

UNIVERSIDADE ESTADUAL PAULISTA  
FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

## **Análise e Desempenho de Redes de Acesso Sem Fio**

**Ligia Rodrigues Prete**

**Orientador: Prof. Dr. Ailton Akira Shinoda**

Ilha Solteira – SP, Abril de 2011



UNIVERSIDADE ESTADUAL PAULISTA  
"JÚLIO DE MESQUITA FILHO"  
Campus de Ilha Solteira

**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

“Análise e Desempenho de Redes de Acesso Sem Fio”

**LÍGIA RODRIGUES PRETE**

**Orientador:** Prof. Dr. Ailton Akira Shinoda

Dissertação apresentada à Faculdade de Engenharia - UNESP - Campus de Ilha Solteira, para obtenção do título de Mestre em Engenharia Elétrica.  
Área de Conhecimento: Automação.

Ilha Solteira - SP  
Abril/2011

## FICHA CATALOGRÁFICA

Elaborada pela Seção Técnica de Aquisição e Tratamento da Informação  
Serviço Técnico de Biblioteca e Documentação da UNESP - Ilha Solteira.

P912a

Prete, Lígia Rodrigues.

Análise e desempenho de redes de acesso sem fio / Lígia Rodrigues Prete. -- Ilha Solteira: [s.n.], 2011  
140 f.: il.

Dissertação (mestrado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira. Área de conhecimento: Automação, 2011

Orientador: Ailton Akira Shinoda

Inclui bibliografia

1. Redes de computadores. 2. Simulação (Computadores). 3. Redes de computação – Protocolos. 4. Mobilidade do usuário. 5. Vazão na transmissão de dados.



**UNIVERSIDADE ESTADUAL PAULISTA**  
CAMPUS DE ILHA SOLTEIRA  
FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA

### CERTIFICADO DE APROVAÇÃO

**TÍTULO:** Análise e Desempenho de Redes de Acesso Sem Fio

**AUTORA:** LÍGIA RODRIGUES PRETE

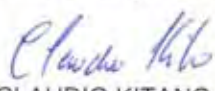
**ORIENTADOR:** Prof. Dr. AILTON AKIRA SHINODA

Aprovada como parte das exigências para obtenção do Título de Mestre em Engenharia Elétrica ,  
Área: AUTOMAÇÃO, pela Comissão Examinadora:



Prof. Dr. AILTON AKIRA SHINODA

Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira



Prof. Dr. CLAUDIO KITANO

Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira



Prof. Dr. RUY DE OLIVEIRA

Departamento de Informática / Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso - Cuiabá/MT

Data da realização: 29 de abril de 2011.

*Dedico aos meus queridos pais, Nelson  
Antonio Prete e Zenaide Rodrigues Prete, e a  
minha irmã Márcia Donizeth Prete*

## **AGRADECIMENTOS**

A Deus que nos concede o direito à vida.

À família sempre presente nos momentos mais difíceis.

Ao Professor Dr. Ailton Akira Shinoda, pela compreensão e por ter se mostrado um exemplo de orientador, na elaboração deste trabalho. Agradeço por sua dedicação, seus ensinamentos e por toda sua paciência no desenvolvimento da pesquisa e das simulações.

À banca examinadora que analisará o conteúdo deste trabalho, contribuindo com sugestões.

*“Sofremos demasiado pelo pouco que nos falta  
e alegramo-nos pouco pelo muito que temos...”*

*Autor: William Shakespeare*

## RESUMO

Este estudo provê uma análise de redes infraestruturadas (com e sem fio), usando o simulador de redes de computadores *Network Simulator*. Este simulador oferece suporte à simulação de um grande número de tecnologias de rede, fazendo um detalhamento de suas características e funcionalidades, por meio da implementação de modelos de sistemas de redes, mostrando resultados de simulações. No decorrer deste estudo, serão abordados três tipos de redes de acesso sem fio, para avaliação de desempenho, iniciando com uma topologia de rede simples e sem nenhuma mobilidade do usuário, estendendo para uma topologia de rede intermediária, em que o usuário terá mobilidade restrita até uma topologia de rede complexa e sem nenhuma restrição de mobilidade para o usuário. O primeiro tipo de rede, “Rede de Acesso Sem Fio e Sem Mobilidade”, terá uma topologia de rede simples infraestruturada e sem mobilidade dos nós, composta por um nó cabeado, cinco estações base e vinte nós sem fio. O objetivo é avaliar o desempenho da rede em função da vazão média e instantânea na transmissão de notícias em tempo real e o comportamento da rede em função da vazão média na transmissão de arquivos dentro de uma intranet. Já o segundo tipo de rede, “Rede de Acesso Sem Fio e com Mobilidade Restrita” (movimentação dos nós em uma única direção), a topologia de rede intermediária infraestruturada, será composta por um nó cabeado, seis estações base e sessenta nós sem fio. A finalidade é analisar a movimentação dos nós sem fio e a comparação média de vazão, quando ocorre o aumento de nós na rede, transmitindo dados em um ambiente externo, considerando uma avenida com uma distância de 2000 metros. Finalmente, o terceiro tipo de rede, “Rede de Acesso Sem Fio e com Mobilidade Irrestrita” (movimentação dos nós em várias direções), a topologia de rede complexa infraestruturada será composta por um nó cabeado, nove estações base e trinta nós sem fio. O objetivo é avaliar o desempenho da vazão média e as possíveis razões de descarte, quando usuários se movimentam e trocam dados em um ambiente externo, considerando uma área geográfica de 500 x 500 metros com um conglomerado de blocos de prédios divididos por ruas ou avenidas. Por fim, serão apresentados os gráficos com os resultados das análises de desempenho nos três tipos de rede infraestruturada.

**Palavras-chave:** Rede. Simulação. Protocolo. Mobilidade. Vazão.



## ABSTRACT

This study foresees an analysis of infrastructured network (wired and wireless), using the network simulator computer Network Simulator. This simulator supports simulation of a large number of networking technologies, making a detailment of its features and functionalities, through the improvements of models of network system, showing results of simulations. During this study, it will be approached three types of wireless access networks, for performance assessment starting with a simple network topology and without user's mobility, extending to an intermediate network topology, where the user will have restricted mobility to a complex network topology and without any restriction of mobility for the user. The first type of network, "Wireless Network Access and Without Mobility" will have a topology of a simple and infrastructured network and without node mobility, consisting of a wired node, five base station and twenty wireless nodes. The objective is to evaluate the network performance depending on the instantaneous and average throughput in the transmission of news in real time and network behavior depending on the average throughput in the transmission of files within an intranet service. The second type of network, "Wireless Network Access and with Restricted Mobility" (movement of the nodes in a single direction), the topology of an intermediate infrastructured network will be composed by a wired node, six base stations and sixty wireless nodes. The aim is to analyze the movement of the wireless nodes and the comparison of the average throughput, when there is an increase of nodes in the network, transmitting data in an external environment, considering an avenue with a distance of 2000 meters. Finally, the third type of network, "Wireless Network Access and with Unrestricted Mobility" (movement of the nodes in various directions), the topology of the infrastructured and complex network will be composed by a wired node, nine base stations and thirty wireless nodes. The objective is to evaluate the performance of the average throughput and the possible reasons for disposal, when users move and exchange data in an external environment, considering a geographical area of 500 x 500 meters with a cluster of buildings blocks divided by streets or avenues. Finally, the graphics will be presented with the results of the performance analyses in the three types of the infrastructured network.

**Keywords:** Network. Simulation. Protocol. Mobility. Throughput.

## LISTA DE ILUSTRAÇÕES

Figura 2.1 -	Posição do IEEE 802.11 no modelo de referência OSI.....	20
Figura 2.2 -	Camada física (PHY) e de enlace (MAC) em uma rede 802.11 .....	21
Figura 2.3 -	Rede infraestruturada .....	21
Figura 2.4 -	Rede <i>ad hoc</i> .....	22
Figura 2.5 -	Funcionamento da função DCF .....	26
Figura 2.6 -	Problema do terminal escondido.....	26
Figura 2.7 -	Funcionamento da função DCF (RTS/CTS).....	27
Figura 2.8 -	Funcionamento da função PCF (ERGEN, 2002).....	28
Figura 2.9 -	Quadro MAC do padrão 802.11.....	29
Figura 3.1 -	Correspondência entre objetos C++ e Otcl .....	35
Figura 3.2 -	Estrutura do NS.....	36
Figura 3.3 -	Arquitetura geral do NS .....	38
Figura 4.1 -	Criação de rota para o nó MH4 empregando o Protocolo DSDV.....	41
Figura 4.2 -	Mudança na topologia da Figura 4.1.....	42
Figura 4.3 -	Criação de rota no protocolo AODV (PERKINS; ROYER, 1999) .....	44
Figura 4.4 -	Criação de rota no protocolo DSR (JOHNSON; MALTZ, 1996) .....	45
Figura 4.5 -	Criação de rota no protocolo TORA (PARK; CORSON, 1997) .....	46
Figura 4.6 -	Exemplo de uma rede sem fio <i>ad hoc</i> com três nós.....	47
Figura 5.1 -	Topologia da rede sem nenhuma mobilidade do usuário.....	61
Figura 5.2 -	Estatística média de vazão dos pacotes transmitidos do nó cabeado W(0) até as estações bases.....	63
Figura 5.3 -	Vazão instantânea do nó cabeado W(0) até os nós sem fio relativos à EB1 ....	64
Figura 5.4 -	Vazão instantânea do nó cabeado W(0) até os nós sem fio relativos à EB2 ....	65
Figura 5.5 -	Vazão instantânea do nó cabeado W(0) até os nós sem fio relativos à EB3 ....	65
Figura 5.6 -	Vazão instantânea do nó cabeado W(0) até os nós sem fio relativos à EB4 ....	66
Figura 5.7 -	Vazão instantânea do nó cabeado W(0) até os nós sem fio relativos à EB5 ....	67
Figura 5.8 -	Estabelecimento de conexão TCP, entre os nós sem fio, por meio da EB1 .....	69
Figura 5.9 -	Estatística média de vazão do roteamento de pacotes, feito pelos nós sem fio pertencentes à área de cobertura de sua respectiva estação base. ....	70
Figura 5.10 -	Estabelecimento de conexão TCP, entre o nó sem fio n6 e o nó fixo W(0), sendo roteado pela EB1. ....	71

Figura 5.11 - Estatística média de vazão do roteamento de pacotes transmitidos dos nós sem fio até as estações base e estas, roteam os pacotes até o nó fixo $W(0)$ .....	71
Figura 5.12 - Estabelecimento de conexão TCP, entre o nó sem fio $n6$ e o nó sem fio $n11$ , roteados por meio da EB1, EB2 e $W(0)$ . .....	72
Figura 5.13 - Estatística média de vazão do roteamento de pacotes feito pelos nós sem fio de diferentes estações base. ....	73
Figura 5.14 - Topologia da rede com mobilidade restrita do usuário.....	74
Figura 5.15 - Movimento do nó sem fio $N1$ em função do tempo quando este se movimenta por uma avenida em um ambiente externo. ....	77
Figura 5.16 - Distribuição dos nós sem fio conectados às estações base em função do tempo.....	78
Figura 5.17 - Comparação média de vazão à medida que as mesmas escalonam .....	79
Figura 5.18 - Topologia da rede com mobilidade irrestrita do usuário .....	80
Figura 5.19 - Estatística média de vazão (com aumento normalizado de nós sem fio).....	83
Figura 5.20 - Estatística média de vazão (com aumento normalizado de mobilidade) .....	84
Figura 5.21 - Razão do descarte de pacotes com 10% da carga do sistema (3 nós sem fio)..	85
Figura 5.22 - Razão do descarte de pacotes com 20% da carga do sistema (6 nós sem fio)..	86
Figura 5.23 - Razão do descarte de pacotes com 30% da carga do sistema (9 nós sem fio)..	86
Figura 5.24 - Relação média de pacotes descartados entre 10% a 30% da carga do sistema.	87

## LISTA DE TABELAS

Tabela 2.1 -	Características das variantes do padrão 802.11 .....	23
Tabela 4.1 -	Tabela de roteamento da Figura 4.1 .....	41
Tabela 4.2 -	Tabela de roteamento da Figura 4.1 .....	42
Tabela 4.3 -	<i>Script</i> para uma rede <i>ad hoc</i> .....	47
Tabela 4.4 -	Configuração dos componentes dos <i>MobileNodes</i> .....	50
Tabela 4.5 -	Inicialização de variáveis para abertura de escrita de arquivos .....	51
Tabela 4.6 -	Funções que realizam <i>log</i> do comportamento da rede.....	51
Tabela 4.7 -	Criação da topografia da rede .....	51
Tabela 4.8 -	Criação do objeto GOD .....	52
Tabela 4.9 -	Configuração dos <i>MobileNodes</i> .....	52
Tabela 4.10 -	Criação dos <i>MobileNodes</i> .....	52
Tabela 4.11 -	Posição inicial e movimentação dos <i>MobileNodes</i> .....	53
Tabela 4.12 -	Criação da conexão TCP entre os <i>MobileNodes</i> .....	54
Tabela 4.13 -	Definição da posição inicial do <i>MobileNode</i> no NAM.....	54
Tabela 4.14 -	Aviso aos <i>MobileNodes</i> do tempo final da simulação .....	55
Tabela 4.15 -	Procedimento de encerramento da simulação.....	55
Tabela 4.16 -	Execução da simulação de exemplo <i>wireless_adhoc.tcl</i> .....	55
Tabela 4.17 -	Execução do programa “ <i>setdest</i> ” para geração de um cenário de rede sem fio	56
Tabela 4.18 -	Geração do padrão de tráfego com o <i>script</i> “ <i>cbrgen.tcl</i> ” .....	57
Tabela 4.19 -	Configuração do novo formato <i>trace</i> .....	58
Tabela 4.20 -	Exemplo de arquivo com o novo formato <i>trace</i> .....	58
Tabela 4.21 -	Especificação dos campos do novo formato <i>trace</i> .....	59
Tabela 5.1 -	Parâmetros da rede considerada na transmissão de notícias em tempo real .....	62
Tabela 5.2 -	Parâmetros da rede considerada na transmissão de arquivos em uma intranet.	68
Tabela 5.3 -	Parâmetros da rede considerado na simulação de transmissão de dados em um ambiente externo.....	75
Tabela 5.4 -	Criação do cenário de mobilidade com o <i>setdest</i> .....	76
Tabela 5.5 -	Parâmetros da rede considerada na simulação de transmissão de dados em um ambiente externo .....	81
Tabela 5.6 -	Criação do cenário de mobilidade com o <i>BonnMotion</i> .....	82
Tabela 5.7 -	Conversão de arquivos com a aplicação <i>NSFile</i> .....	82

## LISTA DE SIGLAS

ACK	<i>Acknowledgment</i> (Reconhecimento)
AODV	<i>Ad Hoc On Demand Distance Vector</i> (Vetor de Distância <i>Ad Hoc</i> Sob Demanda)
AGT	<i>Agent</i> (Agente)
AP	<i>Access Point</i> (Ponto de Acesso)
ARP	<i>Address Resolution Protocol</i> (Protocolo de Resolução de Endereços)
AWK	Alfred V. <b>Aho</b> , Peter J. <b>Weinberger</b> e Brian W. <b>Kernighan</b>
bps	bits por segundo
BSS	<i>Basic Service Set</i> (Conjunto Básico de Serviço)
CBQ	<i>Class-Based Queueing</i> (Enfileiramento Baseado em Classes)
CBK	<i>Callback</i> (Retorno)
CBR	<i>Constant Bit Rate</i> (Taxa de Bit Constante)
CMU	<i>Carnegie Mellon University</i> (Universidade <i>Carnegie Mellon</i> )
CONSER	<i>Collaborative Simulation for Education and Research</i> (Simulação de Colaboração para Educação e Pesquisa)
CRC	<i>Cyclic Redundancy Check</i> (Checa Erros de Transmissão)
CSMA/CA	<i>Carrier Sense Multiple Access/Collision Avoidance</i> (Acesso Múltiplo com Detecção de Portadora/Prevenção de Colisão)
CTS	<i>Clear to Send</i> (Liberado para Envio)
DARPA	<i>Defense Advanced Research Projects Agency</i> (Agência de Apoio a Projetos e à Pesquisa Avançada)
DCF	<i>Distributed Coordination Function</i> (Função de Coordenação Distribuída)
DFWMAC	<i>Distributed Foundation Wireless Medium Access Control</i> (Controle de Acesso ao Meio Sem Fio por Base de Distribuição)
DIFS	<i>Distributed Inter Frame Space</i> (Espaço Distribuído Entre Quadros)

DRR	<i>Deficit Round Robin</i>
DS	<i>Distribution System</i> (Sistema de Distribuição)
DSDV	<i>Destination-Sequenced Distance-Vector</i> (Vetor Distância com Destino Sequenciado)
DSR	<i>Dynamic Source Routing</i> (Roteamento Dinâmico pela Fonte)
DSSS	<i>Direct Sequence Spread Spectrum</i> (Espalhamento Espectral de Sequência Direta)
EB	Estação Base
ESM	Estação de Suporte à Mobilidade
FCS	<i>Frame Check Sequence</i> (Sequência de Verificação do Quadro)
FHSS	<i>Frequency Hopping Spread Spectrun</i> (Espalhamento Espectral por Salto em Frequência)
FQ	<i>Fair Queueing</i> (Enfileiramento no Buffer)
FTP	<i>File Transfer Protocol</i> (Protocolo de Transferência de Arquivos)
GHz	<i>Gigahertz</i>
GOD	General Operations Director (Diretor de Operações Gerais)
HM	Host Móvel
HR-DSSS	<i>High Rate - Direct Sequence Spread Spectrum</i> (Taxa Alta de Sequência Direta de Espalhamento do Espectro)
HTTP	<i>Hypertext Transfer Protocol</i> (Protocolo de Transferência de Hipertexto)
ICSI	<i>International Computer Science Institute</i> (Instituto Internacional de Ciência e Computação)
IEEE	<i>Institute of Electrical and Eletronics Engineers</i> (Instituto de Engenharia Elétrica e Eletrônica)
IFQ	<i>Interface Queue</i> (Interface de Fila)
IFS	<i>Inter Frame Space</i> (Espaço entre Frames)
IP	<i>Internet Protocol</i> (Protocolo de Internet)
Kbps	Kilobits por segundo

LANs	Local Area Networks (Redes de Área Local)
LBNL	<i>Lawrence Berkeley National Laboratory</i> (Laboratório Nacional <i>Lawrence Berkeley</i> )
LLC	<i>Logic Link Control</i> (Controle Lógico do <i>Link</i> )
MAC	<i>Media Access Control</i> (Controle de Acesso ao Meio)
MANET	<i>Mobile Ad hoc Network</i> (Rede <i>Ad Hoc</i> Móvel)
Mbps	Megabits por segundo
Monarch	<i>Mobile Networking Architectures</i> (Arquitetura de Rede Móvel)
NAM	<i>Network AniMator</i> (Animação de Rede)
NAV	<i>Network Allocation Vector</i> (Vetor de Alocação da Rede)
NIST	<i>National Institute of Standard and Technology</i> (Instituto Nacional de Padrões e Tecnologia)
NS	<i>Network Simulator</i> (Simulador de Redes)
NSF	<i>National Science Foundation</i> (Fundação Nacional de Ciência)
OFDM	<i>Orthogonal Frequency Division Multiplexing</i> (Multiplexação Ortogonal por Divisão de Frequência)
OSI	<i>Open System Interconnection</i> (Interconexão de Sistemas Abertos)
OTcl	<i>Object-oriented Tool Command Language</i> (Linguagem de Comandos Orientada a Objeto)
PARC	<i>Palo Alto Research Center</i> (Centro de Pesquisa Palo Alto)
PCF	<i>Point Coordination Function</i> (Função de Coordenação de Ponto)
PHY	<i>Physical Layer</i> (Camada Física)
PIFS	<i>Priority Inter Frame Space</i> (Espaço Prioritário Entre Quadros)
pps	pacotes por segundo
QoS	<i>Quality of Service</i> (Qualidade de Serviço)
RED	<i>Random Early Discard</i> (Descarte Aleatório Antecipado)
RTR	<i>Router</i> (Roteamento)

RTS	<i>Request to Send</i> (Requisição de Envio)
SAMAN	<i>Simulation Augmented by Measurement and Analysis for Networks</i> (Simulação Aumentada de Medição e Análise de Redes)
SFQ	Stochastic Fair Queueing (Enfileiramento Estocástico no Buffer)
SIFS	<i>Short Inter Frame Space</i> (Espaço Curto Entre Quadros)
SSH	Secure Shell (Rede Segura)
SSID	<i>Service Set Identifier</i> (Identificador do Conjunto de Serviço)
TclCL	<i>Tool Command Language with Classes</i> (Linguagem de Comandos de Ferramentas com Classes)
TCP	<i>Transmission Control Protocol</i> (Protocolo de Controle de Transmissão)
TORA	<i>Temporally Ordered Routing Algorithm</i> (Algoritmo de Roteamento Ordenado Temporariamente)
TTL	<i>Time To Live</i> (Tempo de Vida)
UCB	<i>University of California Berkeley</i> (Universidade Berkeley da Califórnia)
UDP	<i>User Datagram Protocol</i> (Protocolo de Transmissão de Dados)
USC/ISI	<i>University of Southern California/ Information Sciences Institute</i>
VBR	<i>Variable Bit Rate</i> (Taxa de Bit Variável)
VINT	<i>Virtual InterNetwork Testbed</i> (Teste Virtual de Rede)
WEP	Wired Equivalent Privacy (Segurança Equivalente à das Redes Cabeadas)
WiMAX	<i>Worldwide Interoperability for Microwave Access</i> (Interoperabilidade Mundial para Acesso de Micro-ondas)
WLAN	<i>Wireless Local Area Networks</i> (Rede Local Sem Fio)



# SUMÁRIO

1	INTRODUÇÃO.....	18
2	REDES LOCAIS SEM FIO PADRÃO IEEE 802.11.....	20
2.1	Arquitetura.....	21
2.2	Camada Física.....	22
2.3	Camada de Enlace.....	23
2.3.1	<i>Subcamada LLC</i> .....	24
2.3.2	<i>Subcamada MAC</i> .....	24
2.3.2.1	<i>Função de Coordenação Distribuída (DCF)</i> .....	25
2.3.2.2	<i>Função de Coordenação Pontual (PCF)</i> .....	27
2.3.3	<i>Estrutura de Quadros</i> .....	29
3	O SIMULADOR DE REDES NETWORK SIMULATOR.....	33
3.1	Breve Histórico.....	33
3.2	A Linguagem do NS.....	34
3.3	Arquitetura do NS.....	37
4	SIMULAÇÃO DE REDES SEM FIO NO NETWORK SIMULATOR.....	39
4.1	Roteamento em Redes Sem Fio.....	39
4.2	<i>Table-Driven Routing Protocol</i> (Orientado a tabela) - Pró-ativo.....	40
4.2.1	<i>Destination-Sequenced Distance-Vector (DSDV)</i> .....	41
4.3	<i>Source-Initiated On-Demand Routing</i> (Sob demanda) - (Reativo).....	42
4.3.1	<i>Ad-hoc On-Demand Distance Vector (AODV)</i> .....	43
4.3.2	<i>Dynamic Source Routing (DSR)</i> .....	44
4.3.3	<i>Temporally Ordered Routing Algorithm (TORA)</i> .....	45
4.4	Criação e Configuração de Nós <i>MobileNode</i> no NS.....	46
4.5	Geração Automática de Cenários e Padrões de Tráfego.....	55
4.5.1	<i>Geração de Cenários com “setdest”</i> .....	56
4.5.2	<i>Geração de Padrões de Tráfego com “cbrgen.tcl”</i> .....	57
4.6	Analisando o Arquivo <i>Trace</i> da Simulação.....	57

<b>5</b>	<b>RESULTADOS .....</b>	<b>60</b>
<b>5.1</b>	<b>Rede de Acesso Sem Fio e Sem Mobilidade.....</b>	<b>60</b>
<b>5.1.1</b>	<b><i>Topologia da Rede.....</i></b>	<b>61</b>
<b>5.1.2</b>	<b><i>Parâmetros da Rede - Primeiro Cenário Sem Mobilidade .....</i></b>	<b>62</b>
<b>5.1.2.1</b>	<b><i>Script de Simulação .....</i></b>	<b>63</b>
<b>5.1.2.2</b>	<b><i>Resultados .....</i></b>	<b>63</b>
<b>5.1.2.3</b>	<b><i>Conclusão .....</i></b>	<b>67</b>
<b>5.1.3</b>	<b><i>Parâmetros da Rede - Segundo Cenário Sem Mobilidade .....</i></b>	<b>67</b>
<b>5.1.3.1</b>	<b><i>Resultados.....</i></b>	<b>69</b>
<b>5.1.3.1.1</b>	<b><i>Primeira Análise.....</i></b>	<b>69</b>
<b>5.1.3.1.2</b>	<b><i>Segunda Análise.....</i></b>	<b>70</b>
<b>5.1.3.1.3</b>	<b><i>Terceira Análise .....</i></b>	<b>72</b>
<b>5.1.3.2</b>	<b><i>Conclusões .....</i></b>	<b>73</b>
<b>5.2</b>	<b>Rede de Acesso Sem Fio e com Mobilidade Restrita .....</b>	<b>74</b>
<b>5.2.1</b>	<b><i>Topologia da Rede .....</i></b>	<b>74</b>
<b>5.2.2</b>	<b><i>Script de Simulação .....</i></b>	<b>76</b>
<b>5.2.3</b>	<b><i>Resultados.....</i></b>	<b>76</b>
<b>5.2.3.1</b>	<b><i>Primeira Análise .....</i></b>	<b>76</b>
<b>5.2.3.2</b>	<b><i>Segunda Análise.....</i></b>	<b>77</b>
<b>5.2.4</b>	<b><i>Conclusão .....</i></b>	<b>79</b>
<b>5.3</b>	<b>Rede de Acesso Sem Fio e com Mobilidade Irrestrita .....</b>	<b>80</b>
<b>5.3.1</b>	<b><i>Topologia de Rede .....</i></b>	<b>80</b>
<b>5.3.2</b>	<b><i>Script de Simulação .....</i></b>	<b>81</b>
<b>5.3.3</b>	<b><i>Resultados.....</i></b>	<b>83</b>
<b>5.3.4</b>	<b><i>Conclusão .....</i></b>	<b>87</b>
<b>6</b>	<b>CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS .....</b>	<b>89</b>
	<b>REFERÊNCIAS .....</b>	<b>92</b>
	<b>APÊNDICES .....</b>	<b>95</b>

# CAPÍTULO 1

## INTRODUÇÃO

Os avanços nas comunicações possibilitaram o surgimento de várias tecnologias, que procuram atender à real necessidade de seus usuários, com a melhor qualidade possível. Nos últimos anos, a comunicação sem fio ganhou um espaço considerável nas tecnologias de transmissão de dados, deixando de existir apenas nas comunicações de longa distância (feitas através de satélite), para fazer parte de ambientes locais (SIQUEIRA, 1997). As redes locais sem fio ou WLAN (*Wireless Local Area Networks*) foram desenvolvidas para oferecer aos usuários uma experiência de uso similar às redes locais cabeadas. O alcance típico de uma WLAN varia de 100 a 500 metros, dependendo das condições de transmissão do ambiente. Essa tendência foi fortalecida pelo investimento de instituições e empresas no sentido de aplicar a transmissão sem fio em redes de computadores (TANENBAUM, 2003).

Apostando nessa nova tecnologia, o IEEE (*Institute of Electrical and Eletronics Engineers*) constituiu um grupo de pesquisa para criar padrões abertos que pudessem tornar a tecnologia sem fio cada vez mais próxima da realidade. Esse projeto, denominado de padrão IEEE 802.11, nasceu em 1990, mas ficou inerte por, aproximadamente, sete anos, devido a fatores que não permitiam que a tecnologia sem fio saísse do papel (KUROSE; ROSS, 2006). Um dos principais fatores era a baixa taxa de transferência de dados que, inicialmente, a tecnologia oferecia, na faixa de Kbps (Kilobits por segundo).

À medida que a taxa de transferência aumentava, e passou a atingir a faixa de Mbps (Megabits por segundo), a rede sem fio começou a ser vista como uma tecnologia promissora e a receber ainda mais investimentos para a construção de equipamentos que possibilitassem a comunicação sem fio entre computadores.

Há uma tendência moderna de se utilizar sempre mais as redes sem fio, em vez de redes com fio. Essa propensão é motivada tanto por aspectos de inviabilidade da instalação de redes com fio, em certos lugares, como pela redução de custo dos equipamentos sem fio e da interoperabilidade oferecida por esta tecnologia. Outros fatores relacionam-se com as facilidades de mobilidade e flexibilidade que as comunicações sem fio oferecem.

A tecnologia WLAN vem sendo muito usada na medicina móvel para atendimento a pacientes, às transações comerciais e bancárias, a ambientes onde não possam transpassar fios (como construções antigas ou tombadas pelo patrimônio histórico), lugares onde há a

necessidade de mobilidade e flexibilidade nas estações de trabalho, entre outros (BERNAL, 2002).

Este estudo apresenta informações sobre as redes locais sem fio que atendem ao padrão IEEE 802.11, comentado nesta introdução. Sendo assim, o trabalho foi organizado em capítulos, incluindo esta introdução.

No capítulo 2, serão apresentadas as redes locais sem fio, destacando sua arquitetura, o meio físico e de enlace do modelo OSI (*Open System Interconnection*). No capítulo 3, será apresentado o simulador de eventos discretos para redes, intitulado NS (*Network Simulator*). No capítulo 4, será mostrada a estrutura básica para simulação em uma rede sem fio, usando o NS. No capítulo 5, será abordado um estudo com três tipos de redes de acesso sem fio, para avaliação de desempenho, empregando o NS nas simulações e apresentando os resultados obtidos. Finalmente, o capítulo 6 mostra as conclusões finais e sugestões para trabalhos futuros.

## CAPÍTULO 2

### REDES LOCAIS SEM FIO PADRÃO IEEE 802.11

Neste capítulo, serão apresentadas as redes locais sem fio padrão IEEE 802.11, sua arquitetura, camada física e de enlace. Na camada de enlace, serão detalhadas as subcamadas LLC (*Logic Link Control*) e MAC (*Media Access Control*).

O padrão 802.11 define basicamente uma arquitetura para as WLANs, que abrange a camada física e de enlace (especificamente na subcamada MAC), no modelo de referência OSI (CALI; CONTI; GREGORI, 2000). A Figura 2.1 mostra a relação das camadas do modelo de referência OSI com o modelo de referência IEEE 802.11. As camadas superiores controlam aspectos como endereçamento, roteamento, integridade de dados e formato dos dados contidos dentro de cada pacote, não fazendo diferença se elas estão transportando pacotes por meio de fios, fibra óptica ou sinais de rádio. A Figura 2.2 apresenta detalhes da camada física (PHY) e de enlace (MAC) de uma rede local sem fio com o padrão 802.11. A descrição de cada camada do padrão 802.11 será comentada nos subcapítulos 2.2 e 2.3.

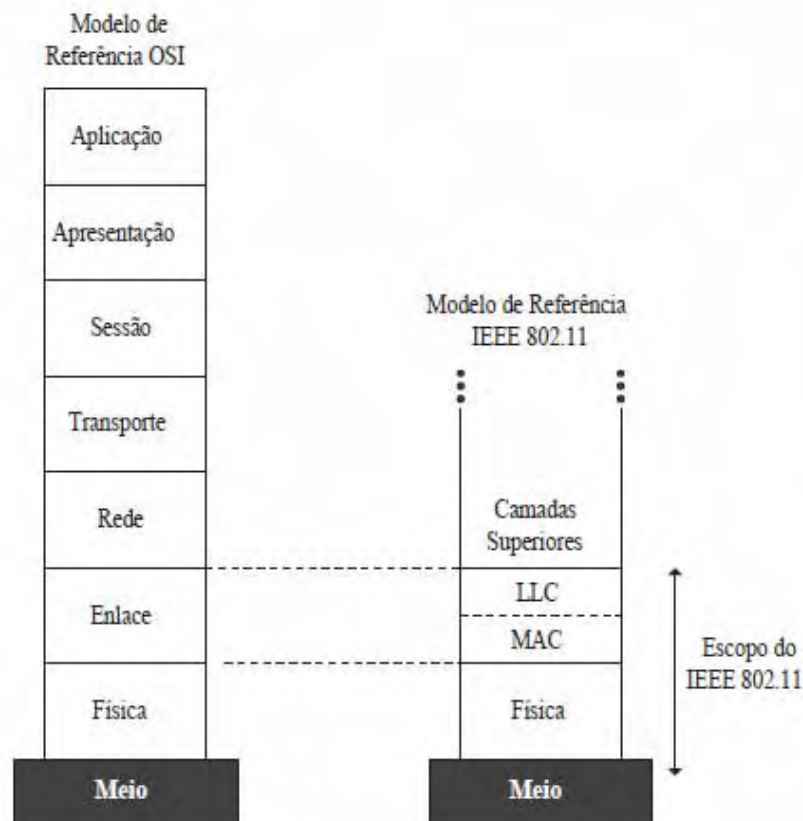


Figura 2.1 - Posição do IEEE 802.11 no modelo de referência OSI

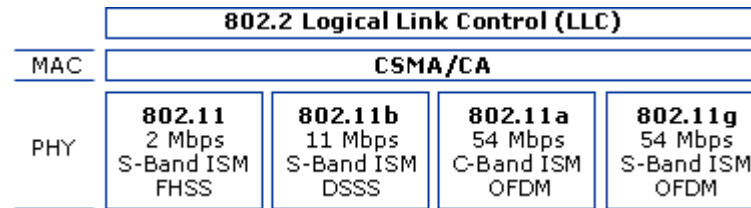


Figura 2.2 - Camada física (PHY) e de enlace (MAC) em uma rede 802.11

## 2.1 Arquitetura

Segundo o grupo de pesquisa IEEE, uma rede local sem fio, padrão 802.11, pode ser classificada como independente, no caso das redes *ad hoc* ou infraestruturadas (IEEE 802.11, 2010).

Redes infraestruturadas são aquelas em que o nó sem fio está em contato direto com uma Estação de Suporte à Mobilidade (ESM) ou Estação Base (EB), na rede fixa. O funcionamento deste tipo de rede é semelhante ao da telefonia celular, em que toda a comunicação deve necessariamente passar por um ponto central, mesmo que os equipamentos sem fio estejam a uma distância em que poderiam eventualmente comunicar-se diretamente. Toda a comunicação entre os nós sem fio é feita por meio de Estações de Suporte à Mobilidade. Neste caso, os nós sem fio, mesmo próximos uns dos outros, estão impossibilitados de realizar qualquer tipo de comunicação direta. A Figura 2.3 ilustra um modelo de comunicação em redes infraestruturadas onde a ESM está conectada a *gateways* que permitem a comunicação entre os nós sem fio ou host móvel (HM) e a parte fixa da rede.

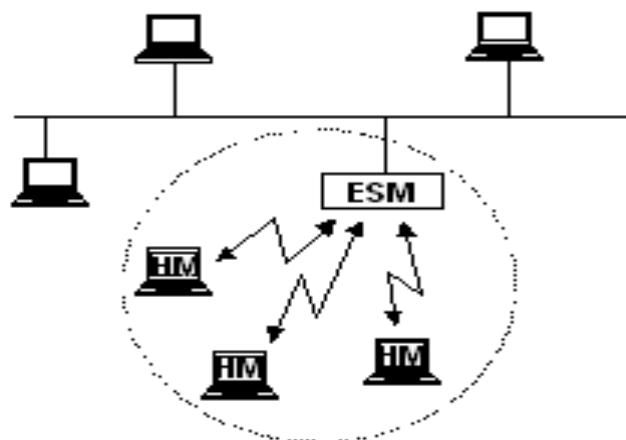


Figura 2.3 - Rede infraestruturada

Na rede *ad hoc*, também conhecida como MANET (*Mobile Ad hoc Network*), os dispositivos são capazes de trocar informações diretamente entre si. Ao contrário do que ocorre em redes convencionais, não há pontos de acesso, ou seja, não existem Estações de Suporte à Mobilidade, e os nós dependem uns dos outros para manter a rede conectada. Por esse motivo, redes *ad hoc* são indicadas, principalmente, em situações em que não se pode, ou não faz sentido, instalar uma rede fixa.

Os nós de uma rede *ad hoc* podem se mover arbitrariamente. Deste modo, a topologia da rede muda frequentemente e de forma imprevisível. Assim, a conectividade entre os nós sem fio modifica-se constantemente, requerendo uma permanente adaptação e reconfiguração de rotas. A Figura 2.4 apresenta um modelo de comunicação em redes *ad hoc*.

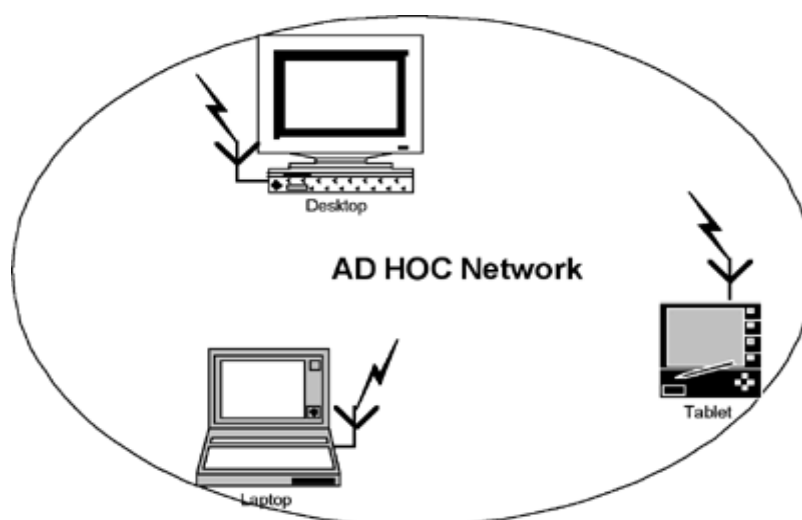


Figura 2.4 - Rede *ad hoc*

## 2.2 Camada Física

A camada física no padrão 802.11 é responsável pela transmissão dos quadros por um canal de comunicação. O padrão 802.11, de 1997, definiu três técnicas de transmissão para as redes sem fio (TANENBAUM, 2003). São essas o infravermelho e outras duas utilizando métodos de radiofrequência: o FHSS (*Frequency Hopping Spread Spectrum*) e o DSSS (*Direct Sequence Spread Spectrum*). Em 1999, foram apresentadas duas novas técnicas de radiofrequência, para alcançar maior largura de banda: o OFDM (*Orthogonal Frequency Division Multiplexing*) e o HR-DSSS (*High Rate Direct Sequence Spread Spectrum*).

