

SUÊNIO TRINDADE ALVES

**SISTEMA PARA CONFIGURAÇÃO E ANÁLISE DE REDES *WIRELESS*
PARA CLIENTES EM AMBIENTE LINUX**

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências da disciplina Projeto Orientado para obtenção do título de Bacharel em Ciência da Computação.

Orientador

Professor Rêmulo Maia Alves

Co-orientador

Professor Anderson Bernardo dos Santos

LAVRAS
MINAS GERAIS - BRASIL
2002

SUÊNIO TRINDADE ALVES

**SISTEMA PARA CONFIGURAÇÃO E ANÁLISE DE REDES *WIRELESS*
PARA CLIENTES EM AMBIENTE LINUX**

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências da disciplina Projeto Orientado para obtenção do título de Bacharel em Ciência da Computação.

Aprovada em ____ de _____ de 2002.

Prof. Anderson Bernardo dos Santos
CIN/UFLA

Prof. Rêmulo Maia Alves
DCC/UFLA
(Orientador)

LAVRAS
MINAS GERAIS – BRASIL

SISTEMA PARA CONFIGURAÇÃO E ANÁLISE DE REDES WIRELESS PARA CLIENTES EM AMBIENTE LINUX

RESUMO

Sistema desenvolvido para a configuração de drivers de suporte para redes *Wireless LAN* e análise gráfica da conexão de uma rede *Wireless LAN* utilizando interface gráfica para um cliente em ambiente Linux. O Sistema foi desenvolvido na Universidade Federal de Lavras - MG para auxiliar no controle do servidor *Wireless* no Centro de Informática da UFLA – (CIN-UFLA) e para ser utilizado por clientes Linux. A ferramenta utilizada para o desenvolvimento do sistema foi a Qt-Designer que trabalha com a linguagem de programação C++. O sistema operacional usado é o Red Hat 7.3.

Palavras Chaves: wireless, interface gráfica, sistema, linux.

SYSTEM FOR PATTERN AND ANALYSIS OF NETWORKS WIRELESS FOR CUSTOMERS IN ENVIRONMENT LINUX

ABSTRACT

System developed for the pattern of drivers of bed for networks *Wireless LAN* and graphical analysis of the connection of a network *Wireless LAN* using graphical interface for a customer in Linux environment the System was developed in the University Federal of Lavras - MG to assist in the control of the *Wireless* server in the Center of Computer science of the UFLA - (CIN-UFLA) and to be used by Linux customers. The tool used for the development of the system was the Qt-Designer that works with the programming language C++. The used operational system is the Red Hat 7.3.

Words Keys: wireless, graphical interface, system, linux.

Sumário

1	Introdução	1
2	Redes Wireless	3
2.1	Redes Locais Sem Fio IEEE 802.11	5
2.2	Arquitetura de Rede Sem Fio 802.11	7
2.3	<i>Roaming</i>	9
2.4	Estações Perdidas (<i>hidden node</i>)	9
2.5	Transmissão em Redes Locais Sem Fio (<i>WLAN</i>)	11
2.6	Novo Padrão para Rede Local Sem Fio	12
2.7	Outras Considerações Sobre Redes Sem Fio	14
2.8	Componentes Para Redes Sem Fio	15
2.9	Sistemas de monitoração, configuração de redes sem fio	18
2.10	Conclusão	20
3	Implementação do Sistema	22
3.1	Análise dos Arquivos	23
3.1.1	Arquivo de Configuração de Drivers de Suporte para Redes Wireless	23
3.1.2	Arquivo de Análise da Conexão Wireless	27
3.2	Construção das Classes	28
3.2.1	Classe <i>clWireless.h</i>	28
3.2.2	Classe <i>clAdapter.h</i>	33
3.2.3	Classe <i>clink.h</i>	34
4	Implementação das Interfaces	36
4.1	Interface de Configurações de Drivers de Suporte para Redes Wireless LAN	37
4.1.1	Interface Inicial	37
4.1.2	Janela de Exibição dos Drivers	40
4.1.3	Interface de edição, inserção e remoção de drivers	43
4.1.4	Interface de Configuração do Cliente e da Rede	50
4.2	Interface de Análise da Conexão da Rede Wireless LAN	51

5 Conclusões e propostas futuras	58
5.1 Conclusões	58
5.2 Propostas Futuras	59
Bibliografia	60
Anexos	63
Anexo A	64
Anexo B	65
Anexo C	66
Anexo D	67
Anexo E	68
Anexo F	69
Anexo G	70
Anexo H	71
Anexo I	72
Anexo J	73
Anexo L	74

Lista de Figuras

2.1 - Conexão de uma rede sem fio com uma convencional com fio . . .	5
2.2 - Comparação do padrão 802.11 com o RM-OSI	6
2.3 - União de duas BSS formando uma ESS [Soares95, Pag 271] . . .	8
2.4 - Perda de conexão com a estação móvel por razão geográfica	10
2.5 - AP escolhe uma estação móvel mais próxima da estação perdida para usar como ponte	10
2.6 - Rede local sem fio <i>Ad Hoc</i>	13
2.7 - Rede local sem fio com infra-estrutura	13
2.8 - Cartão PCMCIA para notebooks	15
2.9 - Cartão ISA para computadores	16
2.10 - Comunicação de redes sem fio de longa distância, usando antenas direcionais.	16
2.11 - Antena onidirecional baseada em célula	17
2.12 - Ponto de Acesso (AP - <i>Access Point</i>)	18
4.1 - Janela Inicial	38
4.2 - Menu <i>Action</i>	38
4.3 - Menu <i>Advanced</i>	38
4.4 - menu <i>help</i>	39
4.5 - figuras da situação do link	39
4.6 - Formulário de Adição, edição e remoção de drivers.	40
4.7 - Caixa de seleção de drivers.	41
4.8 - Informações de drivers no campo <i>Settings</i>	41
4.9 - Janela de edição e inserção de driver (<i>configuration</i>)	44
4.10 - Janela de edição e inserção de driver (<i>configuration 1</i>)	45
4.11 - Janela de edição e inserção de driver (<i>configuration 2</i>)	46
4.12 - Janela de edição e inserção de driver (<i>configuration 3</i>)	47
4.13 - Janela Client Configuration.	50
4.14 - Interface de Análise (<i>Link Test</i>)	52
4.15 - Interface de Análise (<i>Level Graphic</i>)	53
4.16 - Interface de Análise (<i>Noise Graphic</i>)	14
4.17 - Interface de Análise (<i>Link Graphic</i>)	55
4.18 - Interface de Análise (<i>Total Graphic</i>)	56

Lista de Tabelas

2.1 - Relações de distância x velocidade	17
4.1 – Informações	39

Capítulo 1

Introdução

Os avanços da comunicação nos últimos anos possibilitaram o surgimento de várias tecnologias, que desde então, procuram atender a real necessidade de seus usuários, com a melhor qualidade possível. No início eram máquinas mono-usuário, muito se teve que evoluir até chegar as redes de computadores atuais. Hoje em dia, as empresas estão apostando numa das mais novas e revolucionárias tendências tecnológicas: a comunicação *Wireless* (Rede sem fio).

As redes sem fio são consideradas a quarta revolução da computação, antecedida pelos centros de processamento de dados da década de sessenta, o surgimento dos terminais nos anos setenta e as redes de computadores na década de oitenta como disse [Lima00]. As redes domésticas sem fio (*wireless home networks*) estão cada vez mais populares na área de produtos eletrônicos e de computação voltados para o consumidor final, doméstico. Essa tecnologia está sendo utilizada para interligar computadores, impressoras, *handhelds*, *DVDs*, etc., criando um ambiente doméstico com alto grau de integração e conectado à Internet.

O advento da Internet mostrou a viabilidade e as possibilidades abertas pela interligação de computadores espalhados pelo mundo - de correio eletrônico a *e-commerce*. Profissionais envolvidos com tecnologia já há muito tempo tem

considerado a conexão de equipamentos instalados em residências, porém tais conexões são pouco viáveis quando efetuada via cabo, principalmente em função dos custos e das dificuldades de instalação das redes domésticas.

As redes *wireless* superam essas dificuldades. Já disponíveis há cerca de dois anos, estão agora se tornando suficientemente baratas, confiáveis e simples de instalar e operar, a ponto de passarem a interessar à massa de consumidores leigos, devendo em breve ser criada massa crítica que vai tornar sua utilização cada vez mais freqüente, a ponto de mudar a forma pela qual esses consumidores pensam e aplicam tecnologia em suas residências.

É importante observar que pela primeira vez, grandes empresas estão trazendo produtos ao mercado: *Sony*, *Philips*, *Cisco Systems* e *AOL Time Warner* estão entre elas. A *Microsoft* diz que seu *Windows XP* foi projetado para tornar o uso das redes sem fio simples; o envolvimento com o assunto é tal que Bill Gates considera que o padrão 802.11, que suporta as redes sem fio, é a mais importante inovação tecnológica dos últimos anos.

Com o crescimento das redes *wireless*, ouve a necessidade do surgimento de softwares para a configuração, manutenção e análise destas redes. No mercado atual existem vários softwares para o controle destas redes, sendo que as maiorias destes softwares já acompanham os próprios dispositivos e hardwares de redes sem fios que auxiliam no controle e manutenção destes hardwares. Muitos destes softwares que acompanham os dispositivos de conexão são feitos exclusivamente para determinados sistemas operacionais.

Observando este crescimento e vantagens das redes *wireless*, e a necessidade de sistemas que auxiliam na configuração, manutenção e análise das redes sem fio, procuramos neste projeto criar um sistema com interface gráfica e prática que facilite a configuração e análise de redes *wireless* para clientes em ambiente *Linux*. Deve-se observar que este software já existe para ambientes *Windows* e por isso houve a necessidade de se criar um sistema que auxilie nestes requisitos (configuração, manutenção e análise) trabalhando no sistema operacional *Linux* para o qual não existem muitos softwares com interfaces gráficas que auxiliem nestes requisitos.

Capítulo 2

Redes Wireless

Nos últimos anos a comunicação sem fio ganhou um espaço considerável nas tecnologias de transmissão de dados, deixando de existir apenas nas comunicações de longa distância (feitas através de satélite), para fazer parte de ambientes locais. Essa tendência foi fortalecida pelo investimento de instituições e empresas no sentido de aplicar a transmissão sem fio em redes de computadores.

Também apostando nessa nova tecnologia, o **IEEE** (*Institute of Electrical and Eletronics Engeneers*) constituiu um grupo de pesquisa para criar padrões abertos que pudessem tornar a tecnologia sem fio cada vez mais realidade. Esse projeto, denominado de **Padrão IEEE 802.11**, nasceu em 1990, mas ficou inerte por aproximadamente sete anos devido a fatores que não permitiam que a tecnologia sem fio saísse do papel. Um dos principais fatores era a baixa taxa de transferência de dados que inicialmente a tecnologia oferecia, que era em torno de Kbps.

De acordo com a elevação dessa taxa de transferência de dados que passou a atingir Mbps, a rede sem fio começou a ser vista como uma tecnologia promissora e a receber ainda mais investimentos para a construção de equipamentos que possibilitassem a comunicação sem fio entre computadores.

Atualmente o foco das redes de computadores sem fio (*Wireless*) se encontra no contexto das redes locais de computadores (*Wireless Local Area Network - WLAN*), tanto em soluções proprietárias como no padrão do IEEE. Primeiramente foram colocados em prática alguns padrões proprietários, através

de empresas como IBM, CISCO, Telecom e 3COM. Hoje essas e outras empresas baseiam seus produtos no padrão do IEEE, devido às inúmeras e já conhecidas vantagens que o padrão aberto oferece: interoperabilidade, baixo custo, demanda de mercado, confiabilidade de projeto, entre outras.

Fora das redes de computadores, muitas tecnologias sem fio proprietárias têm sido usadas para possibilitar a comunicação entre dispositivos sem fio. Essas tecnologias têm o propósito de permitir o controle remoto de equipamentos domésticos e interligar os periféricos (teclado, mouse, impressoras, etc) aos computadores, eliminando os fios e tornando mais flexível e prático o uso desses equipamentos. O *Bluetooth* e *HomeRF* são exemplos dessas tecnologias.

O padrão IEEE 802.11 define basicamente uma arquitetura para as WLANs que abrange os níveis *físico* e de *enlace*. No nível físico são tratadas apenas as transmissões com frequência de rádio (RF) e infravermelho (IR), embora outras formas de transmissão sem fio possam ser usadas, como microondas e laser, por exemplo. No nível de enlace, o IEEE definiu um protocolo de controle de acesso ao meio (protocolo MAC), bastante semelhante ao protocolo usado em redes locais Ethernet (CSMA/CD). O padrão IEEE 802.11 possibilita a transmissão de dados numa velocidade de 1 à 2Mbps, e especifica uma arquitetura comum, métodos de transmissão, e outros aspectos de transferência de dados sem fio, permitindo a interoperabilidade entre os diversos produtos WLAN [Soares95].

Apesar da significativa elevação da taxa de transferência de dados que subiu de algumas poucas dezenas de kilobits por segundo para 2Mbps, as WLANs não atendiam satisfatoriamente a necessidade de banda das empresas. Com isso, o IEEE investiu no melhoramento do padrão 802.11 (que passou a ser chamado de 802.11b), com a mesma arquitetura e tecnologia, mas com taxa de transferência de dados maiores, entre 5 e 11 Mbps, impulsionando de vez a tecnologia e estimulando as comunidades científica e industrial a padronizarem, projetarem e produzirem produtos para essas redes.

Portanto, a adoção de sistema sem fio vem crescendo significativamente, em que muitas soluções WLAN estão ou já foram implantadas em empresas, universidades e outras instituições do mundo inteiro. Isso indica, sem dúvida, que as redes de computadores sem fio são uma realidade e, provavelmente, nos próximos anos, substituirão ou serão adicionais aos sistemas com fio já existentes, passando a ser uma solução bastante interessante para as organizações, pois desta forma os pontos que necessitam de mobilidade são conectados à rede pelo meio “Wireless” e as estações fixas são ligadas à rede via cabo. “Fibra ótica e comunicação sem fio são as tecnologias do futuro”[Tenenbaum96]. Não só isso. Há uma tendência moderna de se implantar cada vez mais as redes sem fio ao invés de redes com fio. Essa propensão é motivada tanto por aspectos da inviabilidade da instalação de redes com fio em

certos lugares, como pelo barateamento dos equipamentos sem fio e da interoperabilidade oferecida pela tecnologia Wireless. Outros fatores relacionam-se com as facilidades de mobilidade e flexibilidade que as comunicações sem fio oferecem.

