

Alexandre Coelho Gonçalves  
Danilo Bovo de Azevedo  
Leonardo Yoshihara  
Lucas Prado Bessa

Roteador Móvel

Trabalho de Formatura apresentado  
a Escola Politécnica da  
Universidade de São Paulo para  
obtenção do diploma de graduação  
em Engenharia

São Paulo  
2007

Alexandre Coelho Gonçalves  
Danilo Bovo de Azevedo  
Leonardo Yoshihara  
Lucas Prado Bessa

## Roteador Móvel

Trabalho de Formatura apresentado  
a Escola Politécnica da  
Universidade de São Paulo para  
obtenção do diploma de graduação  
em Engenharia

Área de Concentração:  
Engenharia Elétrica ênfase em  
Computação

Orientador: Tereza C. M. B.  
Carvalho  
Co-Orientador: Denis Gabos

São Paulo  
2007

## **Abstract**

This project consisted in the development of a mobile router for Internet access sharing through several interfaces where infrastructure is scarce or completely absent. In order to achieve this goal, a computer connected to the Internet via mobile phone network using GPRS or EDGE was used. This computer shares its connection to the others through wireless or Ethernet connection.

A software for charging for the services provided was developed. The charge takes some factors into consideration: service use duration and amount of data transferred. The amount charged is a result of a combination of these factors.

This router has a wide range due to the use of mobile phone networks as means for Internet connection, given that these networks have a wide coverage throughout the world, and also due to the possibility of the use a car battery as power source.

The router uses the Ubuntu 7.04 as operating system, without graphical interface. MySQL was used for user subscription. The software for charging was written in Java. The interfaces used for subscription and for showing charge details were developed in HTML.

The Linux application "iptables" was used to define the rules that manage the packets exchanged between users and network. DHCP was used for users IP addresses assignments.

## **Resumo**

Esse projeto consistiu no desenvolvimento de um roteador móvel para distribuição de acesso à Internet através de diversas interfaces em locais sem infraestrutura ou infraestrutura deficiente, para isso foi utilizado um computador conectado a Internet através da rede de telefonia celular utilizando o sistema GPRS ou EDGE. Esse distribuirá sua conexão aos demais via conexão Wireless ou rede Ethernet.

Foi desenvolvido também um software para realizar a cobrança do serviço oferecido, essa cobrança será feita levando em conta alguns fatores: Tempo de utilização do serviço e Quantidade de dados trocados com a rede. O valor cobrado será obtido através de uma combinação desses fatores.

Esse roteador tem um alcance ampliado devido ao fato de utilizarmos a rede de telefonia celular como conexão a internet, e essa rede já possuir cobertura em quase todas as localidades do planeta, e também devido à possibilidade de alimentação do sistema por uma bateria de carro.

O roteador utiliza como base o sistema operacional Ubuntu 7.04 sem interface gráfica, o cadastramento dos usuários foi feito em MySQL, o software de cobrança do acesso foi desenvolvido em Java e as interfaces utilizadas para o cadastramento e exibição do saldo devedor foi desenvolvida em HTML.

Utilizamos o aplicativo "iptables" do Linux para definir as regras que gerenciam os pacotes trocados entre os usuários e a rede, e a atribuição dos endereços IP dos usuários pelo DHCP.

## LISTA DE FIGURAS

Figura 1 - Alterações nos cabeçalhos TCP e IP realizadas por NAT .....	22
Figura 2 - Exemplo de requisição .....	24
Figura 3 - Exemplo de resposta HTTP.....	25
Figura 4 - Placa FOX Board LX.....	27
Figura 5 - Placa STK1000 .....	28
Figura 6 - Placa Yiso u893 .....	30
Figura 7 - Celular Samsung d900 .....	31
Figura 8 - Arquitetura do Sistema .....	32
Figura 9 - Gerenciador de dispositivos do Ubuntu indicando as propriedades do celular Samsung .....	38
Figura 10 - Roteador usado como access point .....	40
Figura 11 - Tela inicial do Access Point .....	41
Figura 12 - Tela de Password .....	42
Figura 13 - Tela de definição de endereço .....	43
Figura 14 - Tela de escolha do canal, SSID e modo .....	44
Figura 15 - Desabilitando a Segurança do roteador.....	45
Figura 16 - Script de Inicialização .....	46
Figura 17 - Comando para habilitar NAT .....	47
Figura 18 - Comandos para ajustar rota principal .....	47
Figura 19 - Comandos para liberar acesso a internet para um endereço.....	48
Figura 20 - Fluxo de dados no roteador móvel.....	49
Figura 21 - Alteração para gerar arquivo de log.....	50
Figura 22 - Comando para gerar eventos de log.....	50
Figura 23 - Exemplo de log.....	51
Figura 24 - Conversor DC/AC da Batelco .....	53
Figura 25 - Arquitetura do Software .....	55
Figura 26 - Tela de Cadastro .....	57
Figura 27 - Tela de Cobrança .....	58
Figura 28 - Dados recebidos após execução do "wvdial" .....	68

## Lista de tabelas

Tabela 1 - Comparação entre taxas de transferências de diferentes tecnologias .....	14
Tabela 2 - Protocolos IEEE 802.11 .....	17
Tabela 3 - Mensagem DHCP .....	19
Tabela 4 - Relação de tarifas oferecidas pelas operadoras .....	29
Tabela 5 - Relação dos endereços IP usados pelas interfaces .....	36
Tabela 6 - Atores dos casos de uso .....	61

## LISTA DE ABREVIATURAS E SIGLAS

AC	Alternating Current
BD	Banco de Dados
DC	Direct Current
DHCP	Dynamic Host Configuration Protocol
EDGE	Enhanced Data Rates for GSM Evolution
EVDO	Evolution-Data Optimized
GB	Giga Byte
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
MB	Mega Byte
NAT	Network Address Translation
PCI	Peripheral Component Interconnect
PCMCIA	Personal Computer Memory Card International Association
PPP	Point-to-Point Protocol
RAM	Random Access Memory
RFC	Request For Comments
SSH	Secure Shell
TCP	Transfer Control Protocol
Telnet	Telecommunication Network
URI	Uniform Resource Identifier
USB	Universal Serial Bus
V	Volt
VAC	Volts – Alternating Current
VDC	Volts – Direct Current
W	Watt

## SUMÁRIO

<b>INTRODUÇÃO.....</b>	<b>11</b>
1. Objetivo.....	11
2. Justificativa .....	12
<b>METODOLOGIA.....</b>	<b>13</b>
<b>TECNOLOGIAS UTILIZADAS.....</b>	<b>14</b>
1. GPRS.....	14
2. EDGE.....	16
3. IEEE 802.11 .....	17
4. DHCP.....	18
5. NAT.....	21
6. HTTP Headers.....	24
7. Java .....	26
<b>INFRA ESTRUTURA.....</b>	<b>27</b>
1. Sistema base usado no projeto.....	27
2. Conexão com a rede de telefonia celular .....	29
2.1. Operadoras disponíveis e preços oferecidos.....	29
2.2. Dispositivos para efetuar a conexão .....	30
2.3. Escolha final.....	31
3. Montagem do roteador .....	32
4. Sistema operacional.....	33
5. Configurando as interfaces Ethernet.....	35
5.1. Definindo os endereços IP .....	35
5.2. Configurando o DHCP .....	36
6. Configurando o celular .....	37
6.1. Funcionamento .....	37
6.2. Estabelecendo a conexão.....	38
7. Configuração do access point.....	40
7.1. Dispositivo utilizado.....	40
7.2. Configuração.....	40
7.2.1. <i>Habilitando o serviço de Access Point .....</i>	<i>41</i>
7.2.2. <i>Configurando a conexão Wireless .....</i>	<i>44</i>
7.2.3. <i>Desabilitando as opções de segurança.....</i>	<i>45</i>
8. Configurando o roteamento.....	46
8.1. Comandos utilizados.....	46
8.2. Gerando o arquivo de "log" .....	49
8.3. Arquivos de scripts.....	51
9. Java e MySQL Server .....	52



10.	Bateria .....	53
<b>SOFTWARE.....</b>		<b>54</b>
1.	Arquitetura .....	54
2.	Servidor Web.....	56
3.	Desenvolvimento do Banco de Dados .....	59
4.	Casos de Uso .....	61
4.1.	Atores .....	61
4.2.	Descrição dos Casos de Uso.....	61
<b>TESTES E AVALIAÇÃO .....</b>		<b>65</b>
1.	Infra estrutura .....	65
1.1.	Acesso remoto .....	65
1.2.	Configurando a rede local .....	66
1.3.	Uso do access point.....	67
1.4.	Conexão pelo celular .....	67
1.5.	Teste integrado da infra-estrutura.....	69
2.	Software.....	70
2.1.	Testes iniciais.....	70
2.2.	Servidor Web .....	70
2.3.	Banco de dados .....	71
2.4.	Leitura do arquivo de "log" .....	71
2.5.	Teste integrado .....	71
<b>CONSIDERAÇÕES FINAIS .....</b>		<b>73</b>
<b>REFERENCIAS .....</b>		<b>74</b>

## **Introdução**

### **1. Objetivo**

Este projeto consistiu no desenvolvimento de um roteador móvel para distribuição de conexão através de diversas interfaces, que permitirá que qualquer terminal consiga se conectar a rede. Foi desenvolvido também um sistema de tarifação, que pode ser configurado pelo administrador da rede da forma que melhor o convir, cobrando por dados, banda, tempo ou qualquer combinação dos possíveis métodos.

## 2. Justificativa

Como não existe um equipamento equivalente que reúna todas as funcionalidades e características, o projeto tem um grau de inovação importante e apresenta alguns desafios interessantes a serem enfrentados. Embora aparentemente simples, este projeto apresenta como desafio a implementação de um sistema de tarifação configurável seja por critérios de tempo, de dados, ou ambos, sem que isso gere perdas significativas no desempenho do acesso. Este recurso facilitaria a venda de acesso à Internet de forma portátil e móvel.

Optamos por criar a nossa própria versão de um roteador baseado em Linux para que possamos inserir qualquer funcionalidade desejada, tal como QoS (quality of service), e permitir a adequação do firmware às placas que iremos utilizar no projeto.

## **Metodologia**

No desenvolvimento deste roteador com a utilização de um computador foi dado um maior foco na parte de tarifação do acesso, para isso estudamos os vários métodos de tarifação existentes no mercado, tanto os que cobram por tempo quanto os que cobram por dados recebidos/enviados.

Para desenvolver o processo de tarifação que no momento de venda do acesso à rede deste roteador, o administrador pode escolher o melhor método de cobrança para este cliente, seja ele tempo de acesso, quantidade de dados recebidos/enviados ou a combinação das duas atribuindo um peso a cada uma delas.

Foi realizada também uma avaliação do hardware a ser utilizado devido ao fato de utilizarmos Linux como sistema operacional e alguns hardwares não possuem suporte para o sistema e acarretam algum atraso ao roteador.

## Tecnologias Utilizadas

### 1. GPRS

General Packet Radio Service (GPRS) é o serviço de dados sem fios mais difundido do mundo, disponível em quase todas as redes GSM. GPRS é uma solução de conectividade baseada em IP que serve uma vasta gama de aplicações corporativas e de consumidor final. Com vazões de até 40kbps, os usuários tem uma experiência semelhante ao acesso discado, mas com a facilidade do acesso em qualquer lugar.

A classe do dispositivo determina a velocidade em que GPRS pode ser utilizado. A maioria dos terminais GPRS realiza downloads à taxa de 24kB/s. A velocidade máxima teórica é 171.2kbps, com 8 timeslots alocados para um usuário. Na prática, a velocidade esperada é 40-50kbps.

Type	Uplink (Enviar)	Downlink (Receber)
<b>GPRS</b>	14 kbps	28-64 kbps
<b>GSM CSD</b>	9.6-14 kbps	9.6-14 kbps
<b>HSCSD</b>	28 kbps	28 kbps
<b>Dial-UP</b>	56 kbps	56 kbps
<b>ISDN Standard</b>	64 kbps	64 kbps
<b>ADSL</b>	256 kbps	512 kbps
<b>Broadband</b>	2 Mbps	2 Mbps

Tabela 1 - Comparação entre taxas de transferências de diferentes tecnologias<sup>1</sup>

- Classe A – Telefones móveis podem se conectar a serviços GPRS e GSM simultaneamente.
- Classe B – Telefones móveis podem se conectar a serviços GPRS e GSM, um serviço por vez. É possível realizar/receber chamadas ou enviar/receber SMS durante uma conexão GPRS. Durante ligações ou SMS, os serviços GPRS são suspensos e reiniciados automaticamente após a conclusão da chamada ou SMS.

<sup>1</sup> <http://www.gsmworld.com/technology/gprs/index.shtml>

- Classe C – Telefones móveis podem se conectar a serviços GPRS e GSM, um serviço por vez. A troca entre serviços é manual.

## 2. EDGE

Enhanced Data rates for GSM Evolution<sup>2</sup> (EDGE) é um padrão móvel de transferência de dados em alta-velocidade baseado em rádio. Permite que taxas de transmissão de dados de 384kbps sejam alcançadas quando todos os oito timeslots são utilizados. Isso significa uma taxa máxima de 48kbps por timeslot.

EDGE foi inicialmente desenvolvido para operadores de redes móveis que não conseguiram espectro UMTS. EDGE permite que operadores GSM ofereçam serviços de dados em velocidades que se assemelham aquelas de redes UMTS.

EDGE também permite um caminho de migração evolucionário de GPRS para UMTS, implementando agora as mudanças em modulação que serão necessárias para implementar UMTS no futuro.

---

<sup>2</sup> [http://www.3g-generation.com/gprs\\_and\\_edge.htm](http://www.3g-generation.com/gprs_and_edge.htm)

### 3. IEEE 802.11

Trata-se de um conjunto de padrões para redes locais sem fios (WLAN). No modelo de camadas de redes, essa tecnologia encontra-se na camada de enlace.

Dispositivos da família IEEE 802.11 se comunicam por ondas de rádio, em frequências de operação que variam dependendo do padrão utilizado e do país onde o dispositivo é operado.

