|  |  |  |
| --- | --- | --- |
|  | Universidade Federal de PernambucoCentro de Informática | ..\..\..\..\tg\figuras\cin.gif |

Trabalho de Graduação em Engenharia da Computação

**Um estudo sobre pilhas Bluetooth e** **suas limitações em sistemas embarcados**

**Autor:** Hugo Rodrigues de Albuquerque (hra@cin.ufpe.br)

**Orientador:** Sérgio Vanderlei Cavalcante (svc@cin.ufpe.br)

**Co-orientador:** Paulo Urbano (paulo.urbano@cesar.org.br)

Recife, 21 dezembro de 2010

**Folha de Aprovação**

Hugo Rodrigues de Albuquerque

**Um estudo sobre pilhas Bluetooth e suas limitações em sistemas embarcados**

Banca avaliadora

Prof. Sérgio Vanderlei Cavalcante (svc@cin.ufpe.br) – UFPE

(Orientador)

Prof. Abel Guilhermino da Silva Filho (agsf@cin.ufpe.br) – UFPE

(Avaliador)

**Resumo**

Sistemas computacionais são cada vez mais presentes no nosso dia-a-dia e a troca de informações entre eles é fundamental. Um caso particular são os sistemas embarcados, que são embutidos em outros sistemas maiores e provêem serviços específicos. Em razão da necessidade de mobilidade, redes de comunicação sem fio mostram-se muito atraentes, sendo o Bluetooth um padrão bastante utilizado atualmente.

O Bluetooth é uma tecnologia baseada na transmissão de dados através de ondas de rádio e tem certa imunidade a interferências de outras tecnologias que utilizam a mesma faixa de frequência. O conjunto (pilha) de protocolos que define o funcionamento tem sua estrutura dividida em camadas e provê interoperabilidade entre os dispositivos da rede. A especificação estabelece ainda perfis de utilização que agrupam protocolos de várias camadas de forma a definir quais serão necessários para o desenvolvimento de determinado tipo de aplicação. Devido às restrições de recursos dos sistemas embarcados, existem certas limitações quanto ao desenvolvimento de pilhas Bluetooth voltadas para este nicho. Elas envolvem questões como quantidade de memória disponível, consumo de energia, dificuldades de implementação, eficiência na comunicação e taxa de transferência de dados.

**Agradecimentos**

Agradeço primeiramente a Deus e à minha família, que sempre foi base para a minha formação. À minha mãe, Maria do Carmo Rodrigues de Albuquerque, que sempre me apoiou em todas as minhas decisões, dando suporte psicológico e de conhecimento necessários para que eu pudesse seguir em frente sem me deixar abater pelas dificuldades encontradas. Ao meu pai, Evaristo Paulino de Albuquerque, por ajudar na minha criação, que permitiu saber discernir entre o certo e o errado, e também por seu lado perfeccionista, que me fez ter sempre o pensamento de fazer tudo da melhor maneira possível, de preferência nas primeiras tentativas. Ao meu irmão, Marcelo, pelo apoio, consideração e até pelas discussões que já tivemos ao longo de todo esse tempo. Aos meus avós, tios, tias, primos, primas e agregados por todos os bons momentos que compartilhamos.

Agradeço à minha namorada, Talita, pelos mais de 4 anos que estamos juntos e todos os momentos de extrema felicidade que vivemos. Por ser minha amiga, companheira, confidente e estar sempre ao meu lado dando apoio, carinho e amor, fundamentais para o meu bom estado de espírito. Também agradeço à sua família pelos bons momentos de confraternizações, conversas e horas de lazer.

Agradeço ao meu orientador e professor Sérgio Cavalcante e ao meu coorientador Paulo Urbano, por toda a ajuda essencial que obtive para a realização deste trabalho.

Agradeço a todos os meus amigos, que são inúmeros e de diversos lugares, incluindo os vários colégios e cursos por que passei (Colégio Boa Viagem, Escola Madre de Deus, CPI, Colégio Militar, GGE, preparatório para o ITA), vizinhos, amigos do meu irmão, da minha namorada e amigos de amigos, que passei a incluir também na minha lista. Aos meus colegas e amigos do Centro de Informática que dividiram comigo esses cinco anos de preocupações, vitórias, nervosismo, dias e noites de estudos e projetos, felicidades e por que não, momentos de lazer também. Em especial aos meus amigos mais próximos, que me proporcionaram muitos momentos de lazer, descontração, viagens e conversas.

Enfim, agradeço e dedico este trabalho a todos que em algum momento se fizeram importantes na minha vida e me permitiram chegar até aqui. Obrigado!

Índice

[**Lista de Acrônimos 7**](#_Toc280539259)

[**1. Introdução 9**](#_Toc280539260)

[**2. Conceitos Básicos 11**](#_Toc280539261)

[2.1. Redes de comunicação de dados 11](#_Toc280539262)

[2.1.1. Wi-Fi (IEEE 802.11) 12](#_Toc280539263)

[2.1.2. ZigBee (IEEE 802.15.4) 13](#_Toc280539264)

[2.1.3. Infravermelho (*Infrared Data Association* - IrDA) 14](#_Toc280539265)

[2.1.4. Bluetooth 15](#_Toc280539266)

[2.2. Sistemas Embarcados 17](#_Toc280539267)

[**3. Bluetooth 19**](#_Toc280539268)

[3.1. Arquitetura 19](#_Toc280539269)

[3.2. Máquina de Estados 21](#_Toc280539270)

[3.3. Pilha de Protocolos Bluetooth 24](#_Toc280539271)

[3.3.1. Camada de Rádio Bluetooth 26](#_Toc280539272)

[3.3.2. Camada *Baseband* 26](#_Toc280539273)

[3.3.3. Camada de Gerenciamento de Link (*Link Manager*) 27](#_Toc280539274)

[3.3.4. Camada de Interface Host-Controlador (HCI) 28](#_Toc280539275)

[3.3.5. Camada de Controle de Link Lógico e Adaptação (L2CAP) 29](#_Toc280539276)

[3.3.6. Protocolo de Descobrimento de Serviços (SDP) 29](#_Toc280539277)

[3.3.7. Protocolo de Comunicação de Radiofrequência (RFCOMM) 29](#_Toc280539278)

[3.3.8. Protocolos PPP, UDP/TCP/IP 30](#_Toc280539279)

[3.3.9. Protocolo de Troca de Objetos (OBEX) 30](#_Toc280539280)

[3.4. Perfis Bluetooth 31](#_Toc280539281)

[3.5. Aplicações 33](#_Toc280539282)

[**4. Implementações e Análises 35**](#_Toc280539283)

[4.1. Pilhas Bluetooth para Sistemas Embarcados 35](#_Toc280539284)

[4.1.1. lwBT 35](#_Toc280539285)

[4.1.2. BTnode 35](#_Toc280539286)

[4.1.3. IAR Embedded Bluetooth Protocol Stack 36](#_Toc280539287)

[4.1.4. BeeCon MicroBlue / AmbiCompTM Bluetooth Sandwich Module 38](#_Toc280539288)

[4.1.5. AVE-BlueTM 38](#_Toc280539289)

[4.1.6. IVT Bluetooth Host Stack (BlueletTM) 39](#_Toc280539290)

[4.1.7. BluRapport© (antigo BL-Rappore) 39](#_Toc280539291)

[4.1.8. Bluetopia® 40](#_Toc280539292)

[4.2. Avaliações 41](#_Toc280539293)

[4.3. Sugestões de melhorias 44](#_Toc280539294)

[**5. Conclusões e trabalhos futuros 46**](#_Toc280539295)

[**Referências 48**](#_Toc280539296)

# Lista de Acrônimos

A2DP – *Advanced Audio Distribution Profile*

AVRCP – *Audio Video Remote Control Profile*

ACL – *Asynchronous Connectionless*

AES – *Advanced Encryptation Standard*

API – *Application Programming Interface*

BIP – *Basic Imaging Profile*

BPP – *Basic Printing Profile*

CTP – *Cordless Telephony Profile*

CRC – *Cyclic Redundancy Check*

DUN – *Dial-up Networking Profile*

EAP – *Extensible Authentication Protocol*

EDR – *Enhanced Data Rate*

ETSI - European Telecommunications Standards Institute

FAX – *Fax Profile*

FEC – *Forward Error-Correcting*

FHSS – *Frequency Hopping Spread Spectrum*

FTP – *File Transfer Profile*

GAP – *Generic Access Profile*

GAVDP – *Generic Audio/Video Distribution Profile*

GFSK – *Gaussian Frequency Shift Keying*

GOEP – *Generic Object Exchange Profile*

HCI – *Host Command Interface*

HCRP – *Hardcopy Cable Replacement Profile*

HFP – *Hands-Free Profile*

HID – *Human Interface Device Profile*

HS – *High Speed*

HSP – *Headset Profile*

IEEE – *Institute of Electrical and Eletronic Engineers*

IETF – *Internet Engineering Task Force*

IP – *Internet Protocol* ou *Intercom Profile*

IrDA – *Infrared Data Association*

ISM – *Industrial, Scientific and Medical*

kbps – Quilobit por segundo

L2CAP – *Logical Link Control and Adaptation Protocol*

LAN – *Local Area Network*

LAP – *LAN Access Profile*

LMP – *Link Manager Protocol*

LR-WPAN – *Low-Rate Wireless Personal Area Network*

MAN – *Metropolitan Area Network*

Mbps – Megabit por segundo

OBEX – *Object Exchange Protocol*

OPP – *Object Push Profile*

PAN – *Personal Area Network* ou *Personal Area Network Profile*

PC – *Personal Computer*

PPP – *Point-to-Point Protocol*

QoS – *Quality of Service*

RF – Radiofrequência

RFCOMM – *Radio Frequency Communication Protocol*

SAP – *SIM Access Profile*

SCO – *Synchronous Connection-Oriented*

SDAP – *Service Discovering Application Profile*

SDP – *Service Discovering Protocol*

SPP – *Serial Port Profile*

SYNC – *Synchronization Profile*

TCP – *Transport Control Protocol*

TCS – *Telephony Control Signaling*

TDD – *Time Division Duplexing*

TDMA – *Time Division Multiple Access*

TKIP – *Temporal Key Integrity Protocol*

UART – *Universal Asynchronous Receiver/Transmitter*

UDP – *User Datagram Protocol*

USB – *Universal Serial Bus*

WAE – *Wireless Application Environment*

WAN – *Wide Area Network*

WAP – *Wireless Application Protocol*

WEP – *Wired Equivalent Privacy*

WLAN – *Wireless Local Area Network*

WPA – *Wi-Fi Protected Access*

WPAN – *Wireless Personal Area Network*

1. **Introdução**

Com o avanço contínuo da tecnologia, os sistemas computacionais encontram-se cada vez mais presentes nas mais diversas tarefas cotidianas da sociedade, várias vezes até de maneira imperceptível, permitindo a introdução de funcionalidades inteligentes e o controle e gerenciamento de sistemas.

Sejam computadores pessoais, aparelhos eletrodomésticos, sistemas de segurança, telefonia ou controle, são inúmeros os equipamentos e dispositivos que abrigam um componente que permite processar computacionalmente as informações recebidas e armazenadas. Embora seja muito mais comum relacionar a computação com computadores pessoais, os sistemas embarcados, que são sistemas computacionais embutidos em outros sistemas maiores e geralmente executam tarefas específicas, vêm ganhando um grande destaque nesse universo, sendo utilizados tanto em tarefas mais simples e corriqueiras, como também em outras bem mais complexas e sofisticadas.

É claro que a computação desempenha um papel fundamental nos dispositivos computacionais, mas diversas vezes a comunicação entre eles também representa um papel essencial, pois permite que eles compartilhem informações necessárias ao funcionamento do sistema como um todo. Muitos desses dispositivos encontram-se interligados e necessitam comunicar-se de alguma forma. É possível realizar esta comunicação das mais variadas formas, através de fios, cabos, ondas eletromagnéticas, ondas sonoras, satélites, microondas, radiação, entre diversos outros meios disponíveis.

Uma forma de comunicação que tem chamado bastante atenção ultimamente, devido à ausência da necessidade de um meio físico para propagação dos dados, é a utilização de tecnologias sem fio (wireless), que utilizam ondas eletromagnéticas (ondas de rádio, por exemplo) para a transmissão de sinais. Entre outras, podemos citar o Wi-Fi, o infravermelho, o ZigBee e o Bluetooth.

Na tentativa de adicionar um elemento de comunicação aos dispositivos, dependendo dos requisitos e características do sistema em questão, algumas tecnologias se mostram mais adequadas que outras. Nesse caso é interessante a existência de análises técnicas que abordem as configurações e os resultados da tecnologia utilizada de modo a obter informações acerca das limitações e dificuldades que podem existir durante a implementação da solução. É também através dessas análises que são apresentadas sugestões de melhorias que facilitem o processo de desenvolvimento.

