

Universidade Federal de Campina Grande
Centro de Ciências e Tecnologia
Departamento de Engenharia Elétrica
Laboratório de Sistemas Embutidos

Estágio Supervisionado

Implantação de uma Rede Privada Utilizando as Tecnologias
Ethernet, Wi-Fi e Bluetooth a partir de um Servidor Linux

José Luís do Nascimento
jluisn@dee.ufcg.edu.br

Angelo Perkusich
perkusic@dee.ufcg.edu.br

Campina Grande, Junho de 2005



Biblioteca Setorial do CDSA. Fevereiro de 2021.

Sumé - PB

Universidade Federal de Campina Grande
Centro de Ciências e Tecnologia
Departamento de Engenharia Elétrica
Laboratório de Sistemas Embutidos

Implantação de uma Rede Privada Utilizando as Tecnologias
Ethernet, Wi-Fi e Bluetooth a partir de um Servidor Linux

José Luís do Nascimento
Aluno

Angelo Perkusich
Orientador

Glossário

ACL: *Asynchronous Connectionless Link*
AT: *Attention Code*
CSMA/CD: *Carrier Sense with Multiple Access and Collision Detection*
DARPA: *Defense Advanced Research Projects Agency*
DHCP: *Dynamic Host Configuration Protocol*
DNAT: *Destination NAT*
DSSS: *Direct Sequence Spread Spectrum*
FDMA-TDD: *Frequency Division Multiple Access - Time Division Duplex*
Gnome: *GNU Object Model Environment*
GPE: *The GPE Palmtop Environment*
GPL: *General Public License*
HCI: *Host Controller Interface*
IANA: *Internet Assigned Numbers Authority*
ICMP: *Internet Control Message Protocol*
IEEE: *Institute of Electrical and Electronic Engineers*
IP: *Internet Protocol*
IrDA: *Infrared Data Association*
IrMC: *IrDA Mobile Communications*
KDE: *K Desktop Environment*
L2CAP: *Logical Link Control and Adaptation Protocol*
LAN: *Local Area Network*
LMP: *Link Manager Protocol*
MAC: *Media Access Control*
MTU: *Maximum Transfer Unit*
M-VoIP: *Mobile VoIP*
NAT: *Network Address Translation*
OBEX: *Object Exchange*
OFDM: *Orthogonal Frequency Division Multiplexing*
Opic: *Open Palmtop Integrated Environment*
P2P: *Peer-to-peer*
PAN: *Personal Area Network*
PBX: *Private Branch Exchange*
PDA: *Personal Digital Assistant*
PPP: *Point to Point Protocol*
PSTN: *Public Switched Telephone Network*
QoS: *Quality of Service*
RFCOMM: *Radio Frequency - oriented emulation of serial COM ports*
SDP: *Service Discovery Protocol*
SNAT: *Secure Network Address Translation*
SIG: *Special Interest Group*
ToS: *Type of Service*
TCP: *Transmission Control Protocol*

TDMA: *Time Division Multiple Access*
UDP: *User Datagram Protocol*
vCal: *virtual calendar*
VoIP: *Voice over IP*
vCard: *virtual card*
VPN: *Virtual Private Network*
WAP: *Wireless Application Protocol*
Wi-Fi: *Wireless Fidelity*

Conteúdo

1	Introdução	1
2	Objetivos	1
2.1	Objetivos gerais	1
2.2	Objetivos específicos	2
3	Tecnologias de Acesso Utilizadas	2
3.1	Pilha de Protocolos TCP/IP	2
3.1.1	Protocolo IP	3
3.1.2	Protocolo TCP	4
3.2	Ethernet	4
3.3	Redes sem fio	5
3.4	Bluetooth	6
3.4.1	Pilha Bluetooth	6
3.4.2	Aplicações	8
3.5	Wi-Fi	10
3.5.1	802.11a	10
3.5.2	802.11b	11
3.5.3	802.11g	11
4	Aplicações utilizadas	11
4.1	Iptables	11
4.2	NAT	12
4.3	DHCP	13
4.4	Bridge	13
4.5	BlueZ	14
5	Arquitetura Proposta	15
6	Implementação	16
6.1	Configuração das máquinas	16
6.2	O Sistema Operacional Linux	16
6.2.1	Instalação do Fedora Core 3	17
6.2.2	Instalação do driver da placa Wi-Fi	25
6.2.3	Configuração do BlueZ	28
6.2.4	Instalação do Familiar Linux	30
6.3	Implantação do Gateway Ethernet - Wi-Fi	31
6.4	Implantação do Gateway Bluetooth	33
6.5	Implantação do serviço DHCP	35

Lista de Figuras

1	Formas de interconexão entre estações	7
2	Pilha de Protocolos Bluetooth	8
3	Topologia da rede	15
4	Arquitetura M-VoIP	16
5	Tela de Boot	17
6	Testar CDs	18
7	Linguagem de instalação	19
8	Tipo de instalação	19
9	Escolha do tipo de Particionamento	20
10	Escolha do tipo de Particionamento	20
11	Configuração da interface de rede	21
12	Segurança	22
13	Senha do Administrador	22
14	Seleção dos pacotes	23
15	Instalação dos Pacotes	24
16	Tela de boas vindas	24
17	Familiar Linux <i>Bootloader</i>	31
18	Opie	32
19	Script Gateway	32
20	Script para o estabelecimento da ponte	34
21	Script dev-up	34
22	Script para conexões Bluetooth	35
23	Script de configuração DHCP	36

1 Introdução

O uso de redes de comunicações tornou-se imprescindível aos padrões de vida do homem moderno. A mobilidade em redes de computadores passou a ser um objetivo almejado a partir do momento em que dispositivos móveis como laptops, PDA e telefones celulares passaram a ser usados com frequência por grande parte da população mundial.

Neste contexto, torna-se importante o uso de um mecanismo de roteamento capaz de abstrair a tecnologia de acesso usada pelos diferentes tipos de redes. Uma vez que os usuários só estão interessados no produto final, ou seja, o uso da rede. Não importando pra ele se a conexão será Bluetooth, Wi-Fi ou Ethernet, dese é claro que ele possa acessar a tecnologia.

O uso do sistema operacional Linux para aplicações que envolvem tráfego de pacotes em redes TCP/IP, torna-se uma escolha natural, uma vez que aplicações sob a GNU GPL [2], são distribuídas de forma livre. Tornando-se cada vez mais usadas, seja em aplicações tradicionais, quanto em aplicações com restrições críticas de recursos, como é o caso dos sistemas embarcados.

Neste trabalho serão apresentados os passos necessários para a configuração do *gateway* Bluetooth, Wi-Fi e Ethernet.

O propósito do *gateway* apresentado é atuar como a camada de rede no projeto M-VoIP. O M-VoIP tem como objetivo disponibilizar uma arquitetura de rede sem fio com cobertura para várias tecnologias, mais especificamente Bluetooth e Wi-Fi, que dê suporte as aplicações de VoIP. A arquitetura de rede do M-VoIP estaria conectada a um PBX VoIP que faria o controle dos clientes com *softphones* e conectaria o sistema de telefonia IP à PSTN através de uma placa de telefonia.

Nas seções seguintes são apresentadas as tecnologias de acesso utilizadas, as aplicações utilizadas, a arquitetura proposta, os passos necessários para a implantação e por fim a conclusão.

2 Objetivos

2.1 Objetivos gerais

Compreender e integrar as tecnologias de rede Ethernet, Wi-Fi e Bluetooth, a partir do sistema operacional Linux. O qual disponibiliza uma série de ferramentas para implantação e monitoramento de redes, ferramentas estas, disponibilizadas de forma livre. O fato das aplicações serem disponibilizadas de forma livre com acesso ao código fonte, possibilita que as aplicações sejam alteradas pela comunidade Linux, sendo implantadas novas funcionalidades a partir das

existentes.

2.2 Objetivos específicos

Implementar uma rede privada tendo o sistema operacional Linux como roteador para redes Wi-Fi, Bluetooth, Ethernet permitindo a configuração dinâmica através do protocolo DHCP.

3 Tecnologias de Acesso Utilizadas

A seguir, tem-se uma visão geral dos protocolos e arquiteturas usadas no presente trabalho.

3.1 Pilha de Protocolos TCP/IP

Atualmente, redes baseadas nos protocolos TCP/IP são usadas como solução de convergência em aplicações de redes. Uma vez que estes protocolos apresentam implementação e manutenção consagradas, além de permitirem a interligação de redes locais através de outras redes de longa distância, com desempenho considerado.

Entre 1960 e 1970, muitas redes tinham seus próprios protocolos. O compartilhamento de informação entre estas redes logo se tornou um problema, e tornou-se necessário o desenvolvimento de um protocolo comum. A DARPA fundou a exploração deste protocolo comum e o pacote de protocolos ARPANET, que introduziu o conceito fundamental de camadas. O pacote de protocolos TCP/IP evoluiu a partir do ARPANET e tomou sua forma em 1978. Com o uso do TCP/IP, uma rede foi criada para ser usada principalmente por agências do governo e institutos de pesquisa com a finalidade de compartilhamento de informação e colaboração em pesquisas.

No início da década de 80 o protocolo TCP/IP tornou-se a espinha dorsal dos protocolos de múltiplos vendedores tais como ARPANET [14], NFSNET [15] e redes regionais. O protocolo foi integrado ao sistema operacional UNIX, na universidade da Califórnia em Berkeley, e tornou-se disponível ao público de maneira nominal. Deste ponto em diante o protocolo TCP/IP tornou-se amplamente utilizado devido a sua disponibilidade gratuita no UNIX [16] e a sua expansão para outros sistemas operacionais.

Hoje em dia, o TCP/IP provê às corporações a habilidade do uso de tecnologias de redes com níveis físicos diferentes dando aos usuários uma gama de funções comuns. Ele permite interoperabilidade entre equipamentos supridos por fabricantes diversos em múltiplas plataformas provendo acesso a internet.

A internet de hoje consiste de uma grande espinha dorsal de redes internacionais, nacionais e regionais que permitem aos indivíduos e redes regionais acesso a recursos globais.

3.1.1 Protocolo IP

O protocolo IP implementa um serviço orientado a pacotes. Sua função é transferir blocos de dados, denominados datagramas, da origem ao destino, onde a origem e o destino são *hosts* identificados pelos endereços IP. O protocolo IP também fornece o serviço de fragmentação e remontagem de datagramas, para que estes possam ser transportados em redes onde o tamanho máximo de cada pacote pode variar.

Como o serviço fornecido pelo protocolo IP é sem conexão, cada datagrama é tratado como uma unidade independente que não possui nenhuma relação com qualquer outro datagrama. A comunicação é não confiável, pois não são utilizados reconhecimentos fim a fim ou entre nós intermediários. Não são empregados mecanismos de controle de fluxo e de controle de erros. Apenas uma conferência simples do cabeçalho é realizada, para garantir que as informações nele contidas, usadas pelos roteadores para encaminhar datagramas, estejam corretas.

