FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
mantenedora do CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA – UNIVEM
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ADRIANO DOS SANTOS LUCAS

AVALIAÇÃO DE DESEMPENHO DO PROTOCOLO WEP EM REDES SEM FIO AD HOC USANDO UM SIMULADOR DE REDES

MARÍLIA
2006

ADRIANO DOS SANTOS LUCAS

AVALIAÇÃO DE DESEMPENHO DO PROTOCOLO WEP EM REDES SEM FIO AD HOC USANDO UM SIMULADOR DE REDES

Monografia apresentada ao Curso de Graduação em Ciência da Computação, do Centro Universitário Eurípides de Marília, mantido pela Fundação de Ensino Eurípides Soares da Rocha, como requisito para obtenção do grau de Bacharel em Ciência da Computação.

Orientadora:
Profa. Dra. Kalinka R. L. J. Castelo Branco

MARÍLIA
2006

ADRIANO DOS SANTOS LUCAS

AVALIAÇÃO DE DESEMPENHO DO PROTOCOLO WEP EM REDES SEM FIO AD HOC USANDO UM SIMULADOR DE REDES

Banca examinadora do Trabalho de Conclusão de Curso apresentado à UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação. Área de concentração: Arquitetura de computadores.

Resultado: _____

_____

ORIENTADORA: Profa. Dra. Kalinka Regina Lucas Jaquie Castelo Branco.

_____

1º EXAMINADOR: André Luiz Satoshi Kawamoto

_____

2º EXAMINADOR: José Remo Ferreira Brega

Marília, 06 de dezembro de 2006.

Dedico este trabalho primeiramente a Deus, por ele ter me dado à benção da vida, ter me dado sabedoria e por estar comigo em todos os momentos.

Ao meu pai Silas de Oliveira Lucas e minha mãe Dolores dos Santos Lucas, que sempre me apoiaram e não mediram esforços para que eu pudesse me formar. Amo vocês.

Ao meu irmão Adauto dos Santos Lucas que mesmo longe sempre deu uma força. A minha irmã Andréia dos Santos Lucas que sempre que eu pedia um favor, uma tradução, mesmo com seus afazeres me dava atenção e sua ajuda.

ÓRGÃO FINANCIADOR

Fundação de Amparo à Pesquisa do Estado de São Paulo

AGRADECIMENTOS

Quero agradecer a Deus por ter me abençoado, e ter me dado à oportunidade de concluir a faculdade, mesmo com as dificuldades, sempre esteve comigo.

Aos meus pais, Silas e Dolores pelo apoio que me deram durante todo o curso. Confiaram em meu potencial e fizeram todo o esforço necessário para que eu pudesse concluir a faculdade. Muito obrigado, amo vocês.

Ao meu irmão Adauto e minha irmã Andréia que me ajudaram muito, e tiveram paciência quando tinha que passar o dia e a noite no computador.

A minha namorada Renata Bortolotti pela paciência nesses 4 anos que por vezes dediquei mais tempo aos estudos, pela sua compreensão da importância de eu me formar, pelo carinho e apoio de sempre. Te amo.

À minha orientadora Kalinka Regina Lucas Jaquie Castelo Branco, pela confiança concedida, paciência, compreensão e por sempre me apoiar mesmo nas situações adversas.

Aos meus queridos amigos da 1ª Igreja Batista de Marília, pelas orações feitas nas células de quarta-feira. Deus abençoe muito a todos.

Aos meus colegas de classe, que sempre me apoiaram em tudo e deram vários conselhos. Em especial quero agradecer aos amigos Rodrigo, Mariana, Danilo, Bárbara e Ana que passei dois anos convivendo intensamente.

À FAPESP, pelo apoio financeiro.

A todos os excelentes professores de Ciência da Computação do UNIVEM que me passaram o conhecimento necessário para que eu pudesse me formar.

Enfim, a todos os meus familiares e amigos que contribuíram para que eu chegasse até aqui, agradeço de coração.

"Bendito o homem que confia no Senhor, e cuja esperança é o Senhor".
A Bíblia Sagrada, em Jeremias 7:7

LUCAS, Adriano dos Santos. Avaliação de desempenho do protocolo WEP em redes sem fio *Ad Hoc* usando um simulador de redes. 2006. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação). Curso de Ciência da Computação, Fundação de Ensino "Eurípides Soares da Rocha", mantenedora do Centro Universitário de Marília – Univem.

RESUMO

As tecnologias de redes de computadores têm sofrido um grande crescimento nos últimos anos. Segundo Torres (2001), "As redes de computadores surgiram da necessidade da troca de informações, onde é possível ter acesso a um dado que está fisicamente localizado distante de você". As redes convencionais apresentam uma etapa complicada que é a passagem dos cabos para ser montada, exigindo-se às vezes obras civis, ou deixando tubulações e canaletas à mostra, além disso, essas redes convencionais restringem o movimento dos equipamentos. A dificuldade em manusear o cabeamento convencional somada à necessidade de mobilidade e a fácil instalação, vem aumentando o uso das redes sem fio e tornando-as um sucesso (ZANNETI, GONÇALVES, 2006). Embora esse tipo de rede seja muito conveniente, existem algumas vulnerabilidades colocando em risco a confidencialidade, integridade, autenticidade, disponibilidades da comunicação entre outros, que devem ser levadas em consideração pelos seus usuários (MAIA, 2006). O protocolo de segurança presente na maioria dessas redes sem fio é o WEP (*Wired Equivalent Privacy*) (MAIA, 2006), e é este protocolo que será estudado nesse trabalho, com o uso da ferramenta *Network Simulator* (NS) (VINT, 2006), um simulador que oferece suporte a diversas tecnologias de rede. Porém o *Network Simulator* não fornece nenhum suporte para simulação de segurança em redes sem fio, sendo necessário o desenvolvimento de um novo agente. Assim, será realizado nesse trabalho a inserção de um novo agente no simulador contendo características do protocolo WEP para permitir que sejam proferidas simulações baseadas na segurança em redes sem fio.

Palavras chave: Redes de computadores, Redes sem fio, protocolo WEP, *Network Simulator*, topologia *ad-hoc*.

LUCAS, Adriano dos Santos. Avaliação de desempenho do protocolo WEP em redes sem fio Ad Hoc usando um simulador de redes. 2006. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação). Curso de Ciência da Computação, Fundação de Ensino "Eurípides Soares da Rocha", mantenedora do Centro Universitário de Marília – Univem.

Abstract

The computer net technologies have been growing in the last years. According to Torres (2001), "The computer nets had its origin from the necessity of information exchange, and it makes possible to access an information which is physically located far from you". The ordinary nets present a complicated step of assembling the cables, which sometimes may demand a construction work, or may involve tubings and besides the fact that these ordinary nets restrict the movements of the equipments. The difficulties in handling the ordinary cabling and the necessity of mobility and easy installation cause the progress of the wireless nets making it a success (ZANNETI, GONÇALVES, 2006). Even though this kind of net is very convenient, there are some weaknesses that can put the confidentiality, the integrity, the authenticity and the availability of communication in risk. And this should be considered by the users (MAIA, 2006).The security protocol presented in most of the wireless nets is the WEP (Wired Equivalent Privacy) (MAIA, 2006), which is going to be analyzed in this project through the Network Simulator (NS) (VINT, 2006). It is a simulator that offers support to many net technologies. It is a simulator that offers support to many net technologies. However the Network Simulator does not supply to no support simulation of security in wireless network, being necessary the development of a new agent. Thus, will make in this project the insertion of a new agent in the simulator contends characteristic of protocol WEP to allow that simulations based on the wireless network security are pronounced.

Key words: Computer Nets, wireless net, WEP protocol, *Network Simulator*, topology *ad-hoc.*

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ACK: Acknowledgment

AES: Advanced Encryption Standard

AODV: Ad-hoc On Demand Distance Vector

AP: Acess Point

ASCII: American Standard Code for Information Interchange

BSA: Basic Service Area

BSS: Basic Service Set

CBR: Constant Bit Rate

CCNA: Cisco Certified Network Administrator

CCNP: Cisco Certified Network Professional

CISCO: Corps Information Systems Control Officer

CSMA: Carrier Sense Multiple Access

CSMA/CA: Carrier Sense Multiple Access With Collision Avoidance

CSMA/ CD: Carrier Sense Multiple Access with Collision Detection

CTS: Clear to Send

DARPA: Defense Advanced Research Projects Agency

DCF: Distributed Coordination Function

DNS: Domain Name System

DoD: Departamento de Defesa Americano

DS: Distribution System

DSSS: Direct Sequence Spread Spectrum

EAP: Extensible Authentication Protocol

EBCDIC: Extended Binary Coded Decimal Interchange Code

ESA: Extended Service Area

ESM: Estação de Suporte à Mobilidade

ESS: Extended Service Set

FHSS: Frequency Hopping Spread Spectrum

FTP: File Transfer Protocol

GHz: Gigahertz

GLOMOSIM: Global Mobile Information Systems Simulation Library

GPRS: General Packet Radio Service

HR-DSSS: High Rate Direct Sequence Spread Spectrum

HTTP: Hypertext Transport Protocol

IBM: International Business Machines

IBSS: Independent Basic Service Set

ICV: Integrity Check Calue

IEEE: Institute of Electrical and Electronics Engineers

IP: Internet Protocol

ISM: Industrial, Scientific and Medical

ISO: International Standards Organization

LAN: Local Area Network

LLC: Link Layer Control

MACAW: Multiple Access with Collision Avoidance for Wireless

MAC: Media Access Control

MAN: Metropolitan Area Network

Mbps: Megabits Per Second

MIF: Maker Interchange Format

MIT: Massachusetts Institute of Technology

NCP: Network Control Program

NEST: Network Simulation Testbed

NAM: Network Animator

NS-2 : Network Simulator

NSF: National Science Foundation

np-CSMA: non-persistent Carrier Sense Multiple Access

OFDM: Orthogonal Frequency Division Multiplexing

OMNet++: Objective Modular Network Testbed in C++

OSI: Reference Model for Open System Interconnection

p-CSMA: p-persistent Carrier Sense Multiple Access

PARC: Palo Alto Research Center

PCF: Point Coordination Function

PCI: Protocol Control Information

PDU: Protocol Data Unit

PRNG: Pseudo Random Number Generators

RSA: Rivest Shamir Adleman

RTS: Request to Sendo

SDU: Service Data Unit

SMTP: Simple Mail Transfer Protocol

SNMP: Simple Network Management Protocol

SSID: Service Set Identifier Differentiates

STA: Station

TCP: Transmission Control Protocol

TCP/IP: Transmission Control Protocol/ Internet Protocol

TDMA: Time Division Multiple Access

TKIP: Temporal Key Integrity Protocol

UDP: User Datagram Protocol

VBR: Variable Bit Rate

VPN: Virtual Private Network

WAN: Wide Area Network

WEP: Wired Equivalent Privacy

WFQ: Weight Fair Queueing

WI-FI: Wireless Fidelity

WWW: World Wide Web

SUMÁRIO

INTRODUÇÃO

A comunicação sempre foi uma das maiores necessidades do ser humano. As redes de computadores têm por objetivo compartilhar informações, pois se vive em um mundo informatizado onde o tempo tem um preço elevado.

As redes de computadores vêm para disponibilizar informações que estão distantes trazendo benefícios a todos. É praticamente impossível hoje em dia não pensar em redes quando o assunto é informática. Segundo Tanenbaum (1997), o segmento de mercado que mais cresce na informática é o dos computadores móveis, como os *notebooks* e os *palmtops*. Com esse crescimento da computação móvel e a necessidade de transmissão de dados interligando redes onde não é possível a passagem de cabos, a tecnologia de redes sem fio está se fixando cada vez mais.

As redes locais sem fio (*Wireless Lan*) já são uma realidade em vários ambientes de redes, principalmente nos que requerem mobilidade dos usuários. Essas redes sem fio têm proporcionado uma melhora no acesso às informações, em rapidez e mobilidade. Basicamente existem dois tipos de redes móveis sem fio: a infra-estruturada e as redes *ad hoc* (MATHIAS, 2006).

Neste projeto serão trabalhadas as redes com a topologia *ad hoc*. Neste tipo de topologia os dispositivos são capazes de trocar informações diretamente entre si, não precisando de um Ponto de Acesso (PA) ligado em uma rede fixa como na topologia infra-estruturada.

Um problema inerente às redes *wireless* tem sido a segurança, devido à utilização de ondas de rádio ao invés de cabos, o que facilita o acesso às informações transmitidas. Com objetivo de solucionar esse problema, diversos protocolos de segurança surgiram, sendo um deles o WEP (*Wired Equivalent Privacy*) (DUARTE, 2003).

O WEP que utiliza a implementação do protocolo RC4 para realizar criptografia, está presente na maioria das redes sem fio. Este protocolo serve para impedir que intrusos consigam ler ou modificar os dados transmitidos além de impedir que tenham acesso à rede *wireless*, proporcionando confidencialidade, integridade dos dados e controle de acesso à rede *wireless* (MAIA, 2006). Entretanto, este protocolo não está atendendo ainda totalmente esses aspectos, necessitando ser ainda estudado, ter o seu desempenho analisado nas redes sem fio e, através da conclusão de sua análise, ser melhorado para atender cada vez mais os aspectos que o fazem ser um protocolo de segurança.

A ferramenta *Network Simulator* (VINT, 2006) que será utilizada não contém suporte para efetuar simulações onde se pretende analisar segurança. Por este motivo será realizada a implementação de um novo agente, para que posteriormente esse tipo de análise em relação à segurança possa ser analisado através de simulação. A vantagem de se utilizar um ambiente de simulação, é que estes permitem o estudo e avaliação de diversas tecnologias de redes com custo reduzido, desempenhando também um papel muito importante quando se pretende estudar cenários que muitas vezes são impossíveis de implementar na prática, ou quando se pretende avaliar o desempenho de um projeto inexistente.

O *Network Simulator* - NS-2 é um dos simuladores mais populares e permite criar quase todos os tipos de topologias de rede e analisar vários protocolos (VINT, 2006). Esta ferramenta tem como principais vantagens o fato de ser gratuita e possuir código aberto permitindo dessa maneira a adição de novos módulos ou a alteração dos existentes de acordo com as necessidades do usuário.

Além disso, o NS-2 dá suporte à simulação de diversas tecnologias de rede: redes baseadas nos protocolos TCP e UDP, redes locais, redes sem fio, satélite, *multicast*, entre outras.

Assim neste trabalho será inserido um novo agente, adequando características do protocolo WEP. Pretende-se tornar a ferramenta apta a avaliar segurança em redes, e assim obter resultados conclusivos a respeito do uso desse protocolo em redes *wireless* para propor melhorias que possibilitem o aumento do seu grau de segurança.

Este trabalho foi desenvolvido em seis capítulos, para uma melhor compreensão dos tópicos a serem desenvolvidos.

No primeiro capítulo é realizada uma introdução sobre redes de computadores convencionais contendo um breve histórico, definição, objetivos, classificação, topologias além de serem abordados os modelo OSI e o modelo TCP.

O segundo capítulo irá abordar as redes de computadores sem fio, desde o seu surgimento, a sua importância, vantagens e características, sendo muito relevante o entendimento dessa tecnologia, pois este trabalho irá usá-la para a implementação.

No terceiro capítulo o assunto é segurança para redes sem fio, onde um enfoque maior é dado ao protocolo WEP, assunto principal deste trabalho.

O quarto capítulo irá discutir alguns simuladores existentes e dará um destaque ao simulador *Network Simulator*, o qual foi usado neste trabalho para fazer a implementação.

O Capítulo 5 mostra os resultados obtidos, abordando o que foi preciso para realizar o trabalho e como este foi desenvolvido.

O Capítulo 6 descreve as conclusões obtidas nesse trabalho. Também aborda os trabalhos futuros e as publicações.

CAPÍTULO 1 - REDES DE COMPUTADORES

Redes de computadores são hoje uma realidade, e este capítulo tem por objetivo apresentar os aspectos gerais de redes de computadores convencionais e suas características.

1.1 - Considerações Iniciais

A comunicação sempre foi uma necessidade da sociedade, desde o início de sua existência, tendo o objetivo de aproximar comunidades. A troca de informações entre pessoas geograficamente dispersas, e a evolução dos sistemas de computação, migrando do centralizado para a distribuição do poder computacional, causou a necessidade de interligação dos computadores, fazendo surgir às redes de computadores.

Novas tecnologias e computadores interligados é uma realidade no contexto em que se vive atualmente, sendo muito relevante no meio empresarial. Conforme Tanenbaum (1994), hoje em dia os computadores, desde os pessoais aos supercomputadores, têm grande chance de fazer parte de uma rede do que de não estar em nenhuma.

Portanto neste capítulo será abordado um breve histórico das redes, conceitos sobre o que são as redes de computadores, topologias, os modelos de referência mais conhecidos OSI e TCP/IP.

1.2 - Histórico de redes de computadores convencionais

A comunicação sempre foi importante e desde a antiguidade vêm sendo desenvolvidas técnicas para suprir tais necessidades de comunicação, como toques de tambor, o uso de sinais de fumaça, pombos correio entre outras.

Uma nova época nas comunicações surgiu em 1838, com a invenção do telégrafo. Desde então, com a evolução dos sinais elétricos surgiu a maior parte dos sistemas de comunicação mais atuais como telefone, rádios, televisores entre outros (SOARES, 1995).

Na informática houve o desenvolvimento dos minicomputadores e dos microcomputadores, onde a utilização de redes de informação proporcionou melhorias na estruturação organizacional. Antes da evolução dos computadores, estes eram máquinas grandes e complexas, sendo operadas por pessoas especializadas, tendo o processamento em lote, por meio da leitura de cartões ou fitas magnéticas (SOARES et al, 1995).

Na década de 70 começou uma mudança nos sistemas de computação. De um sistema único centralizado e de grande porte, o qual servia todas as necessidades computacionais de uma organização, migrava-se para a distribuição do poder computacional, onde um número de computadores separados, mas interconectados, executavam a tarefa. Esses sistemas são chamados redes de computadores (TANENBAUM, 1994).

A descentralização do processamento permitiu o compartilhamento de recursos, como impressoras, meios de armazenamento de dados, *software* e ainda proporcionaram uma maior confiabilidade, modularidade do sistema, novos serviços dentre outras vantagens que facilitaram a comunicação entre as pessoas. Assim, a interconexão entre os vários sistemas para o uso compartilhado de dispositivos periféricos tornou-se importante (SOARES et al, 1995).

Além do compartilhamento de recursos, a capacidade de troca de informações também foi uma razão importante para a interconexão dos computadores. Usuários individuais de sistemas de computação não trabalham isolados e necessitam de alguns dos benefícios oferecidos por um sistema centralizado. Entre esses a capacidade de troca de mensagens entre os diversos usuários e a facilidade de acesso a dados e programas de várias fontes quando da preparação de um documento. Ambientes de trabalho cooperativos se

tornaram uma realidade tanto nas empresas como nas universidades, exigindo a interconexão dos equipamentos nessas organizações.

Mediante todos os fatos, tem-se a convicção que atualmente há uma necessidade muito grande pela conexão de computadores em diversos setores (SOARES et al, 1995).

1.3 - Definição de redes de computadores

Uma rede de computadores pode ser definida como um conjunto de computadores autônomos interconectados através de um sistema de comunicação (cabo, fibra óptica, luz) permitindo o compartilhamento de recursos computacionais e trocando informações via regras contidas em protocolos (TANENBAUM, 2003).A definição de redes de computadores é usada para sistemas de todos os tamanhos e tipo, desde a Internet até uma rede simples.

É preciso salientar que um sistema distribuído vai além de um sistema de redes de computadores. No sistema distribuído o usuário não percebe que há vários computadores interligados sendo que a distribuição de tarefas é realizada pelo sistema operacional (TANENBAUM, 1994). Para o usuário tudo parece como um sistema mono-processador virtual, sendo que o mesmo não percebe a existência de vários processadores. Assim, pode-se dizer que os sistemas distribuídos utilizam as redes de computadores através de um *software* que dá um alto grau de transparência ao usuário. Portanto a distinção entre uma rede de computadores e um sistema distribuído está no *software* (especialmente no sistema operacional), e não no *hardware*. O sistema distribuído é um caso especial de rede (TANENBAUM, 1994). Na Figura 1 é dado um exemplo de rede de computadores genérico.

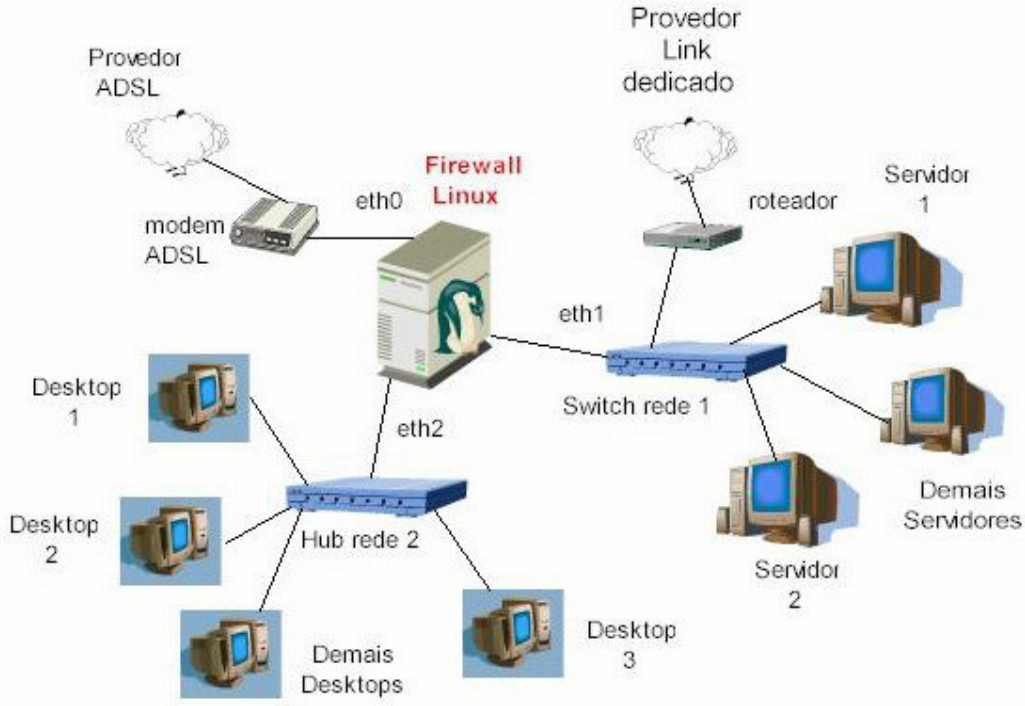

Figura 1 - Exemplo de uma rede de computadores.
Fonte: http://www.antunes.eti.br/ADSL/Rede1.jpg

1.4 - Objetivos das redes de computadores

Atualmente, além de prover o compartilhamento de recursos, redes de computadores pessoais tem por objetivo deixar todos os programas, dados e equipamentos disponíveis para qualquer um na rede, fornecendo um sistema de comunicação eficiente entre pessoas trabalhando distante umas das outras.

Um outro objetivo da rede é promover a alta confiabilidade tendo várias fontes alternativas (TANENBAUM, 1997). Geralmente precisa-se do uso dessa alta confiabilidade em aplicações militares, bancária, de controle de tráfego aéreo entre outras que precisa de operação constante mesmo acontecendo problemas com o *hardware*. Essa alta confiabilidade tem por objetivo colocar os arquivos em duas ou mais máquinas diferentes para quando ocorrerem problemas existem outras disponíveis para serem usadas.

A economia é outro objetivo da rede, onde computadores de grande porte chamados *mainframes* estão sendo substituídos gradativamente por computadores de pequeno porte que têm, geralmente, um custo menor.

1.5 - Classificação das redes de computadores

As classificações das redes de computadores não são exatas, mas existem dois tópicos que se destacam: Escala e Tecnologia de transmissão (TANENBAUM, 2003).

1.5.1 - Escala

É uma classificação baseada no alcance que a rede possui. A distância é importante como uma métrica de classificação, pois são empregadas diferentes técnicas em escalas distintas. Será feita uma abordagem sobre redes locais, redes metropolitanas e redes de longa distância. Na Tabela 1 tem-se a classificação de sistemas, organizadas por seu tamanho físico.

Tabela 1 - Classificação de tipos de rede de acordo com a escala.
Fonte: (TANENBAUM, 2003).

Distância entre processadores	**Processadores localizados no(a) mesmo (a)**	**Exemplo**
1m	Metro Quadrado	Rede pessoal
10m 100m 1 km	Sala Edifício Campus	Rede Local (LAN)
10 km	Cidade	Rede Metropolitana (MAN)
100 km 1.000 km	País Continente	Rede geograficamente distribuída (WAN)
10.000 km	Planeta	A Internet

As redes locais, chamadas LANs (*Local Area Network*) são redes geralmente de propriedades privadas, geralmente mais usadas por computadores pessoais e estações de trabalho, permitindo o compartilhamento de recursos e a troca de informações. As LANs tem características que as distinguem de outros tipos de redes como tamanho restrito, tecnologia de transmissão e topologia.

O tamanho restrito permite conhecer o pior tempo de transmissão com antecedência, podendo servir de base para desenvolvimento de projetos e simplificar a gerência de rede. A tecnologia de transmissão se caracteriza por possuir um cabo simples, ao qual são conectadas todas as máquinas (TANENBAUM, 2003). Na seção 1.5.2, tem-se uma abordagem mais detalhada sobre esse tipo de tecnologia.

Uma rede MAN (*Metropolitan Area Network*) tem por característica o alcance de distâncias metropolitanas, sendo esses sistemas chamados de Redes Metropolitanas. As redes do tipo MANs tem características semelhantes às LANs, tendo diferencial de cobrirem distâncias maiores operando em velocidades maiores, e suportam em geral dados e voz (telefonia), podendo estar associadas à rede de televisão via cabo (SOARES et al, 1995).

