

Uma API para Monitoração e Análise de Redes 802.11 e Bluetooth

Ricardo A. Rabelo Oliveira*, Rainer P. Couto*, Antonio A.F. Loureiro

¹Departamento de Ciência da Computação
Universidade Federal de Minas Gerais

(rabelo, rainerpc, loureiro) at dcc.ufmg.br

Resumo. *A redução do tempo de desenvolvimento é uma das características mais importantes na criação de software. Uma das estratégias com essa finalidade é o uso de uma interface de programação, API, Application Programming Interface, cujo objetivo é descrever como ocorre a interação entre as diversas partes de um sistema. O objetivo deste trabalho é desenvolver uma API para monitoração e análise de duas das principais tecnologias de redes sem fio usadas na computação móvel: as WLANs, com o IEEE 802.11, e as WPANs, com o Bluetooth. A API proposta acessa as informações das interfaces de um dispositivo, assim como possibilita a sua reconfiguração de acordo com as necessidades da aplicação. Neste artigo são apresentados a especificação, projeto, implementação e avaliação desta API.*

Abstract. *Companies today expend a high effort in reducing software development time. An important strategy employed by programmers with this aim is the use of APIs, Application Programming Interface, which are sets of protocols, routines and tools for building software applications. In this work we present a API for monitoring and analysis of two of the most important wireless network technologies: 802.11 for WLANs (Wireless Local Area Networks), and Bluetooth for WPANs (Wireless Personal Area Networks). The proposed API can access information in both communication interfaces and can configure them in order to attend specific application needs.*

1. Introdução

A redução do tempo de desenvolvimento é um dos objetivos mais importantes no processo de criação de softwares. Com essa finalidade, diversas estratégias de planejamento e implementação vem sendo continuamente propostas e utilizadas na prática. Uma dessas estratégias é o uso de uma interface de programação ou API (*Application Programming Interface*), que consiste de um conjunto de rotinas, protocolos e ferramentas que disponibiliza todos os elementos básicos necessários à programação de um sistema. Com o uso de uma API, o papel do programador se reduz à interligação dos componentes básicos, o que facilita a aprendizagem e reduz o tempo de programação de novos softwares.

No ambiente de computação móvel existe uma grande diversidade de dispositivos e protocolos de comunicação que demandam uma abordagem de programação diferenciada daquela adotada nas redes fixas. Fatores como escassez de recursos, propriedades do meio de comunicação, mobilidade e interoperabilidade também devem ser considerados durante o desenvolvimento de um novo software. Nesse caso, o uso de uma API para acesso aos recursos do ambiente facilita esse processo, abstraindo os detalhes de implementação de componentes básicos do sistema e fornecendo uma interface única para programação.

As diferentes tecnologias para a comunicação sem fio se distinguem por suas capacidades de transmissão, qualidade de recepção e a maneira como utilizam os recursos disponíveis. Estas

*Financiado pelo CNPq.

interfaces sem fio possuem diversas aplicações e objetivos, indo desde comunicações celulares de longo alcance até a comunicação entre sensores. O enfoque deste trabalho cobre duas das principais tecnologias de redes sem fio usadas na computação móvel: as *Wireless Personal Area Networks*, WPANs [1], redes de acesso pessoal, e as *Wireless Local Area Networks*, WLANs [2], as redes de acesso local. Os dois tipos de redes foram desenvolvidos com objetivos específicos e atendem a diferentes tipos de necessidades. As WPANs são redes sem fio onde os elementos comunicantes da rede estão próximos do usuário, sendo muitos deles de uso pessoal. Uma das principais características deste tipo de dispositivos é possuírem baixa capacidade computacional, pouca fonte de energia e serem de uma implementação mais simples que WLANs. Além disso, as WPANs são caracterizadas por terem menores capacidades de transmissão, menor mobilidade entre os dispositivos e menores distâncias. Um dos cenários mais comuns previstos nas WPANs são as redes estabelecidas entre os dispositivos de um usuário, como celulares, PDAs, relógios e fones de ouvido, permitindo um acesso condizente com as características destes, tanto de seus dados como os diferentes tipos de serviços oferecidos. As WLANs são conhecidas como a extensão das redes de acesso local. A maior motivação para a sua criação foi permitir a mobilidade de usuários das redes conhecidas como LANs, onde os usuários da rede possam mover com certas restrições e acessar as redes fixas e as infra-estruturas existentes, sem necessidade de modificações nas instalações. Outro fator foi a possibilidade de diversos dispositivos poderem ter um acesso também irrestrito, sem a necessidade de cabos ou uma infra-estrutura de suporte, criando redes *ad hoc*. Das diversas tecnologias sem fio mais difundidas, este trabalho foca duas que mais se destacam, por serem financiadas por várias empresas de grande porte e terem finalidades distintas: Wi-Fi [2], do padrão IEEE 802.11, é voltado para as *Wireless LANs*, e *Bluetooth* [3], das WPANs.

Este trabalho tem como objetivo especificar, projetar, implementar e avaliar uma API para utilizar os recursos das duas tecnologias de maneira integrada, permitindo com que as aplicações disponham de acesso transparente ao controle das duas interfaces sem fio. A API proposta acessa as informações de seus dispositivos e de outros que estejam comunicando, assim como possibilita a reconfiguração do dispositivo de acordo com as necessidades da aplicação.

2. Trabalhos Relacionados

A adaptabilidade das aplicações para estes dois padrões de comunicação são fontes de diversos estudos. Em [4, 5], por exemplo, analisam as propriedades do canal Wi-Fi e propõem modelos de adaptação para alguns serviços como vídeo. Por contemplarem cenários complementares, os protocolos Wi-Fi e *Bluetooth* também podem trabalhar em conjunto, atendendo as disponibilidades da aplicação e maximizando a comunicação. [6] provê uma arquitetura de interoperabilidade entre as duas interfaces, para uso em redes domésticas. Todos processos de adaptação propostos geralmente necessitam acessar informações do ambiente e do dispositivo de forma eficiente, portanto o estudo e a criação de APIs voltadas para a comunicação sem fio têm crescido rapidamente. Em [7], é descrito um ambiente de teste de algoritmos de roteamento *ad hoc*, chamado *APE - Ad Hoc Protocol Evaluation testbed*, utilizando o sistema operacional Linux. Uma API para a interface Wi-Fi é fornecida como ferramenta de coleta de informações de conectividade na rede *ad hoc* formada. Com estas informações, mais as estatísticas de perdas de pacotes e atrasos de transmissão, os protocolos *ad hoc* podem ter informações mais reais com relação às conexões envolvidas.

