

UNIVERSIDADE DO VALE DO RIO DOS SINOS — UNISINOS  
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA  
NÍVEL MESTRADO

SAMUEL FRANCISCO FERRIGO

*MOMICARE:*

Um *middleware* móvel para *smartphones* para gerenciamento de uma rede de sensores sem fio corporais

SÃO LEOPOLDO  
2017



Samuel Francisco Ferrigo

*MOMICARE:*

Um *middleware* móvel para *smartphones* para gerenciamento de uma rede de sensores sem fio corporais

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre pelo Programa de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos — UNISINOS

Orientador:  
Prof. Dr. Cristiano André da Costa

Co-orientador:  
Prof. Dr. Rodrigo da Rosa Righi

São Leopoldo  
2017

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)

Ferrigo, Samuel Francisco

*MOMICARE*:: um *middleware* móvel para *smartphones* para gerenciamento de uma rede de sensores sem fio corporais / Samuel Francisco Ferrigo — 2017.

101 f.: il.; 30 cm.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, São Leopoldo, 2017.

“Orientador: Prof. Dr. Cristiano André da Costa, Unidade Acadêmica de Pesquisa e Pós-Graduação”.

1. Computação Móvel. 2. Informática médica. 3. Cuidados médicos. 4. Rede de sensores sem fio. 5. Eficiência energética. 6. Sincronização. I. Título.

CDU 004.75.057.5:61

Bibliotecário Alessandro Dietrich — CRB 10/2338

Samuel Francisco Ferrigo

MOMICARE

Um *middleware* móvel para smartphones para gerenciamento de uma rede de sensores sem fio corporais

Dissertação apresentada à Universidade do Vale do Rio dos Sinos – Unisinos, como requisito parcial para obtenção do título de Mestre em Computação Aplicada.

Aprovado em 03/08/2017

BANCA EXAMINADORA

---

Prof. Dr. Cristiano André da Costa - UNISINOS

---

Prof. Dr. Rodrigo da Rosa Righi - UNISINOS

---

Prof. Dr. Gerson G. H Cavalheiro - UFPEL

---

Pof. Dr. Jorge Luis Victória Barbosa - UNISINOS

Prof. Dr. Cristiano André da Costa (Orientador)  
Prof. Dr. Rodrigo da Rosa Righi (Co-orientador)

Visto e permitida a impressão.  
São Leopoldo,

Prof. Dr. Sandro José Rigo  
Coordenador PPG em Computação Aplicada



À minha família.

*Fé inabalável só o é a que pode encarar frente a frente a razão,  
em todas as épocas da Humanidade.*

— ALLAN KARDEC





## **AGRADECIMENTOS**

Primeiramente, ao meu orientador Prof. Cristiano André da Costa, pela paciência, ponderação e disponibilidade durante esta jornada.

Ao meu co-orientador Prof. Dr. Rodrigo da Rosa Righi, pelas intervenções e questionamentos em relação aos aspectos técnicos do trabalho.

Aos professores Sandro Rigo e Jorge Barbosa, que contribuíram com ideias durante a concepção da proposta.

Aos meus colegas de setor da Câmara de Vereadores de Caxias do Sul, especialmente ao colega Machado, por auxiliar-me na troca do disco rígido e na restauração dos arquivos que perdi durante a confecção deste trabalho.

Ao meu irmão Jonatas, por ter cedido seu monitor para que pudesse montar os protótipos e executar as avaliações deste trabalho.

À minha namorada Juliana, pela compreensão nos momentos que não pude dar a atenção que merecia.

Aos meus pais, Paulo e Elisa, pelos ensinamentos da vida que levaram-me a ser o que sou.

E, finalmente, ao meu filho Artur, que é a razão de minha vida.



## RESUMO

A Internet das Coisas, ou simplesmente *IoT*, apresenta inúmeras soluções para a área da saúde. Diversos sensores podem ser colocados ao redor do corpo humano para monitoramento de seus sinais vitais e, a partir disso, prever problemas de saúde. Gerenciar esses sensores apresenta uma série de desafios que não são resolvidos pelos trabalhos relacionados pesquisados, como a busca pela eficiência energética na transmissão de dados, a segurança, a interoperabilidade e sincronização entre sensores que formam uma rede de sensores sem fio corporais. Uma solução para estes problemas é apresentada pelo modelo *MOMICARE*, um *middleware* móvel para *smartphones* para gerenciamento de uma rede de sensores sem fio corporais. O *MOMICARE* permite que um *smartphone* receba dados de diversos sensores, armazene-os e retransmita-os às aplicações médicas que necessitem usar esses dados, oferecendo baixo consumo de energia, segurança, independência de protocolos de redes de sensores sem fios e sincronismo entre os dados coletados pelos sensores. As principais contribuições científicas deste trabalho são a redução do consumo energético de dispositivos que formam uma rede *Wireless Body Area Network*, e o oferecimento de um sincronismo temporal entre os dados desses dispositivos. O modelo foi analisado por meio da implementação de um protótipo e um ambiente de simulação, onde foram realizadas avaliações de: (1) consumo de energia dos dispositivos da rede *Wireless Body Area Network*, (2) sincronização entre os dispositivos e o *middleware*, e (3) carga sobre o dispositivo onde o protótipo foi executado, considerando o uso de diversas aplicações simultâneas. Como resultados obtidos, verificou-se que, em dispositivos *Bluetooth Low Energy*, a redução no consumo de energia superou 10% em relação a aplicações convencionais, enquanto que em dispositivos padrão *IEEE 802.15.4* a redução no consumo energético chegou a até 7,8%, quando comparado ao uso de aplicações convencionais. Outro resultado obtido foi a variação de sincronização inferior a 0,5 segundos, valores aceitáveis no que diz respeito a dispositivos médicos. Por fim, em todos os testes realizados, não foi verificado qualquer tipo de sobrecarga, sendo que a média de uso do processador apresentou valores médios inferiores a 1%.

**Palavras-chave:** Computação Móvel. Informática médica. Cuidados médicos. Rede de sensores sem fio. Eficiência energética. Sincronização.



## ABSTRACT

The Internet of Things, or simply IoT, presents numerous solutions for the health area. Several sensors can be placed around the human body to monitor your vital signs and, from there, predict health problems. Managing these sensors presents a number of challenges that are not resolved by related research, such as the search for energy efficiency in data transmission, security, interoperability and synchronization between sensors that form a network of wireless body sensors. One solution to these problems is presented by the MOMICARE model, a mobile middleware for smartphones for managing a network of wireless body sensors. MOMICARE allows a smartphone to receive data from multiple sensors, store and re-transmit it to medical applications that need to use that data, offering low power consumption, security, wireless sensor network protocol independence and synchronization between data collected by sensors. The main scientific contributions of this work are the reduction of the energy consumption of devices that form a Wireless Body Area Network, and the offer of a temporal synchronism between the data of these devices. The model was analyzed through the implementation of a prototype and simulation environment, where evaluations were made of: (1) energy consumption of Wireless Body Area Network devices, (2) synchronization between devices and middleware, and 3) load on the device where the prototype was executed, considering the use of several simultaneous applications. As results obtained, it has been found that in Bluetooth Low Energy devices, the reduction in energy consumption exceeded 10% compared to conventional applications, whereas in standard devices IEEE 802.15.4 reduction in energy consumption reached up to 7.8%, when compared to the use of conventional applications. Another result obtained was the synchronization variation less than 0.5 seconds, acceptable values with respect to medical devices. Finally, in all the tests performed, no type of overload was verified, and the average use of the processor presented average values lower than 1%.

**Keywords:** Mobile computing. Medical informatics. Healthcare. Wireless sensor network. Energy efficiency. Synchronization.



## LISTA DE FIGURAS

Figura 1:	Uma arquitetura <i>m-health</i> . . . . .	24
Figura 2:	A proposta de solução para a questão de pesquisa. . . . .	26
Figura 3:	Arquitetura <i>BLE</i> . . . . .	31
Figura 4:	Estrutura de um quadro <i>BLE</i> . . . . .	32
Figura 5:	Topologias de rede do padrão <i>IEEE 802.15.4</i> . . . . .	35
Figura 6:	Camadas do padrão <i>IEEE 802.15.4</i> . . . . .	36
Figura 7:	O superquadro do protocolo <i>IEEE 802.15.4</i> . . . . .	37
Figura 8:	O estrutura de quadros no padrão <i>IEEE 802.15.4</i> . . . . .	37
Figura 9:	Um <i>gateway</i> de conversão do <i>Zigbee</i> para o <i>IPv6</i> . . . . .	38
Figura 10:	Pilhas de cabeçalhos <i>6LoWPAN</i> normalmente usadas . . . . .	39
Figura 11:	Localizações mais comuns de sensores corporais não intrusivos . . . . .	41
Figura 12:	O funcionamento do algoritmo de Berkeley . . . . .	42
Figura 13:	O funcionamento do relógio de Lamport . . . . .	43
Figura 14:	A arquitetura da solução <i>MOMICARE</i> . . . . .	56
Figura 15:	Estrutura de dados utilizada pelo <i>MOMICARE</i> . . . . .	58
Figura 16:	Fluxo de comunicação entre os sensores de uma <i>WBAN</i> e o <i>smartphone</i> . Na esquerda é mostrado o envio de dados dos sensores para o <i>smartphone</i> , enquanto que na direita é mostrado o envio de dados do <i>smartphone</i> aos sensores. . . . .	62
Figura 17:	Fluxo de comunicação entre as aplicações (externas ou do próprio <i>smartphone</i> e o <i>smartphone</i> . Na esquerda é mostrado o envio de dados das aplicações para o <i>smartphone</i> , enquanto que na direita é mostrado o envio de dados do <i>smartphone</i> às aplicações. . . . .	63
Figura 18:	A arquitetura do protótipo . . . . .	65
Figura 19:	Um trecho do formato XML gerado pelo protótipo e que é executado pelo <i>nRF Connect</i> . . . . .	66
Figura 20:	Telas do protótipo . . . . .	67
Figura 21:	A cinta cardíaca (à esquerda) e a pulseira (à direita) utilizadas para avaliação do protótipo . . . . .	69
Figura 22:	A arquitetura do ambiente de simulação . . . . .	71
Figura 23:	Telas do emulador <i>Cooja</i> . . . . .	72
Figura 24:	Os resultados da variação do <i>MTU</i> referente à cinta . . . . .	77
Figura 25:	Comparativo do consumo energético da cinta utilizando as diferentes para- metrizações de <i>MTU</i> no protótipo . . . . .	77
Figura 26:	Os resultados da variação do <i>MTU</i> referente à pulseira . . . . .	78
Figura 27:	Comparativo do consumo energético da pulseira utilizando as diferentes pa- rametrizações de <i>MTU</i> no protótipo . . . . .	79
Figura 28:	Comparativo do consumo energético da cinta quando diversas aplicações são executadas simultaneamente, e seu comparativo com o consumo registrado no aplicativo <i>EliteHRV</i> . . . . .	80
Figura 29:	Comparativo do consumo energético da pulseira quando diversas aplicações são executadas simultaneamente, e seu comparativo com o consumo regis- trado no aplicativo <i>HBand</i> . . . . .	81

Figura 30:	Comparativo do consumo energético dos dispositivos <i>IEEE</i> 802.15.4 quando alterado o MTU do dispositivo-pai <i>Tmote-sky</i> , e em relação ao cenário sem realizar a configuração . . . . .	82
Figura 31:	Comparativo da média do consumo energético dos 2 dispositivos <i>IEEE</i> 802.15.4 quando utilizado aplicações simultâneas executadas independentemente e quando executadas de forma simultânea pelo dispositivo-pai <i>Tmote-sky</i> . . . . .	83
Figura 32:	Comparativo da média do consumo energético dos 4 dispositivos <i>IEEE</i> 802.15.4 quando utilizado aplicações simultâneas executadas independentemente e quando executadas de forma simultânea pelo dispositivo-pai <i>Tmote-sky</i> . . . . .	84
Figura 33:	Variação do horário entre o horário registrado pelo relógio interno da pulseira e o horário recalculado pelo protótipo. . . . .	86
Figura 34:	Avaliação de uso de processamento do protótipo considerando o uso de 1, 2, 4 e 8 aplicações simultâneas. . . . .	89
Figura 35:	Avaliação de uso de memória do protótipo considerando o uso de 1, 2, 4 e 8 aplicações simultâneas. . . . .	90



## LISTA DE TABELAS

Tabela 1:	Tabela de serviços <i>GAP</i> . . . . .	34
Tabela 2:	Comparativo entre os trabalhos relacionados . . . . .	52
Tabela 3:	Exemplo de sincronização de tempo entre sensores . . . . .	60
Tabela 4:	Resumo dos resultados esperados das avaliações. . . . .	76
Tabela 5:	Valores sincronizados pelo protótipo, tendo como base o horário da chegada dos dados da pulseira . . . . .	87
Tabela 6:	Valores sincronizados pelo protótipo, considerando que o horário da pulseira não foi sincronizado. . . . .	88
Tabela 7:	Resumo das contribuições científicas do modelo <i>MOMICARE</i> . . . . .	94



## LISTA DE SIGLAS

<i>ACK</i>	<i>Acknowledgement</i>
<i>ACL</i>	<i>Asynchronous Connectionless</i>
<i>AP</i>	<i>Access Point</i>
<i>API</i>	<i>Application Programming Interface</i>
<i>BAN</i>	<i>Body Area Network</i>
<i>BAN</i>	<i>Body and Area Sensor Network</i>
<i>BLE</i>	<i>Bluetooth Low Energy</i>
<i>BMU</i>	<i>Biosignal Monitoring Unit</i>
<i>BO</i>	<i>Beacon Order</i>
<i>BS</i>	<i>Bridge Sensor</i>
<i>CSMA</i>	<i>Carrier Sense Multiple Access</i>
<i>CAP</i>	<i>Contension Access Period</i>
<i>CFP</i>	<i>Contension Free Period</i>
<i>CSMA/CA</i>	<i>CSMA with Collision Avoidance</i>
<i>DAC</i>	<i>Discretionary Access Control</i>
<i>ED</i>	<i>Energy Detection</i>
<i>ER</i>	<i>Entidade-Relacionamento</i>
<i>EHR</i>	<i>Electronic Health Record</i>
<i>ESB</i>	<i>Enterprise Serial Bus</i>
<i>FFD</i>	<i>Full-funcion Device</i>
<i>GAP</i>	<i>Generic Access Profile</i>
<i>GATT</i>	<i>Generic Attribute Profile</i>
<i>GPS</i>	<i>Global Position System</i>
<i>GTS</i>	<i>Garantee Time Slots</i>
<i>HC1</i>	<i>IPv6 Header Compression</i>
<i>HC2</i>	<i>UDP Header Compression</i>
<i>HTTP</i>	<i>Hypertext Transfer Protocol</i>
<i>IEEE</i>	<i>Institute of Electrical and Electronics Engineers</i>
<i>IETF</i>	<i>Internet Engineering Task Force</i>
<i>IoT</i>	<i>Internet of Things</i>
<i>IP</i>	<i>Internet Protocol</i>
<i>ISM</i>	<i>Industrial Medical Scientific</i>
<i>LAN</i>	<i>Local Area Network</i>

<i>L2CAP</i>	<i>Logical Link Control Adaptation Protocol</i>
<i>LLC</i>	<i>Logical Link Control</i>
<i>LQI</i>	<i>Link Quality Indicator</i>
<i>LTE</i>	<i>Long Term Evolution</i>
<i>m-health</i>	<i>Mobile Health</i>
<i>MAC</i>	<i>Media Access Control</i>
<i>MFR</i>	<i>MAC Footer</i>
<i>MHR</i>	<i>MAC Header</i>
<i>MSDU</i>	<i>MAC Service Data Unit</i>
<i>MTU</i>	<i>Maximum Transmission Unit</i>
<i>OSI</i>	<i>Open Systems Interconnection</i>
<i>P2P</i>	<i>Peer-to-peer</i>
<i>PHR</i>	<i>PHY Header</i>
<i>PIB</i>	<i>Produto Interno Bruto</i>
<i>PPDU</i>	<i>Physical Protocol Data Unit</i>
<i>PSDU</i>	<i>PHY Service Data Unit</i>
<i>QoS</i>	<i>Quality of Service</i>
<i>RAM</i>	<i>Random Access Memory</i>
<i>REST</i>	<i>Representational State Transfer</i>
<i>RFC</i>	<i>Request for Comments</i>
<i>RFD</i>	<i>Reduced-function Device</i>
<i>SBMHM</i>	<i>Smartphone-based Mobile Health Monitoring</i>
<i>SCO</i>	<i>Synchronous Connection Oriented</i>
<i>SGBD</i>	<i>Sistema Gerenciador de Banco de Dados</i>
<i>SHR</i>	<i>Synchronization Header</i>
<i>SLR</i>	<i>Systematic Literature Review</i>
<i>SO</i>	<i>Superframe Order</i>
<i>TDM</i>	<i>Time-division multiplexing</i>
<i>UART</i>	<i>Universal Asynchronous Receiver-Transmitter</i>
<i>UDP</i>	<i>User Datagram Protocol</i>
<i>URI</i>	<i>Uniform Resource Identifier</i>
<i>USB</i>	<i>Universal Serial Bus</i>
<i>UTC</i>	<i>Universal Time Coordinated</i>
<i>WBAN</i>	<i>Wireless Body Area Network</i>

<i>WiFi</i>	<i>Wireless Fidelity</i>
<i>WPAN</i>	<i>Wireless Personal Area Network</i>
<i>WSN</i>	<i>Wireless Sensor Network</i>



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>23</b>
1.1	Questão de pesquisa	25
1.2	Objetivos	26
1.3	Organização do Texto	27
<b>2</b>	<b>FERRAMENTAL TÉCNICO</b>	<b>29</b>
2.1	<i>Bluetooth Low Energy (BLE)</i>	29
2.1.1	Camada física	30
2.1.2	Camada de controle do enlace	30
2.1.3	Camada de aplicações	32
2.1.4	Demais camadas	32
2.2	Padrão <i>IEEE 802.15.4</i>	34
2.3	<i>6LowPAN</i>	38
2.3.1	<i>6LowPAN no Bluetooth</i>	39
2.4	<i>WBAN</i>	40
2.5	Sincronização	41
2.6	Considerações parciais	43
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>45</b>
3.1	<i>SBMHM</i>	45
3.2	<i>BeWell+</i>	46
3.3	<i>HyHealthAssistant</i>	46
3.4	<i>EcoHealth</i>	47
3.5	<i>Home Health Gateway Based Healthcare Services Through U-Health Platform</i>	47
3.6	<i>WellPhone</i>	48
3.7	<i>Integration of wearable devices in a wireless sensor network for an E-health application</i>	48
3.8	Comparativo entre as soluções	49
3.9	Avaliação geral das soluções	50
3.10	Lacunas de pesquisa	51
3.11	Considerações parciais	52
<b>4</b>	<b>MODELO <i>MOMICARE</i></b>	<b>55</b>
4.1	Arquitetura do modelo	55
4.1.1	Módulo de Gerenciamento	55
4.1.2	Módulo do Dispositivo	57
4.1.3	Módulo de Comunicação	58
4.1.4	Módulo de Integração	61
4.2	Funcionamento geral do <i>MOMICARE</i>	61
4.3	Considerações parciais	63
<b>5</b>	<b>METODOLOGIA DE AVALIAÇÃO</b>	<b>65</b>
5.1	Protótipo	65
5.1.1	Metodologia de Avaliação Energética	68
5.1.2	Metodologia de Avaliação de Sincronismo	69
5.1.3	Metodologia de Avaliação de Carga	69

5.1.4	Limitações do Protótipo . . . . .	70
<b>5.2</b>	<b>Ambiente de Simulação . . . . .</b>	<b>70</b>
5.2.1	Metodologia de Avaliação Energética . . . . .	71
5.2.2	Limitações do Ambiente de Simulação . . . . .	73
<b>5.3</b>	<b>Considerações parciais . . . . .</b>	<b>73</b>
<b>6</b>	<b>RESULTADOS E DISCUSSÃO . . . . .</b>	<b>75</b>
<b>6.1</b>	<b>Consumo Energético . . . . .</b>	<b>75</b>
6.1.1	Protótipo . . . . .	75
6.1.2	Ambiente de Simulação . . . . .	81
<b>6.2</b>	<b>Sincronismo . . . . .</b>	<b>86</b>
<b>6.3</b>	<b>Carga . . . . .</b>	<b>89</b>
<b>6.4</b>	<b>Considerações parciais . . . . .</b>	<b>91</b>
<b>7</b>	<b>CONCLUSÃO . . . . .</b>	<b>93</b>
<b>7.1</b>	<b>Contribuição Científica . . . . .</b>	<b>93</b>
<b>7.2</b>	<b>Artigos Publicados . . . . .</b>	<b>95</b>
<b>7.3</b>	<b>Trabalhos futuros . . . . .</b>	<b>95</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>97</b>



## 1 INTRODUÇÃO

A *mobile healthcare (m-health)*, também conhecida por *mobile telemedicine*, é uma área da telemedicina que se utiliza de dispositivos de computação e comunicação móveis para a prestação de serviços médicos (CONSTANTINESCU; KIM; FENG, 2012) (MERRELL; DOARN, 2014). Com a popularização das tecnologias de comunicação sem fio, e a consequente popularização dos dispositivos móveis, a *m-health* tornou-se algo acessível à grande parte da população (REN; PAZZI; BOUKERCHE, 2010) e que vem revolucionando o atendimento médico a pacientes (CONSTANTINESCU; KIM; FENG, 2012).

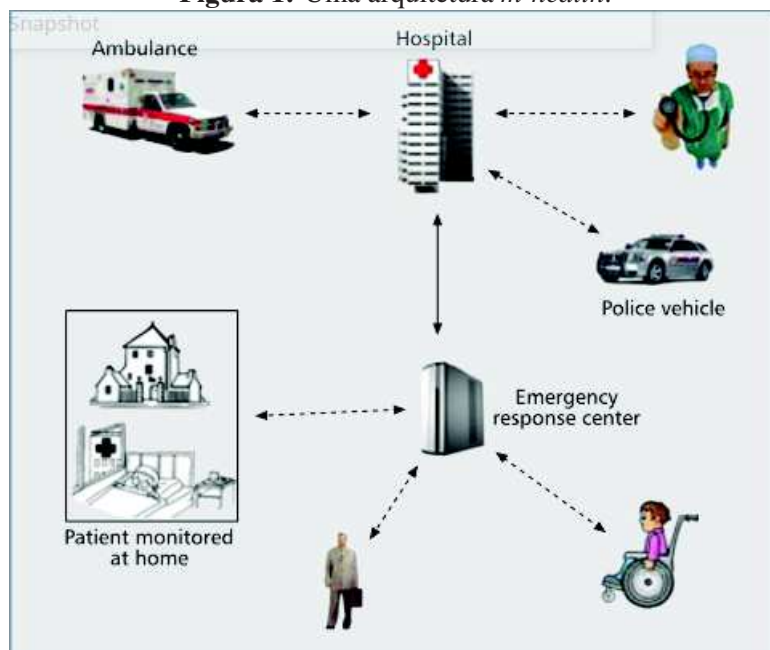
A área da saúde, historicamente, consome muitos recursos financeiros. Somente nos Estados Unidos, cerca de 17% do Produto Interno Bruto (PIB) é direcionado a cuidados médicos, mostrando o quão ineficientes e caros são os serviços oferecidos naquele país (JIN; CHEN, 2015). No Brasil a situação não é diferente. Segundo o IBGE (2015), somente o governo federal investe quase 2% do PIB em saúde. Ainda assim, em virtude de suas dimensões continentais, os serviços médicos do país são ineficientes e acessíveis a apenas uma parcela da população, principalmente àquelas que vivem nos grandes centros urbanos.

Dessa forma, a *mobile healthcare* acaba sendo uma aliada na melhora da eficiência em cuidados médicos. Ela permite que pacientes, médicos, enfermeiros, hospitais, emergências e quaisquer outros recursos médicos necessários estejam em comunicação direta e contínua (REN; PAZZI; BOUKERCHE, 2010). A *m-health* possui uma arquitetura onde a Internet das Coisas, ou *Internet of Things (IoT)* (Riazul Islam; Humaun Kabir; HOSSAIN, 2015), exerce um papel preponderante nessa estrutura.

Um cenário de uso dessa arquitetura é apresentado na Figura 1, onde um carro de polícia pode relatar a um hospital um acidente fornecendo os dados das vítimas. O hospital, por sua vez, verifica se estes constam em seu prontuário eletrônico, também conhecido como *Electronic Health Record (EHR)* (WU et al., 2017). A partir do *EHR*, o hospital pode alertar à equipe médica que se desloca ao atendimento da ocorrência, normalmente em uma ambulância, cuidados específicos que devem ser tomados em relação às vítimas. A comunicação entre polícia, hospital e ambulância se dá por meio de conexões 3G/4G e outros protocolos compatíveis com o protocolo *IP* (REN; PAZZI; BOUKERCHE, 2010).

Outro cenário, também apresentado na Figura 1, permite que um centro de emergência possa realizar a triagem de pacientes a partir de dados capturados anteriormente por meio de sensores instalados no corpo de uma pessoa, facilitando o diagnóstico. Essa comunicação entre sensores é realizada por intermédio de protocolos específicos para dispositivos *IoT*, tais como *Bluetooth* e 802.15.4. Em ambientes específicos, como num ambiente doméstico (também demonstrado na Figura 1), isso torna-se possível por meio de equipamentos que realizam a função de *gateway*, convertendo os dados enviados por esses protocolos num formato que possa ser recebido por redes padrão *IP*, como redes *Wi-Fi* (SEEGER; VAN LAERHOVEN; BUCHMANN, 2015) (IPV6.BR, 2013). Todavia, na maioria dos ambientes - especialmente em ambientes externos

**Figura 1:** Uma arquitetura *m-health*.



Fonte: Retirado de Ren, Pazzi e Boukerche (2010)

- esses equipamentos não estão disponíveis, dificultando o uso dos dados coletados por esses sensores (LEE; JEONG; YOON, 2012).

Ainda, conforme demonstrado na Figura 1, os dois cenários podem ser combinados, permitindo que os dados coletados nos sensores possam ser correlacionados com os dados armazenados no *EHR*. Isso proporciona uma avaliação ainda mais precisa, seja pela equipe de atendimento de emergência, seja por um profissional médico ao realizar uma consulta, ou mesmo pela equipe médica da ambulância.

A *m-health* permite também que problemas de saúde possam ser identificados antecipadamente e mais facilmente tratados, por meio de um monitoramento contínuo. Esse monitoramento consiste em medir a condição de saúde de um paciente, obtida por intermédio da medição de seus sinais vitais. Tradicionalmente, há cinco sinais vitais que devem ser medidos: temperatura corporal, pressão arterial, pulso, respiração e saturação de oxigênio (ELLIOTT; COVENTRY, 2012) (SHAHAMABADI et al., 2013) (Riazul Islam; Humaun Kabir; HOSSAIN, 2015).

Esse monitoramento contínuo não é possível atualmente somente pelo uso de *smartphones* e *smartwatches*. Hoje poucos aparelhos possuem sensores dedicados para monitoramento dos sinais vitais. O aparelho Samsung Galaxy S6, por exemplo, conta com sensor de batimento cardíaco (SAMSUNG, 2015). Existem, também, aplicativos para *smartphones* que permitem o monitoramento de sinais vitais por meio do uso da câmera do aparelho (HAN et al., 2015), todavia, seu uso contínuo é inviável pois exige que o usuário mantenha um de seus dedos sobre a câmera do aparelho. Nesse contexto, os dispositivos *IoT* acabam realizando a tarefa de registrar os sinais vitais de maneira contínua. Esses dispositivos podem ser implantados por meio de intervenções cirúrgicas ou fixados ao corpo de uma pessoa (ULLAH et al., 2012), podendo

comunicar-se com outros dispositivos por meio da utilização ou não de fio (CASAMASSIMA; FARELLA; BENINI, 2013). Assim, cada dispositivo *IoT* possui uma função específica de monitoramento, e muitos dispositivos podem ser utilizados para monitoramento dos sinais vitais.

O conjunto de dispositivos *IoT* corporais acaba formando uma rede chamada de *Body Area Network (BAN)* (CHUNG; LEE; JUNG, 2008) ou, quando somente faz uso de dispositivos sem fio, de *Wireless Body Area Network (WBAN)* (CASTILLEJO et al., 2013). Alguns desses dispositivos, como *smartphones*, *tablets* e *smartwatches* em geral possuem grande capacidade de processamento e armazenamento. Todavia, a maior parte dos dispositivos *IoT* possuem capacidade limitada de processamento que possuem um rádio integrado para transmissão dos dados coletados (SHAHAMABADI et al., 2013) (WANG et al., 2015). Dessa forma, os dispositivos *IoT* necessitam de uma aplicação terceira que receba os dados capturados e os processem. Todavia, nem sempre esses dispositivos estão em locais que possuam os recursos de rede que permitam a comunicação com essas aplicações. Nesse cenário, alguns desses dispositivos *wearables*, como os *smartphones*, podem atuar como um intermediário que recebe esses dados e os enviem para as aplicações no momento em que haja uma rede disponível, conforme demonstrado na Figura 2.

Todavia, o uso *WBANs* cria outros desafios que existem nos sistemas distribuídos em geral: o sincronismo de dados e a interoperabilidade de comunicação entre esses dispositivos. Dependendo do tipo de monitoramento médico a ser realizado, o sincronismo pode exercer um papel fundamental na avaliação de um paciente (CASAMASSIMA; FARELLA; BENINI, 2013), enquanto que a interoperabilidade de comunicação permite que um maior número de dispositivos *IoT* voltados à saúde possam ser utilizados (FOTOUHI et al., 2016).

