

UNIVERSIDADE FEDERAL FLUMINENSE

Robson Araújo Lima

**Uma Proposta de Adaptação da Camada de
Aplicação de Dispositivos Pessoais de Saúde em
Redes Corporais Sem Fio**

NITERÓI

2019

UNIVERSIDADE FEDERAL FLUMINENSE

Robson Araújo Lima

**Uma Proposta de Adaptação da Camada de
Aplicação de Dispositivos Pessoais de Saúde em
Redes Corporais Sem Fio**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Sistemas de Computação

Orientadora:

Débora Christina Muchaluat Saade

Co-orientador:

Célio Vinicius Neves de Albuquerque

NITERÓI

2019

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

L732p Lima, Robson Araújo
Uma Proposta de Adaptação da Camada de Aplicação de Dispositivos Pessoais de Saúde em Redes Corporais Sem Fio / Robson Araújo Lima ; Débora Christina Muchaluat Saade, orientadora ; Célio Vinicius Neves De Albuquerque, coorientador. Niterói, 2019.
76 p. : il.

Dissertação (mestrado)-Universidade Federal Fluminense, Niterói, 2019.

DOI: <http://dx.doi.org/10.22409/PGC.2019.m.05234386139>

1. Redes Corporais Sem Fio. 2. Personal Health Device. 3. X73-PHD. 4. Castalia. 5. Produção intelectual. I. Muchaluat Saade, Débora Christina, orientadora. II. De Albuquerque, Célio Vinicius Neves, coorientador. III. Universidade Federal Fluminense. Instituto de Computação. IV. Título.

CDD -

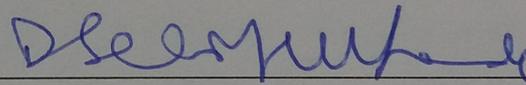
Robson Araújo Lima

Uma Proposta de Adaptação da Camada de Aplicação de Dispositivos Pessoais de Saúde em Redes Corporais Sem Fio

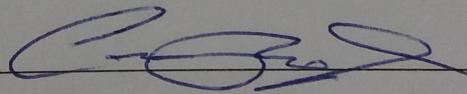
Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Sistemas de Computação

Aprovada em Setembro de 2019.

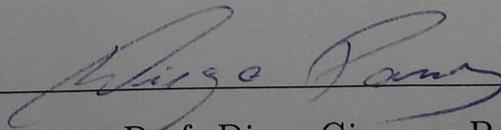
BANCA EXAMINADORA



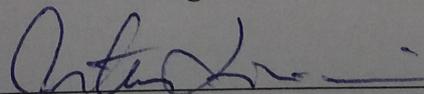
Prof. Débora Christina Muchaluat Saade - Orientadora,
UFF



Prof. Célio Vinicius Neves de Albuquerque - Coorientador,
UFF



Prof. Diego Gimenez Passos, UFF



Prof. Artur Ziviani, LNCC

*Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.*

Lauro César Araújo

Agradecimentos

Os agradecimentos principais são direcionados aos meu pais, Claudete de Araújo Lima e Carlos Antônio de Lima, os quais, proporcionaram minha permanência no programa. Também, à Professora Débora Christina Muchaluat Saade e ao Professor Célio Vinicius Neves de Albuquerque, os quais deram início ao tema e não mediram esforços para contribuir com o trabalho. E por fim, aos colegas do grupo de pesquisa, Vinicius C. Ferreira e Egberto Caballero.

Agradecimentos especiais são direcionados à Universidade Federal Fluminense, principalmente ao MídiaCom, que me acolheu de braços abertos. As instituições de fomento à pesquisa FAPESP, INCT-MACC, FAPERJ, CAPES e CNPq, pelos recursos financeiros recebidos.

Agradeço a todos os meus amigos, minha segunda família, por todos os momentos juntos e a Deus, por todas as oportunidades que tive.

Resumo

Simular novos protocolos para sistemas de saúde digital (*e-health*) é muito importante pois permite uma avaliação inicial antes que uma implementação real seja feita. Por outro lado, os simuladores atuais não oferecem suporte adequado para representar aplicações ou componentes médicos nas simulações de saúde digital. Com o objetivo de preencher essa lacuna, este trabalho propõe o uso da norma ISO/IEEE 11073 *Personal Health Devices* (X73-PHD) em simulações de redes de *e-health*, representando aplicações médicas realistas e investigando o comportamento de dispositivos médicos (sensores ou atuadores) em cenários onde são usadas Redes Corporais Sem Fio (WBAN - *Wireless Body Area Networks*). Sendo assim, esta dissertação propõe uma adaptação da camada de aplicação, seguindo as especificações do padrão X73-PHD, para o Simulador Castalia. Esta camada de aplicação fornece cinco tipos diferentes de Dispositivos Pessoais de Saúde (PHD - *Personal Health Devices*) para atuar como dispositivos reais ISO/IEEE 11073 em simulações de WBAN. A implementação realizada suporta o modo *agent-initiated*, onde os PHDs têm a iniciativa de enviar medições para o *hub*. Também é suportado o modo *manager-initiated*, onde o *hub* solicita explicitamente medições para os PHDs. A implementação provê suporte também a dois modos de comunicação, o modo com confirmação e sem confirmação. No primeiro modo, o receptor envia uma confirmação toda vez que recebe um pacote e, no segundo modo, não é enviada nenhuma confirmação de recebimento. Os resultados de simulação mostraram que o modo com confirmação não apresentou bom desempenho em cenários WBANs quando o intervalo entre as transmissões é muito pequeno, devido a longos *timeouts* propostos na norma ISO/IEEE 11073. Portanto, esta dissertação propõe um novo modo de comunicação chamado Modo de Retransmissão, que diminui a sobrecarga de pacotes de controle na rede, usando *timeouts* menores e entregando mais pacotes no destino.

Palavras-chave: Redes Corporais Sem Fio, Personal Health Device, X73-PHD, Castalia, Antidote.

Abstract

Simulating new protocols for e-health systems is very important, as it allows an initial evaluation before a real implementation is made. On the other hand, network simulators do not offer proper support to represent medical applications or components to facilitate running simulations modeling e-health applications. Aiming at fulfilling this gap, this work proposes the use of the ISO/IEEE 11073 standard for Personal Health Devices (X73-PHD) in e-health network simulations, representing realistic medical applications and investigating the behavior of medical devices (sensors or actuators) in Wireless Body Area Network (WBAN) scenarios. We developed a free and open-source implementation of X73-PHD for the Castalia Simulator, providing five different PHD types to act as real ISO/IEEE 11073 devices in WBAN simulations. Our implementation supports Agent-initiated mode, where PHDs take the initiative to send measurements to the hub, and also Manager-initiated mode, where the hub requires measurements to PHDs. Our implementation also supports both unconfirmed and confirmed communication modes, where the receiver sends an acknowledgment to the sender every time it receives a packet. Simulation results showed that the confirmed communication mode did not perform well in WBANs when the interval between transmissions is too small, due to the long timeout period proposed in the ISO/IEEE 11073 standard. Therefore, we propose a new extension to the confirmed mode standard, named retransmission mode, that decreases the overhead of control packets over the network, using smaller timeouts and delivering more packets to the destination.

Keywords: . Wireless Body Area Network, Personal Health Device, X73-PHD, Castalia, Antidote.

Lista de Figuras

| | | |
|------|--|----|
| 2.1 | Tipos de Sensores WBAN [7]. | 7 |
| 2.2 | Cobertura dos padrões. | 12 |
| 2.3 | Termômetro - Modelo de Domínio de Informação. | 14 |
| 2.4 | Modelo de Domínio de Informação para dispositivos pessoais de saúde. . . | 16 |
| 2.5 | Uma <i>PM-Store</i> com dois <i>Segments</i> | 23 |
| 2.6 | Diagrama da máquina de estados de um agente. Fonte: [23]. | 27 |
| 2.7 | Diagrama de sequência da associação de um agente com um gerente (adaptado de [23]). | 29 |
| 2.8 | Transmissão de dados utilizando o modo <i>Agent-initiated</i> (adaptado de [23]). | 30 |
| 2.9 | <i>Manager-initiated</i> utilizando <i>time period mode</i> (adaptado de [23]). | 32 |
| 2.10 | <i>Manager-initiated</i> utilizando <i>no limit time mode</i> (adaptado de [23]). | 33 |
| 4.1 | Diagrama de sequência para os modos de operações com e sem confirmação de uma aplicação X73-PHD. | 43 |
| 4.2 | Diagrama de sequência do modo de retransmissão demonstrando o comportamento quando uma mensagem ACK e uma leitura são perdidas. . . . | 45 |
| 4.3 | Diagrama de sequência de operação de uma aplicação <i>manager-initiated</i> . . | 45 |
| 4.4 | Arquitetura em alto nível da proposta. | 46 |
| 4.5 | Organização interna do simulador Castalia, fonte: [5] | 47 |
| 4.6 | Síntese da organização interna da Biblioteca Antidote (Adaptado de [38]). . | 49 |
| 5.1 | Diagrama de como é realizado a comunicação entre a Biblioteca Antidote e o Simulador Castalia, utilizando um plug-in de comunicação. | 54 |
| 5.2 | Módulo de comunicação manipulando vários nós. Cada seta representa uma função sendo chamada do módulo de comunicação. | 55 |

| | | |
|------|---|----|
| 6.1 | A topologia utilizada na simulação. | 57 |
| 6.2 | Pacotes de leituras entregues com sucesso no gerente usando <i>Agent-initiated mode</i> | 60 |
| 6.3 | Número médio de pacotes de controle enviados e recebidos utilizando <i>Agent-initiated mode</i> | 61 |
| 6.4 | Média de associações feitas por cada nó. | 62 |
| 6.5 | Média de retransmissões de pacotes feita na camada de aplicação utilizando <i>retries=4, timeout=0.4</i> | 62 |
| 6.6 | Latência na camada de aplicação. | 63 |
| 6.7 | Pacotes de leitura entregues com sucesso por nó usando <i>Manager-initiated mode</i> | 64 |
| 6.8 | Número médio de associações feitas por cada dispositivo durante cada simulação. | 65 |
| 6.9 | Número médio de pacotes de controle enviados/recebidos por cada agente. | 65 |
| 6.10 | Latência da camada de aplicação para os três modos <i>manager-initiated</i> | 66 |
| 6.11 | Pacotes de leituras entregues com sucesso no gerente usando o Modo de Retransmissão com <i>timeOutToRetransmitPacket</i> calculado dinamicamente. | 67 |
| 6.12 | Número médio de associações feitas por cada nó utilizando o Modo de Retransmissão com <i>timeOutToRetransmitPacket</i> calculado dinamicamente. | 68 |
| 6.13 | Pacotes de controle enviados/recebidos utilizando o Modo de Retransmissão com <i>timeOutToRetransmitPacket</i> calculado dinamicamente. | 69 |
| 6.14 | Número de retransmissões de pacotes de leituras utilizando o Modo de Retransmissão com <i>timeOutToRetransmitPacket</i> calculado dinamicamente. | 69 |
| 6.15 | Média dos valores de <i>timeOutToRetransmitPacket</i> calculados dinamicamente para cada um dos agentes. | 70 |

Lista de Tabelas

| | | |
|-----|--|----|
| 2.1 | Atributos da classe <i>Medical Device System</i> (MDS). | 17 |
| 2.2 | Atributos da classe <i>Metric</i> | 18 |
| 2.3 | Atributos da classe <i>Numeric</i> | 19 |
| 2.4 | Atributos da classe <i>RT-SA</i> | 20 |
| 2.5 | Atributos da classe <i>Enumeration</i> | 21 |
| 2.6 | Atributos da classe <i>PM-Store</i> | 21 |
| 2.7 | Atributos da classe <i>PM-Segment</i> | 22 |
| 2.8 | Atributos da classe <i>Scanner</i> | 22 |
| 4.1 | Parâmetros configuráveis de simulação. | 40 |
| 6.1 | Número de pacotes de leitura entregues no gerente variando os parâmetros de <i>timeOutToRetransmitPacket</i> e <i>maxNumOfRetransmission</i> | 59 |

Lista de Abreviaturas e Siglas

| | | |
|---------------|--|------|
| ACK | <i>Acknowledgement</i> | 2 |
| APDU | <i>Application Protocol Data Unit</i> | 15 |
| API | <i>Application Programming Interface</i> | 48 |
| ASN.1 | <i>Abstract Syntax Notation One</i> | 13 |
| BER | <i>Basic Encoding Rules</i> | 15 |
| CDG | <i>Continua Design Guidelines</i> | 13 |
| CoAP | <i>Constrained Application Protocol</i> | 36 |
| DIM | <i>Domain Information Model</i> | 4 |
| DPS | <i>Dispositivo Pessoal de Saúde</i> | 1 |
| ECG | <i>Eletrocardiograma</i> | 6 |
| FCC | <i>Federal Communications Commission</i> | 7 |
| HL7 | <i>Health Level Seven</i> | 12 |
| ICNIRP | <i>International Commission on Non-Ionizing Radiation Protection</i> | 9 |
| IHE | <i>Integrating the Healthcare Enterprise</i> | 12 |
| IoT | <i>Internet of Things</i> | 4 |
| ISM | <i>Industrial, Scientific and Medical</i> | 6 |
| M2M | <i>Machine-to-Machine</i> | 36 |
| MAC | <i>Medium Access Control</i> | 3 |
| MBAN | <i>Medical Body Area Network</i> | 8 |
| MDER | <i>Medical Data Encoding Rules</i> | 15 |
| MDS | <i>Medical Device System</i> | viii |
| MQTT | <i>Message Queuing Telemetry Transport</i> | 36 |
| OEP | <i>Optimized Exchange Protocol</i> | 13 |
| OSI | <i>Open System Interconnection</i> | 15 |

| | | |
|--------------------|---|----|
| PCHAlliance | <i>Personal Connected Health Alliance</i> | 12 |
| PHD | <i>Personal Health Device</i> | 1 |
| PM-Segment | <i>Persistent Metric Segment</i> | 20 |
| PM-Store | <i>Persistent Metric Store</i> | 21 |
| RF | <i>Radio Frequency</i> | 8 |
| RTT | <i>Round-trip Time</i> | 66 |
| RSSF | <i>Redes de Sensores Sem Fio</i> | 1 |
| RT-SA | <i>Real-Time Sample Array</i> | 16 |
| SAR | <i>Specific Absorption Rate</i> | 9 |
| TCP | <i>Transmission Control Protocol</i> | 66 |
| X73-PHD | <i>11073 Personal Health Device</i> | 2 |
| X73-PoC | <i>11073 Point of Care Device</i> | 5 |
| XER | <i>XML Encoding Rules</i> | 15 |
| WBAN | <i>Wireless Body Area Network</i> | xi |

Sumário

| | | |
|----------|--|----------|
| 1 | Introdução | 1 |
| 1.1 | Objetivos | 3 |
| 1.2 | Organização do Texto | 4 |
| 2 | Fundamentação Teórica | 5 |
| 2.1 | Redes Corporais Sem Fio | 6 |
| 2.1.1 | Requisitos de comunicação | 8 |
| 2.1.2 | Segurança para o corpo humano | 9 |
| 2.1.3 | Métricas para <i>Wireless Body Area Network</i> (WBAN) | 10 |
| 2.2 | Padrão X73-PHD | 11 |
| 2.3 | Perfis de especialização | 14 |
| 2.4 | Optimized Exchange Protocol - IEEE 11073-20601 | 15 |
| 2.5 | Domain Information Model (DIM) | 15 |
| 2.6 | Modelo de Serviço | 24 |
| 2.6.1 | Serviços de Acesso | 24 |
| 2.6.2 | Serviço de Associação | 24 |
| 2.6.3 | EVENT REPORT com e sem confirmação | 25 |
| 2.7 | Modelo de comunicação | 26 |
| 2.7.1 | Máquina de estados | 26 |
| 2.7.2 | Associating | 28 |
| 2.7.3 | Operating | 29 |
| 2.8 | Casos de Uso do padrão X73-PHD | 31 |

| | | |
|----------|--|-----------|
| 2.9 | Segurança | 34 |
| 3 | Trabalhos Relacionados | 36 |
| 4 | Proposta | 39 |
| 4.1 | Modos de Comunicação | 42 |
| 4.1.1 | Modo sem Confirmação | 42 |
| 4.1.2 | Modo com Confirmação | 42 |
| 4.1.3 | Proposta do Modo de Retransmissão | 43 |
| 4.1.4 | Modo <i>Manager-initiated</i> | 44 |
| 4.2 | Arquitetura do Sistema | 46 |
| 4.2.1 | Simulador Castalia | 46 |
| 4.2.2 | Biblioteca Antidote | 48 |
| 5 | Implementação | 51 |
| 5.1 | Castalia Plug-in | 51 |
| 5.2 | Mudanças no módulo de comunicação | 54 |
| 6 | Avaliação | 56 |
| 6.1 | Cenário de simulação | 56 |
| 6.2 | Resultados | 58 |
| 6.2.1 | Agent-initiated | 59 |
| 6.2.2 | Manager-initiated | 63 |
| 6.2.3 | Modo de Retransmissão com <i>Timeout</i> Calculado Dinamicamente | 66 |
| 7 | Conclusão | 71 |
| 7.1 | Trabalhos Futuros | 72 |
| | Referências | 73 |

Capítulo 1

Introdução

Redes de Sensores Sem Fios (RSSFs) têm se tornado um ramo de estudo importante para inúmeros cenários de aplicação e simulação (Internet das coisas, cidades inteligentes, informática médica, etc). Devido ao aumento de pesquisas na área de RSSF, juntamente com telemedicina, um novo tipo de rede surgiu: as Redes Corporais Sem Fio ou WBANs [33]. Uma WBAN é formada por dispositivos inteligentes colocados sobre pele ou implantados no corpo, que são capazes de trocar dados através de uma rede sem fio [7].

Com o grande avanço de tecnologias para o monitoramento de pacientes e o crescente interesse por tecnologias digitais aplicadas à saúde, desencadeou-se o desenvolvimento de *Dispositivo Pessoal de Saúdes* (DPSs) (ou *Personal Health Device* (PHD), em inglês) com interfaces de comunicação embutidas, tais como USB, *Bluetooth* e *ZigBee*. Em cenários que fazem uso de DPS, o paciente também é responsável pelo monitoramento de sua própria saúde e bem-estar [32].

Um dispositivo médico típico é geralmente utilizado em hospitais e manuseado por especialistas. Os dados gerados por esses dispositivos são utilizados para realizar diagnósticos e para tratamentos. Dispositivos similares são muitos comuns no dia-a-dia com propósitos de melhorar a saúde, bem-estar, estado físico, etc. Esses aparelhos coletam dados que podem se tornar úteis se compartilhados e reutilizados de forma inovadora. Por exemplo, uma balança pode enviar leituras para um *personal trainer* específico, ou um médico pode receber dados de leitura de um monitor de pressão sanguínea de seu paciente. Antes que os dados cheguem em seu destino final (para um especialista da saúde por exemplo), eles devem passar por vários outros processos e transformações.

A ausência de dispositivos pessoais de saúde no mercado e os riscos que testes práticos podem oferecer à saúde tornam experimentos reais com WBANs complexos [28]. Com o

avanço e aperfeiçoamento de RSSF, simuladores se tornaram uma importante ferramenta que permite viabilizar testes com menor custo e tempo. O Castalia [5] é um simulador de redes de sensores sem fio, sensores corporais e dispositivos embutidos de baixa potência. Ele é um software livre e de código aberto, desenvolvido para que os usuários testem seus próprios algoritmos e protocolos.

O Castalia fornece uma estrutura genérica, confiável e realista para a primeira validação de um algoritmo, antes de passar para uma plataforma de sensor específica. Cada sensor é representado por um nó genérico, que executa funções de transmissão e recepção de dados. Entretanto, as aplicações disponíveis para testes de novos protocolos no simulador são genéricas, isto é, não especificam o tipo de sensor nem seus requisitos e procedimentos de comunicação na camada de aplicação. Isso representa um ponto fraco para o uso do simulador para aplicações de saúde digital, já que os protocolos de aplicação padronizados não são considerados nas simulações.

Para tornar as simulações mais realistas, o uso de aplicações reais e padronizadas é um ponto vital. O padrão *11073 Personal Health Device* (X73-PHD) [21] descreve aplicações e informações para dispositivos de telessaúde. O padrão também especifica a troca e a representação de dados para comunicação entre dispositivos pessoais de saúde. Desta forma, o principal objetivo da norma X73-PHD é prover um protocolo de comunicação de dados que torne possível implementar DPS com pouca capacidade de processamento e memória.

O padrão X73-PHD define dois tipos de dispositivos: *agentes* e *gerentes*. Os agentes são tipicamente sensores ou atuadores de baixa potência e com pouco poder de processamento, enquanto os gerentes são dispositivos com um poder de processamento maior, que podem ou não estar conectados a uma fonte de energia. Apesar do padrão X73-PHD não comentar explicitamente, dispositivos agentes e gerentes podem ser nós de uma WBAN.

Existem dois modos de comunicação no padrão X73-PHD, o modo com confirmação e sem confirmação. No modo sem confirmação, as leituras são transmitidas sem nenhum tipo de confirmação de recebimento por parte do gerente. No modo com confirmação, para cada pacote de leitura enviado pelos agentes, espera-se que um *Acknowledgement* (ACK) seja transmitido de volta, reconhecendo que o pacote foi entregue no gerente. De acordo com o padrão, caso o agente não receba o ACK em um período de 3s, uma nova associação com o gerente deve ser feita para finalizar a transmissão de leituras.

Este último modo de comunicação pode aumentar consideravelmente o número de pacotes de controle trocados entre agente e gerente em um cenário WBAN, quando um

ACK é perdido. Portanto, há espaço para propostas de novos modos de comunicação de forma a tornar o uso do padrão X73-PHD em WBANs mais eficiente.

1.1 Objetivos

Esta dissertação propõe o uso do padrão X73-PHD para simulação de aplicações de saúde digital em cenários de WBAN. Desta forma, simulações de novos protocolos em WBANs podem se tornar mais realistas com o uso de uma camada de aplicação baseada no padrão X73-PHD.

O objetivo principal deste trabalho, portanto, é propor uma camada de aplicações médicas para o simulador Castalia, que simula um dispositivo médico real (sensor ou atuador) e analisar o comportamento desses dispositivos em redes WBAN. Alguns exemplos de dispositivos agentes simulados em redes WBAN são: termômetro, monitor de pressão arterial, medidor de glicose e etc. Todas as extensões são desenvolvidas de acordo com a padronização de comunicação de dispositivos pessoais de saúde definidos pela norma X73-PHD. O simulador Castalia foi escolhido para simulações de WBAN, pois possui a camada *Medium Access Control* (MAC) padrão IEEE 802.15.6 de WBAN já implementada.

Devido a limitações de recursos, incluindo bateria, muitos desses dispositivos podem não ter uma camada de transporte confiável. Tendo isso em vista e considerando que os modos de comunicação propostos no padrão X73-PHD não obtêm um bom desempenho em cenários WBANs, esta dissertação propõe a retransmissão de pacotes na camada de aplicação, a qual não está presente atualmente no padrão. Portanto, como segundo objetivo, este trabalho propõe um novo modo de comunicação, chamado *modo de retransmissão* para o padrão X73-PHD. Este modo é baseado em *stop-and-wait*, fazendo a retransmissão de pacotes que foram perdidos, ao invés de solicitar imediatamente uma nova associação após a perda de um pacote, como é feito no modo com confirmação do padrão X73-PHD.

Este trabalho faz a união entre um padrão criado para comunicação de WBANs (IEEE 802.15.6) e um padrão de camada de aplicação criado para dispositivos pessoais de saúde X73-PHD. Em relação ao padrão X73-PHD, é ainda proposto nesta dissertação um novo modo de comunicação que melhora o desempenho da comunicação quando os dois padrões são usados em conjunto.

1.2 Organização do Texto

O restante do texto é organizado da seguinte forma. No Capítulo 2 é apresentada a fundamentação teórica. É feita primeiramente uma apresentação geral sobre WBANs, apresentando suas características físicas e técnicas, limitações, casos de uso, exemplos de nós e requisitos de comunicação. Logo após, é explicado como o padrão X73-PHD está estruturado, apresentando os perfis de especialização, os componentes do *Domain Information Model* (DIM), suas relações considerando orientação a objetos e o que cada um de seus componentes significa. Por fim, é discutida sua máquina de estados, assim como as condições de entrada e saída em cada um dos estados.

Os trabalhos relacionados são abordados no Capítulo 3. Propostas de uso de protocolos de *Internet of Things* (IoT) para comunicação de dispositivos de saúde, adaptadores para dispositivos de saúde legados, controle remoto de gerentes 11073 compatíveis e modelagens realistas do consumo de energia no Castalia são alguns dos trabalhos discutidos.

A proposta desta dissertação de mestrado é apresentada no Capítulo 4. A arquitetura da solução e suas relações com a biblioteca Antidote, usada na implementação, são discutidas. Por fim, uma visão geral do funcionamento e divisão dos módulos do simulador Castalia e da biblioteca Antidote é apresentada.

No Capítulo 5, é detalhado como foi desenvolvido o *plug-in* que permite a comunicação entre a biblioteca Antidote e o simulador. São abordadas também as mudanças que a biblioteca Antidote sofreu para se adaptar ao simulador e ao cenário de WBAN.

No Capítulo 6, são apresentados os cenários de simulação que avaliam a proposta desta dissertação, discutindo os resultados obtidos, que confirmam as melhorias obtidas com o modo de retransmissão proposto em WBANs.

Por fim, conclusões e considerações finais são feitas no Capítulo 7. Um resumo das análises dos resultados obtidos e trabalhos futuros também são discutidos nesse capítulo.

Capítulo 2

Fundamentação Teórica

Devido ao surgimento em massa dos PHDs no mercado, um grupo de trabalho do IEEE definiu a família de normas denominadas ISO/IEEE 11073 [16]. Esta família de normas começou a ser especificada na década de 1990 e tinha como finalidade conectar dispositivos médicos em unidades de saúde. Monitores de sinais vitais e monitores de pressão arterial são alguns exemplos desses dispositivos. Seu uso era feito principalmente por profissionais da saúde (médicos, enfermeiras, etc). Contudo, dispositivos de saúde e equipamentos para exercícios físicos (*fitness*) alcançaram o mercado e seu uso doméstico vem crescendo a cada ano. O termo PHD envolve tanto dispositivos médicos, quanto dispositivos de saúde e *fitness* usados por usuários “leigos” em suas residências [16], [21]. É comum a venda desses dispositivos juntamente com produtos eletrônicos de consumo geral.