A escolha da técnica dependerá de vários fatores relacionados com a aplicação dos usuários e o ambiente onde a rede operará e, assim, os dados poderão ser codificados e

modulados para equilibrar velocidade, distância e capacidade de transmissão. Para a transmissão em radiofrequência, são usadas as técnicas que transmitem os quadros de dados, enviando-os por vários canais disponíveis dentro de uma frequência, ao invés de usar um único canal, possibilitando, dessa forma, a transmissão simultânea de vários quadros.

Para atualizar a frequência e a taxa nominal de transmissão, outros padrões foram criados para redes sem fio 802.11 (IEEE 802.11, 2010). Os precursores foram os padrões: 802.11a, 802.11b e 802.11g, conforme apresentado na Figura 2.2 da camada física (PHY). A Tabela 2.1 reúne as principais características destes padrões.

Tabela 2.1 - Características das variantes do padrão 802.11

	<b>Frequência</b>	<b>Técnica de Modulação</b>	<b>Taxa de Transmissão de Dados</b>
802.11b	2,4 GHz	DSSS	até 11 Mbps
802.11g		DSSS, OFDM	até 54 Mbps
802.11a	5 GHz	OFDM	até 54 Mbps

Novos padrões de rede sem fio 802.11 foram criados com alterações na frequência, modulação, taxa de transmissão de dados e outras especificações. Como, exemplos, citam-se os padrões 802.11i, 802.11e e 802.11n. O padrão 802.11i aumentou consideravelmente a segurança, definindo melhores procedimentos para autenticação, autorização e criptografia. O padrão 802.11e agregou qualidade de serviço (QoS - *Quality Of Service*) às redes 802.11. O padrão 802.11n usa múltiplas antenas para transmissão e recepção, atingindo taxa nominal de transmissão de até 600 Mbps.

### 2.3 Camada de Enlace

A camada de enlace detecta e corrige erros que possam acontecer na camada física. Ela é responsável pela transmissão e recepção de quadros e pelo controle de fluxo. Esta camada, em uma rede local sem fio, pode ser dividida em duas subcamadas: LLC e MAC. A subcamada LLC é idêntica a qualquer outro padrão 802.x, mas a camada MAC é exclusiva da WLAN (CALI; CONTI; GREGORI, 2000). Na Figura 2.1, foi apresentada a divisão da camada de enlace.



### 2.3.1 Subcamada LLC

A subcamada LLC provê um enlace de comunicação entre entidades da camada de rede, para que elas possam trocar dados de maneira controlada e ordenada sobre uma conexão física cujo acesso é controlado por um protocolo do nível MAC (TANENBAUM, 2003). A camada LLC não faz parte do padrão 802.11, é a mesma usada na subcamada LLC da *Ethernet*.

### 2.3.2 Subcamada MAC

Além de definir um mecanismo para transmissão física, via radiofrequência ou infravermelho, o IEEE definiu um mecanismo de acesso ao meio denominado DFWMAC (*Distributed Foundation Wireless Medium Access Control*), prevendo dois modos de acesso denominados funções de coordenação. Ambas as funções de coordenação destinam-se a definir quando uma estação pode ou não transmitir. Estas funções são (SOARES, 1995):

- **DCF (*Distributed Coordination Function*):** é o modo de acesso distribuído básico, cuja decisão de que uma estação pode ou não transmitir é realizada individualmente pelas próprias estações da rede, existindo a possibilidade de ocorrer colisões.
- **PCF (*Point Coordination Function*):** é o modo de acesso com controle centralizado, em que a decisão de transmitir é centralizada em uma estação, que determina qual estação pode transmitir e em que momento, minimizando, assim, a possibilidade de colisões.

Em ambos os modos de acesso, existem parâmetros para regular o tempo necessário de espera, antes de liberar o acesso ao meio para uma estação, uma vez que o meio pode estar ocupado com a transmissão de quadros de dados, quadros de controle ou, ainda, estar disponível, para que alguma estação possa tomar o meio de transmissão.

Para uma estação ter acesso ao meio e assim poder transmitir, ela deve detectar se o meio está livre por um período de silêncio mínimo (IFS - *Inter Frame Space*), antes de utilizá-lo. O DFWMAC define três prioridades de acesso ao meio, através de diferentes períodos de tempo (SOARES, 1995):

- **DIFS (*Distributed Inter Frame Space*):** parâmetro que indica o maior tempo de espera aguardando um intervalo de silêncio mínimo, para ter o acesso ao meio, para transmitir dados (menor prioridade) e é definido pelo espaço distribuído entre quadros da função de coordenação distribuída (DCF);
- **PIFS (*Priority Inter Frame Space*):** é um intervalo de tempo intermediário do DIFS e o SIFS (prioridade média), já que é usado por uma estação que controla outras estações e, por isto, tem maior prioridade que as estações comuns e é definido pelo espaço entre os quadros da função de coordenação pontual (PCF);
- **SIFS (*Short Inter Frame Space*):** é o intervalo de tempo de espera para funções de maior prioridade. Em geral, são transmissões de quadros que contêm respostas curtas como, por exemplo, os pacotes ACK (*Acknowledgment*).

Todas as estruturas *ad hoc* e infraestruturada devem operar com a DCF. No entanto, a operação com a PCF é opcional, no caso de redes infraestruturadas.

### 2.3.2.1 Função de Coordenação Distribuída (DCF)

A função DCF realiza o acesso ao meio através do método CSMA/CA (*Carrier Sense Multiple Access/Collision Avoidance*), cuja colisão é evitada, solicitando a uma estação esperar certo período de tempo aleatório, antes da transmissão (SOARES, 1995).

Neste esquema de acesso ao meio, a estação que deseja transmitir um quadro verifica o canal por um período chamado DIFS e, ao final desse período, se o canal estiver livre, realiza a transmissão. Ao receber o quadro de dados, o receptor aguarda por um tempo chamado SIFS e, então, envia um reconhecimento positivo (ACK).

Se o meio estiver ocupado após um tempo DIFS, as outras estações entram em uma fase de contenção. Nesta fase, cada estação escolhe um tempo aleatório, dentro de uma janela de contenção. Após este intervalo de tempo aleatório, as estações tentam acessar o meio. Permanecendo ocupado após este intervalo, determina-se que essa estação perdeu este ciclo, iniciando pela espera da duração de um tempo DIFS. Para proporcionar que as estações que já estão no ciclo tenham vantagem em relação às novas que tentarem o acesso, é acrescentado um contador (*backoff*). Se uma determinada estação não consegue acessar no primeiro ciclo, espera o canal estar inativo novamente por um tempo DIFS e o seu contador começa a decair. Provavelmente, a estação antiga terá o seu contador com um tempo menor, assim, quando o

contador expirar, essa estação acessa o meio (SOARES, 1995). A Figura 2.5 apresenta o funcionamento da função DCF.

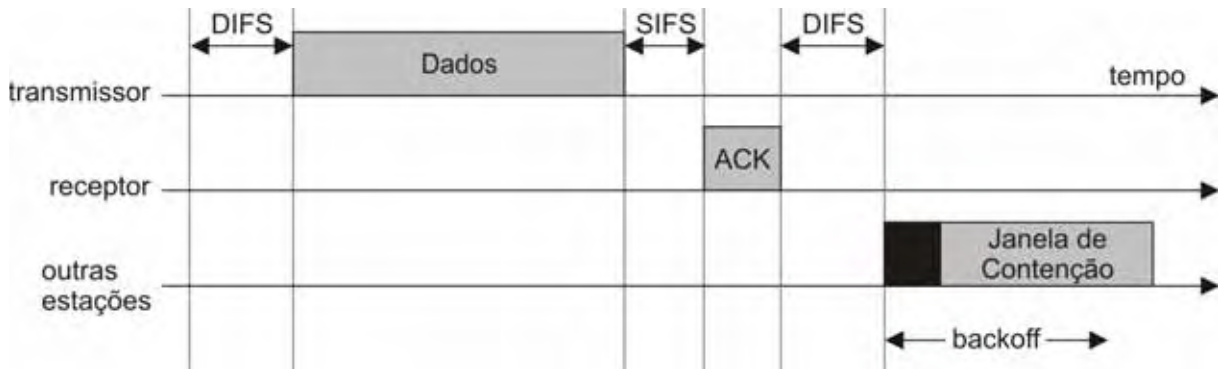


Figura 2.5 - Funcionamento da função DCF

Além do esquema básico de acesso ao meio, a função de coordenação distribuída disponibiliza um esquema opcional baseado em pacotes de solicitação (RTS - *Request to Send*) e permissão de transmissão (CTS - *Clear to Send*), para evitar problemas causados por terminais escondidos. O problema do terminal escondido acontece, por exemplo, como apresentado na Figura 2.6, quando uma estação B é capaz de receber quadros das estações A e C, mas as mesmas estão fora do alcance uma da outra. Desta forma, A e C podem transmitir ao mesmo tempo, gerando colisão no receptor B (SOARES, 1995).

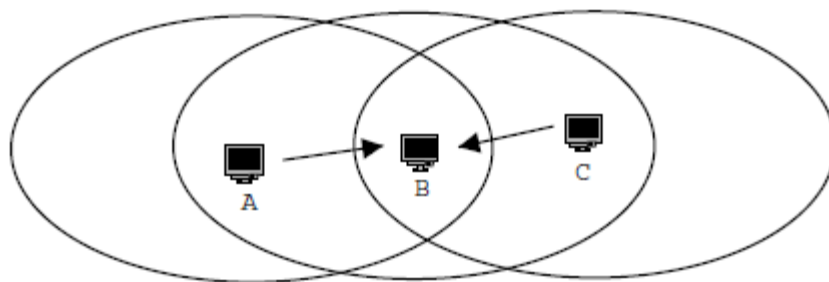


Figura 2.6 - Problema do terminal escondido

Assim, quando uma estação pretende realizar uma transmissão, ela emite um quadro chamado RTS, contendo a duração do quadro de dados endereçado à estação de destino e informando que irá começar a transmissão. Se a estação de destino receber este RTS corretamente, ela espera um tempo igual ao SIFS para verificar se nenhuma outra transmissão está ocorrendo. Caso o canal esteja ocioso, a estação de destino transmite o quadro CTS, informando que o canal está livre para transmissão e reservando o meio de acesso para esta

transmissão. Após um tempo igual ao SIFS, a estação que recebeu o CTS inicia a sua transmissão. Ao receber o quadro de dados, o receptor aguarda por um tempo chamado SIFS e, então, envia um reconhecimento positivo (ACK). Nos quadros utilizados pelo mecanismo RTS/CTS, são especificados os limites utilizados para envio dos dados que a estação deseja transmitir. Como o canal é de difusão, as estações que receberem o RTS/CTS podem usar esta informação para atualizar seu vetor de alocação da rede NAV (*Network Allocation Vector*). Desta forma, as estações que não participam da transmissão, não precisam “escutar” o meio durante todo o tempo, ativando o modo estado de espera. Assim, apenas quando o contador de tamanho igual ao NAV estiver zerado, a estação volta a monitorar o meio físico (SOARES, 1995). A Figura 2.7 mostra o esquema de funcionamento da função DCF (RTS/CTS).

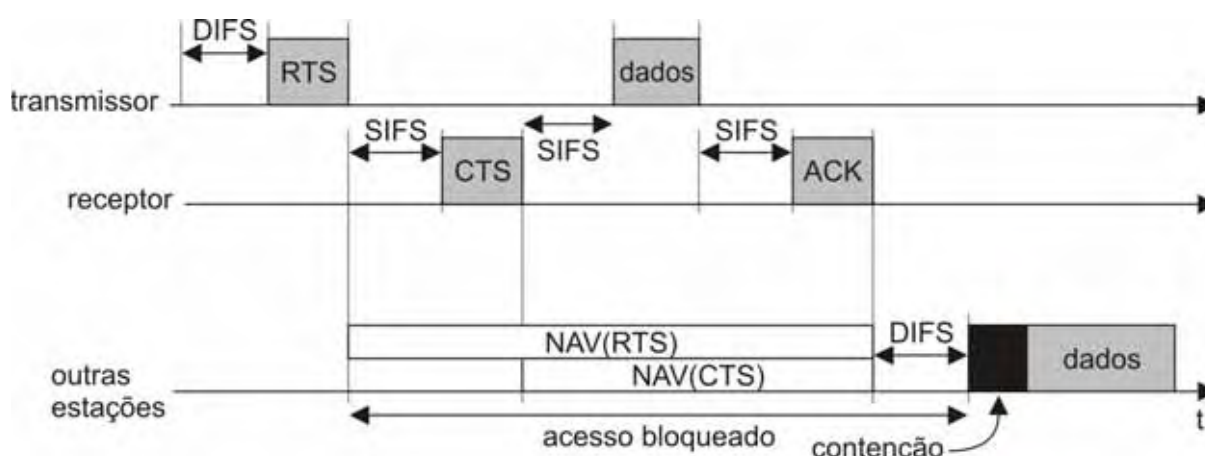


Figura 2.7 - Funcionamento da função DCF (RTS/CTS)

### 2.3.2.2 Função de Coordenação Pontual (PCF)

A função PCF baseia-se em consultas periódicas às estações usando um AP (*Access Point*), para controlar o acesso ao meio, determinando, em cada momento, qual estação deve transmitir (SOARES, 1995). Redes *ad hoc* não podem usar essa função, uma vez que não possuem nenhum nó central controlador.

Para promover a integração das duas funções de coordenação (pontual e distribuída), o método de acesso PCF utiliza o conceito de superquadro, fazendo com que a função de coordenação pontual assuma o controle da transmissão, evitando a ocorrência de colisões. Um superquadro consiste em dois intervalos de tempo consecutivos: no primeiro, controlado pela PCF, o acesso é ordenado (não ocorrem colisões). Após esperar um intervalo de tempo (especificado pelo PIFS), o ponto de coordenação dá acesso a primeira estação, que pode

responder após SIFS. Depois de aguardar mais SIFS, o coordenador dá a vez para a segunda estação e, assim por diante. Quando uma estação não responde após SIFS, o coordenador aguarda PIFS e passa a vez para a próxima. No segundo tempo, controlado pela DCF, o acesso baseia-se na disputa pela posse do meio, podendo ocorrer colisões.

Na Figura 2.8, é mostrado o esquema de integração das duas funções de coordenação (pontual e distribuída) em que, no tempo  $t_0$ , o período livre de contenção do superquadro deveria ter iniciado. Entretanto, outra estação está transmitindo (o meio está ocupado). Por causa disso, o PCF espera o DCF, ou seja, o início do superquadro é adiado (SOARES, 1995).

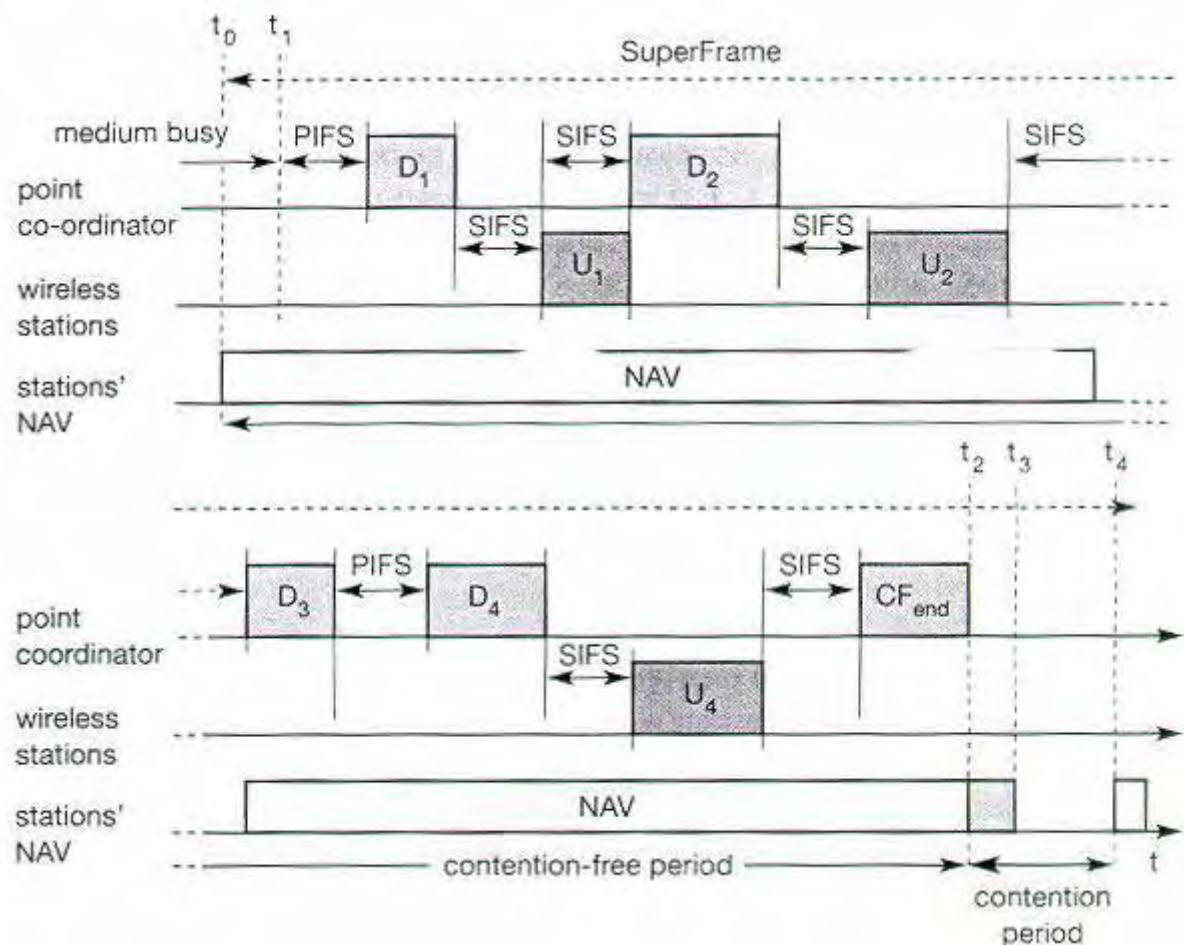


Figura 2.8 - Funcionamento da função PCF (ERGEN, 2002)

Depois que o meio se tornar inativo (no tempo  $t_1$ ), o coordenador pontual (AP) tem que esperar por PIFS antes de acessar o meio. O AP, agora, requer a transmissão, enviando o dado  $D_1$  para a primeira estação que responde com o dado  $U_1$  depois de SIFS. Após esperar SIFS novamente, o AP requer a transmissão, enviando o dado  $D_2$  para a segunda estação (a Figura 2.8 ilustra o exemplo de como seria o procedimento incorreto, em que o AP não espera

o SIFS novamente). A estação responde, enviando o dado  $U_2$ . Novamente, o AP envia um requerimento para uma terceira estação, mas, desta vez, a estação não tem nada para enviar. Então, o AP não vai receber nada depois de SIFS. Após aguardar por PIFS, o AP pode requerer a transmissão da quarta estação através de  $D_4$ . A estação responde com  $U_4$  e, depois de SIFS, o AP envia um sinal de finalização ( $CF_{end}$ ), ou seja, o período de contenção pode ser iniciado. Na utilização do PCF, são configurados todos os NAVs, evitando a transmissão de outras estações. Neste exemplo, o período livre de contenção esperado era de  $t_0$  até  $t_3$ . Entretanto, como a terceira estação não enviou dados, o período terminou em  $t_2$ . Em  $t_4$ , o ciclo se reinicia com outro superquadro.

### 2.3.3 Estrutura de Quadros

O padrão 802.11 define o quadro MAC, um cabeçalho com uma variedade de campos usados na subcamada MAC (MATTHEW, 2002).

O quadro MAC do padrão 802.11, como mostrado na Figura 2.9, consiste nos campos *MAC Header* (cabeçalho), *Frame Body* (corpo do quadro) e FCS (*Frame Check Sequence*). Os números, na Figura 2.9, representam o número de *bytes* de cada campo.

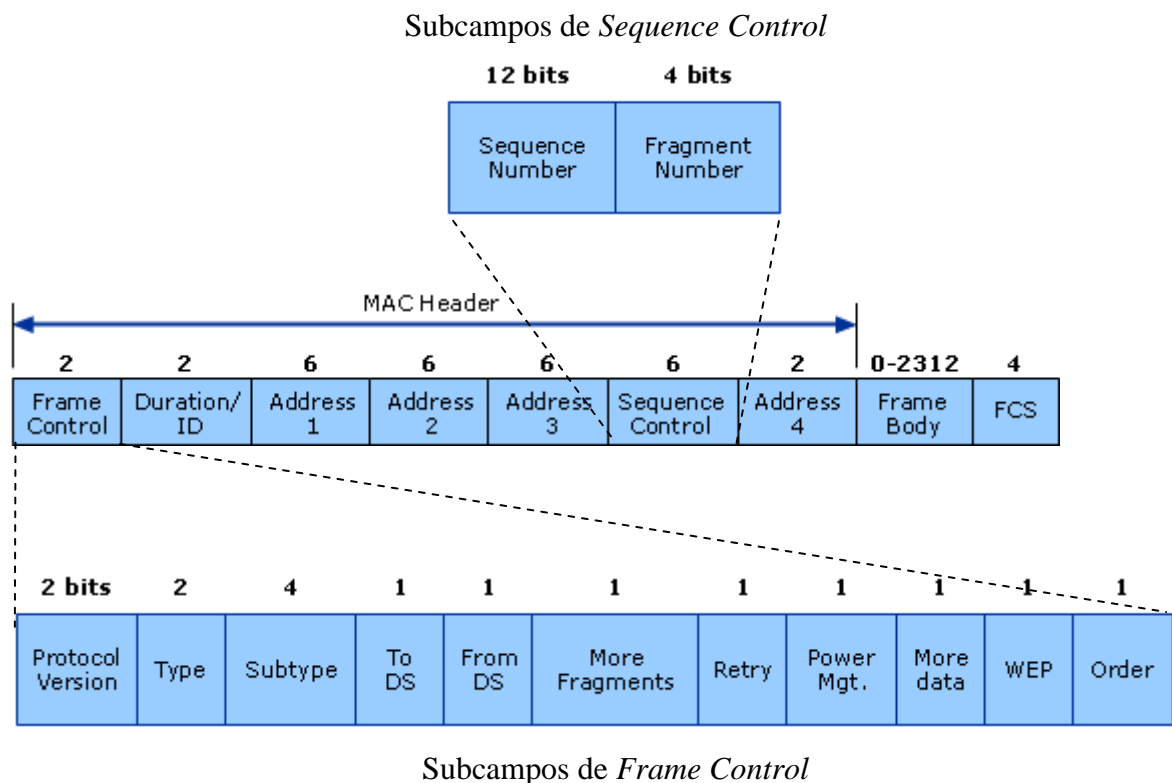


Figura 2.9 - Quadro MAC do padrão 802.11

A definição dos campos é apresentada a seguir (MATTHEW, 2002):

- **Frame Control (Controle de Quadro):** esse quadro contém informações de controle usado para definir o tipo de quadro MAC 802.11. A estrutura desse campo é composta por onze subcampos:
  - **Protocol Version (Versão do Protocolo):** indica a versão corrente do protocolo 802.11 utilizado. As estações receptoras usam esse valor para determinar se a versão do protocolo do quadro recebido é suportada.
  - **Type e Subtype (Tipo e Subtipo):** determina a função do quadro. Há três diferentes tipos de quadro: controle, dados e gerenciamento. Há múltiplos subtipos para cada tipo de quadro. Cada subtipo determina uma função específica desempenhada com o seu tipo de quadro associado, por exemplo, RTS ou CTS.
  - **To DS (Para Sistema de Distribuição) e From DS (Do Sistema de Distribuição):** indica se a informação será enviada para um sistema de distribuição ou se foi originada em um sistema de distribuição. Esses campos somente são utilizados em um quadro de dados transmitidos por estações conectadas a um ponto de acesso.
  - **More Fragments (Mais Fragmentos):** campo utilizado para informar se existem mais fragmentos do quadro, seja de dados ou de gerenciamento, que devem ser encaminhados.
  - **Retry (Repetir):** indica se a informação (dado ou gerenciamento) está ou não sendo retransmitida.
  - **Power Management (Gerenciamento de Potência):** é usado pelo ponto de acesso para deixar ou retirar o receptor do estado de espera.
  - **More Data (Mais Dados):** indica que o transmissor tem quadros adicionais para o receptor.
  - **WEP (Wired Equivalent Privacy):** indica ou não se está sendo usado no quadro o processo de criptografia e autenticação. Isso pode ser configurado para todos os quadros de dados e gerenciamento que têm o *subtype* configurado para autenticação.
  - **Order (Ordem):** indica se todos os quadros de dados recebidos devem ser processados em ordem.

- **Duration ID (Duração/ID):** este campo é utilizado para indicar o tempo necessário para aguardar a próxima transmissão a ser recebida. Para a reserva do canal, realizada através do mecanismo RTS/CTS, esse campo contém o período de tempo que o meio ficará ocupado.
  
- **Address Field (Campo de Endereço):** contém os endereços de origem/destino do quadro e de origem/destino do ponto de acesso:
  - **BSS Identifier (Identificador de BSS):** o SSID (*Service Set Identifier*) identifica cada BSS quando o quadro é vindo de uma estação que opera em modo infraestruturado. Quando o quadro é proveniente de uma estação que opera em modo *ad hoc*, o SSID é um valor aleatório gerado e gerenciado localmente pela própria estação que iniciou a transmissão.
  - **Destination Address (Endereço Destino):** indica o endereço MAC do destino final para a recepção do quadro.
  - **Source Address (Endereço Fonte):** indica o endereço MAC da fonte que originou e transmitiu inicialmente o quadro.
  - **Receiver Address (Endereço do Receptor):** indica o endereço MAC da próxima estação que irá receber o quadro.
  - **Transmitter Address (Endereço do Transmissor):** indica o endereço MAC da estação que transmitiu o quadro na rede sem fio.
  
- **Sequence Control (Controle de Sequência):** formado por dois subcampos:
  - **Fragment Number (Número de Fragmento):** considerado apenas quando existe um fragmento relacionado ao quadro enviado.
  - **Sequence Number (Número de Sequência):** registra o número de sequência para cada quadro. Esse número é sempre igual, caso o quadro enviado seja um quadro fragmentado.
  
- **Frame Body (Corpo do Quadro):** contém a informação específica sobre o quadro de dados ou quadro de gerenciamento.
  
- **Frame Check Sequence (Sequência de Verificação do Quadro):** o transmissor do quadro aplica um CRC-32 (*Cyclic Redundancy Check*) sobre todos os campos do cabeçalho MAC e sobre o corpo do quadro para gerar o FCS. Assim, é possível que a



estação receptora do quadro, através do CRC-32, analise se o valor recebido no campo FCS é o mesmo recebido, possibilitando a verificação de possíveis erros durante a transmissão.

No próximo capítulo, será apresentado o simulador *Network Simulator* usado nas simulações dos cenários de rede nesta dissertação.

## CAPÍTULO 3

### O SIMULADOR DE REDES NETWORK SIMULATOR

Neste capítulo, serão apresentados um breve histórico do simulador de redes NS (*Network Simulator*), as linguagens usadas e a sua arquitetura.

Uma das dificuldades encontradas para a criação e a otimização de protocolos de redes é a análise de tais protocolos em diferentes ambientes. A implantação da infraestrutura para testes destas redes se mostraria custosa com o aumento do número de estações, no ponto de vista financeiro e no tempo de desenvolvimento.

Com base nesses fatos, uma solução com custos reduzidos e que atenda à flexibilidade necessária para os testes é a simulação (JAIN, 1991). Com a simulação, pode-se obter resultados a partir de diferentes topologias e número de estações, com o gasto de tempo relativo apenas na configuração dos parâmetros de simulação.

O NS é um simulador de rede de eventos discretos. Trata-se de um software de código aberto que possui várias extensões desenvolvidas no meio acadêmico e é amplamente usado em pesquisas de redes de computadores com fio, sem fio e satélite (ISI, 2009). Ele simula o funcionamento de vários protocolos de aplicação, transporte, rede e enlace e implementa fontes de tráfego simulando aplicações FTP (*File Transfer Protocol*), *Telnet*, *Web*, CBR (*Constant Bit Rate*) e VBR (*Variable Bit Rate*).

O NS emprega protocolos de transporte UDP (*User Datagram Protocol*) e TCP (*Transmission Control Protocol*) com suas variantes *Tahoe*, *Reno*, *Newreno* e *Vegas*. Possui mecanismos de controle de filas de roteadores como *DropTail*, FQ (*Fair Queueing*), DRR (*Deficit Round Robin*), SFQ (*Stochastic Fair Queueing*), RED (*Random Early Discard*) e CBQ (*Class-Based Queueing*). O NS usa protocolos de roteamento *Distance Vector*, *Link State*, entre outros. Por fim, ele implementa *multicasting* e alguns dos protocolos da camada MAC para simulação de LANs (*Local Area Networks*) (CHUNG; CLAYPOOL, 1995).

#### 3.1 Breve Histórico

A primeira versão do NS surgiu em 1989, proveniente de uma variação do software *REAL Network Simulator*. Em 1995, seu desenvolvimento passou a ter o apoio do DARPA (*Defense Advanced Research Projects Agency*), por meio do Projeto VINT (*Virtual InterNetwork Testbed*), envolvendo os colaboradores USC/ISI (*University of Southern*

*California/ Information Sciences Institute*), Xerox PARC (*Palo Alto Research Center*), LBNL (*Lawrence Berkeley National Laboratory*) e UCB (*University of California Berkeley*), culminando no lançamento da versão 2. Atualmente, seu desenvolvimento é feito pelo DARPA, com o projeto SAMAN (*Simulation Augmented by Measurement and Analysis for Networks*) e pela NSF (*National Science Foundation*), com o projeto CONSER (*Collaborative Simulation for Education and Research*), ambos em colaboração com outros pesquisadores, inserindo o ICSI (*International Computer Science Institute*). O NS ainda inclui contribuições substanciais de outros pesquisadores, incluindo códigos de redes sem fio do UCB *Daedalus*, dos projetos do CMU (*Carnegie Mellon University*) e da *Sun Microsystems* (ISI, 2009).

O NS encontra-se atualmente na versão 2.34, atualizada em junho de 2008. Ele foi desenvolvido para operar preferencialmente em plataformas *Unix* (*FreeBSD*, *SunOS*, *Solaris*, *Linux*, entre outras). Pode funcionar ainda em plataforma *Windows*, apesar da sua instalação não ser tão automatizada e não seguir os padrões de instalação de software feitos para *Windows* (ISI, 2009). A distribuição do NS, além de ser gratuita, proporciona a flexibilidade de que o seu código fonte pode ser alterado para refletir a pesquisa que está sendo desenvolvida. Um grupo de estudo é mantido pelos usuários, cujos pesquisadores de diversas partes do mundo podem trocar ideias e experiências e, do mesmo modo, propor correções para o código do simulador, que, depois de avaliadas, podem ser incorporadas. Estes pesquisadores, oriundos de países como Estados Unidos, Índia, Inglaterra, Itália, Taiwan e Brasil, contribuem para o valor desta ferramenta.

### **3.2 A Linguagem do NS**

O NS é um simulador que foi desenvolvido utilizando-se duas linguagens de programação. O núcleo do NS é escrito em C++, o que lhe confere uma especificação de protocolos no nível de bits/bytes, garantindo rápida velocidade de execução, uma característica da linguagem. Como um jeito de tornar a descrição dos modelos mais prática e rápida para o usuário, o NS usa a linguagem OTcl (*Object-oriented Tool Command Language*) como interpretador de comandos e de configuração de modelos para simulações. Com a escolha das linguagens citadas, os desenvolvedores do NS souberam adequar, de uma forma prática para os usuários e desenvolvedores da ferramenta, o poder de representação e eficiência da linguagem C++, quanto à praticidade da linguagem OTcl conveniente à

execução de várias simulações que necessitam de grandes ou pequenas alterações rapidamente (CHUNG; CLAYPOOL, 1995).

De fato, a interface de comando e configuração do NS interpreta comandos OTcl, que podem ser fornecidos interativamente pelo usuário, de forma semelhante à linha de comando de um sistema *Unix* (bash, sh, csh, entre outros), ou na forma de *scripts* de simulação em OTcl, que são constituídos por linhas de código em sequência, de tal modo a descrever completamente um modelo a ser simulado. Assim, o OTcl funciona como um *frontend* mais amigável e prático para o usuário interagir com o simulador.

Para que seja possível a utilização dessas duas linguagens, OTcl e C++, uma como *frontend* de usuário e outra como implementação dos componentes de rede, existe no NS uma correspondência entre objetos na interface OTcl e objetos C++, conforme mostrado na Figura 3.1 (CHUNG; CLAYPOOL, 1995).

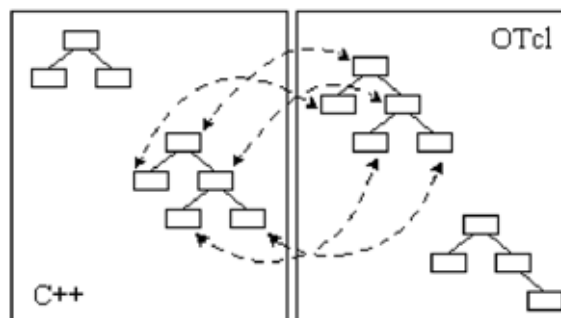


Figura 3.1 - Correspondência entre objetos C++ e Otcl

Com o intuito de reduzir o tempo de processamento de pacotes e eventos, o agendador de eventos e os componentes básicos de rede são escritos e compilados em C++. Esses objetos compilados são disponibilizados ao interpretador OTcl através de um “acoplamento” entre a biblioteca OTcl e os componentes compilados C++, criando um objeto correspondente no interpretador OTcl para cada objeto construído em C++. Da mesma maneira isso é feito para métodos e variáveis desses objetos, podendo os mesmos serem referenciados via OTcl, ainda que sua especificação real esteja feita em C++.

Uma visão detalhada, a partir da codificação de um *script* de simulação em OTcl, seu processamento pelo simulador e, conseqüentemente, os resultados que podem ser analisados ou visualizados, é exibida na Figura 3.2 (CHUNG; CLAYPOOL, 1995).

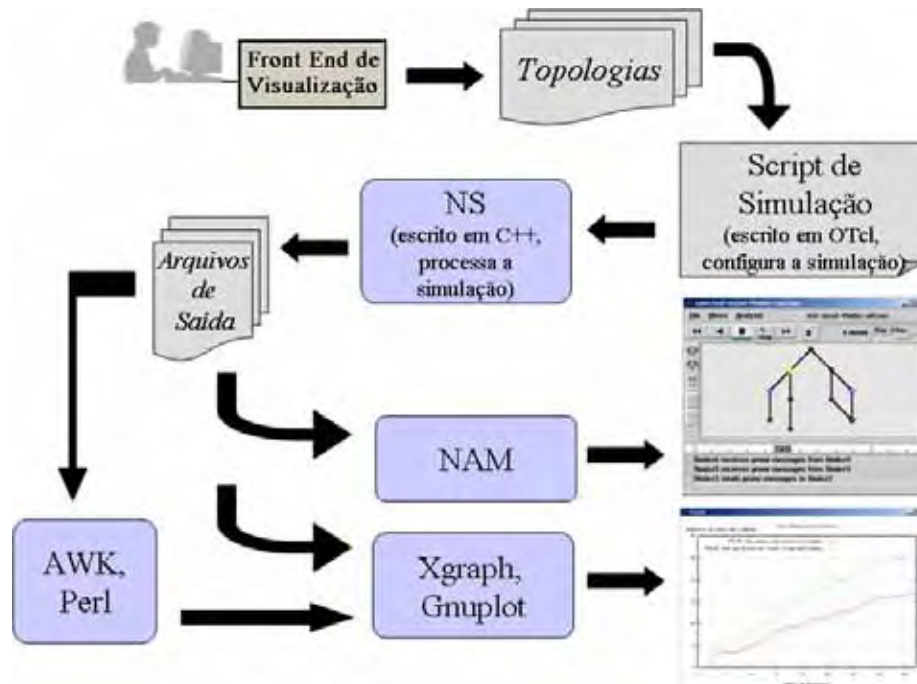


Figura 3.2 - Estrutura do NS

A Figura 3.2 sintetiza a estrutura do NS onde, inicialmente, o usuário define a topologia, escrevendo um *script* de simulação em OTcl, contendo os objetos que representam os componentes da rede (nós, ligação entre os nós, agentes, tráfego da rede, entre outros) e configurando o seu comportamento por meio do agendador de eventos. Em outras palavras, definir quando o agendador deve disparar uma determinada ação. Na rede, uma ação pode ser, por exemplo, a transmissão de dados entre dois nós, ou mesmo, a movimentação de um nó na rede em função do tempo durante a simulação. Este *script* é submetido ao núcleo do NS para processar a simulação.

Concluída a simulação, inicia-se uma das fases mais importantes: a análise dos resultados. É fundamental que se busquem consistência e coerência nesses resultados antes de apresentá-los. O NS gera um log<sup>1</sup> com todos os eventos ocorridos durante o processo de simulação em um arquivo *trace* ou arquivo de saída. Por definição da língua inglesa, *trace* significa “encontrar ou descobrir por investigação; descrever a origem ou desenvolvimento de algo” (MCKEAN, 2005). É justamente esse o objetivo das informações geradas no arquivo *trace* da simulação: investigar o comportamento da rede modelada.