As redes WANs (*Wide Area Network*) surgiram da necessidade de compartilhar recursos por uma população de usuários dispersos geograficamente, um país ou continente por exemplo. Esse tipo de rede tem um custo de comunicação elevado por usarem circuitos para satélite e microondas. Sendo assim essas redes são em geral públicas, isto é, sistema de comunicação chamado sub-rede de comunicação que é mantido, gerenciado e de propriedade de grandes operadores, públicas ou privadas (SOARES et al, 1995).

1.5.2 - Tecnologia de Transmissão

Segundo Tanenbaum (2003), há dois tipos de tecnologias de transmissão em uso disseminado atualmente, sendo elas redes de difusão e redes ponto a ponto.

As redes de difusão têm por característica um único canal de comunicação, onde este é compartilhado por todas as máquinas da rede. Têm um tráfego de pequenas mensagens, onde são enviados por uma máquina e recebidas por todas. Existe um campo de endereço onde é especificado para qual máquina o mesmo deve ser entregue (*unicasting*). Sempre que se recebe essa informação a máquina receptora verifica esse campo de endereço, para checar se a mensagem foi destinada a ela, caso seja ela processa esse pacote, caso não seja o pacote é ignorado.

Nos sistemas de difusão pode-se também fazer o endereçamento a todas as máquinas da rede ao mesmo tempo, com a utilização de um código especial no campo de endereço, caracterizando o chamado *broadcasting.* Um sistema de difusão permite ainda que seja endereçado algumas máquinas da rede ao mesmo tempo, caracterizando o chamado *multicasting.*

A outra tecnologia de transmissão chamada ponto a ponto tem por característica um canal exclusivo de comunicação para interligação de quaisquer duas máquinas, além do tráfego de envio de uma máquina origem para uma única máquina destino, sendo que para ir da origem ao destino um pacote pode ter de passar por uma ou mais máquinas intermediárias fazendo uso de algoritmos de roteamento para isso (TANENBAUM, 2003).

1.6 - Topologias – Definições, classificação e características

A topologia é muito importante na construção de qualquer sistema de comunicação. A distribuição da rede, ou seja, os componentes e a maneira como estão conectados, é denominada de topologia de rede. É também a relação lógica e física dos nós numa rede (SOARES et al, 1995).

A topologia de uma rede descreve como é o *layout* do meio através do qual há o tráfego de informações, e também como os dispositivos estão conectados a ele. O arranjo topológico que deve ser utilizado e quais as alternativas existentes, sendo essas alternativas dependentes do tipo de rede (LAN, MAN, WAN), são uma das questões mais observadas na construção de qualquer sistema de comunicação. A topologia de uma rede irá caracterizar o seu tipo, eficiência e velocidade (SOARES et al, 1995).

Ao longo da história das redes várias topologias foram empregadas com maior ou menor sucesso. Os três tipos básicos empregados na conexão dos computadores são barramento, estrela e anel. Há outros tipos variantes desses, como as topologias híbridas (SOARES et al, 1995).

1.7 - Modelo OSI

Para facilitar o processo de padronização e para permitir a comunicação entre máquinas heterogêneas, a Organização Internacional de Padronização (ISO) definiu o modelo OSI (*Open System Interconnection*) no início dos anos 80. Esse modelo serve de base para qualquer tipo de rede, seja de curta, média ou longa distância (TANENBAUM, 1997).

O modelo de referência ISO/OSI como é chamado, trata da interconexão de sistemas abertos. Esse modelo é dividido em sete camadas (TANENBAUM, 2003), sendo que cada

uma delas possui uma função distinta no processo de comunicação entre dois sistemas abertos. O modelo OSI em si não é uma arquitetura de rede, pois não especifica os serviços e protocolos exatos que devem ser usados em cada camada, informando apenas o que cada camada deve fazer (TANENBAUM, 2003).

Na Figura 2 são apresentados os sete níveis do modelo OSI. Pode-se ver através da figura que cada nível possui um ou mais protocolos que realizam as funções específicas daquele nível, e esses protocolos são compatíveis entre as máquinas que estão se comunicando.

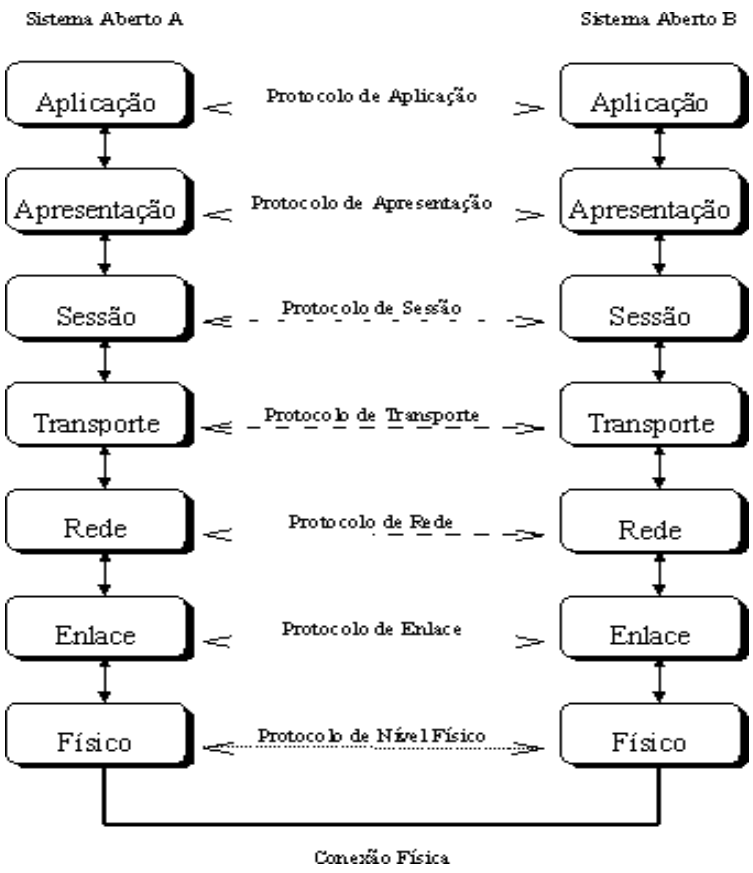

Figura 2 - Modelo de Referência OSI.
Fonte: http://www.wirelessbrasil.org/wirelessbr/colaboradores/cabral_leite/seg_wap_05.html

Entre cada nível existe uma interface. Essa interface permite que dois níveis quaisquer troquem informações. A interface também define quais primitivas, operações e

serviços o nível inferior oferece ao imediatamente superior. Cada nível é independente entre si e executa somente suas funções, sem se preocupar com as funções dos outros níveis. A seguir serão abordados todos os sete níveis do modelo OSI.

1.7.1 - Camada Física

A camada física provê características físicas, elétricas, funcionais e procedimentos. Tem a função de transmitir uma seqüência de *bits* através de um canal de comunicação. Não é função desta camada tratar de problemas como erros de transmissão (SOARES et al, 1995). A função típica dos protocolos deste nível é fazer com que um *bit* "1" transmitido por uma estação seja entendido pelo receptor como *bit* "1" e não como *bit* "0". Assim, este nível trabalha basicamente com as características mecânicas e elétricas do meio físico. Os protocolos deste nível são os que realizam a codificação/decodificação de símbolos e caracteres em sinais elétricos lançados no meio físico, que fica logo abaixo dessa camada de nível 1 (TANENBAUM, 2003).

1.7.2 - Camada de enlace de dados

O objetivo dessa camada é providenciar a transmissão de dados para a camada de rede, convertendo um canal de transmissão não confiável em um canal confiável e detectar, e possivelmente corrigir erros que possam ocorrer no meio físico (SOARES et al, 1995). Para executar essas tarefas, a camada de enlace de dados faz com que o emissor divida os dados de entrada em quadros de dados (*frames*), e transmita os quadros seqüencialmente, podendo o receptor confirmar a recepção correta de cada quadro enviando de volta um quadro de confirmação. Quase todos os protocolos do nível de enlace incluem *bits* de redundância em

seus quadros para detecção de erros. Há também o controle de fluxo para garantir que um receptor lento não receba muitos dados de um transmissor rápido (TANENBAUM, 2003).

1.7.3 - Camada de rede

A camada de rede tem a função de controlar a operação da rede de um modo geral. Converte endereços lógicos em endereços físicos de forma que os pacotes consigam chegar corretamente ao destino. Em redes de longa distância é comum que a mensagem chegue do nó fonte ao nó destino passando por diversos nós intermediários no meio do caminho, e é tarefa do nível de rede escolher o melhor caminho para essa mensagem.

A escolha da melhor rota pode ser baseada em tabelas estáticas, que são configuradas na criação da rede e são raramente modificadas, pode também ser determinada no início de cada conversação, ou ser altamente dinâmica, sendo determinada a cada novo pacote, a fim de refletir exatamente a carga da rede naquele instante (TANENBAUM, 2003). O controle de congestionamento, que controla a taxa em que os dados são transmitidos de forma que o transmissor não envie mais dados do que o receptor tenha a capacidade de receber, também é uma função da camada de rede.

1.7.4 - Camada de transporte

A camada de transporte inclui funções relacionadas com conexões entre a máquina fonte e máquina destino, dividindo os dados enviados pela camada de sessão em unidades de tamanho apropriado e transmitindo ou repassando para utilização na camada de rede. Cabe a essa camada assegurar ainda que todos os fragmentos chegarão corretamente à outra extremidade (TANENBAUM, 2003). Essa camada de transporte separa as camadas de nível

de aplicação (camadas 5 a 7) das camadas de nível físico (camadas de 1 a 3). As camadas de 1 a 3 estão preocupadas com a maneira com que os dados serão transmitidos e recebidos pela rede, mais especificamente com os quadros transmitidos pela rede. Já as camadas de 5 a 7 estão preocupadas com os dados contidos nas mensagens de dados, para serem enviados ou recebidos para a aplicação responsável pelos dados. A camada 4, Transporte faz a ligação entre esses dois grupos.

Segundo Tanenbaum (1994), a camada de transporte é o primeiro que trabalha com conexões lógicas origem-destino ou camada fim a fim, ou seja, um programa na máquina fonte conversa com um programa similar na máquina destino, diferentemente dos níveis inferiores, onde os protocolos são trocados entre cada uma das máquinas e seus vizinhos imediatos, e não entre as máquinas de origem e destino, que podem estar separadas por muitos roteadores. Vale ressaltar que a conexão criada pelo nível de transporte é uma conexão lógica, e os dados são transmitidos somente pelo meio físicos, através do nível 1 do modelo. Assim, os dados devem descer nível a nível até atingir o nível 1, para então serem transmitidos à máquina remota.

1.7.5 - Camada de sessão

A função da camada de sessão do modelo OSI é permitir que usuários de máquinas diferentes possam estabelecer sessões entre si. Os principais serviços fornecidos pelo nível de sessão são: gerenciamento de *token* impedindo que duas partes tentem executar a mesma operação ao mesmo tempo, controle de diálogo que mantém o controle de quem deve transmitir em cada momento e sincronização realizando a verificação constante de transmissões longas para permitir que elas continuem a partir do ponto em que estavam ao acontecer uma falha (TANENBAUM, 2003).

1.7.6 - Camada de apresentação

Esta camada é responsável pela representação da informação para entidades de aplicação, comunicando-se em um determinado caminho, e preservação do sentido em determinado espaço de tempo resolvendo diferenças de sintaxe. Ela tem que assegurar que a informação seja transmitida de tal forma que possa ser entendida e usada pelo receptor. Dessa forma, este nível pode modificar a sintaxe da mensagem, mas preservando sua semântica. Por exemplo, uma aplicação pode gerar uma mensagem em ASCII mesmo que a estação interlocutora utilize outra forma de codificação (como EBCDIC). A tradução entre os dois formatos é feita neste nível (TANENBAUM, 2003).

O nível de apresentação também é responsável por outros aspectos da representação dos dados, como criptografia e compressão de dados (SOARES et al, 1995).

1.7.7 - Camada de aplicação

Essa camada é totalmente voltada para os usuários finais, fornecendo serviços como a transferência de arquivos ou a troca de mensagens de correio eletrônico. Ela faz a interface entre o protocolo de comunicação e o aplicativo que pediu ou receberá a informação através da rede. O nível de aplicação sem dúvida nenhuma é o nível que possui o maior número de protocolos existentes devido ao fato de estar mais perto do usuário e os usuários possuírem necessidades diferentes. Um exemplo de protocolo é o HTTP que constitui a base para a *World Wide Web* (TANENBAUM, 2003).

1.7.8 - Funcionamento Modelo OSI

Uma maneira bastante fácil e simplificada de se enxergar a funcionalidade de um modelo em camadas, como o modelo OSI, é imaginar que cada camada tem como função adicionar um cabeçalho aos dados do usuário a serem transmitidos para outro sistema, conforme Figura 3. Deste modo à função de cada camada do outro sistema é exatamente a inversa, ou seja, retirar os cabeçalhos dos dados que chegam e entregá-los ao usuário em sua forma original.

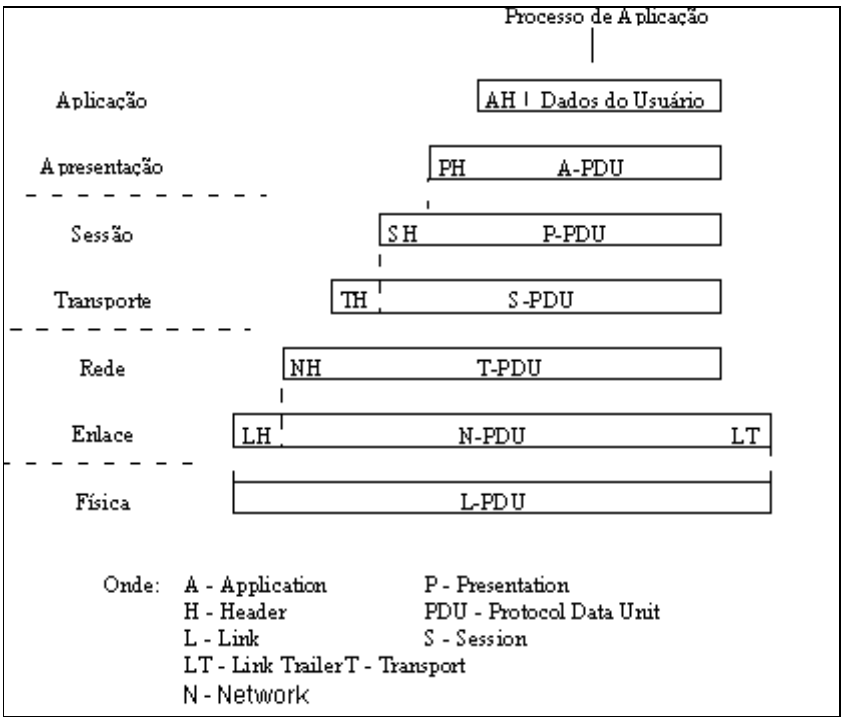

Figura 3 - Transferência de Dados entre Camadas.
Fonte: http://www.wirelessbrasil.org/wirelessbr/colaboradores/cabral_leite/seg_wap_05.html

Os dados passados pelo usuário à camada de aplicação do sistema recebem a denominação de SDU (*Service Data Unit*). A camada de aplicação, então, junta à SDU (no caso, os dados do usuário) um cabeçalho chamado PCI *(Protocol Control Information*). O

objeto resultante desta junção é chamado de PDU (*Protocol Data Unit*), que corresponde à unidade de dados especificada de um protocolo da camada em questão.

1.8 - Arquitetura TCP/IP

Outro modelo bastante difundido é o modelo TCP/IP. Segundo Tanenbaum (1997), o modelo TCP/IP surgiu através de uma rede de pesquisa patrocinada pelo Departamento de Defesa Americano (DoD) para resolver o problema de interconexão de computadores heterogêneos.

Uma das tarefas essenciais dessa rede era manter a comunicação mesmo que acontecessem guerras ou catástrofes. Dessa necessidade, surgiu a ARPANET, uma rede que permaneceria intacta caso um dos servidores perdesse a conexão. A ARPANET cresceu e tornou-se a rede mundial de computadores, também chamada de Internet.

Aos poucos, centenas de universidades e repartições públicas foram sendo conectadas a ela através de linhas telefônicas dedicadas. Com o surgimento das redes de rádio e satélite, apareceram vários problemas com os protocolos existentes, forçando a criação de uma nova arquitetura de referência. O objetivo era conectar várias redes ao mesmo tempo. Essa arquitetura ficou conhecida como Modelo de Referência TCP/IP, graças a seus dois principais protocolos TCP (*Transmission Control Protocol*) e IP (*Internet Protocol*) (TANENBAUM, 2003).

Em 1974, um estudo feito por Vinton Cerf e Robert Kahn, propôs um grupo de protocolos centrais para satisfazer necessidades como roteamento entre redes diferentes, independência do *hardware*, possibilitando recobrar-se de falhas entre outras. Originalmente, esses protocolos foram chamados de NCP (*Network Control Program*), mas, em 1978, passaram a ser chamado de TCP/IP.

Em 1980, o DARPA começou a implementar o TCP/IP na ARPANET, dando origem à Internet. Em 1983, o DARPA finalizou a conversão de todos seus computadores e exigiu a implementação do TCP/IP em todos os computadores que quisessem se conectar à ARPANET. Assim, o TCP/IP ficou sendo utilizado como o padrão de fato para interconectar sistemas de diferentes fabricantes, não apenas na Internet, mas em diversos ramos de negócios que requerem tal forma de comunicação (COMER, 1997).

Ao contrário do modelo ISO/OSI, que designou 7 camadas para o sistema de redes locais, o protocolo TCP/IP é constituído basicamente por 4 camadas: a camada de interface de rede, a camada de rede, a camada de transporte e a camada de aplicação. A Figura 4 apresenta o modelo TCP/IP em comparação ao modelo OSI (SOARES et al, 1995).

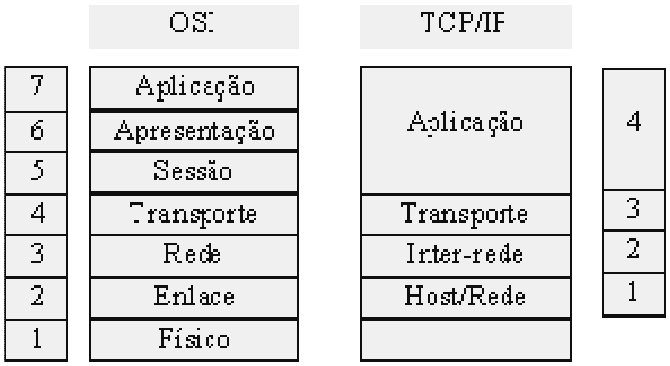

Figura 4 - Modelo OSI versus TCP/IP.
Fonte: http://www.inf.unisinos.br/~cbi/aularedes/redescbi.htm

A seguir será feita uma descrição de cada uma das 4 camadas do modelo TCP/IP, para melhor compreensão das funções desempenhadas por elas.

1.8.1 - Camada host/rede

Esta camada não tem um protocolo definido, e o modelo TCP/IP não defini muito bem o que acontece nela, somente diz que o *host* tem que se conectar à rede utilizando algum

protocolo que deve enviar os pacotes IP e receber estes pacotes, sendo este protocolo indefinido e podendo variar de *host* para *host* e de rede para rede (TANENBAUM, 2003). Por ser uma camada não normatizada pelo modelo, isso provê uma das grandes virtudes do modelo TCP/IP que é a possibilidade de interconexão e interoperação de redes heterogêneas.

1.8.2 - Camada inter-rede

A camada inter-rede trata das informações de uma máquina para outra. Também conhecida como camada Internet, é responsável pelo endereçamento, roteamento dos pacotes, controle de envio e recepção (erros, *bufferização*, fragmentação, seqüência, reconhecimento). Essa camada define um formato de pacote oficial e um protocolo chamado IP (*Internet Protocol*). O roteamento de pacotes é muito importante nessa camada. (TANENBAUM, 2003). A camada de rede é uma camada não orientada à conexão, portanto se comunica através de datagramas.

1.8.3 - Camada de Transporte

A camada de transporte é uma camada fim-a-fim, isto é, uma entidade desta camada só se comunica com a sua entidade par do *host* destinatário. É nesta camada que se faz o controle da conversação entre as aplicações intercomunicadas da rede. O *software* da camada de transporte divide o fluxo de dados transmitidos em pequenas partes, chamadas de pacotes passando-os juntamente com o endereço de destino à camada seguinte para ser transmitido (COMER, 1997).

A camada de transporte utiliza dois protocolos: o TCP e o UDP. O primeiro é orientado à conexão confiável que permite a entrega sem erros de um fluxo de *bytes* e o

segundo é não orientado à conexão e não confiável, destinado para aplicações que não exigem controle de fluxo nem controle de seqüência de mensagens enviadas. Ambos os protocolos podem servir a mais de uma aplicação simultaneamente (TANENBAUM, 2003).

1.8.4 - Camada de Aplicação

Os usuários executam programas de aplicações para acessar serviços disponíveis através de uma interligação em redes TCP/IP (COMER, 1997). As aplicações interagem com um dos protocolos do nível de transporte para enviar e receber dados. Esta camada não possui um padrão comum. O padrão estabelece-se para cada aplicação, isto é, o FTP possui seu próprio protocolo, o TELNET possui o seu próprio, bem como o SMTP (*Simple Mail Transfer Protocol*), GOPHER, DNS (*Domain Name System*), entre outros. É na camada de aplicação que se estabelece o tratamento das diferenças entre representação de formato de dados (COMER, 1997).

1.9 - Considerações finais

Foi apresentada uma introdução ao assunto redes de computadores. Nele foram apresentados conceitos relevantes paras os capítulos posteriores, tais como surgimento das redes, definição, objetivos e características gerais, com a finalidade de contextualizar o assunto redes de computadores.

Visto que este projeto trabalha com redes sem fio, no próximo capítulo será discutida essa tecnologia emergente.

CAPÍTULO 2 - REDES SEM FIO

De acordo com o Capítulo 1, as redes de computadores são hoje uma realidade e estão em crescente desenvolvimento. Esse cenário foi propício para o surgimento de novas tecnologias, tais como as redes *wireless*. Sendo assim, este capítulo tem por objetivo apresentar os aspectos gerais e as características das redes sem fio.

2.1 - Considerações Iniciais

As redes sem fio tem se expandido, por oferecem inúmeras vantagens. Uma das grandes vantagens deste tipo de rede é a flexibilidade oferecida. Sem a necessidade de cabos, os usuários estão livres para se moverem enquanto estão conectados à rede de forma transparente.

Redes sem fio constituem um dos assuntos de extrema relevância para este trabalho, pois foi usada para avaliação do protocolo WEP (*Wired Equivalent Privacy*) (JUNIOR et al, 2004). Neste capítulo será realizado um estudo das redes sem fio, partindo do seu surgimento, conceito, vantagens e desvantagens, padrões, tipos e topologias de redes sem fio, que está sendo um dos segmentos de mais rápido crescimento na informática, possibilitando uma grande melhoria no acesso a informações, principalmente em relação à mobilidade (TANENBAUM, 1997).

2.2 - Surgimento das redes sem fio e protocolos de acesso.

As transmissões sem fio têm percorrido um longo caminho desde que Henrich Hertz descobriu as ondas de rádio e desenvolveu o primeiro transmissor e receptor em 1887. A

primeira rede a utilizar a comunicação sem fio foi o Aloha, desenvolvida na Universidade do Havaí pelo professor Norman Abramson e seus colegas, no início da década de 70. Eles elaboraram um método novo e sofisticado para resolver o problema de alocação de canais, usando um sistema de difusão por rádio como meio de transmissão, ao invés de cabo ponto a ponto. Com a utilização do rádio, não precisou utilizar linhas telefônicas que eram caras e não confiáveis (TANENBAUM, 1994).

O objetivo principal desta pesquisa era a interligação de 4 campos, situados em 4 ilhas com o computador central, que ficava na ilha de Oahu. As redes sem fio não se difundiram na época da rede Aloha devido à baixa taxa de transmissão e um alto custo, mas a partir dela, vários pesquisadores ampliaram este trabalho, pois houve um grande interesse em aperfeiçoar esta tecnologia, tornando-a viável (ARAUJO et al, 2004).

A comunicação no Aloha foi através de um transmissor/receptor de rádio FM contido em cada estação, que se comunicava com o transmissor/receptor do centro de computação. Possuía um alcance estimado de 30 km, sendo que depois de um tempo foi introduzido um repetidor mais potente com um alcance estimado de 500 km. Toda a comunicação trafegava do centro de computação para uma estação ou de uma estação para o centro de computação, não havendo comunicação entre estações. A explicação para dois canais distintos foi o fluxo do tráfego de entrada e o de saída, sendo que na entrada há muitos usuários desordenados competindo pelo acesso ao recurso compartilhado e na saída há um único lugar com controle sobre o canal, não havendo contenção nem colisões (TANENBAUM, 1994).

A partir da rede Aloha foi utilizado pela primeira vez protocolos com acesso baseado em contenção.

Segundo Soares (1995) os métodos de acesso podem ser divididos em dois grandes grupos: métodos baseados em contenção e os de acesso ordenado sem contenções. Nas Tabela 2 e Tabela 3 são apresentados alguns dos protocolos referentes a esses métodos e as

funções que desempenham. Estas tabelas foram obtidas a partir de informações obtidas em (SOARES et al, 1995).