A família de padrões ISO/IEEE 11073 é dividida em três grupos a primeira e mais antiga parte é a ISO/IEEE 11073 *Lower Layer*, que especifica protocolos e serviços de comunicação orientados à conexão, utilizando camadas físicas como infravermelho, tecnologia RF sem fio ou Ethernet [20]. Já a parte ISO/IEEE 11073 *Point of Care Device (X73-PoC)* especifica padrões de comunicação para dispositivos que são usados exclusivamente em unidades de saúde. Por fim, a ISO/IEEE 11073 X73-PHD, ou DPS em português, define padrões para dispositivos usados pelos usuários em suas casas. O foco deste trabalho será apenas no padrão ISO/IEEE 11073 X73-PHD, que especifica dispositivos usados mais frequentemente por usuários comuns, tais como pacientes em suas casas ou atletas que precisam ter sua atividade física monitorada, representando uma parte significativa das aplicações de saúde digital. Por questões de simplificação, este padrão será referenciada apenas como X73-PHD neste texto.

2.1 Redes Corporais Sem Fio

Redes corporais sem fio, ou WBAN, são redes compostas de dispositivos sensores ou atuadores que possuem como principal característica o baixo consumo energético, tamanho e capacidade de processamento limitados. Estes dispositivos podem ser implantados sobre, dentro ou nos arredores do corpo humano. Os sensores monitoram funções do corpo e do ambiente, com o objetivo de coletar e transferir dados biofísicos. Os atuadores recebem comandos para atuar conforme a necessidade. Além das aplicações médicas, outros campos como entretenimento, esportes, jogos, aplicações militares podem se beneficiar desta nova tecnologia.

Tendo em vista que WBANs são comumente utilizadas na banda *Industrial, Scientific and Medical* (ISM) de 2.4GHz, a concorrência do canal por outras tecnologias de comunicação sem fio deve ser considerada para que não haja colisões ou ruídos. Uma das principais questões envolvidas acerca da implementação de WBANs é o superaquecimento dos dispositivos, o que pode resultar em desconforto e problemas sérios de saúde [7]. Mecanismos sofisticados para economia de energia devem ser implementados.

WBANs são capazes de coletar amostras, monitorar, processar e comunicar diferentes sinais vitais e parâmetros fisiológicos ao médico responsável, provendo informações em tempo real, sem interferir na vida cotidiana de um paciente. Com amostras de dados devidamente armazenadas, é possível ter conhecimento do estado atual do paciente, realizar diagnósticos precoces e inferir se no futuro, este paciente poderá desenvolver algum tipo de doença crônica.

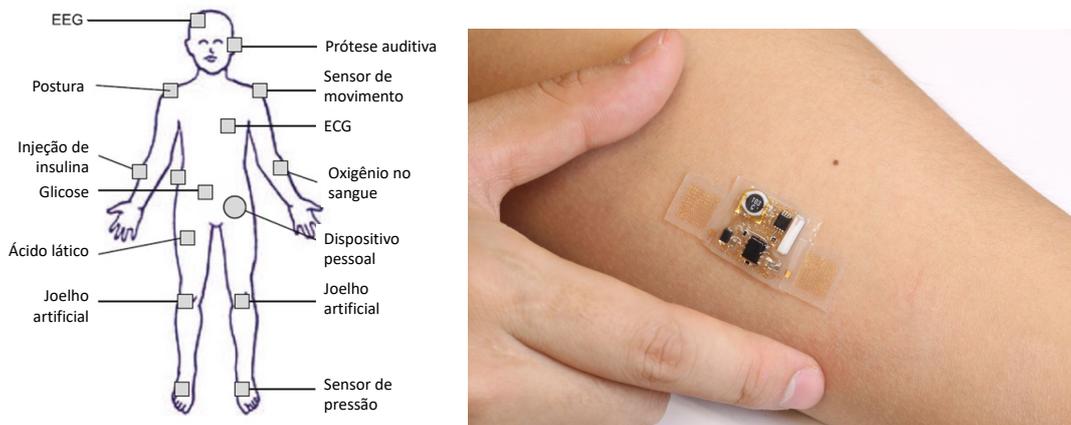
Espera-se que as redes WBANs possam interagir com outras tecnologias de comunicação como *ZigBee*, *Bluetooth*, *Wi-Fi*, redes celulares e etc. Essa interação pode abrir possibilidades para a criação de novas aplicações, serviços e dispositivos que trarão maior qualidade de vida. Estas novas aplicações devem seguir o padrão X73-PHD, que normaliza a comunicação entre dispositivos médicos de uso doméstico, isto é, dispositivos que podem ser administrados pelos próprios usuário em suas casas.

Os principais componentes destes dispositivos são o transceptor de rádio para comunicação sem fio, microprocessador, memória, sensores bioquímicos ou analógicos, como por exemplo, *Eletrocardiograma* (ECG), medidor de glicose e termômetro. Geralmente, o sistema operacional utilizado exige pouco poder de processamento e pouca memória.

Dispositivos médicos podem ser colocados no corpo humano por meio de adesivos, implantes ou anexados na roupa. Atualmente, já existem alguns desses dispositivos como:

- *Comprimidos* que contêm transceptores sem fio que podem detectar enzimas, ácidos nucleicos, acidez intestinal, pressão e outros fatores fisiológicos de forma não invasiva [11].
- *Dispositivos vestíveis* portáteis que utilizam transceptores sem fio montados na superfície do corpo como, por exemplo, *smartwatches* ou cintos com acelerômetros.
- *Dispositivos fisiológicos implantados* são dispositivos que podem ser implantados dentro do corpo, como marcapasso ou atuadores para injeção de insulina.

Nas Figuras 2.1(a) e 2.1(b) é possível ver alguns tipos de sensores posicionados no corpo humano [26] e um sensor desenvolvido pela Universidade de Illinois [41].



(a) Alguns sensores posicionados no corpo humano. (b) Sensor médico desenvolvido pela Universidade de Illinois.

Figura 2.1: Tipos de Sensores WBAN [7].

A comunicação de WBANs podem ser classificadas em três classes da seguinte forma:

- *Classe 1*: Comunicação intra-WBAN, acontece apenas entre nós da mesma WBAN.
- *Classe 2*: Comunicação inter-WBAN, isto é, comunicação é feita entre redes WBAN.
- *Classe 3*: Comunicação além-WBAN, a comunicação é realizada entre a WBAN e redes externas, como *Wi-Fi*, *bluetooth*, redes celulares e etc.

O escopo deste trabalho está concentrado apenas na classe 1. O restante do trabalho trata sobre a comunicação entre sensores e gerentes e suas tecnologias.

A maioria das tecnologias WBAN operam na banda ISM. Portanto, é importante para evitar problemas de interferência, a coexistência com outras redes sem fio. A *Federal*

Communications Commission (FCC) disponibilizou uma banda de 40 MHz alocada em uma faixa de 2,36 GHz a 2,4 GHz exclusivamente para aplicações médicas, denominada *Medical Body Area Network* (MBAN) [26].

2.1.1 Requisitos de comunicação

A diversidade de dispositivos médicos e tipos de dados impõe vários requisitos de funcionamento e desenvolvimento de WBANs. Alguns desses requisitos são apresentados por [39].

- *Taxa de dados e Qualidade de serviço (QoS)*: a taxa de transmissão de um dispositivo WBAN pode variar de 1 kb/s até 10 Mb/s (por exemplo, streaming de vídeo). Caso a WBAN possua mais de um dispositivo, mecanismos de Qualidade de Serviço devem garantir a priorização de tráfego. Outros parâmetros importantes de transmissão são latência e o *jitter*. É esperado que em situações de emergência uma WBAN forneça uma resposta rápida e confiável.
- *Alcance*: a distância entre nós de uma WBAN não deve exceder 6 metros, utilizando a topologia em estrela, visto que o corpo humano já representa um obstáculo natural à propagação de *Radio Frequency* (RF). Uma WBAN geralmente implementa um mecanismo confiável de associação, permitindo que um novo nó seja agregado ou separado de outros em um grupo de nós preexistente.
- *Segurança*: A segurança de dados é um requisito primário para WBANs. A segurança, como em qualquer dispositivo médico, é tratada em termos de privacidade, confidencialidade, métodos de autorização e integridade de dados. Infelizmente, métodos de atuais de criptografia de dados não são apropriados para o cenário de WBAN, uma vez que os nós possuem recursos limitados de processamento, memória e consumo de energia.
- *Antena e canal de rádio*: Dimensões quase minúsculas levam um projeto de antena altamente eficiente. Além disso, o corpo afeta o perfil da radiação da antena, o que requer um canal de rádio adequado para este cenário em particular.
- *Consumo de energia*: Geralmente são dispositivos móveis que usam pequenas baterias, e o consumo varia de acordo com a aplicação. Espera-se que o tempo de vida da bateria seja o maior possível, principalmente para dispositivos implantados. Me-

canismos de *duty-cycle* podem ser utilizados para este fim. Existem também estudos que consideram o calor [13] e o movimento corporal [40] como fonte de energia.

- *Coesistência*: Como citado anteriormente, dispositivos médicos utilizam a banda ISM, a qual é utilizada pela maioria dos dispositivos sem fio, inclusive *Wi-Fi* (IEEE 802.11), *Bluetooth* (IEEE 802.15.1), *ZigBee* (IEEE 802.15.4). O uso de várias redes sem fio na mesma faixa de frequência gera interferência e o aumento na taxa de perda de pacotes. Isso se torna mais preocupante durante eventos de alto risco, como em casos de ataque cardíaco, derrames, demaios e etc.
- *Projeto do Hardware*: O ponto mais crítico se concentra no tamanho do dispositivo. O desafio é incorporar uma bateria e uma antena em um dispositivo quase minúsculo. Alguns dispositivos são desenvolvidos para serem acoplados na roupa, oferecendo menos incômodo e, relaxando a restrição de tamanho. Atividades militares e esportivas já usam este tipo de dispositivo.
- *Processamento de sinais*: O consumo de energia é um dos pontos mais críticos na implementação de WBAN. Além de prejudicar a saúde em casos de superaquecimento, nem sempre é possível trocar a bateria de um dispositivo médico. No entanto, técnicas de processamento de sinais mais eficientes podem melhorar o controle sobre o consumo de energia relacionado à aquisição de sinais biológicos.

2.1.2 Segurança para o corpo humano

Radiação é o processo pelo qual a energia é emitida como partículas ou ondas. A radiação pode ser classificada como ionizantes e não ionizantes. A radiação ionizante, que tem uma frequência mais alta do que a radiação não ionizante, representa uma ameaça à saúde. A maioria das radiações não ionizantes, como as ondas de rádio ou micro-ondas é considerada prejudicial à saúde para uma determinada quantidade de energia térmica. Organizações governamentais e não-governamentais definem alguns limites de exposição relacionados à radiofrequência ou campos eletromagnéticos de forma geral. A *International Commission on Non-Ionizing Radiation Protection* (ICNIRP) define restrições de tempo de exposição do corpo humano a campos eletromagnéticos não ionizantes. Tais restrições são tipicamente definidas em termos da *Specific Absorption Rate* (SAR). A SAR é definida como a taxa de absorção de energia eletromagnética por tecido corporal específico, medida por seis minutos e sua unidade de medida é W/kg . Considerando todo o corpo humano, a taxa SAR estimada pelos dispositivos WBAN é insignificante [7].

A transmissão de dados por um longo período de tempo, de dispositivos implantados ou sobre a pele, pode aumentar a temperatura do tecido local e causar efeitos indesejados no corpo humano [34]. Os principais efeitos térmicos no corpo são:

- Redução do fluxo sanguíneo;
- Danos térmicos aos órgãos mais sensíveis;
- Alguns tipos de crescimento bacteriano;
- Efeitos sobre os relacionamentos enzimáticos.

2.1.3 Métricas para WBAN

Algumas métricas para avaliar o desempenho de WBANs são definidas em [8]. Algumas são métricas já comumente utilizadas em redes como taxa de entrega, taxa de perda de pacotes, atraso fim-a-fim e vazão.

1. *Vida útil da rede*: Define o tempo máximo de operação dos nós até o último nó ficar inativo;
2. *Energia residual*: É a diferença entre a energia inicial e a energia utilizada durante a operação da rede;
3. *Número de nós ativos e inativos*: É o número de nós que depois de um tempo predefinido de operação de rede, ainda têm energia ou consumiram toda sua energia, respectivamente;
4. *Período de estabilidade*: É o tempo antes do primeiro nó ficar inativo.
5. *Path loss*: É a diferença entre a potência do sinal transmitido pelo nó transmissor e a potência recebida no nó receptor.
6. *Atraso fim-a-fim*: É o tempo tomado por um pacote de dados para alcançar o nó gerente (*sink node*) da WBAN desde o nó origem.
7. *Taxa de entrega de pacotes*: Determinada pelo número de pacotes recebidos no *sink node*, dividido pelo número de pacotes enviados desde o nó origem.
8. *Taxa de perda de pacotes*: Determinada pelo número de pacotes perdidos, dividido pelo número de pacotes enviados desde o nó origem.

9. *Aumento da temperatura*: Geralmente é uma métrica utilizada para avaliar protocolos que têm como principal objetivo evitar o aquecimento.
10. *Vazão*: É definida como a taxa de transmissão de dados através de uma rede. Normalmente é medida em bits por segundo.
11. *Pacotes reenviados*: É o número de pacotes que são reenviados em razão de um erro, como a perda de um ACK.

Das métricas apresentadas, as quatro primeiras estão relacionadas à eficiência energética. As métricas 6, 7, 8 estão associadas à qualidade de serviço. A métrica 9 diz respeito à absorção de radiação e aquecimento dos nós. A métrica 10 oferece uma ideia geral de desempenho da rede, pois vários aspectos como: topologia de rede, particionamento topológico, perdas do caminho e limitação de recursos influenciam em seu desempenho. A métrica 11 diz respeito a mecanismos de entrega confiável.

2.2 Padrão X73-PHD

Interoperabilidade entre dispositivos de diferentes fabricantes é alcançada quando todos os sistemas envolvidos implementam um conjunto comum de normas. A família de normas 11073 foi projetada para a comunicação entre qualquer dispositivo médico.

O padrão X73-PHD define dois tipos de dispositivos: *agentes* e *gerentes*. Os dispositivos agentes são frequentemente sensores ou atuadores de baixo custo, possuem capacidades limitadas de hardware, como memória RAM, CPU e pequenas fontes de energia. Por razões de simplicidade, possuem configuração fixa e são desconectados quando inativos. Em contrapartida, os dispositivos gerentes têm mais capacidades de hardware e podem se conectar a vários agentes. Possuem fontes de energia fixa e CPU com maior poder de processamento. *Smartphones* e *desktops* são alguns exemplos de gerentes.

Tecnologias utilizadas para a transmissão de dados de medições biofísicas, entre agentes e gerentes, são divididas em dois grandes grupos: tecnologias de comunicação cabeadas e sem fio. *Wireless Personal Area Network* (WPAN) é suportada por vários PHDs presentes no mercado e está largamente disponível em *laptops* e *smartphones*. Uma WPAN é frequentemente utilizada para transferir dados de dispositivos para um *smartphone* em curtas distâncias, que posteriormente, transmite os dados para um centro especializado por meio de uma rede WAN. *Cabled PAN bus* é uma tecnologia amplamente utilizada para conectar dispositivos eletrônicos como *laptops* e *desktops* [16]. É utilizada também

para o carregamento da baterias de eletrônicos e *smartphones*. Essa tecnologia é usada com PHDs quando o uso de uma tecnologia sem fio não é viável por motivos técnicos ou por razões de usabilidade. Dispositivos pequenos e de baixa potência costumam utilizar tecnologias WPAN ou WLAN para curtas distância, por exemplo, dentro de uma quarto ou prédio.

Os dados médicos que estão em um *smartphone* ou *desktop*, precisam ser transferidos para um centro ou profissional especializados para serem analisados. Os dados gerados por agentes são transmitidos, primeiramente, para os gerentes, que, posteriormente enviam os dados para processamento em centros de telessaúde, profissionais da saúde, amigos, parentes, *personal trainers*, entre outros. O padrão X73-PHD, portanto, cobre apenas a primeira fase dessa jornada, ou seja, a transmissão dos dados entre os agentes e gerentes.

Portanto, o envio desses dados pela Internet não é coberto pelo padrão X73-PHD. Entidades como *Health Level Seven (HL7)* e *Integrating the Healthcare Enterprise (IHE)* incentivam e promovem o uso de normas internacionais para representação de dados médicos, especificando, inclusive como devem ser transmitidos pela Internet. Para garantir o uso dos padrões já existentes e evitar duplicatas, a *Personal Connected Health Alliance (PCHAlliance)* é uma organização que implementa um *framework* geral de todas as normas relacionadas a dados médicos já criadas e especifica como esses padrões devem ser integrados e utilizados em conjunto. Na Figura 2.2 é possível ver que o padrão X73-PHD é responsável apenas pela comunicação local entre agentes e gerentes.

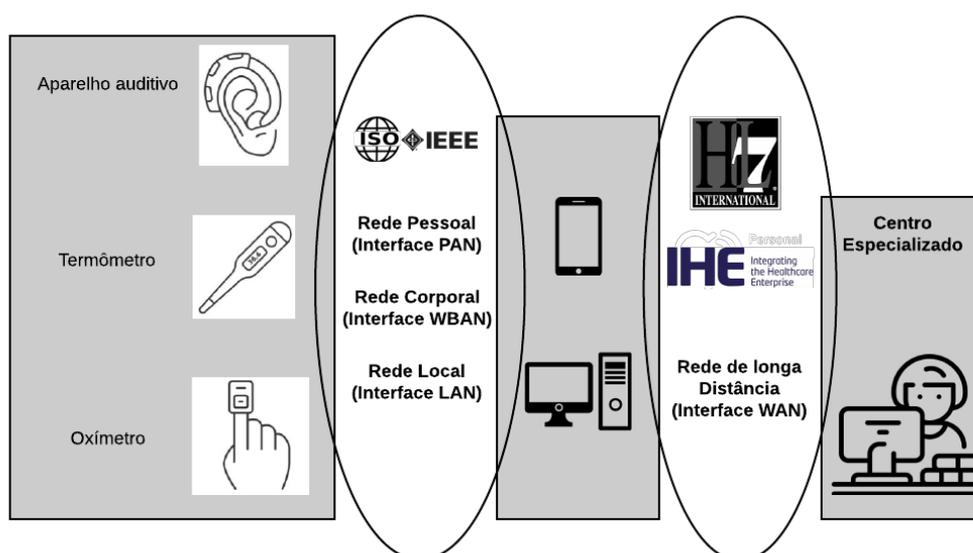


Figura 2.2: Cobertura dos padrões.

A fim de oferecer um conjunto completo de padrões que cubra não apenas a comunicação entre os agentes e os gerentes, mas sim todo o caminho que os dados fazem até chegarem ao seu destino final, a PCHAlliance publica e promove o *Continua Design Guidelines* (CDG), um framework para conexão fim-a-fim de DPS, usando padrões abertos, para criar uma conexão segura e interoperável entre esses dispositivos.

O CDG oferece um conjunto claro de interfaces que permite o fluxo seguro de dados entre os sensores, *gateways* e centros de telessaúde. Além disso, esse guia remove ambiguidade dos padrões para assegurar um ecossistema consistente e interoperável para DPS [17].

A norma X73-PHD descreve a estrutura dos dados dos agentes e gerentes utilizando o conceito de *Domain Information Model* (DIM). O DIM descreve as partes e mostra como essas partes devem ser reunidas para formar um elemento maior. Nas implementações, essas partes são geralmente chamadas de “classes” e “objetos”. A linguagem *Abstract Syntax Notation One* (ASN.1) descreve o formato dos dados contidos nos objetos do DIM. ASN.1 é uma notação formal usada para descrever dados transmitidos por protocolos de telecomunicações, independentemente da linguagem e da representação física desses dados [19].

Há também o modelo de serviço, que descreve os serviços disponíveis para cada uma das classes do DIM. Estes serviços ditam a forma de transmitir, solicitar ou modificar dados de um agente. Os serviços são governados pelo modelo de comunicação, o qual descreve quando e em qual contexto um serviço deve ser utilizado. O modelo de comunicação faz isso utilizando máquina de estados.

O padrão X73-PHD possui especificações distintas para cada PHD, chamadas de especializações. As especializações definem um conjunto de padrões para cada um dos PHDs. Cada especialização possui seu próprio DIM, isto é, quais atributos e objetos um dispositivo deve possuir. Definir um DIM separadamente para cada dispositivo é importante, pois um eletrocardiograma possui necessidades técnicas diferentes de um oxímetro de pulso. Todas as especializações seguem um modelo de serviço e comunicação único, definidos no protocolo *Optimized Exchange Protocol* (OEP). O protocolo OEP é o coração da norma X73-PHD e é onde são definidos os DIMs comuns a todos os dispositivos, assim como o modelo de serviço e o modelo de comunicação.

Serão abordados nas próximas seções as especializações e suas características e o protocolo OEP, que define o DIM, o modelo de serviço e o modelo de comunicação.

2.3 Perfis de especialização

A família de normas 11073 inclui perfis de especialização, isto é, cada PHD possui uma norma associada que descreve como esse dispositivo faz sua representação de dados e como as informações são transmitidas para os gerentes.

Por exemplo, a norma IEEE 11073-10408 define padrões para um termômetro, já a norma IEEE 11073-10415 para uma balança. Essas normas são chamadas de “especializações de dispositivos”. Os padrões de especialização de dispositivos definem por exemplo, o modelo de informação do dispositivo DIM. Neste modelo de informação, são especificados todos os atributos de cada classe ou objeto, assim como métodos e eventos. A Figura 2.3 apresenta o modelo de domínio de informação para um termômetro, conforme a norma IEEE 11073-10408.

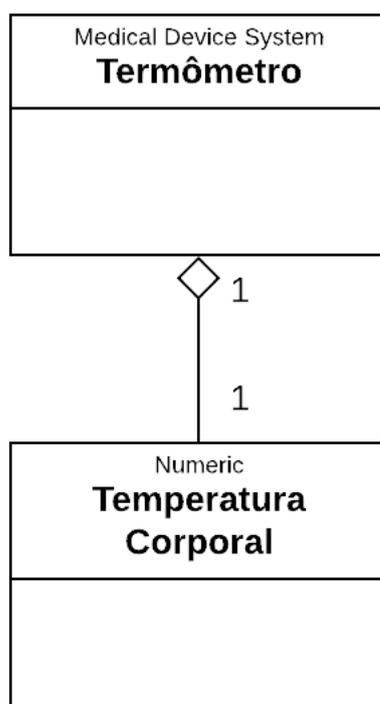


Figura 2.3: Termômetro - Modelo de Domínio de Informação.

O objeto Termômetro, do tipo *Medical Device System*, possui um objeto Temperatura Corporal, do tipo *Numeric*. O objeto *Medical Device System* pode estar relacionado com quantos objetos for necessário no caso do termômetro, ele apenas possui um.

Neste padrão, a nomenclatura IEEE 11073-104zz é usada para se referir à coleção de especializações, onde zz pode ser um número entre 01 e 99.

2.4 Optimized Exchange Protocol - IEEE 11073-20601

Dentro da família de normas 11073, a parte 11073-20601 define um protocolo chamado OEP que estabelece um *framework* para um modelo abstrato, o qual implementa uma conexão lógica entre sistemas, garantindo a interoperabilidade entre dispositivos de diferentes fabricantes [15].

As conexões são normalmente feitas ponta-a-ponto e um agente trabalha com um único gerente. Entretanto, um gerente pode trabalhar com múltiplos agentes. Os dados são transportados por *Application Protocol Data Units* (APDUs). Estas APDUs devem ser processadas automaticamente e podem ser segmentadas e remontadas. O tamanho máximo de uma APDU quando enviada de um agente para um gerente é de 63 kB, e quando enviada de um gerente para um agente é de 8 kB. Esses limites podem ser reduzidos pelas especializações dos dispositivos.

O padrão X73-PHD é *transport-agnostic*. Isso quer dizer que as APDUs podem ser transportadas por quase todas as tecnologias baseadas em pacotes, como TCP/IP, Bluetooth e USB e etc. Este padrão preenche as três camadas mais altas do modelo *Open System Interconnection* (OSI): sessão, apresentação e aplicação.

As especificações do protocolo OEP são escritas na linguagem ASN.1. Atualmente, codificações de mensagens ASN.1 para fluxos de bytes são ditadas por um conjunto de regras como *Basic Encoding Rules* (BER), *XML Encoding Rules* (XER) e *Medical Data Encoding Rules* (MDER). No caso de dispositivos para saúde, especialmente agentes e gerentes 11073, implementam a codificação MDER.

Outras características importantes deste protocolo, como máquina de estados, modelo de dados, modelo de serviço e modelo de comunicação, serão discutida nas próximas seções.

2.5 Domain Information Model (DIM)

PHDs, dentro do contexto do padrão X73-PHD, são definidos por um modelo de dados orientado a objetos. O DIM define várias classes, atributos e serviços para os agentes e os gerentes. A comunicação entre agentes e gerentes é realizada através de métodos bem definidos, como GET e SET. Leituras feitas pelo agente são transmitidas utilizando métodos chamados EVENT REPORT.

Um ponto chave na definição do DIM é a utilização de códigos e nomenclaturas de-

finidos pelo padrão ISO/IEEE 11073-10101 [14]. Este padrão define códigos e nomenclaturas usados para comunicação de dados de dispositivos médicos. Por exemplo, mensagens com dados sobre o peso em quilogramas de um paciente têm como identificador o id MDC_DIM_KILO_G o qual representa o código 1731. Usando uma nomenclatura consistente, a interoperabilidade é aprimorada, pois todas as implementações mantêm o mesmo significado semântico para os códigos numéricos.

Para representar informações e fazer o envio de eventos de leituras, o DIM idealiza e define várias estruturas de dados dentro de cada agente. Essas estruturas são definidas por um modelo de dados orientado a objetos. Cada classe pode ser estendida por suas respectivas especializações. Na Figura 2.4, é mostrado o diagrama de classe comum a todos os PHD e suas relações. Podem existir nenhum ou vários objetos *Numeric*, *Real-Time Sample Array* (RT-SA), *Enumeration*, *PM-Store* e *Scanner* associados com a classe MDS. As classes *Enumeration*, *Numeric* e RT-SA são derivadas da classe base *Metric*, a qual contém e compartilha atributos comuns entre essas três classes filhas.

