

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL  
FACULDADE DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

ALEXANDRE GADDO

**AVALIAÇÃO DE TRANSFERÊNCIA DE INFORMAÇÕES  
DE RASTREAMENTO VEICULAR SOBRE REDES GPRS**

Porto Alegre  
Dezembro de 2007

ALEXANDRE GADDO

**AVALIAÇÃO DE TRANSFERÊNCIA DE INFORMAÇÕES  
DE RASTREAMENTO VEICULAR SOBRE REDES GPRS**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre em Ciência da Computação pelo Programa de Pós-graduação em Ciência da Computação da Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador: Dr. Eduardo Augusto Bezerra

Porto Alegre  
Dezembro de 2007

## **Dados Internacionais de Catalogação na Publicação (CIP)**

G123a Gaddo, Alexandre  
Avaliação de transferência de informações de rastreamento  
veicular sobre redes GPRS / Alexandre Gaddo. – Porto Alegre,  
2008.  
102 f.  
  
Diss. (Mestrado) – Fac. de Informática, PUCRS  
  
Orientador: Prof. Dr. Eduardo Augusto Bezerra  
  
1. Informática. 2. Protocolos de Comunicação. 3. GPS.  
4. Protocolos de Aplicação sem Fio (Protocolos de Rede de  
Computação). 5. Simulação (Programação de Computadores).  
6. Transmissão de Dados (Computação). I. Bezerra, Eduardo  
Augusto. II. Título.  
  
CDD 004.62

**Ficha Catalográfica elaborada pelo**  
**Setor de Tratamento da Informação da BC-PUCRS**



Pontifícia Universidade Católica do Rio Grande do Sul  
FACULDADE DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

## TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "**Avaliação de Transferência de Informações de Rastreamento Veicular Sobre Redes GPRS**", apresentada por Alexandre Gaddo, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Sistemas Embarcados e Sistemas Digitais, aprovada em 25/03/08 pela Comissão Examinadora:

Prof. Dr. Eduardo Augusto Bezerra –  
Orientador

PPGCC/PUCRS

Prof. Dr. Fabiano Passuelo Hessel –

PPGCC/PUCRS

Profa. Dra. Ana Cristina Benso da Silva –

FACIN/PUCRS

Homologada em 02/06/09, conforme Ata No. 009 pela Comissão Coordenadora.

Prof. Dr. Fernando Gehm Moraes  
Coordenador.



PUCRS

**Campus Central**

Av. Ipiranga, 6681 – P32 – sala 507 – CEP: 90619-900  
Fone: (51) 3320-3611 – Fax (51) 3320-3621  
E-mail: [ppgcc@inf.pucrs.br](mailto:ppgcc@inf.pucrs.br)  
[www.pucrs.br/facin/pos](http://www.pucrs.br/facin/pos)

## LISTA DE FIGURAS

Figura 1 - Rede GSM.....	16
Figura 2 - Rede GPRS .....	18
Figura 3 - Estados de uma Estação Móvel.....	20
Figura 4 - Plano de Transmissão .....	21
Figura 5 - Plano de Sinalização.....	23
Figura 6 - Plano de Sinalização: SGSN-HLR, SGSN-EIR, SGSN-MSC/VLR.....	24
Figura 7 - Transferindo Pacotes de Dados Fim a Fim em uma Rede GPRS .....	25
Figura 8 - Portadoras de Frequência GSM, Duplexação e Quadros TDMA .....	28
Figura 9 - Divisão da Interface Aérea GPRS em Canais Físicos.....	29
Figura 10 - Mapeamento de Canais Lógicos em Canais Físicos .....	31
Figura 11 - Codificação de Pacotes de Dados GPRS.....	32
Figura 12 - Visão Geral do Sistema.....	38
Figura 13 - Formato do Cabeçalho TCP .....	47
Figura 14 - Formato do Cabeçalho UDP .....	48
Figura 15 - Arquitetura do NS .....	54
Figura 16 - Esquema de Utilização do NS .....	55
Figura 17 - Esquema de um <i>MobileNode</i> .....	56
Figura 18 - Pilha de Protocolos de Rede em um Nó Móvel no NS .....	59
Figura 19 - Pilha de Rede Modificada para um Nó Móvel GPRS .....	61
Figura 21 - Topologia Utilizada nas Simulações .....	76
Figura 22 - Vazão do Sistema nos Quatro Cenários de Codificação .....	79
Figura 23 - Percentual de Blocos Descartado na Transferência de Informações para os Quatro Cenários de Codificação .....	80
Figura 24 - Vazão para Diferentes Tipos de Pacotes de Rastreamento Veicular.....	81
Figura 25 - Tempo de Recuperação do Histórico de Rastreamento no Restabelecimento de Sinal GPRS .....	84
Figura 26 - Vazão para Diferentes Números de Estações Móveis.....	86
Figura 27 - Vazão para Diferentes Tipos de Pacotes de Rastreamento Veicular.....	85
Figura 28 - Vazão para Múltiplos <i>Timeslots</i> .....	86
Figura 29 - Vazão Variando o Modo de Transmissão RLC .....	87
Figura 30 - Vazão para Diferentes Números de Estações Móveis.....	88
Figura 31 - Vazão com Diferentes Tamanhos de Pacotes de Dados sobre UDP.....	89
Figura 32 - Variação da Taxa de Erros, Vazão e Tamanho dos Pacotes .....	89
Figura 33 - Variação da Taxa de Erros, Vazão e Número de Veículos .....	90
Figura 34 - Vazão TCP x UDP para CS-1 e CS-4 .....	91