Endereços IP

Os endereços IP são números de 32 bits, divididos em quatro octetos na forma decimal. A primeira parte do endereço identifica uma inter-rede específica na inter-rede, a segunda parte identifica um *host* dentro desta rede. Este endereço, portanto, pode ser usado para nos referirmos tanto a redes quanto a um *host* individual. É através do endereço IP que os *hosts* conseguem enviar e receber mensagens pela rede, em uma arquitetura TCP/IP.

A atribuição dos endereços IP é realizada pela IANA [6].

O protocolo IP utiliza três classes diferentes de endereços. A definição de classes de endereços deve-se ao fato do tamanho das redes que estão interligadas variar muito, indo desde redes locais de computadores a redes públicas interligando milhares de *hosts*.

Na primeira classe de endereços, a classe A, o *bit* mais significativo é 0, os outros 7 *bits* do primeiro octeto identificam a rede, e os 24 *bits* restantes definem o endereço local. Esta classe é usada para redes de grande porte, seus valores variam de 1 a 126, e cada rede tem a capacidade de endereçar cerca de 16 milhões de *hosts*.

A classe B de endereços usa dois octetos para o número da rede e dois para endereços de *hosts*. Os endereços de redes classe B variam na faixa de 128.1 a 191.255, e cada rede pode interligar cerca de 65 mil *hosts*.

Faixa de Endereços IP	Quantidade de Hosts
10.0.0.0 - 10.255.255.255	16.777.216
172.16.0.0 - 172.31.255.255	1.048.576
192.168.0.0 - 192.168.255.255	65.536

Tabela 1: Faixas de endereços IP para redes privadas

Já os endereços da classe C, utilizam três octetos para identificar a rede e apenas um octeto para a *host*. Os endereços de rede situam-se na faixa de 192.1.1 até 223.254.254 (os endereços acima de 223 no primeiro octeto foram reservados para uso futuro), e cada rede pode endereçar 254 *hosts*.

Os endereços definidos a partir da RFC 1918 [1] ("*Address Allocation for Private Internets*") são usados em redes privadas sem a necessidade de autorização prévia de nenhuma entidade. Estes endereços pertencem as faixas de endereçamento mostradas na tabela 1.

3.1.2 Protocolo TCP

O protocolo TCP (Transmission Control Protocol) é um protocolo de transporte da arquitetura TCI/IP. O protocolo é orientado a conexão e fornece um serviço confiável de transferência de arquivos fim a fim. Ele é responsável por inserir as mensagens das aplicações dentro do datagrama de transporte, reenviar datagramas perdidos e ordenar a chegada dos datagramas recebidos.

O protocolo TCP interage de um lado com processos das aplicações e do outro com o protocolo da camada de rede. A interface entre o protocolo e a camada superior consiste em um conjunto de chamadas. Existem chamadas, por exemplo, para abrir e fechar conexões e para enviar e receber dados em conexões previamente estabelecidas. Já a interface entre o TCP e a camada inferior define um mecanismo através do qual as duas camadas trocam informações assincronamente.

3.2 Ethernet

Ethernet é o padrão mais utilizado no âmbito de redes locais LAN. Onde as máquinas são interconectadas em uma área limitada fisicamente, que pode ser um escritório, uma empresa ou uma universidade. O padrão Ethernet baseia-se em um barramento comum ao qual várias máquinas podem se conectar. Para evitar colisão de pacotes na rede é usado o protocolo CSMA/CD para controlar o acesso ao barramento.

No início, o hardware Ethernet consistia de um grosso cabo que era ligado às máquinas usando conectores especiais que furavam a camada de proteção

externa do próprio cabo. Posteriormente chegou o tipo 10base5, que usava conectores do tipo BNC (*British Naval Connector*) para inserir em conectores especiais em T da própria máquina, com terminais em ambas as extremidades da linha. Hoje o barramento Ethernet está contido dentro dos comutadores (*switchs*) e dos *hubs*, e os cabos são do tipo par trançado, 10baseT para as redes de 10 Mbit/s e 100baseT para as redes de 100Mbit/s. A comunicação entre o nó e o barramento Ethernet pode ser *full-duplex* ou *half-duplex*.

3.3 Redes sem fio

Usuários de redes sem fio podem usar as mesmas aplicações de redes tradicionais. Cartões de adaptação para redes sem fio usados em *laptops* e *desktops* suportam os mesmos protocolos que cartões Ethernet. As redes sem fio são usadas para prover aos usuários as seguintes funcionalidades:

- Mobilidade: a produtividade aumenta quando as pessoas têm acesso aos dados em qualquer lugar dentro da faixa de operação de uma rede sem fio. Decisões administrativas baseadas em informações em tempo real podem aumentar significativamente a eficiência do trabalhador.
- Baixos custos de implementação: as redes sem fio são de fácil montagem, administração, mudanças e realocações. Redes que são mudadas freqüentemente, fisicamente e logicamente, podem ser beneficiadas pela fácil implementação das redes sem fio. E elas podem ainda operar em locais onde instalações com fios são impraticáveis.
- Instalação e expansão da rede: a instalação de uma rede sem fio é rápida e fácil permitindo a eliminação de cabos que passariam através de paredes e tetos.
- Solução barata: as redes sem fio têm preços compatíveis com os preços dos dispositivos Ethernet tradicionais.
- Escalabilidade: as redes sem fio podem ser instaladas de formas variadas de modo a satisfazer as necessidades de instalações e aplicações específicas. As configurações de rede são alteradas facilmente, indo de redes com um número reduzido de usuários até uma infra-estrutura de rede que acomode centenas ou milhares de usuários, dependendo do número de dispositivos sem fio usados.

3.4 Bluetooth

A tecnologia Bluetooth [17] é definida para suportar comunicações sem fio. Suas especificações foram publicadas em 1999. Desenvolvida pelo consórcio denominado *Special Interest Group* [11], composto principalmente por as seguintes empresas:

- Nokia Mobile Phones
- Ericsson Mobile Communications AB
- IBM Corporation
- Intel Corporation
- Toshiba Corporation

A tecnologia Bluetooth é designada para ser usada em enlaces de curto alcance, de 10 metros para dispositivos de classe 2 e 100 metros para dispositivos de classe 1, conectando duas ou mais estações móveis. Onde a classe do dispositivo é definida pelo alcance máximo obtido no enlace. Esta tecnologia provê conexões ponto a ponto entre duas estações ou conexões ponto a multiponto onde o meio é compartilhado por várias estações.

O Bluetooth opera na frequência de 2,4 GHz, podendo alcançar uma taxa de transmissão máxima de 1 Mbps, usando as tecnologias TDMA ou FDMA-TDD para acesso ao meio físico.

Os dispositivos Bluetooth podem compartilhar conexões através de *piconets* e *scatternets*, conforme Figura 1. Numa *piconet* uma das estações atua como mestre e as outras como escravos, de modo que a rede é constituída por um único mestre e no máximo sete escravos. O mestre é definido como o nó que inicia a conexão. Um escravo pode ser sincronizado com outra *piconet*: A estação que é mestre em uma *piconet* pode ser escrava em outra *piconet*. Neste sentido, múltiplas redes com cobertura superposta podem constituir uma *scatternet*. Numa *scatternet*, dispositivos em diferentes *piconets* não são sincronizados em frequência ou tempo.

3.4.1 Pilha Bluetooth

Um exemplo de pilha de protocolos Bluetooth é mostrada na figura 2. Nela pode-se observar protocolos específicos do Bluetooth, como L2CAP, e protocolos não específicos, como TCI/IP e PPP. Os protocolos podem ser divididos em quatro grupos de acordo com seus propósitos.

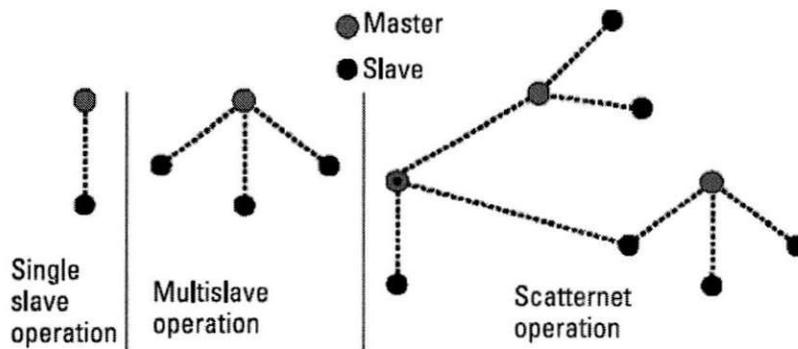


Figura 1: Formas de interconexão entre estações

- Protocolos de núcleo (*baseband*, LMP, L2CAP e SDP): protocolos desenvolvidos pelo Bluetooth SIG.
- Protocolos para substituição de cabos(RFCOMM): é constituído pelo Bluetooth SIG, mas é baseado no padrão ETSI TS 07.10 [18].
- Protocolos de controle e especificação de telefonia (TCS, BIN, comandos AT): também é constituído pelo Bluetooth SIG, mas é baseado na recomendação ITU-T Q.931.
- Protocolos adotados(PPP, UDP/TCP/IP, WAP/WAE, OBEX, vCard, vCal e IrMC).

L2CAP

A camada L2CAP multiplexa camadas superiores numa única conexão ACL (*Asynchronous ConnectionLess*) entre dois dispositivos e, no caso do dispositivo mestre, encaminha os dados ao escravo correto. Ela também segmenta e junta dados em pacotes que se ajustam a máxima carga útil do HCI, que é responsável pela conexão das camadas mais altas de um nó às camadas mais baixas do dispositivo Bluetooth.

RFCOMM

A camada RFCOMM emula o padrão RS232 de 9 pinos sobre um canal L2CAP. Ela é baseada no padrão TS 07.10 para emulação da interface RS232. O padrão TS 07.10 inclui a habilidade de multiplexar várias portas seriais emuladas em uma única conexão usando diferentes identificadores de conexão para cada porta. Entretanto, cada sessão TS 07.10 pode estar conectada a um canal L2CAP de modo a se comunicar com um único dispositivo. Um dispositivo

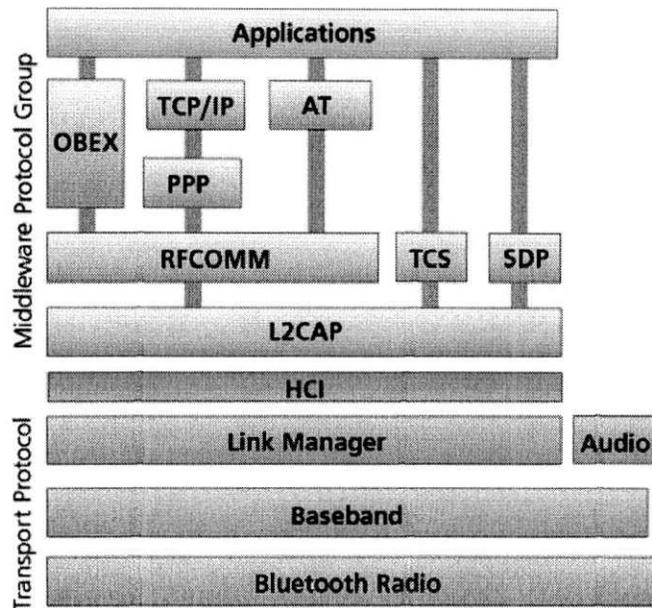


Figura 2: Pilha de Protocolos Bluetooth

mestre deve ter sessões RFCOMM separadas rodando, para cada escravo que requirite uma conexão pela porta serial.