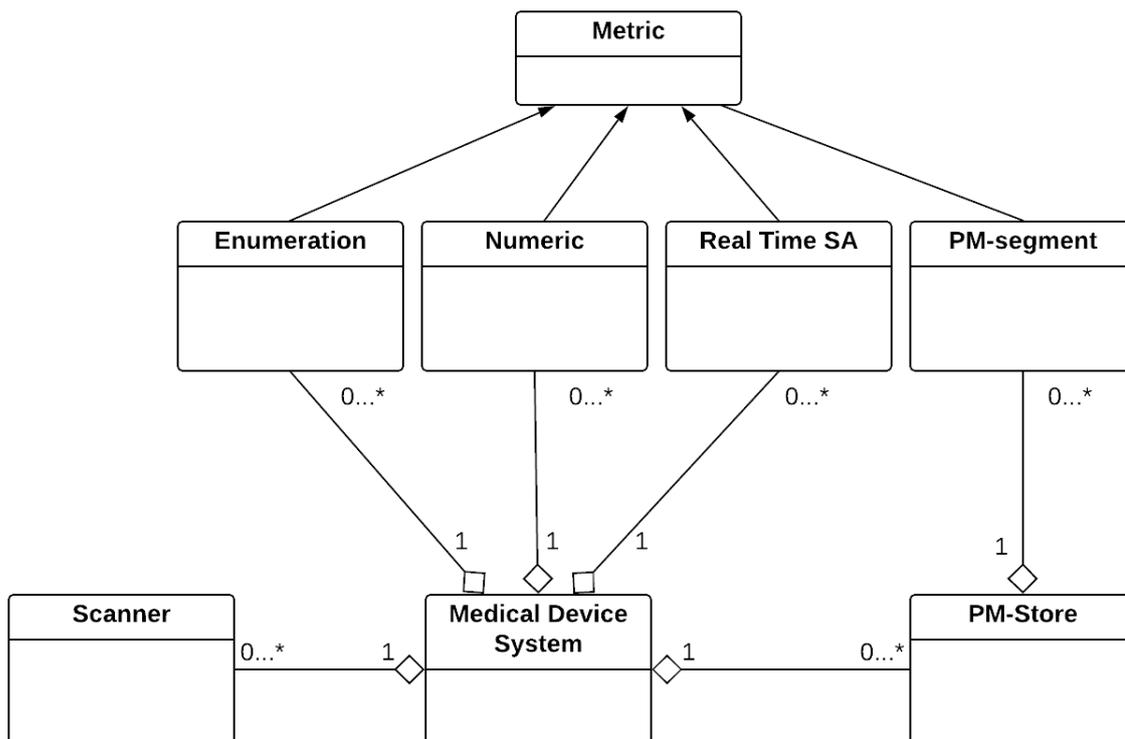


Figura 2.4: Modelo de Domínio de Informação para dispositivos pessoais de saúde.

Cada agente contém exatamente um objeto do tipo MDS. A classe MDS possui atributos como id do sistema, nome do fabricante, tipo de especialização e etc. Essa classe também contém objetos “filhos” - aqueles que são herdados da classe MDS, e são esses objetos que contém de fato os dados médicos. Os atributos de todas as classes

podem ser acessados e modificados através de métodos como SET e GET.

Os atributos de todas as classes podem ser opcionais, condicionais ou obrigatórios. Os atributos opcionais são implementados opcionalmente pelos agentes. Os atributos condicionais são implementados de acordo com a medição que o agente realiza, ou seja, o atributo apenas existirá caso seja necessário. Os atributos obrigatórios devem ser implementados por todos os agentes.

A classe MDS representa informações técnicas sobre o dispositivo como, por exemplo, tipo, fabricante, status sobre a bateria e modelo.

Alguns desses atributos são apresentados brevemente. A tabela é composta por três colunas. A primeira mostra o nome do atributo na coluna nome. A segunda coluna ID do Atributo mostra o nome que este atributo terá no código, isto é, qualquer implementação seguindo o padrão 11073 possuirá, por exemplo, o atributo System-type definido como MDC_ATTR_SYS_TYPE. A última coluna, Observação, é utilizada para detalhar o uso do atributo.

O atributo *Attribute-Value-Map* é utilizado para representar a estrutura dos dados para a transmissão. Como o próprio nome sugere, este atributo mapeia valores. Esse mapeamento é feito para atualizações de medições. Na prática, este atributo define o tamanho do campo que irá carregar as medições e uma marca de tempo para cada medição.

Tabela 2.1: Atributos da classe MDS.

| Nome | ID do Atributo | Observação |
|---------------------|----------------------------|---|
| System-type | MDC_ATTR_SYS_TYPE | Este atributo define o tipo do agente, como definido na nomenclatura (e.g. <i>weighting scale</i>). Os valores devem vir da norma ISO/IEEE 11073-10101 |
| System-Model | MDC_ATTR_ID_MODEL | Este atributo define o fabricante e o número do modelo. |
| Attribute-Value-Map | MDC_ATTR_ATTRIBUTE_VAL_MAP | Este atributo define um formato fixo para o envio dos dados de leitura. |
| Battery-Level | MDC_ATTR_VAL_BATT_CHARGE | Este atributo reporta a percentagem restante da bateria. |

A classe *Metric* é a classe base para as classes que de fato definem um tipo de dado

médico. Esta classe é responsável por definir, por exemplo, que um dispositivo do tipo termômetro envia medições de temperatura em graus celsius. Como mostrado na Tabela 2.2, o atributo *Type* determina a leitura que tal dispositivo realiza. *Unit-Code* é o atributo no qual é definido a unidade de medida.

Tabela 2.2: Atributos da classe *Metric*.

| Nome | ID do Atributo | Observação |
|-----------|--------------------|---|
| Type | MDC_ATTR_ID_TYPE | Este atributo define o tipo da métrica utilizada, como por exemplo, peso ou batimentos do coração. |
| Unit-Code | MDC_ATTR_UNIT_CODE | Este atributo define a unidade de medida utilizada, como por exemplo, peso em quilogramas ou batimentos por minuto. |

Uma instância da classe *numeric* representa uma medição numérica. Esta classe herda atributos e métodos da classe *metric*. É nesta classe onde é de fato armazenada uma medição. Todos os atributos desta classe definem tipos para armazenar dados médicos. Pode-se ver na Tabela 2.3 que todos os atributos são definições de tipos numéricos, e não contêm nenhum atributo de configuração. A diferença entre os atributos é o tamanho em bits que cada tipo suporta.

Um balança que envia dados de peso e um termômetro que envia informações sobre temperatura são alguns dos dispositivos que utilizam esta classe para transmitir valores numéricos.

Uma instância da classe RT-SA representa uma leitura na forma de onda. Geralmente são dados que precisam ser enviados em tempo real como, por exemplo, um eletrocardiograma. Esta classe também herda atributos da classe *metric*.

É possível ver três atributos da classe RT-SA na Tabela 2.4. O atributo *Sample-Period* apenas determina o intervalo de tempo em que as leituras foram realizadas. *Simple-Sa-Observed-Value* determina o tamanho do vetor de leituras que será transmitido e o tamanho, em bits, de cada valor do vetor.

O atributo *Scale-and-Range-Specification* define um coeficiente para um algoritmo mapear valores absolutos - leituras que podem ser em ponto flutuante e negativos, em valores escalares positivos. Isso é útil para enviar medições absolutas como -0.060 , o que exigiria pelo menos 4 bytes no formato de ponto flutuante da norma X73-PHD. Quando

Tabela 2.3: Atributos da classe Numeric.

| Nome | ID do Atributo | Observação |
|-----------------------------------|--------------------------------|--|
| Simple-Nu-Observed-Value | MDC_ATTR_NU_VAL_OBS_SIMP | Este atributo define o valor numérico de uma medição. |
| Compound-Simple-Nu-Observed-Value | MDC_ATTR_NU_CMPD_VAL_OBS_SIMP | Este atributo define um vetor de valores Simple-Nu-Observed-Value. |
| Basic-Nu-Observed-Value | MDC_ATTR_NU_VAL_OBS_BASIC | Este atributo define o valor numérico de uma medição, porém, com uma representação numérica menor. |
| Compound-Basic-Nu-Observed-Value | MDC_ATTR_NU_CMPD_VAL_OBS_BASIC | Este atributo define um vetor de valores Basic-Nu-Observed-Value |

convertido, o número -0.060 torna-se 388, o que requer apenas dois bytes para ser enviado. O padrão X73-PHD define seus próprios tipos de ponto flutuante compostos por 32 bits que compreendem um expoente inteiro com sinal de 8 bits seguido por uma mantissa com sinal de 24 bits.

No gerente, a equação para fazer a conversão dos valores escalares (convertidos) para valores absolutos é dada pela expressão $Y = M \times X + B$, onde:

$$Y = \text{valor absoluto}$$

$$M = \frac{(\text{maior valor absoluto} - \text{menor valor absoluto})}{(\text{maior valor escalar} - \text{menor valor escalar})}$$

$$B = \text{maior valor absoluto} - (M \times \text{maior valor escalar})$$

$$X = \text{valor escalar}$$

No agente, a conversão de valores absolutos para valores escalares é dada pela expressão $X = \frac{(R-B)}{M}$, onde:

$$R = \text{valor atual da leitura (valor absoluto)}$$

Um termômetro que faz leituras de -45°C até 50°C com uma resolução de 0.5°C tem os seguintes valores:

Menor valor absoluto = -45.0

Maior valor absoluto = 50.0

Menor valor escalar = 0

Maior valor escalar = 190

Tabela 2.4: Atributos da classe RT-SA.

| Nome | ID do Atributo | Observação |
|-----------------------------------|------------------------------|--|
| Sample-Period | MDC_ATTR_ TIME_PD_SAMP | Este atributo define o intervalo de tempo entre cada medição. |
| Simple-Sa- Observed-Value | MDC_ATTR_SIMP_SA_O BS_VAL | Este atributo define o vetor com as medições. |
| Scale-and- Range-Specification | MDC_ATTR_ SCALE_SPECN | Este atributo define o valores necessários para conversões das medições para valores inteiros sem sinal. |

A classe *Enumeration* basicamente é utilizada para informar status do dispositivo ou informações extras. Os valores do objeto *Enumeration* são codificados de acordo com a nomenclatura definida no padrão [14]. Esta classe herda atributos da classe *Metric*.

Um eletrocardiograma possui dois objetos *Enumeration*, um para informar o status do dispositivo e outro para informar a razão pela qual uma transmissão teve início. Por exemplo, pode-se informar a perda de um sinal de um eletrodo utilizando o mnemônico `leadsignal-loss`, ou ainda, pode-se especificar o eletrodo que perdeu o sinal utilizando `leadsignal-loss-first-lead`.

Uma transmissão pode ser disparada quando o usuário pressiona um botão ou quando uma atividade cardíaca suspeita é detectada. Assim, podemos usar informações como `MDC_ECG_EVT_CTXT_USER` para informar que a transmissão se deu devido a uma ação do usuário. Três atributos dessa classe podem ser consultados na Tabela 2.5.

Alguns dispositivo agentes podem coletar dados enquanto estão fora do alcance de um gerente, ou podem também enviar esses dados para mais de um gerente, e.g., descarregar uma cópia na casa do usuário e outra cópia em um hospital. A norma X73-PHD estabelece duas classes que configuram o armazenamento temporário de leituras nos agentes. A primeira é a *Persistent Metric Segment* (PM-Segment), que armazena de fato as leituras.

Tabela 2.5: Atributos da classe Enumeration.

| Nome | ID do Atributo | Observação |
|------------------------------------|------------------------------------|---|
| Enum-Observed-Value-Simple-OID | MDC_ATTR_ENUM_OBS_VAL_SIMP_OID | Este atributo define o status de uma leitura ou do dispositivo, utilizando nomenclaturas predefinidas. |
| Enum-Observed-Value-Simple-Bit-Str | MDC_ATTR_ENUM_OBS_VAL_SIMP_BIT_STR | Este atributo define o status de uma leitura ou do dispositivo. Neste caso, utilizando uma string de 32 bits. |
| Enum-Observed-Value | MDC_ATTR_VAL_ENUM_OBS | Este atributo define o status de uma leitura ou do dispositivo. Neste caso, o tamanho do stream é definido pelo fabricante. |

Pode existir um número variável de PM-Segments em um único agente. Já a classe *Persistent Metric Store* (PM-Store) gerencia, organiza e envia as leituras contidas nas PM-Segment. Tabelas com alguns atributos das classes PM-Store e PM-Segment são mostradas em 2.6 e 2.7 respectivamente.

Tabela 2.6: Atributos da classe PM-Store.

| Nome | ID do Atributo | Observação |
|--------------------|------------------------------------|---|
| Sample-Period | MDC_ATTR_ENUM_OBS_VAL_SIMP_OID | Este atributo define o tipo de leitura que será transmitida, utilizando nomenclaturas predefinidas. |
| Number-Of-Segments | MDC_ATTR_ENUM_OBS_VAL_SIMP_BIT_STR | Este atributo define o tipo de medição que será transmitida. Neste caso, utilizando uma string de 32 bits |
| Operational-State | MDC_ATTR_OP_STAT | Este atributo indica se dados estão sendo inserido no PM-segment que está contido neste PM-store. |

Agentes não enviam os dados de leituras coletados e armazenados temporariamente enquanto um gerente não os solicitar. Ou seja, dados armazenados nesta classe apenas são enviados utilizando o modo *Manager-initiated*, que será explicado posteriormente.

Na Figura 2.5 é possível ver o diagrama da representação de uma *PM-Store* com dois *PM-Segments*. Não há limites para o número de entradas que um *PM-Segment* pode

Tabela 2.7: Atributos da classe PM-Segment.

| Nome | ID do Atributo | Observação |
|-----------------------|--------------------------------|--|
| PM-Seg-Person-Id | MDC_ATTR_ENUM_OBS_VAL_SIMP_OID | Este atributo permite que o agente possua medições de múltiplas pessoas. |
| PM-Segment- Entry-Map | MDC_ATTR_PM_SEG_MAP | Este atributo define o formato que as medições estão no PM-segment. |
| Instance- Number | MDC_ATTR_ID_INSTNO | Este atributo identifica um PM-segment de forma única dentro do contexto de um PM-store. |

conter, nem para o número de *PM-Segments* que uma *PM-Store* pode ter. Imagine um oxímetro de pulso que mede a frequência cardíaca, sua velocidade, e a SpO₂ (saturação do oxigênio no sangue) a cada segundo. Neste caso, são três valores numéricos que podem ser armazenados em um *PM-Segment*, cada entrada contendo os três valores, lidos a cada segundo.

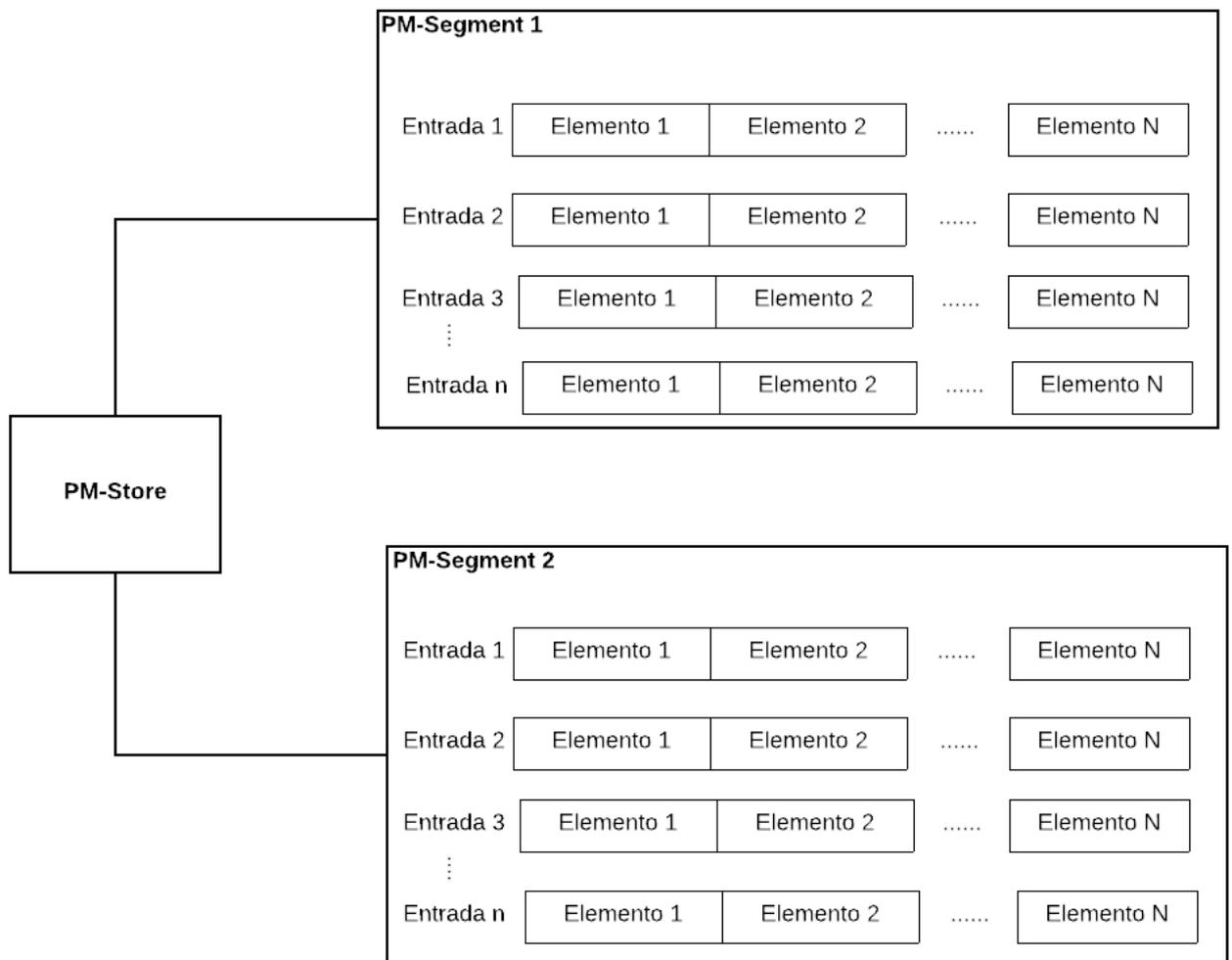
| Frequência Cardíaca | Velocidade | SpO ₂ |
|---------------------|------------|------------------|
| 120 | 10 | 98 |

A classe *Scanner* é responsável por combinar dois ou mais objetos diferentes da classe *Metric* e enviá-los de forma periódica ou episódica. Quando os eventos são episódicos, não é conhecido o intervalo de tempo entre cada evento. Quando o evento é periódico, o intervalo de tempo entre cada evento é conhecido.

Note que, apesar do nome, esta não é uma classe para escanear objetos e transmitir dados quando valores são atualizados. Ela apenas otimiza o envio de leituras, combinando elas em apenas um EVENT REPORT. Esta classe se caracteriza como sendo uma ferramenta auxiliar no envio de leituras. Dois atributos são apresentados na Tabela 2.8.

Tabela 2.8: Atributos da classe Scanner.

| Nome | ID do Atributo | Observação |
|-------------------|---------------------------|---|
| Operational-State | MDC_ATTR_OP_STAT | Este atributo indica se o scanner está mandando atributos ou não. |
| Scan-Handle-List | MDC_ATTR_SCAN_HANDLE_LIST | Este atributo contém as medições que irão ser enviadas. |

Figura 2.5: Uma *PM-Store* com dois *Segments*.

2.6 Modelo de Serviço

O modelo de serviço define os tipos de mensagens que podem ser trocadas entre um agente e um gerente, assim como o contexto em que elas são trocadas. A forma como acessar valores de atributos, fazer solicitações de ações remotas e enviar dados são alguns tipos de serviços definidos neste modelo.

Cada uma das classes citadas na Seção 2.5 possuem serviços que podem ser suportados ou não. Nesta seção, abordaremos de forma geral o que cada serviço faz, não se preocupando como esse serviço é implementado por cada classe.

2.6.1 Serviços de Acesso

Serviços de acesso são usados para obter informações sobre dados de objetos definidos no DIM. Estes serviços são, portanto, especificamente para enviar leituras e acessar informações sobre agentes. Os serviços suportados são:

- **GET:** Usado pelo gerente para recuperar informações das classes MDS e PM-Store dos agentes, e.g., durante o processo de associação o gerente solicita informações de modelo, fabricante, tipo, utilizando um serviço GET.
- **SET:** Usado pelo gerente para definir valores nos atributos da classe *Scanner* do agente.
- **EVENT REPORT:** Este serviço é o mais utilizado tanto pelo agente quanto pelo gerente. Este serviço é utilizado pelo agente para transmitir dados de leitura e configurações e pelo gerente para responder com um ACK, ou quaisquer solicitações que requerem uma resposta.
- **ACTION:** Serviço utilizado pelo gerente para invocar ações no agente. Por exemplo, quando o gerente solicita que o agente inicie uma transmissão de leituras.

2.6.2 Serviço de Associação

O serviço de associação gerencia o processo de associação entre um agente e um gerente. As mensagens envolvidas neste processo são:

- *Association Request:* Utilizado exclusivamente pelo agente. Esta mensagem é a primeira enviada para iniciar o processo de associação.

- *Association Response*: Mensagem enviada pelo gerente confirmando uma solicitação de associação.
- *Association Release*: Solicita o término da associação, pode ser enviado por ambos dispositivos.
- *Association Release Response*: Confirma a solicitação de desassociação.
- *Abort*: Encerra uma associação imediatamente sem esperar confirmação. É utilizado frequentemente em situações de erro.

2.6.3 EVENT REPORT com e sem confirmação

EVENT REPORT é o principal mecanismo de envio de leituras e mensagens de configurações. Este serviço está presente apenas nas classes MDS e *Scanner*. Ele pode ter várias formas e propriedades, como será comentado nesta seção.

O dispositivo que envia um EVENT REPORT pode, opcionalmente, solicitar uma confirmação do destinatário desta mensagem, utilizando uma mensagem do tipo CONFIRMED EVENT REPORT. O receptor desta mensagem deve confirmar o recebimento da mensagem enviado um ACKNOWLEDGED CONFIRMED EVENT REPORT.

Nos eventos sem confirmação, uma resposta não é esperada do destinatário. Esta mensagem é denominada como UNCONFIRMED EVENT REPORT.

Nos eventos *Agent-initiated data transmission*, o agente tem a iniciativa de enviar dados, como resultado de uma ação do usuário, ou uma alteração detectada pelo sensor.

Manager-initiated data transmission ocorre quando o gerente dispara um comando, solicitando explicitamente que o agente inicie uma transmissão de dados. O gerente por sua vez, pode tanto solicitar o início quanto o fim da transmissão de dados de um agente. Dispositivos agentes e gerentes devem suportar esses dois modos de transmissão (*Agent* e *Manager-initiated*).

Dispositivos agentes que serão usados em vários pacientes como, por exemplo, em um hospital devem suportar o serviço MULTIPLE-PERSON EVENT REPORT. Quando deseja-se transmitir esses dados para um gerente, o ideal é enviar os dados de todas as pessoas em apenas um EVENT REPORT. Caso não seja possível, o agente pode enviar as leituras de cada paciente, separadamente, utilizando SINGLE-PERSON EVENT REPORT. Agentes e gerentes devem suportar SINGLE e MULTIPLE-PERSON EVENT REPORT.

2.7 Modelo de comunicação

No modelo de comunicação, temos as definições de máquina de estado, condições de entrada e saída de cada estado e definições de *timeouts*.

A topologia prevista na norma X73-PHD consiste em um ou mais agentes conectados a um gerente, na qual a comunicação é ponta-a-ponta. Cada conexão com os agentes é tratada de forma independente.

2.7.1 Máquina de estados

O funcionamento dos agentes e gerentes é governado por uma máquina de estados. Na Figura 2.6 é apresentada uma visão completa de todos os estados da máquina de estados de um agente. As transições e as mensagens trocadas também podem ser vistas nesta imagem. Perceba que esta máquina de estados impõe uma hierarquia de estados, e.g., não se pode passar do estado *connected* para *associated* antes de entrar no estado *associating*. De forma resumida, temos os seguintes estados e suas hierarquias:

- Disconnected
- Connected
 - Associating
 - Associated
 - * Operating
 - Disassociating
 - Unassociated

Quando um agente é ligado, ele inicialmente entra no estado *disconnected*, indicando que uma conexão de nível de transporte ainda não foi estabelecida com o gerente. Caso a conexão na camada de transporte seja finalizada, o agente volta imediatamente para o estado *disconnected*.

Após a camada de transporte indicar que uma conexão foi estabelecida com o gerente, o agente imediatamente transita para o estado de máquina *connected*. O agente permanece nesse estado enquanto há uma conexão na camada de transporte. Dentro deste estado, o agente pode transitar para o sub-estado *unassociated*. Caso a conexão na camada de transporte seja encerrada, o agente retorna para o estado *disconnected*.

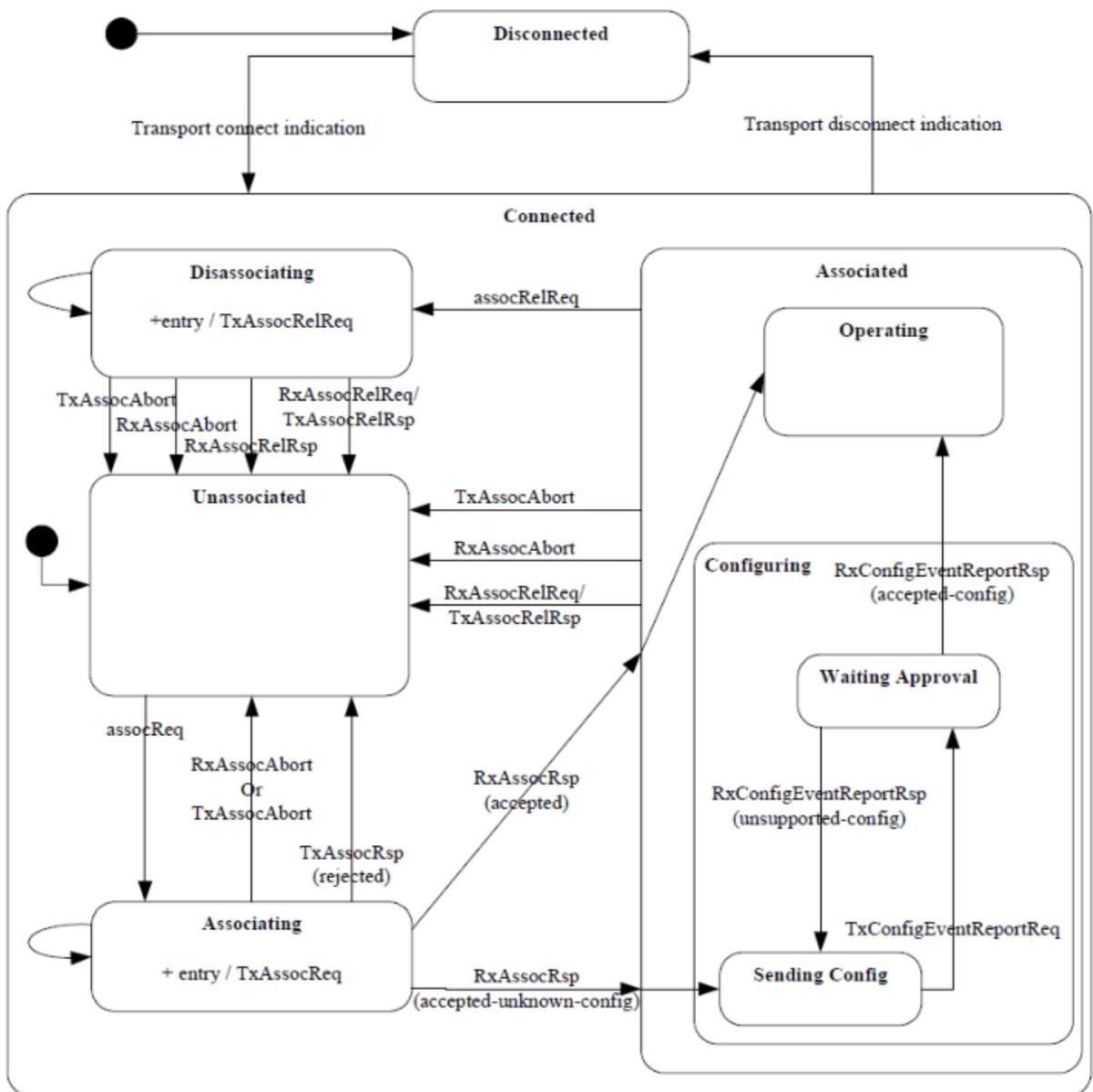


Figura 2.6: Diagrama da máquina de estados de um agente. Fonte: [23].