## 1.1 Questão de pesquisa

O presente trabalho propõe-se a responder a seguinte questão de pesquisa:

*Como coletar os dados de sensores que captam os sinais vitais de uma pessoa de maneira contínua, síncrona, com baixo consumo de energia e, no momento requisitado, disponibilizá-los a aplicações externas que necessitem desses dados?*

A continuidade do monitoramento requer o uso de sensores *IoT*, de forma que não haja necessidade da intervenção do paciente para realização do registro dos indicadores de saúde de um paciente, especialmente os sinais vitais. Ainda, a comunicação entre esses dispositivos e o *middleware* deve ocorrer em intervalos que não prejudiquem a confiabilidade dos dados registrados por estes dispositivos. Além disso, estes sensores comunicam-se por meio de diversos protocolos de comunicação, assim a interoperabilidade entre esses padrões faz-se necessária para que não haja uma restrição no uso de sensores numa rede *WBAN*. O uso de diversos sensores também requer que haja uma sincronização entre os sensores de uma *WBAN*, permitindo a correlação entre os dados capturados. Especialmente na saúde, essa correlação é importante para que possa ser feito um diagnóstico de saúde correto. Tornar a comunicação entre os sensores

**Figura 2:** A proposta de solução para a questão de pesquisa.



Fonte: Elaborado pelo autor.

energeticamente mais eficiente permite que o tempo de carga de uma bateria de um sensor possa ser prolongado, ou que baterias com menor capacidade de carga possam ser utilizadas. Por fim, todos os dados enviados pelos sensores devem estar disponíveis a aplicações externas voltadas a saúde para que possam ser armazenadas processadas e/ou avaliadas por um profissional médico.

Dessa forma, a contribuição científica deste trabalho é a proposta de um modelo que permita que a comunicação entre dispositivos *IoT* - como, por exemplo, sensores localizados no corpo de uma pessoa - e aplicações médicas - localizadas em qualquer parte do mundo - possam ocorrer de maneira energeticamente eficiente e de forma a haver uma sincronização de tempo nos dados enviados por esses sensores. Isso permite que o tempo de funcionamento desses dispositivos possa ser prolongado, aumentando sua disponibilidade, bem como oferece a possibilidade de correlacionamento entre os dados por eles enviados.

## 1.2 Objetivos

O objetivo geral deste trabalho é desenvolver um modelo de *middleware* que funcione em um *smartphone* que possa comunicar-se de maneira contínua, síncrona e energeticamente eficiente com os diversos tipos de sensores *IoT* existentes. Como esse dispositivo está normalmente próximo de um corpo de uma pessoa, ele não teria problemas de comunicação pela distância próxima. Ainda, o *smartphone*, em função de sua maior capacidade de processamento e armazenamento, pode armazenar e transmitir os dados para aplicações médicas externas que os requisitem, eliminando a necessidade de retransmissão dos mesmos dados pelos sensores. Além disso, os *smartphones* possuem sensores internos - tais como acelerômetro e *GPS* - que permitem que os dados capturados pelos sensores *IoT* possam ser complementados. O *GPS*, inclusive, permite que a análise de saúde de uma pessoa possa ser avaliada a partir de um determinado contexto.

Para chegar a esse objetivo geral, este trabalho visa atender também aos seguintes objetivos específicos:

- Analisar e descrever os conceitos teóricos envolvidos na elaboração do modelo;
- Identificar os trabalhos relacionados à área de cuidados médicos que façam uso de *smartphones* e dispositivos que fazem o uso de redes de sensores sem fio;
- Descrever o modelo *MOMICARE*;
- Implementar o protótipo com base no modelo descrito;
- Avaliar o modelo com base no protótipo desenvolvido, a partir de cenários definidos;
- Descrever os resultados obtidos nessa análise.

### 1.3 Organização do Texto

O texto deste trabalho está organizado da seguinte maneira: no Capítulo 2 são apresentados os conceitos necessários para entendimento da proposta deste trabalho. No Capítulo 3 são apresentados os trabalhos relacionados que servem de base para a elaboração da questão de pesquisa e, conseqüentemente, a elaboração da proposta. No Capítulo 4 é apresentado o modelo *MOMICARE* e detalhado o seu funcionamento. No Capítulo 5 é descrita a metodologia de avaliação do modelo, composta por um protótipo e um ambiente de simulação. No Capítulo 6 são apresentados os resultados e discussões referentes à avaliação efetuada. Por fim, no Capítulo 7 são apresentados as conclusões deste trabalho, bem como os trabalhos futuros que podem ser desenvolvidos a partir deste trabalho.



## 2 FERRAMENTAL TÉCNICO

Neste capítulo são apresentados os conceitos básicos que envolvem o entendimento da proposta deste trabalho. Primeiramente são explicados os protocolos de redes utilizados para comunicação entre dispositivos que formam uma *WBAN*, iniciando pelo padrão *Bluetooth*, especificamente a sua versão voltada à dispositivos *IoT*, chamada de *Bluetooth Low Energy (BLE)*, com foco em economia de energia. Em seguida, é detalhada a especificação *IEEE 802.15.4*, que também é muito utilizada em *WBANs* devido à sua economia de energia. Na sequência, é explicado o protocolo *6LoWPAN*, que possibilita a integração dos protocolos *Bluetooth* e *IEEE 802.15.4* ao protocolo *IP*. Em seguida, é explicado o que é uma *WBAN* e as suas diferenças em relação às demais redes de sensores sem fio. Finalmente, como as redes *WBAN* formam um sistema distribuído, foi explanado as formas de sincronização de dispositivos que formam este tipo de sistema.

### 2.1 *Bluetooth Low Energy (BLE)*

Criado em 1999, o *Bluetooth* é um padrão sem fios utilizado para conectar dispositivos de computação e intercomunicação, e também acessórios, utilizando rádios sem fio de curto alcance (TANENBAUM; WETHERALL, 2011). O *Bluetooth* utiliza o padrão mestre-escravo para comunicação entre dispositivos. Vários dispositivos eletrônicos utilizam *Bluetooth*, desde telefones móveis, *smartphones*, *notebooks*, fones de ouvido, impressoras, teclados, mouses, jogos, relógios, aparelhos de música (TANENBAUM; WETHERALL, 2011) e dispositivos *IoT* em geral (SIG, 2016a).

A especificação do protocolo foi evoluindo desde seu surgimento. Desde a versão 4.0 até a versão atual (5.0)<sup>1</sup>, especifica a operação em baixa potência, útil para dispositivos *IoT*. Devido a esse fato, o *Bluetooth 4.+* também é chamado de *Bluetooth Low Energy (BLE)* (GONNOT et al., 2015) (SIG, 2016a) ou também *Bluetooth Smart* (SIG, 2016a). O *BLE* possui diferenças significativas em relação às versões anteriores do *Bluetooth*, compostos de *hardwares* diferentes (SIG, 2016d) e, portanto, são incompatíveis. Os atuais equipamentos que utilizam *Bluetooth* costumam vir com ambas implementações, e são chamados de *Bluetooth Smart Ready* (PES-SOA, 2016).

Uma rede *Bluetooth* tradicional é formada por um dispositivo mestre e até sete dispositivos escravos ativos que estejam localizados a uma distância de até 10 metros. Esses dispositivos forma uma *piconet* (TANENBAUM; WETHERALL, 2011) (SIG, 2016e). É possível também formar uma rede com vários *piconets*, que formam uma rede chamada *scatternet* (TANENBAUM; WETHERALL, 2011) (SIG, 2016e). A ligação entre as *piconets* se dá por meio de um nó escravo que faz a função de ponte *bridge* entre duas *piconets*.

---

<sup>1</sup>O *Bluetooth Special Interest Group*, grupo responsável pelo desenvolvimento do *Bluetooth* lançou a versão 5.0 da tecnologia *Bluetooth* em dezembro de 2016 (SIG, 2016b)(SIG, 2016c).



Já no *BLE* normalmente são criadas apenas *piconets*. Em função de sua capacidade de endereçamento de 48 *bits* (SIG, 2016e), a *BLE* permite a criação de grandes redes na topologia estrela que podem formar grandes *piconets*, praticamente acabando com a utilidade das *scatter-nets*.

Como a arquitetura tradicional do *Bluetooth* difere significativamente da arquitetura do *BLE*, será detalhado apenas o funcionamento deste último, devido ao fato de ser a arquitetura relevante para entendimento deste trabalho. Conforme pode ser visto na Figura 3, dependendo da aplicação, uma pilha específica de protocolos de comunicação pode ser usada. O conjunto de protocolos destinado a uma aplicação específica é chamada de *profile*, ou perfil. Por exemplo, no que diz respeito a cuidados médicos, há perfis para monitoramento cardíaco, para controle da pressão arterial e para controle da glicose. Por padrão, um perfil necessita utilizar, no mínimo, três protocolos: um protocolo de rádio, um protocolo de controle de enlace (ou banda-base) e um protocolo de aplicação. A seguir, nas seções 2.1.1, 2.1.2, 2.1.3 e 2.1.4 são descritas as camadas típicas utilizadas pelo protocolo *BLE*.

### 2.1.1 Camada física

A camada física de rádio é a base da pilha. Semelhante à camada física do modelo *Open Systems Interconnection (OSI)*, essa camada tem a função de lidar com a transmissão e a modulação de rádio. Essa camada é um sistema de baixa potência com alcance de cerca de 10 metros que opera na banda de frequências *Industrial Medical Scientific (ISM)* não-licenciada<sup>2</sup> de 2,4 GHz.

No *BLE*, a banda é dividida em 40 canais de 2 MHz cada, sendo que os três últimos canais são reservados para a comunicação de anúncio e os demais utilizados para transferência de dados (SIG, 2016e) (GUPTA, 2013). A comunicação de anúncio existe apenas no *BLE* e serve para que um dispositivo inativo avise a outro dispositivo *BLE* - chamado de *scanner* ou mestre - que deseja transmitir dados. Esse recurso de inatividade fornecido pelo *BLE* é a base para a melhor eficiência energética em relação às versões anteriores do *Bluetooth*. Para otimizar ainda mais o consumo de energia, o *BLE* permite taxas de transferência de até 1Mbps, enquanto que as versões anteriores podem chegar a até 3 Mbps (GUPTA, 2013).

### 2.1.2 Camada de controle do enlace

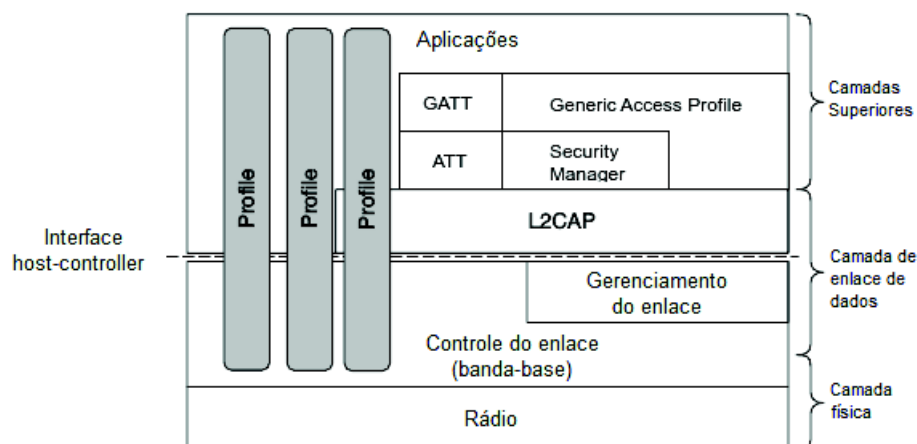
No *BLE*, como os dispositivos normalmente encontram-se com a comunicação inativa, a camada de enlace trabalha com cinco estados que definem a situação de um dispositivo *BLE* (GUPTA, 2013) (BATTISTI, 2016):

---

<sup>2</sup>As bandas de rádio *ISM* são divisões do espectro de radiofrequência reservadas internacionalmente. Por padrão, todo o espectro deve ser licenciado. Porém, três faixas (900 MHz, 2,4GHz e 5,8GHz) não precisam de licenciamento. Devido a isso, essas faixas são amplamente utilizadas em telecomunicações. A faixa de 2,4GHz é a mais utilizada atualmente por ser um padrão mundial (BATTISTI, 2016) (TANENBAUM; WETHERALL, 2011).



**Figura 3:** Arquitetura *BLE*.



Fonte: Adaptado de Tanenbaum e Wetherall (2011) e SIG (2016e)

- **Inativo:** o dispositivo encontra-se inativo, não enviando nem recebendo dados;
- **Anúncio:** o dispositivo que está no estado inativo passa para esse estado quando deseja enviar dados para outro dispositivo. Nesse estado, o dispositivo passa a ser chamado de anunciante ou escravo e faz esse anúncio por meio dos três canais dedicados explicados na Seção 2.1.1 e também pode responder a solicitações de outros dispositivos *BLE*;
- **Escaneamento:** nesse estado o dispositivo, chamado de mestre ou *scanner*, escuta anúncios de dispositivos anunciantes e pode solicitar informações adicionais a outros dispositivos;
- **Inicialização:** o dispositivo que encontra-se nesse estado chama-se de iniciador e tem a intenção de se conectar a um anunciante. Ao localizar um anunciante, ele responde requisitando o início de uma conexão. Normalmente um dispositivo mestre é quem passa para esse estado, após escanear um anúncio da rede;
- **Conexão:** os dispositivos iniciador e anunciante entram nesse estado quando a conexão é formada entre ambos, formando uma conexão *P2P*, e passam a se comunicar por intermédio dos canais de dados. Somente o dispositivo iniciador pode interromper uma conexão.

Um dispositivo *BLE* entra em estado de anúncio em períodos que podem variar entre 20ms a 10,24s, utilizando-se dos atributos *Advertising\_Interval\_Min* e *Advertising\_Interval\_Max* especificados na *API* do padrão (SIG, 2016e). Dessa forma, quanto maior for o intervalo de anúncio, maior o período de inatividade do dispositivo, salvando assim mais energia.

A camada de controle do enlace também define o formato do quadro para transmissão pela camada física. O *BLE* conta com um formato de quadro enxuto, de forma a diminuir a quantidade de dados a serem transmitidos. O *BLE* utiliza o mesmo quadro para fazer anúncios de

**Figura 4:** Estrutura de um quadro *BLE*.

Fonte: Retirado de SIG (2016e)

transmissão ou efetivamente transmitir dados, dependendo da configuração do quadro (SIG, 2016e). Esse quadro é mostrado na Figura 4 e contém os seguintes campos:

- **Preâmbulo:** possui o tamanho de 1 *byte*;
- **Endereço de acesso:** contém o endereço de acesso ao canal e ao dispositivo *BLE*. É atribuído de forma randômica pelo protocolo, por meio do canal de transmissão que é utilizado por cada par de dispositivos que formam a conexão *P2P*. Caso o quadro seja um quadro de anúncio, esse campo recebe um valor fixo. Possui o tamanho de 4 *bytes*;
- **Cabeçalho *PDU*:** caso trate-se de um quadro de anúncio, contém informações referentes ao tipo de *payload* que está sendo enviado: dados de anúncio ou dados de inicialização. Caso se trate de um quadro de transmissão, contém informações de controle da transmissão, como um identificador de controle do enlace lógico, o número sequencial do próximo quadro a ser recebido, bem como a informação de que se existe ainda algum quadro a ser transmitido. Possui o tamanho de 2 *bytes*;
- ***PDU Payload:*** contém os dados de anúncio ou transmissão, conforme o canal utilizado e os parâmetros informados no campo *PDU Header*. Possui o tamanho variável, podendo variar de 23 a 255 *bytes*;
- ***CRC:*** faz o *checksum* do quadro. Possui o tamanho de 3 *bytes*.

### 2.1.3 Camada de aplicações

Por fim, a última das camadas básicas do *BLE* é chamada de camada de aplicações. Ela possui as aplicações *BLE* propriamente ditas. Conforme a Figura 3, entre as camadas de controle de enlace e a camada de aplicações, temos uma linha chamada *interface host-controller*, ou interface hospedeiro-controlador (SIG, 2016e), que define o que é implantado no dispositivo *BLE* e o que é implantado no *chip Bluetooth* instalado no dispositivo. Os protocolos localizados acima da linha normalmente são implementados no dispositivo, enquanto que os protocolos abaixo da linha são normalmente implementados no *chip*.

### 2.1.4 Demais camadas

Além das camadas básicas, grande parte dos perfis *BLE* necessitam de funcionalidades que não estão contidas nos três protocolos base da arquitetura, conforme demonstrado na Figura

3. Dessa forma, outros protocolos adicionais são normalmente utilizados para comunicação *Bluetooth*. Entre esses protocolos, destacam-se o protocolo de controle adaptativo do enlace lógico (*L2CAP*), o protocolo de gerenciamento de segurança (*SMP*) e o protocolo de perfil de acesso genérico (*GAP*).

O protocolo de controle adaptativo do enlace lógico, localizado após a camada de controle do enlace e também conhecido por *Logical Link Control Adaptation Protocol (L2CAP)*, possui quatro funcionalidades principais. A primeira é, no dispositivo transmissor, aceitar pacotes de até 64KB das camadas superiores e fracioná-los em quadros comportados pela camada de controle do enlace e fazer o processo inverso no dispositivo receptor (TANENBAUM; WETHERALL, 2011). No caso do *BLE*, o *L2CAP* fraciona os pacotes das camadas superiores em quadros de tamanho máximo de 48 *bytes* (SIG, 2016d). A segunda funcionalidade é realizar, no dispositivo transmissor, a multiplexação de várias origens de pacotes, realizados por diferentes protocolos localizados na camada imediatamente acima, e entregá-los à camada correspondente no dispositivo receptor. A terceira funcionalidade é realizar o controle de erros e retransmissão. Por fim, a última função do *L2CAP* é impor requisitos de qualidade de serviço (*QoS*) entre múltiplos enlaces. Apesar de o *L2CAP* ser um protocolo opcional, seu uso torna-se obrigatório para a implementação do *6LowPAN*, conforme tratado na Seção 2.3.1.

Logo acima do *L2CAP* há uma série de protocolos que podem utilizar os serviços fornecidos por este protocolo e, conforme já comentado, são multiplexados pelo protocolo *L2CAP*. No *BLE*, dois protocolos bastante utilizados são o protocolo de gerenciamento da segurança, também conhecido por *Security Manager Protocol (SMP)* e o protocolo de perfil de acesso genérico, também conhecido por *Generic Access Profile (GAP)* (PESSOA, 2016). O protocolo *GAP* permite que os dispositivos *Bluetooth* interoperem entre si, definindo um comportamento padrão para cada perfil *BLE*, oferecendo um conjunto de serviços específicos por intermédio do protocolo de atributo genérico, também chamado de *GATT*. Cada serviço, ainda, possui uma ou mais características. Cada perfil, serviço e característica possui uma identificação em hexadecimal que o define na arquitetura *BLE*. Alguns exemplos de perfis *GAP* - e seus respectivos identificadores - são mostrados na Tabela 1.

Já o protocolo *SMP* tem a função de fornecer aos protocolos das camadas inferiores a possibilidade de troca segura de chaves de segurança entre os dispositivos *BLE*, caso configurado. Essa troca é possível por meio de três etapas:

- Pareamento: gera uma chave criptográfica comum temporária para criação de um enlace provisório seguro que permita a troca de uma chave criptográfica permanente;
- Ligação: gera chaves criptográficas permanentes entre os dispositivos e realiza a troca dessas chaves de forma a estabelecer um canal seguro permanente;
- Restabelecimento: Reestabelece a comunicação segura entre os dispositivos a partir das chaves permanentes estabelecidas na etapa de ligação.

**Tabela 1:** Tabela de serviços *GAP*

<b>Código</b>	<b>Serviço</b>
0x180D	Serviço de monitoramento cardíaco
0x1810	Serviço de pressão arterial
0x181F	Serviço de monitoramento de glicose
0x180F	Serviço de monitoramento da bateria

Fonte: Adaptada de SIG (2017)

## 2.2 Padrão *IEEE 802.15.4*

O padrão *IEEE 802.15.4* foi criado pela *IEEE* em 2003 tendo como objetivo principal fornecer um padrão de conexão para redes pessoais sem fio, ou *Wireless Personal Area Network (WPANs)* de baixo custo e com baixo consumo de energia (*IEEE*, 2011), de maneira análoga ao *BLE*. O padrão opera nas faixas ISM não licenciadas de 868 MHz, 915 MHz e 2,4 GHz. A taxa bruta de transferência do padrão é de 250 KB/s, considerada muito baixa se comparada às outras tecnologias de transmissão sem fio, porém suficiente para dispositivos *IoT*.

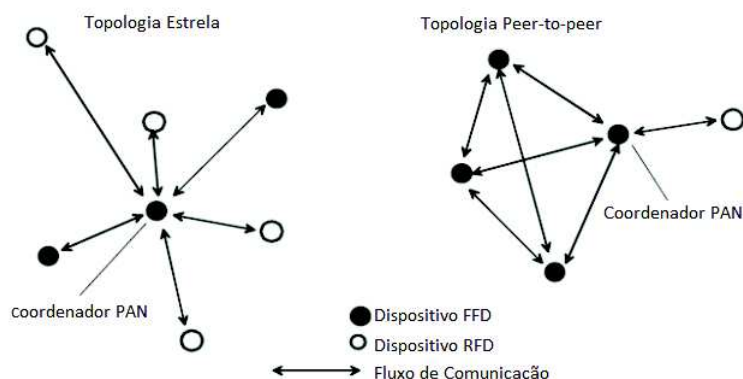
O padrão é composto basicamente por dois componentes: os dispositivos e os nós coordenadores (*IEEE*, 2011). Um dispositivo normalmente tem alguma aplicação associada e é o ponto inicial ou o ponto final da rede de comunicação. Já o nó coordenador, também chamado de *PAN coordinator*, também pode ter uma aplicação associada, porém ele também pode realizar as tarefas de início, fim ou roteamento de comunicações por meio da rede. O nó coordenador é o controlador primário da rede no padrão *IEEE 802.15.4*, portanto qualquer rede precisa de, no mínimo, um nó coordenador.

Os dispositivos podem ser de dois tipos: um dispositivo de função completa (também chamado de *full-function device - FFD*), ou um dispositivo de função reduzida (também chamado de *reduced-function device - RFD*) (*IEEE*, 2011). Um *RFD* é um dispositivo com baixa capacidade de processamento, baixa capacidade de memória e baixa capacidade de comunicação, portanto destinado a aplicações extremamente simples. Já um *FFD* é um dispositivo com mais poder de processamento, maior memória e mais capacidade de comunicação. Naturalmente, somente um *FFD* pode se tornar um nó coordenador numa *WPAN* padrão *IEEE 802.15.4*.

O padrão pode operar nas topologias estrela ou *peer-to-peer (P2P)* (*IEEE*, 2011). Na topologia estrela, o nó coordenador é o nó central da rede, e toda a comunicação da rede passa por ele. Já na topologia *P2P* também possui um nó coordenador, porém a comunicação pode ser realizada diretamente entre os dispositivos que compõem a rede, conforme mostrado na Figura 5.

Conforme pode ser verificado na Figura 6, a arquitetura do padrão *IEEE 802.15.4* é formada por apenas duas camadas: a camada física (*PHY*) e a subcamada de acesso ao meio (*MAC*) (*IEEE*, 2011), equivalentes, respectivamente, às camadas física e à subcamada de acesso ao meio descritas em modelos como *Wi-Fi* e *Bluetooth* e outras tecnologias baseadas no modelo

**Figura 5:** Topologias de rede do padrão *IEEE 802.15.4*



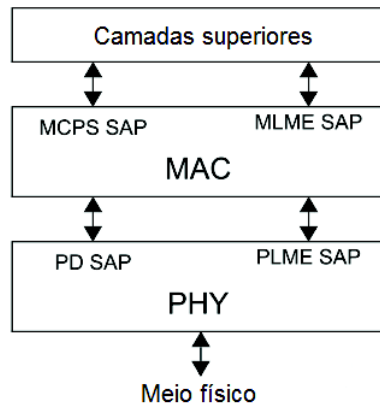
Fonte: Retirado de IEEE (2011)

*OSI*. As camadas superiores não são abordadas pelo padrão, porém podem ser integradas ao mundo *IP* por meio do uso do modelo *6LowPAN*, a ser abordado na Seção 2.3.

A camada física possui o serviço de dados físico e o serviço de gerenciamento físico. O serviço de dados possibilita a transmissão e recepção de pacotes de dados *PHY*, ou *PHY protocol data units (PPDUs)*, ou unidades de dados de protocolo físico. Já o serviço de gerenciamento provê basicamente mecanismos de controle das configurações de rádio. Além desses serviços, a camada física também possui as seguintes funcionalidades:

- Ativa e desativa o rádio *tranceiver*;
- Seleciona a frequência e troca de canal;
- Gera a frequência portadora e detecta o sinal;
- Executa a transmissão e recepção de dados, por meio de *PPDUs*;
- Detecta os níveis de energia dos canais. Essa funcionalidade, chamada de *Energy Detection (ED)*, fornece à subcamada *MAC* a informação de picos de energia entre os canais.
- Mede a qualidade do enlace a partir dos *PPDUs* recebidos. Essa funcionalidade é chamada de *Link Quality Indication (LQI)*;
- Verifica a ocupação do canal, de forma a controlar o acesso ao canal por meio do *CSMA/CA*. Essa funcionalidade é chamada de *Clear Channel Assessment (CCA)*.

Já a subcamada *MAC* é responsável por manipular todo o acesso o canal físico de rádio e possui dois modos de operação: *beacon* ativo e *beacon* inativo. O modo *beacon* ativo é normalmente utilizado quando há um nó coordenador e emprega um mecanismo cíclico de trabalho que visa reduzir o consumo de energia causado pela comunicação dos dispositivos (KARTHI; RAO; PILLAI, 2015). Esse modo funciona por meio de estruturas chamadas de superquadros, ou *superframes*.

**Figura 6:** Camadas do padrão *IEEE 802.15.4*

Fonte: Retirado de IEEE (2011)

Um superquadro é delimitado por *beacons* que são enviados pelo nó coordenador, e é composto por um período de atividade e outro período de inatividade, onde o dispositivo poupa energia. O período de atividade inicia após o recebimento de um *beacon* pelo dispositivo, e possui um tempo de acesso de contenção (*CAP*) onde o dispositivo disputa o acesso ao meio utilizando-se do protocolo *Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)* e, em seguida, há um intervalo tempo de transmissão garantido para cada dispositivo, chamado de *Guarantee Time Slots, (GTS)*. O conjunto de *GTS* compõe o intervalo de tempo chamado de *Contention Free Period (CFP)*. Após o *CFP*, o dispositivo pode entrar em inatividade até a chegada do próximo *beacon*. Essa estrutura é mostrada na Figura 7.

Os períodos de atividade e inatividade do superquadro são definidas pelas Fórmulas 2.1 e 2.2, a seguir:

$$PerodoAtivo = BaseSuperQuadro \cdot 2^{SO} \text{ simbolos} \quad (2.1)$$

onde *SO*, chamado de *Superframe Order*, varia entre 0 e 14, e Base Super Quadro é o produto da quantidade de *slots* e a duração de um *slot*.

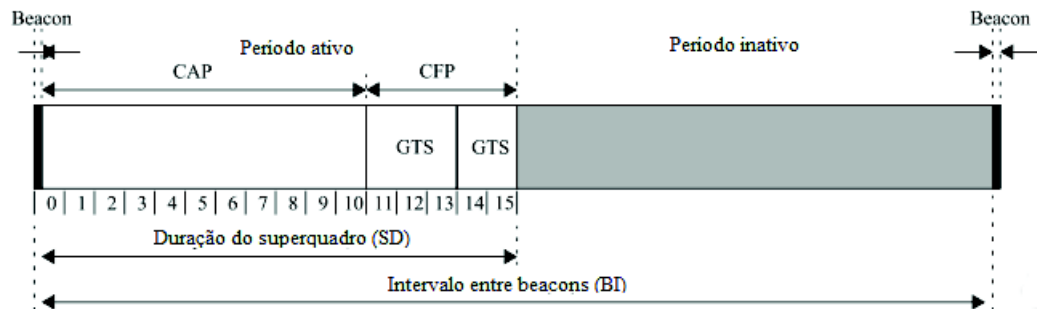
$$PerodoInativo = PerodoAtivo - (BaseSuperQuadro \cdot 2^{BO} \text{ simbolos}) \quad (2.2)$$

onde *BO*, chamado de *Beacon Order*, varia entre 0 e 14 e é sempre igual ou menor que *SO*, e Base Super Quadro é o produto da quantidade de *slots* e a duração de um *slot*.

Já o modo *beacon* inativo não trabalha com os *beacons* enviados pelos coordenadores e não possui um tempo específico para transmissão. Esse modo é normalmente utilizado em redes *PAN* onde os onde não há um nó coordenador fixo. Este modo tem os parâmetros *BO* e *SO* ajustados para o valor 15. Além dos modos de operação abordados, a subcamada *MAC* também possui as seguintes funcionalidades:

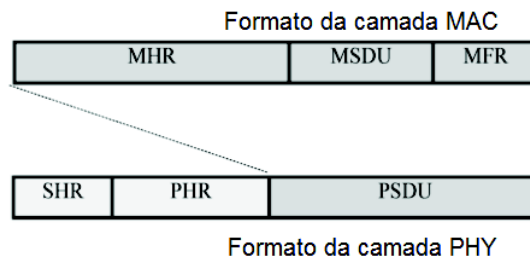
- Gera os *beacons* de informação e sinalização da rede, em intervalos determinados, caso o nó seja o coordenador da rede;

**Figura 7:** O superquadro do protocolo *IEEE 802.15.4*



Fonte: Retirado de IEEE (2011)

**Figura 8:** O estrutura de quadros no padrão *IEEE 802.15.4*



Fonte: Retirado de Karthi, Rao e Pillai (2015)

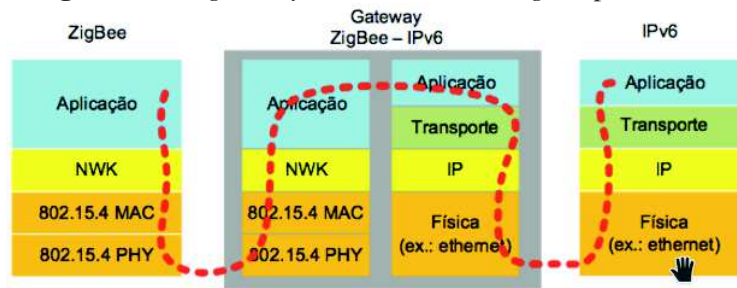
- Sincroniza os *beacons* entre os diversos nós;
- Executa a associação e desassociação dos dispositivos em uma *PAN*, bem como inicia uma rede *PAN*, caso trate de um nó coordenador;
- Controla o acesso ao canal, por meio da detecção de energia *ED*, fornecida pela camada *PHY*;
- Suporta a segurança de dispositivo.

Os quadros das camadas do *IEEE 802.15.4* são mostrados na Figura 8. Na camada *PHY*, o campo *SHR* ajuda na sincronização do receptor com a transmissão que chega até ele. Já o campo *PHR* contém o tamanho do *payload* do quadro. O *payload*, também chamado de *PHY Service Data Unit (PSDU)*, contém o quadro da subcamada *MAC* e tem o tamanho máximo de 127 bytes.