Em [8], é proposta uma API de comunicação, que estende algumas funcionalidades do *APE*, sendo possível a implementação de suas funcionalidades nos sistemas operacionais Windows e Windows CE. A proposta desse trabalho é a mais genérica possível, possibilitando uma programação multi-plataforma, sem extrair muitas informações das interfaces de comunicação sem fio.

Em [9], foi desenvolvida uma API integrada para o Bluetooth, definida BlipNet, onde um elemento definido como BlipServer, está conectado a rede ethernet e a diversos elementos BlipNodes. Estes elementos possuem interfaces Bluetooth e coletam as informações sobre os elementos que desejam conectar a eles. O BlipServer coleta informações sobre estes dispositivos e fornece uma API de programação, usando J2ME, para que as aplicações controlem as conexões vindas pelos BlipNodes. Dessa maneira as aplicações podem coordenar o acesso destes elementos à rede fixa, fornecendo os serviços da Internet para esses dispositivos, de acordo com suas características. Esta implementação foi feita tanto para Linux quanto para Windows.

3. Características das Interfaces sem fio

Na descrição do modelo OSI da ISO [10], a interface para a comunicação sem fio encontra-se nas duas primeiras camadas, a camada física e a de enlace, divididas respectivamente no modem de rádio transmissão, responsável por enviar e receber os dados, e o controlador de acesso ao meio, que coordena o tráfego de dados. A Figura 1 mostra como é a estrutura de uma interface de comunicação sem fio e sua visão interna.

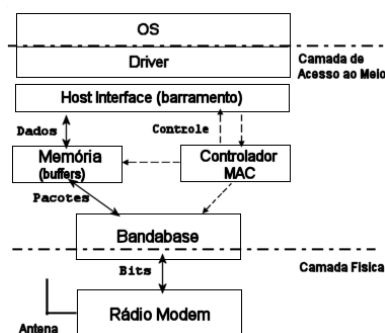


Figura 1. Diagrama Funcional da interface sem fio

As principais características do modem de rádio é o controle da frequência de transmissão, a taxa de sinalização, a modulação e a potência de transmissão. A parte digital, responsável pelo controle dessas funcionalidades, se encontra na Bandabase, que funciona como a interface com a camada de enlace, fornecendo essas informações para o controlador de acesso ao meio. No modem encontra-se a parte analógica da comunicação, onde é feita a modulação dos dados para a transmissão.

A camada de controle de acesso ao meio, MAC, é responsável por executar o protocolo de acesso ao meio. Ele coordena o espaço de memória limitado para os buffers e dados de configuração e estatísticas, e é responsável pelo formato e definição dos quadros, o controle de acesso ao meio e algumas características de gerenciamento de rede. O *host interface* provê a comunicação entre o software, normalmente o driver da placa, com o MAC.

- **Camada Física:** A camada física das interfaces sem fio podem utilizar tanto a comunicação através de rádio-frequência, como comunicação em infra-vermelho. Este trabalho é focado na primeira classe. As interfaces de rádio utilizadas pelo *Bluetooth* e *Wi-Fi* para a comunicação usam a banda conhecida como *ISM (Industrial, Scientific and Medical)*, sendo utilizadas as faixas de frequência de 900 MHz a 2.4GHz. A técnica definida para o uso é o *Spread Spectrum*, que possui uma maior confiabilidade na transmissão. O *Spread Spectrum* utiliza mais largura de banda para transmitir do que é realmente necessário, reduzindo o impacto das interferências localizadas. Na banda de 2.4GHz, a regulamentação especifica que os sistemas devem utilizar dois tipos de técnicas de transmissão: *Direct Sequence* ou *Frequency Hopping*.

O princípio de funcionamento do *Direct Sequence* consiste em espalhar o sinal sobre uma banda de transmissão, multiplexando o sinal com uma assinatura de identificação. Cada bit é modulado por um código que cobre grande parte do canal de transmissão. Para uma taxa de 2Mb/s, normalmente utilizada na WLAN, o sinal é espalhado sobre 22MHz da banda. Dessa maneira, interferências em determinados canais são minimizadas, e a demodulação consegue restaurar os sinais recebidos. O *Frequency Hopping* (FH) utiliza um conjunto de canais de largura pequena e transmite a cada momento em um deles, seguindo uma seqüência. No caso da banda ISM de 2.4GHz, ela é dividida em 79 canais de 1 MHz. Periodicamente, num intervalo entre 20 a 400 ms, o sistema efetua um salto para um canal diferente, a partir de um padrão pré-determinado, e assim evitando as interferências, pois permanece pouco tempo no mesmo canal. A principal vantagem desse funcionamento, com relação ao *Direct Sequence* é que uma interferência localizada em determinado canal de frequência é prontamente evitada pela utilização de outros canais livres. O *Bluetooth* permite que até 79 elementos possam comunicar-se diretamente, a uma curta distância, e sem degradar a qualidade da comunicação, enquanto que no *Direct Sequence* a taxa de transmissão degrada devido as colisões. Quando a comunicação chega ao limite, as colisões começam acontecer, onde diversos sistemas utilizam os mesmos canais ao mesmo tempo interferindo uns com os outros.

O modem que trabalha com o FH é relativamente mais simples que o que usa o *Direct Sequence*, sendo seu custo de implementação mais barato, compensando o custo de uma camada de enlace mais robusta.