O agente permanece no estado *unassociated* enquanto não existir uma associação, na camada de aplicação, com o gerente. Este também é sempre o estado inicial depois que uma associação é terminada ou abortada. Normalmente, o agente não faz nada neste estado. O gerente por sua vez, espera uma mensagem de *Association Request* neste estado. Apenas quando o agente decide iniciar a associação, é feita a transição para o estado *Associating*.

2.7.2 Associating

O agente entra neste estado a partir do momento que ele envia a mensagem de *Association Request*. O gerente entra neste estado quando recebe a mensagem de *Association Request*. Mesmo sob condições de falha, isto é, a mensagem de *Association Request* não foi respondida pelo gerente, o agente mesmo assim continua neste estado até o número máximo de tentativas de associações ser atingido. O procedimento de associação permite que o agente e o gerente concordem em usar o mesmo protocolo de dados e um conjunto comum de parâmetros operacionais.

O gerente apenas prossegue para o próximo estado quando envia a resposta da associação, *association response*. O agente, portanto, transita para o próximo estado no momento em que recebe o *association response*.

O agente deve esperar pelo *association response* por um tempo de TO_{assoc} (timeout). Se o período TO_{assoc} expirar, deve-se reenviar a mensagem *association request*. Este processo é repetido até que um *association response* seja recebido ou até que RC_{assoc} (retry count) seja atingido. A mensagem de *association request* é enviada $RC_{assoc} + 1$ vezes. Se nenhuma resposta é recebida neste período, o agente envia uma mensagem de *association abort* e transita de volta para o estado *unassociated*.

Se o agente ou gerente receberem uma mensagem de *association abort* durante o processo de associação, eles deverão transitar para o estado *unassociated*.

Vários agentes podem estar associados a um gerente. A conexão/associação normalmente é iniciada pelo agente, pois é ele quem detecta novos dados a serem transmitidos. A associação consiste nos seguintes passos, como apresentados na Figura 2.7.

O procedimento de associação começa com o agente enviando uma mensagem de *association request* para o gerente. Essa mensagem contém, por exemplo, a identificação do agente, informações das funcionalidades que o agente suporta, informações de codificação e decodificação de mensagens. Após o gerente receber esta mensagem, ele verifica se é

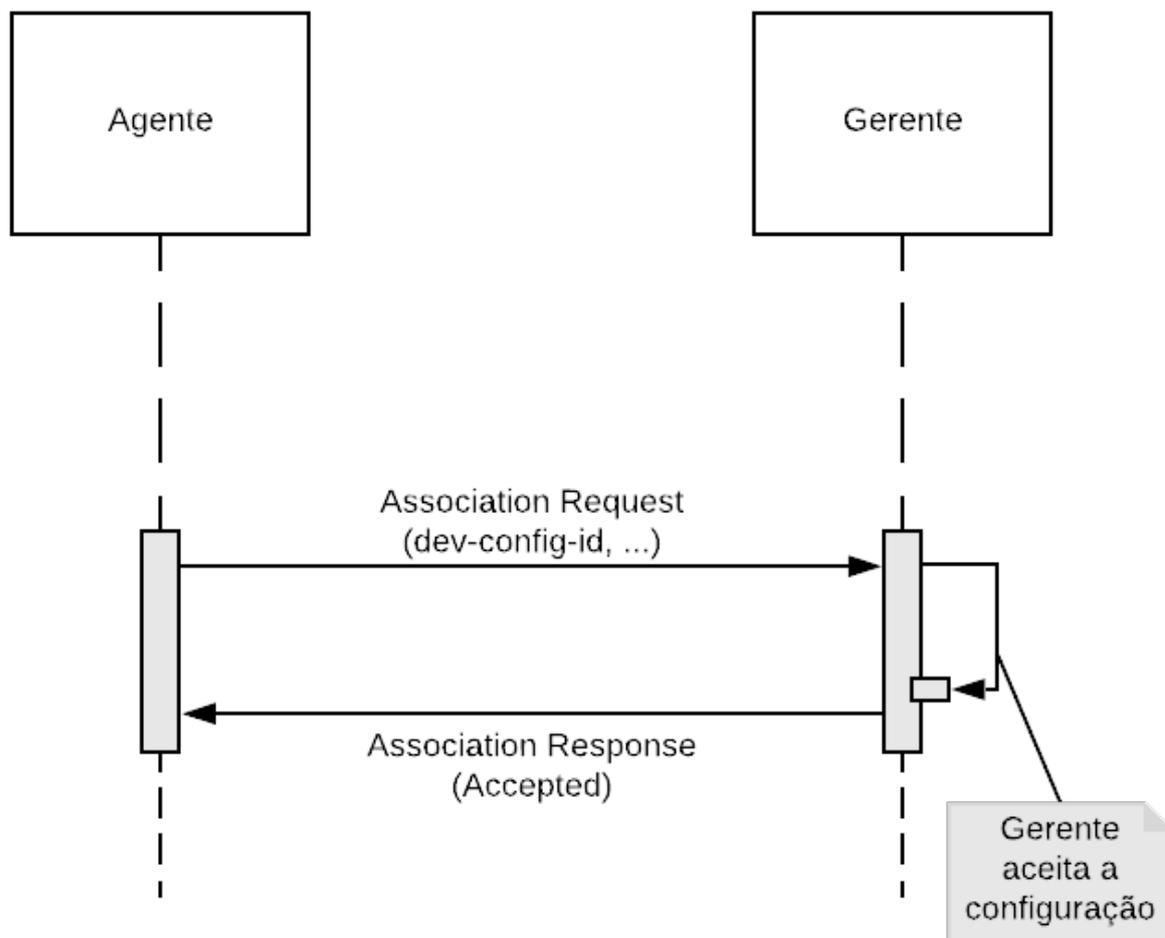


Figura 2.7: Diagrama de sequência da associação de um agente com um gerente (adaptado de [23]).

compatível com este tipo de agente, ou se a configuração já é conhecida. Caso a configuração não seja conhecida, o gerente responde com uma mensagem de *association response* dizendo que a associação pode ser feita, porém, parâmetros de configuração do agente são necessários. Na Figura 2.7, é considerado que o gerente já havia se conectado previamente ao agente. Portanto, a mensagem de *association response* não contém o campo de solicitação de parâmetros de configuração do agente.

2.7.3 Operating

O agente entra neste estado quando recebe a mensagem de *association response* do gerente, e o gerente quando envia o *association response*.

A transferência de leituras e o envio de status do agente acontecem neste estado. Como já comentado previamente, existem dois modos de transmissão de leituras: *Agent-initiated*

e *Manager-initiated*. Geralmente agentes que transmitem pequenas quantidade de dados como, por exemplo, um termômetro, usam o modo *Agent-initiated*. Já para agentes que geram grande quantidade de dados em pequenos intervalos de tempo, é preferível o uso do modo *Manager-initiated*, o qual permite controlar o fluxo de dados. Se algum erro ocorrer durante este estado, uma mensagem de *association abort* é enviada e ambos os dispositivos voltam para o estado *unassociated*.

O gerente pode encerrar a transmissão de dados iniciado pelo agente enviando uma mensagem de *Association Release Request* ou *Association Abort* para encerrar a associação. Se o agente está usando um objeto *Scanner*, o gerente pode desabilitar a transmissão utilizando o serviço SET.

Na Figura 2.8 é possível ver o digrama de sequência da transmissão de leituras iniciada por um agente. Neste diagrama, o agente tem a iniciativa de enviar leituras, solicitando uma confirmação de recebimento. O parâmetro *Invoke*, na mensagem enviada do agente para o gerente, significa que esta mensagem requer confirmação de recebimento. O agente poderia não solicitar a confirmação e apenas enviar as leituras. Assim, a mensagem vindo do gerente não existiria.

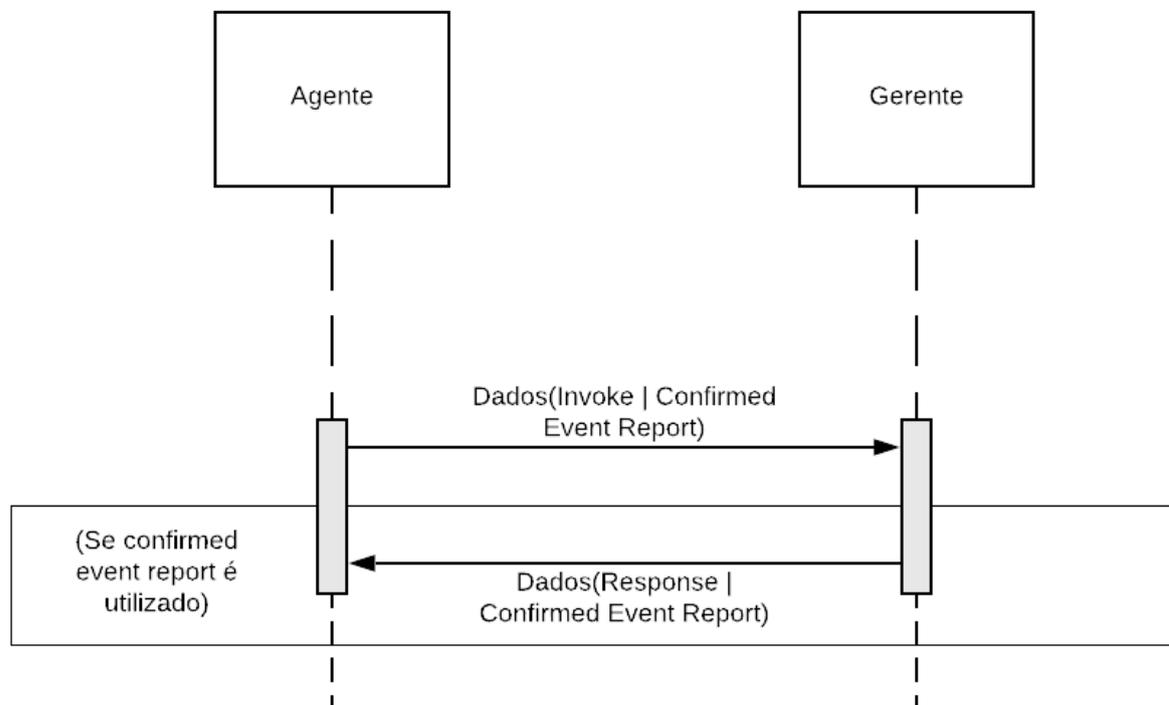


Figura 2.8: Transmissão de dados utilizando o modo *Agent-initiated* (adaptado de [23]).

No modo *manager-initiated*, o gerente utiliza o serviço ACTION para solicitar dados do agente. Quando o gerente deseja fazer essa solicitação, ele deve enviar uma APDU,

indicando o início (*Start DataRequest*) ou término (*Stop DataRequest*) de uma transmissão. Se é uma requisição para iniciar, o modo deve ser especificado como, *single mode*, *time period mode* ou *no time limit mode*.

O modo *single response* permite que o gerente solicite dados para o agente e receba a leitura já na mensagem de resposta. Se o agente não tem nada para enviar no momento da solicitação, uma lista vazia é retornada. Apenas neste modo, dados de leitura não são enviados usando EVENT REPORT, mas sim, ACTION RESPONSE. Neste modo também, não é possível solicitar confirmação de recebimento.

O modo *Manager-initiated time period mode* permite que o agente envie dados coletados por um período de tempo predeterminado pelo gerente. Quando o agente recebe a mensagem de *Start DataRequest* do gerente, o agente deve enviar uma mensagem de *DataResponse*, sem incluir nesta mensagem dados de leitura. Quando dados já estiverem prontos para serem enviados, o agente deve utilizar o serviço EVENT REPORT para fazer a transmissão durante o tempo determinado na mensagem *Start DataRequest*. A transmissão pode ser interrompida por uma mensagem de *Stop DataRequest* ou se a associação terminar.

Diferente do *single mode*, neste modo o agente pode escolher entre enviar mensagens com confirmação ou sem confirmação. Isso pode ser útil em casos onde o agente verifica se a leitura foi entregue com sucesso antes de removê-la da memória ou cache local. É apresentado o diagrama de sequência da troca de mensagens deste modo na Figura 2.9.

O modo *no time limit* permite o agente enviar dados continuamente, sem limite de tempo predeterminado pelo gerente. A transmissão deve ser interrompida apenas quando o agente receber uma mensagem de *Stop DataRequest*, ou a associação entre o agente e gerente terminar. O procedimento pode ser visto na Figura 2.10.

2.8 Casos de Uso do padrão X73-PHD

Em vários contextos, PHDs são úteis para o suporte de atividades individuais de diferentes formas. Três grandes famílias de casos de usos podem ser destacadas: Saúde e Atividade Física, Independência para Terceira Idade e Gerenciamento de Doenças.

Equipamentos de força, monitores de pressão sanguínea e balanças são equipamentos frequentemente utilizados para Saúde e Atividades físicas. Os usuários deste grupo, em geral, são pessoas que utilizam vários equipamentos de atividades físicas para manter a

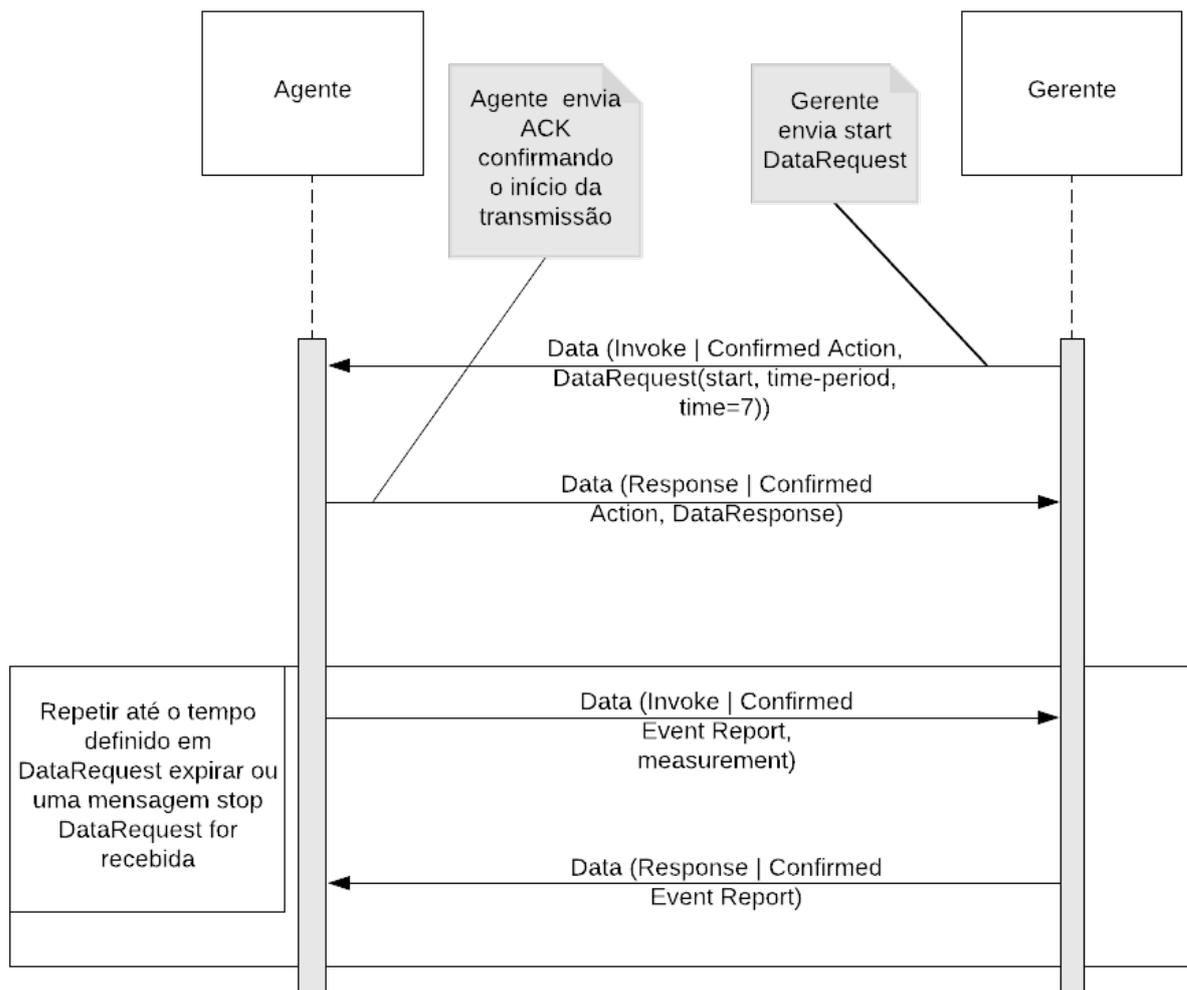


Figura 2.9: Manager-initiated utilizando time period mode (adaptado de [23]).

saúde em dia. Os dados coletados pelos agentes podem ser armazenados em um PC, e posteriormente, transmitidos para *personal trainers* para avaliação. Os usuários possuem e operam seus próprios equipamentos, e buscam por ajuda de profissionais para montar ou melhorar seus planos de atividades físicas.

Já para Independência para a Terceira Idade, dispositivos como *hub* de atividade de vida independente, monitor de medicação, monitor de pressão sanguínea, balança, termômetro, medidor de glicose e oxímetro de pulso são equipamentos tipicamente utilizados. “Independência” ou “envelhecimento independente” são termos usados, por exemplo, pela *Personal Health Alliance*. Casas inteligentes, com dispositivos sensores e atuadores conectados, podem estender o período de tempo de vida independente para pessoas idosas. Funcionalidades como gerenciamento de aquecedores e abertura de portas podem melhorar a qualidade de vida substancialmente.

Por fim, para o gerenciamento de doenças, também pode-se utilizar o *hub* de ativi-

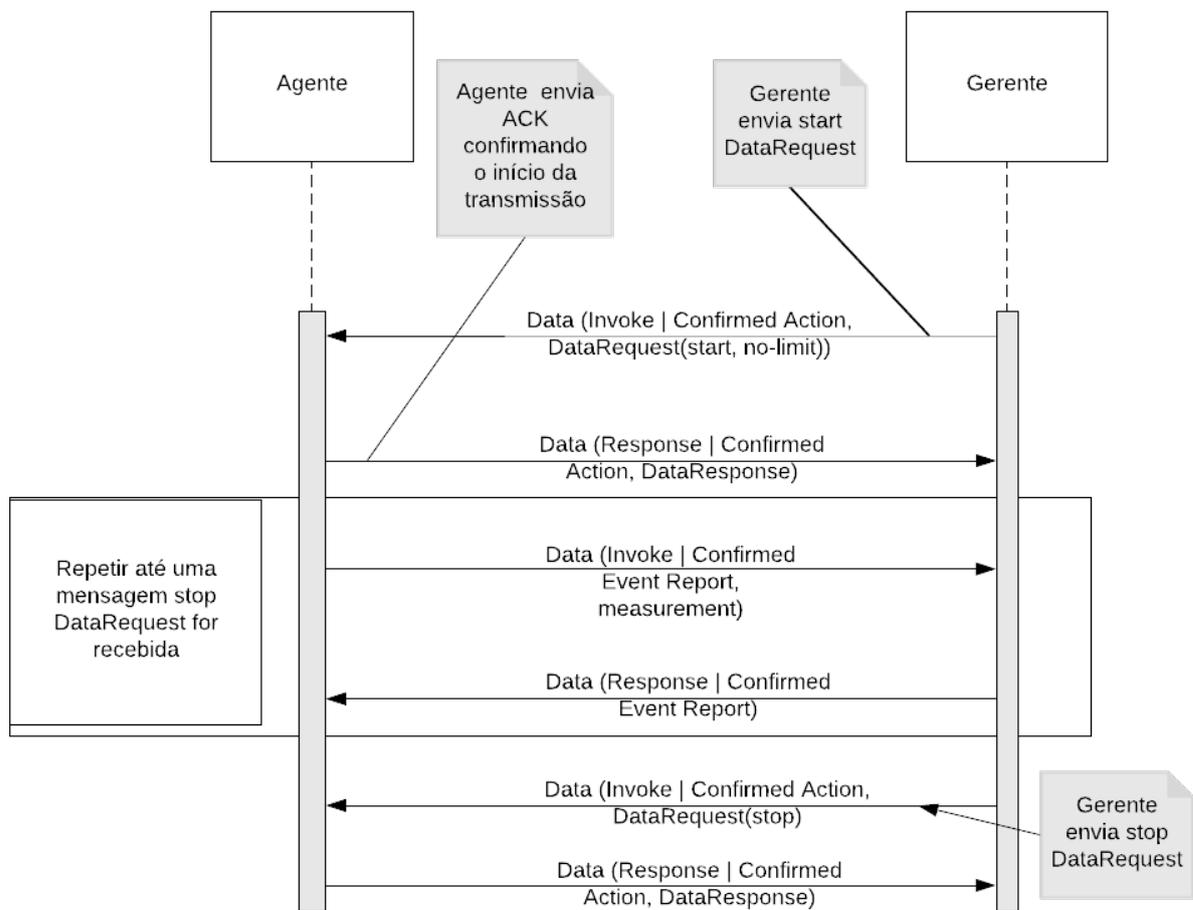


Figura 2.10: *Manager-initiated* utilizando *no limit time mode* (adaptado de [23]).

idade de vida independente, monitor de medicação, monitor de pressão sanguínea, balança, termômetro, medidor de glicose e oxímetro de pulso. Neste grupo, estão pessoas que possuem, por exemplo, sobrepeso e hipertensão, e se não houver tratamento e acompanhamento, podem correr o risco de desenvolver problemas mais graves posteriormente, como doenças crônicas. Dispositivos que fazem o acompanhamento e gerenciamento de tratamentos podem evitar futuros gastos com consultas médicas, além de proporcionar um acompanhamento da saúde do paciente sem que ele precise sair de casa.

Em implementações atuais de PHD, os usuários precisam manualmente transferir as medidas dos dispositivos para portais na Internet ou sistemas, se desejarem compartilhá-las. Espera-se que, no futuro, esses dispositivos possam fazer esse compartilhamento de forma autônoma e inteligente para sistemas de telessaúde.

Em ambientes onde um PHD é compartilhado, como famílias, lares para idosos e hospitais, um único dispositivo fará várias medições de diferentes pacientes. Uma vez que esses dispositivos estão conectados a uma rede, métodos seguros de identificação de

pacientes são necessários. Sendo assim, o X73-PHD permite o transporte de identificação de pacientes nos dados.

2.9 Segurança

Dentro do contexto de comunicação de DPS, a segurança é muito importante, pois informações médicas são consideradas altamente sensíveis. Proteção adequada para estas informações são necessárias tanto para a troca de dados entre agentes e gerentes, quanto para a transmissão pela Internet. Entretanto, até o momento, a família de normas 11073 não provê métodos que garantam a segurança. Ela assume que essa proteção é garantida por outras tecnologias.

Quatro princípios centrais regem a segurança da informação em DPSs: confidencialidade, integridade, disponibilidade e repúdio.

Confidencialidade é definida pela ISO como “assegurar que a informação seja acessível apenas àqueles que possuem autorização”. Dentro do contexto de saúde, uma quebra de confidencialidade pode ocorrer quando uma “escuta” ou vazamento de informação ocorre entre o agente e o gerente.

Integridade significa que um dado não pode ser modificado ou apagado sem autorização. Um vírus, por exemplo, pode quebrar a integridade dos dados no sistema de um agente ou gerente. Verificar se os dados vieram do remetente certo e não de alguém que pretende se passar pela fonte ou garantir que os dados originais alcancem o destino sem alterações são também questões inerentes à integridade dos dados.

Disponibilidade, em segurança da informação, significa que a informação deve estar disponível quando necessária. Assim, os sistemas que armazenam, processam e protegem e os canais de comunicação usados para acessar os dados devem estar funcionando corretamente e de forma confiável.

Repúdio, neste contexto, significa ter clareza de como o processo de obtenção dos dados foi realizado. Por exemplo, em um hospital, ter conhecimento da enfermeira que realizou uma medição e qual aparelho foi utilizado, é de suma importância. Portanto, todo esse processo deve ser registrado.

Garantir a segurança física dos aparelhos é outro ponto crítico quando se trata de aparelhos para a saúde. Agentes podem ser roubados e trocados por outros, os quais podem fornecer dados errôneos para a decisão médica. Outros problemas, como canal não

confinado (conexão sem fio entre agentes e gerentes), uso indevido dos gerentes (pessoas não autorizadas acessam informações nos *smartphones* ou *desktops*) e a não proteção dos dados nos centros especializados, são outras questões críticas de segurança da informação na saúde [23].

Capítulo 3

Trabalhos Relacionados

Com a popularização da família de padrões 11073, vários aperfeiçoamentos e integrações com outros sistemas foram feitos. Nos trabalhos [31], [10] e [37], é proposta a integração do padrão X73-PHD com protocolos de IoT, como *Message Queuing Telemetry Transport* (MQTT) e *Constrained Application Protocol* (CoAP), para serem usados na camada de transporte de PHDs. Esses protocolos auxiliam o compartilhamento de informação de saúde diretamente na Internet, usando baixo consumo energético e poucos pacotes de controle. Nos três trabalhos citados acima, também é apresentado o mapeamento das mensagens X73-PHD para mensagens MQTT e CoAP e a biblioteca Antidote é utilizada na camada de aplicação.