Tabela 2 - Protocolos métodos de acesso baseados em contenção

Método baseado em contenção	Descrição
ALOHA puro	Cada terminal escuta apenas o canal computador-terminal, não podendo saber se o canal terminal-computador está sendo utilizado por algum outro terminal. Sendo assim, quando o terminal possui alguma mensagem pronta para ser enviada ele simplesmente envia sem se preocupar se o meio está ocupado ou não. Ao completar a transmissão, liga um temporizador e aguarda uma resposta do computador central que vai indicar o reconhecimento da mensagem. Se o reconhecimento não for recebido até se esgotar o tempo de espera, o terminal entende que a mensagem foi corrompida e que deve ser retransmitida.
ALOHA em intervalos (*slotted*)	Dobra a eficiência do Aloha puro. Transmissão em *Slots* de tempo bem definidos. Permiti que transmissões só sejam iniciadas em intervalos fixos de tempo. O computador central se encarrega de dividir o tempo total em intervalos de tempo fixos. Um terminal que deseja transmitir, só pode iniciar a transmissão no começo de um intervalo. Isto diminui consideravelmente o total de tempo perdido na ocorrência de colisão
CSMA	Uma estação só transmite sua mensagem após "escutar" o meio de transmissão e determinar que o mesmo não esteja sendo utilizado. Caso a estação detecte o meio ocupado, ela deve aguardar até que o sinal desapareça para então iniciar a sua transmissão. Pode ocorrer que duas ou mais estações estejam aguardando que o meio fique desocupado para iniciarem suas transmissões e isto ocasionará uma colisão de mensagens das estações ao transmitirem simultaneamente. Várias estratégias foram desenvolvias para aumento da eficiência.
np-CSMA	Tem um retardo aleatório quando o canal é detectado como ocupado. Se o nó detecta o meio livre, ele transmite sua mensagem. Se o nó detecta o meio ocupado, tenta transmitir mais tarde, de acordo com uma distribuição aleatória de atrasos. O algoritmo é repetido na nova tentativa.
p-CSMA	Se estiver ocupado, espera até desocupar e então transmite com probabilidade p.
CSMA/CD	Interrompe a transmissão no caso de colisão. Este método de acesso foi um dos escolhidos como padrão, e é de fato o método mais difundido em redes locais.

Tabela 3 - Protocolos métodos de acesso ordenado sem contenção

Método acesso ordenado sem contenção	**Descrição**
Polling	Uma estação controla quem pode ou não transmitir
Quadro vazio ou *Slot* Vazio	O tempo de transmissão é dividido em *slots* (fatias) iguais
Token Bus	Anel lógico em barramento físico. O *token* não transporta qualquer informação, mas permite que o seu proprietário transmita sua mensagem. Apenas um *token* deve existir na rede em um determinado instante.
Token Ring	Uso de *Token* em rede em anel. A estação que deseja transmitir aguarda a chegada da "ficha". Uma vez de posse da ficha, a estação transmite a sua mensagem. Ao final da transmissão, a estação devolve a ficha enviando-a ao próximo nó da rede.

As redes sem fio possuem algumas vantagens e desvantagens em relação às redes cabeadas. Pode-se apontar como vantagens o fato de ser uma rede com mais flexibilidade e mobilidade, atingindo lugares onde os fios não poderiam chegar, e a alta produtividade e serviços; a robustez, que faz com que as redes sem fio, mesmo com um desastre (Ex: terremoto), possam continuar inteiras garantindo a comunicação, a facilidade de instalação, uma vez que geralmente é rápida, não precisando passar cabos por paredes ou canaletas. Mesmo tendo um custo inicial mais alto que as redes convencionais, têm-se a redução do custo agregado, pois junta-se vantagens como: facilidade de expansão, menor necessidade de manutenção, robustez e outros fatores que ajudam na recuperação do custo inicial (MATHIAS, 2006).

Entretanto, existem algumas desvantagens presentes como: qualidade de serviço, que é menor do que as redes cabeadas, uma vez que possui uma pequena banda passante devido às limitações da rádio transmissão e a alta taxa de erro devido à interferência.

A segurança é outra desvantagem além de ser um grande problema das redes sem fio. As redes sem fio são mais propensas a possuir interceptores, podendo estes ser intrusos, equipamentos elétricos que interferem e causam a perda de dados, entre outros.

Apesar da baixa taxa de transferência de dados ser um problema, essa taxa vem crescendo rapidamente, mesmo sendo ainda menor que as da rede cabeada (MATHIAS, 2006).

2.3 - Padrão IEEE 802.11

As primeiras redes sem fio utilizavam um tipo de transmissão chamada *spread spectrum,* que é uma tecnologia de modulação por espalhamento espectral utilizada comumente em rádios que operam na faixa de freqüência isenta de licença. Utilizavam também infravermelha difusa. No entanto, possuíam uma baixa interoperabilidade devido ao fato dos fabricantes (como IBM, CISCO, Telecom e 3COM) utilizarem padrões proprietários, o que causava a falta de padronização atrasando o desenvolvimento e, sobretudo, a popularização de redes sem fio (BATISTA, 2002).

Para padronizar o acesso ao meio físico sem fio e permitir a interoperabilidade entre fabricantes o IEEE em 1991 aprovou a norma 802.11, que regulamenta as redes sem fio. Devido a atrasos, apenas em 1997 foi publicada a especificação que padronizava a conectividade sem fio entre equipamentos em uma área local e que permitia utilização de equipamentos de diferentes fabricantes. Esse padrão 802.11, à medida que era elaborado, foi sendo adotado pelos fabricantes de redes sem fio, que então passaram a elaborar seus produtos baseando-se nessas normas, uma vez que a padronização oferece interoperabilidade, confiabilidade e diminuição nos custos, provendo assim uma boa aceitação do mercado (ARAUJO et al, 2004).

A padronização das comunicações sem fio pela norma 802.11 permitia taxas de transmissão de 1 a 2 Mbps, operando na banda de freqüência não licenciada de 2,4GHz e utilizando a técnica de transmissão de FHSS (*frequency-hopping spread spectrum*) ou DSSS (*direct-sequence spread spectrum*). Estas técnicas serão melhores estudadas no item 2.3.1.1.

Por essa taxa de transmissão ser baixa, o IEEE em 1999 descreve um novo padrão para redes sem fio, o 802.11 b, que possuía a mesma tecnologia e arquitetura, mas com taxas maiores na faixa de 5.5 a 11 Mbps, valores esses aproximados ao das redes cabeadas (ARAUJO et al, 2004).

As normas 802.11 e 802.11b descrevem como a tecnologia trata os pontos principais para o desempenho das redes sem fio (BATISTA, 2002). A partir de 1999 novas tecnologias foram surgindo, entre elas os padrões 802.11a, 802.11g, além de vários outros. Na Tabela 4 tem-se alguns padrões e sua respectiva descrição.

Tabela 4 - Resumo padrões do IEEE 802.11
Fonte: (ARAUJO et al, 2004).

Padrão	Descrição	Observação	Avaliação
802.11a	Uma camada física numa banda de rádio de 5 GHz. Possibilitando 8 canais de rádio. Máximo 54 Mbps por link por canal.	Especifica uma rede até cinco vezes mais rápida que a 802.11b.	Padrão definido em 1999.
802.11b	Uma camada física numa banda de rádio de 2.4 GHz. Possibilitando 3 canais de rádio. Máximo 11 Mbps por link.	Performance equivalente a uma rede *ethernet*.	Padrão definido em 1999.
802.11d	Este padrão é suplementar para a camada MAC.	Permite alterar a freqüência de transmissões permitindo o uso de equipamentos em países de diferentes legislações de uso de frequências.	Trabalhos em andamento.
802.11e	Este padrão é suplementar para a camada MAC, para dar suporte de QoS (Quality of Service) às aplicações LAN. O propósito é servir diversos níveis de aplicações de dados, voz e vídeo.	Serviço de qualidade. Muitas WLAN têm o objetivo de garantir QoS para diferenciarem seus produtos.	O padrão foi finalizado no segundo semestre de 2002.
802.11f	Este documento é para recomendar o uso de um AP (Access Point) para as conexões WLAN.	Inter-operabilidade padrão encerrado no segundo semestre de 2002.	Avaliação esperada.
802.11g	802.11g Esta camada especifica WLANs com taxas entre 2.4 GHz e 5GHz, com no máximo 54 Mbps por canal.	Performance do 802.11a em 2,4 GHz.	Definido no segundo semestre de 2002, Ratificado em junho de 2003
802.11h	Este padrão é suplementar com a camada MAC, obedecendo as regulamentações Européias sobre WLANs.	Concordância das regulamentacoes Européias. Isto se faz necessário para que os produtos sejam aceitos na Europa.	Produtos avaliados no primeiro semestre de 2003 e aprovação esperada.
802.11i	Padrão suplementar a camada MAC para melhorar a segurança. Revisão dos padrões 802.11 a, b e g. Oferecendo uma nova alternativa ao WEP com novos métodos de encriptação e autenticação.	Aperfeiçoamento da segurança. A maior fraqueza das WLANs é a insegurança. Soluções estão surgindo através de do TKIP (Temporal Key Integrity Protocol).	Este protocolo era esperado para o final de 2002, mas não se confirmou.

2.3.1 - Arquitetura do Padrão IEEE 802.11

A idéia da arquitetura do padrão IEEE 802.11 é a de especificar um conjunto de componentes que interagem entre si para fornecer uma rede local sem fio que seja transparente para as camadas superiores e que ofereça suporte à mobilidade das estações.

Esta arquitetura baseia-se na divisão da área de cobertura da rede em células. Essas células, chamadas de BSA (*Basic Service Área*) têm tamanho variado, dependendo de fatores como a potência dos transmissores e receptores, e do ambiente que estão operando (SOARES et al, 1995). Outro componente da arquitetura IEEE 802.11 é o BSS (*Basic Service Set*), que é definido como um grupo de estações (*Stations* – STAs) que estão no controle de uma única função de coordenação (SOARES et al, 1995).

Os membros de um BSS se encontram na área da BSA. Apesar de ser possível a existência de uma rede sem fio com apenas 1 célula (*Ad-Hoc*), normalmente as redes sem fio são formadas por várias células. Nesse caso múltiplas BSAs são interligadas através de um DS (*Distribution System*), outro componente da arquitetura. Tal sistema pode ser uma rede que possui um meio de transmissão sem fio, ou mesmo outro meio como par trançado, cabo coaxial, fibra ótica.

O AP (*Acess Point*), mais um componente da arquitetura 802.11, é um equipamento especial que captura as transmissões realizadas pelas estações de sua BSA e retransmite ao destino, localizado em outra BSA, utilizando o Sistema de Distribuição. Um Ponto de Acesso é comparado a um HUB das redes cabeadas. Os BSAs que são interligados pelo sistema de distribuição através dos APs formam as chamadas ESA (*Extended Service Area*). Já os chamados ESSs (*Extended Service Set*) são definidos por um conjunto de estações formado pela união de vários BSSs interligado por um sistema de distribuição (SOARES et al, 1995).

A Figura 5 ilustra os componentes da arquitetura do 802.11.

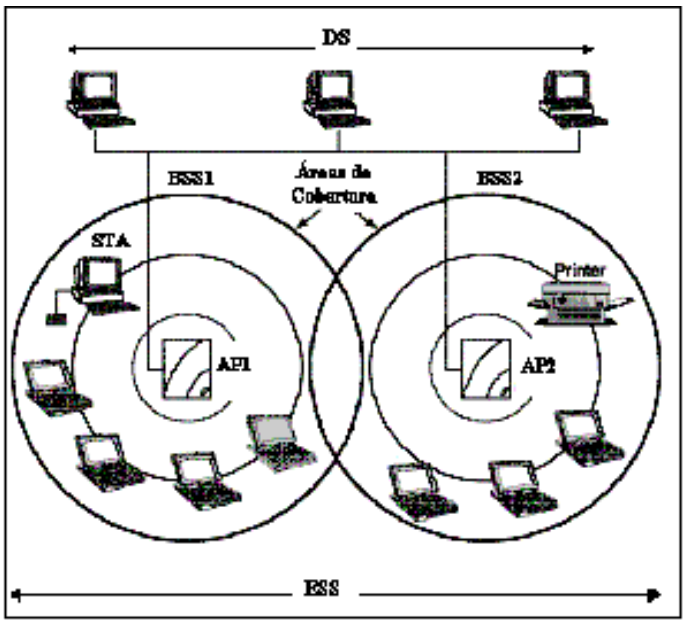

Figura 5 - Arquitetura de rede IEEE 802.11
Fonte: http://www.teleco.com.br/tutoriais/tutorialwlan/pagina_3.asp

O padrão IEEE 802.11 enfoca dois níveis do Modelo de Referência OSI: a camada física e de enlace, conforme ilustrado na Figura 6.

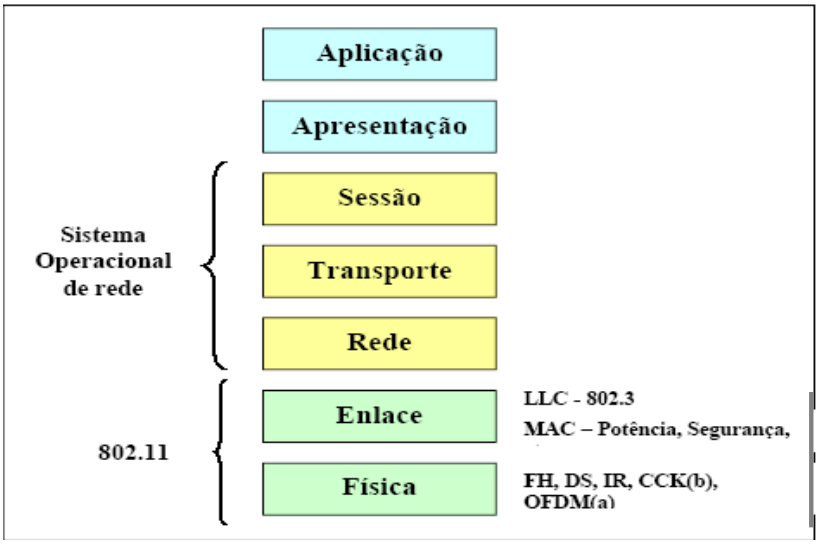

Figura 6 - 802.11 – Modelo OSI.
Fonte: (BATISTA, 2002).

2.3.1.1 - Camada física 802.11

Há três tipos diferentes de técnicas de transmissão permitidas pela camada física, definido pelo padrão 802.11 de 1997: espalhamento de espectro por salto em freqüências (FHSS), espalhamento de espectro por seqüência direta (DSSS) e infravermelho. Cada camada física oferece taxas de 1 a 2 Mbps, e em cada um delas existe um sinal de avaliação de canal livre (CCA), que serve para indicar à camada física se o meio está livre, evitando colisões (JUNIOR, 2003).

Em 1999 foram apresentadas duas novas técnicas, que são chamadas OFDM (Multiplexação Ortogonal por divisão de Freqüência) e HR-DSSS (*High Rate Direct Sequence Spread Spectrum*), para alcançar maior largura de banda atuando com 54Mbps e 11Mbps respectivamente (TANENBAUM, 2003). Todas essas técnicas de transmissão permitem o envio de quadros MAC (*Media Access Control*) de uma estação para outra, diferindo-se apenas na tecnologia usada e nas velocidades que podem ser alcançadas.

O FHSS usa um esquema de modulação *spread-spectrum* que divide a banda passante em vários canais menores, alterando a freqüência segundo um padrão conhecido pelo transmissor e pelo receptor, que sincronizados adequadamente mantêm um único canal lógico. Eles usam essa banda por um tempo que é ajustável e depois mudam para outro canal, permitindo a coexistência de várias redes em uma mesma área. Essa técnica fornece alguma segurança, pois um intruso que não conheça a seqüência de saltos não poderá ter acesso às informações. O FHSS usa a banda ISM (*Industrial, Scientific, and Medical*) que vai de 2,4000 a 2,4835 GHz (JUNIOR, 2003). Sua principal desvantagem é a baixa largura de banda, 1Mhz.

O DSSS também é um método de espalhamento de espectro, se restringindo a velocidade de 1 a 2 Mbps e utilizando a banda ISM de 2,4 GHz, diferenciando-se da técnica FHSS por separar por código as transmissões simultâneas e não por freqüência. Funciona com

o espalhamento da banda disponível em 11 subcanais, cada um com 11Mhz, e com a utilização da seqüência de Barker de 11 *chips*, para efetuar o espalhamento de cada símbolo de dados (MELO, 2004).

Esse padrão de *bits* chamado *chip* permite aos receptores filtrar sinais que não utilizam o mesmo padrão, incluindo ruídos e interferências. A vantagem desta técnica é que ela reduz os efeitos de interferência de fontes de banda estreita (ZANNETI, GONÇALVES, 2006).

A utilização do infravermelho usa transmissão difusa, com comprimento de onda de 850 a 950 nano metro, e são permitidas duas velocidades: 1Mbps e 2Mbps. Os sinais de infravermelho não atravessam paredes não permitindo, portanto, células situadas em salas diferentes. Isso é possível, pois foi projetado para ambientes fechados, operando com transmissões não direcionais que alcançam no máximo aproximadamente 10 metros, caso não haja luz solar interferindo. Este é um dos problemas para que o infravermelho não seja uma opção popular, outro é a baixa largura de banda (TANENBAUM, 2003).

O OFDM é baseado na idéia de multiplexação por divisão de freqüência, enviando múltiplos sinais em diferentes freqüências. São usadas 52 freqüências, sendo 48 para dados e 4 para sincronização. A primeira das LANs sem fio de alta velocidade, a LAN 802.11a, utiliza OFDM, para transmissão com velocidade de até 54Mbps, utilizando também a banda ISM, porém na mais larga de 5Ghz. Essa divisão do sinal em várias bandas estreitas tem algumas vantagens como, melhor imunidade à interferência de banda estreita e a possibilidade de uso de bandas não contíguas (TANENBAUM, 2003).

O HR-DSSS consiste na técnica de espalhamento de *Spectrum* e que usa 11 milhões de *chips*/sec para alcançar 11Mbps na faixa de 2.4 GHz. É chamado de 802.11b, mas não é um seguimento da 802.11a, na realidade seu padrão foi o primeiro a ser aprovado e comercializado. Suporta taxa de dados 1.2, 5.5 e 11Mbps, sendo que as 2 taxas mais baixas

funcionam com 1 e 2 *bits* por banda respectivamente para tornar-se compatível com a DSSS. A taxa de dados pode ser adaptada durante a operação a fim de alcançar a melhor velocidade com baixas cargas e ruídos (TANENBAUM, 2003).

2.3.1.2 - Camada enlace do 802.11

Para determinar qual estação tem o direito de transmitir e receber dados utilizando o meio sem fio. o controle de acesso ao meio (MAC) é baseado em funções de coordenação, sendo duas definidas pelo 802.11: uma distribuída conhecida como DCF (*Distributed Coordination Function*), não usando nenhum controle central, sendo de implementação obrigatória proporcionando um acesso com contenção. A outra é centralizada, chamada de PCF (*Point Coordination Function*), a qual possui uma implementação opcional, e utiliza a estação base para controlar toda a atividade em sua célula. A Figura 7 ilustra os modos de acesso do MAC 802.11, sendo que o PCF, como pode ser visto, utiliza as regras de acesso do DCF (MELO, 2004).

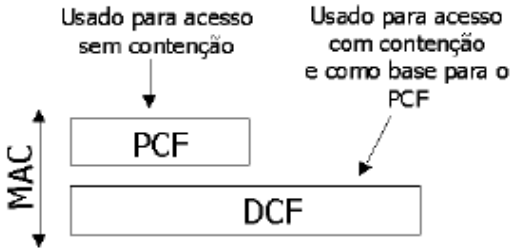

Figura 7 - Modos de acesso do MAC 802.11
Fonte: (MELO, 2004).

No modo DCF, o 802.11 utiliza o protocolo CSMA/CA (CSMA com abstenção de colisão). A diferença entre o CSMA/CA e o CSMA/CD é justamente o fato de não haver a detecção de colisão no meio sem fio. Para suprir essa ausência de detecção de colisão utiliza-

se um mecanismo de prevenção de colisão junto com o reconhecimento positivo (*ACK*) dos quadros de dados (MELO, 2004).

O CSMA/CA admite dois métodos de operação:

- no primeiro a estação irá efetuar a escuta do canal, se esse estiver ocioso a estação começará a transmitir. A estação não escuta o canal enquanto está acontecendo à transmissão, mas envia o quadro inteiro, que poderá ser destruído no receptor devido à interferência. Caso o canal esteja ocupado, espera-se o canal ficar ativo para efetuar a transmissão. Outra maneira é basear-se no protocolo MACAW (Multiple *Access with Collision Avoidance for Wireless*), empregando a detecção do canal virtual. Neste modo de operação a estação que deseja transmitir reserva o canal por um tempo através de um pacote RTS (*Request to Send*) que é confirmado através de um pacote CTS (*Clear to Send*), reservando o meio para a estação solicitante. Em ambas as formas, a transmissão é considerada com sucesso após a confirmação *ACK* (*Acknowledgement*) da estação receptora (TANENBAUM, 2003);
- segundo modo permitido é o PCF, que implementa um mecanismo de acesso ordenado ao meio, suportando a transmissão de tráfego com retardo limitado ou tráfego assíncrono. Esse modo é usado em redes com infra-estrutura e onde não haja intersecção entre BSSs que operam na mesma faixa de freqüência (SOARES et al, 1995).

2.4 - Topologias Padrão IEE 802.11

Este padrão, 802.11 define dois modos de operação: redes com infra-estrutura e redes sem infra-estrutura denominadas *ad-hoc*. As sessões que seguem apresentam esses dois modos, dando um enfoque maior às redes *ad-hoc*.

2.4.1 - Redes com Infra-estrutura

As redes infra-estruturadas têm a característica de possuir a comunicação dos *hosts* móveis sempre com um *host* fixo, ou seja, mesmo que dois *hosts* móveis estejam próximos com capacidade de fazer uma comunicação direta, tem que passar pelo *host* fixo. Existe pelo menos um ponto de acesso, também chamado de estação de suporte a mobilidade (ESM), conectado em uma infra-estrutura de rede fixa.

Esse tipo de rede utiliza uma arquitetura em células chamadas de BSS *(basic service set)*, já abordada anteriormente em arquitetura do 802.11 no item 2.3.1. Uma típica BSS contém pelo menos uma estação e uma estação central chamada de ponto de acesso.

Toda a comunicação dos nós móveis é feita através do ESM. O maior problema desse tipo de rede é com o controle de acesso ao meio. Não existe o problema do roteamento, pois toda comunicação tem que obrigatoriamente passar pelo *host* fixo (CÂMARA, 2006). Na Figura 8 encontra-se um exemplo de uma rede infra-estrutura.

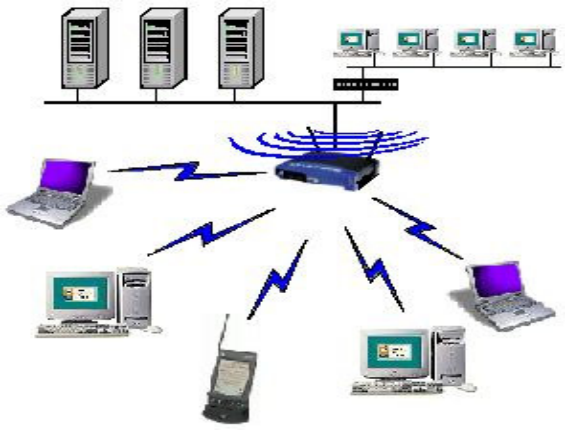

Figura 8 - Rede infra-estruturada.
Fonte: http://www.richard.eti.br/duvidas58.html

Em uma rede com infra-estrutura acontece mais comunicação que em uma rede *ad-hoc*. Essa exigência maior de capacidade de transmissão acontece pelo fato de toda comunicação ter que passar pelo AP. O processo de comunicação fará dois saltos, transmitindo primeiro da origem ao AP, e depois do AP ao destino. Mesmo exigindo maior capacidade de transmissão existem vantagens nas redes infra-estruturadas. O fato de ter um AP diminui o custo com relação à complexidade da camada física. Se a comunicação fosse direta como na rede *ad-hoc* cada estação (nó) tem que ter a tabelas de todos os relacionamentos possíveis com outros nós. Os APs permitem também economizar energia, pois permitem que estações permaneçam inativas enquanto não recebem quadros (ARAUJO et al, 2004).

2.4.2 - Redes ad-hoc

Essas redes são também conhecidas pelo nome de *peer-to-peer* ou IBSS (*Independent Basic Service Set).* Nessas redes os nós não precisam de um AP para controlar o acesso ao meio e para se comunicar, ocorrendo comunicação direta entre as estações, e não há qualquer conexão com uma rede fixa (BATISTA, 2002). A Figura 9 ilustra uma rede *ad-hoc*.

Figura 9 - Rede ad-hoc.
Fonte: http://www.richard.eti.br/duvidas58.html

Neste tipo de rede toda comunicação é direta entre os *hosts* móveis, e se o destino não está ao alcance solicita-se o serviço de outro *host* móvel vizinho, utilizando estes *hosts* como ponte para a informação chegar ao destino. O *host* fixo não é considerado, se existir ele fica como sendo um *host* móvel (CÂMARA, 2006). A Figura 10 demonstra a utilização de *hosts* móveis como pontes para uma mensagem ir da origem A ao destino H.

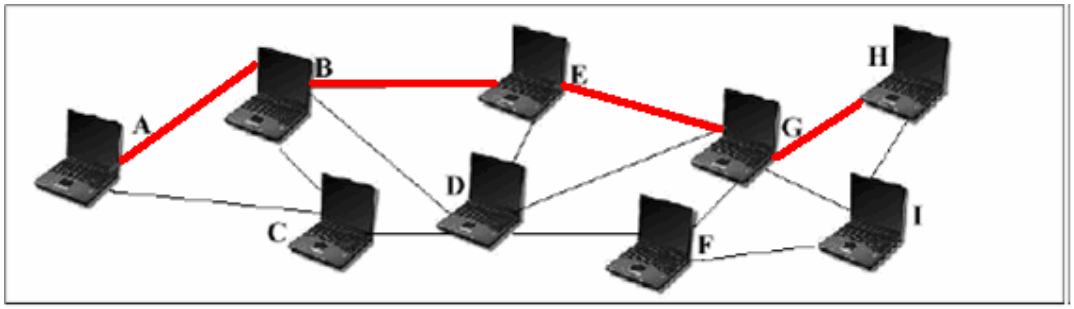

Figura 10 - Rede *ad-hoc* usando estações como pontes**.**
Fonte: (ARAUJO et al, 2004).