- **Modulação:** Para a modulação da transmissão, existe uma troca entre a velocidade de modulação e a qualidade do sinal recebido. Quanto mais rápido for o sistema de modulação, mais forte deve ser o sinal recebido. Para compensar essas flutuações, os sistemas normalmente trabalham com estas duas abordagens mescladas, trocando para taxas menores caso o sinal recebido tenha um nível mais fraco.
- **Propagação:** A propagação do sinal é influenciada por diversos fatores, desde absorção do sinal por obstáculos até interferências por diversos tipos de ruído. Dessa maneira é difícil prever, com precisão, o comportamento das transmissões em diferentes cenários. Não existe um padrão, ou método conhecido para medir o alcance da propagação de maneira correta. As melhores métricas para definir a qualidade da propagação do sinal são relativas a potência de transmissão e a sensibilidade do sinal recebido. Com esses dois valores é possível calcular a atenuação do sinal, medida a cada recepção dos pacotes, sendo essa a diferença do valor do sinal transmitido pelo recebido. Outro fator diretamente relacionado com a recepção é a razão *Signal to Noise Ratio* (SNR), que define a diferença da potência de um sinal e um ruído, no momento da recepção. Dessa forma valores pequenos de SNR significam uma melhor sensibilidade a recepção. Algumas das interfaces sem fio fornecem um acesso a essas medidas de qualidade e potência do sinal, e do nível de energia do sinal medido, *Radio Signal Strength Indicator*.
- **Contexto de rede:** A informação obtida através do contexto de rede é a descrição que as aplicações e algoritmos possuem sobre o ambiente de rede em que estão executando, relativas a comunicação direta com seus vizinhos. No caso da comunicação sem fio, são os dados relativos as constantes mudanças que ocorrem durante a comunicação, abrangendo os cenários das redes infra-estruturadas e das ad hoc. Este pode ser descrito a partir de alguns parâmetros, definidos em [11], como conectividade da rede, capacidade de transmissão do canal, mobilidade, entre outros.

As camadas de enlace sem fio fornecem o ferramental para se extrair os dados necessários para estes parâmetros, a partir das medidas do nível de sinal ruído, os níveis de potência de transmissão e a recepção dos pacotes.

4. Wi-Fi

O grupo de trabalho do IEEE, o 802.11, [2], é responsável pelo desenvolvimento de um padrão de comunicação de *WLAN (Wireless Local Area Network)*, também conhecido como Wi-Fi. O padrão IEEE 802.11 foi criado em 1989 e no final da década de 90 começou a se destacar no mercado. Diversas empresas se associaram na produção das placas de rede, criando a designação Wi-Fi [12], dada aos produtos que implementam o IEEE 802.11b, sendo este uma extensão do padrão original, com capacidade de transmissão de 11Mbps. O principal objetivo desse grupo é homologar a conformidade da implementação destas interfaces com a especificação criada pelo IEEE.

A especificação do IEEE 802.11 define uma camada MAC e diversas camadas físicas, possibilitando uma variedade de características de acesso ao meio. Basicamente ele trabalha em duas configurações:

- **Independente (ad hoc):** as estações comunicam diretamente umas com as outras, sem infraestrutura de suporte, formando as redes ad hoc. Neste modo eles efetuam a comunicação ponto a ponto com os elementos que se encontram mais próximos do alcance da antena.
- **Infra-estruturada:** as estações comunicam via pontos de acesso, os quais são parte de um sistema de distribuição de comunicação, cobrindo uma área maior. Os pontos de acesso coordenam o funcionamento e a comunicação com as estações, controlando a autenticação, o acesso e a troca de acessos.

A camada física das interfaces sem fio podem utilizar tanto a comunicação através de radio-freqüência como comunicação em infra-vermelho. Este trabalho é focado na primeira classe, através do *spread spectrum*, sobre a banda ISM. Neste trabalho foram utilizadas as interfaces que trabalham com o *Direct Sequence Spread Spectrum (DSSS)*. Este funciona sobre a banda ISM, em múltiplos canais, de 2.4 a 2.4835 GHz, com uma separação entre os canais de 30Mhz. Ele atinge a taxa de transmissão de 1 e 2Mbps, utilizando a modulação DQPSK e DBPSK e a potência de transmissão é de aproximadamente até 1000mW, com possibilidades de ajuste até valores de 100mW. Uma extensão a esse padrão, chamado de *High-Rate Direct Sequence Spread Spectrum (HR/DSSS)*, que utiliza o *Complementary Code Keying (CCK)*, permite valores de conexões que vão de 5.5 a 11 Mbps. Essa extensão é o 802.11b, e é a mais utilizada atualmente no mercado, a qual recebeu a denominação de Wi-Fi.

A camada de acesso ao meio descrita pelo 802.11 possui diversas funcionalidades de fragmentação de quadros, retransmissão e controle de acknowledges. Ela provê dois mecanismos de acesso básico, o *Distributed Coordination Function (DCF)*, onde existe o compartilhamento eficiente do meio de acesso usando contenção do canal, e o, *Point Coordination Function (PCF)*, um método de acesso de polling, onde um coordenador seleciona qual elemento tem o direito de transmissão. O modo de uso do PCF é opcional e estabelece um método de acesso ao meio sem contenção. O procedimento de acesso ao meio usando o *DCF* é conhecido como o (CSMA/CA), *Carrier Sense Multiple Access with Collision Avoidance* onde a estação que deseja mandar alguma mensagem detecta se o meio está livre, senão espera por um tempo definido, para novamente tentar efetuar a transmissão. Ele espera por um tempo aleatório, evitando assim possíveis colisões.

Para contornar o problema do nó escondido é utilizado um mecanismo de *handshaking*, chamado RTS/CTS. Neste processo são trocadas duas mensagens entre os nós comunicantes, *Request To Send (RTS)* e *Clear To Send (CTS)*. Ambos os quadros informam o tamanho da transmissão, onde os outros elementos da rede sabem por quanto tempo o canal será utilizado. Quando o canal está liberado, um quadro de controle RTS é enviado, anunciando que a estação deseja transmitir. Este quadro é respondido com um quadro CTS, indicador da estação receptora, informando que está preparada para receber os dados.

5. Bluetooth

O termo Bluetooth [13] se refere a uma tecnologia que permite a comunicação de voz e dados através de um enlace de rádio de baixo alcance. Esta tecnologia foi, inicialmente, desenvolvida para a substituição dos cabos associados a dispositivos periféricos e evoluindo para o conceito de WPANs. A estrutura básica de comunicação no Bluetooth é chamada de piconet, e pode ser descrita como uma rede onde um nodo central, denominado mestre, se comunica ativamente com os outros nodos, chamados de escravos, formando uma topologia em estrela. Segundo a especificação, podemos ter no máximo oito elementos ativos dentro de uma piconet.

Os dispositivos Bluetooth operam na faixa de frequência de 2.4 GHz, na banda ISM, usando o (*Frequency Hopping* - FH), dividida em 73 canais, numa taxa de saltos de frequência é de 1600 hops/seg no modo conectado e 3200 hops/seg no modo de procura / sincronização. Este esquema completo é chamado de *Frequency Hopping Code Division Multiple Access*, FHCDMA. Cada um desses canais é usado por 625 μ s (um slot), e a seqüência desses canais é escolhida de maneira pseudo-aleatória, de acordo com o identificador e o relógio do mestre da piconet.