No quadro *MAC*, o campo *MHR* basicamente indica os endereços de origem e destino do quadro, indica a qual rede *PAN* o quadro pertence, bem como informa se haverá um cabeçalho adicional de segurança pode fornecer diversos mecanismos de segurança às camadas superiores. Esse cabeçalho de segurança possui informações que informam se o quadro será autenticado e se seu *payload* será criptografado. Já o campo *MSDU* indica o *payload* do quadro e o campo *MFC* faz a verificação de integridade do quadro.



**Figura 9:** Um *gateway* de conversão do Zigbee para o IPv6



Fonte: Retirado de IPv6.Br (2013)

### 2.3 6LowPAN

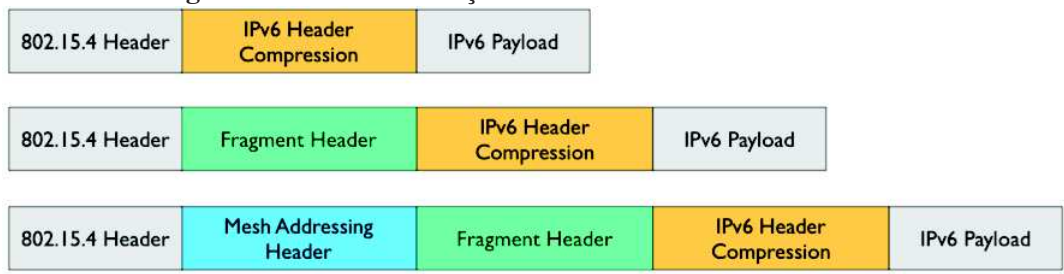
Por muitos anos, o protocolo *IP* n o foi considerado uma opç o para uso em redes de sensores sem fio, devido   complexidade de seu cabeçalho que acabava gerando um *overhead* na rede (IPv6.Br, 2013). Para realizar as tarefas correspondentes   camada de rede *Open Systems Interconnection (OSI)*, utilizavam-se outros protocolos - normalmente propriet rios, como o protocolo *Zigbee* (HUI; CULLER; CHAKRABARTI, 2009). Todavia, esses protocolos nativamente compat veis com a *Internet*, sendo necess rio o uso de *gateways* para a convers o, a n vel de aplicaç o, dos dados gerados no padr o *IEEE 802.15.4* para o protocolo *IP* (IPv6.Br, 2013) (Riazul Islam; Humaun Kabir; HOSSAIN, 2015). A Figura 9 mostra o processo de convers o de dados do *Zigbee*, que utiliza o modelo *IEEE 802.15.4*, para a comunicaç o *IPv6*.

Para resolver esse problema, o *Internet Engineering Task Force (IETF)* prop s, em 2007, a criaç o de uma camada que faça a convers o de pacote *IPv6* para quadros que possam ser suportados pelas redes *IEEE 802.15.4* (IETF, 2007) (HUI; CULLER; CHAKRABARTI, 2009). Essa camada acabou recebendo o nome de *6LowPAN*.

O *6LowPAN*   formado por tr s componentes prim rios: compress o de cabeçalho, fragmenta o e encaminhamento a n vel de enlace (HUI; CULLER; CHAKRABARTI, 2009). A compress o de cabeçalho tem a funç o de reduzir um cabeçalho do *IPv6* para um formato que possa ser carregado pela subcamada *MAC* do *IEEE 802.15.4*. Para realizar essa tarefa, alguns campos do cabeçalho *IPv6* s o omitidos e outros s o reduzidos. J  a fragmenta o realiza a tarefa de fragmentar um pacote *IPv6* em quadros que possam ser suportados pela subcamada *MAC* do *IEEE 802.15.4*. Por fim, a funcionalidade de roteamento a n vel de enlace permite que endereços de enlace local *IPv6* possam ser roteados em n vel de enlace como de forma an loga ao protocolo *IP*.

Cada uma dessas funcionalidades   definida por meio de um cabeçalho do *6LowPAN*: cabeçalho de fragmenta o (*fragment header*), cabeçalho de endereçamento *mesh* (*mesh addressing header*) e cabeçalho de compress o (*compression header*). Cada um desses cabeçalhos   opcional e pode ser usado de acordo com as necessidades de cada rede (IETF, 2007) (HUI; CULLER; CHAKRABARTI, 2009). Algumas possibilidades de uso desses cabeçalhos s o apresentados



**Figura 10:** Pilhas de cabeçalhos *6LoWPAN* normalmente usadas

Fonte: Retirado de Hui, Culler e Chakrabarti (2009)

na Figura 10.

O cabeçalho de fragmentação possui o tamanho de 5 *bytes* e é usado quando o *payload* do *IPv6* é maior que o campo *MSDU* da subcamada *MAC* do *IEEE 802.15.4*. De forma semelhante ao protocolo *IP*, ele possui campos que identificam a posição do fragmento e o tamanho total do *payload* do datagrama *IP*. Já o cabeçalho de endereçamento *mesh* possui que varia de 5 a 17 *bytes*, dependendo do tipo de endereçamento usado, e permite o roteamento de pacotes a nível de enlace no *IEEE 802.15.4*. Ele possui campos que identificam a origem, o destino e a quantidade máxima de saltos que o quadro pode ter numa rede *IEEE 802.15.4*. Por fim, o cabeçalho de compressão, também chamado de *HC1*, possui tamanho de 4 *bytes* e omite campos desnecessários para as redes *IEEE 802.15.4*, assumindo valores que são normalmente usados, bem como comprime os endereços *IPv6* de enlace local, que possuem tamanho de 128 *bits*, para endereços de apenas 2 *bits*. Além do *HC1*, há outro cabeçalho de compressão chamado de *HC2*, que realiza a compressão de cabeçalhos *UDP*. O *HC2* obrigatoriamente deve vir após um cabeçalho *HC1*. O *HC2* reduz o tamanho dos campos de origem e destino do *UDP* de 16 para apenas 4 *bits* cada campo. Isso é possível pois o *6LoWPAN* usa apenas o range de portas *UDP* 61616 – 61631, podendo assim omitir os 12 *bits* mais significativos.

### 2.3.1 *6LoWPAN* no *Bluetooth*

O mesmo problema de compatibilidade com o protocolo *IP* existente no padrão *IEEE 802.15.4* é verificado no *Bluetooth*. Percebendo esse problema, em 2013, um grupo da *IETF* propôs a adaptação do *6LoWPAN* para o *BLE*, onde um pacote *IPv6* é convertido para um formato aceito pelo protocolo *L2CAP*, materializando-se na RFC 7668 (*IETF*, 2015). Essa RFC descreve o uso de *6LoWPAN* em dispositivos *BLE*. Desde a versão 4.2 (*SIG*, 2016f), o *6LoWPAN* pode ser usado em redes *Bluetooth*.

No *BLE*, o *6LoWPAN* é responsável por realizar a conversão de um pacote *IPv6* de forma que os dados possam ser enviados às camadas inferiores do *BLE* por intermédio do protocolo *L2CAP*. Conforme visto na Seção 2.1.4, uma das funções do *L2CAP* é fracionar os pacotes das camadas superiores de forma que possa ser transmitido pelas camadas inferiores do *Bluetooth*, sendo que no *BLE*, a partir da versão 4.2, o *payload* do quadro é de de até 255 *bytes*.

O *6LowPAN* para *BLE* possui pequenas variações em relação ao *6LowPAN* original, apresentando as seguintes mudanças (IETF, 2015):

- Não há a necessidade de utilização de um cabeçalho específico para fragmentação dos pacotes, pois o protocolo *L2CAP* já implementa essa funcionalidade.
- Não há a necessidade de utilização do cabeçalho de endereçamento *mesh*, pois o *BLE* não suporta redes *mesh*.
- O cabeçalho de compressão *HCI* é adaptado para reduzir endereços de enlace local para endereços *BLE* de 48 *bits*.

## 2.4 *WBAN*

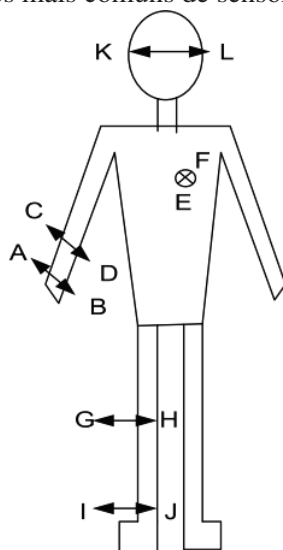
Uma *WBAN* (Wireless Body Area Network), também conhecida apenas como *BAN*, é uma rede de sensores sem fio (RSSF) utilizada para interconectar sensores implantados dentro de um corpo de uma pessoa ou dispostos sobre o corpo de uma pessoa (IEEE, 2016), utilizados, entre outras, para aplicações militares, de entretenimento pessoal e, especialmente, cuidados médicos (ULLAH et al., 2012) (CASTILLEJO et al., 2013).

Considerada por Shankar e Tomar (2016) uma subárea das RSSF, ou *Wireless Sensor Network* (*WSN*), uma *WBAN* é formada por dispositivos chamados de *motes*. Esses dispositivos possuem baixa capacidade de armazenamento e processamento e possuem restrições de consumo de energia, visto que seu tamanho reduzido não permite que sejam utilizadas baterias com grande capacidade de armazenamento de energia (KO et al., 2010) (ULLAH et al., 2012) (GOSWAMI et al., 2016). Os *motes* comunicam-se normalmente utilizando-se dos padrões *BLE* e *IEEE 802.15.4*, de forma que a troca de dados entre esses dispositivos consuma o mínimo de energia. Dessa forma, esses dispositivos têm capacidade suficiente apenas para processar e repassar informação para uma estação-base para posterior diagnose e análise por outros dispositivos computacionais (ULLAH et al., 2012)

As *WBANs* diferem das demais RSSF pelo fato de que as primeiras têm um foco maior em consumo de energia em relação ao desempenho, enquanto que as últimas preocupam-se primeiramente no desempenho da rede em relação ao consumo de energia (ULLAH et al., 2012). Além disso, as RSSF contemplam o uso de sensores em diversos ambientes inteligentes, tais quais prédios, casas e veículos. Dessa forma, uma *WBAN* faz uso de protocolos como *BLE* e *IEEE 802.15.4* para realizar a comunicação entre os sensores e os demais dispositivos e aplicações que requisitam esses dados.

Ainda, as redes *WBAN* diferenciam-se das demais RSSF por serem formadas por *motes* que possuem normalmente apenas um sensor acoplado que realiza a medição de um ou mais sinais vitais (SHANKAR; TOMAR, 2016). Esses sensores são classificados em dois tipos: intrusivos e não-intrusivos (SHANKAR; TOMAR, 2016). Os sensores intrusivos são inseridos dentro do corpo humano por meio de procedimento cirúrgico e são usados para medições específicas,

**Figura 11:** Localizações mais comuns de sensores corporais não intrusivos



Fonte: Retirado de Ullah et al. (2012)

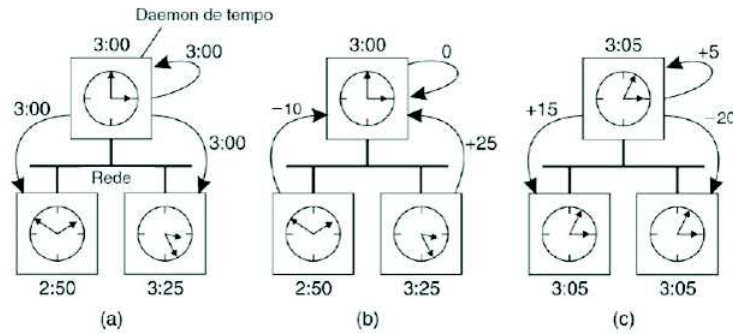
como a verificação da pressão intra-craniana. Já os sensores não-intrusivos, também conhecidos como *wearables*, são fixados externamente ao corpo humano utilizando-se de pingentes, braceletes, meias, entre outros, e normalmente realizam a medição de sinais vitais, como pressão arterial, temperatura corporal, batimento cardíaco, etc.

Em se tratando de sensores não intrusivos, as localizações mais utilizadas são mostradas na Figura 11. Eles podem estar em volta das mãos (Figura 11 A-B), em volta dos braços (Figura 11 C-D), em frente ou atrás do peito (Figura 11 E-F), em volta da coxa (Figura 11 G-H), em volta do tornozelo (Figura 11 I-J) ou próximo às orelhas (Figura 11 K-L).

## 2.5 Sincronização

As redes *WBAN* podem ser consideradas um sistema distribuído, devido ao fato de que tratam-se de redes cujos dispositivos comunicam-se e coordenam suas ações por meio de trocas de mensagens (COULOURIS et al., 2013). A sincronização é um problema importante em sistemas distribuídos e, portanto, em redes *WBAN*. Por exemplo, para sabermos em que hora do dia um evento específico ocorreu num *mote* que forma a rede é necessário configurar seu relógio com uma fonte de tempo confiável e aceita por todos os componentes que formam o sistema distribuído (COULOURIS et al., 2013) e, conseqüentemente, a rede *WBAN*. De maneira geral, existem duas formas de efetuarmos a sincronização entre dispositivos que compõem um sistema distribuído. A primeira forma é por meio da sincronização aproximada de relógios físicos por meio da troca de mensagens, enquanto que a segunda maneira é por intermédio do uso de relógios lógicos, onde a ordem em que os eventos ocorrem é que define a sincronização entre os dispositivos.

Uma forma de sincronização de relógios físicos utiliza o método de Cristian (COULOURIS

**Figura 12:** O funcionamento do algoritmo de Berkeley

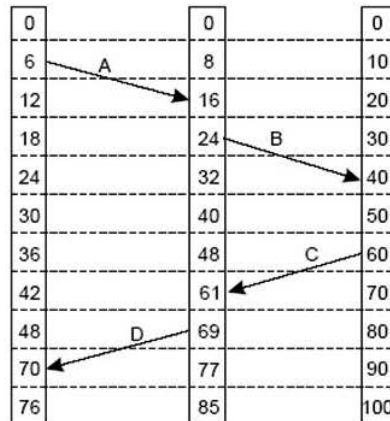
Fonte: Retirado de Tanenbaum e Van Steen (2007)

et al., 2013) (TANENBAUM; VAN STEEN, 2007). Esse método utiliza sincronização externa, onde um dispositivo envia uma mensagem  $M$  a um servidor conectado a uma fonte de tempo universal coordenada (*Universal Time Coordinated - UTC*) externa. O servidor, então, envia uma mensagem  $M_t$  ao dispositivo, informando o tempo  $T$ . O dispositivo, ao receber a mensagem, calcula o tempo transcorrido entre o envio da mensagem e o retorno da mensagem do servidor ( $T_{viagem}$ ). O cálculo normalmente efetuado considera  $T + T_{viagem}/2$ . Após o término do cálculo, então, o dispositivo atualiza seu relógio físico com o novo horário.

Outra maneira de sincronização de relógios físicos é por intermédio do algoritmo de Berkeley (COULOURIS et al., 2013). Esse algoritmo utiliza sincronização interna, onde um computador coordenador é escolhido para atuar como mestre. Ao contrário do método de Cristian, o dispositivo mestre periodicamente consulta os outros dispositivos - chamados de escravos - cujos relógios devem ser sincronizados, conforme mostra a Figura 12 (a). Os dispositivos escravos, então, retornam ao mestre os seus valores de relógio, conforme Figura 12 (b). O mestre, assim, faz uma estimativa dos tempos locais desses relógios, observando os tempos de ida e volta das mensagens, e faz uma média dos valores obtidos - incluindo o valor de seu próprio relógio. Efetuado o cálculo, o dispositivo mestre envia o valor pelo qual o relógio de cada escravo deve ser ajustado, conforme Figura 12 (c). O valor pode ser positivo ou negativo.

Porém, a sincronização por meio do uso de relógios físicos não é perfeita. A outra maneira de sincronização de dispositivos, portanto, é por meio da utilização de relógios lógicos de Lamport (TANENBAUM; VAN STEEN, 2007). O relógio lógico é um contador de *software* que aumenta a contagem conforme vão ocorrendo os eventos lógicos. A Figura 13 mostra o funcionamento do relógio lógico de Lamport onde os tempos são corrigidos a cada troca de mensagem.

Um problema do relógio lógico de Lamport é a impossibilidade de confirmarmos a ordem de execução de todos os eventos ocorridos num sistema distribuído. Para contornar essa fraqueza, então, surgiram os relógios vetoriais (TANENBAUM; VAN STEEN, 2007). Um relógio vetorial para um sistema de  $N$  processos é um vetor de  $N$  inteiros (COULOURIS et al., 2013). Cada um dos processos possui seu relógio vetorial  $V_i$ , o qual utiliza para indicar o tempo dos eventos

**Figura 13:** O funcionamento do relógio de Lamport

Fonte: Retirado de Tanenbaum e Van Steen (2007)

locais.

## 2.6 Considerações parciais

Neste capítulo foram abordados conceitos relacionados à comunicação de dispositivos que formam uma rede de sensores corporais, também chamada de *WBAN*. Os dispositivos que formam esta rede comunicam-se por intermédio dos padrões de comunicação *Bluetooth*, em sua versão *BLE*, e pelo padrão *IEEE 802.15.4*. Ambos padrões priorizam o baixo consumo de energia em suas comunicações, deixando em segundo plano questões relacionadas ao desempenho da comunicação. Para a interligação dos padrões *BLE* e *IEEE 802.15.4* pode-se utilizar o protocolo *6LowPAN* para comportar pacotes *IPv6* nos quadros desses padrões, permitindo assim que esses dispositivos possam ser diretamente ligados à Internet. Por fim, como as redes *WBAN* formam um sistema distribuído, foi explanado as formas de sincronização de dispositivos que formam este tipo de sistema.



### 3 TRABALHOS RELACIONADOS

Neste capítulo são apresentados os trabalhos relacionados que servem de base para elaboração da questão de pesquisa abordada na Seção 1.1 e da elaboração do modelo abordado no Capítulo 4. Em seguida, é feita uma comparação entre os trabalhos, analisando as lacunas de pesquisa encontradas.

O processo de busca pelos trabalhos relacionados utilizou como ferramenta de busca o *Google Acadêmico*. Como critério de exclusão, foram considerados os artigos publicados a partir do ano de 2012, ano de publicação do artigo "*Wireless body area networks for healthcare: A survey*" (CROSBY et al., 2012), que explica a situação do uso de redes *WBAN* em cuidados médicos. As palavras-chave utilizadas na pesquisa foram:

(*MIDDLEWARE AND SMARTPHONE AND "HEALTHCARE"*) AND  
(*"BODY AREA NETWORK" AND ENERGY*)

Como resultado da pesquisa acima, foram retornados 191 resultados. Para cada um dos resultados, foi lido o seu resumo de forma a selecionar os artigos que (1) permitiam a utilização de um *smartphone* com utilitário de verificação de saúde; (2) utilizavam algum tipo de rede de transmissão para coleta dos dados de sensores de uma *RSSF* e/ou que transmitiam dados coletados do próprio equipamento para uma aplicação externa e; (3) que possuíam algum tipo de integração com aplicações externas.

Nas Seções a seguir são apresentados os trabalhos relacionados selecionados a partir da metodologia apresentada.

#### 3.1 *SBMHM*

O *SBMHM*, acrônimo para *Smartphone-based Mobile Health Monitoring* (LEE; JEONG; YOON, 2012), apresenta um sistema de monitoramento de saúde que funciona a partir de um *smartphone*. A criação dessa solução foi motivada pelo fato do crescimento da procura por atendimento médico em qualquer lugar e a qualquer momento, ao invés dos atendimentos tradicionais em consultórios médicos e ambientes hospitalares.

Basicamente, a solução conta com uma unidade de monitoramento de sinais biológicos, chamado de *biosignal monitoring unit (BMU)*. O *BMU* é um hardware específico que contém sensores que realizam a captura de sinais vitais. Os dados capturados são transmitidos a um *smartphone Android* por meio de *Bluetooth*. Por meio de um aplicativo específico, o *smartphone* armazena os dados recebidos do *BMU* e também armazena dados de localização, por meio de *GPS*, caso o aparelho possua a tecnologia.

Esse aplicativo possui também um modo de monitoramento que permite que um usuário visualize, em tempo real, os resultados coletados pelos sensores. Por fim, os dados armazenados pela aplicação podem ser transmitidos para para aplicações externas por meio de *Wi-Fi* e/ou *3G*.



### 3.2 *BeWell+*

A solução *BeWell+* (LANE et al., 2014) é uma evolução da aplicação *BeWell* proposta por Lane et al. (2011) que tem por objetivo monitorar o bem-estar de uma pessoa, por meio de três dimensões de saúde: qualidade do sono, atividade física e interação social.

A solução funciona a partir de um aplicativo para *smartphone* que monitora as atividades de uma pessoa por meio do uso de um acelerômetro e de um microfone que já vêm instalados nos *smartphones*. A partir dos dados coletados, o aplicativo realiza inferências sobre esses dados e transmite os resultados para a nuvem *BeWell+*. A nuvem armazena os dados enviados por todos os *smartphones* que possuem a aplicação, formando assim uma comunidade. Como resultado, a nuvem devolve uma nota para a aplicação instalada no *smartphone*. Essa nota é chamada de nota de bem-estar, e é atualizada dinamicamente, conforme o bem-estar geral da comunidade.

Além disso, o aplicativo instalado no *smartphone* possui um módulo de gerenciamento de energia que otimiza o uso da bateria por meio do intervalo de troca de informações entre o *smartphone* e a nuvem. Esse intervalo é definido por meio da análise da variação do bem-estar do usuário. Quanto menor a variação do bem-estar, maior o intervalo de troca de informações, diminuindo assim o uso de *Wi-Fi* e/ou 3G.

### 3.3 *HyHealthAssistant*

Proposto por Seeger, Van Laerhoven e Buchmann (2015), o *HyHealthAssistant* é um *middleware* voltado a eventos projetado para gerenciar sensores de uma *WBAN*, e também sensores que estão localizados num ambiente, formando assim uma rede *BASN* (*Body and Area Sensor Network*). O *middleware* funciona captando os dados dos sensores de uma *BASN*. Os dados dos sensores do ambiente são enviados via *Wi-Fi*, enquanto que os dados dos sensores corporais são enviados via *Bluetooth*.

Esses dados são interpretados por um módulo chamado de *Event Composer*, criando eventos. O *Event Composer* também verifica a validade dos dados recebidos dos sensores, e cria eventos que informam isso. Em seguida, esses eventos são transmitidos por meio de um ou mais canais de transmissão. Um canal de transmissão contém eventos relacionados a um interesse específico, como exercícios físicos, telemedicina ou cuidados a idosos. Esses interesses podem ser previamente configurados. As aplicações *healthcare* podem se inscrever em um ou mais canais de transmissão, por meio do método *publish-subscribe*, recebendo assim apenas os eventos desejados.

Esse *middleware* ainda possui módulos que verificam o estado do sistema e detectam situações críticas (como a baixa carga de bateria) e que fornecem mecanismos de autenticação, fornecendo uma camada mínima de segurança.

Como resultado, o *middleware* apresenta uma significativa redução de consumo de energia, devido principalmente ao fato de tornar-se o único ponto de comunicação entre as aplicações



*healthcare* e os sensores, eliminando assim comunicações redundantes entre aplicações e sensores.

### 3.4 *EcoHealth*

O *EcoHealth*, acrônimo de *Ecosystem of Health Care Devices* (MAIA et al., 2014), é uma plataforma de *middleware Web* que conecta médicos e pacientes utilizando-se de sensores corporais ligados ao corpo, a fim de monitorar a saúde de um paciente e diagnosticar possíveis problemas de saúde (MAIA et al., 2015). A plataforma é capaz de interligar diversos sensores heterogêneos e notifica condições de saúde de um paciente, bem como seus sinais vitais.

O *middleware* possui um módulo de conexão de dispositivos, que interliga os diversos sensores ao *middleware* por intermédio de *drivers* baseados no padrão *REST* e no protocolo *HTTP*, necessitando assim de um servidor *web* para funcionar. Esses *drivers* estruturam os dados e os enviam ao módulo de manipulação de dados, onde os dados são tratados e armazenados no módulo de armazenamento, que contém um banco de dados relacional.

O *EcoHealth* conta também com um módulo de atuação que envia comandos do *middleware* para os dispositivos, permitindo que estes tenham sua configuração alterada. Serviços de infraestrutura são fornecidos pelo módulo de serviços comuns e a interface com o usuário é feita por meio do módulo de visualização e gerenciamento, que é uma interface gráfica *Web* que pode ser acessada a partir de um computador ou um *smartphone*.

### 3.5 *Home Health Gateway Based Healthcare Services Through U-Health Platform*

Jung et al. (2013) propõe um *gateway* doméstico voltado a cuidados médicos, chamada de *Home Health Gateway Based Healthcare Services Through U-Health Platform*. Essa solução propõe dietas, exercícios e outros cuidados médicos ao usuário a partir de decisões tomadas por um ambiente ubíquo e também permite que um profissional médico contate remotamente esse paciente por meio da solução.

A arquitetura desse *gateway* é formada por um gerente embutido que recebe os dados de dispositivos *Personal Health Device (PHD)* que atendem à especificação *ISO/IEEE 11073*<sup>1</sup> (*PHD*) via conexão *Bluetooth*. Os dados recebidos pelo *gateway* são manipulados e retransmitidos esses dados a clientes de visualização e interação com o usuário, como *smartTVs* e *tablets*, ou a servidores remotos por meio de *Wi-Fi*. Além disso, ele pode trabalhar com ubiquidade conectando-se a outros serviços de contexto, como previsão do tempo.

---

<sup>1</sup> A *ISO/IEEE 11073* define o padrão para armazenamento e transmissão de dados de saúde entre pacientes e seus dispositivos sem fio, médicos e centros de saúde, por meio do uso de perfis. Ela também define dispositivos que funcionam como agentes, capturando dados e transmitindo-os a um gerente, que armazenam e processam os dados recebidos (JUNG et al., 2013).

### 3.6 *WellPhone*

A solução proposta por Moser e Melliar-Smith (2015), chamada de *WellPhone*, trata um *smartphone* voltado exclusivamente para o monitoramento de saúde de uma pessoa. Além de coletar dados de sensores de sinais vitais pré-determinados de uma *WBAN*, ele fornece diversas análises estatísticas sobre os dados coletados e também faz a correlação entre a atividade física e as medições fisiológicas.

Nessa solução, os dispositivos se conectam ao *smartphone* por meio de uma conexão *Bluetooth* criptografada. Para cada dispositivo conectado há um módulo de processamento e visualização dos dados, que são armazenados em uma área de dados também criptografada.

A partir dos dados coletados pelos sensores ao longo do tempo, o *WellPhone* cria gráficos que mostram a evolução da condição de saúde de um usuário. Ele também interage com o usuário por meio do reconhecimento de voz, utilizando perguntas e respostas pré-determinadas. Caso não haja uma resposta e/ou haja alguma alteração significativa nos sinais vitais, a solução pode contatar um médico, um familiar ou um serviço de emergência por meio de uma conexão à *Internet* criptografada ou por meio de uma ligação telefônica.

No que diz respeito à segurança, além da criptografia nas conexões e no armazenamento dos dados, o *WellPhone* permite que a própria pessoa controle o acesso de terceiros aos dados armazenados no dispositivo.

### 3.7 *Integration of wearable devices in a wireless sensor network for an E-health application*

O *Integration of wearable devices in a wireless sensor network for an E-health application* (CASTILLEJO et al., 2013) é uma solução de *e-health* voltada para ambientes internos que interliga dispositivos *IoT* de *healthcare* que tem por objetivo monitorar a condição de saúde de um paciente quanto este está praticando algum exercício físico. Isso é possível pois a solução trabalha interligando uma *WBAN* a uma rede de sensores sem fio que pode possuir outros dispositivos *IoT* em um ambiente que podem atuar em conjunto com *wearables* de uma *WBAN*.

Para isso, a solução conta com um subsistema de operação, onde um agente solicita qual informação deseja obter (temperatura do ambiente, temperatura corporal, etc), por meio de um *smartphone* ou computador. Esse subsistema envia a requisição ao subsistema RSSF, onde um nó chamado de *Broker* verifica se a informação desejada pode ser obtida na própria rede. Em caso positivo, a informação é retornada ao subsistema de operação. Caso contrário, uma nova requisição é enviada ao subsistema *WBAN* que coletará os dados do(s) sensor(es) desejado(s), retornando-os ao subsistema RSSF que retornará os dados coletados ao subsistema de operação.

Para realizar a comunicação entre os três subsistemas, dois *gateways* são necessários. O primeiro, chamado de *Enterprise Serial Bus (ESB)*, interliga o subsistema de operação ao subsistema RSSF. O segundo gateway é chamado de *Bridge Sensor (BS)* e faz a ligação entre

os subsistemas RSSF e *WBAN*. Esses *gateways* são necessários pois cada subsistema utiliza um padrão diferente de comunicação. Os dispositivos do subsistema de operação comunicam-se por meio do padrão *REST*, os dispositivos do subsistema RSSF comunicam-se por meio do 802.15.4, e, por fim, o subsistema *WBAN* utiliza *Bluetooth* para comunicação entre os sensores.

### 3.8 Comparativo entre as soluções

As soluções apresentadas neste capítulo foram comparadas considerando os quesitos de segurança na transmissão de dados entre dispositivos de uma rede *WBAN*, escalabilidade da rede *WBAN*, interoperabilidade dos padrões de comunicação de redes *WBAN*, eficiência energética, utilização de sensores do próprio *smartphone*, integração com aplicações externas, forma de captura dos dados, tipos de sensores externos captados e sincronização entre sensores de uma rede *WBAN*. A Tabela 2 apresenta uma síntese da avaliação das soluções apresentadas neste capítulo.

No que diz respeito à segurança na transmissão de dados entre sensores que formam uma rede *WBAN*, apenas a solução apresentada na Seção 3.6 trata dessa questão quando conectado por meio do *Bluetooth*. A solução apresentada na Seção 3.4 utiliza o termo segurança no que diz respeito à utilização da nuvem como garantia de disponibilidade e integridade dos dados capturados.