A tecnologia sem fio não é recente, mas seus produtos caros e sua baixa taxa transferência de dados, inviabilizaram seu uso. Porém, a tecnologia WLAN (*Wireless Local Area Network*) vem sendo muito usada na medicina móvel no atendimento aos pacientes, transações comerciais e bancárias, usadas onde não possa traspasar fios como construções antigas ou tombadas pelo patrimônio histórico, entre outros, lugares onde há a necessidade de mobilidade e flexibilidade nas estações de trabalho.

2.1 Redes Locais Sem Fio IEEE 802.11

Uma rede sem fio (*Wireless*) é tipicamente uma extensão de uma rede local (*Local Area Network - LAN*) convencional com fio, criando-se o conceito de rede local sem fio (*Wireless Local Area Network - WLAN*). Uma WLAN converte pacotes de dados em onda de rádio ou infravermelho e os envia para outros dispositivos sem fio ou para um ponto de acesso que serve como uma conexão para uma LAN com fio.

“Uma rede sem fio é um sistema que interliga vários equipamentos fixos ou móveis utilizando o ar como meio de transmissão” [IEEE 802.11a].

A Figura 1.1 ilustra uma rede sem fio conectada por um ponto de acesso (AP) a uma rede convencional com fio.

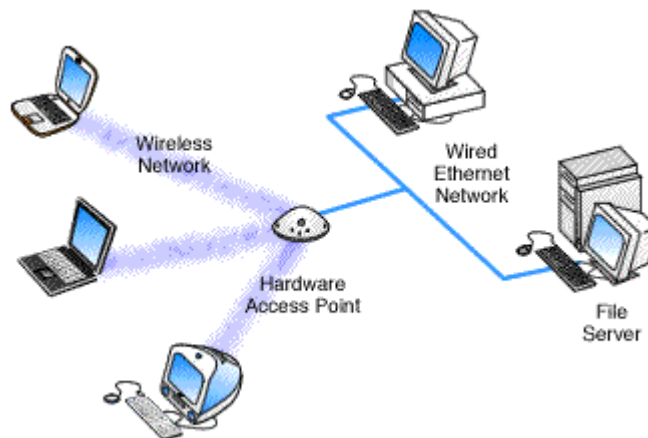


Figura 2.1- Conexão de uma rede sem fio com uma convencional com fio

O IEEE constituiu um grupo chamado de *Wireless Local-Area Networks Standard Working Group*, com a finalidade de criar padrões para redes sem fio, definindo um nível físico para redes onde as transmissões são realizadas na frequência de rádio ou infravermelho, e um protocolo de controle de acesso ao meio, o DFWMAC (*Distributed Foundation Wireless MAC*). Esse padrão é denominado de Projeto IEEE 802.11 [Soares95] e tem, entre outras, as seguintes premissas: suportar diversos canais; sobrepor diversas redes na mesma área de canal; apresentar robustez com relação a interferência; possuir mecanismos para evitar nós escondidos; oferecer privacidade e controle de acesso ao meio [Camara00].

A Figura 1.2 ilustra o padrão IEEE 802.11, comparando com o modelo padrão de redes de computadores, o RM-OSI da ISO (*Reference Model – Open Systems Interconnection of the International Standardization Organization*).

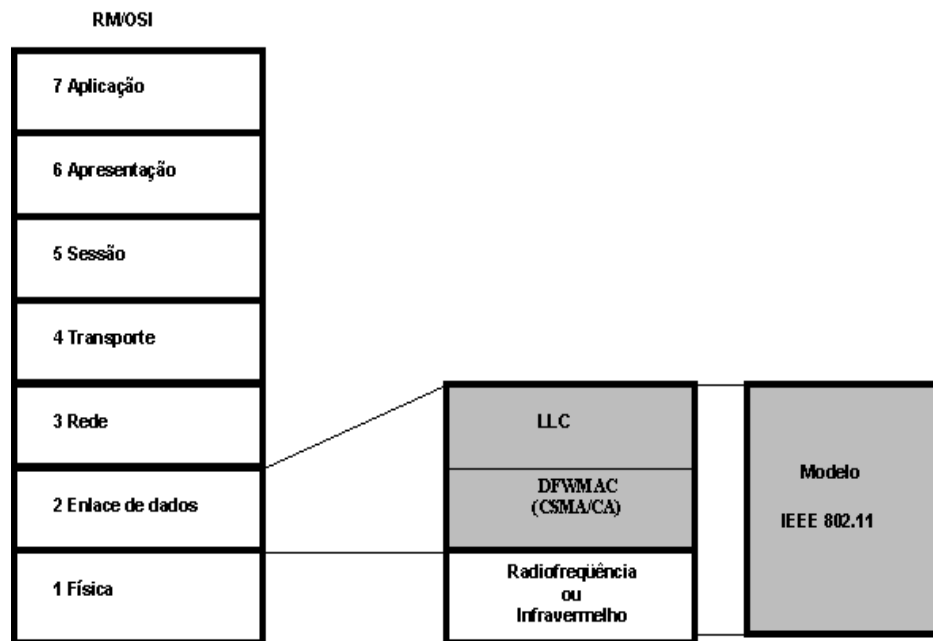


Figura 2.2 - Comparação do padrão 802.11 com o RM-OSI

A maioria das redes sem fio é baseada nos padrões IEEE 802.11 e 802.11b (sendo este último evolução do primeiro), para comunicação sem fio entre um dispositivo e uma rede LAN. Esses padrões permitem transmissão de dados de 1 a 2Mbps, para o padrão IEEE 802.11, e de 5 a 11Mbps, para o padrão IEEE 802.11b, e especificam uma arquitetura comum, métodos de transmissão, e

outros aspectos de transferência de dados sem fio, permitindo a interoperabilidade entre os produtos.

Duas razões contribuíram bastante para que a tecnologia sem fio avançasse: a aprovação do padrão IEEE 802.11, em 1997, o que ajudou a tornar as WLAN uma realidade; e o barateamento dos equipamentos para WLAN, que fizeram com que as redes sem fio ficassem mais acessíveis para algumas empresas, aumentando consideravelmente a comercialização de produtos para computadores móveis, como o cartão PCMCIA para Notebook e o cartão ISA/PCI para PCs.

2.2 Arquitetura de Rede Sem Fio 802.11

O padrão IEEE 802.11 define uma arquitetura para as redes sem fio, baseada na divisão da área coberta pela rede em células. Essas células são denominadas de BSA (*Basic Service Area*). O tamanho da BSA (célula) depende das características do ambiente e da potência dos transmissores/receptores usados nas estações. Outros elementos fazem parte do conceito da arquitetura de rede sem fio, quais sejam [Soares95]:

- **BSS (*Basic Service Set*)** – representa um grupo de estações comunicando-se por radiodifusão ou infravermelho em uma BSA.
- **Ponto de acesso (*Access Point – AP*)** – são estações especiais responsáveis pela captura das transmissões realizadas pelas estações de sua BSA, destinadas a estações localizadas em outras BSAs, retransmitindo-as, usando um sistema de distribuição.
- **Sistema de distribuição** – representa uma infra-estrutura de comunicação que interliga múltiplas BSAs para permitir a construção de redes cobrindo áreas maiores que uma célula.
- **ESA (*Extend Service Area*)** – representa a interligação de várias BSAs pelo sistema de distribuição através dos APs.
- **ESS (*Extend Service Set*)** – representa um conjunto de estações formado pela união de vários BSSs conectados por um sistema de distribuição.

A Figura 1.3, apresenta união de duas BSSs conectados por um sistema de distribuição.

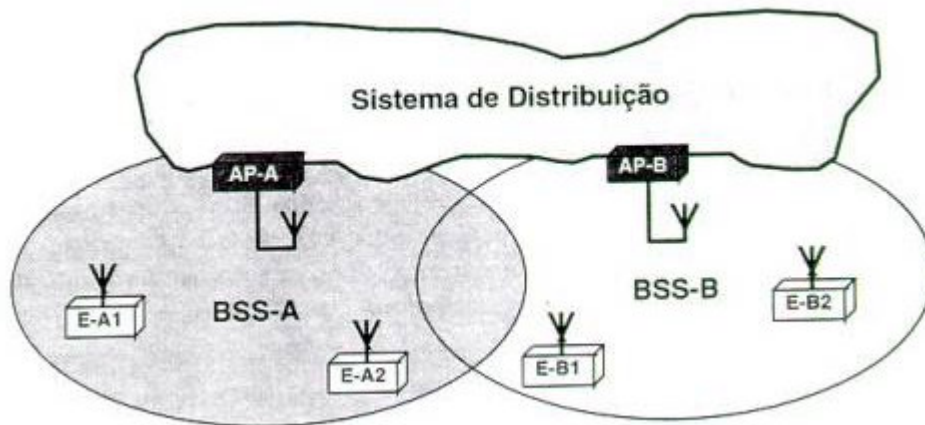


Figura 2.3 – União de duas BSS formando uma ESS [Soares95, Pag 271]

A identificação da rede ocorre da seguinte maneira: cada um dos ESSs recebe uma identificação chamada de ESS-ID; dentro de cada um desses ESSs, cada BSS recebe uma identificação chamada de BSS-ID. Então, o conjunto formado por esses dois identificadores (o ESS-ID e o BSS-ID), forma o *Network-ID* de uma rede sem fio padrão 802.11 [Soares95].

Apesar dos elementos que fazem parte da arquitetura sem fio possibilitar a construção de uma rede abrangendo áreas maiores do que um ambiente local, o projeto do IEEE 802.11 limita o padrão IEEE 802.11 às redes locais, com ou sem infra-estrutura.

Numa rede WLAN sem infra-estrutura (conhecidas por redes *Ad Hoc*), as estações se comunicam numa mesma célula, sem a necessidade de estações especiais, ou seja, sem necessidade dos APs para estabelecer as comunicações. Numa rede local com infra-estrutura, é necessária a interconexão de múltiplos BSSs, formando um ESS. Nesse caso, a infra-estrutura é representada pelos APs, e pelo sistema de distribuição que interliga esses APs. O sistema de distribuição, além de interligar os vários pontos de acesso, pode fornecer os recursos necessários para interligar a rede sem fio a outras redes, e ele, o sistema de distribuição, geralmente é representado por um sistema de comunicação com fio (cobre ou fibra) [IEEE802.11a].

Um elemento fundamental na arquitetura de rede local sem fio com infra-estrutura é o ponto de acesso, que desempenha as seguintes funções[Soares95]:

- *autenticação, associação e reassociação*: permite que uma estação móvel mesmo saindo de sua célula de origem continue conectada à infra-estrutura e não perca a comunicação.

A função que permite manter a continuidade da comunicação quando um usuário passa de uma célula para outra, é conhecida como *handoff* [Alencar98].

- *gerenciamento de potência*: permite que as estações operem economizando energia, através de um modo chamado de *power save*.

- *Sincronização*: garante que as estações associadas a um AP estejam sincronizadas por um relógio comum.

2.3 Roaming

O *roaming* é uma importante característica de comunicação sem fio. Permite que estações mudem de célula e continuem enviando e recebendo informações. Sistemas de *roaming* empregam arquiteturas de microcélulas que usam pontos de acesso estrategicamente localizados. O *handoff* entre pontos de acesso é totalmente transparente para o usuário.

Redes sem fio típicas dentro de prédios requerem mais que apenas um AP para cobrir todos os ambientes. Dependendo do material de que é feito as paredes dos prédios, um AP tem um raio transmissão que varia de 10 a 20 metros, se a transmissão for de boa qualidade. Se um usuário passeia com uma estação (aparelho sem fio), a estação tem que se mover de um célula para outra. A função do *roaming* funciona da seguinte forma [Alencar00]:

- Uma estação móvel, ao entrar em uma nova célula, e não estando em conversação, registra-se automaticamente pelo AP que controla a célula destino.

- Na célula visitada, o AP desta, irá verificar se a estação móvel visitante não havia se registrado anteriormente. Caso esse procedimento não tenha sido efetuado, o referido AP irá informar ao AP da célula origem sobre a nova posição.

- Com isso, o AP da célula origem fica sabendo da nova posição da estação móvel, e envia a informação a ela destinada, como se a referida estação estivesse em sua própria célula.

2.4 Estações Perdidas (*hidden node*)

Um dos grandes problemas em redes sem fio ocorre quando uma estação fica incomunicável por um período de tempo com o AP. São vários os motivos porque isto ocorre. O desligamento da estação móvel, a saída da estação móvel da área de atuação do AP ou entrada da estação móvel em uma área onde as ondas de rádio proveniente de outro lugar não se propagam ou locais com

grande degradação de sinal, que pode ser por motivos geográficos ou ambientais (área de sombra).

A Figura 2.6 ilustra uma perda de conexão do AP com a estação móvel por razões geográficas.



Figura 2.4 - Perda de conexão com a estação móvel por razão geográfica

O protocolo MAC trata o problema de estações perdidas da seguinte forma [Camara00]:

- Ao tentar comunicar-se com a estação móvel inúmera vezes sem obter resposta, o AP envia um *rquest* para todas as outras estações móveis sob sua área de cobertura. Cada uma destas envia um *request communication* para a estação perdida, esta por sua vez, envia um *response request* para todos avisando que está ativo.
- As estações que ouvirem esta comunicação envia um *bridge request*, diretamente para o AP, podendo que assim encontrar a melhor opção de comunicação entre o AP e a estação perdida.

A Figura 2.7 ilustra o AP escolhendo uma estação móvel para usar como ponte para comunicar-se com a estação perdida.



Figura 2.5 - AP escolhe uma estação móvel mas próxima da estação perdida para usar como ponte.

A comunicação do AP com a estação perdida será via “ponte”. O AP deve enviar dados para a ponte, como diretamente para a estação perdida. Assim se esta receber a comunicação, não há mais a necessidade da ponte.

Se o AP perder a comunicação com a ponte ou a ponte perde a comunicação com a estação perdida, o AP escolhe outra ponte entre as estações que respondera inicialmente.

Com este método o AP tem a chance de recuperar uma estação que por algum motivo tornou-se incomunicável com a rede.

2.5 Transmissão em Redes Locais Sem Fio (WLAN)

O Padrão IEEE 802.11 trata da tecnologia sem fio enfocando as redes locais sem fio (WLAN). Essas redes basicamente utilizam radiofrequência para a transmissão de dados, através de duas técnicas conhecidas como DSSS (*Direct Sequence Spread Spectrum*) e FHSS (*Frequency Hopping Spread Spectrum*), codificando dados e modulando sinais de modos diferentes para equilibrar velocidade, distância e capacidade de transmissão. A escolha da técnica DSSS ou FHSS dependerá de vários fatores relacionados com a aplicação dos usuários e o ambiente onde a rede operará.