Os principais padrões no mercado são descritos na tabela abaixo:

Protocolo	Data	Taxa de Transferência (típica)	Taxa de Transferência (máxima)	Técnica de Modulação
802.11b	1999	4.3 Mbit/s	11 Mbit/s	DSSS
802.11g	2003	19 Mbit/s	54 Mbit/s	OFDM

Tabela 2 - Protocolos IEEE 802.11<sup>3</sup>

Os protocolos 802.11b e 802.11g apresentam em comum as seguintes características:

- Frequência de operação: 2.4 GHz
- Alcance (raio indoor): 38m
- Alcance (raio outdoor): 140m

Alguns dispositivos podem causar interferência na frequência de 2.4 GHz afetando o desempenho de redes baseadas nos protocolos 802.11b e 802.11g, incluindo, mas não limitado a, fornos de microondas, dispositivos Bluetooth, monitores de bebês e telefones sem fios.

<sup>3</sup> <http://standards.ieee.org/getieee802/802.11.html>



## 4. DHCP

Um protocolo padrão desde outubro de 1993, o Dynamic Host Configuration Protocol (DHCP) fornece parâmetros de configuração para hosts Internet. O DHCP consiste de dois componentes: um protocolo para a entrega de parâmetros de configuração específicos de um servidor DHCP para um host e um mecanismo para a alocação de endereços de rede para hosts.

DHCP segue o modelo cliente-servidor, onde um servidor DHCP aloca endereços de rede e entrega parâmetros de configuração para hosts dinamicamente configurados.

O protocolo oferece três mecanismos para a alocação de endereços IP, descritos a seguir:

- Alocação automática: DHCP designa um endereço IP permanente para o cliente.
- Alocação dinâmica: DHCP designa um endereço IP por um período de tempo limitado para o cliente, ou até que o cliente declare o abandono do endereço fornecido.
- Alocação manual: o endereço IP do cliente é definido manualmente pelo administrador de rede, sendo papel do DHCP simplesmente conduzir o endereço designado para o cliente.

O mecanismo de alocação dinâmica é o único dos três mecanismos que permite o reuso automático de um endereço IP que não é mais necessário para o cliente para o qual ele foi designado. Portanto, o mecanismo de alocação dinâmica é particularmente útil para designar endereços IP para clientes que permanecerão conectados à rede temporariamente ou para compartilhar uma quantidade limitada de endereços IP entre um grupo de clientes que não precisam de endereços IP permanentes na rede.

O mecanismo de alocação manual permite que o DHCP seja utilizado para eliminar o processo altamente propenso a erros de manualmente configurar clientes com endereços IP em ambientes onde seja desejável gerenciar as alocações de endereços IP fora dos mecanismos do DHCP.

O formato das mensagens DHCP é baseado no formato das mensagens BOOTP, protocolo que o DHCP sucede. A tabela a seguir descreve os campos de uma mensagem DHCP.

<b>Campo</b>	<b>Octetos</b>	<b>Descrição</b>
Op	1	Tipo de mensagem
Htype	1	Tipo de endereço de hardware
Hlen	1	Tamanho do endereço de hardware
Hops	1	Opcionalmente utilizado quando realiza boot via agente relay
Xid	4	ID da transação, um número aleatório definido pelo cliente e utilizado por cliente e servidor para associar mensagens e respostas entre eles
Sécs	2	Segundos decorridos desde que o cliente iniciou o processo de aquisição ou renovação de endereço
Flags	2	Conjunto de flags (contém somente um flag de broadcast)
ciaddr	4	Endereço IP do cliente
yiaddr	4	"seu" (cliente) endereço IP
siaddr	4	Endereço IP do próximo servidor a ser utilizado
giaddr	4	Endereço IP de agente relay
chaddr	16	Endereço de hardware do cliente
sname	64	Nome do servidor, opcional
File	128	Nome do arquivo de boot
options	Variável	Parâmetros opcionais

**Tabela 3 - Mensagem DHCP**

O DHCP foi projetado para fornecer aos clientes DHCP os parâmetros de configuração definidos nas RFCs de requerimentos de hosts. Após obter os parâmetros via DHCP, um cliente deve ser capaz de trocar pacotes com qualquer outro host na Internet.

O primeiro serviço oferecido pelo DHCP é fornecer armazenamento de parâmetros de rede para os clientes, ou seja, uma tabela no servidor contendo os parâmetros de configuração de todos os clientes, indexados por uma chave única

por cliente. Um cliente pode requisitar seus parâmetros de configuração através de uma interface com o servidor que consiste de mensagens de protocolo e respostas do servidor contendo os parâmetros requisitados.

O segundo serviço oferecido pelo DHCP é a alocação de endereços de rede temporários ou permanentes para os clientes. O mecanismo básico para a alocação de endereços de rede é simples: o cliente requisita o uso de um endereço de rede por algum período de tempo. O mecanismo de alocação garante não realocar aquele endereço dentro do período requisitado e tenta fornecer o mesmo endereço de rede todas as vezes que o cliente requisitar um endereço. O cliente pode enviar uma mensagem liberando o endereço requisitado. O servidor reusa os endereços cujo período expirou ou o cliente liberou.

## 5. NAT

A necessidade por tradução de endereços IP surge quando endereços IP de uma rede interna não podem ser utilizados em uma rede externa, seja por motivos de privacidade ou porque eles são inválidos para uso em uma rede externa.

A topologia de rede fora de um domínio local pode mudar de várias formas. Clientes podem mudar de provedores, backbones de empresas podem ser reorganizados, ou provedores podem se fundir ou se separar. Sempre que a topologia externa muda com o tempo, a alocação de endereços para nós dentro da rede local também deve mudar para refletir as alterações externas. Mudanças deste tipo podem ser ocultadas dos usuários dentro do domínio através da centralização das mudanças em um roteador de tradução de endereço simples.

A tradução básica de endereços permitiria que hosts em uma rede privada tivessem acesso transparentemente à rede externa e ative acesso para hosts locais selecionados por fora. Organizações com uma configuração de rede predominantemente para uso interno, com uma necessidade para acesso externo ocasional são bons candidatos para esse esquema.

Muitos usuários Small Office, Home Office (SOHO) e empregados que trabalham em casa possuem muitos nós de redes em seus escritórios, rodando aplicações TCP/UDP, mas têm um único endereço IP alocado para seus roteadores de acesso remoto por seus provedores de serviço para acessar redes remotas. Essa comunidade sempre crescente de usuários de acesso remoto seria beneficiada por NAT, que permitiria que múltiplos nós em uma rede local pudessem acessar redes remotas simultaneamente usando um único endereço IP alocado para seus roteadores.

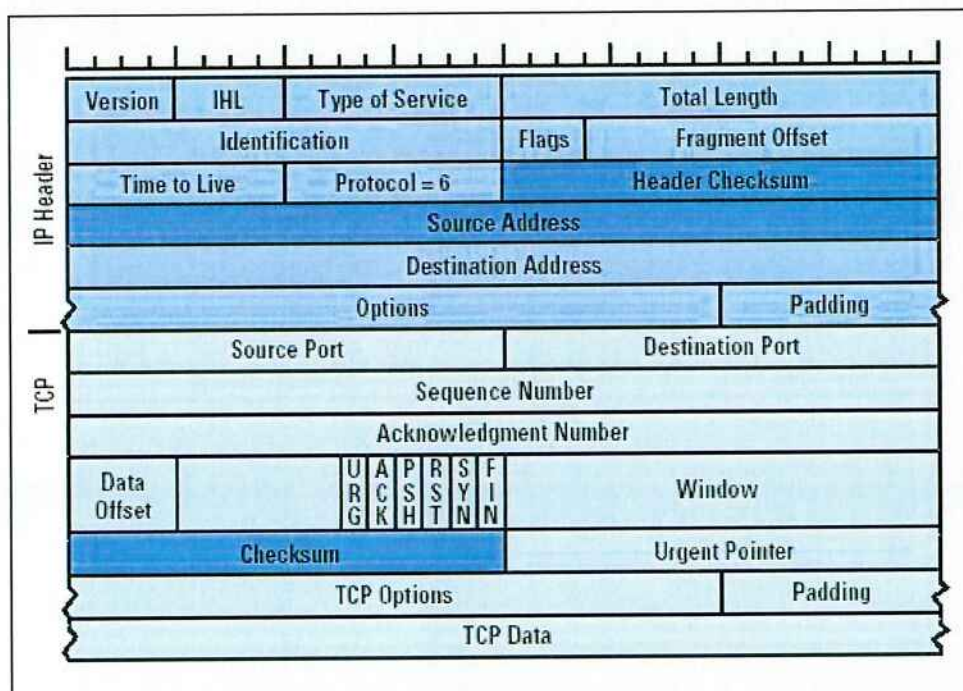


Figura 1 - Alterações nos cabeçalhos TCP e IP realizadas por NAT<sup>4</sup>

Network Address Translation (NAT) permite que hosts dentro de uma rede privada acessem hosts em redes externas transparentemente. No NAT, as sessões são unidirecionais, saindo das redes privadas. Sessões no sentido oposto podem ser permitidas em caráter excepcional, usando mapas de endereços estáticos para hosts pré-determinados.

Um domínio stub com um conjunto de endereços pode ser ativado para comunicar-se com a rede externa por mapeamento dinâmico do conjunto de endereços privados para o conjunto de endereços de rede válidos globalmente. Se o número de nós locais for menor ou igual ao de endereços no conjunto global, cada nó global tem a garantia de um endereço global para ser mapeado. Caso contrário, nós com permissão para acessar simultaneamente a redes externas são limitados pelo número de endereços no conjunto global. Endereços locais individuais podem ser estaticamente mapeados para endereços globais específicos para garantir acesso ao exterior ou para permitir acesso ao host local através de hosts externos

<sup>4</sup> [http://www.cisco.com/web/about/ac123/ac147/archived\\_issues/ipj\\_7-3/anatomy.html](http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_7-3/anatomy.html)

via um endereço de rede público. Múltiplas sessões simultâneas podem ser inicializadas pelo nó local, usando o mesmo mapeamento de endereços.

Endereços dentro de um domínio stub são locais para esse domínio e não são válidos fora do domínio. Portanto, endereços dentro de um domínio stub podem ser reutilizados por outros domínios stub.



## 6. HTTP Headers

Hypertext Transfer Protocol<sup>5</sup> (HTTP) é um protocolo da camada de aplicação para sistemas de informação hipermídia distribuídos e colaborativos. HTTP está em uso na World-Wide Web desde 1990. Em suas primeiras versões, tratava-se simplesmente de um protocolo de transferência de dados através da Internet. Hoje, HTTP contém um conjunto de métodos e cabeçalhos que indicam o propósito do pedido e carregam informações sobre os dados sendo transportados.

HTTP é um protocolo de requisição/resposta. Um cliente envia uma requisição para um servidor, na forma de um método, URI, versão do protocolo, seguidos por uma mensagem contendo informações do cliente e características do pedido. O servidor responde com uma mensagem contendo uma linha de status, incluindo a versão do protocolo e um código de sucesso ou erro, seguida por uma mensagem contendo informações sobre o servidor e características dos dados transmitidos na resposta.

A seguir, um exemplo de requisição HTTP é detalhado:

```
GET /index.html HTTP/1.0
User-Agent: Mozilla/3.0Gold (WinNT;I)
Host: www.exemplo.com
Accept: image/gif, image/x-bitmap, image/jpeg, */*
```

**Figura 2 - Exemplo de requisição**

A primeira linha da mensagem indica o método (GET), a URI (/index.html) e a versão do protocolo (HTTP/1.0). A segunda linha apresenta informações sobre o cliente. A terceira linha indica o servidor requisitado. A quarta linha indica informações sobre os formatos de dados que o cliente espera como resposta do servidor.

---

<sup>5</sup> <http://www.w3.org/Protocols/rfc2616/rfc2616.html>

A seguir, um exemplo de resposta HTTP é detalhado:

```
HTTP/1.0 200 OK
Server: NSCA/1.4.2
MIME-Version: 1.0
Content-type: text/html
Content-length: 46

<html>
<title>
  A pagina HTML simples
</title>
</html>
```

Figura 3 - Exemplo de resposta HTTP

A primeira linha indica a versão do protocolo (HTTP/1.0), o código da mensagem de resposta (200) e um texto com o status da resposta (OK). A segunda linha fornece informações sobre o servidor. As três linhas seguintes fornecem informações sobre o corpo da mensagem. As seis linhas seguintes são o corpo da mensagem, os dados propriamente transferidos do servidor para o cliente.



## 7. Java

Java é uma linguagem de programação orientada a objetos desenvolvida pela Sun Microsystems<sup>6</sup>. Sua estrutura é semelhante a C e C++, facilitando seu aprendizado por programadores familiares com essas linguagens.

Trata-se de uma linguagem multithread e independente de plataforma, devido ao uso da máquina virtual Java. Um programa escrito em Java é compilado para bytecode. Um programa em bytecode não contém código específico de sistema operacional ou processador, portanto qualquer plataforma que tenha máquina virtual Java é capaz de executar o programa.

---

<sup>6</sup> <http://tools.ietf.org/html/rfc2131>

## Infra Estrutura

### 1. Sistema base usado no projeto

A idéia inicial para o projeto era simplificar ao máximo o hardware que seria usado, de modo a reduzir o consumo de energia consumida pela bateria e assim aumentar o tempo de funcionamento. Pensando nisso, começamos a procurar informações sobre kits de desenvolvimento que utilizassem microcontrolador com o sistema operacional Linux embarcado.

Alguns modelos de kit de desenvolvimento foram encontrados, como o STK1000 da Atmel<sup>7</sup> e o FOX Board LX da Acme Systems<sup>8</sup>. Porém, o alto custo de aquisição desses kits, que facilmente ultrapassam o valor R\$ 300,00, e a dificuldade de utilizá-los fizeram com que esta opção fosse abandonada pelo grupo.



**Figura 4 - Placa FOX Board LX**

<sup>7</sup> ATMEL CORPORATION. ATSTK1000. Disponível em: <<http://www.atmel.com/>>. Acesso em: 02 abr. 2007.

<sup>8</sup> ACME SYSTEMS. FOX board a complete Linux System. Disponível em: <<http://www.acmesystems.it/>>. Acesso em: 02 abr. 2007.