Referente à escalabilidade da rede *WBAN*, apenas a solução apresentada na Seção 3.7 apresenta esse recurso, onde as tarefas de gerenciamento da rede são divididas entre os *smartphones*, conforme o crescimento da rede, principalmente pelo fato de utilizar não somente *wearables*, mas também sensores que podem estar num ambiente específico. Soluções apresentadas nas Seções 3.2 e 3.4 fornecem escalabilidade por meio do armazenamento de dados e processamento utilizando a nuvem.

No que diz respeito à interoperabilidade de padrões de comunicação de dispositivos de uma *WBAN*, todas as soluções apresentam algum tipo de comunicação entre o dispositivo e aplicações externas utilizando conexões G (3G, 4G) ou *Wi-Fi*. A solução apresentada na Seção 3.4 faz apenas referência ao uso do protocolo *HTTP*, não fazendo menção ao tipo de tecnologia das camadas física e/ou de enlace que são utilizadas. No que diz respeito à comunicação de redes *WBAN*, apenas a solução apresentada na Seção 3.2 não apresenta nenhum tipo de comunicação entre o *smartphone* e sensores, enquanto que a solução apresentada na Seção 3.4 utiliza um dispositivo *Arduíno* para comunicação entre os sensores e a plataforma de *e-health*. No que diz respeito à comunicação em uma rede *WBAN*, as soluções apresentadas nas Seções 3.1, 3.5 e 3.6 realizam a comunicação entre *smartphone* e uma rede *WBAN* por meio de *Bluetooth*, enquanto que as soluções apresentadas nas Seções 3.3 e 3.7 permitem a comunicação por meio dos padrões *Bluetooth* e *IEEE 802.15.4*.

Em relação à eficiência energética, a solução descrita na Seção 3.2 apresenta otimização no consumo de energia por intermédio de algoritmo próprio que avalia a periodicidade de intera-

ções com o usuário ou de comunicação os os servidores que formam a comunidade da solução. Já a solução apresentada na Seção 3.3 sugere uma otimização na eficiência energética pelo fato de ser um *middleware* que o torna um ponto único de comunicação das diversas aplicações externas e/ou do próprio *smartphone*, diminuindo assim a quantidade de comunicações entre os *smartphones* e os sensores de uma *WBAN*.

Referente à utilização de sensores do próprio *smartphone*, as soluções apresentadas nas Seções 3.2 e 3.6 fazem uso do acelerômetro e do microfone, enquanto que a solução apresentada na Seção 3.3 faz uso do acelerômetro. As outras quatro soluções não fazem uso de nenhum sensor disponível em *smartphones*.

No que diz respeito a integração dos dispositivos que capturam dados de sensores com aplicações externas, apenas as soluções apresentadas nas Seções 3.1 e 3.2 não permitem a comunicação do *smartphone* com aplicações externas. As demais soluções possuem algum mecanismo de integração dos dados capturados e armazenados no *smartphone* com outras aplicações, especialmente médicas.

Em relação a locais onde os dados de sensores de uma *WBAN* podem ser capturados, as soluções apresentadas nas Seções 3.1, 3.2, 3.3 e 3.7 podem capturar dados de sensores, especialmente os localizados em uma *WBAN*, em qualquer lugar, pois utilizam exclusivamente um *smartphone* para o envio dos dados. As demais soluções utilizam dispositivos que dependem de outros equipamentos fixos para o seu completo funcionamento.

No que diz respeito à sincronização entre sensores, nenhuma solução apresenta qualquer mecanismo de sincronismo, dificultando o estabelecimento de relações entre os dados captados pelos diferentes sensores instalados em uma *WBAN*.

### 3.9 Avaliação geral das soluções

A partir da avaliação das soluções apresentadas neste capítulo, percebe-se que, de maneira geral, as aplicações de cuidados médicos voltadas à *smartphones* focam-se basicamente em executarem a função de um *gateway* entre uma rede *WBAN* e aplicações externas, onde o *smartphone* captura os dados de sensores da rede *WBAN*, armazena-os, e, por fim, envia-os a aplicações *healthcare* que farão a análise desses dados, conforme a necessidade verificada ou à finalidade médica da aplicação.

Nesse sentido, o termo *middleware* é pouco utilizado, pois normalmente as aplicações que executam essas tarefas já possuem uma finalidade médica específica e, portanto, realizam algum tipo de processamento sobre os dados coletados.

No que diz respeito à comunicação entre *smartphone* e a rede *WBAN*, o *Bluetooth* é o padrão normalmente utilizado. O padrão *IEEE 802.15.4*, apesar de ser sido pensado para uso em RSSF, é ainda muito pouco usado. Isso se deve principalmente ao fato de que os *smartphones* atuais não contarem com a tecnologia *IEEE 802.15.4*. Outro problema existente para a popularização do *IEEE 802.15.4* é sua incompatibilidade com o protocolo *IP*. Todavia, esse é um problema

que pode ser resolvido por meio do uso do *6LowPAN*. No que tange à comunicação externa, todos os padrões atuais de comunicação a nível de enlace (*Wi-Fi*, *Bluetooth* e tecnologias G) são utilizadas.

A segurança na comunicação entre os dispositivos também é pouco explorada. Isto deve-se ao fato de que as redes *WBAN* tem seu foco na confiabilidade e e eficiência na transmissão dos dados, conforme menciona Lee, Jeong e Yoon (2012).

A escalabilidade da rede é trabalhada apenas nas soluções que utilizam outros sensores RSSF além dos sensores de uma *WBAN*. Essas redes, no momento, não fazem uso de grandes quantidades de sensores que justifiquem a preocupação com escalabilidade.

No que diz respeito à preocupação com consumo de energia dos sensores, os trabalhos limitam-se ao uso de tecnologias de conexão voltados à eficiência energética, como *BLE* e *IEEE 802.15.4*.

### 3.10 Lacunas de pesquisa

A partir do estudo dos trabalhos relacionados apresentados neste capítulo, verifica-se inicialmente que as soluções apresentadas não atendem em sua totalidade os requisitos necessários para que um *smartphone* seja um *gateway* genérico entre sensores corporais e aplicações externas voltadas à saúde localizadas no próprio dispositivo ou na *Internet*. Dessa forma, as soluções não atendem a todas funções de um *middleware* voltado para uma rede sensores corporais. Além dessa questão mais genérica, verificam-se também as seguintes lacunas de estudo:

- **Uso de sensores do *smartphone*:** uma das vantagens do uso de *smartphones* atuando como *gateways* entre *WBANs* e aplicações externas é a possibilidade de uso de sensores oferecidos pelo próprio dispositivo, como acelerômetros e *GPS*, que permitem o uso de dados de contexto para validação dos dados de sensores. Por exemplo, a temperatura corporal de uma pessoa aumenta caso esta esteja em atividade física, como uma corrida. Um dado coletado a partir de um sensor de temperatura pode ser incorretamente interpretado como febre a partir de sua análise individual, enquanto que os dados do sensor *GPS* poderiam afirmar que a pessoa estaria em movimento;
- **Eficiência energética:** um *middleware* que se propõe interligar sensores de uma *WBAN* a aplicações externas não deve limitar-se apenas à gestão de energia fornecidos pelos próprios protocolos de comunicação. Deve-se buscar meios de otimizá-los, de forma a prolongar a carga das baterias de sensores de uma rede *WBAN*;
- **Segurança:** o sistema deve oferecer uma solução completa de segurança, desde a forma de comunicação entre *WBAN* e *smartphone*, passando pelo armazenamento e processamento interno, e seu posterior envio a aplicações externas;
- **Escalabilidade:** apesar das redes *WBAN* serem relativamente recentes e planejadas para

**Tabela 2:** Comparativo entre os trabalhos relacionados

<b>Critério/ Trabalho</b>	<b>(LEE; JEONG; YOON, 2012)</b>	<b>(LANE et al., 2014)</b>	<b>(SEEGER; VAN LA- ERHO- VEN; BUCH- MANN, 2015)</b>	<b>(MAIA et al., 2015)</b>	<b>(JUNG et al., 2013)</b>	<b>(MOSER; MELLIAR- SMITH, 2015)</b>	<b>(CASTILLEJO et al., 2013)</b>
Segurança na transmissão	Não	Não	Não	Não	Não	Somente Bluetooth	Não
Escalabilidade	Não	Não	Não	Não	Não	Não	Sim
Interoperabilidade	3G, WiFi, Bluetooth	3G, WiFi	WiFi, Bluetooth	802.15.4, Bluetooth	WiFi, Bluetooth	3G, WiFi, Bluetooth	802.15.4, WiFi, Bluetooth
Eficiência energética	Não	Sim	Não	Não	Não	Não	Não
Uso de sensores do próprio <i>smartphone</i>	Não	Acelerômetro, microfone	Acelerômetro	Não	Não	Acelerômetro, microfone	Não
Captura dos dados dos sensores	Em qualquer lugar	Em qualquer lugar	Em qualquer lugar	Locais determinados	Locais determinados	Em qualquer lugar	Locais determinados
Sincronização entre sensores	Não	Não	Não	Não	Não	Não	Não

Fonte: Elaborada pelo autor.

poucos dispositivos, a tendência é de que a quantidade de sensores que uma pessoa poderá vestir aumente com o passar dos anos, necessitando que o recebimento e o processamento dos dados recebidos dos diversos sensores sejam divididos entre outros dispositivos. Isso permitirá que a rede *WBAN* possa tornar-se escalável.

- **Independência de protocolos:** O *middleware* deve ser capaz de trabalhar com os protocolos de comunicação de redes *WBAN Bluetooth* e *IEEE 802.15.4*;
- **Sincronização entre sensores:** nenhuma solução apresenta qualquer tipo de sincronização entre sensores, de forma a correlacionar temporalmente os dados enviados por cada um dos sensores.

### 3.11 Considerações parciais

Neste capítulo foram descritos e analisados sete trabalhos relacionados. Nessa análise, verificou-se a existência das lacunas de pesquisa no que diz respeito ao uso de sensores disponíveis no próprio *smartphone*, à segurança na transmissão de dados em sensores de uma *WBAN*, à eficiência energética na comunicação desses sensores, na escalabilidade de uma rede *WBAN*, na independência de protocolos de comunicação de sensores e na sincronização entre sensores de uma rede *WBAN*.

A partir da análise da Seção 3.10, verifica-se que a questão de pesquisa e a contribuição científica deste trabalho propõem-se a resolver algumas dessas lacunas. A primeira delas é reduzir a troca de dados entre os dispositivos que compõem uma rede *WBAN*, diminuindo, dessa maneira, o consumo energético desses dispositivos, independentemente de sua quantidade. A segunda é permitir a sincronização do tempo entre os dispositivos que compõem a rede *WBAN*, de forma que os dados coletados pelos diversos sensores possam ser correlacionados. E, por fim, a terceira é oferecer um nível de escalabilidade à rede *WBAN*.





## 4 MODELO MOMICARE

A partir da análise do Capítulo 3, verificou-se a inexistência de uma solução específica para comunicação de sensores corporais - especialmente os sensores voltados a cuidados médicos - a aplicações externas, onde um *smartphone* seja um *gateway* genérico entre esses sensores corporais e as diversas aplicações. Devido a isso, propõe-se a criação de um *middleware* que facilite a integração entre diversas aplicações médicas - sejam elas um aplicativo instalado no próprio *smartphone* para controle pessoal ou uma aplicação médica externa localizada em um centro médico, um consultório, um hospital ou uma aplicação instalada em um *smartphone* de um profissional médico, oferecendo um baixo consumo de energia e um sincronismo de tempo entre dispositivos que permita a correlação de dados médicos, bem como possibilita um certo grau de escalabilidade à rede *WBAN*.

### 4.1 Arquitetura do modelo

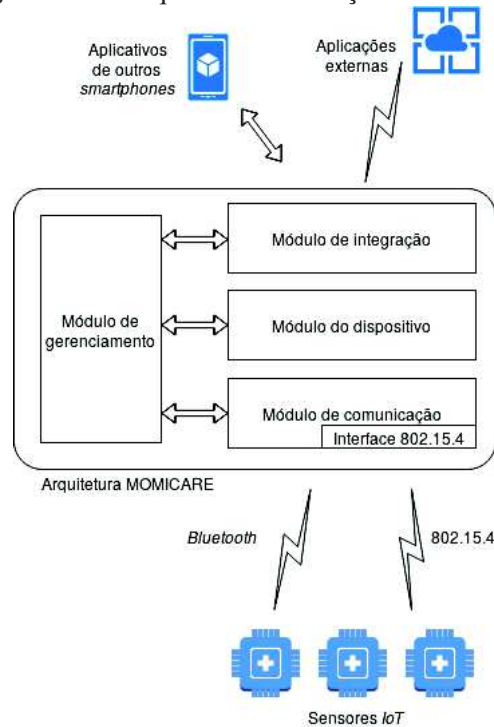
O *MOMICARE*, acrônimo de *MOBILE MIDDLEWARE FOR HEALTHCARE*, é um *middleware* projetado especialmente para atender as lacunas de sincronismo temporal entre dispositivos que formam uma rede *WBAN*, de escalabilidade da rede *WBAN* e de eficiência energética, lacunas essas identificadas no Capítulo 3. O *middleware* também atenderá secundariamente a requisitos que envolvem a segurança na comunicação entre sensores, a interoperabilidade entre sensores e a possibilidade de uso de sensores do próprio dispositivo.

Esse *middleware* possui quatro módulos: módulo de gerenciamento, módulo de integração, módulo do dispositivo e módulo de comunicação. Conforme a Figura 14, os sensores que formam uma rede *WBAN* comunicam-se com o *middleware* por intermédio do módulo de comunicação e armazenam os dados enviados no módulo do dispositivo. As aplicações externas acessam os dados armazenados no *middleware* por meio do módulo de integração, enquanto que o módulo de gerenciamento é responsável pela administração e comunicação entre os demais módulos.

#### 4.1.1 Módulo de Gerenciamento

O módulo de gerenciamento é responsável pela administração dos demais módulos, bem como pela configuração dos parâmetros do *middleware*, que afetarão o funcionamento dos demais módulos. Todos os demais módulos somente se comunicam com o módulo de gerenciamento, que controla a execução de todo o *middleware*. Para realizar tais tarefas, esse módulo possui uma interface para que o usuário possa configurá-lo. Esse módulo também comunica-se diretamente com os demais módulos. Entre as opções de configuração possíveis, estão as seguintes:

- **Configuração de uso dos sensores do dispositivo:** Permitir que o usuário ative ou de-

**Figura 14:** A arquitetura da solução *MOMICARE*

Fonte: Elaborado pelo autor

sative sensores do dispositivo, fornecidos por meio do módulo do dispositivo, que podem auxiliar da captura de dados.

- **Segurança:** Configurar parâmetros de criptografia e autenticação no armazenamento dos dados coletados, a serem executados pelo módulo do dispositivo; informar credenciais para autenticação de aplicações de terceiros instaladas no *smartphone* e externas, a serem executadas pelo módulo de integração; criptografar os dados enviados para aplicações externas, por meio de trocas de chaves a nível de aplicação, a serem executadas pelo módulo de integração; utilização de mecanismos de segurança a nível de enlace oferecidas pelos protocolos para segurança nas comunicações entre *smartphone* e rede *WBAN*, a serem executadas pelo módulo de comunicação.
- **Configurações de energia:** Configurar o intervalo de captura de dados dos sensores, bem como demais parametrizações nos protocolos *BLE* e *IEEE 802.15.4* que podem auxiliar na otimização do consumo de bateria, a serem executadas pelo módulo de comunicação.

O módulo de gerenciamento também é responsável por alertar o usuário sobre possíveis problemas na estrutura administrada por ele, a partir das informações fornecidas pelos demais módulos do *MOMICARE*.

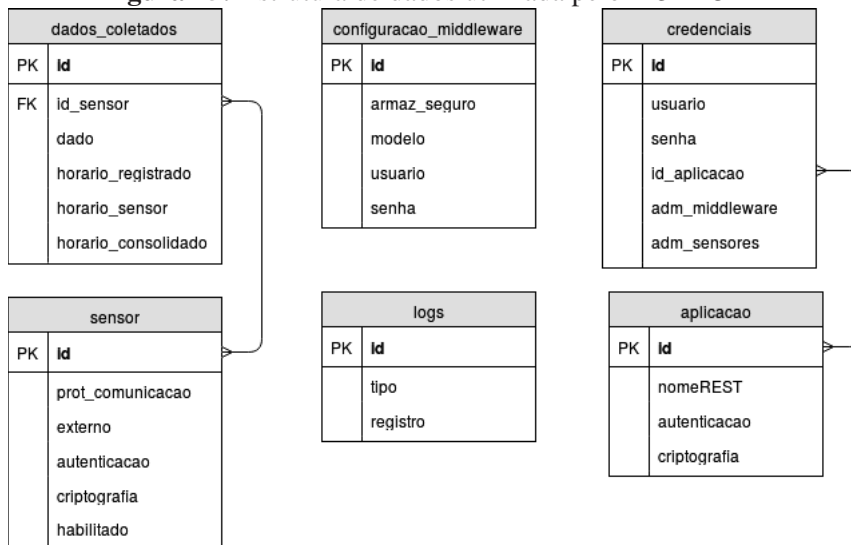
#### 4.1.2 Módulo do Dispositivo

O módulo do dispositivo do *MOMICARE* é responsável por administrar os recursos do *smartphone* que serão utilizados pelo *middleware*, como os sensores instalados no aparelho, a bateria, o armazenamento interno e os *hardwares* de comunicação existentes. Ele atende às requisições solicitadas pelo módulo de gerenciamento e relata a estas informações diversas referentes aos recursos utilizados pelo *middleware*. Entre eles, podem ser destacados os seguintes:

- Bateria: esse recurso deve ser monitorado de forma a verificar a carga restante do *smartphone*, alertando o usuário, por intermédio do módulo de gerenciamento, caso a carga entre em um estado crítico para que o mesmo possa tomar alguma providência. De forma a otimizar o consumo de bateria, o módulo do dispositivo verifica a carga da bateria a cada envio de dados dos sensores para o *middleware*;
- Armazenamento interno: o armazenamento interno do *smartphone* deve ser monitorado de forma a verificar se a quantidade de espaço livre, alertando o usuário, por meio do módulo de gerenciamento, caso o espaço de armazenamento entre em um nível crítico para que o mesmo possa tomar alguma providência. De forma análoga ao monitoramento da carga da bateria, o módulo do dispositivo verifica o espaço livre e armazenamento do *smartphone* a cada envio de dados dos sensores para o *middleware*;
- Sensores instalados: o *middleware* deve ser capaz de verificar quais sensores do *smartphone* estão disponíveis e repassar essa informação ao módulo de gerenciamento, quando solicitado por este. A disponibilidade de sensores disponíveis no dispositivo é repassada ao *middleware* pelo próprio sistema operacional instalado no *smartphone*. Esse módulo também deve ser capaz de repassar problemas e falhas encontrados nos sensores disponíveis no *smartphone* e que estão em uso. Exemplos de sensores normalmente instalados em *smartphones* e que podem ser utilizados em *healthcare* estão o *GPS*, acelerômetro, câmera e microfone.

A manipulação desses recursos é possível devido ao uso das *Application Programming Interface (API)* fornecidas pelos sistemas operacionais de dispositivos móveis. No *Android*, por exemplo, a manipulação de sensores pode ser feita pelas classes *JAVA Sensor* e *SensorManager*.

O módulo do dispositivo ainda é responsável por armazenar os dados de configuração do *middleware*, bem como os dados coletados pelos diversos sensores, conforme a estrutura de dados definida no modelo entidade-relacionamento (ER) representado na Figura 15. Como pode ser observado, o *MOMICARE* conta com uma tabela de configuração do próprio *middleware*, chamada de *configuracao\_middleware*, que registra os dados do aparelho e sistema operacional ao qual está instalado, bem como grava as credenciais de acesso aos recursos de configuração do *middleware*. O registro e configuração dos sensores controlados pelo *MOMICARE* fica armazenado na tabela *sensor*, onde ficam registrados o sensor encontrado, seu protocolo de

**Figura 15:** Estrutura de dados utilizada pelo *MOMICARE*

Fonte: Elaborado pelo autor

comunicação e se serão utilizados os recursos de autenticação e criptografia disponíveis pelo protocolo de comunicação do sensor. Essa tabela ainda conta com o campo habilitação, que informa se o dispositivo está desabilitado, se foi habilitado pela configuração do próprio *middleware* ou se foi habilitado por uma aplicação externa.

Os dados enviados pelos sensores são armazenados na tabela *dados\_coletados*, que registram também os horários de sincronização obtidos por meio do método descrito na Seção 4.1.3. Para configuração das aplicações que terão acesso ao *middleware*, utiliza-se a tabela *aplicacao*, que contém a identificação *REST* da aplicação, bem como informa se a aplicação utilizará recursos de segurança e autenticação disponíveis pelo protocolo *REST*. Como cada aplicação pode possuir vários usuários que desejam conectar-se ao *MOMICARE*, há uma tabela específica (*credenciais*) para registro dos dados de acesso de cada aplicação ao *middleware*. Essa tabela também registra se o usuário poderá alterar alguma configuração do *MOMICARE* (campo *adm\_middleware*) ou de algum sensor conectado (campo *adm\_sensores*).

Para implementação desse modelo de dados, recomenda-se o uso do banco de dados relacional *SQLite* para armazenamento dos dados coletados, bem como as informações de configuração informadas pelo módulo de gerenciamento. Opta-se pelo uso desse banco de dados pelo fato de ser um *software* livre, multiplataforma e implementado a nível de biblioteca de programação (SQLITE, 2016), consumindo assim menos recursos de *hardware* em relação aos sistemas gerenciadores de banco de dados (SGBD) tradicionais.

#### 4.1.3 Módulo de Comunicação

O módulo de comunicação é responsável por realizar a comunicação do *middleware* com os sensores corporais de uma rede *WBAN*, conforme as configurações de energia e segurança

definidas no módulo de gerenciamento. Dentro desse módulo há um componente *6LowPAN* responsável por realizar a conversão dos cabeçalhos enviados em *IPv6* para o cabeçalho reduzido utilizado pelo padrão *IEEE 802.15.4* e pelo padrão *BLE*, por meio das técnicas descritas na Seção 2.3, permitindo assim que qualquer dispositivo *IoT* possa ter um endereçamento válido no mundo *IP*. Dessa forma, não há a necessidade de um *gateway* que efetue a conversão dos padrões a nível de aplicação, conforme abordado na Seção 2.3, exigindo assim um menor processamento.

Esse módulo também é responsável por fornecer uma comunicação energeticamente mais eficiente entre o *smartphone* e os sensores de uma *WBAN*, reduzindo o ciclo de trabalho dos sensores. O ciclo de trabalho pode ser calculado por meio da seguinte fórmula (DEMENTYEV et al., 2013) (NAIR et al., 2015):

$$CicloDeTrabalho = TempoDeAtividade / TempoTotal \quad (4.1)$$

onde *TempoTotal* é a soma do tempo de atividade (*TempoDeAtividade*) e do tempo de inatividade de um sensor. Conforme expressado pela Fórmula 4.1, quanto menor o ciclo de trabalho, maior será o tempo de inatividade de uma comunicação. Como consequência, teoricamente, quanto menor o ciclo de trabalho, maior será a redução de consumo de energia.

O ciclo de trabalho pode ser otimizado de maneiras distintas no *BLE* e no *IEEE 802.15.4*. No *BLE* o ciclo de trabalho pode ser controlado configurando-se o tempo de anúncio de um dispositivo, conforme explanado na Seção 2.1.2. Já no *IEEE 802.15.4*, o ciclo de trabalho pode ser configurado parametrizando-se os campos *BO* e *SO*, que definem o tempo de atividade e inatividade de um superquadro, conforme descrito na Seção 2.2.

O *MOMICARE* permitirá a configuração do ciclo de trabalho. Todavia, por padrão, o *middleware* trabalhará com ciclos de trabalho inferiores a 1 segundo, de forma a não prejudicar o sincronismo temporal entre os dispositivos. Esse valor foi definido com base nos trabalhos de Elliott e Coventry (2012) e Benharref, Serhani e Mizouni (2014), que definem que a frequência de batimento cardíaco precisa ser colhida em frequências de 1 segundo, por um período mínimo de 30 segundos.

Por meio do uso do ciclo de trabalho, o *MOMICARE* fornece um recurso de sincronização entre os sensores de uma *WBAN*, por intermédio do modelo de ordem de eventos (DJENOURI; BAGAA, 2016). Esse modelo é uma variação do algoritmo de Berkeley explanado no Capítulo 2, porém sem efetuar a resincronização do dispositivo escravo. Nesse modelo, os dados recebidos por cada sensor são reordenados conforme seu intervalo de chegada, conforme metodologia abaixo:

- O *middleware* registra o horário de chegada dos dados de um determinado sensor;
- O *middleware* calcula o intervalo de transmissão de cada sensor, a partir do cálculo da diferença entre os dois últimos instantes de recebimento de dados;

**Tabela 3:** Exemplo de sincronização de tempo entre sensores

Sensor	Tempo decorrido	Horário de chegada dos dados	Horário do sensor	Horário consolidado	Dados recebidos
Sensor 1	0s	2016-10-26 09:00:00:059	2001-01-01 10:00:00:000	2016-10-26 09:00:01:059	x=19
Sensor 1	0s	2016-10-26 09:00:00:059	2001-01-01 10:00:03:000	2016-10-26 09:00:04:059	y=20
Sensor 2	2s	2016-10-26 09:00:02:074	2003-01-17 09:00:05:000	2016-10-26 09:00:03:599	a=32
Sensor 1	5s	2016-10-26 09:00:05:059	2001-01-01 10:00:07:000	2016-10-26 09:00:07:059	x=19
Sensor 1	5s	2016-10-26 09:00:05:059	2001-01-01 10:00:08:000	2016-10-26 09:00:08:059	y=22
Sensor 2	5s	2016-10-26 09:00:05:079	2003-01-17 09:00:09:000	2016-10-26 09:00:08:604	a=35
Sensor 2	8s	2016-10-26 09:00:08:084	2003-01-17 09:00:13:000	2016-10-26 09:00:13:609	a=34
Sensor 1	10s	2016-10-26 09:00:10:059	2001-01-01 10:00:11:000	2016-10-26 09:00:10:559	x=20
Sensor 1	10s	2016-10-26 09:00:10:059	2001-01-01 10:00:15:000	2016-10-26 09:00:14:559	y=22

Fonte: Elaborada pelo autor.

- O *middleware* analisa os horários recebidos na última transmissão e calcula o intervalo obtido entre o horário do primeiro e do último dado recebido do sensor nesse horário;
- O *middleware* calcula a média da diferença entre o intervalo de transmissão e o intervalo dos dados;
- O *middleware* registra o horário consolidado, somando o valor obtido da média ao horário de chegada dos dados.

Esse método permite que os dados recebidos de cada sensor possam ser relacionados marcando-se o horário de chegada desses dados no *middleware*. A Tabela 3 mostra um cenário de uma possível relação de sincronização entre dois sensores, onde o Sensor 1 envia dados ao *middleware* a cada 5 segundos, enquanto que o Sensor 2 envia dados a cada 3,05 segundos. Com o registro do horário de recebimento dos dados dos sensores pelo *middleware*, é possível correlacionar os dados entre os sensores.

O módulo de comunicação conta ainda com uma interface externa para o padrão *IEEE 802.15.4*, permitindo que qualquer dispositivo que utilize esse padrão de comunicação possa se comunicar ao *smartphone*, de forma que hardwares externos possam cumprir essa função, como os utilizados por Wang et al. (2015) e por Gonnot et al. (2015). Esse módulo, além de contar com a camada de comunicação *6LowPAN*, conta ainda com uma conexão serial por



meio de interface *USB* que permite que outros pequenos dispositivos possam ser conectados fisicamente ao *smartphone*, por meio do padrão *UART*. Essa conexão física se faz necessária pois os atuais *smartphones* não possuem o módulo *IEEE 802.15.4* nativos. Dessa forma, é necessária a utilização de um dispositivo externo que implemente essa funcionalidade. Assim, por intermédio de um pequeno cabo *USB*, pode-se fornecer a funcionalidade do *IEEE 802.15.4* ao *smartphone*.

A interligação entre os sensores de uma *WBAN* e o *smartphone* é feito por meio do protocolo *6LowPAN*, convertendo os cabeçalhos *IP* para os cabeçalhos suportados pelas camadas *L2CAP* do *Bluetooth* e pela subcamada *MAC* do *IEEE 802.15.4*, especificados na Seção 2.3.

Esse módulo também é responsável por aplicar o protocolo *SMP*, descrito nas Seções 2.1.4 e 2.2, para estabelecer uma comunicação segura entre o *smartphone* e os sensores, por meio de conexões *BLE* e *IEEE 802.15.4*, respectivamente, caso essa configuração seja marcada no módulo de gerenciamento do *MOMICARE*.

#### 4.1.4 Módulo de Integração

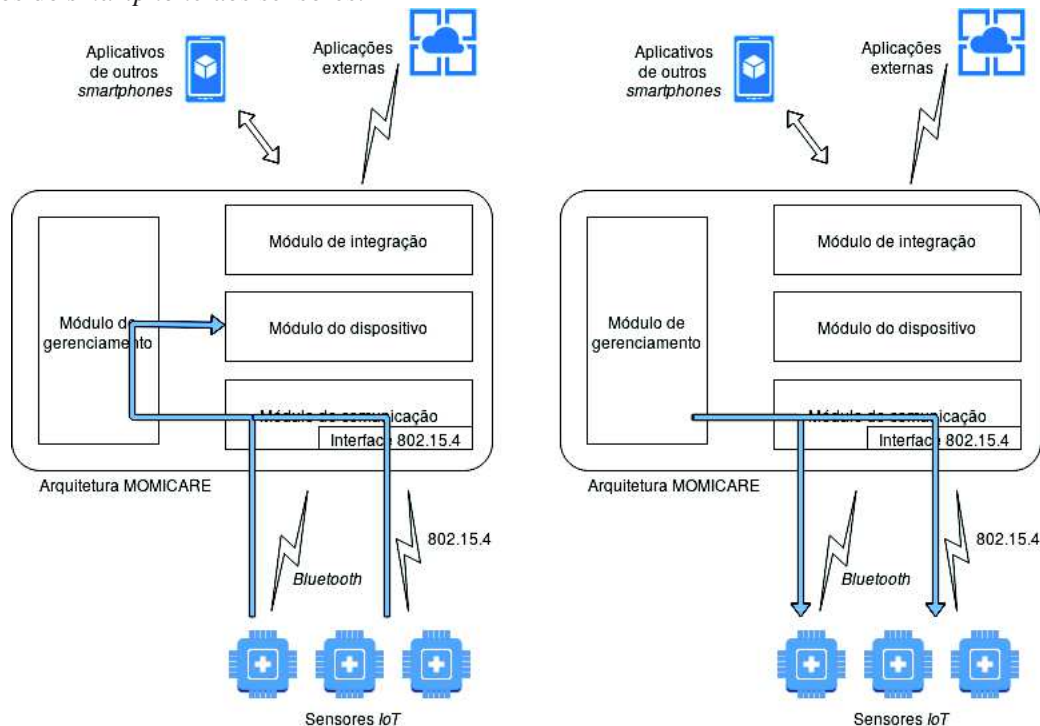
O módulo de integração tem a função de realizar a comunicação entre o *middleware* e os aplicativos de terceiros instalados no próprio *smartphone*, em *smartphones* externos ou em qualquer outro dispositivo, a partir das configurações definidas no módulo de gerenciamento. Caso a solicitação de comunicação seja enviada a uma aplicação terceira instalada no próprio *smartphone*, ela será efetuada por meio de chamada de sistema do próprio sistema operacional do dispositivo.