Outras formas de transmissão também podem ser usadas em redes locais sem fio, como a transmissão em infravermelho, por exemplo. Mas transmissões com infravermelho não atravessam certos tipos de materiais, apesar de poder enviar mais dados do que a transmissão com radiofrequência. Com isso, a transmissão através de radiofrequência acaba sendo o padrão adotado nas transmissões WLAN.

Para a transmissão em radiofrequência são usadas as técnicas DSSS e FHSS. Essas técnicas transmitem os quadros de dados enviando-os por vários canais disponíveis dentro de uma frequência, ao invés de usar um único canal, possibilitando, dessa forma, a transmissão simultânea de vários quadros.

A técnica DSSS distribui o sinal em cima de uma gama extensiva da faixa de frequência e reorganiza os pacotes no receptor. A técnica FHSS envia segmentos curtos de dados que são transmitidos através de frequências específicas, controlando o fluxo com o receptor, que negocia velocidades menores comparadas às velocidades oferecidas pela técnica DSSS, mas menos suscetíveis a interferências.

O padrão 802.11 usa as duas técnicas, enquanto que outras tecnologias, como o *HomeRF* e *Bluetooth*, usam apenas a técnica FHSS, que é mais eficiente para ambientes que possuem outros tráficos de rádio, como áreas públicas abertas, por exemplo.

As WLANs baseadas em radiofrequência usam as faixas de frequência ISM (*Industrial - Scientific - Medical*), que assumem frequências de 900MHz, 2.4GHz e 5GHz. Quanto maior a frequência maior é a quantidade de informação

que um dispositivo pode enviar num canal. As primeiras WLANs operavam na frequência de 900MHz, atingindo uma taxa de 256Kbps. O padrão IEEE 802.11 aumentou a taxa de transmissão para 1Mbps, usando a técnica FHSS, e posteriormente para 2Mbps, usando a técnica DSSS, trabalhando na frequência de 2.4GHz.

A maioria das empresas optou pela técnica DSSS porque oferece frequências mais altas do que a FHSS.

2.6 Novo Padrão para Rede Local Sem Fio

Um novo padrão foi proposto pelo IEEE em 1998, o padrão IEEE 802.11b [IEEE802.11c], que tem se tornado o padrão atual para as WLANs. Esse novo padrão especifica a técnica básica de transmissão na camada física usando a técnica DSSS, passando a taxa de transmissão real de 2Mbps para 5Mbps (com a possibilidade de se chegar a 11Mbps), tornando as redes locais sem fio mais atrativas.

Esse novo padrão define os protocolos que cada estação tem que observar de forma que cada uma dessas estações tenha acesso justo ao meio de transmissão. Para isso, um método de controle é implementado de maneira que seja assegurada a possibilidade de uma estação transmitir num dado tempo.

O padrão IEEE 802.11b também define o protocolo para dois tipos de redes: redes *Ad Hoc* e redes com infra-estrutura (Cliente/Servidor).

- uma rede *Ad Hoc* é um sistema onde as comunicações são estabelecidas entre várias estações de uma mesma área (célula), sem o uso de um ponto de acesso ou servidor e sem a necessidade de infra-estrutura [Alencar98].
- uma rede Cliente/Servidor é sistema com infra-estrutura onde várias células fazem parte da arquitetura, e estações se comunicam com estações de outras células através de pontos de acesso usando um sistema de distribuição.

As Figuras 1.7 e 1.8 apresentam um exemplo de uma rede local sem fio *Ad Hoc* e uma rede com infra-estrutura, respectivamente.

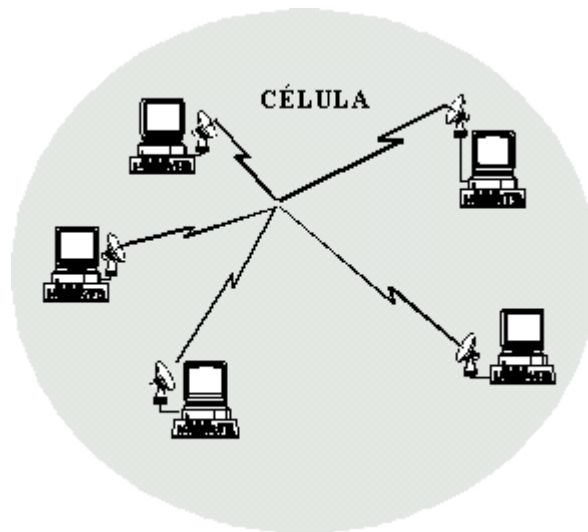


Figura 2.6 – Rede local sem fio *Ad Hoc*

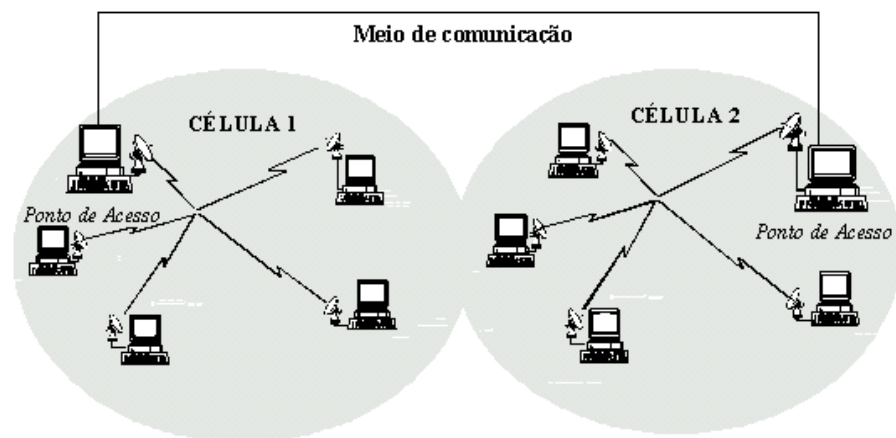


Figura 2.7 – Rede local sem fio com infra-estrutura

2.7 Outras Considerações Sobre Redes Sem Fio

O propósito de uma rede sem fio não é a de substituir as redes com fio, e sim, estendê-las. Hoje podemos ter num escritório uma WLAN operando aproximadamente a 5Mbps com uma distância máxima entre as estações de 25 metros. Numa comparação com uma LAN padrão com fio (uma rede Ethernet com cobre, por exemplo), essa taxa pode chegar até 100Mbps e a distância entre as estações até 100 metros. Isso prova que as redes sem fio ainda não substituirão com total eficiência às redes com fio. Além disso, a taxa de 11 Mbps das WLANs ainda não é praticada, sendo atingida apenas de 4 a 6Mbps, por várias razões, entre elas:

- o padrão 802.11b só é 85% eficiente no que diz respeito à camada física, devido à codificação, sincronização, e protocolos de transmissão acrescentarem cargas em cima do pacote de dados no nível de enlace;
- a subcamada de controle de acesso ao meio (MAC) trabalha com contenção, tendo que encontrar o melhor momento para transmitir, o que diminui a eficiência.

Por outro lado, as redes sem fio permitem maior mobilidade e flexibilidade na transmissão de dados. Elas são fáceis de montar, precisando apenas da colocação de cartões PCMCIA ou adaptadores PCI/ISA nas estações, e da instalação de pontos de acesso (*Access Points – APs*), que servem como intermediários entre uma rede local com fio e uma WLAN.

Segurança é a principal preocupação a cerca das redes sem fio, pois dados irão trafegar pelo ar e poderão ser interceptados por pessoas com equipamentos apropriados. Para essa questão de segurança, o padrão IEEE 802.11 definiu um mecanismo de segurança opcional e privativo, que provoca uma sobrecarga (*overhead*) na rede, mas que oferece segurança às redes sem fio tanto quanto às com fio. Para impedir que usuários não autorizados acessem sua rede sem fio, um valor de identificação chamado de ESS-ID, é programado em cada AP para identificar a subrede de comunicação de dados e funciona como ponto de autenticação das estações da rede. Se uma estação não puder identificar esse valor, não poderá se comunicar com o AP respectivo. Outros fabricantes duplicam a tabela de controle de endereços MAC sobre o AP, permitindo, dessa forma, que apenas estações com o endereço MAC reconhecido possam acessar a WLAN.

A existência de diversas tecnologias sem fio, como o *HomeRF*, *Bluetooth*, e HiperLAN2 (Europa), podem causar confusão para os consumidores e apresentar problemas de interoperabilidade, sem contar ainda que essas tecnologias podem apresentar interferências entre si, quando implantadas num mesmo ambiente, tendo em vista que esses padrões utilizam a mesma frequência de 2.4GHz, e apesar de usarem técnicas de transmissão diferentes, pacotes aerotransportados podem facilmente colidir. Atualmente a

probabilidade de isso acontecer é muito remota, mas de acordo com o crescimento dos usuários sem fio, essa probabilidade pode aumentar e esse problema pode se tornar uma realidade a ser considerada.

Embora ainda hajam muitas questões sendo analisadas a respeito das redes sem fio, a comunidade científica tem investido de forma significativa no melhoramento dos padrões, tentando oferecer uma velocidade que possa chegar até 50Mbps, e um alcance maior de transmissão que possa se aproximar à distância do padrão Ethernet (100 metros), de maneira que são esperados produtos com essas tecnologias ainda para os próximos anos.

É bom lembrar que a tecnologia de redes com fio, logo no início, também teve seus problemas que foram com o passar do tempo sendo corrigidos ou melhorados. Embora a tecnologia sem fio seja diferente das com fio, muitos investimentos estão sendo feitos para tornar as WLANs mais seguras, rápidas, e, conseqüentemente, mais atrativa.

2.8 Componentes Para Redes Sem Fio

Os componentes essenciais de LANs sem fio são os mesmos ou similares aos das LANs convencionais (cabeadas). A mudança maior está na substituição de cartões de interface de redes Ethernet e Token Ring pelos seus similares nas LANs sem fio, e a ausência de conectores de cabo, e do próprio cabo, evidentemente.

Os principais componentes são:

- **Cartões de interface de rede NICs (*Network Interface Cards*)**. Este deve ser da forma PCMCIA para notebooks ou cartões padrão ISA para computadores de mesa (PC).

A Figura 2.9 mostra um Cartão PCMCIA para notebooks de 11 Mbps.



Figura 2.8 – Cartão PCMCIA para notebooks

A Figura 2.10 mostra um Cartão ISA IEEE Turbo 10Mbps.



Figura 2.9 - Cartão ISA para computadores

- **Antenas para captar e difundir sinais de rádio.** Diversos tipos de antenas podem ser utilizados.
- Antenas direcionais que levam sinais de rede para longas distâncias, tais como edifício para edifício. Elas são montadas em postes ou mastros em telhados para assim aumentar o alcance.

A Figura 2.11 ilustra uma comunicação de redes sem fio de longa distância com visada direta (sem obstáculos), usando antenas direcionais com formato de parabólica.

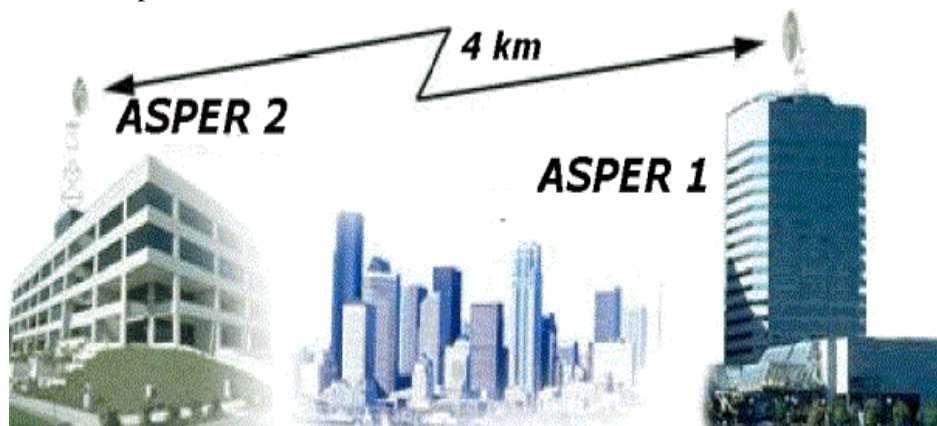


Figura 2.10 - Comunicação de redes sem fio de longa distância, usando antenas direcionais.

- Antenas omnidirecionais que são usadas em sistemas onde as comunicações são baseadas em células.

A Figura 2.12 mostra uma antena omnidirecional usada em comunicações baseadas em células.



Figura 2.11 - Antena omnidirecional baseada em célula

A taxa de transferência de dados depende muito da distância entre prédios, a tabela 4.1 apresenta as relações de distância x velocidade:

Transferência de Dados	Distância
Alta 11 Mbps	5.5 Km
Média 5.5 Mbps	7.9 Km
Padrão 2 Mbps	11.2 Km
Baixa 1 Mbps	15.8 Km

Tabela 2.2 - Relações de distância x velocidade

• **Pontos de acesso ou módulos de controle.** Equipamento com uma porta Ethernet e um slot PCMCIA para placa de rede sem fio, que funciona como bridge (ponte) entre a rede Ethernet tradicional e a rede sem fio, cada ponto de acesso pode atender até 200 estações, sendo recomendável um número de até 50 estações por AP de forma a manter adequado o nível de utilização da rede é composta por uma pequena antena opcional para ser utilizada no AP quando este é colocado dentro de um rack ou em PCs de forma a aumentar o alcance do sinal. Cria uma célula com raio de até 171m de alcance em ambiente aberto e 53m de alcance em ambiente semi-aberto.

A Figura 2.13 mostra um Ponto de Acesso (AP) com uma antena opcional.

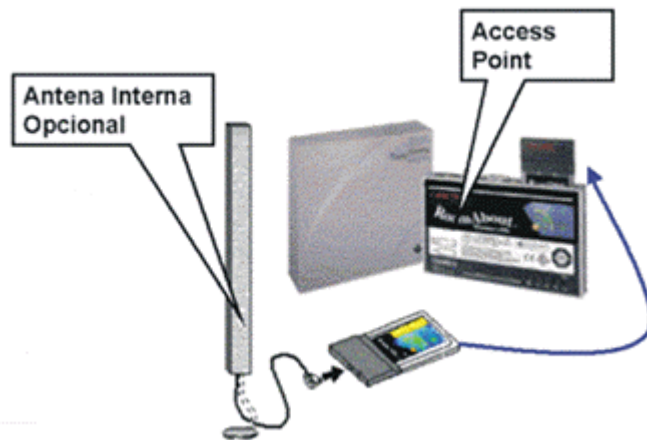


Figura 2.12 - Ponto de Acesso (AP - *Access Point*)

2.9 Sistemas de monitoração, configuração de redes sem fio

A importância da interface gráfica auxiliando na relação homem-máquina é uma realidade e muito visada hoje. A necessidade de criar softwares com boa qualidade devem satisfazer uma das principais áreas na engenharia de software: a usabilidade.