**Figura 5 - Placa STK1000**

A solução encontrada foi utilizar um PC comum, que além de ser fácil de obter tem a vantagem de podermos alterar o número e tipos de interfaces que podem ser usadas. Isso evita também que uma grande carga de retrabalho do ponto de vista de configuração do hardware seja gerada toda vez que uma alteração for necessária, o que provavelmente ocorreria se fosse utilizado um dos kits de desenvolvimento citados acima.

O primeiro computador usado foi um notebook Compaq Presario 1200. Ele possuía um processador AMD K6 II com 96 MB de memória RAM, uma interface USB e uma interface Ethernet através de uma placa PCMCIA. Nos primeiros testes realizados, o notebook estava funcionando bem. No entanto, alguns problemas durante a inicialização da máquina, e que não tinham relação à implementação do projeto, começaram a atrapalhar o andamento das atividades. Como solução emergencial, decidimos montar o equipamento em um computador AMD Athlon com 128 MB de memória RAM.

Ao usar este computador, contamos no final com 3 interfaces Ethernet e uma interface USB para serem usadas. Ressaltamos que não há necessidade neste projeto do uso dos 128 MB de memória RAM disponível no computador, essa quantidade foi usada pois o grupo não possuía um pente de memória com menos de 128 MB. O uso desse computador, também gerou um problema de consumo de energia pela bateria, mas como este projeto é uma prova de conceito, esse fator não foi levado em conta durante a implementação.

## 2. Conexão com a rede de telefonia celular

Uma das principais funções do nosso roteador é estabelecer uma conexão com a Internet usando a rede de telefonia celular. Para isso, precisávamos saber quais eram os preços cobrados pelo serviço pelas operadoras, quais dispositivos poderíamos usar para fazer a conexão e como configurar o dispositivo para trabalhar no nosso roteador.

### 2.1. Operadoras disponíveis e preços oferecidos

Foram consultadas as 3 principais operadoras que oferecem esse tipo de serviço: Vivo<sup>9</sup>, Claro<sup>10</sup> e Tim<sup>11</sup>. A Vivo trabalha com a tecnologia EVDO enquanto que a TIM e a Claro trabalham principalmente com GPRS e EDGE. As informações sobre a cobrança dos serviços foram retiradas dos sites oficiais das empresas.

Tarifa - Vivo - Sistema EVDO	
Pacote Avulso	
Preço por MB	R\$ 5,90
Plano de Franquia de consumo	
100 MB	R\$ 49,90
MB Excedente	R\$ 0,75

Tarifa - Claro - Sistema GPRS	
Pacote Avulso	
Preço por MB	R\$ 6,00
Plano de Franquia de consumo	
40 MB	valor não disponível
MB Excedente	valor não disponível

Tarifa - TIM - Sistema GPRS	
Pacote Avulso - Preço por MB	
até 1	R\$ 5,99
entre 1 e 5	R\$ 5,79
entre 5 e 40	R\$ 4,99
acima de 40	R\$ 3,99
Plano de Franquia de consumo	
40 MB	R\$ 79,00
MB Excedente	valor não disponível

Tabela 4 - Relação de tarifas oferecidas pelas operadoras

<sup>9</sup> VIVO. Preços e Tarifas Vivo Zap Empresas. Disponível em: <<http://www.vivo.com.br/>>. Acesso em: 30 mai. 2007.

<sup>10</sup> CLARO. Claro – Acesso à Internet. Disponível em: <<http://www.claro.com.br/>>. Acesso em: 30 mai. 2007.

<sup>11</sup> TIM. Quanto custa. Disponível em: <<http://www.tim.com.br/>>. Acesso em: 30 mai. 2007.

Como neste projeto não temos a intenção de transferir muitos dados, apenas o necessário para realizar testes e demonstrações, não houve a necessidade de adquirir um plano de franquia de consumo, bastando optar pelo pacote avulso. Como os preços praticados pelas operadoras são bem próximos, o que acabou definindo a escolha foi o preço do dispositivo a ser usado no projeto.

## 2.2. Dispositivos para efetuar a conexão

Desde o início, a intenção era usar um dispositivo que possuísse uma interface USB. Isso foi decidido para que se houvesse necessidade de mudar o hardware base do sistema, não haver a preocupação de adaptar o dispositivo ao novo hardware.

A primeira opção que surgiu durante a fase de levantamento de preços foi o uso da placa Yiso u893<sup>12</sup>, oferecido pela Vivo pelo valor de R\$ 399,00 (preço informado em loja). Essa placa possuía interface USB, e atendia as necessidades do projeto.



Figura 6 - Placa Yiso u893

---

<sup>12</sup> **Modem Yiso Modelo U893:** Manual de Instruções. São Paulo [2006]. 56 p

Durante as pesquisas, descobriu-se que também era possível usar o celular Samsung d900<sup>13</sup>, que pertence a um dos integrantes do grupo. Ele também atende as necessidades do projeto e com ele poderíamos nos conectar pela operadora Tim.



Figura 7 - Celular Samsung d900

### 2.3. Escolha final

Com o objetivo de poupar gastos com o projeto, acabamos escolhendo o celular da Samsung. Desse modo, o nosso roteador se conectará à Tim através da tecnologia GPRS ou EDGE para estabelecer a conexão com a Internet.

Os detalhes de como o celular foi configurado para ser usado se encontram no item 6.

---

<sup>13</sup> SAMSUNG ELETRONICS. **SGH-D900**: User's Guide. Coréia do Sul [2006]. 135 p. (GH68-12771A)

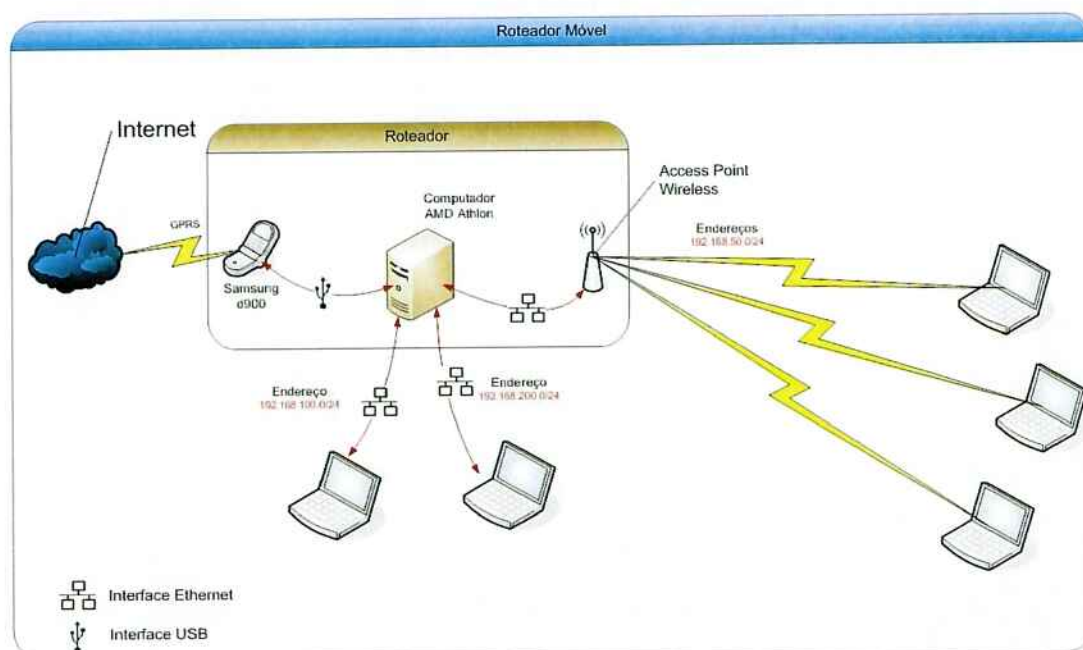


### 3. Montagem do roteador

Como já dito anteriormente, o roteador foi montado em um computador AMD Athlon com 128 MB de memória RAM. A placa-mãe do computador já possui uma interface USB e uma interface Ethernet. As outras duas interfaces Ethernet foram obtidas com o uso de 2 placas de rede PCI.

O sistema operacional, os arquivos de configuração e os programas que serão executados ficarão armazenados em um disco rígido de 4 GB. Esse disco rígido foi usado pois pertencia a um dos integrantes do grupo e não estava em uso.

A interface USB foi usada para conectar o celular Samsung, as duas placas de rede serão usadas por quem precisar usar os recursos do roteador por cabo de rede convencional e a interface Ethernet disponível na placa mãe será usada para conectar os dispositivos que farão uso do roteador através do "access point". Segue abaixo uma ilustração que mostra como o roteador foi montado.



**Figura 8 – Arquitetura do Sistema**

#### 4. Sistema operacional

Como já mencionado, o sistema operacional escolhido para desenvolver o projeto foi o Linux. Ele foi escolhido por ser um sistema aberto que permite ao desenvolvedor realizar alterações que julgar apropriadas para que o sistema funcione do modo desejado.

No entanto, por se tratar de um sistema aberto, existem diversas distribuições e versões, o que muitas vezes faz com que algumas versões não sejam bem documentadas ou tenham pouco ou quase nenhum suporte para solucionar alguns problemas específicos.

Por essa razão, escolhemos usar a distribuição Ubuntu 7.04<sup>14</sup> do Linux. Ela foi escolhida, por ser uma distribuição bastante popular entre as pessoas que usam o Linux, além de possuir um bom suporte tanto da organização que a desenvolveu tanto de diversas comunidades que desenvolvem projetos para essa distribuição. Além também de ser de fácil uso, principalmente para instalar recursos como compiladores, editores de texto e atualização de programas.

Durante o processo de instalação, selecionamos apenas os componentes necessários para que o Linux funcionasse de forma adequada e possuísse as configurações de acesso remoto e de rede (configuração das interfaces de rede, roteamento, criação de regras, etc). Evidentemente, a interface gráfica que acompanha a distribuição do Ubuntu não foi instalada, sendo toda a configuração feita por linhas de comando.

Atualmente o Ubuntu se encontra na versão 7.10, lançada em outubro de 2007, mas como a implementação deste projeto se iniciou no mês de agosto, quando a última versão era a 7.04, decidimos não atualizar a versão da distribuição para não

---

<sup>14</sup> CANONICAL LTD. Ubuntu Home Page. Disponível em: <<http://www.ubuntu.com/>>. Acesso em: 24 jun. 2007.



correr o risco de o processo de atualização alterar algumas das configurações que já haviam sido feitas no início do desenvolvimento.

## 5. Configurando as interfaces Ethernet

Cada interface Ethernet do roteador estará em uma sub-rede diferente. Isso foi definido pensando na segurança do sistema depois que um usuário tiver seu acesso liberado. As explicações sobre como essa segurança foi pensada se encontram no item 8.

Para cada interface de rede que o Linux detectou durante o seu processo de instalação, ele atribuiu um nome. As interfaces das placas de rede receberam o nome de eth0 e eth1 e a interface da placa-mãe recebeu o nome de eth2. A inicial eth se refere ao fato das interfaces serem Ethernet.

### 5.1. Definindo os endereços IP

O usuário que se conectar pela interface eth0 receberá o endereço IP 192.168.100.0/24, quem se conectar na eth1 receberá o endereço IP 192.168.200.0/24 e quem se conectar na eth2 receberá o endereço IP 192.168.50.0/24. A interface eth2 será usada apenas para aqueles que se conectarem pelo o "access point".

O arquivo onde se define o endereço de uma interface Ethernet é o '/etc/network/interfaces'. No arquivo que esta no roteador tem as duas configurações para cada interface, ela tanto pode assumir um endereço fixo ou obter um endereço via DHCP. Por padrão, todas as interfaces Ethernet possuem endereço fixo, o uso de atribuição por DHCP só foi usado somente para testes. A configuração que não é usada foi deixada como comentário no arquivo.

## 5.2. Configurando o DHCP

Neste projeto, para oferecer o serviço de DHCP foi usado o DHCP3 Server<sup>15</sup> para atribuir os endereços para os dispositivos que se conectarem ao nosso roteador. Ele foi escolhido por ser de fácil instalação e configuração.

O arquivo de configuração do serviço de DHCP é `/etc/dhcp3/dhcpd.conf`. Ele foi ajustado para atribuir endereços em cada interface Ethernet de acordo com os endereços de rede mencionados no item anterior. Basicamente, se indica quais sub-redes que terão atribuição de endereços, intervalo de endereços que serão disponíveis, o endereço de gateway (no caso, é o endereço da interface) e endereços de servidores DNS.

O arquivo `"/etc/default/dhcp3-server"` define em quais interfaces o servidor de DHCP irá atuar, neste caso as interfaces `eth0`, `eth1` e `eth2`.

A tabela a seguir mostra a relação das interfaces com todos os endereços utilizados.

Interface	Endereço de rede	Endereço IP da interface	Intervalo de endereços disponíveis para DHCP
eth0	192.168.100.0/24	192.168.100.1	192.168.100.11 a 192.168.100.50
eth1	192.168.200.0/24	192.168.200.1	192.168.200.11 a 192.168.200.50
eth2	192.168.50.0/24	192.168.50.1	192.168.50.11 a 192.168.50.50

**Tabela 5 - Relação dos endereços IP usados pelas interfaces**

<sup>15</sup> INTERNET SYSTEMS CONSORTIUM. Dynamic Host Configuration Protocol. Disponível em: <http://www.isc.org/>. Acesso em: 15 jul. 2007.

## 6. Configurando o celular

### 6.1. Funcionamento

Após algumas pesquisas sobre o funcionamento do celular Samsung d900, foi averiguado que o mesmo funciona como um modem convencional ao ser conectado pela interface USB. No entanto, precisamos saber como o Linux trabalhava com o celular e onde ele ficaria mapeado para uso.

Isso foi resolvendo conectando o celular a outro computador que possuía a mesma versão do Ubuntu rodando com interface gráfica. O Ubuntu com interface gráfica possui um gerenciador de dispositivos que reúne as principais propriedades de todos os componentes de hardware que foram detectados.

No caso do celular Samsung, com o uso do gerenciador, descobrimos que o celular fica mapeado na pasta `/dev/ttyACM0`. De posse dessa informação, foi possível realizar as configurações necessárias para estabelecer uma conexão com a operadora de telefonia celular.