OBEX

O padrão OBEX foi desenvolvido pela *Infrared Data Association* (IrDA) para facilitar operações comuns aos dispositivos usando infra-vermelho. Ao invés de desenvolver um novo padrão, o Bluetooth SIG tornou algumas implementações obrigatórias e usou o padrão da IrDA nos perfis de transferência de arquivos, sincronização e troca de objetos. O padrão OBEX permite aos usuários receber e enviar objetos, criar e apagar pastas e objetos, e especificar o diretório de trabalho no dispositivo remoto.

Além disso, a especificação compreende a interface controladora de *hosts* (HCI), que provê uma interface de comandos para o controlador *baseband*, o gerenciador do enlace (LMC - *Link Manager Controller*) e acesso ao estado do hardware e registradores de controle.

3.4.2 Aplicações

Existem várias aplicações disponíveis a partir da implementação *Open Source* do Bluetooth (Bluez, [9]) para Linux, as quais são descritas sucintamente a seguir:

- SP (*Serial Port*): Permite a um dispositivo Bluetooth emular um cabo

serial RS232, permitindo conexões seriais entre dispositivos.

- DUN (*Dial Up Network*): usado em dois casos principais, para uso de um telefone celular ou modem por um computador como modem sem-fio para conectar à internet através de um servidor de acesso discado, ou usar outros serviços de acesso discado, e uso de um telefone celular ou modem por um computador para receber chamadas de dados.
- LAN (*Local Area Network*): esta aplicação define como os dispositivos Bluetooth conectam-se a redes locais usando o protocolo ponto a ponto (PPP). Esta aplicação permite que dispositivos Bluetooth se conectem a outros dispositivos Bluetooth que estejam participando de uma LAN. Após a conexão, os dispositivos poderão acessar nós da rede, como impressoras.
- FAX: Envio de Fax a partir da conexão Bluetooth.
- OPUSH (*OBEX Push*): permite a transferência de arquivos entre dispositivos Bluetooth.
- FTRN (*File Transfer*): esta aplicação permite que um dispositivo Bluetooth acesse um sistema de arquivos, crie ou apague arquivos e pastas, ou transfira arquivos para outro dispositivo.
- HS (*Headset*): esta aplicação permite a transferência de áudio entre um dispositivo Bluetooth, como um telefone celular ou um PDA, e um fone de ouvido. Os usos mais comuns desta aplicação incluem chamadas telefônicas e gravação de voz usando um fone sem fio.
- HF (*Handsfree*): esta aplicação permite, entre outras coisas, a comunicação de um veículo com um dispositivo Bluetooth.
- NAP (*Network Access Point*) : um dispositivo mestre atua como *proxy*, roteador ou ponte entre uma infraestrutura de rede existente (tipicamente uma LAN) para clientes Bluetooth (no máximo 7).
- GN (*Group ad-hoc Network*): um nó atua, em uma rede no estilo *peer-to-peer*, como encaminhador de pacotes, interconectando no máximo 7 escravos a uma rede *peer-to-peer* externa.
- HID (*Human Interface Device*): definido para a interconexão de dispositivos como teclados e *mouses* a um micro computador.
- CIP (*Common ISDN Access*): permite o uso de conexões ISDN (*Integrated Service Digital Network*) a partir do enlace Bluetooth.

- CTP (*Cordless Telephony*): a partir desta aplicação, dispositivos Bluetooth podem atuar como telefones sem fio quando estiverem próximos a um ponto de acesso Bluetooth, ou a outro dispositivo Bluetooth que tenha acesso à rede de voz.
- A2SRC (*Audio Source*): aplicação usada por uma fonte sonora no perfil A2DP (*Advanced Audio Distribution Profile*), que é um perfil para a transferência de áudio de alta qualidade a partir do enlace Bluetooth, no envio do som.
- A2SNK (*Audio Sink*): aplicação usada no perfil A2DP para recebimento de som.

3.5 Wi-Fi

As redes Wi-Fi são usadas para a substituição de fios em uma infra-estrutura de rede Ethernet tradicional. As redes Wi-Fi são definidas segundo os padrões 802.11a, 802.11b e 802.11g definidos pelo IEEE. Cada padrão apresenta suas peculiaridades, sendo descritos mais precisamente nas seções abaixo.

3.5.1 802.11a

Devido à grande procura por mais largura de banda, e o número crescente de tecnologias a trabalhar na banda 2,4GHz, foi criado o 802.11a para WLAN a ser utilizado nos Estados Unidos. Este padrão utiliza a frequência de 5GHz, onde a interferência não é problema. Graças à frequência mais alta, o padrão é quase cinco vezes mais rápido, atingindo taxas da ordem de 54 Mbps.

Note que esta é a velocidade de transmissão nominal que inclui todos os sinais de modulação, cabeçalhos de pacotes, correção de erros, etc. A velocidade real das redes 802.11a é de 24 a 27 megabits por segundo, pouco mais de 4 vezes mais rápido que no 802.11b. Outra vantagem é que o 802.11a permite um total de 8 canais simultâneos, contra apenas 3 canais no 802.11b. Isso permite que mais pontos de acesso sejam utilizados no mesmo ambiente, sem que haja perda de desempenho.

O grande problema é que o padrão é mais caro. Além disso, por utilizar uma frequência mais alta, os transmissores 802.11a também possuem um alcance mais curto, teoricamente metade do alcance dos transmissores 802.11b. Desta forma torna-se necessário a utilização de mais pontos de acesso para cobrir a mesma área, o que contribui para aumentar ainda mais os custos.

3.5.2 802.11b

A camada física do 802.11b utiliza espalhamento espectral por seqüência direta (DSSS) utilizando transmissão aberta de rádio, operando nas freqüências de 2,4000 a 2,4835 GHz, usando um total de 14 canais com uma capacidade de transmissão de 11 Mbps.

O alcance em ambiente abertos pode chegar a 450 metros, enquanto que em ambiente fechados ele cai para aproximadamente 50 metros. A taxa de transmissão pode ser reduzida a 5,5 Mbps ou menos, dependendo das condições do ambiente no qual as ondas estão se propagando (paredes, interferências, etc).

A topologia das redes 802.11b é semelhante a das redes de par trançado, com um *hub* central. Neste caso o dispositivo Wi-Fi é denominado ponto de acesso.

3.5.3 802.11g

O padrão 802.11g é uma extensão do padrão 802.11b. Ele aumenta a taxa de transmissão para 54 Mbps dentro da banda de 2,4 GHz usando a tecnologia OFDM.

O padrão 802.11g traz suporte nativo ao padrão de segurança WPA [12], que se encontra implementado em alguns produtos 802.11b. O alcance e aplicações também são basicamente os mesmos do 802.11b e ele é claramente uma tecnologia que, aos poucos, irá substituir as implementações do 802.11b, já que mantém a compatibilidade e oferece maior velocidade.

4 Aplicações utilizadas

Nesta seção serão apresentadas as principais aplicações utilizadas para a realização do presente trabalho.

4.1 Iptables

Concebido por Rusty Russell em 1999, o Iptables vem a ser a maior referência de *firewalls* para Linux da atualidade, sendo o *firewall* padrão em todas as distribuições Linux sob *kernels* com versões a partir do 2.4.

O Iptables, além de realizar suas tarefas de forma veloz, segura, eficaz e econômica, tanto no aspecto financeiro, quanto no requerimento de hardware, por utilizar o mínimo de recursos possíveis de um computador, abre um amplo leque de possibilidades, tais como:

- Filtros de pacotes com controle de estado de conexão (*Stateful Package Filtering*);
- Implementação de NAT com suporte a SNAT e DNAT;
- Mascaramento de conexões;
- Desenvolvimento de QoS e TOS sobre o tráfego;
- Redirecionamento de endereços e portas;
- Monitoramento de tráfego;
- Controle de tráfego P2P via módulo *string*;
- Controle de tráfego por endereçamento IP, MAC e *host name* (origem/destino);
- Demais módulos externos para expansão de funcionalidades;

Dentre as possibilidades disponíveis pelo Iptables, neste trabalho são usadas apenas o NAT, descrito a seguir, com mascaramento de conexões.

4.2 NAT

O NAT é uma série de tarefas que um roteador deve realizar para converter endereços IPs entre redes distintas. O NAT, como o próprio nome diz, faz a tradução de endereços IP e portas TCP de uma rede privada para a Internet. Por exemplo, um pacote enviado de por um nó numa rede privada tem seu endereço IP trocado pelo IP do servidor para ser enviado à rede externa, se um pacote é destinado a rede privada, ele é endereçado ao servidor, o qual por sua vez encaminha o pacote a porta a qual o dispositivo esteja conectado.

O NAT define algumas outras tarefas mais complexas que devem ser realizadas pelo roteador para que ele funcione corretamente, entre elas:

- Os pacotes de dados TCP/IP tem um campo para verificação de erro: o *checksum* IP. Como esse campo é dependente dos dados contidos no cabeçalho do pacote de dados (que inclui os endereços IP de origem e destino) ele deve então ser modificado corretamente pelo roteador.
- O roteador com NAT deve armazenar a informação sobre qual servidor na Internet os computadores locais estão acessando para poder encaminhar as respostas que chegarem. Porém pode haver instantes onde duas máquinas da rede local podem estar acessando o mesmo servidor na Internet. Nesse caso, não basta armazenar os endereços IPs das máquinas que estão sendo acessadas, deve-se também armazenar a porta de origem de cada máquina local para que o roteador possa encaminhar corretamente as respostas.

- A solução apresentada no item anterior também traz um problema: a porta de origem de cada máquina é definida por cada uma dessas máquinas e é aleatória. Portanto existe a possibilidade de duas máquinas na rede local acessarem o mesmo servidor e escolherem a mesma porta de origem. Neste caso o roteador é obrigado a modificar também a porta de origem em uma das conexões, manter essa informação na memória e modificar a porta de destino da resposta.
- Alguns protocolos, como o ICMP e protocolos de roteamento, carregam dentro do campo de dados informações sobre o endereço da máquina de origem. O roteador com NAT deve então observar esses pacotes e modificar essas informações adequadamente. Como cada protocolo utiliza essa informação de maneira diferente o roteador deve ter conhecimento específicos sobre o protocolo, caso contrário o protocolo não irá funcionar.

4.3 DHCP

O protocolo para configuração dinâmica de nós da rede é usado para atribuição de endereços IP e outras configurações de rede (máscara de sub-rede, DNS, endereço de *broadcast*) aos dispositivos de uma rede.

Um cliente configurado para usar DHCP envia uma requisição em *broadcast* destinada a um servidor DHCP pedindo um endereço. O servidor DHCP, então atribuirá um endereço IP ao cliente por um dado intervalo de tempo, que pode ser especificado no servidor. O servidor DHCP reduz o tempo necessário para configurar clientes, e permite que ele mude de uma rede para outra sendo configurado com IP, *gateway* e máscara de sub-rede apropriados. Para os provedores de acesso o DHCP, conserva o número limitado de endereços que podem ser usados. O servidor DHCP pode atribuir endereços IP estáticos para dispositivos específicos, através do endereço MAC.