A usabilidade é definida tradicionalmente como a conjunção de cinco atributos:

- ◇ Facilidade de aprendizado: o sistema deve permitir que o usuário aprenda a executar suas tarefas no prazo mais curto possível;
- ◇ Eficiência de uso: o sistema, uma vez dominado pelo usuário, permite um alto grau de produtividade;
- ◇ Retenção: o sistema deve ser lembrado facilmente mesmo pelo usuário, permite um alto grau de produtividade;
- ◇ Minimização de erros: o sistema deve ter uma taxa baixa de erros de utilização. Além disso, os erros cometidos pelo usuário devem ser facilmente recuperáveis (por exemplo, volta a um estado seguro) e erros catastróficos não podem ocorrer;
- ◇ Satisfação: o sistema deve ser agradável de usar, ou seja, seus usuários ficam subjetivamente satisfeitos com ele.

Segundo [Nielsen93] a interface de um programa deve ter dialogo simples e natural, fale a língua do usuário e minimize a carga de memória do usuário.

Os sistemas atuais para monitoração, configuração e análise de redes sem fio têm buscado atingir estas interações com os usuários. Existem várias empresas que trabalham na criação e elaboração tanto de dispositivos para a construção de redes WLAN e na configuração, monitoração e análise das redes sem fio. Citaremos abaixo uma destas empresas que trabalham com esta tecnologia.

A Lucent Technologies Inc. trabalha com tecnologia de redes sem fio a vários anos e já tem vários produtos para wireless. Um dos kits de wireless é o Orinoco que junto com os dispositivos para conexão vem acompanhado softwares para a monitoração, configuração e análise da WLAN.

Este sistema dispõe para sistemas operacionais Windows softwares para monitoração da rede wireless como o Orinoco AP Manager e Orinoco Client Manager. Estes softwares consistem em ferramentas que exibem e modificam as configurações dos componentes da rede, configuram os componentes tais como os Access Points, criam diagnósticos da performance e se necessário identificam e resolvem os problemas da rede e ainda fazem a manutenção e otimização da conexão.

O Orinoco Client Manager é uma ferramenta de diagnostico para monitorar comunicações entre uma estação wireless e um ponto de acesso Orinoco, ou para monitorar o link entre duas estações de redes independentes [LUCENT00].

O Orinoco AP Manager é primeiramente uma ferramenta para administradores de LAN ou sistema supervisores. Ele também pode ser utilizado para configurar os AP's e para monitorar a performance de sua rede wireless.

Para Linux ainda não existe um software com interface gráfica prática para a análise e configuração da rede sem fio utilizando produtos da Lucent Technologies. Em relação aos softwares da Lucent Technologies toda a configuração e análise são feitas em arquivos localizados em diretórios no Linux.

Existe vários modos de configurar uma rede wireless utilizando os produtos da Orinoco. Abaixo citamos alguns modos configuração uma rede sem fio usando esta tecnologia.

Estação-a-Estação ("peer-to-peer network"):

As redes sem fio Orinoco são baseadas em uma estrutura celular em que cada estação Orinoco pode comunicar-se com outra ou com um ponto de acesso ligado à rede.

Pode-se criar um grupo de trabalho ou segmentos sem fio na rede cabeada existente.

Adicionando-se mais células, cria-se uma infraestrutura na qual os usuários podem se mover livremente mantendo-se conectados à rede.

Extensão de uma rede cabeada:

Você pode colocar seu computador rapidamente em rede sem a necessidade de lançamento de cabos, apenas instalando Orinoco/ISA e Orinoco/PCMCIA em seu PC ou Laptop.

Expansão de uma rede:

Não é necessário que as redes terminem onde terminam os cabos. Usando-se um ponto de acesso sem fio com o AP-1000, é possível adicionar estações de trabalho e/ou Notebooks na rede. Cada estação de trabalho ou notebook deverá possuir um cartão Orinoco/ISA ou Orinoco/PCMCIA para comunicação com o AP-1000.

Ponte entre duas redes:

Instalando-se vários Access Point`s é possível obter uma grande rede sem fio.

2.10 Conclusão

O crescimento das redes de computadores tem sido uma realidade e vem acontecendo de forma muito rápida. As formas de acesso a dados também têm mudado radicalmente, em que transações que antes eram feitas de formas fixas e centralizadas, hoje podem ser feitas de formas móveis e distribuídas. A cultura de utilização da informação também vem recebendo novas filosofias da era digital.

A propensão é que no futuro próximo, as redes de longa distância que são usadas para transporte de grande quantidade de dados, sejam construídas com a tecnologia de fibra ótica, devido às inúmeras vantagens que esse meio de comunicação oferece: largura de banda, imunidade a interferências, altíssima velocidade, alcance de grandes distâncias, etc.

Não é só isso. Uma outra tendência é que os dispositivos computacionais e domésticos usados em ambientes locais (residências, empresas, etc) sejam sem fio.

A rede local de computadores sem fio é um exemplo de que essa realidade já faz parte do avanço tecnológico. Por isso que o IEEE tem investido no melhoramento da tecnologia sem fio, para impulsionar a tecnologia e permitir que empresas projetem, construam e comercializem componentes para a comunicação sem fio.

Baseado nesse pensamento teremos sem dúvida, uma adoção crescente a sistemas sem fio no contexto das redes locais, substituindo gradativamente as

redes tradicionais cabeadas, para que no futuro tenhamos transmissões locais totalmente sem fio.

Observando este grande crescimento das redes sem fio, e a necessidade de softwares com boa usabilidade, este projeto visará a criação de um sistema com interface gráfica para clientes que trabalham em ambiente Linux que auxiliará nas configurações e da análise de da conexão do próprio cliente com a rede wireless.

Capítulo 3

Implementação do Sistema

Dividimos a construção do software em duas seções baseada nas necessidades que foram estabelecidas que são:

- Configuração
- Implementação de uma interface gráfica para configuração dos drivers de suporte para *Wireless LAN* em plataforma Linux.
- Análise
- Implementação de uma interface gráfica para análise da conexão do cliente Linux em sua rede *Wireless LAN*.

Iniciamos os estudos dos arquivos de configuração de drivers de suportes para redes Wireless e de análise da conexão do cliente Linux. Depois foi construído as classes na linguagem de programação C++ que iriam trabalhar com estes arquivos e finalmente foi construída a interface gráfica para plataforma Linux usando a ferramenta QT Designer tanto para a configuração quanto para a análise da conexão wireless.

3.1 Análise dos Arquivos

3.1.1 Arquivo de Configuração de Drivers de Suporte para Redes Wireless (/net/pcmcia/wireless.opts)

Os drivers de suporte para redes Wireless LAN estão listados no arquivo `wireless.opts` que esta na pasta `/net/pcmcia` do Linux e apresentadas no anexo A. Neste arquivo já existem algumas configurações de drivers como o Orinoco da Lucent Wavelan IEEE, Cisco/Aironet 4800/340, entre outros.

A configuração para trabalhar com Access Point é o scheme `essidany,*,*,)` que tem como campos INFO e ESSID que são respectivamente o nome do cliente Linux e o nome da rede *Wireless LAN*.

As próximas linhas do arquivo se referem a alguns dos drivers de suporte para redes *Wireless LAN* com suas respectivas configurações possíveis. Abaixo está uma lista dos drivers pré-listados no arquivo `wireless.opts`.

- Lucent Wavelan IEEE(+ Orinoco, RoamAbout and ELSA
- Cisco/Aironet 4800/340
- Samsung MagicLan (+ some other PrismII cards)
- Raytheon Raylink/WebGear Aviator2.4
- Old Lucent Wavelan
- Netwave (Xircom Netwave/Netwave Airsurfer)
- Proxim RangeLan2/Symphony
- No Wires Needed Swallow 550 and 1100 setting
- Symbol Spectrum24 setting
- Generic example (descreve todas as configurações possíveis)

Cada driver é reconhecido pelo seu endereço de máquina (*Mac Adress*) representado por seis dígitos pares hexadecimais no formato `xx:xx:xx*` depois das string `*,*,*,.` Este endereço é definido pela fábrica do cartão PCMCIA. Alguns drivers podem ter mais de um Mac Adress e são separados por “|” no arquivo `wireless.opts`.

Exemplo : `*,*,*,00:60:1D:*,*,*,00:02:2D:*`)

Os campos de configuração de um driver são:

INFO

Indica uma informação qualquer sobre o driver. Geralmente é o nome do driver ou a fábrica do mesmo.

Exemplo: INFO="Wavelan IEEE example (Lucent default settings) "

ESSID (extended network name)

Refere-se ao nome da rede wireless definida pelo cliente a ser usada na conexão. Como default pode se usar any[Tourrilhes2000].

Exemplo: ESSID = "SEMFIO"

NWID/Domain (cell identifier)

Refere-se ao domínio ou ID da rede para que se possa se distinguir logicamente diferentes redes[Tourrilhes97].

Exemplo: NWID="100"

MODE (Operation mode)

Os seis tipos de definição Ad-Hoc, Managed, Master, Repeater, Secondary e auto. Define o modo(ponto a ponto, gateway ou outros) que a conexão será usada[Tourrilhes].

Exemplo: MODE="Ad-Hoc"

FREQ ou CHANNEL (Frequency or channel)

Os dois indicam a parcela do spectrum de rádio limitado para um uso particular que será usada para conexão. A frequência é a separação física entre as redes. A maioria das redes wireless LANs usa na banda entre 2.4 to 2.48 GHz [Tourrilhes97].

Exemplo: FREQ="2.425G"
CHANNEL="5"

SENS (cell size + roaming speed)

Este campo define o nível do sinal para iniciar a recepção de pacotes (sensibilidade). O primeiro valor define o tamanho da célula e o segundo a velocidade em dBm da transmissão[Tourrilhes2002].

Exemplo: SENS = "1;-70" (dBm)

RATE (Bit rate)

Campo que define a velocidade com que cada bit será transmitido na camada física.

Exemplo: `RATE="auto"`

`KEY` (Encryption key)

Especifica uma de 4 chaves possíveis para criptografar os dados. Uma destas chaves, identificado por `Transmit_Key_ID`, é usado para cifrar os dados que são transmitidos por esta estação. Todas as chaves especificadas podem ser usadas para decifrar os dados que são recebidos. O valor chave pode ser uma string de caracteres ASCII ou um valor hexadecimal. O tamanho do valor chave pode ser 5 caracteres ou 10 dígitos hexadecimais para a criptografia padrão (cartão prata ou ouro), ou 13 caracteres ou 26 dígitos hexadecimais para a criptografia com chaves avançadas (cartão ouro somente). As chaves definidas na estação devem combinar as chaves definidas nos pontos de acesso(AP) [Valley2002].

Exemplo: `KEY = "2d73-31b7-96"`

`RTS` (RTS threshold - Request To Send)

Especifica o uso de *handshaking* na transmissão.

Exemplo: `RTS="500"`

`RTS="off"`

`FRAG` (Fragmentation threshold)

A fragmentação é a transmissão dos pacotes grandes em partes pequenas. Naturalmente, isto adiciona algumas sobrecargas, porque duplica o cabeçalho do pacote em cada fragmentos. Cada fragmento individualmente é verificado e retransmitido se necessário. A primeira vantagem é que caso de erro, o host necessita retransmitir somente um fragmento pequeno, sendo bem mais rápido. A segunda vantagem é que se o a média de ruído for grande, um pacote pequeno tem uma probabilidade mais elevada de começar completamente sem erros, assim que o nó aumenta sua possibilidade de sucesso em más condições[Valley2002].

Exemplos: `FRAG="off"`

`FRAG="1000"`

`IWCONFIG`

Define outros parâmetros para configuração da rede wireless. Abaixo alguns exemplos de atributos possíveis.

Exemplos: IWCONFIG="power on"
IWCONFIG="ap 01:23:45:67:89:AB"

IWSPY

Iwspy foi projetado para testar o suporte para IP móvel. Permite que o usuário ajuste uma lista de endereços de rede no driver. O driver recolherá a informação da qualidade para cada um daqueles endereços (atualizados cada vez que recebe um pacote desse endereço). A ferramenta permite que o usuário indique a informação associada com cada endereço na lista [Tourrilhes97].

Iwspy aceita o IP ADDRESS assim como endereços de máquina (MAC). Os endereços do IP serão convertidos aos endereços de máquina antes de ser transmitido ao driver. Nenhuma verificação é feita sobre o endereço de máquina [Tourrilhes97].

Este campo aceita o caractere + para adicionar um novo endereço na lista.

Exemplo: IWSPY = "01:23:45:67:89:AB +
01:23:45:67:89:AB"

IWPRIV

O iwpriv é um suporte experimental para extensões específicas do dispositivo. Alguns drivers (como o Wavelan) podem definir alguns parâmetros de funcionalidade extra, iwpriv são usados para manipular estes parâmetros [Tourrilhes97].

Exemplos: IWPRIV = "set_port 2"
IWPRIV = "set_histo 50 60"

Existem outros parâmetros possíveis em relação aos campos acima que poderão ser inseridos pelo cliente Linux para a conexão do driver utilizado. A maioria dos drivers não exigem todos os campos, sendo que o usuário pode utilizar apenas as que desejar ou que o driver requisiõe para uma melhor configuração.

Outros drivers podem ser inseridos no arquivo sendo que o endereço de máquina (MAC) define qual é o cartão ligado ao computador. Os campos necessários para a configuração de um driver geralmente é especificado pela fábrica.

3.1.2 Arquivo de Análise da Conexão Wireless

(/proc/net/wireless)

Este na pasta /proc/ e é um pseudo-arquivo de sistema que dá a informação e a estatística sobre o sistema atual. Estas entradas agem como arquivos, então dando um comando cat nele dará a informação requerida. /proc/net/wireless é projetado para exibir estatísticas específicas da conexão wireless em cada interface wireless no sistema (Anexo B). Esta entrada é de fato um clone do /proc/net/dev que dá as estatísticas dos drivers padrão[Tourrilhes97].