No artigo [35], os autores propõem um sistema interoperável de mensagens para serviços de saúde, onde os agentes e o gerente possam se comunicar com o mínimo de esforço, utilizando protocolos leves de IoT e de forma segura. Os protocolos IoT utilizados são o MQTT e CoAP, e a arquitetura é composta por três componentes: *publisher*, *subscriber* e *message broker*. Toda a comunicação é realizada por intermédio do servidor *message broker*, que recebe requisições de dispositivos *publishers* e *subscribers*. O desempenho desta arquitetura foi comparado com outras arquiteturas *publish* e *subscribe* que utilizam o protocolo MQTT. Detalhes da implementação não foram abordados no artigo.

Outro projeto interessante utilizando o padrão X73-PHD, apresentado em [6], desenvolve uma plataforma de monitoramento remoto interoperável usando *ZigBee Health Care Profile* como camada de transporte e uma solução *Machine-to-Machine* (M2M) para fornecer conectividade de rede de longa distância. Esse trabalho também inclui um aplicativo Web no lado clínico (lado servidor) e usa os padrões e estruturas fornecidos pela IHE [18] e HL7 [12] para garantir interoperabilidade fim-a-fim. Essas duas instituições defendem um mundo no qual todos possam acessar e usar com segurança os dados de

saúde quando e onde precisarem.

O padrão X73-PHD é aplicado em várias áreas da saúde digital, como por exemplo, diagnóstico precoce de doenças crônicas e monitoramento remoto de pacientes em suas residências. No trabalho [24], os autores propõem uma coleção de dispositivos vestíveis X73-PHD, *plug-and-play*, que coletam continuamente dados fisiológicos e do ambiente. Neste projeto, três dispositivos são utilizados: dispositivos sensores, que coletam informações fisiológicas e do ambiente; os *data loggers*, que são dispositivos móveis, que armazenam temporariamente os dados dos sensores; e por fim, as *base stations*, terminais que recebem, processam, armazenam e exibem informações recebidas dos *data loggers*. Como camada de transporte, os autores utilizaram *Bluetooth* e, na camada de aplicação, o padrão X73-PHD.

A versão do padrão X73-PHD para hospitais, X73-PoC [20], fornece um mecanismo para controlar remotamente os agentes. Este mecanismo é definido nas partes X73-10201 e X73-20301. No entanto, o X73-PHD não possui especificações de mecanismos para fazer isso. Assim, em [1] o autor propõe adaptar as capacidades de controle remoto de dispositivos X73-PoC em dispositivos X73-PHD com um *overhead* aceitável e sem custo extra para os fabricantes. Esse mecanismo deve ser instalado no gerente e nos agentes, e o controle remoto consiste em poder alterar as unidades de medida (por exemplo, de quilogramas para libras) diretamente do gerente, de um *smartphone* ou de um computador nas unidades de enfermagem.

Em vários dispositivos de saúde, os protocolos de comunicação são fechados e privados. Em [27], o autor defende que mudar os protocolos já existentes demandará muita complexidade de implementação, além de mudanças no hardware e software. Assim, em seu trabalho, ele propõe um sistema para dispositivos que não seguem o padrão X73-PHD. Este sistema converte o protocolo de comunicação nativo do dispositivo para comunicação X73-PHD. É assumido neste trabalho que o gerente será um dispositivo X73-PHD compatível. A arquitetura deste sistema consiste em um gerente X73-PHD, um dispositivo legado (não padronizado), e um adaptador X73-PHD. Os dispositivos poderão se conectar ao adaptador por conexão cabeada ou sem fio. Os dispositivos suportados para este adaptador são: FORA TD-3250G (medidor de pressão arterial e medidor de glicose) e TD-8002 (medidor de pressão, medidor de glicose, termômetro e oxímetro de pulso). Detalhes de implementação não foram fornecidos, apenas diagramas do funcionamento do adaptador.

Outro trabalho interessante é o desenvolvimento de um adaptador para dispositivos

que não seguem o padrão X73-PHD. Como mostrado em [25], os autores sugerem um dispositivo intermediário entre os sensores não padronizados e o gerente. Este dispositivo faz a tradução das mensagens vindas desses sensores para mensagens X73-PHD, e então transmite os dados traduzidos para um gerente padronizado. Esse trabalho não deixa claro como foi feito o mapeamento das mensagens e em quais dispositivos o adaptador proposto funciona.

Em [2], os autores apresentam um *framework* de código aberto, chamado *GreenCastalia*, para simulações que têm por objetivo analisar *energy-harvesting*. O foco do projeto *GreenCastalia* é simular descarga real da bateria dos dispositivos. As principais modificações no código Castalia original foram feitas no módulo de Gerenciador de Recursos. Também neste módulo, os autores acrescentam os sistemas de coleta de energia que fornecem um modelo de bateria mais realista.

Em [30], é feita uma revisão do padrão X73-PHD e também é proposto um *middleware* para uma plataforma de telemonitoramento. Este *middleware* segue o modelo de comunicação OEP, implementando as definições ASN.1 e codificação BER. De acordo com o artigo, essa implementação pode ser usada como um testbed para testes de comunicação X73-PHD, porém, nenhum código foi disponibilizado e, nenhuma informação sobre o código ser livre ou aberto foi dada no artigo.

Capítulo 4

Proposta

Este trabalho propõe o uso do padrão ISO/IEEE 11073 para simulação de aplicações de saúde digital em cenários de WBAN [8], [29]. Conforme o padrão, são propostos um componente gerente e outro agente na Camada de Aplicação do simulador Castalia. O envio de dados de leitura sempre é feito por um sensor, porém este sensor pode tomar a iniciativa de enviar ou o gerente pode solicitar leituras a um sensor. Neste trabalho os dois modos de comunicação são suportados, *agent-initiated* e *manager-initiated*.

O agente é o primeiro e o único a enviar a mensagem de *Association request* para solicitar uma associação ao gerente antes de iniciar o envio de leituras. Quando não há mais leituras para serem enviadas, o agente envia a mensagem *Association release* para encerrar a associação. No modo *manager-initiated*, o gerente pode solicitar tanto o início do envio das leituras quanto o fim. O gerente envia uma mensagem *Stop Request* para o agente solicitando o encerramento da transmissão.

A proposta de camada de aplicação para dispositivos pessoais de saúde desta dissertação oferece cinco tipos de dispositivos como agentes: oxímetro de pulso, medidor de glicose, termômetro, medidor de pressão e eletrocardiograma (ECG de 3 derivações). O oxímetro transmite a frequência cardíaca em batimentos por minuto e a porcentagem de saturação de oxigênio na hemoglobina arterial (SpO₂). O medidor de glicose transmite o nível de glicose, isto é, a concentração de açúcar no sangue em miligramas por decilitro (mg/dL). O termômetro transmite a temperatura corporal em Celsius (°C). O medidor de pressão envia uma mensagem composta por três medidas arteriais, sistólica, diastólica e a média arterial em milímetros de mercúrio (mmHg). O ECG envia oitenta amostras do potencial elétrico do coração em milivolts (mV) por pacote. Cada amostra em *mV* deve ser convertida para valores escalares inteiros, antes de serem enviadas. Os valores abso-

lutos inferior e superior e os valores escalares inferior e superior são transmitidos na fase de configuração. Todos os dados dos agentes são produzidos de forma pseudo-aleatória, exceto o ECG que transmite valores reais obtidos a partir de uma base de dados [9].

Se um agente usa o modo de comunicação confirmado e o ACK de uma leitura enviada ao gerente não é recebido, o padrão indica que o agente deve realizar uma nova associação com o gerente para transmitir novamente a leitura. Em um cenário de rede sem fio, a perda de quadros pode ocorrer com frequência, por isso, o modo de comunicação com confirmação pode levar a uma sobrecarga de pacotes de controle com necessidade de estabelecer uma nova associação toda vez que um ACK for perdido na WBAN.

Para contornar este problema, esta dissertação de mestrado propõe um novo modo de comunicação, chamado de *Modo de Retransmissão*, como extensão ao padrão X73-PHD. As próximas subseções detalham os três modos de operação e discutem como a proposta deste trabalho pode melhorar o funcionamento de PHDs em cenários WBANs.

Alguns parâmetros de simulação são oferecidos para permitir a personalização da simulação. A lista completa com todos os parâmetros pode ser consultada na Tabela 4.1. Uma breve introdução a cada um dos parâmetros é listada a seguir:

Tabela 4.1: Parâmetros configuráveis de simulação.

| Nome do parâmetro | Valor |
|--|---|
| ApplicationName | “Manager”, “Agent” |
| hubNode | valor double |
| application_type | “pulseoximeter”, “thermometer” “bloodpressure”, “basicECG” “glucometer” |
| measurements_per_second | valor double |
| confirmed_event | true, false |
| retransmitPacket | true, false |
| maxNumOfRetransmission | valor inteiro |
| timeOutToRetransmitPacket | valor double |
| dynamicTimeout | true, false |
| managerInitiated | true, false |
| managerInitiateMode | “singleMode”, “timePeriodMode” “noTimePeriodMode” |
| managerInitiatedTime | double Usado em “timePeriodMode” |
| numberOfReceived MeasurementsToSendStop | integer Usado em “noTimePeriodMode” |

- *ApplicationName*: Utilizado para definir a aplicação que o nó utilizará. A aplicação “Manager” é utilizada pelo nó gerente e “Agent” pelos nós agentes (sensores e

atuadores). Apenas um nó da rede pode executar a aplicação “Manager”.

- *hubNode*: Valor numérico que identifica o nó hub. Este nó deve ser o mesmo que executa a aplicação “Manager”.
- *application_type*: Parâmetro utilizado apenas em nós “Agent” para especificar o tipo do agente. Atualmente os tipos disponíveis são: “pulseoximeter”, “thermometer”, “bloodpressure”, “basicECG” e “glucometer”.
- *measurements_per_second*: Valor numérico onde é definida a quantidade de leituras que serão enviadas por segundo.
- *confirmed_event*: Valor booleano, onde *true* significa que o nó solicitará uma confirmação de recebimento para cada leitura enviada, equivalente ao Modo com Confirmação. Se o valor for definido para *false*, as leituras serão enviadas sem nenhum mecanismo de confiabilidade, equivalente ao Modo sem Confirmação.
- *retransmitPacket*: Valor booleano. Quando definido como *true*, retransmite os pacotes que tiveram o ACK perdido, equivalente ao Modo de Retransmissão, proposto nesta dissertação. Os parâmetros *maxNumOfRetransmission* e *timeOutToRetransmitPacket* devem ser definidos caso este modo seja escolhido.
- *maxNumOfRetransmission*: Valor numérico inteiro, que define o máximo de tentativas que o nó agente utilizará para retransmitir um pacote.
- *timeOutToRetransmitPacket*: Valor numérico, representado em segundos, que define o tempo que um nó agente aguarda pelo ACK do gerente.
- *dynamicTimeOut*: Valor booleano. Quando *true*, calcula o valor de *timeOutToRetransmitPacket* dinamicamente, se *false* usa os valores especificados pelo usuário. Caso não seja especificado nenhum valor, é usado 0,4 segundos como valor padrão.
- *managerInitiated*: Valor booleano. Caso seja *true*, o gerente tomará a iniciativa de solicitar leituras para os agentes. O parâmetro *managerInitiatedMode* deve ser definido caso este modo seja escolhido.
- *managerInitiatedMode*: Aceita as seguintes *strings*: “singleMode”, “timePeriodMode”, “noTimePeriodMode”. Usado para definir o modo de solicitação que o gerente fará para o agente. Caso o modo “timePeriodMode” seja escolhido, o parâmetro *managerInitiatedTime* deve ser definido. Caso o modo “noTimePeriodMode” seja escolhido, o parâmetro *numberOfReceivedMeasurementToSendStop* deve ser definido.

- *managerInitiatedTime*: Valor numérico, representado em segundos. Usado pelo gerente para solicitar leituras ao agente por um tempo pré-determinado.
- *numberOfReceivedMeasurementToSendStop*: Valor numérico, inteiro. Usado pelo gerente para identificar quando enviar uma mensagem de *Stop Request* para o agente.

4.1 Modos de Comunicação

4.1.1 Modo sem Confirmação

A Figura 4.1(a) mostra um diagrama de sequência do procedimento de troca de mensagens de uma operação normal de um agente com configuração padrão e usando eventos de envio de leituras sem confirmação. O agente pretende se associar com o gerente pela primeira vez e então envia um *Association request*. Quando o gerente recebe o *Association request*, ele verifica se já houve associações feitas previamente com esse agente. Se for a primeira associação, o gerente envia uma mensagem de *Association response* e, na sequência uma mensagem de *Get attributes* solicitando as informações de configuração deste agente. Logo após, o agente envia sua configuração e atributos para iniciar a transmissão de leituras. Quando todas as leituras tiverem sido enviadas, o agente envia uma mensagem de *Association release* e o gerente responde com uma mensagem *Association release response* para confirmar o término desta associação.

Como este modo não possui nenhuma mecanismo de confiabilidade, não há garantia que os leituras foram entregues no gerente. Este modo é ativado definindo-se nas configurações de simulação *confirmed_event = false* e *retransmitPacket = false*. Pela sua natureza, este modo tende a realizar apenas uma associação durante as transmissões. Uma segunda associação só será feita se uma mensagem de *Abort* é recebida do gerente e ainda existirem leituras a serem enviadas.

4.1.2 Modo com Confirmação

A Figura 4.1(b) descreve o diagrama de sequência da operação de um agente com configuração padrão usando eventos de envio de leitura com confirmação. O processo inicial é idêntico ao explicado na Seção 4.1.1. A diferença é que para cada leitura que o agente envia, o gerente confirma com uma mensagem ACK. Após enviar uma leitura, o agente espera a confirmação por três segundos, conforme *timeouts* definido no padrão X73-PHD. Se nada chegar durante esse período, o agente envia uma mensagem de interrupção, *As-*

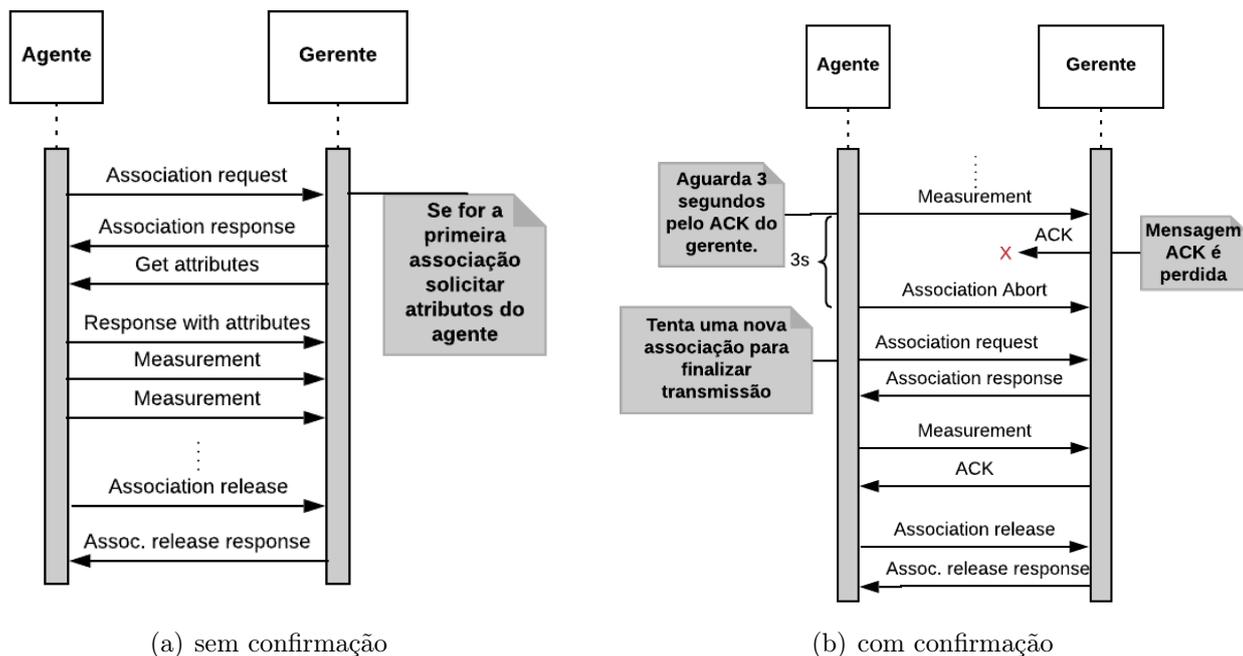


Figura 4.1: Diagrama de sequência para os modos de operações com e sem confirmação de uma aplicação X73-PHD.

sociation abort, e transita para o estado de máquina não associado. Se ainda existirem leituras para serem enviadas, o agente deve tentar uma nova associação.

Neste modo, é possível ter a garantia do recebimento das leituras no gerente. Em contrapartida, realizar uma nova associação a cada ACK não recebido de uma leitura enviada pode causar grandes atrasos na transmissão. Para ativar este modo, basta fazer *confirmed_event = true*.

4.1.3 Proposta do Modo de Retransmissão

O padrão X73-PHD assume que, em dispositivos reais, haverá uma camada de transporte confiável disponível para comunicação, como o *Bluetooth Health Device Profile* [3]. Em uma WBAN, pode não haver camada de transporte confiável disponível, já que os sensores têm poucos recursos de memória, processamento e bateria. Desta forma, esta dissertação propõe um novo modo de comunicação, baseado em *stop-and-wait*, para realizar retransmissões de pacotes perdidos ou não confirmados pelo gerente.

Isso reduz a troca desnecessária de vários pacotes de controle no processo de associação. Ao invés de fazer uma nova associação quando um ACK não é recebido, retransmite-se o pacote até que um ACK seja recebido ou seja atingido um número máximo de tentativas

de retransmissões. O usuário define o número n de retransmissões e, então, aquela leitura será retransmitida até n vezes ou até que um ACK seja recebido. Se o gerente receber uma mensagem duplicada, ele retransmitirá imediatamente o ACK que não foi recebido pelo agente, ou seja, o último ACK enviado.

Na fase de associação, nossa proposta reduz o *timeout* de 10 (indicado pelo padrão) para 0,4 segundos, caso algum pacote de controle seja perdido. Após a associação, o dispositivo usando o *Modo de Retransmissão*, envia sua primeira leitura, aguarda o período de *timeOutToRetransmitPacket* conforme definido pelo usuário no arquivo de configuração da simulação. Se nenhuma resposta for recebida, o pacote é retransmitido no máximo *maxNumOfRetransmission* vezes, conforme definido também pelo usuário, ou até que uma mensagem de reconhecimento do gerente seja recebida. Na Figura 4.2, é apresentado o diagrama de sequência para os casos onde uma mensagem ACK e uma leitura são perdidos. Se todas as tentativas definidas em *maxNumOfRetransmission* forem utilizadas, então uma nova associação é feita com o gerente e a variável *maxNumOfRetransmissionCounter* é redefinida para 0. Esse processo continua até o agente realizar 3 associações, conforme definido no padrão X73-PHD. Na quarta tentativa, o agente envia uma mensagem de *Association abort* e volta para estado de máquina *não associado*.

No Capítulo 6.2, é avaliada a eficiência desta proposta usando como métrica a redução no envio de pacotes de controle, o número de novas associações feitas por cada dispositivo, a quantidade de pacotes retransmitidos e a número de leituras transmitidas com sucesso para o gerente. Vale ressaltar que esta é uma nova proposta e não está presente no padrão X73-PHD.

4.1.4 Modo *Manager-initiated*

Esta dissertação também oferece o modo *manager-initiated*, além do *agent-initiated* para comunicação entre gerente e agentes. A Figura 4.3 descreve um diagrama de sequência do procedimento de mensagens correspondente a uma operação de um agente usando o modo *manager-initiated*. O procedimento inicial é o mesmo explicado nas seções anteriores. A diferença é o pacote de controle *Data Request* enviado pelo gerente. Este pacote pode solicitar dados de medição aos agentes de três maneiras: *Single*, *Time Period* and *No Time Limit mode*.

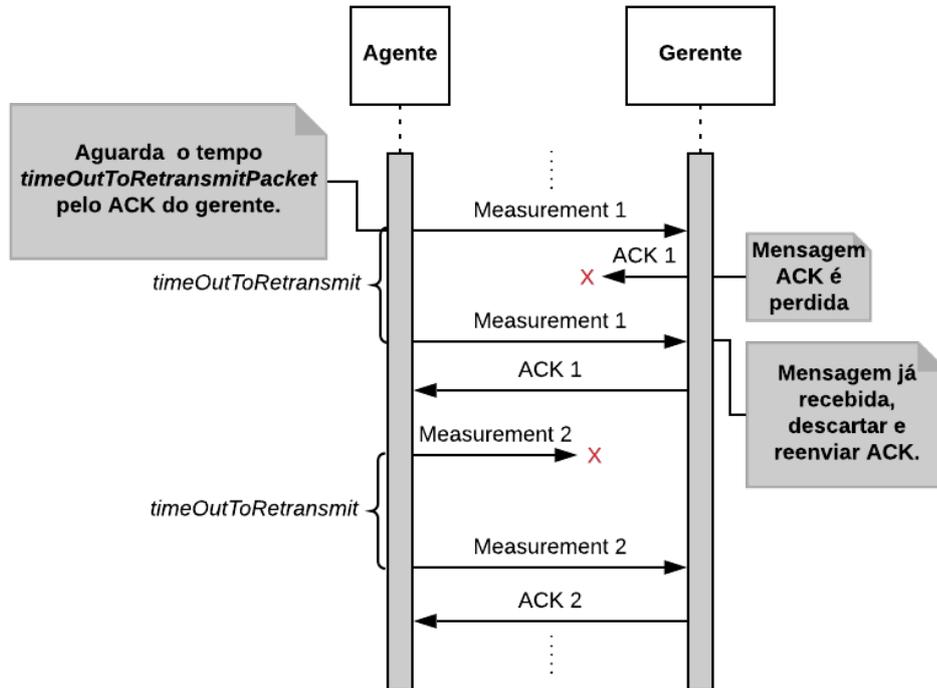


Figura 4.2: Diagrama de sequência do modo de retransmissão demonstrando o comportamento quando uma mensagem ACK e uma leitura são perdidas.

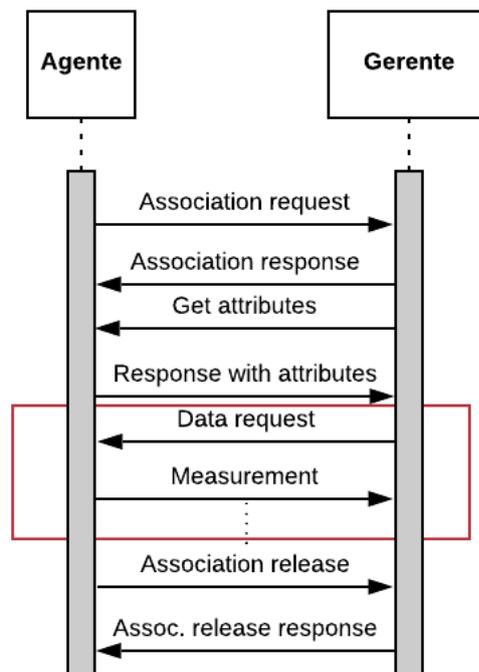


Figura 4.3: Diagrama de sequência de operação de uma aplicação *manager-initiated*.

4.2 Arquitetura do Sistema

O principal objetivo deste trabalho é tornar as simulações de redes corporais sem fio mais realistas. Para isso, são propostos dispositivos de saúde que se comportam como um dispositivo real, que segue o padrão X73-PHD. O simulador Castalia foi escolhido como plataforma de simulação, já que oferece a camada MAC IEEE 802.15.6 já implementada, desenvolvida para comunicação em WBANs. Para auxiliar no desenvolvimento dos dispositivos pessoais de saúde, foi utilizada a biblioteca Antidote.

Assim, uma camada de aplicação foi implementada no simulador Castalia. Para esta aplicação se comportar como uma aplicação de um dispositivo médico real, é preciso enviar as mensagens para o Antidote, decodificá-las, decidir a próxima ação e retornar uma resposta para o simulador. Essa interação é possível através de *plug-ins* criados para a biblioteca Antidote.

A arquitetura do sistema é exibida na Figura 4.4. O *plug-in*, que auxilia na comunicação com a biblioteca, é implementado também na camada de aplicação, sendo independente das camadas de roteamento e MAC do simulador. Toda e qualquer mensagem trocada entre a biblioteca e o simulador obrigatoriamente passa pelo *plug-in* de comunicação. Todas as tarefas de controle de máquina de estados, tipos de serviço e codificação e decodificação das APDUs, são realizados na biblioteca Antidote.

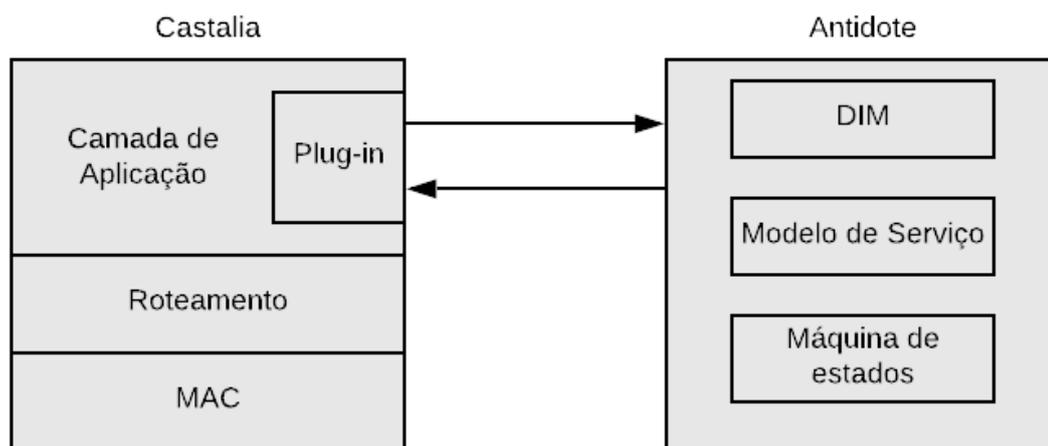


Figura 4.4: Arquitetura em alto nível da proposta.

4.2.1 Simulador Castalia

O Castalia é um simulador para redes de sensores e redes corporais sem fio. É baseado na plataforma OMNeT++, e pode ser usado por pesquisadores e desenvolvedores que

querem testar seus algoritmos distribuídos e protocolos em um canal sem fio e modelo de rádio realistas [5].

Na Figura 4.5, pode-se ver a organização interna de um nó no Castalia. As setas sólidas significam trocas de mensagens e as tracejadas chamadas de funções. Os nós se comunicam por meio do canal *wireless*, nunca diretamente.