Caso contrário, ela se dará por qualquer outra aplicação que possa comunicar-se pelo padrão *Representational State Transfer (REST)*. Esse padrão utiliza os recursos existentes no protocolo *HTTP* para realizar comunicação entre aplicações remotas (CASTILLEJO et al., 2013), utilizando as *URIs* para identificação dos recursos oferecidos pelo *middleware* e utilizando os métodos como *GET*, *POST*, etc. para a troca de informações entre *smartphone* e aplicação externa. Ainda, o uso do *HTTPS* permite o uso da criptografia na comunicação entre *middleware* e aplicação.

## 4.2 Funcionamento geral do *MOMICARE*

O *MOMICARE* possui diferentes formas de funcionamento. A Figura 16 mostra o fluxo de comunicação entre o *smartphone* e os sensores. Os sensores comunicam-se com o *smartphone* por intermédio dos padrões de comunicação definidos pelo módulo de comunicação. O desenho à esquerda mostra o fluxo de comunicação dos sensores ao *smartphone*. Recebidos os dados dos sensores, o módulo de comunicação os envia para o módulo de gerenciamento, que analisa os dados recebidos e os marca como íntegros ou não e, por fim, armazena-os na memória do *smartphone* por meio do módulo do dispositivo para posterior uso das aplicações de terceiros

**Figura 16:** Fluxo de comunicação entre os sensores de uma *WBAN* e o *smartphone*. Na esquerda é mostrado o envio de dados dos sensores para o *smartphone*, enquanto que na direita é mostrado o envio de dados do *smartphone* aos sensores.



Fonte: Elaborado pelo autor

que desejam armazenar esses dados.

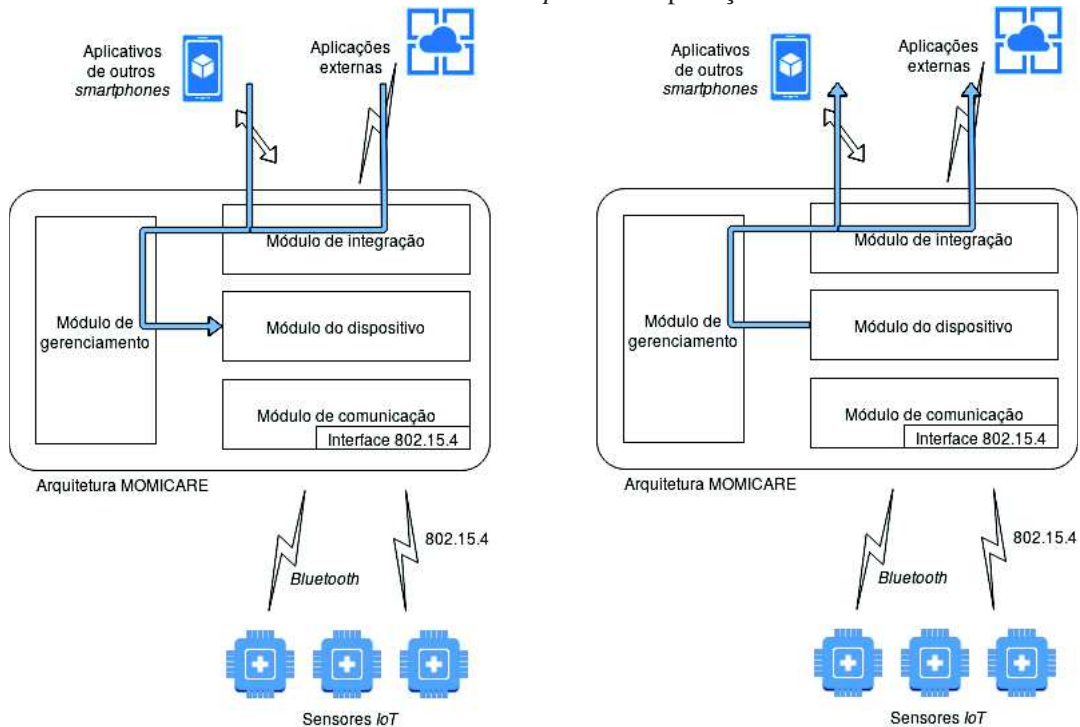
Já o desenho da direita mostra o fluxo de comunicação do *smartphone* aos sensores, quando há uma requisição de alteração da parametrização do sensor, por exemplo. Nesse desenho, nota-se que o processo de comunicação inicia-se pelo módulo de gerenciamento, que detém as configurações dos sensores, passa pelo módulo de comunicação, e chega ao(s) sensor(es) desejado(s) por meio dos padrões de comunicação suportados pelo *middleware*.

As aplicações de terceiros solicitam esses dados por meio do módulo de integração, que repassa as requisições para o módulo de gerenciamento, que envia a requisição para o módulo do dispositivo, conforme mostrado na Figura 17 à esquerda. Em seguida, o módulo do dispositivo verifica se os dados requisitados existem e, em caso positivo, retorna os dados solicitados ao módulo de gerenciamento, que faz uma verificação de consistência e, caso esteja tudo certo, repassa esses dados ao módulo de aplicação que então os retorna a aplicação. Esse processo é mostrado na Figura 17 à direita.

Uma aplicação pode ainda solicitar uma configuração dos sensores do dispositivo ou que compõem a rede *WBAN*, ou uma configuração do *middleware*. Essa requisição é feita a partir do módulo de integração, que repassa a requisição para o módulo de gerenciamento. Esse módulo verifica as permissões do usuário vinculado à aplicação e, caso o usuário tenha a permissão, a configuração é repassada aos sensores pelo módulo de comunicação ou aos sensores internos pelo módulo do dispositivo. Em relação aos sensores, ainda, um usuário não poderá desativar



**Figura 17:** Fluxo de comunicação entre as aplicações (externas ou do próprio *smartphone* e o *smartphone*. Na esquerda é mostrado o envio de dados das aplicações para o *smartphone*, enquanto que na direita é mostrado o envio de dados do *smartphone* às aplicações.



Fonte: Elaborado pelo autor

um sensor que tenha sido ativado por outra aplicação. Essa desativação só poderá ser feita pela configuração do *middleware*. Dessa forma, há um controle para que a configuração de uma aplicação não interfira em parametrizações de sensores efetuadas anteriormente por outras aplicações.

### 4.3 Considerações parciais

O modelo *MOMICARE*, proposto neste capítulo, gerencia uma rede de sensores sem fio corporais, chamadas de *WBAN*. Essas redes apresentam desafios relacionados à segurança na comunicação entre sensores, ao sincronismo e interoperabilidade entre esses dispositivos, bem como em relação à eficiência energética na comunicação entre aplicações médicas e os sensores.

O modelo é composto pelos módulos de gerenciamento, de integração, de comunicação e do dispositivo. O módulo de gerenciamento administra o funcionamento dos demais módulos e permite a parametrização do *middleware*. O módulo de integração realiza a integração entre o *middleware* e às aplicações de terceiros. O módulo de comunicação realiza a comunicação do *middleware* com os dispositivos que formam uma rede *WBAN*, conforme as configurações fornecidas pelo módulo de gerenciamento. Esse módulo também é responsável por realizar a sincronização de tempo entre os dados recebidos pelos sensores que compõem essa rede. Finalmente, o módulo do dispositivo é responsável por administrar os recursos do *smartphone*

que terá o *middleware* instalado.

## 5 METODOLOGIA DE AVALIAÇÃO

Para a avaliação do modelo proposto no Capítulo 4 foi criado um protótipo e um ambiente de simulação que complementa a análise do protótipo. O protótipo é executado em um *smartphone* com o Sistema Operacional *Android*<sup>1</sup>, que controla uma rede de sensores sem fio que comunicam-se por intermédio do protocolo *BLE*. Já o ambiente de simulação é executado a partir do *software Cooja*<sup>2</sup>, que simula o comportamento de dispositivos que comunicam-se por meio do uso do protocolo *IEEE 802.15.4*. O uso do protótipo e do simulador deve-se ao fato de que tanto o *smartphone* quanto o SO *Android* não suportarem nativamente o protocolo *IEEE 802.15.4* (HU et al., 2013).

### 5.1 Protótipo

O protótipo, desenvolvido a partir do código-fonte *BluetoothLeGatt Sample*<sup>3</sup>, implementa os módulos de dispositivo, de gerenciamento e de comunicação do modelo *MOMICARE*, conforme demonstrado na Figura 18.

O módulo de comunicação gera um arquivo no padrão XML que contém comandos de requisições de leitura. Esse arquivo é importado para a aplicação *nRF Connect*<sup>4</sup>. Um trecho do código gerado é mostrado na Figura 19. O aplicativo, por sua vez, lê o arquivo XML, executa os comandos solicitados diretamente sobre os dispositivos *BLE* e grava os resultados na própria aplicação.

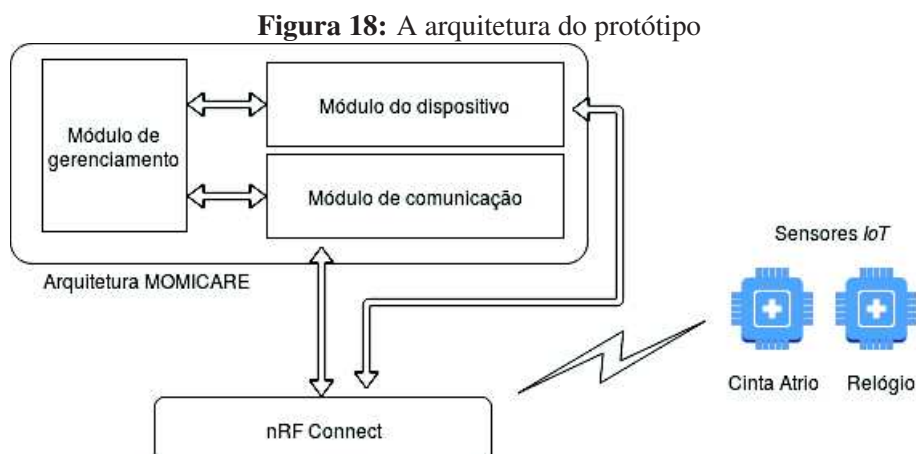
Como o *nRF Connect* permite a exportação dos dados por meio do padrão CSV, o módulo do dispositivo do protótipo *MOMICARE* foi adaptado para importar diretamente esses dados e

<sup>1</sup><https://www.android.com>

<sup>2</sup><http://www.contiki-os.org>

<sup>3</sup><https://github.com/googlesamples/android-BluetoothLeGatt>

<sup>4</sup><https://www.nordicsemi.com/eng/Products/Nordic-mobile-Apps/nRF-Connect-for-mobile-previously-called-nRF-Master-Control-Panel>



**Figura 19:** Um trecho do formato XML gerado pelo protótipo e que é executado pelo *nRF Connect*

```

<macro name="Coração" icon="PLAY">
  <assert-service description="Ensure Heart Rate" uuid="0000180d-0000-1000-8000-00805f9b34fb">
    <assert-characteristic description="Ensure Heart Rate Measurement" uuid="00002a37-0000-1000-8000-00805f9b34fb">
      <property name="NOTIFY" requirement="MANDATORY"/>
    </assert-characteristic>
  </assert-service>
  <wait-for-notification description="Wait for notification to Heart Rate Measurement" characteristic-uuid="00002a37-0000-1000-8000-00805f9b34fb" service-uuid="0000180d-0000-1000-8000-00805f9b34fb">
    <assert-value description="Assert value equals 0x16470A03" value="16470A03"/>
  </wait-for-notification>
  <wait-for-notification description="Wait for notification to Heart Rate Measurement" characteristic-uuid="00002a37-0000-1000-8000-00805f9b34fb" service-uuid="0000180d-0000-1000-8000-00805f9b34fb">
    <assert-value description="Assert value equals 0x16470F03" value="16470F03"/>
  </wait-for-notification>
  <wait-for-notification description="Wait for notification to Heart Rate Measurement" characteristic-uuid="00002a37-0000-1000-8000-00805f9b34fb" service-uuid="0000180d-0000-1000-8000-00805f9b34fb">
    <assert-value description="Assert value equals 0x16472B03" value="16472B03"/>
  </wait-for-notification>
  <wait-for-notification description="Wait for notification to Heart Rate Measurement" characteristic-uuid="00002a37-0000-1000-8000-00805f9b34fb" service-uuid="0000180d-0000-1000-8000-00805f9b34fb">
    <assert-value description="Assert value equals 0x16480703EF02" value="16480703EF02"/>
  </wait-for-notification>

```

Fonte: Elaborado pelo autor

armazená-los em sua base de dados interna implementada por meio do SGBD *SQLite*. Optou-se pelo uso dessa comunicação indireta visto que o padrão *BLE* possui inúmeros perfis, serviços e características de comunicação oferecidas pelo protocolo *GAP*, explanado na Seção 2.1.4, e o *software nRF Connect* já possui esses perfis incorporados à sua base de dados. O módulo do dispositivo, ainda, contém códigos que executam requisições de leituras de dados sobre os dispositivos *BLE* que formam a rede *WBAN*, simulando assim as requisições de diversas aplicações de saúde.

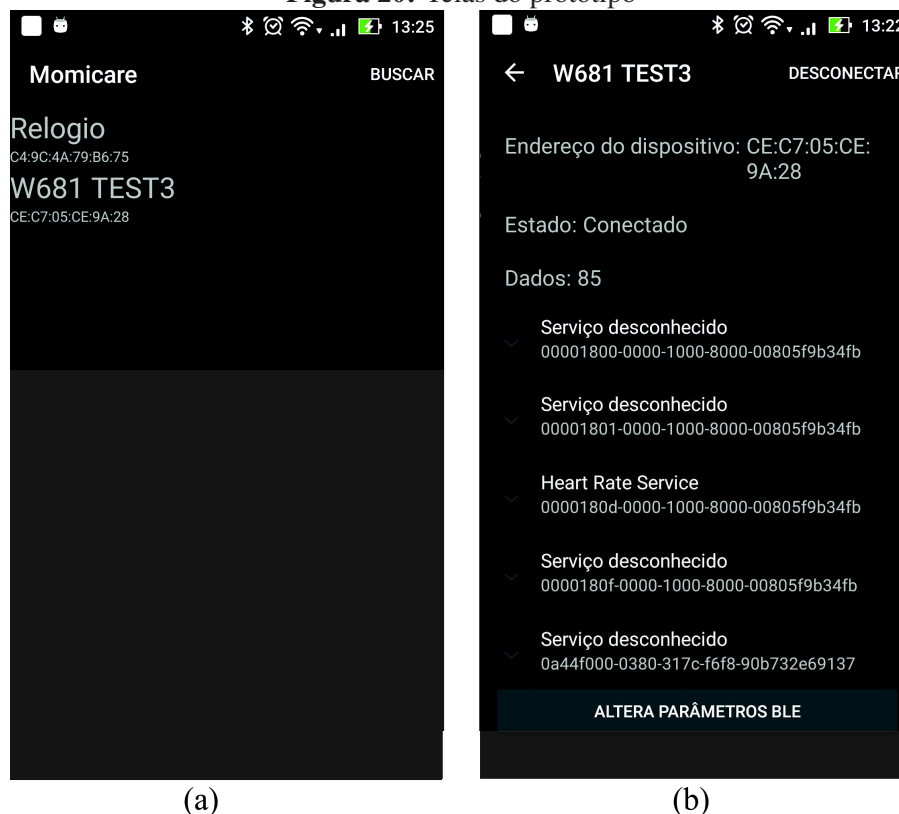
Por fim, o módulo de gerenciamento permite a visualização dos dispositivos *BLE* próximos ao *smartphone*, conforme demonstrado na Figura 20 (a). Esse módulo também permite a visualização dos serviços oferecidos e a parametrização do tamanho do *payload* do quadro *BLE*, bem como o intervalo de comunicação (ciclo de trabalho) entre o *smartphone* e os dispositivos, conforme demonstrado na Figura 20 (b). No que diz respeito a esse intervalo, também cabe ressaltar que o *Android* permite somente três níveis de parametrização. O primeiro nível trabalha com intervalos de 11,2 a 15ms, o segundo nível trabalha com intervalos de 30 a 50 ms, e o último nível trabalha com intervalos de 100 a 125 ms. Para fins de avaliação, o protótipo teve seu ciclo de trabalho definido no seu último nível, de forma a manter o maior tempo de inatividade possível, conforme definido nas premissas do modelo, descritas na Seção 4.1.3.

O protótipo é executado no *smartphone Asus Zenfone 2*<sup>5</sup>, que executa o *Android Marshmallow* 6.0.1 e que possui como especificações de *hardware* dois processadores (*Intel Atom Quad Core Z3580* e *Intel Atom Quad Core Z3560*), 3GB de memória *RAM* e 32GB de armazenamento interno. Para execução dos experimentos que compõem a avaliação, foram efetuadas as seguintes parametrizações no *smartphone*:

- O aparelho teve sua configuração restaurada para os padrões de fábrica, contendo apenas os aplicativos necessários para execução dos experimentos que compõem a avaliação do modelo. Isso evitava que outras aplicações pudessem comprometer os resultados dos experimentos realizados, especialmente em relação à troca de dados e ao consumo de bateria do dispositivo;

<sup>5</sup>[https://www.asus.com/br/Phone/ZenFone\\_2\\_ZE551ML](https://www.asus.com/br/Phone/ZenFone_2_ZE551ML)

Figura 20: Telas do protótipo



Fonte: Elaborado pelo autor

- Todas as redes de dados do *smartphone* - 3G/4G, *Wi-Fi* e *BLE* estavam habilitados, enquanto que o *GPS* estava desabilitado;
- O brilho da tela do dispositivo estava configurado em seu valor mínimo, para evitar consumo excessivo de bateria;
- Antes da execução de cada experimento, o *smartphone* tinha sua bateria totalmente recarregada.

Já a rede de sensores sem fio é formada por uma cinta cardíaca modelo Átrio ES055<sup>6</sup> e por uma pulseira modelo *Smart Band*<sup>7</sup>, conforme mostra a Figura 21. Esses dois dispositivos também são visualizados na Figura 20 (a) com os nomes "W681 TEST3" e "Relógio", respectivamente. A cinta cardíaca possui um medidor de frequência cardíaca e não possui memória interna. Já a pulseira possui um medidor de frequência cardíaca, um monitor de pressão arterial e um pedômetro, bem como memória interna de 64KB. Tanto a medição cardíaca, bem como a pressão arterial, são verificadas por meio do uso de uma pequena câmera acoplada à pulseira. Esse dispositivo também possui um visor de informações e possui um sistema de vibração que pode ser configurado.

<sup>6</sup><http://www.multilaser.com.br/cinta-cardiaca-bluetooth-atrrio-es055/p>

<sup>7</sup>[https://pt.aliexpress.com/store/product/New-Smart-band-A06/1394406\\_32711754112.html](https://pt.aliexpress.com/store/product/New-Smart-band-A06/1394406_32711754112.html)

### 5.1.1 Metodologia de Avaliação Energética

A avaliação do consumo de energia consiste basicamente em verificar o desempenho energético do protótipo em relação à variação do tamanho do *payload* - ou *Maximum Transmission Unit (MTU)* - implementado pelo protocolo *BLE*, comparado ao uso de aplicações utilizadas para leitura dos dispositivos que utilizam essa tecnologia de comunicação. Essa metodologia baseia-se numa adaptação dos modelos desenvolvidos por Siekkinen et al. (2012) e Karthi, Rao e Pillai (2015), onde quadros com *MTUs* variados são enviados utilizando os ciclos de trabalho padrão dos protocolos e o menor ciclo de trabalho possíveis nos protocolos, durante um intervalo de 24 horas, ou 86400 segundos. No protótipo, a avaliação foi efetuada considerando o consumo da bateria informada pelo próprio dispositivo *BLE*. Além dessa avaliação, foi implementada uma avaliação de consumo energética considerando o uso de aplicações simultâneas executadas sobre o protótipo.

Para a avaliação do protótipo foram utilizados *MTUs* com tamanhos de 30, 60, 120 e em seu valor máximo, correspondente a 255 *bytes*, bem como também foi analisado o impacto energético utilizando 1, 2 e 4 instâncias de um código desenvolvido dentro do módulo do dispositivo do protótipo. Esse código consiste em fazer a leitura de alguns serviços oferecidos por cada um dos 2 dispositivos *BLE* que formam a rede *WBAN*.

No que diz respeito à cinta cardíaca, o código realiza a leitura do nome do dispositivo, da carga da bateria e do batimento cardíaco. Essas são as leituras efetuadas pelo aplicativo *Elite HRV*<sup>8</sup>, utilizado como comparativo nessa análise. No que tange à pulseira, o código realiza a leitura do nome do dispositivo e do serviço que informava a quantidade de passos efetuados, o batimento cardíaco, a pressão arterial, a carga da bateria, entre outros. Esse serviço não é um padrão de perfil do *BLE*, mas um serviço customizado desenvolvido pelo próprio fabricante e que é lido pelo aplicativo *HBand 2.0*<sup>9</sup>, utilizado como comparativo nessa análise.

Além disso, para todas as simulações realizadas sobre o protótipo, foi considerado o maior ciclo de trabalho que o protótipo pode implementar, conforme descrito na Seção 5.1. Por fim, todas as avaliações energéticas realizadas sobre esse protótipo consideraram um tempo de troca de dados de 86400 segundos, ou 24 horas. Esse tempo de avaliação foi assim definido devido ao fato de que a cinta cardíaca consome muito pouca energia e porque ambos dispositivos trazem em seus resultados de leitura da carga da bateria valores inteiros. Ainda, para cada bateria de testes, a bateria da cinta cardíaca e da pulseira eram, respectivamente, trocada e recarregada. Por fim, com o objetivo de facilitar a análise, foram armazenados os registros dos dispositivos a cada 30 segundos.

---

<sup>8</sup><https://elitehrv.com/>

<sup>9</sup><https://play.google.com/store/apps/details?id=com.timaimiee.hband>



**Figura 21:** A cinta cardíaca (à esquerda) e a pulseira (à direita) utilizadas para avaliação do protótipo



Fonte: Elaborado pelo autor

### 5.1.2 Metodologia de Avaliação de Sincronismo

A avaliação do sincronismo tem por objetivo avaliar o método de sincronização por ordem de eventos proposto no Capítulo 4. Nessa análise, foi verificada a precisão dos tempos registrados pelo protótipo em relação aos tempos registrados pelo dispositivo. Essa verificação baseou-se no modelo de Djenouri e Bagaa (2016), que utiliza um relógio externo para verificação da precisão. Esse relógio externo está sincronizado com o protótipo.

Como a cinta cardíaca não possui em seu perfil *GAP/GATT* o serviço de verificação e ajuste de horário, pressupõe-se que a definição de sincronização nestes dispositivos ocorre somente pelo registro do horário dos quadros de dados recebidos pelo protótipo, esse passa a ser definido arbitrariamente como o horário de registro no protótipo, bem como não há a possibilidade de ser analisada variações temporais em relação ao sincronismo. Todavia, em relação à pulseira *BLE*, essa avaliação pode ser efetuada conjuntamente com o protótipo.

Para realização dessa avaliação, desconsiderou-se o cálculo do primeiro recebimento dos dados, visto que neste caso o protótipo não possuía dados anteriores para a realização do cálculo da média do intervalo de recebimento. O protótipo, ainda, foi configurado para utilizar o maior ciclo de trabalho disponível no *Android*, conforme especificado na Seção 5.1. Ainda, para a execução dessa avaliação, a bateria da pulseira foi totalmente recarregada, de maneira análoga ao cenário descrito na Seção 5.1.1.

### 5.1.3 Metodologia de Avaliação de Carga

A avaliação de carga consiste em avaliar o funcionamento do *middleware* considerando o uso de múltiplas aplicações externas requisitando dados ao *smartphone*. Nessa análise, foi verificado se o protótipo não apresenta problemas de comunicação com os sensores, de gravação

dos dados recebidos e de envio desses dados às aplicações externas, considerando o uso simultâneo de 1, 2, 4 e 8 aplicações externas. Para análise desses dados, foi utilizado o software *CPU Monitor*<sup>10</sup>, que faz o monitoramento do consumo dos processos *Android*.

O protótipo também teve sua carga comparada em relação aos aplicativos *EliteHRV* e *HBand*, utilizados na avaliação energética descrita na Seção 5.1.1. Para a execução dessa avaliação, ainda, o protótipo foi parametrizado com *MTU* de 255 *Bytes*, bem como teve definido seu ciclo de trabalho no seu valor máximo permitido pelo *Android*. Nesta análise foram executadas duas baterias de testes: uma bateria considerando o uso da pulseira, utilizando o aplicativo *HBand* como parâmetro de comparação; e outra bateria utilizando a cinta cardíaca, utilizando o aplicativo *Elite HRV* como parâmetro para análise.

Ainda, para a execução dessa avaliação, o *smartphone* estava calibrado conforme descrito na Seção 5.1.1, enquanto que a bateria da cinta cardíaca e da pulseira eram, respectivamente, trocada e recarregada, conforme já descrito na Seção 5.1.1

#### 5.1.4 Limitações do Protótipo

O protótipo desenvolvido possui algumas limitações. Primeiramente, não foi implementando o módulo de integração, visto que a implementação desse recurso é onerosa, sendo que as requisições externas podem ser simuladas pelo próprio protótipo, sem comprometer os resultados das avaliações propostas nas Seções 5.1.1 a 5.1.3. Em função disso, o módulo de gerenciamento também não implementa os recursos de configuração referentes às aplicações externas.

Além disso, outra restrição existente está relacionada ao número limitado de sensores *BLE* utilizados, que não permite avaliações do protótipo contendo um maior número de dispositivos. Outra limitação existente é a impossibilidade de comunicação com dispositivos *IEEE 802.15.4*, visto que o *hardware* do *smartphone* não possui esse módulo de comunicação. Além disso, o próprio *SO Android* não suporta nativamente o protocolo. Essa limitação, todavia, é em parte resolvida por meio do desenvolvimento do ambiente de simulação, descrito na Subseção 5.2.

## 5.2 Ambiente de Simulação

O ambiente de simulação é executado a partir do *software Cooja*, disponível no sistema operacional *Contiki*, voltado para aplicações *IoT* (CONTIKI, 2017) que utilizam o protocolo *IEEE 802.15.4*. O *Cooja* destina-se a simular redes de sensores sem fio em diversos níveis de avaliação, possibilitando assim a realização de avaliações mais detalhadas e precisas em relação ao comportamento da rede.

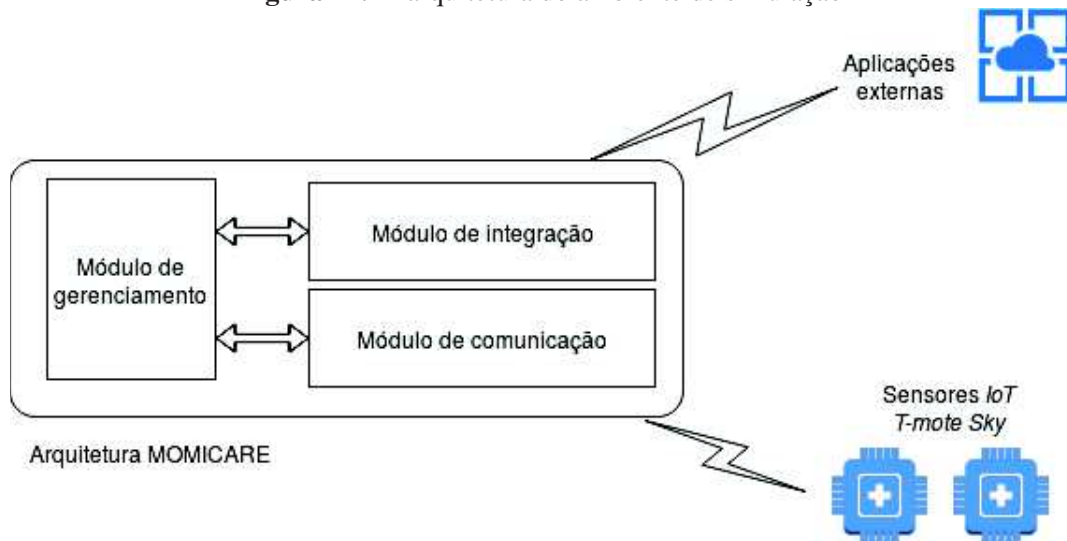
O ambiente utilizou diversos dispositivos *Tmote-sky*<sup>11</sup>, disponíveis no próprio *Cooja*. Num

<sup>10</sup><https://play.google.com/store/apps/details?id=com.bigbro.ProcessProfiler>

<sup>11</sup><http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf>



**Figura 22:** A arquitetura do ambiente de simulação



Fonte: Elaborado pelo autor

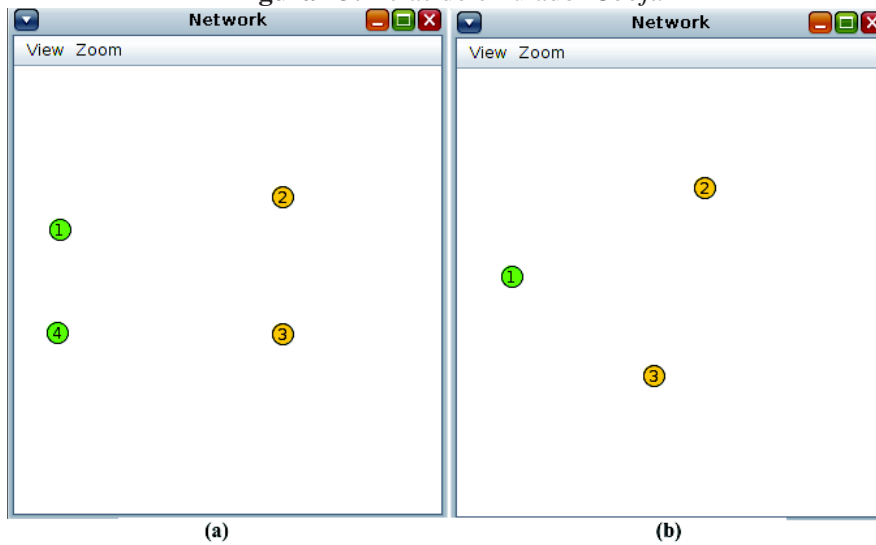
desse dispositivo, chamado de dispositivo-pai, foi implementado os módulos do dispositivo, de gerenciamento e de comunicação, conforme Figura 22. O módulo do dispositivo permite a parametrização da comunicação com os demais dispositivos *IEEE* 802.15.4 a partir da própria interface gráfica do *software*. O módulo de comunicação realiza a comunicação com os demais sensores, conforme os parâmetros executados. O módulo de integração recebe as requisições das aplicações externas, executadas a partir de um *notebook*, que executam comandos *ping* com tamanho de 32 *bytes*, simulando as requisições de leituras de dados sobre os dispositivos *IEEE* 802.15.4 que formam a rede *WBAN*.

