Desenvolvimento de Sistemas Pervasivos com Bluetooth e Linux/Python

Henry M. M. Bilby[1], Antonio Carvalho Jr[1], Samuel F. Oliveira[1], Nilson Silva[1], Jonatas Isvi[1], André L.C. Portela[1], Carlos M.S. Figueiredo[1]

[1]Projeto Zagaia - Mobile Linux Development Center (Fucapi/INDT)
Av. Gov. Danilo de Mattos Areosa, 381. Lab A01 – Manaus – AM.

{h2mbilby, brankinhu, sf.oliveira, fergus.mao, jonatas.nona, portela.eng, mauriciofigueiredo}@gmail.com

Abstract. *The concept of pervasive computing is in our lives and is considered the power of propulsion for the next technological advances in mobile devices. This paper shows the basic concept of pervasive computing, bluetooth tecnology, some considerations about the applications programming interfaces (API's) to use this technology in GNU/Linux operational system with python programming language and case studies, of authors, in development of two dedicated softwares for mobile devices using the API's approached.*

Resumo. *O conceito de Computação Pervasiva adentra cada vez mais em nossas vidas e está sendo considerado a força propulsora para os próximos avanços tecnológicos em dispositivos móveis. Este artigo apresenta os conceitos básicos de computação pervasiva, da tecnologia BlueTooth, bem como considerações sobre algumas bibliotecas de programação(API's) disponíveis para utilização desta tecnologia, em sistemas operacionais GNU/Linux com a linguagem de programação Python e estudos de caso, dos autores, no desenvolvimento de dois softwares voltados para dispositivos móveis utilizando as API's abordadas.*

1. Introdução

Computação Pervasiva é uma área de pesquisa que está sendo considerada a terceira onda da computação [Jansen et al 2005]. Originou-se da proposta de Mark Weisen chamada Computação Ubíqua (*Ubiquitous Computing*) [Weiser 1991], que se baseia na idéia de um ambiente computacional inteligente onde um conjunto de aplicativos interagem com os seres humanos de forma simbiótica e transparente. Esta transparência está diretamente relacionada à mobilidade, pois para que ela exista a computação não pode impor limitações ao comportamento do usuário, ou seja, não deve cercear sua capacidade natural de movimentação/deslocamento físico.

Como o cenário pervasivo prevê uma mobilidade física, as redes sem fio assumiram fundamental importância no contexto e, a partir de então, a necessidade crescente de

transmissão direta de dados utilizando esse tipo de conexão contribuiu bastante para a popularização dos dispositivos de transmissão que utilizam ondas de rádio, como o *Bluetooth* [Huang e Rudolph 2007]. Entretanto, esta transmissão exige meios de comunicação e protocolos confiáveis, aplicações com métodos de operação bem implementados para o dispositivo que irá realizar a comunicação, tecnologia de baixo custo e que seja ideal para sistemas móveis à curta distância; é nesse contexto que o *Bluetooth* se enquadra no conceito de Computação Pervasiva. Este trabalho apresenta aspectos desta tecnologia, ferramentas de software livre, neste caso *Linux* e *Python*, para desenvolvimento de aplicações pervasivas e móveis utilizando bluetooth e dois estudos de caso utilizando softwares desenvolvidos no Mobile Linux Development Center [Zagaia Project 2008].

2. Fundamentos sobre o BlueTooth

O Bluetooth é uma especificação industrial para áreas de redes pessoais sem fio (Wireless personal area network – PAN), e provê uma maneira de conectar e trocar informações entre dispositivos através de uma frequência de rádio de curto alcance. *Segundo* [Huang e Rudolph 2007] *Bluetooth* é um padrão global de comunicação sem fio e de baixo consumo de energia, baseado num *microchip* com transmissor de rádio embutido, que permite a transmissão de dados entre dispositivos compatíveis com a tecnologia. Para isso, uma combinação de *hardware* e *software* é utilizada para permitir que essa comunicação ocorra entre os mais diferentes tipos de aparelhos. As conexões são realizadas instantaneamente sem um único centímetro de cabo e a transmissão de dados é feita através de uma radiofrequência de 2.4 GHz que não precisa de licença, permitindo que um dispositivo detecte o outro independente de suas posições, desde que estejam dentro do limite de proximidade.