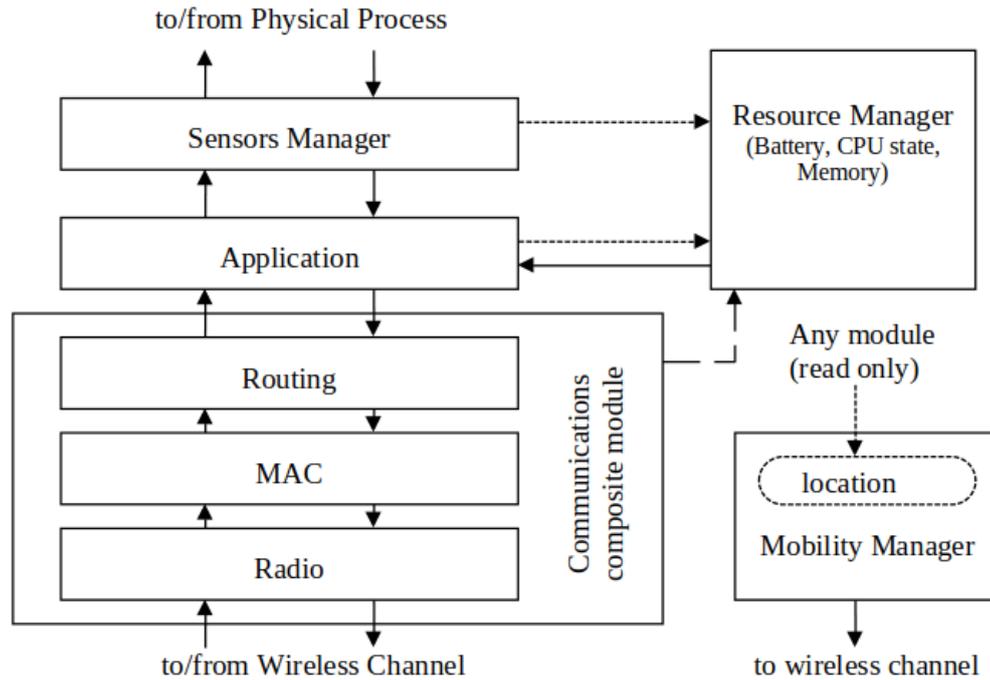


Figura 4.5: Organização interna do simulador Castalia, fonte: [5]

O Castalia é organizado em camadas. Para criar um novo módulo de roteamento, é preciso criar um novo diretório dentro de `Castalia/src/node/communication/routing`. Já para criar um novo módulo de aplicação, cria-se um novo diretório em `Castalia/src/node/application`. Neste trabalho, foram desenvolvidos dois novos módulos de aplicação, um para o agente outro para o gerente. Todos os códigos relacionados com a aplicação do agente encontram-se em `Castalia/src/node/application/agent` e do gerente em `Castalia/src/node/application/manager`.

Cada uma das aplicações possui um *plug-in* para se comunicar com a biblioteca Antidote. No módulo do agente, além do *plug-in*, também existe o módulo que gera os dados dos dispositivos médicos. Esses dados são gerados de forma pseudo-aleatória, mas em intervalos que façam sentido para cada tipo de dispositivo. Por exemplo, o termômetro produz valores entre 36.5 até 40.5 graus celsius.

4.2.2 Biblioteca Antidote

A biblioteca Antidote é portátil e usa a linguagem ANSI-C para criar um código limpo e multiplataforma. Para realizar a comunicação entre o Antidote e outros sistemas, como o Castalia, é necessário desenvolver e adaptar *plug-ins* de comunicação no Antidote que permitam essa comunicação. Como esta biblioteca foi desenvolvida para ser usada em dispositivos médicos reais, modificações foram necessárias para que o funcionamento dentro do simulador ocorresse como em um dispositivo real. Inúmeras partes do código foram alteradas, porém os principais módulos alterados foram: *communication*, *encoders*, *agent* and *manager*.

Antidote é uma implementação do protocolo *Optimized Exchange Protocol*, desenvolvido pela Signove, como parte da *SigHealth Platform*¹. Esta biblioteca é a primeira implementação de código aberto desse protocolo. Esta biblioteca também permite a comunicação com dispositivos certificados da *Bluetooth Continua Health Alliance* [32].

A biblioteca como um todo é portátil. Porém, dependências externas para uma plataforma específica são necessárias quando *plug-ins* de transportes são utilizados, como USB, TCP/IP, sockets e etc.

Na Figura 4.6, é exibida a organização interna em alto nível da Biblioteca Antidote. O coração da biblioteca é o módulo de comunicação, onde toda a “regra de negócio” é realizada. É neste módulo que o protocolo OEP é implementado. A partir deste módulo, são feitas todas as chamadas para os *plug-ins* de comunicação (*Bluetooth*, USB, sockets). A biblioteca também oferece módulos codificadores que convertem dados dos agentes para o formato XML/JSON. Este codificador não faz parte da especificação do padrão X73-PHD. No módulo *Device Specialization* se encontram as configurações de cada especialização implementada. A biblioteca também oferece *Application Programming Interfaces* (APIs) que são utilizadas pelas aplicações.

Para fazer o seu trabalho, o Antidote precisa se comunicar com o mundo exterior utilizando *plug-ins*. Um *plug-in* de comunicação deve fornecer alguns serviços para o Antidote, através de uma interface escrita na linguagem C. Algumas destas tarefas são:

- Notificar a biblioteca quando uma conexão está aberta e quando é fechada;
- Criar um ID único para cada dispositivo/conexão;

¹*SigHealth* é uma plataforma de monitoramento remoto de pacientes e gestão de dados utilizando dispositivos pessoais sem fio para a saúde.

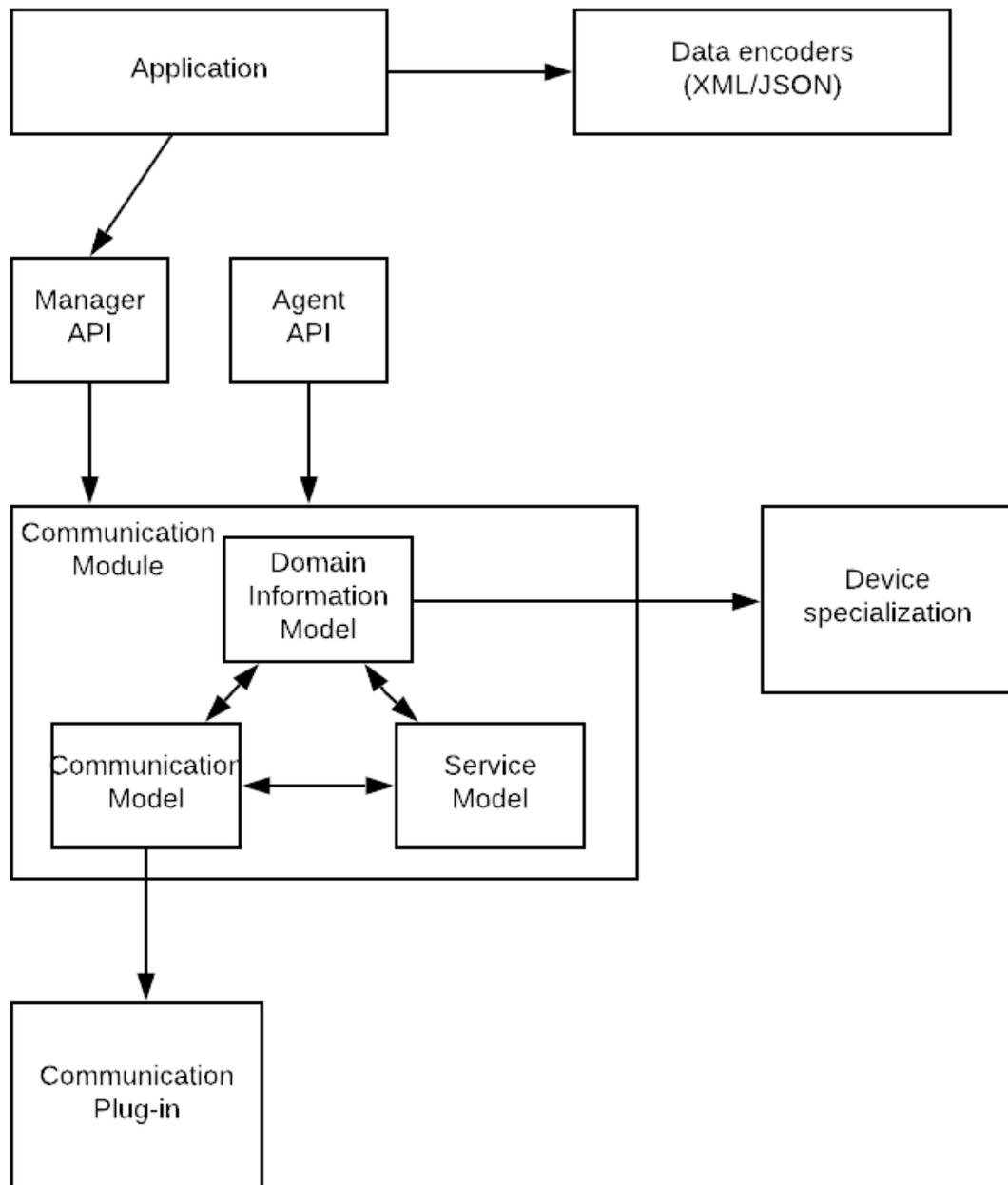


Figura 4.6: Síntese da organização interna da Biblioteca Antidote (Adaptado de [38]).

- Finalizar uma conexão quando solicitado pela biblioteca;
- Notificar quando um novo pacote é recebido.

Na prática, um *plug-in* precisa implementar mais tarefas que as quatro supracitadas. Um *plug-in* não precisa, necessariamente, ser escrito dentro da estrutura de código da biblioteca antidote *src/communication/plugin*. Um *plug-in* é apenas uma lista de *callback functions* que são registradas na biblioteca. O funcionamento independe de onde o *plug-in* é implementado, uma vez registrado, a Biblioteca realiza seu trabalho.

O próximo capítulo descreve o *plug-in* implementado neste trabalho para o simulador Castalia.

Capítulo 5

Implementação

5.1 Castalia Plug-in

O *plug-in* proposto permite que o Simulador Castalia e a biblioteca Antidote conversem entre si através de um módulo que chamamos de Castalia Plug-in ou `plugin_castalia.cc`. Dentro do Castalia, pode-se chamar qualquer função da biblioteca Antidote e recuperar qualquer informação dela, mas transmitir os dados do Castalia para o Antidote não é uma tarefa simples. Para resolver isso, a referência das funções criadas no módulo Castalia *plug-in* foi passada para a biblioteca Antidote. Assim, em qualquer momento que o Antidote necessite transferir os dados, ele utiliza essas funções.

Há um conjunto de *callback functions* que são passadas para o Antidote durante o processo de inicialização de cada agente. Essas funções são passadas para o módulo de comunicação do Antidote como ponteiros para funções. Desta forma, sempre que o módulo de comunicação precisar recuperar ou enviar alguma mensagem, ele apenas aciona a *callback function* correspondente.

No Algoritmo 5.1, é apresentada a função na qual é feita a passagem por referência das funções do `castalia_plugin.cc` para a estrutura *CommunicationPlugin*. Esta estrutura é enviada, posteriormente, para a Biblioteca Antidote. A implementação completa de cada método do arquivo `castalia_plugin.cc` pode ser encontrado em <https://github.com/conqlima/11073PhdApplication>.

```

int plugin_network_castalia_agent_setup(CommunicationPlugin *plugin
    , ... )
{
    plugin->network_init = network_init;
    plugin->network_wait_for_data = network_castalia_wait_for_data;
    plugin->network_get_apdu_stream = network_get_apdu_stream;
    plugin->network_send_apdu_stream = network_send_apdu_stream;
    plugin->network_disconnect = network_disconnect;
    plugin->network_finalize = network_finalize;

    return CASTALIA_ERROR_NONE;
}

```

Algoritmo 5.1: Registrando as funções do Castalia Plug-in na biblioteca Antidote.

Após a estrutura *CommunicationPlugin* receber todas as funções necessárias, é preciso passar esta estrutura para a biblioteca. Isto é feito chamando a função `agent_init`, que além desta estrutura, recebe outras informações importantes para seu funcionamento como o tipo de especialização e a estrutura `CONTEXT_ID`, como ilustrado no Algoritmo 5.2. Esta estrutura é uma identificação única daquele nó dentro da biblioteca Antidote.

```

//Initialize the agent
agent_init(CONTEXT_ID, plugins, specialization, ...);

```

Algoritmo 5.2: Passagem da estrutura *CommunicationPlugin* de um agente para a biblioteca Antidote.

O mesmo é feito no gerente, Algoritmo 5.3, porém com apenas dois argumentos, *CommunicationPlugin* e `CONTEXT_ID`.

```

//Initialize the manager
manager_init(CONTEXT_ID, plugins);

```

Algoritmo 5.3: Passagem da estrutura *CommunicationPlugin* de um gerente para a biblioteca Antidote.

Além das funções do *plug-in*, outras funções são passadas para a biblioteca Antidote. *Listeners* são funções que notificam qualquer mudança no estado de máquina de um agente ou gerente. Elas também notificam quando uma APDU é recebida ou enviada. Essas funções são úteis para realizar ações necessárias quando um agente entra, por exemplo, no

estado *connected*. Assim, essas funções oferecem um controle maior sobre o funcionamento dos dispositivos. No Algoritmo 5.4, é apresentada a estrutura que o Antidote oferece para os desenvolvedores de *Plug-ins*. Essas funções devem ser criadas e passadas por referência como feito no Algoritmo 5.1.

```
typedef struct AgentListener
{
    /**
     * Called after device is connected
     */
    void (*device_connected)(Context *ctx, const char *addr);
    /**
     * Called after device is disconnected
     */
    void (*device_disconnected)(Context *ctx, const char *addr);
    /**
     * Called after device is operational
     */
    void (*device_associated)(Context *ctx);
    /**
     * Called after device is not operational
     */
    void (*device_unavailable)(Context *ctx);
    /**
     * Called after timeout occurs
     */
    void (*timeout)(Context *ctx);
} AgentListener;
```

Algoritmo 5.4: *Listeners* utilizados na biblioteca Antidote.

A estrutura *ContextId* carrega um *id* único, no nível de aplicação, para cada nó da rede. Esta estrutura é muito importante, pois, a partir dela, é possível acessar informações do dispositivo, como a máquina de estado atual. A atribuições dos *ids* únicos é de inteira responsabilidade do *plug-in*, pois é este que de fato estabelece o canal de comunicação.

Foram desenvolvidos dois *plug-ins*, um para os dispositivos agentes e outro para o gerente.

A Figura 5.1 descreve como o Antidote recebe as mensagens do Castalia. As mensagens que chegam são copiadas para uma variável com escopo externo no Simulador Castalia - a variável externa é visível para os módulos do Castalia e Castalia Plug-in. Como o módulo de comunicação do Antidote possui ponteiros para as funções do módulo Castalia Plug-in, ele pode recuperar as mensagens usando esses ponteiros (*callback functions*).

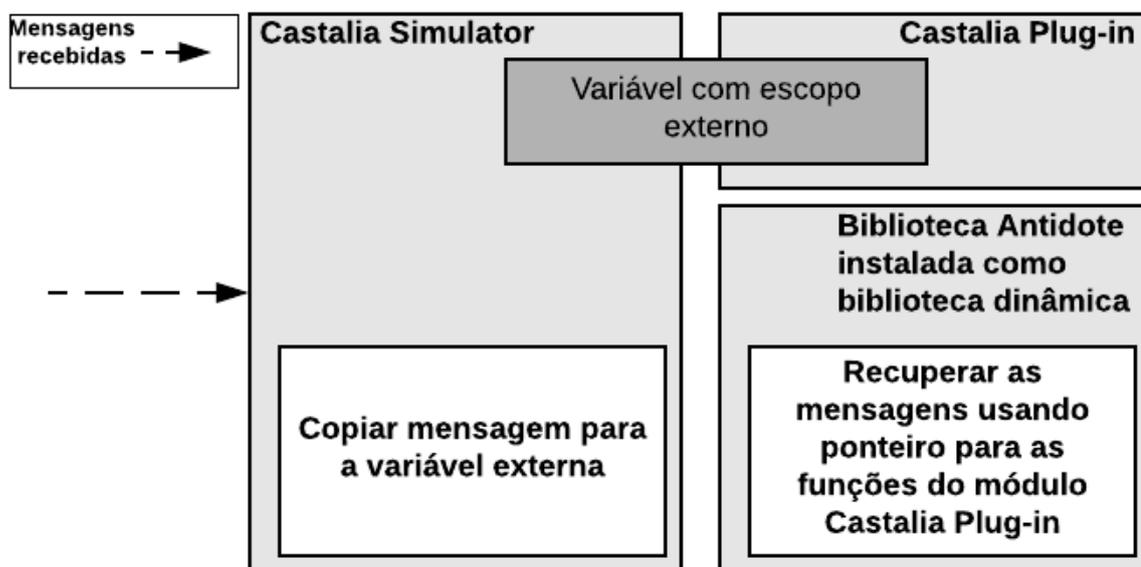


Figura 5.1: Diagrama de como é realizada a comunicação entre a Biblioteca Antidote e o Simulador Castalia, utilizando um plug-in de comunicação.

5.2 Mudanças no módulo de comunicação

O módulo de comunicação é um dos módulos mais importantes na biblioteca Antidote. A inicialização, a finalização, as fases da máquina de estados de todos os nós são controladas por este módulo. Em um cenário real, cada dispositivo possui seu próprio módulo de comunicação, executando sua própria cópia do código. Mas em uma simulação executada em uma única máquina é um pouco diferente. Somente um módulo de comunicação manipula todos os nós, uma vez que o Antidote é instalado no sistema operacional como uma biblioteca dinâmica.

Para garantir o funcionamento adequado do módulo de comunicação, a identificação do nó é necessária em cada chamada de função. Desta forma sabe-se com qual nó se está trabalhando e aplica-se a ação para apenas este nó, sem alterar o estado dos outros nós.

Por exemplo, se um nó transitar do estado de máquina não associado para associando, sem a modificação proposta, essa ação afetaria todos os nós, mesmo aqueles que não deveriam alterar o estado da máquina.

A Figura 5.2 ilustra a modificação do sistema. Em cada função do módulo de Comunicação, foi inserido o novo parâmetro *node id*. Desta forma, o módulo de comunicação sempre conhece o nó em questão.

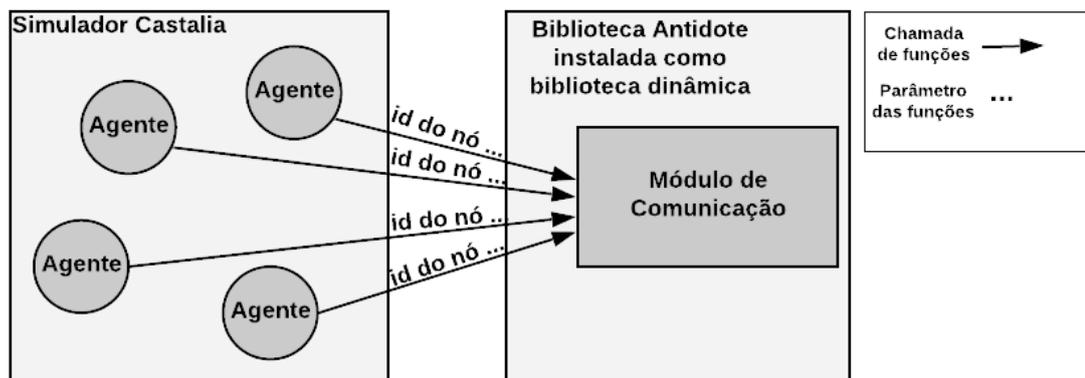


Figura 5.2: Módulo de comunicação manipulando vários nós. Cada seta representa uma função sendo chamada do módulo de comunicação.

Capítulo 6

Avaliação

Este capítulo apresenta resultados de simulações que utilizam a proposta desta dissertação. Todos os resultados apresentados neste capítulo são relativos à camada de aplicação implementada por este trabalho. As simulações foram feitas em um computador com 8GB de memória RAM, Intel Core i5-7200U e sistema operacional Ubuntu 18.04.1 LTS.

6.1 Cenário de simulação

Monitoramento remoto e independência para a terceira idade é um dos casos de uso do padrão X73-PHD. Os sensores e atuadores propostos para este caso de uso são: medidor de pressão sanguínea, termômetro, medidor de glicose, oxímetro de pulso e um ECG. Neste trabalho utilizamos um paciente hipotético que possui diabetes e hipertensão e que precisa ser monitorado em sua casa.

As simulações têm duração de 43201s, sendo o primeiro segundo apenas para a configuração da camada MAC (e.g., requisições de conexões da camada MAC). Foram executadas 15 repetições da simulação com intervalo de confiança de 95%.

A Figura 6.1 mostra a topologia utilizada nas simulações. O hub, que é o nó gerente, está localizado no quadril direito, um nó sensor em cada pulso e um em cada tornozelo e um sensor no meio do peito. Utilizando essas posições, há a vantagem de usar um canal sem fio, que já possui um modelo de perda de sinal, entre cada par de dispositivos, pré-definido com experimentos reais, conforme apresentado em [4].

Neste trabalho, foram simulados seis cenários, três *agent-initiated* e três *manager-initiated*. Os três primeiros cenários *agent-initiated* são:

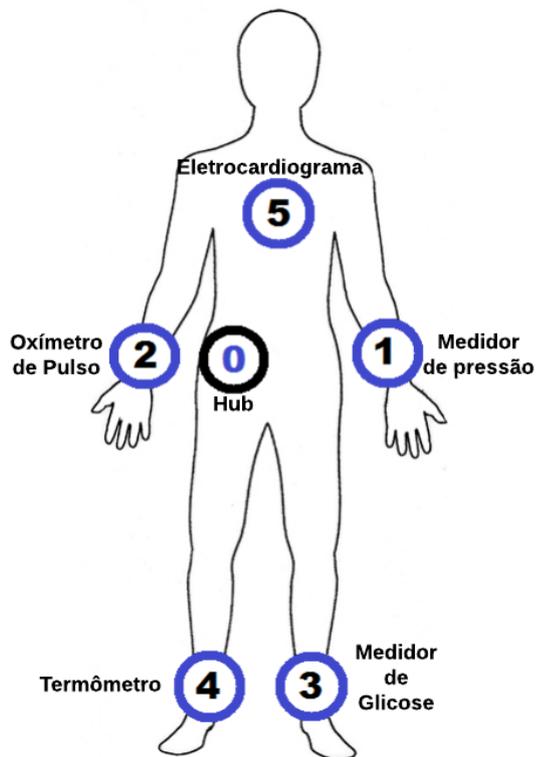


Figura 6.1: A topologia utilizada na simulação.

- Modo sem Confirmação, em que os agentes tomam a iniciativa de enviar as medições ao gerente sem confirmação.
- Modo com Confirmação, onde os agentes aguardam por três segundos o ACK do gerente, quando transmitem leituras. Se um ACK não for recebido, o agente terá que tentar estabelecer uma nova associação para finalizar a transmissão dos pacotes de medição.
- Modo de Retransmissão, em que os agentes esperam um ACK do gerente durante o período de tempo definido pelo usuário em *timeOutToRetransmitPacket* e, caso ele não seja recebido, o pacote é retransmitido para *maxNumOfRetransmission* também definido pelo usuário. Se todas as novas tentativas forem feitas e uma confirmação não for recebida, uma nova associação será feita.

Os três últimos cenários *manager-initiated* são:

- *Single-Mode*, o gerente solicita apenas uma medição para o agente.
- *Time Period Mode* permite que o gerente solicite que os agentes enviem dados de medições durante um período de tempo.

- e finalmente *No Time Period Mode*, em que o gerente solicita dados de medições aos agentes continuamente até que a associação seja interrompida ou o gerente envie uma mensagem *Stop Data Request*.

Foi utilizada a camada MAC definida pelo padrão IEEE 802.15.6 (WBAN) [22] com mapa de perda de sinal e modelo temporal para o canal sem fio fornecidos pelo simulador Castalia. A taxa de transmissão na camada física de 1024 kb/s e o rádio utilizado se enquadra na proposta do padrão IEEE 802.15.6, trabalhando com -15dBm para transmissão.

As configurações da simulação são definidas da seguinte forma. O tempo total de simulação é de 43201 segundos (12 horas). O nó zero é o hub e usa a aplicação *Manager*. Os demais nós são agentes e usam a aplicação *Agent*. O monitor de pressão arterial transmite uma medição a cada 15 minutos, totalizando 48 medições a serem enviadas em nossa simulação. O oxímetro envia uma leitura a cada 15 segundos, então, 2880 leituras devem ser enviadas. O medidor de glicose transmite uma medida a cada 5 minutos, ou seja, 144 em 43201 segundos. O termômetro envia dados de temperatura a cada 3 minutos, assim, 240 no total. Neste trabalho, assumimos que ECG é um dispositivo que recebe sinais de todos os eletrodos implantados no corpo e transmite esses sinais para o gerente. Então, ele irá transmitir 80 amostras do potencial elétrico do coração em milivolts a cada 0,8 segundos, o que totaliza 54 mil pacotes para serem enviados nos 43201s de simulação [9].

Nos três modos *agent-initiated*, os nós tentarão transmitir todos os pacotes descritos no parágrafo anterior, mas nos modos *manager-initiated*, isso não é verdade. No *Single-Mode*, por exemplo, é transmitido apenas um pacote de dados de medição. No *Time Period Mode*, foram definidos 21600 segundos (6 horas) de transmissão de pacotes de leituras, que é a metade das medições totais transmitidas nos modos *agent-initiated*. E para o modo *No Time Period*, o gerente envia uma mensagem de *Stop Request* quando 10 leituras são recebidas de cada nó.