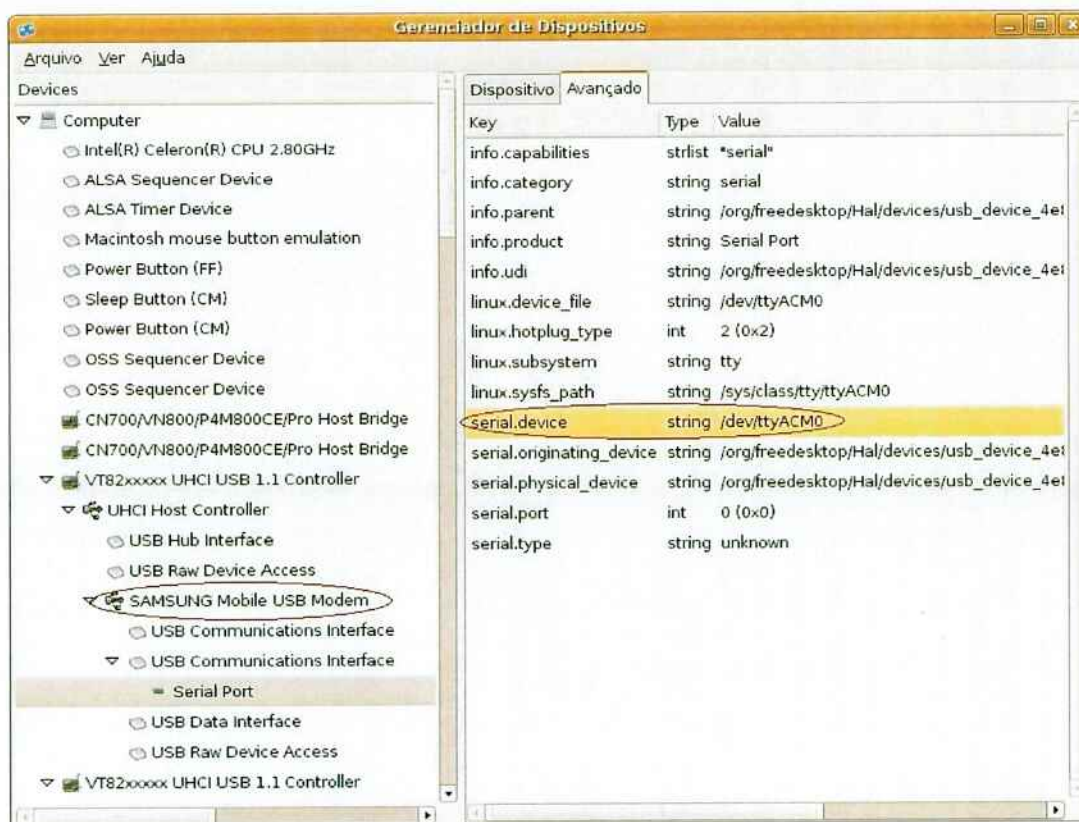


Figura 9 - Gerenciador de dispositivos do Ubuntu indicando as propriedades do celular Samsung

## 6.2. Estabelecendo a conexão

Para estabelecer a conexão com a operadora, utilizamos o aplicativo “wvdial”. Ele é um discador que utiliza o protocolo ponto-a-ponto (PPP) para se conectar a Internet. Ele também possui um arquivo de configuração que se encontra em /etc/wvdial.conf e contem as informações necessárias para realizar a conexão. É neste arquivo que foi preciso inserir a informação de onde o celular Samsung estava mapeado no Linux.

Esse arquivo precisa ter o número a ser discado, o nome do usuário e uma senha. Todas essas informações são padrão da operadora e fornecida pela mesma. Uma vez que o aplicativo esteja corretamente configurado, para iniciar o processo basta executar o comando “wvdial &”. Ao estabelecer a conexão, o Linux atribui o

nome ppp0 para a interface do celular. Com isso, já podemos montar as regras de roteamento do roteador móvel.

## 7. Configuração do access point

### 7.1. Dispositivo utilizado

Neste projeto, para fornecer a conexão wireless ao Roteador Móvel foi utilizado o dispositivo Belkin Wireless G Router, modelo F5D7230-4<sup>16</sup>. Apesar de ser um roteador, as funcionalidades de roteamento foram desativadas e conforme descrição abaixo, apenas o serviço de “access point” foi habilitado.



**Figura 10 - Roteador usado como access point**

### 7.2. Configuração

O “access point” ficará na interface de rede que possui endereço de rede 192.168.50.0/24 e suas configurações podem ser alteradas por uma interface Web pelo endereço 192.168.50.250 utilizando um navegador Web.

---

<sup>16</sup> BELKIN. Wireless G Router: User Manual. Estados Unidos [2007]. 110 p. (P74559-B)



O dispositivo não permite acesso às configurações por terminal como Telnet ou SSH. Por essa razão, apenas as configurações de canal foram feitas pela interface Web, deixando a segurança e liberação de acesso à Internet sob responsabilidade do roteador móvel.

### 7.2.1. Habilitando o serviço de Access Point

Ao acessar o endereço 192.168.50.250, será exibida a página Web a seguir onde a opção "Use as Access Point" deve ser selecionada.

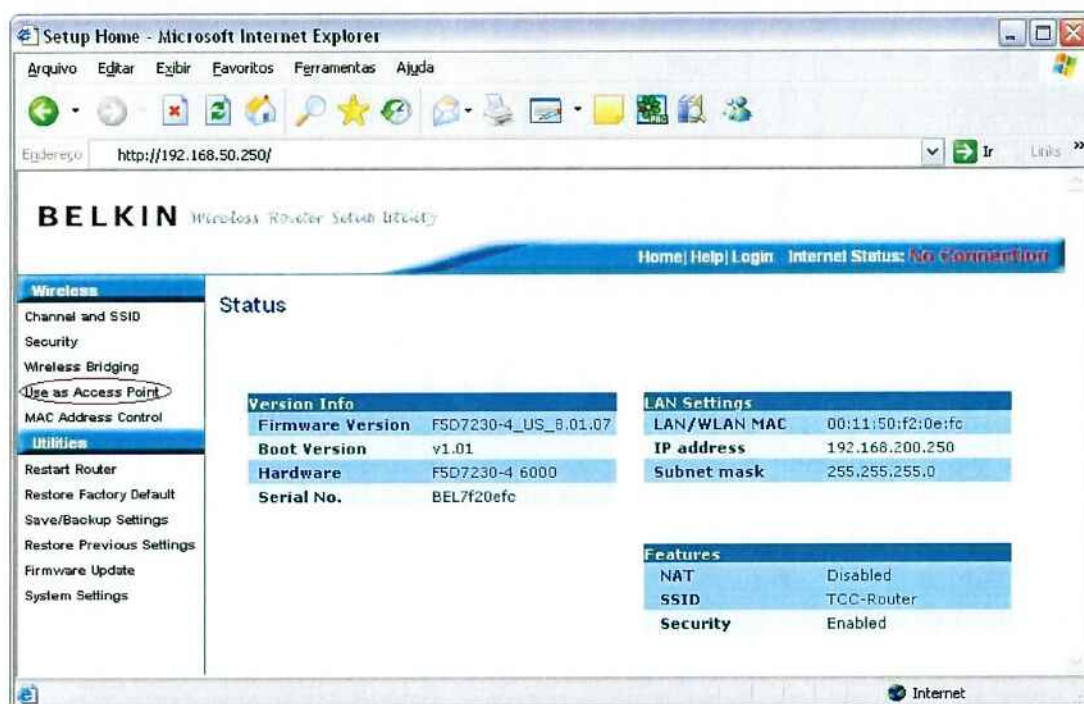


Figura 11 - Tela inicial do Access Point



Em seguida, deve ser fornecida a senha de segurança para dar continuidade.

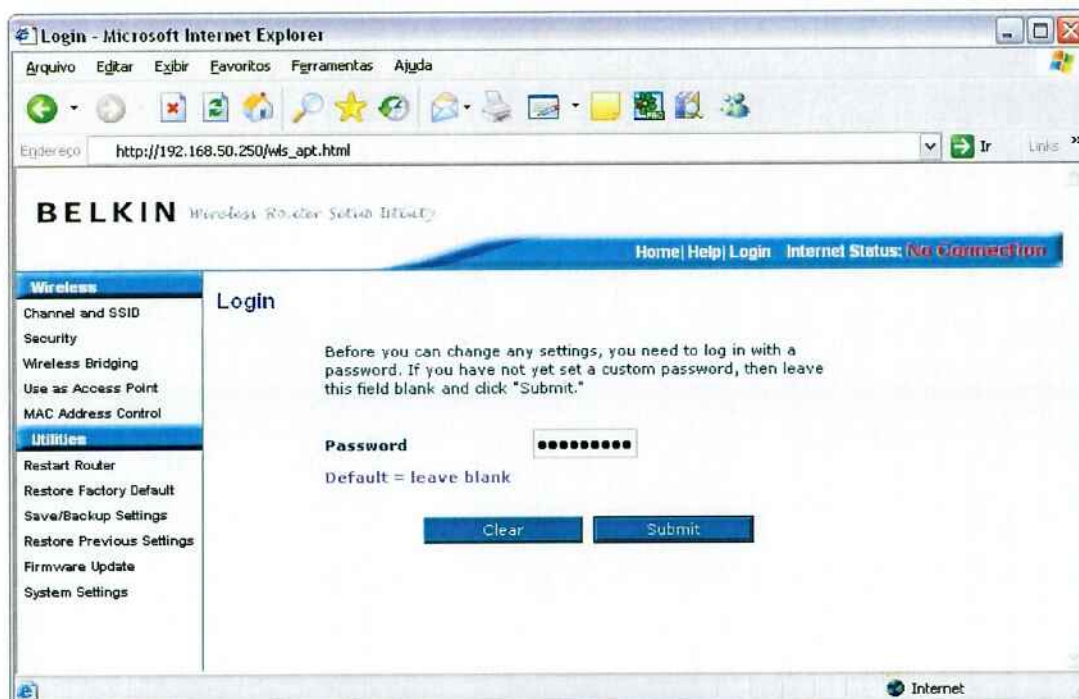


Figura 12 - Tela de Password

Para finalizar deve-se definir um endereço para que as configurações possam ser acessadas posteriormente em caso de necessidade. Depois que a operação for confirmada, deve-se aguardar o reinicialização do dispositivo. Após isso, o dispositivo passará a funcionar apenas como um “access point”.

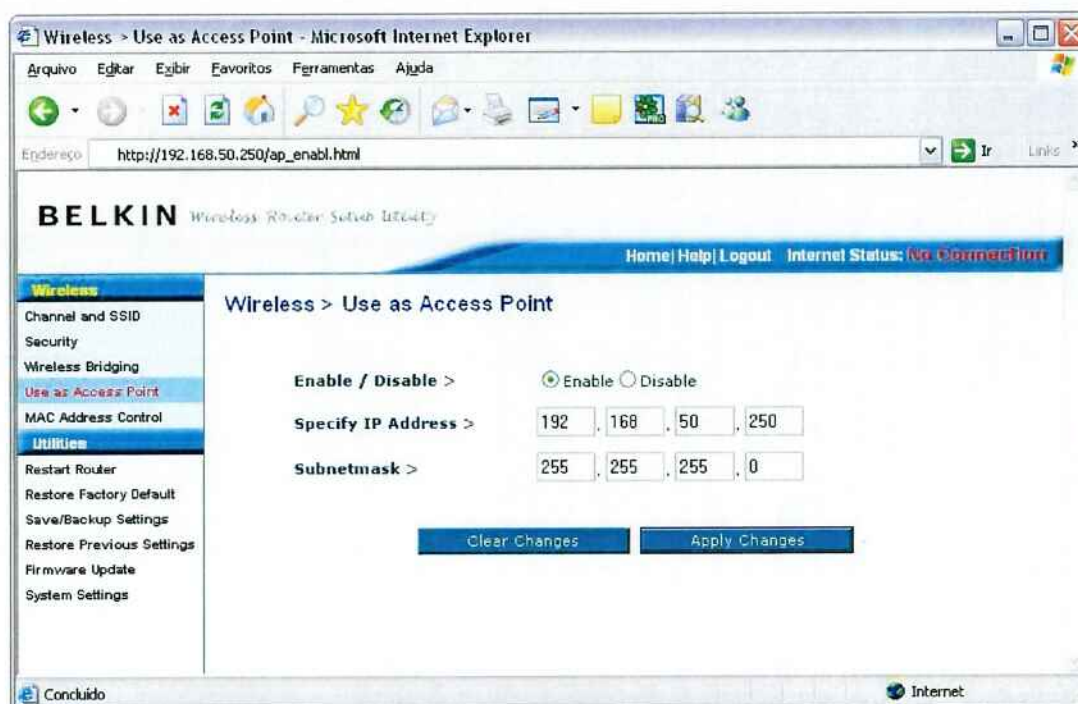


Figura 13 - Tela de definição de endereço

### 7.2.2. Configurando a conexão Wireless

Para definir o nome da conexão wireless foi utilizada a opção “Channel and SSID” do “access point”.

Após inserir novamente a senha de segurança, foi definido que o roteador móvel usará o canal 1 e terá o nome “TCC Router”. Após a confirmação da operação, o “access point” será novamente reinicializado.