A arquitetura do Bluetooth aborda especificações de protocolos das diversas camadas, permitindo assim o desenvolvimento de soluções completas e com interoperabilidade, como mostrado na figura 1:

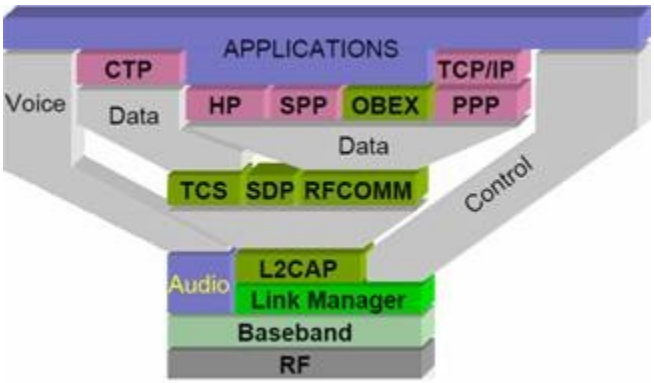

Figura 1. Interação de protocolos na arquitetura Bluetooth. (Fonte:Peters 2008)

Nas camadas mais altas da arquitetura podemos notar a existência de vários perfis de aplicações, que podem ser implementados por determinados dispositivos dando a eles

uma função padrão. No entanto, novas aplicações podem ser desenvolvidas com os principais protocolos descritos a seguir:

- ***Logical Link Control Adaptation Protocol* (L2CAP):** Fornece serviços de conexão de dados com e sem conexão para as camadas superiores de protocolo. Executa funções de multiplexação, segmentação, controle de fluxo e de erro e gerenciamento de grupo. O L2CAP é utilizado para multiplexar canais lógicos em um único enlace físico.
- **RFCOMM:** Canal criado sobre o L2CAP que emula uma interface serial. Desta forma, pode-se transmitir dados entre os dispositivos *Bluetooth* através de uma interface semelhante, por exemplo, a "/dev/ttyS1".
- **OBEX:** Protocolo usado para transferência de dados entre dispositivos móveis. Possui perfil chamado Object Push que é destinado a transferências rápidas de mídias, sem a necessidade de pareamento dos dispositivos.

3. Desenvolvimento com Bluetooth

Para o desenvolvimento de aplicações *Bluetooth*, dispõe-se de diferentes bibliotecas de funções e ferramentas, dependendo do sistema operacional e da linguagem de programação adotada. Uma plataforma de destaque consiste no uso do *GNU/Linux* e *Python*, que vêm se tornando comuns em dispositivos móveis como software embarcado. Um exemplo disso é a plataforma Maemo, que é uma customização da distribuição GNU/Linux Debian, que foi desenvolvida pela Nokia, para ser embarcada nos aparelhos da série N8*0. A seguir são apresentadas as principais API's bluetooth disponíveis em ambiente GNU/Linux, inclusive para o Maemo.

3.1 Bluez

A BlueZ [BlueZ 2000] é uma API (*Application-Programming Interface*) que fornece suporte para camadas e protocolos do núcleo *Bluetooth* para ambientes *Linux*. Dentre suas principais características, a flexibilidade e a eficiência são garantidas através de uma implementação modular, processamento de dados multithread, multiprocessamento simétrico seguro, abstração real do hardware e interface de socket padrão para todas as camadas. Possui compatibilidade com a maioria das distribuições linux como Debian GNU/ Linux, Red Hat Linux e OpenSUSE Linux, trabalhando perfeitamente em várias arquiteturas de hardware, e está licenciado sobre a licença GNU/GPL2.

3.2 PyBluez

Estrutura de dados em outras linguagens de programação tendem a obscurecer as operações básicas, *Python* não sofre desta desvantagem, pois tem como pilares a clareza e a rapidez no desenvolvimento do código.

Neste contexto surge uma biblioteca que contém os bindings para Python de bibliotecas *Bluetooth* escritas em linguagem C da API Bluez e licenciada sob GNU/GPL2.

A PyBluez permite um rápido desenvolvimento através de classes e de funções

encapsuladas, tanto para ambiente Windows XP (utilizando a pilha Bluetooth da Microsoft ou da Widcomm) quanto GNU/Linux (utilizando a pilha Bluez), suportando os principais protocolos de comunicação, RFCOMM, L2CAP, SCO e HCI. Em virtude dessas características a PyBluez [PyBluez 2008] foi escolhida pelos autores como biblioteca base no desenvolvimento dos softwares BlueHi e PC Remote apresentados no capítulo 4.

3.3 LightBlue

A LightBlue [LightBlue 2008] é um binding multi-plataforma *Bluetooth* para Python, oferecendo um acesso simples as principais operações, tais quais: descoberta de aparelhos com *Bluetooth* ativo e dos serviços por ele oferecidos, interface padrão de *sockets* utilizando os protocolos RFCOMM e L2CAP, envio e recepção de arquivos via protocolo OBEX. Ela possui dependência dos bindings da API PyBluez e está licenciada sob GNU/GPL3.