Com a finalidade de extrair os dados necessários para a análise, esse arquivo *trace* pode ser processado utilizando *scripts* escritos em Perl e AWK (Alfred V. **Aho**, Peter J. **Weinberger** e Brian W. **Kernighan**), duas linguagens de programação muito eficientes para

<sup>1</sup> É o termo utilizado para descrever o processo de registro de eventos relevantes numa simulação.

se manipular *strings* de texto. Com esses *scripts*, consegue-se extrair do arquivo *trace* as seguintes informações: vazão de pacotes na rede, atraso médio dos pacotes, variação do atraso de pacotes, taxa de perdas de pacotes, entre outros. A ilustração de um estudo comparativo, por meio de gráficos, constitui-se em uma das principais contribuições do NS. Isso é possível usando ferramentas de plotagem como o XGraph, componente opcional do NS, ou o Gnuplot do *Unix* (ALTMAN; JIMENEZ, 2003).

Além do arquivo *trace*, voltado para a análise do comportamento da rede, ainda é possível produzir outro tipo de arquivo, voltado à análise visual do comportamento da rede. Este arquivo é utilizado pelo NAM (*Network AniMator*), outro componente opcional do NS, que foi desenvolvido como parte do projeto VINT e possibilita uma visualização animada do comportamento da rede, assim como: movimentação de nós, transmissão de pacotes de várias fontes de tráfego, estouros de fila em roteadores, ocasionando perdas de pacotes e alcance de sinal em redes sem fio (VINT, 1996). Contudo, as informações da visualização animada não podem ser utilizadas para análise profunda de simulações.

### 3.3 Arquitetura do NS

Neste subcapítulo, é dada uma visão um pouco mais detalhada da arquitetura do NS, ou seja, como está estabelecida a hierarquia entre os componentes do sistema e sua interligação. A arquitetura do NS é formada pelos seguintes componentes (CHUNG; CLAYPOOL, 1995):

- Escalonador de eventos
- Componentes de rede
- TclCL (*Tool Command Language with Classes*)
- Biblioteca OTcl
- Linguagem de *script* Tcl

O NS inclui uma extensão simples de OTcl chamada TclCL, que fornece uma ponte entre C++ e OTcl. A Figura 3.3 mostra a arquitetura geral do NS.

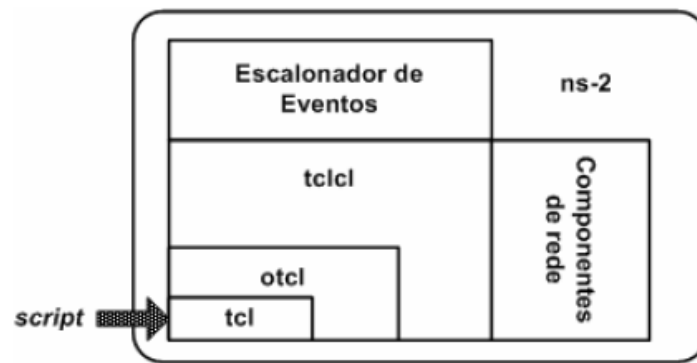


Figura 3.3 - Arquitetura geral do NS

Nesta figura, o usuário do NS está localizado no canto inferior esquerdo, onde projeta e executa simulações em *script* Tcl, usando os objetos do simulador em OTcl. O escalonador de eventos e os componentes de rede são implementados em C++. A interface de ambos, com a linguagem OTcl, é feita por meio de interligação usando o TclCL. A união desses cinco componentes forma o NS, localizado no canto superior direito da Figura 3.3.

No próximo capítulo, será mostrada a estrutura básica para simulação em uma rede sem fio, usando o Network Simulator, bem como os protocolos de roteamento orientado a tabela e, sob demanda, a criação e configuração dos nós *MobileNode*, a geração automática de cenários e padrões de tráfego e a análise do arquivo de saída da simulação.

## CAPÍTULO 4

### SIMULAÇÃO DE REDES SEM FIO NO NETWORK SIMULATOR

O objeto de estudo deste trabalho tem foco principal na simulação de redes sem fio padrão 802.11. Neste capítulo, será exibida a estrutura de configuração de uma simulação sem fio no NS, assim como trechos de exemplo de um *script* básico de configuração da rede sem fio escrito em OTcl. Além disso, será abordada a geração de modelos de cenário (disposição dos nós) e de tráfego de dados, assim como a análise do arquivo *trace* produzido pela simulação da rede.

O modelo de simulações de rede sem fio utilizado pelo NS foi desenvolvido por pesquisadores do Projeto Monarch (*Mobile Networking Architectures*), originalmente pertencente à CMU, e, agora, pertencente à Rice University (JOHNSON, 2004). Inicialmente, esse modelo foi adotado apenas como uma extensão do NS. Atualmente, o modelo é parte integrante do NS e tem sido continuamente aprimorado ao longo dos anos. Os componentes principais deste modelo são o *MobileNode* (nó sem fio), mecanismos de roteamento e componentes de rede que são empregados para configurar o *MobileNode*.

#### 4.1 Roteamento em Redes Sem Fio

O roteamento em redes sem fio tem sido tema frequente de pesquisas na área de redes. Dentre os diversos problemas, que envolvem o roteamento neste tipo de rede, está a possibilidade de mobilidade nos nós da rede, a não obrigatoriedade de que todos os nós estejam no raio de transmissão uns dos outros e a falta de informações prévias a respeito da localização de cada nó no espaço (HEIDEMANN et al., 2000).

Visando resolver todos estes problemas, vários tipos de protocolos de roteamento foram desenvolvidos. Entretanto, eles estão divididos em duas categorias de protocolos (ROYER; TOH, 1999):

- *Table-Driven Routing Protocol* (Orientado a tabela) - Pró-Ativo
- *Source-Initiated On-Demand Routing* (Sob demanda) - Reativo

Os tipos de transmissão utilizados nas redes sem fio pelos protocolos de roteamento podem ser classificados em (CHUNG; CLAYPOOL, 1995):



- **Transmissões em *unicast*:** transmissões somente entre dois nós (fonte e destino).
- **Transmissões em *broadcast*:** o nó fonte transmite um pacote através da rede para todos os seus vizinhos. As transmissões em *broadcast* podem ser chamadas de *flooding*, quando estas se espalham descontroladamente, por toda a rede, por meio de simultâneas retransmissões dos nós que a recebem. Para evitar que isto ocorra, os protocolos de roteamento podem implantar formas de contenção deste *flooding*, por meio da aplicação de identificadores únicos a estas mensagens ou de tempo de vida para elas, conhecido como TTL (*Time To Live*).

Apesar destes protocolos de roteamento terem sido projetados para o mesmo tipo de rede, as suas características são bem distintas.

Os próximos tópicos do capítulo descrevem de forma resumida as características e funcionalidades de alguns destes protocolos, organizados por suas categorias.

#### **4.2 *Table-Driven Routing Protocol (Orientado a tabela) - Pró-ativo***

Os protocolos pró-ativos englobam todos os protocolos cuja estratégia de roteamento requer que todas as informações sobre possíveis rotas sejam mantidas atualizadas durante todo o ciclo de vida da rede (PERKINS; BHAGWAT, 2000). Portanto, a cada mudança significativa na rede (movimentação, aparecimento ou desaparecimento de um ou mais nós), esta alteração é sinalizada a todos os nós, para que suas respectivas tabelas de roteamento possam ser atualizadas.

A grande vantagem dos protocolos pró-ativos está na maior agilidade no início de uma transmissão para um destino qualquer, uma vez que as informações sobre as rotas estão atualizadas para todos os possíveis destinos da rede. Quando uma requisição de transmissão é recebida esta é iniciada imediatamente.

Entretanto, a grande desvantagem destes protocolos está no excessivo tráfego de rede gerado pela enorme quantidade de mensagens de controle necessárias para manter as tabelas de roteamento de todos os nós da rede sempre atualizadas. O principal protocolo de roteamento pró-ativo em estudo para as redes sem fio é o *Destination-Sequenced Distance-Vector* (PERKINS; BHAGWAT, 1994).

### 4.2.1 Destination-Sequenced Distance-Vector (DSDV)

O DSDV toma como base o algoritmo clássico de roteamento *Bellman-Ford*. É um protocolo de roteamento pró-ativo, baseado em vetor de distâncias, que trabalha requisitando, periodicamente, de cada um dos nós vizinhos, suas tabelas de roteamento, com a finalidade de manter suas tabelas atualizadas. Cada nó da rede mantém uma tabela de roteamento que contém o próximo salto e o número de saltos para alcançar o destino. As tabelas mantêm rotas para todos os nós da rede, mesmo que nunca seja necessário enviar pacote para este nó. Cada nó mantém apenas uma rota para cada destino (PERKINS; BHAGWAT, 1994).

A Figura 4.1 ilustra a criação de rota do protocolo DSDV para os nós vizinhos de MH4 e a Tabela 4.1 mostra a tabela de roteamento da Figura 4.1.

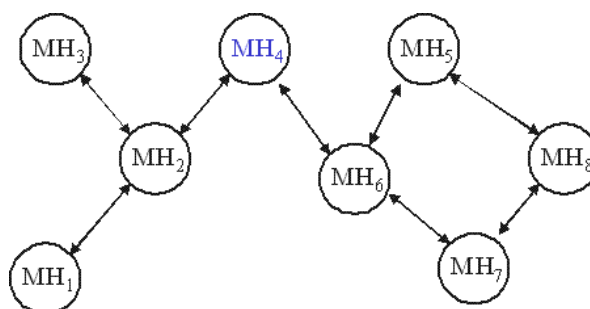


Figura 4.1 - Criação de rota para o nó MH4 empregando o Protocolo DSDV

Tabela 4.1 - Tabela de roteamento da Figura 4.1

<i>Destination</i>	<i>Next Hop</i>	<i>Metric</i>	<i>Seq. No</i>
MH4	MH4	0	S406_MH4
MH1	MH2	2	S128_MH1
MH2	MH2	1	S564_MH2
MH3	MH2	2	S710_MH3
MH5	MH6	2	S392_MH5
MH6	MH6	1	S076_MH6
MH7	MH6	2	S128_MH7
MH8	MH6	3	S050_MH8

A vantagem principal do DSDV sobre os protocolos baseados em vetor de distâncias tradicionais é que eles garantem ausência de *loops*, usando o conceito de número de sequência mantido em cada destino, para indicar qual a rota mais recente. As rotas mais recentes possuem um número de sequência maior e são as mais favoráveis. Caso os números de sequências sejam iguais, a rota que tiver a menor distância será a mais favorável. Os *loops* de rotas podem ocorrer quando informações de roteamento incorretas são mantidas na rede após

uma troca de topologia. O DSDV inicia um processo de atualização de rota periodicamente ou quando a topologia da rede muda.

Após a movimentação do nó MH1 pela rede, os novos vizinhos deste nó passam a ser MH7 e MH8.

A Figura 4.2 apresenta a mudança na topologia da Figura 4.1 e a Tabela 4.2 exibe a atualização do roteamento.

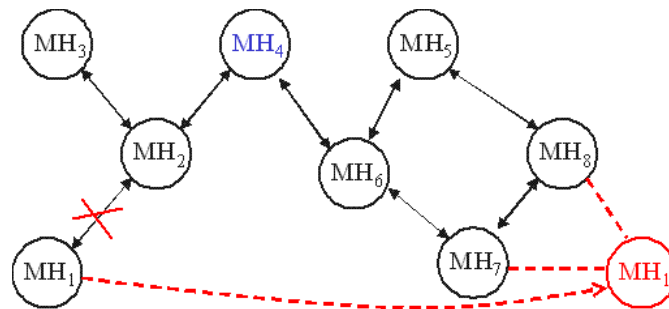


Figura 4.2 - Mudança na topologia da Figura 4.1

Tabela 4.2 - Tabela de roteamento da Figura 4.1

<i>Destination</i>	<i>Next Hop</i>	<i>Metric</i>	<i>Seq. No</i>
MH4	MH4	0	S516_MH4
MH1	MH6	3	S238_MH1
MH2	MH2	1	S674_MH2
MH3	MH2	2	S820_MH3
MH5	MH6	2	S502_MH5
MH6	MH6	1	S186_MH6
MH7	MH6	2	S238_MH7
MH8	MH6	3	S160_MH8

### 4.3 *Source-Initiated On-Demand Routing (Sob demanda) - (Reativo)*

Os protocolos reativos, ao contrário dos pró-ativos, não mantêm informações atualizadas sobre a rede. Estes protocolos têm como estratégia de roteamento descobrir a rota de transmissão quando houver uma requisição por esta rota no nó de origem (JOHNSON; MALTZ, 1996).

Quando um nó requer uma rota para um destino, este inicia um processo de descoberta de rota dentro da rede. Este processo é completado, se uma rota é encontrada ou se todas as possíveis permutações de rota tenham sido examinadas. Uma vez que uma rota tenha sido estabelecida, esta é mantida por um procedimento de manutenção de rota, até que o destino se torne inalcançável por meio de todos os caminhos, a partir da origem, ou até que a rota não seja mais desejada.

Uma vantagem dos protocolos reativos é a baixa taxa de tráfego na rede com pacotes de controle. Isto se deve ao fato de que as rotas de transmissão somente são definidas momentos antes do início das transmissões. Entretanto, a espera pela descoberta de uma rota antes da transmissão pode aumentar consideravelmente o atraso no início desta. Entre os reativos podem ser citados como principais protocolos o *Ad-hoc On-Demand Distance Vector*, *Dynamic Source Routing* e *Temporally Ordered Routing Algorithm*.

#### 4.3.1 *Ad-hoc On-Demand Distance Vector (AODV)*

O AODV é um protocolo de roteamento reativo de vetor de distância. Ele permite que os dispositivos móveis obtenham rapidamente rotas para novos destinos e não exige que eles mantenham rotas para destinos que não estão em comunicação ativa (PERKINS; ROYER, 1999).

Quando há uma necessidade de comunicação entre dois pontos, o nó solicitante envia uma mensagem de requisição de rota (*Route Request*), por *broadcasting*, a todos os nós vizinhos da rede e, se nenhum destes for o nó destino, estes repassam a mensagem à frente, continuamente, até chegar ao nó destino, gravando um caminho temporário a partir do nó anterior. Neste caminho, é designado um número de sequência a cada nó por qual a requisição da rota (*Route Request*) passa, e, ainda, é adicionado a ela o número de saltos ocorridos, que é salvo numa tabela de requisição. Por essa tabela, o algoritmo evita que haja uma repetição das requisições por uma rota já estabelecida e, do mesmo modo, as limita a um tempo de vida, pela qual controla a quantidade de vezes que pode ser retransmitida.

Após a requisição chegar ao nó destino, este envia uma resposta (*Route Reply*) ao nó solicitante, que passa a utilizar o caminho com o menor número de *hops* (saltos) em *unicasting*, e as rotas temporárias que não foram utilizadas são “limpas” a partir de um limite de tempo. A começar daí, são geradas várias mensagens “*hello*” aos nós vizinhos, pelos quais passa a rota. Se uma resposta (*Route Reply*) não for recebida por um determinado tempo, presume-se que ocorreu uma quebra do *link*, e um sinal de erro de rota (*Route Error*) é transmitido ao nó solicitante, que repete o processo de requisição da rota (*Route Request*).

AODV oferece a vantagem de não necessitar de uma grande quantidade de memória ou um grande volume de cálculo, podendo, assim, economizar mais sinal de banda e economizar energia no dispositivo. No entanto, ele necessita de mais tempo para estabelecer uma conexão (PERKINS; ROYER, 1999).

A Figura 4.3 apresenta a criação de rota entre a origem (S) e o destino (D), empregando o protocolo AODV.

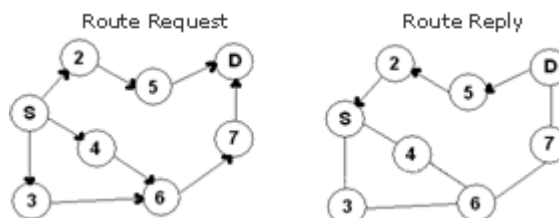


Figura 4.3 - Criação de rota no protocolo AODV (PERKINS; ROYER, 1999)

### 4.3.2 *Dynamic Source Routing (DSR)*

O DSR é um protocolo de roteamento reativo que usa roteamento na fonte para entregar pacotes de dados, ou seja, o nó origem determina toda a sequência de nós por onde passará o pacote até chegar ao seu destino e os cabeçalhos dos pacotes de dados carregam esta sequência de nós (JOHNSON; MALTZ, 1996). Cada nó mantém um *cache*, onde todas as suas rotas conhecidas são armazenadas.

O DSR permite que cada nó mantenha múltiplas rotas para o mesmo destino. Ele consiste de dois mecanismos: descoberta de rotas e manutenção de rotas.

Quando um nó precisa enviar um pacote para outro nó, o nó de origem verifica se possui uma rota para o nó de destino em seu *cache*. Caso a rota não exista, inicia um processo de descoberta de rotas para encontrar dinamicamente uma rota para o destino. O mecanismo de descoberta de rotas do DSR consiste em inundar a rede com pacotes de requisição de rota (*Route Request*).

Quando um nó recebe este pacote, verifica no seu *cache* se tem uma rota para o destino requisitado. Se o nó conhece uma rota, envia para a origem um pacote de resposta (*Route Reply*), que contém uma lista com a sequência de todos os nós até o destino. Caso o nó não tenha uma rota para o destino, encaminha um pacote de requisição da rota (*Route Request*), por *broadcast*, para os seus vizinhos, após ter inserido seu próprio endereço no registro de rotas armazenado no pacote.

No mecanismo de manutenção de rotas, o nó origem detecta se ocorreram alterações na topologia da rede que poderão comprometer o uso das rotas. Desta forma, se um nó detecta algum problema de comunicação com o nó vizinho, envia um pacote de erro de rota (*Route Error*) de volta para o nó de origem.

A Figura 4.4 apresenta a criação de rota entre a origem (S) e o destino (D), exibindo o *cache* das rotas armazenadas, empregando o protocolo DSR.

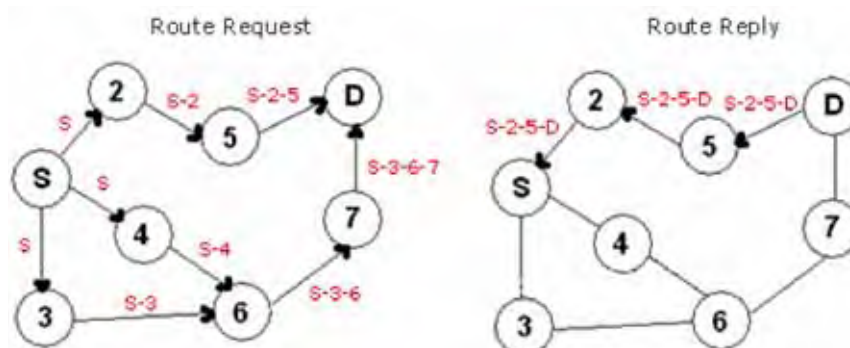


Figura 4.4 - Criação de rota no protocolo DSR (JOHNSON; MALTZ, 1996)

### 4.3.3 Temporally Ordered Routing Algorithm (TORA)

O TORA é baseado no conceito do *link* reverso. Ele é proposto para ser utilizado em redes altamente dinâmicas, é do tipo *source-initiated* (descoberta de rotas baseado na origem). TORA é capaz de encontrar múltiplas rotas de um nó origem até um nó destino. A característica principal deste protocolo é que as mensagens de controle ficam restritas a um conjunto muito pequeno de nós próximos da ocorrência de uma mudança topológica. Para isto, os nós mantêm informação de roteamento sobre os nós adjacentes (PARK; CORSON, 1997).

O TORA executa basicamente três funções, que são a criação, manutenção e deleção de rotas.

No processo de criação, a fonte envia um pacote de requisição de rota (*Route Request*) para o destino desejado. Este pacote se propaga pela rede até chegar ao destino ou até que algum nó intermediário possua uma rota válida. Assim que a rota for descoberta, é enviado um pacote de resposta (*Route Reply*) para a fonte, contendo a altura do nó destino, ou seja, quantos *hops* (saltos) são necessários para que a informação chegue até o seu destino.

O processo de manutenção de rotas ocorre quando se percebe uma quebra no *link*. O nó, então, faz uma verificação de rota e ajusta a sua tabela para a nova configuração.

A deleção da rota ocorre ao se perceber uma fragmentação da rede. É informada, assim, a necessidade de se retirar estas rotas das tabelas.

A Figura 4.5 apresenta o processo de criação de rota. O nó fonte (*Source ID*) envia um pacote de requisição de rota (*Route Request*) pela rede (assim que é feita a análise da

estabilidade do *link*). O nó intermediário que recebe o pacote de requisição de rota (*Route Request*), verifica se este chegou por uma rota estável ou instável. Neste ponto, os pacotes que chegaram pelo caminho instável e os duplicados são retirados da rede. Se o pacote chegou por caminhos estáveis, ele armazena o endereço do nó em sua tabela e, então, repassa o pacote de requisição de rota (*Route Request*) para o nó seguinte e, assim sucessivamente, até que se chegue ao nó destino (*Destination ID*), que envia um pacote de resposta (*Route Reply*) pelo mesmo caminho.

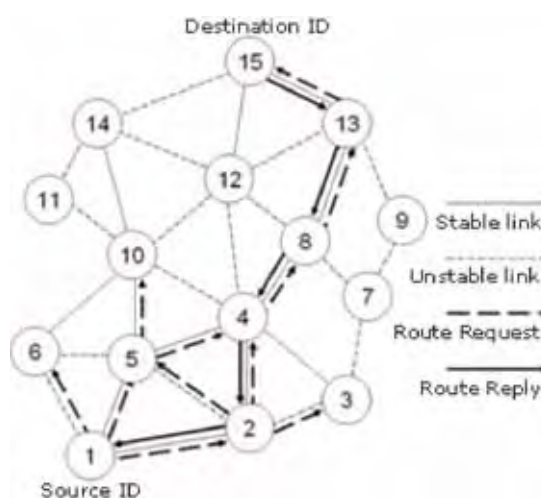


Figura 4.5 - Criação de rota no protocolo TORA (PARK; CORSON, 1997)

#### 4.4 Criação e Configuração de Nós *MobileNode* no NS

Um nó da classe *MobileNode* (nó sem fio) possui as mesmas características de um nó da classe *Node* (nó com fio), com algumas características adicionais, assim como a habilidade de transmitir e receber pacotes (encapsulados em quadros da camada de enlace) em um canal de comunicação sem fio e a capacidade de realizar movimento (ALTMAN; JIMENEZ, 2003).

A Figura 4.6 mostra um exemplo de cenário para a simulação de uma rede sem fio *ad hoc* com três nós em uma área de 500 x 400 metros.

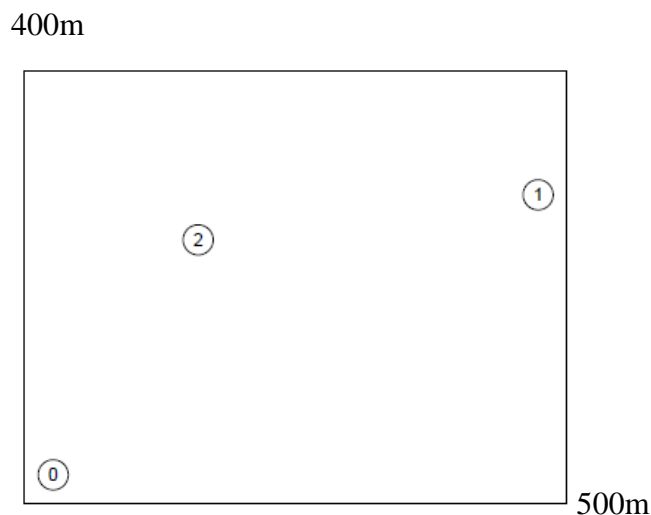


Figura 4.6 - Exemplo de uma rede sem fio *ad hoc* com três nós

A Tabela 4.3 mostra o *script* escrito em OTcl que descreve a simulação para o cenário apresentado na Figura 4.6 (ALTMAN; JIMENEZ, 2003). A sua exibição e posterior detalhamento do código são de fundamental importância para o entendimento dos componentes básicos de configuração de um *MobileNode* de qualquer simulação em rede sem fio empregando o NS.

Tabela 4.3 - *Script* para uma rede *ad hoc*

```
# wireless_adhoc.tcl
# A 3-node example for ad-hoc simulation with DSDV

# Define options
set val(chan) Channel/WirelessChannel ;# Channel Type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
set val(nn) 3 ;# number of mobilenodes
set val(rp) DSDV
set val(x) 500
set val(y) 400
set val(stop) 150
```



```

set ns          [new Simulator]
set tracefd     [open simple.tr w]
set namtrace    [open simwrls.nam w]

$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)

# set up topography object
set topo new Topography]
$topo load_flatgrid $val(x) $val(y)

# create nn mobilenodes [$val(nn)] and attach them to the channel.
create-god $val(nn)

# configure the nodes
    $ns node-config -adhocRouting    $val(rp) \
                    -llType         $val(ll) \
                    -macType        $val(mac) \
                    -ifqType        $val(ifq) \
                    -ifqLen         $val(ifqlen) \
                    -antType         $val(ant) \
                    -propType        $val(prop) \
                    -phyType         $val(netif) \
                    -channelType     $val(chan) \
                    -topoInstance    $topo \
                    -agentTrace      ON \
                    -routerTrace     ON \
                    -macTrace        OFF \
                    -movementTrace   ON

    for {set i 0} {$i < $val(nn) } { incr i } {
        set node_($i) [$ns node]
    }

# provide initial location of mobilenodes
$node_(0) set X_ 5.0
$node_(0) set Y_ 5.0
$node_(0) set Z_ 0.0

$node_(1) set X_ 490.0
$node_(1) set Y_ 285.0
$node_(1) set Z_ 0.0

```

```

$node_(2) set X_ 150.0
$node_(2) set Y_ 240.0
$node_(2) set Z_ 0.0

# generation of movements
$ns at 10.0 "$node_(0) setdest 250.0 250.0 3.0"
$ns at 15.0 "$node_(1) setdest 45.0 285.0 5.0"
$ns at 110.0 "$node_(0) setdest 480.0 300.0 5.0"

# set a TCP connection between node_(0) and node_(1)
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(0) $tcp
$ns attach-agent $node_(1) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 10.0 "$ftp start"

# define node initial position in nam
for {set i 0} {$i < $val(nn)} { incr i } {
# 30 defines the node size for nam
$ns initial_node_pos $node_($i) 30}

# telling nodes when the simulation ends
for {set i 0} {$i < $val(nn)} { incr i } {
    $ns at $val(stop) "$node_($i) reset";}

# ending nam and the simulation
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "stop"
$ns at 150.01 "puts \"end simulation\" ; $ns halt"
proc stop {} {
    global ns tracefd namtrace
    $ns flush-trace
    close $tracefd
    close $namtrace
}

$ns run

```

Para facilitar o trabalho de configuração das simulações, é comum definir as variáveis referentes aos componentes do *MobileNode* no início do *script*, que descreve o modelo de rede a ser simulado (ISI, 2009).

Na Tabela 4.4, além do protocolo de roteamento, os componentes principais de configuração de um *MobileNode* são: Canal de Comunicação (*Channel*), Modelo de Propagação de Ondas de Rádio (*Propagation*), Interface de Rede (*Phy*), Protocolo MAC (*MAC protocol*), Fila da Interface de Rede (*Queue*), Camada de Enlace (LL) e Modelo de Antena (*Antenna*). Outro componente necessário à configuração do *MobileNode*, que está presente na configuração da Tabela 4.4, é a topografia (ou, em outras palavras, o terreno) onde os nós estão dispostos. A topografia define um tipo de terreno e uma área limite para a localização dos nós, a qual é especificada em metros quadrados. Sendo assim, as coordenadas x e y, que definem suas dimensões, devem estar em metros.

Tabela 4.4 - Configuração dos componentes dos *MobileNodes*



A Tabela 4.5 mostra a inicialização das variáveis globais, “ns” é a variável responsável por instanciar um objeto do tipo “Simulator”. Este objeto é responsável pelo agendador de eventos da simulação, assim como por parâmetros da simulação como o formato dos pacotes e o endereçamento padrão da rede. Ele possui funções de fundamental importância a qualquer simulação, como criação de objetos compostos como nós e enlaces, conexões entre agentes de tráfego da rede e especificação de opções de exibição de animações

pelo NAM. As variáveis “tracefd” e “namtrace” são atribuídas respectivamente para abertura de escrita dos arquivos “simple.tr” e “simwrls.nam”

Tabela 4.5 - Inicialização de variáveis para abertura de escrita de arquivos

```
set ns [new Simulator]
set tracefd [open simple.tr w]
set namtrace [open simwrls.nam w]
```

A Tabela 4.6 apresenta o trecho no qual a função “trace-all” do objeto “Simulator” realiza um log de todo o comportamento da rede durante a simulação em um formato geral no arquivo *trace* especificado pela variável “\$tracefd”. A função “namtrace-all-wireless” realiza o log de todo o comportamento da rede no formato apropriado para leitura pelo NAM no arquivo especificado pela variável “\$namtrace”.

Tabela 4.6 - Funções que realizam *log* do comportamento da rede

```
$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
```

A Tabela 4.7 mostra a inicialização da variável *topo*. Ela será responsável por instanciar um objeto do tipo “Topography”. Este objeto é responsável pela criação da área de movimentação dos nós.

Tabela 4.7 - Criação da topografia da rede

```
# set up topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
```

O *General Operations Director* (usualmente tratado pela sigla GOD) é um objeto utilizado pelo NS, para armazenar informações globais sobre o estado do cenário constituído pela configuração da rede e os nós que nela estão presentes (ISI, 2009). Este objeto atua fornecendo uma visão completa da rede. A finalidade do instanciamento deste objeto (Tabela 4.8) e de sua configuração é acelerar a computação do roteamento e conectividade da rede, fornecendo informações para serem usadas em tempo de simulação. Após ter sido instanciado, o objeto guarda o número de nós envolvidos no modelo e pode ser configurado com o menor número de saltos entre um nó e outro.

Tabela 4.8 - Criação do objeto GOD

```
# create nn mobilenodes [$val(nn)] and attach them to the channel.
Create-god $val(nn)
```

A Tabela 4.9 apresenta a configuração dos nós e os níveis de *trace* a serem produzidos pelos nós, existindo quatro níveis: *trace* de agentes (*agentTrace*), *trace* de roteadores (*routerTrace*), *trace* de camada MAC (*macTrace*) e *trace* de movimento (*movementTrace*).

Tabela 4.9 - Configuração dos *MobileNodes*

```
# configure the nodes
    $ns node-config -adhocRouting    $val(rp) \
        -llType                    $val(ll) \
        -macType                    $val(mac) \
        -ifqType                    $val(ifq) \
        -ifqLen                     $val(ifqlen) \
        -antType                    $val(ant) \
        -propType                   $val(prop) \
        -phyType                    $val(netif) \
        -channelType                $val(chan) \
        -topoInstance               $topo \
        -agentTrace                 ON \
        -routerTrace                ON \
        -macTrace                   OFF \
        -movementTrace              ON
```

Após a configuração do *MobileNode*, deve-se realmente criar os nós da rede. Este procedimento está descrito na Tabela 4.10. Os nós *MobileNode* previamente configurados são criados com auxílio da função “node” da classe Simulator.

Tabela 4.10 - Criação dos *MobileNodes*

```
for {set i 0} {$i < $val(nn)} {incr i} {
    set node_($i) [$ns node]
}
```

A configuração de posição e movimento dos nós em uma rede sem fio é, em sua essência, a configuração do cenário desejado para a simulação. Os *MobileNode* foram projetados e implementados para se movimentar numa topologia tridimensional, mas a

terceira dimensão ainda não é usada pelas simulações realizadas com o NS (ISI, 2009). Para especificar a posição e o movimento de um nó, basta configurar manualmente a sua posição inicial e destino numa movimentação com o uso de coordenadas. Este tipo de configuração está citado na Tabela 4.11. A posição de um nó deve ser configurada apenas por meio das coordenadas x e y, sendo a coordenada z configurada como zero. A configuração da movimentação acontece com a configuração discreta do agendador de eventos, utilizando a função auxiliar “at”. A *string* de comando faz com que o simulador movimente, por exemplo, no tempo 10.0 segundos, o nó `node_(0)` para o ponto (x=250.0, y=250.0), no plano definido pela topografia da simulação, com uma velocidade de 3.0 milissegundos.

Tabela 4.11 – Posição inicial e movimentação dos *MobileNodes*

```
# provide initial location of mobilenodes
$node_(0) set X_ 5.0
$node_(0) set Y_ 5.0
$node_(0) set Z_ 0.0

$node_(1) set X_ 490.0
$node_(1) set Y_ 285.0
$node_(1) set Z_ 0.0

$node_(2) set X_ 150.0
$node_(2) set Y_ 240.0
$node_(2) set Z_ 0.0

# generation of movements
$ns at 10.0 "$node_(0) setdest 250.0 250.0 3.0"
$ns at 15.0 "$node_(1) setdest 45.0 285.0 5.0"
$ns at 110.0 "$node_(0) setdest 480.0 300.0 5.0"
```

A Tabela 4.12 apresenta a criação dos fluxos de tráfego pelos agentes TCP (envio) e *Sink* (recepção) e da aplicação FTP. Instancia-se um novo objeto da classe *Agent/TCP* e o associa à variável “tcp”. Instancia-se um novo objeto da classe *Agent/TCPSink* e o associa à variável “sink”. Instancia-se um novo objeto da classe *Application/FTP* e o associa à variável “ftp”. Isto é necessário para a simulação de um determinado protocolo da camada de aplicação que utilizará um agente de tráfego (protocolo da camada de transporte) previamente configurado. No caso, como o FTP é uma aplicação que se baseia no protocolo da camada de transporte TCP, esta aplicação fará uso do agente de tráfego TCP, instanciado anteriormente.

A escolha do tipo de aplicação é especialmente importante para modelar determinados tipos de tráfego de maior interesse em certas configurações de rede. A “função attach-agent” associa o objeto de uma das classes Agent, por exemplo, \$tcp a um objeto nó, neste caso, \$node\_(0). Isso significa que, durante a execução da simulação, a ação de tráfego (envio) associada ao agente de tráfego referenciado pela variável \$tcp será realizada pelo nó \$node\_(0). A ação de tráfego (recebimento) associada ao agente de tráfego referenciado pela variável \$sink será realizada pelo nó \$node\_(1). Logo, é especificada a ligação (connect) entre os agentes de tráfego previamente definidos, conectando a origem (\$tcp) a um destino (\$sink) para o tráfego de dados. A função membro do agendador de eventos “at” configura uma ação a ser realizada pelo simulador e o tempo em que esta ação deve ser executada. Na tabela 4.12, o agendador de eventos é configurado para iniciar a aplicação \$ftp.

Tabela 4.12 - Criação da conexão TCP entre os *MobileNodes*

```
# set a TCP connection between node_(0) and node_(1)
set tcp [new Agent/TCP/Newreno]
set sink [new Agent/TCPSink]
$ns attach-agent $node_(0) $tcp
$ns attach-agent $node_(1) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 10.0 "$ftp start"
```

Na Tabela 4.13, é apresentada a posição inicial dos nós sem fio no NAM.

Tabela 4.13 - Definição da posição inicial do *MobileNode* no NAM

```
# define node initial position in nam
for {set i 0} {$i < $val(nn)} { incr i } {
# 30 defines the node size for nam
$ns initial_node_pos $node_($i) 30
}
```

Na Tabela 4.14, é mostrado o trecho em que é avisado para todos os nós quando a simulação termina.

Tabela 4.14 - Aviso aos *MobileNodes* do tempo final da simulação

```
# telling nodes when the simulation ends
for {set i 0} {$i < $val(nn) } { incr i } {
    $ns at $val(stop) "$node_($i) reset";
}
}
```

Concluídas as etapas mostradas anteriormente, é necessário finalizar a simulação. A Tabela 4.15 mostra o procedimento “stop”. Este procedimento é responsável por fechar os arquivos e encerrar efetivamente a simulação. A última linha (\$ns run) executa a simulação e inicia o escalonador de eventos do NS.

Tabela 4.15 - Procedimento de encerramento da simulação

```
# ending nam and the simulation
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "stop"
$ns at 150.01 "puts \"end simulation\" ; $ns halt"
proc stop {} {
    global ns tracefd namtrace
    $ns flush-trace
    close $tracefd
    close $namtrace
}
$ns run
```

Após a configuração do modelo e parâmetros da simulação, conforme exibido na Tabela 4.3, basta executar o NS, passando, como primeiro argumento da linha de comando, o nome do script OTcl, como mostra a Tabela 4.16.

Tabela 4.16 - Execução da simulação de exemplo wireless\_adhoc.tcl

```
$ ns wireless_adhoc.tcl
```

#### 4.5 Geração Automática de Cenários e Padrões de Tráfego

Para auxiliar a realização de simulações com várias configurações de posição, movimento e tráfego entre nós, em uma rede sem fio, a equipe do Projeto Monarch desenvolveu dois programas que estão disponíveis no NS (GREIS, 1996): o “setdest” e o



“cbrgen.tcl”. Ambos os programas estão localizados no diretório ns-2.34/indep-utils/cmu-scen-gen. O “setdest” é um gerador de cenários, ou seja, posições e movimentos dos nós, desenvolvido em C++ e já compilado. O “cbrgen.tcl”, apesar do nome indicar apenas o tráfego do UDP do tipo CBR, pode ser usado para gerar padrões de tráfego tanto para fluxos UDP, simulando aplicações do tipo CBR, quanto TCP, simulando aplicações FTP.

#### 4.5.1 Geração de Cenários com “setdest”

A geração de cenários com o programa “setdest” é bastante direta. Basta ter uma ideia de como se deseja modelar a rede a ser simulada e especificar as *flags* disponibilizadas pelo programa (GREIS, 1996). A Tabela 4.17 mostra um exemplo de cenário simples a ser criado.

Tabela 4.17 - Execução do programa “setdest” para geração de um cenário de rede sem fio

```
./setdest -v 1 -n 50 -p 900 -M 20 -t 900 -x 1500 -y 300 > scenario.tcl
```

Os parâmetros utilizados no exemplo da Tabela 4.17 correspondem a:

- v 1: versão do cenário a ser gerada (a versão 1 corresponde à versão original desenvolvida pela equipe do Projeto Monarch);
- n: número de nós;
- p: “*pause time*” (tempo em que os nós devem ficar parados) em segundos;
- M: velocidade máxima dos nós em metros/segundo;
- t: tempo total da simulação em segundos;
- x: limite da coordenada x do terreno usado pela simulação em metros;
- y: limite da coordenada y do terreno usado pela simulação em metros.