Em particular, a maioria dos sistemas embarcados possui restrições de recursos que atrapalham o processo de desenvolvimento. A implementação de tecnologias de comunicação (no caso deste estudo, o Bluetooth) nesses sistemas pode apresentar certas dificuldades, uma vez que podem ser necessárias certas modificações ou ajustes para que as restrições sejam contornadas.

O objetivo deste trabalho é estudar a fundo os conceitos e características do conjunto (também conhecido como *pilha*) de protocolos Bluetooth, como meio de formar um conhecimento básico que permita realizar análises sobre as limitações e dificuldades de sua implementação em sistemas embarcados, além da sugestão de melhorias a serem realizadas, de forma a evitar ou amenizar os problemas encontrados.

Este relatório contém outros cinco capítulos, organizados como é mostrado a seguir. No capítulo 2 são apresentados vários conceitos básicos sobre os assuntos abordados, como redes de comunicação, o padrão Bluetooth e sistemas embarcados. No capítulo 3 o Bluetooth é descrito de forma mais aprofundada, onde são especificadas as características e a pilha de protocolos Bluetooth, dividida em camadas. O capítulo 4 é dedicado a apresentar algumas implementações da pilha Bluetooth voltadas para sistemas embarcados, assim como as análises e características de cada uma delas. Ainda no mesmo capítulo são discutidas sugestões de melhorias em relação às implementações anteriormente mostradas.

1. **Conceitos Básicos**

Neste capítulo, iremos introduzir alguns conceitos básicos que serão necessários para o entendimento do trabalho. Serão abordadas algumas redes de comunicação de dados, em especial as redes sem fio (wireless), incluindo a tecnologia Bluetooth. Em seguida, são passadas algumas informações gerais sobre sistemas embarcados.

* 1. **Redes de comunicação de dados**

Com o uso cada vez maior dos computadores e sistemas computacionais integrados aos vários itens utilizados no nosso dia-a-dia, um dos recursos mais fundamentais nesse contexto é a comunicação entre dispositivos. Além de processar as informações, os dispositivos precisam conectar-se para trocar informações de controle e dados, de forma que possam operar em conjunto de forma natural.

De acordo com a extensão geográfica, as redes de computadores podem ser classificadas em PANs (*Personal Area Networks*), LANs (*Local Area Networks*), MANs (*Metropolitan Area Networks*) e WANs (*Wide Area Networks*) [1]. As PANs são redes pequenas que visam à comunicação de dispositivos em volta de uma pessoa, como por exemplo, um conjunto de periféricos ou dispositivos móveis conectados a um PC. As LANs são redes locais que interligam vários computadores localizados fisicamente próximos, a fim de realizar compartilhamento de arquivos, da Internet, por exemplo. Nas MANs, computadores que se encontram em localizações mais distantes, como por exemplo, em bairros diferentes de uma cidade, podem se conectar de forma a compartilhar dados. Por fim, existem as WANs, que são redes de grande extensão e visam conectar grandes áreas geográficas, como países e até continentes, fornecendo serviços de voz e de dados, por exemplo.

Com relação à tecnologia utilizada, os dados podem ser transmitidos com fio ou sem fio. Na primeira opção, um dos maiores problemas detectados é a obrigatoriedade da utilização de cabos, que dificulta a conexão e torna o ambiente “poluído”. Dentre as mais conhecidas estão as conexões de modem (via linha telefônica) e o padrão Ethernet (IEEE 802.3), amplamente utilizado em redes domésticas e comerciais para o compartilhamento de arquivos e distribuição de Internet. Já nas redes sem fio, a flexibilidade e mobilidade da conexão são imensamente superiores. No decorrer desta seção vamos apresentar conceitos relativos a algumas redes de comunicação sem fio.

* + 1. **Wi-Fi (IEEE 802.11)**

O Wi-Fi (abreviação do termo *Wireless Fidelity* – fidelidade sem fio) é uma marca utilizada por produtos pertencentes à classe de redes sem fio WLAN (*Wireless Local Area Network*) que utilizam a tecnologia 802.11 definida pela IEEE. Foi criado pela Wi-Fi Alliance, uma organização sem fins lucrativos formada por centenas de grandes, médias e pequenas empresas, com o objetivo de estabelecer uma especificação internacional de conectividade sem fio entre computadores, celulares, consoles de vídeo-game, impressoras, tocadores de música e vídeo, câmeras digitais, entre outros dispositivos eletrônicos.

Entre os mais diversos usos de uma rede Wi-Fi, podemos citar a conexão dos aparelhos eletrônicos uns aos outros, à Internet ou ainda a outras redes previamente estabelecidas que façam uso de outras tecnologias. Assim, é possível compartilhar facilmente dispositivos e recursos de mídia, como impressoras, arquivos de áudio/vídeo e acesso à Internet.

O Wi-Fi opera utilizando ondas de rádio (de modo similar aos telefones celulares, TVs, rádios e sistemas de navegação de GPS) para transmissão de dados nas faixas não-licenciadas de frequência de 2,4 GHz e 5 GHz (às vezes simultaneamente). O alcance de acesso entre dois dispositivos Wi-Fi pode ir de algumas dezenas a centenas de metros, dependendo da antena, da tecnologia empregada e dos obstáculos físicos e interferências eletromagnéticas entre eles. Devido ao consórcio formado pelas empresas, a certificação Wi-Fi garante a interoperabilidade entre os dispositivos, isto é, que aparelhos de marcas e modelos diferentes irão trabalhar em conjunto corretamente.

A especificação ainda encontra-se em desenvolvimento e possui várias versões: 802.11a, 802.11b, 802.11g e 802.11n. Cada certificação é definida por um conjunto de funcionalidades relacionadas ao desempenho, faixa de frequência e largura de banda (“velocidade”). Cada avanço de geração também promove melhorias na segurança, como o uso de padrões de segurança (WEP, WPA, WPA2, TKIP, AES, etc.) e de autenticação EAP, no alcance e em outras funcionalidades que os fabricantes desejem adicionar. A seguir, na *Tabela 1*, encontra-se uma tabela com algumas características estabelecidas para cada geração.

|  |  |  |
| --- | --- | --- |
| Tecnologia | Faixa de frequência | Largura de banda (taxa máxima de dados) |
| 802.11a | 5 GHz | 54 Mbps |
| 802.11b | 2.4 GHz | 11 Mbps |
| 802.11g | 2.4 GHz | 54 Mbps |
| 802.11n | 2.4 GHz / 5 GHz (uso alternável ou simultâneo) | 450 Mbps |

Tabela 1. Características das gerações do Wi-Fi [2]

* + 1. **ZigBee (IEEE 802.15.4)**

Uma tecnologia também interessante é o ZigBee, desenvolvido pela ZigBee Alliance, outra associação de mais de 300 empresas com o objetivo de estabelecer uma especificação padrão para redes sem fio. Porém, diferentemente do Wi-Fi, que provê um conjunto de protocolos que garante a qualidade para transmissão de voz ou de dados com altas taxas de transferência, o ZigBee é voltado para as aplicações em redes locais que usam sensores e outros dispositivos de controle. Em detrimento das altas taxas de transferência de dados, essas redes priorizam a simplicidade e o baixo custo (para baratear os gastos com aquisição, instalação, manutenção e mão-de-obra dos equipamentos), a segurança (para assegurar que as informações do sistema não serão acessadas nem modificadas indevidamente), o baixo tempo de resposta (visando garantir que as informações recebidas são atuais e, portanto, mais precisas) e o baixo consumo de energia (de forma a aumentar a vida útil das baterias utilizadas nos equipamentos). A tecnologia suporta tráfegos de dados periódicos (sensores), intermitentes (interruptores e chaves) e provenientes de dispositivos repetitivos de baixa frequência de atualização (mouses, teclados, por exemplo) [5].

Os dispositivos ZigBee operam na faixa ISM (*Industrial, Scientific and Medical*), que de acordo com a localidade corresponde às frequências 868 MHz (Europa), 915 MHz (EUA e Austrália) e 2,4 GHz (Global), com baixas taxas de transmissão de dados (250 kbps, 40 kbps e 20 kbps, respectivamente). Baseado num padrão de redes de baixo consumo (LR-WPAN – Low-Rate Wireless Personal Area Network, padronizado pelo IEEE como 802.15.4), os dispositivos podem ser classificados em dois tipos:

* RFD (*Reduced-Function Device* – Dispositivos de Função Reduzida) – são apenas nós terminais (sensores, atuadores, etc.), sem funções de roteamento. Geralmente são dispositivos móveis e alimentados por bateria, com um modo de baixo consumo de energia (sleep);
* FFD (*Full-Function Device* – Dispositivos de Função Completa) – além de desempenharem as funções básicas, como sensoriamento, podem agir como roteadores ou coordenadores ZigBee, ajudando a formar a rede. De modo geral, são dispositivos fixos, alimentados pela rede elétrica e não entram em modo de baixo consumo de energia.

Um conceito muito importante na arquitetura de uma rede ZigBee é a possibilidade de ela ter uma estrutura em forma de malha, onde os nós (dispositivos) encontram-se fortemente interconectados, visando aumentar a confiabilidade e o alcance relativo de cada dispositivo. Como a rede torna-se bastante descentralizada, devido à falta de um dispositivo mestre que controle os outros, caso ocorra uma falha individual em um dispositivo a rede pode recuperar-se mais facilmente ao proporcionar rotas alternativas, aumentando assim a estabilidade [6]. Além disso, mesmo que um dispositivo não tenha uma “visão” restrita, ele pode utilizar suas conexões para “enxergar” outros nós da rede que a princípio são inalcançáveis. Essa estrutura proporciona ainda uma alta escalabilidade, isto é, permite que novos dispositivos sejam adicionados sem a necessidade de grandes modificações.

A ZigBee Alliance apresenta os seguintes perfis de soluções disponíveis através do uso da tecnologia, alguns deles ainda em desenvolvimento [4]: gerenciamento de energia refinado (ZigBee Smart EnergyTM); controles remotos mais rápidos e confiáveis através do uso da radiofrequência (ZigBee Remote ControlTM); automação de residências, provendo controle de iluminação, climatização, segurança, gerenciamento de multimídia e de energia, sensoriamento de água, fogo e fumaça, aparelhos inteligentes e controle de acesso (ZigBee Home AutomationTM); segurança, monitoramento confiável e gerenciamento de serviços de saúde não críticos (ZigBee Health CareTM); automação para empresas, semelhante à automação residencial, provendo controle, conservação, flexibilidade e segurança (ZigBee Building AutomationTM); distribuição de informações, jogos móveis, serviços baseados em localização, pagamentos móveis seguros, serviços de compartilhamento peer-to-peer (P2P) e publicidade móvel (ZigBee Telecommunication ServicesTM); funcionalidades para serviços de varejo, como informações de loja, serviços baseados em localização, vendas a domicílio, gerenciamento da cadeia de suprimentos, gerenciamento de energia (ZigBee Retail ServicesTM).

* + 1. **Infravermelho (*Infrared Data Association* - IrDA)**

Outro meio de transmissão de dados sem fio bastante conhecido e difundido através do seu uso em controles remotos, são as ondas de luz infravermelha (radiação com frequência abaixo da luz visível, que não é percebida pelo olho humano). O funcionamento básico ocorre através da codificação das informações digitais em forma de sinais elétricos, traduzidos em forma de luz infravermelha, e enviados ao receptor. Do outro lado, um aparelho possui componentes que recebem essas ondas e convertem-nas de volta no padrão digital original da informação.

Porém, para que dois dispositivos possam realmente se comunicar e transmitir dados de maneira completa é necessário estabelecer uma ligação bi-direcional de transferência de informações, que é na verdade bem mais complexa do que o procedimento explicado anteriormente [7]. Para especificar como deve ocorrer essa transferência bi-direcional e garantir a interoperabilidade entre os dispositivos, foi criada a IrDA, uma associação de centenas de empresas de computação e telecomunicação, que busca definir o padrão de mesmo nome. Através das especificações IrDA-Data e IrDA-Control, esse padrão define as características físicas da interface, os protocolos de comunicação que provêem as diferentes funcionalidades e a velocidade de transmissão com a qual os dispositivos se comunicam [7]. Os padrões não são proprietários e estão relativamente livres de regulação, uma vez que as frequências infravermelhas localizam-se abaixo da luz visível na escala eletromagnética e não há restrições de uso.

Devido à natureza do meio de transmissão, os dispositivos necessitam de linha de visada para estabelecer uma conexão, pois o dispositivo transmite através de uma área de cone com ângulo de apenas 30°, num raio de curta distância de no máximo alguns metros, e o sinal não é capaz de atravessar paredes e objetos opacos. Isso pode ser um empecilho para a comunicação, mas também adiciona um nível maior de segurança, pois o usuário emissor necessita apontar o dispositivo para o receptor das informações, evitando assim escutas indesejadas. Algumas outras vantagens são que os dispositivos são baratos e consomem pouca energia. Além disso, essa tecnologia não sofre interferência de ondas de rádio e, devido à alta velocidade da luz (300.000 km/s), as taxas de transmissão de dados também são altas, chegando à ordem de alguns Mbps. Podemos encontrar essa tecnologia empregada em celulares, dispositivos portáteis, impressoras, entre diversos outros dispositivos eletrônicos, para fins de sincronização, controle remoto ou sistemas de pagamento.