O tamanho da área de cobertura de uma rede *ad-hoc* depende de fatores como obstáculos, proximidade dos *hosts*, entre outros.

Existem algumas vantagens em se utilizar redes *ad-hoc* em relação a redes com infra-estruturada ou até mesmo rede fixa sendo: o fato delas serem muito úteis por facilitar e agilizar a instalação em qualquer lugar onde não exista uma infra-estrutura fixa; existe maior tolerância a falhas, pois uma estação que apresente problema ou que seja desativada, rapidamente é solucionado esse problema, o que não acontece em uma rede fixa ou rede infra-estruturada.

A comunicação estabelecida por duas estações, se estiverem enxergando uma a outra, é mais uma vantagem, o que não ocorre em uma rede fixa ou infra-estruturada. A principal vantagem provida por essas redes é a mobilidade que proporcionam (CÂMARA, 2006). Porém existem algumas desvantagens no uso de redes *ad-hoc* como: a banda passante menor

em comparação com redes fixas, taxa de erros maior, a localização do *host* móvel é dificultada por sua mobilidade, a topologia muda constantemente diferentemente das redes fixas que quase não tem alteração em curto espaço de tempo.

Há uma preocupação maior com roteamento, visto que não existe um ponto central para a distribuição de informações (CÂMARA, 2006). Por esta ausência necessita-se de algoritmos mais sofisticados para fazer o roteamento, uma vez que a topologia das redes *ad-hoc* mudam frequentemente e são imprevisíveis, causando mudanças na conectividade entre equipamentos móveis, forçando o sistema a mudar frequentemente as rotas.

Segundo Câmara (2006), há três tipos de algoritmos de roteamento mais utilizados, sendo eles: *flooding* (inundamento), em que todo pacote que chega ao nó é enviado para todos os outros *hosts* que tem contato, menos ao de origem. Este é um método simples, e por isso apresenta vários problemas. Outro algoritmo é o *link State* (estado do *link*), que faz com que um nó, ao perceber uma mudança no estado de seus vizinhos, efetua um *flooding* dessa mudança pela rede, fazendo com que os demais nós atualizem sua topologia ao saber da mudança. *Distance Vector* (Vetor Distância) é outro algoritmo muito usado que mantém uma tabela (atualizada periodicamente) com o menor caminho até todos os outros nós.

2.5 - Tipos de Redes Sem Fio

De uma maneira geral redes sem fio podem ser divididas em três categorias principais (TANENBAUM, 2003): interconexão de sistemas, LANs sem fio e WANs sem fio. Abaixo serão descritas as principais categorias de redes sem fio.

2.5.1 - Interconexão de sistemas

Neste tipo de rede usa-se rádio de alcance limitado para interconectar componentes de um computador, tais como teclado, impressora, mouse, entre outros. A interconexão de sistemas é representada pela tecnologia *Bluetooth,* de alcance limitado. Não precisa de cabos nem instalação de *drivers*, basta ligar os componentes e eles funcionarão, sendo essa facilidade de operação uma grande vantagem. A comunicação é feita simplesmente trazendo os componentes para o alcance da rede. Em sua forma mais simplificada a unidade do sistema é quem gerencia a comunicação impondo os limites de acesso, de freqüência e assim por diante (TANENBAUM, 2003).

2.5.2 - Redes LAN sem fio – WLAN

As redes locais sem fio (WLANs) constituem uma alternativa às redes convencionais com fio, fornecendo as mesmas funcionalidades, mas de forma flexível com fácil configuração e com boa conectividade em áreas prediais ou em um campus. Dependendo da tecnologia utilizada, rádio freqüência ou infravermelho, e do receptor, as rede WLANs podem atingir distâncias de até 18 metros com velocidades de até 50Mbps. Esses sistemas possuem um *modem* de rádio e uma antena em cada computador, para efetuar a comunicação com outros sistemas, podendo existir ou não uma antena acima dos computadores, permitindo a comunicação entre as máquinas.

Esse tipo de rede vem se tornando comum, tanto em lares como em escritórios. Tem substituído à rede *Ethernet* nesses lugares, uma vez que a mesma possui uma instalação trabalhosa. O padrão para as LANs sem fio é o IEEE 802.11 abordado na seção 2.3 (TANENBAUM, 2003). A Figura 11 ilustra em exemplo de rede LAN sem fio.

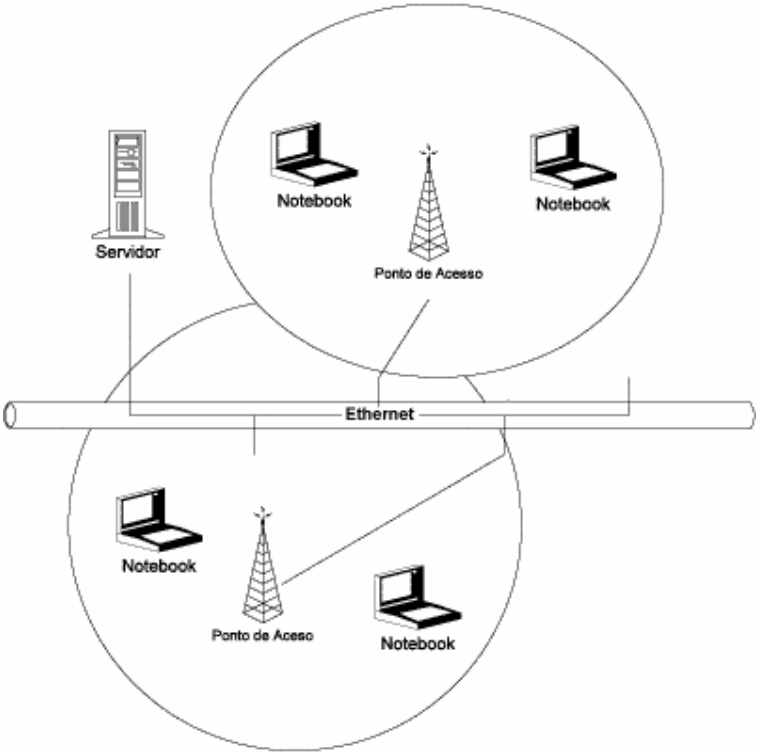

Figura 11 - Rede sem fio WLAN típica.
Fonte: http://www.rnp.br/newsgen/9805/wireless.html#ng-introducao

2.5.3 - Redes WAN sem fio – WWAN.

As redes sem fio do tipo WAN ou WWAN (*Wireless Wide Area Network*) são utilizadas em sistemas geograficamente distribuídos e baseia-se principalmente nas redes de telefonia celular. Esta foi desenvolvida a princípio para comunicação de voz e atualmente suporta também a transferência de dados, porém tem uma baixa largura de banda. Os sistemas celulares operam abaixo de 1Mbps, mas alcançam uma distância entre a estação-base e o computador medida em quilômetros, diferentemente das LANs sem fio que atingem dezenas de metros (TANENBAUM, 2003).

Essa facilidade que os usuários finais têm de conectar-se de localizações distantes onde não existe nenhum acesso com fios para grandes distâncias é um benefício da WWAN. A limitação da tecnologia é a linha de visão que é exigida para a conectividade. As condições

climáticas também possuem potencial para interferir na transmissão de tráfego. Semelhante a LAN sem fio, todas as pessoas têm acesso e compete pela mesma largura de banda, o que está se tornando um problema menor já que a capacidade de largura de banda continua a aumentar com cada geração de sistemas (TANENBAUM, 2003).

2.6 - Considerações Finais

Como este trabalho faz uso das redes sem fio, foi apresentado neste capítulo um breve histórico das redes sem fio, uma análise de suas principais tecnologias, seus principais padrões e os seus respectivos conceitos, principalmente em relação ao padrão IEEE 802.11.

Um problema que tem impedindo uma ascensão maior das redes sem fio no mercado é a segurança. Na maioria das redes sem fio usa-se o protocolo WEP (*Wired Equivalent Privacy*) para prover a segurança. Este protocolo foi estudado detalhadamente no próximo capítulo.

CAPÍTULO 3 - SEGURANÇA PARA REDES SEM FIO

3.1 - Considerações Iniciais

A segurança em redes sem fio é um assunto que merece muita atenção e pesquisa. As redes locais sem fio vêm se tornando cada vez mais utilizadas no meio corporativo, pois há um ganho na flexibilidade e na mobilidade dos equipamentos destes padrões. Porém, existem restrições quanto ao seu uso, devido às diversas vulnerabilidades encontradas no padrão 802.11x. Sendo assim, há várias medidas de segurança que devem ser adotadas ao implementar este padrão, impondo a este campo uma grande importância (CARVALHO FILHO, 2005).

Neste capítulo serão abordados protocolos de segurança para redes sem fio dando maior destaque ao protocolo WEP, presente na maioria das redes sem fio.

3.2 - Segurança no Padrão IEEE 802.11

As redes sem fio estão evoluindo a passos largos e cada vez mais vem sendo utilizadas na comunicação entre variados dispositivos como computadores pessoais, *notebooks*, telefones entre outros, e tem sido empregadas em diferentes ambientes como residências, edifícios, empresas.

Essas redes sem fio têm sido uma alternativa às redes fixa, pois dispensam o uso de cabos e permitem conectividade aos dispositivos mesmo em movimento. Porém um dos maiores problemas dessas redes é a falha na segurança, onde pessoas mal intencionadas podem invadir essas redes, comprometendo assim a segurança dos usuários e dos dados. Essa fragilidade tem barrado um desenvolvimento maior dessa tecnologia.

O padrão IEEE 802.11b é uma das soluções mais adotadas para redes locais sem fio. Esse padrão está cada vez mais presente nas empresas, hotéis, fábricas e lugares públicos como aeroportos, universidades, hospitais e centros comerciais, oferecendo a possibilidade de acesso à rede com suporte à mobilidade.

O IEEE 802.11 especifica dois métodos de segurança para as redes sem fio que são a autenticação (uso do SSID (*Service Set Identifier*)) e a privacidade (uso do WEP (*Wired Equivalent Privacy*)), que agem na camada de enlace (MILANEZ et al, 2004). A seguir serão estudados esses dois métodos.

3.2.1 - Tipos de autenticação

A autenticação garante que apenas usuários devidamente autorizados tenham acesso à rede, bloqueando o acesso às demais pessoas. Para uma rede local sem fio a autenticação pode ser realizada de duas maneiras: ou na camada de enlace de dados, ou na camada de rede. A autenticação na camada de enlace é facilitada através do uso de *WEP– Wired Equivalent Privacy*, que é um protocolo de segurança que foi projetado para tornar a segurança de uma LAN sem fio tão boa quanto à de uma LAN fisicamente conectada. Já a autenticação na camada de rede pode ser garantida com o uso do protocolo *IEEE* 802.1x, capaz de garantir a autenticação tanto na estação quanto na entidade autenticadora (TANENBAUM, 2003).

Segundo Junior (2004), o padrão IEEE 802.11b define duas formas de autenticação: *open system* e *shared key.* Independente da forma de autenticação, esta tem que ocorrer entre pares de estações, não podendo haver comunicação *multicast*.

3.2.1.1 - Autenticação por sistema aberto - *Open system*

Este método é adotado como padrão pelo IEEE 802.11, disponibilizando um algoritmo simples, sendo um algoritmo de autenticação nula. A autenticação do tipo *Open System* foi desenvolvida focando redes que não precisam de segurança para autenticidade de dispositivos. Nenhuma informação sigilosa deve trafegar nestas redes já que não existe qualquer proteção. Uma estação é aceita na rede, simplesmente solicitando uma requisição (VASCONCELOS, 2005).

3.2.1.2 - Autenticação com chave compartilhada – *Shared Key*

Este algoritmo de autenticação é baseado em mecanismo de criptografia, utilizando a opção WEP para criptografar e decriptografar, dependente de ambas as estações terem uma cópia da chave WEP, sendo o administrador da rede responsável por isso.

3.2.2 - Privacidade – Uso do WEP

As redes sem fio devem prover à mesma privacidade equivalente às redes com fio, não permitindo o acesso de intrusos, que poderão capturar ou modificar os pacotes que estão sendo transferidos. O Padrão IEEE 802.11b, tenta aplicar alguns mecanismos de segurança para que os dados que trafegam na rede possam obter sua confidencialidade e integridade desejada, sendo o WEP o protocolo mais popular responsável pela execução.

3.2.2.1 - Características do protocolo WEP

O protocolo WEP (*Wired Equivalent Privacy*) é um protocolo de segurança do nível de enlace dos dados, para redes sem fio que surgiu para garantir um nível de segurança equivalente ao existente nas redes cabeadas, tentando prover confidencialidade, integridade dos dados e controle de acesso à rede (VASCONCELOS, 2005).

A confidencialidade tenta garantir que o protocolo irá impedir que intrusos consigam ler, inserir ou modificar dados que estão sendo transmitidos. Em relação à integridade dos dados o protocolo faz com que a mensagem chegue até o seu destino sem alterações, implementando uma função linear chamada *checksum,* garantindo assim que o conteúdo da mensagem transmitida seja protegido e inalterado. O controle de acesso à rede sem fio impede a entrada de intrusos na rede, garantindo a autenticidade.

Com a escolha do WEP, todos os outros pacotes que não o estejam utilizando podem ser descartados, garantindo assim que somente aqueles que tenham a chave de criptografia WEP façam parte da comunicação (JUNIOR et al, 2004).

Como qualquer outro protocolo o WEP também possui suas falhas, porém seu uso é essencial para garantir uma maior segurança das redes. Ele é constituído de um algoritmo simétrico utilizando chaves compartilhadas, que devem ser as mesmas tanto no cliente quanto no ponto de acesso. Essas chaves criptográficas simétricas são chamadas de chaves WEP (JUNIOR et al, 2004).

Uma vez que as chaves são distribuídas, não sendo especificadas pelo padrão como, elas podem ser pré-carregadas pelo fabricante, trocadas pela rede fisicamente conectada ou a estação-base escolhe uma chave aleatoriamente e envia à outra máquina codificada com a chave pública da outra máquina. Como essas chaves permanecem estáveis por um longo período o padrão WEP recomenda que o vetor de inicialização seja modificado em cada

pacote, com o objetivo de evitar ataques de reutilização de fluxo de chaves (TANENBAUM, 2003).

Para codificar os pacotes que serão trocados numa rede de comunicação sem fio o WEP é baseado na implementação do método criptográfico RC4 da RSA projetado por Ronald Rivest em 1987, que é um algoritmo de fluxo, ou seja, criptografa os dados ao mesmo tempo em que são transmitidos (JUNIOR et al, 2004).

O WEP contém uma chave secreta de 40 ou 104 *bits* e um vetor de inicialização (VI) público de 24 *bits* (MARTINS, 2003). Esse vetor de inicialização somado com a chave WEP de 40 ou 104 *bits* forma uma chave RC4 de 64 ou 128 *bits*, que é usada para criptografar os dados (VASCONCELOS, 2005). Na Figura 12 está ilustrada a composição da chave WEP.

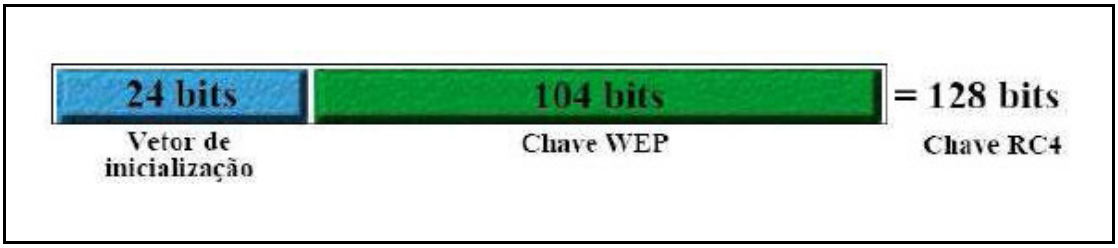

Figura 12 - Composição da chave WEP.
Fonte: (VASCONCELOS, 2005)

O WEP se encarrega de cifrar todos os dados que são transmitidos via rede. O padrão WEP de 64 bits é compatível com todo produto que siga o padrão *WI-FI (Wireless Fidelity)*, ou seja, todos os produtos comercializados atualmente.

Já para utilizar o padrão 128 *bits*, faz-se necessário que todos os componentes da rede suportem esse padrão, caso contrário os equipamentos que não possuem suporte ficarão fora da rede (TOLARI, ZANARDI, 2005). Além da chave gerada de 64 ou 128 *bits* o WEP utiliza CRC-32 para calcular o *checksum* da mensagem, que é incluso no pacote, para garantir a integridade dos dados.

3.2.2.2 - Criptografia do protocolo WEP

O processo de cifragem inicia-se com uma chave secreta que é distribuída entre as estações por um administrador de rede ou por um serviço de gerenciamento de chaves. Esse administrador irá digitar a chave secreta de 40 ou 104 *bits* em todos os usuários e todos os pontos de acesso (PA) da rede.

O WEP é um algoritmo simétrico, no qual a mesma chave é usada para cifragem e decifragem. É aplicado no arquivo que se deseja transmitir dois processos, um para realizar a cifragem dos dados e outro tem o objetivo de evitar qualquer alteração no arquivo enquanto ele é enviado. Como visto anteriormente a chave composta é formada pela concatenação do IV que é gerado a cada pacote a ser transmitido e pode assumir 16.777.216 valores válidos (ou 2^{24}), com a chave secreta, caracterizando o *stream Cipher* no algoritmo RC4. Esse *stream Cipher* gera uma mudança de chave para cada mensagem a ser criptografada pelo RC4, tendo como resultado uma chave composta diferente para os dados (MATTOS, 2005).

Essa chave composta criada é inserida no algoritmo gerador de número pseudo-randômico PRNG (*Pseudo-random Number Generator*), que é baseado no RC4. O PRNG irá gerar um conjunto de *bits* aleatório que será utilizado para cifrar o arquivo através de uma operação XOR, o que não mudará o tamanho do arquivo original. No início deste resultado é concatenado o vetor de inicialização, e no final 32 *bits*, que é o resultado de um processo *ICV (Integrity Check Calue)*, através da análise da carga útil que é verificada usando-se o polinômio CRC-32, que irá proteger os dados contra modificações inesperadas (MATTOS, 2005). A saída de todo esse processo é a mensagem contendo o endereço MAC e o IV, ambos não cifrados, mais o texto cifrado. A Figura 13 ilustra todo esse processo de cifragem.

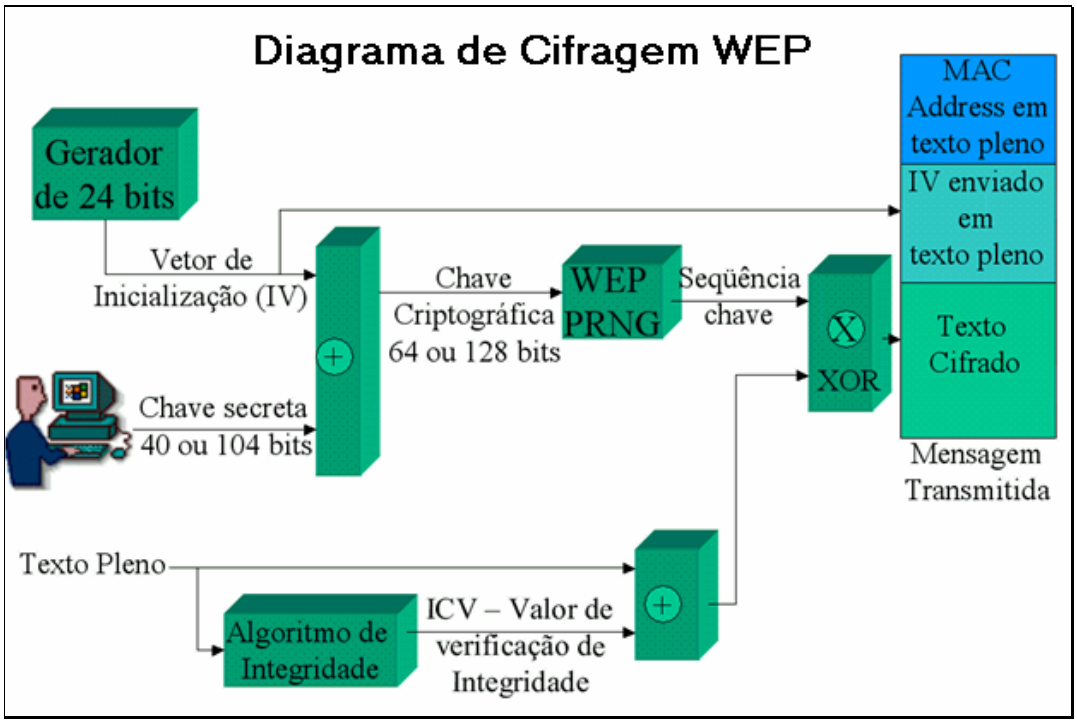

Figura 13 - Diagrama de Cifragem do WEP.
Fonte: (TOLARI, ZANARDI, 2005).

Quando o destinatário recebe a mensagem, ele já tem o conhecimento da chave secreta. Ele utiliza o IV do início do pacote para criar a mesma *string* do PRNG e assim decifrar o arquivo. Depois do arquivo já estar decifrado é executado o CRC-32 que irá gerar uma ICV, sendo comparado com o ICV transmitido, onde se houver alguma diferença quer dizer que os dados sofreram modificação durante a transmissão, e assim uma indicação de erro é enviada e a mensagem deverá ser descartada, caso contrário à mensagem é aceita (MATTOS, 2005).

Na Figura 14 tem-se o processo de decifragem efetuado pelo protocolo. Essa abordagem aparentemente parece boa, porém já foi rompida por alguns métodos criados, de modo que possui falhas.

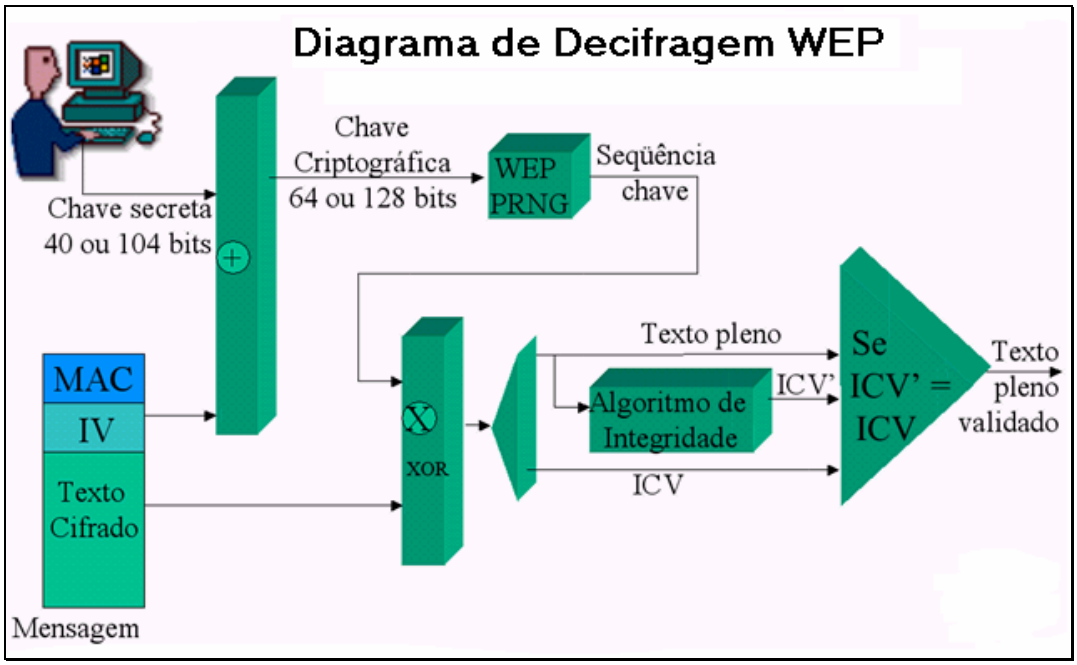

Figura 14 - Diagrama de Decifragem do WEP.
Fonte: (TOLARI, ZANARDI, 2005).

3.2.2.3 - Problemas do protocolo WEP

Há muitas falhas, já comprovadas, com relação ao protocolo WEP. Muitos lugares e instalações utilizam a mesma chave compartilhada para todos os usuários, permitindo assim a leitura de todo o tráfego por todos os usuários. Mesmo que cada usuário possuísse uma chave distinta, o protocolo ainda teria vulnerabilidades e poderia ser atacado, pois as chaves são estáveis por longos períodos, acontecendo o ataque de reutilização de fluxo de chaves se não for variado o IV (TANENBAUM, 2003).

Chaves com valores do IV arbitrários podem demorar algumas dezenas de minutos a mais para serem quebradas. Entretanto, a dimensão desse vetor de iniciação é um problema, pois é muito curta, além de ser alterado a cada pacote enviado, indo de zero até seu valor máximo (16.777.216 números diferentes), causando assim a reutilização desse vetor e podendo-se calcular quanto tempo vai demorar até que o vetor se repita (MATTOS, 2005).

O RC4 apresenta chaves fracas que permitem chegar à chave de encriptação, para isso, basta escutar suficientemente o tráfego. Este defeito é utilizado pelos *softwares* de *cracking* (*Aisnort* e *Wepcrack*, entre outros) disponíveis na *Web*. O WEP possui ainda outra falha de peso que é o sistema de autenticação por pacotes. Como se baseia numa assinatura do pacote por segmentação linear, é fácil deduzir um pacote forjado partindo de um pacote encriptado e bem formado.

Outro defeito do protocolo de segurança WEP, conforme apresentado por Vasconcelos (2005), é a função detectora de erros CRC-32 que é uma função linear e que não possui chave, duas característica que fazem com que seja suscetível a ataques.