4.4 Bridge

As pontes Ethernet implementam o padrão ANSI/IEEE 802.1d e são uma forma de conectar redes de modo a formar uma rede maior. Uma ponte é o meio de conectar dois segmentos de rede separados em uma única rede de forma transparente às aplicações. Os pacotes são encaminhados com base nos endereços Ethernet, ao invés dos endereços IP. Desde que o roteamento é realizado na camada de enlace de dados, todos os protocolos podem passar de maneira transparente pela ponte.

Uma ponte em Linux é mais poderosa que uma ponte puramente em hardware, pois ela também pode filtrar e formatar o tráfego. As restrições quanto ao

uso de pontes são as seguintes:

- Todos os dispositivos devem ter o mesmo MTU, pois as pontes não fragmentam pacotes.
- Os dispositivos devem se parecer com dispositivos Ethernet, isto é, ter endereços fonte e destino com 6 *bytes*.
- Suportar operação promíscua. A ponte recebe todo o tráfego da rede, não só o tráfego a ela destinado.
- Permitir *spoofing* do endereço fonte. A ponte deve poder enviar dados sobre a rede como se eles originassem do outro nó.

4.5 BlueZ

A pilha oficial de protocolos Bluetooth para Linux, BlueZ, trata-se de um projeto destinado a implementação do padrão Bluetooth no Linux. O código deste projeto é licenciado sob a GNU GPL [2] e inclui suporte para *kernels* 2.4 e 2.6.

O projeto BlueZ consiste de vários módulos separados que implementam quase todas as funções previstas pelo SIG. O projeto BlueZ trabalha em diferentes arquiteturas e distribuições de Linux, entre as arquiteturas, temos:

- Intel e AMD x86
- AMD64 (x86-64)
- SUN SPARC 32/64bit
- PowerPC 32/64bit
- Intel StrongARM e XScale
- Hitachi/Renesas SH processors
- Motorola DragonBall

Entre as distribuições de Linux que tem suporte ao BlueZ, pode-se destacar as seguintes:

- Debian GNU/Linux [19]
- Fedora Core / Red Hat Linux [20]
- SuSE Linux [21]
- Mandrake Linux [22]

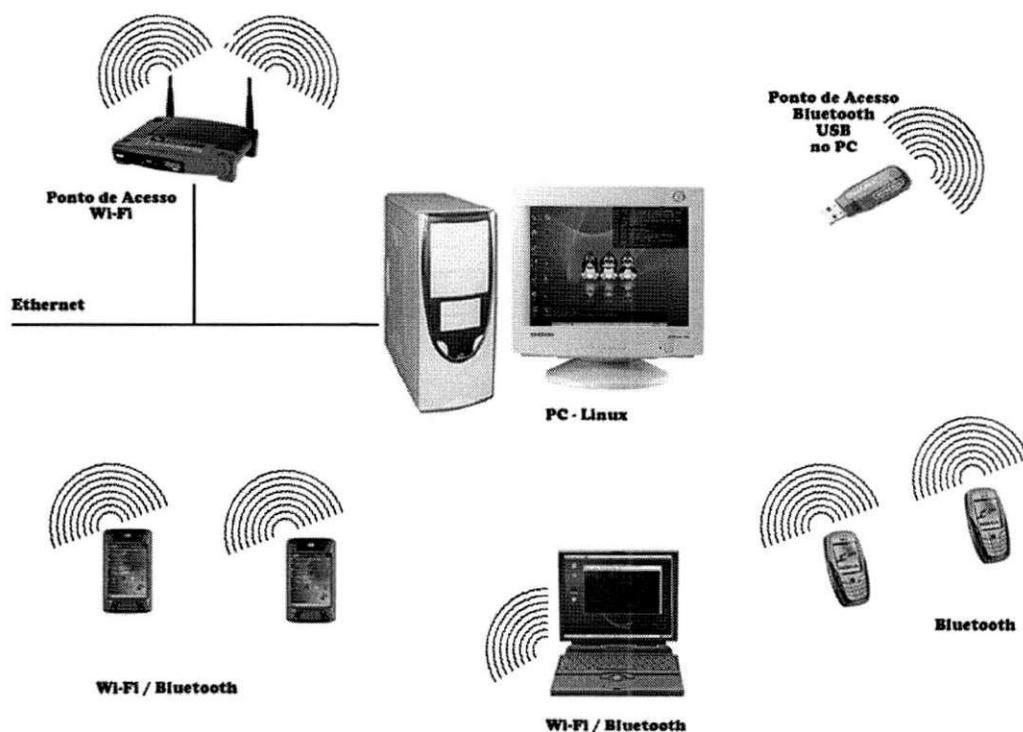


Figura 3: Topologia da rede

5 Arquitetura Proposta

A arquitetura do projeto consiste de um *gateway* Linux, com uma placa Wi-Fi, uma placa Ethernet e um dispositivo Bluetooth USB, um ponto de acesso Wi-Fi 54g e os dispositivos móveis que usarão as conexões tanto sem fio quanto pela rede Ethernet existente, conforme apresentado na Figura 3.

O *gateway* proposto torna-se transparente ao tipo de conexão, ou seja, a conexão pode ser via *Ethernet*, *Bluetooth*, *Wi-Fi* e extensível às novas tecnologias para redes sem fio que venham surgir. De modo, que ao servidor Linux podem se conectar dispositivos diversos, como *laptops*, *smart phones*, *PDA*s e computadores pessoais.

Conforme observado na introdução o *gateway* Linux implementado neste projeto foi usado no M-VoIP, Figura 4. No M-VoIP não são usadas conexões Ethernet, apenas Bluetooth e Wi-Fi.

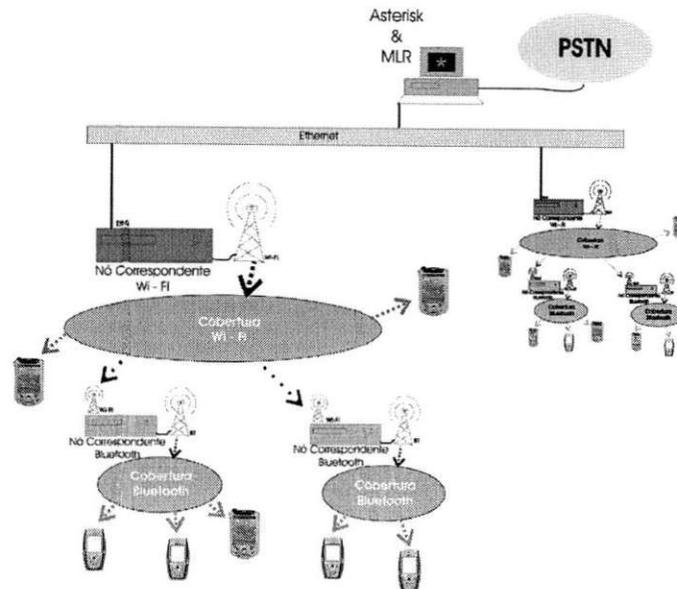


Figura 4: Arquitetura M-VoIP

6 Implementação

Para a implementação da arquitetura apresentada, as seguintes etapas foram seguidas.

6.1 Configuração das máquinas

As duas máquinas disponíveis para o projeto M-Voip apresentam as seguintes configurações:

- Processador AMD Sempron 2400+
- 256MB de RAM
- HD de 40 GB
 - Com uma partição NTFS de 25GB com o Windows XP
 - Uma partição EXT3 de 15GB com o Fedora Core 3
- Placa Wi-Fi DWL-G520+ 802.11g/2.4GHz
- Bluetooth USB Sitecom

6.2 O Sistema Operacional Linux

O sistema operacional Linux foi escolhido por ser um sistema que vem se tornando cada vez mais utilizado, não só por usuários corporativos, quanto por usuários domésticos. Além do fato de ser um sistema de código aberto e disponibilizar soluções fáceis de serem implementadas, com uma grande quantidade de aplicativos disponíveis, principalmente quando se trata de aplicações para redes.

A versão de Linux escolhida para ser instalada foi a versão *Fedora Core 3* [20] que é baseada no Red Hat. Que se trata de uma versão de distribuição livre e de grande aceitação no mundo Linux, além de dispor de grande quantidade de aplicações disponibilizadas na Internet. Cabe dizer, que as soluções aqui apresentadas são aplicáveis a outras distribuições Linux, com algumas modificações possivelmente.

6.2.1 Instalação do Fedora Core 3

Uma vez que as máquinas já tinham o Windows XP instalado, tornou-se necessária a criação de uma partição estendida ext3, a partir do Software Partition Magic, de 15GB para a instalação do Linux.

Com a partição criada dá-se o *boot* pelo CD 1 do Fedora Core 3, ver figura 5, iniciando-se a instalação. Em seguida aparece uma tela pedindo para checar os CDs de instalação, ver figura 6, trata-se de uma etapa um pouco lenta que pode ser pulada, dependendo da confiabilidade da fonte de obtenção dos CDs de instalação.

Depois é escolhida a linguagem usada no processo de instalação, figura 7. Em seguida é escolhido o tipo de instalação, figura 8, no caso do projeto em questão, o tipo escolhido é o personalizado (*Custom*).

Conforme observado na figura os tipos de instalação possíveis, são:

- *Personal Desktop*: É o tipo de instalação padrão, que proporciona um ambiente de trabalho gráfico com aplicações para escritório, aplicações Internet e softwares multimídia.
- *Workstation*: Este tipo de instalação inclui os pacotes instalados na instalação *Personal Desktop*, junto com softwares para desenvolvimento de aplicações e administração do sistema.
- *Server*: Este tipo de instalação prove servidores de rede como o Apache e o Samba além de ferramentas administrativas. Porém, a instalação não vem com interface gráfica.