* + 1. **Bluetooth**

O Bluetooth é uma especificação aberta, desenvolvida pelo Bluetooth SIG (*Special Interest Group*, uma associação com milhares de companhias presentes atualmente) e padronizada pelo IEEE como o padrão 802.15.1. Seu objetivo é estabelecer um padrão único para o desenvolvimento de WPANs através de ondas de rádio, provendo uma infraestrutura de rede capaz de possibilitar a comunicação sem fio de curta distância de dados e voz. Inicialmente projetado para substituir os cabos que conectam os dispositivos eletrônicos, o Bluetooth compreende tanto componentes de software como de hardware, tendo como objetivos principais a robustez, o baixo consumo de energia (potência) e o baixo custo. O padrão utiliza um dispositivo com baixa potência de transmissão (100 mW), o que garante o baixo consumo de energia e restringe a comunicação a pequenas distâncias (10 a 100 m).

Hoje se pode encontrar essa tecnologia em uma grande diversidade de dispositivos, como dispositivos inteligentes (PDAs, celulares, PCs), periféricos de dados (mouses, teclados, joysticks, câmeras, canetas digitais, impressoras, pontos de acesso LAN), periféricos de áudio (fones de ouvido, alto-falantes, receptores estéreo), e aplicações embarcadas (sistemas de partida automotivos, atualizações de mercearias, sistemas industriais, instrumentos musicais MIDI).

Uma rede Bluetooth consiste na formação de pequenos grupos (clusters) de até oito dispositivos, chamados de *piconets*, onde um deles, o mestre (*master*), coordena a rede e os outros, os escravos (*slaves*), apenas respondem às requisições do primeiro. Existe ainda a possibilidade de que um dispositivo esteja conectado a mais de uma piconet simultaneamente, contanto que ele não seja mestre em ambas, de forma a juntar piconets e constituir redes maiores, chamadas *scatternets*. Devido a essa característica da arquitetura, um dispositivo pode conectar-se a outro que não esteja ao seu alcance, através de outros dispositivos ou redes intermediárias. A *Figura 1* mostra um exemplo de uma scatternet formada por três piconets, onde alguns dos nós podem ser mestre e escravo simultaneamente, porém em redes diferentes, ou escravo em mais de uma piconet.

**Piconet 3**

Escravo em apenas uma piconet

Mestre

**Piconet 1**

**Piconet 2**

Mestre e escravo (ponte)

Escravo em mais de uma piconet (ponte)

Figura 1. Exemplo de uma scatternet formada por três piconets

Devido à escolha da faixa não-licenciada ISM do espectro de frequência de 2,4 GHz para a transmissão dos dados, o tráfego na rede Bluetooth fica muito suscetível a interferências de outras tecnologias que utilizam a mesma banda. Um conceito bastante importante utilizado no Bluetooth é o FHSS (*Frequency Hopping Spread Spectrum*), que define canais de transmissão e possibilita ao dispositivo mestre estabelecer saltos de frequência a cada período previamente determinado. Como esse período é bastante pequeno, mesmo que outro dispositivo que esteja transmitindo na mesma faixa de frequência interfira no sinal, os danos causados não serão de grande porte, uma vez que em breve o canal será trocado e a interferência deixará de existir. Esse conceito de salto de frequência também é interessante do ponto de vista de segurança, pois como apenas quem está conectado ao mestre pode conhecer a sequência de saltos a ser realizada, dispositivos localizados fora da rede que queiram obter os dados trafegados não irão conseguir obter uma sucessão correta dos mesmos.

* 1. **Sistemas Embarcados**

Sistemas embarcados são sistemas computacionais encontrados dentro de sistemas maiores e, em contraste com os computadores pessoais (conhecidos como PCs), que executam tarefas de caráter genérico, geralmente realizam tarefas específicas, sendo por isso também conhecidos como sistemas especialistas. Por exemplo, um forno microondas possui um sistema embarcado que controla o tempo e a potência a serem utilizados para cada programa de aquecimento ou descongelamento especificado. Segundo alguns dados estimados por pesquisas de alta tecnologia, mais de 90% dos microprocessadores fabricados mundialmente são destinados a máquinas que usualmente não são chamadas de computadores [14].

Atualmente, os sistemas embarcados estão presentes em grande quantidade e diversidade no nosso dia-a-dia. Exemplos de dispositivos embarcados podem ser encontrados em verticais de indústria tão diversas como a automobilística, petróleo & gás e telecomunicações, em especial nos telefones celulares. Dispositivos de integração dos sistemas de entretenimento e controle do veículo, instrumentação de maquinaria industrial e acessórios para celulares podem ser identificados como exemplos de sistemas embarcados em cada uma das verticais mencionadas anteriormente.

Uma característica marcante nos sistemas embarcados é a interação constante mantida com o meio a sua volta, seja ela feita de forma automática ou por meio de interfaces. Para isso eles podem apresentar sensores, que captam do ambiente informações importantes ao funcionamento do sistema, e atuadores, que modificam as condições de forma a obter uma resposta ou efeito desejado. Num sistema autônomo, pode ser fundamental analisar as condições ao seu redor para realizar as decisões adequadas à situação atual e evitar problemas de funcionamento ou a inferência de informações incorretas.

Os sistemas maiores que abrigam os sistemas embarcados podem possuir algumas restrições de recursos, como memória, processamento, energia, entre outros. Assim, os sistemas embarcados terminam por adquirir essas limitações. Por exemplo, o sistema embarcado de um celular deve controlar o uso de energia, de modo que ele não gaste rapidamente a bateria, ou de memória, permitindo que ainda exista uma sobra suficiente para o uso natural dos outros componentes do aparelho. De maneira geral, algumas características como tamanho, peso, consumo de energia e robustez são extremamente importantes quando tratamos desse tipo de sistema. Tamanho e peso, devido ao fato de eles estarem fisicamente alojados em outros dispositivos ou sistemas; nesse caso, quanto menor e mais leve, melhor. A robustez se refere à necessidade de alguns dispositivos suportarem condições adversas, como vibrações, diferenças de temperatura, poeira, umidade, corrosão, variações na tensão de alimentação, interferências eletromagnéticas, entre outras.

Com relação às aplicações para os quais os sistemas embarcados são desenvolvidos, podemos analisar quatro tipos [15]:

* Propósito geral – Apesar dos sistemas específicos serem mais comuns, alguns deles desempenham tarefas genéricas no interior de outros aparelhos, como é o caso dos consoles de vídeo-game, por exemplo.
* Sistemas de controle – Possuem controles em malha fechada com realimentação em tempo real. Geralmente são sistemas com pouca interação com o usuário, com algumas sinalizações simples do estado do programa.
* Processamento de sinais – Apresenta certa quantidade de dados a serem processados em pouco tempo. Podem ser utilizados conversores analógico-digital (AD) e digital-analógico (DA) para trabalhar com o formato correto dos dados. É o caso de radares, sonares, filtros e tratamento de áudio e vídeo.
* Comunicações e redes – Realizam o chaveamento e distribuição de informações, como os sistemas de telecomunicações.

Diferente do caso dos PCs, que são dispositivos que permitem ao usuário realizar tarefas bastante variadas ou inclusive instalar aplicativos e softwares adicionais, os sistemas operacionais dos sistemas embarcados (quando existem), são pouco flexíveis, sendo configurados no momento da especificação e se restringindo a executar as tarefas para as quais foram desenhados e desenvolvidos [15]. A flexibilidade permitida e desejada nesse caso seria a possibilidade de atualização para uma nova versão do sistema como um todo, possibilitando correções, melhorias e adição de novas funcionalidades que aperfeiçoem o dispositivo. Porém isso é realizado pelos fabricantes e raramente pelos usuários finais.

1. **Bluetooth**

Como já foi apresentado, o Bluetooth possibilita a formação de uma rede de comunicação de dados sem fios, garantindo assim uma grande mobilidade e flexibilidade aos seus componentes. Neste capítulo será mostrada uma visão um pouco mais detalhada do funcionamento do padrão Bluetooth.

Visando estabelecer o padrão Bluetooth que é hoje internacionalmente utilizado, em 1998 algumas empresas uniram-se e formaram o Bluetooth SIG, uma associação privada sem fins lucrativos. Hoje esse grupo já ultrapassa os 13000 membros, incluindo empresas líderes em telecomunicações, computação, eletrônicos de consumo, automotivos, automação industrial e indústrias de rede, que são responsáveis pelo desenvolvimento, implementação e venda da tecnologia [16]. As principais tarefas do Bluetooth SIG são publicar as especificações Bluetooth, administrar o programa de qualificação, proteger as marcas registradas e divulgar a tecnologia.

A especificação do Bluetooth, que pode ser encontrada no site oficial (<www.bluetooth.com>), define vários protocolos, agrupados em camadas que formam a chamada *pilha Bluetooth*, e perfis, que explicitam exatamente quais partes de cada protocolo são necessárias para satisfazer uma determinada função. Várias funcionalidades são opcionais, permitindo, assim, que haja uma diferenciação nos produtos.

Atualmente essa especificação encontra-se em desenvolvimento e a melhoria mais recente consiste na versão Bluetooth 4.0 (liberada em Dezembro de 2009), que abriga três especificações menores: a tecnologia clássica Bluetooth, a tecnologia de baixo consumo de energia Bluetooth, e a tecnologia de alta velocidade Bluetooth [8]. Essas três podem ser combinadas ou utilizadas separadamente em diferentes dispositivos, de acordo com suas necessidades.

Nas seções que se seguem iremos estudar mais profundamente como está especificado o padrão Bluetooth, abordando suas características, a arquitetura, a estrutura da pilha, os perfis e as aplicações nas quais ele é empregado.

* 1. **Arquitetura**

A arquitetura Bluetooth é constituída basicamente de um dispositivo de transmissão, conhecido como *transceiver*, e uma pilha de protocolos. A pilha será discutida posteriormente na seção 3.3.

O transceiver Bluetooth opera na faixa não-licenciada ISM (*Industrial, Scientific and Medical*) de 2,4 GHz (2400 a 2483.5 MHz). Devido ao fato dessa faixa do espectro eletromagnético ser amplamente utilizada por diversas outras tecnologias (aparelhos Wi-Fi, fornos microondas, telefones sem fio, alarmes de carro, etc.), o padrão Bluetooth estabelece saltos pseudo-randômicos de frequência para minimizar a ocorrência de interferência na transmissão dos dados. Existem 79 canais disponíveis (porém alguns países permitem o uso de apenas 23 canais), com largura nominal de 1 MHz. Para compensar regularizações em alguns países, existe uma banda de guarda inferior de 2 MHz e uma superior de 3,5 MHz [8]. A taxa de transmissão é de até 1 Mbps, mas pode chegar a 2 ou 3 Mbps caso seja usado o modo EDR (*Enhanced Data Rate*) [9] [19].

O máximo pico de potência de saída permitido é de 100 mW e um dispositivo não pode operar num mesmo canal por mais de 0,4 segundos (400 ms) dentro de um período de 30 segundos. Quando há uma conexão entre dispositivos Bluetooth, o salto de frequência ocorre tipicamente a 1600 vezes por segundo, com um tempo de permanência de 625 μs. Quando o dispositivo está nos modos de *Solicitação* (*Inquiry*) ou *Página* (*Page*) (explicados a seguir, na seção 3.2) salta a 3200 vezes por segundo, com tempo de permanência de 312,5 μs [9].

Segundo os critérios de potência de emissão e alcance, os dispositivos Bluetooth podem ser divididos em três classes, como mostra a *Tabela 2* abaixo.

|  |  |  |  |
| --- | --- | --- | --- |
| **Classe** | **Potência mínima** | **Potência máxima** | **Alcance** |
| Classe 1 | 1 mW (0 dBm) | 100 mW (20 dBm) | Longo alcance (100 m) |
| Classe 2 | 0,25 mW (-6 dBm) | 2,5 mW (4 dBm) | Curto alcance (10 m) |
| Classe 3 | N/A | 1 mW (0 dBm) | Curtíssimo alcance (1 m) |

Tabela 2. Classificação dos dispositivos Bluetooth

A especificação Bluetooth define como os dispositivos da rede irão agrupar-se para realizar a comunicação. Numa piconet, cada um dos dispositivos é identificado por um endereço de 3 bits. Apesar da limitação de oito dispositivos ativos numa piconet, existe a possibilidade de que escravos excedentes permaneçam sincronizados à sequência de saltos de frequência, porém inabilitados a transmitir. Esses escravos, que permanecem num estado inativo (também denominado passivo ou estacionado) podem ser acessados de modo direto, por meio de 255 endereços do tipo *parked slave address* (endereços de escravos estacionados), ou indireto, através do *Bluetooth device address* (endereço do dispositivo Bluetooth, obrigatoriamente presente e único para cada dispositivo) [9]. Para reativar um dispositivo inativo, o mestre deve primeiramente transferir um dispositivo em estado ativo para estacionado.