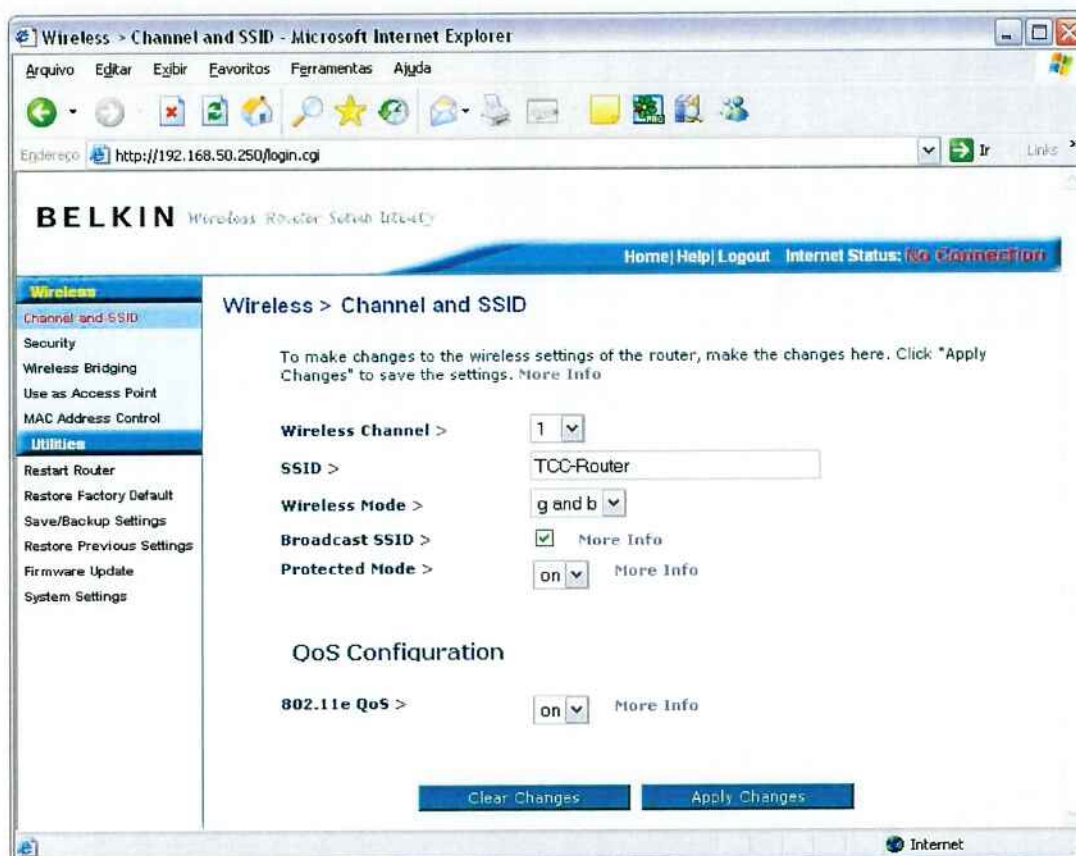


Figura 14 - Tela de escolha do canal, SSID e modo

### 7.2.3. Desabilitando as opções de segurança

Como o roteador móvel será responsável pela segurança e permissão de uso de suas funcionalidades. As opções de segurança que poderiam ser usadas pelo dispositivo de "access point" foram desabilitadas. Essa operação foi feita através da opção "Security".

Após inserir novamente a senha, deve-se desabilitar as opções de segurança através da opção "Security Mode". Após a confirmação da operação, o "access point" será novamente reinicializado.

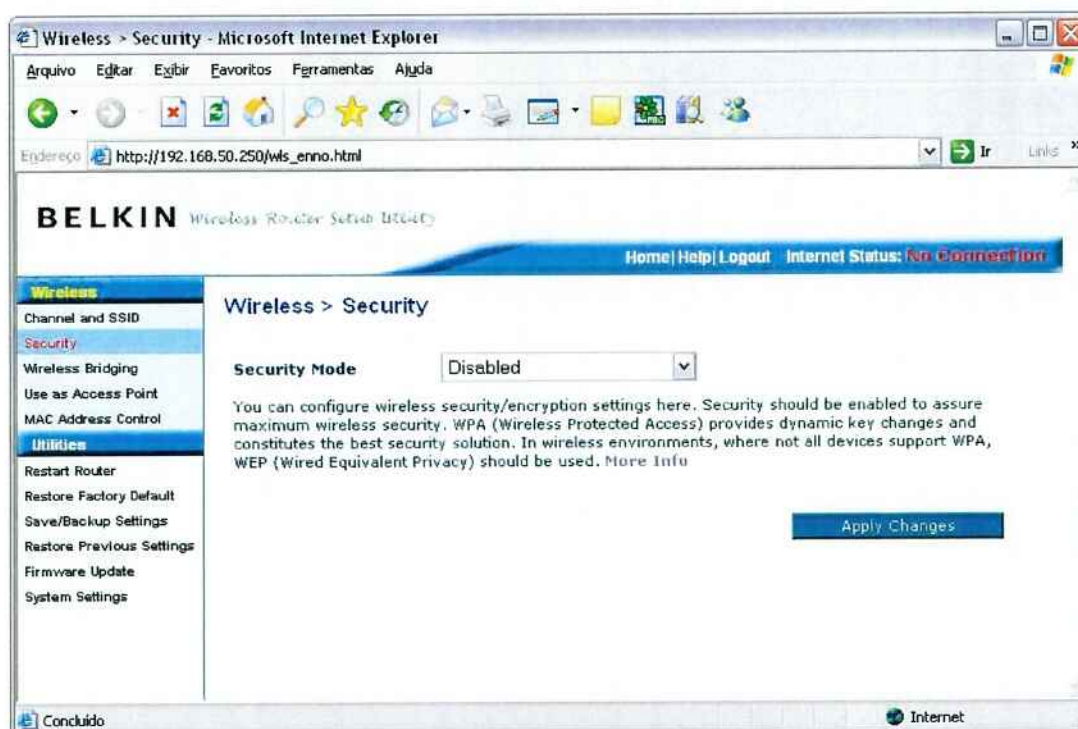


Figura 15 - Desabilitando a Segurança do roteador

Após todas as alterações mencionadas acima, o "access point" já pode ser colocado em funcionamento no roteador móvel.



## 8. Configurando o roteamento

Durante a fase de levantamento dos requisitos do projeto, cogitou-se a possibilidade do grupo implementar o protocolo de roteamento que seria utilizado. Porém, dadas as dificuldades em fazê-lo e o prazo estipulado para desenvolver todo o projeto, o grupo optou por utilizar as ferramentas de roteamento já existentes no sistema operacional Linux e que acompanham a distribuição do Ubuntu.

Após consulta em alguns guias, concluímos que o uso das ferramentas “iptables” e “ip” ofereciam os recursos necessários para aplicar as regras que queríamos.

### 8.1. Comandos utilizados

Para o correto funcionamento do roteador, após ele ser inicializado, deve-se primeiro estabelecer a conexão com a Internet através do celular para que tenhamos a interface ppp0 criada pelo sistema operacional. Com isso feito, devem-se executar os seguintes comandos:

```
#iptables -F
#iptables -Z
#iptables -t nat -F
#iptables -P INPUT ACCEPT
#iptables -A INPUT -i ppp0 -j DROP
#iptables -F FORWARD DROP
#iptables -F OUTPUT ACCEPT
```

Figura 16 - Script de Inicialização

Esses comandos realizam respectivamente as seguintes operações:

- Apagar todas as regras para entrada, saída e encaminhamento de pacotes que estivessem habilitadas.
- Reiniciar os contadores de pacotes e bytes.
- Apagar todas as regras de NAT que estejam habilitadas.
- Permite que qualquer endereço possa acessar o roteador.
- Impede que endereços externos se conectem ao roteador pela interface ppp0, desse modo, apenas os usuários do roteador podem acessá-lo.
- Impede que os pacotes possam ser encaminhados a outros endereços. A liberação de encaminhamento de um endereço interno para um externo deve ser feita individualmente.
- Permite que o roteador acesse endereços externos.

Como ao estabelecer a conexão como Internet, recebemos apenas um endereço IP da operadora, precisamos habilitar o serviço de NAT para mascarar os endereços de rede local do roteador para que os usuários possam acessar endereços externos. Isso foi feito executando o comando:

```
#iptables -t nat -A POSTROUTING -s 192.168.0.0/16 -o eth2 -j MASQUERADE
```

**Figura 17 - Comando para habilitar NAT**

Com esse comando, todos os endereços de rede 192.168.0.0/16, que incluem todos os endereços das interfaces eth0, eth1 e eth2, serão mascarados através do NAT.

Para definir que a rota principal para acessar endereços externos passaria pela interface ppp0, foi utilizado os seguintes comandos:

```
#ip route del default dev eth0  
#ip route del default dev eth1  
#ip route del default dev eth2  
#ip route add default dev ppp0
```

**Figura 18 - Comandos para ajustar rota principal**

Os 3 primeiros comandos impedem que as interfaces eth0, eth1, e eth2 sejam usadas erroneamente como rota para endereços externos e o último comando garante que apenas a interface ppp0 será usada para esse fim.

Finalmente, para liberar um endereço interno para que ele tenha acesso aos endereços externos da Internet precisamos dos seguintes comandos:

```
#iptables -A FORWARD -p ALL -s endereço -o ppp0 -j ACCEPT  
#iptables -A FORWARD -p ALL -i ppp0 -d endereço -j ACCEPT
```

**Figura 19 - Comandos para liberar acesso a internet para um endereço**

Nos comandos acima, a palavra endereço deve ser substituída por um número de endereço IP qualquer que tenha sido atribuído para um usuário. Esses comandos garantem também que o fluxo de dados ficará restrito entre um usuário e a interface ppp0, sem que um usuário possa interferir em outro. Quando se desejar bloquear o acesso do usuário, basta substituir o parâmetro -A por -D nos comandos acima.

O fluxo de dados que atuará no roteador está demonstrado na figura abaixo.

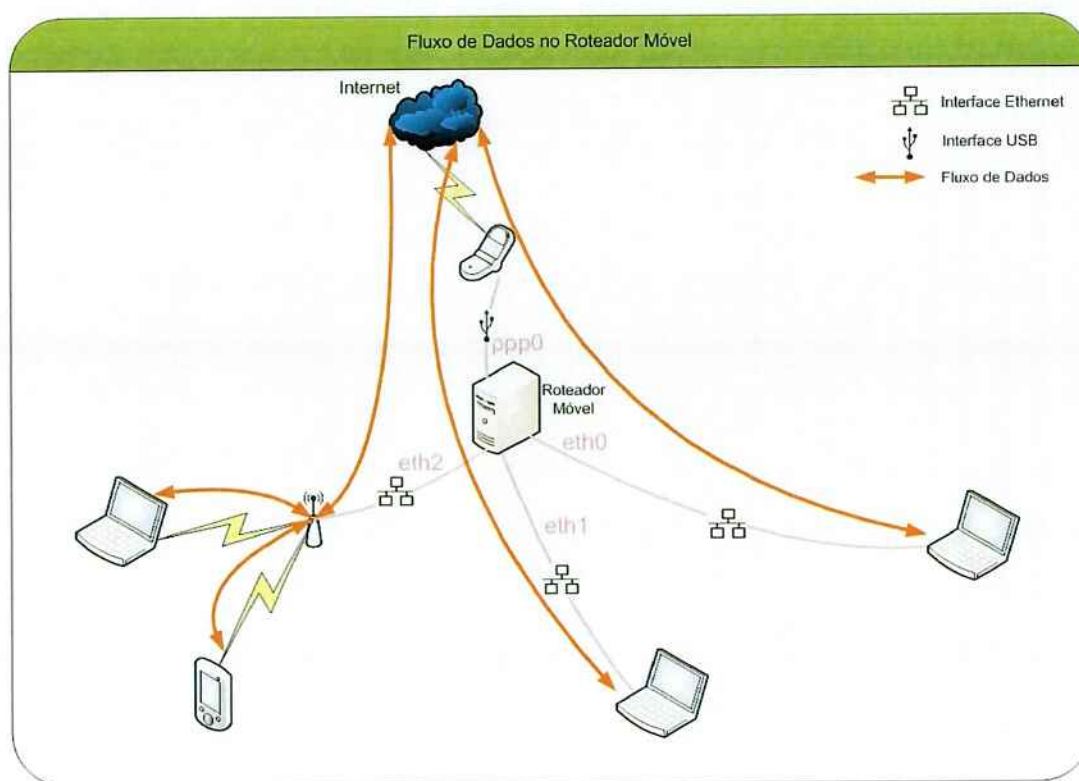


Figura 20 - Fluxo de dados no roteador móvel

## 8.2. Gerando o arquivo de "log"

Para que o software responsável pelo gerenciamento dos usuários e da tarifação funcione corretamente, era preciso que um arquivo de "log" fosse criado para registrar a quantidade de pacotes que estão trafegando pelo roteador, bem como qual usuário foi responsável pelo pacote e a quantidade de dados contida nele.

Isso foi resolvido usando uma das funcionalidades do "iptables" que permite a geração dessas informações no arquivo de "log" geral do sistema operacional. Ao analisar o arquivo `/etc/syslog.conf`, que responsável por configurar a geração de



“log” do sistema, percebemos que nenhum “log” para eventos do tipo alert é criado e assim adicionamos a seguinte linha ao arquivo:

```
*.=alert -/var/log/iptables
```

Figura 21 - Alteração para gerar arquivo de log

Com isso, todos os eventos do tipo alert serão gravados no arquivo /var/log/iptables e mesmo que outros processos do sistema operacional gerem evento desse tipo, constatamos que eles são muitos raros e não irão interferir no desempenho da análise do arquivo. Para que o “iptables” gere eventos do tipo alert, usamos o comando:

```
#iptables -I FORWARD -j LOG --log-level 1 --log-prefix 'TCC >'
```

Figura 22 - Comando para gerar eventos de log

Esse comando inclui nas regras de encaminhamento de pacotes a regra para que sejam gerados eventos toda vez um pacote for encaminhado pelo roteador tanto de um endereço interno para um externo ou vice-versa. O parâmetro 'log-level 1' indica que o evento será do tipo alert e o parâmetro 'log-prefix' incluirá o prefixo 'TCC >' a cada novo evento gerado, dessa forma poderemos distinguir os eventos gerados pelo “iptables” de qualquer outro que seja eventualmente gerado pelo sistema.

Segue abaixo, um exemplo do “log” gerado pelo sistema é mostrado abaixo, os dados que são usados pelo software foram destacados em negrito.

```
Sep 22 17:51:23 bessa kernel: [ 6277.922027] TCC >IN=eth2 OUT=eth0
SRC=192.168.200.48 DST=64.233.169.103 LEN=40 TOS=0x00 PREC=0x00 TTL=63
ID=2699 DF PROTO=TCP SPT=1129 DPT=443 WINDOW=64261 RES=0x00 ACK
FIN URGP=0
Sep 22 17:51:23 bessa kernel: [ 6278.060077] TCC >IN=eth0 OUT=eth2
SRC=64.233.169.103 DST=192.168.200.48 LEN=40 TOS=0x00 PREC=0x00 TTL=245
ID=20817 PROTO=TCP SPT=80 DPT=1133 WINDOW=8190 RES=0x00 ACK FIN
URGP=0
Sep 22 17:51:23 bessa kernel: [ 6278.060090] TCC >IN=eth0 OUT=eth2
```

Figura 23 - Exemplo de log

### 8.3. Arquivos de scripts

Para facilitar o processo de inicialização do roteador foram criados 2 scripts que executam alguns dos comandos citados acima. O script *inicia\_rotador* executa todos os comandos acima, com exceção dos dois comandos que liberam o acesso de cada endereço individual.