Como se pode notar, foi utilizado um valor de “*pause time*” igual ao valor de tempo total da simulação. Isto foi feito propositalmente, para geração de um cenário onde não haverá movimento dos nós. Caso se deseje gerar padrões de movimento em redes sem fio, basta especificar um valor de “*pause time*” menor que o tempo total da simulação. Assim, o “setdest” criará padrões de movimento usando o algoritmo *Random Waypoint*. Como o “setdest” cria o cenário e o imprime na saída padrão (stdout) no *shell* do sistema operacional, é necessário redirecionar a saída em um arquivo, para posterior carregamento do mesmo,

dentro do *script* que define a simulação. Isto é feito através do caractere “>”, seguido do nome do arquivo a conter as informações capturadas da saída padrão.

#### 4.5.2 Geração de Padrões de Tráfego com “cbrgen.tcl”

A geração de padrões de tráfego com o auxílio do *script* “cbrgen.tcl” é tão simples quanto à geração de cenários com o “setdest”. A Tabela 4.18 apresenta um exemplo para gerar padrões de tráfego (GREIS, 1996).

Tabela 4.18 - Geração do padrão de tráfego com o *script* “cbrgen.tcl”

```
$ ns cbrgen.tcl -type cbr -nn 50 -seed 2 -mc 30 -rate 4.0 > traffic.tcl
```

Os parâmetros utilizados no exemplo da Tabela 4.18 correspondem a:

- type: tipo de tráfego (CBR/TCP) da camada de aplicação;
- nn: número total de nós;
- seed: semente para gerador de números aleatórios;
- mc: número máximo de conexões entre os nós;
- rate: taxa de envio (pacotes por segundo).

Como o “cbrgen.tcl” está escrito em OTcl, o próprio binário do NS (denominado “ns”) deve ser usado como interpretador para execução do *script* que utiliza uma semente para o gerador de números aleatórios que, por sua vez, é usado para definir o tempo de início de cada um dos fluxos de tráfego. Este tempo é um valor aleatório entre 0 e 180 segundos.

Assim como o “setdest”, o “cbrgen.tcl” imprime a configuração dos fluxos de tráfego na saída padrão e, por este motivo, a saída deve ser redirecionada em um arquivo para posterior carregamento por um *script* de configuração de uma simulação.

#### 4.6 Analisando o Arquivo *Trace* da Simulação

O processo de simulação no NS envolvendo redes sem fio é consideravelmente diferente em relação ao das redes cabeadas. Uma das mudanças mais significativas ocorre em relação ao arquivo *trace* que possui um *layout* diferente daquele das redes com fio. Na

realidade, existem duas opções de arquivo *trace* quando se utiliza uma simulação com redes sem fio: o *trace* clássico e o novo *trace* (ISI, 2009).

Para este trabalho, será utilizado apenas o novo formato *trace*. Como mostrado na Tabela 4.19, a configuração do novo formato deve ser feita antes do comando de *trace* universal, “*trace-all*”, já que a primitiva, “*use-newtrace*”, configura uma variável do simulador denominada *newTraceFormat*, de forma a possibilitar a produção de *trace* no novo formato.

Tabela 4.19 - Configuração do novo formato *trace*

```
$ns_ use-newtrace
$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
```

O novo formato *trace* do NS disponibiliza uma gama de informações sobre as camadas, os nós e o roteamento dos pacotes entre eles. Algumas linhas de saída do novo formato *trace* podem ser vistas na Tabela 4.20. Nela, estão transcritas seis linhas de *trace* que, dado seu comprimento, estão dispostas com quebras de linha. Para facilitar a identificação, cada nova linha de *trace* começa com o caractere “s” ou “r”.

Tabela 4.20 - Exemplo de arquivo com o novo formato *trace*

```
s -t 0.086515825 -Hs 3811 -Hd -2 -Ni 3811 -Nx 5821.40 -Ny 1658.53 -Nz
0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 3811.1 -Id
3812.0 -It cbr -Il 64 -If 0 -Ii 0 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 0
s -t 0.121304076 -Hs 1439 -Hd -2 -Ni 1439 -Nx 9551.09 -Ny 2971.31 -Nz
0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 1439.0 -Id
1440.0 -It cbr -Il 64 -If 0 -Ii 1 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 0
s -t 0.129406862 -Hs 730 -Hd -2 -Ni 730 -Nx 1782.55 -Ny 3459.60 -Nz
0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 730.1 -Id
731.1 -It cbr -Il 64 -If 0 -Ii 2 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 0
s -t 0.139662372 -Hs 50 -Hd -2 -Ni 50 -Nx 5148.92 -Ny 426.73 -Nz 0.00
-Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 50.1 -Id 52.0 -
It cbr -Il 64 -If 0 -Ii 3 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 0
s -t 0.151689062 -Hs 3262 -Hd -2 -Ni 3262 -Nx 511.00 -Ny 2046.89 -Nz
0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 3262.0 -Id
3263.0 -It cbr -Il 64 -If 0 -Ii 4 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 0
r -t 0.195079601 -Hs 3812 -Hd 3812 -Ni 3812 -Nx 5451.95 -Ny 1095.33 -
Nz 0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 13a -Md ee4 -Ms 2257 -Mt 800 -
Is 3811.1 -Id 3812.0 -It cbr -Il 84 -If 0 -Ii 0 -Iv 27 -Pn cbr -Pi 0
```

Na Tabela 4.21, são descritas as especificações de cada campo do novo formato *trace*. Essas informações estão separadas por um caractere de espaço em branco e, com exceção da primeira informação, que é um caractere de indicação do tipo do evento, todas as demais informações são precedidas por flags que indicam seu significado (ISI, 2009).

Tabela 4.21 - Especificação dos campos do novo formato *trace*

<b>Campo</b>	<b>Especificação</b>
<letra>	A <letra> que antecede o campo -t define o tipo de evento: s = envio de pacote, f = encaminhamento de pacote, r = recepção de pacote, d = perda de pacote.
-t	Tempo de simulação na ocorrência do evento
-Hs	Identificação do nó onde ocorreu o evento
-Hd	Identificação do próximo nó a ser percorrido pelo pacote
-Ni	Identificação do nó
-Nx	Coordenada X do nó
-Ny	Coordenada Y do nó
-Nz	Coordenada Z do nó
-Ne	Nível de energia do nó
-Nl	Nível do <i>trace</i> : AGT (Agent), RTR (Router), MAC, IFQ (Interface Queue)
-Nw	Razão do evento: END: descarte no fim da simulação; TTL: tempo de vida do pacote alcançou zero antes de chegar ao destino CBK ( <i>Callback</i> ): nenhuma entrada de rota para o destino IFQ: pacotes descartados, devido à fila cheia no buffer de transmissão ARP ( <i>Address Resolution Protocol</i> ): falha na resolução de endereços entre endereços IP ( <i>Internet Protocol</i> )
-Ma	Duração do pacote no nível MAC
-Md	Endereço <i>ethernet</i> de destino
-Ms	Endereço <i>ethernet</i> de envio
-Mt	Tipo <i>ethernet</i>
-Is	Endereço (IP) fonte (porta fonte)
-Id	Endereço (IP) destino (porta destino)
-It	Tipo do pacote (CBR, Pareto, Exponencial etc)
-Il	Tamanho do pacote
-If	Identificação do fluxo ao qual o pacote pertence
-Ii	Identificação única do pacote
-Iv	Valor do TTL
-Pn	Dependendo do tipo de pacote CBR, ARP, DSR, TCP, estes campos podem assumir significados diferentes.
-Pi	
-Pf	
-Po	

No próximo capítulo, será abordado um estudo com três tipos de redes de acesso sem fio (sem mobilidade, com mobilidade restrita e com mobilidade irrestrita), para avaliação de desempenho, empregando o NS nas simulações e apresentando os resultados obtidos.

## CAPÍTULO 5

### RESULTADOS

Neste capítulo são apresentados três tipos de redes de acesso sem fio para avaliação de desempenho. Inicia-se com uma topologia de rede simples e sem nenhuma mobilidade do usuário, estendendo para uma topologia de rede intermediária, cujo usuário terá mobilidade restrita, até uma topologia de rede complexa e sem nenhuma restrição de mobilidade para o usuário. Nos três tipos de redes, a topologia empregada será infraestruturada, em que, parte da infraestrutura da rede é cabeada e parte é sem fio e ambas interagem por meio de estações base. Será empregado o NS, para processar os *scripts* que possuem os parâmetros para as simulações, a linguagem interpretada AWK, utilizada para extração de informações nos arquivos *trace* e o Gnuplot, para plotagem dos gráficos.

O termo vazão, que será usado como métrica nas simulações deste capítulo, é definido como a razão (requisições por unidade de tempo) em que as requisições podem ser servidas por um sistema. Para análise da rede, a vazão é medida em pacotes por segundo (pps) ou bits por segundo (bps), podendo, ainda, ser medida em kilobits por segundo (Kbps) (TANENBAUM, 2003). O estudo da vazão da rede, realizado neste trabalho, compreende uma análise da razão entre a quantidade de dados que é transferida pela rede entre um nó de origem e um nó de destino e o tempo decorrido até que essa quantidade de dados seja totalmente transferida. O Apêndice A, apresenta o arquivo “avgStats.awk”, que será usado neste capítulo, para calcular a vazão média da rede.

O estudo realizado com o NS, nas simulações de redes com e sem fio, originou seis trabalhos aceitos e apresentados em três congressos (PRETE; SHINODA, 2008) (PRETE; SHINODA, 2009a) (PRETE; SHINODA, 2010a), um simpósio (PRETE; SHINODA, 2009b), um encontro (PRETE; SHINODA, 2010b), uma conferência (PRETE; SHINODA, 2010c) e um workshop (PRETE; SHINODA, 2010d).

#### 5.1 Rede de Acesso Sem Fio e Sem Mobilidade

Neste subcapítulo, serão abordados dois cenários, a partir de uma topologia de rede simples, sem nenhuma mobilidade dos usuários. O primeiro cenário avalia o desempenho da rede, em função da vazão média e instantânea, na transmissão de notícias em tempo real (PRETE; SHINODA, 2009b). Já o segundo cenário avalia o comportamento da rede, em

função da vazão média, na transmissão de arquivos dentro de uma intranet (PRETE; SHINODA, 2010d).

### 5.1.1 Topologia da Rede

A topologia de rede infraestruturada (com e sem fio) é composta por um nó cabeado  $W(0)$  conectado a cinco estações base (EB1, EB2, EB3, EB4 e EB5), com uma distância de aproximadamente 2 km. As estações base (EB) são responsáveis pelo roteamento dos pacotes para os nós sem fio que pertencem à área de alcance de cada estação. A distância entre os nós sem fio na área de contorno da estação base é de 200 m.

Em simulações de redes infraestruturadas, é usado o endereçamento hierárquico, a fim de rotear pacotes entre domínios com e sem fio. Na Figura 5.1, são criados 6 domínios, 1 domínio para rede cabeada ( $W0: 0.0.0$ ) e 5 domínios para rede sem fio (EB1: 1.0.0, EB2: 2.0.0, EB3: 3.0.0, EB4: 4.0.0 e EB5: 5.0.0). Assim como o domínio, são definidos os endereços para os *clusters* e nós de cada domínio. Por exemplo, o endereçamento hierárquico (domínio.cluster.nó) do nó n7 seria representado como 1.0.2, onde 1 é o domínio, 0 é o cluster e 2 é o nó sem fio da rede.

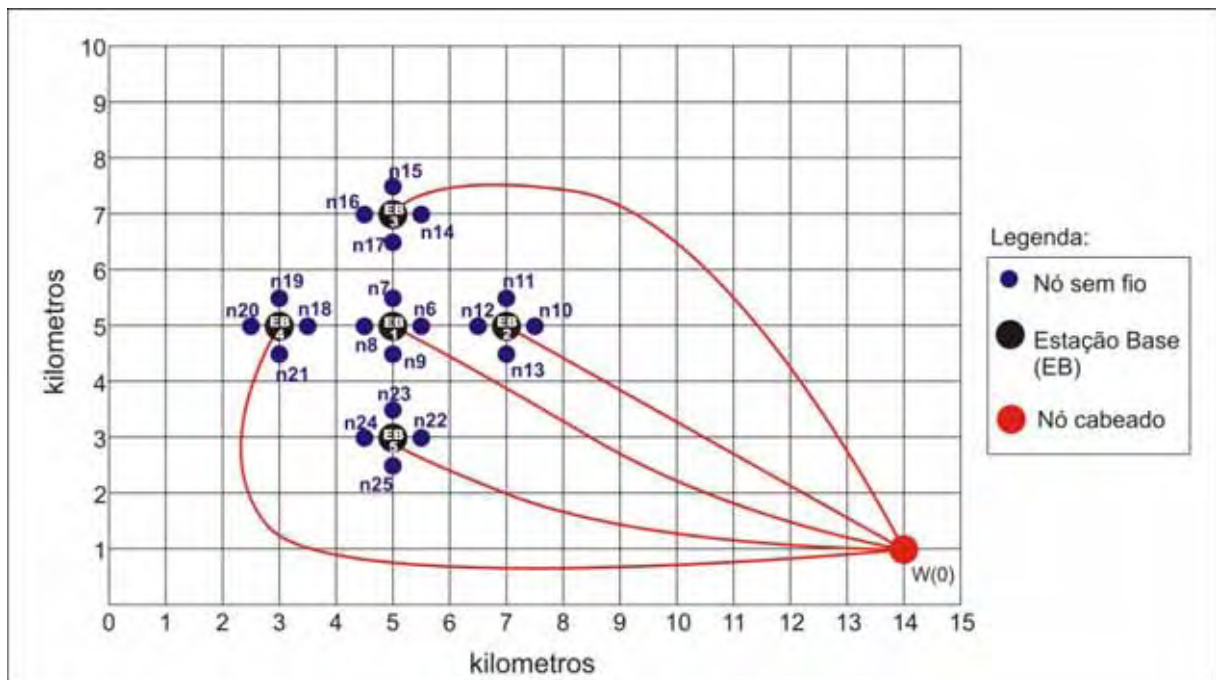


Figura 5.1 - Topologia da rede sem nenhuma mobilidade do usuário

### 5.1.2 Parâmetros da Rede - Primeiro Cenário Sem Mobilidade

Neste cenário, será abordada a transmissão de pacotes do nó cabeado  $W(0)$  até as estações base. A Tabela 5.1 apresenta os principais parâmetros da rede, considerada na simulação de transmissão de notícias em tempo real (PRETE; SHINODA, 2009b).

Tabela 5.1 - Parâmetros da rede considerada na transmissão de notícias em tempo real

Parâmetro	Valor
Canal	WirelessChannel
Propagação	TwoRayGround
Interface de rede	Wireless Phy
Camada MAC	802_11
Tipo de fila	DropTail
Camada de enlace	LL
Modelo da antena	OmniAntenna
Número máximo de pacotes na fila	50
Número de nós sem fio	20
Protocolo de roteamento	DSDV
Área de cobertura (XxY)	10 km x 15 km
Número de nós cabeado	1
Número de Estação Base	5
Taxa de transmissão	11 Mb
Tamanho dos pacotes	1023
Tipo de aplicação	CBR

No NS, todo e qualquer protocolo de transporte é definido através de um agente. Na Figura 5.1, aplicando os parâmetros da Tabela 5.1, temos dois agentes definidos. Um agente UDP anexado ao nó  $W(0)$  e outro agente de recepção (*Null*) vinculado aos nós sem fio que empregam também o protocolo de roteamento sem fio DSDV. Além disso, a aplicação associada entre o nó  $W(0)$  e os nós sem fio é o CBR. Os agentes de recepção servem para receber os pacotes enviados pelos protocolos de transporte. O agente *Null* recebe os pacotes do protocolo UDP e os anula. O protocolo pró-ativo DSDV mantém uma tabela de rotas possíveis para o tráfego de pacotes por toda a rede. A vantagem para aplicações usarem o

UDP é que a transmissão de dados fica mais rápida, acelerando o envio de pacotes, já que o transmissor não precisará esperar receber uma mensagem do receptor para enviar o próximo pacote. Um dos usos mais conhecidos do protocolo UDP é a transmissão de informações em tempo real. Por este motivo, optou-se pela utilização desse agente para a aplicação de transmissão de notícias em tempo real nas simulações deste cenário (ALTMAN; JIMENEZ, 2003).

### 5.1.2.1 Script de Simulação

Para a topologia definida na Figura 5.1 e empregando os parâmetros da Tabela 5.1, foi desenvolvido o arquivo “wireless\_srmais.tcl” que apresenta o *script* de simulação da rede, conforme detalhado no Apêndice B (PRETE; SHINODA, 2009b).

### 5.1.2.2 Resultados

A Figura 5.2 apresenta a comparação da estatística média de vazão dos pacotes transmitidos do nó cabeado W(0) até as estações base. Estas, por sua vez, fazem o roteamento dos pacotes até os nós sem fio na sua área de alcance. Observa-se que a vazão média apresentou-se semelhante e com poucas oscilações (PRETE; SHINODA, 2009b).

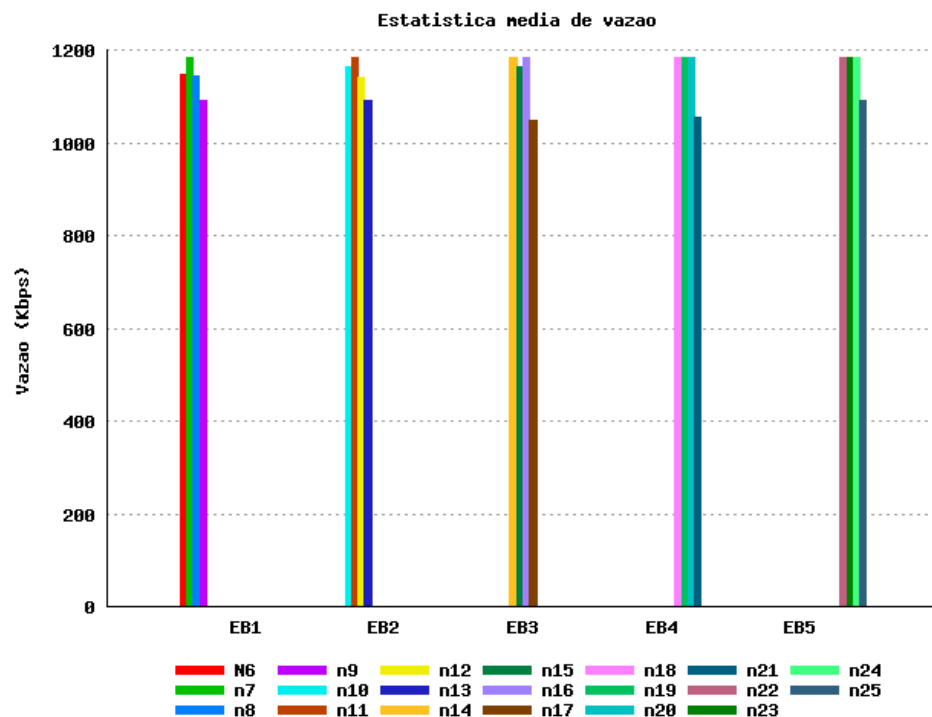


Figura 5.2 - Estatística média de vazão dos pacotes transmitidos do nó cabeado W(0) até as estações bases.



Analisando as próximas figuras (5.3, 5.4, 5.5, 5.6 e 5.7), verificou-se que a vazão instantânea (medida a cada 1.0 segundo) dos pacotes transmitidos do nó cabeado até os nós sem fio foi quase constante e com poucas oscilações (PRETE; SHINODA, 2009b). O Apêndice C apresenta o script do arquivo “instantThroughput.awk” para calcular a vazão instantânea da rede.

A Figura 5.3 apresenta a vazão instantânea dos pacotes transmitidos do nó cabeado  $W(0)$  até os nós sem fio (n6, n7, n8 e n9) relativos à estação base (EB1) apresentados na topologia da Figura 5.1.

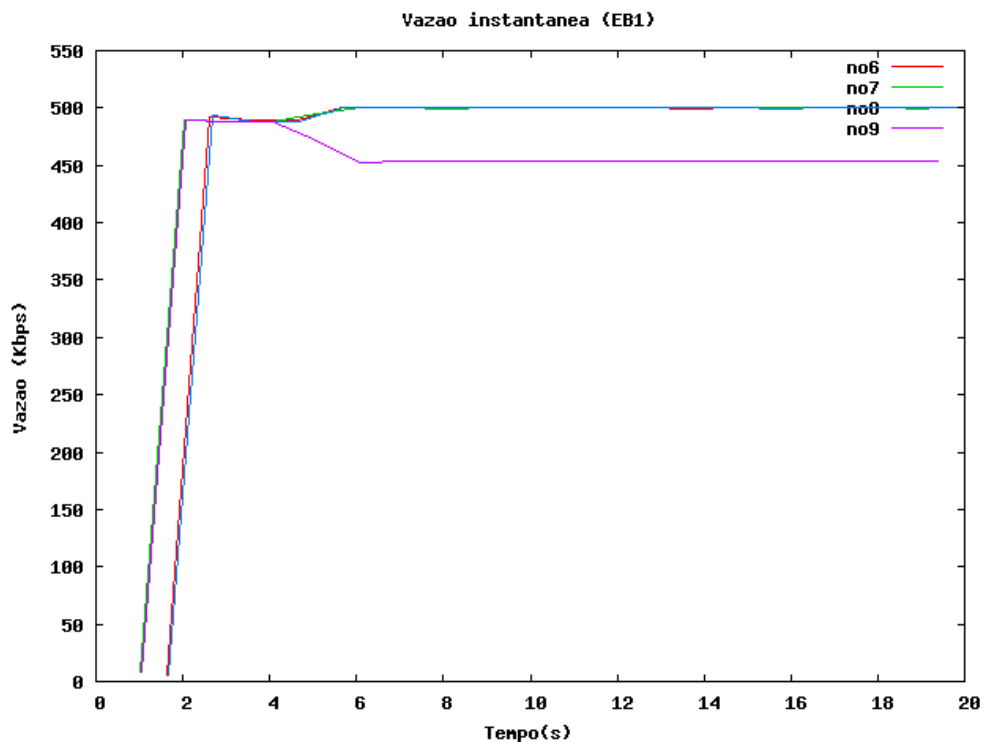


Figura 5.3 - Vazão instantânea do nó cabeado  $W(0)$  até os nós sem fio relativos à EB1

A Figura 5.4 apresenta a vazão instantânea dos pacotes transmitidos do nó cabeado  $W(0)$  até os nós sem fio (n10, n11, n12 e n13), relativos à estação base (EB2), apresentados na topologia da Figura 5.1.

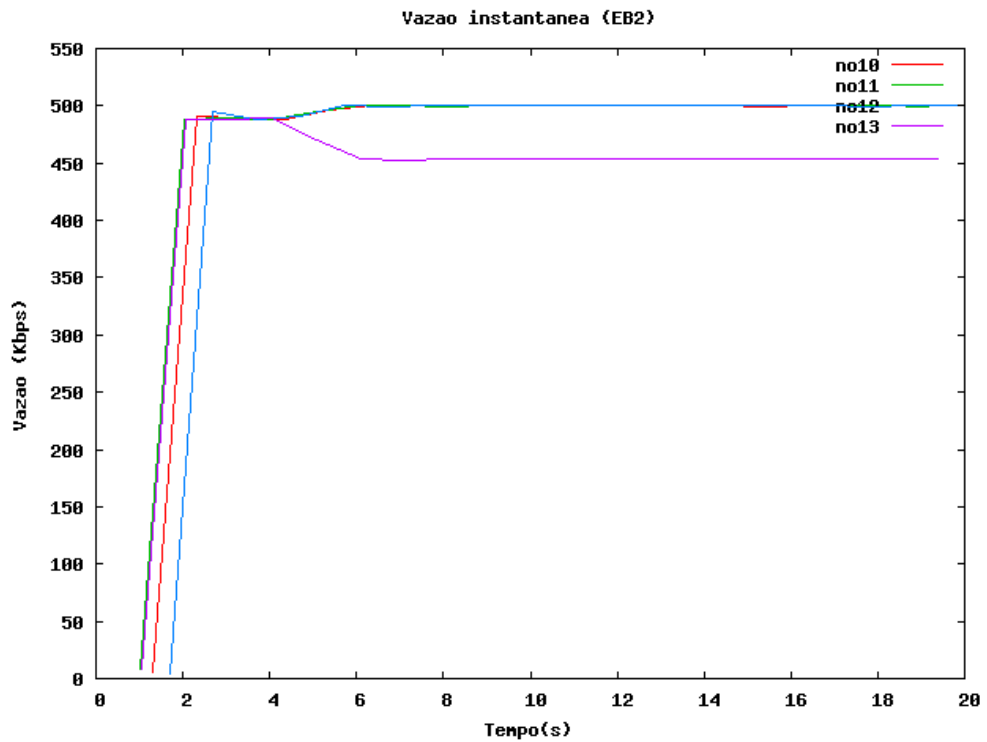


Figura 5.4 - Vazão instantânea do nó cabeado W(0) até os nós sem fio relativos à EB2

A Figura 5.5 apresenta a vazão instantânea dos pacotes transmitidos do nó cabeado W(0) até os nós sem fio (n14, n15, n16 e n17), relativos à estação base (EB3), apresentados na topologia da Figura 5.1.

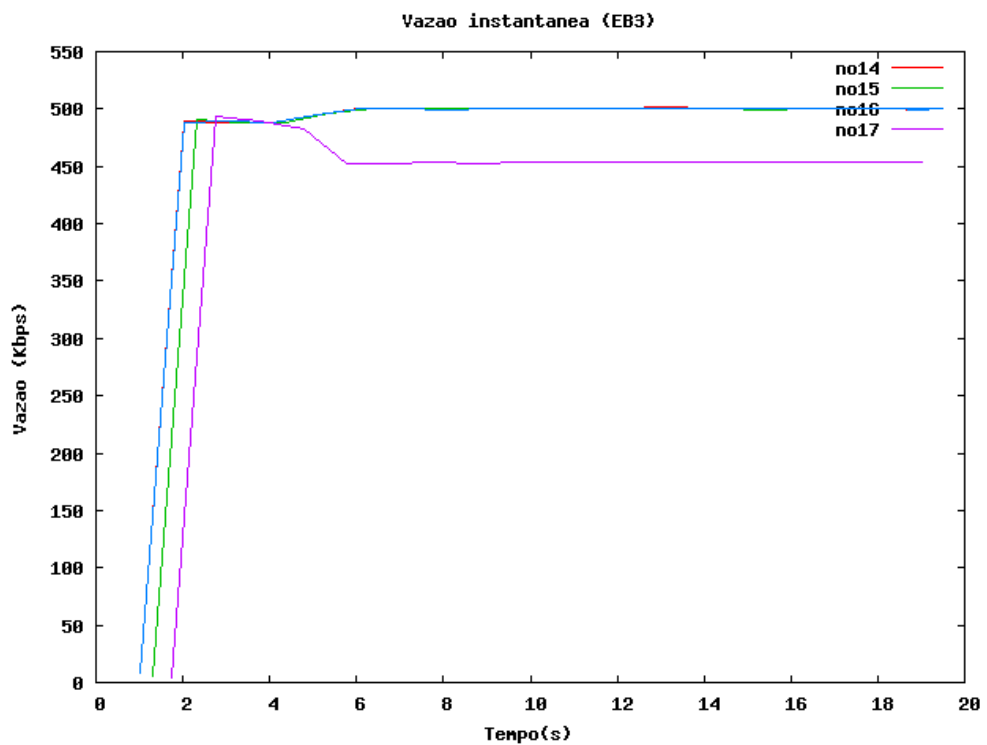


Figura 5.5 - Vazão instantânea do nó cabeado W(0) até os nós sem fio relativos à EB3

A Figura 5.6 apresenta a vazão instantânea dos pacotes transmitidos do nó cabeado W(0) até os nós sem fio (n18, n19, n20 e n21) relativos à estação base (EB4) apresentados na topologia da Figura 5.1.

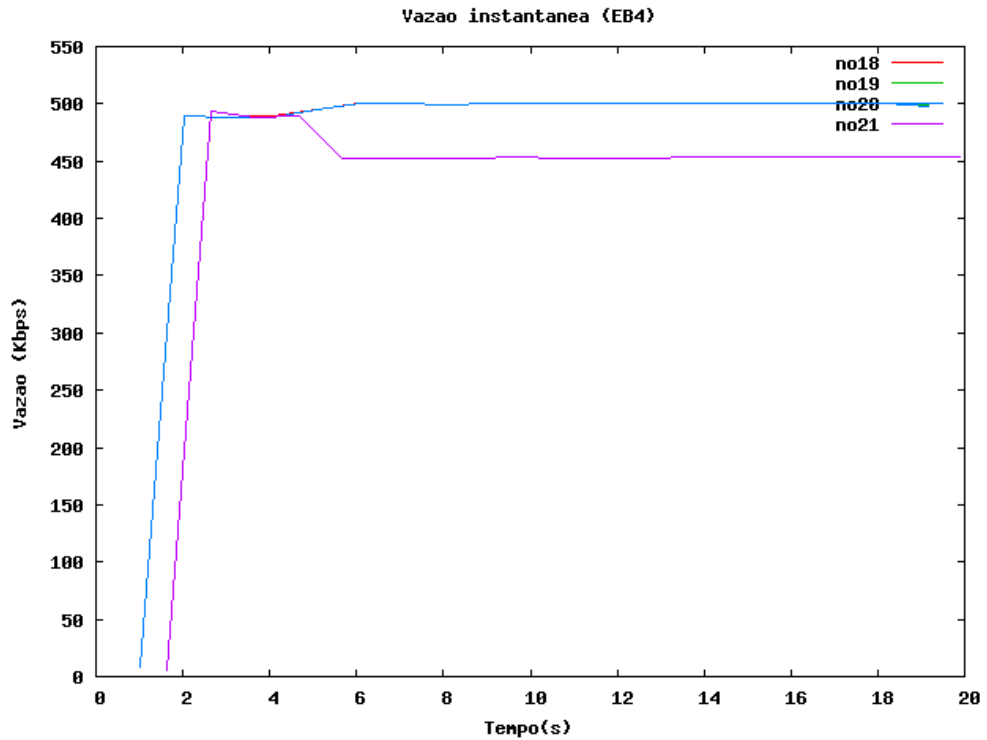


Figura 5.6 - Vazão instantânea do nó cabeado W(0) até os nós sem fio relativos à EB4

A Figura 5.7 apresenta a vazão instantânea dos pacotes transmitidos do nó cabeado W(0) até os nós sem fio (n22, n23, n24 e n25), relativos à estação base (E5), apresentados na topologia da Figura 5.1.

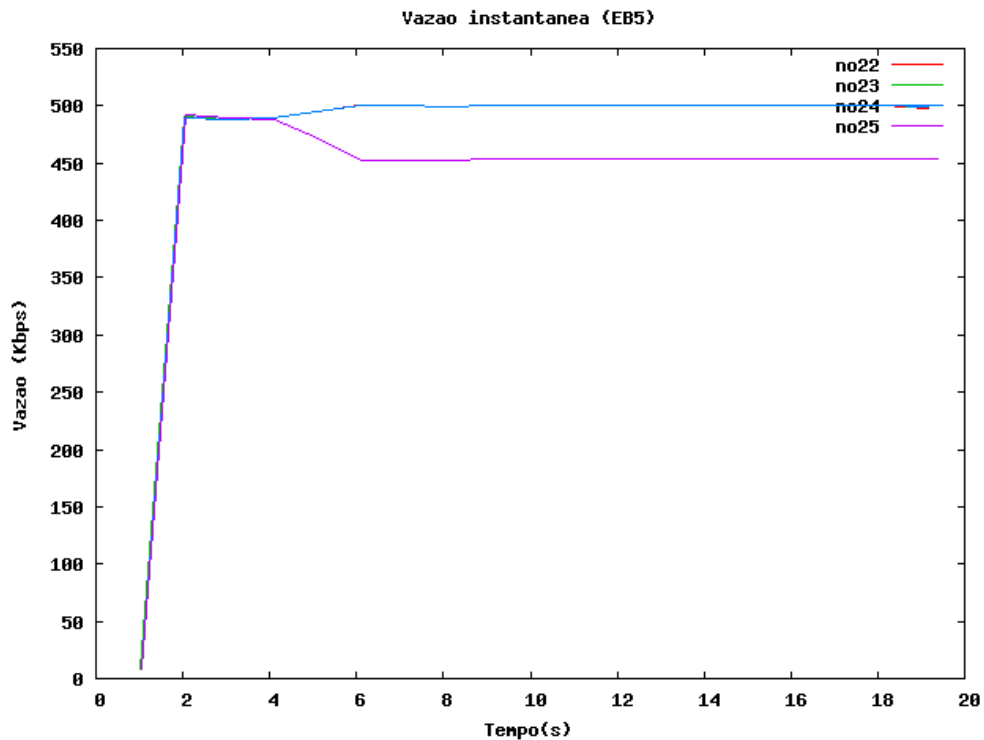


Figura 5.7 - Vazão instantânea do nó cabeado W(0) até os nós sem fio relativos à EB5

### 5.1.2.3 Conclusão

Os resultados mostraram que a utilização do protocolo de transporte UDP e o protocolo de roteamento DSDV conduziram os pacotes com uma vazão média e instantânea quase estável para todos os nós sem fio da rede, ocasionando poucas oscilações ao longo da simulação.

Portanto, para a aplicação de transmissão de notícias em tempo real (CBR), é importante que todos os receptores recebam a notícia ao mesmo tempo, sem muita perda na taxa de transmissão de dados. A proposta da topologia da Figura 5.1, com os parâmetros da Tabela 5.1, evidenciaram resultados satisfatórios para esta aplicação.

### 5.1.3 Parâmetros da Rede - Segundo Cenário Sem Mobilidade

Neste segundo cenário, alguns parâmetros da rede foram alterados e uma análise mais profunda foi realizada em relação ao primeiro cenário do subcapítulo 5.1.2. Na Tabela 5.2, observa-se a alteração do protocolo de aplicação para FTP. Consequentemente, o protocolo de transporte foi modificado para TCP, já que o objetivo deste segundo cenário é transmitir arquivos em uma intranet, sendo estes protocolos os mais adequados. Para uma análise mais

profunda, em relação ao primeiro cenário, que considerou apenas a análise dos pacotes transmitidos do nó cabeado  $W(0)$  até os nós sem fio, neste segundo cenário, será analisado, além do roteamento dos pacotes do nó cabeado  $W(0)$  até os nós sem fio, o roteamento de pacotes feito pelos nós sem fio pertencentes à área de cobertura de sua respectiva estação base e o roteamento de pacotes feito pelos nós sem fio de diferentes estações base (PRETE; SHINODA, 2010d).

Tabela 5.2 - Parâmetros da rede considerada na transmissão de arquivos em uma intranet

<b>Parâmetro</b>	<b>Valor</b>
Canal	WirelessChannel
Propagação	TwoRayGround
Interface de rede	Wireless Phy
Camada MAC	802_11
Tipo de fila	DropTail
Camada de enlace	LL
Modelo da antena	OmniAntenna
Número máximo de pacotes na fila	50
Número de nós sem fio	20
Protocolo de roteamento	DSDV
Área de cobertura (XxY)	10 km x 15 km
Número de nós cabeado	1
Número de estações base	5
Taxa de transmissão	11 Mb
Tamanho dos pacotes	1023
Tipo de aplicação	FTP

Na Figura 5.1, aplicando os parâmetros da Tabela 5.2, tem-se dois agentes definidos. Um agente TCP anexado ao nó  $W(0)$  e outro agente de recepção (Sink) vinculado aos nós sem fio, que empregam também o protocolo de roteamento sem fio DSDV. Além disso, a aplicação associada entre o nó  $W(0)$  e os nós sem fio é o FTP. O agente Sink recebe os pacotes do protocolo TCP e gera os pacotes de reconhecimento ACKs (Acknowledgements). O protocolo pró-ativo DSDV tem a mesma finalidade descrita no subcapítulo 5.1.2. A vantagem do TCP para aplicações é a versatilidade e robustez deste protocolo, tornando-o

adequado a redes globais, já que este verifica se os dados são enviados de forma correta, na sequência apropriada e sem erros pela rede, garantindo a transmissão de pacotes. Um dos usos mais conhecidos do protocolo TCP é nas aplicações cibernéticas, como o SSH (*Secure Shell*), FTP, HTTP (*Hypertext Transfer Protocol*), entre outros. Por este motivo, optou-se pela utilização desse agente para a aplicação FTP na transmissão de arquivos em uma intranet nas simulações deste cenário (ALTMAN; JIMENEZ, 2003).

### 5.1.3.1 Resultados

Os resultados apresentados correspondem a três análises, em decorrência do desempenho no tráfego de pacotes em uma rede infraestruturada.

#### 5.1.3.1.1 Primeira Análise

Na primeira análise, é verificada a vazão média do roteamento de pacotes, feito pelos nós sem fio pertencentes à área de cobertura de sua respectiva estação base. O arquivo “wireless\_wpgcps\_1.tcl” apresenta o *script* de simulação da rede, conforme detalhado no Apêndice D (PRETE; SHINODA, 2010d).

A Figura 5.8 ilustra o estabelecimento de conexão TCP, entre os nós sem fio, por meio da EB1. O mesmo ocorre com as outras estações base. O nó sem fio n6 inicia a conexão enviando um pacote TCP que é roteado pela EB1 e espera-se que o nó sem fio n7 aceite a conexão enviando um pacote ACK de confirmação. Durante o estabelecimento de conexão, são trocados números de sequência iniciais entre os sistemas finais que irão servir para identificar os dados ao longo do fluxo, bem como servir de contador de bytes transmitidos durante a fase de transferência de dados.

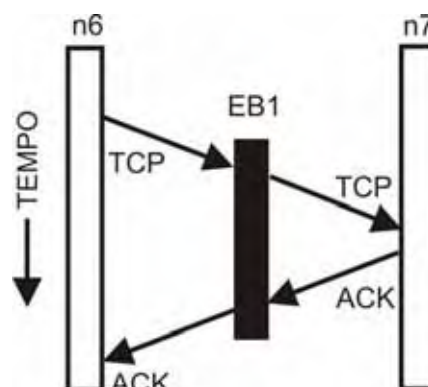


Figura 5.8 - Estabelecimento de conexão TCP, entre os nós sem fio, por meio da EB1

Observa-se na Figura 5.9 uma vazão homogênea dos pacotes transmitidos entre as estações base. O comportamento é coerente pelas condições do tráfego das cinco EBs serem idênticas.