4. Estudos de Caso

Os estudos de caso abaixo foram realizados utilizando softwares desenvolvidos no Mobile Linux Development Center [Zagaia Project 2008].

4.1 BlueHI

Plugin de compartilhamento de mídias para o Canola Media Center. O BlueHI [BlueHI 2009] foi projetado seguindo os padrões de arquitetura e interface definidos para o Canola [Canola 2008], com o principal objetivo de compartilhar arquivos de música, vídeo e imagem, entre diversos dispositivos móveis, como celulares, handhelds e internet tablets.

O BlueHI está licenciado sob a licença GPL3, é baseado no padrão MVC de 3 camadas e foi desenvolvido com a linguagem Python. Dentre as classes do plugin, a BluetoothManager utiliza a API Lightblue e define os métodos que utilizam o protocolo Bluetooth, entre os quais estão:

- O método abaixo "*def search_devices(self):"* detecta os dispositivos móveis que estão com o bluetooth ativado, através da função *lightblue.finddevices()*, retornando uma lista contendo tuplas que possuem o mac-address do bluetooth e o nome do aparelho;

```
def search_devices(self):
    devices=lightblue.finddevices();
    return devices;
```

Figura 2. Método para procura de dispositivos móveis.

- O método "*def connect(self, address):"* tem o objetivo de conectar com o aparelho informando o mac-address. Primeiro é feita uma busca no aparelho pelo serviço OBEX Object Push onde é gravada a porta que será utilizada para estabelecer a conexão, através de um client pelo protocolo OBEX:

```
def connect(self, address):
    services=lightblue.findservices(address)
```

```
for i in services:
    print i
    if i[2] == "OBEX Object Push":
    port = i[1]
client = lightblue.obex.OBEXClient(address, port)
client.connect()
```

Figura 3. Método para conexão com o dispositivo móvel.

- *def send_file(self, path):* é o método que envia o arquivo para o client, através do protocolo OBEX. Para isso é utilizado o método *put* do objeto *client,* nele devemos passar o nome e tamanho do arquivo e o objeto *file*.

```
put_response = client.put({"name": file_name, "length": file_length}, file(path))
```

Figura 4. Método para envio de arquivo via lightblue.

4.2 PC Remote

Aplicativo que provê a mobilidade dos serviços de uma estação de trabalho fixa, o computador de mesa, utilizando o dispositivo móvel Nokia N8*0. A interligação das estações é disponibilizada através da execução de dois módulos distintos: o cliente e o servidor.

O PC Remote [PcRemote 2009] está licenciado sob a licença GPL3 e foi totalmente desenvolvido com a linguagem *Python* e projetado para ser executado no ambiente *Linux*, sendo que o servidor roda sobre o *Ubuntu*; e o cliente, *Maemo*. O limite de distância entre cliente e servidor é de 100m, visto que a conexão entre eles é feita via *Bluetooth* (tecnologia já citada antes como sendo eficiente e de baixo custo) e utilizando o protocolo L2CAP. Para tal foi desenvolvida a classe *BlueToothConnectionManager* que se utiliza dos recursos da biblioteca *PyBlueZ.* Após a conexão ser estabelecida, a transmissão de comandos e a coordenação das ações entre os módulos é feita pura e simplesmente através de troca de mensagens de texto, o que remete à definição de Sistemas Distribuidos feita por [Birmam 1996], a qual se refere como "*um conjunto de programas de computador executando em um ou mais computadores, e coordenando suas ações por meio de troca de mensagens*". O cliente recebe do usuário a informação de qual serviço ele quer iniciar, este processa e manda uma mensagem para o servidor, que por sua vez, interpreta e inicia o serviço, se o mesmo estiver disponível. A partir de então, o processo de troca de mensagens é constante até que o usuário resolva interromper o serviço (o fluxo dessa informação é demonstrado na figura 5; e parte do código que demonstra a troca de mensagem, na figura 6). Os serviços implementados até o momento da elaboração deste artigo são: *Tablet* (controle remoto de mouse e teclado), *Slide Show* (utilizado para controlar apresentações) e *Player* (controle dos aplicativos mais comuns do gênero, tais como *Amarok* e *RythmBox*).