A primeira característica permite que ocorra modificação na mensagem, que eventualmente foram capturadas na transmissão, sem que o receptor final perceba. A segunda característica permite que se descubra uma seqüência secreta RC4, podendo ser feito uma autenticação na rede e introduzir mensagens clandestinas (VASCONCELOS, 2005).

Conforme visto, o protocolo de segurança WEP não é seguro o bastante para deixar uma rede sem fio em um estado confortável no quesito segurança. Como tentativa para melhorar o protocolo WEP, pesquisas foram realizadas e sugestões foram estudadas e testadas (JUNIOR et al, 2004). Como solução imediata foi desenvolvido o padrão de segurança conhecido como WPA (*Wi-Fi Protected Access*). Há ainda outros métodos de segurança que são usados junto com WEP para prover uma melhor segurança nas redes sem fio.

3.2.2.4 - Outros protocolos e métodos de segurança

O WPA (*Wi-Fi Protected Access*) é um outro protocolo, chamado de WEP2, que foi desenvolvido pela necessidade de se obter uma maior segurança nas redes sem fio, melhorando as vulnerabilidades do WEP. Este protocolo, diferentemente do WEP, tem o

propósito de ter uma constante troca da chave criptográfica dificultando assim a invasão ou a descoberta da chave. Essa vantagem ocorre pela melhoria na criptografia dos dados, que usa um protocolo de chave temporária chamada TKIP (*Temporal Key Integrity Protocol*), que faz a criação dinâmica das chaves por quadro, tendo um mecanismo de distribuição de chaves.

As chaves possuem 128 *bits* e o vetor de inicialização, possui 48 *bits*, diferente do WEP, que tem 24 *bits*. O protocolo TKIP de certa forma veio a substituir o WEP com um forte algoritmo de criptografia, usando também padrão do RC4, porém com outras técnicas com o objetivo de suprir a fragilidade do WEP.

Outra vantagem do WPA é em relação a autenticação de usuários, que passa a ser obrigatória ao uso do 802.1x e do EAP (*Extensible Authentication Protocol*), para fazer a autenticação de cada usuário que entrar na rede. O EAP é um modelo para autenticação imposta no nível de usuário, utilizando o modelo 802.1x (CARVALHO FILHO, 2005).

Outro protocolo para prover segurança em redes sem fio é WPA2, que possui técnicas eficazes de autenticação e encriptação, como o AES (*Advanced Encryption Standard*). O WPA2 tem compatibilidade com WAP, suportando assim o TKIP e AES, garantindo um nível de privacidade de dados exigido por muitos órgãos governamentais.

Esses protocolos surgiram com intuito de aliviar as preocupações com segurança, entretanto, os *hackers* também desenvolvem ferramentas para a quebra da segurança, assim, é recomendável a implementação de outras técnicas para ajudar na segurança.

Umas das várias técnicas que pode ser usada para melhoria da segurança em redes sem fio é o uso do filtro de endereços MAC. Cada equipamento da rede possui um identificador único chamado endereço físico ou endereço MAC. Os pontos de acesso ou roteadores registram esses endereços de todos os equipamentos que se conectam a ele. Se um dispositivo cliente tentar se conectar ao ponto de acesso ou roteador e o seu endereço não

estiver registrado na lista de endereços MAC, as funcionalidades do filtro MAC não deixarão ele se conectar.

Filtro MAC é uma boa solução para pequenas empresas e residências, ficando inviável a utilização para grandes empresas visto que o administrador da rede terá que digitar todos os endereços MAC das placas sem fio da empresa (TOLARI, ZANARDI, 2005).

Um outro método que ajuda a manter a segurança em redes sem fio é o VPN (*Virtual Private Network*) ou Rede Privada Virtual. A VPN protege uma WLAN criando um túnel que simula uma conexão ponto-a-ponto, fazendo os dados ficarem isolados de acesso não autorizado ou de outras redes, além do dado ser encapsulado e cifrado. Esta tecnologia possibilita o tráfego de várias fontes, via diferentes túneis sobre a mesma infra-estrutura, também permite que diferentes protocolos de rede se comuniquem através de uma infra-estrutura incompatível. Conforme Tolari e Zanardi (2005) uma VPN fornece proteção de privacidade, integridade, autenticação dos dados de origem, proteção contra *Replay*, proteção contra análise de tráfego.

Há várias outras técnicas que são utilizadas para fornecer uma maior segurança nas redes sem fio, porém estudar todas essas técnicas não é relevante para este trabalho.

3.3 - Considerações Finais

Em todos os meios em que se usam redes sem fio o uso de uma política de segurança eficaz é imprescindível, pois há a necessidade de diminuir as vulnerabilidades e os acessos indevidos à rede. Com o que foi apresentado neste capítulo, é possível entender como os protocolos de segurança nas redes sem fio são importantes, mas que ainda são suscetíveis à falhas.

Foi analisado mais especificamente o protocolo WEP que, apesar de apresentar vulnerabilidades, ainda é um protocolo amplamente utilizado. Para tal avaliação serão efetuadas simulações, utilizando uma ferramenta de simulação chamada *Network Simulator* (NS-2). No próximo capítulo serão abordados alguns simuladores existentes e dar-se-á maior ênfase ao *Network Simulator*.

CAPÍTULO 4 - SIMULADORES

Simuladores são muito úteis para implementar algo difícil de executar em plataformas práticas ou que possui restrições orçamentárias. Neste trabalho a simulação será utilizada como ferramenta de avaliação de desempenho.

4.1 - Considerações Iniciais

A vantagem de se utilizar ambientes de simulação, é que estes permitem o estudo e avaliação de diversas tecnologias incluindo redes com custo reduzido, desempenhando também um papel muito importante quando se pretende estudar cenários que muitas vezes são complicados de implementar em plataformas práticas, ou quando se pretende avaliar o desempenho de um projeto inexistente.

Neste capítulo foi feito o estudo de alguns simuladores existentes no mercado atualmente, além de descrever o que é simular e o porquê simular. Será dado um enfoque maior ao simulador utilizado por este projeto: o *Network Simulator.*

4.2 - Simulação

Geralmente usa-se a simulação de sistemas quando não é possível realizar experimentos no sistema real devido a adversidades, como o tempo necessário para a realização do experimento, o alto custo ou a dificuldade de realizar fisicamente o experimento. Soluções viáveis e possíveis para resolução de alguns problemas em redes de computadores podem ser simuladas com objetivo de verificar sua funcionalidade. Através da

simulação é possível reproduzir a realidade da rede, possibilitando inserir, excluir ou modificar componentes da rede simulada (SALES, 2004).

Na maioria as propostas de segurança para sistemas móveis sem fio são verificadas através de simuladores com suporte a ambientes de rede sem fio. Após a execução de uma simulação é muito importante a análise dos resultados para gerar conclusões a respeito do ambiente e da plataforma simulada (SALES, 2004).

4.2.1 - O que é simular?

Segundo o conceito que está no dicionário (AURÉLIO, 2006), simular é uma imitação, representar com semelhança, fingir, aparentar. No contexto deste projeto a simulação é uma implementação de um modelo abstrato do mundo, no caso uma rede sem fio *ad-hoc* que se deseja estudar e analisar o uso de um protocolo chamado WEP (BORISOV et al, 2006). A simulação foi efetuada em uma linguagem específica compatível com o simulador utilizado, a Otcl (OTCL, 2006) e o simulador *Network Simulator*. A simulação permite abordar procedimentos de emergência e avarias, a fim de monitorar, analisar e examinar as reações do cenário em que esta sendo efetuada a análise.

No contexto deste trabalho a simulação não pretende reproduzir a realidade em sua totalidade, mas servir como uma preparação para o enfrentamento da mesma. Ela pode ser usada para construir ambientes que tentam imitar algum possível cenário futuro, mas certamente de maneira muito limitada.

4.2.2 - Por que simular?

Compreender o funcionamento da segurança em redes sem fio baseado no protocolo WEP e realizar estudos de avaliação de desempenho baseados em experimentos reais são atividades de grande complexidade. Além disso, nem sempre é possível ter acesso a redes e equipamentos, e essas experiências práticas geralmente têm um custo alto. Por esse motivo, simulação é uma técnica utilizada com muita freqüência, pela flexibilidade em testar cenários variados, incluindo o comportamento de protocolos e novas tecnologias e efeito de diferentes topologias (GAONA,1995).

Um importante papel da simulação de redes é desenvolver a intuição das pessoas sobre o comportamento de aplicações, protocolos e tecnologias de rede. Isso é muito útil para pesquisadores, professores, alunos e até para profissionais que trabalham com o projeto de novas redes, onde podem convencer seus clientes, mostrando a eles como a nova rede vai se comportar (GAONA,1995).

Portanto a simulação é uma ferramenta importante e útil para testar o comportamento de componentes da rede. Existem vários simuladores de redes de computadores presentes na literatura, e alguns deles serão analisados a seguir.

4.3 - Simuladores mais populares

Vários simuladores têm surgido nos últimos anos para permitir a avaliação de cenários de redes de computadores antes da sua implantação. Foram analisados a seguir os simuladores de redes mais utilizados no meio acadêmico. Uma maior ênfase foi dada ao simulador *Network Simulator* (NS-2), pois o mesmo será utilizado para fazer a parte de implementação desde projeto.

4.3.1 - Simulador OMNeT++

A ferramenta de simulação OMNeT++ (*Objective Modular Network Testbed in C++*) tem um domínio público e é implementada em C++, para a simulação de eventos discretos orientados a objeto. Ela é baseada em componentes e oferece interface gráfica e animação.

Apesar de ser indicada para a simulação de redes de comunicação nos seus vários aspectos, como modelagem de tráfego, modelagem de protocolos entre outros, pela sua flexibilidade pode ser usada para modelagem de sistemas distribuídos de *hardware* ou multiprocessadores e modelagens de quaisquer sistemas que possa ser simulado baseando-se em eventos discretos.

É completamente portável podendo ser instalado em sistemas Unix, Linux e *Windows* e está transformando-se rapidamente em uma plataforma popular de simulação na comunidade científica. A OMNeT++ possui também uma biblioteca de funções úteis para o tratamento estatístico das amostras recolhidas, bem como estruturas de dados suplementares (OMNeT, 2006).

Por possuir uma interface gráfica, através da mesma os detalhes internos dos módulos são expostos ao usuário, permite o controle do modelo durante a simulação. Módulos aninhados compõem o modelo OMNeT++, que permite ao usuário refletir a estrutura lógica do sistema real. Os componentes (módulos) são programados em C++ e posteriormente montados em componentes e em modelos maiores usando uma linguagem de alto nível (NED).

OMNeT++ foi desenvolvida por András Varga na universidade técnica de Budapeste, departamento de telecomunicações (BME-HIT) (OMNeT, 2006).

4.3.2 - OPNET

OpNet (OpNet,2006) é um pacote de produtos que permite projetar, desenvolver, gerenciar e simular a infra-estrutura, os equipamentos e as aplicações de uma rede de computadores. Ele também oferece editor gráfico e animação da simulação. Essa ferramenta permite a especificação de um grande número de componentes de comunicação via satélite, redes sem fio, LANs, dentre outros.

Ao contrário do NS-2, como será visto na seção 4.3.5, essa ferramenta é comercial com um custo elevado, tornando inviável a sua utilização neste projeto.

O OpNet é um pacote de simulação que possui várias ferramentas de simulação, análise de performance de comunicação de redes e especificação. Este *software* pode ser usado para análise de desempenho de uma rede já implantada ou para redes criadas no próprio *software*, ele dispõe de recursos para captura de tráfegos numa rede variando o número de usuários que acessam serviços diferenciados (OpNet,2006). Na Figura 15 são ilustradas as etapas do processo de modelagem e simulação usando o OpNet.

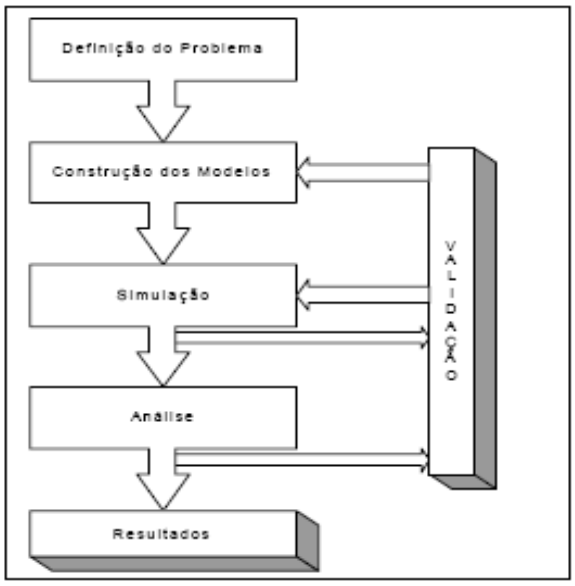

Figura 15 - Etapas da modelagem e simulação no OpNet.
Fonte: http://www.inf.pucpcaldas.br/~joao/materiaispesquisa/Relatorio%20Tecnico%20Opnet.pdf

4.3.3 - GloMoSim

O GloMoSim (*Global Mobile Information Systems Simulation Library*) é um simulador de redes sem fio baseado em uma biblioteca de funções que quando compilada e utilizada permite a simulação de forma escalável, aproveitando assim a capacidade de simulação de eventos paralelos fornecido pela linguagem PARSEC, o que permite o processamento paralelo (GLOMOSIM 2003). Para executar a biblioteca de funções do GloMoSim é necessário um compilador C como o GCC (GNU C *Compiler*) e uma versão do compilador PARSEC o PCC (UCLA, 2001), que atualmente já vem incluído no pacote de distribuição do GloMoSim.

Este simulador foi desenvolvido na UCLA (*University of California*) (UCLA, 2001), e é usado para realizar simulações de redes de diferentes tamanhos, permitindo a comunicação de nós em diversos cenários. É um simulador paralelo para redes móveis baseado em uma biblioteca modular implementada usando PARSEC (*PARallel Simulation Environment for Complex systems*). Esta linguagem de simulação é baseada em C para descrição de simulações seqüenciais e paralelas (GLOMOSIM 2003).

O GloMoSim implementa um conjunto de protocolos de rede para comunicação sem fio, que são organizados em uma arquitetura em camadas. Novos protocolos e módulos implementados em PARSEC podem ser facilmente adicionados à biblioteca do GloMoSim para compor diferentes simulações. O núcleo do PARSEC implementa uma máquina de simulação de alto desempenho, que permite ao GloMoSim fazer a simulação de redes de larga escala com total transparência de simulação, tanto para o programador do protocolo quanto para o usuário do simulador.

Dessa maneira o GloMoSim é um simulador que tem foco específico para simulação de redes de computação móvel de larga escala. Esse simulador está ainda em

desenvolvimento, mas o GloMoSim é a base de um produto comercial denominado QualNet (QualNet, 2006), que ao contrário do GloMoSim é pago. Este produto comercial possui suporte a mais recursos e suas variáveis permitem maiores configurações (GLOMOSIM 2003).

4.3.4 - Boson NetSim

O Boson é um simulador já utilizado há vários anos, por esse motivo tem bases sólidas, além de possuir uma rede global de escritores e de instrutores, que são profissionais capacitados em oferecer três linhas de produto para os profissionais que trabalham para a certificação da Cisco: *Software* de ExSim, *software* de NetSim, e treinamento de sala de aula (BOSON,2006).

O Boson NetSim emula tabelas de chaveamento de pontes bem como tabelas de protocolo de roteamento para permitir ao usuário ir além dos exemplos do simulador. Com o Boson NetSim é possível projetar e configurar uma rede com 40 modelos diferentes de roteadores.

Este simulador de rede é versátil e realístico. Outros produtos simulam *scripts* para um usuário experiente, sem simular o que esta acontecendo realmente dentro da rede. A tecnologia virtual do pacote do Boson cria os pacotes individuais que são distribuídos e comutados através da rede simulada, permitindo que o Boson NetSim construa uma tabela de roteamento virtual apropriada, e simule o trabalho em rede real.

O Boson oferece dois produtos focalizados na certificação Cisco, NetSim para CCNA e NetSim para CCNP. Cada um suporta as tecnologias e as habilidades que um usuário necessitará para as respectivas certificações. O Boson NetSim inclui também um menu detalhado do laboratório que contém as lições e os laboratórios que cobrem protocolos de

roteamento, dispositivos de Cisco, *Switching*, projeto topológico, entre outros (BOSON,2006).

4.3.5 - Network Simulator (NS-2)

O NS-2 é um simulador que tem sido utilizado com grande freqüência em pesquisas em redes de computadores sempre com suporte e apoio de várias organizações acadêmicas e comerciais, pois ele é um *software* de código livre e fornecido gratuitamente (NS-2, 2006).

4.3.5.1 - Histórico do NS-2

O NS-2 teve origem no ano de 1989 a partir de uma variação do REAL *Network Simulator* (KESHAV, 1997), um projeto da Cornell University, EUA, que por sua vez derivou do NEST (*Network Simulation Testbed*) da universidade de Columbia, NY. REAL foi desenvolvido por S. Keshav da Universidade de Cornell com o intuito de ser uma ferramenta para o estudo do comportamento dinâmico do controle de fluxo e congestionamento em redes comutadas por pacotes.

O simulador NS-2 foi criado originalmente para o estudo do desempenho do protocolo TCP, e desde então tem sido estendido de modo a suportar trabalhos em várias áreas. O NS-2 têm evoluído sempre com suporte e apoio de várias organizações acadêmicas e comerciais. Atualmente o seu desenvolvimento é suportado pelo DARPA (*Defense Advanced Research Projects Agency*, EUA) através do projeto SAMAN (HEIDEMANN et al, 2001) e pela NSF (*National Science Foundation*, EUA) através do projeto CONSER (CONSER, 2002), em colaboração com outros pesquisadores como o centro ICIR (ICIR, 2006).

O simulador já recebeu apoio do *Lawrence Berkeley National Laboratory*, do Xerox PARC (*Palo Alto Research Center*), da Universidade da Califórnia em Berkeley, *Sun Microsystems* e também agrega diversos módulos construídos por pesquisadores independentes (VINT, 2006). Esses módulos que podem ser adicionados por pesquisadores independentes devido ao fato de o simulador possuir código livre e gratuito o torna mais vantajoso em relação aos demais, pois o usuário pode adequar o simulador para quaisquer necessidades de uso.

Há também uma lista de discussão, mantida pelos desenvolvedores do NS-2, onde os pesquisadores de vários lugares do mundo podem interagir e trocar experiências, e também propõem correções para o código do simulador, que depois de devidamente avaliadas podem ser incorporadas.

4.3.5.2 - Característica do NS-2

O NS-2 foi desenvolvido para várias plataformas computacionais, sendo possível instalar o pacote em várias plataformas Unix como, FreeBSD, Linux, SunOS e Solaris, além da plataforma Windows.

Este simulador é baseado em eventos discretos e orientado a objetos escrito em C++ e Otcl (*Object Tool Control Language*) para simulação de redes de computadores (NS-2, 2006). A idéia de se utilizar essas duas linguagens de programação advém da necessidade de se ter um ambiente que permita eficientemente desenvolver protocolos e manipular estruturas de dados, no caso utilizando a linguagem C++, e também possibilite a configuração rápida dos parâmetros da simulação utilizando a linguagem Otcl (SALES, 2004). No item 4.3.5.3 é explicado melhor o uso das duas linguagens no NS-2.

O código fonte básico do NS-2 foi desenvolvido em C++, mas grande parte dos módulos utilizados para tecnologias específicas de rede foi desenvolvida em Otcl. A Figura 16 ilustra a arquitetura geral do NS-2.

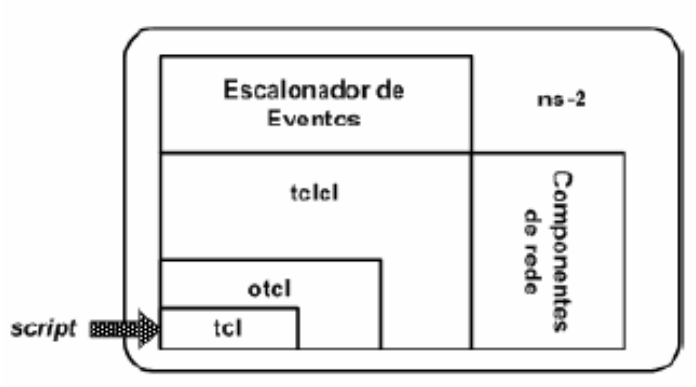

Figura 16 - Arquitetura do NS-2
Fonte: http://www.geocities.com/locksmithone/articles/networksimulator-sepa.pdf.

A Figura 17 mostra a árvore parcial de hierarquia de classes do NS-2. Os objetos de rede são criados segundo essas classes.

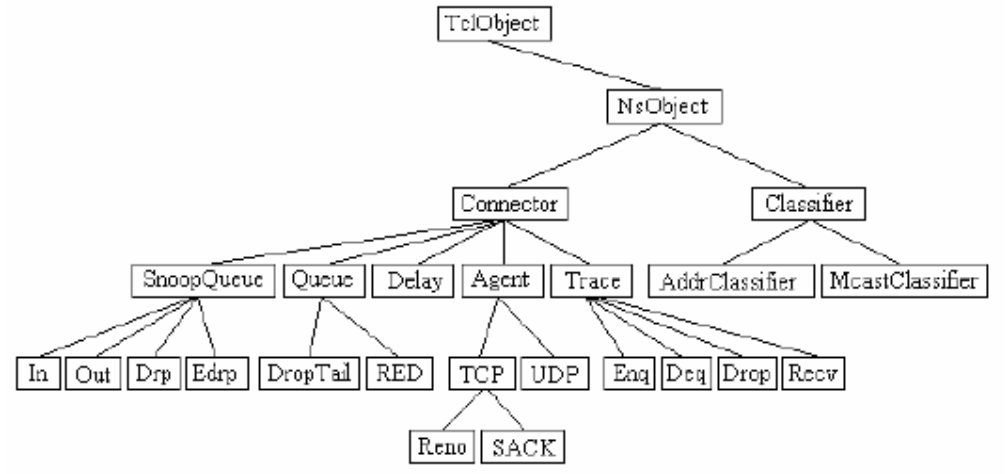

Figura 17 - Hierarquia de classes parcial NS-2
Fonte: http://www.geocities.com/locksmithone/articles/networksimulator-sepa.pdf

A biblioteca de protocolos e mecanismos implementados no NS-2 é bastante vasta, abrangendo implementação dos protocolos TCP, UDP, IP além de disciplinas de

serviços,como, WFQ (*Weight Fair Queueing*), protocolos para redes móveis, como o IP móvel, tecnologias de redes sem fio locais e de longa distância como o 802.11, *Bluetooth* e GPRS (*General Packet Radio Service*). Alguns destes não são distribuídos diretamente no pacote do NS-2, sendo contribuições disponibilizadas como *patchs* na página que hospeda as informações sobre o NS-2 e que podem ser baixadas e adicionadas ao módulo básico através de re-compilação do núcleo do simulador.

O NS-2 fornece também bibliotecas de funções para a geração de alguns tipos de tráfego como: CBR (*Constant Bit Rate*) utilizado para simular tráfego constante e voz, ON-OFF para tráfego em rajada e voz comprimida, FTP para gerar tráfego correspondente a aplicações de transferência de arquivos e VBR (*Variable Bit Rate*) para tráfego com taxa de dados variável. Além das bibliotecas com os módulos específicos de protocolos, tecnologias e geração de tráfego, o NS-2 possui funções específicas de simulação e geração de números aleatórios (FALL e VARADHAN, 2006).

O NS-2 pode ainda ser utilizado como um emulador de rede, o qual é capaz de interagir com uma rede real. Ele inclui um gerador de cenários, o qual permite a geração automática de diferentes topologias de redes, padrões de tráfego e de falhas.

Para as redes *wireless* o NS-2 implementa um modelo que consiste essencialmente no *MobileNode* (mobilenode.h), com características adicionais que permite simulações de redes *ad hoc multi-hop*, LANs *wireless* entre outras. Um nó móvel é derivado do nó básico o *Node* (node.h) com funcionalidades adicionadas de um nó *wireless* e mobilidade como a habilidade de mover-se dentro de uma dada topologia e habilidade de receber e transmitir sinais a partir de um canal *wireless*. A diferença principal entre eles é que um *host* móvel não está conectado por meio de *links* (cabos) a outros *hosts* ou *host* móvel (NS-2, 2006).

Na criação de um objeto *MobileNode* é especificado a criação de um agente de roteamento, cria-se a pilha de rede consistindo de uma camada de enlace (*link layer*), camada

mac e uma interface de rede com uma antena, interconectando estes componentes e conectando a pilha ao canal de comunicação. O *MobileNode* está ilustrado na Figura 18 .