O script *desativa\_rotador* foi criado para ser usado apenas em casos de emergência, como uma possível falha do software do gerenciamento, e deve ser executado somente por linha de comando. Ele desativa as funcionalidades de “log” e encaminhamento de pacotes para que se possa fazer uma análise de alguma anormalidade que possa ter surgido.

## 9. Java e MySQL Server

Seguindo a proposta de criar um software de tarifação flexível, que permita o administrador definir os parâmetros de tempo e de quantidade de dados trafegados, geramos um arquivo de “log” de todos os pacotes que passarem pelo roteador.

Como esperado, esse arquivo cresce rapidamente o que dificulta a administração das informações. Como solução a este problema, adotamos um banco de dados simples para otimizar a organização e as buscas. O sistema gerenciador de banco de dados escolhido foi o MySQL por não necessitar de muito processamento e ser suficiente para as nossas necessidades.

O software de tarifação consiste de um programa Java que busca as informações de destino do pacote, tamanho, e hora que o pacote foi recebido no arquivo de “log” e as transmite a base de dados.

Para instalar tanto o Java quanto o MySQL, foi usado o utilitário apt-get que acompanha o Ubuntu. Além de instalar ambos os programas, esse utilitário os configura automaticamente para que funcionem corretamente no sistema operacional. A versão do Java instalado é a 1.4 e a versão do MySQL é a 5.0.

## 10. Bateria

Para prover a mobilidade do roteador móvel, além de garantir que ele se conecte por uma operadora de telefonia celular, temos que garantir que ele funcione sem depender da rede elétrica local, ou seja, usando uma bateria.

Uma das alternativas propostas foi utilizar de baterias que alimentariam um circuito que teria como saída as tensões necessárias ( $\pm 12V$ ,  $\pm 5V$ ,  $\pm 3.3V$ ) para alimentar os componentes do roteador. Outra opção foi de utilizar um conversor de tensão contínua para tensão alternada 110VAC e alimentar a fonte padrão do computador.

Escolhemos usar uma bateria de carro ligado a um conversor de tensão. Desse modo, garantimos que a fonte de alimentação do computador irá fornecer corretamente as tensões necessárias para os componentes do roteador. A opção usar o circuito foi abandonada, pois os componentes do roteador, principalmente a placa-mãe do computador, são feitos de circuitos delicados que poderiam sofrer danos graves caso o circuito não fornecesse as tensão de maneira constante e com poucas variações.

Um possível conversor que pode ser utilizado é o conversor DC/AC 12VDC – 120VAC potência de 500W da Batelco<sup>17</sup>. Ele possui as características necessárias para manter o roteador em funcionamento e possui mecanismos de alerta para o caso de tensão baixa da bateria, sobre carga na saída e temperatura interna muito alta. O preço de aquisição é por volta de R\$ 350,00.



**Figura 24 - Conversor DC/AC da Batelco**

<sup>17</sup> MARINE SERVICE. Inversor Batelco DC/AC 12VDC – 120VAC 500W. Disponível em: <<http://www.marine.com.br/>>. Acesso em: 22 set. 2007.

## Software

### 1. Arquitetura

A arquitetura do sistema ficou definida em 3 camadas: a mais baixa é o banco de dados e o software de roteamento; uma camada a cima fica o software de interface e a camada superior é o servidor web.

O software de interface é responsável por articular todo o trânsito de dados entre as demais camadas, dessa forma ele age como interface entre o servidor web e o banco de dados, e como interface entre o software de roteamento e o banco de dados.

O software de roteamento gera um arquivo de “log” que contém o cabeçalho de todos os pacotes que transitaram por ele. O software de interface gera uma cópia desse arquivo e limpa o original, liberando-o para o roteador, e lê o arquivo cópia e separa apenas as informações úteis à tarifação – Origem, Destino, Data e Tamanho dos pacotes trafegados. De posse dessas informações a interface já conectada ao banco de dados, insere na tabela correspondente todos os pacotes que estavam presentes no arquivo.

Em paralelo a esse processo, a interface interage com o servidor web, de onde ele recebe solicitação para incluir um novo cliente na base, requisições de quanto cada cliente consumiu e autenticação de novos usuários.

O servidor web é o “front-end” do roteador e será por este que o administrador poderá fazer modificações nas configurações pré-estabelecidas e os usuários poderão verificar quanto já consumiram do serviço previamente pago.





Figura 25 - Arquitetura do Software

## 2. Servidor Web

O software de tarifação possui interface web, portanto é necessária a presença de um servidor web para apresentar o sistema para o usuário final e para o administrador do sistema.

Como o servidor web não necessita de muitas funcionalidades, optou-se por desenvolver um simples ao invés de utilizar um de mercado como o Apache, por exemplo.

Foi desenvolvido um software servidor web em Java, utilizando as facilidades que a linguagem disponibiliza para operações de rede.

O servidor web desenvolvido tem como função receber as requisições HTTP e tratá-las. Ele identifica os pedidos HTTP enviados pelos browsers, reconhecendo o método, GET ou POST, e o arquivo requisitado.

Quando um pedido usa o método GET, o servidor simplesmente busca o arquivo solicitado, lê em um stream, adiciona o cabeçalho HTTP e envia para o usuário que solicitou. Se o arquivo não for encontrado, o servidor envia uma mensagem "404 – Not Found", formatada em HTML e com cabeçalho HTTP.

Quando um pedido usa o método POST, o servidor identifica qual arquivo o pedido HTTP requisita e trata o pedido, utilizando códigos Java. Os pedidos esperados são para inserção e autenticação de usuários. Nestes casos, o servidor utiliza a interface com o banco de dados, para acrescentar uma entrada na tabela de clientes (cadastro de usuário), verificar senha (autenticação) e resgatar horário de início de acesso (tarifação).

O cadastro de usuários é simplesmente acrescentar uma entrada ao banco de dados com os dados do usuário.

No caso de autenticação, o servidor resgata a senha armazenada no banco de dados para o usuário requisitante, compara com a senha digitada pelo usuário e, se a verificação de senha for bem sucedida, libera o acesso para o usuário, armazenando no banco de dados o IP e o horário de início do acesso e criando uma regra no iptables.

No caso de tarifação, o servidor resgata do banco de dados as informações do usuário, incluindo o horário de início do acesso, e exibe as informações na tela, incluindo o tempo de acesso, a quantidade de dados transferidos e a cobrança, baseado na política de cobrança (participação do tempo de acesso e da quantidade de dados transferidos).



The screenshot shows a Mozilla Firefox browser window with the title "Cadastro - Mozilla Firefox". The address bar displays "http://192.168.50.1/inicial.html". The main content area features a registration form titled "Cadastro de usuário". The form includes five input fields labeled "Nome:", "Sobrenome:", "Celular:", "CPF/Login:", and "Senha:". Below these fields is a blue "Cadastrar" button.

Nome:	<input type="text"/>
Sobrenome:	<input type="text"/>
Celular:	<input type="text"/>
CPF/Login:	<input type="text"/>
Senha:	<input type="password"/>
<input type="button" value="Cadastrar"/>	

Figura 26 - Tela de Cadastro





## Conclusão de acesso - Cobrança

Nome: Denis  
Sobrenome: Gabos  
Celular: 555-1234

Horário de início: 14:46:27  
Horário de encerramento: 17:13:05  
Tarifa por minuto: R\$0,10  
**Preço por tempo: R\$14,70**

Dados transferidos: 17,42 MB  
Tarifa por MB: R\$0,05  
**Preço por tráfego: R\$0,87**

**Preço final: R\$15,57**

Figura 27 - Tela de Cobrança

### 3. Desenvolvimento do Banco de Dados

A adoção de um banco de dados foi necessária para programar o sistema de tarifação. Para tarifar trabalhamos com dois parâmetros: tempo que o usuário permanece conectado e quantidade de dados trafegados, obtidos da leitura do cabeçalho IP de todos os pacotes que passam pelo roteador móvel.

As informações referentes a cada pacote é salva em um arquivo de “log”, que como o esperado, cresce rapidamente. Para organizar, otimizar e viabilizar a tarifação, esse arquivo é organizado em uma tabela do banco de dados. Essa tabela tem as informações do endereço IP de origem, endereço IP de destino, data / hora e tamanho do pacote, ou seja, apenas as informações que são estritamente necessárias.

Há outra tabela no banco, que contem os usuários conectados do roteador. As informações são: Nome, documento, data/ hora de inicio de uso, celular para contato, IP designado ao usuário e o plano escolhido. O relacionamento é feito através do endereço IP de origem para totalizar a quantidade de “Upload” e pelo endereço IP de destino para totalizar a quantidade de “download”. O tempo de uso é calculado pela diferença de tempo entre o último pacote destinado ao usuário registrado e o tempo que está registrado como inicio do uso.

Outra tabela presente no banco de dados é a tabela planos, nesta tabela contem o preço por KB, o preço por tempo e a identificação do plano. A relação entre a tabela com os dados do cliente e a tabela plano é feita pela identidade do plano que será guardada na tabela cliente.

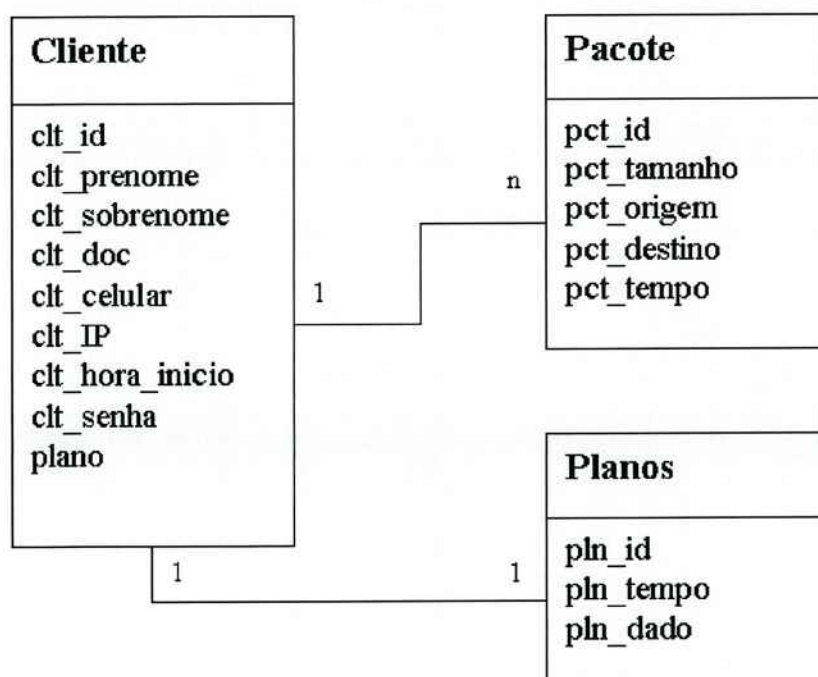


Figura 28 - Diagrama de entidade e relacionamento do Banco de Dados

## 4. Casos de Uso

### 4.1. Atores

Nome	Papel
Clientes	Usuários do roteador móvel, acesso mediante a cadastro no sistema.
Administradores	Possuem acesso irrestrito ao sistema

Tabela 6 - Atores dos casos de uso

### 4.2. Descrição dos Casos de Uso

#### Caso de uso 1: Registro de pacote

**Descrição:** Este caso de uso descreve o processo pelo qual os pacotes IP são registrados no banco de dados.

**Evento iniciador:** Processo cíclico, ocorre a cada 2 minutos

**Atores:** não se aplica

**Pré-condição:** arquivo de “log” salvo no roteador.

**Seqüência de eventos:**

1. Reinicia o contador de tempo para iniciar a próxima chamada de registro.
2. Software abre o arquivo de “log”.
3. Programa lê cada linha do arquivo, separando pelos marcadores as informações de data, origem, destino e tamanho do pacote (Insere pacote).
4. De posse dessas informações o programa registra na tabela pacote do banco de dados o pacote.

**Pós-condição:** Programa volta ao estado de espera.

**Extensões:** não há.

**Inclusões:**

1. Caso de Uso Insere pacote.

**Caso de uso 2:** Insere pacote.

**Descrição:** Insere pacote IP no BD.

**Evento iniciador:** Chamada do método pelo sistema.

**Atores:**

**Pré-condição:** pacote IP instanciado e conexão aberta com o BD.

**Seqüência de eventos:**

1. Recebe o pacote com as informações de data, origem, destino e tamanho.
2. Faz requisição de inclusão ao banco.
3. Recebe confirmação de sucesso.

**Exceções:** Ao fazer a requisição o banco retorna erro. Neste caso o pacote é descartado.

**Pós-condição:** Sistema liberado para próxima inclusão

**Extensões:** não há.

**Inclusões:** não há

**Caso de uso 3:** Abre conexão com o BD

**Descrição:** Este caso de uso trata da rotina para abrir uma conexão com o BD.

**Evento iniciador:** Início do programa.

**Atores:** administrador

**Pré-condição:** não há

**Seqüência de eventos:**

1. O sistema faz uma requisição de conexão ao gerenciador do BD.
2. O gerenciador de banco de dados retorna uma mensagem informando que a conexão foi realizada com sucesso.
3. O Sistema armazena o caminho até o BD.

**Exceções:** Em caso de falha na conexão o software retorna ao início da rotina, para tentar novamente a conexão.

**Pós-condição:** Sistema encerra a rotina, devolvendo o controle para o método principal.

**Extensões:** não há.

**Inclusões:** não há

**Caso de uso 4:** Fechar conexão com o banco de dados

**Descrição:** Encerra a conexão com o banco de dados.

**Evento iniciador:** o Administrador faz uma requisição para encerrar o roteador.

**Atores:** Administrador.

**Pré-condição:** Não há.

**Seqüência de eventos:**

1. O sistema recebe a requisição para encerrar.
2. O sistema pára todas as "threads" relacionadas a acesso ao banco de dados.
3. O sistema envia um pedido de desconexão ao gerenciador de banco de dados.
4. O gerenciador envia uma confirmação de desconexão.
5. O sistema encerra a sua atividade.