A saída deste arquivo está listado abaixo:

```
[suenio@localhost]$cat /proc/net/wireless
Inter-|sta| Quality | Discarded packets
face |tus|link level noise| nwid crypt misc
eth2: f0      15. 24.    4.   181    0    0
```

Para cada dispositivo, a informação é dada na listagem:

- *Interface* : Refere-se a interface wireless usada.
- *Status* : É o corrente estado da conexão. Esta é uma informação dependente do dispositivo.
- *Quality - link* : Qualidade geral da recepção.
- *Quality - level* : Nível da força do sinal da estação wireless no receptor.
- *Quality - noise* : Nivel de interferência no receptor.
- *Discarded - nwid* : número dos pacotes rejeitados devido à identificação de rede inválida.
- *Discarded - crypt* : número do pacote incapazes de decifrar.
- *Discarded - misc* : não utilizado (para agora)..

Estas informações permitem que o usuário tenha um feedback melhor sobre seu sistema. Um valor elevado do Discarded - nwid pode indicar um problema da configuração do nwid ou uma rede adjacente. O Quality - level pode ajudar a descobrir faixas de areas shadow. A diferença básica entre a Quality - link e Quality - level é que a primeira indica como bom à recepção é (para o exemplo a porcentagem de pacotes corretamente recebidos) e o segundo como "forte" o sinal é. Quando os valores da qualidade foram atualizados depois da última entrada lida, um ponto seguirá esse valor (tipicamente, ele significa que uma medida nova foi feita) [Tourrilhes97].

3.2 Construção das Classes

Depois de havermos estudado e analisado o arquivo de configuração `/net/pcmcia/wireless.opts` e o arquivo de estatística `/proc/net/wireless`, partimos então para a construção das classes em C++ que trabalharam com estes arquivos.

Para a configuração do arquivo `wireless.opts` foi criada as classes `clWireless` que está no arquivo `clwireless.h` e a classe `clAdapter` que está no arquivo `cladapter.h`. Ambos estão listadas no Anexo C e D respectivamente.

Para a análise do arquivo de estatística `wireless` foi criada a classe `clLink` que está no arquivo `cllink.h`. Este arquivo está listado no Anexo I.

Utilizamos da criação de classes para que pudéssemos trabalhar independente da interface gráfica do sistema que seria criada posteriormente.

Todo o código está escrito em inglês para futuras distribuições.

3.2.1 Classe *clWireless.h* (Anexo C)

Esta classe tem como funções principais a alteração nas configurações dos drivers de suporte para Wireless Lan listados no arquivo `wireless.opts` e também a inserção de novos drivers no arquivo, remoção de drivers.

O algoritmo consiste basicamente na seqüência abaixo listado:

- Abre o arquivo `wireless.opts`;
- Remove-se todos os comentários do arquivo inserindo os num vetor e um identificador de qual linha o comentário estava;
- Escreve todas as linhas não comentadas em um arquivo temporário;
- Cria-se objetos da classe `clAdapter` que contem todas as informações utilizadas pelos drivers e os insere em um vetor;
- Realiza as funções de alteração nos campos dos drivers, inserção e remoção de drivers direto no vetor de adaptadores(drivers) caso tenham sido requisitadas;
- Salva todas as alterações, caso tenham sido feitas dentro do arquivo `wireless.opts`;
- Fecha o arquivo `wireless.opts`.

Principais classes incluídas

```
#include<fstream>
```

Para abrir o arquivo `wireless.opts` e criar o arquivo temporário `wirelesstmp.tmp` para armazenar as linhas não comentados do arquivo.

```
#include<string>
```

Para trabalhar com as palavras lidas do arquivo e analisa-las.

```
#include<vector>
```

Para a criação de vetores que serão utilizados na classe para o armazenamento de dados.

```
#include "clAdapter.h"
```

Para a criação dos objetos que irão armazenar separadamente as informações de cada driver.

```
struct ESSIDANY
```

Para armazenar o nome do cliente (`info`) e o nome da rede usada (`ssid`).

```
Struct COMMENT
```

Para armazenar o comentário (`comment`) e sua respectiva linha (`lin`).

Atributos

```
vector<clAdapter> adapters;
```

Armazena os adaptadores existente nos arquivo `wireless.opts`

```
ifstream File;
```

Cria um canal de entrada para o arquivo `wireless.opts`

```
ifstream Temp;
```

Cria um canal de entrada para o arquivo `wirelesstmp.tmp`.

```
string wireless;
```

Armazena a leitura de cada palavra sobre um dado driver do arquivo para ser criado posteriormente o objeto da classe `clAdapter`

```
vector<string> vectorWireless;
```

Armazena todas as strings wireless num vetor que sera usado depois para a criação dos adaptadores pela classe `clAdapter`.

```
ifstream arqin;
```

Objeto de entrada para ser usado no arquivo temporário.

```
ofstream arqout;
```

Objeto de saída para ser usado no arquivo temporário

```
vector<COMMENT> vectorComment;
```

Armazena os struct contendo o comentário e sua respectiva linha no arquivo `wireless.opts`

```
unsigned int numAdapterOld;
```

Contem o numero e adaptadores antes da nova atualização para ser utilizado caso tenha sido inserido novos adaptadores

```
vector<string> newComment;
```

Armazena o comentário dos novos drivers inseridos.

Construtor

Inicializa removendo os comentários do arquivo `wireless.opts` e então trabalha com o arquivo `wirelesstmp.tmp` contendo todos os adaptadores wireless. Pega todos os valores e informações de cada adaptador contidos no arquivo temporário e os armazena em strings para adaptadores já definidos e outros adaptadores que foram inseridos são armazenados em um vetor (`vector<string> vectorWireless`). Então é chamado a função `fillAdapters()` que então, insere os adaptadores num vetor (`vector<clAdapter> adapters`).

Funções usadas pela classe

```
bool insertAdapter(clAdapter adapter ,string comment);
```

Inserir novos adaptadores no arquivo wireless.opts. Como parâmetros tem "clAdapter adapter" que é o novo adaptador, e "string comment" que contém um comentário sobre este novo adaptador que deverá ser inserido no arquivo wireless.opts.

Constrói-se a string que contém o endereço MAC do novo driver, e logo depois é avaliado cada campo que é utilizado por este driver observando se está preenchido e então ativando os booleanos de cada campo.

Terminado os campos ativados é então inserido no vetor de adaptadores. A função retorna true se o endereço mac foi definido.

```
bool removeAdapter(string macAddress);
```

Remove os novos adaptadores que foram inseridos no arquivo wireless.opts. Como parâmetros tem "string macAddress" que é indicará qual adaptador deverá ser removido do arquivo. Os adaptadores default (Lucent Wavelan IEEE (+ Orinoco, RoamAbout and ELSA), Cisco/Aironet 4800/34, Samsung MagicLan (+ some other PrismII cards), Raytheon Raylink/WebGear Aviator2.4, Old Lucent Wavelan e Netwave (Xircom Netwave/Netwave Airsurfer), não são removíveis.

Constrói-se a string que contém o endereço MAC do driver a ser removido, localiza-o no vetor de drivers e então é removido deste vetor. Ele retorna true se foi encontrado o driver e removido. Falso caso não exista o driver com o endereço MAC requisitado.

```
void fillAdapters();
```

Cria objetos da classe clAdapter que será inserido no vetor de adaptadores wireless.

As strings contido no vetor Wireless são transformados em objetos da classe clAdapter e então inseridos no vetor de adaptadores.

```
void saveAlterations();
```

Esta função é responsável pela escrita de todas as informações no arquivo `wireless.opts`.

É criado um vetor estático (`conf`) que irá armazenar todas as linhas do arquivo, tanto os comentários como as de informação dos drivers.

A primeira palavra inserida no vetor é o a linha de início do código - `case "$ADDRESS" in`. Depois é inserido o campo `essidany, *, *, *` que como vimos na seção 4.1.1, é responsável pela configuração do nome do cliente e do nome da rede.

Depois é inseridas as informações de cada driver neste vetor, e caso tenha sido inserido algum novo driver seu comentário será inserido no cabeçalho do mesmo. Isto é avaliado na linha de código `if(i>=numAdapterOld)` que define que o numero de adaptadores aumentou.

Então é inserido os comentários no vetor em suas respectivas linhas definidas pelo atributo `lin` do `struct COMMENT`.

E finalmente é gravado todo o vetor `conf` no arquivo `wireless.opts`.

```
clAdapter item(string macAdress)
```

Retorna o adaptador solicitado no parametro `string macAdress`.

```
void createVectorComment()
```

Cria o vetor que irá conter todos os comentários do arquivo `wireless.opts`.

Recebe o arquivo `wireless.opts` e realiza a leitura das linhas e caso sejam linhas de comentário `#` e então é inseridos no vetor de comentários.

```
bool comments(string a) e bool empty(string a)
```

São utilizadas na função `createVectorComment()` e servem para determinar se uma linha é um comentário ou se está vazia respectivamente.

3.2.2 Classe *clAdapter.h* (Anexo D)

A função desta classe é criar um objeto que irá conter todos as informações de um driver de suporte *Wireless LAN* que estão contidos dentro do arquivo `wireless.opts`, e será lido pela classe `clWireless` e inserido as informações dentro dos atributos deste objeto.

O algoritmo desta classe pode ser definido assim:

- apartir da string passada no parâmetro do construtor pela classe `clWireless`;
- analisa-se cada campo localizando-os dentro da string;
- armazena-os dentro das suas respectivas strings (atributos) ativando os respectivos booleanos deste campo;

Principais classes incluídas

```
#include<string>
```

É a única classe usada dentro da `clAdapter` e foi incluída para trabalhar com a string lida pela classe `clWireless` contida dentro do arquivo `wireless.opts`.

Atributos

```
string macAdress, info, essid, nwid, mode, freq, channel,  
sens, rate, key, rts, frag, iwconfig, iwspy, iwpriv;
```

Estas strings armazenam os campos lidos do arquivo de um dado driver.

```
bool Info, Essid, Nwid, Mode, Freq, Channel, Sens, Rate,  
Key, Rts, Frag, Iwconfig, Iwspy ,Iwpriv;
```

Quando um campo é usado o seu respectivo booleano é ativado para definir que este campo deste driver pode ser preenchido.

Construtor

Como foi dito antes este construtor recebe uma string que foi lida do arquivo `wireless.opts` pela classe `clWireless` e pega todos as definições deste adaptador e os insere nos atributos desta classe.

Inicializa primeiro os atributos booleanos com `false`. Depois localiza-se primeiro o endereço MAC e então é armazenado na string `macAddress`. Agora então, é localizado cada campo com ESSID, INFO, NWID e armazenado em suas strings caso tenham sido usadas e então ativando seus booleanos.

3.2.3 Classe `clink.h` (Anexo E)

Agora iremos trabalhar com o arquivo `wireless` que contem todas as estatísticas da conexão wireless de um cliente Linux. Esta classe irá trabalhar com este arquivo e então armazenar os valores da conexão.

O Algoritmo desta classe esta definido abaixo:

- Abre se o arquivo `wireless`;
- Realiza a leitura dos valores armazenando os em seus respectivos atributos;
- Fecha o arquivo;

Principais classes incluídas

```
#include<fstream>
```

Para abrir o arquivo `wireless` para leitura.

```
#include<string>
```

Para trabalhar com as palavras lidas do arquivo e analisa-las.

```
#include<vector>
```

Para a criação de vetores que serão utilizados na classe para o armazenamento de dados.

Atributos

```
string inter;
```

Armazena a interface usada na conexão (`eth0,eth1`);

```
string status;
```

Armazena o status corrente da conexão.

```
int link;
```

Armazena a qualidade geral da recepção.

```
int level;
```

Armazena o nível do sinal no receptor.

```
int noise;
```

Armazena o nível de ruído no receptor.

```
vector <string> discardsPackets;
```

Armazena os nomes das configurações relacionadas com os pacotes perdidos.

```
vector <string> valueDiscardsPackets;
```

Armazena os valores dos campos de configurações relacionados com os pacotes perdidos.

```
ifstream arqin;
```

Para a leitura de arquivo.

Construtor

O construtor recebe como entrada o arquivo "wireless", pula as duas primeiras linhas e pega os valores da interface, o status, o link, o level e o noise , os dados sobre Discarded Packet.

Funções usadas pela classe

```
int str2int(string str);
```

Esta função converte um numero no formato de string em um inteiro.

Capítulo 4

Implementação das Interfaces

Neste capítulo vamos iniciar o estudo sobre a interface do sistema de configuração e análise para plataforma Linux. Estudaremos cada janela do programa, as entradas e saídas do programa e o código que realiza a configuração e análise da rede.

Decidimos criar uma interface gráfica que facilita-se a configuração dos drivers de suporte para redes *Wireless LAN* existentes no arquivo `wireless.opts` e analisa-se graficamente a conexão wireless. Utilizamos a ferramenta de criação de interface gráfica QT Designer que trabalha com as bibliotecas da linguagem de programação C++ e cria programas para serem executados em ambiente Linux.

A interface do sistema basicamente se divide em dois segmentos:

- Interface de Configuração de Drivers;
- Interface de Análise da Conexão Wireless;

A interface de configuração de drivers realiza as alterações no arquivo `wireless.opts`, modificando os campos de informação dos adaptadores, inserindo novos adaptadores ou removendo algum.

A interface de análise da conexão wireless realiza a análise da conexão exibindo graficamente o nível da recepção, o nível do ruído e o nível do link, podendo salvar o histórico diário da conexão.

A interface é praticamente livre do arquivo, claro que obedecendo ao padrão do arquivo, podendo assim, ser feito também alterações no arquivo diretamente. A interface irá reconhecer naturalmente qualquer alteração feita diretamente no arquivo.

4.1 Interface de Configurações de Drivers de Suporte para Redes Wireless LAN

4.1.1 Interface Inicial

O programa de configuração dos drivers de suporte para redes *Wireless LAN* é formado por três formulários (janelas) principais responsáveis pela análise e configuração da rede wireless.

O primeiro formulário é a janela inicial do programa e exibe a condição da conexão wireless e contem menus que dão acesso as configurações de drivers e análise da conexão da rede.

O Segundo formulário é responsável pela exibição dos dados dos drivers de suporte para redes Wireless LAN e acesso ao formulário de edição, inserção e remoção de drivers.

O Terceiro formulário é responsável pela edição, inserção e remoção de drivers. Ele é formado por campos e opções para a definição dos atributos dos drivers.