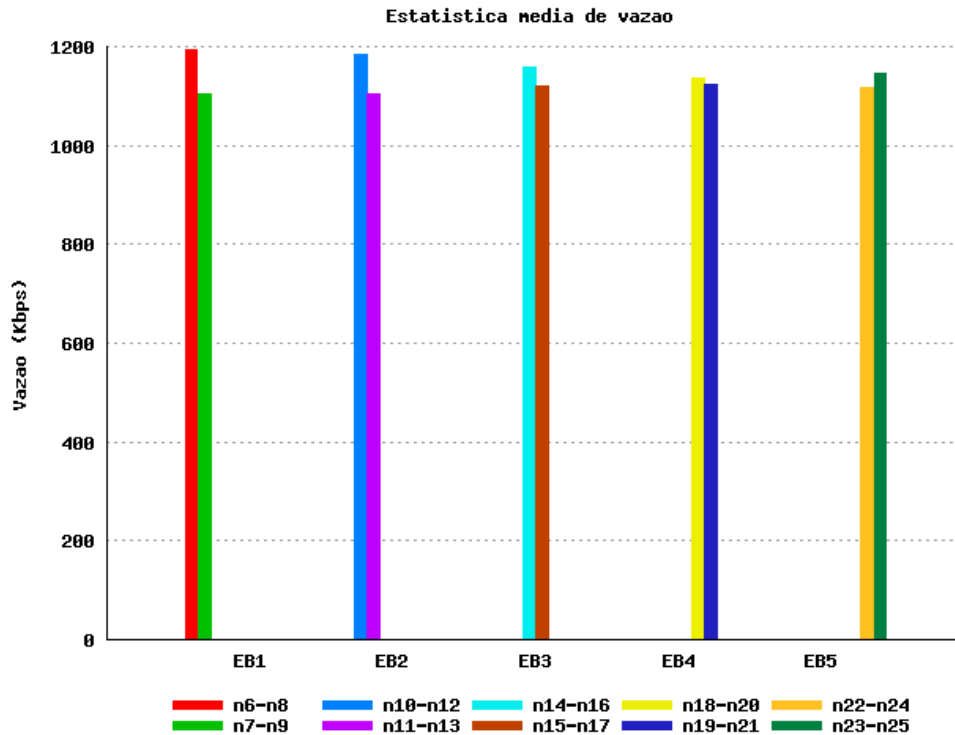


Figura 5.9 - Estatística média de vazão do roteamento de pacotes, feito pelos nós sem fio pertencentes à área de cobertura de sua respectiva estação base.

#### 5.1.3.1.2 Segunda Análise

Na segunda análise, é observada a vazão média dos pacotes transmitidos dos nós sem fio até as estações base. Estas, por sua vez, roteam os pacotes até o nó fixo W(0). O arquivo “wireless\_wpgcps\_2.tcl” apresenta o *script* de simulação da rede, conforme detalhado no Apêndice E (PRETE; SHINODA, 2010d).

A Figura 5.10 ilustra o estabelecimento de conexão TCP, entre o nó sem fio n6 e o nó fixo W(0), sendo roteado pela EB1, como exemplo. O nó sem fio n6 inicia a conexão, enviando um pacote TCP e espera-se que o nó fixo W(0) aceite a conexão, enviando um pacote ACK de confirmação. Como todas as mensagens dos nós sem fio são destinadas para W(0) (gateway), é provável que haja uma degradação do desempenho do sistema em relação à primeira análise.

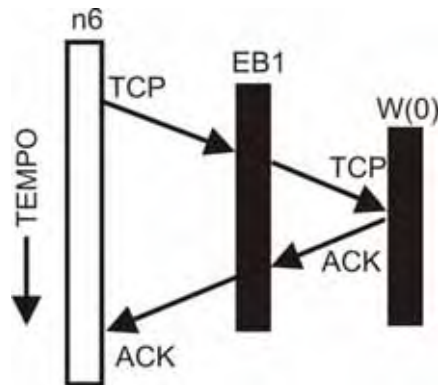


Figura 5.10 - Estabelecimento de conexão TCP, entre o nó sem fio n6 e o nó fixo W(0), sendo roteado pela EB1.

A Figura 5.11 ilustra a modelagem do comportamento dos nós sem fio em que ocorre uma degradação do desempenho de vazão. Exceto nas EB4 e EB5, todas as demais estações base apresentaram um perfil semelhante.

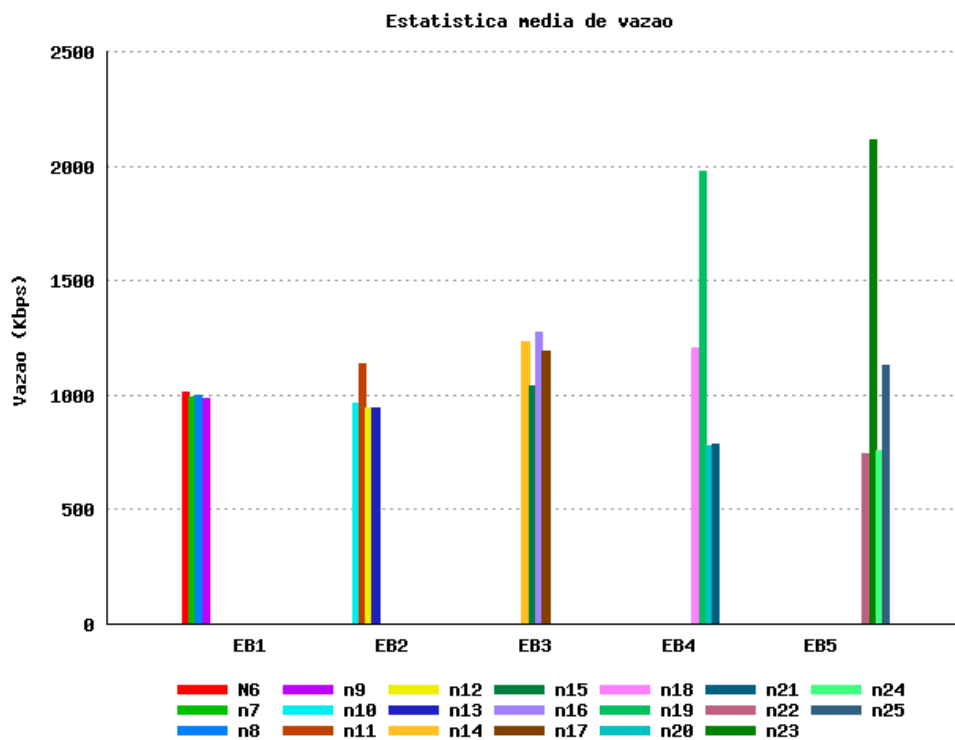


Figura 5.11 - Estatística média de vazão do roteamento de pacotes transmitidos dos nós sem fio até as estações base e estas, roteiam os pacotes até o nó fixo W(0).



### 5.1.3.1.3 Terceira Análise

Na terceira análise, é analisada a vazão média do roteamento de pacotes (bytes) feita pelos nós sem fio de diferentes estações base. O arquivo “wireless\_wpgcps\_3.tcl” apresenta o *script* de simulação da rede, conforme detalhado no Apêndice F (PRETE; SHINODA, 2010d).

A Figura 5.12 ilustra o estabelecimento de conexão TCP, entre o nó sem fio n6 e o nó sem fio n11, roteados por meio da EB1, EB2 e W(0), como exemplo. O nó sem fio n6 inicia a conexão, enviando um pacote TCP que é roteado pela EB1, W(0) e EB2 e espera-se que o pacote seja enviado até o nó sem fio n11. Este, por sua vez, envia um pacote ACK de confirmação.

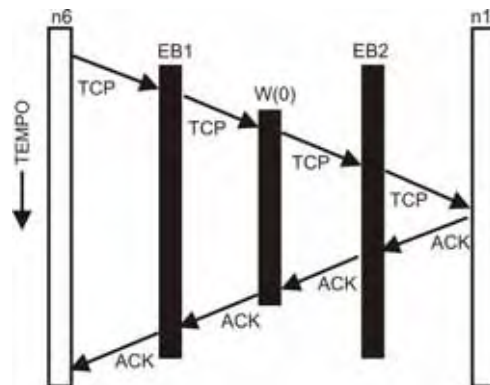


Figura 5.12 - Estabelecimento de conexão TCP, entre o nó sem fio n6 e o nó sem fio n11, roteados por meio da EB1, EB2 e W(0).

Observa-se na Figura 5.13 um desequilíbrio da vazão nos fluxos analisados, em função do alto tráfego gerado pela rede, principalmente pela necessidade de roteamento entre duas estações base e um nó fixo.

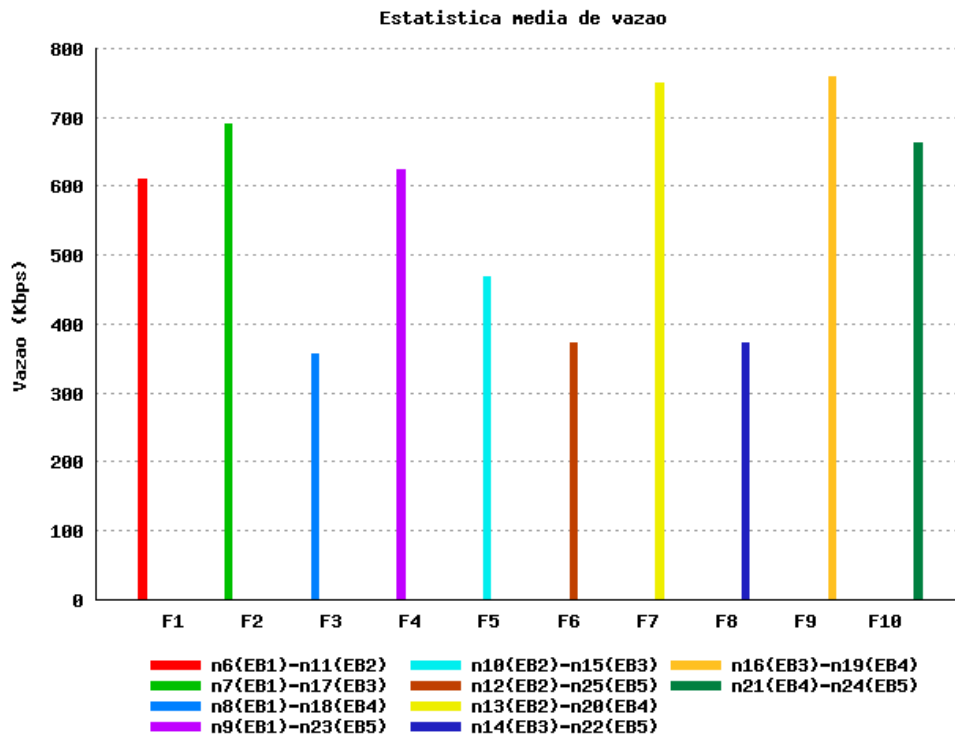


Figura 5.13 - Estatística média de vazão do roteamento de pacotes feito pelos nós sem fio de diferentes estações base.

### 5.1.3.2 Conclusões

Os resultados mostraram que a utilização do protocolo de transporte TCP e o protocolo de roteamento DSDV conduziram os pacotes com uma vazão média diversificada nas três análises. Na primeira análise, a vazão média dos pacotes apresentou-se quase estável, ocasionando poucas oscilações ao longo da simulação isto devido à vazão ser analisada somente na estação base correspondente ao nó sem fio. Já na segunda análise, onde é considerado o desempenho da vazão média dos pacotes de todos os nós sem fio até o nó fixo W(0) houve uma degradação da vazão para alguns nós, devido ao grande número de nós sem fio na rede, transmitindo e recebendo pacotes ocasionando congestionamento. Por fim, na terceira análise que foi avaliada, na vazão média dos pacotes, verificou-se, do mesmo modo, uma degradação na vazão em decorrência dos nós sem fio estarem se comunicando com os nós de outras estações base, provocando congestionamento e perda de vazão para alguns nós da rede.

Portanto, para a aplicação de transmissão de arquivos em uma intranet (FTP), é importante que todos os receptores recebam os arquivos com confiabilidade e sem erros. A

proposta de topologia da Figura 5.1, com os parâmetros da Tabela 5.2, evidenciaram resultados que preveem o desempenho da rede em alguns cenários do cotidiano.

## 5.2 Rede de Acesso Sem Fio e com Mobilidade Restrita

Neste subcapítulo, será abordado o estudo de um cenário proposto para análise do desempenho de uma rede infraestruturada na transmissão de dados em um ambiente externo, considerando uma avenida com uma distância de 2000 metros. As métricas de modelagem consideradas neste cenário foram a movimentação dos nós sem fio e a comparação média de vazão quando ocorre o aumento de nós móveis na rede (PRETE; SHINODA, 2010c).

### 5.2.1 Topologia da Rede

A Figura 5.14 mostra o cenário implementado para avaliar o comportamento da rede em função da mobilidade e vazão média na transmissão de dados em um ambiente externo. A topologia infraestruturada é composta por um nó cabeado  $W(0)$  conectado a seis estações base (EB1, EB2, EB3, EB4, EB5 e EB6), com uma distância de aproximadamente 2000 m. A distância entre os nós sem fio na área de contorno da EB é de 200 m (PRETE; SHINODA, 2010c).

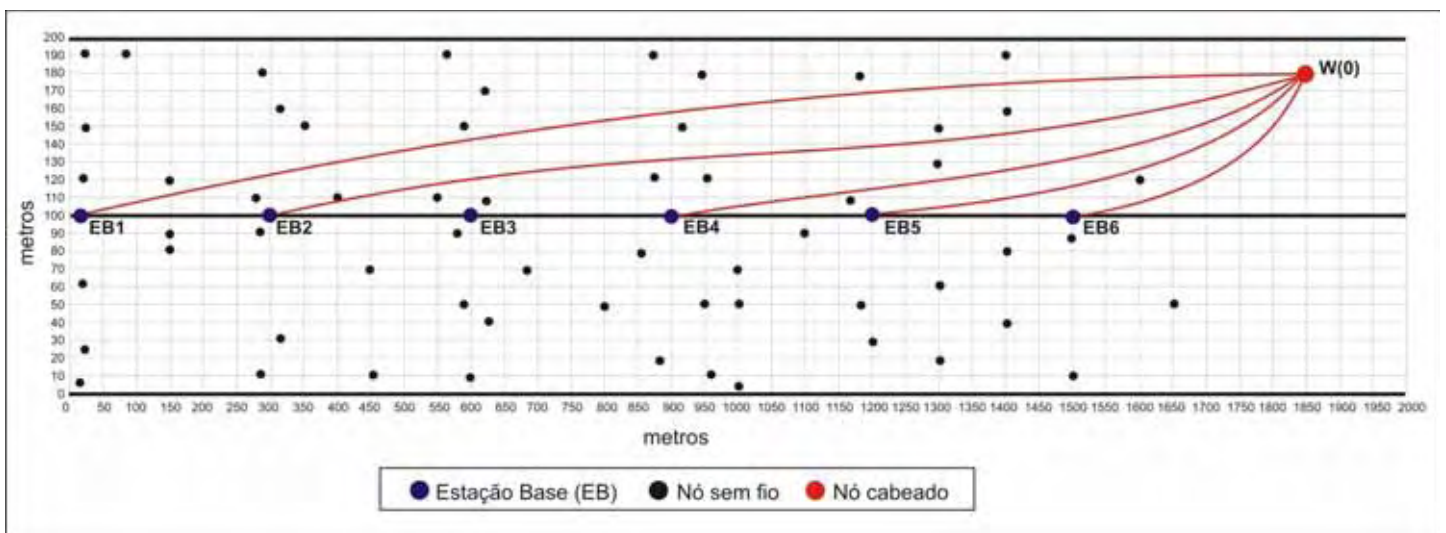


Figura 5.14 - Topologia da rede com mobilidade restrita do usuário

Na Figura 5.14, são criados 7 domínios, 1 domínio para rede cabeada (W0: 0.0.0) e 6 domínios para rede sem fio (EB1: 1.0.0, EB2: 2.0.0, EB3: 3.0.0, EB4: 4.0.0, EB5: 5.0.0 e EB6: 6.0.0).

A Tabela 5.3 apresenta os principais parâmetros do modelo de rede considerado na simulação de transmissão de dados em um ambiente externo.

Tabela 5.3 - Parâmetros da rede considerado na simulação de transmissão de dados em um ambiente externo.

<b>Parâmetro</b>	<b>Valor</b>
Canal	WirelessChannel
Propagação	TwoRayGround
Interface de rede	Wireless Phy
Camada MAC	802_11
Tipo de fila	DropTail
Camada de enlace	LL
Modelo da antenna	OmniAntenna
Número máximo de pacotes na fila	50
Número de nós sem fio	60
Protocolo de roteamento	DSDV
Área de cobertura (XxY)	2000 m x 200 m
Número de nós cabeado	1
Número de estações base	6
Taxa de transmissão	11 Mb
Tamanho dos pacotes	1023
Tipo de aplicação	FTP

Na Figura 5.14, aplicando os parâmetros da Tabela 5.3, temos dois agentes definidos. Um agente TCP anexado ao nó W(0) e outro agente de recepção (Sink), vinculado aos nós sem fio, que empregam também o protocolo de roteamento sem fio DSDV. Além disso, a aplicação associada entre o nó W(0) e os nós sem fio é o FTP.

Com o intuito de auxiliar a realização de simulações com várias configurações de posição e movimento entre nós em uma rede sem fio, foi empregado o programa “setdest”, descrito no subcapítulo 4.5.1.

### 5.2.2 *Script de Simulação*

Para a topologia definida na Figura 5.14 e empregando os parâmetros da Tabela 5.3, foi desenvolvido o arquivo “wireless\_dincon\_60n.tcl” que apresenta o *script* de simulação da rede para 60 nós sem fio, conforme detalhado no Apêndice G (PRETE; SHINODA, 2010c). Outros arquivos foram criados, baseados no arquivo “wireless\_dincon\_60.tcl”, para obter resultados de simulações com 6 nós, 12 nós, 18 nós, 24 nós, 30 nós, 36 nós, 42 nós, 48 nós e 54 nós.

No *script* de simulação “wireless\_dincon\_60n.tcl”, foi carregado o arquivo “setdest\_60n” (Apêndice H). Este arquivo traz todo o cenário de mobilidade dos nós sem fio em função do tempo, gerado pelo software “setdest”. A Tabela 5.4 apresenta os parâmetros para gerar o cenário de mobilidade para 60 nós.

Tabela 5.4 - Criação do cenário de mobilidade com o setdest

<pre>setdest -v 1 -x 2000 -y 200 -n 60 -p 0 -M 20 -t 250 &gt; setdest_60n</pre>
---

Outros cenários de mobilidade foram criados para 6 nós, 12 nós, 18 nós, 24 nós, 30 nós, 36 nós, 42 nós, 48 nós e 54 nós. Os parâmetros foram baseados na Tabela 5.4, alterando somente o campo `-n` (números de nós sem fio).

### 5.2.3 *Resultados*

Os resultados apresentados a seguir correspondem a duas análises em decorrência do desempenho no tráfego de pacotes em uma rede infraestruturada.

#### 5.2.3.1 *Primeira Análise*

Na primeira análise, é considerado o comportamento do nó sem fio N1 em função do tempo quando este se movimenta por uma avenida em um ambiente externo. Nota-se na Figura 5.15 o tempo em que N1 é roteado em cada estação base, conforme a topologia da Figura 5.14. O nó N1 passa a se comunicar com a estação base na seguinte escala de tempo: de 0 a 67s (EB1), de 67 a 96s (EB2), de 96 a 127s (EB3), de 127 a 153s (EB4), de 153 a 192s (EB5) e de 192 a 250s (EB6). Nas EB1 e EB6, o tempo de comunicação é maior, devido às estações base se localizarem nas extremidades da avenida, onde o nó N1 possui uma área

maior de comunicação, já que, antes da EB1 e depois da EB6, não há outras estações base para comunicação. Nas demais estações base, o tempo de comunicação torna-se menor, devido à curta distância entre elas, cerca de 150 metros para troca de roteamento (PRETE; SHINODA, 2010c).

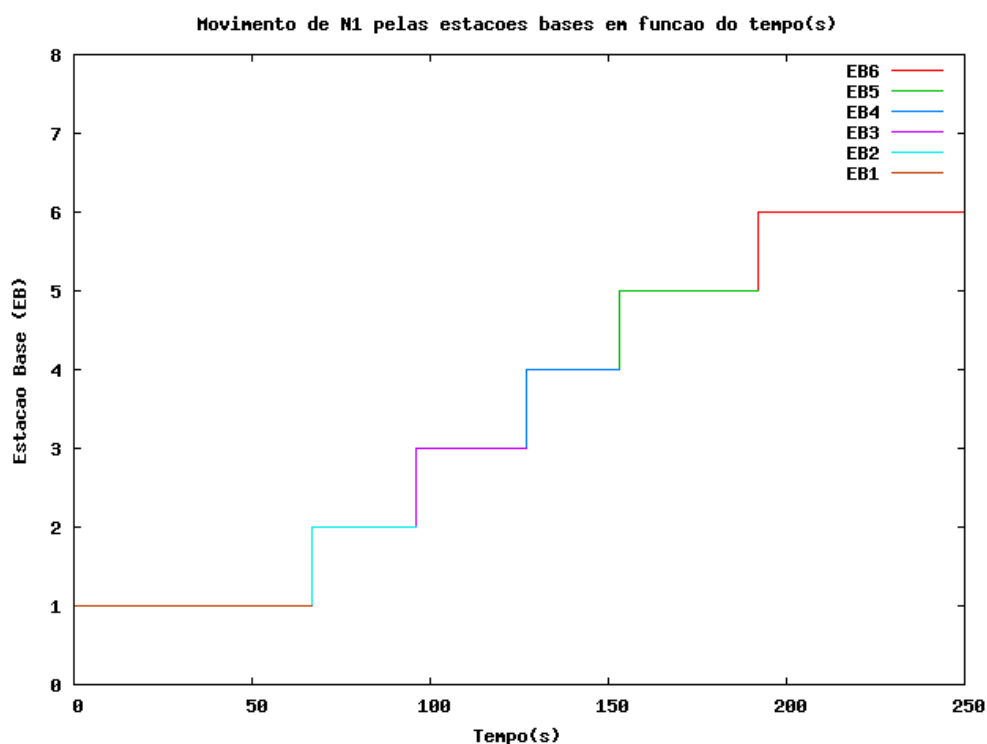


Figura 5.15 - Movimento do nó sem fio N1 em função do tempo quando este se movimenta por uma avenida em um ambiente externo.

### 5.2.3.2 Segunda Análise

Na segunda análise, foram considerados dois aspectos da simulação. No primeiro, é avaliada a quantidade de nós sem fio conectados às estações base em função do tempo, quando estes se movimentam por uma avenida em um ambiente externo, conforme mostrado na Figura 5.16. No final da simulação, observou-se o posicionamento dos nós sem fio com acúmulo gradativo de roteamento pela EB2 (8 nós sem fio), EB3 (14 nós sem fio) e, principalmente, EB4 (17 nós sem fio) (PRETE; SHINODA, 2010c).

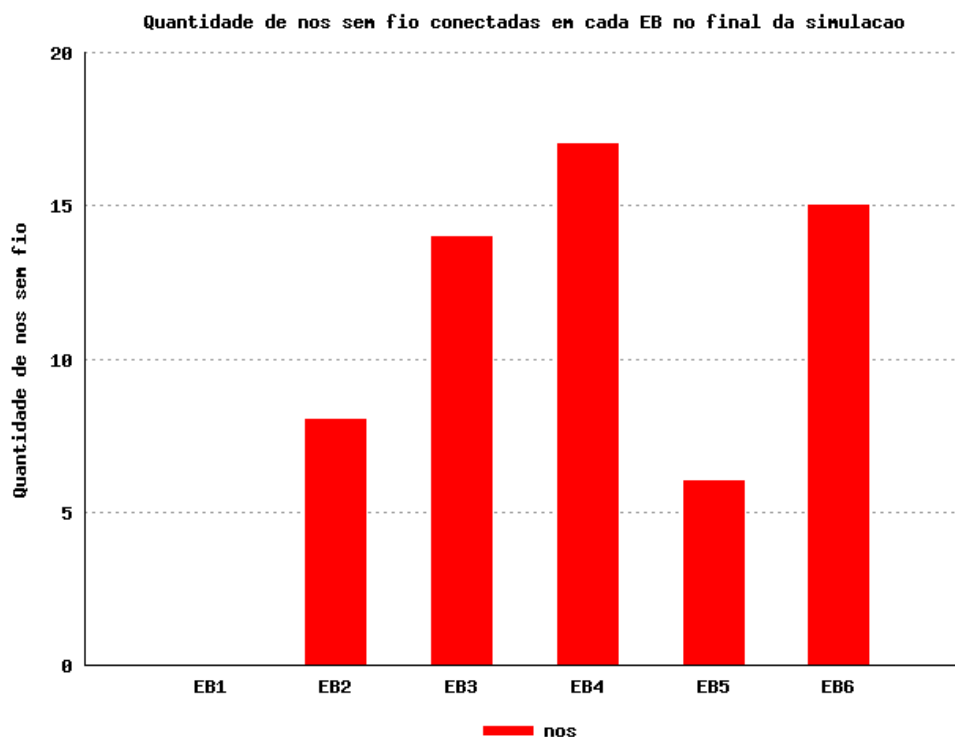


Figura 5.16 - Distribuição dos nós sem fio conectados às estações base em função do tempo

A movimentação dos nós sem fio gerado pelo programa *setdest*, usa o algoritmo *Random Waypoint* (Ponto de Mudança de Rota) onde o nó fica parado em um local por um determinado tempo e logo se desloca para um novo local selecionado aleatoriamente com uma velocidade uniformemente distribuída. Quando o destino é alcançado, o nó para novamente por um tempo determinado e repete o processo selecionando um novo destino e velocidade. Quando chega ao término da simulação, é registrada no arquivo *trace* a última posição do nó nas coordenadas x e y do terreno percorrido, onde é verificada qual estação base é responsável pela área em seu torno de roteamento.

Com essa característica de funcionamento do programa *setdest*, os nós que estavam na EB1 se movimentaram todos para a EB2, não ficando nenhum nó sem fio se comunicando com a EB1. Uma boa parte dos nós que estavam na EB2 se movimentaram para a EB3. Vários nós da EB3 se movimentaram para a EB4 e poucos nós da EB4 se deslocaram para a EB5, onde se registrou na EB4 o maior número de nós conectados no final da simulação. Vários nós que estavam na EB5 se deslocaram para a EB6 e poucos nós que estavam em torno da EB6 se deslocaram para outras EB, registrando um alto número de nós ao seu alcance.

No segundo aspecto da simulação é analisado o comportamento da rede, à medida que as mesmas escalonam, ou seja, quando se aumenta a quantidade de nós sem fio presentes e,

proporcionalmente, a quantidade de nós comunicantes, conforme mostrado na Figura 5.17. A comparação apresentada é em virtude do aumento normalizado de nós sem fio a cada simulação, ou seja, 6 nós (10%), 12 nós (20%), 18 nós (30%), 24 nós (40%), 30 nós (50%), 36 nós (60%), 42 nós (70%), 48 nós (80%), 54 nós (90%) e 60 nós (100%) (PRETE; SHINODA, 2010c).

Como todas as mensagens dos nós sem fio são destinadas para W(0) (gateway), há uma degradação no desempenho do sistema quando se aumenta o número de nós sem fio na rede, ocasionando perda de vazão e, em alguns momentos, descarte de pacotes.

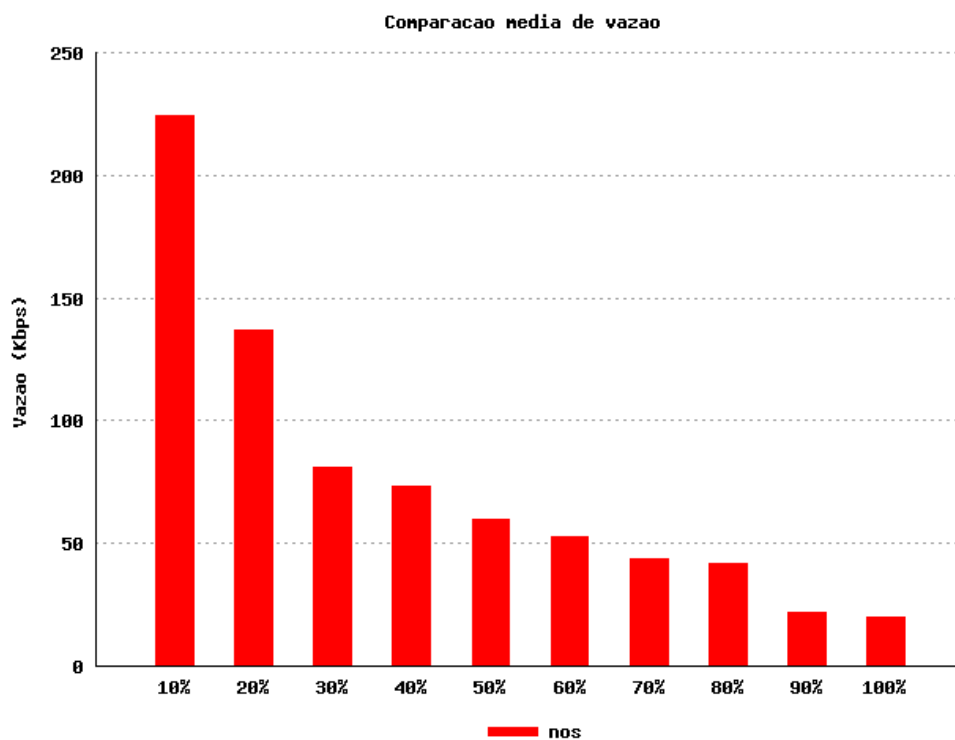


Figura 5.17 - Comparação média de vazão à medida que as mesmas escalonam

#### 5.2.4 Conclusão

A topologia apresentada na Figura 5.14 mostrou um estudo de duas análises de mobilidade, ambas desenvolvidas para redes infraestruturadas.

Os resultados das análises feitas, em relação às métricas estudadas, mostram uma degradação da eficiência da rede com o escalonamento do número de nós, devido à necessidade de roteamento e aumento da interferência com o crescimento do número de transmissores.



### 5.3 Rede de Acesso Sem Fio e com Mobilidade Irrestrita

Neste subcapítulo, será viabilizada por meio de simulações, a avaliação de desempenho da vazão média de pacotes e as possíveis razões de descarte, quando usuários se movimentam e trocam dados em um ambiente externo, usando rede infraestruturada. As estações base que roteam os dados, estão situadas em várias quadras em uma área de 500 x 500 metros (PRETE; SHINODA, 2010a).

#### 5.3.1 Topologia de Rede

A Figura 5.18 apresenta o cenário implementado para avaliar o comportamento da rede em função da mobilidade e vazão média na transmissão de dados em um ambiente externo. A topologia infraestruturada é composta por um nó cabeado  $W(0)$ , conectado a nove estações base (EB1, EB2, EB3, EB4, EB5, EB6, EB7, EB8 e EB9). A distância entre os nós sem fio, na área de contorno da EB, é de 120 metros (PRETE; SHINODA, 2010a).

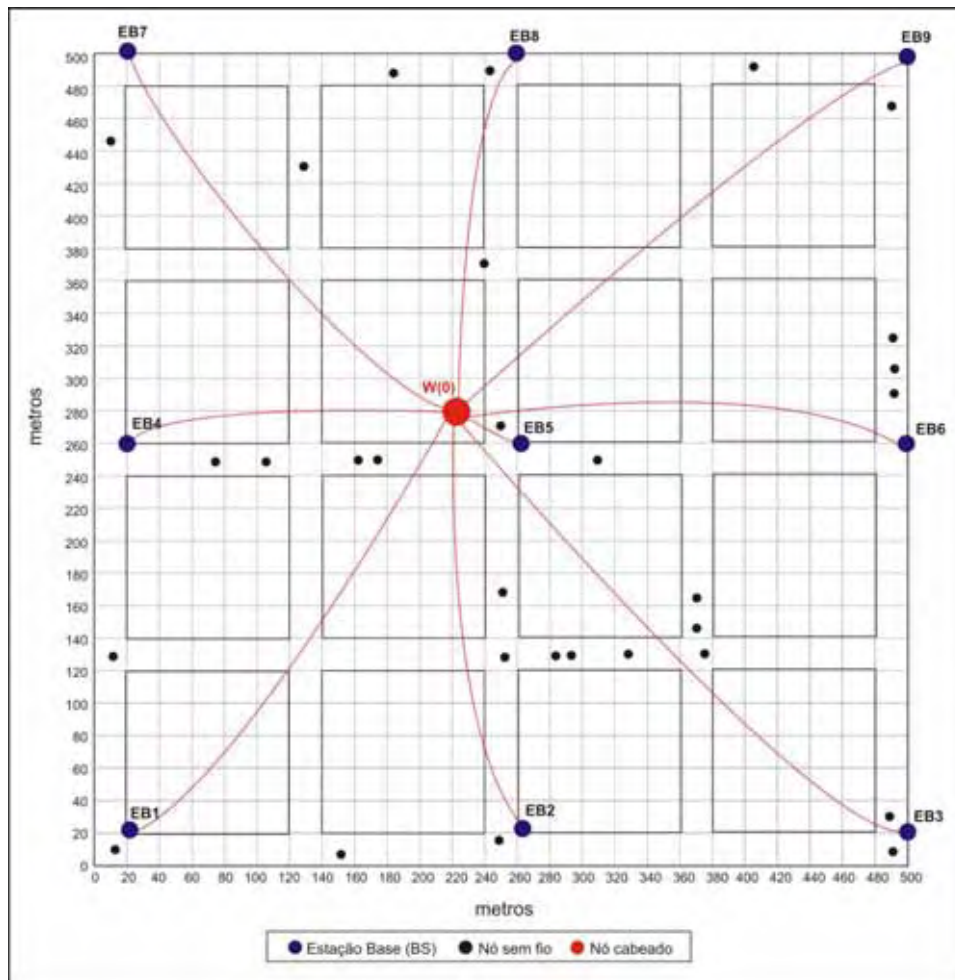


Figura 5.18 - Topologia da rede com mobilidade irrestrita do usuário

A Tabela 5.5 apresenta os principais parâmetros do modelo de rede considerada na simulação de transmissão de dados em um ambiente externo (PRETE; SHINODA, 2010a).

Tabela 5.5 - Parâmetros da rede considerada na simulação de transmissão de dados em um ambiente externo

<b>Parâmetro</b>	<b>Valor</b>
Canal	WirelessChannel
Propagação	TwoRayGround
Interface de rede	Wireless Phy
Camada MAC	802_11
Tipo de fila	DropTail
Camada de enlace	LL
Modelo da antenna	OmniAntenna
Número máximo de pacotes na fila	50
Número de nós sem fio	30
Protocolo de roteamento	DSDV
Área de cobertura (XxY)	500 x 500 metros
Número de nós cabeado	1
Número de estações base	9
Taxa de transmissão	11 Mb
Tamanho dos pacotes	1023
Tipo de aplicação	FTP

No esquema mostrado na Figura 5.18, tem-se dois agentes definidos. Um agente TCP, anexado ao nó W(0) e outro agente de recepção (Sink), vinculado aos nós sem fio que empregam também o protocolo de roteamento sem fio DSDV. Além disso, a aplicação associada entre o nó W(0) e os nós sem fio é o FTP.

### 5.3.2 Script de Simulação

Para a topologia definida na Figura 5.18, foi desenvolvido o arquivo “wireless\_emc\_cilamce.tcl”, que apresenta o *script* de simulação da rede, conforme detalhado no Apêndice I (PRETE; SHINODA, 2010a). No *script* de simulação “wireless\_emc\_cilamce.tcl”, foi carregado o arquivo “scen\_30n”, sendo que este arquivo traz

todo o cenário de mobilidade, gerado pelo software BonnMotion. Este software foi desenvolvido pelo grupo de pesquisa *Institute of Computer Science IV*, da *University of Bonn*, na Alemanha (ASCHENBRUCK, 2009). Para gerar o cenário com o BonnMotion, foram usados os parâmetros listados na Tabela 5.6.

Tabela 5.6 - Criação do cenário de mobilidade com o BonnMotion

```
bm -f scen ManhattanGrid -n 30 -d 900 -x 500 -y 500 -R 2.0 -u 4.0 -v
4.0
```

Com o intuito de auxiliar a simulação com várias configurações de posição e movimento para 30 nós em uma rede sem fio, foi empregado o modelo de mobilidade ManhattanGrid, com duração de 900 segundos, em uma área de 500 x 500 metros. Este modelo representa uma área geográfica com um conglomerado de blocos de prédios divididos por ruas ou avenidas, sendo que, na Tabela 5.6, foi definido 4 x 4 quadras, com forma tipicamente quadrada ou retangular. Um nó sem fio é posicionado, a princípio, em um local escolhido aleatoriamente, em algum ponto, em uma das ruas. Este nó movimentar-se somente pelas ruas estabelecidas.

Após a execução da instrução listada na Tabela 5.6, foram gerados dois arquivos: o primeiro, “scen.params” (Apêndice J), contém os parâmetros detalhados da configuração usados na simulação. O segundo, “scen.movements.gz” (Apêndice K), apresenta a posição inicial dos nós e a sua movimentação em um determinado tempo. Os dois arquivos citados não são compatíveis com os parâmetros de configuração do NS e, para convertê-los, foram usados os parâmetros listados na Tabela 5.7:

Tabela 5.7 - Conversão de arquivos com a aplicação NSFile

```
bm NSFile -f scen
```

A aplicação NSFile (ASCHENBRUCK, 2009) foi empregada para gerar dois novos arquivos: “scen.ns\_params” (Apêndice L) e “scen.ns\_movements.gz” (Apêndice M). Esses arquivos possuem os formatos de variáveis e objetos usados pelo NS, e podem ser integrados em um arquivo “.tcl”, sendo carregados pelo comando *source*. Nas simulações deste subcapítulo, o arquivo “scen.ns\_movements.gz” foi integrado no arquivo “wireless\_emc\_cilamce.tcl”, com o nome alterado para “scen\_30n”.