6.2 Resultados

Nesta seção serão apresentados resultados utilizando os modos *agent-initiated* e *manager-initiated*. Para o Modo de Retransmissão, utilizamos os seguintes valores: *timeOutToRetransmitPacket* = 0,4s, 0,6s, 0,8s e 1s e *maxNumOfRetransmission* = 2, 4, 6, 8 e 10. Primeiro serão apresentados resultados utilizando os valores fixos de *timeOutToRetrans-*

mitPacket e depois com o os valores sendo dinamicamente calculados. Na Tabela 6.1, são apresentados os resultados para a taxa de entrega de pacotes de leitura no gerente para todos os valores de *timeOutToRetransmitPacket* e *maxNumOfRetransmission* avaliados.

| | Medidor de Pressão Arterial | Oxímetro | Medidor de Glicose | Termômetro | ECG |
|------------------------------|-----------------------------|-------------------|--------------------|------------|----------------|
| retries=2,timeout=0.4 | 44 | 2684 | 125.2 | 224.46667 | 36067 |
| retries=2,timeout=0.6 | 47.06667 | 2871 | 142.86667 | 239.8 | 39831.93333 |
| retries=2,timeout=0.8 | 47.06667 | 2870.73333 | 142.93333 | 240 | 39720.86667 |
| retries=2,timeout=1 | 47.13333 | 2870.46667 | 143.26667 | 240 | 38242.13333 |
| retries=4,timeout=0.4 | 47.06667 | 2871.86667 | 133.8 | 240 | 41374.6 |
| retries=4,timeout=0.6 | 47.06667 | 2871 | 143.13333 | 240 | 39821.26667 |
| retries=4,timeout=0.8 | 47.06667 | 2870.6 | 143.06667 | 240 | 39700.4 |
| retries=4,timeout=1 | 47.06667 | 2870.53333 | 143.06667 | 240 | 38208 |
| retries=6,timeout=0.4 | 47.06667 | 2872 | 133.46667 | 240 | 41371.6 |
| retries=6,timeout=0.6 | 47.06667 | 2871 | 143.06667 | 240 | 39818.26667 |
| retries=6,timeout=0.8 | 47.06667 | 2870.46667 | 143.06667 | 240 | 39701.06667 |
| retries=6,timeout=1 | 47.06667 | 2870.6 | 143.06667 | 240 | 38204.2 |
| retries=8,timeout=0.4 | 47.06667 | 2872 | 133.46667 | 240 | 41369 |
| retries=8,timeout=0.6 | 47.06667 | 2871 | 143.06667 | 240 | 39818.26667 |
| retries=8,timeout=0.8 | 44.06667 | 2682.6 | 133.8 | 224.4 | 37097.8 |
| retries=8,timeout=1 | 47.06667 | 2870.6 | 143.06667 | 240 | 38204.2 |
| retries=10,timeout=0.4 | 47.06667 | 2872 | 133.46667 | 240 | 41369 |
| retries=10,timeout=0.6 | 47.06667 | 2871 | 143.06667 | 240 | 39818.26667 |
| retries=10,timeout=0.8 | 47.06667 | 2870.46667 | 143.06667 | 240 | 39701.06667 |
| retries=10,timeout=1 | 47.06667 | 2870.6 | 143.06667 | 240 | 38204.2 |

Tabela 6.1: Número de pacotes de leitura entregues no gerente variando os parâmetros de *timeOutToRetransmitPacket* e *maxNumOfRetransmission*.

Para ser possível comparar os resultados do Modo de Retransmissão com os outros modos, foi escolhido o melhor resultado dentre todas as combinações apresentadas na Tabela 6.1. O critério para a escolha deste resultado foi o desempenho do ECG, o qual apresenta a maior taxa de perda de pacotes e que, com essa combinação apresentou a maior taxa de entrega entre todas as outras combinações. O resultado escolhido pode ser visto em destaque na tabela. Esta mesma combinação de *retries* e *timeout* será utilizada para todos os resultados do Modo de Retransmissão e não apenas para a taxa de entrega de pacotes de leitura.

6.2.1 Agent-initiated

Nesta seção serão utilizados os parâmetros fixos de **retries=4**, **timeout=0.4** para o Modo de Retransmissão, com exceção da Figura 6.4 que utiliza **retries=10**, **timeout=0.4** para apresentar o comportamento quando o máximo de **retries** é utilizado. O primeiro resultado que será discutido é o total de leituras entregues no gerente usando

os modos *agent-initiated*. É possível ver na Figura 6.2, que o número de pacotes entregues foi maior para todos os agentes quando o modo de retransmissão é usado. Para alguns agentes, como o oxímetro, o modo sem confirmação entregou quase todos os pacotes, embora não forneça confiabilidade. Como esperado, o modo com confirmação melhora a confiabilidade, mas em contrapartida, requer muitas reassociações. O modo de retransmissão melhorou o resultado em todos os cenários, entregando quase 100% de todas as mensagens, evitando o tempo de espera de um novo *handshake* de associação e retransmitindo as mensagens mais rapidamente.

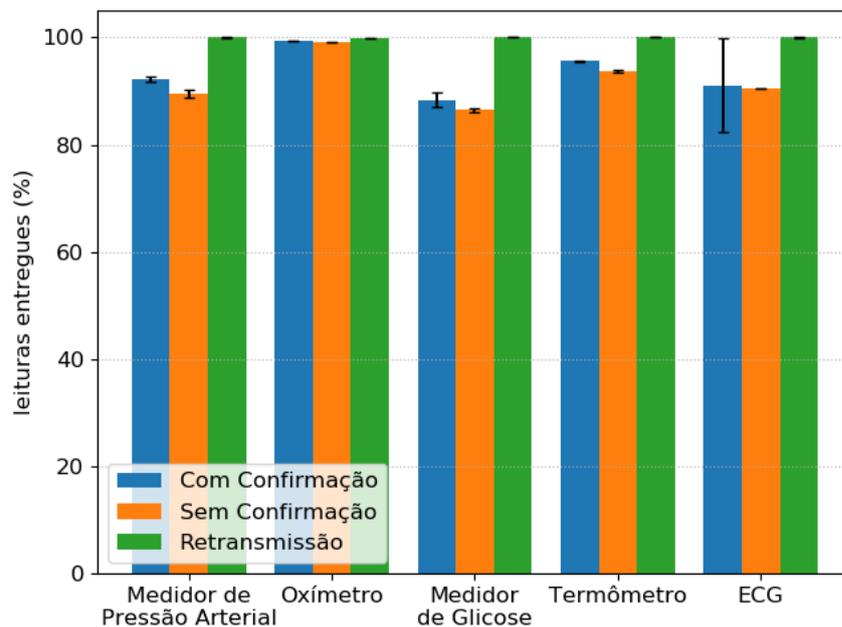


Figura 6.2: Pacotes de leituras entregues com sucesso no gerente usando *Agent-initiated mode*.

O número de pacotes de controle trocados entre um nó agente e o gerente é crucial nesse cenário, principalmente no caso de uma nova associação devido a um ACK não recebido. O procedimento de associação envolve no máximo a troca de quatro pacotes para um novo nó e um mínimo de dois pacotes, quando os atributos do agente são conhecidos previamente.

O número médio de pacotes de controle trocados entre cada nó por modo de operação é mostrado na Figura 6.3. Observe que os nós transmitem diferentes números de pacotes na mesma simulação. Enquanto o monitor de pressão arterial transmite 48 medições em 12 horas, o ECG tem que transmitir 54.000 no mesmo período de tempo. No modo de retransmissão, menos pacotes de controle são trocados, quando comparado ao modo confirmado. Embora para o ECG, na Figura 6.3, o modo de retransmissão tenha usado mais pacotes de controle do que o modo confirmado, o modo de retransmissão entregou

mais pacotes de leitura do que o modo confirmado, como já visto na Figura 6.2. O modo sem confirmação, como esperado, é aquele que transmite menos pacotes de controle.

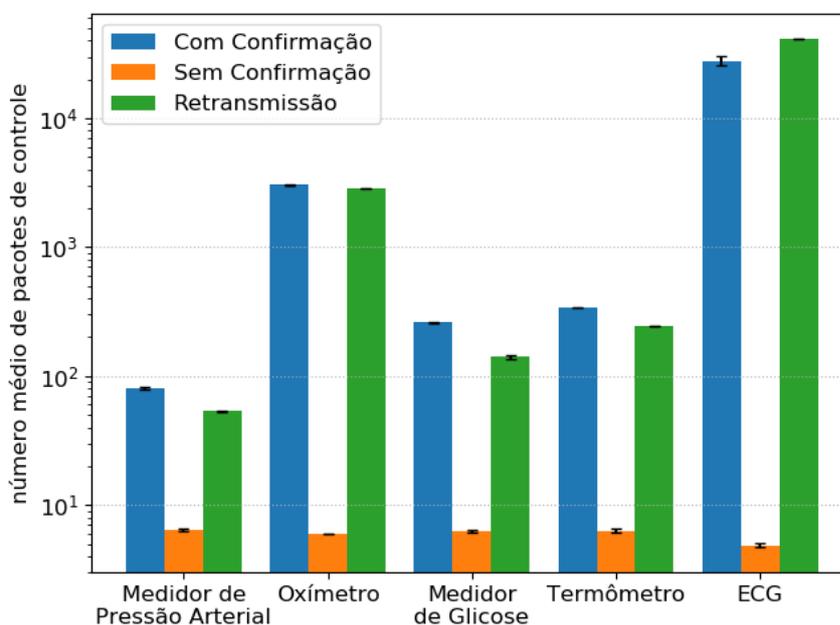


Figura 6.3: Número médio de pacotes de controle enviados e recebidos utilizando *Agent-initiated mode*.

O número de associações feitas por cada nó é alto no modo com confirmação, pois é feita uma nova associação após cada perda de pacote. A Figura 6.4 mostra o número médio de associações que cada nó fez nos modos *agent-initiated*. Como esperado, o modo com confirmação tem a maior média de tentativas de associação, enquanto o modo sem confirmação fez apenas uma associação. O modo de retransmissão tenta uma nova associação após todas as tentativas de reenviar uma mensagem ou se o agente receber uma mensagem *Abort* do gerente. O ECG, que tem o pior enlace sem fio entre todos os nós, requer quase 2000 novas associações para finalizar a transmissão de 54000 medições no modo confirmado. O medidor de glicose possui uma média menor que uma associação pois em uma das 15 simulações, ele não conseguiu estabelecer nenhuma associação.

Na Figura 6.5, é possível ver o número médio de tentativas de retransmissão feitas no modo de retransmissão pela camada de aplicação. A maioria dos pacotes é entregue com sucesso na primeira tentativa e mais de 1000 pacotes usam uma tentativa de retransmissão. Na sétima tentativa de retransmissão, os valores tendem a zero.

A Figura 6.6 mostra que a maioria dos pacotes tem uma latência média de 180 ms no modo de retransmissão e no modo com confirmação e de 30 ms no modo sem confirmação. Os modos de retransmissão e com confirmação possuem uma latência maior pois requerem *timeouts* nas transmissões dos pacotes na camada de aplicação. O tempo é calculado a

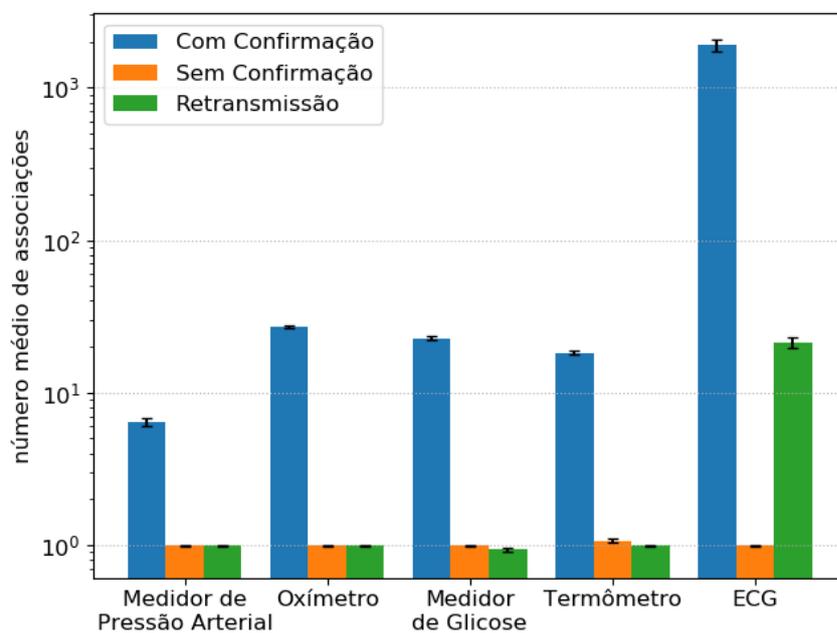
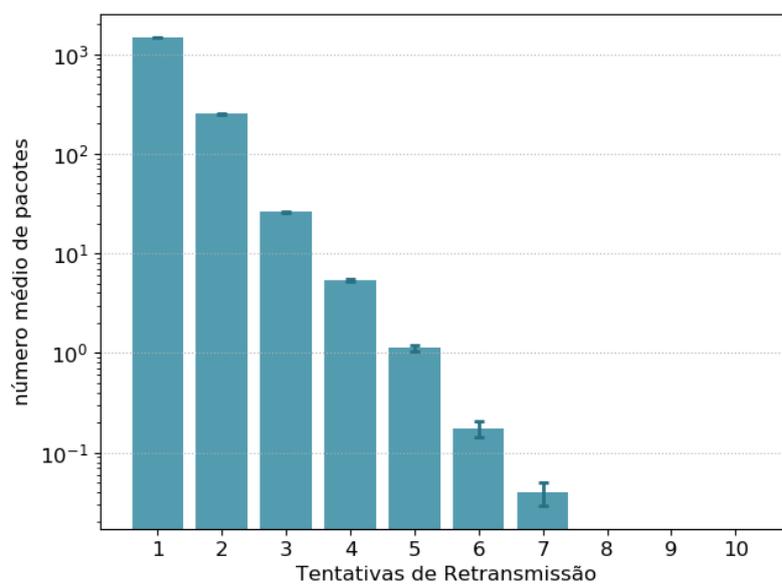


Figura 6.4: Média de associações feitas por cada nó.

Figura 6.5: Média de retransmissões de pacotes feita na camada de aplicação utilizando $\text{retries}=4$, $\text{timeout}=0.4$.

partir do momento da criação do pacote, na camada de aplicação do agente, até o momento em que é recebido na camada de aplicação do gerente.

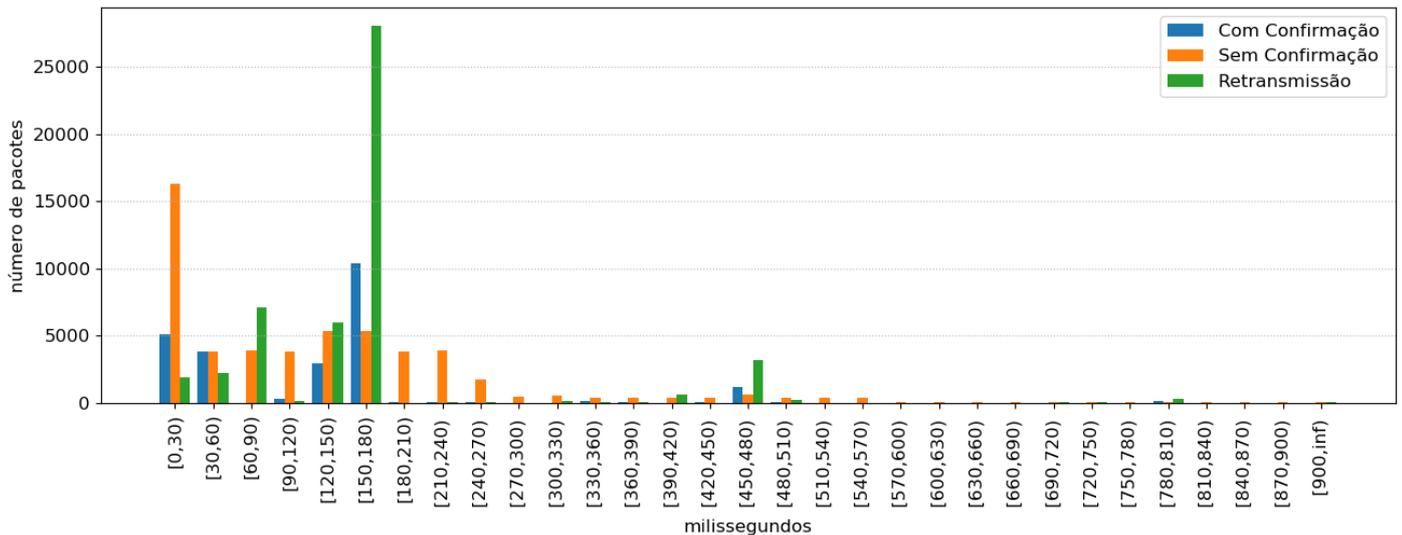


Figura 6.6: Latência na camada de aplicação.

6.2.2 Manager-initiated

Os três modos *manager-initiated* (*No Time Mode*, *Single Mode*, *Time Period Mode*) utilizam em todos as simulações transmissões sem confirmação. Conforme explicado na Seção 6.1, o número de pacotes transmitidos é diferente para cada modo, devido à sua natureza e objetivo.

O número médio de pacotes de leituras entregues no modo *manager-initiated* podem ser vistos na Figura 6.7. No *Single Mode*, é enviado apenas um pacote, assim, se um dispositivo recebe um pacote, ele já recebeu 100% dos pacotes. Já os dispositivos oxímetro e medidor de glicose tiveram uma perda deste único pacote em algumas das quinze repetições da simulação, por isso a queda no gráfico no modo *Single Mode*. No modo *No Time Mode*, o gerente envia uma mensagem de *Stop* a cada dez leituras recebidas de cada agente. Porém, cada agente teve que enviar mais de dez pacotes para que o gerente recebesse os dez pacotes esperados. O *Time Period Mode* entregou mais de 80% dos pacotes em todos os dispositivos.

Na Figura 6.8, é possível observar que todos os dispositivos utilizaram em média apenas uma associação para finalizar a transmissão dos dados. Apesar de ter sido utilizado o modo sem confirmação nas transmissões, alguns dispositivos utilizaram mais de uma associação para finalizar a transmissão dos dados de leitura. Isso se deu devido a erros

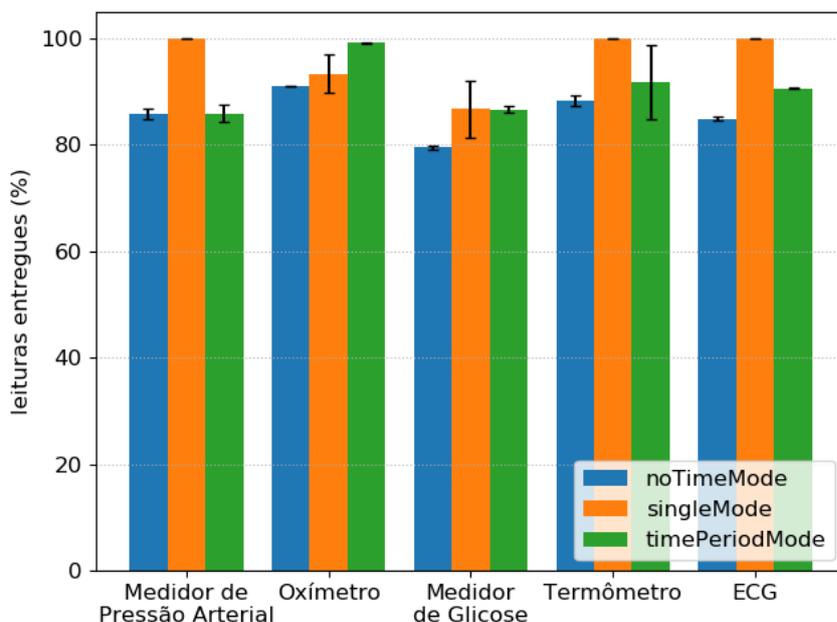


Figura 6.7: Pacotes de leitura entregues com sucesso por nó usando *Manager-initiated mode*.

durante o processo de associação. Por exemplo, a perda do pacote *Response With Attributes*, enviado por cada agente antes de iniciar a transmissão dos dados, pode causar uma nova reassociação.

Na Figura 6.9, é mostrado o número médio de pacotes de controle enviados/recebidos por cada agente. O modo *No Time Period Mode* teve a maior taxa de pacotes de controle trocados pois este modo possui dois pacotes de controle extras que não existem em nenhum outro modo do padrão X73-PHD. No modo *No Time Period Mode*, quando o gerente decide parar a transmissão de dados de leitura, ele deve enviar o pacote *DataRequest(stop)* para o agente. Logo após, o agente confirma o fim da transmissão enviando para o gerente a resposta *DataResponse*. Esses dois pacotes de controle são exclusivos deste modo e podem ser perdidos, causando sua retransmissão e conseqüentemente o aumento na taxa de pacotes de controle.

É possível observar na Figura 6.10 a latência dos pacotes nos três modos *manager-initiated*. Majoritariamente, o gráfico é composto por pacotes do modo *Time Period Mode*, pois dentre os três, este é o modo que transmite o maior número de pacotes.

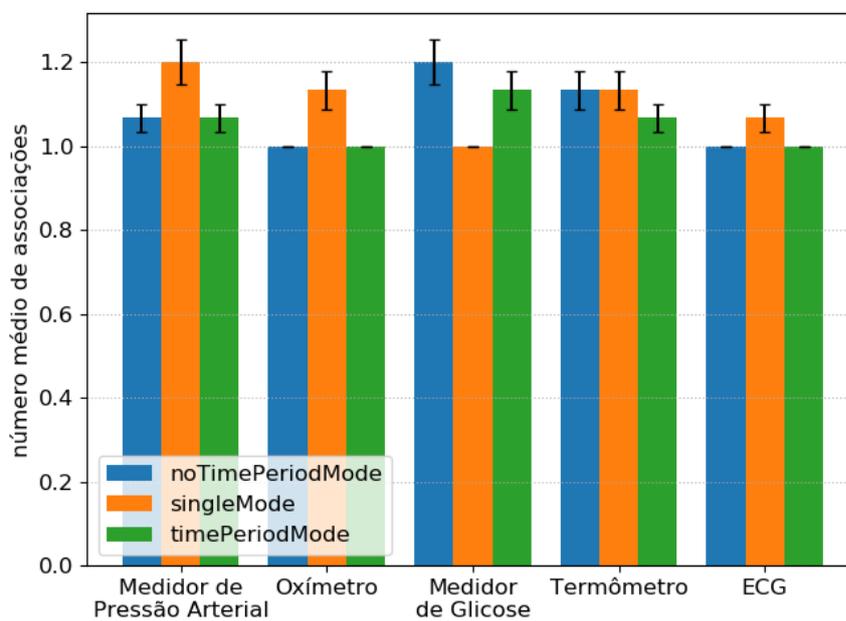


Figura 6.8: Número médio de associações feitas por cada dispositivo durante cada simulação.

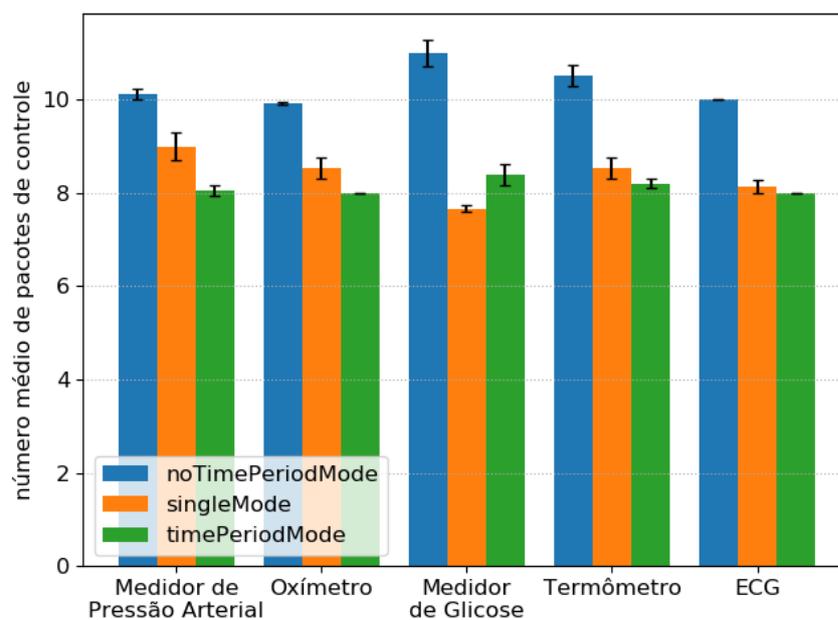


Figura 6.9: Número médio de pacotes de controle enviados/recebidos por cada agente.

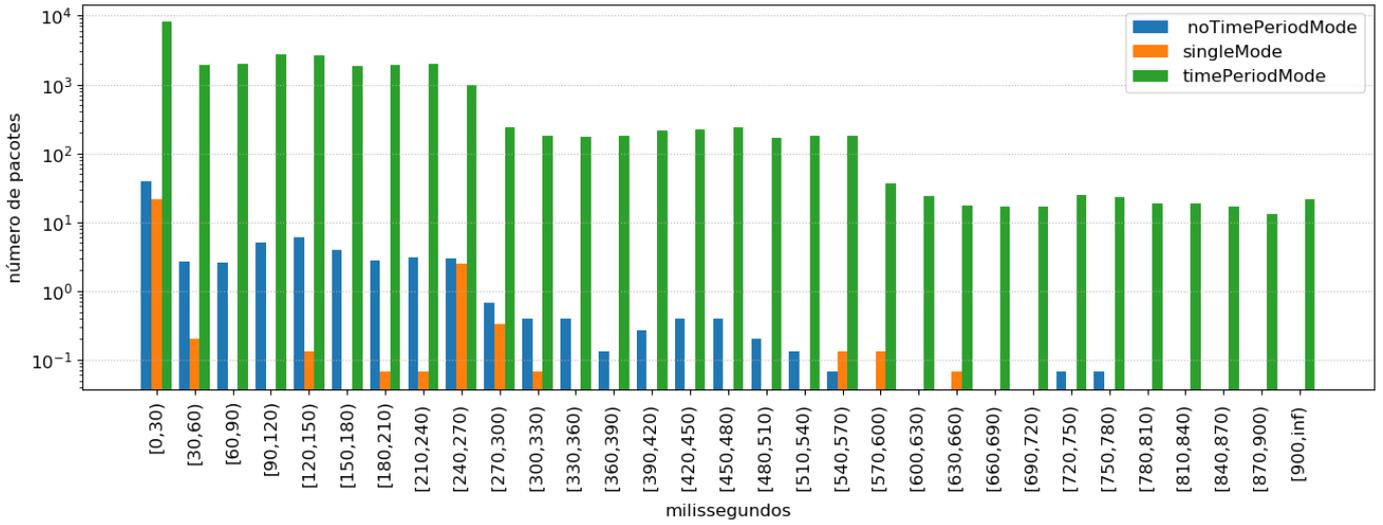


Figura 6.10: Latência da camada de aplicação para os três modos *manager-initiated*.

6.2.3 Modo de Retransmissão com *Timeout* Calculado Dinamicamente

Tendo em vista a natureza dinâmica da rede e a diversidade de cenários, é proposta a utilização de *timeouts* calculados dinamicamente para que o tempo de espera pelo ACK seja ajustado de forma automática. Nesta seção, será analisado somente o comportamento do Modo de Retransmissão utilizando o *timeOutToRetransmitPacket* calculado dinamicamente, baseando-se na equação que é utilizada pelo *Transmission Control Protocol* (TCP), onde:

$$EstimatedRTT = (1 - \alpha) \times EstimatedRTT + \alpha \times SampleRTT \quad (6.1)$$

$$DevRTT = (1 - \beta) \times DevRTT_{prev} + \beta \times |SampleRTT - EstimatedRTT| \quad (6.2)$$

$$Timeout = EstimatedRTT + 4 \times DevRTT \quad (6.3)$$

sendo $\alpha = 0.125$, $\beta = 0.25$ e *SampleRTT* o valor atual do *Round-trip Time* (RTT). Os valores iniciais para *EstimatedRTT* e *DevRTT_prev* são definidos conforme [36]:

$$EstimatedRTT = SampleRTT$$

$$DevRTT_{prev} = \frac{SampleRTT}{2}$$

Este *timeout* dinâmico é utilizado apenas para retransmissões de pacotes de leitura, não se aplicando às outras fases, como associação ou finalização de uma transmissão. O *timeout* é sempre atualizado no envio de leituras. O valor do *timeout* não é atualizado em casos de erro, como na retransmissão de um pacote de leitura, recebimento de um *Association Abort* ou na realização de uma nova associação. Os *timeouts* utilizados nestas duas últimas

fases são fixos, e os valores utilizados são os propostos no padrão X73-PHD. Neste teste, todos os dispositivos agentes tomam a iniciativa de enviar leituras (*agent-initiated*).