Figura 4.1 – Janela Inicial

É formada por quatro menus, um botão de ok e um botão de ajuda (*help*) exibidos na figura 4.1 e um painel de informações sobre a conexão.

O menu *File* apenas é formado por um submenu *close* que fecha o programa. O menu *Action* exibido na figura 4.2 é formado pelo submenu *Add/Edit Configuration Drivers* que realiza a configuração dos drivers e pelo submenu *Client Configuration* que dá acesso a configuração do nome da rede e do nome do cliente.

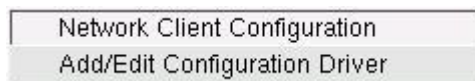


Figura 4.2 – Menu Action

O menu *Advanced* exibido na figura 4.3 é formado pelo submenu *Link Test* que dá acesso ao formulário de análise da conexão.

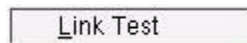


Figura 4.3 – Menu Advanced

O menu *Help* exibido na figura 4.4 dá acesso aos documentos de ajuda (*Contents*) e sobre as informações do programa(*Version*).



Figura 4.4 – menu *help*

Neste formulário é exibido as informações sobre a conexão e a condições dela no campo *informations*. Abaixo está uma lista de todas as mensagens possíveis sobre a situação atual da rede.

Informação	Situações Possíveis	Significado
Client Name		- Nome do Computador
Network Name		- Nome da Rede a qual o cliente está conectado
Link	<ul style="list-style-type: none"> - <i>Bad</i> - <i>Marginal</i> - <i>Good</i> 	<ul style="list-style-type: none"> - Conexão Ruim - Conexão Razoável - Conexão Boa

Tabela 4.1 – Informações

As figuras que definem a situação em relação ao *Link* estão abaixo citadas:



Bad Link



Marginal Link



Good Link

Figura 4.5 – figuras da situação do link

Análise do Código – wireless.ui.h (Anexo F)

Classe : wirelessForm

O código do formulário de entrada do sistema exibido no anexo II é formado basicamente pelas funções que dão acesso aos formulários de drivers, de configuração do cliente e de análise da conexão.

Abaixo está as funções contidas no arquivo wireless.ui.h que é o código do formulário principal.

```
void wirelessForm::execAddEdit()
```

Executa o construtor do formulário de edição e adição de drivers e exhibe este formulário.

```
void wirelessForm::execClientConf()
```

Executa o construtor do formulário de edição do nome do cliente e do nome da rede wireless usada.

```
void wirelessForm::execMonitorLink()
```

Executa o construtor do formulário de análise da conexão wireless

4.1.2 Janela de Exibição dos Drivers

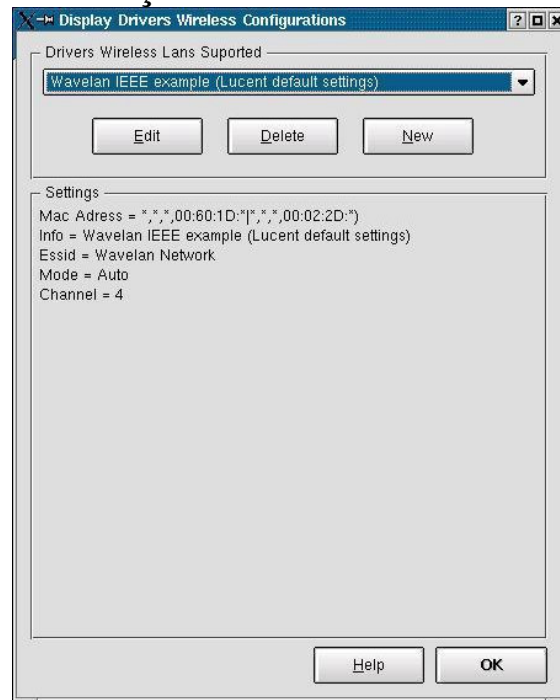


Figura 4.6 – Formulário de Adição, edição e remoção de drivers.

Esta janela apresentada na figura 4.6 é responsável pela exibição das informações dos drivers existentes no arquivo wireless.opts e acesso aos formulários de configuração, adição ou remoção de drivers de suporte para redes *Wireless LAN*. Ela é formado por três botões onde o primeiro (Edit) dá acesso ao formulário de edição de drivers; o botão do meio dá acesso ao formulário de adição de um novo driver que o cliente deseja inserir no arquivo wireless.opts. O terceiro botão é responsável pela remoção de um dos drivers que tivessem sido inseridos pelo cliente.

Na caixa de seleção de drivers exibido na figura 4.7 estão todos os drivers existentes no arquivo wireless.opts, e podendo ser selecionado para ver todos os campos de atributos dos *drives* que são exibidos no campo *Settings*.

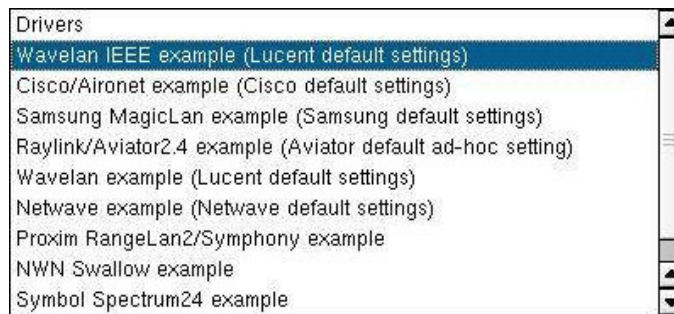


Figura 4.7 – Caixa de seleção de drivers

No campo *Settings* é exibido todas as informações sobre o driver selecionado na caixa de seleção como mostrado na figura 4.8.

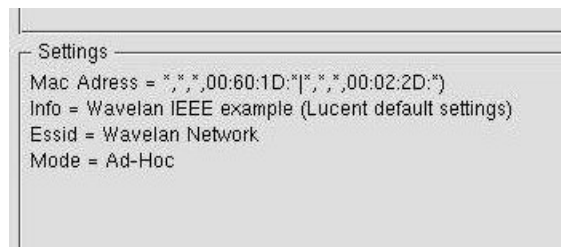


Figura 4.8 – Informações de drivers no campo *Settings*

Nesta parte é exibido o nome do campo (Ex.: Mac Adress) e o valor (Ex.: *,*,*,00:60...) que estão preenchidos no arquivo wireless.opts.

O botão *Ok* e o botão *Help* finalizam o formulário e chama a ajuda para esta janela respectivamente.

Análise do Código – displayform.ui.h (Anexo G)

Classe : displayForm

O código do formulário de exibição dos driver contido no arquivo displayForm.ui.h que pode ser visto no anexo II é formado pelas funções abaixo e trabalha basicamente com os objetos das classes clWireless e clAdapter.

```
void displayForm::init()
```

Construtor do formulário que inicia as variáveis globais.

```
void displayForm::newDriver()
```

Executa o formulário de edição e adição para a inserção de um novo driver no arquivo wireless.opts.

O algoritmo desta função está apresentado abaixo:

- Cria-se um objeto do formulário de edição e adição de drivers (addEditForm *addeditform = new addEditForm;);
- O formulário de edição e adição de drivers realiza a inserção do novo driver no arquivo.

```
void displayForm::editDriver()
```

Executa o formulário de edição de drivers ativando as opções neste formulário para que possam ser editadas e salvas no arquivo wireless.opts.

O algoritmo desta função está apresentado abaixo:

- Apartir do driver selecionado na caixa de seleção;
- Abrir o formulário de edição de drivers;
- Ativar as opções de informações deste driver já contida no arquivo utilizando as informações contidas nos objetos das classes clWireless e clAdapter;
- As alterações serão salvas pelo formulário de edição de drivers.

```
void displayForm::deleteDriver()
```

Remove o driver que estiver selecionado na caixa de seleção.

O algoritmo de remoção esta abaixo exibido:

- Apartir do driver selecionado na caixa de seleção;
- Exibir a mensagem para confirmar se deseja mesmo remover o driver.
- Caso confirme remover o driver usando a função `removeAdapter()` da classe `clWireless` para remover o driver;

```
void displayForm::selectionDriver()
```

Exibe o driver selecionado na caixa de seleção. É ativada a qualquer alteração na caixa de seleção.

O algoritmo de seleção de drivers está exibido abaixo:

- Apartir do driver selecionado na caixa de seleção;
- Exibir suas informações no campo *Settings* apartir das informações contidas no objeto que contém todos os campos usados no arquivo para este driver;

A remoção e seleção são as únicas funções realizadas diretamente nestes arquivos. As de adição e edição dos drivers são feitas no formulário da classe `addEditForm`.

Outras informações mais detalhadas deste código estão no anexo II.

4.1.3 Interface de edição, inserção e remoção de drivers

O formulário de edição, inserção e remoção de drivers apresentado nas figuras 4.9, 4.10, 4.11 e 4.12 são responsável pela edição dos campos de informações dos drivers contidos no arquivo `wireless.opts`, a inserção de novos drivers e a remoção dos drivers.

Este formulário é formado por quatro orelhas (*Configuration*, *Configuration 1*, *Configuration 2*, *Configuration 3*) que contem os campos para configuração do respectivo driver.

Na orelha *Configuration* apresentado na figura 4.9 estão os campos *Mac Adress* que informa o *Mac Adress* do driver a ser editado. O campo de informações sobre o driver (*informtation*) e o campo *comment* que é usado exclusivamente se o formulário foi chamado para a inserção de um novo driver. Este comentário será inserido no arquivo `wireless.opts` como uma linha comentada(#). Quando o formulário foi chamado para edição este campo não será ativado.

O campo *Mac Adress* é um campo obrigatório, pois é necessário para o reconhecimento do driver instalado no computador. O *Mac Adress* deve ser digitado exatamente igual ao exemplo ao lado do nome dele. Inicia com a string **,*,** seguida pelos três dígitos hexadecimais do driver (definido pela fábrica) e no final o parênteses). No caso de o driver ter mais de um *Mac Adress*, separa-los pelo pine l.

O campo *Information* também é obrigatório, pois ele será usado na caixa de seleção do formulário de exibição de drivers.



Figura 4.9 – Janela de edição e inserção de driver (*configuration*)

Na orelha *Configuration 1* apresentado na figura 4.10 estão os campos *Essid* (*Extended Network Name*), *Nwid*, *Bit Rate* e *Channel*.

O campo *Essid* é responsável pela configuração do nome da rede wireless definida pelo cliente a ser usada na conexão.

O campo *Nwid* é responsável pela configuração do domínio ou ID da rede para que se possa se distinguir logicamente diferentes redes. O *NetWork Identification Designator (NWID)*, é um código de quatro dígitos hexadecimal, que 'conecta' logicamente um número de estações wireless entre um *wireless workgroup* (célula).

O campo *Bit Rate* é responsável pela configuração da velocidade com que cada bit será transmitido na camada física.

O campo *Channel* define o modo da parcela do spectrum de rádio limitado para um uso particular que será usada para conexão. São usados 11 canais definidos na caixa de seleção onde cada um destes canais definem um frequência entre 2.4 to 2.48 GHz .

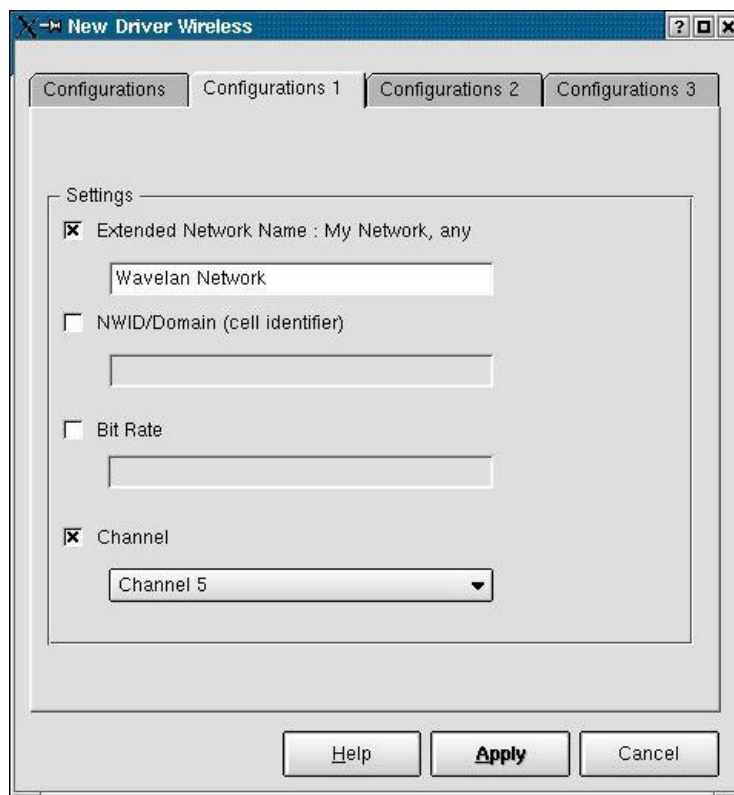


Figura 4.10 – Janela de edição e inserção de driver (*configuration 1*)

Cada campo que o cliente desejar configurar deverá ser primeiramente ativado usando as caixas de confirmação e assim este

campo será liberado para edição. Caso nada seja inserido no campo quando ativado, será inserido no arquivo apenas a informação vazia entre aspas. Ex.: `NWID=""`.

Devido a grande diversidade dos atributos que podem ser inseridos com exceção do campo *Channel* os demais campos foram configurados para aceitar o que o usuário digitar sendo de responsabilidade deste a inserção dos dados corretamente e de valores reais para o funcionamento correto da conexão.

Na orelha *Configuration 2* apresentado na figura 4.11 estão os campos de configuração *Mode*, *Rts*, *Sens* e *Channel*.