Neste ambiente não foi implementado o módulo de integração, visto que o sensor *Tmote-sky* não possui capacidade para armazenamento dos dados capturados. Dessa forma, a análise dos dados enviados pelos demais sensores foi feita a partir dos *logs* gerados pelo próprio emulador. A rede de sensores sem fio também foi formada por sensores *Tmote-sky*. Devido ao fato de tratar-se de um simulador, é possível que sejam configurados inúmeros sensores para composição de uma rede. Portanto, nesse ambiente, foram criadas redes *WBAN* compostas por 2 dispositivos - similar à rede descrita na Seção 5.1 - e por 4 dispositivos. A Figura 23 (a) mostra a configuração de uma rede *WBAN* formada por dois dispositivos (em amarelo), comunicando-se com duas aplicações distintas (em verde). Já a Figura 23 (b) representa os mesmos dois dispositivos (em amarelo) comunicando-se com o sensor que teve os módulos configurados (em verde).

### 5.2.1 Metodologia de Avaliação Energética

A avaliação do ambiente de simulação foi efetuada a partir das metodologias descritas na Seção 5.1.1, com variações em relação ao tamanho do *MTU* e em relação à quantidade de

Figura 23: Telas do emulador *Cooja*



Fonte: Elaborado pelo autor

dispositivos que compõem a rede *WBAN*. Em relação ao *MTU*, o ambiente foi configurado para a utilização de *MTUs* com tamanhos de 30, 60 e em seu valor máximo, correspondente a 127 *bytes*, bem como avaliado o impacto energético utilizando 1, 2 e 4 instâncias de um código desenvolvido dentro do módulo do dispositivo do protótipo. Esse código consiste em executar comandos *ping* que simulam requisições de aplicações remotas.

Nesta avaliação foram utilizadas duas configurações de rede *WBAN*: uma contendo 2 dispositivos *IEEE 802.15.4* e outra contendo 4 dispositivos. Foi também efetuada uma avaliação preliminar simulando o cenário de consumo sem a configuração de um dos sensores com os módulos descritos na Seção 5.2, considerando que cada um dos sensores que formam a *WBAN* comunica-se diretamente com outro sensor que representa a aplicação. Ainda, para todas as simulações realizadas sobre o protótipo, foi considerado o ciclo de trabalho padrão do dispositivo (MOTEIV, 2006).

Como o *Cooja* não possui um módulo para verificação do consumo de energia dos sensores, foi desenvolvida uma aplicação dentro do módulo de comunicação que calcula o consumo energético da interface de rádio de cada um dos sensores-filho, a partir da seguinte fórmula, efetuada a partir das especificações de *hardware* do dispositivo (MOTEIV, 2006):

$$\text{ConsumoEnergetico} = (TX + RX) * 20 * 3 \quad (5.1)$$

onde *TX* representa a porcentagem de tempo que o sensor esteve transmitindo dados; *RX* representa a porcentagem de tempo que o sensor esteve recebendo dados; a constante 20 representa a corrente, em *miliamperes*, que a interface de comunicação do sensor utiliza ao transmitir ou receber dados; e a constante 3 representa a tensão, em *volts*, utilizada pelo sensor. O consumo energético é dado em *miliwatts*.

Por fim, todas as avaliações energéticas realizadas sobre esse ambiente consideraram um

tempo de comunicação de 86400 segundos, ou 24 horas, de forma a ter um indicador de tempo de execução semelhante ao protótipo.

### 5.2.2 Limitações do Ambiente de Simulação

Da mesma forma que ocorreu no protótipo, o ambiente de simulação também possui algumas limitações. Enquanto que no protótipo o consumo energético é avaliado a partir do consumo da bateria, demonstrando o consumo do dispositivo como um todo, no caso do ambiente de simulação consumo energético é avaliado somente sob a perspectiva da transmissão de dados por meio do protocolo *IEEE 802.15.4*, desconsiderando o consumo energético do processador e demais componentes que constituem o *hardware* do dispositivo *IEEE 802.15.4*. Outra limitação deste protótipo é que utilizam-se dispositivos genéricos que não possuem características específicas para o controle de sinais vitais corporais, como no caso do protótipo.

## 5.3 Considerações parciais

Para uma melhor avaliação do modelo *MOMICARE*, foram desenvolvidos um protótipo e configurado um ambiente de simulação, de forma a ampliar as análises sobre o referido modelo, bem como permitir também que a questão de pesquisa elaborada na Seção 1.1 possa ser respondida. O protótipo é executado em um *smartphone* com o Sistema Operacional *Android*, que controla uma rede *WBAN*, formada por uma cinta cardíaca e uma pulseira, onde a comunicação é realizada por intermédio do protocolo *BLE*. Para este protótipo, estão descritas as metodologias referentes à avaliação de consumo energético, à avaliação de sincronismo e à avaliação de carga.

Já o ambiente de simulação é executado a partir do *software Cooja*, que simula o comportamento de sensores que comunicam-se por meio do uso do protocolo *IEEE 802.15.4*. Neste ambiente, está descrita a metodologia de avaliação energética, que é complementar à avaliação energética prevista no protótipo.



## 6 RESULTADOS E DISCUSSÃO

Este capítulo descreve os resultados obtidos a partir das metodologias de avaliação descritas no Capítulo 5. São descritos e discutidos três tipos de resultados: de consumo energético, de sincronismo e de carga, cujos resultados esperados são mostrados na Tabela 4. Em relação à avaliação energética, espera-se que o consumo de energia dos dispositivos que formam a rede *WBAN* seja reduzido conforme o *MTU* do protótipo e do ambiente de simulação são aumentados. Espera-se também que haja um maior consumo de energia desses dispositivos conforme o número de instâncias concorrentes - que simulam as aplicações - é aumentado. Em relação à avaliação de sincronismo, espera-se que os dados enviados pelos dispositivos possam oferecer uma sincronização de tempo que seja tolerável para aplicações médicas. Em relação à avaliação de carga, espera-se um incremento no tempo de uso do processador e um incremento no uso da memória conforme a quantidade de instâncias concorrentes aumenta.

### 6.1 Consumo Energético

Os resultados referentes ao consumo energético são divididos em duas partes: uma referente aos resultados do protótipo, que realiza a comunicação entre dispositivos *BLE*; e outra referente aos resultados do ambiente de simulação, que realiza a comunicação entre dispositivos *IEEE 802.15.4*.

#### 6.1.1 Protótipo

Conforme descrito na Seção 5.1.1, a avaliação do consumo de energia foi feita considerando a variação do tamanho do *MTU* implementado pelo protocolo *BLE* e também considerando o uso de aplicações simultâneas executadas sobre os protótipos.

No que diz respeito à variação do *MTU* da cinta cardíaca, os resultados obtidos dessa análise são mostrados na Figura 24. Verificou-se que, quando utilizado o aplicativo *Elite HRV*, o consumo da bateria atingiu 1% na marca dos 32880 segundos, ou 9 horas e 8 minutos. Já a marca de 2% de consumo foi atingida na marca dos 65880 segundos, ou 18 horas e 18 minutos. Quando o protótipo foi configurado com o valor *MTU* de 30 *bytes*, o consumo da bateria do dispositivo atingiu 1% na marca dos 34140 segundos, ou 8 horas e 53 minutos. Já a marca de 2% de consumo foi atingida na marca dos 63930 segundos, ou 17 horas e 45 minutos. Quando o *MTU* do protótipo foi configurado para 60 *bytes*, o consumo da bateria do dispositivo atingiu 1% na marca dos 34140 segundos, ou 9 horas e 29 minutos. Já a marca de 2% de consumo foi atingida na marca dos 68250 segundos, ou 18 horas e 57 minutos. Quando o *MTU* do protótipo foi configurado para 120 *bytes*, houve uma pequena diminuição do consumo da bateria, atingindo 1% na marca dos 34200 segundos, ou 9 horas e 30 minutos. Já a marca de 2% de consumo foi atingida na marca dos 68340 segundos, ou 18 horas e 59 minutos. Quando o *MTU*

**Tabela 4:** Resumo dos resultados esperados das avaliações.

<b>Tipo de avaliação</b>	<b>Metodologia</b>	<b>Resultado esperado</b>
Avaliação energética	Variação do <i>MTU</i> do quadro <i>BLE</i> e <i>IEEE 802.15.4</i> . Execução de instâncias concorrentes nos protótipos.	Menor consumo de energia dos dispositivos conforme o <i>MTU</i> do protótipo é aumentado. Maior consumo de energia dos dispositivos quando executada várias instâncias concorrentes no protótipo.
Avaliação de sincronismo	Sincronização por ordem de eventos	Sincronização de tempo compatível com o exigido para aplicações médicas em relação aos dados enviados pelos dispositivos <i>WBAN</i> .
Avaliação de carga	Quantidade de aplicações externas solicitando dados ao <i>middleware</i>	Maior tempo de processamento e maior uso de memória do <i>smartphone</i> , sem comprometimento do funcionamento do <i>middleware</i> .

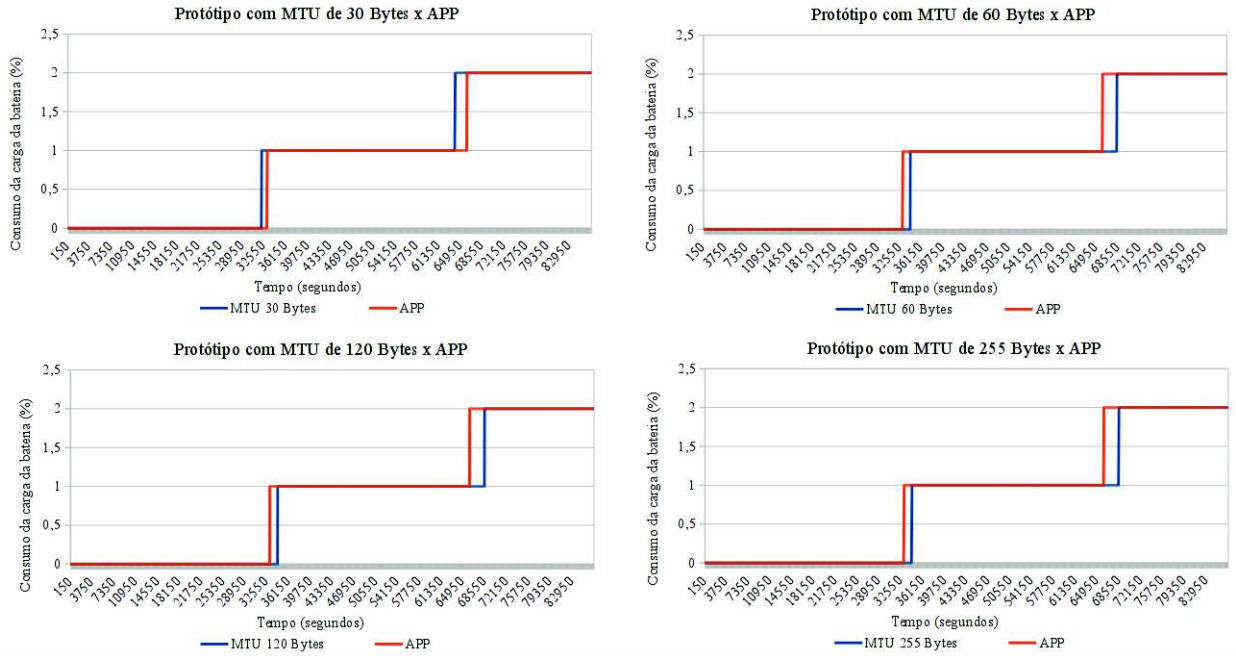
Fonte: Elaborada pelo autor.

do protótipo foi configurado para 255 bytes, o consumo da bateria do dispositivo manteve-se praticamente estável, atingindo 1% na marca dos 34200 segundos, ou 9 horas e 30 minutos. Já a marca de 2% de consumo foi atingida na marca dos 68370 segundos, ou 18 horas e 59 minutos.

Nessa mesma análise, a Figura 25 mostra a variação do consumo energético da cinta considerando as diferentes parametrizações de *MTU*. Nesse cenário, a variação do tamanho do *MTU* demonstrou que a configuração do *MTU* para 255 bytes causou uma redução de aproximadamente 6,5% no consumo de energia, quando comparado à configuração de *MTU* de 30 bytes. Verificou-se também que, quando o protótipo foi configurado com o valor *MTU* de 30 bytes, houve um aumento de aproximadamente 3% em relação à aplicação (*Elite HRV*) utilizada para leitura desse dispositivo. Quando o *MTU* do protótipo foi configurado para 60 bytes, houve uma redução no consumo de aproximadamente 3,5% em relação à aplicação. Quando o protótipo teve seu *MTU* configurado para uso de 120 e 255 bytes, a redução no consumo energético aumentou para 3,6%.

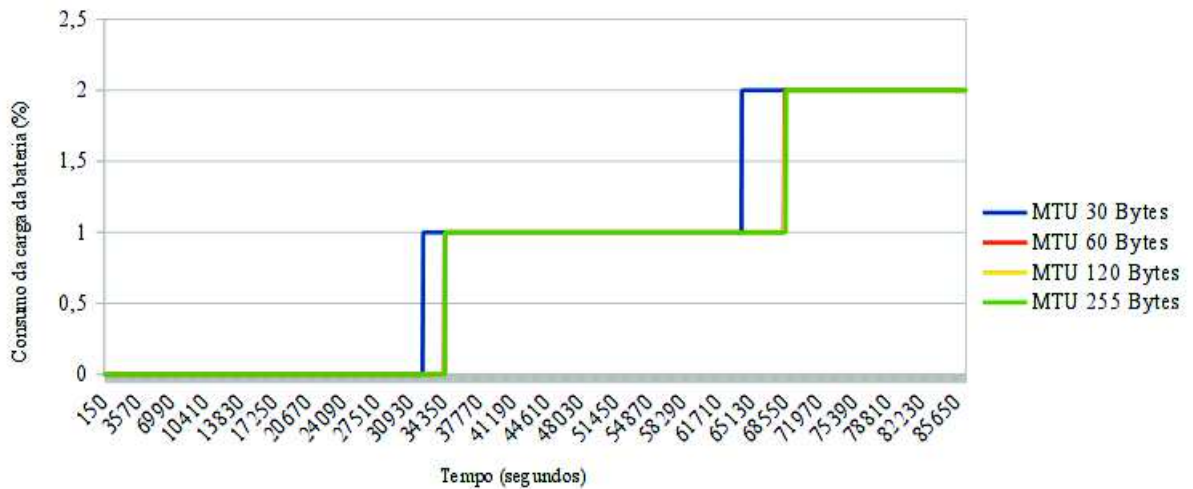
No que diz respeito à variação do *MTU* da pulseira, os resultados obtidos dessa análise são mostrados na Figura 26. Verificou-se que, quando utilizado o aplicativo *HBand*, houve um decréscimo de 17% na carga da bateria, sendo que o dispositivo perdia 1% de sua carga a cada 5111 segundos em média, ou cerca de 1 hora e 18 minutos. Quando o protótipo foi configurado com o valor *MTU* de 30 bytes, houve um decréscimo de 17% na carga da bateria, sendo que o dispositivo perdia 1% de sua carga a cada 4812 segundos em média, ou cerca de 1 hora e 20 minutos. Quando o *MTU* do protótipo foi configurado para 60 bytes, houve um decréscimo de 16% na carga da bateria, sendo que o dispositivo perdia 1% de sua carga a cada 5235 segundos em média, ou cerca de 1 hora e 27 minutos. Quando o *MTU* do protótipo foi configurado para 120 bytes, houve um decréscimo de 16% na carga da bateria, sendo que o dispositivo perdia 1% de sua carga a cada 5391 segundos em média, ou cerca de 1 hora e 29 minutos. Quando o *MTU* do protótipo foi configurado para 255 bytes, houve um decréscimo de 15% na carga da bateria,

**Figura 24:** Os resultados da variação do MTU referente à cinta



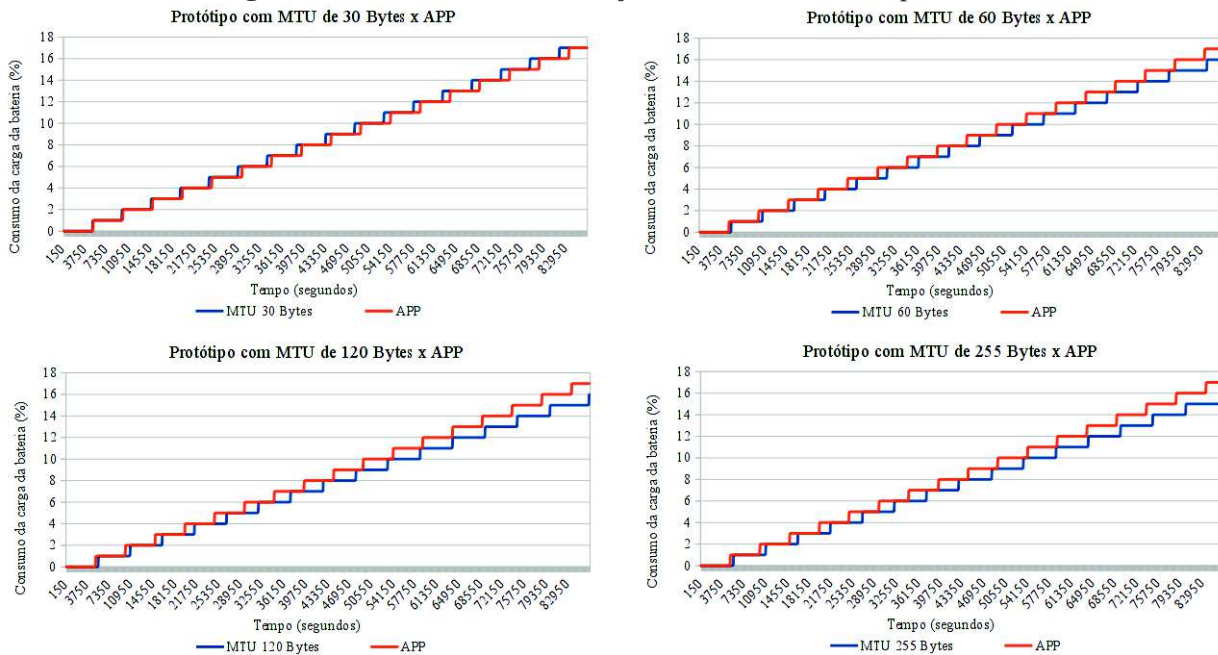
Fonte: Elaborado pelo autor

**Figura 25:** Comparativo do consumo energético da cinta utilizando as diferentes parametrizações de MTU no protótipo



Fonte: Elaborado pelo autor



**Figura 26:** Os resultados da variação do *MTU* referente à pulseira

Fonte: Elaborado pelo autor

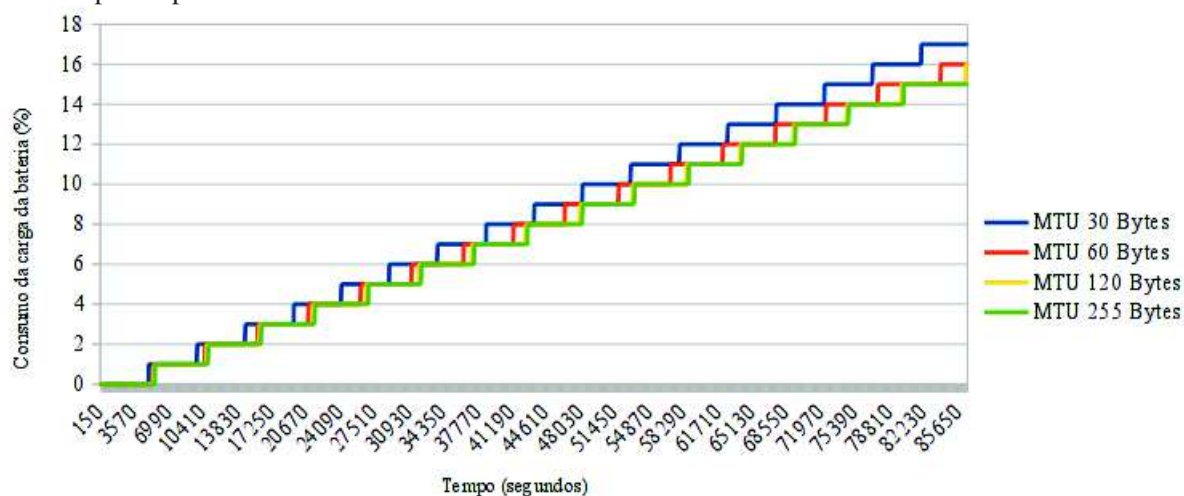
sendo que o dispositivo perdia 1% de sua carga a cada 5560 segundos em média, ou cerca de 1 hora e 32 minutos.

Já a Figura 27 mostra a variação do consumo energético da pulseira considerando as diferentes parametrizações de *MTU*. Nesse cenário, a variação do tamanho deste demonstrou que a configuração do *MTU* para 255 bytes causou uma redução de aproximadamente 12% no consumo de energia, quando comparado à configuração de *MTU* de 30 bytes. Verificou-se também que, quando o protótipo foi configurado com o valor *MTU* de 30 bytes, houve um aumento de aproximadamente 2% em relação à aplicação (*Elite HRV*) utilizada para leitura desse dispositivo. Quando o *MTU* do protótipo foi configurado para 60 bytes, houve uma redução no consumo de aproximadamente 6,3% em relação à aplicação. Quando o protótipo teve seu *MTU* configurado para uso de 120 bytes, a redução chegou a 9%. Por fim, quando o protótipo foi configurado com *MTU* no valor de 255 bytes, a redução no consumo energético alcançou 10,2%.

Em relação ao uso de diversas aplicações simultâneas executadas no protótipo que requisitam dados da cinta cardíaca, os resultados obtidos dessa análise são mostrados na Figura 28. Verificou-se que quando executada uma aplicação apenas no protótipo, o consumo da bateria do dispositivo atingiu 1% na marca dos 34230 segundos, ou 9 horas e 30 minutos. Já a marca de 2% de consumo foi atingida na marca dos 68370 segundos, ou 18 horas e 59 minutos. Quando executadas 2 aplicações simultâneas no protótipo, o consumo da bateria do dispositivo atingiu 1% na marca dos 33780 segundos, ou 9 horas e 23 minutos. Já a marca de 2% de consumo foi atingida na marca dos 67650 segundos, ou 18 horas e 47 minutos. Finalmente, quando executadas 4 aplicações simultâneas no protótipo, o consumo da bateria da cinta atingiu 1% na marca dos 33570 segundos, ou 9 horas e 17 minutos. Já a marca de 2% de consumo foi atingida na



**Figura 27:** Comparativo do consumo energético da pulseira utilizando as diferentes parametrizações de MTU no protótipo



Fonte: Elaborado pelo autor

marca dos 67110 segundos, ou 18 horas e 38 minutos.

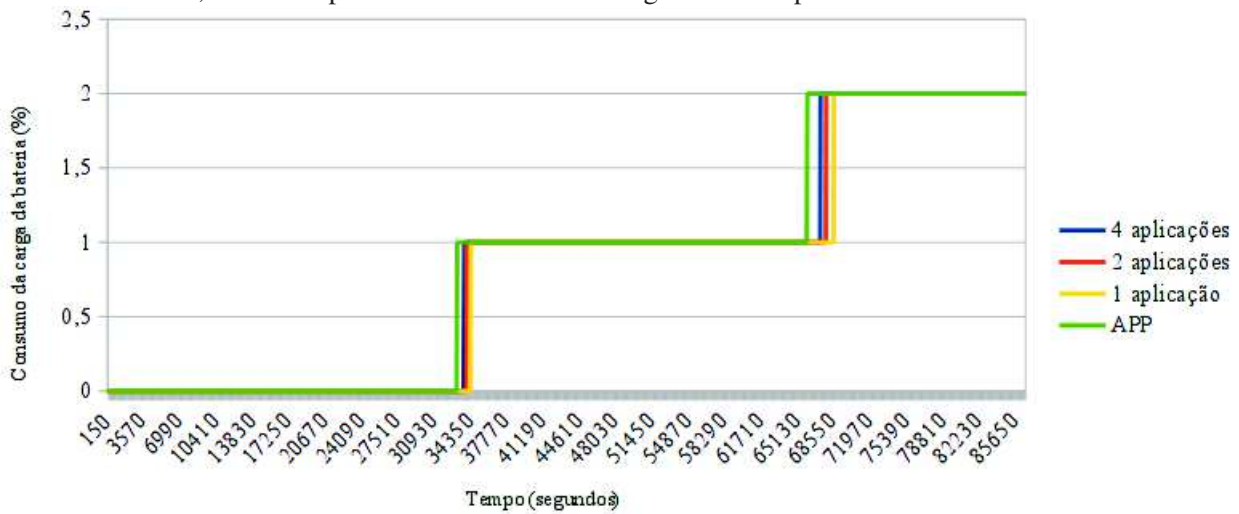
Nessa mesma análise, ainda, conforme demonstrado na Figura 28, verificou-se que, no que diz respeito ao uso da cinta, quando executada uma aplicação apenas no protótipo, há uma redução no consumo de energia de aproximadamente 4% quando comparado com o aplicativo *EliteHRV*. Quando executadas duas aplicações simultâneas no protótipo, há uma redução de energia de cerca de 3%. Finalmente, quando executadas 4 aplicações simultâneas no protótipo, a redução do consumo de energia da cinta cai para cerca de 1% quando comparada ao aplicativo.

Em relação ao uso de diversas aplicações simultâneas executadas no protótipo que requisitam dados da pulseira, os resultados obtidos dessa análise são mostrados na Figura 29. Verificou-se que quando executada uma aplicação apenas no protótipo, houve um decremento de 15% na carga da bateria, sendo que o dispositivo perdia 1% de sua carga a cada 5546 segundos em média, ou cerca de 1 hora e 54 minutos. Quando executadas 2 aplicações simultâneas no protótipo, houve um decremento de 17% na carga da bateria, sendo que o dispositivo perdia 1% de sua carga a cada 4983 segundos em média, ou cerca de 1 hora e 38 minutos. Quando executadas 4 aplicações simultâneas no protótipo, houve um decremento de 17% na carga da bateria, sendo que o dispositivo perdia 1% de sua carga a cada 4800 segundos em média, ou cerca de 1 hora e 33 minutos.

Nessa mesma análise, ainda, conforme demonstrado na Figura 29, verificou-se que quando executada uma aplicação apenas no protótipo, há uma redução no consumo de energia de aproximadamente 11% quando comparado com o aplicativo *HBand 2.0*. Quando executadas duas aplicações simultâneas no protótipo, há uma redução de energia de cerca de 2%. Por fim, quando executadas quatro aplicações simultâneas no protótipo, ocorre a situação inversa: o protótipo consome cerca de 2% a mais de energia quando comparada ao aplicativo.

Com base nos resultados apresentados, há uma relação inversamente proporcional entre

**Figura 28:** Comparativo do consumo energético da cinta quando diversas aplicações são executadas simultaneamente, e seu comparativo com o consumo registrado no aplicativo *EliteHRV*



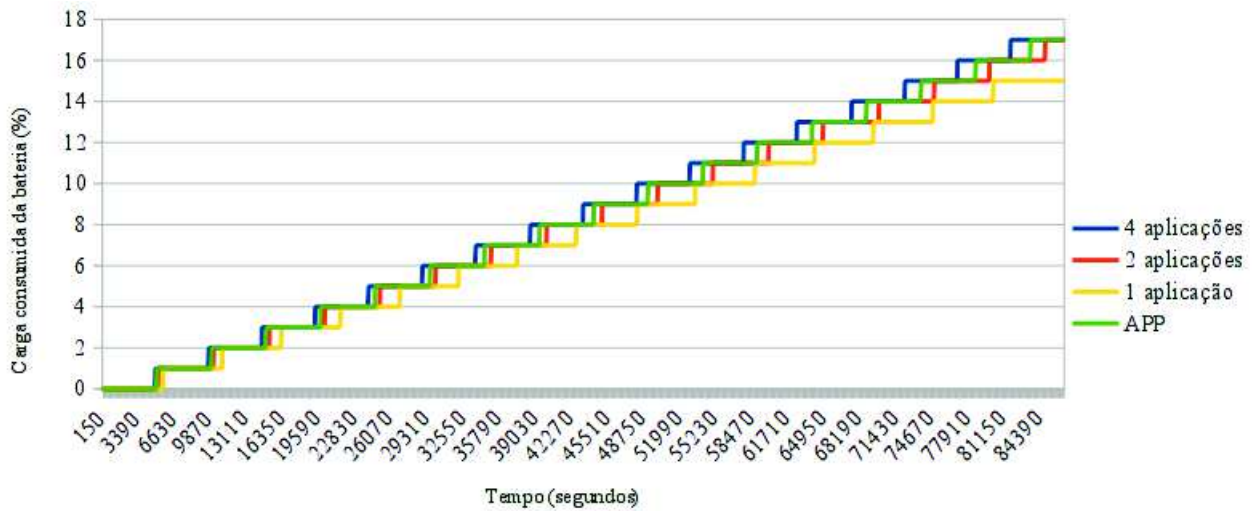
Fonte: Elaborado pelo autor

o tamanho do *MTU* e o consumo energético, conforme esperado nos resultados descritos na Tabela 4. Até um certo valor, quanto maior o *MTU* aplicado, menor era o consumo da cinta cardíaca e da pulseira. O que determina esse valor é o tipo de perfil *GATT/GAP* utilizado pelo *BLE*, que determina a quantidade de dados que trafega na rede. No caso da cinta cardíaca, o consumo energético estabilizou-se quando teve seu *MTU* configurado a partir dos 60 *bytes*, enquanto que na pulseira o consumo energético estabilizou-se a partir dos 120 *bytes*. Ainda, em relação ao *MTU*, percebe-se que, quando utilizados valores baixos (30 *bytes*), percebe-se que o protótipo consumiu mais energia, se comparado às aplicações utilizadas como parâmetro nessa análise.