Durante a comunicação do *Bluetooth*, são definidos alguns tipos de canais lógicos, para controle e transmissão de dados, sendo entre eles os mais importantes, um orientado a conexão o *Synchronous Connection Oriented* (SCO), usado exclusivamente para comunicação de voz e outro orientado a pacote *Asynchronous Connectionless* (ACL), usado para transmissão de dados. O canal SCO é um canal ponto a ponto entre um mestre e um escravo, estabelecido pela reserva de slots duplex durante intervalos regulares. O canal ACL é um canal ponto a multiponto que é alocado dinamicamente de acordo com a demanda.

Os tipos de pacotes numa piconet são relacionados aos canais lógicos, podendo ser utilizados de acordo com as necessidades do tráfego. Estes pacotes podem ser agrupados em três conjuntos: controle, dados e áudio. Os pacotes de controle são utilizados em todos os canais lógicos e são: ID, FHS, NULL, POLL e DM1. O ID e o FHS são utilizados durante os procedimentos de estabelecimento de conexão, e contém as informações necessárias para as etapas de sincronização. Os pacotes NULL e POLL não possuem payload e são usados durante o modo conectado para os nodos manterem a sincronia do mestre com o escravo. Os pacotes do estilo DM1 são usados para as mensagens de controle e dados possuindo um bit indicador de sua natureza e prioridade; Os pacotes de dados usados no canal ACL são transmitidos de forma assíncrona e contendo informações de dados e controle. São os pacotes DM1, DH1, DM3, DH3, DM5 e DH5. Os pacotes DM são os *Data Medium Rate* e contém um código de correção 2/3 FEC. Os pacotes DH são os *Data High Rate* e não possuem o código FEC, permitindo a transmissão de mais dados. Os pacotes DM1 e DH1 usam um slot, os DH3 e DM3 usam três slots, e os DM5 e DH5 usam 5 slots. A vazão máxima, de 752kbps, é alcançada quando utiliza o pacote DH5.

Na Tabela 1 é mostrado o tamanho pacotes, a sua utilização dos slots e a prioridade de transmissão define o uso da banda passante. Os *payloads* dos pacotes de dados variam de zero até o tamanho máximo de cada tipo. Como citado anteriormente, os pacotes de maior uso de slots propiciam as melhores taxas, mas são mais sensíveis a falhas e ruídos.

6. Arquitetura da API

O desenvolvimento desta API para as interfaces sem fio foi feito no ambiente Linux sobre os drivers dos dispositivos Wi-Fi e *Bluetooth*, sendo a implementação feita em na linguagem C, onde foram criadas bibliotecas para o acesso as funções. Devido ao seu código aberto, os *drivers* de cada padrão, Wi-Fi e *Bluetooth*, possuem visíveis todas as funções de acesso aos recursos do hardware em questão. O desenvolvimento da API é feito de maneira a agrupar essas funções, extraindo a funcionalidade necessária para as aplicações em questão. A API não modifica nenhuma

Tabela 1. Velocidade e capacidade dos pacotes do *Bluetooth*

Nome	Payload (bytes)	FEC	Taxa max. Assimétrica (kbps)	Tempo de transmissão (μ s)
ID	na	na	na	312
NULL	na	na	na	312
POLL	na	na	na	312
FHS	18	2/3	na	312
DM1	0-17	2/3	108.8	625
DH1	0-27	-	172.8	625
DM3	0-121	2/3	387.2	1875
DH3	0-183	-	585.6	1875
DM5	0-224	2/3	477.8	3125
DH5	0-339	-	723.2	3125

funcionalidade dos *drivers* já implementados, apenas atua como um *middleware* entre os *drivers* e as aplicações usuárias. As operações básicas da API foram definidas e agrupadas em blocos funcionais de acordo com as suas características. A Figura 2 mostra os blocos funcionais da API.

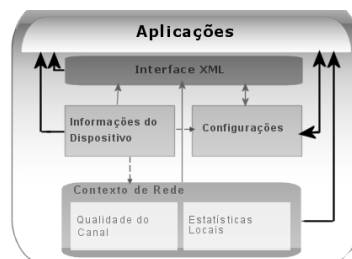


Figura 2. Arquitetura Funcional da API

- **Informações do Dispositivo:** define as operações de identificação dos *drivers* das interfaces e das funcionalidades implementadas nestas. Nas interfaces Wi-Fi, identificam quais as características de controle são acessíveis, a versão e o fabricante do chipset e quais parâmetros podem ser configurados. Nas interfaces *Bluetooth*, identificam a versão da especificação utilizada, o fabricante, quais as operações HCI estão acessíveis e as configurações do rádio modem;
- **Configuração:** este bloco acessa todas as informações de configuração da interface, desde as que permitem seu funcionamento básico, até o ajuste de parâmetros para o melhorias no desempenho. Algumas das operações são passíveis de ajuste remoto, via comandos do bloco XML. Devido a questões de segurança, apenas os parâmetros de ajuste de desempenho são habilitados para a execução remota. Tanto no Wi-Fi e no *Bluetooth* são informações de ajuste dos pacotes e de nível de potência do sinal;
- **Contexto de Rede:** este bloco de funções acessam as informações referente ao contexto de comunicação da rede, sendo o ponto de partida para a inferência do comportamento do dispositivo na rede. Ele foi subdividido em dois blocos:
 - **Qualidade do Canal:** coleta as informações da qualidades dos canais de comunicação, através de amostragens temporais das medidas do RSSI, da qualidade e da potência de transmissão da interface. A partir desses valores, determina o nível de conectividade e mobilidade do dispositivo, criando um histórico desses parâmetros, de maneira que é possível determinar o comportamento da conexão da interface com elementos vizinhos. Também permitem estipular um posicionamento dos dispositivos, a partir de um referencial pré-determinado. No Wi-Fi, a determinação da posição pode ser utilizada nos nós ligados aos pontos de acesso, no caso do modo de funcionamento infra-estruturado, e entre vizinhos,

no modo ad hoc. No caso do Bluetooth, essa localização é possível no mestre da piconet, acessando as posições relativas com os seus escravos;