**Exceções:** não há

**Pós-condição:** Sistema encerrado

**Extensões:** não há.

**Inclusões:** não há.

**Caso de uso 5:** Busca por um cliente.

**Descrição:** Este caso de uso busca um cliente na base de dados que armazena o histórico de clientes do Administrador do roteador.

**Evento iniciador:** Administrador faz uma requisição ao sistema de um cliente.

**Atores:** Administrador.

**Pré-condição:** Sistema estar disponível e existir uma conexão com o banco de dados.

**Seqüência de eventos:**

1. Sistema recebe uma informação sobre o cliente (Nome completo, documento ou IP)
2. Sistema identifica qual a informação que o administrador forneceu.
3. Sistema faz um requerimento ao gerenciador de banco de dados passando a informação sobre o cliente.
4. Sistema retorna todas as informações sobre o primeiro cliente que ele encontrar que respeita a condição informada.

**Exceções:** não há

**Pós-condição:** Sistema apresenta as informações do cliente.

**Extensões:** não há.

**Inclusões:** não há

## **Testes e avaliação**

### **1. Infra estrutura**

Depois da escolha de qual seria o sistema base usado no projeto, seguimos instalando o sistema operacional Linux. Com a instalação concluída, seguimos uma série de passos para verificar o funcionamento dos componentes e validar a infra-estrutura usada e em seguida validar o sistema por inteiro.

As atividades realizadas para montar a infra-estrutura e testar o software foram as seguintes:

- Providenciar acesso remoto;
- Configurar a rede local;
- Habilitar o access point;
- Configurar a conexão por celular;
- Teste da infra-estrutura;

#### **1.1. Acesso remoto**

Apesar do computador usado no projeto possuir suporte para monitor de vídeo e teclado, o roteador não poderá ter esse recursos, assim se fez necessário o uso de acesso remoto ao roteador.

No início, configuramos uma das interfaces de rede Ethernet para receber endereço IP por DHCP e conectamos essa interface a uma rede local para que fosse possível ter um acesso remoto ao roteador e iniciar as atividades seguintes.



O acesso em geral é feito conectando-se a “port” 22 do roteador usando o protocolo SSH, mas alternativamente a acesso pode ser feito pela “port” 23 usando o protocolo “Telnet”.

## 1.2. Configurando a rede local

Com o acesso remoto estabelecido, o passo seguinte foi configurar as interfaces Ethernet, conforme descrito na parte de Infra Estrutura no item 5. Com essa atividade concluída, instalamos e configuramos o serviço de DHCP no roteador.

A partir desse momento, não precisaríamos mais conectar o roteador em uma rede local para ter acesso remoto à máquina, bastando apenas conectar outro computador ou um notebook através de um cabo de rede, pois o próprio roteador se tornou capaz de atribuir endereços IP.

Ainda mantivemos uma das interfaces Ethernet para receber endereço por DHCP para que tivéssemos um acesso a Internet enquanto o acesso pelo celular Samsung não estivesse operacional. Isso foi necessário principalmente para que pudéssemos testar as regras de roteamento sem que ficar na dependência de precisar do celular toda vez que um teste precisasse ser feito.

Para finalizar a configuração da rede local, montamos algumas regras de roteamento e de NAT para que cada usuário possa apenas buscar por endereços externos sem interferir com outros usuários.

### 1.3. Uso do access point

Com as interfaces Ethernet configuradas, podemos passar para a configuração do "access point". O dispositivo usado para oferecer o sinal wireless foi configurado conforme o item 7 da parte de Infra Estrutura.

Essa etapa foi relativamente fácil, pois apenas precisamos conectá-lo a uma das interfaces Ethernet e ligá-lo. Testamos a conectividade com um notebook com suporte a rede wireless. O teste foi bem sucedido e o notebook recebeu um endereço IP de nosso roteador dentro da faixa de endereço estabelecida pelo serviço de DHCP.

### 1.4. Conexão pelo celular

Configuramos os procedimentos para usar o celular Samsung conforme o item 6 da parte de Infra Estrutura. Nas primeiras tentativas de estabelecer a conexão, realizadas em agosto, não conseguíamos ter um acesso estável. Perdíamos a conexão após aproximadamente 5 segundos de operação.

Pensávamos que se tratava de algum problema ao configurar o aplicativo responsável por fazer a discagem, mas alguns dias depois, depois de contatar a operadora Tim, descobrimos que na verdade havia um problema na disponibilidade do serviço por parte da operadora. Depois que o serviço foi normalizado, conseguimos realizar a conexão sem problemas. Segue na figura abaixo os dados recebidos da operadora após a execução do "wvdial".

```

root@bessa:~# wvdial &
[1] 5622
root@bessa:~# WvDial<*1>: WvDial: Internet dialer version 1.56
WvModem<*1>: Cannot get information for serial port.
WvDial<*1>: Initializing modem.
WvDial<*1>: Sending: ATZ
WvDial Modem<*1>: ATZ
WvDial Modem<*1>: OK
WvDial<*1>: Sending: AT&FH0L3
WvDial Modem<*1>: AT&FH0L3
WvDial Modem<*1>: OK
WvDial<*1>: Modem initialized.
WvDial<*1>: Sending: ATDT*99**1#
WvDial<*1>: Waiting for carrier.
WvDial Modem<*1>: ATDT*99**1#
WvDial Modem<*1>: CONNECT
WvDial<*1>: Carrier detected. Starting PPP immediately.
WvDial<Notice>: Starting pppd at Thu Nov 22 18:06:51 2007
WvDial<Notice>: Pid of pppd: 5623
WvDial<*1>: pppd: H00000[06][08]§[06][08]
WvDial<*1>: Using interface ppp0
WvDial<*1>: pppd: H00000[06][08]§[06][08]
WvDial<*1>: pppd: H00000[06][08]§[06][08]
WvDial<*1>: pppd: H00000[06][08]§[06][08]
WvDial<*1>: pppd: H00000[06][08]§[06][08]
root@bessa:~# WvDial<*1>: pppd: H00000[06][08]§[06][08]
WvDial<*1>: local IP address 189.65.114.145
WvDial<*1>: pppd: H00000[06][08]§[06][08]
WvDial<*1>: remote IP address 192.200.1.21
WvDial<*1>: pppd: H00000[06][08]§[06][08]
WvDial<*1>: primary DNS address 189.40.238.6
WvDial<*1>: pppd: H00000[06][08]§[06][08]
WvDial<*1>: secondary DNS address 189.40.238.7
WvDial<*1>: pppd: H00000[06][08]§[06][08]
root@bessa:~#

```

Figura 29 - Dados recebidos após execução do "wvdial"

Para validá-la, efetuamos o comando "ping" para o site principal da USP e tivemos retorno, confirmando que a conexão estava operante.

Com essa atividade bem-sucedida, passamos a não precisar daquela interface Ethernet que havia sido configurada anteriormente para funcionar com DHCP. Fazendo com que ela volte ter a configuração de endereço fixo, podemos realizar um teste com toda a infra-estrutura montada.

### 1.5. Teste integrado da infra-estrutura

O teste da infra-estrutura relativamente simples. Como nesta etapa ainda não tínhamos o software operando, os comandos para realizar conexões e liberar usuários foram feitos manualmente através de linhas de comando e scripts já mencionados em itens anteriores.

Em primeiro lugar, estabelecemos a conexão com a operadora de telefonia celular. Após valida-la conectamos um notebook a uma interface de rede e conectamos outro através de sinal wireless. Ambos receberam os endereços IP pelo DHCP referentes à interface em que estavam conectados.

Feito isso, realizamos a liberação de acesso primeiramente para o usuário que estava usando o notebook por wireless inserindo as regras apropriadas através do *"iptables"*. Para validar o acesso, ele acessou sites diversos da internet e navegou sem problemas. Ao mesmo tempo, o usuário que estava conectado pelo cabo de rede não conseguia de forma alguma acessar endereços externos, mostrando que o roteamento impedia tal ação a menos que as regras de liberação fossem aplicadas a ele também.

Após isso, este último usuário teve seu acesso liberado e assim conseguiu acessar endereços externos. Por final, testamos se um usuário podia de alguma forma enviar pacotes para outro. O teste foi feito tentando executar o comando *"ping"* de um usuário para outro. Como esperado, o *"ping"* não foi bem sucedido, pois devido as regras implementadas, cada usuário que envia pacotes tendo o roteador como intermediário só poderá mandá-los para endereços IP externos e somente quando ele estiver liberado para isso.

Com a infra-estrutura validada, demos continuidade com as atividades envolvendo o software de gerenciamento e tarifação do roteador.

## 2. Software

### 2.1. Testes iniciais

A primeira bateria de testes aplicada foi em cada funcionalidade que foi desenvolvida. As funcionalidades testadas foram: Servidor Web, acesso ao banco de dados, leitura do arquivo de log.

### 2.2. Servidor Web

O primeiro teste consistiu em verificar se as requisições de páginas estavam funcionando corretamente. Para testar, criaram-se páginas HTML simples que estavam salvas em um diretório do roteador. Por meio de um browser, o usuário fazia uma requisição do servidor, este por sua vez devolvia a página. O sucesso desse teste foi visualizar a página solicitada.

O segundo teste aplicado foi trabalhar com páginas que tivessem formulários, estes serão utilizados em telas de “login” e cadastro de cliente. O teste realizado seguiu os seguintes passos, o usuário preenchia o formulário e clicava no botão de “enviar”. As informações deveriam ser recebidas pelo servidor web que deveria tratá-las e devolver uma página informando o sucesso ou a falha no recebimento das informações, e todos os dados que foram enviados pelo formulário. O sucesso desse teste foi observado pelo recebimento das informações enviadas.

O último teste de funcionalidade do servidor web foi a interação do servidor com o roteador, para habilitar e desabilitar as funcionalidades do roteador. Foi considerado sucesso neste teste, quando foi possível verificar no roteador se a funcionalidade foi ativada ou desativada.

### 2.3. Banco de dados

O acesso ao banco de dados via Java foi testado executando os comandos SQL, e validados pelo administrador de banco de dados. Depois de definidas as “queries” que serão utilizadas pelo software, uma bateria de chamadas foi realizada para verificar se não ocorreria nenhuma situação não prevista.

### 2.4. Leitura do arquivo de “log”

A parte mais importante à tarifação é o cadastro dos pacotes que trafegam pelo roteador. Existem duas limitações que deveriam ser levadas em conta, o desempenho do software e o desempenho do gerenciador de banco de dados. Como essa é uma aplicação crítica, esse software foi testado com diversos tipos de pacotes registrados e com arquivos de “log” dos mais variados tamanhos.

A validação pacote a pacote é inviável devido à quantidade que foram registradas em cada teste. Funcionar nos testes com menor quantidade de pacotes não é uma condição suficiente para garantir que não haverá erros com uma quantidade muito maior de arquivos. Dessa forma para validar esse teste, utilizamos amostragem de pacotes e a quantidade total que foi registrada.

### 2.5. Teste integrado

Como teste final das funcionalidades do roteador junto com o software, foi criado uma situação simulada de uso. Nesta simulação tivemos três personagens, dois clientes com “notebook” e um administrador do roteador.

Os clientes solicitam o uso do roteador ao administrador, que através da tela de registro de clientes cadastra cada um dos dois clientes. Individualmente, cada cliente acessa a página de "login", que irá liberar o acesso a Internet.

Após obterem acesso à Internet, os usuários passam a utilizar a Internet da forma que desejarem, ao final, durante o uso, aleatoriamente acessam a página que contém as informações de quanto já utilizaram e quanto devem pagar ao administrador.

Após terminar o uso, os clientes procuram o administrador para pagar o acesso. O Administrador consulta o valor a ser pago, e registra o pagamento, bloqueando o acesso a internet referente ao endereço IP do cliente.

Durante toda essa simulação, uma pessoa monitorava o software através das saídas previamente determinadas, suficientes para garantir que o funcionamento seguiu os requisitos levantados durante a fase de concepção do projeto.



## **Considerações Finais**

O roteador foi desenvolvido a partir de um computador, mas pode ser desenvolvida em qualquer equipamento em arquitetura i386, a escolha de desenvolver no computador foi devido ao preço de desenvolver em outro tipo de equipamento.

A conexão a rede a ser distribuída pelo roteador pode ser provida de qualquer tipo de provedor das empresas de telefonia celular, a utilização de EDGE foi devido a um dos integrantes do grupo apresentar um celular que tinha esse tipo de conexão.

Apesar de o sistema operacional instalado estar sem a interface gráfica reduzindo assim o consumo de recursos do equipamento, este poderia ser reduzido ainda mais retirando outros pacotes de aplicações do sistema operacional que não são utilizados, mas para efeito de desenvolvimento deste protótipo esta diferença não acarretará uma mudança significativa.

Para a implementação do software foram seguidas as boas praticas da engenharia de software para garantir a qualidade do serviço e para facilitar a identificação de erros, o software foi desenvolvido baseado em threads para otimizar os processos cargas dos pacotes, na conexão com o banco de dados a thread permite conexões simultâneas, na conexão com o servidor web foi utilizada para viabilizar acessos simultâneos.

Desta forma ficou demonstrado que é possível o desenvolvimento de um hardware móvel que seja capaz de distribuir conexão a rede a aparelhos nas proximidades através das interfaces mais populares.