Qualquer dispositivo Bluetooth pode atuar numa piconet como mestre, escravo ou ponte, sendo estas regras temporárias e existentes apenas durante a existência da própria rede. O dispositivo mestre é o único que pode inicializar um link de comunicação Bluetooth, porém uma vez estabelecido, um dispositivo escravo pode requisitar uma troca mestre/escravo para adquirir o controle da rede [11]. O mestre escolhe a frequência, a sequência do salto de frequência, a temporização (quando os saltos realmente ocorrem) e a ordem de polling dos escravos, além de ser responsável por instruí-los a mudar para diferentes estados em períodos de inatividade. Para que o dispositivo escravo se conecte à piconet, o mestre deve compartilhar seu identificador global (*Global ID*) e o offset do clock. O escravo deve ser capaz de recriar o salto de frequência, saber qual frequência utilizar em determinado tempo e sincronizar-se com o clock do mestre.

Um dispositivo é classificado como ponte (bridgeou gateway) quando está conectado a mais de uma piconet, podendo ser escravo em todas elas ou mestre em uma e escravo nas restantes. Isso resulta numa conexão entre as duas ou mais piconets formando uma scatternet. A ponte se comunica com todas as piconets conectadas a ela alinhando-se com o clock de cada piconet, porém comunicando-se com uma piconet por vez. Devido ao overhead causado pela mudança do clock, existe certa possibilidade do dispositivo ponte tornar-se um gargalo [9].

O Bluetooth realiza a transmissão de pacote baseada em polling. Toda comunicação entre dispositivos ocorre exclusivamente entre um mestre e um escravo, utilizando TDD (*Time Division Duplexing*) e TDMA (*Time Division Multiple Access*), sem comunicação direta entre dois escravos. O mestre verifica se cada escravo ativo tem dados a transmitir, e este último só transmite após a realização do polling, enviando os dados no slot(quadro) de tempo imediatamente seguinte ao da verificação. O mestre transmite apenas em slots de tempo pares, enquanto os escravos transmitem nos ímpares, e em cada slot de tempo é utilizado um canal de frequência diferente (um salto na sequência de saltos) [9] [11] [17].

* 1. **Máquina de Estados**

Os dispositivos Bluetooth seguem uma sequência de estados para se conectar a uma rede e podem ser encontrados em um dos seguintes estados: *Espera* (*Standby*), *Solicitação* (*Inquiry*), *Página* (*Page*), *Conectado* (*Connected*), *Transmissão* (*Transmit*), *Escuta* (*Sniff*), *Bloqueado* (*Hold*) ou *Estacionado* (*Park*) [9] [17]. A seguir, na *Figura 2*, segue um gráfico mostrando todos os estados disponíveis, assim como a sequência de mudanças de estados, explicadas posteriormente.

**Estados de baixo consumo de energia**

**Estados ativos**

**Estados de conexão**

**Estado de desconexão**

Figura 2. Máquina de estados dos dispositivos [9]

O dispositivo está no modo *Espera* (*Standby*) quando está ligado, porém ainda não se encontra conectado a nenhuma piconet. Nesse estado o ciclo de trabalho (*duty cycle*) é de menos de 1% [12]. O estado de *Solicitação* (*Inquiry*) é atingido quando o dispositivo envia requisições para encontrar outros dispositivos com os quais ele pode comunicar-se. Ele gera um subconjunto com 32 canais (dos 79 disponíveis) da sequência de saltos e envia mensagens de solicitação ao passo que troca de canal conforme foi pré-estabelecido. Periodicamente dispositivos detectáveis entram no estado de verificação de solicitações (*Inquiry Scan*) e todos os dispositivos dentro do raio de alcance neste estado respondem à solicitação com seu endereço [38].

Após verificar os dispositivos próximos, no estado de *Página* (*Page*) o mestre gera uma sequência de saltos baseada no endereço e clock estimado do dispositivo remoto a se conectar, sendo possível assim enviar mensagens de requisição de conexão aos dispositivos os quais deseja convidar para juntar-se à sua piconet. Os dispositivos remotos conectáveis entram periodicamente no estado de verificação de página de forma a receber as requisições de conexão. Ao recebê-las, respondem ao mestre com um pacote de resposta de página, que envia de volta uma sincronização de saltos de frequência (FHS – Frequency Hopping Synchronization), incluindo o endereço Bluetooth e clock do mestre. O dispositivo remoto envia então um aviso de reconhecimento (ack) e o mestre, após recebê-lo, gera uma nova sequência de saltos de frequência baseada em seu próprio endereço e clock. O escravo por fim utiliza o endereço e clock do mestre para recriar uma sequência de saltos idêntica [38]. Nesse momento os dispositivos estão conectados e sincronizados e podem começar a comunicar-se através do polling do mestre. A *Figura 3* mostra a sequência dos estados alcançados durante o processo de solicitação e conexão explicado anteriormente.

Figura 3. Máquina de estados do processo de solicitação e conexão

Quando ocorre uma conexão bem sucedida entre o mestre e um novo dispositivo, este último assume o papel de escravo, entrando no estado *Conectado* (*Connected*) e recebendo um endereço ativo que o identifica na piconet. Em seguida, escuta o slot de transmissão do mestre em busca de pacotes com seu endereço. Caso não existam pacotes com seu endereço, ele “dorme” até a próxima escuta. O dispositivo mestre realiza o polling nos escravos regularmente. Enquanto está conectado, o escravo está apto a transmitir dados quando o mestre o solicitar, ou seja, quando verificar um pacote com seu endereço, passando assim ao estado de *Transmissão* (*Transmit*). No final da transmissão, ele volta ao estado *Conectado*.

O estado de *Escuta* (*Sniff*) é um estado de baixo consumo de energia no qual o dispositivo “dorme” durante um número pré-determinado de slots de tempo, concordado entre o mestre e o escravo. Ele então acorda no slot de tempo definido para a transmissão de dados e volta ao estado inativo até que o próximo slot designado chegue. Dentre os três estados de baixo consumo de energia, este é o que têm o maior ciclo de trabalho (menor economia). O estado *Bloqueado* (*Hold*) é outro estado de baixo consumo no qual o dispositivo fica inativo por um determinado período, sem transmitir dados. Após esse período o dispositivo reinicia as transferências instantaneamente.

Quando um dispositivo escravo não possui dados para enviar ou receber, o mestre deve instruí-lo a entrar no estado *Estacionado* (*Park*). Enquanto está neste estado, o escravo renuncia a seu endereço, que poderá ser entregue a outro dispositivo reativado a partir do estado *Estacionado* ou a um novo dispositivo descoberto. Este estado é o que possui o menor consumo de energia entre todos os estados ativados (conectados).

* 1. **Pilha de Protocolos Bluetooth**

A especificação Bluetooth divide a pilha de protocolos em três grupos lógicos: Protocolos de Transporte, Protocolos de Middleware e Aplicações [9] [17].

O grupo de protocolos de Transporte da pilha Bluetooth permite que os dispositivos Bluetooth descubram uns aos outros e possibilita o gerenciamento dos links físicos e lógicos para protocolos e aplicações de camadas mais altas. Ele corresponde às camadas Física e de Enlace do modelo OSI (*Open Systems Interconnection Reference Model*), porém não possuem relação direta com a camada de Transporte do mesmo. Os protocolos RF (*Radio Frequency*), *Baseband*, *Link Manager*, L2CAP (*Logical Link Control and Adaptation Protocol*) e HCI (*Host Controller Interface*), os quais estão inclusos nesse grupo, suportam transmissão síncrona e assíncrona e são necessários para dar suporte à comunicação entre os dispositivos Bluetooth.

O grupo de protocolos de Middleware inclui protocolos de terceiros e de padrões industriais, assim como os protocolos desenvolvidos pelo Bluetooth SIG. Protocolos de padrão industrial incluem o PPP (*Point-to-Point Protocol*), IP (*Internet Protocol*), TCP (*Transmission Control Protocol*), WAP (*Wireless Application Protocols*) e protocolos OBEX (*Object Exchange*). Dos protocolos desenvolvidos pelo Bluetooth SIG, podemos citar um emulador de porta serial que permite que aplicações legadas operem sem problemas sobre os protocolos de transporte Bluetooth (RFCOMM - *Radio Frequency Communication Protocol*), um protocolo de sinalização de controle de telefonia baseado em pacote (TCS - *Telephony Control Signaling*), e um protocolo de descobrimento de serviço (SDP - *Service Discovery Protocol*) que permite aos dispositivos obterem informações sobre os serviços disponíveis.

O grupo de Aplicações consiste em aplicações reais que usam os links Bluetooth, podendo incluir tanto aplicações legadas quanto aquelas orientadas ao Bluetooth.

As camadas *Baseband* e *Link Manager* são implementadas no chip do controlador Bluetooth (hardware), que oferece ao host a interface HCI para o controle de todas as funções de baixo nível. O host (software) implementa as camadas HCI, L2CAP e protocolos de mais alto nível como SDP e RFCOMM [18].

A seguir são exploradas as camadas de protocolos da pilha que são mais importantes no desenvolvimento de aplicações para sistemas embarcados, partindo dos níveis mais baixos (próximos ao hardware) até os mais altos (próximos às aplicações). Na *Figura 4* pode-se visualizar a posição de cada camada dentro da pilha.

Figura 4. Estrutura da pilha de protocolos Bluetooth

Bluetooth RF

Bluetooth Baseband

Link Manager Protocol (LMP)

L2CAP

PPP

TCP/UDP

SDP

Audio

OBEX

IP

Host-Controller Interface (HCI)

RFCOMM

**Host**

**Controlador**

* + 1. **Camada de Rádio Bluetooth**

Essa é a camada mais baixa, responsável por especificar detalhes como a frequência de operação, potência e as técnicas de modulação e transmissão utilizadas. Inclui a antena de transmissão, uma interface de radiofrequência (RF) e o transceiver [17]. A antena é o componente que gera as ondas de rádio que transmitem os dados, podendo ser interna ou externa e com possibilidade de integração a componentes de terceiros. A interface contém um filtro de banda passante da antena e um amplificador de energia, sendo responsável pelas trocas entre os estados de transmissão e recepção. O transceiver RF utiliza modulação e demodulação GFSK (*Gaussian Frequency Shift Keying*), abrigando um sintetizador de radiofrequência, filtros Gaussianos, recuperação de clock e detector de dados.

* + 1. **Camada *Baseband***

A camada *Baseband*, considerada a camada física da pilha Bluetooth, fica imediatamente acima da camada de rádio, definindo como os dispositivos procuram e conectam-se uns aos outros. Nela são estabelecidas as regras de mestre e escravo utilizadas, como também a sequência de salto, os tipos de pacotes, procedimentos de processamento de pacotes e estratégias para detecção e correção de erros, cifragem de sinais (*whitening*), encriptação, transmissão e retransmissão de pacotes [9]. Também é nessa camada onde são feitos o gerenciamento de links síncronos e assíncronos e as paginações e solicitações aos dispositivos.

Aqui são suportados dois tipos de links: ACL (*Asynchronous Connectionless*) e SCO (*Synchronous Connection-Oriented*). Um dispositivo pode possuir um link ACL e até dois links SCO (três para o mestre) simultaneamente [12]. O primeiro é um link ponto-a-multiponto entre o mestre e todos os escravos participantes da piconet, utilizado apenas para dados. Um dispositivo com link ACL pode enviar pacotes com tamanho variável de 1, 3 ou 5 slots de tempo, porém não possui slots reservados [9]. Já os links SCO são simétricos do tipo ponto-a-ponto entre o mestre e o escravo, podendo conter informações de áudio (voz), dados genéricos ou uma combinação dos dois. São caracterizados por uma atribuição periódica de pacote em um único slot (reservas de slot) e usados primariamente para transmissão de voz, que requer uma transferência de dados consistente e rápida. Links SCO são tratados como prioritários, sendo servidos antes de quaisquer pacotes ACL, e apenas os assíncronos são retransmitidos para garantir a integridade dos dados.