- **Estatísticas Locais:** A medida do uso da interface, como quantidade de pacotes retransmitidos e erros de transmissão. No Bluetooth essa informação é associada aos canais lógicos, o ACL e SCO, permitindo a medida da comunicação de cada um;
- **Interface XML:** O bloco XML é responsável por coletar as informações dos outros módulos, e convertê-los de acordo com os DTDs pré-determinados. As informações dos blocos responsáveis pelo contexto de rede são geradas a cada nova requisição, devido as novas amostragens necessárias, enquanto que as informações dos outros blocos são geradas na sua inicialização. O *parser* efetua o retorno das operações para o bloco de controle, que as executa dependendo da sua prioridade. O uso do XML como método para a manipulação da informação em ambientes móveis é de extrema importância pois é possível um mapeamento das funcionalidades de maneira flexível, onde os recursos e serviços disponíveis podem ser descritos de maneira independente da tecnologia;

6.1. Implementação

O método utilizado no Linux, para acessar os parâmetros de uma interface de rede é através da função `ioctl`, uma chamada de sistema que trabalha sobre um descritor de arquivo, um *socket* associado a essa interface. Os drivers de cada dispositivo definem as operações que podem ser executadas, através de constantes específicas para o acesso. Assim na chamada da função `ioctl` são definidas as operações que são efetuadas sobre o dispositivo. São operações que definem modos básicos de configuração e funcionamento, como definições de parâmetros da interface e outras que permitem a análise da qualidade da recepção de quadros e do sinal recebido.

No caso do Wi-Fi foram utilizadas as implementações da biblioteca Wireless Extensions [14], sobre o driver da interface Wi-Fi, que já estão integradas ao Kernel do Linux. Com o Bluetooth, foi utilizado a pilha de acesso do Affix [15], através da qual as funções da camada HCI foram acessadas para programação das funcionalidades da camada Baseband. Por serem tecnologias com características distintas, cada driver permite um conjunto de operações comuns e outras específicas, tanto do padrão, quanto do próprio hardware envolvido. A Tabela 2 mostra algumas das operações feitas sobre os *drivers* Wi-Fi e Bluetooth.

7. Experimentos

Para demonstrar a aplicação da API desenvolvida realizamos uma série de testes com os dois ambientes de estudo. Os primeiros testes incluem a análise de interferência de um ambiente sobre outro e seu impacto no desempenho das aplicações. A segunda bateria de testes relaciona-se às características do canal sob condições de mobilidade. Dentre as diversas informações que se pode obter através da API, decidimos nos concentrar em três medidas:

- *Qualidade do canal:* é a qualidade da recepção dos dados. Os seus valores e a sua métrica é dependente do fabricante da interface. Na interface 802.11 utilizada este valor varia de 0 a 92, enquanto que na interface Bluetooth, varia de 0 a 255;
- *Força do sinal de recepção:* Na interface 802.11 este valor varia 0 a 255. No Bluetooth, os valores do RSSI estão associados as regiões de alcance da transmissão, existindo três áreas distintas de medida. Na primeira, se a interface estiver numa região próxima a unidade transmissora, onde o nível do sinal se encontra próximo de 20dB, essa medida retorna o valor 0. Caso a interface esteja fora dessa região, retorna um valor entre -1 e -127. Caso a interface se aproxime em excesso, ela entra na região em que a recepção do sinal é muito forte, e a função retorna valores de 1 a 128, que indica a sobrecarga em cima da recepção;
- *Vazão:* é a quantidade de bytes recebidos pela aplicação em um determinado instante.

Tabela 2. Operações sobre os drivers Wi-Fi e Bluetooth

<i>Driver</i>	Comando	Descrição
Wi-Fi	SIOCSIWFREQ	define o canal de frequência (Hz)
	SIOCGIWNWID	define o network id
	SIOCSIWMODE	define o modo de operação (Ad Hoc, Managed)
	SIOCSIWESSID	obtém o SSID da rede
	SIOCSIWNICKN	seta o nickname do nó
	SIOCGIWAP	define o endereço MAC do ponto de acesso
	SIOCSIWSENS	define o nível de sensibilidade ao canal (dBm)
	SIOCGIWSTATS	obtém as estatísticas no <code>/proc/net/wireless</code>
	SIOCGIWSPY	define o modo de observação de outros nós
Bluetooth	HCI_reset	Reinicia o dispositivos
	HCI_create_connection	Cria uma conexão ACL
	HCI_read_BD_ADDR	Lê o endereço do dispositivo
	HCI_disconnect	Desconecta o canal
	HCI_inquiry	Efetua o <i>Inquiry</i>
	HCI_HoldMode	Entra em modo <i>Hold</i>
	HCI_ReadRssi	Lê o valor do RSSI da interface
	HCI_GetLinkQuality	Lê o valor da qualidade do canal

7.1. Análise da Interferência

Para estes experimentos utilizamos o sistema operacional Suse Linux 7.1, kernel 2.4.18, em um notebook Compaq Armada, Pentium II, 365 MHz, 128Mb e dois desktops, Pentium II, 400Mhz e 256 Mb de memória. As interfaces 802.11 utilizadas foram uma placa PCMCIA Interlink, com o chipset Prism I, e uma placa PCMCIA Wavelan, com o chipset Hermes 1. A do *Bluetooth* foi o kit da Ericsson, com os chipsets, ROK 101 007 e ROK 101 008, e a interface USB-Bluetooth da TDK, com o chipset da Cambridge Silicon Systems, o Bluecore. Também foram utilizados dois pontos de acesso, um *Bluetooth* da WidComm, o Bluegate 2100, e um ponto de acesso 802.11, que utilizou a placa Wavelan.

Os gráficos 3(a) e 3(b) representam a qualidade dos sinais quando submetidos a um tráfego CBR no canal *Bluetooth* e um tráfego FTP no canal 802.11. No início de monitoração (tempo 0 s), somente o tráfego *Bluetooth* ocupa o canal, portanto a qualidade observada atinge o ponto máximo em ambos os canais. Ao iniciar a aplicação FTP, a qualidade do canal *Bluetooth* reduz, oscilando entre 210 e 235 até o fim da monitoração (tempo 120 s). Pela figura 3(b), podemos notar que a qualidade do canal 802.11 mantém-se estável em seu valor máximo, porém se torna menor e com grandes oscilações a partir do início da transferência de dados. A conclusão, já prevista, é que a transmissão de dados em um canal exerce grande impacto sobre o desempenho do outro canal. Entretanto, é interessante notar que a variação da qualidade do canal 802.11 é muito superior àquela observada no canal *Bluetooth*, o que se deve à forma como os protocolos utilizam o meio (seção 3.). Uma vez que o *Bluetooth* foi feito tendo a robustez como guia, é de se esperar que sofra menos impacto que as tecnologias 802.11. A amostragem desses dados é feita a cada 100ms e cada ponto corresponde ao valor médio e a variância de 10 amostragens consecutivas, ou seja, a cada 1 segundo.