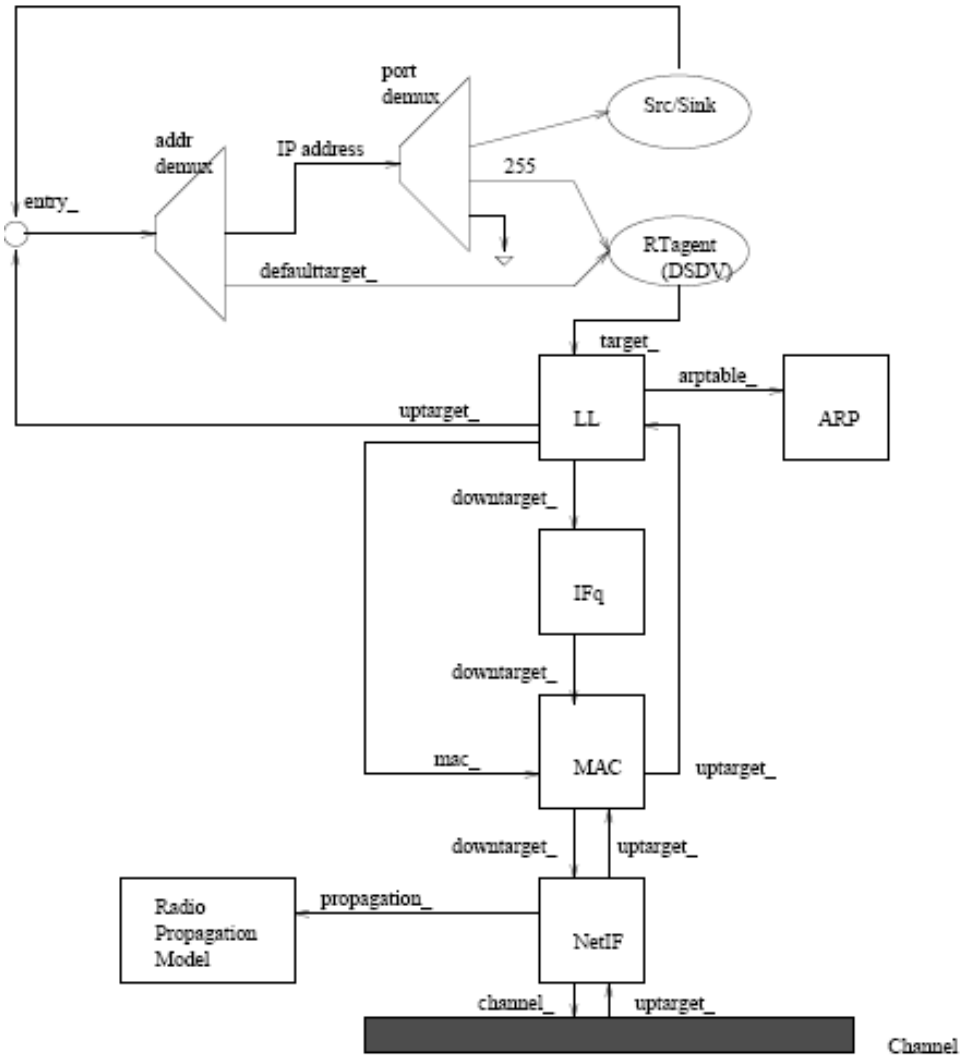

Figura 18 - Esquema de um *Mobile Node*
Fonte: (FALL e VARADHAN, 2006).

Para simulação de redes *wireless* existem quatro protocolos de roteamento: DSDV, DSR, AODV e TORA. Para a subcamada MAC existem duas opções: IEEE 802.11 ou TDMA, enquanto que para a camada de transporte estão disponíveis os protocolos TCP e UDP. É possível definir ainda qual é o tipo de antena utilizada, como omnidirecional entre

outras, e também qual é o consumo de energia, e o alcance do sinal de rádio de cada nó móvel.

Os nós móveis são dispostos, através de coordenadas cartesianas, em um espaço tridimensional delimitado e gradeado, sendo que a resolução da grade pode ser ajustada. Para movimentar um nó móvel neste espaço são informadas as coordenadas de seu destino e a sua velocidade até este destino (FALL e VARADHAN, 2006).

4.3.5.3 - Modelo de programação do NS-2

Ao invés de adotar uma única linguagem de programação, o projeto do simulador levou em consideração o fato de que diferentes simulações requerem diferentes modelos de programação. Portanto, o NS-2 adota duas linguagens de programação: C++ para o núcleo do simulador (*back-end*) e Otcl (OTCL, 2006) para construção de *scripts* e modelagem da simulação (*front-end*). O objetivo é prover flexibilidade sem prejudicar o desempenho (FALL e VARADHAN, 2006).

A estrutura do código do NS-2 está dividida com base no seu nível de processamento, isto é, se são funções, procedimentos, classes, que necessitem de elevado processamento deverão utilizar a linguagem C++. A linguagem C++ é utilizada por possuir grande desempenho, tornando-se ideal para a implementação de protocolos detalhados que utilizem grandes conjuntos de dados. Porém a linguagem C++ não é adequada para implementar as mudanças rápidas dos parâmetros das simulações, pois seu processo de compilação, retirada de erros, e execução é lento (FALL e VARADHAN, 2006).

Por isso se o objetivo é desenvolver algo que necessite de constante aperfeiçoamento e pouco processamento computacional, deverá ser desenvolvido em OTcl (OTCL, 2006), que é

uma linguagem baseada em *scripts*, flexibilizando e facilitando as mudanças dos parâmetros de simulação (FALL e VARADHAN, 2006).

OTcl foi desenvolvida pelo MIT (*Massachusetts Institute Of Technology*) , e é uma linguagem interpretada e interativa, uma vez que seus programas podem ser alterados de maneira rápida e ser facilmente re-executados. OTcl atua como interface para o usuário (as simulações são escritas em OTcl), permitindo a manipulação de parâmetros e configuração. No entanto, por ser uma linguagem interpretada, Otcl é bem mais lenta que C++ (SALES, 2004).

A implementação de novos mecanismos e protocolos em C++ é possível porque praticamente todo objeto C++ tem um correspondente Otcl e vice-versa. Pode-se implementar um novo protocolo através de herança de classes existentes em C++ e depois ligar o novo objeto C++ a um objeto Otcl correspondente. O novo objeto Otcl, resultado da ligação citada anteriormente, poderá então ser invocado a partir de *scripts* Otcl. Assim, será possível executar as novas funções implementadas em C++ através do desenvolvimento de *scripts* de simulação em Otcl. Estes *scripts* conterão uma chamada a um objeto Otcl ligado a um objeto C++ incluído pelo usuário no *back-end* do NS-2. Além da possibilidade de chamar funções em C++ a partir do Otcl pode-se também chamar funções definidas em Otcl no C++ (FALL e VARADHAN, 2006).

A junção entre a linguagem C++ e Otcl é realizado através da linguagem Tclcl, que é um conjunto de módulos específicos que acompanha o NS-2. Através de tclcl, uma classe escrita em C++ pode ser instanciada usando-se código Otcl, e qualquer parâmetro modificado nesse código Otcl será refletido no objeto C++ instanciado (FALL e VARADHAN, 2006). A Figura 19 ilustra o uso das duas linguagens dentro do contexto do NS-2.

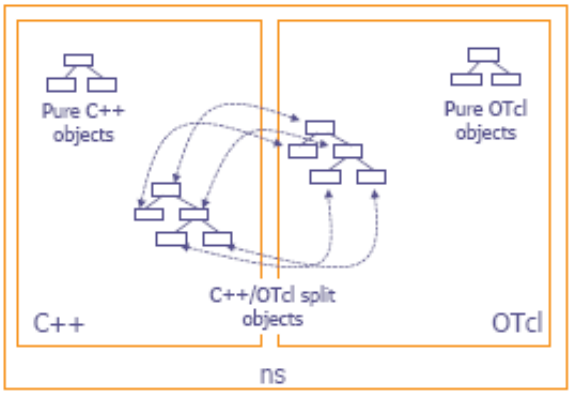

Figura 19 - Linguagem C++ e OTCL no NS-2
Fonte: http://www.gta.ufrj.br/~rezende/cursos/cpe710/

4.3.5.4 - Ferramentas de análise de resultados - NAM e Xgraph

O NAM (*Network Animator*) (FALL e VARADHAN, 2006) é um animador de redes que acompanha o NS-2, utilizado para compreender o que ocorre durante a simulação. Através do NAM pode-se visualizar a topologia da rede, bem como acompanhar o fluxo dos pacotes e seus conteúdos.

Durante a simulação, o NS-2 gera um ou mais arquivos de *trace* que contêm dados detalhados da simulação, para visualização posterior. A criação destes arquivos é opcional e depende da adição de alguns comandos no *script* de simulação Otcl. Esses comandos são detalhados no Capítulo 5 na seção 5.3 que aborda o desenvolvimento de um *script* Otcl. No final da simulação, o NAM pode ser acionado explicitamente ao *script* para interpretar o arquivo de *trace* e mostrar a animação da simulação.

O arquivo de *trace* do NAM contém todas as informações necessárias para animação, tanto do *layout* estático da rede quanto eventos dinâmicos, como saída e chegada de pacotes e quedas de enlaces. A

Figura 20 ilustra uma simulação visualizada no NAM de dois nós móveis caracterizando uma rede *wireless ad-hoc* com textos explicativos nas suas funções.

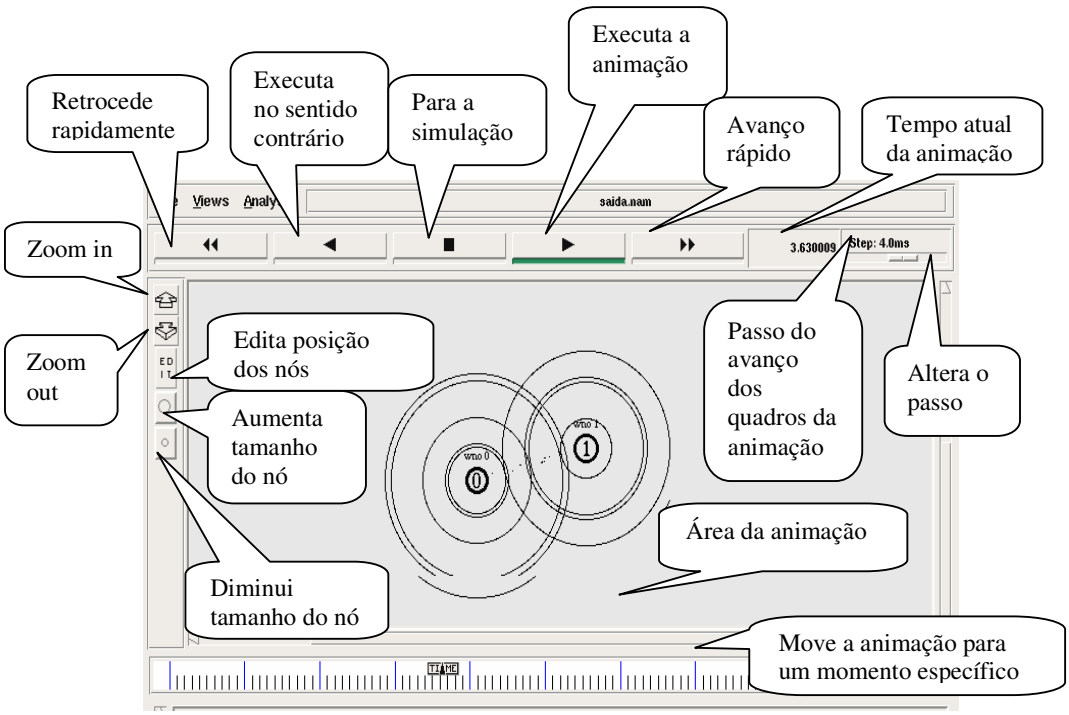

Figura 20 - NAM – *Network Animator*

Para um melhor controle por parte do usuário, o NAM proporciona uma interface com botões no estilo de um CD-Player (*play*, *fast forward*, *rewind*, *stop*). Pode-se controlar o momento particular da simulação que se deseja visualizar, bem como, redefinir a apresentação do desenho da topologia (FALL e VARADHAN, 2006).

Além de servir para a animação de modelos de simulação, o NAM aceita arquivos formatados de *traces* de dados gerados por redes reais. Esta é uma característica importante dada à necessidade de obter resultados de desempenho baseados em tráfegos que modelem de forma mais fiel as aplicações existentes.

Outra ferramenta de visualização que o NS-2 fornece é o Xgraph (XGRAPH,2006), que é um gerador de gráficos X-Y de uso geral com opções interativas de "*zoom*", impressão e opções de visualização. Ele está ilustrado na Figura 21 e permite a visualização de uma ou mais curvas de tráfego no mesmo gráfico.

São criados gráficos a partir de dados contidos em um ou mais arquivos gerados pelo *script* TCL da simulação. O Xgraph produz arquivos *PostScript*, PDF e MIF (*Maker Interchange Format*) para serem impressos, armazenados, compartilhados ou inseridos em outros arquivos. Permite configuração de cores e espessura das linhas.

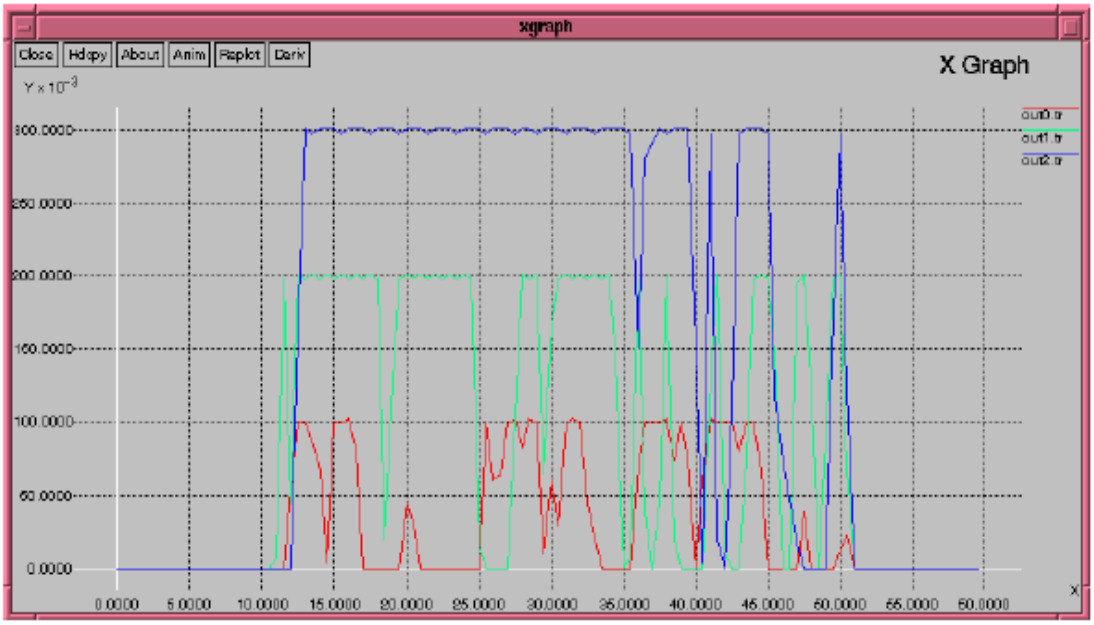

Figura 21 - Exemplo do uso do Xgraph
Fonte: http://www.isi.edu/nsnam/ns/tutorial/index.html

4.3.5.5 - Funcionamento geral NS-2

Depois de abordadas várias características do NS-2 nos tópicos anteriores, a Figura 22 ilustra o funcionamento do simulador de uma maneira geral.

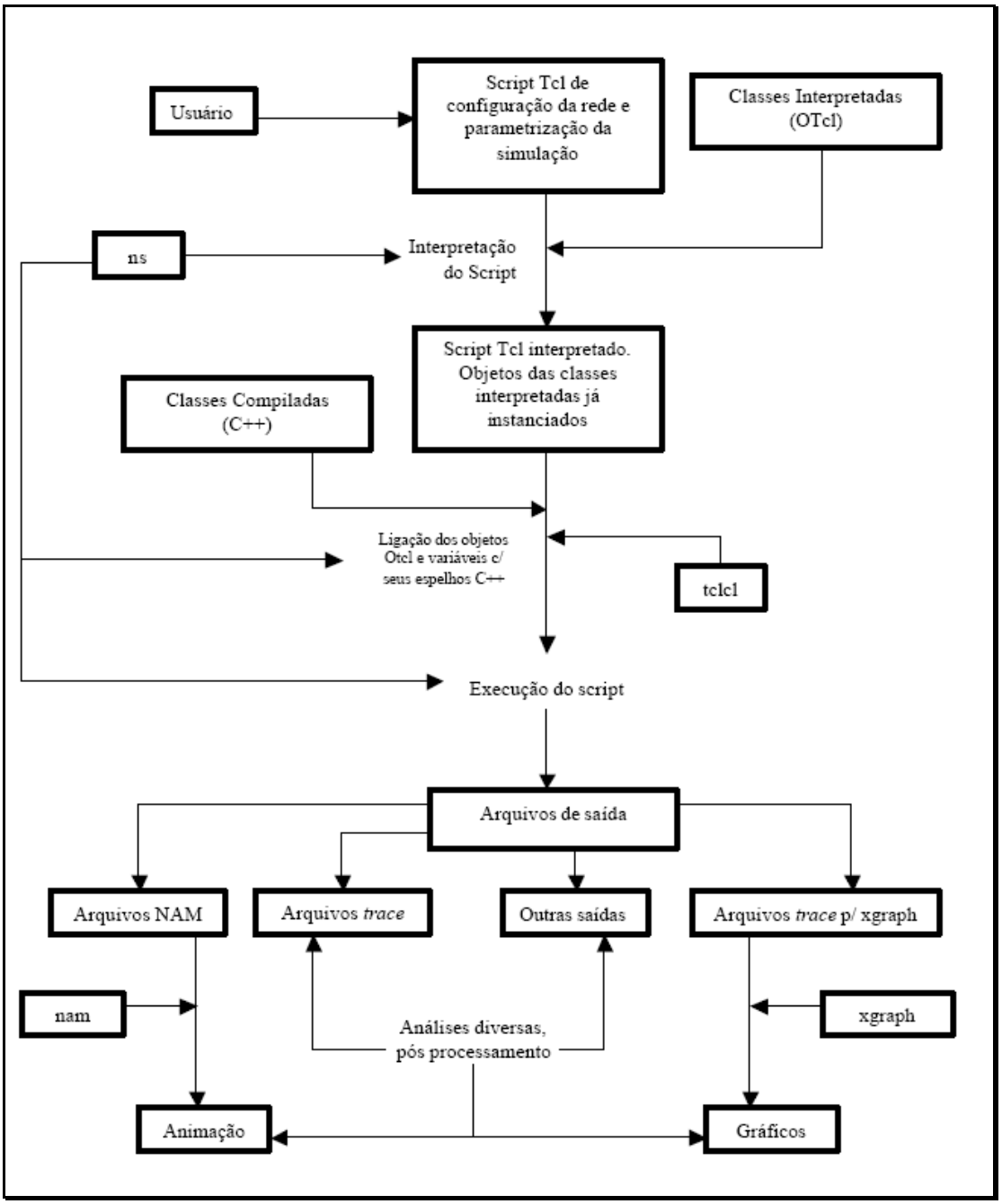

Figura 22 - Funcionamento do NS-2
Fonte: http://www.jabour.com.br/ufjf/redes/ns2.pdf

4.4 - Considerações finais

Há muitos simuladores de rede disponíveis. Alguns dos mais populares como o OMNeT++, Opnet, Glomosim, Boson NetSim além do *Network Simulator*, foram citados neste capítulo. Cada um possui suas vantagens e desvantagens, e após análise optou-se por utilizar o *Network Simulator*. Os fatores que fizeram com que fosse escolhido o NS-2 neste projeto foi a característica de possuir código aberto para estudo e modificação aliado ao fato de ter um alto poder e versatilidade para criação de novos módulos, mesmo sendo complexo o desenvolvimento desses módulos.

O NS-2 permite acessar o código do núcleo do simulador, podendo ser criados quaisquer ambientes de simulação, protocolos ou alterações necessárias. Além disso, o NS-2 possui uma comunidade muito grande e ativa formada pelos próprios usuários do simulador.

No próximo capítulo será abordado a implementação no NS-2, explicando partes dos *scripts* utilizados.

CAPÍTULO 5 - PROCEDIMENTO DE AVALIAÇÃO

Neste capítulo é descrito como foi efetuado a implementação deste projeto, os materiais e os métodos usados, além de mostrar os resultados obtidos.

5.1 - Considerações Iniciais

O conceito de redes sem fio *ad-hoc*, o protocolo de segurança WEP para redes sem fio e detalhes sobre os simuladores foram descritos nos capítulos anteriores. Nesse capítulo são apresentados os detalhes sobre o novo agente que será acoplado ao NS-2 e as simulações realizadas para a avaliação do protocolo WEP. Entretanto, antes de analisar como foram feitas as simulações e avaliados os resultados obtidos, será abordado na seção 5.2 o que foi utilizado para confecção das simulações bem como a instalação e configuração do NS-2. Os resultados das simulações são comentados na seção 5.4.

5.2 - Materiais e Métodos

Para o desenvolvimento desse trabalho, foi utilizado um computador AMD Athlon XP 2400+, com 512 Mb de memória e 40 Gb de HD. O sistema operacional utilizado para instalar o NS-2 foi o Fedora Core 5 que está disponível na página http://fedora.redhat.com/. Este sistema operacional foi instalado no computador com sucesso.

No capítulo anterior foi abordado as características do NS-2. Como fora dito é um *software* gratuito e está disponível no *website* oficial do NS-2 em www.isi.edu/nsnam/ns.

Para efetuar as simulações foi utilizada a versão *ns-allinone* 2.29 para Linux que possui 50 megabytes (.tar.gz) e requer aproximadamente 250 MB de espaço em disco.

Existem duas opções: instalar os módulos separados, ou instalar todo o pacote de uma só vez. Para os iniciantes sugere-se a instalação do formato *allinone* (tudo em um), e por isso optou-se pela mesma (NS-2, 2006).

Com o pacote *allinone*, a instalação é bem mais simples. Este pacote vem com todos os componentes necessários e alguns adicionais usados quando o NS-2 é executado.

5.2.1 - Procedimentos de instalação do *Network Simulator (NS-2)*

O NS-2 foi construído para rodar preferencialmente em plataformas Unix (FreeBSD, SunOS, Solaris, Linux, dentre outras). Por este motivo foi escolhido o sistema operacional Linux Fedora Core 5. Depois de baixar o *software* do site oficial do NS-2 (www.isi.edu/nsnam/ns), foi preciso inicialmente entrar no console do Linux Fedora Core 5 e descompactar o arquivo obtido através do comando:

```
tar -xzf ns-allinone-2.29.3.tar.gz
```

O próximo passo é efetuar a instalação do NS-2. Para isso foi necessário entrar no diretório onde foi descompactado o NS-2 e digitar o *script* de instalação ./install .

```
cd ns-allinone-2.29
./install
```

Um arquivo *batch* fez todo o resto. Através desse arquivo *batch* foi efetuada a instalação que automaticamente configura, compila e instala os componentes do pacote *allinone*. Este pacote já vem com os componentes:

- Tcl/Tk: Interpretador de linguagem Tcl, que é a interface do simulador com o usuário (componentes necessários);
- OTcl: suplemento de orientação a objetos para o Tcl (componente necessário);

- Tclcl: implementação de classes para Tcl (componente necessário);
- ns-2: classes do simulador propriamente dito (componente necessário);
- nam-1: visualizador e animador gráfico de topologias de rede e simulação (componente opcional);
- xgraph: ferramenta de plotagem de gráficos (componente opcional);
- cweb e SGB: bibliotecas requeridas para sgb2-ns e gt-itm (componentes opcionais);
- Gt-itm, gt-itm e sgb2-ns: gerador de topologias (componentes opcionais);
- zlib: ferramenta de compressão de arquivos (componente opcional).

O tempo de instalação varia de acordo com o equipamento, porém é razoavelmente lento na média. Esperou-se em torno de 30 minutos até se concretizar essa etapa da instalação.

O processo de instalação basicamente compila os arquivos da estrutura central do NS-2, que são escritos em C++, gerando um binário capaz de executar as simulações escritas em TCL.

Após a instalação, é necessária a modificação de variáveis de ambiente do sistema operacional, como por exemplo, a variável *path* do sistema, onde se inclui os diretórios dos executáveis do NS-2. Faz-se isso ao final da instalação, se não houve erro de compilação o *script* dará uma mensagem de sucesso e fornecerá instruções para modificação das variáveis. São mostrados alguns caminhos que devem ser copiados da interface textual e adicionados ao arquivo de inicialização como mostrado abaixo. Esse aviso foi retirado da instalação realizada, e a parte destacada de vermelho é o *path* que foi acrescentado no sistema Fedora. As partes destacadas de azul são variáveis que necessitam serem acrescentadas no sistema para o bom funcionamento do NS-2.

```
Here are the installation places:
tcl8.4.11:      /root/ns-allinone-2.29/{bin,include,lib}
tk8.4.11:               /root/ns-allinone-2.29/{bin,include,lib}
otcl:           /root/ns-allinone-2.29/otcl-1.11
tclcl:          /root/ns-allinone-2.29/tclcl-1.17
ns:             /root/ns-allinone-2.29/ns-2.29/ns
nam:    /root/ns-allinone-2.29/nam-1.11/nam
xgraph: /root/ns-allinone-2.29/xgraph-12.1
gt-itm:   /root/ns-allinone-2.29/itm, edriver, sgb2alt, sgb2ns,
sgb2comns, sgb2hierns

----------------------------------------------------------------------------

Please put /root/ns-allinone-2.29/bin:/root/ns-allinone-2.29/tcl8.4.11/unix:/root/ns-allinone-
2.29/tk8.4.11/unix
into your PATH environment; so that you'll be able to run itm/tclsh/wish/xgraph.

IMPORTANT NOTICES:

(1) You MUST put /root/ns-allinone-2.29/otcl-1.11, /root/ns-allinone-2.29/lib,
   into your LD_LIBRARY_PATH environment variable.
   If it complains about X libraries, add path to your X libraries
   into LD_LIBRARY_PATH.
   If you are using csh, you can set it like:
            setenv LD_LIBRARY_PATH <paths>
   If you are using sh, you can set it like:
            export LD_LIBRARY_PATH=<paths>

(2) You MUST put /root/ns-allinone-2.29/tcl8.4.11/library into your
TCL_LIBRARY environmental
   variable. Otherwise ns/nam will complain during startup.

(3) [OPTIONAL] To save disk space, you can now delete directories tcl8.4.11
   and tk8.4.11. They are now installed under
/root/ns-allinone-2.29/{bin,include,lib}

After these steps, you can now run the ns validation suite with
cd ns-2.29; ./validate

For trouble shooting, please first read ns problems page
http://www.isi.edu/nsnam/ns/ns-problems.html. Also search the ns
mailing list archive
for related posts.
```

A alteração do *path* foi feita no arquivo *profile,* que está localizado no diretório /etc, pois assim todos os usuários do computador poderão executar o NS-2 sem problemas. Foi utilizado o editor de textos vi para fazer essa alteração. Abaixo é mostrado o arquivo *profile* atualizado. As atualizações aparecem em destaque no texto.

```
# /etc/profile
# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

pathmunge () {
        if ! echo $PATH | /bin/egrep -q "(^|:)$1($|:)" ; then
           if [ "$2" = "after" ] ; then
              PATH=$PATH:$1
           else
              PATH=$1:$PATH
           fi
        fi
}

# ksh workaround
if [ -z "$EUID" -a -x /usr/bin/id ]; then
        EUID=`id -u`
        UID=`id -ru`
fi

# Path manipulation
if [ "$EUID" = "0" ]; then
        pathmunge /sbin
        pathmunge /usr/sbin
        pathmunge /usr/local/sbin
fi

# No core files by default
ulimit -S -c 0 > /dev/null 2>&1

if [ -x /usr/bin/id ]; then
        USER="`id -un`"
        LOGNAME=$USER
        MAIL="/var/spool/mail/$USER"
fi

HOSTNAME=`/bin/hostname`
HISTSIZE=1000
LD_LIBRARY_PATH="/root/ns-allinone-2.29/otcl-1.11, /root/ns-allinone-2.29/lib"
TCL_LIBRARY="/root/ns-allinone-2.29/tcl8.4.11/library"

if [ -z "$INPUTRC" -a ! -f "$HOME/.inputrc" ]; then

   INPUTRC=/etc/inputrc
fi

export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE INPUTRC
LD_LIBRARY_PATH TCL_LIBRARY

export PATH="$PATH:/root/ns-allinone-2.29/bin:/root/ns-allinone-2.29/tcl8.4.11/unix:/root/ns-
allinone-2.29/tk8.4.11/unix"

for i in /etc/profile.d/*.sh ; do
   if [ -r "$i" ]; then
        . $i
   fi
done

unset i
unset pathmunge
```

Depois de efetuado esse ajuste do *path* nas variáveis de ambiente do sistema, é opcional fazer o teste para verificar se a instalação foi bem sucedida. Optou-se por fazer esse teste para a validação do NS-2, portanto foi digitado o comando ./validate, e esperou-se em torno de 40 minutos. A validação consiste em executar várias simulações pré-programadas e comparar seus resultados com resultados-padrões.