O pacote da camada *Baseband* é dividido em três partes: código de acesso (72 ou 68 bits), cabeçalho (54 bits) e carga útil (variável, de 0 a 2745 bits). Os códigos de acesso contêm o endereço da piconet e são utilizados para sincronização de tempo, compensação de deslocamento, paginação (*paging*) e solicitação (*inquiry*), podendo ser de três tipos diferentes: *Channel Access Code* (CAC), usado para identificar uma piconet; *Device Access Code* (DAC), usado para paginação; e *Inquiry Access Code* (IAC), usado para solicitações de busca de dispositivos. Um dispositivo pode saber se um pacote foi endereçado a ele apenas verificando o cabeçalho. Caso não seja, ele pode voltar a “dormir” durante o resto do slot de tempo, permitindo uma maior economia de energia mesmo em modo ativo. O cabeçalho possui informações para reconhecimento, numeração e reordenação de pacotes, controle de fluxo, endereço do escravo e checagem de erro de cabeçalho, sendo codificado com um código FEC (*Forward Error-Correcting*) com taxa de alta confiabilidade de 1/3 (cada bit do cabeçalho é retransmitido três vezes). O campo de carga útil (*payload*) também pode ser codificado com o FEC e inclui um código CRC de 16 bits para detecção e correção de erros (exceto para pacotes SCO). Com relação às taxas de transmissão, canais síncronos suportam um link de 64 kbps; canais assíncronos suportam um link assimétrico máximo de 723,2 kbps em uma direção e 57,6 kbps na direção contrária, ou um link simétrico de 433,9 kbps em ambas as direções [13]. O pacote pode ocupar mais de um slot de tempo (pacotes multi-slot), continuando a transmissão no próximo slot.

* + 1. **Camada de Gerenciamento de Link (*Link Manager*)**

Essa camada, localizada no controlador Bluetooth, é a responsável por implementar o LMP (*Link Manager Protocol*), que gerencia as conexões entre os dispositivos, garantindo segurança, configuração e controle dos links. Dentre suas principais funções, o protocolo controla a alocação da largura de banda (*bandwidth*) para dados genéricos, reserva largura de banda necessária para o tráfego de áudio, atende a questões de segurança como autenticação (através do uso de métodos desafio-resposta) e encriptação de dados, define os tamanhos de pacotes da camada *Baseband* e gerencia o uso de energia. Este último leva em consideração a negociação dos modos de atividade de baixo consumo e a determinação de níveis de energia de transmissão.

O LMP trabalha através da troca de mensagens para a realização de transações. Essas mensagens são filtradas e depois interpretadas pelo gerenciador de conexão. As mensagens de gerenciamento de link têm maior prioridade que os dados do usuário, de forma que elas não devem ser atrasadas devido a tráfego nas camadas superiores (L2CAP), embora possa ser atrasada por múltiplas retransmissões de pacotes individuais da camada *Baseband* [13].

* + 1. **Camada de Interface Host-Controlador (HCI)**

A HCI (*Host Controller Interface*) é um misto da camada de nível mais baixo no lado do host com a de nível mais alto no lado do controlador e define uma interface padrão para o acesso das aplicações de nível mais alto às camadas mais baixas da pilha de protocolos. Essa camada não é uma parte essencial da especificação, mas seu objetivo é possibilitar a interoperabilidade entre dispositivos e o uso de protocolos de aplicações de nível mais alto existentes, através de uma interface de driver que abstrai as dependências de transporte das camadas inferiores.

A HCI pode ser dividida em três partes: driver, firmware e um módulo de transporte entre eles. O driver está localizado na parte de software (host), tendo a responsabilidade de receber notificações assíncronas dos acontecimentos. Após o recebimento, ele avalia o pacote e determina qual evento ocorreu. O firmware fica no hardware (controlador) e implementa os comandos para acessar a camada *Baseband*, a camada *Link Manager* e os registradores de status, controle e eventos. O módulo de transporte localiza-se entre o driver e o firmware provendo uma transferência de dados sem a necessidade de entendê-los. O host deve receber notificações de eventos HCI independente do módulo de transporte utilizado (geralmente são usados protocolos seriais, como RS-232, UART – *Universal Asynchronous Receiver/Transmitter* – ou USB – *Universal Serial Bus*) [19].

Basicamente, existem três tipos de pacotes: comando, evento e dados. Pacotes de comando podem ser enviados apenas do host para o controlador e são utilizados para ler e escrever parâmetros gerais como nome do dispositivo, timeouts e classe de dispositivo e para instanciar consultas e inicializar conexões de dados. Pacotes de evento são enviados apenas do controlador para o host, utilizados para transportar resultados e informações de um ao outro, como uma resposta a um comando previamente enviado ou uma requisição de outro dispositivo. Pacotes de dados podem ser transmitidos de dois modos: utilizando um canal síncrono (SCO), onde a largura de banda de upload e download são iguais; ou assíncrono, com taxas de download e upload assimétricas. O tamanho do pacote HCI é ajustável entre 17 e 339 bytes de carga útil (payload), e os pacotes podem conter detecção ou correção de erro [18].

* + 1. **Camada de Controle de Link Lógico e Adaptação (L2CAP)**

A partir dessa camada, os protocolos são todos localizados no lado do host (em software). Através da implementação do L2CAP (*Logical Link Control and Adaptation Protocol*) obtém-se um alto nível de multiplexação, segmentação e remontagem de pacotes, uma garantia de Qualidade de Serviço (QoS) e suporte a abstração de Grupos. A camada não provê confiabilidade, sendo a *Baseband* a responsável por essa garantia [11].

Canais são utilizados para estabelecer caminhos diferentes entre as aplicações em dispositivos Bluetooth. Estes canais são identificados de forma a nomear cada ponto terminal de conexão. Os pacotes trafegados podem ser de dois tipos: dados e sinalização [18]. Os de sinalização são usados para configuração e manipulação de conexões. O tamanho máximo de cada pacote é 64 KB, mas é possível fragmentar pacotes maiores para que os protocolos mais altos não precisem enviar o cabeçalho várias vezes. O L2CAP ainda provê serviços de dados orientados ou não a conexões para as camadas superiores, porém sua comunicação com a camada *Baseband* se dá apenas através de links ACL.

* + 1. **Protocolo de Descobrimento de Serviços (SDP)**

O SDP (*Service Discovery Protocol*) é o protocolo base que permite às aplicações descobrir quais serviços estão disponíveis na rede e determinar suas características, otimizado para a natureza altamente dinâmica das comunicações Bluetooth. Possui apenas mecanismos de descoberta, ficando a cargo dos outros protocolos os métodos de acesso aos serviços. Pode ser utilizado sem problemas em conjunto com outros protocolos de descobrimento, como o Jini e o UPnP.

Esse protocolo é implementado utilizando-se um modelo de comunicação cliente-servidor. O servidor mantém uma lista de registros dos serviços associados a ele. O cliente busca informações de um registro de serviço mantido pelo servidor SDP através de uma solicitação SDP, e caso deseje utilizar o serviço, deve abrir uma conexão em separado ao servidor do serviço. Um dispositivo Bluetooth pode atuar tanto como cliente quanto como servidor [13].

* + 1. **Protocolo de Comunicação de Radiofrequência (RFCOMM)**

O RFCOMM (*Radio Frequence Communication Protocol*), baseado no padrão do ETSI (European Telecommunications Standards Institute) TS 07.10, é um emulador de interface serial que atua sobre o L2CAP. Possui capacidade para emular os nove circuitos de portas seriais RS-232 e suporta até 60 conexões simultâneas entre dois dispositivos Bluetooth, no entanto, esse número depende de cada dispositivo. O protocolo provê um fluxo de dados confiável, múltiplas conexões concorrentes, controle de fluxo e configurações de linha de cabo serial [11].

O RFCOMM acomoda dois tipos de dispositivos: pontos terminais de comunicação, como computadores e impressoras; e dispositivos que fazem parte do segmento de comunicação, como modems [13].

* + 1. **Protocolos PPP, UDP/TCP/IP**

A especificação Bluetooth dá suporte ao uso da suíte de protocolos de transporte e rede UDP/TCP/IP, bastante conhecidos e utilizados em larga escala em aplicações que envolvem a Internet e redes locais. Apesar de ser possível rodar diretamente os protocolos UDP, TCP e IP sobre a camada L2CAP, atualmente ainda não existem perfis definidos para esse propósito, de modo que a maioria das implementações dos fabricantes é feita sobre o protocolo PPP (*Point-to-Point Protocol*) rodando sobre o RFCOMM. Devido à grande popularidade do TCP/IP, seu desempenho sobre links Bluetooth é de muito interesse a muitos desenvolvedores, e diversos estudos ainda estão em andamento na tentativa de solucionar os problemas e as limitações existentes [12].

* + 1. **Protocolo de Troca de Objetos (OBEX)**

O OBEX (*Object Exchange Protocol*) é um protocolo de sessão definido pela IrDA para interconectar os dispositivos que suportam os seus protocolos. É um protocolo binário compacto e eficiente, que habilita os dispositivos a trocarem dados de maneira simples e espontânea e permite o desenvolvimento de aplicações que funcionam bem tanto em um ambiente infravermelho como em um de radiofreqüência de curto alcance, possibilitando uma integração entre as duas tecnologias num mesmo dispositivo. Embora a IrDA também defina uma forma não orientada à conexão, no Bluetooth apenas o padrão orientado à conexão é utilizado. [13]

Opera através do uso de um modelo cliente-servidor e independe de mecanismo e API de transporte (escolhido pelo protocolo RFCOMM). O dispositivo que deseja configurar uma sessão comunicação OBEX é considerado o cliente. Três perfis de aplicações foram desenvolvidos para esse protocolo: sincronização (SYNC), transferência de arquivos (FTP) e *object push profile* (OPP). [24]

* 1. **Perfis Bluetooth**

Os perfis representam soluções padrão para os modelos de uso em nível de aplicação do Bluetooth, que podem ser vistos como cenários de uso para aplicações específicas. Um perfil descreve o comportamento através de um corte vertical da pilha de protocolos, definindo os protocolos e funcionalidades suportadas em cada modelo de uso. Cada dispositivo Bluetooth pode suportar um ou mais perfis. Definições de perfis possibilitam a interoperabilidade entre dispositivos (inclusive de fabricantes diferentes) não só no nível do rádio, mas também no nível da aplicação. A *Figura 5* mostra alguns perfis importantes definidos para o Bluetooth.

Figura 5. Exemplos de perfis Bluetooth [37]

 **Generic Access Profile**

 **Serial Port Profile**

Service Discovery Application Profile

**TCS-BIN Based Profiles**

**Generic Object Exchange Profile**

LAN Access Profile

Headset Profile

Dial-up Networking Profile

Fax Profile

File Transfer Profile

Synchronization Profile

Object Push Profile

Intercom Profile

Cordless Telephony Profile

Entre os mais diversos perfis Bluetooth disponíveis podemos citar os seguintes principais: *Advanced Audio Distribution Profile* (A2DP), *Audio Video Remote Control Profile* (AVRCP), *Basic Imaging Profile* (BIP), *Basic Printing Profile* (BPP), *Cordless Telephony Profile* (CTP), *Dial-up Networking Profile* (DUN), *Fax Profile* (FAX), *File Transfer Profile* (FTP), *Generic Access Profile* (GAP), *Generic Audio/Video Distribution Profile* (GAVDP), *Generic Object Exchange Profile* (GOEP), *Hardcopy Cable Replacement Profile* (HCRP), *Hands-Free Profile* (HFP), *Human Interface Device Profile* (HID), *Headset Profile* (HSP), *Intercom Profile* (IP), *LAN Access Profile* (LAP), *Object Push Profile* (OPP), *Personal Area Networking Profile* (PAN), *SIM Access Profile* (SAP), *Service Discovery Application Profile* (SDAP), *Synchronization Profile* (SYNC), *Serial Port Profile* (SPP) [25].

Dentre eles, alguns têm um maior destaque em aplicações de sistemas embarcados. Existem quatro perfis genéricos que são usados por diferentes perfis baseados em modelos de uso: um perfil de porta serial (*Serial Port Profile* – SPP), um perfil de acesso genérico (*Generic Access Profile* – GAP), um perfil para descobrimento de serviços (*Service Discovery Application Profile* – SDAP) e um perfil de troca de objetos genéricos (*Generic Object Exchange Profile* – GOEP) [11]. A seguir são apresentadas descrições um pouco mais detalhadas dos perfis mais importantes e utilizados [37].

Através do GAP, que faz uso tanto de protocolos de mais baixo nível (LMP e *Link Control*) quanto os mais altos (L2CAP, RFCOMM e OBEX), é provido um conjunto de descobrimento básico de dispositivo, gerenciamento de conexões e capacidades de segurança. Por razão de interoperabilidade e compatibilidade dos dispositivos Bluetooth, este perfil é um requisito obrigatório para todos eles.

O SDAP descreve a descoberta de serviços oferecidos (outros perfis inclusive) pelos dispositivos Bluetooth na vizinhança, fazendo uso do protocolo SDP da pilha de protocolos.

O SPP utiliza o protocolo RFCOMM para permitir cenários de substituição de cabeamento. Não existe necessidade de papéis mestre/escravo, sendo definidos papéis de dispositivo de mesmo nível (*peer device roles*) para comunicação serial geral. Como mostra a Figura 4, o SPP atua como base ou perfil abstrato para vários outros perfis que utilizam esse tipo de comunicação, como o DUN, FAX, LAP, entre outros.