## Referencias

- BELKIN. **Wireless G Router**: User Manual. Estados Unidos [2007]. 110 p. (P74559-B)
- SAMSUNG ELETRONICS. **SGH-D900**: User's Guide. Coréia do Sul [2006]. 135 p. (GH68-12771A)
- YISO. **Modem Yiso Modelo U893**: Manual de Instruções. São Paulo [2006]. 56 p.
- A Look Inside Network Address Translator  
<[http://www.cisco.com/web/about/ac123/ac147/archived\\_issues/ipj\\_7-3/anatomy.html](http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_7-3/anatomy.html)> Acesso em: 18 nov 2007
- About the Java Technology  
<<http://java.sun.com/docs/books/tutorial/getStarted/intro/definition.html>>  
Acesso em: 15 nov 2007.
- ACME SYSTEMS. FOX board a complete Linux System. Disponível em:  
<<http://www.acmesystems.it/>>. Acesso em: 02 abr. 2007.
- ATMEL CORPORATION. ATSTK1000. Disponível em:  
<<http://www.atmel.com/>>. Acesso em: 02 abr. 2007.
- CANONICAL LTD. Ubuntu Home Page. Disponível em:  
<<http://www.ubuntu.com/>>. Acesso em: 24 jun. 2007.
- CLARO. Claro – Acesso à Internet. Disponível em:  
<<http://www.claro.com.br/>>. Acesso em: 30 mai. 2007.
- GPRS and EDGE to 3G <[http://www.3g-generation.com/gprs\\_and\\_edge.htm](http://www.3g-generation.com/gprs_and_edge.htm)>  
Acesso em: 13 out 2007
- GSM World <<http://www.gsmworld.com/technology/gprs/index.shtml>> Acesso em: 20 nov 2007
- HUMBERT, B. et al. **Linux Advanced Routing & Traffic Control HOWTO**. Holanda, 2003. 158 p. Disponível em: <<http://lartc.org/lartc.pdf>>. Acesso em: 24 jun. 2007.
- Hypertext Transfer Protocol – HTTP/1.1  
<<http://www.w3.org/Protocols/rfc2616/rfc2616.html>> Acesso em: 20 nov 2007.
- IEEE-SA 802.11 <<http://standards.ieee.org/getieee802/802.11.html>> Aceso em: 15 jul 2007.

- INTERNET SYSTEMS CONSORTIUM. Dynamic Host Configuration Protocol. Disponível em: <<http://www.isc.org/>>. Acesso em: 15 jul. 2007.
- MARINE SERVICE. Inversor Batelco DC/AC 12VDC – 120VAC 500W. Disponível em: <<http://www.marine.com.br/>>. Acesso em: 22 set. 2007.
- RUSSEL, Paul. Linux 2.4 Packet Filtering HOWTO. Disponível em: <<http://www.netfilter.org/documentation/HOWTO/packet-filtering-HOWTO.html>>. Acesso em: 24 jun. 2007.
- TIM. Quanto custa. Disponível em: <<http://www.tim.com.br/>>. Acesso em: 30 mai. 2007.
- Tradicional IP Network Address Translator <<http://tools.ietf.org/html/rfc3022>> Acesso em: 15 nov 2007
- VIVO. Preços e Tarifas Vivo Zap Empresas. Disponível em: <<http://www.vivo.com.br/>>. Acesso em: 30 mai. 2007.

## APÊNDICE I

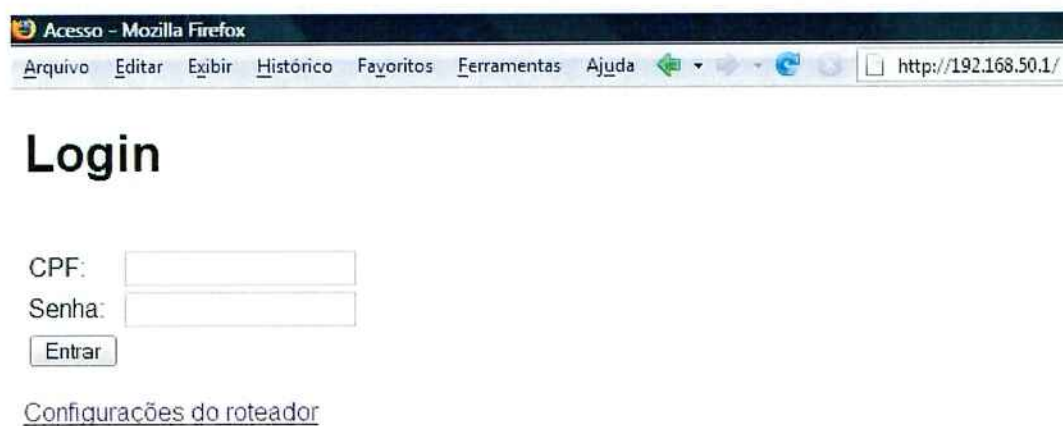
### Manual de operação

Os itens a seguir descrevem o funcionamento do software de tarifação do roteador.

#### 1. Acesso ao software

Após ligar o roteador, para acessar o software de tarifação basta conectar alguns dispositivo nele através de um cabo de rede ou através da “conexão wireless” e em seguida, com algum navegador, acessar o endereço 192.168.50.1.

A tela que será exibida segue na figura abaixo.



The screenshot shows a web browser window titled "Acesso - Mozilla Firefox". The address bar displays "http://192.168.50.1/". The page content includes a "Login" heading, two input fields for "CPF:" and "Senha:", and an "Entrar" button. Below the login fields is a link labeled "Configurações do roteador".

**Figura 30** Tela inicial do software de tarifação

## 2. Acesso do administrador

A maior parte das funcionalidades do software são de acesso restrito ao administrador do negócio. Elas estão disponíveis ao selecionar a opção “Configurações do roteador” na tela inicial de para autenticação, o usuário será levado a uma nova tela de autenticação onde apenas o administrador deve conhecer o nome de usuário e senha de acesso.



**Figura 31** Tela de autenticação do administrador

### 3. Opções de configurações do roteador

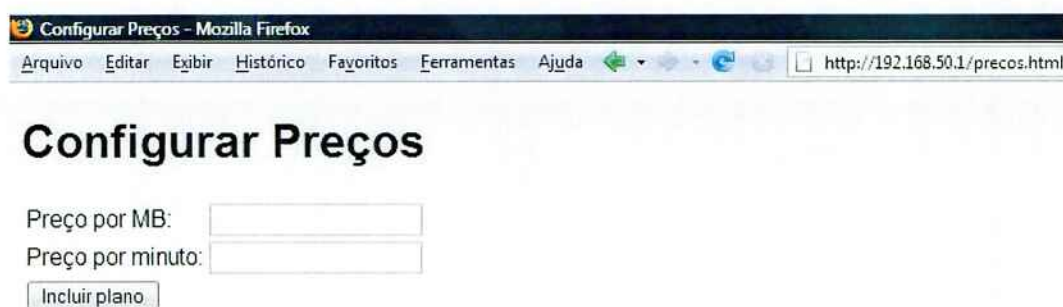
Após a autenticação do administrador, será exibida a tela abaixo.



**Figura 32** Configuração do roteador

### 3.1. Adicionar plano de acesso

Esta opção é usada para cadastrar novos planos de acesso tarifado no roteador. Pode-se inserir valores para cobrança de dados transferidos e por tempo utilizado.



The screenshot shows a web browser window with the title "Configurar Preços - Mozilla Firefox". The address bar displays the URL "http://192.168.50.1/precos.html". The main content area has the heading "Configurar Preços". Below the heading, there are two input fields: "Preço por MB:" and "Preço por minuto:". Below these fields is a button labeled "Incluir plano".

**Figura 33** Cadastramento de plano de acesso

Para excluir algum plano, basta utilizar a opção "apagar plano de acesso".

### 3.2. Cadastrar novo usuário

Para que um cliente possa ter acesso à Internet, ele primeiro deve ser cadastrado. Os seus dados serão inseridos pelo administrador através da opção “Cadastrar novo usuário”. Na tela do cadastro, serão exibidas as opções de planos de acesso disponíveis no até o momento.



Cadastro - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda <http://192.168.50.1/inicial.html>

## Cadastro de usuario

Nome:

Sobrenome:

Celular:

CPF/Login:

Senha:

Plano:

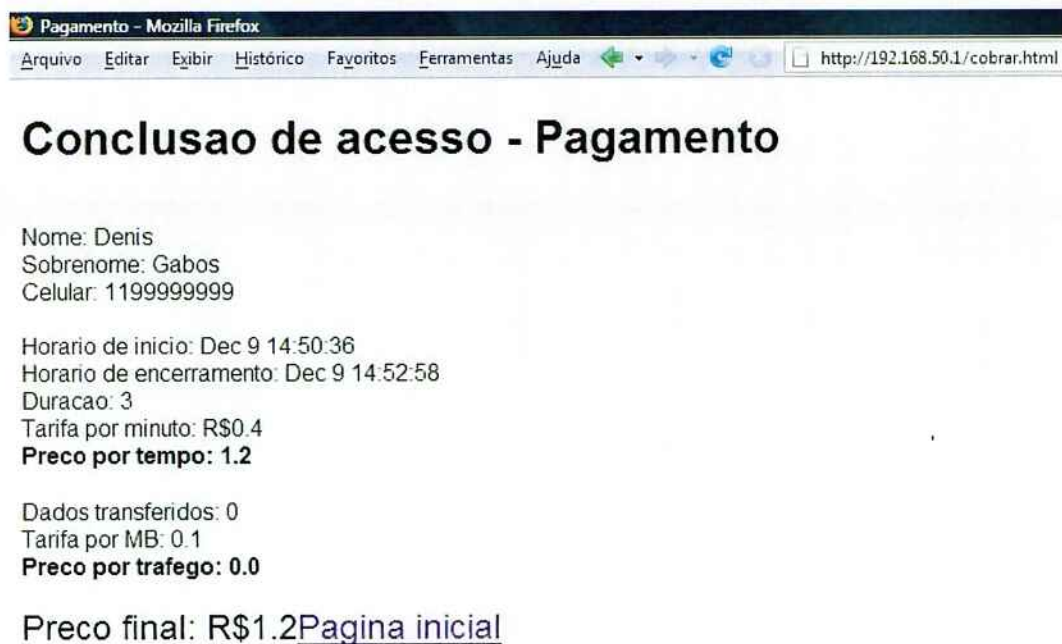
Plano	Tarifa(R\$)/MB	Tarifa(R\$)/minuto
1	0.1	0.4

**Figura 34** Tela de cadastro de cliente



### 3.3. Finalizando a conexão

Para encerrar o acesso basta utilizar a opção "Finalizar conexão". No final do processo, será exibida a valor total a ser paga pelo cliente.



**Figura 35** Tela de cobrança

### 3.4. Controle do roteador

Existem ainda alguns botões que executam operações que configuram o Linux para correto funcionamento do roteador.

O botão "Ativa roteador" limpa as tabelas de roteamento e bloqueia o acesso para endereços externos em todos dispositivos conectados ao roteador.

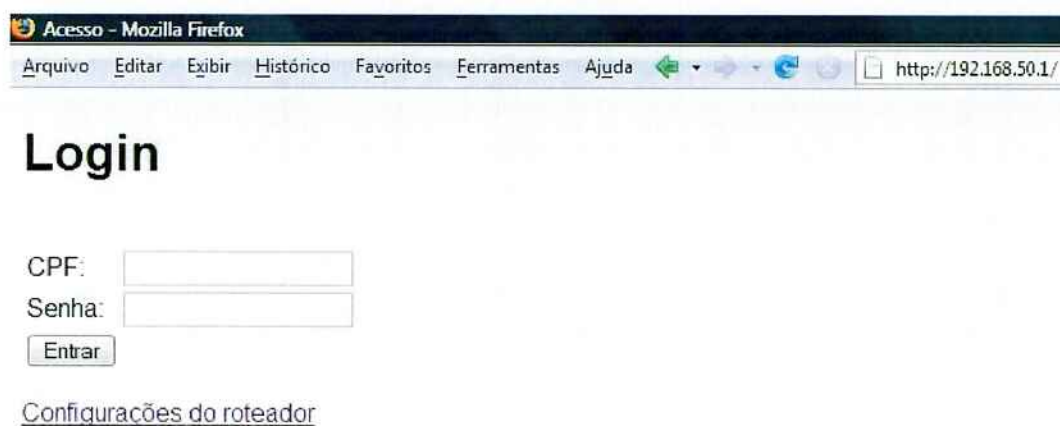
O "Desativa roteador" foi uma opção criada para ser usada em casos de malfuncionamento do roteador. Ela libera o acesso para todas as estações conectadas e desativa o registro de "log" dos dados trafegados.

Os botões "Conectar celular" e "Desconectar celular" servem respectivamente, para iniciar a conexão com a rede de telefonia celular e para finalizar a mesma.

O "Ajusta rota" é usado ajustar a rota principal de saída de dados para a Internet após que a conexão do celular é estabelecida com sucesso.

#### 4. Cliente liberando o seu acesso

Após ser cadastrado, basta o cliente digitar seu CPF e senha, criada durante o cadastro, acessando o endereço 192.168.50.1 para ter seu acesso liberado e utilizar a Internet.



Acesso - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://192.168.50.1/

## Login

CPF:

Senha:

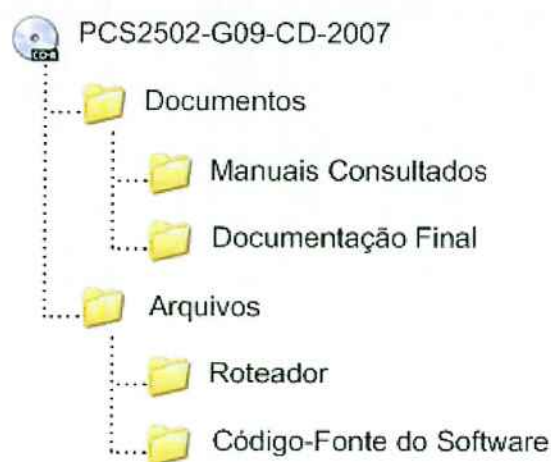
[Configurações do roteador](#)

**Figura 36** Tela inicial do software de tarifação

## APÊNDICE II

**Apresentação do CD**

Foi elaborado um CD contendo alguns documentos utilizados para desenvolver o projeto. Este contém também os códigos-fonte, scripts e modificações feitas no sistema operacional Linux.



**Figura 37** Estrutura de pastas do CD

**1. Pasta "Manuais Consultados"**

Esta pasta contém a versão em formato PDF dos manuais do "access point", do celular Samsung d900 e de outros documentos que foram consultados ao longo do desenvolvimento do projeto.

## 2. Pasta “Documentação Final”

Contém a versão em formato PDF deste documento, além da apresentação em formato “Power Point” realizada pelo grupo para avaliação da banca de professores.

## 3. Pasta “Roteador”

Esta pasta possui todos os scripts criados para o executar comandos do Linux como liberar acesso a Internet para um endereço IP, configurar o NAT, ajustar rota principal de saída de dados, entre outros. Foram adicionados também, todos os arquivos de configuração que foram criados ou modificados para que os requisitos do projeto fossem atendidos.

## 4. Pasta “Código-Fonte do Software”

Contém os códigos-fonte em linguagem Java do software de tarifação que controla o roteador.