A seguir analisaremos o efeito de variações de parâmetros no canal *Bluetooth* e suas consequências no canal 802.11. Como explicado na seção 5., a protocolo *Bluetooth* pode ser configurado para transmitir dados a diferentes taxas, cada qual relacionada a um tipo de pacote distinto (tabela 1). As figuras 4(a) e 4(b) demonstram o efeito do uso desses diferentes tipos de pacotes sobre a qualidade observada nos dois canais de comunicação quando submetidos ao mesmo tráfego anterior. Os gráficos representam a distribuição acumulada dos valores de qualidade observada em

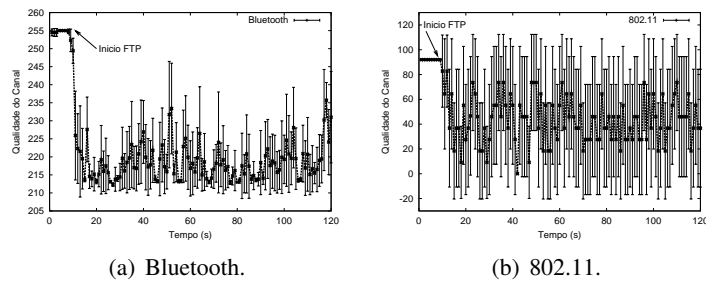


Figura 3. Qualidade dos canais Bluetooth (pacote DH1 e tráfego CBR) e 802.11 (taxa de transmissão de 1Mb e tráfego FTP).

cada experimento. Na figura 4(a), os tipos de pacotes DH1 e DM1 são os que mais apresentam valores reduzidos para qualidade do canal *Bluetooth*: 90% dos valores observados estão abaixo de 231 para o DH1 e abaixo de 227 para o DM1. O tipo de pacote DH5 é aquele que apresenta a melhor distribuição de valores na faixa aceitável para comunicação (200 e 255). Esse comportamento também tem reflexos no canal 802.11. No gráfico da figura 4(b), os tipos de pacotes DM1 e DH1 são os que mais degradam a qualidade do canal 802.11, sendo o DH5 aquele que possui a maior quantidade de pontos no valor máximo de qualidade. Isso é um dado animador pois, além de apresentar a maior taxa de transmissão de dados, o DH5 é o que menos afeta os outros ambientes de comunicação e, portanto, o que menos degrada o desempenho de outros canais. Compreender os efeitos dos diferentes tipos de carga e configurações é extremamente importante para se definir políticas de acesso em ambientes compartilhados. No nosso caso, poderíamos definir uma mudança automática do tipo de pacote para DH5 no momento em que o dispositivo móvel detectar uma comunicação semelhante à que foi modelada nesse experimento.

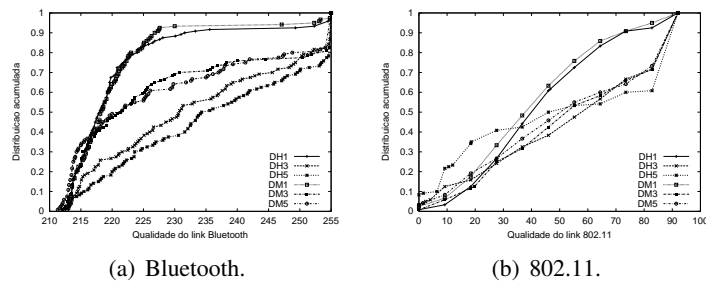


Figura 4. Distribuição acumulada da qualidade dos canais Bluetooth e 802.11 (taxa de transmissão de 1Mb).

Os gráficos 5(a) e 5(b) ilustram a força do sinal para os ambientes estudados para a mesma carga descrita acima. Nota-se que, no *Bluetooth*, há uma clara distinção entre a força de sinal para os pacotes da classe DH (valores inferiores) para os da classe DM (valores superiores). Nesse gráfico, quanto menor o valor da escala, maior é a força de recepção do sinal. Para o ambiente 802.11, a força do sinal possui uma maior variação. No gráfico 5(b), quanto maior o valor da escala, mais fraco é o sinal de recepção do canal. Nesse último caso, não há uma relação clara entre o tipo de pacote do canal *Bluetooth* com a força do sinal no canal 802.11. Para todos os testes, a faixa de valores é aproximadamente a mesma e suas variâncias são sempre elevadas.

Mudanças na configuração do ambiente 802.11 também alteram o comportamento da qualidade dos canais. Os gráficos das figuras 6(a) e 6(b) refletem a qualidade do canal 802.11 quando alteramos a sua taxa de transmissão de 1Mb para 2Mb e geramos o mesmo tráfego anterior, com tipo de pacote DH5 no canal *Bluetooth*. Podemos notar que a taxa de transmissão mais alta é menos

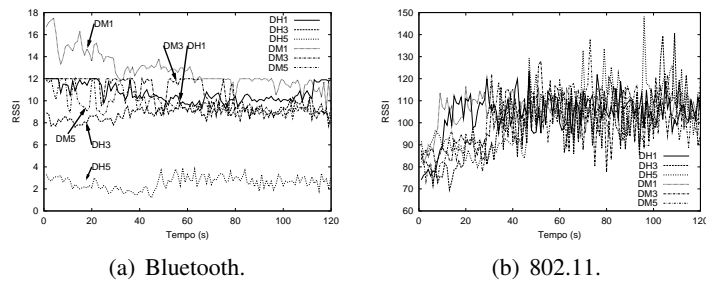


Figura 5. Força do sinal nos canais Bluetooth e 802.11 (taxa de transmissão de 1Mb).

afetada pela interferência do *Bluetooth*, o que impacta diretamente a performance das aplicações nos dois ambientes. As figuras 7(a) e 7(b) ilustram a vazão medida pela aplicação FTP no dispositivo móvel sem interferência no canal *Bluetooth* e com a interferência descrita acima. O valor médio da vazão sem interferência é cerca de 240kbps para taxa de transmissão de 1Mbps e cerca de 320Kbps para taxa de transmissão de 2Mbps, porém esses valores caem para 127kbps e 208Kbps sob efeitos de interferência, respectivamente. Apesar da queda visível no valor médio, é interessante notar que a qualidade alta do canal no caso de taxa de transmissão de 2Mb permite picos de vazão próximos aos valores sem interferência. O mesmo não acontece no caso de taxa de 1Mb: a qualidade do canal é tão degradada que o *throughput* máximo chega a apenas 82% do valor médio sem interferência.