O LAP descreve como conexões PPP podem ser configuradas sobre o RFCOMM para que os dispositivos se conectem e obtenham acesso aos serviços da LAN. Ele também define como dispositivos de uma piconet podem formar uma LAN independente usando o mesmo mecanismo de conexões. Existem dois papéis definidos para os dispositivos Bluetooth que suportam o LAP: ponto de acesso LAN (provê acesso de serviço) e terminal de dados LAN (requisita acesso à rede).

O GOEP atua de forma similar ao SPP, servindo como um perfil abstrato para outros perfis. Ele define aspectos comuns, incluindo papéis de dispositivos, estabelecimentos de conexões e considerações de segurança. O protocolo OBEX é utilizado para habilitar operações comuns de troca de dados suportadas pelos perfis OPP, FTP e SYNC.

O SYNC define os requisitos, comportamento de protocolos e procedimentos que devem ser usados pelos dispositivos Bluetooth para o oferecimento de aplicações de sincronização, fazendo uso do GOEP para definir os requisitos de interoperabilidade.

* 1. **Aplicações**

Com base nas características apresentadas, podemos analisar várias áreas e exemplos de aplicações do padrão Bluetooth. A tecnologia Bluetooth apresenta flexibilidade suficiente para ser utilizada nos mais diversos aparelhos, desde computadores pessoais e dispositivos móveis a sistemas embarcados.

Por exemplo, a partir de um computador e um telefone celular (ou PDA) pode-se compartilhar o acesso à Internet, sincronizar e editar contatos, mensagens e outras informações pertinentes, fazer ligações, enviar mensagens e gravar conversas telefônicas no computador, sem a necessidade de fios ou até mesmo de intervenção manual, uma vez que é possível manter uma comunicação permanente entre eles.

Outra funcionalidade bastante interessante é o estabelecimento de PANs (*Personal Area Networks*) para conectar computadores (celulares ou dispositivos de multimídia) e periféricos, como teclados, mouses, scanners, impressoras, headsets (fones de ouvido), câmeras digitais, etc. Telefones residenciais e intercomunicadores de curta distância também podem utilizar Bluetooth para realizar transmissões sem fio.

Além disso, é possível listar o conteúdo de um dispositivo remoto (desde que esteja autorizado) e realizar a transferência de objetos, como arquivos, diretórios, contatos, inclusive através de operações do tipo “*push*” na troca de cartões de visita, por exemplo. Podemos citar ainda o uso em casas inteligentes, onde o usuário seria reconhecido no momento de chegada através de seu identificador e algumas tarefas seriam realizadas automaticamente, como abrir portas, acender luzes, ajustar a temperatura ambiente, etc.

Visando um pouco mais as áreas comercial e industrial, o Bluetooth também poderia ser aplicado em sistemas de sensoriamento remoto, redes de sensores sem fio (para monitoração ambiental, gerenciamento de suprimentos e localização ou rastreamento de pessoas, por exemplo), automação industrial (controle de processos) e predial (iluminação, climatização, segurança, controle de acesso). Na área médica existem alguns estudos [como 32] que implementam e analisam o uso do Bluetooth para monitoramento contínuo de pacientes.

1. **Implementações e Análises**

Nesse capítulo serão apresentadas as soluções encontradas de pilhas Bluetooth para sistemas embarcados e as respectivas análises do ponto de vista de dificuldades e limitações da implementação. Em seguida são sugeridas melhorias com base nas avaliações realizadas.

* 1. **Pilhas Bluetooth para Sistemas Embarcados**

Existem hoje no mercado algumas implementações de pilhas Bluetooth desenvolvidas para sistemas embarcados. Entre elas, podemos citar as pilhas de protocolos lwBT, Bluetooth BTnode, IAR, BeeCon MicroBlue, AVE-Blue, BlueletTM, BlueRapport© e Bluetopia®. Primeiro abordaremos as pilhas de código aberto (lwBT e BTnode), em seguida, as de soluções proprietárias.

* + 1. **lwBT**

lwBT é implementação da pilha de protocolos Bluetooth para sistemas embarcados leve e open source (disponível sob uma licença BSD). Ela atua como uma interface de rede para a pilha de protocolos lwIP, transportando os dados IP sobre o Bluetooth. O foco principal da implementação Bluetooth lwBT é a redução do uso de memória RAM, sem abrir mão das funcionalidades básicas Bluetooth. O conjunto lwBT/lwIP se faz adequado para sistemas embarcados que dispõem de apenas algumas dezenas de kbytes de memória RAM livre e espaço para aproximadamente 60 kbytes de código ROM.

Ela suporta vários protocolos e camadas Bluetooth, incluindo: HCI, L2CAP, SDP, RFCOMM e PPP. Os perfis suportados são PAN, LAP, DUN e SPP. O código fonte da implementação da pilha lwBT pode ser encontrado no seguinte endereço: <https://github.com/lwalkera/lwBT>.

* + 1. **BTnode**

Esta solução, desenvolvida no Instituto Federal Suíço de Tecnologia de Zurich (ETH Zurich), está na sua terceira revisão, sendo composta por um microcontrolador e dois rádios independentes: um rádio de baixa potência de broadcast na faixa ISM, que permite formar redes multissalto (multi-hop), e um módulo Bluetooth.

A pilha Bluetooth BTnode está licenciada sob a GNU General Public License, de forma que pode ser utilizada abertamente para fins não comerciais. Ela oferece suporte às camadas HCI, L2CAP e RFCOMM, possibilitando alguma interoperabilidade entre os dispositivos.

Com relação ao uso de memória, a alocação é feita estaticamente. Isso facilita a portabilidade entre os sistemas operacionais, mas restringe em tempo de compilação o número de buffers que o host pode manipular. Porém essa desvantagem não é tão acentuada nesse caso, pois sistemas embarcados (microcontroladores) não possuem muita memória RAM para implementar muitos outros buffers. Para manter uma baixa utilização da memória, o host aloca o menor número possível de buffers [18].

Quanto à chamada de métodos, estão disponíveis chamadas bloqueantes ou não. Chamadas bloqueantes suspendem a aplicação até que o dado seja enviado e a resposta recebida, o que pode resultar em um longo tempo de espera. Já as chamadas não-bloqueantes retornam um manipulador para o rastreamento da chamada, devolvendo imediatamente o controle à aplicação, que fica responsável por verificar se a chamada já foi completada.

Na camada de transporte a especificação de transporte uma UART foi implementada. Ela oferece a inicialização de um dispositivo serial e métodos para enviar e receber pacotes para camadas mais altas. Várias funções da camada HCI também foram implementadas. Um contador global foi introduzido para que fosse possível que cada conexão utilizasse um manipulador de conexão único, diferente do que acontece normalmente com o controlador, uma vez que este pode reutilizar os manipuladores de conexões fechadas anteriormente, impedindo a diferenciação entre conexão novas e antigas.

* + 1. **IAR Embedded Bluetooth Protocol Stack**

IAR Systems ([http://www.iar.com](http://www.iar.com/)) é um provedor de soluções Bluetooth empregado pela Ericsson Microeletronics, que oferece uma compacta pilha de protocolos Bluetooth, escrita de baixo para cima para uso com restrição de recursos em sistemas embarcados e qualificada de acordo com a versão 1.1 da especificação Bluetooth [27]. Porém o código é distribuído apenas em formato binário, sem a possibilidade de análise ou alteração do código fonte.

A pilha é versátil e suporta vários protocolos de transporte, como UART, USB e conexões HCI embarcadas para verdadeiras soluções de único processador, onde a aplicação e a pilha são baixadas para o módulo Bluetooth. O núcleo da pilha implementa as camadas HCI, TCI, L2CAP, RFCOMM e SDP. Um grande número de perfis da pilha também é implementado, como os perfis frequentemente utilizados GAP, SDAP, SPP e HSP. A *Test Control Interface* (TCI) é uma funcionalidade essencial que torna a certificação do produto mais fácil, já que habilita o teste automático de conformidade usando o IAR PreQual, uma ferramenta de teste e certificação usada por muitos laboratórios de teste.

IAR Systems oferece uma ferramenta de configuração e otimização do estado-da-arte da pilha de protocolos Bluetooth, baseada na tecnologia IAR MakeApp, de forma a facilitar e tornar mais rápida a adaptação e modificação da pilha para o uso aprimorado na plataforma alvo particular. Usando a ferramenta de configuração e otimização IAR MakeApp é possível configurar facilmente as funcionalidades da pilha necessárias para a aplicação e a adaptação da pilha para uso em um novo ambiente de hardware ou software é feita em poucos passos, gerando uma pilha feita completamente sob medida para a plataforma alvo. Após a configuração das funcionalidades da pilha, a ferramenta gera um código otimizado com apenas as funcionalidades implementadas. O otimizador da pilha faz um corte vertical através da pilha e remove todo o código fonte desnecessário para a configuração selecionada. A pilha é automaticamente implementada para se adaptar à plataforma alvo selecionada e é modificada para se ajustar ao processador, sistema operacional e módulo ou chipset Bluetooth.

A título de exemplo, uma pilha com as camadas de núcleo e perfis necessários para implementar a funcionalidade de uma porta serial virtual precisa de menos de 25 kbytes de ROM e 2 kbytes de RAM em alguns sistemas. Uma vez que requisitos de ROM e RAM são extremamente baixos em sistemas embarcados, a pilha se adapta em um hardware menos poderoso, o que reduz o custo de produção em produção serial.

A pilha é projetada para sistemas embarcados reais e funciona bem em sistemas com ou sem sistema operacional. O código que depende do sistema operacional ou do hardware é separado da pilha por camadas de abstração para melhorar a portabilidade. A pilha já se encontra portada para Win32, CMX, Nucleus, e pode ser facilmente portada a novos sistemas operacionais pelo desenvolvedor da aplicação. A pilha suporta vários chipsets Bluetooth, como ALPS, CSR, Ericsson, Fujitsu, Infineon, Mitsubishi, OKI, Taiyo Yuden, Toshiba, Xemics e Zeevo, e faz uso de comandos HCI específicos de fabricantes se necessário.

* + 1. **BeeCon MicroBlue / AmbiCompTM Bluetooth Sandwich Module**

O Módulo Sanduíche Bluetooth é um componente do projeto AmbiCompTM (<http://www.beecon.de/ambicomp.html>), que provê Unidades de Controle para Inteligência de Ambientes (AICUs – *Ambient Intelligence Controller Units*) com uma interface Bluetooth, permitindo a comunicação, por exemplo, com telefones celulares, PDAs e computadores pessoais. Ele também pode integrar uma AICU em um sistema AmbiCompTM de maior escala [28].

A solução contém uma camada de abstração de hardware bem leve, a chamada BIOS, que provê acesso através da HCI ao transceiver Bluetooth e às linhas de entrada e saída do módulo. Ele também fornece uma interface a outros módulos sanduíches na AICU através do barramento *backplane*.

Uma leve pilha (MicroBlue) implementa nativamente os protocolos Bluetooth de baixo nível, tendo sido desenvolvida para atender requisitos de microcontroladores pequenos, baratos e com baixo consumo de energia. Através do perfil de porta serial (SPP) provê acesso em conformidade com a especificação Bluetooth. Alternativamente, o módulo oferece o protocolo L2CAP.

Uma instância da Máquina Virtual AmbiCompTM (ACVM) executa aplicações em Java transcodificadas. Também provê suporte Bluetooth - tanto nativamente, através do SPP, como com bibliotecas de Java flexíveis no topo do L2CAP. De acordo com o conceito AmbiCompTM, a ACVM roda sem problemas através de todos os módulos em uma AICU. Logo, os desenvolvedores de software não precisam endereçar particularmente o Módulo Sanduíche Bluetooth, podendo simplesmente usar suas funcionalidades em todos os módulos sanduíches na AICU.

* + 1. **AVE-Blue**TM

AVE-BlueTM é a pilha de protocolos Bluetooth original da ACCESS ([http://www.access-company.com](http://www.access-company.com/)), projetada para sistemas embarcados e de acordo com a versão 1.1 da especificação Bluetooth. Sua principal vantagem está na facilidade em adicionar a funcionalidade Bluetooth a uma aplicação de Internet que já utiliza o navegador NetFront da mesma empresa, uma vez que a pilha encontra-se totalmente integrada a ele. Permite a emulação do ambiente num PC, além de acelerar o desenvolvimento das aplicações através de suas APIs, exemplos de aplicações para cada perfil e funcionalidades de configuração para implementações de perfil seletivo [29].

A solução dá suporte às camadas HCI, L2CAP, RFCOMM, SDP e TCSBIN e a 12 perfis (GAP, SDAP, CTP, ICP, SPP, HSP, DUN, LAP, GOEP, OPP, FTP e SYNC). Seu núcleo da pilha é bastante compacto, sendo necessário dispor de apenas 60 KB de ROM e 5 KB de RAM. Para a realização de testes e certificação do produto é utilizada a interface TCI.