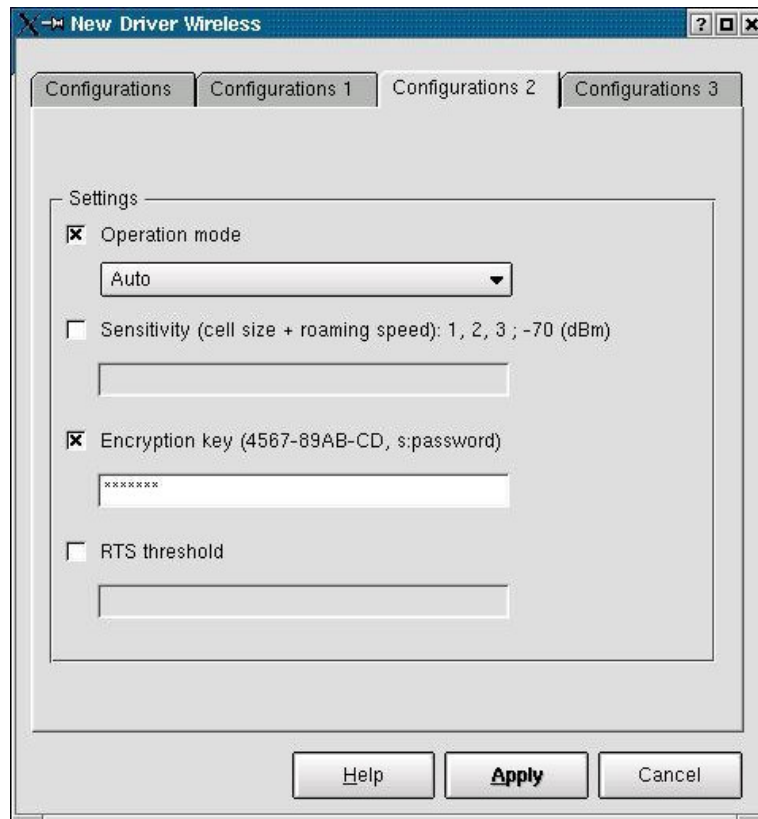


Figura 4.11 – Janela de edição e inserção de driver (*configuration 2*)

O campo *RTS threshold* é responsável pela definição no uso de *handshaking* na transmissão.

A caixa de seleção *Operation Mode* define os seis tipos de conexão: Ad-Hoc, Managed, Master, Repeater, Secondary e auto.

O campo *Sensitivity (cell size + roaming speed)* define o valor do nível do sinal para iniciar a recepção de pacotes (sensibilidade).

Também como foi definida para as outras configurações, devido a grande diversidade dos atributos que podem ser inseridos com exceção do campo Mode os demais campos foram configurados para aceitar o que o usuário digitar sendo de responsabilidade deste a inserção dos dados corretamente e de valores reais para o funcionamento correto da conexão.

Na ultima orelha da interface (*Configuration 3*) apresentado na figura 4.12 estão os campos de configuração *Frag*, *Iwconfig*, *Iwspy* e *Iwpriv*

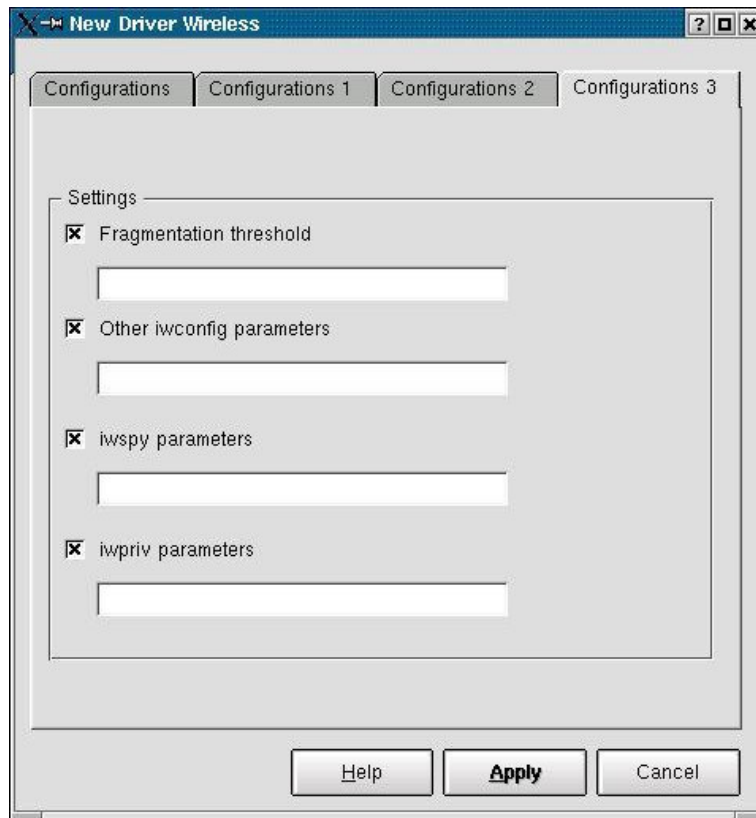


Figura 4.12 - - Janela de edição e inserção de *driver (configuration 3)*

O campo *Fragmentation threshold* define o tamanho dos pacotes que serão transmitidos.

O campo *Iwconfig Parameters* define os valores de configuração

O campo *Iwspy Parameters* é responsável pela definição de alguns parâmetros de funcionalidade extra.

O campo *Iwpriv Parameters* define o valor do endereço IP assim como endereços de máquina (MAC).

Todas as configurações alteradas deverão ser salvas clicando no botão *Apply*. Este botão tem a função de salvar as alterações nos campos dentro do arquivo *wireless.opts*.

Este formulário é ativado tanto quando for usado para inserir um novo driver como para editar um dos drivers já existentes neste arquivo. A única diferença é que quando for optado para editar um driver, os campos com valores no arquivo serão ativados. Caso deseje-se inserir novos campos nos drivers já existentes basta ativa-los.

Abaixo está o estudo do código deste formulário que está anexado no Anexo II e está escrito no arquivo *addeditform.ui.h*.

Análise do Código – arquivo *addeditform.ui.h* (Anexo H)

Classe : `addEditForm`

`void addEditForm::init()`

Construtor do formulário que inicia as variáveis globais para serem usadas no programa.

`void addEditForm::apply()`

Função ligada ao botão *apply* e é responsável pela aplicação das mudanças nos campos do arquivo *wireless.opts* a partir das alterações feitas no formulário.

O algoritmo desta função está apresentado abaixo:

- Apartir da opção escolhida (edição ou inserção);
- Caso edição `if(edit_adapt);`
- Capturar todas informações do driver selecionado no formulário *displayform* contido no arquivo *wireless.opts* e inserir-las nos seus respectivos campos na tela e ativando-os para edição;

- Salvar o driver no arquivo wireless.opts (wireless.saveAlterations()) caso o campo de *Mac adress* e *Infomation* tenha sido corretamente inserido;
- Caso adição;
- Salvar o driver no arquivo wireless.opts (wireless.saveAlterations()) caso o campo de *Mac adress* e *Infomation* tenha sido corretamente inserido;
- Caso os campos *Mac adress* e *Information* não tenha sido corretamente inserido exibir a mensagem da figura 5.13.

```
void addeditForm1::informeOperation( bool edit , int pos)
```

Esta função determina se a operação utilizada é de edição ou adição. Se ela for usada ativa a variável booleana `edit_adapt` que indica que o formulário está sendo usado para editar um driver já existente no arquivo `wireless.opts`. Ele também informa qual adaptador deverá ser alterado (`pos_adapt`).

```
bool newForm1::insertedNewDriver()
```

Esta função retorna o valor da variável booleana `newDriver` que determina se o novo driver foi inserido. O resultado será utilizado para a atualização da caixa de seleção de drivers.

```
void newForm1::alterNewDriver()
```

Esta função tem a finalidade apenas de atualizar a variável `newDriver`.

4.1.4 Interface de Configuração do Cliente e da Rede

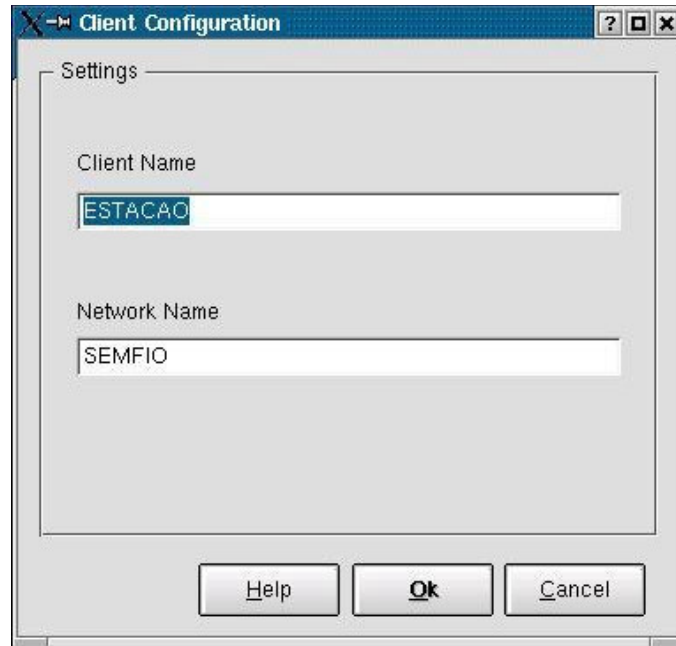


Figura 4.13 – Janela Client Configuration

Esta janela apresentada na figura 4.13 é responsável pela edição do nome do cliente ou nome do computador (campo Client Name) que esta usando a conexão e o nome da rede wireless (Wireless Network Name) a qual este computador deverá se conectar.

O nome do cliente (Client Name) pode se qualquer um, desde que não esteja sendo usado por outro cliente. Este nome será inserido no arquivo dentro do campo `INFO` do código `essidany,*,*,*`.

O nome da rede (Wireless Network Name) deve ser o nome que foi definido para rede wireless a qual o driver deverá ser conectado, e este nome será inserido no campo `ESSID` que está contido também no código `essidany,*,*,*`. O Network Name é usado para conexão logica a estação wireless. O Network Name é um nome compartilhado que combina múltiplas células wireless. Este nome é uma string alfanumérica nos valores ' α ', ' A ' e ' 0 ' com no máximo de 32 caracteres.

Nota : O Network Name é case-sensitive.

Análise do Código – arquivo clientform.ui.h (Anexo I)

O código deste formulário está contido no anexo II e é o arquivo clientconfform.ui.h. Esta classe é formada pela funções abaixo listadas e ele trabalha com os atributos `info` e `essid` contidos no objeto wireless da classe `clWireless` criado a partir do arquivo wireless.opts.

void clientForm::saveConf()

Salva as alterações criadas pelo cliente no arquivo wireless.opts referentes ao nome do cliente e ao nome da rede wireless.

O algoritmo desta função pode ser definido assim:

- Apartir dos nomes editados nas caixas de textos;
- Criar o objeto wireless da classe `clWireless` e caso já exista algum nome nos campos INFO e ESSID inseri-los respectivamente nos campos de Client Name e Network Name;
- Salva as alterações no arquivo wireless.opts;

4.2 Interface de Análise da Conexão da Rede Wireless LAN

Agora vamos estudar a interface de análise da conexão utilizada pelo cliente Linux em seu computador. Esta interface fornece uma análise gráfica e numérica sobre a conexão no momento atual para que possa avaliar a situação da transmissão.

Nas Figuras 4.14, 4.15, 4.16, 4.17 e 4.18 é apresentado a interface de análise da conexão. Ela é formada por quatro partições.

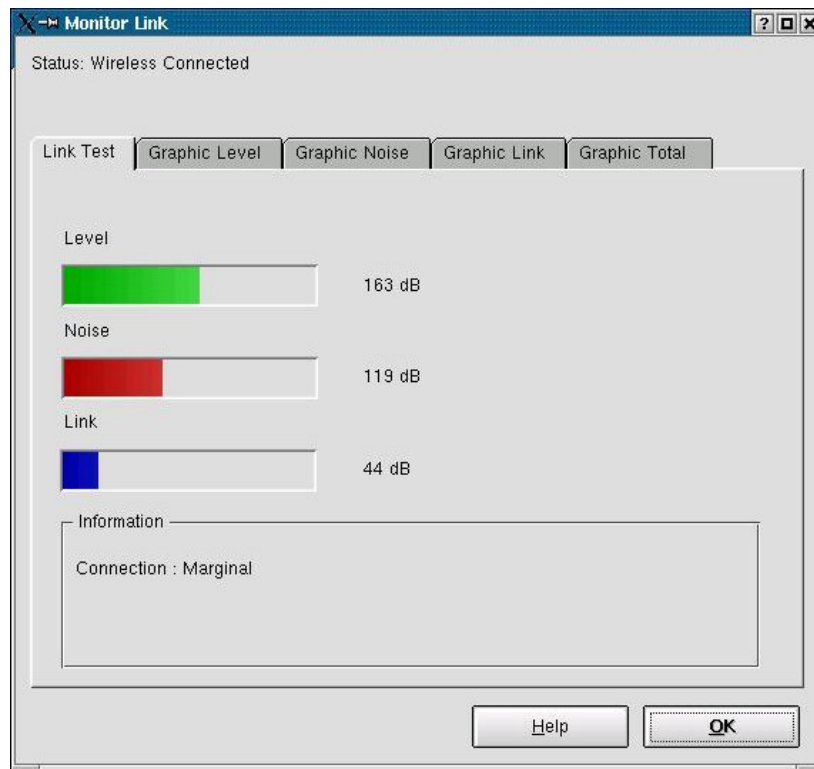


Figura 4.14 – Interface de Análise (*Link Test*)

A primeira partição apresentada na figura 4.14 exhibe a conexão em barras de progresso onde é mostrado o valor do *link*, *noise* e *level* lido a cada segundo no arquivo wireless analisado no capítulo 4. Os valores atualizados no arquivo são lidos pelo classe *cLink* e exibidos nas barras e nos valores aos lados. Este valor está em dB (decibéis) e está definido por um valor entre 0 dB e 300 dB.

Uma conexão estará numa condição boa quando a diferença entre o *level* (força da transmissão) e o *noise* (ruído) for num valor maior do que 30 dB. Este valor representa o valor do *Link*. Um valor entre 20 dB e 29 dB indica que a conexão está *marginal* e um valor menor do que 20 dB indica que o link está ruim. O valor de 0 dB indica que não a conexão com nenhuma rede wireless.

No campo *information* e apresentado a situação da conexão (*bad,reasonable,good*) e outras informações referente ao campo *Discarded packets(nwid, crypt, frag)*.

A Segunda partição (*Level Graphic*) apresentada na figura 4.15 refere-se exclusivamente ao nível da transmissão e é apresentado graficamente no gráfico level x tempo. O tempo é de 1 minuto para se completar um gráfico. Depois disso é reiniciado o gráfico. O valor do link é representado pela linha verde. Ao lado em *information* é exibido o valor do ponto atual do *level* do gráfico.