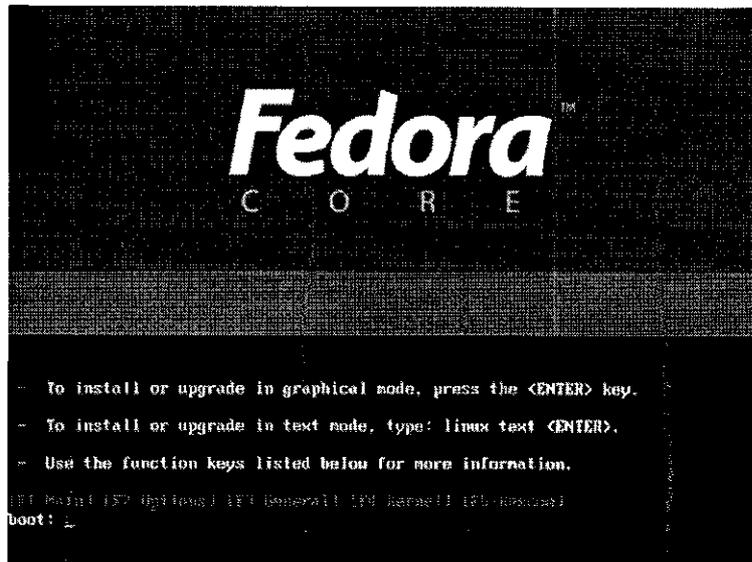


Figura 5: Tela de Boot

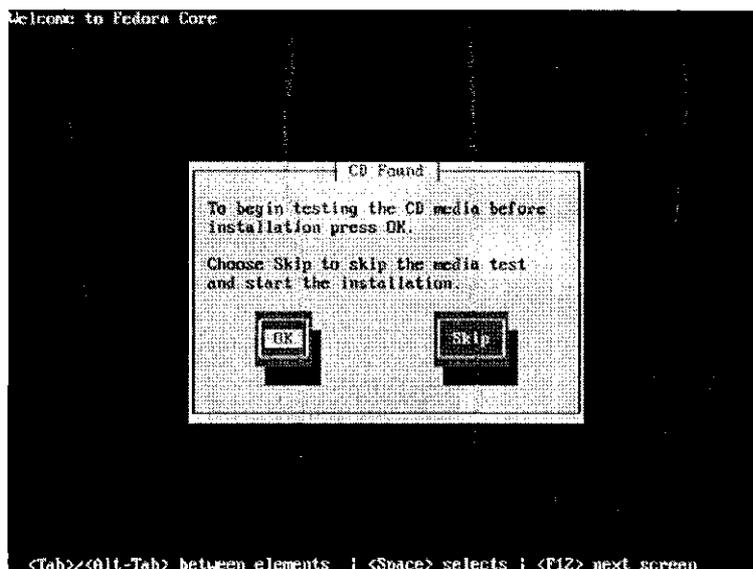


Figura 6: Testar CDs

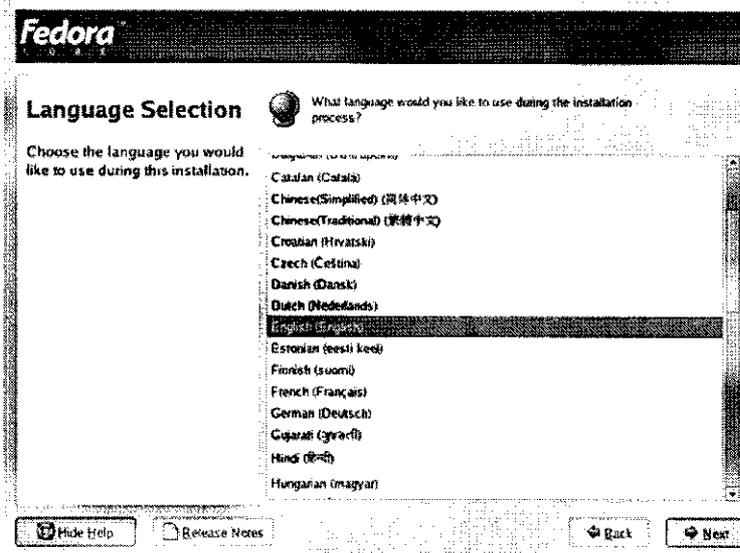


Figura 7: Linguagem de instalação

- *Custom*: Neste tipo de instalação o usuário seleciona os pacotes a serem instalados.

Após selecionado o tipo de instalação, aparece um diálogo questionando sobre o tipo de particionamento a ser escolhido na instalação, neste ponto escolhe-se particionamento automático, figura 9. Na próxima tela, figura 10, escolhe-se o modo de particionamento. Uma vez que foi criada uma partição Linux com o Partition Magic, pode-se escolher a opção "Remove all Linux partitions on this system", de modo que o instalador do Fedora já criará as partições corretamente.

Em seguida configura-se o Grub que é o *bootloader* do Linux. O Grub é instalado na MBR (*Master Boot Record*) e reconhece tanto partições Linux quanto partições Windows.

Uma vez instalado o Grub, o próximo passo trata-se da configuração da interface de rede Ethernet, figura 11. Para configurar a rede seleciona-se a opção *manually* e os seguintes parâmetros são adicionados:

- Máquina 1:

Hostname: sip.dee.ufcg.edu.br

IP Address: 150.165.61.141

Netmask: 255.255.255.0

Gateway: 198.168.1.1

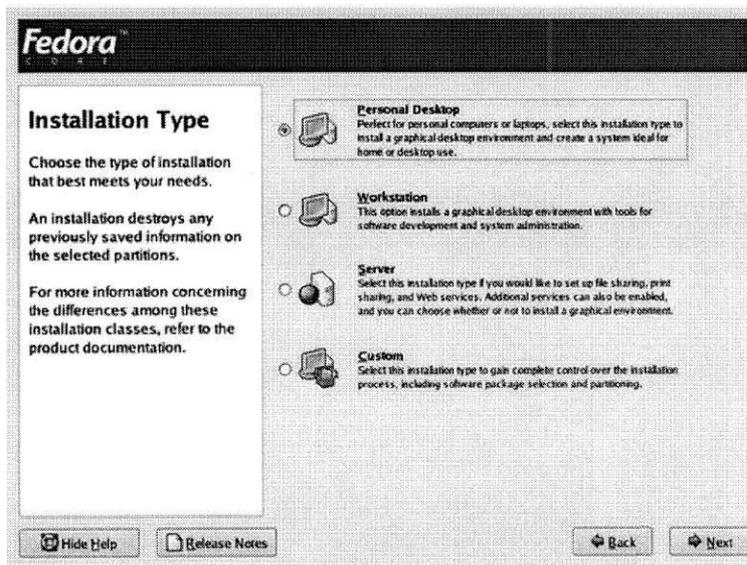


Figura 8: Tipo de instalação

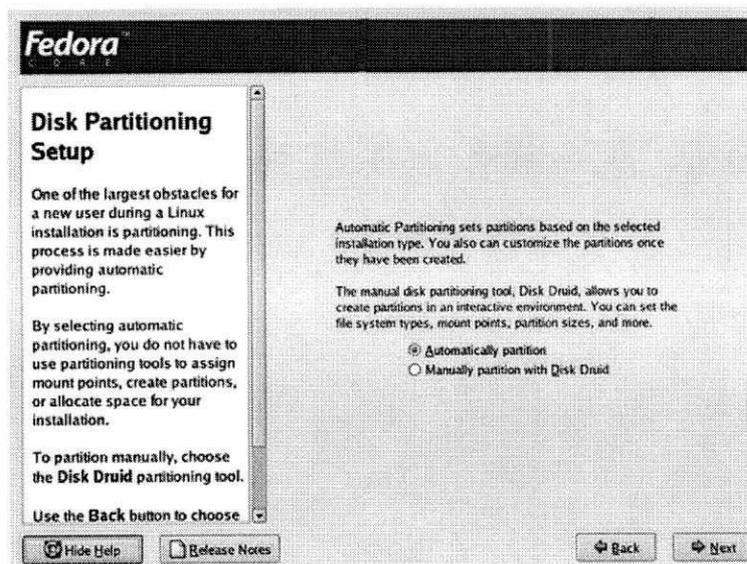


Figura 9: Escolha do tipo de Particionamento

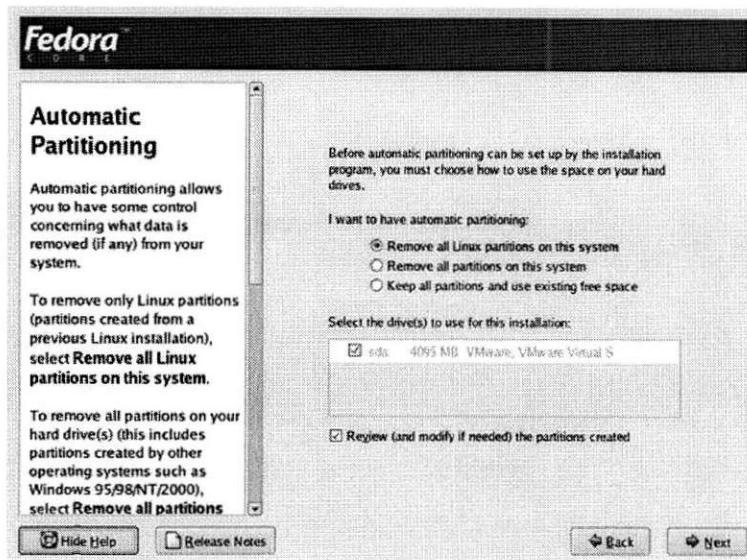


Figura 10: Escolha do tipo de Particionamento

Primary DNS: 150.165.61.3

- Máquina 2:

Hostname: rtp.dee.ufcg.edu.br

Na máquina 2 só foi configurado o *Hostname*, pois ele só usa a interface de rede Wi-Fi.

Em seguida aparece uma tela perguntando sob configurações de segurança, ver figura 12. Como a rede do DEE já se encontra protegida por um *Firewall* relativamente confiável, esta opção é desabilitada.

A seguir, configura-se a senha de administrador do sistema a qual será utilizada para a realização de tarefas administrativas e deve ser restrita aos usuários com conhecimento avançado do sistema. Este passo é observado na figura 13.

Em seguida, como foi utilizada a instalação personalizada, aparece uma tela com os pacotes disponíveis para serem instalados, ver figura 14. Uma vez concluída a seleção dos pacotes tem-se a etapa de instalação propriamente dita, ver figura 15, na qual os pacotes são copiados dos CDs para o disco rígido da máquina e configurados.

Após terminada a instalação retira-se o último CD e reinicia-se o computador. Após a reinicialização surge uma tela de boas vindas, ver figura 16, com alguns passos adicionais que necessitam ser realizados. Os passos são os seguintes:

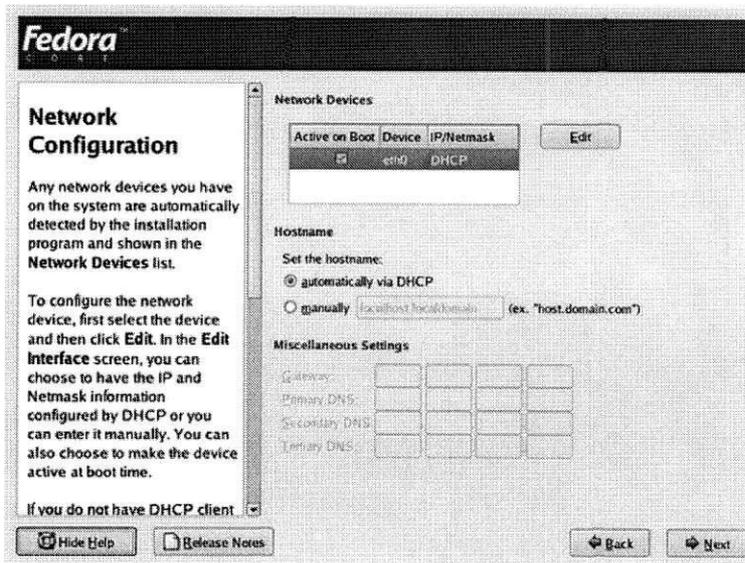


Figura 11: Configuração da interface de rede

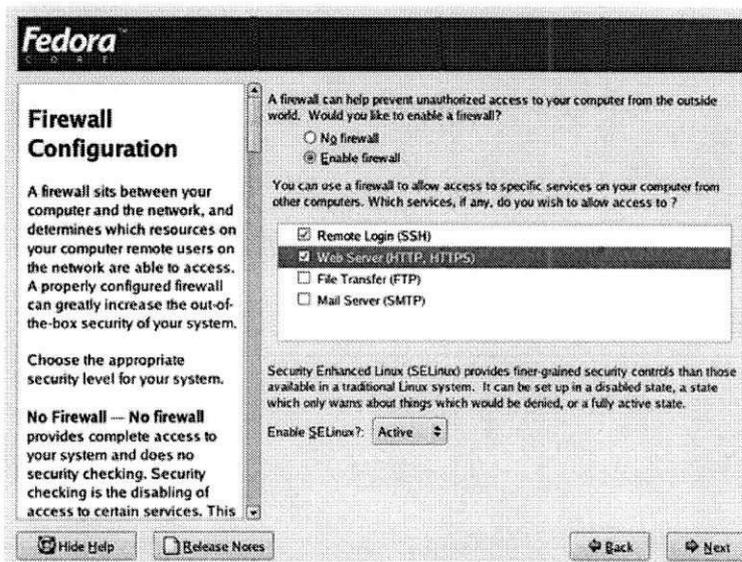


Figura 12: Segurança

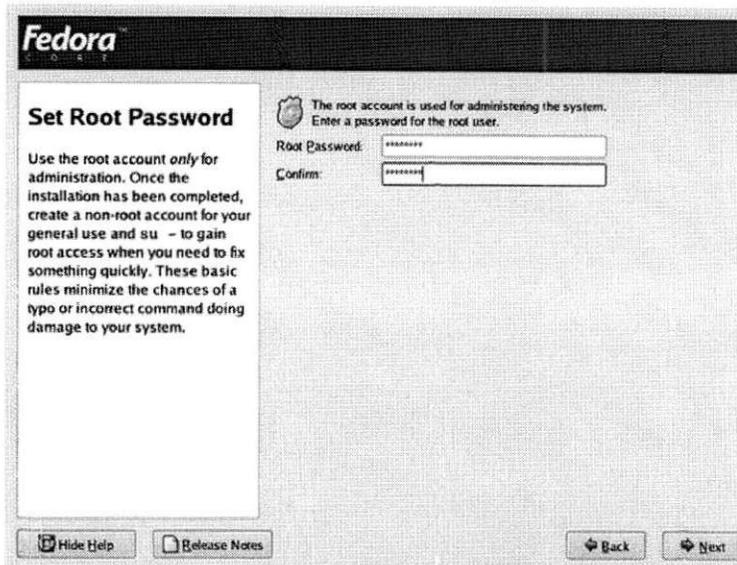


Figura 13: Senha do Administrador

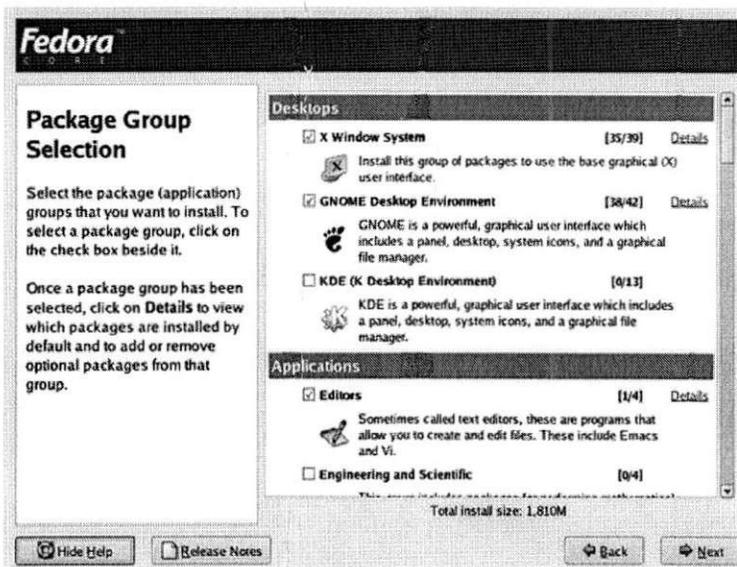


Figura 14: Seleção dos pacotes

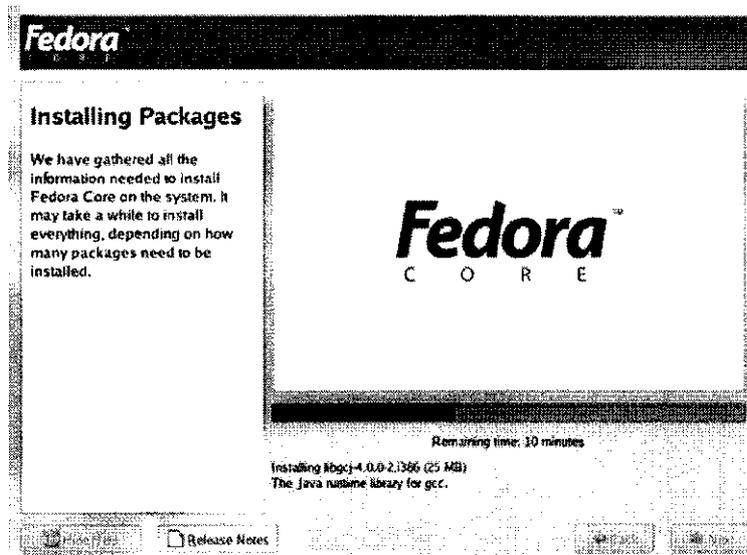


Figura 15: Instalação dos Pacotes

- Aceitação do acordo de licença;
- Configurações de data e tempo;
- Resolução da tela;
- Criação de usuários
- Detecção da placa de som;
- Instalação de CDs adicionais.

Finalmente a instalação é concluída e o usuário pode logar e usar o Linux. Um passo adicional necessita ser realizado apenas na configuração da rede, uma vez que as rotas para encaminhamento de pacotes na rede não ficam completas. Para a solução deste problema deve-se editar o arquivo `/etc/sysconfig/network-scripts/ifcfg-eth0` adicionando as duas linhas seguintes:

```
/sbin/route add -net 198.168.1.0 netmask 255.255.255.0 dev eth0
/sbin/route add default gw 198.168.1.1
```

6.2.2 Instalação do driver da placa Wi-Fi

O *driver* da placa de rede Wi-Fi não é distribuído junto com a versão do Linux instalada. Os arquivos necessários para a sua instalação são obtidos da Internet, nos sítios www.houseofcraig.net e <http://rhlx01.fht-esslingen.de/andi/acx100/>.

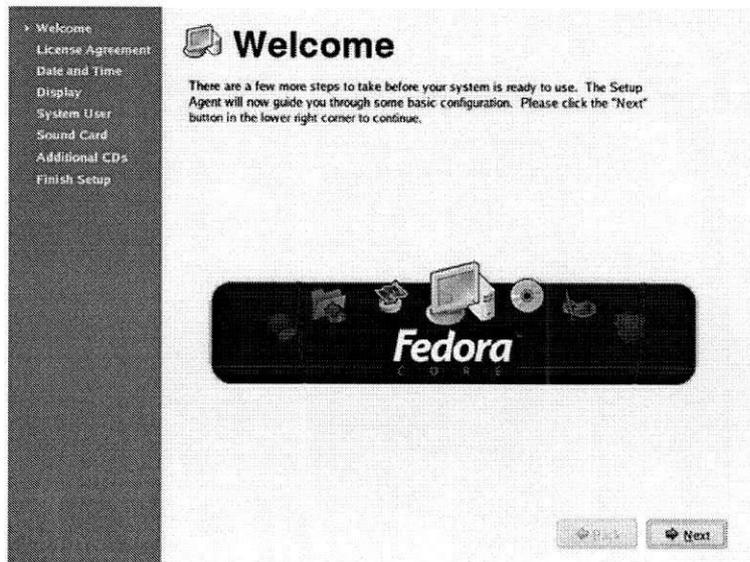


Figura 16: Tela de boas vindas

Após os arquivos serem obtidos, descompacta-se o arquivo `acx100-0.2.0pre8_plus_fixes_51.tar.bz2` com o comando:

```
tar -xjvf acx100-0.2.0pre8_plus_fixes_51.tar.bz2
```

A seguir, instala-se o *firmware* do driver ACX, com o comando:

```
tar Pjxf acx_firmware.tar.bz2
```

Este comando extrai os arquivos na pasta `/usr/share/acx` com o *firmware* usado pela placa Wi-Fi. Em seguida, com os seguintes comandos, instala-se o *driver* da placa.

```
cd acx100-0.2.0pre8_plus_fixes_51
make
make install
```

O comando *make* compila os arquivos fonte gerando o *driver* e o *make install* copia o driver gerado para a pasta `/lib/modules/`uname -r``, onde o comando `'uname -r'` retorna a versão do *kernel* em uso.

Com o driver instalado, deve-se editar o arquivo `start_net` com as informações relativas a rede Wi-Fi. No campo *ESSID* coloca-se o nome da rede Wi-Fi em uso, no caso em questão o nome é **mvoip**. A taxa de transmissão da rede também é alterada para 54 Mbps. O canal usado foi o 11.

Em seguida são configurados os endereços IP, a máscara de sub-rede, e o *gateway*. Como na interface de rede Ethernet, deve-se adicionar rotas extras para que a interface de rede Wi-Fi possa acessar a rede externa. Estas configurações podem ser visualizadas no script abaixo.

```
#!/bin/bash
```

```
#####  
# start_net script #  
# acx100 project #  
# acx100.sourceforge.net #  
# edited by arnie <urnotwelcome@gmx.de> #  
#####  
# with modifications by craig #  
# summary at end of file #  
#####
```

```
# Please edit below
```

```
# syntax is: VARIABLENAME=VALUE, with _no_ spaces in between  
# make sure to _preserve_ any double-quotes ("  
# text beginning with the comment delimiter (#) is ignored  
# make sure to _preserve_ at least one space before any  
# comment delimiters (#) that do not begin a line  
# "uncommenting" a line means to remove it's leading "#" character
```

```
ESSID="mvoip" # THIS IS CASE SeNsItIvE!! any == associate to any ESSID  
# Default rate configured as 11Mbps to not cause speed problems (while  
# using auto rate) or connection problems (while not using auto rate)  
# with non-22Mbps hardware...
```

```
RATE=54M
```

```
AUTORATE=1 # only disable auto rate if you know what you're doing...
```

```
CHAN=11 # it's useful to try to stick to channels 1, 6 or 11 only,  
since these don't overlap with other channels
```

```
#SHORTPREAMBLE=1 #set a value of 1 in order to force  
"Short Preamble" instead of peer autodetect
```

```
#TXPOWER=20 # 0..20 (dBm) (18dBm is firmware default) overly  
large setting might perhaps destroy your radio eventually!
```

```
MODE=Managed # Managed for infrastructure, Ad-hoc for
```

peer-to-peer. NOTE: Auto mode is not supported any more,
you HAVE to select a specific mode!