* + 1. **IVT Bluetooth Host Stack (BlueletTM)**

O IVT Bluetooth Host Stack (BlueletTM) é uma implementação completa da pilha Bluetooth V2.1 + EDR utilizando ANSI C. Incluindo todos os protocolos da pilha de protocolos Bluetooth (TCS, SDP, RFCOMM, L2CAP e HCI), o IVT realiza validação de protocolo usando a técnica de projeto formal, Linguagem de Descrição de Especificação (SDL), para garantir que todos os protocolos estão logicamente corretos antes da codificação. Suporta uma grande variedade de sistemas operacionais e quase todos os perfis Bluetooth e diferentes drivers de transporte HCI (USB, UART, PCMCIA, BCSP, SDIO), além de não ter requisitos particulares em relação a processadores [30].

IVT BlueletTM é apropriado tanto para aplicações embarcadas como para aplicações desktop. Foi cuidadosamente projetado para sistemas embarcados com tamanho de código extremamente pequeno. Em situações gerais, o tamanho da RAM é menor que 12 KB. O IVT BlueletTM é o núcleo do Bluesoleil.

A IVT também desenvolveu uma nova pilha de host para suportar a versão Bluetooth 3.0 + HS (*High Speed*). Essa versão possui a velocidade do padrão 802.11 e a Camada de Adaptação de Protocolo (PAL) do 802.11 provê uma vazão de transferência de dados maior, a uma taxa aproximada de 24 Mbps. Além disso, essa versão permite maior economia de energia devido ao melhor controle de potência e é retrocompatível com versões anteriores do Bluetooth.

* + 1. **BluRapport© (antigo BL-Rappore)**

Em maio de 2010, o antigo BL-Rappore trocou seu nome para BluRapport©. A solução é uma pilha Bluetooth desenvolvida pela DSR ([http://www.dsr-wireless.com](http://www.dsr-wireless.com/)) desde 2004, adequada tanto para aplicações desktop como para embarcadas. O BluRapport© implementa os protocolos e perfis Bluetooth acima da camada HCI para prover uma pilha Bluetooth funcional, robusta e flexível, que interage sem problemas com chips Bluetooth da Broadcom, CSR e outros. Os perfis suportados incluem: GAP, SDAP, SPP, GOEP, OPP, FTP, LAN, DUN, PAN, HSP, HFP e BIP [31].

A seguir são apresentadas algumas características e vantagens do BluRapport©:

* pilha Bluetooth completamente funcional - extenso suporte para protocolos e perfis, disponíveis como código fonte ou binário
* otimizada para ambientes embarcados e desktop - uso mínimo de recursos, execução robusta, alta performance
* portável - suporta vários chips Bluetooth, sistemas operacionais e processadores
* cuidadosamente testado - certificado pelo Bluetooth SIG e testado para interoperabilidade
* extensivamente implantado - disponível em milhares de dispositivos

A solução é certificada pelo Bluetooth SIG segundo a versão 2.0 + EDR da especificação, tendo sido testada para interoperabilidade entre dispositivos.

* + 1. **Bluetopia®**

Bluetopia® é a implementação da Stonestreet One ([http://www.stonestreetone.com](http://www.stonestreetone.com/)) das camadas superiores da pilha de protocolos Bluetooth, qualificada segundo a versão 2.1 + EDR da especificação Bluetooth. A solução facilita o desenvolvimento de aplicações fornecendo uma ferramenta de desenvolvimento de software robusta e flexível que implementa os protocolos e perfis Bluetooth acima da interface HCI. A API (Application Programming Interface) provê acesso a todos os protocolos e perfis das camadas mais altas e pode trabalhar diretamente com os módulos Bluetooth mais populares da Broadcom, CSR, TI e outras empresas do ramo. O Bluetopia® possui portabilidade para vários sistemas operacionais (Windows, Linux, QNX, Nucleus, uCOS, ThreadX, NetBSD, VxWorks, etc.) e processadores (ARM7/ARM9/ARM11, Intel x86, TI DSP 54xx/55xx, etc.) e é atualmente incluído em dispositivos de companhias como Motorola, Kodak, Honeywell, Garmin, VTech e Harris [33].

Os seguintes protocolos são suportados: HCI, L2CAP, SDP, RFCOMM, AVDTP (*Audio/Video Distribution Transport Protocol*), AVCTP (*Audio/Video Control Transport Protocol*), BNEP (Bluetooth Network Encapsulation Protocol) e OBEX. Quanto aos perfis disponíveis, temos: GAP, SPP, FAX, DUN, GOEP, OPP, FTP, HSP, HFP, HCRP, HID, SYNC, SAP, GAVDP, A2DP, BIP e PAN.

Dentre as funcionalidades oferecidas pela solução, podemos citar a possibilidade de habilitar ou desabilitar perfis e protocolos, uma interface de depuração (*debugging*) e uma interface API completamente documentada. O Bluetopia® é disponibilizado em código binário (precompilado para o sistema operacional) e código fonte completo.

* 1. **Avaliações**

Apesar da grande quantidade de soluções de pilhas Bluetooth para sistemas embarcados disponíveis no mercado, ainda existem algumas limitações relacionadas à sua implementação e seu uso.

Alguns aspectos de desenvolvimento estão intimamente relacionados entre si e são essenciais quando se trata do desenvolvimento para sistemas embarcados, como personalização, modularização, flexibilidade, manutenabilidade e extensibilidade. Devido à existente restrição de recursos nesses sistemas, é muito interessante que a implementação da pilha possa ser altamente personalizável, permitindo uma fácil escolha entre os diversos perfis e protocolos disponibilizados. Dessa forma, apenas os trechos necessários são utilizados, sem a adição de código que apenas aumenta o tamanho da pilha, sem prover funcionalidades úteis para a aplicação. A modularização da pilha de protocolos ajuda a prover essa personalização, garantindo também uma maior flexibilidade, manutenabilidade e extensibilidade.

Num projeto realizado no Centro de Estudos e Sistemas Avançados do Recife - C.E.S.A.R ([www.cesar.org.br](http://www.cesar.org.br)), existiram algumas limitações com uma implementação da pilha Bluetooth embarcada desenvolvida por um parceiro da Sony Ericsson. Essa solução era distribuída em código binário, não permitindo o acesso nem o acréscimo de novas funcionalidades ou correções das já existentes. Foram detectados também problemas relacionados ao limite de conexões simultâneas entre dispositivos Bluetooth, que eram inferiores tanto em relação ao número de dispositivos ativos quanto inativos (7 escravos ativos e 255 inativos, segundo a especificação Bluetooth). Outro porém foi o fato de a conexão ser inicializada apenas a partir de um PC para um dispositivo móvel, sem possibilidade de realizá-la no sentido inverso. Esses problemas foram de certa forma inesperados, dificultando a implementação da aplicação.

Num estudo realizado por Bonnet et al. em 2003 [35], são apresentadas algumas desvantagens do Bluetooth [34]. Em primeiro lugar, existe uma grande lentidão no descobrimento de dispositivos devido ao mecanismo de salto de frequências utilizado, sendo necessárias consideráveis quantidades de tempo (5 segundos, a partir da versão 1.2 da especificação Bluetooth [35]) e energia para que os dispositivos se encontrem, sem possibilidade de troca de dados durante esse período. Isso deteriora bastante o desempenho e aumenta o consumo, principalmente quando dispositivos precisam se desconectar e reconectar constantemente. Outro problema diz respeito à manutenção das conexões abertas ser bastante cara, resultando em um maior consumo de potência no estado de comunicação quando em comparação com outras tecnologias de rádio similares. Com relação à estrutura da pilha, o fato de ela ser projetada em camadas impede uma fina granularidade da sincronização de tempo e restringe otimizações entre as camadas. Quanto à topologia de rede, o Bluetooth é mais bem adaptado a conexões ponto-a-ponto. Se a aplicação necessita de uma rede de sensores composta de grandes grupos de nós, é aconselhável adotar uma abordagem com dois módulos de rádio de forma a construir uma rede multissalto (multi-hop). No caso da utilização de uma UART como interface de conexão no intuito de generalizar ou padronizar a comunicação externa com outros dispositivos, o valor máximo da vazão de dados (*throughput*) diminui de 90 kbps a 45 kbps. Para aplicações que necessitam um tráfego mais intenso isso se torna um problema notável, sendo necessário escolher entre a padronização da interface de comunicação e a alta taxa de transferência.

Devido à adoção da política de melhor esforço, comandos de solicitação e conexão (*inquiry* e *page*) não garantem 100% dos resultados esperados, pois são métodos não-determinísticos. Uma busca por dispositivos ou conexão pode não ser efetuada corretamente dependendo do estado em que cada nó encontra-se e das ações executadas por eles em determinado momento. Por exemplo, no caso de todos os nós estarem realizando umasolicitação exatamente ao mesmo tempo, nenhum deles conseguirá “enxergar” os outros, passando a falsa impressão de que não existem nós a serem descobertos na rede.

Os modos de baixo consumo de energia nos estados de Escuta (*Sniff*), Bloqueado (*Hold*) e Estacionado (*Park*) tem um limite máximo de 65440 slots de tempo (aproximadamente 40 segundos) para o período no qual um dispositivo pode permanecer dormindo (*sleep*). Num cenário em que dispositivos podem permanecer inativos por minutos ou até por horas, isso é bastante indesejável, pois eles seriam acordados em períodos de tempos bem mais frequentes do que o ideal. Seria interessante então que as aplicações pudessem ter alguma forma de acesso às informações de tempo gerenciadas pelas camadas de baixo nível, como a *Baseband* e as relacionadas ao gerenciamento de links (para o reconhecimento de transmissão de mensagens ou informações sobre retransmissões).

Em [36] a funcionalidade Bluetooth foi implementada numa aplicação de rede de sensores utilizando uma solução de mercado (*off-the-shelf*) desenvolvida pela Stollmann Entwicklungs - und Vertriebs-GmbH ([http://www.stollmann.de](http://www.stollmann.de/)). O dispositivo provê um conjunto de comandos para controle das conexões Bluetooth através de uma interface serial RS-232. Isso permite uma rápida prototipação, sem a necessidade do desenvolvimento da pilha Bluetooth ajustável ao microcontrolador. No entanto, existem alguns inconvenientes: os comandos não permitem um controle total dos protocolos e configurações Bluetooth; o protocolo SDP não é suportado, dificultando a descoberta de serviços disponíveis na rede; apenas o perfil de porta serial (SPP) é suportado; e o tamanho do dispositivo de cada nó não é negligenciável (5 cm x 5 cm x 5 cm), dificultando sua implantação nos nós sensores da rede.

Outra análise que pode ser feita diz respeito à topologia da rede Bluetooth. Devido à necessidade de o mestre coordenar toda a piconet, ele termina tornando-se um ponto crítico de falha (*single point of failure*). Isso significa que caso o mestre tenha algum problema e desconecte-se da rede, toda a piconet deverá ser restabelecida, com a nomeação de um novo dispositivo mestre. Além disso, toda a comunicação da piconet deve passar pelo mestre, de forma que ele necessita de um bom controle de mensagens, roteamento, largura de banda e capacidade de tomada de decisão. Em resumo, o dispositivo mestre precisa de mais poder de computação, comunicação e uma maior tolerância a falhas que os dispositivos escravos. Essa necessidade do tráfego passar obrigatoriamente pelo mestre acrescenta ainda um overhead caso dois dispositivos escravos queiram comunicar-se.

Em [32] Jasemian & Arendt-Nielsen implementaram a pilha Bluetooth embarcada em um sistema de telemedicina para monitoramento remoto de pacientes, verificando as limitações e problemas encontrados durante a implementação e realizando análises de performance dos resultados obtidos. Os testes técnicos realizados por eles mostraram que o módulo Bluetooth pôde armazenar apenas 2500 bytes nos buffers, relativos à quantidade de memória RAM disponível, o que torna o controle de fluxo de dados não confiável. Nesse experimento, ao utilizar-se o protocolo TCP/IP como interface para conectar o módulo Bluetooth a uma rede GPRS para transmissão de dados, ele pôde ainda enviar continuamente ao servidor um pacote TCP/IP a uma taxa de 800 bytes por segundo. Com uma taxa de dados de 7,5 kbps, houve uma perda de pacotes de 0,5%, considerando o sistema online durante 4 semanas, 24 horas por dia. Com um controle de fluxo mais inteligente entre o manipulador de dados do módulo Bluetooth e a pilha TCP/IP, o buffer TCP/IP se esgotou, resultando em um deadlock na transmissão dos dados. O problema nesse caso é que o protocolo TCP/IP no módulo Bluetooth não é um protocolo de “streaming”, ou seja, os requisitos de tempo nem sempre garantem a taxa de transmissão desejada, tornando a confiabilidade dependente do tráfego de dados momentâneo.

* 1. **Sugestões de melhorias**

Em vista das análises realizadas anteriormente, existem alguns tópicos nos quais podem ser realizadas algumas melhorias.