### 5.3.3 Resultados

Os resultados apresentados correspondem à análise de vazão de pacotes e sua perda em decorrência do desempenho no tráfego de dados em uma rede infraestruturada.

A Figura 5.19 analisa o comportamento da rede, à medida que as mesmas escalonam, ou seja, quando se aumenta a quantidade de nós sem fio presentes e, proporcionalmente, a quantidade de nós comunicantes. A comparação apresentada é em virtude do aumento normalizado de nós sem fio a cada simulação, ou seja, 3 nós (10%), 6 nós (20%), 9 nós (30%), 12 nós (40%), 15 nós (50%), 18 nós (60%), 21 nós (70%), 24 nós (80%), 27 nós (90%) e 30 nós (100%). Como todas as mensagens dos nós sem fio são destinadas para W(0) (gateway), há uma degradação no desempenho do sistema quando se aumenta o número de nós sem fio na rede, ocasionando perda de vazão e, em alguns momentos, descarte de pacotes (PRETE; SHINODA, 2010a).

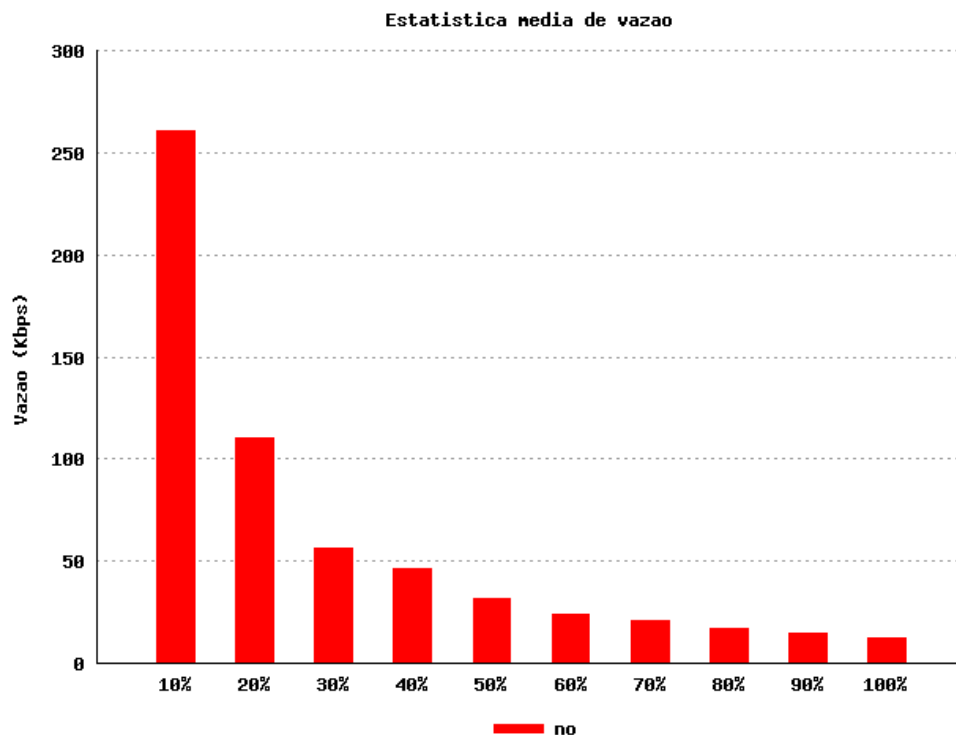


Figura 5.19 - Estatística média de vazão (com aumento normalizado de nós sem fio)

Na Figura 5.20, é analisada a influência da mobilidade no desempenho da vazão, o número de nós sem fio é 30. O que muda é a relação entre nós parados e móveis, por exemplo, no caso de 10% de mobilidade serão 3 nós móveis e 27 parados (PRETE; SHINODA, 2010b).

Observam-se na figura três faixas de desempenho em função da mobilidade: entre 10% e 30%, o desempenho é uniforme; de 40% a 70%, há uma leve degradação no desempenho e de 70% a 100%, ocorre uma degradação da vazão de pacotes, mas não chega a ser tão dramática (22,2% de degradação entre a mobilidade de 10% a 100%). Isso mostra que, no caso da topologia estudada, a mobilidade não é um fator crítico na degradação do sistema, existindo outras características mais relevantes para a perda de vazão.

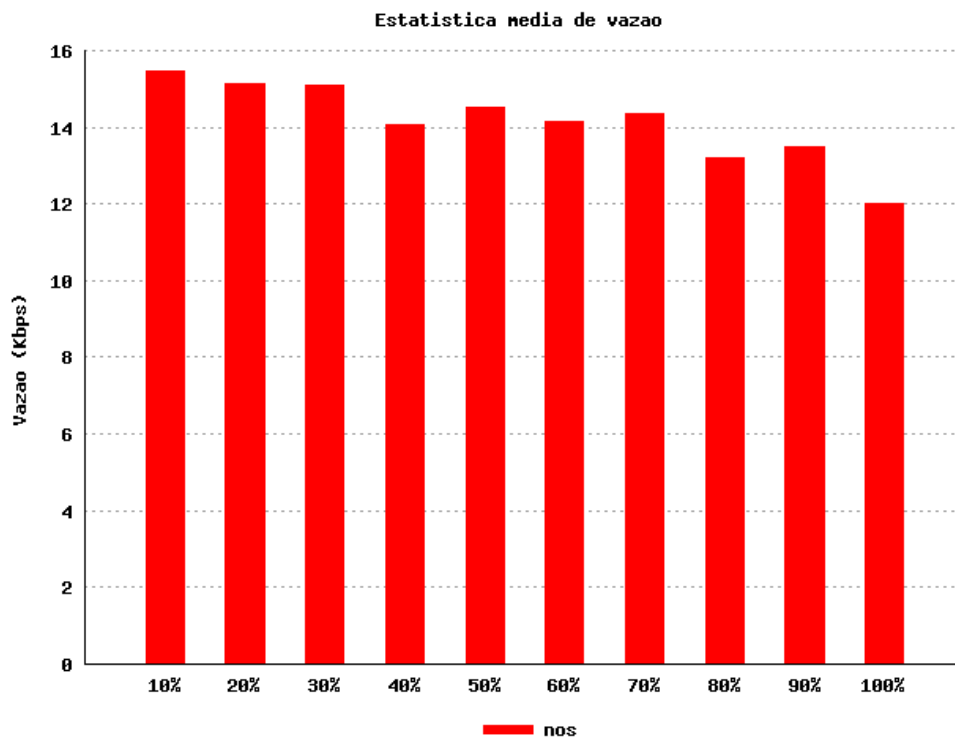


Figura 5.20 - Estatística média de vazão (com aumento normalizado de mobilidade)

As Figuras 5.21, 5.22 e 5.23 exibem os motivos de razão de descarte de pacotes coletados no arquivo *trace* da simulação, nas cargas mais críticas do sistema, exibido na Figura 5.19 (entre 10% a 30%). Analisou-se que a falha de roteamento dos nós sem fio com as estações de suporte, a mobilidade e o congestionamento no tráfego de pacotes foram as causas da perda de vazão apresentada na Figura 5.19 (PRETE; SHINODA, 2010a).

No arquivo *trace* da simulação (Apêndice N), o tipo de evento marcado com “d” (descarte de pacotes), o campo -NI (nível de rastreo), marcado com IFQ ou RTR e o campo -Nw (razões de descarte), marcado com END, TTL, IFQ, ARP ou outros, identificam os motivos da perda de vazão.

O nível de rastreo IFQ expressa fila cheia no *buffer* de transmissão com as seguintes razões de descarte: IFQ --- (descarte, devido à fila cheia), IFQ ARP (descarte, devido à falha

na resolução de endereços entre endereços IP), IFQ END (descarte no fim da simulação). O nível de rastreo RTR denota falha de roteamento com as seguintes razões de descarte: RTR CBK (nenhuma entrada de rota para o destino) e RTR TTL (tempo de vida do pacote alcançou zero antes de chegar ao destino).

A Figura 5.21 mostra os motivos de razão de descarte coletados no arquivo *trace* da simulação (Apêndice N.1), com 10% da carga de nós do sistema (3 nós móveis). Observam-se poucas diferenças entre as razões de descarte, sendo a mais relevante o descarte de pacotes devido à fila cheia no *buffer* de transmissão marcado com o nível de rastreo IFQ.

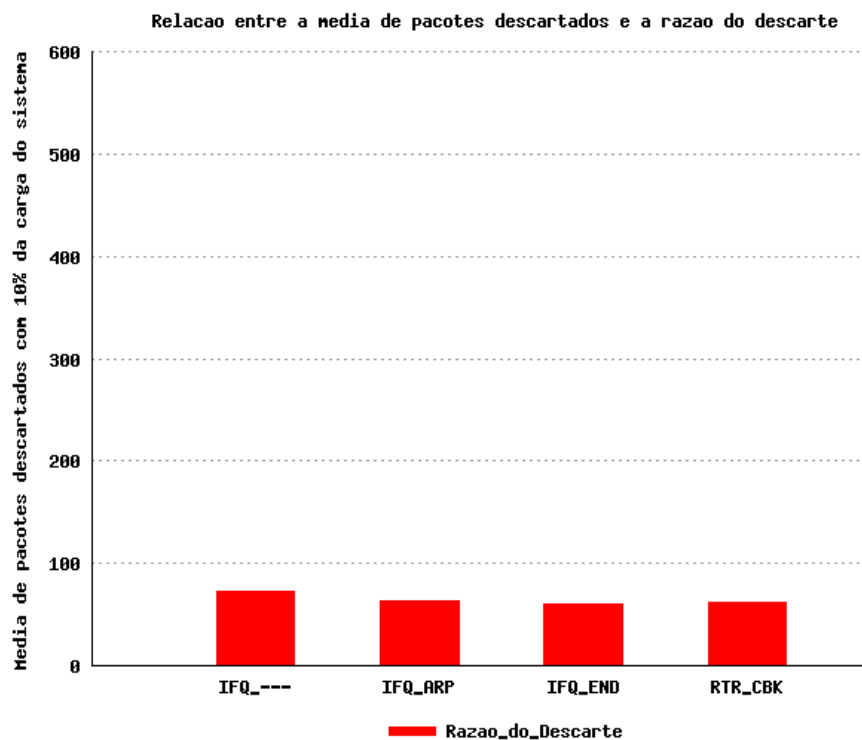


Figura 5.21 - Razão do descarte de pacotes com 10% da carga do sistema (3 nós sem fio)

A Figura 5.22 mostra os motivos de razão de descarte coletados no arquivo *trace* da simulação (Apêndice N.2), com 20% da carga de nós do sistema (6 nós móveis). Observa-se com ênfase duas razões de descarte. Caso não se encontre a resolução de endereço do IP (IFQ\_ARP), provavelmente, não haverá nenhuma entrada de rota para o destino (RTR\_CBK), causando o descarte de pacotes.

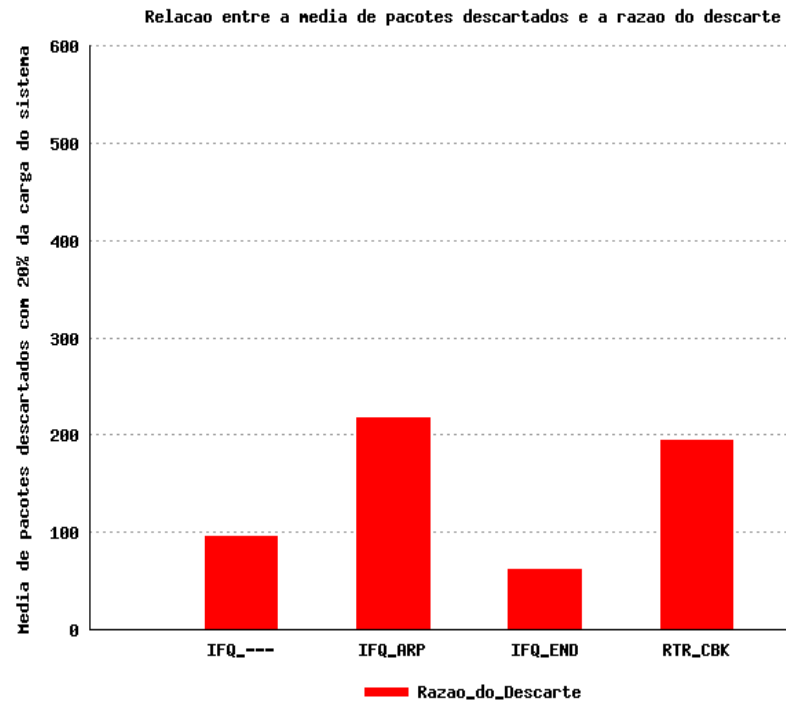


Figura 5.22 - Razão do descarte de pacotes com 20% da carga do sistema (6 nós sem fio)

A Figura 5.23 mostra os motivos de razão de descarte coletados no arquivo *trace* da simulação (Apêndice N.3), com 30% da carga de nós do sistema (9 nós móveis). Observou-se que o maior descarte de pacotes foi pela razão de tempo de vida do pacote, que alcançou o valor zero antes de chegar ao destino (RTR TTL).

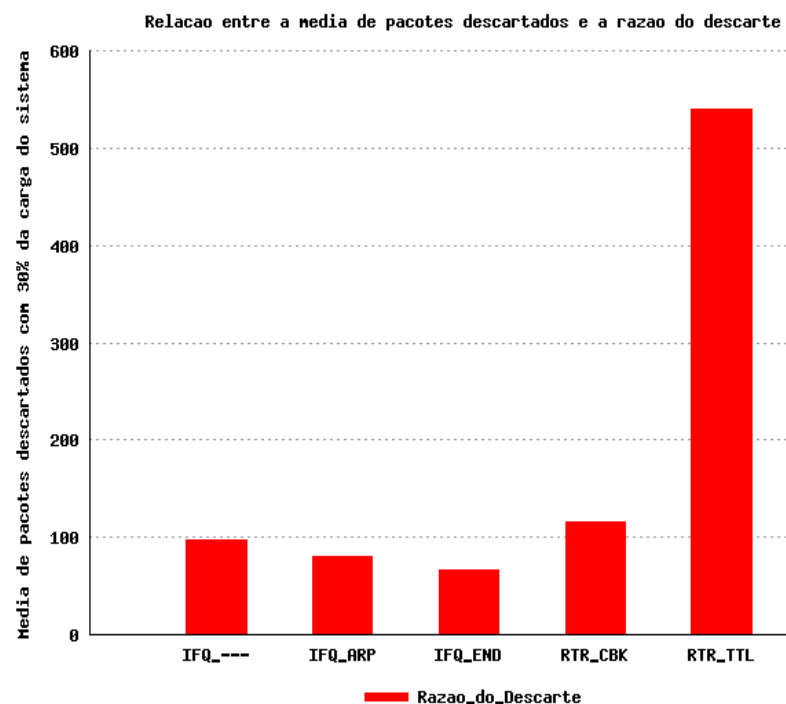


Figura 5.23 - Razão do descarte de pacotes com 30% da carga do sistema (9 nós sem fio)

A Figura 5.24 mostra a relação média de pacotes descartados entre 10% a 30% da carga mais crítica do sistema. Em 10%, ocasionou um descarte menor de pacotes, elevando a vazão de dados, proporcionalmente com o aumento de nós móveis no sistema. O descarte de pacotes foi maior, devido ao congestionamento de pacotes e à falha de roteamento com as estações de suporte à mobilidade (PRETE; SHINODA, 2010a).

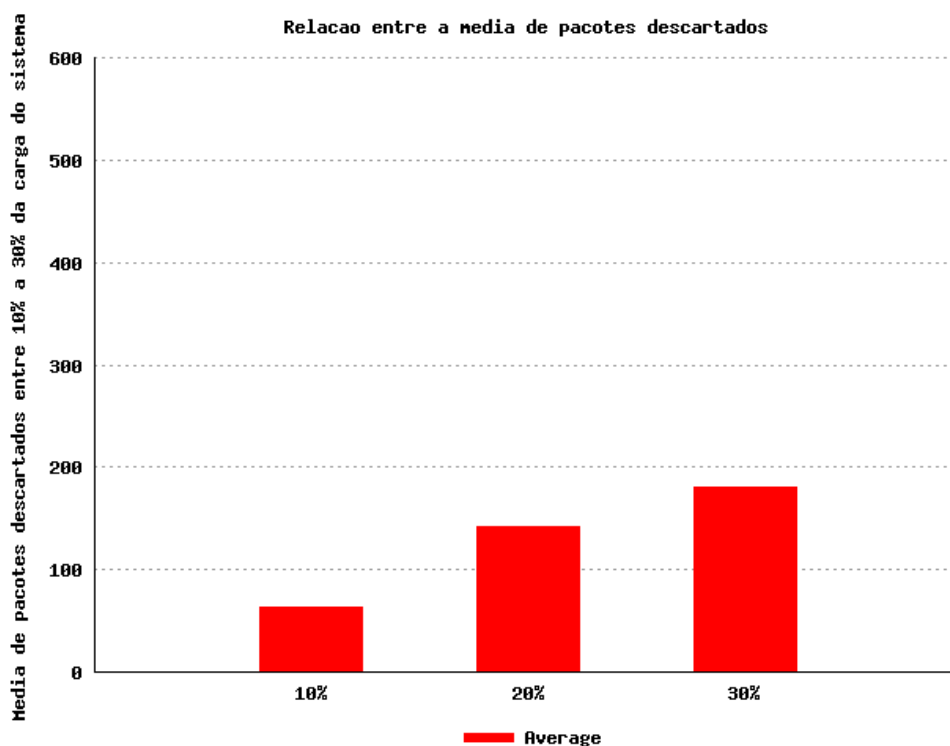


Figura 5.24 - Relação média de pacotes descartados entre 10% a 30% da carga do sistema

### 5.3.4 Conclusão

O resultado das análises mostrou claramente uma degradação da eficiência da rede com o escalonamento do número de nós, devido à necessidade de roteamento e o aumento da interferência com o crescimento do número de transmissores. Observou-se que a influência da mobilidade no desempenho da vazão, em relação à quantidade de nós parados e em movimento, não chega a ser tão dramática. Isso mostra que, no caso da topologia da Figura 5.18, a mobilidade não é um fator crítico na degradação do sistema.

Em relação aos motivos de razão de descarte de pacotes coletados no arquivo *trace* da simulação, nas cargas mais críticas do sistema, entre 10% (3 nós móveis) a 30% (9 nós móveis), o congestionamento marcado com IFQ, no campo -NI (nível de rastreo) do



arquivo *trace*, foi apontado como o maior problema de razão de descarte de pacotes, nas simulações com 10% e 20% da carga do sistema.

A relação média de pacotes descartados entre 10% a 30% da carga do sistema levou à conclusão de que o congestionamento de pacotes e, em alguns casos, a falha de roteamento com as estações base foram as causas mais relevantes de perda da vazão.

## CAPÍTULO 6

### CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

A simulação é uma importante ferramenta na pesquisa de redes, que permite modelar sistemas complexos e obter resultados para diversos cenários de redes propostos.

Existem diversos softwares que podem ser utilizados para simulação, entretanto, optou-se pela utilização do *Network Simulator*, por se tratar de uma ferramenta gratuita e de código aberto, que oferece suporte à simulação de tecnologias de redes com e sem fio. Outro ponto forte do NS é a contribuição de pesquisadores, principalmente do meio acadêmico, na atualização e acréscimo de novas bibliotecas.

A análise do escalonamento de redes infraestruturadas, proposta nesta dissertação, é relevante para aplicações dessas redes em ambientes diversos, quanto à necessidade crescente de comunicação que ocorre com a evolução natural da tecnologia, onde equipamentos, antes destinados a uma única finalidade, como aparelhos celulares para telecomunicação, são atualmente usados para os mais diversos fins, como acesso à Internet. Para que haja uma análise e um desempenho eficientes do sistema, sem a necessidade de uma infraestrutura física, é importante que sejam feitos testes, inicialmente por meio de simulações, como os que foram realizados, principalmente pelo custo e pela complexidade de gerenciamento serem menores que a realização de testes com equipamentos reais.

Este trabalho apresentou um estudo de mobilidade desenvolvido para redes infraestruturadas. Quando há necessidade de enviar dados para ambientes à longa distância, onde a comunicação é inviável, realizada somente por meio de rede cabeada, uma alternativa é utilizar ambientes de redes infraestruturadas (com e sem fio). Fundamentado neste estudo, pôde-se realizar uma avaliação de desempenho de vazão, percebendo-se que há variação significativa, de acordo com cada simulação.

Analisando os três tipos de redes de acesso sem fio, tratados neste trabalho, observou-se que o uso do protocolo de roteamento DSDV, associado aos protocolos de transporte (UDP ou TCP) e aplicação (CBR ou FTP) foi adequado para as topologias simuladas do Capítulo 5, isto devido à pouca quantidade de nós sem fio parados ou se movimentando. Neste caso, o DSDV inicia um processo periódico de atualização de rota, especialmente quando a topologia da rede muda. Se existir um aumento elevado de nós sem fio na rede, parados ou se deslocando, os protocolos mais adequados para serem empregados seriam o AODV, DSR ou

TORA, pelo fato de não manterem constante atualização de rota. Somente quando existir requisição de transmissão de dados, é feito um procedimento de manutenção de rota, conservando baixa taxa de tráfego na rede. Outra análise a ser considerada, referente aos três tipos de redes de acesso sem fio simuladas no Capítulo 5, é a vazão, devido a sua importância no desempenho de uma rede. Conclui-se que o aumento no tráfego de dados, do primeiro tipo de rede, o crescimento do número de nós sem fio transmissores, do segundo tipo de rede e a falha de roteamento e o congestionamento, do terceiro tipo de rede, foram as causas mais relevantes de degradação da rede, ocasionando atraso no envio e na recepção de dados, seja em um ambiente interno ou externo.

Como sugestão para trabalhos futuros, o prosseguimento inclui a análise de desempenho de redes com outros protocolos de roteamento em topologias com um elevado número de nós sem fio parados ou se movimentando. Outra proposta é a continuação do estudo das topologias de rede deste trabalho, substituindo o nó cabeado  $W(0)$  em sem fio. A análise dos resultados pode ser aprofundada com métricas, para medir o desempenho da rede com gráficos de atraso médio dos pacotes, variação do atraso, entre outros.

Neste trabalho, empregou-se o uso do NS, exclusivamente em redes Wi-Fi, padrão 802.11. Como sugestão para trabalhos posteriores, deve-se aplicar o NS em redes WiMAX (*Worldwide Interoperability for Microwave Access*), padrão 802.16. As redes banda larga sem fio, baseadas no padrão IEEE 802.16, apresentam algumas características particulares, tais como a capacidade de prover QoS, a oferta de altas taxas de dados e uma ampla área de cobertura num ambiente sem fio de rede metropolitana.

Um estudo mais avançado seria a simulação, por meio do NS, utilizando a rede WiMAX, juntamente com a tecnologia Smart Grid (rede inteligente), em um sistema de distribuição de energia elétrica. De acordo com (CLEVELAND, 2006), o termo “Smart Grid” refere-se à modernização do sistema de distribuição de energia elétrica, integrado aos sistemas de comunicação e à infraestrutura de rede automatizada, de modo que monitore, proteja e otimize automaticamente a operação dos seus elementos interligados.

No recente relatório da NIST (*National Institute of Standard and Technology*), algumas tecnologias de comunicação com e sem fio são identificados para Smart Grid (ARNOLD, 2009).

Sistemas avançados sem fio oferecem vantagens de produtos e instalações de baixo custo, implantação rápida, acesso generalizado, enquanto a comunicação celular, tecnologias com fio e até mesmo as antigas tecnologias sem fio, muitas vezes não podiam oferecer. No entanto, até recentemente as tecnologias sem fio tinham relativamente pouca aceitação pelas

indústrias, devido à baixa taxa de dados, relacionados com problemas de interferência, preocupações com a segurança, disponibilidade limitada de produtos, entre outros. Várias atividades já foram iniciadas para atender às questões técnicas e identificar as tecnologias sem fio adequadas, especialmente para Smart Grid. No último relatório do NIST, sobre o Smart Grid, uma das ações priorizadas é abordar a “interferência de comunicações no espectro de rádio sem licença”. O sucesso desta ação irá aliviar muitos problemas relacionados com comunicação sem fio no espectro de rádio sem licença, fornecendo canais dedicados à comunicação, para interoperações críticas da rede inteligente. Um tópico de pesquisa nessa linha poderia ser “rádio cognição”, um novo conceito em que o próprio sistema se encarrega de alocar as frequências com menores níveis de interferência no ambiente.

## REFERÊNCIAS

- ALTMAN, E.; JIMENEZ, T. **NS simulator for beginners**. Mérida: Universidad de Los Andes, 2003. 146 p.
- ARNOLD, G. **Smart grid interoperability standards roadmap**. Gaithersburg: National Institute of Standards and Technology, 2009.
- ASCHEBRUCK, N. **BonnMotion**: a mobility scenario generation and analysis tool. Alemanha: Institute of Computer Science IV, University of Bonn, 2009.
- BERNAL, P. S. **Comunicações móveis**: tecnologias e aplicações. São Paulo: Érica, 2002. 204 p.
- CALI F.; CONTI M.; GREGORI E. IEEE 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism. **IEEE Journal on Selected Areas in Communications**, New York, v. 18, p. 1774-1786, 2000.
- CHUNG, J.; CLAYPOOL, M. **NS by example**. Worcester: Worcester Polytechnic Institute, 1995. Disponível em: <<http://nile.wpi.edu/NS>>. Acesso em: 16 ago. 2010.
- CLEVELAND, F. Use of wireless data communications in power system operations. **PROCEEDINGS IEEE POWER SYSTEM CONFERENCE AND EXPOSITION**, 2006, Atlanta. **Proceedings...** New York: IEEE, 2006, p. 631-640.
- ERGEN, M. **IEEE 802.11**. Califórnia: Berkeley, 2002. 90 p.
- GREIS, M. **Tutorial for the Network Simulator "NS"**. California: University of Southern California, 1996. Disponível em: <<http://www.isi.edu/nsnam/ns/tutorial/index.html>>. Acesso em: 20 mar. 2010.
- HEIDEMANN, J. et al. Effects of detail in wireless network simulation. In: **SCS COMMUNICATION NETWORKS AND DISTRIBUTED SYSTEMS MODELING AND SIMULATION CONFERENCE**, 5., 2000, San Diego. **Proceedings...** San Diego: [s.n.], 2000.
- IEEE 802.11. **IEEE 802.11 wireless local area networks**. New York: IEEE, 2010. Disponível em: <<http://www.ieee802.org/11/index.shtml>>. Acesso em: 10 ago. 2010.
- ISI. **The network simulator NS-2**. California: University of Southern California, 1995. Disponível em: <<http://www.isi.edu/nsnam/ns/index.html>>. Acesso em: 24 fev. 2010.
- JAIN, R. **The art of computer systems performance analysis**: techniques for experimental design, measurement, simulation, and modeling. New York: John Wiley & Sons, 1991.
- JOHNSON, D. B. **The Rice university monarch project**: mobile networking architectures. Houston: Rice University, 2004.

JOHNSON, D. B.; MALTZ, D. A. Dynamic source routing in ad hoc networks. In: IMIELINSKI, T.; KORTH, H. F. (Eds.). **Mobile computing**. Boston: Kluwer Academic Publishers, 1996. Cap. 5, p. 153-181.

KUROSE, J. F.; ROSS K. W. **Redes de computadores e a internet: uma abordagem top-down**. 3. ed. São Paulo: Pearson Addison Wesley, 2006. 656 p.

MATTHEW, S. G. **802.11 wireless networks: the definitive guide**. 2. ed. Cambridge: O'Reilly Media, 2002.

MCKEAN, E. **The new oxford american dictionary**. 2. ed. New York: Oxford University Press, 2005.

PARK, V. D.; CORSON, M. S. A highly adaptative distributed routing algorithm for mobile wireless networks. In: IEEE COMPUTER AND COMMUNICATIONS SOCIETIES - INFOCOM '97, 1997, Kobe. **Proceedings...** New York: IEEE, 1997. v. 3, p. 1405-1413.

PERKINS, C. E.; BHAGWAT, P. **DSDV routing over a multihop wireless network of mobile computers**. Boston: Addison-Wesley Longman, 2000. p. 53-74. (Perkins-47190). Disponível em: <<http://ptgmedia.pearsoncmg.com/images/0201309769/samplechapter/perk03.pdf>>. Acesso em: 10 ago. 2010.

PERKINS, C. E.; BHAGWAT, P. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for mobile computers. In: IEEE CONFERENCE ON COMMUNICATIONS ARCHITECTURES, PROTOCOLS AND APPLICATIONS - ACM SIGCOMM '94, 1994, London. **Proceedings...** New York: IEEE, 1994. p. 234-244.

PERKINS, C. E.; ROYER, E. Ad-hoc On-Demand Distance Vector Routing. In: IEEE WORKSHOP MOBILE COMPUTERS SYSTEM AND APPLICATIONS, 2., 1999, New Orleans. **Proceedings...** New York: IEEE, 1999. p. 90-100.

PRETE, L. R; SHINODA, A. A. Análise do comportamento das variações do protocolo TCP. In: CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL, 32., 2009, Cuiabá. **Anais...** Cuiabá: [s.n.], 2009a.

PRETE, L. R; SHINODA, A. A. Analysis reasons of disposal in the routing of mobile infrastructured network. In: CONGRESS MECOM DEL BICENTENARIO 2010; ARGENTINIAM CONGRESS ON COMPUTATIONAL MECHANICS, 9.; IBERIAN-LATIN-AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGINEERING, 31.; SOUTH AMERICAN CONGRESS ON COMPUTATIONAL MECHANICS, 2., 2010, Buenos Aires. **Proceedings...** Buenos Aires: [s.n.], 2010a.

PRETE, L. R; SHINODA, A. A. Avaliação de desempenho na mobilidade de usuários e seu impacto no roteamento das redes móveis infraestruturadas. In: ENCONTRO DE MODELAGEM COMPUTACIONAL, 13., 2010, Nova Friburgo. **Anais...** Nova Friburgo: [s.n.], 2010b.

PRETE, L. R; SHINODA, A. A. Comportamento do escalonamento de redes infraestruturadas através de métodos computacionais. In: BRAZILIAN CONFERENCE ON DYNAMICS, CONTROL AND THEIR APPLICATIONS, 9., 2010, Serra Negra. **Proceedings...** Serra Negra: [s.n.], 2010c.

PRETE, L. R; SHINODA, A. A. Desempenho de redes com monitoração de tráfego através de descarte aleatória antecipada de pacotes. In: CONGRESSO DE COMPUTAÇÃO DO SUL DE MATO GROSSO, 4., 2008, Rondonópolis. **Anais...** Rondonópolis: [s.n.], 2008.

PRETE, L. R; SHINODA, A. A. Desempenho de redes de ponto de acesso sem fio através de medidas de vazão. In: SIMPÓSIO REGIONAL DE MATEMÁTICA E SUAS APLICAÇÕES DE ILHA SOLTEIRA, 2., 2009, Ilha Solteira. **Anais...** Ilha Solteira: [s.n.], 2009b.

PRETE, L. R; SHINODA, A. A. O avanço da tecnologia de comunicação com o uso de simulador para a modelagem em redes mistas cabeadas e sem fio. In: WORKSHOP DE PÓS-GRADUAÇÃO E PESQUISA DO CENTRO PAULA SOUZA, 5., 2010, São Paulo. **Anais...** São Paulo: [s.n.], 2010d.

ROYER, E.; TOH, C. A review of current routing protocols for ad hoc mobile wireless networks. In: IEEE COMPUTER AND COMMUNICATIONS SOCIETIES, 1999, New York. **Proceedings...** New York: IEEE, 1999. v. 6, p. 46-55.

SIQUEIRA, E. **Três momentos da história das telecomunicações no Brasil**. São Paulo: Dezembro Editorial, 1997. 108 p.

SOARES, L. F. G. **Redes de computadores: das LANs, MANs e WANs às redes ATM**. Rio de Janeiro: Campus, 1995.

TANENBAUM, A. S. **Redes de computadores**. 4. ed. Rio de Janeiro: Campus, 2003. 945 p.

VINT. **Virtual InterNetwork Testbed**. California: University of Southern California, 1996. Disponível em: <<http://www.isi.edu/nsnam/vint/index.html>>. Acesso em: 11 set. 2009.

## APÊNDICES

### Apêndice A - Arquivo avgStats.awk

avgStats.awk

```
BEGIN {
    recvdSize = 0
    startTime = 1e6
    stopTime = 0
}

{
    # Trace line format: normal
    if ($2 != "-t") {
        event = $1
        time = $2
        if (event == "+" || event == "-") node_id = $3
        if (event == "r" || event == "d") node_id = $4
        flow_id = $8
        pkt_id = $12
        pkt_size = $6
        flow_t = $5
        level = "AGT"
    }

    # Trace line format: new
    if ($2 == "-t") {
        event = $1
        time = $3
        node_id = $5
        flow_id = $39
        pkt_id = $41
        pkt_size = $37
        flow_t = $45
        level = $19
    }

    # Store packets send time
    if (level == "AGT" && flow_id == flow && node_id == src &&
```



```

sendTime[pkt_id] == 0 && (event == "+" || event == "s") &&
  pkt_size >= pkt) {
  if (time < startTime) {
    startTime = time
  }
  sendTime[pkt_id] = time
  this_flow = flow_t
}

# Update total received packets' size and store packets arrival time
if (level == "AGT" && flow_id == flow && node_id == dst &&
  event == "r" && pkt_size >= pkt) {
  if (time > stopTime) {
    stopTime = time
  }
  # Rip off the header
  hdr_size = pkt_size % pkt
  pkt_size -= hdr_size
  # Store received packet's size
  recvdSize += pkt_size
  # Store packet's reception time
  recvTime[pkt_id] = time
}
}

END {
  # Compute average delay
  delay = avg_delay = recvdNum = 0
  for (i in recvTime) {
    if (sendTime[i] == 0) {
      printf("\nError in delay.awk: receiving a packet that
        wasn't sent %g\n",i)
    }
    delay += recvTime[i] - sendTime[i]
    recvdNum ++
  }
  if (recvdNum != 0) {
    avg_delay = delay / recvdNum
  } else {
    avg_delay = 0
  }
}

```

```

# Compute average jitters
jitter1 = jitter2 = jitter3 = jitter4 = jitter5 = 0
prev_time = delay = prev_delay = processed = deviation = 0
prev_delay = -1
for (i=0; processed<recvdNum; i++) {
    if(recvTime[i] != 0) {
        if(prev_time != 0) {
            delay = recvTime[i] - prev_time
            e2eDelay = recvTime[i] - sendTime[i]
            if(delay < 0) delay = 0
            if(prev_delay != -1) {
                jitter1 += abs(e2eDelay - prev_e2eDelay)
                jitter2 += abs(delay-prev_delay)
                jitter3 += (abs(e2eDelay-prev_e2eDelay) -
                    jitter3) / 16
                jitter4 += (abs(delay-prev_delay) - jitter4) /
                    16
            }
            prev_delay = delay
            prev_e2eDelay = e2eDelay
        }
        prev_time = recvTime[i]
        processed++
    }
}
if (recvdNum != 0) {
    jitter1 = jitter1*1000/recvdNum
    jitter2 = jitter2*1000/recvdNum
}

# Output
if (recvdNum == 0) {

    printf("#####\n" \
#####\n" \
"# Warning: no packets were received, simulation may be too
short #\n" \
"#####\n\n")
}
printf("\n")

```

```

printf(" %15s: %g\n", "flowID", flow)
printf(" %15s: %s\n", "flowType", this_flow)
printf(" %15s: %d\n", "srcNode", src)
printf(" %15s: %d\n", "destNode", dst)
printf(" %15s: %d\n", "startTime", startTime)
printf(" %15s: %d\n", "stopTime", stopTime)
printf(" %15s: %g\n", "receivedPkts", recvdNum)
printf(" %15s: %g\n", "avgTput[kbps]", (recvdSize/(stopTime-
      startTime))*(8/1000))

printf(" %15s: %g\n", "avgDelay[ms]", avg_delay*1000)
printf(" %15s: %g\n", "avgJitter1[ms]", jitter1)
printf(" %15s: %g\n", "avgJitter2[ms]", jitter2)
printf(" %15s: %g\n", "avgJitter3[ms]", jitter3*1000)
printf(" %15s: %g\n", "avgJitter4[ms]", jitter4*1000)
}

function abs(value) {
    if (value < 0) value = 0-value
    return value
}

```

Instruções de uso do arquivo `avgStats.awk` no prompt de comando:

```
awk -f avgStats.awk <values> <tracefile>
```

Onde:

<values>: `src` - identificação do nó de origem  
`dst` - identificação do nó de destino  
`flow` - identificação do fluxo  
`pkt` - tamanho (bytes) dos pacotes no fluxo de dados gerados na simulação

<tracefile>: nome do arquivo *trace* a ser analisado.

Exemplo:

```
awk -f avgStats.awk src=0 dst=10 flow=0 pkt=1000 wireless.tr
```

Estatística do resultado obtido no seguinte formato:

```
    flowID:    0
    flowType:  tcp
    srcNode:   0
    destNode:  10
    startTime: 100
    stopTime:  116
    receivedPkts: 1331
    avgTput[kbps]: 396.05
    avgDelay[ms]: 142.693
    avgJitter1[ms]: 7.96962
    avgJitter2[ms]: 2.45535
    avgJitter3[ms]: 7.87847
    avgJitter4[ms]: 2.40291
```

Para gerar o gráficos que mostram a vazão média da rede, foram usadas as informações do campo `avgTput [Kbps]`.