DEBUG=0xb # 0xffff for maximum debug info, 0 for none

WEP Key(s)

ascii keys (passphrase) should look like this: KEY="s:ascikey"

hex keys should look like this: KEY="4378c2f43a"

most wep users will want to use this line

KEY=""

.
.
.

#IP address

USE_DHCP=0 # set to 1 for auto configuration instead of fixed IP setting

IP=150.165.61.142 # set this if you did not set USE_DHCP=1

NETMASK=255.255.255.0 # set this if you did not set USE_DHCP=1

GATEWAY=198.168.1.1 # set this if you did not set USE_DHCP=1

LED_OFF=1 # set to 1 to turn off the power LED to save power

MTU_576=0 # set to 1 if you have buffer management problems

.
.
.

/sbin/route add -net 198.168.1.0 netmask 255.255.255.0 dev wlan0

/sbin/route add default gw 198.168.1.1

end of file

Com o arquivo *start_net* configurado, edita-se o arquivo *rc.local* inserindo
as três linhas seguintes no final do mesmo.

#Bring acx100 device up

cd /usr/local/acx/scripts

./start_net

Finalmente, digita-se os seguintes comandos no terminal para desabilitar a detecção automática pelo software de detecção de hardware (*kudzu*) do Fedora Core.

```
echo 'DEVICE=wlan0' > /etc/sysconfig/network-scripts/ifcfg-wlan0
kudzu
```

No *kudzu* escolhe-se a opção ignorar dispositivo. Com isto feito, reiniciliza-se o computador com a placa de rede Wi-Fi configurada corretamente.

6.2.3 Configuração do BlueZ

Para que o pilha de protocolos BlueZ funcione corretamente é necessária a alteração de dois scripts usados em sua configuração. Os scripts a serem modificados estão localizados na pasta */etc/bluetooth*. O primeiro deles é o *hcid.conf* e o segundo é o *pin*.

No *hcid.conf* são comentadas as linhas com respeito à autenticação e a segurança além de ser removida a opção do PC poder se comunicar como escravo, uma vez que o PC atuará como servidor e não como cliente. O conteúdo modificado deste script se encontra a seguir:

```
#
# HCI daemon configuration file.
# /etc/bluetooth/hcid.conf
# $Id: hcid.conf,v 1.4 2004/04/29 20:14:21 holtmann Exp $
#

# HCID options
options {
    # Automatically initialize new devices
    autoinit yes;

    # Security Manager mode
    # none - Security manager disabled
    # auto - Use local PIN for incoming connections
    # user - Always ask user for a PIN
    #
    security none;
```

```

# Pairing mode
# none - Pairing disabled
# multi - Allow pairing with already paired devices
# once - Pair once and deny successive attempts
pairing multi;

# PIN helper
pin_helper /usr/bin/bluepin;
}

# Default settings for HCI devices
device {
    # Local device name
    # %d - device id
    # %h - host name
    name "%h-%d";

# Local device class
    class 0x120104;

# Default packet type
    #pkt_type DH1,DM1,HV1;

# Inquiry and Page scan
    iscan enable; pscan enable;

# Default link mode
# none - no specific policy
# accept - always accept incoming connections
# master - become master on incoming connections,
#          deny role switch on outgoing connections
#
    lm accept, master;

# Default link policy
# none - no specific policy
# rswitch - allow role switch
# hold - allow hold mode
# sniff - allow sniff mode

```

```

# park    - allow park mode
#
lp rswitch,hold,sniff,park;

# Authentication and Encryption
#auth enable;
#encrypt enable;
}

```

No script *pin* coloca-se o código PIN usado caso algum dispositivo Bluetooth requirite uma autenticação durante o estabelecimento de uma conexão. Este script é encontrado abaixo:

```

#!/bin/bash
echo "1234"

```

6.2.4 Instalação do Familiar Linux

O Familiar Linux é uma distribuição Linux para PDAs baseada no Debian [19]. Assim como as distribuições de Linux para computadores que apresentam ambientes gráficos como o KDE e o Gnome, no Familiar Linux existem os ambientes gráficos Opie e GPE.

A diferença em relação aos computadores pessoais e que como o PDA utilizado, Ipaq H3900, apresenta uma memória reduzida (48MB) só se pode instalar um ambiente gráfico. De modo que foi escolhido o Opie por apresentar uma interface com o usuário mais sofisticada que as demais.

Os passos necessários para a instalação do Familiar Linux são descritos no sítio do projeto Familiar [24]. As etapas a serem seguidas são a instalação do *bootloader*, figura 17, e a instalação do Familiar Linux propriamente dita.

A instalação é realizada a partir da porta serial do PDA conectada a um computador pessoal através do Hyperterminal [26].

Concluída a instalação, o próximo passo consiste na atualização dos pacotes existentes na distribuição, para isso deve ser realizada uma conexão com a Internet. Esta conexão poderia ser feita através do Bluetooth, mas na instalação o Bluetooth não é configurado corretamente. Para resolver este problema conectase o PDA a Internet pela porta serial, com a execução dos comandos abaixo:

No PDA

comandos a serem inseridos

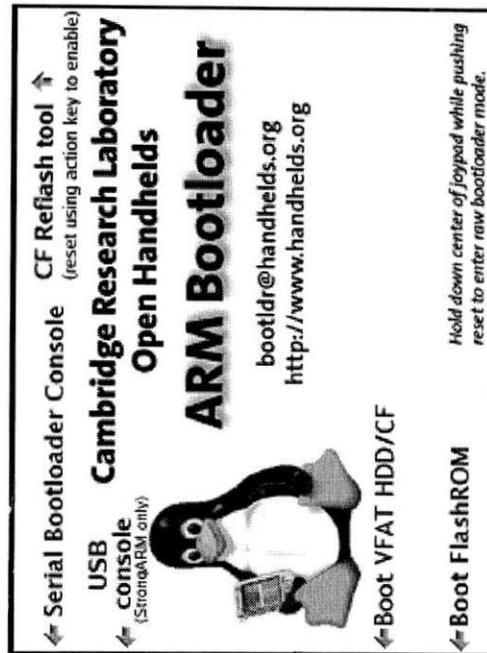


Figura 17: Familiar Linux *Bootloader*

No computador pessoal:

comandos a serem inseridos

Com a conexão realizada pode-se obter os pacotes através dos comandos:

```
ipkg update
ipkg upgrade
ipkg install nome_do_pacote
```

Após baixados os pacotes necessários o ambiente gráfico Opie, figura 18, já se pronto para ser usado no projeto como forma de validar a parte do *gateway* que usa conexão Bluetooth.

6.3 Implantação do Gateway Ethernet - Wi-Fi

Uma vez que a tecnologia Wi-Fi apresenta a mesma forma de conexão que uma conexão Ethernet, a configuração a seguir é aplicável para a atribuição de conexões tanto WI-FI quanto Ethernet.

Para a definição de um *gateway* no Linux define-se um *script*, figura 19, que executará as tarefas necessárias para uma configuração correta.

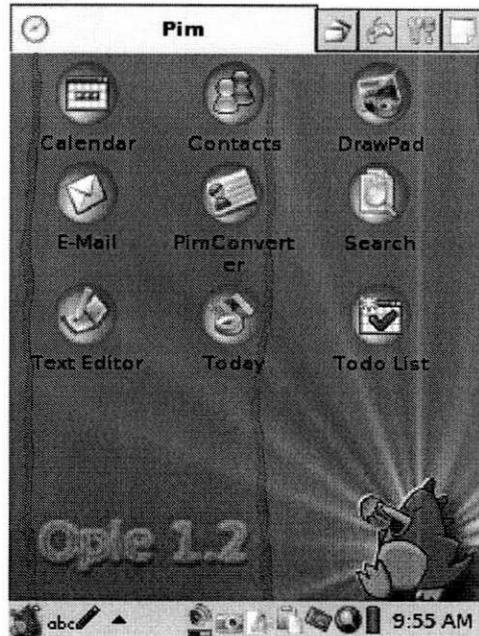


Figura 18: Opie

```

jluish@localhost:~/TCC/relato/Relatorio/scripts - Shell Núm. 2 - Konsole
Sessão Editar Ver Favoritos Configurações Ajuda
| /bin/sh
/sbin/ifconfig eth0 192.168.10.1 netmask 255.255.255.0 broadcast 192.168.10.255
# Flush all the rules in filter and nat tables
/sbin/iptables --flush
/sbin/iptables --table nat --flush
# Delete all chains that are not in default filter and nat table
/sbin/iptables --delete-chain
/sbin/iptables --table nat --delete-chain
# Set up IP FORWARDing and Masquerading
/sbin/iptables --table nat --append POSTROUTING --out-interface wlan0 -j MASQUERADE
# Assuming one NIC to local LAN
/sbin/iptables --append FORWARD --in-interface eth0 -j ACCEPT
# Enables packet forwarding by kernel (disabled by default)
echo 1 > /proc/sys/net/ipv4/ip_forward
# Create a route for internal packets
/sbin/route add -net 192.168.10.0 netmask 255.255.255.0 gw 198.168.1.1 dev eth0
-
1,1 All
Shell Shell Núm. 2 Shell Núm. 3

```

Figura 19: Script Gateway

IP	192.168.10.1
Máscara de sub-rede	255.255.255.0
Broadcast	192.168.10.255

Tabela 2: Configuração do *gateway*

No *gateway* configurado segundo o script mostrado na figura 19 a placa Ethernet é usada para a conexão com os demais dispositivos da rede e a placa Wi-Fi é usada para a comunicação com a Internet. O endereço IP, conforme observado na Tabela 2, do *gateway* trata-se de um endereço de classe C, uma vez que os IPs necessários para a rede privada usando o *gateway* são suficientes.

O encaminhamento de pacotes é realizado através da aplicação *Iptables*. Conforme pode ser observado no *script* da Figura 19 são usadas tabelas de NAT, que são responsáveis pelo encaminhamento dos pacotes provenientes dos nós a partir da interface Ethernet, para a Internet, a partir da Interface Wi-Fi (*wlan0*). Sempre que o Linux é reiniciado o *kernel* desabilita o encaminhamento de pacotes IP, de modo que deve-se habilitar este serviço toda vez que o *script* for executado. Para finalizar acrescenta-se uma rota para o *gateway* que tem conexão com o mundo externo (última linha do *script*).

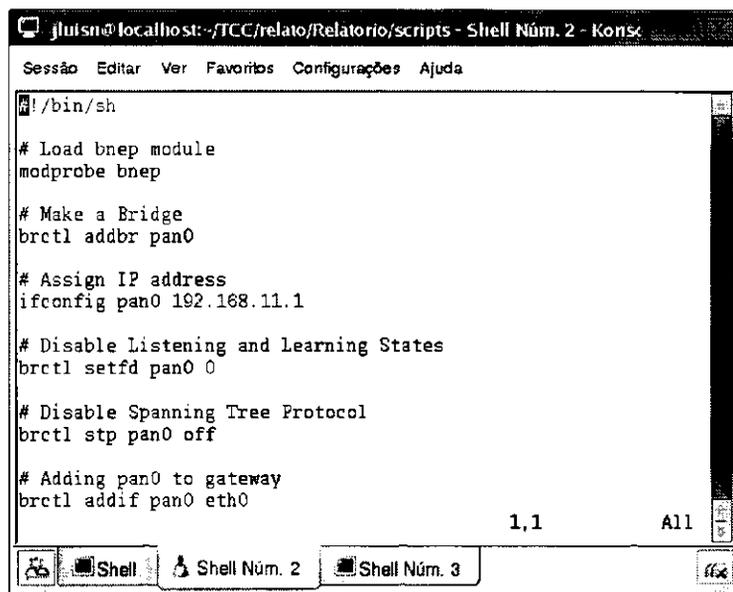
6.4 Implantação do Gateway Bluetooth

Para a implantação da interface Bluetooth no *gateway* usou-se o serviço NAP disponível na pilha BlueZ. Para usar este serviço, os seguintes pacotes têm que estar instalados.