Na Figura 6.11, é apresentada a taxa de recebimento de pacotes de leitura no gerente. A posição dos nós e o número de pacotes enviados por cada dispositivo é igual ao descrito na Seção 6.1. A taxa de entrega de leituras se mostrou estável durante as 15 repetições, isto é, o Modo de Retransmissão conseguiu entregar no gerente quase todos os pacotes enviados.

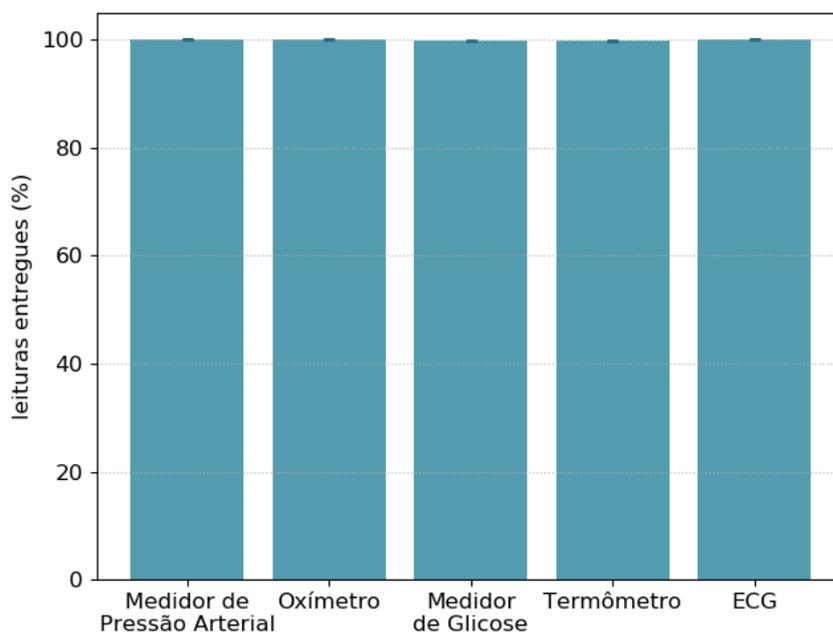


Figura 6.11: Pacotes de leituras entregues com sucesso no gerente usando o Modo de Retransmissão com *timeOutToRetransmitPacket* calculado dinamicamente.

O número de associações feitas por cada dispositivo pode ser visto na Figura 6.12. O medidor de glicose e o termômetro fizeram em média uma associação durante as simulações. O ECG realizou em média quase mil associações durante cada simulação. Este número de associações do ECG tende a diminuir conforme o número de tentativas de retransmissões cresce. Pode-se ver este resultado comparando o número de associações feitas quando foi utilizado o *timeOutToRetransmitPacket* fixo e com quatro retransmissões, ilustrado na Figura 6.4. O ECG tende a realizar muitas associações devido à qualidade baixa do enlace sem fio que ele possui com o gerente e, o número de pacotes que ele tem que enviar é o maior entre os cinco dispositivos.

O número de pacotes de controle recebidos/enviados por cada dispositivo é exibido na Figura 6.13. Os dois nós que mais enviam pacotes de leituras são consequentemente os que mais enviam pacotes de controle, como é o caso do oxímetro e ECG. É considerado neste resultado todo e qualquer pacote que não seja pacote de leitura, incluindo ACKs,

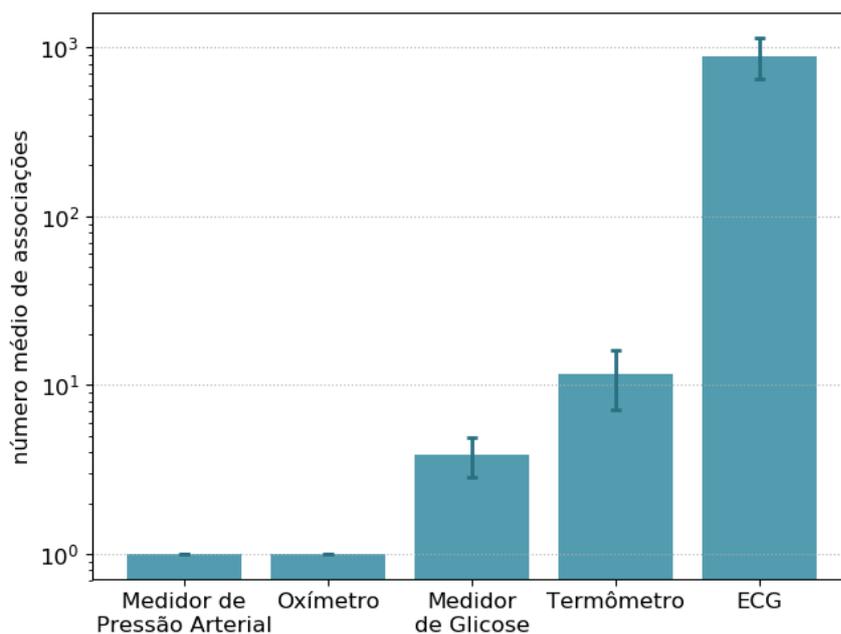


Figura 6.12: Número médio de associações feitas por cada nó utilizando o Modo de Retransmissão com *timeOutToRetransmitPacket* calculado dinamicamente.

que são contabilizados como pacotes de controle.

Diferente dos outros cenários de testes, foram utilizadas no máximo oito tentativas de retransmissões dos pacotes de leitura. Na Figura 6.14, é apresentado o número de pacotes e suas retransmissões. O ECG, sozinho, foi responsável pelo aumento no número de pacotes retransmitidos cinco e seis vezes. Naturalmente, a maioria dos pacotes foram retransmitidos pelo menos uma vez, e a minoria até 8 vezes.

Na Figura 6.15 é apresentada a média dos valores de *timeOutToRetransmitPacket* utilizados por cada um dos agentes no Modo de Retransmissão. Utilizar o *timeout* calculado dinamicamente pode oferecer mais comodidade para quem realiza as simulações e economizar alguns *gigabytes* de dados de *trace*.

O Modo de Retransmissão melhorou significativamente a taxa de entrega de pacotes de leituras em relação ao Modo com Confirmação, além de oferecer um mecanismo de entrega confiável de dados, na ausência de uma camada de transporte confiável. O Modo de Retransmissão chegou a entregar 13% a mais de pacotes, no caso do medidor de glicose. A taxa de entrega em todos os casos gira em torno de 99%. Para pequenas quantidades de dados a serem transmitidos, recomenda-se sempre o usar o modo *agent-initiated*, já que uma grande quantidade de nós pode sobrecarregar o gerente.

O Modo *manager-initiated* é ideal para grandes quantidade de dados, visto que ele possui a capacidade de controlar o fluxo de dados vindo dos agentes. Este modo tem

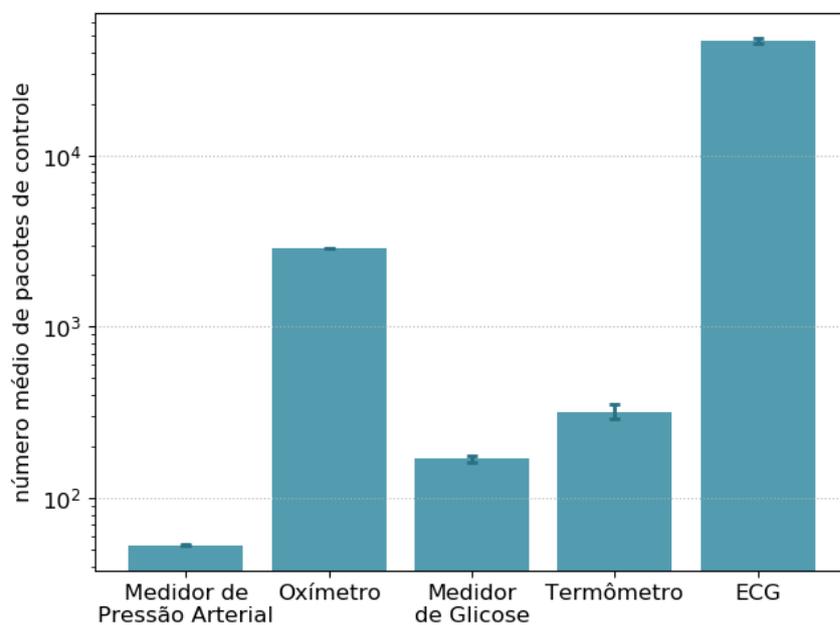


Figura 6.13: Pacotes de controle enviados/recebidos utilizando o Modo de Retransmissão com *timeOutToRetransmitPacket* calculado dinamicamente.

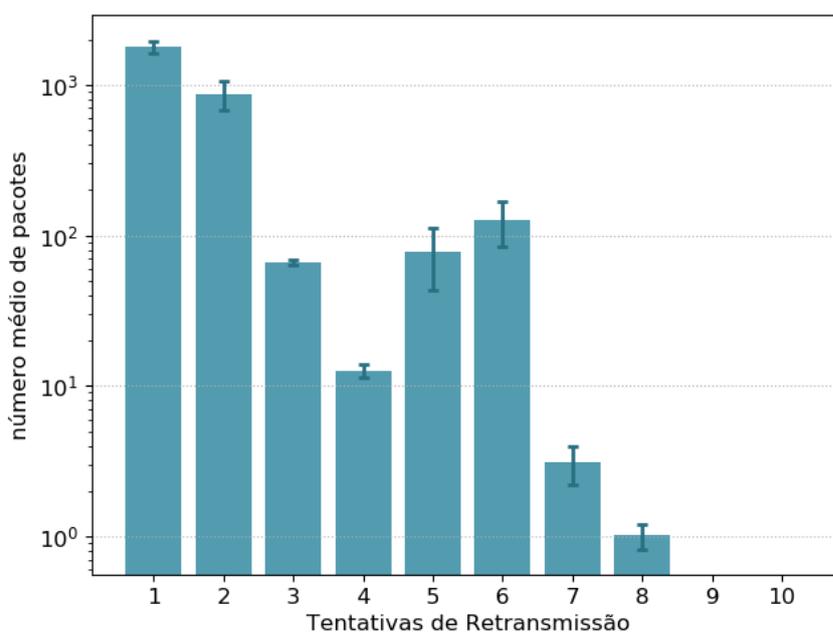


Figura 6.14: Número de retransmissões de pacotes de leituras utilizando o Modo de Retransmissão com *timeOutToRetransmitPacket* calculado dinamicamente.

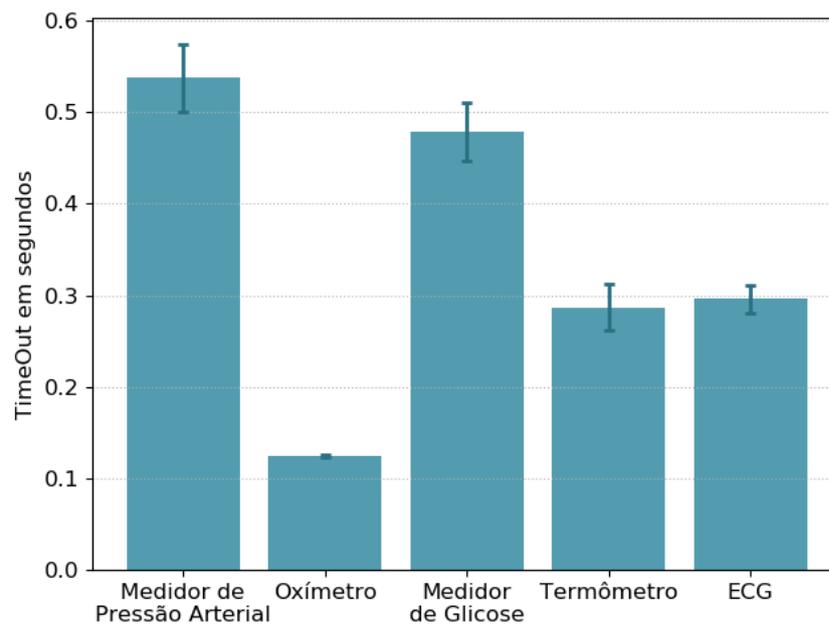


Figura 6.15: Média dos valores de *timeOutToRetransmitPacket* calculados dinamicamente para cada um dos agentes.

a vantagem de controlar a transmissão especificando tempo ou quantidade de pacotes, porém depende da ação de um usuário para acionar a solicitação de dados aos agentes.

Capítulo 7

Conclusão

Neste trabalho, foi apresentada uma adaptação de camada de aplicação para simulação de dispositivos pessoais de saúde em redes corporais sem fio. A camada de aplicação proposta segue o padrão X73-PHD com o auxílio da biblioteca Antidote e foi implementada no simulador Castalia. Cinco agentes foram implementados, que simulam dispositivos médicos reais. Além disso, foi proposto um novo modo confiável de transferência de dados, o modo de retransmissão, para ajustar o protocolo X73-PHD aos cenários WBAN, em que uma camada de transporte confiável não está disponível. O modo de retransmissão visa reduzir o número de dissociações/reassociações que ocorrem após uma mensagem ou um ACK perdido.

O modo de retransmissão, proposto nesta dissertação, além de reenviar uma leitura mais rapidamente, tende a diminuir o número de pacotes de controle necessários durante a transmissão de dados. Esta proposta apresenta bom desempenho em cenários onde é necessário enviar muitas leituras em um curto período de tempo ou, quando o envio é esporádico. Também foi reduzido o *timeout* de 10 segundos para 0,4 segundos no processo de associação. Este valor é fixo e foi escolhido por apresentar o melhor desempenho nos resultados.

O modo sem confirmação utiliza menos pacotes de controle entre todos os modos discutidos, porém não garante a entrega e tem um desempenho baixo, apresentando muitas perdas de pacotes quando o enlace sem fio é ruim. Em contrapartida, tem um bom desempenho quando o envio de leituras deve ser feito de forma contínua e quando o canal sem fio não tem perdas.

O modo com confirmação apresenta um desempenho moderado em cenários onde os dispositivos enviam leituras esporadicamente ou quando existe um intervalo de tempo

entre os envios maior que o *timeout* de associação, que é de 10 segundos neste modo. Nos dispositivos que enviam leituras com intervalos maiores que 1 minuto, este modo consegue reestabelecer uma associação perdida e reenviar a leitura sem grandes impactos no fluxo de dados. Infelizmente, apresenta um comportamento ruim em qualquer outro cenário.

7.1 Trabalhos Futuros

Como trabalhos futuros, pode-se implementar ferramentas de segurança na camada de aplicação, questão que não foi abordada nesta dissertação e que é um ponto muito importante a ser considerado.

Outros dispositivos médicos podem ser implementados para serem incluídos no simulador. Até o momento da escrita desta dissertação, o padrão X73-PHD especificou os seguintes dispositivos médicos: Medidor de Glicose, Medidor de Glicose Contínuo, Monitor de Apneia (*Sleep Apnoea Breathing Therapy Equipment*), ECG de três derivações, Monitor de Atividades Cardiovasculares, Bomba de Insulina, Termômetro, Oxímetro de Pulso, Balança, Monitor de atividades (*Independent living activity hub*), Analisador de Urina, Medidor de Pressão Sanguínea, Medidor de Pico de Fluxo Expiratório e Monitor de Medicação.

A inserção de uma camada de transporte leve (como as utilizadas em IoT) no simulador Castalia, seria uma proposta interessante e que já tem um vasto campo de pesquisa na área de *Wireless Body Area Network*.

Os códigos das aplicações do Castalia e o código modificado da biblioteca Antidote podem ser encontrados, respectivamente, em <https://github.com/conqlima/Antidote> e <https://github.com/conqlima/11073PhdApplication>.

Referências

- [1] BARRÓN-GONZÁLEZ, H. G.; MARTÍNEZ-ESPRONCEDA, M.; TRIGO, J. D.; LED, S.; SERRANO, L. Lessons learned from the implementation of remote control for the interoperability standard ISO/IEEE 11073-20601 in a standard weighing scale. *Computer methods and programs in biomedicine* 123 (2016), 81–93.
- [2] BENEDETTI, D.; PETRIOLI, C.; SPENZA, D. GreenCastalia: An Energy-Harvesting-Enabled Framework for the Castalia Simulator. *Proceedings of the 1st International Workshop on Energy Neutral Sensing Systems - ENSSys '13* (2013), 1–6.
- [3] BLUETOOTH. The Bluetooth SIG Health Device Profile (HDP) Data Exchange Specifications. <https://www.bluetooth.com/specifications/assigned-numbers/health-device-profile>, 2019. Online; acesso 22 de Janeiro de 2019.
- [4] BOULIS, A.; TSELISHCHEV, Y. Contention vs. polling: a study in body area networks MAC design. In *Proceedings of the fifth international conference on body area networks* (2010), ACM, pp. 98–104.
- [5] BOULIS, ATHANASSIOS. Castalia: A simulator for Wireless Sensor Networks and Body Area Networks (User's manual). <https://github.com/boulis/Castalia>, 2013. Online; acesso 7 Janeiro de 2019.
- [6] CLARKE, M.; DE FOLTER, J.; VERMA, V.; GOKALP, H. Interoperable end-to-end remote patient monitoring platform based on IEEE 11073 PHD and zigbee health care profile. *IEEE Transactions on Biomedical Engineering* 65, 5 (2018), 1014–1025.
- [7] FERREIRA, V.; BALBI, H.; SEIXAS, F.; ALBUQUERQUE, C.; MUCHALUAT-SAADE, D. Wireless body area networks: An overview. *XXXV Brazilian Communications and Signal Processing Symposium* (September 2017).
- [8] FERREIRA, V. C.; CABALLERO, E.; LIMA, A. R.; BALBI, H.; SEIXAS, F.; ALBUQUERQUE, C. V. N. D.; SAADE, D. C. M.-S. Redes Corporais Sem Fio e Suas Aplicações em Saúde. In *37o JAI – JORNADAS DE ATUALIZAÇÃO EM INFORMÁTICA*. Natal - RN, 2018, pp. 149–198.
- [9] GOLDBERGER, A.; AMARAL, L.; GLASS, L.; HAVLIN, S.; M. HAUSDORG, J.; IVANOV, P.; G. MARK, R.; E. MIETUS, J.; B. MOODY, G.; PENG, C.-K.; STANLEY, H.; PHYSIOBANK, P. Components of a new research resource for complex physiologic signals. *PhysioNet* 101.
- [10] GOMES, Y. F.; SANTOS, D. F.; ALMEIDA, H. O.; PERKUSICH, A. Integrating MQTT and ISO/IEEE 11073 for health information sharing in the internet of things. In *Consumer Electronics (ICCE), 2015 IEEE International Conference on* (2015), IEEE, pp. 200–201.

- [11] HAO, Y.; FOSTER, R. Wireless body sensor networks for health-monitoring applications. *Physiological Measurement* 29, 11 (2008), R27.
- [12] HL7 INTERNATIONAL. Health Level Seven International. <http://www.hl7.org/index.cfm>, 2019. Online; acesso 7 de Janeiro de 2019.
- [13] HOANG, D. C.; TAN, Y. K.; CHNG, H. B.; PANDA, S. K. Thermal energy harvesting from human warmth for wireless body area network in medical healthcare system. In *International Conference on Power Electronics and Drive Systems (PEDS)* (2009), IEEE, pp. 1277–1282.
- [14] IEEE. ISO/IEEE health informatics – point-of-care medical device communication – part 10101: Nomenclature. *ISO/IEEE 11073-10101:2004(E)* (Dezembro 2004), 1–504.
- [15] IEEE. Health informatics - personal health device communication - application profile: Optimized exchange protocol. Standard, ISO/IEEE, New York, USA, 2010.
- [16] IEEE. Health informatics - personal health device communication: Overview. Standard, ISO/IEEE, New York, USA, 2012.
- [17] IEEE. Introduction to the continua design guidelines. Standard, Personal Connected Health Alliance, 2017.
- [18] IHE INTERNATIONAL. Integrating the Healthcare Enterprise (IHE). <https://www.ihe.net/>, 2019. Online; acesso 7 de Janeiro de 2019.
- [19] INTERNATIONAL TELECOMMUNICATION UNION. Introduction to ASN.1. <https://www.itu.int/en/ITU-T/asn1/Pages/introduction.aspx>, 2019. Online; acesso 23 de Junho de 2019.
- [20] ISO/IEEE. IEEE standard for health informatics - point-of-care medical device communication - part 20101: Application profile - base standard. *ISO/IEEE 11073-20101:2004(E)* (2004), 01–04.
- [21] ISO/IEEE. Health informatics - personal health device communication part 00103: Overview. *IEEE Std 11073-00103-2012* (Aug 2012), 1–80.
- [22] ISO/IEEE. IEEE standard for local and metropolitan area networks - part 15.6: Wireless body area networks. *IEEE Std 802.15.6-2012* (Feb 2012), 1–271.
- [23] ISO/IEEE. IEEE health informatics–personal health device communication part 20601: Application profile–optimized exchange protocol - corrigendum 1. *IEEE Std 11073-20601-2014/Cor 1-2015 (Corrigendum to IEEE Std 11073-20601-2014)* (Outubro 2015), 1–24.
- [24] JIANCHU, Y.; STEVE, W. Applying the ISO/IEEE 11073 standards to wearable home health monitoring systems. *Journal of Clinical Monitoring and Computing* 19, 6 (2005), 427–436.
- [25] KORSANOV, I. N.; S.M, K.; D.A, R.; V.V, F.; M.A, S. On demand plugins for X73-PHD manager. *International Journal of Advanced Trends in Computer Science and Engineering* 2, 5 (2013), 35–40.

- [26] LATRÉ, B.; BRAEM, B.; MOERMAN, I.; BLONDIA, C.; REUSENS, E.; JOSEPH, W.; DEMEESTER, P. A low-delay protocol for multihop wireless body area networks. In *Mobile and Ubiquitous Systems: Networking & Services, 2007. MobiQuitous 2007. Fourth Annual International Conference on* (2007), IEEE, pp. 1–8.
- [27] LEE, Y. F. An interoperability solution for legacy healthcare devices. *IT Professional* 17, 1 (2015), 51–57.
- [28] LIMA, R.; FERREIRA, V.; CABALLERO, E.; ALBUQUERQUE, C.; MUCHALUAT-SAADE, D. Simulation of ISO/IEEE 11073 personal health devices in wbans. In *22nd Int'l ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems* (Miami Beach, 2019), ACM, p. 4.
- [29] LIMA, R.; FERREIRA, V.; CABALLERO, E.; ALBUQUERQUE, C.; MUCHALUAT-SAADE, D. Uma proposta de camada de aplicação baseada no padrão IEEE 11073 para simulação de aplicações de saúde digital em redes corporais sem fio. In *Anais do XIX Simpósio Brasileiro de Computação Aplicada à Saúde* (Porto Alegre, RS, Brasil, 2019), SBC, pp. 234–245.
- [30] MARTÍNEZ, I.; ESEAYOLA, J.; MARTÍNEZ-ESPRONEEDA, M.; SERRANO, L.; TRIGO, J.; LED, S.; GARCÍA, J. Standard-based middleware platform for medical sensor networks and u-health. *Proceedings - International Conference on Computer Communications and Networks, ICCCN* (2008), 714–719.
- [31] MARTINS, A. F.; SANTOS, D. F.; PERKUSICH, A.; ALMEIDA, H. O. IEEE 11073 and connected health: Preparing personal health devices for the internet. In *Consumer Electronics (ICCE), 2014 IEEE International Conference on* (2014), IEEE, pp. 274–275.
- [32] MARTINS, A. F.; SANTOS, D. F.; PERKUSICH, A.; ALMEIDA, H. O. UPnP and IEEE 11073: Integrating personal health devices in home networks. *2014 IEEE 11th Consumer Communications and Networking Conference, CCNC 2014* (2014), 1–6.
- [33] MOVASSAGHI, S.; ABOLHASAN, M.; LIPMAN, J.; SMITH, D.; JAMALIPOUR, A. Wireless body area networks: A survey. *IEEE Communications Surveys Tutorials* 16, 3 (Third 2014), 1658–1686.
- [34] MOVASSAGHI, S.; SHIRVANIMOGHADDAM, M.; ABOLHASAN, M.; SMITH, D. An energy efficient network coding approach for wireless body area networks. In *IEEE 38th Conference on Local Computer Networks (LCN)* (2013), IEEE, pp. 468–475.
- [35] ORYEMA, B.; KIM, H. S.; LI, W.; PARK, J. T. Design and implementation of an interoperable messaging system for IoT healthcare services. *2017 14th IEEE Annual Consumer Communications and Networking Conference, CCNC 2017* (2017), 45–51.
- [36] PAXSON, V.; ALLMAN, M.; CHU, J.; SARGENT, M. Computing tcp’s retransmission timer. RFC 6298, RFC Editor, Junho 2011.
- [37] SANTOS, D. F.; BUBLITZ, F. M.; ALMEIDA, H. O.; PERKUSICH, A. Integrating iee 11073 and constrained application protocol for personal health devices. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing* (2014), ACM, pp. 466–467.

-
- [38] SIGNOVE. Antidote: Program Guide, documentation for developers of applications based on Antidote IEEE 11073 library. <http://oss.signove.com/images/c/c7/AntidoteProgramGuide.pdf>, 2011. Online; acesso 7 de Janeiro de 2019.
- [39] TAO, W.; LIU, T.; ZHENG, R.; FENG, H. Gait analysis using wearable sensors. *Sensors* 12, 2 (2012), 2255–2283.
- [40] VON BUREN, T.; MITCHESON, P. D.; GREEN, T. C.; YEATMAN, E. M.; HOLMES, A. S.; TROSTER, G. Optimization of inertial micropower generators for human walking motion. *Sensors* 6, 1 (2006), 28–38.
- [41] XU, S.; ZHANG, Y.; JIA, L.; MATHEWSON, K. E.; JANG, K.-I.; KIM, J.; FU, H.; HUANG, X.; CHAVA, P.; WANG, R., ET AL. Soft microfluidic assemblies of sensors, circuits, and radios for the skin. *Science* 344, 6179 (2014), 70–74.