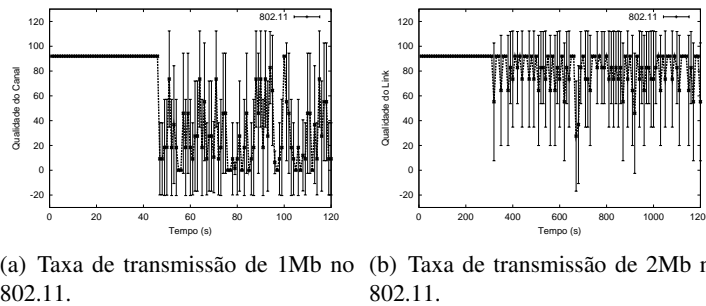


Figura 6. Qualidade do link 802.11 (interferência Bluetooth com pacote DH5).

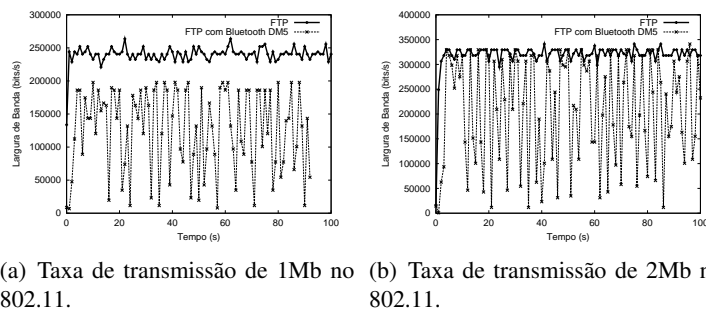


Figura 7. Largura de banda do link 802.11 para FTP sem interferência e com interferência do Bluetooth com pacote DH5.

Apesar de obter uma qualidade melhor e um *throughput* maior, a taxa de transmissão de 2Mb é mais susceptível a perdas de conexões que a de 1Mb. O gráfico da figura 8(a) ilustra uma queda de conexão do canal 802.11 em um cenário de taxa de transmissão de 2Mb e interferência *Bluetooth* com tipo de pacote DH1. As quedas de conexões são mais frequentes para os

tipos de pacotes DH1 e DM1. A análise deste comportamento é um fator crucial para se determinar políticas que avaliam tanto a performance quanto a confiabilidade de comunicação de uma aplicação em um ambiente compartilhado. O gráficos das figuras 8(b) e ilustra o que acontece com a força do sinal durante todo o processo. Nesses gráficos é possível ver claramente as propriedades que caracterizam uma desconexão no ambiente sem fio: qualidade do sinal reduzida e força de recepção do sinal fraca.

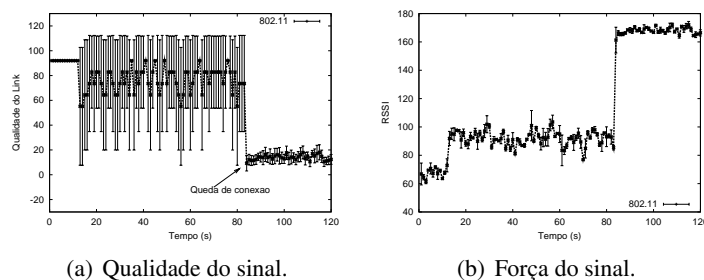


Figura 8. Qualidade e força do sinal do canal 802.11 durante uma queda de conexão (interferência do Bluetooth com pacote DH1)

7.2. Análise da mobilidade

Neta seção serão analisados alguns resultados obtidos durante a avaliação dos ambiente 802.11 e Bluetooth na presença de mobilidade. As figuras 9(a) e 9(b) representam a qualidade dos canais Bluetooth e 802.11 ao se aproximar do ponto de comunicação. Podemos notar uma aumento gradativo na qualidade do canal 802.11 e um aumento mais brusco para o *Bluetooth*. Também é possível notar um pequeno aumento e estabilização de valores nos momentos de paradas. Os gráficos 10(a) e 10(b) ilustram os valores da força do sinal durante o movimento. Podemos notar que, para o ambiente 802.11, essa última curva acompanha a curva de qualidade do sinal e aumenta à medida que se aproxima do ponto de comunicação. Também é possível notar o aumento relativo dos valores nos momentos de parada. Já no caso do *Bluetooth*, notamos que a força do sinal aumenta abruptamente para próximo do maior valor possível, indicando uma possível entrada na área de cobertura. O comportamento dos padrões da qualidade do sinal e da força do sinal nesses gráficos diferem daqueles presentes nos gráficos de interferência, o que pode indicar uma possível solução para aplicações que desejam distinguir entre movimentação e interferência externa durante tomadas de decisões. Nesse caso, o auxílio de uma ferramenta que faça a monitoração contínua dos ambientes é essencial.

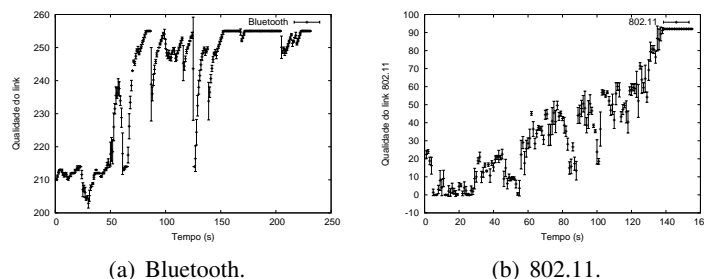


Figura 9. Qualidade dos links Bluetooth e 802.11 aproximando-se do ponto de comunicação.