Após efetuada a validação passa-se a fase de desenvolvimento do *script* que se deseja para efetuar a simulação. Como o NS-2 não oferece suporte para simulação do protocolo WEP nem de nenhum tipo de segurança, foi necessário o desenvolvimento de um novo agente chamado AgentWep para poder realizar alguns estudos de casos. O desenvolvimento e explicação do novo agente e do *script* serão descritos na seção 5.3. Basicamente uma simulação com o NS-2 consiste em 5 passos:

1- Planejar a simulação
2- Definir os nós
3- Definir a ligação entre os nós (topologia)
4- Definir o tráfego que será injetado na rede
5- Analisar os resultados

Para escrever a simulação qualquer editor de textos pode ser utilizado, desde os baseados em texto como o ***emacs***, ***vi*** ou o ***mcedit*** até os editores gráficos como o ***kedit*** ou o ***kate*** que já vêm com a interface gráfica kde. Os arquivos devem ser gravados com a extensão ***.tcl***. Para se executar a simulação basta digitar:

```
ns nome-do-arquivo.tcl
```

5.3 - Desenvolvimento do protocolo WEP e do *Script* Otcl

A interface entre o usuário e o NS-2 dá-se através da linguagem em *script* Otcl. Deste modo, para desenvolver uma simulação no NS-2 é preciso em primeiro lugar montar um *script* em Otcl, que deverá conter as seguintes partes básicas:

- criação do objeto Simulador;
- abertura de arquivos para *tracing* e análise posterior;
- criação da topologia de rede;
- criação de nós ou nodos;
- conexão dos nós entre si (*links*);
- criação das filas de saída;
- criação dos agentes de 4ª. Camada e conexão com *hosts*;
- criação dos geradores de tráfego (nível de Aplicação) e conexão com agentes de 4ª. Camada (nível de Transporte);
- programação dos escalonadores e *timer* e
- fechamento da simulação, animação e geração de estatísticas.

O processo de simulação pode ser resumido conforme mostra a Figura 23. Primeiramente a criação do *script* contendo os itens básicos citados acima, que pode ser feito como um arquivo texto comum (salvar com extensão .tcl), em segundo lugar a execução desse *script* no NS-2 com o comando *ns nome-script.tcl* que irá fazer a geração dos arquivos de *tracing* com registro de cada evento simulado e por último após a conclusão da simulação é preciso analisar os resultados gerados, cálculos feitos de preferência com a ajuda do NAM e xgraph, para poder extrair os resultados desejados.

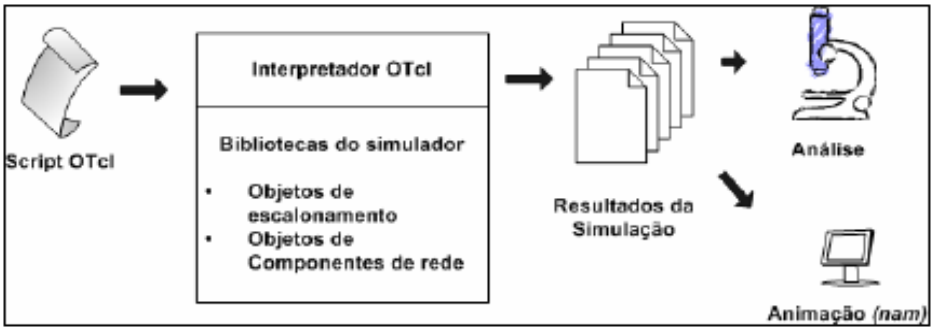

Figura 23 - Utilização do NS-2
Fonte: http://www.geocities.com/locksmithone/articles/networksimulator-sepa.pdf

O desenvolvimento do *script* foi feito totalmente na linguagem Otcl denominado projeto.tcl (Apêndice A). Mas antes de ser feito o desenvolvimento do *script* foi preciso adequar o NS-2, adicionando o protocolo WEP. Os detalhes da implementação do novo agente adicionado ao NS-2 será detalhado no item 5.3.1.

5.3.1 - Implementação do protocolo WEP no *Network Simulator*

Visando entender melhor como criar um novo agente no *Network Simulator*, antes de efetuar a inserção do protocolo WEP foi realizada a criação de um agente simples (CHUNG, CLAYPOOL 2006).

Supondo já tenha criado em C++ uma nova classe para o NS, por exemplo "MeuAgente". Esta é filha da classe Agent. É desejável que se torne possível criar uma instância desse objeto em OTcl. Para fazer isso, tem que definir um objeto de ligação, como "MeuAgenteClass", que deve ser derivado de TclClass. Este objeto de ligação cria um objeto OTcl de nome específico(Agent/MeuAgenteOTcl, neste exemplo), e cria um link entre o objeto OTcl e o objeto C++(MeuAgente, neste exemplo). A seguir é mostrado a definição da classe MeuAgente:

```
Class MeuAgente : public Agent {
  public:
            MeuAgente( );
  protected:
            int command (int  argc, const char*const* argv);
  private:
            int my_var1;
            double my_var2;
            void MinhaFunção(void);
  };
```

Agora, a classe MeuAgenteClass, filha de TclClass e responsável pelo link das classes MeuAgente e MeuAgenteOTcl.

```
static class MeuAgenteClass : public TclClass {
   public:
                MeuAgenteClass( ) : TclClass (“Agent/MeuagenteOtcl”) {}
                TclObject* create (int, const char*const*) {
                        return (new MeuAgente( ) );
                }
   } class_meu_agente;
```

Depois que se exportou as classes em C++ para OTcl, tem que analisar as variáveis. O objeto MeuAgente tem duas variáveis, que são my_var1 e my_var2, que é desejável que elas possam ser modificadas facilmente a partir do OTcl. Para isso basta usar os métodos bindings.

```
MeuAgente :: MeuAgente( ) : Agent(PT_UDP)
 {
   bind(“my_var1_otcl”, &my_var1);
   bind(“my_var2_otcl”, &my_var2);
}
```

Note que as funções de *binding* estão colocadas no construtor de “MeuAgent”, para que o link seja concretizado assim que o objeto seja criado. Feito isso, é necessário criar comandos em OTcl para o agente criado. Consequentemente, é necessário que se exporte os objetos em C++ de controle de comando para o OTcl. Para isso, é preciso definir uma função command do objeto C++(“MeuAgente”).

```
int MeuAgente ::command(int argc, const char*const* argv) {
        if (argc = = 2) {
                if (strcmp(argv[1], "chama-minha-função") = = 0) {
                        MinhaFunção( );
                        return ( TCL_OK);
                }
        }
        return (Agent :: command (argc, argv));
}
```

Depois de implementado um novo objeto em C++, é desejável executar um comando OTcl a partir do objeto C++. Confira a implementação de "MinhaFunção", método da classe "MeuAgente", que faz o interpretador imprimir nas variáveis "my_var1" e "my_var2".

```
    void MeuAgente :: MinhaFunção(void) {
        Tcl& tcl = Tcl :: instance( );
        tcl.eval ("puts \ "Message From MinhaFunção\ "");
        tcl.evalf("puts \ "    my_var1 = %d\"", my_var1);
        tcl.evalf("puts \ "    my_var2 = %f\"", my_var2);
}
```

Agora, para ilustrar a criação do agente "MeuAgente", um script será mostrado.

```
set myagent [new Agent/MeuAgenteOTcl]
$myagent set my_var1_otcl  2
$myagent set my_var2_otcl  3.14
$myagent  chama-minha-função
```

Como saída, tem-se:

```
Message From MinhaFunção
        my_var1 = 2
        my_var2 = 3.140000
```

Essa é a forma básica para se criar um novo agente. Para esse agente foi uma tarefa fácil, entretanto para agentes com reais funções, que tenham realmente uma aplicação prática, a complexidade aumenta consideravelmente, mesmo sendo seguida a idéia do procedimento. No presente trabalho, foram inseridos dois agentes, um para envio de mensagens chamado WEP, e outro para recepção chamado WEPRecv.

A implementação deste trabalho consiste na abertura de sessão do protocolo. Outras funcionalidades do protocolo WEP, em relação à criptografia poderão ser adicionadas a partir desta implementação inicial.

O tempo exigido nessa etapa foi elevado devido à necessidade de interligação entre os códigos dos novos agentes e a interface em OTcl no simulador. Esse é um trabalho complexo levando em consideração que as linguagens Otcl e C++ devem ser estudadas com maior profundidade. Os códigos dos novos agentes podem ser visualizados no Anexo I.

Para que o novo Agente WEP possa funcionar, foi necessário adicionar algumas linhas de código aos programas: *ns~/common/packet.h, ns~/tcl/lib/ns-packet.tcl e ns~/tcl/lib/ns-default.tcl* do NS-2. Nesta seção serão mostradas os ajustes feitos nestes programas.

Inicialmente foi definido no programa *packet.h* a existência de um novo pacote com a estrutura *hdr_wep* e seu método *access()* de acesso ao pacote.

```
#define HDR_WEP(p)      (hdr_wep::access(p))
```

Após isso, foi incluído o novo pacote wep na pilha de pacotes já existentes. Os dois exemplos abaixo mostram a parte do código onde ele foi acrescentado.

```
enum packet_t {
        PT_TCP,
        PT_UDP,
        PT_WEP,
        …
class p_info {
        public:
                p_info() {
                        name_[PT_TCP]= "tcp";
                        name_[PT_UDP]= "udp";
                        name_[PT_WEP]= "wep";
                        ...
```

Foi necessário adicionar também, o novo pacote wep, ao programa *ns-packet.tcl*. Veja abaixo:

```
foreach prot {
        AODV
        ARP
        aSRM
        WEP
```

E posteriormente foi necessário setar os valores *default* de algumas variáveis no programa *ns-default.tcl* necessárias ao funcionamento do Agente WEP e ao Agente WEPRecv:

```
#Agente WEP
Agent/WEP set seqno_            0
Agent/WEP set t_seqno_          0
Agent/WEP set maxburst_         0
Agent/WEP set maxcwnd_          0
Agent/WEP set numdupacks_       3
Agent/WEP set first_ true ;     # Permite o envio do pacote first_
Agent/WEP set packetSize_       1000
Agent/WEP set tcpip_base_hdr_size_ 40
Agent/WEP set tcpTick_          0.01;
Agent/WEP instproc done {} { }
Agent/WEP set dupacks_          0
Agent/WEP set cntl_             0
Agent/WEP set cwnd_             0
Agent/WEP set maxseq_           0
Agent/WEP set useHeaders_       true;
Agent/WEP set ndatapack_        0
Agent/WEP set ndatabytes_       0
Agent/WEP set nackpack_         0
Agent/WEP set trace_all_oneline_        false
Agent/WEP set nam_tracevar_             false

#Agente WEPRecv
Agent/WEPRecv set sport_        0
Agent/WEPRecv set dport_        0
Agent/WEPRecv set packetSize_ 40
Agent/WEPRecv set bytes_        0
```

Depois de fazer essas mudanças, o NS-2 foi recompilado através do comando *make*, efetuado no diretório do ns2.29 e o script em OTCl de teste pode ser executado.

Para simular os novos agente WEP e WEPRecv , foi escrito um script simples em OTCL contendo a configuração dos agentes e a topologia da rede. O script pode ser visualizado no Apêndice A.

5.4 - Discussão dos Resultados

Como resultados obtidos têm-se a inserção de um novo agente WEP no NS-2 que inicia o processo de inserção de segurança nesta ferramenta de simulação.

O desenvolvimento e inserção de novos agentes no NS-2 é uma tarefa muito árdua. O desenvolvimento do agente WEP não foi diferente. Surgiram diversas dificuldades para

inserir o novo agente no simulador, tais como a geração dos arquivos objetos referente à implementação do wep.cc e do weprecv.cc., levando um tempo muito maior que o esperado.

Após realizado o procedimento de inserção dos novos agentes foi confeccionado um *script* teste, disponível na íntegra no Apêndice A. Nele é apresentado um cenário *wireless* com 3 nós.

Com a inserção do agente foi possível verificar a troca de pacotes em uma rede *ad hoc* e garantir que é possível, apesar de custosa, a inserção de segurança no NS-2. Uma vez inserido o agente é possível que novas alterações sejam efetuadas buscando aprimorar o funcionamento do mesmo.

Os resultados obtidos são apenas uma prévia do que realmente se deseja fazer no âmbito de simulação de redes sem fio, porém são relevantes, pois constituem a base para proferir simulações envolvendo segurança em redes *wireless*.

CAPÍTULO 6 - CONCLUSÕES

Através desse trabalho pode-se concluir que simulação é muito importante para avaliar o desempenho de várias tecnologias. Além disso, ela proporciona uma pré-visualização do *layout* e dos possíveis erros. Sendo assim a utilização dessa técnica em rede *wireless* é relevante e viável.

Tendo em vista que simulação de segurança em redes sem fio é pouco freqüente na literatura a proposta do trabalho tem relevância no contexto literário. Além disso, para que se possa providenciar segurança em redes sem fio através de simulações e utilizando o protocolo WEP é necessário que se construa a base para o mesmo, por meio de inserção de agentes para abertura de seção e transmissão de pacotes.

Uma vez que o NS-2 não provê segurança, a inserção de agentes que possibilitem a geração e inclusão de características de segurança, principalmente em rede *wireless*, a inserção e a configuração, *linkagen*, consistem fatores importantes no tocante a redes de computadores sem fio.

6.1 - Publicações

- 14º Simpósio Internacional de Iniciação Científica da Universidade de São Paulo
  Forma de apresentação: resumo enviado e aceito para apresentação.
  Título: Avaliação de Desempenho do Protocolo TCP com Controle de Congestionamento Highspeed em Redes com Topologia Estrela
  Local: USP - Universidade de São Paulo **Data:** 2006

- XIV Congresso de Iniciação Científica da UFSCar
  Forma de apresentação: resumo enviado e aceito para apresentação.
  Título: Avaliação do Desempenho do Protocolo TCP/IP em Redes com Topologia Estrela Utilizando a Ferramenta *Network Simulator*.
  Local: Universidade Federal de São Carlos **Data:** 2006

- 14º Simpósio Internacional de Iniciação Científica da Universidade de São Paulo
  Forma de apresentação: resumo enviado e aceito para apresentação.

Título: Avaliação de Desempenho do Protocolo WEP em Redes Sem Fio *Ad-Hoc* Usando um Simulador de Redes
Local: USP - Universidade de São Paulo **Data:** 2006

6.2 - Trabalhos Futuros

Como trabalhos futuros propõe-se a melhoria do agente WEP e uma maior bateria de testes. Propõe-se ainda a inserção de novos agentes de segurança como o WPA *(Wi-Fi Protected Acess Protocol)* e a possível comparação entre os dois. Além disso, pode-se ainda trabalhar com outras topologias de rede sem fio, buscando aprimorar e comparar o desempenho das mesmas.

REFERÊNCIAS BIBLIOGRÁFICAS

(ANDRADE et al, 2003) ANDRADE, Rogério; KAMIENSKI, Carlos; SOUSA, Dênio e SADOK, Djamel. **O Algoritmo SQM-Response para Controle de Congestionamento do Protocolo TCP.** 21º Simpósio Brasileiro de Redes de Computadores (SBRC-2003), Natal-RN, Maio de 2003.

(ARAUJO et al, 2004) ARAUJO, Marcel Ferraro; LOPES, Eyder J. Sabá; FARIAS, Max A. Salviano. **Estudo de Integração do Sistema SUS a uma Rede Sem Fio Padrão**. 2004. Universidade da Amazônia. Trabalho de conclusão de curso para obtenção do grau de Bacharel em Ciências da Computação. Disponível na Internet em http://www.cci.unama.br/margalho/portaltcc/tcc2004/eydermarcel&max.pdf, acesso em 27 de março de 2006.

(AURÈLIO, 2006) **Dicionário Aurélio Disponível para Consulta On-Line.** Disponível na Internet em http://200.225.157.123/dicaureliopos/home.asp?logado=true, acesso em 10 de novembro de 2006.

(BATISTA, 2002) BATISTA, Marcelo H. Euzébio. **Protótipo de Gerenciamento para Redes Móveis (PGRM)**. 2002. Unilasalle – Centro Universitário La Salle Disponível na Internet em http://www.sinprors.org.br/paginasPessoais/layout3/..%5Carquivos%5CProf_250%5Cmonografia.pdf , acesso em 07 de abril de 2006.

(BORISOV et al,2006) BORISOV ,Nikita; GOLDBERG, Ian; WAGNER,David. **Security of the WEP algorithm.** Disponível na Internet em http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html, acesso em 28 de outubro de 2006.

(BOSON,2006) **Boson Holdings**. Disponível na Internet em http://www.boson.com/, acesso em 25 de outubro de 2006.

(CÂMARA, 2006) CÂMARA, Daniel. **Roteamento em Redes Ad-Hoc**. Universidade Federal de Minas Gerais. Instituto de Ciências Exatas. Disponível na Internet em http://www.dcc.ufmg.br/~danielc/redes/roteamento.html, acesso em 28 de março de 2006.

(CARVALHO FILHO, 2005) CARVALHO FILHO, João Rogério Lima de. **Um estudo de protocolos empregados na segurança de dados em redes sem fio – Padrão 802.11**. Monografia apresentado ao Centro Universitário de João Pessoa – UNIPÊ para obtenção do grau Bacharel em Ciências da Computação. Disponível na Internet em http://www.unipe.br/graduacao/computacao/projetos/tcc20052/UM%20ESTUDO%20DE%20PROTOCOLOS%20EMPREGADOS%20NA%20SEGURAN%C7A%20DE%20DADOS%20%20EM.pdf , acesso em 20 de maio de 2006.

(COMER, 1997) COMER, Douglas E. **Interligação em rede com TCP/IP** – tradução de Internetworking with TCP/IP. Rio de Janeiro: Editora Elsevier, 1997.

(CHUNG, CLAYPOOL 2006) CHUNG, Jae ; CLAYPOOL, Mark. **WPI – WORCESTER POLYTECHNIC INSTITUTE.** Disponível na Internet em http://nile.wpi.edu/NS/, acesso em 5 de novembro de 2006

(CONSER,2002) **CONSER - Collaborative Simulation for Education and Research**. Disponível na Internet em http://www.isi.edu/conser/index.html, acesso em 14 de outubro de 2006.

(DUARTE, 2003, 2003) DUARTE, Luiz Otavio. **Análise de Vulnerabilidades e Ataques Inerentes a Redes Sem Fio 802.11x**. 2003. 55f. Projeto Final de Curso submetido ao Departamento de Ciências de Computação e Estatística do Instituto de Biociências, Letras e Ciências Exatas (IBILCE) da Universidade Estadual Paulista Júlio de Mesquita Filho, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação. Disponível na Internet em angel.acmesecurity.org/hp_ng/files/testes_monografias/acme-monografia-Wireless-2003-LOD.pdf , acesso em 5 de janeiro de 2006.

(FALL e VARADHAN, 2006) FALL, Kevin, VARADHAN, Kannan., "**The ns Manual", The VINT Project**. Disponível na Internet em http://www.isi.edu/nsnam/ns/doc/index.html, acesso 2 de novembro de 2006.

(GAONA,1995) Gaona, Hugo Blas Mendieta. **O Uso da Simulação para Avaliar Mudanças Organizacionais na Produção.** Dissertação submetida à Universidade Federal de Santa Catarina para a obtenção do Grau de Mestre em Engenharia. Disponível na Internet em http://www.eps.ufsc.br/disserta/gaona/indice/index.htm, acesso em 26 de outubro de 2006.

(GLOMOSIM 2003) **GloMoSim Web Site**. Disponível na Internet em http://pcl.cs.ucla.edu/projects/glomosim/, acesso em 8 de novembro de 2006.

(HEIDEMANN et al, 2001) HEIDEMANN, John; ESTRIN, Deborah; GOVINDAN, Ramesh, GOEL,Ashish. **SAMAN - Simulation Augmented by Measurement and Analysis for Networks**. Disponível na Internet em http://www.isi.edu/saman/index.html, acesso em 14 de outubro de 2006.

(ICIR, 2006) **ICIR - The International Computer Science Institute Center for Internet Research**. Disponível na Internet em http://www.icir.org/, acesso em 15 de outubro de 2006.

(JUNIOR, 2003) JUNIOR, Egídio Ieno. **Uma proposta de Metodologia para Análise de desempenho de Redes IEEE 802.11 Combinado a Gerência SNMP e Ferramentas de Simulação**. Dissertação apresentada ao Instituto Nacional de Telecomunicações, como parte dos requisitos para obtenção do Título de Mestre em Engenharia Elétrica. Disponível na Internet em http://cict.inatel.br/nova2/docentes/lucianol/opnet/teses/TESEEgidioFINAL_4-12_.pdf,acesso em 4 de maio de 2006.

(JUNIOR et al, 2004) JUNIOR, Carlos A. C. Vaz Pereira; BRABO, Gustavo da Silva; AMORAS, Rômulo A. De Sales. **SEGURANÇA EM REDES WIRELESS PADRÃO IEEE 802.11b: PROTOCOLOS WEP, WPA E ANÁLISE DE DESEMPENHO.** Trabalho de Conclusão de Curso apresentado, como requisito parcial para obtenção de do Grau de Bacharel em Ciência da Computação, à Universidade da Amazônia, Disponível na Internet

em http://www.cci.unama.br/margalho/portaltcc/tcc2004/carlosgustavo&romulo.pdf , acesso em 13 de maio de 2006.

(KESHAV, 1997) KESHAV, S. **REAL 5.0 overview**. Disponível na Internet em http://www.cs.cornell.edu/skeshav/real/overview.html, acesso em 14 de outubro de 2006.

(MAIA, 2006) MAIA, Roberto. **Segurança em Redes Wireless 802.11i.** GTA - Universidade Federal do Rio de Janeiro. Disponível na Internet em http://www.gta.ufrj.br/~rmaia/802_11i.html, acesso em 12 de janeiro de 2006.

(MARTINS, 2003). MARTINS, Marcelo. **Protegendo Redes Wireless 802.11b.** Disponível na Internet em **http://www.planetarium.com.br/planetarium/noticias/2003/3/1048024279/protegendo_redes_wireless.pdf,** acesso em 19 de maio de 2006.

(MATHIAS, 2006) MATHIAS, André Pimenta. **IEEE 802.11 – Redes Sem Fio**. Disponível na Internet em www.gta.ufrj.br/grad/00_2/ieee/ , acesso em 27 de janeiro de 2006.

(MATTOS, 2005) MATTOS, José H. Lara, **SEGURANÇA EM REDES WIRELESS**. Projeto final de graduação apresentado ao curso de Sistemas de Informação, do IESUR – Instituto de Ensino Superior de Rondônia, 2005.

(MELO, 2004) MELO, Rogério de Castro**. FERRAMENTA DE AUXÍLIO À INSTALAÇÃO DE REDES 802.11 INFRA-ESTRUTURADAS**. Projeto de conclusão de curso apresentado a Universidade Federal do Rio de Janeiro para obtenção do grau de Engenheiro Eletrônico. Disponível na Internet em http://www.gta.ufrj.br/ftp/gta/TechReports/Rogerio04/Rogerio04.pdf , acesso em 5 de maio de 2006.

(MILANEZ et al, 2004) MILANEZ, Marcelo; MAZIERO, Carlos; JAMHOUR, Edgard. IV **WSeg - Workshop em Segurança de Sistemas Computacionais**, SBRC. Gramado RS, 2004. Disponível na Internet em http://www.ppgia.pucpr.br/pesquisa/sisdist/papers/2004-wseg-milanez-maziero-jamhour.pdf , acesso em 14 de maio de 2006.
(NS-2, 2006) **The Network Simulator – ns -2**. Disponível na Internet em http://www.isi.edu/nsnam/ns/index.html, acesso em 15 de outubro de 2006.