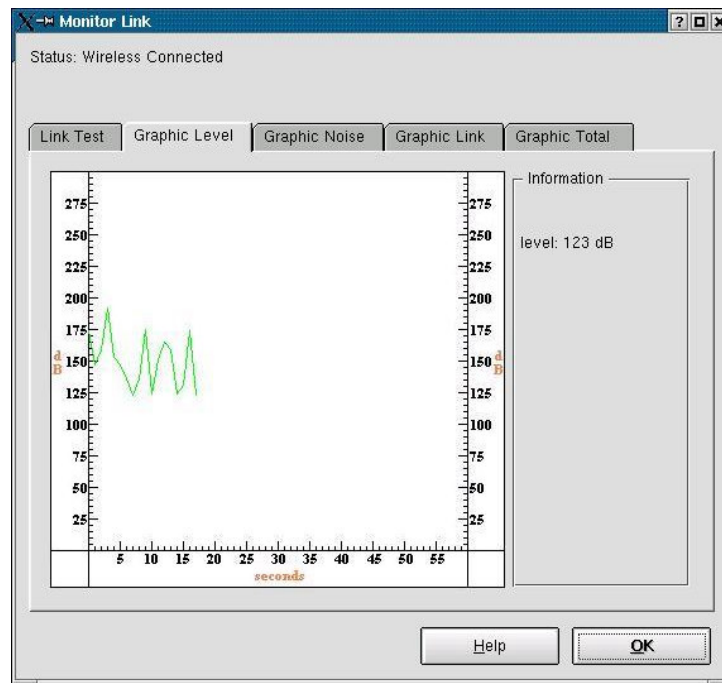


Figura 4.15 – Interface de Análise (*Level Graphic*)

A terceira partição (*Noise Graphic*) apresentada na figura 4.16 refere-se exclusivamente ao nível da interferência e é apresentado graficamente no gráfico noise x tempo. O tempo é de 1 minuto para se completar um gráfico. Depois disso é reiniciado o gráfico. O valor do *noise* é representado pela linha vermelha. Ao lado em *information* é exibido o valor do ponto atual do *noise* do gráfico.

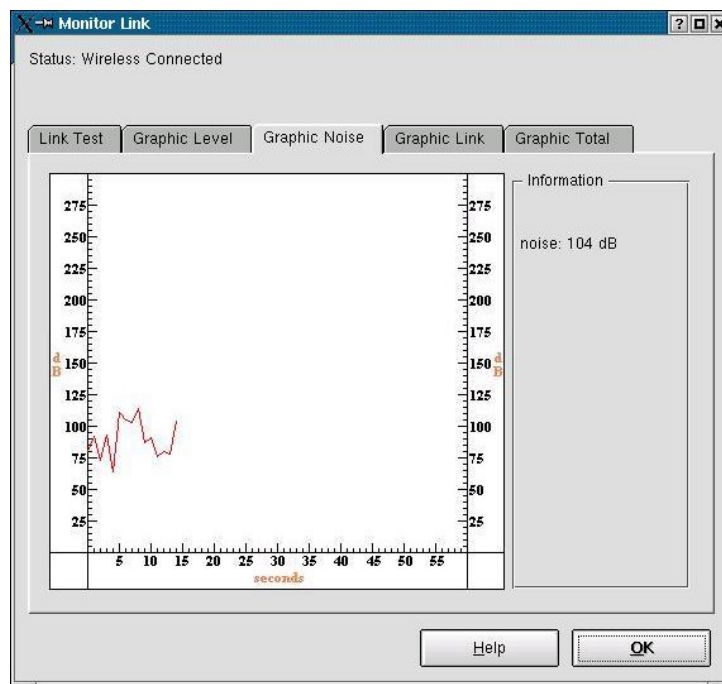


Figura 4.16 – Interface de Análise (*Noise Graphic*)

A quarta partição (*Link Graphic*) apresentada na figura 4.17 refere-se exclusivamente ao link da conexão e é apresentado graficamente no gráfico link x tempo. O tempo é de 1 minuto para se completar um gráfico. Depois disso é reiniciado o gráfico. O valor do *link* é representado pela linha azul. Ao lado em *information* é exibido o valor do ponto atual do *link* do gráfico.

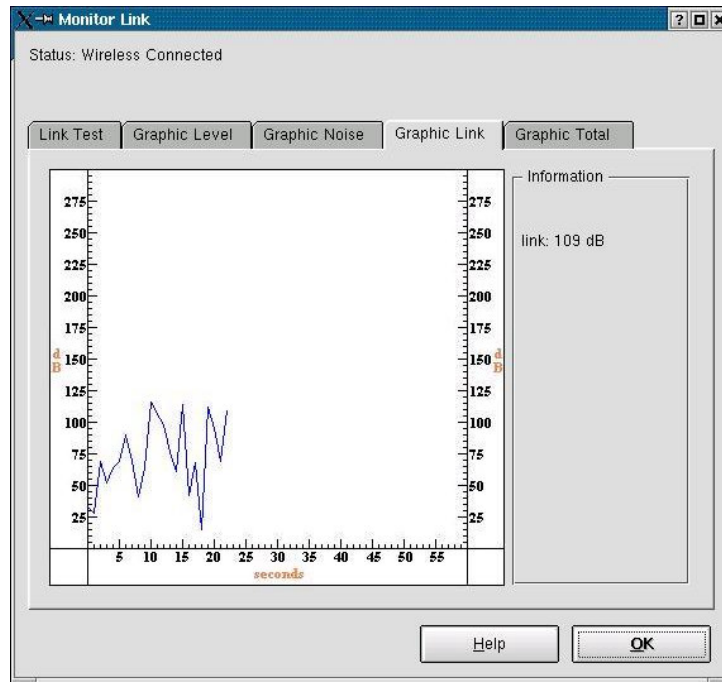


Figura 4.17 – Interface de Análise (*Link Graphic*)

A última partição (*Total Graphic*) apresentada na figura 4.18 apresenta as os três valores graficamente no gráfico (*level*, *noise* e *link*) x tempo. O tempo é de 1 minuto para se completar um gráfico. Depois disso é reiniciado o gráfico. O valor do *link* é representado pela linha azul, o do *level* pela cor verde e o do *noise* pela cor vermelha. Ao lado em *information* é exibido o valor do ponto atual do *link*, *level* e *noise* do gráfico.

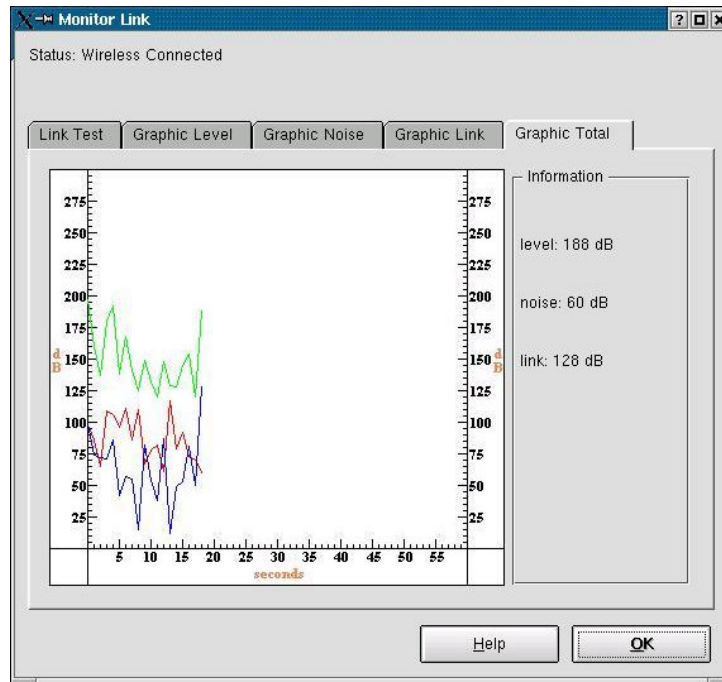


Figura 4.18 – Interface de Análise (*Total Graphic*)

Análise do Código – arquivo monitorform.ui.h (Anexo J)

O código desta interface está no arquivo *monitorform.ui.h* apresentado no anexo II. O código trabalha com a classe *cLink* que lê as informações contidas no arquivo *wireless*. A cada segundo o formulário pega os dados contidos no arquivo e os imprime nos formatos apresentados acima (*gráfico, barra de progresso...*).

```
void monitorForm::init()
```

Este é o construtor da classe e nele é inicializado os valores globais a serem usados na classe. A linha de código `startTimer(1000)`; determina o tempo de 1 segundo para que a cada este tempo seja feita a leitura do arquivo *wireless* pela função `void monitorForm::timerEvent(QTimerEvent *e)`.

```
void monitorForm::timerEvent( QTimerEvent *e )
```

Esta função é responsável pela leitura do arquivo e a exibição dos dados avaliados na janela *Monitor Link*. O algoritmo desta função pode ser representado assim:

- A cada segundo faça:
- Cria-se um objeto da classe `clLink` (`monitor`);
- Se houver conexão wireless;
- Exibe os valores nas barras de progresso referentes ao *link*, *level* e *noise*;
- Imprime os valores nos gráficos a partir da função `void monitorForm::print_graphic(int level, int noise, int link);`

```
void monitorForm::print_graphic(int level, int noise, int link)
```

Função que desenha as linhas dos gráficos referente ao *level*, *Noise* e *link*. O algoritmo desta função está na listagem abaixo:

- A partir dos valores passados por parâmetros (`int level, int noise, int link`)
- Imprimir nos gráficos os valores depois de definido as cores de cada linha.
- Caso tenha sido terminado o tempo de 1 minuto atualizar os gráficos.

Capítulo 5

Conclusões e Propostas Futuras

5.1 Conclusões

Este sistema foi idealizado com o objetivo de monitorar a conexão wireless para clientes linux existente no Centro de Informática da Universidade Federal de Lavras – MG. Visando facilitar para o usuário linux, o sistema atendeu os requisitos necessários para uma rápida configuração dos drivers de suporte e para também uma análise rápida de como está a conexão com a estação wireless.

O sistema naquilo que foi a principio objetivado, conseguiu atender de uma forma satisfatória os seus requisitos.

O uso de conexão wireless tem crescido grandemente e futuramente devido ser muito mais prático sua instalação do que o uso dos cabos que dão muito trabalho para a instalação. Por isso visamos criar este sistema que facilite ao usuário sua configuração e também a análise da conexão a uma estação wireless.

5.2 Propostas Futuras

Este sistema está em sua primeira versão sendo sua função como foi visto a configuração dos drivers de suporte para redes Wireless LAN e para análise da conexão. Uma melhoria que deve ser feita é em relação aos

Campos de configuração que neste programa estão abertos a qualquer inserção de parâmetros e então definir proteções usando máscaras para garantir a inserção de dados válidos para que a conexão possa ocorrer corretamente.

Em termos de aumentar a configuração de outros atributos de uma rede wireless as propostas são a de configurar mais arquivos que estão relacionados com redes wireless usando ambiente gráfico para facilitar a configuração do cliente.

Em termos de análise da conexão de um cliente linux pode ser criado mais análise em relação a pacotes enviados, recebidos ou perdidos na conexão.

Outros projetos poderão ser implantados ao sistema para que sejam criadas novas versões e assim aumentado o campo de configuração e análise da rede wireless LAN.

Referências Bibliográficas

[Lima00] LIMA, Abiel R; et. Al. Apostila Wireless. Curso PCT Motorola. Florianópolis. 2000.

[Alencar98] ALENCAR, Marcelo Sampaio de Alencar. “Telefonia Digital”. São Paulo, Érica, 1998.

[Camara00] CÂMARA, Daniel Câmara. “Proposta para Cobertura de Área de Sombra em Redes Wireless”, UFMG, 2000 .

[IEEE802.11a] IEEE Doc. IEEE P802.11-96/49C. ‘802.11 Tutorial – 802.11 MAC Entity: MAC Basic Access Mechanism Privacy and Access Control’. U.S.A., 1996.

[IEEE802.11b] IEEE Standard 802.11. “The IEEE 802.11 Standard”. U.S.A., 1997.

[IEEE802.11c] IEEE Standard 802.11. “The IEEE 802.11b Standard”. U.S.A., 1998.

[Soares95] SOARES, L. F. G. Et al. – ‘Redes de Computadores – Das LANs, MANs e WANs às redes ATM’ - Ed. Campus – 1995.

- [Tenenbaum96] TENENBAUM, Adrew S. “Redes de Computadores”, Editora Campus, 1996.
- [Areias99], Gonçalo Menezes, Wireless IP, Universidade do Minho, 1999.
- [Arnett97], Matthew Flint; DULANEY, Emmett; HARPER, Eric et al. Desvendando o TCP/IP. Rio de Janeiro: Campus, 1997.
- [Duarte2001], Emeide Nóbrega; NEVES, Dulce Amélia de B.; SANTOS, Bernadete de L. O. dos. Manual técnico para realização de trabalhos monográficos: dissertações e teses. 4 ed. João Pessoa: Ed. Universitária da UFPB, 2001. 96 p.
- [França98], Júnia Lessa et al. Manual para normalização de publicações técnico-científicas. 4 ed. Belo Horizonte: Ed. UFMG, 1998.
- [Johnon98], David B., Maltz David A., Dynamic Source Routing in Ad Hoc Wireless Networks, 1996.
- [Mateus98], Geraldo Robson, Loureiro, Antonio Alfredo, Introdução à Computação Móvel, 11^a Escola de Computação, Rio de Janeiro, 1998.
- [Mathias2000], André Pimenta, “IEEE 802.11 – Redes Sem Fio”, UFRJ, 2000.
- [Menezes2000], Rodrigo Saldanha, “IEEE 802.11 – Wireless”, UFRJ, 2000.
- [Lucent2000], Technologies Inc., “ORINOCO Manager Suíte User’s Guide”, 2000
- [Tourrilhes],Jean, Wireless Extensions, http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html .
- [Tourrilhes97], Jean, “Wireless Extensions for Linux”, 1997
- [Tourrilhes2000], Jean, “Linux Wireless LAN Howto”, 2000

[Tourrilhes2002], Jean, ‘Linux Wireless LAN Howto’, 2002

[Valley2002] Linux, *Linux Documentation Project – Man Page - Wavelan2_cs*, Available from http://www.linuxvalley.it/encyclopedia/ldp/manpage/man4/wavelan2_cs.4.php Nov 2002.

Anexos

Anexo A

Arquivo Wireless.opts

Anexo B

Arquivo Wireless

Anexo C

Arquivo clWireless.h

Anexo D

Arquivo clAdapter.h

Anexo E

Arquivo clLink.h

Anexo F

Arquivo wireless.ui.h

Anexo G

Arquivo displayform.ui.h

Anexo H

Arquivo addeditform.ui.h

Anexo I

Arquivo clientform.ui.h

Anexo J

Arquivo monitorform.ui.h

Anexo L

Arquivo main.cpp