## Apêndice B - Arquivo wireless\_srmais.tcl

wireless\_srmais.tcl

```

set val(chan) Channel/WirelessChannel ;# Channel Type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
set val(nn) 20 ;# number of mobilenodes
set val(rp) DSDV
set val(x) 15e3
set val(y) 15e3
set num_wired_nodes 1
set num_BS 5
set temp_BS {1.0.0 2.0.0 3.0.0 4.0.0 5.0.0} ;# hierarchical addresses BS
set temp_x {5.0e3 7.0e3 5.0e3 3.0e3 5.0e3} ;# BS x position
set temp_y {5.0e3 5.0e3 7.0e3 5.0e3 3.0e3} ;# BS y position
set temp_no_ind {0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4}
set temp_no_ad {1.0.1 1.0.2 1.0.3 1.0.4 \
                2.0.1 2.0.2 2.0.3 2.0.4 \
                3.0.1 3.0.2 3.0.3 3.0.4 \
                4.0.1 4.0.2 4.0.3 4.0.4 \
                5.0.1 5.0.2 5.0.3 5.0.4}
set temp_no_x {5.2e3 5.0e3 4.8e3 5.0e3 \
               7.2e3 7.0e3 6.8e3 7.0e3 \
               5.2e3 5.0e3 4.8e3 5.0e3 \
               3.2e3 3.0e3 2.8e3 3.0e3 \
               5.2e3 5.0e3 4.8e3 5.0e3}
set temp_no_y {5.0e3 5.2e3 5.0e3 4.8e3 \
               5.0e3 5.2e3 5.0e3 4.8e3 \
               7.0e3 7.2e3 7.0e3 6.8e3 \
               5.0e3 5.2e3 5.0e3 4.8e3 \
               3.0e3 3.2e3 3.0e3 2.8e3}

Mac/802_11 set dataRate_ 11Mb

Phy/WirelessPhy set Pt_ 2

```

```

# Initialize Global Variables
set ns_ [new Simulator]

# set up for hierarchical routing
$ns_ node-config -addressType hierarchical

AddrParams set domain_num_ 6          ;# number of domains
lappend cluster_num 1 1 1 1 1 1      ;# number of clusters in each domain
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel 1 1 1 1 1 1      ;# number of nodes in each cluster
AddrParams set nodes_num_ $eilastlevel ;# of each domain

$ns_ use-newtrace
set tracefd [open out.tr w]
$ns_ trace-all $tracefd

#create wired nodes
set temp {0.0.0}                      ;# hierarchical addresses
for {set i 0} {$i < $num_wired_nodes} {incr i} {
    set W($i) [$ns_ node 0.0.0]
}

# set up topography object
set topo [new Topography]

$topo load_flatgrid $val(x) $val(y)

# Create God
create-god $val(nn)

# Create channel
set chan_1_ [new $val(chan)]

$ns_ node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \

```

```

        -phyType $val(netif) \
        -wiredRouting ON \
        -topoInstance $topo \
        -agentTrace ON \
        -routerTrace ON \
        -macTrace ON \
        -movementTrace ON \
        -channel $chan_1_

for {set i 0} {$i < $num_BS} {incr i} {
    set BS($i) [$ns_ node [lindex $temp_BS $i]]
    $BS($i) random-motion 0                ;# disable random motion
    $BS($i) set X_ [lindex $temp_x [expr $i]]
    $BS($i) set Y_ [lindex $temp_y [expr $i]]
    $BS($i) set Z_ 0.0
}

$ns_ node-config -wiredRouting OFF

for {set i 0} {$i < $val(nn)} {incr i} {
    set node($i) [$ns_ node [lindex $temp_no_ad $i]]

    $node($i) base-station [AddrParams addr2id [$BS([lindex $temp_no_ind
[lexpr $i]]) node-addr]]

    $node($i) random-motion 0                ;# disable random motion
    $node($i) set X_ [lindex $temp_no_x [expr $i]]
    $node($i) set Y_ [lindex $temp_no_y [expr $i]]
    $node($i) set Z_ 0.0
}

#create links between wired and BS nodes
for {set i 0} {$i < $num_BS} {incr i} {
    $ns_ duplex-link $W(0) $BS($i) 2Mb 2ms DropTail
}

Application/Traffic/CBR set packetSize_ 1023
Application/Traffic/CBR set rate_ 512Kb

for {set i 0} {$i < $val(nn)} {incr i} {
    set udpFix($i) [new Agent/UDP]

```

```
$ns_ attach-agent $W(0) $udpFix($i)
set cbrFix($i) [new Application/Traffic/CBR]
$cbrFix($i) attach-agent $udpFix($i)
set node_rx($i) [new Agent/Null]
$ns_ attach-agent $node($i) $node_rx($i)
$ns_ connect $udpFix($i) $node_rx($i)
$ns_ at 1.0 "$cbrFix($i) start"
}
$ns_ at 20.0 "stop"
$ns_ at 20.0 "puts \"NS EXITING...\" ; $ns_ halt"

proc stop {} {
    global ns_ tracefd
    $ns_ flush-trace
    close $tracefd
    exit 0
}

puts "Starting Simulation..."
$ns_ run
```



**Apêndice C - Arquivo instantThroughput.awk**

instantThroughput.awk

```
BEGIN {
    recv = 0
    currTime = prevTime = 0
    printf("# %10s %10s %5s %5s %15s %18s\n\n", \
        "flow", "flowType", "src", "dst", "time", "throughput")
}
{
    # Trace line format: normal
    if ($2 != "-t") {
        event = $1
        time = $2
        if (event == "+" || event == "-") node_id = $3
        if (event == "r" || event == "d") node_id = $4
        flow_id = $8
        pkt_id = $12
        pkt_size = $6
        flow_t = $5
        level = "AGT"
    }
    # Trace line format: new
    if ($2 == "-t") {
        event = $1
        time = $3
        node_id = $5
        flow_id = $39
        pkt_id = $41
        pkt_size = $37
        flow_t = $45
        level = $19
    }

    # Init prevTime to the first packet recv time
    if(prevTime == 0)
        prevTime = time

    # Calculate total received packets' size
    if (level == "AGT" && flow_id == flow && node_id == dst &&
```

```

event == "r" && pkt_size >= pkt) {
    # Rip off the header
    hdr_size = pkt_size % pkt
    pkt_size -= hdr_size
    # Store received packet's size
    recv += pkt_size
    # This 'if' is introduced to obtain clearer
    # plots from the output of this script
    if((time - prevTime) >= tic*10) {
        printf(" %10g %10s %5d %5d %15g %18g\n", \
            flow,flow_t,src,dst,(prevTime+1.0),0)
        printf(" %10g %10s %5d %5d %15g %18g\n", \
            flow,flow_t,src,dst,(time-1.0),0)
    }
    currTime += (time - prevTime)
    if (currTime >= tic) {
        printf(" %10g %10s %5d %5d %15g %18g\n", \
            flow,flow_t,src,dst,time,(recv/currTime)*(8/1000))
        recv = 0
        currTime = 0
    }
    prevTime = time
}

}

END {
    printf("\n\n")
}

```

Instruções de uso do arquivo `instantThroughput.awk` no prompt de comando:

```
awk instantThroughput.awk <values> <tracefile>
```

Onde:

<values>: `tic` - tempo de amostragem (segundos)  
`src` - identificação de fluxo do nó de origem  
`dst` - identificação de fluxo do nó de destino

`flow` - identificação do fluxo

`pkt` - tamanho (bytes) dos pacotes de dados gerado pelo fluxo da aplicação

`<tracefile>`: nome do arquivo *trace* a ser analisado.

**Exemplo:**

```
awk -f instantThroughput.awk tic=1.0 src=0 dst=6 flow=0 pkt=1000
wireless_srmais.tr > stats.out
```

Estatística do resultado obtido no seguinte formato:

#	flow	flowType	src	dst	time	throughput
	0	cbr	0	6	1.6209	4.95184
	0	cbr	0	6	2.62867	492.174
	0	cbr	0	6	3.64337	488.816
	0	cbr	0	6	4.6579	488.898
	0	cbr	0	6	5.66442	500.735
	0	cbr	0	6	6.67175	500.331
	0	cbr	0	6	7.67769	501.023
	0	cbr	0	6	8.68449	500.596
	0	cbr	0	6	9.69106	500.712
	0	cbr	0	6	10.698	500.551
	0	cbr	0	6	11.7047	500.636
	0	cbr	0	6	12.7109	500.901
	0	cbr	0	6	13.722	498.452
	0	cbr	0	6	14.7283	500.859
	0	cbr	0	6	15.7349	500.693
	0	cbr	0	6	16.7419	500.501
	0	cbr	0	6	17.7481	500.882
	0	cbr	0	6	18.7552	500.457
	0	cbr	0	6	19.7619	500.629

Para gerar os gráficos que mostram a vazão instantânea da rede, foram usadas as colunas “time” e “throughput”.

## Apêndice D - Arquivo wireless\_wpgcps\_1.tcl

wireless\_wpgcps\_1.tcl

```

set val(chan) Channel/WirelessChannel ;#Channel Type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
set val(nn) 20 ;# number of mobilenodes
set val(rp) DSDV
set val(x) 15e3
set val(y) 15e3
set num_wired_nodes 1
set num_BS 5
set temp_BS {1.0.0 2.0.0 3.0.0 4.0.0 5.0.0} ;# hierarchical addresses BS
set temp_x {5.0e3 7.0e3 5.0e3 3.0e3 5.0e3} ;# BS x position
set temp_y {5.0e3 5.0e3 7.0e3 5.0e3 3.0e3} ;# BS y position
set temp_no_ind {0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4}
set temp_no_ad {1.0.1 1.0.2 1.0.3 1.0.4 \
                2.0.1 2.0.2 2.0.3 2.0.4 \
                3.0.1 3.0.2 3.0.3 3.0.4 \
                4.0.1 4.0.2 4.0.3 4.0.4 \
                5.0.1 5.0.2 5.0.3 5.0.4}
set temp_no_x {5.2e3 5.0e3 4.8e3 5.0e3 \
               7.2e3 7.0e3 6.8e3 7.0e3 \
               5.2e3 5.0e3 4.8e3 5.0e3 \
               3.2e3 3.0e3 2.8e3 3.0e3 \
               5.2e3 5.0e3 4.8e3 5.0e3}
set temp_no_y {5.0e3 5.2e3 5.0e3 4.8e3 \
               5.0e3 5.2e3 5.0e3 4.8e3 \
               7.0e3 7.2e3 7.0e3 6.8e3 \
               5.0e3 5.2e3 5.0e3 4.8e3 \
               3.0e3 3.2e3 3.0e3 2.8e3}

Mac/802_11 set dataRate_ 11Mb

Phy/WirelessPhy set Pt_ 2

```

```

# Initialize Global Variables
set ns_          [new Simulator]

# set up for hierarchical routing
$ns_ node-config -addressType hierarchical

AddrParams set domain_num_ 6          ;# number of domains
lappend cluster_num 1 1 1 1 1 1      ;# number of clusters in each domain
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel 1 1 1 1 1 1      ;# number of nodes in each cluster
AddrParams set nodes_num_ $eilastlevel ;# of each domain

$ns_ use-newtrace
set tracefd      [open out.tr w]
$ns_ trace-all $tracefd

#create wired nodes
set temp {0.0.0}          ;# hierarchical addresses
for {set i 0} {$i < $num_wired_nodes} {incr i} {
    set W($i) [$ns_ node 0.0.0]
}

# set up topography object
set topo          [new Topography]

$topo load_flatgrid $val(x) $val(y)

# Create God
create-god $val(nn)

# Create channel
set chan_1_ [new $val(chan)]

$ns_ node-config -adhocRouting $val(rp) \
                -llType $val(ll) \
                -macType $val(mac) \
                -ifqType $val(ifq) \
                -ifqLen $val(ifqlen) \
                -antType $val(ant) \
                -propType $val(prop) \

```

```

        -phyType $val(netif) \
        -wiredRouting ON \
        -topoInstance $topo \
        -agentTrace ON \
        -routerTrace ON \
        -macTrace ON \
        -movementTrace ON \
        -channel $chan_1_

for {set i 0} {$i < $num_BS} {incr i} {
    set BS($i) [$ns_ node [lindex $temp_BS $i]]
    $BS($i) random-motion 0                ;# disable random motion
    $BS($i) set X_ [lindex $temp_x [expr $i]]
    $BS($i) set Y_ [lindex $temp_y [expr $i]]
    $BS($i) set Z_ 0.0
}

$ns_ node-config -wiredRouting OFF

for {set i 0} {$i < $val(nn)} {incr i} {
    set node($i) [$ns_ node [lindex $temp_no_ad $i]]

    $node($i) base-station [AddrParams addr2id [$BS([lindex $temp_no_ind
[lexpr $i]]) node-addr]]

    $node($i) random-motion 0                ;# disable random motion
    $node($i) set X_ [lindex $temp_no_x [expr $i]]
    $node($i) set Y_ [lindex $temp_no_y [expr $i]]
    $node($i) set Z_ 0.0
}

#create links between wired and BS nodes
for {set i 0} {$i < $num_BS} {incr i} {
    $ns_ duplex-link $W(0) $BS($i) 2Mb 2ms DropTail
}

Application/FTP set packetSize_ 1040
Application/FTP set rate_ 512Kb

# create connection between nos
set tcp1 [new Agent/TCP/Newreno]

```

```
$tcp1 set class_1 ;#identificacao do fluxo de pacotes
set sink1 [new Agent/TCPSink]
$ns_ attach-agent $node(0) $tcp1
$ns_ attach-agent $node(2) $sink1
$ns_ connect $tcp1 $sink1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns_ at 1.0 "$ftp1 start"

set tcp2 [new Agent/TCP/Newreno]
$tcp2 set class_2 ;#identificacao do fluxo de pacotes
set sink2 [new Agent/TCPSink]
$ns_ attach-agent $node(1) $tcp2
$ns_ attach-agent $node(3) $sink2
$ns_ connect $tcp2 $sink2
set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2
$ns_ at 1.0 "$ftp2 start"

set tcp3 [new Agent/TCP/Newreno]
$tcp3 set class_3 ;#identificacao do fluxo de pacotes
set sink3 [new Agent/TCPSink]
$ns_ attach-agent $node(4) $tcp3
$ns_ attach-agent $node(6) $sink3
$ns_ connect $tcp3 $sink3
set ftp3 [new Application/FTP]
$ftp3 attach-agent $tcp3
$ns_ at 1.0 "$ftp3 start"

set tcp4 [new Agent/TCP/Newreno]
$tcp4 set class_4 ;#identificacao do fluxo de pacotes
set sink4 [new Agent/TCPSink]
$ns_ attach-agent $node(5) $tcp4
$ns_ attach-agent $node(7) $sink4
$ns_ connect $tcp4 $sink4
set ftp4 [new Application/FTP]
$ftp4 attach-agent $tcp4
$ns_ at 1.0 "$ftp4 start"

set tcp5 [new Agent/TCP/Newreno]
$tcp5 set class_5 ;#identificacao do fluxo de pacotes
```

```
set sink5 [new Agent/TCPSink]
$ns_ attach-agent $node(8) $tcp5
$ns_ attach-agent $node(10) $sink5
$ns_ connect $tcp5 $sink5
set ftp5 [new Application/FTP]
$ftp5 attach-agent $tcp5
$ns_ at 1.0 "$ftp5 start"

set tcp6 [new Agent/TCP/Newreno]
$tcp6 set class_ 6 ;#identificacao do fluxo de pacotes
set sink6 [new Agent/TCPSink]
$ns_ attach-agent $node(9) $tcp6
$ns_ attach-agent $node(11) $sink6
$ns_ connect $tcp6 $sink6
set ftp6 [new Application/FTP]
$ftp6 attach-agent $tcp6
$ns_ at 1.0 "$ftp6 start"

set tcp7 [new Agent/TCP/Newreno]
$tcp7 set class_ 7 ;#identificacao do fluxo de pacotes
set sink7 [new Agent/TCPSink]
$ns_ attach-agent $node(12) $tcp7
$ns_ attach-agent $node(14) $sink7
$ns_ connect $tcp7 $sink7
set ftp7 [new Application/FTP]
$ftp7 attach-agent $tcp7
$ns_ at 1.0 "$ftp7 start"

set tcp8 [new Agent/TCP/Newreno]
$tcp8 set class_ 8 ;#identificacao do fluxo de pacotes
set sink8 [new Agent/TCPSink]
$ns_ attach-agent $node(13) $tcp8
$ns_ attach-agent $node(15) $sink8
$ns_ connect $tcp8 $sink8
set ftp8 [new Application/FTP]
$ftp8 attach-agent $tcp8
$ns_ at 1.0 "$ftp8 start"

set tcp9 [new Agent/TCP/Newreno]
$tcp9 set class_ 9 ;#identificacao do fluxo de pacotes
set sink9 [new Agent/TCPSink]
```



```
$ns_ attach-agent $node(16) $tcp9
$ns_ attach-agent $node(18) $sink9
$ns_ connect $tcp9 $sink9
set ftp9 [new Application/FTP]
$ftp9 attach-agent $tcp9
$ns_ at 1.0 "$ftp9 start"

set tcp10 [new Agent/TCP/Newreno]
$tcp10 set class_ 10 ;#identificacao do fluxo de pacotes
set sink10 [new Agent/TCPSink]
$ns_ attach-agent $node(17) $tcp10
$ns_ attach-agent $node(19) $sink10
$ns_ connect $tcp10 $sink10
set ftp10 [new Application/FTP]
$ftp10 attach-agent $tcp10
$ns_ at 1.0 "$ftp10 start"

$ns_ at 20.0 "stop"
$ns_ at 20.0 "puts \"NS EXITING...\" ; $ns_ halt"

proc stop {} {
    global ns_ tracefd
    $ns_ flush-trace
    close $tracefd
    exit 0
}

puts "Starting Simulation..."
$ns_ run
```

## Apêndice E - Arquivo wireless\_wpgcps\_2.tcl

wireless\_wpgcps\_2.tcl

```

set val(chan) Channel/WirelessChannel ;#Channel Type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
set val(nn) 20 ;# number of mobilenodes
set val(rp) DSDV
set val(x) 15e3
set val(y) 15e3
set num_wired_nodes 1
set num_BS 5
set temp_BS {1.0.0 2.0.0 3.0.0 4.0.0 5.0.0} ;# hierarchical addresses BS
set temp_x {5.0e3 7.0e3 5.0e3 3.0e3 5.0e3} ;# BS x position
set temp_y {5.0e3 5.0e3 7.0e3 5.0e3 3.0e3} ;# BS y position
set temp_no_ind {0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4}
set temp_no_ad {1.0.1 1.0.2 1.0.3 1.0.4 \
                2.0.1 2.0.2 2.0.3 2.0.4 \
                3.0.1 3.0.2 3.0.3 3.0.4 \
                4.0.1 4.0.2 4.0.3 4.0.4 \
                5.0.1 5.0.2 5.0.3 5.0.4}
set temp_no_x {5.2e3 5.0e3 4.8e3 5.0e3 \
              7.2e3 7.0e3 6.8e3 7.0e3 \
              5.2e3 5.0e3 4.8e3 5.0e3 \
              3.2e3 3.0e3 2.8e3 3.0e3 \
              5.2e3 5.0e3 4.8e3 5.0e3}
set temp_no_y {5.0e3 5.2e3 5.0e3 4.8e3 \
              5.0e3 5.2e3 5.0e3 4.8e3 \
              7.0e3 7.2e3 7.0e3 6.8e3 \
              5.0e3 5.2e3 5.0e3 4.8e3 \
              3.0e3 3.2e3 3.0e3 2.8e3}

Mac/802_11 set dataRate_ 11Mb

Phy/WirelessPhy set Pt_ 2

```

```

# Initialize Global Variables
set ns_          [new Simulator]

# set up for hierarchical routing
$ns_ node-config -addressType hierarchical

AddrParams set domain_num_ 6          ;# number of domains
lappend cluster_num 1 1 1 1 1 1      ;# number of clusters in each domain
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel 1 1 1 1 1 1      ;# number of nodes in each cluster
AddrParams set nodes_num_ $eilastlevel ;# of each domain

$ns_ use-newtrace
set tracefd      [open out.tr w]
$ns_ trace-all $tracefd

#create wired nodes
set temp {0.0.0}          ;# hierarchical addresses
for {set i 0} {$i < $num_wired_nodes} {incr i} {
    set W($i) [$ns_ node 0.0.0]
}

# set up topography object
set topo          [new Topography]

$topo load_flatgrid $val(x) $val(y)

# Create God
create-god $val(nn)

# Create channel
set chan_1_ [new $val(chan)]

$ns_ node-config -adhocRouting $val(rp) \
                -llType $val(ll) \
                -macType $val(mac) \
                -ifqType $val(ifq) \
                -ifqLen $val(ifqlen) \
                -antType $val(ant) \
                -propType $val(prop) \

```

```

        -phyType $val(netif) \
        -wiredRouting ON \
        -topoInstance $topo \
        -agentTrace ON \
        -routerTrace ON \
        -macTrace ON \
        -movementTrace ON \
        -channel $chan_1_

for {set i 0} {$i < $num_BS} {incr i} {
    set BS($i) [$ns_ node [lindex $temp_BS $i]]
    $BS($i) random-motion 0                ;# disable random motion
    $BS($i) set X_ [lindex $temp_x [expr $i]]
    $BS($i) set Y_ [lindex $temp_y [expr $i]]
    $BS($i) set Z_ 0.0
}

$ns_ node-config -wiredRouting OFF

for {set i 0} {$i < $val(nn)} {incr i} {
    set node($i) [$ns_ node [lindex $temp_no_ad $i]]

    $node($i) base-station [AddrParams addr2id [$BS([lindex $temp_no_ind
[lexpr $i]]) node-addr]]

    $node($i) random-motion 0                ;# disable random motion
    $node($i) set X_ [lindex $temp_no_x [expr $i]]
    $node($i) set Y_ [lindex $temp_no_y [expr $i]]
    $node($i) set Z_ 0.0
}

#create links between wired and BS nodes
for {set i 0} {$i < $num_BS} {incr i} {
    $ns_ duplex-link $W(0) $BS($i) 2Mb 2ms DropTail
}

Application/FTP set packetSize_ 1040
Application/FTP set rate_ 512Kb

for {set i 0} {$i < $val(nn)} {incr i} {
    set tcpFix($i) [new Agent/TCP/Newreno]
}

```

```
    set node_rx($i) [new Agent/TCPSink]
    $ns_ attach-agent $W(0) $tcpFix($i)
    $ns_ attach-agent $node($i) $node_rx($i)
    $ns_ connect $tcpFix($i) $node_rx($i)
    set ftpFix($i) [new Application/FTP]
    $ftpFix($i) attach-agent $tcpFix($i)
    $ns_ at 1.0 "$ftpFix($i) start"
}

$ns_ at 20.0 "stop"
$ns_ at 20.0 "puts \"NS EXITING...\" ; $ns_ halt"

proc stop {} {
    global ns_ tracefd
    $ns_ flush-trace
    close $tracefd
    exit 0
}

puts "Starting Simulation..."
$ns_ run
```

## Apêndice F - Arquivo wireless\_wpgcps\_3.tcl

wireless\_wpgcps\_3.tcl

```

set val(chan) Channel/WirelessChannel ;#Channel Type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
set val(nn) 20 ;# number of mobilenodes
set val(rp) DSDV
set val(x) 15e3
set val(y) 15e3
set num_wired_nodes 1
set num_BS 5
set temp_BS {1.0.0 2.0.0 3.0.0 4.0.0 5.0.0} ;# hierarchical addresses BS
set temp_x {5.0e3 7.0e3 5.0e3 3.0e3 5.0e3} ;# BS x position
set temp_y {5.0e3 5.0e3 7.0e3 5.0e3 3.0e3} ;# BS y position
set temp_no_ind {0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4}
set temp_no_ad {1.0.1 1.0.2 1.0.3 1.0.4 \
                2.0.1 2.0.2 2.0.3 2.0.4 \
                3.0.1 3.0.2 3.0.3 3.0.4 \
                4.0.1 4.0.2 4.0.3 4.0.4 \
                5.0.1 5.0.2 5.0.3 5.0.4}
set temp_no_x {5.2e3 5.0e3 4.8e3 5.0e3 \
               7.2e3 7.0e3 6.8e3 7.0e3 \
               5.2e3 5.0e3 4.8e3 5.0e3 \
               3.2e3 3.0e3 2.8e3 3.0e3 \
               5.2e3 5.0e3 4.8e3 5.0e3}
set temp_no_y {5.0e3 5.2e3 5.0e3 4.8e3 \
               5.0e3 5.2e3 5.0e3 4.8e3 \
               7.0e3 7.2e3 7.0e3 6.8e3 \
               5.0e3 5.2e3 5.0e3 4.8e3 \
               3.0e3 3.2e3 3.0e3 2.8e3}

Mac/802_11 set dataRate_ 11Mb

Phy/WirelessPhy set Pt_ 2

```

```

# Initialize Global Variables
set ns_          [new Simulator]

# set up for hierarchical routing
$ns_ node-config -addressType hierarchical

AddrParams set domain_num_ 6          ;# number of domains
lappend cluster_num 1 1 1 1 1 1      ;# number of clusters in each domain
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel 1 1 1 1 1 1      ;# number of nodes in each cluster
AddrParams set nodes_num_ $eilastlevel ;# of each domain

$ns_ use-newtrace
set tracefd      [open out.tr w]
$ns_ trace-all $tracefd

#create wired nodes
set temp {0.0.0}          ;# hierarchical addresses
for {set i 0} {$i < $num_wired_nodes} {incr i} {
    set W($i) [$ns_ node 0.0.0]
}

# set up topography object
set topo          [new Topography]

$topo load_flatgrid $val(x) $val(y)

# Create God
create-god $val(nn)

# Create channel
set chan_1_ [new $val(chan)]

$ns_ node-config -adhocRouting $val(rp) \
                -llType $val(ll) \
                -macType $val(mac) \
                -ifqType $val(ifq) \
                -ifqLen $val(ifqlen) \
                -antType $val(ant) \
                -propType $val(prop) \

```

```

        -phyType $val(netif) \
        -wiredRouting ON \
        -topoInstance $topo \
        -agentTrace ON \
        -routerTrace ON \
        -macTrace ON \
        -movementTrace ON \
        -channel $chan_1_

for {set i 0} {$i < $num_BS} {incr i} {
    set BS($i) [$ns_ node [lindex $temp_BS $i]]
    $BS($i) random-motion 0                ;# disable random motion
    $BS($i) set X_ [lindex $temp_x [expr $i]]
    $BS($i) set Y_ [lindex $temp_y [expr $i]]
    $BS($i) set Z_ 0.0
}

$ns_ node-config -wiredRouting OFF

for {set i 0} {$i < $val(nn)} {incr i} {
    set node($i) [$ns_ node [lindex $temp_no_ad $i]]

    $node($i) base-station [AddrParams addr2id [$BS([lindex $temp_no_ind
[lexpr $i]]) node-addr]]

    $node($i) random-motion 0                ;# disable random motion
    $node($i) set X_ [lindex $temp_no_x [expr $i]]
    $node($i) set Y_ [lindex $temp_no_y [expr $i]]
    $node($i) set Z_ 0.0
}

#create links between wired and BS nodes
for {set i 0} {$i < $num_BS} {incr i} {
    $ns_ duplex-link $W(0) $BS($i) 2Mb 2ms DropTail
}

Application/FTP set packetSize_ 1040
Application/FTP set rate_ 512Kb

# create connection between nodes
set tcp1 [new Agent/TCP/Newreno]

```



```
$tcp1 set class_ 1 ;#identificacao do fluxo de pacotes
set sink1 [new Agent/TCPSink]
$ns_ attach-agent $node(0) $tcp1
$ns_ attach-agent $node(5) $sink1
$ns_ connect $tcp1 $sink1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns_ at 1.0 "$ftp1 start"

set tcp2 [new Agent/TCP/Newreno]
$tcp2 set class_ 2 ;#identificacao do fluxo de pacotes
set sink2 [new Agent/TCPSink]
$ns_ attach-agent $node(1) $tcp2
$ns_ attach-agent $node(11) $sink2
$ns_ connect $tcp2 $sink2
set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2
$ns_ at 1.0 "$ftp2 start"

set tcp3 [new Agent/TCP/Newreno]
$tcp3 set class_ 3 ;#identificacao do fluxo de pacotes
set sink3 [new Agent/TCPSink]
$ns_ attach-agent $node(2) $tcp3
$ns_ attach-agent $node(12) $sink3
$ns_ connect $tcp3 $sink3
set ftp3 [new Application/FTP]
$ftp3 attach-agent $tcp3
$ns_ at 1.0 "$ftp3 start"

set tcp4 [new Agent/TCP/Newreno]
$tcp4 set class_ 4 ;#identificacao do fluxo de pacotes
set sink4 [new Agent/TCPSink]
$ns_ attach-agent $node(3) $tcp4
$ns_ attach-agent $node(17) $sink4
$ns_ connect $tcp4 $sink4
set ftp4 [new Application/FTP]
$ftp4 attach-agent $tcp4
$ns_ at 1.0 "$ftp4 start"

set tcp5 [new Agent/TCP/Newreno]
$tcp5 set class_ 5 ;#identificacao do fluxo de pacotes
```

```
set sink5 [new Agent/TCPSink]
$ns_ attach-agent $node(4) $tcp5
$ns_ attach-agent $node(9) $sink5
$ns_ connect $tcp5 $sink5
set ftp5 [new Application/FTP]
$ftp5 attach-agent $tcp5
$ns_ at 1.0 "$ftp5 start"

set tcp6 [new Agent/TCP/Newreno]
$tcp6 set class_ 6 ;#identificacao do fluxo de pacotes
set sink6 [new Agent/TCPSink]
$ns_ attach-agent $node(6) $tcp6
$ns_ attach-agent $node(19) $sink6
$ns_ connect $tcp6 $sink6
set ftp6 [new Application/FTP]
$ftp6 attach-agent $tcp6
$ns_ at 1.0 "$ftp6 start"

set tcp7 [new Agent/TCP/Newreno]
$tcp7 set class_ 7 ;#identificacao do fluxo de pacotes
set sink7 [new Agent/TCPSink]
$ns_ attach-agent $node(7) $tcp7
$ns_ attach-agent $node(14) $sink7
$ns_ connect $tcp7 $sink7
set ftp7 [new Application/FTP]
$ftp7 attach-agent $tcp7
$ns_ at 1.0 "$ftp7 start"

set tcp8 [new Agent/TCP/Newreno]
$tcp8 set class_ 8 ;#identificacao do fluxo de pacotes
set sink8 [new Agent/TCPSink]
$ns_ attach-agent $node(8) $tcp8
$ns_ attach-agent $node(16) $sink8
$ns_ connect $tcp8 $sink8
set ftp8 [new Application/FTP]
$ftp8 attach-agent $tcp8
$ns_ at 1.0 "$ftp8 start"

set tcp9 [new Agent/TCP/Newreno]
$tcp9 set class_ 9 ;#identificacao do fluxo de pacotes
set sink9 [new Agent/TCPSink]
```

```
$ns_ attach-agent $node(10) $tcp9
$ns_ attach-agent $node(13) $sink9
$ns_ connect $tcp9 $sink9
set ftp9 [new Application/FTP]
$ftp9 attach-agent $tcp9
$ns_ at 1.0 "$ftp9 start"

set tcp10 [new Agent/TCP/Newreno]
$tcp10 set class_ 10 ;#identificacao do fluxo de pacotes
set sink10 [new Agent/TCPSink]
$ns_ attach-agent $node(15) $tcp10
$ns_ attach-agent $node(18) $sink10
$ns_ connect $tcp10 $sink10
set ftp10 [new Application/FTP]
$ftp10 attach-agent $tcp10
$ns_ at 1.0 "$ftp10 start"

$ns_ at 20.0 "stop"
$ns_ at 20.0 "puts \"NS EXITING...\" ; $ns_ halt"

proc stop {} {
    global ns_ tracefd
    $ns_ flush-trace
    close $tracefd
    exit 0
}

puts "Starting Simulation..."
$ns_ run
```

## Apêndice G - Arquivo wireless\_dincon\_60n.tcl

wireless\_dincon\_60n.tcl

```

# wireless3.tcl
# simulation of a wired-cum-wireless topology running with mobileIP
# =====
# Define options
# =====

set opt(chan)    Channel/WirelessChannel        ;# channel type
set opt(prop)    Propagation/TwoRayGround       ;# radio-propagation model
set opt(netif)   Phy/WirelessPhy               ;# network interface type
set opt(mac)     Mac/802_11                    ;# MAC type
set opt(ifq)     Queue/DropTail/PriQueue       ;# interface queue type
set opt(ll)      LL                             ;# link layer type
set opt(ant)     Antenna/OmniAntenna           ;# antenna model
set opt(ifqlen)  50                            ;# max packet in ifq
set opt(nn)      60                            ;# number of mobilenodes
set opt(adhocRouting) DSDV                     ;# routing protocol

set opt(sc) "setdest_60n"                      ;# node movement file.
set opt(cp) ""                                  ;# Trafego
set opt(x)      2000                            ;# x coordinate of topology
set opt(y)      200                             ;# y coordinate of topology
set opt(seed)   0.0                             ;# random seed
set opt(stop)   250                             ;# time to stop simulation

set opt(ftp1-start)  1.0

set num_wired_nodes  1
set num_BS           6

set temp_BS {1.0.0 2.0.0 3.0.0 4.0.0 5.0.0 6.0.0};# hierarchical addr BS
set temp_x {1.0 300.0 600.0 900.0 1200.0 1500.0}      ;# BS x position
set temp_y {100.0 100.0 100.0 100.0 100.0 100.0}     ;# BS y position

set temp_no_ind { 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2
2 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 }
set temp_no_ad {1.0.1 1.0.2 1.0.3 1.0.4 1.0.5 1.0.6 1.0.7 1.0.8 1.0.9
1.0.10 \

```

```

2.0.1 2.0.2 2.0.3 2.0.4 2.0.5 2.0.6 2.0.7 2.0.8 2.0.9 2.0.10 \
3.0.1 3.0.2 3.0.3 3.0.4 3.0.5 3.0.6 3.0.7 3.0.8 3.0.9 3.0.10 \
4.0.1 4.0.2 4.0.3 4.0.4 4.0.5 4.0.6 4.0.7 4.0.8 4.0.9 4.0.10 \
5.0.1 5.0.2 5.0.3 5.0.4 5.0.5 5.0.6 5.0.7 5.0.8 5.0.9 5.0.10 \
6.0.1 6.0.2 6.0.3 6.0.4 6.0.5 6.0.6 6.0.7 6.0.8 6.0.9 5.0.10 }

set temp_no_x {2.0 22.0 10.0 10.0 10.0 30.0 150.0 150.0 150.0 90.0 \
290.0 290.0 280.0 320.0 280.0 320.0 450.0 450.0 350.0 400.0\
550.0 600.0 580.0 580.0 560.0 630.0 630.0 680.0 570.0 630.0 \
850.0 890.0 870.0 870.0 910.0 950.0 950.0 950.0 800.0 960.0 \
1000.0 1190.0 1170.0 1000.0 1200.0 1180.0 1000.0 1300.0 1100.0
1300.0 \
1500.0 1300.0 1500.0 1400.0 1400.0 1400.0 1400.0 1600.0 1300.0
1650.0 \ }

set temp_no_y {2.0 22.0 120.0 60.0 190.0 150.0 90.0 120.0 90.0 190.0 \
90.0 110.0 180.0 30.0 10.0 160.0 10.0 70.0 150.0 110.0 \
110.0 10.0 150.0 50.0 190.0 40.0 160.0 70.0 90.0 110.0 \
80.0 20.0 120.0 190.0 150.0 180.0 50.0 120.0 50.0 10.0 \
70.0 180.0 110.0 5.0 30.0 50.0 50.0 60.0 90.0 150.0 \
90.0 130.0 10.0 190.0 40.0 160.0 80.0 120.0 20.0 50.0 }

# =====

# check for boundary parameters and random seed
if { $opt(x) == 0 || $opt(y) == 0 } {
    puts "No X-Y boundary values given for wireless topology\n"
}
if {$opt(seed) > 0} {
    puts "Seeding Random number generator with $opt(seed)\n"
    ns-random $opt(seed)
}

# create simulator instance
set ns_ [new Simulator]

# set up for hierarchical routing
$ns_ node-config -addressType hierarchical

AddrParams set domain_num_ 7 ;# number of domains
lappend cluster_num 1 1 1 1 1 1 1 ;# number of clusters in each domain
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel 1 1 1 1 1 1 1 ;# number of nodes in each cluster

```

```

AddrParams set nodes_num_ $eplastlevel ;# of each domain

$ns_ use-newtrace
set tracefd [open wireless_60n.tr w]
set namtrace [open wireless_60n.nam w]
$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace $opt(x) $opt(y)

# Create topography object
set topo [new Topography]

# define topology
$topo load_flatgrid $opt(x) $opt(y)

# create God
create-god $opt(nn)

#create wired nodes
set temp {0.0.0} ;# hierarchical addresses
for {set i 0} {$i < $num_wired_nodes} {incr i} {
    set W($i) [$ns_ node [lindex $temp $i]]
}

# Configure for ForeignAgent and HomeAgent nodes
$ns_ node-config -mobileIP ON \
                -adhocRouting $opt(adhocRouting) \
                -llType $opt(ll) \
                -macType $opt(mac) \
                -ifqType $opt(ifq) \
                -ifqLen $opt(ifqlen) \
                -antType $opt(ant) \
                -propType $opt(prop) \
                -phyType $opt(netif) \
                -channelType $opt(chan) \
                -topoInstance $topo \
                -wiredRouting ON \
                -agentTrace ON \
                -routerTrace OFF \
                -macTrace OFF

# Create BS

```

```

for {set i 0} {$i < $num_BS} {incr i} {
    set BS($i) [$ns_ node [lindex $temp_BS $i]]
    $BS($i) random-motion 0                ;# disable random motion
    $BS($i) set X_ [lindex $temp_x [expr $i]]
    $BS($i) set Y_ [lindex $temp_y [expr $i]]
    $BS($i) set Z_ 0.0
}

# create a mobilenode
$ns_ node-config -wiredRouting OFF

for {set i 0} {$i < $opt(nn)} {incr i} {
    set node_($i) [$ns_ node [lindex $temp_no_ad $i]]

    $node_($i) base-station [AddrParams addr2id [$BS([lindex $temp_no_ind
[expr $i]]) node-addr]]

    #$node_($i) random-motion 0            ;# disable random motion
    $node_($i) set X_ [lindex $temp_no_x [expr $i]]
    $node_($i) set Y_ [lindex $temp_no_y [expr $i]]
    $node_($i) set Z_ 0.0
}

# source connection-pattern and node-movement scripts
puts "Loading scenario file..."
source $opt(sc)
puts "Load complete..."

# create links between wired and BaseStation nodes
for {set i 0} {$i < $num_BS} {incr i} {
    $ns_ duplex-link $W(0) $BS($i) 2Mb 2ms DropTail
}

# setup TCP connections between a wired node and the MobileHost
for {set i 0} {$i < $opt(nn)} {incr i} {
    set tcpFix($i) [new Agent/TCP/Newreno]
    set node_rx($i) [new Agent/TCPSink]
    $ns_ attach-agent $W(0) $tcpFix($i)
    $ns_ attach-agent $node_($i) $node_rx($i)
    $ns_ connect $tcpFix($i) $node_rx($i)
}