(OMNET, 2006) **OMNet Web Site**. Disponível na Internet em http://www.omnetpp.org/, acesso em 19 de julho de 2006.

(OpNet, 2006). **OPNET – Making Networks and Applications Perform**. Disponível na Internet em http://www.opnet.com/, acesso em 9 de novembro de 2006.

(OTCL,2006) **MIT Object Tool Command Language**. Disponível na Internet em http://otcl-tclcl.sourceforge.net/otcl/, acesso em 5 de novembro de 2006.

(QualNet, 2006) **Scalable Network Technologies, QualNet products**.Disponível na Internet em http://www.scalable-networks.com/products/qualnet.php, acesso em 2 de novembro de 2006.

(SALES, 2004) SALES, William de Araújo. **Uma Entidade Funcional Para Autenticação de Dispositivos Móveis entre Áreas de Micromobilidade**. Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal do Ceará. Disponível na Internet em http://www.mcc.ufc.br/disser/WilliamSales.pdf, acesso em 15 de novembro de 2006.

(SANTOS, 2003) SANTOS, Izabela Chaves. **WPA: A Evolução do WEP**. Disponível na Internet em http://www.lockabit.coppe.ufrj.br/rlab/rlab_textos?id=70, acesso em 21 de maio 2006.

(SOARES et al, 1995) SOARES, Luiz Fernando; LEMOS, Guido; COLCHER, Sergio. **Redes de Computadores: das LANs, MANs e WANs às redes ATM**. Rio de Janeiro: Editora Campus, 1995.

(THOMAS, 1997) THOMAS, Robert M. **Introdução às Redes Locais**. Ed. Makron Books, 1997.

(TANENBAUM, 1994) TANENBAUM, Andrew S. **Redes de Computadores – tradução de Computer Networks.** Rio de Janeiro: Editora Campus, 1994.

(TANENBAUM, 1997) TANENBAUM, Andrew S. **Redes de Computadores – 3ª edição**. Rio de Janeiro: Editora Campus, 1997.

(TANENBAUM, 2003) TANENBAUM, Andrew S. **Redes de Computadores. 4. ed**. Rio de Janeiro: Editora Campus, 2003.

(TOLARI, ZANARDI, 2005) TOLARI, Luciano Ferreira; ZANARDI, Marco Aurélio. **INSTALAÇÃO E SEGURANÇA EM UMA REDE SEM FIO.** Dissertação apresentada ao Programa de Pós Graduação do Centro Universitário Eurípides de Marília, mantido pela Fundação de Ensino Eurípides Soares da Rocha, para obtenção do Título de Pós Graduação em Redes de Computadores e Internet, 2005.

(TORRES, 2001) TORRES, Gabriel. **Redes de Computadores: Curso Completo. 1ª Edição**. Rio de Janeiro: Axcel Books, 2001.

(UCLA, 2001) **UCLA Parallel Computing Laboratory.** Disponível na Internet em http://pcl.cs.ucla.edu/projects/parsec/, acesso em 5 de novembro de 2006.

(VASCONCELOS, 2005) VASCONCELOS, Daniel de Souza. **SEGURANÇA EM REDES LOCAIS WIRELESS**. Universidade Federal da Bahia. Disponível na Internet em http://twiki.im.ufba.br/pub/MAT057/SemestreAtualProgramacao/DanielSouza.pdf ,acesso em 14 de maio de 2006.

(VINT, 2006) **VINT**, **Virtual InterNetwork Testbed**. Disponível na Internet em http://www.isi.edu/nsnam/vint/, acesso em 10 de janeiro de 2006.

(XGRAPH,2006) **Xgraph**. Disponível na Internet em http://www.isi.edu/nsnam/xgraph/index.html; http://www.atl.external.lmco.com/proj/csim/xgraph/xgraph.html, acesso 10 de novembro de 2006.

(ZANNETI, GONÇALVES, 2006) ZANNETI, Alberto René, GONÇALVES, Leandro de Carvalho. **Redes Locais Sem Fio**. Universidade Federal de São Carlos. Centro de Ciências Exatas e de Tecnologia. Programa de Pós Graduação em Ciências da Computação. Disponível na Internet em http://www.dc.ufscar.br/~carvalho/WLAN/index.html , acesso em 17 de janeiro de 2006.

BIBLIOGRAFIA COMPLEMENTAR

(BARROS, 2005) Barros**,** Emílio Augusto Coelho. **Aplicações de Simulação Monte Carlo e Bootstrap**. Trabalho de conclusão de curso apresentado à Universidade Estadual de Maringá. Centro de Ciências Exatas. Departamento de Estatística. Disponível na Internet em http://www.des.uem.br/graduacao/Monografias/Monografia_Emilio.pdf, acesso em 22 de outubro de 2006.

(MENEZES, 2006) MENEZES, Rodrigo Saldanha de. **IEEE 802.11 – Wireless**. Universidade Federal do Rio de Janeiro. Disponível na Internet em http://www.gta.ufrj.br/grad/98_2/rodrigo/trabalho.html , acesso em 15 de janeiro de 2006.

ANEXO I – Códigos dos agentes inseridos

**********wep.h**********
/* O acesso ao pacote retorna o campo desejado para um ponteiro do pacote
que foi criado. Isso é feito pelas rotinas dentro dos programas
ns~/tcl/lib/ns-packet.tcl e ns~/common/packet.cc.*/

```
#include "agent.h"
#include "packet.h"

struct hdr_wep {

double ts_;          //timestamp do pacote gerado pela fonte
double ts_echo_;     //timestamp do cntl de resposta
int seqno_;          //numero de seqüência
int reason_;         // razão para retx(não usado)
int ptype_;          // tipo de pacote
int cwind_;          //tamanho da janela

//Função de acesso aos campos do pacote
static int offset_;      // offset do cabeçalho wep
inline static int& offset() { return offset_; }
inline static hdr_wep* access(Packet* p) {
return (hdr_wep*) p->access(offset_); }

/* Funções dos campos*/
double& ts_echo( ) { return (ts_echo_); }
double& ts( ) { return (ts_); }
int& cwind( ) { return (cwind_); }
int& ptype( ) { return (ptype_); }
int& seqno() { return (seqno_); }
int& reason() { return (reason_); }

};

// Estrutura dos Pacotes
struct packet_CNTL{
hdr_wep header_CNTL;
int rseq, alloc, echo, cwind;
};
struct packet_FIRST{
hdr_wep header_FIRST;
int alen, aformat, adomain, address;
};
struct packet_DATA{
hdr_wep header_DATA;
int data;
};
```

```
struct packet_ECNTL{
hdr_wep header_ECNTL;
int rseq, aloc, echo, nspan, spans;
};
struct packet_TCNTL{
hdr_wep header_TCNTL;
int tlen, service, tformat;
};
// Inteiro que Identifica os Tipos de Pacotes
#define DADOS       0
#define CNTL        1
#define FIRST               2
#define ECNTL       3
#define TCNTL       4

// Classe Agente WEP

class WepAgent : public Agent {
public:
WepAgent( );
~WepAgent( ) {free(tss);}

virtual void output(int seqno, int reason = 0);
virtual void recv(Packet*, Handler*);
void trace(TracedVar* v);

protected:
virtual int headersize( );      // Tamanho do cabeçalho wep + ip
virtual void delay_bind_init_all( );
virtual int delay_bind_dispatch(const char *varName, const char *localName, TclObject
*tracer);

TracedInt t_seqno_;   /* numero de sequência*/
double ts_peer_;      /* o mais recent timestamp enviado */
double ts_echo_;      /* o mais recente timestamp recebido*/
void reset();         /* reseta as variáveis do agente */
TracedInt dupcntls_;  /* numero de cntls duplicados */
TracedInt curseq_;    /* mais alto numero de sequencia criado pela Aplicação " */
int last_cntl_;       /* último cntl recebido*/
TracedInt highest_cntl_;/* mais alto cntl recebido */
}

**********wep.cc**********

#include <stdlib.h>
#include <math.h>
#include <sys/types.h>
#include "ip.h"
#include "wep.h"
#include "flags.h"
```

```
#include "basetrace.h"

int hdr_wep::offset_;  // variavel offset cabeçalho wep

static class WEPHeaderClass : public PacketHeaderClass {
public:
WEPHeaderClass() : PacketHeaderClass("PacketHeader/WEP", sizeof(hdr_wep)) {
bind_offset(&hdr_wep::offset_);
}
} class_wephdr;

// Chamada do o construtor do agente WEP (Link do OTCL para o Agente WEP)

static class WepClass : public TclClass {
public:
WepClass() : TclClass("Agent/WEP") {}
TclObject* create(int , const char*const*) {
return (new WepAgent( ));
}
} class_wep;

WepAgent::WepAgent() : Agent(PT_WEP), t_seqno_(0), ts_peer_(0), ts_echo_(0),
curseq_(0), maxseq_(0), highest_cntl_(0), cwnd_(0), tss(NULL), tss_size_(100),
lastreset_(0.0), closed_(0), rtx_timer_(this), delsnd_timer_(this), burstsnd_timer_(this) {

// Bind feito pelo construtor do Objeto WEP ( nome da var. em OTCL, endereço da var.
compilada)
#ifdef WEP_DELAY_BIND_ALL
#else /* ! WEP_DELAY_BIND_ALL */
// Não delay-bound por que não existe suporte a delay-bound tracevars ainda
bind("t_seqno_", &t_seqno_);
bind("cwnd_", &cwnd_);
bind("seqno_", &curseq_);
bind("cntl_", &highest_cntl_);
bind("maxseq_", &maxseq_);
bind("ndatapack_", &ndatapack_);
bind("ndatabytes_", &ndatabytes_);
bind("nackpack_", &nackpack_);

#endif /* WEP_DELAY_BIND_ALL */
}

/* A seguir é feita as ligações das variáveis em C++ para OTCL
baseado no arquivo ns~/tcl/lib/ns-default.tcl que
contém os valores default para inicialização das variáveis. */

void WepAgent::delay_bind_init_all() {

//Valores default das variaveis iniciadas no ns-default.tcl.
delay_bind_init_one("useHeaders_");
```

```
delay_bind_init_one("first_");
delay_bind_init_one("wpaTick_");
delay_bind_init_one("packetSize_");
delay_bind_init_one("wpaip_base_hdr_size_");
delay_bind_init_one("maxburst_");
delay_bind_init_one("maxcwnd_");
delay_bind_init_one("nackpack_");
delay_bind_init_one("nam_tracevar_");
delay_bind_init_one("overhead__");

#ifdef WEP_DELAY_BIND_ALL
delay_bind_init_one("t_seqno_");
delay_bind_init_one("seqno_");
delay_bind_init_one("cntl_");
delay_bind_init_one("maxseq_");
delay_bind_init_one("ndatapack_");
delay_bind_init_one("ndatabytes_");
delay_bind_init_one("cwnd_");
delay_bind_init_one("trace_all_oneline_");
delay_bind_init_one("numdupacks_");

#endif /* WEP_DELAY_BIND_ALL */

Agent::delay_bind_init_all();

reset(); // Reinicializa as Variáveis
}

// Bind para o arquivo de trace
int WepAgent::delay_bind_dispatch(const char *varName, const char *localName, TclObject
*tracer){
if (delay_bind_bool(varName, localName, "first_", &first_, tracer)) return TCL_OK;
if (delay_bind(varName, localName, "wpaTick_", &tcp_tick_, tracer)) return TCL_OK;
if (delay_bind(varName, localName, "packetSize_", &size_ , tracer)) return TCL_OK;
if (delay_bind_bool(varName, localName, "useHeaders_", &useHeaders_, tracer)) return
TCL_OK;
if (delay_bind(varName, localName, "wpaip_base_hdr_size_", &wpaip_base_hdr_size_,
tracer)) return TCL_OK;
if (delay_bind(varName, localName, "cwnd_", &cwnd_ , tracer)) return TCL_OK;
if (delay_bind(varName, localName, "maxburst_", &maxburst_ , tracer)) return TCL_OK;
if (delay_bind(varName, localName, "nackpack_", &nackpack_ , tracer)) return TCL_OK;
if (delay_bind(varName, localName, "overhead_", &overhead_, tracer)) return TCL_OK;
if (delay_bind_bool(varName, localName, "nam_tracevar_", &nam_tracevar_ , tracer)) return
TCL_OK;

#ifdef WEP_DELAY_BIND_ALL
// not if (delay-bound - não existe suporte a delay-bound tracevars ainda
if (delay_bind(varName, localName, "t_seqno_", &t_seqno_ , tracer)) return TCL_OK;
if (delay_bind(varName, localName, "seqno_", &curseq_ , tracer)) return TCL_OK;
if (delay_bind(varName, localName, "cntl_", &highest_cntl_ , tracer)) return TCL_OK;
```

```
if (delay_bind(varName, localName, "maxseq_", &maxseq_ , tracer)) return TCL_OK;
if (delay_bind(varName, localName, "ndatapack_", &ndatapack_ , tracer)) return TCL_OK;
if (delay_bind(varName, localName, "ndatabytes_", &ndatabytes_ , tracer)) return TCL_OK;
if (delay_bind_bool(varName, localName, "trace_all_oneline_", &trace_all_oneline_ ,
tracer)) return TCL_OK;

#endif

//Método reset() das variáveis

void WepAgent::reset() {
curseq_ = 0;
t_seqno_ = 0;
maxseq_ = -1;
last_cntl_ = -1;
highest_cntl_ = -1;
ndatapack_ = 0;
ndatabytes_ = 0;
nackpack_ = 0;
lastreset_ = Scheduler::instance().clock();
closed_ = 0;
}

/*
 * Analisa o tamanho do pacote WEP+IP
*/

int WepAgent::headersize() {
int total = wpaip_base_hdr_size_;
if (total < 1) {
fprintf(stderr, "WepAgent(%s): warning: o tamanho do pacote wep+ip é menor que 1: %d
bytes\n", name( ), wpaip_base_hdr_size_);
}
}

void WepAgent::output(int seqno, int reason) {
int tipo;                       //tipo de pacote
Packet* p = allocpkt( );        //Cria e aloca um pacote p
hdr_wep *wpah = hdr_wep::access(p); /*cria um ponteiro de acesso ao cabeçalho wep para o
pacote p*/
hdr_flags* hf = hdr_flags::access(p); /*cria um ponteiro de acesso as flags para o pacote p*/
hdr_ip *iph = hdr_ip::access(p);        /*cria um ponteiro de acesso ao cabeçalho ip para o
pacote p*/
int databytes = hdr_cmn::access(p)->size(); // databytes recebe tamanho do header comum
wpah->seqno() = seqno; // o ponteiro coloca no campo seqno o num. de sequencia
wpah->ts() = Scheduler::instance().clock(); // o ponteiro coloca no ts o instante atual
wpah->ts_echo() = ts_peer_; //o ponteiro coloca em ts_echo o conteudo de ts_peer recebido
wpah->reason() = reason; // campo que informa a razão para transmitir

//Abertura de sessão
```

```
/* Checa se é pra enviar um FIRST packet. Se o numero de sequência inicial for igual a 0 e
variável first_setada preenche com  0 bytes de payload, numero de seqüência , e tamanho do
header WEP+IP é passado para o header comum. O campo ptype =2 indica que é um pacote
FIRST */

if (seqno == 0) {
if (first_) {
databytes = 0;
curseq_ += 1;
hdr_cmn::access(p)->size() = wpaip_base_hdr_size_;
tipo =2;
wpah->ptype() = tipo ; // indica que é um pacote FIRST
}
}else{ /*Se não for mandar first, preenche o tamanho do hdr comum com WEP+IP + size.*/
if (useHeaders_ == true) {
hdr_cmn::access(p)->size() += headersize();
}
}
wpah->ptype == 1 ;    // Seta a flag type para indicar que vai enviar um   pacote de dados
hdr_cmn::access(p)->size();
++ndatapack_;                    // soma +1 na variavel que indica pacote de dados
ndatabytes_ += databytes; /* bytes do pacote de dados recebe numero   de bytes enviados/*
send(p, 0);       // envia o pacote, metodo da classe Connector::send()

if (seqno = = curseq_ && seqno > maxseq_)
idle();  // Avisa a aplicação que já enviou tudo

if (seqno > maxseq_) {
maxseq_ = seqno;
} else {
++nrexmitpack_; // para trabalhos futuros(controle de fluxo)
nrexmitbytes_ += databytes;
}
}

/*
 * Recepção  dos pacote de resposta do WEPRecv
 */

void WepAgent::recv(Packet *pkt, Handler*) {
hdr_wep *wpah = hdr_wep::access(pkt); //acessa o pacote recebido
int valid_cntl = 0;

#ifdef notdef /* Se não for um ACK= CNTL  informa que ele recebeu um pacote diferente de
ACK */
if (pkt->type_ != PT_ACK) {
Tcl::instance().evalf("%s erro \" un não-cntl recebido"", name());
Packet::free(pkt);
return;
```

```
}
#endif

/* Análise do  Pacote CNTL  de negociação da janela: acessa os   campos e seta o novo
tamanho da janela inicial enviado pelo receptor*/
ptype_= wpah->ptype();
if(ptype_= = 1){
cwind_= wpah->cwind();
cwnd_= cwind_;
}

if (wpah->ts() < lastreset_) {
/* Se o tempo do pacote for menor que o último reset ,não faz nada */
Packet::free(pkt);
return;
}

++nackpack_; //incrementa o número de pacotes cntl recebidos
ts_peer_ = wpah->ts(); //coloca o em  ts_peer o tempo em que foi enviado o pacote pelo WEP
receptor
recv_helper(pkt); /* função

/* Checa o numero de sequência do cntl e compara com o ultimo cntl. Essa função é
importante para realizar o controle de fluxo(tamanho da janela) que não foi implementado*/

if (weph->seqno() > last_cntl_) {
// Reajusta a janela
} else if (weph->seqno() == last_cntl_) {
if (dupacks_ < numdupacks_ ) {
send_one();
}
}

if (weph->seqno() >= last_cntl_)
/* Checa pra ver se é uma resposta válida.Se for seta a variável valid_cntl. Contribuição do
código tcp */
valid_cntl= 1;

Packet::free(pkt);
/* Tenta enviar pacote que a janela permite (para trabalhos futuros)

if (valid_cntl || aggressive_maxburst_)
//send_much(0, 0, maxburst_);
}

**********************************weprecv.h*******************************
#include <math.h>
#include "agent.h"
#include "wep.h"
```

```
class WepRecv;
class WepRecv : public Agent { // Classe WepRecv
public:
WepRecv(); //constructor da classe
void recv(Packet* pkt, Handler*); // método recv() dos pacotes
void reset(); // método reset() do Agente

protected:
void cntl(Packet*); //método que prepara o pacote CNTL para envio
virtual void delay_bind_init_all(); // bind das variáveis
virtual int delay_bind_dispatch(const char *varName, const char *localName, TclObject
*tracer);
int bytes_;
double lastreset;
int next_;       /* proximo pacote esperado */
int maxseen_; /* maximo numero pacote seen */
int *seen_;      /* array de pacotes  seen */
double ts_to_echo_; /* timestamp do agente emissor */

**********weprecv.cc**********
#include "flags.h"
#include "ip.h"
#include "weprecv.h"

static class WepRecvClass : public TclClass {
public:
WepRecvClass() : TclClass("Agent/WEPRecv") {}
TclObject* create(int, const char*const*) {
return (new WepRecv());
}
} class_weprecv;

WepRecv::WepRecv: Agent(PT_ACK), next_(0), maxseen_(0), ts_to_echo_(0),
last_cntl_sent_(0)

void WepRecv::delay_bind_init_all() {
delay_bind_init_one("packetSize_");
delay_bind_init_one("bytes_");
Agent::delay_bind_init_all();
}

int WepRecv::delay_bind_dispatch(const char *varName, const char *localName, TclObject
*tracer) {
if (delay_bind(varName, localName, "packetSize_", &size_, tracer)) return TCL_OK;
return Agent::delay_bind_dispatch(varName, localName, tracer);
}

void WepRecv::reset() {
save_ = NULL;
```

```
lastreset_ = Scheduler::instance().clock(); /* guarda o tempo corrente na variável lastreset_,
para detectar pacotes de simulações presos*/
}

void WepRecv::recv(Packet* pkt, Handler*) {
int numToDeliver;
int window;     //tamanho da janela
int numBytes = hdr_cmn::access(pkt)->size(); /* acessa o cabeçalho comum  e guarda em
numBytes o tamanho dos bytes que chegaram*/
hdr_wep *th = hdr_wep::access(pkt); /* acessa o cabeçalho Wep

/* Checa se o pacote é algum que estava preso(de outra simulação)*/
if (th->ts() < lastreset_) { // Se for descarta o pacote e não faz nada
Packet::free(pkt);
return;
}

//Abertura de sessão
// Se o pacote  for um FIRST envia um CNTL com o tamanho da janela
if(th->ptype() = ptype_= = 2){ // o pacote é um pacote FIRST?
Packet* npkt = allocpkt(); /aloca um novo pacote
window=1; //seta a janela =1
hdr_wep *nwep = hdr_wep::access(npkt); //acessa o cabeçalho
nwep->cwind()= window; //seta o tamanho da janela
send(npkt,0); //envia o pacote CNTL  com janela ,para o Agente WEP emissor
} else { /*Se não for um pacote FIRST , envia CNTL normal para cada pacote de dados
recebido */
cntl(pkt); // invoca o método que prepara o envio  do pacote cntl
Packet :: free(pkt); //libera o pacote recebido
}
}

void WepRecv::cntl(Packet* opkt) {
Packet* npkt = allocpkt // npkt é o pacote novo que será construído
double now = Scheduler::instance().clock(); /*coloca em "now "o tempo de envio*/
hdr_wep *owep = hdr_wep::access(opkt); /*acessa o cabeçalho do pacote wep recebido*/
hdr_ip *oiph = hdr_ip::access(opkt); /*acessa o cabeçalho do pacote ip recebido*/
hdr_wep *nwep = hdr_wep::access(npkt); );  /*acessa o cabeçalho do novo pacote wep */
nwep->ts_echo() = owep->ts(); /*novo pacote recebe o tempo do pacote antigo*/
nwep->seqno() = owep->seqno(); /* pacote recebe um numero de sequência*/
nwep->ts() = now; /* pacote recebe o tempo atual de envio */
hdr_ip* nip = hdr_ip::access(npkt);  /*acessa o cabeçalho do novo pacote  ip */
nip->flowid() = oip->flowid(); /* copia a identificação de fluxo do pacote velho para o novo*/
send(npkt, 0); /* finalmente envia o pacote CNTL de resposta */
}
```

APÊNDICE A – Script Otcl para Simulação de rede sem fio

```
############################## projeto.tcl ################################
set val(chan) Channel/WirelessChannel ;          # Tipo de canal
set val(prop) Propagation/TwoRayGround ;     # Modelo de Propagação do Sinal
set val(ant) Antenna/OmniAntenna ;           # Tipo de Antena
set val(ll) LL ;                             # Tipo da camada de Link
set val(ifq) Queue/DropTail/PriQueue ;       # Tipo da fila de pacotes
set val(ifqlen) 50 ;                         # max packet in ifq
set val(netif) Phy/WirelessPhy ;             # Tipo de Interface de rede
set val(mac) Mac/802_11 ;                    # MAC type
set val(rp) DSDV ;                           #  Tipo de roteamento ad-hoc
set val(nn) 2 ;                              # number of mobilenodes

set ns_  [new Simulator] ;   #cria uma instancia do simulador ns
set topo [new Topography] ; #cria a topology e
 $topo load_flatgrid 670 670   ; #define a area em  670x670

#Define standard ns/nam trace
set tracefd  [open demo.tr w]
$ns_ trace-all $tracefd
set namtrace [open demo.nam w]
$ns_ namtrace-all-wireless $namtrace 670 670

#Create  "God"
set god_ [create-god 3]
#God é usado para armazenar um array com o menor numero de #hops entre dois nodos

$ns_ at 899.00 "$god_ setdist 2 3 1"

#Definição de um nodo movel
$ns_ node-config
       -adhocRouting DSDV or DSR or TORA #
     -llType LL \
    -macType Mac/802_11\
    -ifqLen 50 \
    -ifqType Queue/DropTail/PriQueue \
    -antType Antenna/OmniAntenna \
    -propType Propagation/TwoRayGround \
    -phyType Phy/WirelessPhy \
    -channelType Channel/WirelessChannel \
    -topoInstance $topo
    -agentTrace ON \
    -routerTrace OFF \
    -macTrace OFF

#define o modelo de energia do nodo
$ns_ node-config -energyModel EnergyModel
                    -initialEnergy 100.0
        -txPower 0.6
  -rxPower 0.2

#Cria um nodo móvel e coloca -o no canal
set node [$ns_ node]
```

```
$node random-motion 0 ;# disable random motion

# Loop para criar 3 nodos
for {set i < 0} {$i<3} {incr i} {
   set node_($i) [$ns_ node]
}

#Define a posição inicial do nodo no nam
$ns_  at 200.0 “$ns_ nam-end-wireless 200.00”
$ns_ at 200.00 “$ns_ halt”

#Cria o agente WEP e WEPRecv
set wep [new Agent/WEP]
set weprecv [new Agent/WEPRecv]

#Conecta o dois agentes
$ns connect $wep $weprecv

#Cria a aplicação FTP em cima do WEP
set ftp [new Application/FTP]
$ftp attach-agent $wep
$ns at 1.2 “ftp start”

# Envia First para negociação de janela
$wep set first_ true

#Coloca os agentes nos nós criados
$ns attach-agent $n0 $wep
$ns attach-agent $n1 $wep
$ns attach-agent $n2 $weprecv

#Start your simulation
$ns_  run
```