Na seção 7. foi dito que o desempenho e a qualidade dos canais 802.11 e do *Bluetooth* são minimizados quando se usam os tipos de pacotes DM1 e DH1 para comunicação no canal *Bluetooth*. Isso poderia levar ao uso indiscriminado de pacotes do tipo DH5 ou semelhante para

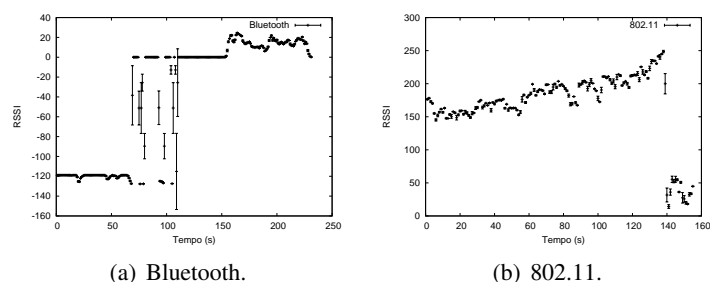


Figura 10. Força sinal Bluetooth e 802.11 aproximando-se do ponto de comunicação.

transferência de dados. No entanto, o uso de pacotes DM1 e DH1 é indicado quando se deseja manter a conexão *Bluetooth* ativa por mais tempo, pois possuem um suporte mais robusto que os outros. Os pontos do gráfico da figura 11 representam a qualidade do canal Bluetooth medido no dispositivo móvel quando nos afastamos de seu ponto de comunicação. É possível notar que a comunicação com pacotes DM1 e DH1 suportam níveis mais baixos de qualidade, o que possibilita manter a conexão ativa por mais tempo. Nesse experimento as conexões com pacotes do tipo DH5 e DM5 foram rompidas entre 10 a 15 metros do ponto de comunicação. As conexões com tipo de pacote DH1 e DM1 foram rompidas somente após 20 metros.

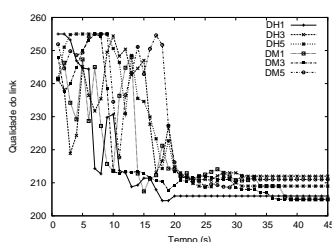


Figura 11. Qualidade do link Bluetooth afastando-se do ponto de comunicação para os 6 tipos de pacotes.

7.3. Informações em XML

Como dito na seção 6., a API é capaz de armazenar dados em XML e disponibilizá-las para acesso remoto. Uma possível aplicação para essa propriedade é a gerência dos dispositivos móveis, principalmente no que concerne a configuração remota de parâmetros das interfaces sem fio. Na figura 12 é demonstrada algumas informações recolhidas da interface *Bluetooth* e visualizadas em um navegador compatível com XML.

8. Conclusão

Neste trabalho foi apresentada uma API para monitoração a análise de duas tecnologias de comunicação sem fio: 802.11 para WLANs e Bluetooth para WPANs. Apesar de utilizarem a mesma faixa de frequência para comunicação, espera-se que essas duas tecnologias co-existam no futuro, o que demanda uma análise mais rigorosa sobre os impactos de interferência de uma sobre a outra. A API proposta tem o objetivo de facilitar esse processo, reduzindo o tempo de desenvolvimento e aprendizado de aplicações para esses ambientes. As funcionalidades da API foram agrupadas em 4 blocos funcionais: informação sobre o dispositivo, configuração, contexto de rede e interface XML. Cada bloco funcional executa operações distintas, porém todos estão a serviço das aplicações, fornecendo dados relevantes para os processos de adaptação tomados nas camadas superiores. Além disso, a API também foi desenvolvido tendo a gerência de redes com

```

- <device>
- <Bluetooth>
  <ifname>bt0</ifname>
  <bda>00:80:98:44:0e:50</bda>
  <localname>barao</localname>
- <class>
  <code>0xB201Computer (Laptop) [Networking, Object Transfer, Audio, Information]</code>
  <name>Computer (Laptop) [Networking, Object Transfer, Audio, Information]</name>
</class>
  <status>UP</status>
  <InquiryScan>DISC</InquiryScan>
  <PageScan>CONN</PageScan>
+ <features>
  <packettypes>DM1 DH1 DM3 DH3 DM5 DH5 HV1 HV2 HV3</packettypes>
+ <security>
+ <role>
+ <scan>
+ <LocalVersionInformation>
+ <timeouts>
+ <buffers>
+ <stats>
</Bluetooth>
</device>

```

Figura 12. Informações da interface Bluetooth geradas pela API em XML.

um referencial, disponibilizando informações em XML e possibilitando reconfiguração remota do dispositivo. Os experimentos sobre impacto de interferência e mobilidade aqui descritos visam demonstrar o potencial da ferramenta, que pode ser utilizada para uma gama muito maior de aplicações. Através do uso da API foi possível identificar e definir políticas de adaptação para contornar problemas comuns na comunicação sem-fio.

Referências

- [1] IEEE 802.15. <http://grouper.ieee.org/groups/802/15/>.
- [2] IEEE 802.11. <http://grouper.ieee.org/groups/802/11/>.
- [3] Bluetooth Forum. <http://www.bluetooth.com>.
- [4] A. F. da Conceição and F. Kon. Adaptação de Fluxos Contínuos UDP sobre Redes IEEE 802.11b. In *Anais do V Workshop de Comunicação sem Fio*, 2003.
- [5] J. Freitag, N. da Fonseca, and O. Branquinho. Diferenciação de Serviços em Redes 802.11 sob Degradação da Taxa de Transmissão. In *V Workshop de Comunicação sem Fio*, 2003.
- [6] P. Manzoni and J. C. Cano. Providing Interoperability Between IEEE 802.11 and Bluetooth Protocols for Home Area Networks. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 42(1):23–37, 2003.
- [7] E. Nordstrom. APE - a Large Scale Ad Hoc Network Testbed for Reproducible Performance Tests. Master's thesis, Information Technology Department of Computer Systems Uppsala University, june 2002.
- [8] C. M. T. Calafate and P. Manzoni. A Multi-Platform Programming Interface for Protocol Development. In *11th Euromicro Conference on Parallel Distributed and Network based Processing in Genoa*, 2003.
- [9] BlipNet. <http://www.blipsystems.com/>.
- [10] Andrew S. Tanenbaum. *Computer Networks*. Prentice-Hall, fourth edition, 2002.
- [11] S. Corson and J. Macker. Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. *Request for Comments document, rf2501*, January 1999.
- [12] Wi-Fi. <http://www.wi-fi.org/>.
- [13] Bluetooth Specification. <http://www.bluetooth.org>.
- [14] Wireless Extensions. http://www.hpl.hp.com/personal/jean_tourrilhes/linux/tools.html.
- [15] Affix. Affix bluetooth protocol stack for linux. <http://affix.sourceforge.net/>.