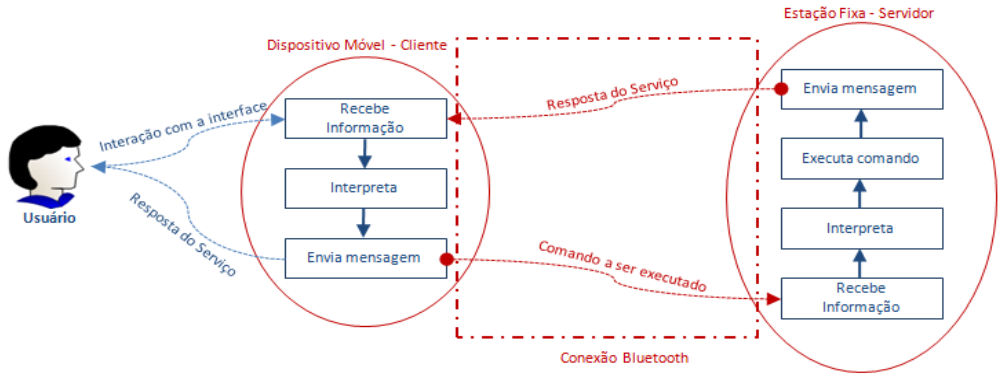

Figura 5. Fluxo de informação no PC Remote .

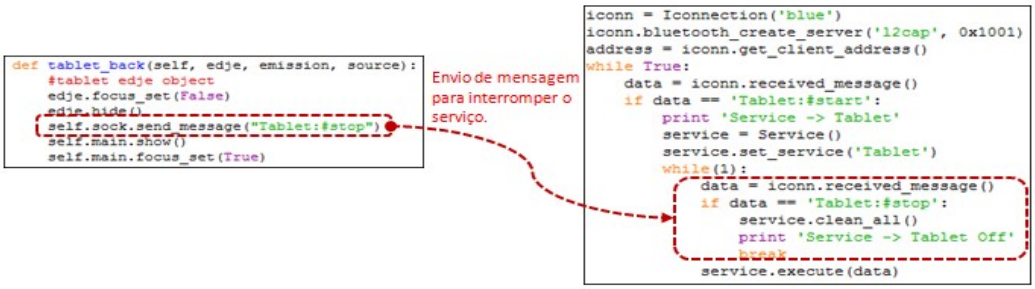

Figura 6. Parte do código que demonstra a troca de mensagens entre cliente e servidor.

O projeto inicial estabelecia a conexão de um cliente para cada servidor, mas testes posteriores mostraram a viabilidade de ter-se muitas conexões para cada servidor e que cada cliente pode executar serviços diferentes, o que vai de encontro ao conceito de computação pervasiva que define que cada usuário seja identificado e diferenciado pelo ambiente para que este se adapte as suas necessidades.

5. Considerações Finais

De uma maneira geral pode-se considerar que a utilização de *Bluetooth* para a Computação Pervasiva, em um ponto de vista econômico e tecnológico, é viável se levarmos em consideração os avanços de dispositivos móveis e a necessidade de automação de tarefas, sejam elas domésticas ou industriais. Outra questão que favorece a utilização do *Bluetooth* é a popularização da tecnologia, que está disponível na maioria dos dispositivos móveis, eletrodomésticos e carros. Particularmente este trabalho mostrou a facilidade de desenvolvimento em plataformas abertas como linux/python, abordando dois estudos de caso, permitindo a popularização de tais soluções, visto que ambos estudos de caso estão licenciados sob GNU/GPL3.

6. Referências

Birman, K. Building secure and reliable network applications, Manning Publications Co, Geenwich, (1996).

Bluez, Project. (2000), http://www.bluez.org/, acessado em 25 de Março de 2009.
BlueHI, Project. (2009), http://portal.fucapi.edu.br/nepcomp/zagaia/bluehi.html
Canola. (2008), http://openbossa.indt.org/canola/, acessado em 31 de Março de 2009
Huang, Albert S e Rudolph, Larry (2007). Bluetooth essential for programmers. Cambridge, New York.
Jansen, E. et al (2005) "A Programming Model for Pervasive Spaces" , International Conference on Service-Oriented Computing" , Netherlands, dec.
Lightblue, Project. (2008), http://lightblue.sourceforge.net/, acessado em 25 de Março de 2009
Lucistnik, Pav. "FreeBSD HandBook".(2003),http://www.openit.com.br/freebsd-hb/network-bluetooth.html , acessado em 04 de Abril de 2009.
PcRemote, Project. (2009), http://portal.fucapi.edu.br/nepcomp/zagaia/pcremote.html
Peters, Eduardo. (2008),"Bluetooth" http://www2.eletronica.org/artigos/eletronica-digital/bluetooth, acessado em 30 de Março de 2009
PyBluez, Project. (2008), http://org.csail.mit.edu/pybluez/, acessado em 25 de Abril de 2009
Sauter, Martin. (2006) Communication Systems for the Mobile Information Society. Wyley, England. 1st Edition.
Weiser, Mark. (1991) "The Computer for the Twenty-First Century", Scientific American, pp. 94-10, http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html, September , acessado em 24 de Março de 2009.
Zagaia, Project. (2008), http://portal.fucapi.edu.br/nepcomp/zagaia