Já em relação ao uso de diversas aplicações simultâneas executadas no protótipo, há uma relação diretamente proporcional entre a quantidade de aplicações simultâneas requisitando dados dos dispositivos e o consumo energético destes, conforme esperado nos resultados descritos na Tabela 4. Quanto maior o número de aplicações simultâneas, maior era o consumo energético da cinta cardíaca e da pulseira. Quando utilizadas até 2 aplicações simultâneas, o consumo do protótipo foi inferior ao consumo de uma única aplicação utilizada como parâmetro nessa análise. Quando utilizadas 4 aplicações simultâneas, porém, o protótipo apresentou um consumo levemente superior (2%) se comparado à aplicação utilizada como parâmetro. Isso demonstra que várias aplicações podem requisitar dados a um mesmo dispositivo, sem comprometimento significativo do consumo energético deste.

Por fim, verificou-se também que os dispositivos utilizados para avaliação desse protótipo apresenta consumos de energia distintos, significando que o *hardware* impacta de maneira decisiva no consumo energético. Isso, todavia, não significa que a comunicação de dados não tenha sua relevância no que diz respeito ao consumo energético, conforme verificou-se na avaliação deste protótipo.

**Figura 29:** Comparativo do consumo energético da pulseira quando diversas aplicações são executadas simultaneamente, e seu comparativo com o consumo registrado no aplicativo *HBand*



Fonte: Elaborado pelo autor

### 6.1.2 Ambiente de Simulação

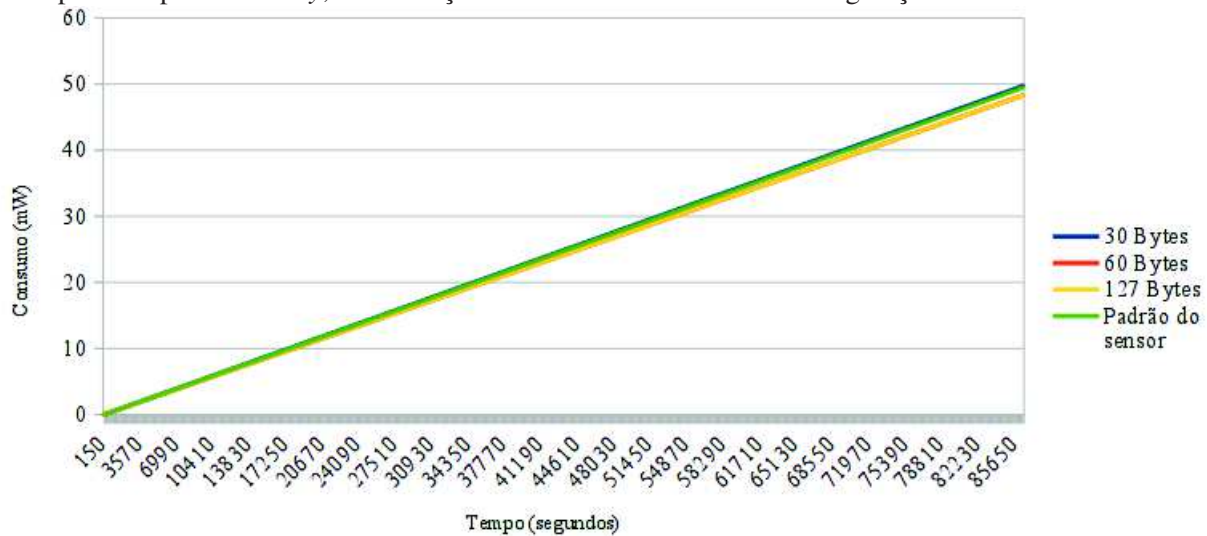
Conforme descrito na Seção 5.2.1, a avaliação do consumo de energia foi feita considerando a variação do tamanho do *MTU* implementado pelo protocolo *IEEE 802.15.4* e também considerando o uso de aplicações simultâneas executadas sobre um dispositivo *Tmote-sky* configurado para executar a função de intermediário entre a comunicação de aplicações e os demais dispositivos da rede *WBAN*, conforme descrito na Seção 5.2. Nesse cenário, esse dispositivo passa a ser chamado de dispositivo-pai.

No que diz respeito à variação do *MTU*, a média do consumo dos dispositivos são mostrados na Figura 30. Nesta configuração, optou-se pela execução considerando que a rede *WBAN* era formada por 2 dispositivos. Como a rede *WBAN* utiliza o mesmo modelo de dispositivo, e a variação dos valores medidos foi inferior a 1%, optou-se pelo cálculo da média, com o objetivo de não poluir o gráfico resultante.

Como resultado do experimento, verificou-se que, quando simulada a comunicação entre os dispositivos sem o uso do dispositivo-pai, a comunicação entre aplicação e dispositivos consumiu, em média, 49,5 mW, ou cerca de 2,06 mW por hora. Quando configurado o dispositivo-pai com *MTU* de 30 bytes, a comunicação entre aplicação e dispositivos consumiu, em média, 49,8 mW, ou cerca de 2,07 mW por hora. Quando configurado o dispositivo-pai com *MTU* de 60 bytes, a comunicação entre aplicação e dispositivos consumiu, em média, 48,3 mW, ou cerca de 2,01 mW por hora. Quando configurado o dispositivo-pai com *MTU* de 127 bytes, a comunicação entre aplicação e dispositivos consumiu, em média, os mesmos 48,3 mW.

Nessa mesma análise, ainda, conforme demonstrado na Figura 30, verificou-se que a configuração do *MTU* em 30 bytes ocasionou um leve aumento do consumo de energia, não superior a 1%, em relação ao cenário sem o uso do dispositivo-pai. Quando o *MTU* do dispositivo-pai

**Figura 30:** Comparativo do consumo energético dos dispositivos *IEEE 802.15.4* quando alterado o MTU do dispositivo-pai *Tmote-sky*, e em relação ao cenário sem realizar a configuração



Fonte: Elaborado pelo autor

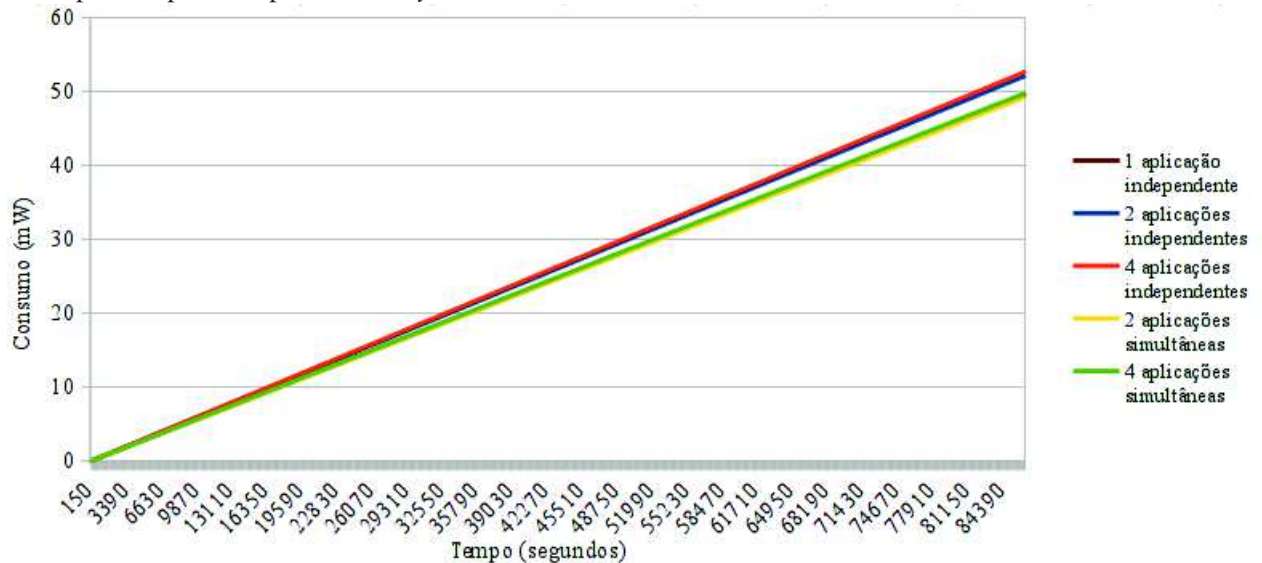
foi configurado para 60 bytes, houve uma redução no consumo de aproximadamente 2,5% em relação ao em relação ao cenário sem o uso do dispositivo-pai. Resultado semelhante foi obtido quando o MTU do dispositivo-pai foi configurado para 127 bytes.

No que diz respeito à avaliação de aplicações simultâneas executadas sobre a rede *WBAN*, a Figura 31 sintetiza o consumo energético quando diversas aplicações simultâneas são executadas sobre o dispositivo-pai numa rede formada por 2 dispositivos. Para efeitos de realização desse teste, ainda, o *MTU* do dispositivo-pai foi configurado para 127 bytes.

Como resultado do experimento, verificou-se que, quando simulada a comunicação entre a aplicação e os dispositivos sem o uso do dispositivo-pai, a comunicação entre a aplicação e os dispositivos consumiu, em média, 49,5 mW, ou cerca de 2,06 mW por hora. Quando simulada a comunicação de 2 aplicações com os dispositivos sem o uso do dispositivo-pai, a comunicação entre as aplicações e os dispositivos consumiu, em média, 52,1 mW, ou cerca de 2,17 mW por hora. Quando simulada a comunicação de 4 aplicações com os dispositivos sem o uso do dispositivo-pai, a comunicação entre as aplicações e os dispositivos consumiu, em média, 52,7 mW, ou cerca de 2,19 mW por hora. Quando simulada a comunicação de 2 aplicações com os dispositivos, tendo o dispositivo-pai como intermediário da comunicação, a comunicação entre as aplicações e os dispositivos consumiu, em média, 49,2 mW, ou cerca de 2,05 mW por hora. Quando simulada a comunicação de 4 aplicações com os dispositivos, tendo o dispositivo-pai como intermediário da comunicação, a comunicação entre as aplicações e os dispositivos consumiu, em média, 49,8 mW, ou cerca de 2,07 mW por hora.

Nesta mesma análise, ainda, conforme demonstrado na Figura 31, verificou se que, quando usado o dispositivo-pai com 2 aplicações simultâneas, houve uma redução de aproximadamente 5,6% no consumo, quando comparado ao mesmo cenário, onde 2 aplicações indepen-

**Figura 31:** Comparativo da média do consumo energético dos 2 dispositivos *IEEE 802.15.4* quando utilizado aplicações simultâneas executadas independentemente e quando executadas de forma simultânea pelo dispositivo-pai *Tmote-sky*



Fonte: Elaborado pelo autor

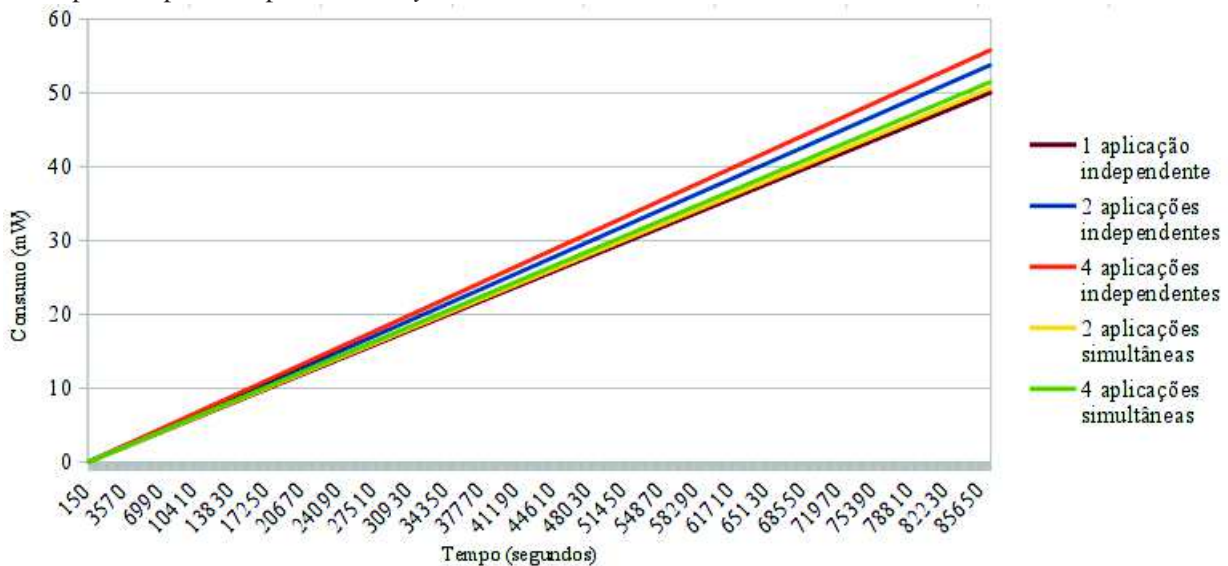
dentes comunicam-se com a rede *WBAN*. Nesta mesma configuração, ainda, verificou-se que o dispositivo-pai gastou aproximadamente 1% a mais de energia se comparado ao cenário onde uma aplicação comunica-se diretamente com os dispositivos da rede. Quando usado o dispositivo-pai com 4 aplicações simultâneas, também apresentou uma redução de aproximadamente 5,6% no consumo, quando comparado ao mesmo cenário, onde 4 aplicações independentes comunicam-se com a rede *WBAN*. Todavia, quando comparado ao cenário onde uma aplicação comunica-se diretamente com os dispositivos da rede, houve um acréscimo de 1% no consumo dos dispositivos.

Já no que diz respeito à avaliação de aplicações simultâneas executadas sobre a rede *WBAN*, a Figura 32 sintetiza o consumo energético quando diversas aplicações simultâneas são executadas numa rede formada por 4 dispositivos. Neste cenário, também, o *MTU* do dispositivo-pai foi configurado para 127 bytes.

Como resultado do experimento, verificou-se que, quando simulada a comunicação entre a aplicação e os dispositivos sem o uso do dispositivo-pai, a comunicação entre a aplicação e os dispositivos consumiu, em média, 50,1 mW, ou cerca de 2,08 mW por hora. Quando simulada a comunicação de 2 aplicações com os dispositivos sem o uso do dispositivo-pai, a comunicação entre as aplicações e os dispositivos consumiu, em média, 53,8 mW, ou cerca de 2,24 mW por hora. Quando simulada a comunicação de 4 aplicações com os dispositivos sem o uso do dispositivo-pai, a comunicação entre as aplicações e os dispositivos consumiu, em média, 55,8 mW, ou cerca de 2,32 mW por hora. Quando simulada a comunicação de 2 aplicações com os dispositivos, tendo o dispositivo-pai como intermediário da comunicação, a comunicação entre as aplicações e os dispositivos consumiu, em média, 50,6 mW, ou cerca de 2,10 mW por



**Figura 32:** Comparativo da média do consumo energético dos 4 dispositivos *IEEE 802.15.4* quando utilizado aplicações simultâneas executadas independentemente e quando executadas de forma simultânea pelo dispositivo-pai *Tmote-sky*



Fonte: Elaborado pelo autor

hora. Quando simulada a comunicação de 4 aplicações com os dispositivos, tendo o dispositivo-pai como intermediário da comunicação, a comunicação entre as aplicações e os dispositivos consumiu, em média, 51,5 mW, ou cerca de 2,14 mW por hora.

Nesta mesma análise, ainda, conforme demonstrado na Figura 32, verificou-se que, quando usado o dispositivo-pai com 2 aplicações simultâneas, houve uma redução de aproximadamente 6% no consumo, quando comparado ao mesmo cenário, onde 2 aplicações independentes comunicam-se com a rede *WBAN*. Nesta mesma configuração, ainda, verificou-se que o cenário composto pelo dispositivo-pai gastou aproximadamente 1% a menos de energia se comparado ao cenário onde uma aplicação comunica-se diretamente com os dispositivos da rede. Quando usado o dispositivo-pai com 4 aplicações simultâneas, também apresentou uma redução de aproximadamente 7,8% no consumo, quando comparado ao mesmo cenário, onde 4 aplicações independentes comunicam-se com a rede *WBAN*. Todavia, quando comparado ao cenário onde uma aplicação comunica-se diretamente com os dispositivos da rede, houve um acréscimo de 3,2% no consumo dos dispositivos.

Com base nos resultados apresentados, da mesma maneira que ocorreu na discussão descrita na Seção 6.1.1, há uma relação inversamente proporcional entre o tamanho do *MTU* e o consumo energético, conforme esperado nos resultados descritos na Tabela 4. Até um certo valor, quanto maior o *MTU* aplicado, menor era o consumo do dispositivo. O que determina esse valor era o tamanho do quadro gerado pelo comando *ping*, que estava configurado com valor de 32 *bytes*. No ambiente simulado, consumo energético estabilizou-se quando teve seu *MTU* configurado a partir dos 60 *bytes*. Ainda, em relação ao *MTU*, percebe-se que, quando utilizados valores baixos (30 *bytes*), percebe-se que o protótipo consumiu mais energia, se comparado às

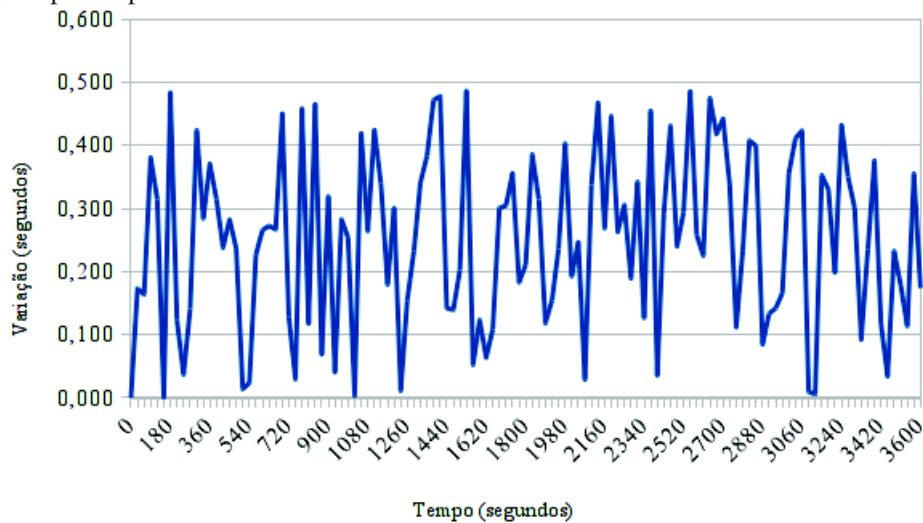
aplicações utilizadas como parâmetro nessa análise. Por fim, os resultados em relação a esse parâmetro seguem basicamente o mesmo padrão de redução do consumo energético apresentado no trabalho de Karthi, Rao e Pillai (2015).

Nesse trabalho, o autor obteve variações de até 40% em relação aos diversos cenários avaliados. Todavia, foram analisados *MTUs* ainda menores (20 *bytes*), bem como o autor também explorou a variação do ciclo de trabalho, aqui não abordada. No que diz respeito à variação do *MTU*, conforme os resultados apresentados nesta avaliação, percebe-se que as maiores diferenças de consumo energético, tanto no protocolo *IEEE 802.15.4*, como no *BLE* foram observadas no intervalo de *MTUs* de 30 a 60 *bytes*. Se fossem analisados *MTUs* ainda menores, a variação do consumo energético seria maior.

Já em relação ao uso de diversas aplicações simultâneas executadas no dispositivo-pai, há uma relação diretamente proporcional entre a quantidade de aplicações simultâneas requisitando dados dos dispositivos e o consumo energético destes, de maneira análoga ao que ocorreu na discussão descrita na Seção 6.1.1, e conforme esperado nos resultados descritos na Tabela 4. Quanto maior o número de aplicações simultâneas, maior era o consumo energético do dispositivo. No cenário onde a rede *WBAN* utilizava 2 dispositivos, quando um dos dispositivos *Tmote-sky* foi configurado para o uso de 2 aplicações simultâneas, houve redução de energia dos dispositivos que formam a rede *WBAN* quando comparado a uma única aplicação requisitando dados aos mesmos dispositivos. Quando utilizadas 4 aplicações simultâneas, porém, houve um leve aumento do consumo (1%) quando comparado a uma única aplicação. De maneira análoga ao verificado na Seção 6.1.1, isso demonstra que várias aplicações podem requisitar dados a um mesmo dispositivo, sem comprometimento significativo do consumo energético deste. Já no cenário onde a rede *WBAN* utilizava 4 dispositivos, quando um dos dispositivos *Tmote-sky* foi configurado para o uso de 2 e 4 aplicações simultâneas, não houve redução de energia dos dispositivos que formam a rede *WBAN* quando comparado a uma única aplicação requisitando dados aos mesmos dispositivos. Todavia, quando comparado à 2 ou 4 aplicações independentes requisitando dados, percebe-se que há uma significativa redução de energia, chegando a valores de até 7,8%.

A explicação para esse aumento do consumo de energia está relacionado às características do próprio protocolo *IEEE 802.15.4* (IEEE, 2011), pois os dispositivos que utilizam esse padrão acabam trocando informações entre si, como uma rede *P2P* (IEEE, 2011). Dessa forma, quanto mais dispositivos na rede, maior a troca de dados. Ainda, como foram utilizadas metodologias de avaliação diferentes na análise dos dispositivos *BLE* e *IEEE 802.15.4*, onde o primeiro é avaliado pela carga da bateria, enquanto que o segundo é calculado efetivamente pelo consumo energético do protocolo de comunicação, não pode ser comparado o consumo energético entre os protocolos *BLE* e *IEEE 802.15.4*, de maneira análoga ao trabalho desenvolvido por Siekkinen et al. (2012).

**Figura 33:** Variação do horário entre o horário registrado pelo relógio interno da pulseira e o horário recalculado pelo protótipo.



Fonte: Elaborado pelo autor

## 6.2 Sincronismo

Conforme descrito na Seção 5.1.2, a avaliação de sincronismo foi feita utilizando o método de sincronização de tempo por ordem de eventos proposto no Capítulo 4. A Figura 33 mostra a variação entre o horário registrado pelo relógio interno da pulseira e o horário recalculado pelo protótipo durante o período de 1 hora, considerando intervalos de registros de medição de 30 segundos.

Durante a execução dessa avaliação, verificou-se que a menor variação de horário foi de 1 milissegundo, registrado aos 150 segundos, ou 2 minutos e 30 segundos. Já a maior variação registrada foi de 487 ms, registrada aos 1530 segundos, ou 25 minutos e 30 segundos. Já a média de variação registrada foi de 251 milissegundos. A Tabela 5 mostra, ainda, um trecho de 5 minutos com os horários registrados pela pulseira, o horário em que os dados foram recebidos pelo protótipo e o horário recalculado.

Com o objetivo de validar a sincronização dos horários considerando o caso de um dispositivo estar com seu horário totalmente dessincronizado, a mesma avaliação foi repetida num cenário onde o relógio da pulseira não estava sincronizado. A Tabela 6 mostra os resultados obtidos dos 5 primeiros minutos considerando esse caso.

Como pode ser verificado nas Tabelas 5 e 6, o horário de registro dos dados pelo protótipo desconsidera o horário do dispositivo que forma a rede *WBAN*. Apesar da variação nos horários registrados pela pulseira e pelo protótipo, devido ao fato de que a medição dos sinais vitais não necessitem de uma alta taxa de precisão e precisar ser medida durante um intervalo considerável e que as amostras são colhidas em intervalos de 1 segundo, conforme descrito na Seção 4.1.3, verifica-se que o método de sincronização apresentada pelo protótipo atende de maneira



**Tabela 5:** Valores sincronizados pelo protótipo, tendo como base o horário da chegada dos dados da pulseira

<b>Tempo</b>	<b>Horário re-gistrado na pulseira</b>	<b>Horário de chegada dos dados</b>	<b>Média do intervalo entre o recebimento dos dados</b>	<b>Horário ajustado</b>	<b>Variação em relação ao horário da pulseira</b>
0 s	22/06/2017 07:55:08,137	22/06/2017 07:55:08,343	0 ms	22/06/2017 07:55:08,343	0 ms
30 s	22/06/2017 07:55:38,122	22/06/2017 07:55:38,454	503 ms	22/06/2017 07:55:37,951	173 ms
60 s	22/06/2017 07:56:08,138	22/06/2017 07:56:08,800	498 ms	22/06/2017 07:56:08,302	164 ms
90 s	22/06/2017 07:56:38,129	22/06/2017 07:56:38,253	505 ms	22/06/2017 07:56:37,748	381 ms
120 s	22/06/2017 07:57:08,133	22/06/2017 07:57:08,319	501 ms	22/06/2017 07:57:07,818	315 ms
150 s	22/06/2017 07:57:38,119	22/06/2017 07:57:38,617	497 ms	22/06/2017 07:57:38,120	1 ms
180 s	22/06/2017 07:58:08,151	22/06/2017 07:58:08,167	500 ms	22/06/2017 07:58:07,667	484 ms
210 s	22/06/2017 07:58:38,122	22/06/2017 07:58:38,498	499 ms	22/06/2017 07:58:37,999	123 ms
240 s	22/06/2017 07:59:08,144	22/06/2017 07:59:08,684	503 ms	22/06/2017 07:59:08,181	37 ms
270 s	22/06/2017 07:59:38,133	22/06/2017 07:59:38,491	499 ms	22/06/2017 07:59:37,992	141 ms
300 s	22/06/2017 08:00:08,139	22/06/2017 08:00:08,217	502 ms	22/06/2017 08:00:07,715	424 ms

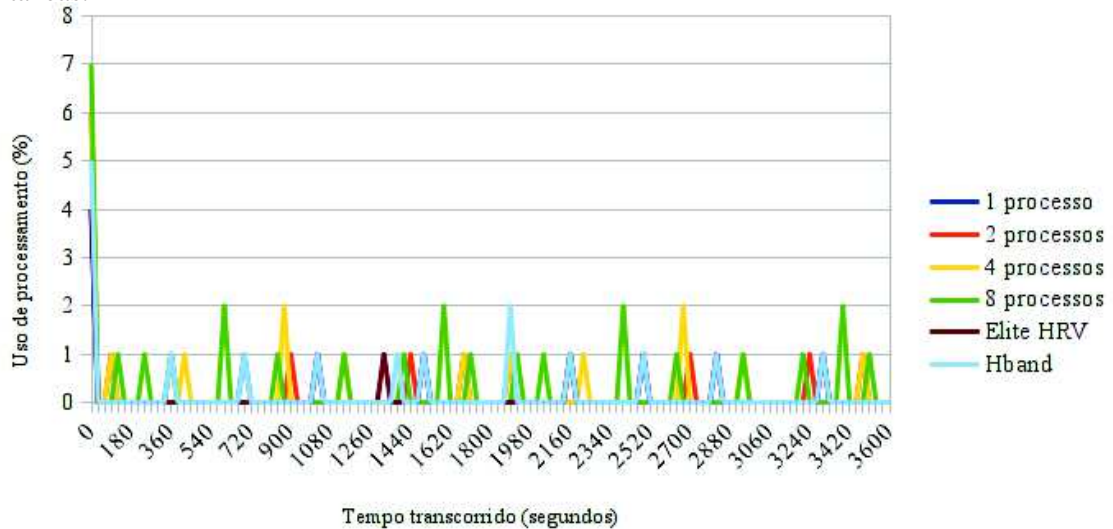
Fonte: Elaborada pelo autor.

**Tabela 6:** Valores sincronizados pelo protótipo, considerando que o horário da pulseira não foi sincronizado.

<b>Tempo</b>	<b>Horário registrado na pulseira</b>	<b>Horário de chegada dos dados</b>	<b>Média do intervalo entre recebimento dos dados</b>	<b>Horário ajustado</b>
0 s	01/03/2016 00:05:06,219	27/06/2017 13:15:07,434	0 ms	27/06/2017 13:15:07,434
30 s	01/03/2016 00:05:36,122	27/06/2017 13:15:37,338	498 ms	27/06/2017 13:15:36,840
60 s	01/03/2016 00:06:06,375	27/06/2017 13:16:07,592	504 ms	27/06/2017 13:16:07,088
90 s	01/03/2016 00:06:36,083	27/06/2017 13:16:37,301	495 ms	27/06/2017 13:16:36,806
120 s	01/03/2016 00:07:06,308	27/06/2017 13:17:07,527	504 ms	27/06/2017 13:17:07,023
150 s	01/03/2016 00:07:36,366	27/06/2017 13:17:37,586	501 ms	27/06/2017 13:17:37,085
180 s	01/03/2016 00:08:06,370	27/06/2017 13:18:07,591	500 ms	27/06/2017 13:18:07,091
210 s	01/03/2016 00:08:36,055	27/06/2017 13:18:37,277	494 ms	27/06/2017 13:18:36,783
240 s	01/03/2016 00:09:06,108	27/06/2017 13:19:07,331	501 ms	27/06/2017 13:19:06,830
270 s	01/03/2016 00:09:36,085	27/06/2017 13:19:37,309	500 ms	27/06/2017 13:19:36,809
300 s	01/03/2016 00:10:06,029	27/06/2017 13:20:07,254	499 ms	27/06/2017 13:20:06,755

Fonte: Elaborada pelo autor.