```

```

    set ftpFix($i) [new Application/FTP]
    $ftpFix($i) attach-agent $tcpFix($i)
    $ns_ at $opt(ftp1-start) "$ftpFix($i) start"
}

# Define initial node position in nam

for {set i 0} {$i < $opt(nn)} {incr i} {
    # 20 defines the node size in nam, must adjust it according to your
    # scenario
    # The function must be called after mobility model is defined
    $ns_ initial_node_pos $node_($i) 1
}

# Tell all nodes when the simulation ends
for {set i 0} {$i < $opt(nn)} {incr i} {
    $ns_ at $opt(stop).0 "$node_($i) reset";
}

for {set i 0} {$i < $num_BS} {incr i} {
    $ns_ at $opt(stop).0 "$BS($i) reset";
}

$ns_ at $opt(stop).0002 "puts \"NS EXITING...\" ; $ns_ halt"
$ns_ at $opt(stop).0001 "stop"
proc stop {} {
    global ns_ tracefd namtrace
    close $tracefd
    close $namtrace
}

# some useful headers for tracefile
puts $tracefd "M 0.0 nn $opt(nn) x $opt(x) y $opt(y) rp \
    $opt(adhocRouting)"
puts $tracefd "M 0.0 sc $opt(sc) cp $opt(cp) seed $opt(seed)"
puts $tracefd "M 0.0 prop $opt(prop) ant $opt(ant)"

puts "Starting Simulation..."
$ns_ run

```



## Apêndice H - Arquivo setdest\_60n

### Trecho do arquivo setdest\_60n

```
#
# nodes: 60, pause: 0.00, max speed: 20.00, max x: 2000.00, max y: 200.00
#
$ns_ at 0.000000000000 "$node_(0) setdest 1748.301345307073 164.256911670193 17.460207322548"
$ns_ at 0.000000000000 "$node_(1) setdest 721.460177112305 147.271527404077 8.223979773602"
$ns_ at 0.000000000000 "$node_(2) setdest 1605.735507326430 42.826252455791 10.377454324279"
$ns_ at 0.000000000000 "$node_(3) setdest 670.591697442130 0.746266393741 5.650440243839"
$ns_ at 0.000000000000 "$node_(4) setdest 1554.131462066282 64.745129427637 17.004596376083"
$ns_ at 0.000000000000 "$node_(5) setdest 1527.164671022969 16.862269469386 8.540497058170"
.
.
.
$ns_ at 10.960148289890 "$node_(20) setdest 696.225153472003 43.983155476976 9.951933186967"
$ns_ at 11.011008742196 "$node_(7) setdest 369.154211316275 32.826268679431 11.311082647346"
$ns_ at 11.742461607525 "$node_(18) setdest 254.559588373731 172.688482962364 3.274990795459"
$ns_ at 12.463376160127 "$node_(9) setdest 1799.208445063643 75.580516765068 13.244472144835"
$ns_ at 14.622940338512 "$node_(58) setdest 1437.027104408862 39.840206896598 1.759997863807"
$ns_ at 17.002611653515 "$node_(26) setdest 1259.366671542699 36.503210056729
19.833346946476"
$ns_ at 17.247721704660 "$node_(25) setdest 1523.586694827777 37.356950056331
13.162233642709"
$ns_ at 17.721240379385 "$node_(24) setdest 1641.116429900801 96.428274469702 7.098850378897"
$ns_ at 18.092650103766 "$node_(47) setdest 1821.717670400924 178.331864181214
1.064785323449"
$ns_ at 19.807655141850 "$node_(49) setdest 282.602265193423 90.594430138274 3.461106326410"
$ns_ at 232.077328729624 "$node_(23) setdest 1204.706907407157 149.096886676772
2.813332831760"
$ns_ at 232.428090980525 "$node_(21) setdest 1139.088056732508 64.989864248292
0.258925006226"
$ns_ at 235.235910939257 "$node_(3) setdest 1588.535846773409 27.422441193803
12.455505759163"
$ns_ at 238.272848267625 "$node_(10) setdest 1090.477911946226 54.379772094775
2.111045620194"
$ns_ at 239.332267973040 "$node_(32) setdest 1565.877807257367 46.489783765253
11.232807681995"
$ns_ at 239.523859700342 "$node_(17) setdest 676.916051376271 161.094603805728
2.148868107927"
$ns_ at 240.291325006891 "$node_(18) setdest 593.234614793398 146.284232155215
15.877933316540"
$ns_ at 242.234474843047 "$node_(3) setdest 584.546475109101 124.264015454546 4.443241717339"
$ns_ at 246.963447669606 "$node_(5) setdest 1213.550570052704 14.039596710409 4.399047464831"
$ns_ at 247.601307516143 "$node_(37) setdest 1257.551362638055 137.859000841778
4.718306814648"
$ns_ at 249.454867909543 "$node_(42) setdest 507.066652055332 59.937921880532
17.279081082449"
```



```

2.0.1 2.0.2 2.0.3 2.0.4 2.0.5 2.0.6 2.0.7 \
3.0.1 3.0.2 \
4.0.1 4.0.2 \
5.0.1 5.0.2 5.0.3 5.0.4 5.0.5 5.0.6 5.0.7 5.0.8 \
6.0.1 6.0.2 6.0.3 \
7.0.1 7.0.2 \
8.0.1 8.0.2 \
9.0.1 9.0.2 }

set temp_no_x { 16.0 14.0 \
                250.0 292.0 152.0 326.0 283.0 377.0 250.0 \
                488.0 490.0 \
                102.0 77.0 \
                370.0 313.0 250.0 240.0 177.0 250.0 168.0 369.0 \
                490.0 490.0 490.0 \
                130.0 10.0 \
                241.0 183.0 \
                409.0 490.0 }

set temp_no_y { 10.0 130.0 120.0 60.0 190.0 \
                17.0 130.0 10.0 130.0 130.0 130.0 132.0 \
                10.0 30.0 \
                250.0 250.0 \
                162.0 250.0 270.0 370.0 250.0 173.0 250.0 143.0 \
                289.0 324.0 308.0 \
                433.0 445.0 \
                490.0 490.0 \
                490.0 468.0 }

# =====
# check for boundary parameters and random seed
if { $opt(x) == 0 || $opt(y) == 0 } {
    puts "No X-Y boundary values given for wireless topology\n"
}

if {$opt(seed) > 0} {
    puts "Seeding Random number generator with $opt(seed)\n"
    ns-random $opt(seed)
}

# create simulator instance
set ns_ [new Simulator]

```

```

# set up for hierarchical routing
$ns_ node-config -addressType hierarchical
# number of domains
AddrParams set domain_num_ 10
# number of clusters in each domain
lappend cluster_num 1 1 1 1 1 1 1 1 1 1
AddrParams set cluster_num_ $cluster_num
# number of nodes in each cluster
lappend eilastlevel 1 3 8 3 3 9 4 3 3 3
# of each domain
AddrParams set nodes_num_ $eilastlevel

$ns_ use-newtrace
set tracefd [open wireless_30n.tr w]
set namtrace [open wireless_30n.nam w]
$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace $opt(x) $opt(y)

# create topography object
set topo [new Topography]

# define topology
$topo load_flatgrid $opt(x) $opt(y)

# create God
create-god $opt(nn)

# create wired nodes
set temp {0.0.0} ;# hierarchical addresses
for {set i 0} {$i < $num_wired_nodes} {incr i} {
    set W($i) [$ns_ node [lindex $temp $i]]
}

# configure for ForeignAgent and HomeAgent nodes
$ns_ node-config -mobileIP ON \
    -adhocRouting $opt(adhocRouting) \
    -llType $opt(ll) \
    -macType $opt(mac) \
    -ifqType $opt(ifq) \
    -ifqLen $opt(ifqlen) \
    -antType $opt(ant) \

```

```

        -propType $opt(prop) \
        -phyType $opt(netif) \
        -channelType $opt(chan) \
        -topoInstance $topo \
        -wiredRouting ON \
        -agentTrace ON \
        -routerTrace OFF \
        -macTrace OFF

# create BS
for {set i 0} {$i < $num_BS} {incr i} {
    set BS($i) [$ns_ node [lindex $temp_BS $i]]
    $BS($i) random-motion 0
    $BS($i) set X_ [lindex $temp_x [expr $i]]
    $BS($i) set Y_ [lindex $temp_y [expr $i]]
    $BS($i) set Z_ 0.0
}

# create a mobilenode
$ns_ node-config -wiredRouting OFF

for {set i 0} {$i < $opt(nn)} {incr i} {
    set node_($i) [$ns_ node [lindex $temp_no_ad $i]]
    $node_($i) base-station [AddrParams addr2id [$BS([lindex $temp_no_ind
        [expr $i]]) node-addr]]
    $node_($i) set X_ [lindex $temp_no_x [expr $i]]
    $node_($i) set Y_ [lindex $temp_no_y [expr $i]]
    $node_($i) set Z_ 0.0
}

# source connection-pattern and node-movement scripts
puts "Loading scenario file..."
source $opt(sc)
puts "Load complete..."

# create links between wired and BaseStation nodes
for {set i 0} {$i < $num_BS} {incr i} {
    $ns_ duplex-link $W(0) $BS($i) 2Mb 2ms DropTail
}

# setup TCP connections between a wired node and the mobilenode

```

```

for {set i 0} {$i < $opt(nn)} {incr i} {
    set tcpFix($i) [new Agent/TCP/Newreno]
    set node_rx($i) [new Agent/TCPSink]
    $ns_ attach-agent $W(0) $tcpFix($i)
    $ns_ attach-agent $node_($i) $node_rx($i)
    $ns_ connect $tcpFix($i) $node_rx($i)
    set ftpFix($i) [new Application/FTP]
    $ftpFix($i) attach-agent $tcpFix($i)
    $ns_ at $opt(ftp1-start) "$ftpFix($i) start"
}

# define initial node position in nam
for {set i 0} {$i < $opt(nn)} {incr i} {
    $ns_ initial_node_pos $node_($i) 1
}

# tell all nodes when the simulation ends
for {set i 0} {$i < $opt(nn)} {incr i} {
    $ns_ at $opt(stop).0 "$node_($i) reset";
}

# ending nam and the simulation
for {set i 0} {$i < $num_BS } {incr i} {
    $ns_ at $opt(stop).0 "$BS($i) reset";
}

$ns_ at $opt(stop).0002 "puts \"NS EXITING...\" ; $ns_ halt"
$ns_ at $opt(stop).0001 "stop"
proc stop {} {
    global ns_ tracefd namtrace
    close $tracefd
    close $namtrace}

# some useful headers for tracefile
puts $tracefd "M 0.0 nn $opt(nn) x $opt(x) y $opt(y) rp \
    $opt(adhocRouting)"
puts $tracefd "M 0.0 sc $opt(sc) cp $opt(cp) seed $opt(seed)"
puts $tracefd "M 0.0 prop $opt(prop) ant $opt(ant)"
puts "Starting Simulation..."

$ns_ run

```

**Apêndice J - Arquivo scen.params**`scen.params`

```
model=ManhattanGrid
ignore=3600.0
randomSeed=2
x=500.0
y=500.0
duration=900.0
nn=30
circular=false
xblocks=4
yblocks=4
updateDist=5.0
turnProb=0.5
speedChangeProb=0.2
minSpeed=0.5
meanSpeed=1.0
speedStdDev=0.2
pauseProb=0.0
maxPause=120.0
```

## Apêndice K - Arquivo scen.movements.gz

### Trecho do arquivo scen.movements.gz

0.0	240.0	7.297033485385771	8.845708735249445	240.0	0.0	16.75572066720406	246.52515515317245
0.0	52.01434251002638	291.5251551531725	0.0	66.92474555937588	306.5251551531725	0.0	
72.76260688947741	311.5251551531725	0.0	113.58865306852931	360.0	0.0	123.2952651885048	360.0
11.525155153172449	160.4554982430368	360.0	51.52515515317245	167.76495200528507	360.0		
61.52515515317245	189.36874902337604	360.0	96.52515515317245	194.89664051809768	360.0		
101.52515515317245	204.1588289217957	360.0	111.52515515317245	211.85739750606353	360.0	120.0	
226.86888651992876	376.5251551531725	120.0	267.0841636758628	416.5251551531725	120.0		
292.77546263351996	446.5251551531725	120.0	297.4489834508581	451.5251551531725	120.0		
319.96690804709215	480.0	120.0	348.8510229221383	480.0	83.47484484682755	368.4513054089216	
480.0	58.47484484682755	377.1881340129921	480.0	53.47484484682755	425.98862380041965	480.0	
0.0	450.19515617784145	453.4748448468275	0.0	476.08820342151967	428.4748448468275	0.0	
484.61329925900736	423.4748448468275	0.0	511.3480482900268	403.4748448468275	0.0		
519.081197945774	393.4748448468275	0.0	549.4095285632065	358.4748448468275	0.0		
588.9833900103768	328.4748448468275	0.0	604.7424821688764	308.4748448468275	0.0		
613.9245433529177	298.4748448468275	0.0	619.3065919473947	293.4748448468275	0.0		
671.069357375688	240.0	0.0	701.5851861912442	240.0	31.52515515317245	713.5894097873634	240.0
41.52515515317245	726.170878924433	240.0	56.52515515317245	754.5532304478947	240.0		
76.52515515317245	807.903845910042	240.0	120.0	809.7754561799275	241.52515515317245	120.0	
825.1259794118896	256.5251551531725	120.0	900.0	315.0178810953694	120.0		
0.0	231.1105652282368	480.0	8.35515572409895	240.0	480.0	10.091434970050159	240.0
478.1526925874125	40.694600120908035	240.0	443.1526925874125	77.69690753627583	240.0		
413.1526925874125	95.0756243920714	240.0	393.1526925874125	102.41276671091282	240.0		
388.1526925874125	133.36510745324767	240.0	368.1526925874125	148.9389070486477	240.0		
353.1526925874125	156.81059804867436	240.0	343.1526925874125	232.8275066131937	240.0		
253.1526925874125	250.31960607821566	240.0	240.0	305.97340158354564	198.1526925874125	240.0	
363.81990380509524	143.1526925874125	240.0	369.52367277545	138.1526925874125	240.0		
392.57080154424693	120.0	240.0	452.04935665171433	120.0	286.8473074125875	494.3301506876196	
120.0	336.8473074125875	514.620864318782	120.0	356.8473074125875	524.766030063678	120.0	
366.8473074125875	530.2434669028335	120.0	371.8473074125875	534.7496078362356	120.0		
376.8473074125875	551.1164364540728	120.0	391.8473074125875	602.8655802971425	120.0		
426.8473074125875	669.7951064627377	120.0	476.8473074125875	673.0186542625561	120.0	480.0	
700.4693416801747	146.8473074125875	480.0	704.3118544049657	151.8473074125875	480.0		
718.4195925063068	166.8473074125875	480.0	723.8253981543585	171.8473074125875	480.0		
732.6893299523508	181.8473074125875	480.0	745.1135244033685	196.8473074125875	480.0		
811.7308018810581	246.8473074125875	480.0	819.6691824852842	256.8473074125875	480.0		
837.8281984967016	276.8473074125875	480.0	859.2887262613613	291.8473074125875	480.0		
875.4380848071596	306.8473074125875	480.0	885.1150127330166	316.8473074125875	480.0		
889.6689196740354	321.8473074125875	480.0	900.0	334.8215679894918	480.0		



**Apêndice L - Arquivo scen.ns\_params**

scen.ns\_params

```
set val(x) 500.0  
set val(y) 500.0  
set val(nn) 30  
set val(duration) 900.0
```

**Apêndice M - Arquivo scen.ns\_movements.gz****Trecho do arquivo scen.ns\_movements.gz**

```
$node_(0) set X_ 250.0
$node_(0) set Y_ 17.29703348538577
$ns_ at 0.0 "$node_(0) setdest 250.0 10.0 0.8249235537575043"
$ns_ at 8.845708735249445 "$node_(0) setdest 256.5251551531725 10.0 0.8249235537575279"
$ns_ at 16.75572066720406 "$node_(0) setdest 301.5251551531725 10.0 1.2762835768398242"
$ns_ at 52.01434251002638 "$node_(0) setdest 316.5251551531725 10.0 1.006009022717492"
$ns_ at 66.92474555937588 "$node_(0) setdest 321.5251551531725 10.0 0.8564780348958786"
$ns_ at 72.76260688947741 "$node_(0) setdest 370.0 10.0 1.1873509532181998"
$ns_ at 113.58865306852931 "$node_(0) setdest 370.0 21.52515515317245 1.1873509532182225"
$ns_ at 123.2952651885048 "$node_(0) setdest 370.0 61.52515515317245 1.0764195138738955"
$ns_ at 160.4554982430368 "$node_(0) setdest 370.0 71.52515515317245 1.3680912863349401"
$ns_ at 167.76495200528507 "$node_(0) setdest 370.0 106.52515515317245 1.6200855789698023"
$ns_ at 189.36874902337604 "$node_(0) setdest 370.0 111.52515515317245 0.9045040056908307"
$ns_ at 194.89664051809768 "$node_(0) setdest 370.0 121.52515515317245 1.0796584526403503"
$ns_ at 204.1588289217957 "$node_(0) setdest 370.0 130.0 1.1008338438584615"
$ns_ at 211.85739750606353 "$node_(0) setdest 386.5251551531725 130.0 1.1008338438584715"
$ns_ at 226.86888651992876 "$node_(0) setdest 426.5251551531725 130.0 0.9946468812063812"
$ns_ at 267.0841636758628 "$node_(0) setdest 456.5251551531725 130.0 1.1677105174574534"
$ns_ at 292.77546263351996 "$node_(0) setdest 461.5251551531725 130.0 1.0698572223002938"
$ns_ at 297.4489834508581 "$node_(0) setdest 490.0 130.0 1.2645412646771965"
$ns_ at 319.96690804709215 "$node_(0) setdest 490.0 93.47484484682755 1.264541264677201"
$ns_ at 348.8510229221383 "$node_(0) setdest 490.0 68.47484484682755 1.275491820939715"
$ns_ at 368.4513054089216 "$node_(0) setdest 490.0 63.47484484682755 0.5722900409961664"
$ns_ at 377.1881340129921 "$node_(0) setdest 490.0 10.0 1.095785002973562"
$ns_ at 425.98862380041965 "$node_(0) setdest 463.4748448468275 10.0 1.0957850029735494"
$ns_ at 450.19515617784145 "$node_(0) setdest 438.4748448468275 10.0 0.9655101527729129"
$ns_ at 476.08820342151967 "$node_(0) setdest 433.4748448468275 10.0 0.5865036704940404"
$ns_ at 484.61329925900736 "$node_(0) setdest 413.4748448468275 10.0 0.7480900597493801"
$ns_ at 511.3480482900268 "$node_(0) setdest 403.4748448468275 10.0 1.2931341620381187"
$ns_ at 519.081197945774 "$node_(0) setdest 368.4748448468275 10.0 1.1540364829669247"
$ns_ at 549.4095285632065 "$node_(0) setdest 338.4748448468275 10.0 0.7580761366956573"
$ns_ at 588.9833900103768 "$node_(0) setdest 318.4748448468275 10.0 1.2691086389270818"
$ns_ at 604.7424821688764 "$node_(0) setdest 308.4748448468275 10.0 1.0890800877454812"
$ns_ at 613.9245433529177 "$node_(0) setdest 303.4748448468275 10.0 0.9290142800142858"
$ns_ at 619.3065919473947 "$node_(0) setdest 250.0 10.0 1.033075501364122"
$ns_ at 671.069357375688 "$node_(0) setdest 250.0 41.52515515317245 1.033075501364121"
$ns_ at 701.5851861912442 "$node_(0) setdest 250.0 51.52515515317245 0.8330401312445505"
$ns_ at 713.5894097873634 "$node_(0) setdest 250.0 66.52515515317245 1.1922296066207736"
$ns_ at 726.170878924433 "$node_(0) setdest 250.0 86.52515515317245 0.7046632476335658"
$ns_ at 754.5532304478947 "$node_(0) setdest 250.0 130.0 0.8148892842234092"
$ns_ at 807.903845910042 "$node_(0) setdest 251.52515515317245 130.0 0.8148892842235881"
$ns_ at 809.7754561799275 "$node_(0) setdest 266.5251551531725 130.0 0.9771653886538362"
$ns_ at 825.1259794118896 "$node_(0) setdest 325.0178810953694 130.0 0.7812152397154059"
```

## Apêndice N - Arquivos *trace*

### N.1 - Trecho do arquivo `wireless_emc_cilamce_3nos.tr`

```

d -t 1.365983348 -Hs 2 -Hd -2 -Ni 2 -Nx 260.00 -Ny 20.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw
IFQ -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 8388609.2 -It tcp -Il 1040 -If 0 -Ii 30 -Iv 30 -Pn
tcp -Ps 1 -Pa 0 -Pf 0 -Po 0
d -t 1.418782605 -Hs 2 -Hd -2 -Ni 2 -Nx 260.00 -Ny 20.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw
IFQ -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 8388609.2 -It tcp -Il 1040 -If 0 -Ii 31 -Iv 30 -Pn
tcp -Ps 2 -Pa 0 -Pf 0 -Po 0
d -t 1.431028400 -Hs 2 -Hd -2 -Ni 2 -Nx 260.00 -Ny 20.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw
IFQ -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 8388609.2 -It tcp -Il 1040 -If 0 -Ii 113 -Iv 30 -Pn
tcp -Ps 3 -Pa 0 -Pf 0 -Po 0
d -t 1.613748192 -Hs 2 -Hd -2 -Ni 2 -Nx 260.00 -Ny 20.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw
IFQ -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 8388609.2 -It tcp -Il 1040 -If 0 -Ii 117 -Iv 30 -Pn
tcp -Ps 1 -Pa 0 -Pf 0 -Po 0
d -t 1.617908192 -Hs 2 -Hd -2 -Ni 2 -Nx 260.00 -Ny 20.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw
IFQ -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 8388609.2 -It tcp -Il 1040 -If 0 -Ii 118 -Iv 30 -Pn
tcp -Ps 4 -Pa 0 -Pf 0 -Po 0
d -t 1.729632998 -Hs 2 -Hd -2 -Ni 2 -Nx 260.00 -Ny 20.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw
IFQ -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 8388609.2 -It tcp -Il 1040 -If 0 -Ii 120 -Iv 30 -Pn
tcp -Ps 5 -Pa 0 -Pf 0 -Po 0
d -t 1.902457180 -Hs 2 -Hd -2 -Ni 2 -Nx 260.00 -Ny 20.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw
IFQ -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 8388609.2 -It tcp -Il 1040 -If 0 -Ii 130 -Iv 30 -Pn
tcp -Ps 6 -Pa 0 -Pf 0 -Po 0
d -t 1.995473207 -Hs 2 -Hd -2 -Ni 2 -Nx 260.00 -Ny 20.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw
IFQ -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 8388609.2 -It tcp -Il 1040 -If 0 -Ii 132 -Iv 30 -Pn
tcp -Ps 7 -Pa 0 -Pf 0 -Po 0
d -t 2.011751529 -Hs 2 -Hd -2 -Ni 2 -Nx 260.00 -Ny 20.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw
IFQ -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 8388609.2 -It tcp -Il 1040 -If 0 -Ii 165 -Iv 30 -Pn
tcp -Ps 8 -Pa 0 -Pf 0 -Po 0
d -t 2.445226283 -Hs 2 -Hd -2 -Ni 2 -Nx 260.00 -Ny 20.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw
IFQ -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 8388609.2 -It tcp -Il 1040 -If 0 -Ii 166 -Iv 30 -Pn
tcp -Ps 9 -Pa 0 -Pf 0 -Po 0
d -t 3.233412405 -Hs 12 -Hd 25165824 -Ni 12 -Nx 412.83 -Ny 490.00 -Nz 0.00 -Ne -1.000000 -Nl
IFQ -Nw ARP -Ma 0 -Md 0 -Ms b -Mt 800 -Is 37748737.0 -Id 25165824.0 -It udp -Il 72 -If 0 -Ii
465 -Iv 32
d -t 3.272238608 -Hs 12 -Hd 25165824 -Ni 12 -Nx 412.87 -Ny 490.00 -Nz 0.00 -Ne -1.000000 -Nl
IFQ -Nw ARP -Ma 0 -Md 0 -Ms b -Mt 800 -Is 37748737.2 -Id 0.2 -It ack -Il 60 -If 0 -Ii 480 -Iv
32 -Pn tcp -Ps 89 -Pa 0 -Pf 0 -Po 0
d -t 12.337643495 -Hs 11 -Hd 29360128 -Ni 11 -Nx 250.00 -Ny 485.58 -Nz 0.00 -Ne -1.000000 -Nl
IFQ -Nw ARP -Ma 0 -Md 0 -Ms a -Mt 800 -Is 33554433.0 -Id 29360128.0 -It udp -Il 72 -If 0 -Ii
2084 -Iv 32
d -t 12.400131780 -Hs 11 -Hd 29360128 -Ni 11 -Nx 250.00 -Ny 485.51 -Nz 0.00 -Ne -1.000000 -Nl
IFQ -Nw ARP -Ma 0 -Md 0 -Ms a -Mt 800 -Is 33554433.2 -Id 0.1 -It ack -Il 60 -If 0 -Ii 2087 -
Iv 32 -Pn tcp -Ps 386 -Pa 0 -Pf 0 -Po 0

```

## N.2 - Trecho do arquivo wireless\_emc\_cilamce\_6nos.tr

```

d -t 0.817625289 -Hs 15 -Hd 37748736 -Ni 15 -Nx 490.00 -Ny 467.55 -Nz 0.00 -Ne -1.000000 -Nl
IFQ -Nw ARP -Ma 0 -Md 0 -Ms e -Mt 800 -Is 37748738.0 -Id 37748736.0 -It udp -Il 92 -If 0 -Ii
8 -Iv 32
d -t 10.002160000 -Hs 2 -Hd -2 -Ni 2 -Nx 260.00 -Ny 20.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw
IFQ -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 33554432.0 -Id 8388610.0 -It udp -Il 48 -If 0 -Ii 11 -Iv 29
d -t 12.387911535 -Hs 5 -Hd 25165824 -Ni 5 -Nx 260.00 -Ny 260.00 -Nz 0.00 -Ne -1.000000 -Nl
IFQ -Nw ARP -Ma 0 -Md 0 -Ms 4 -Mt 800 -Is 0.2 -Id 20971521.2 -It tcp -Il 40 -If 0 -Ii 44 -Iv
29 -Pn tcp -Ps 0 -Pa 0 -Pf 0 -Po 0
d -t 12.387911535 -Hs 5 -Hd 25165824 -Ni 5 -Nx 260.00 -Ny 260.00 -Nz 0.00 -Ne -1.000000 -Nl
IFQ -Nw ARP -Ma 0 -Md 0 -Ms 4 -Mt 800 -Is 0.2 -Id 20971521.2 -It tcp -Il 40 -If 0 -Ii 171 -Iv
29 -Pn tcp -Ps 0 -Pa 0 -Pf 0 -Po 0
d -t 14.226013553 -Hs 9 -Hd 25165824 -Ni 9 -Nx 500.00 -Ny 500.00 -Nz 0.00 -Ne -1.000000 -Nl
IFQ -Nw ARP -Ma 0 -Md 0 -Ms 8 -Mt 800 -Is 0.4 -Id 37748737.2 -It tcp -Il 40 -If 0 -Ii 46 -Iv
29 -Pn tcp -Ps 0 -Pa 0 -Pf 0 -Po 0
d -t 14.226013553 -Hs 9 -Hd 25165824 -Ni 9 -Nx 500.00 -Ny 500.00 -Nz 0.00 -Ne -1.000000 -Nl
IFQ -Nw ARP -Ma 0 -Md 0 -Ms 8 -Mt 800 -Is 0.4 -Id 37748737.2 -It tcp -Il 40 -If 0 -Ii 173 -Iv
29 -Pn tcp -Ps 0 -Pa 0 -Pf 0 -Po 0
d -t 22.002160000 -Hs 2 -Hd -2 -Ni 2 -Nx 260.00 -Ny 20.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw
IFQ -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 29360128.0 -Id 8388610.0 -It udp -Il 48 -If 0 -Ii 12 -Iv 29
d -t 22.432343251 -Hs 5 -Hd 25165824 -Ni 5 -Nx 260.00 -Ny 260.00 -Nz 0.00 -Ne -1.000000 -Nl
RTR -Nw TTL -Ma 13a -Md 5 -Ms 4 -Mt 800 -Is 0.2 -Id 20971521.2 -It tcp -Il 40 -If 0 -Ii 3484
-Iv 0 -Pn tcp -Ps 0 -Pa 0 -Pf 10 -Po 0
d -t 24.440141852 -Hs 9 -Hd 25165824 -Ni 9 -Nx 500.00 -Ny 500.00 -Nz 0.00 -Ne -1.000000 -Nl
RTR -Nw TTL -Ma 13a -Md 5 -Ms 8 -Mt 800 -Is 0.4 -Id 37748737.2 -It tcp -Il 40 -If 0 -Ii 3486
-Iv 0 -Pn tcp -Ps 0 -Pa 0 -Pf 10 -Po 0
d -t 46.002160000 -Hs 2 -Hd -2 -Ni 2 -Nx 260.00 -Ny 20.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw
IFQ -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 20971520.0 -Id 8388610.0 -It udp -Il 48 -If 0 -Ii 13 -Iv 29
d -t 46.179826491 -Hs 5 -Hd 25165824 -Ni 5 -Nx 260.00 -Ny 260.00 -Nz 0.00 -Ne -1.000000 -Nl
RTR -Nw TTL -Ma 13a -Md 5 -Ms 4 -Mt 800 -Is 0.2 -Id 20971521.2 -It tcp -Il 40 -If 0 -Ii 7886
-Iv 0 -Pn tcp -Ps 0 -Pa 0 -Pf 10 -Po 0
d -t 46.249720282 -Hs 8 -Hd 20971520 -Ni 8 -Nx 260.00 -Ny 500.00 -Nz 0.00 -Ne -1.000000 -Nl
RTR -Nw TTL -Ma 13a -Md 4 -Ms 7 -Mt 800 -Is 0.3 -Id 33554433.2 -It tcp -Il 40 -If 0 -Ii 7887
-Iv 0 -Pn tcp -Ps 0 -Pa 0 -Pf 10 -Po 0
d -t 48.603418895 -Hs 9 -Hd 25165824 -Ni 9 -Nx 500.00 -Ny 500.00 -Nz 0.00 -Ne -1.000000 -Nl
RTR -Nw TTL -Ma 13a -Md 5 -Ms 8 -Mt 800 -Is 0.4 -Id 37748737.2 -It tcp -Il 40 -If 0 -Ii 7888
-Iv 0 -Pn tcp -Ps 0 -Pa 0 -Pf 10 -Po 0
d -t 61.412261795 -Hs 13 -Hd 37748738 -Ni 13 -Nx 370.00 -Ny 216.75 -Nz 0.00 -Ne -1.000000 -Nl
IFQ -Nw ARP -Ma 13a -Md c -Ms c -Mt 800 -Is 0.1 -Id 8388610.2 -It tcp -Il 40 -If 0 -Ii 43 -Iv
28 -Pn tcp -Ps 0 -Pa 0 -Pf 1 -Po 0
d -t 61.414128800 -Hs 13 -Hd 37748738 -Ni 13 -Nx 370.00 -Ny 216.75 -Nz 0.00 -Ne -1.000000 -Nl
IFQ -Nw ARP -Ma 13a -Md c -Ms c -Mt 800 -Is 0.1 -Id 8388610.2 -It tcp -Il 40 -If 0 -Ii 170 -
Iv 28 -Pn tcp -Ps 0 -Pa 0 -Pf 1 -Po 0
d -t 61.451242277 -Hs 13 -Hd 37748738 -Ni 13 -Nx 370.00 -Ny 216.78 -Nz 0.00 -Ne -1.000000 -Nl
RTR -Nw CBK -Ma 13a -Md c -Ms c -Mt 800 -Is 0.1 -Id 8388610.2 -It tcp -Il 40 -If 0 -Ii 1273 -
Iv 28 -Pn tcp -Ps 0 -Pa 0 -Pf 1 -Po 0

```

## N.3 - Trecho do arquivo wireless\_emc\_cilamce\_9nos.tr

```

d -t 0.744405687 -Hs 15 -Hd 37748736 -Ni 15 -Nx 490.00 -Ny 467.63 -Nz 0.00 -Ne -1.000000 -Nl
IFQ -Nw ARP -Ma 0 -Md 0 -Ms e -Mt 800 -Is 20971522.0 -Id 37748736.0 -It udp -Il 72 -If 0 -Ii
14 -Iv 32
d -t 0.744405879 -Hs 12 -Hd 37748736 -Ni 12 -Nx 410.07 -Ny 490.00 -Nz 0.00 -Ne -1.000000 -Nl
IFQ -Nw ARP -Ma 0 -Md 0 -Ms b -Mt 800 -Is 8388609.0 -Id 37748736.0 -It udp -Il 72 -If 0 -Ii
15 -Iv 32
d -t 1.022800000 -Hs 1 -Hd -2 -Ni 1 -Nx 20.00 -Ny 20.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw
IFQ -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 4194305.2 -It tcp -Il 1040 -If 0 -Ii 43 -Iv 30 -Pn
tcp -Ps 1 -Pa 0 -Pf 0 -Po 0
d -t 1.026960000 -Hs 1 -Hd -2 -Ni 1 -Nx 20.00 -Ny 20.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw
IFQ -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 4194305.2 -It tcp -Il 1040 -If 0 -Ii 55 -Iv 30 -Pn
tcp -Ps 1 -Pa 0 -Pf 0 -Po 0
d -t 1.031120000 -Hs 1 -Hd -2 -Ni 1 -Nx 20.00 -Ny 20.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw
IFQ -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 4194305.2 -It tcp -Il 1040 -If 0 -Ii 56 -Iv 30 -Pn
tcp -Ps 2 -Pa 0 -Pf 0 -Po 0
d -t 1.035280000 -Hs 1 -Hd -2 -Ni 1 -Nx 20.00 -Ny 20.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw
IFQ -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 4194305.2 -It tcp -Il 1040 -If 0 -Ii 57 -Iv 30 -Pn
tcp -Ps 3 -Pa 0 -Pf 0 -Po 0
d -t 1.039440000 -Hs 1 -Hd -2 -Ni 1 -Nx 20.00 -Ny 20.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw
IFQ -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 4194305.2 -It tcp -Il 1040 -If 0 -Ii 58 -Iv 30 -Pn
tcp -Ps 4 -Pa 0 -Pf 0 -Po 0
d -t 1.079686389 -Hs 1 -Hd -2 -Ni 1 -Nx 20.00 -Ny 20.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw
IFQ -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 4194305.2 -It tcp -Il 1040 -If 0 -Ii 59 -Iv 30 -Pn
tcp -Ps 5 -Pa 0 -Pf 0 -Po 0
d -t 1.083846389 -Hs 1 -Hd -2 -Ni 1 -Nx 20.00 -Ny 20.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw
IFQ -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 4194305.2 -It tcp -Il 1040 -If 0 -Ii 60 -Iv 30 -Pn
tcp -Ps 6 -Pa 0 -Pf 0 -Po 0
d -t 3.844667478 -Hs 12 -Hd 33554432 -Ni 12 -Nx 413.51 -Ny 490.00 -Nz 0.00 -Ne -1.000000 -Nl
RTR -Nw CBK -Ma 13a -Md 7 -Ms b -Mt 800 -Is 8388609.0 -Id 33554432.0 -It udp -Il 72 -If 0 -Ii
200 -Iv 32
d -t 3.888007283 -Hs 12 -Hd 33554432 -Ni 12 -Nx 413.55 -Ny 490.00 -Nz 0.00 -Ne -1.000000 -Nl
RTR -Nw CBK -Ma 13a -Md 7 -Ms b -Mt 800 -Is 8388609.0 -Id 33554432.0 -It udp -Il 72 -If 0 -Ii
205 -Iv 32
d -t 6.500000000 -Hs 14 -Hd 8388608 -Ni 14 -Nx 299.54 -Ny 130.00 -Nz 0.00 -Ne -1.000000 -Nl
IFQ -Nw ARP -Ma 0 -Md 0 -Ms d -Mt 800 -Is 20971521.0 -Id 8388608.0 -It udp -Il 72 -If 0 -Ii
615 -Iv 32
d -t 7.128849964 -Hs 14 -Hd 8388608 -Ni 14 -Nx 300.24 -Ny 130.00 -Nz 0.00 -Ne -1.000000 -Nl
IFQ -Nw ARP -Ma 0 -Md 0 -Ms d -Mt 800 -Is 20971521.0 -Id 8388608.0 -It udp -Il 72 -If 0 -Ii
711 -Iv 32
d -t 7.142590917 -Hs 14 -Hd 8388608 -Ni 14 -Nx 300.26 -Ny 130.00 -Nz 0.00 -Ne -1.000000 -Nl
RTR -Nw CBK -Ma 0 -Md 0 -Ms d -Mt 800 -Is 20971521.0 -Id 8388608.0 -It udp -Il 72 -If 0 -Ii
839 -Iv 32
d -t 7.142590917 -Hs 14 -Hd 8388608 -Ni 14 -Nx 300.26 -Ny 130.00 -Nz 0.00 -Ne -1.000000 -Nl
RTR -Nw CBK -Ma 0 -Md 0 -Ms d -Mt 800 -Is 20971521.2 -Id 0.4 -It ack -Il 60 -If 0 -Ii 844 -Iv
32 -Pn tcp -Ps 156 -Pa 0 -Pf 0 -Po 0

```