Um deles é o tempo necessário para descobrir os dispositivos da piconet através do comando de solicitação (*inquiry*). Um estudo realizado por Woodings et al. em [38] sugere a realização da descoberta de dispositivos da rede através de uma conexão IrDA. Essa técnica reduziria bastante o tempo necessário para realizar uma conexão, mas em compensação, limitaria as condições e número de dispositivos que poderiam ser encontrados caso fosse utilizado o método especificado pelo Bluetooth. O ideal nesse caso seria encontrar alguma técnica que permitisse o envio de mensagens de menor duração, atenuando assim o tempo de descoberta de dispositivos e mantendo as vantagens de mobilidade e escalabilidade oferecidas pelo Bluetooth.

Outro ponto que ainda pode ser melhorado é o consumo de energia. Apesar de já apresentar níveis baixos de consumo, o Bluetooth ainda consome um pouco mais que outras tecnologias, como o ZigBee. Através dos estados de baixo consumo (Estacionado, Escuta ou Bloqueado) e de um controle minucioso entre os tempos de permanência em cada estado pode ser possível economizar ainda mais energia, possibilitando assim que o Bluetooth seja amplamente utilizado em situações que exigem uma longa vida útil da bateria e não contam com fácil fornecimento de energia, como as redes de sensores, por exemplo.

Existem casos também em que é necessário uma alta taxa de transferência de dados, seja para garantir restrições de tempo ou para otimizar o funcionamento da solução. Sistemas de tempo real comumente apresentam situações em que é necessária uma alta vazão de dados que supra os seus requisitos. O problema nesse caso seria relativo à configuração e controle da rede, e não à forma de transmissão através das ondas de rádio, uma vez que outras tecnologias que também utilizam esse meio de comunicação, como a versão 802.11n do Wi-Fi, chegam a transmitir dados na ordem dos 450 Mbps.

A versão 4.0 da especificação Bluetooth já define duas subespecificações relativas ao baixo consumo de energia e à alta velocidade de transferência de dados, porém, devido ao pouco tempo desde a sua liberação (Dezembro de 2009), ainda são raras as soluções que se ajustam a esse padrão.

Atualmente, a estrutura feita em camadas, apesar de abstrair muitos conceitos das camadas de baixo nível para as camadas superiores, que dificultaria o desenvolvimento da aplicação, torna complicada e às vezes impossível a otimização entre elas e uma granularidade mais fina da sincronização de tempo. Um estudo de como essas camadas poderiam ser mantidas oferecendo uma maior flexibilidade quanto ao acesso às informações de controle e sincronização poderia facilitar muito a possibilidade de realização de otimizações para que a aplicação fosse cada vez mais eficiente e capaz de respeitar requisitos ainda mais restritivos.

A implementação de uma interface padrão que permitisse o usuário adicionar dinamicamente (ou seja, sem a necessidade de intervenção do desenvolvedor da pilha) módulos externos contendo uma determinada funcionalidade ou perfil ajudaria muito no sentido de garantir a modularização da pilha, favorecendo assim a personalização, flexibilidade, extensibilidade e manutenabilidade da mesma.

1. **Conclusões e trabalhos futuros**

Durante a pesquisa realizada para este trabalho foram encontrados alguns artigos e resultados que expunham limitações da própria tecnologia Bluetooth ou de certas soluções específicas quando submetidas a ambientes com restrição de recursos. Porém foi relativamente difícil encontrar fontes que relatassem problemas relativos à implementação propriamente dita da pilha Bluetooth nesses sistemas. Além disso, a maior parte dos estudos encontrados datavam do início da década de 2000 (primeiros 5 anos), o que não contempla diversas atualizações e modificações realizadas com o avanço das versões da especificação disponibilizada pelo Bluetooth SIG (a versão 2.1 + EDR foi lançada apenas em Março de 2007; a 3.0 + HS (*High Speed*), em Abril de 2009; e a 4.0, em Dezembro de 2009).

O desenvolvimento da pilha de protocolos Bluetooth para sistemas embarcados apresenta diversas restrições relativas aos requisitos apresentados por estes sistemas. Entre elas estão a pequena quantidade de memória disponível, o alto tempo de procura por dispositivos e conexão e o gasto ainda considerável de energia em alguns estados.

Muitas das soluções disponíveis já vêm integrada com um módulo de hardware, que às vezes possui um tamanho considerável, dificultando assim a sua implementação em sistemas que tenham limites físicos, como por exemplo, redes de sensores. Além disso, parte delas distribui códigos binários que não possibilitam modificações ou inclusões de funcionalidades, restringindo as alternativas possíveis de pilhas Bluetooth que se encaixam nos requisitos do sistema a ser desenvolvido.

A especificação Bluetooth desenvolvida pelo Bluetooth SIG encontra-se em constante desenvolvimento e tem adicionado diversas melhorias relacionadas ao consumo de energia (*Bluetooth Low Energy Technology*) e velocidade (*Bluetooth* *High Speed Technology*), mas, devido ao pouco tempo desde que elas foram anunciadas, ainda é raro encontrar soluções que estão de acordo com a última versão dessa especificação (versão 4.0).

Diversos estudos ainda podem ser realizados visando alguma mudança arquitetural que permita um maior controle ou acesso das informações entre as camadas, possibilitando uma maior comunicação e facilitando assim otimizações e uma maior precisão na sincronização de tempo. Um avanço muito interessante seria o desenvolvimento modularizado de perfis, que pudessem ser adicionados separadamente de modo a fornecer de forma incremental as funcionalidades desejadas pelo usuário. Atualmente as soluções fornecem diversos perfis disponíveis, mas cada uma possui sua lista fixa de perfis suportados, de modo que não é possível adicionar um novo perfil caso seja necessário sem que haja uma intervenção dos fabricantes e/ou desenvolvedores da pilha. Nesse caso, seria necessária uma interface padrão que permitisse a conexão de perfis através de módulos separados.

**Referências**

[1] **redes de comunicação de dados**. Em Infopédia. Porto: Porto Editora, 2003-2010. <[http://www.infopedia.pt/$redes-de-comunicacao-de-dados](http://www.infopedia.pt/%24redes-de-comunicacao-de-dados)>.

[2] **Página oficial da Wi-Fi Alliance**. <[www.wi-fi.org](http://www.wi-fi.org)>

[3] ALECRIM, E. **Tecnologia Wi-Fi (IEEE 802.11)**. InfoWester. <<http://www.infowester.com/wifi.php>>

[4] **Página oficial da ZigBee Alliance**. <[www.zigbee.org](http://www.zigbee.org)>

[5] PINHEIRO, J. M. S. **As Redes com ZigBee**. Eletronica.org. <<http://www2.eletronica.org/artigos/eletronica-digital/as-redes-com-zigbee/>>

[6] **ZigBee.** **Low-cost, low-power, wireless networking for device monitoring and control**. Digi International. <<http://www.digi.com/technology/rf-articles/wireless-zigbee.jsp>>

[7] **Página oficial da IrDA**. <[www.irda.org](http://www.irda.org)>

[8] **The Official Bluetooth Technology Info Site**. <[www.bluetooth.com](http://www.bluetooth.com)>

 [9] McDERMOTT-WELLS, P. **What is Bluetooth?** Em: IEEE Potencials, Volume 23, Edição 5, Dezembro 2004 – Janeiro 2005, pp. 33-35.

[10] BILLO, E. A. **Uma pilha de protocolos Bluetooth adaptável à aplicação**. Universidade Federal de Santa Catarina. Fevereiro 2003

[11] RATHI, S. **Blue Tooth Protocol Architecture. <**<http://www.es2.be/magazine/00q4/2000q4_p028.pdf>>

[12] WANG. H. **Overview of Bluetooth Technology**. 03/07/2001. <<http://www.pori.tut.fi/~mm/BT/Bluetooth_Overview.pdf>>

[13] BHATT, P. **The Bluetooth Technology**. Tata Consultancy Services. 07/2000. <<http://www.pori.tut.fi/~mm/BT/Bluetooth_Technology.pdf>>

[14] CHASE, O. **Sistemas Embarcados.** SBAJovem 2010. <<http://sbajovem.org/publicacoes/SistemasEmbarcados.pdf>>

[15] CUNHA, A. F. **O que são sistemas embarcados?** Revista Saber Eletrônica, 414, Editora Saber, Brasil, 2007. < <http://www.techtraining.eng.br/conteudo/ARTIGO-SIST-EMB.pdf>>

[16] **The Official Bluetooth SIG Member Website**. <<https://www.bluetooth.org/About/bluetooth_sig.htm>>

[17] QUEIROZ, L. E. C. **Protocolo de Redes Bluetooth**. Distribuição e Integração de Sistemas. Barcarena. 2008.

[18] PAYER, M. **Implementation of a Bluetooth Stack for BTnodes and Nut/OS Version 0.9**. Institute for Pervasive Computing, Distributed Systems Group, ETH Zurich. <<http://nebelwelt.net/download/ETH/btnodes_semesterthesis.pdf>>

[19] SILVA, D. F. **Sistema de comunicação Bluetooth utilizando microcontrolador**. dez/2009 <[http://dsc.upe.br/~tcc/20092/TCC\_final\_Davidson.pdf](http://dsc.upe.br/~tcc/20092/TCC_final_Davidson.pdf%20) >

[20] **palowireless. Wireless Resource Center**. <<http://www.palowireless.com/bluetooth/>>

[21] **Bluetooth Specification Version 4.0**. <<http://www.bluetooth.com/Specification%20Documents/Core_V40.zip>>

[22] KOBAYASHI, C. Y. **A Tecnologia Bluetooth e Aplicações**. IME – USP. 2004 < <http://grenoble.ime.usp.br/movel/monografia_bluetooth.pdf> >

[23] **Telephony Control Protocol Specification. TCS Binary**. Bluetooth Specification Version 1.1. <<http://www.bluetooth.com/SiteCollectionDocuments/TCSBinary.pdf>>

[24] **Object Exchange**. The Official Bluetooth Technology Info Site. <<http://www.bluetooth.com/English/Technology/Works/pages/obex.aspx>>

[25] **Funcionamento do Bluetooth**. Kioskea.net <<http://pt.kioskea.net/contents/bluetooth/bluetooth-fonctionnement.php3> >

[26] Código fonte da pilha lwBT. <https://github.com/lwalkera/lwBT>

[27] **IAR Embedded Bluetooth Protocol Stack.** <<http://www.microvision.co.kr/bluetooth/pdf/makeapp_cadalogue.pdf>>

[28] **Bluetooth Sandwich Module.** <<http://www.beecon.de/sandwichmodule_bluetooth.html>>

[29] **AVE-Blue - Bluetooth Protocol Stack for Embedded Applications**. <<http://www.access-company.com/PDF/EmbeddedModules/12_2006_AVE-Blue.pdf>>

[30] **Introduction of IVT - Bluetooth Host Protocol Stack and Profiles - BlueLetTM**. <<http://www.ivtcorporation.com/download/downloadfile/IVT_Stack_April_2009_General.pdf>>

[31] **BluRapport©**. <<http://www.dsr-wireless.com/bluetooth>>

[32] JASEMIAN, Y., ARENDT-NIELSEN, L. **Design and implementation of a telemedicine system using Bluetooth protocol and GSM/GPRS network, for real time remote patient monitoring.** 2005. Em: Technology and Health Care. 13, p. 199-219.

[33] **Bluetooth Protocol Stack – Bluetopia**. Stonestreet One. <<http://www.stonestreetone.com/bluetopia.cfm>>

[34] ELIASSON, J., LINDGREN, P., DELSING, J. **A Bluetooth-based Sensor Node for Low Power Ad Hoc Networks**. Em: Journal of Computers, Vol. 3, No. 5, Maio 2008

[35] BONNET, P., BEAUFOUR, A., DYDENSBORG, M.B., LEOPOLD, M. **Bluetooth-Based Sensor Networks**. Em: ACM SIGMOD Record, Vol. 32, No. 4, Dezembro 2003

[36] KRCO, S. **Bluetooth based wireless sensor networks implementation issues and solutions**. Em: Proceedings of Telfor 2002, Belgrade, Serbia and Montenegro (2002) <<http://www.telfor.rs/telfor2002/radovi/4-19.pdf>>

[37] [PATIL, B.](http://flylib.com/search/en/author/Patil%2BB/), SAIFULLAH[, Y.](http://flylib.com/search/en/author/Saifullah%2BY/), FACCIN, S. Books online: **IP in Wireless Networks/12.6 Bluetooth Profiles**. <<http://flylib.com/books/en/4.215.1.118/1/>>

[38] WOODINGS, R. W., JOOS, D. D., CLIFTON, T., KNUTSON, C. D. **Rapid Heterogeneous Connection Establishment: Accelerating Bluetooth Inquiry Using IrDA**. Em: Wireless Communications and Networking Conference, 2002. <<http://mcl.cs.byu.edu/downloads/IrDA_Assisted_BT_Discovery.pdf>>