**Figura 34:** Avaliação de uso de processamento do protótipo considerando o uso de 1, 2, 4 e 8 aplicações simultâneas.



Fonte: Elaborado pelo autor

satisfatória os requisitos exigidos para dispositivos de saúde, independente do horário apresentado pelo dispositivo, conforme esperado na Tabela 4. Conforme discutido também na Seção 6.1.2, utilizou-se um ciclo de trabalho relativamente alto, devido à limitação do SO *Android* descrita na Seção 5. Isso contribuiu para esse resultado satisfatório. Caso fossem usados ciclos de trabalho menores, possivelmente haveriam maiores distorções de sincronismo temporal.

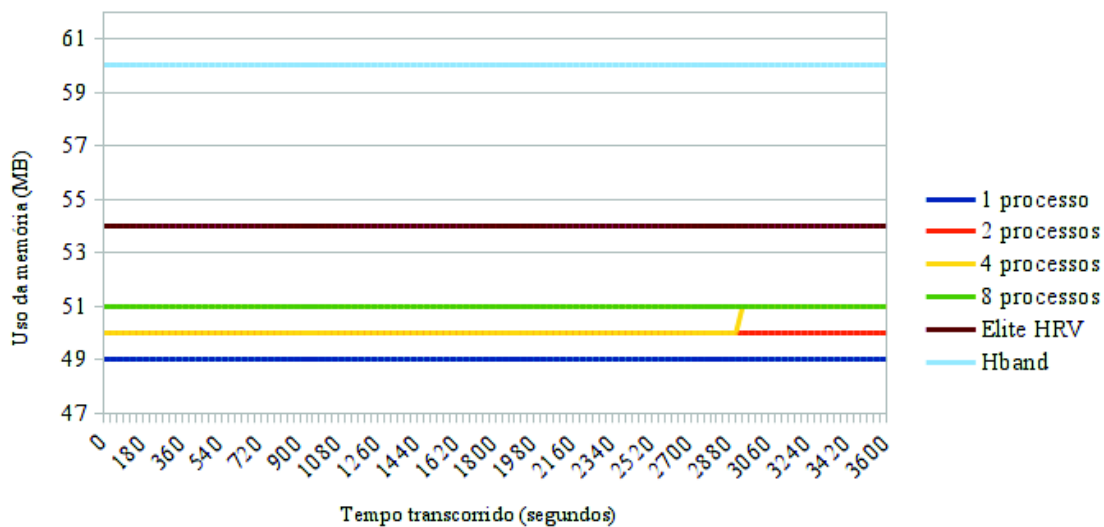
### 6.3 Carga

Conforme descrito na Seção 5.1.3, a avaliação de carga foi feita sobre o protótipo considerando a variação da quantidade de aplicações simultâneas sendo executadas.

A Figura 34 mostra o percentual de processamento utilizado pelo protótipo considerando o uso de 1, 2, 4 e 8 aplicações simultâneas, durante um período de 3600 segundos, ou 1 hora. Os resultados apresentam a média de uso entre as duas baterias de testes.

Nesta avaliação percebe-se que, em todos os cenários analisados, o pico de processamento ocorre no início da transmissão de dados. Ainda, em todos os cenários avaliados, verificou-se que, na maior parte do tempo, o consumo de processamento foi inferior a 1%. Considerando o cenário onde é executada 1 aplicação simultânea sobre o protótipo, percebe-se que o pico de processamento foi de 4%. No caso do cenário onde o protótipo executa 2 aplicações simultâneas, o pico de processamento é de 6%. Em relação ao cenário onde o protótipo executa 4 aplicações simultâneas, o pico de processamento permanece em 6%. Já no cenário onde o protótipo executa 8 aplicações simultâneas, o pico de processamento passa para 7%. No que diz respeito às aplicações utilizadas para comparativo, verificou-se que o pico de processamento do aplicativo *Elite HRV* foi de 4%, enquanto que o aplicativo *HBand* teve um pico de processa-

**Figura 35:** Avaliação de uso de memória do protótipo considerando o uso de 1, 2, 4 e 8 aplicações simultâneas.



Fonte: Elaborado pelo autor

mento de 5%.

Nessa mesma análise, ainda, verificou-se que no cenário onde o protótipo executa uma aplicação simultânea registrou uma média de uso do processador de 0,13%. No caso do cenário onde o protótipo executa 2 aplicações simultâneas, a média de uso do processador foi de 0,16%. No caso do cenário onde o protótipo executa 4 aplicações simultâneas, a média de uso do processador foi de 0,20%. No caso do cenário onde o protótipo executa 8 aplicações simultâneas, a média de uso do processador foi de 0,30%. O aplicativo *EliteHRV*, durante sua execução, apresentou uma média de uso do processador de 0,11%. Por fim, o aplicativo *EliteHRV*, durante sua execução, apresentou uma média de uso do processador de 0,16%.

A Figura 35 mostra a quantidade de memória utilizada pelo protótipo considerando o uso de 1, 2, 4 e 8 aplicações simultâneas, durante um período de 3600 segundos, ou 1 hora. Os resultados apresentam a média de uso entre as duas baterias de testes.

Nesta avaliação verifica-se que, exceto no cenário onde o protótipo executa 4 aplicações simultâneas, o uso de memória mantém-se estável. Considerando o cenário onde é executada 1 aplicação simultânea sobre o protótipo, percebe-se que o uso de memória manteve-se em 49 *MBytes*. No caso do cenário onde o protótipo executa 2 aplicações simultâneas, percebe-se que o uso de memória manteve-se em 50 *MBytes*. No cenário onde o protótipo executa 4 aplicações simultâneas, verifica-se que o consumo de memória inicia-se em 50 *MBytes* e, a partir do minuto 49, a aplicação passa a consumir 51 *MBytes* da memória. Já no caso onde o protótipo executa 8 aplicações simultâneas, verifica-se que o uso da memória manteve-se em 51 *MBytes*. No que diz respeito às aplicações utilizadas para comparativo, verificou-se que o aplicativo *Elite HRV* manteve o consumo em 54 *MBytes*, enquanto que o aplicativo *HBand* manteve o uso da memória em 60 *MBytes*.

Apesar da carga média de processamento do *smartphone* aumentar cerca de 230% em relação aos cenários onde o protótipo executa simultaneamente 1 e 8 aplicações, verifica-se que, ainda assim, a carga média de processamento não chega a 0,5% da capacidade do *smartphone*. Ainda, o pico máximo de processamento registrado em todos os cenários de avaliação foi de 7%. Isso significa que uma eventual sobrecarga de processamento ocorreria somente se houvessem dezenas de aplicações simultâneas sendo executadas pelo *protótipo*.

Já no que se refere ao uso da memória, verificou-se que, nos diversos cenários de execução do protótipo, o uso da memória aumentou em apenas 2 *MBytes*, o que significa que há pouco impacto sobre o *smartphone* caso o aumento o número de aplicações externas executadas simultaneamente. Ainda, percebeu-se que as aplicações *Elite HRV* e *HBand* consumiram mais memória que o protótipo, o que reforça a afirmação de que este utiliza poucos recursos de *hardware*.

Desta maneira, verifica-se que o modelo proposto - centralizado num dispositivo móvel com grande poder de processamento se comparado aos dispositivos *IoT* em geral - fornece um certo nível de escalabilidade às redes *WBAN*. Isso significa que essa solução pode atender às atuais e, possivelmente, futuras demandas relacionadas à requisição de dados de dispositivos que formam uma rede *WBAN* por meio de aplicações externas sem grandes prejuízos no que diz respeito à sobrecarga de *hardware* desse dispositivo, atendendo ao resultado esperado descrito na Tabela 4.

#### 6.4 Considerações parciais

Neste capítulo foram apresentados e discutidos os resultados referentes às avaliações do protótipo e do ambiente de simulação descritos no Capítulo 5. Foram apresentados 3 tipos de resultados: de consumo energético, de sincronismo de tempo e de carga. A avaliação energética, realizada sobre o protótipo e o ambiente de simulação, apresentou resultados satisfatórios quando ambos foram configurados para uso de *MTUs* com tamanhos superiores a 60 *Bytes* e também quando os protótipos utilizaram diversas aplicações simultâneas. Já a avaliação de sincronismo, realizada sobre o protótipo, atingiu resultados satisfatórios, atingindo variações de sincronização inferiores à precisão exigida para dispositivos de monitoramento médico. Por fim, a avaliação de carga, realizada sobre o protótipo, demonstrou que não há sobrecarga no *smartphone* em nenhum dos cenários de uso de aplicações simultâneas utilizadas nesta análise.



## 7 CONCLUSÃO

Este trabalho apresentou o modelo *MOMICARE*, que consiste no projeto de um *middleware* móvel que funcione a partir de um *smartphone* que atue como um intermediário no recebimento dos dados dos diversos sensores espalhados pelo corpo de uma pessoa, armazene-os e os retransmita às aplicações médicas que desejem utilizar esses dados. Esse modelo responde à pergunta de pesquisa abordada na Seção 1.1, por intermédio de uma solução onde um *smartphone* atua como um intermediário entre as aplicações médicas e os sensores corporais, oferecendo, principalmente, uma comunicação energeticamente eficiente.

Inicialmente, foi apresentado os conceitos técnicos que fornecem as bases para o entendimento do modelo. Em seguida, passou-se à avaliação de sete trabalhos relacionados, que acabaram evidenciando as lacunas que serviram de base para a proposta do modelo *MOMICARE*: eficiência energética, sincronização e interoperabilidade. Os protótipos desenvolvidos para validação do modelo objetivaram a realização de testes sobre as lacunas identificadas: consumo energético dos sensores que compõem a rede *WBAN*, sincronização de horários entre os dados do *middleware* e avaliação de carga do *middleware*. De maneira geral, verificou-se que o modelo proposto neste trabalho é válido, apresentando os resultados esperados.

### 7.1 Contribuição Científica

A contribuição científica do modelo *MOMICARE* está relacionada, principalmente, à redução do consumo de energia dos sensores corporais e ao sincronismo de tempo nos dados enviados pelos dispositivos que compõem uma rede *WBAN*. Independente do padrão de comunicação utilizado (*BLE* ou *IEEE 802.15.4*), o modelo apresentou redução do consumo de energia nos dispositivos. No que refere-se a dispositivos médicos, esse resultado significa que o tempo de uso dos sensores sem necessidade de interrupção para troca ou recarga de bateria é ampliado. Isso é extremamente importante quando tratam-se de sensores intrusivos, onde a manutenção de um dispositivo de monitoramento requer intervenção cirúrgica.

Em relação à sincronização temporal, verificou-se que a variação de tempo entre os horários registrados pelos dispositivos e pelo horário recalculado pelo protótipo atende satisfatoriamente aos requisitos de avaliação médica. Em relação a essa contribuição, também, cabe ressaltar que vários dispositivos - como a cinta cardíaca utilizada na avaliação deste trabalho - sequer possuem mecanismo de registro de tempo, reforçando a necessidade de que algum outro mecanismo faça essa medição, permitindo que os dados de outros sensores corporais possam ser correlacionados. A Tabela 7 apresenta um resumo das contribuições científicas acrescentadas pelo modelo *MOMICARE* em relação aos demais trabalhos relacionados avaliados.

**Tabela 7:** Resumo das contribuições científicas do modelo *MOMICARE*

<b>Critério/ Trabalho</b>	<b>(LEE; JE- ONG; YOON, 2012)</b>	<b>(LANE et al., 2014)</b>	<b>(SEEGER; VAN LA- ERHO- VEN; BUCH- MANN, 2015)</b>	<b>(MAIA et al., 2015)</b>	<b>(JUNG et al., 2013)</b>	<b>(MOSER; MELLIAR- SMITH, 2015)</b>	<b>(CASTILLE et al., 2013)</b>	<b>(JOMI- CARE</b>
Segurança na transmissão	Não	Não	Não	Não	Não	Somente <i>Bluetooth</i>	Não	Sim
Escalabilidade	Não	Não	Não	Não	Não	Não	Sim	Sim
Interoperabilidade	3G, <i>WiFi</i> , <i>Bluetooth</i>	3G, <i>WiFi</i>	<i>WiFi</i> , <i>Bluetooth</i>	802.15.4, <i>Bluetooth</i>	<i>WiFi</i> , <i>Bluetooth</i>	3G, <i>WiFi</i> , <i>Bluetooth</i>	802.15.4, <i>WiFi</i> , <i>Bluetooth</i>	802.15.4, <i>Bluetooth</i>
Eficiência energética	Não	Sim	Não	Não	Não	Não	Não	Sim
Uso de sensores do próprio <i>smartphone</i>	Não	Acelerômetro, microfone	Acelerômetro	Não	Não	Acelerômetro, microfone	Não	Sim
Captura dos dados dos sensores	Em qualquer lugar	Em qualquer lugar	Em qualquer lugar	Locais determinados	Locais determinados	Em qualquer lugar	Locais determinados	Em qualquer lugar
Sincronização entre sensores	Não	Não	Não	Não	Não	Não	Não	Sim

Fonte: Elaborada pelo autor.



## 7.2 Artigos Publicados

Durante a elaboração desse trabalho o artigo "Comparativo do uso de OpenCL em CPU e GPU" foi aceito para composição dos anais e apresentação oral no XVI Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul (ERAD/RS). Além disso, o artigo "Um Middleware para Gerenciamento de Sensores de Saúde Corporais em Dispositivos Móveis", que aborda o funcionamento do modelo *MOMICARE*, foi aceito para composição dos anais e apresentação oral no 9º Simpósio Brasileiro de Computação Ubíqua e Pervasiva (SBCUP). Este último trabalho também recebeu menção honrosa entre os três melhores artigos do simpósio. Esses resultados demonstram a relevância do tema abordado perante a comunidade científica em geral.

## 7.3 Trabalhos futuros

Como trabalhos futuros, pretende-se implantar as especificações de segurança previstas no modelo e que, por questões de complexidade, não puderam ser implantadas e analisadas neste trabalho. Outra avaliação que pretende-se fazer é o uso do *6LowPAN*, que permite que os dispositivos *BLE* e *IEEE 802.15.4* possam comunicar-se diretamente pela *Internet*. Durante a avaliação do ambiente de simulação, montado sobre o *software Cooja*, verificou-se que os sensores funcionaram por meio do *6LowPAN*, porém seu funcionamento não pôde ser analisado.

Ainda em relação à eficiência energética, pode ser explorada ainda a variação do ciclo de trabalho, de forma a potencializar ainda mais o consumo energético, sem prejudicar o sincronismo temporal entre os dispositivos. Isso poderia ser possível implementando-se um algoritmo de período adaptativo que permita que o *MOMICARE* possa chegar a um ciclo de trabalho que faça esse equilíbrio entre eficiência energética e sincronismo.

A interoperabilidade entre os padrões *BLE* e *IEEE 802.15.4*, previsto no modelo mas também não avaliado neste trabalho por questões de complexidade e tempo, também pode ser analisada em um trabalho futuro. Conforme descrito no Capítulo 4, essa funcionalidade poderia ser avaliada por meio de uma interface externa para o padrão *IEEE 802.15.4*, permitindo que qualquer dispositivo que utilize esse padrão possa se comunicar a um *smartphone*. Dessa forma, dispositivos *BLE* e *IEEE 802.15.4* poderiam ser avaliados a partir de um único protótipo, oferecendo a possibilidade de verificar qual dos dois padrões consome menos energia no protótipo, de forma a validar os resultados apresentados no trabalho de Siekkinen et al. (2012).

Outra avaliação que pode ser realizada é em relação ao uso dos sensores que estão disponíveis em um *smartphone* (tais como acelerômetro e *GPS*), que permitem que os dados coletados pelos dispositivos da rede *WBAN* possam ser complementados com dados que possam fornecer a informação de um paciente em um determinado contexto, conforme comentado no Capítulo 1.

Ainda, em relação à escalabilidade de uma rede *WBAN*, os testes de avaliação sobre esse

requisito podem ser ampliados, analisando-se a perda de quadros de transmissão e o *throughput* dos protocolos *BLE* e *IEEE 802.15.4*, considerando o uso de um maior número de dispositivos sobre a rede *WBAN*.

Por fim, pretende-se realizar a avaliação do protótipo em conjunto com outros trabalhos desenvolvidos pela equipe do PIPCA, especialmente pelos integrantes do projeto *uHospital*. Um exemplo dessa possibilidade pode ser verificado no trabalho intitulado "Um Modelo Proativo de Antecipação de Ações de Times de Resposta Rápida Baseado em Análise Preditiva", desenvolvido pelo colega Flávio de Oliveira Dias, a análise preditiva dos sinais vitais poderia ser analisado a partir da base de dados do modelo *MOMICARE*.

## REFERÊNCIAS

- BATTISTI, J. **Redes Wireless – Parte XVIII**. Disponível em: <<http://juliobattisti.com.br/tutoriais/paulocfarias/redeswireless018.asp>>. Acesso em: 19.set.2016.
- BENHARREF, A.; SERHANI, M. A.; MIZOUNI, R. Smart Data Synchronization in m-Health Monitoring Applications. **2014 IEEE 16th International Conference on**, [S.l.], p. 140–145, 2014.
- CASAMASSIMA, F.; FARELLA, E.; BENINI, L. Synchronization methods for Bluetooth based WBANs. **2013 IEEE International Conference on Body Sensor Networks**, [S.l.], p. 1–6, 2013.
- CASTILLEJO, P.; MARTINEZ, J. F.; RODRIGUEZ-MOLINA, J.; CUERVA, A. Integration of wearable devices in a wireless sensor network for an E-health application. **IEEE Wireless Communications**, [S.l.], v. 20, n. 4, p. 38–49, aug 2013.
- CHUNG, W.-Y.; LEE, Y.-D.; JUNG, S.-J. A wireless sensor network compatible wearable u-healthcare monitoring system using integrated ECG, accelerometer and SpO<sub>2</sub>. **2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society**, [S.l.], p. 1529–1532, 2008.
- CONSTANTINESCU, L.; KIM, J.; FENG, D. SparkMed: a framework for dynamic integration of multimedia medical data into distributed m-health systems. **IEEE transactions on information technology in biomedicine**, [S.l.], v. 16, n. 1, p. 40–52, 2012.
- CONTIKI. **Contiki Open Source Operating System**. Disponível em: <<http://www.contiki-os.org>>. Acesso em: 05.fev.2017.
- COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T.; BLAIR, G. **Sistemas Distribuídos-: conceitos e projeto**. [S.l.]: Bookman Editora, 2013.
- CROSBY, G. V.; GHOSH, T.; MURIMI, R.; CHIN, C. A. Wireless body area networks for healthcare: a survey. **International Journal of Ad Hoc, Sensor & Ubiquitous Computing**, [S.l.], v. 3, n. 3, p. 1, 2012.
- DEMENTYEV, A.; HODGES, S.; TAYLOR, S.; SMITH, J. Power consumption analysis of Bluetooth Low Energy, ZigBee and ANT sensor nodes in a cyclic sleep scenario. **Wireless Symposium (IWS), 2013 IEEE International**, [S.l.], p. 1–4, 2013.
- DJENOURI, D.; BAGAA, M. Synchronization protocols and implementation issues in wireless sensor networks: a review. **IEEE Systems Journal**, [S.l.], v. 10, n. 2, p. 617–627, 2016.
- ELLIOTT, M.; COVENTRY, A. Critical care: the eight vital signs of patient monitoring. **British journal of nursing (Mark Allen Publishing)**, [S.l.], v. 21, n. 10, p. 621–5, 2012.
- FOTOUHI, H.; VAHABI, M.; BJIRKMAN, M. et al. Interoperability in heterogeneous low-power wireless networks for health monitoring systems. **Communications Workshops (ICC), 2016 IEEE International Conference on**, [S.l.], p. 393–398, 2016.

GONNOT, T.; YI, W.-J.; MONSEF, E.; SANIIE, J. Robust framework for 6LoWPAN-based body sensor network interfacing with smartphone. **2015 IEEE International Conference on Electro/Information Technology (EIT)**, [S.l.], p. 320–323, 2015.

GOSWAMI, S.; ROY, P.; DEY, N.; CHAKRABORTY, S. Wireless Body Area Networks Combined with Mobile Cloud Computing in Healthcare: a survey. In: **Classification and Clustering in Biomedical Signal Processing**. [S.l.]: IGI Global, 2016. p. 388–402.

GUPTA, N. **Inside Bluetooth low energy**. [S.l.]: Artech house, 2013.

HAN, T.; XIAO, X.; SHI, L.; CANNY, J.; WANG, J. Balancing accuracy and fun: designing camera based mobile games for implicit heart rate monitoring. **Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems**, [S.l.], p. 847–856, 2015.

HU, S. Q.; FU, Y.; SHE, C. D.; YAO, H. Enabling ZigBee communications in android devices. **Advanced Materials Research**, [S.l.], v. 756, p. 2125–2130, 2013.

HUI, J.; CULLER, D.; CHAKRABARTI, S. 6LoWPAN: incorporating ieee 802.15. 4 into the ip architecture. **IPSO Alliance White Paper**, [S.l.], v. 3, 2009.

IBGE, I. B. d. G. e. E. **Séries Estatísticas e Históricas**. Disponível em: <<http://seriesestatisticas.ibge.gov.br/series.aspx?no=13&op=0&vcodigo=MS44>>. Acesso em: 29.set.2015.

IEEE. **802.15.4 Low-Rate Wireless Personal Area Networks (LR-WPANs)**. [S.l.: s.n.], 2011. 314 p. v. 2011, n. September.

IEEE. **IEEE 802.15 WPAN Task Group 6 (TG6) Body Area Networks**. Disponível em: <<http://www.ieee802.org/15/pub/TG6.html>>. Acesso em: 21.ago.2016.

IETF. **RFC 7668 - IPv6 Packets over BLUETOOTH Low Energy**. Disponível em: <<https://tools.ietf.org/html/rfc7668>>. Acesso em: 20.out.2016.

IETF, I. E. T. F. **Transmission of IPv6 Packets over IEEE 802.15.4 Networks**. Disponível em: <<https://tools.ietf.org/html/rfc4944/>>. Acesso em: 24.set.2016.

IPV6.BR. **ZigBee usa agora 6LoWPAN! Sua próxima lâmpada terá IPv6?** Disponível em: <<http://ipv6.br/post/zigbee-usa-agora-6lowpan-sua-proxima-lampada-tera-ipv6/>>. Acesso em: 24.set.2016.

JIN, Z.; CHEN, Y. Telemedicine in the Cloud Era: prospects and challenges. **IEEE Pervasive Computing**, [S.l.], v. 14, n. 1, p. 54–61, 2015.

JUNG, E. Y.; KIM, J. H.; CHUNG, K. Y.; PARK, D. K. Home health gateway based healthcare services through U-health platform. **Wireless Personal Communications**, [S.l.], v. 73, n. 2, p. 207–218, 2013.

KARTHI, J.; RAO, S. V.; PILLAI, S. S. Performance analysis of wireless sensor networks with IEEE 802.15. 4 MAC based on variation in packet size. **2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICT)**, [S.l.], p. 46–50, 2015.

KO, J.; LU, C.; SRIVASTAVA, M. B.; STANKOVIC, J. A.; TERZIS, A.; WELSH, M. Wireless sensor networks for healthcare. **Proceedings of the IEEE**, [S.l.], v. 98, n. 11, p. 1947–1960, 2010.

LANE, N. D.; LIN, M.; MOHAMMOD, M.; YANG, X.; LU, H.; CARDONE, G.; ALI, S.; DORYAB, A.; BERKE, E.; CAMPBELL, A. T. et al. Bewell: sensing sleep, physical activities and social interactions to promote wellbeing. **Mobile Networks and Applications**, [S.l.], v. 19, n. 3, p. 345–359, 2014.

LANE, N. D.; MOHAMMOD, M.; LIN, M.; YANG, X.; LU, H.; ALI, S.; DORYAB, A.; BERKE, E.; CHOUDHURY, T.; CAMPBELL, A. Bewell: a smartphone application to monitor, model and promote wellbeing. **5th international ICST conference on pervasive computing technologies for healthcare**, [S.l.], p. 23–26, 2011.

LEE, Y.-G.; JEONG, W. S.; YOON, G. Smartphone-Based Mobile Health Monitoring. **Telemedicine and e-Health**, [S.l.], v. 18, n. 8, p. 585–590, 2012.

MAIA, P.; BAFFA, A.; CAVALCANTE, E.; DELICATO, F. C.; BATISTA, T.; PIRES, P. F. A Middleware Platform for Integrating Devices and Developing Applications in e-Health. **Computer Networks and Distributed Systems (SBRC), 2015 XXXIII Brazilian Symposium on**, [S.l.], p. 10–18, 2015.

MAIA, P.; BATISTA, T.; CAVALCANTE, E.; BAFFA, A.; DELICATO, F. C.; PIRES, P. F.; ZOMAYA, A. A web platform for interconnecting body sensors and improving health care. **Procedia Computer Science**, [S.l.], v. 40, p. 135–142, 2014.

MERRELL, R. C.; DOARN, C. R. m-Health. **Telemedicine and e-Health**, [S.l.], v. 20, n. 2, p. 99–101, 2014.

MOSER, L. E.; MELLIAR-SMITH, P. Personal Health Monitoring Using a Smartphone. **2015 IEEE International Conference on Mobile Services**, [S.l.], p. 344–351, 2015.

MOTEIV, C. **Tmote-sky Datasheet**. Disponível em: <<http://www.eecs.harvard.edu/konrad/projects/shimmer/references/tmote-sky-datasheet.pdf>>. Acesso em: 27.jun.2017.

NAIR, K.; KULKARNI, J.; WARDE, M.; DAVE, Z.; RAWALGAONKAR, V.; GORE, G.; JOSHI, J. Optimizing power consumption in iot based wireless sensor networks using Bluetooth Low Energy. **Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on**, [S.l.], p. 589–593, 2015.

PESSOA, L. **Bluetooth Smart**. Disponível em: <<http://www.embarcados.com.br/serie/bluetooth-smart/>>. Acesso em: 15.out.2016.

REN, Y.; PAZZI, R.; BOUKERCHE, a. Monitoring patients via a secure and mobile healthcare system. **Wireless Communications, IEEE**, [S.l.], v. 17, n. February, p. 59–65, 2010.

Riazul Islam, S. M.; Humaun Kabir, M.; HOSSAIN, M. The Internet of Things for Health Care: a comprehensive survey. **IEEE Access**, [S.l.], v. 3, p. 678–708, 2015.

SAMSUNG. **Samsung Galaxy S6 Full Specifications**. Disponível em: <<http://www.samsung.com/in/consumer/mobile-devices/smartphones/galaxy-s/SM-G920IZDAINS>>. Acesso em: 15.ago.2016.

SEEGER, C.; VAN LAERHOVEN, K.; BUCHMANN, A. MyHealthAssistant: an event-driven middleware for multiple medical applications on a smartphone-mediated body sensor network. **IEEE journal of biomedical and health informatics**, [S.l.], v. 19, n. 2, p. 752–760, 2015.

SHAHAMABADI, M. S.; ALI, B. B. M.; VARAHRAM, P.; JARA, A. J. A network mobility solution based on 6LoWPAN hospital wireless sensor network (NEMO-HWSN). **Proceedings - 7th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2013**, [S.l.], p. 433–438, 2013.

SHANKAR, S. K.; TOMAR, A. S. A survey on wireless body area network and electronic-healthcare. **Recent Trends in Electronics, Information & Communication Technology (RTEICT), IEEE International Conference on**, [S.l.], p. 598–603, 2016.

SIEKKINEN, M.; HIIENKARI, M.; NURMINEN, J. K.; NIEMINEN, J. How low energy is bluetooth low energy? Comparative measurements with ZigBee/802.15.4. **2012 IEEE Wireless Communications and Networking Conference Workshops, WCNCW 2012**, [S.l.], p. 232–237, 2012.

SIG, B. S. I. G. **Bluetooth Low Energy**. Disponível em: <<https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics/low-energy>>. Acesso em: 15.set.2016.

SIG, B. S. I. G. **Bluetooth 5**. Disponível em: <<https://www.bluetooth.com/specifications/bluetooth-core-specification/bluetooth5>>. Acesso em: 10.fev.2017.

SIG, B. S. I. G. **Bluetooth 5 quadruples range, doubles speed, increases data broadcasting capacity by 800%**. Disponível em: <<https://www.bluetooth.com/news/pressreleases/2016/06/16/-bluetooth5-quadruples-rangedoubles-speedincreases-data-broadcasting-capacity-by-800>>. Acesso em: 19.set.2016.

SIG, B. S. I. G. **Bluetooth Core Specification**. Disponível em: <<https://www.bluetooth.com/specifications/bluetooth-core-specification>>. Acesso em: 25.out.2016.

SIG, B. S. I. G. **Specification of the Bluetooth System v4.2**. Disponível em: <<https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx>>. Acesso em: 25.out.2016.

SIG, B. S. I. G. **Bluetooth Quick Reference Guide**. Disponível em: <<https://www.bluetooth.com/media/files/specification/bluetooth-quick-reference-guide.ashx?la=en>>. Acesso em: 15.out.2016.

SIG, B. S. I. G. **GATT Services**. Disponível em: <<https://www.bluetooth.com/specifications/gatt/services>>. Acesso em: 10.jun.2017.

SQLITE. **SQLite Documentation**. Disponível em: <<https://sqlite.org/docs.html>>. Acesso em: 20.out.2016.

TANENBAUM, A. S.; VAN STEEN, M. **Distributed systems: principles and paradigms**. [S.l.]: Prentice-Hall, 2007.

TANENBAUM, A. S.; WETHERALL, D. J. **Computer Networks**. [S.l.]: Prentice Hall, Cloth, 2011.

ULLAH, S.; HIGGINS, H.; BRAEM, B.; LATRE, B.; BLONDIA, C.; MOERMAN, I.; SALEEM, S.; RAHMAN, Z.; KWAK, K. S. A comprehensive survey of wireless body area networks on PHY, MAC, and network layers solutions. **Journal of Medical Systems**, [S.l.], v. 36, n. 3, p. 1065–1094, 2012.

WANG, K. I. K.; DUBEY, S.; RAJAMOCHAN, A.; SALCIC, Z. An Android-Based Mobile 6LoWPAN Network Architecture for Pervasive Healthcare. **Proceedings - 2015 International Conference on Intelligent Environments, IE 2015**, [S.l.], p. 49–56, 2015.

WU, P. Y.; CHENG, C. W.; KADDI, C. D.; VENUGOPALAN, J.; HOFFMAN, R.; WANG, M. D. Omic and Electronic Health Record Big Data Analytics for Precision Medicine. **IEEE Transactions on Biomedical Engineering**, [S.l.], v. 64, n. 2, p. 263–273, Feb 2017.