- Bluez-bluefw (Bluetooth firmware loader)
- Bluez-hcidump (Bluetooth HCI protocol analyser)
- Bluez-libs (Bluetooth libraries)
- Bluez-pin (D-Bus Bluetooth PIN helper)
- Bluez-utils (Bluetooth utilities)

Uma vez que as conexões Bluetooth são *peer-to-peer*, deve-se configurar uma ponte por software de modo que as conexões Bluetooth sejam tratadas como uma única conexão. Para isto é necessária a instalação do aplicativo *bridge-utils*. Com este aplicativo instalado criar-se uma ponte, denominada de **pan0**, a partir do *script* visualizado na Figura 20.

Para que a ponte funcione corretamente, falta a criação do arquivo “dev-up”, ver Figura 21, na pasta `/etc/bluetooth/pan/` que é chamado pelo *daemon*



```
#!/bin/sh

# Load bnep module
modprobe bnep

# Make a Bridge
brctl addbr pan0

# Assign IP address
ifconfig pan0 192.168.11.1

# Disable Listening and Learning States
brctl setfd pan0 0

# Disable Spanning Tree Protocol
brctl stp pan0 off

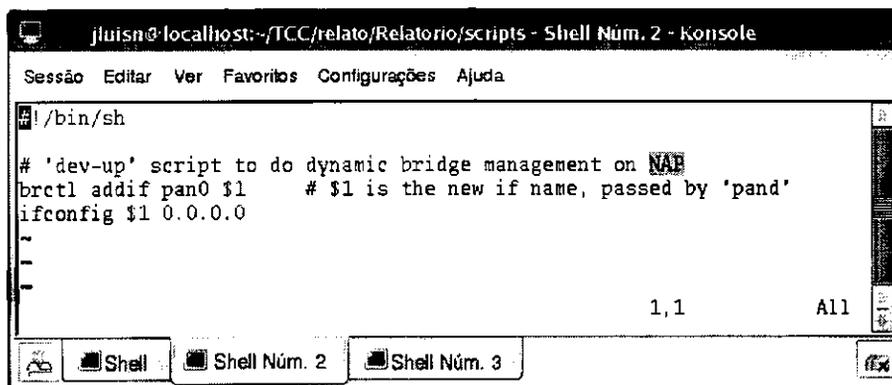
# Adding pan0 to gateway
brctl addif pan0 eth0
```

Figura 20: Script para o estabelecimento da ponte

do PAN sempre que uma interface bnepX for criada, de modo que a função deste arquivo é adicionar à ponte a interface de rede criada.

Com a pilha Bluez corretamente configurada, e o serviço Bluetooth rodando no Linux, a tarefa final trata-se de fazer com que o *daemon* do PAN fique aguardando conexões. Isto é obtido a partir do *script* mostrado na Figura 22. A partir deste momento, os dispositivos Bluetooth podem se conectar ao *gateway* e por conseguinte, a Internet.

Com a ponte configurada, as conexões no *gateway* são realizadas a partir da interface **pan0**. De modo que o endereço IP acessível externamente é o da



```
#!/bin/sh

# 'dev-up' script to do dynamic bridge management on NAB
brctl addif pan0 $1      # $1 is the new if name, passed by 'pand'
ifconfig $1 0.0.0.0
```

Figura 21: Script dev-up

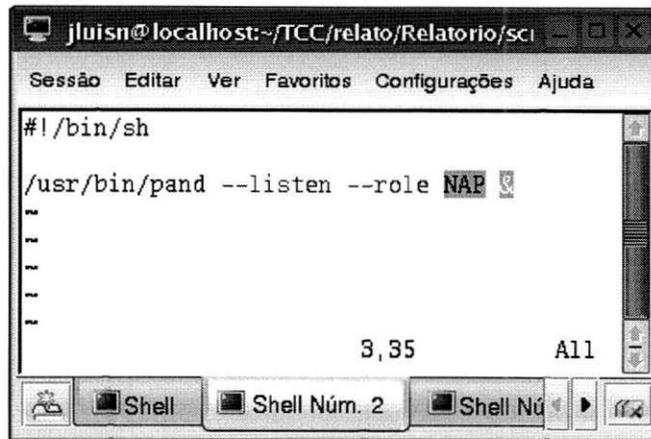


Figura 22: Script para conexões Bluetooth

ponte, tanto para conexões Wi-Fi, Ethernet e Bluetooth.

6.5 Implantação do serviço DHCP

A configuração manual de endereços IP é uma tarefa enfadonha. Nas redes sem fio onde os limites de cada rede não são bem definidos, a determinação dos endereços disponíveis IP torna-se mais difícil, pois é necessário o conhecimento dos endereços IP disponíveis na rede. Além do fato de toda vez que o usuário mudar de rede ele tenha que reconfigurar seu dispositivo com as configurações de rede.

Com a finalidade de resolver este problema, implantou-se no *gateway* Linux o serviço de configuração dinâmica de nós (DHCP). O servidor DHCP envia endereços IP, máscara de sub-rede, o seu endereço IP (*gateway default*) e os IP dos servidores DNS, aos nós que encaminham pedidos em *broadcast*.

No *Fedora Core 3*, como o serviço já se encontra disponível, usa-se apenas um arquivo para configurar o serviço DHCP. Este arquivo, com nome *dhcpd.conf*, é colocado na pasta */etc*. O DHCP é configurado com os parâmetros mostrados na Figura 23.

```
juisn@localhost:~/TCC/relato/Relatorio/scripts - Shell Núm. 2 - Konsole
Sessão Editar Ver Favoritos Configurações Ajuda
/etc/dhcpd.conf
#DHCPCDARGS=pan0;

# Set some defaults for lease time and DNS update method
ddns-update-style ad-hoc;
default-lease-time 600;
max-lease-time 7200;

# Set the subnet mask for the wireless IP network
option subnet-mask 255.255.255.0;

# Set the Broadcast address. This will be 192.168.x.255,
# the "x" will depend upon the network assigned to you.
option broadcast-address 192.168.11.255;

# Set the router address, this will be 192.168.x.1, the address
# of your wireless interface WLAN0
option routers 192.168.11.1;

# Set the Name Server address.
option domain-name-servers 150.165.61.3;

# Set the default domain name for clients on this network.
option domain-name "dee.ufcg.edu.br";

# Allocate a network range for dynamic IP addresses to hand out to clients.
# Again, this range will be in 192.168.x.x, depending upon the network
# allocated to you
subnet 192.168.11.0 netmask 255.255.255.0 {
    range 192.168.11.10 192.168.11.200;
}
~
"dhcpd.conf" 33L, 960C written                               1,1           All
```

Figura 23: Script de configuração DHCP

7 Conclusões

As tecnologias de acesso à redes sem fio são uma tendência global, e ainda não são assuntos esgotados, uma vez que pesquisas apontam para redes com melhores taxas de transmissão e maior alcance, como é o caso da tecnologia WiMax [23]. A integração destas redes possibilita a interconexão de dispositivos com diferentes tecnologias de acesso, possibilitando o aproveitamento das melhores características de cada rede e a integração das mesmas numa arquitetura mista de grande alcance.

A partir da rede proposta, usuários que dispõem de dispositivos com cartões *wireless* podem se conectar à rede sem a necessidade do conhecimento de nenhuma configuração da rede.

Como a arquitetura para a rede foi implementada no sistema operacional Linux, e o Linux vem sendo usado exhaustivamente em dispositivos como pontos de acesso Wi-Fi, PDAs, telefones celulares, entre outros. Pode-se implementar o *gateway* proposto neste trabalho diretamente em um ponto de acesso Wi-Fi, ou em um PDA, por exemplo.

Referências

- [1] The Internet Engineering Task Force. <http://www.ietf.org>. Março de 2005.
- [2] The GNU Operating System. <http://www.gnu.org>. Março de 2005.
- [3] José Duato, Sudhakar Yalamanchili, Lionel Ni. **Interconnection Networks, An Engineering Approach**. IEEE Computer Society Press. USA, 2003.
- [4] Ramjee Prasad, Luis Muñoz. **WLANs e WPANs towards 4G Wireless**. Artech House. Boston, 2003.
- [5] Linux Network Administrators Guide. http://www.faqs.org/docs/linux_network. Março de 2005.
- [6] Internet Assigned Numbers Authority. <http://www.iana.org>. Março de 2005.
- [7] YoLinux Tutorial - Linux Networking. <http://www.yolinux.com>. Março de 2005.
- [8] The Official Bluetooth Website. <http://www.bluetooth.com>. Março de 2005.
- [9] BlueZ - Official Linux Bluetooth protocol stack. <http://www.bluez.org>. Março de 2005.
- [10] Wi-Fi Alliance. <http://www.wi-fi.org>. Março de 2005.
- [11] Bluetooth Special Interest Group. <http://www.bluetooth.com>. Março de 2005.
- [12] Wi-Fi Protected Access. http://www.wi-fi.org/OpenSection/protected_access.asp. Junho de 2005.
- [13] Martin W. Murhammer, Kok-Keong Lee, Payam Motallebi, Paolo Borghi, Karl Wozabal. **IP Network Design Guide**. IBM Red Books, June 1999.
- [14] History of Arpanet. <http://www.dei.isep.ipp.pt/docs/arpa.html>. Junho de 2005.
- [15] NFSNET. <http://www.nfsnet.org>. Março de 2005.
- [16] Unix. <http://www.unix.org>. Junho de 2005.

- [17] Padrão IEEE 802.15.1. <http://www.ieee802.org/15/pub/TG1.html>. Maio de 2005.
- [18] Bluetooth SIG, RFCOMM with TS 07.10. <http://www.bluetooth.com>. Março de 2005.
- [19] Debian - The Universal Operating System. <http://www.debian.org>. Março de 2005.
- [20] Fedora Project. <http://fedora.redhat.com>. Fevereiro de 2005.
- [21] Suse Linux. <http://www.novell.com/pt-br/linux/suse>. Junho de 2005.
- [22] Mandrake Linux. <http://www.mandrakelinux.com>. Junho de 2005.
- [23] WiMax. <http://www.wimax.com>. Junho de 2005.
- [24] Familiar Linux, www.handhelds.org. Abril 2005.
- [25] Opie Linux, www.opie.handhelds.org. Abril 2005.
- [26] Microsoft Hyperterminal. <http://www.microsoft.com>. Abril 2005.