## LISTA DE TABELAS

Tabela 1 - Cenários de Codificação .....	33
Tabela 2 - Posição e descrição dos bytes dos pacotes de informações básicas .....	39
Tabela 3 - Valores possíveis referente ao sinal GSM .....	40
Tabela 4 - Representação de cada bit da posição 31 .....	41
Tabela 5 - Posição e descrição dos bytes do pacote de informações avançadas.....	42
Tabela 6 - Valores possíveis referente ao sinal GSM .....	44
Tabela 7 - Representação de cada bit da posição 31 .....	44
Tabela 8 - BLER .....	69
Tabela 9 - Parâmetros de Simulação.....	77

## LISTA DE ABREVIATURAS E SIGLAS

2G	<i>Second Generation</i>
ACKs	<i>ACKnowledgement</i>
ARP	<i>Address Resolution Protocol</i>
ARQ	<i>Automatic Repeat Request</i>
ATMS	<i>Advanced Technology Management Systems</i>
AuC	<i>Authentication Center</i>
AVI	<i>Automatic Vehicle Identification</i>
BCH	<i>Broadcast Channel</i>
BCS	<i>Block Check Sequence</i>
BLER	<i>Block Error Rate</i>
BSC	<i>Base Station Controller</i>
BSN	<i>Block Sequence Number</i>
BSS	<i>Base Station System</i>
BSSAP+	<i>Base Station System Application Part</i>
BSSGP	<i>Base Station System GPRS Protocol</i>
BTS	<i>Base Transceiver Station</i>
CMS	<i>Congestion Management System</i>
CS	<i>Cenários de Codificação</i>
CWND	<i>Congestion Window</i>
DARPA	<i>Defense Advanced Research Projects Agency</i>
DSSS	<i>Lucent WaveLan Direct-Sequence Spread Spectrum</i>
EIR	<i>Equipment Identity Register</i>
ETSI	<i>European Telecommunications Standards Institute</i>
FCS	<i>Frame Check Sequence</i>
FDMA	<i>Frequency Division Multiple Access</i>
GMM/SM	<i>GPRS Mobility Management and Session Management</i>
GTP	<i>GPRS Tunneling Protocol</i>
GPRS	<i>General Packet Radio Service</i>
GPS	<i>Global Positioning System</i>
GSM	<i>Global System for Mobile Communication</i>
GSNs	<i>GPRS Support Nodes</i>
GGSN	<i>Gateway GPRS Support Nodes</i>
HLR	<i>Home Location Register</i>
IFQ	<i>Interface Queue</i>
IP	<i>Internet Protocol</i>
IPv4	<i>Internet Protocol version 4</i>
IMSI	<i>International Mobile Subscriber Identity</i>
IMEI	<i>International Mobile Equipment Identity</i>
LA	<i>Location Areas</i>
LAN	<i>Local Area Network</i>
LL	<i>Link Layer</i>
LLC	<i>Logical Link Control</i>
MAC	<i>Medium Access Control</i>
MAP	<i>Mobile Application Part</i>
MTP3	<i>Message Transfer Part Level 3</i>
MTP2	<i>Message Transfer Part Level 2</i>

MTU	<i>Maximum Transmission Unit</i>
MS	<i>Mobile Station</i>
MSC	<i>Mobile Switching Center</i>
NACK	<i>Not acknowledged</i>
NAM	<i>Network Animator</i>
NETIF	<i>Network Interface</i>
N-PDUs	<i>Network layer Packet Data Unit</i>
NS	<i>Network Simulator</i>
NSF	<i>National Science Foundation</i>
NOAH	<i>Non Adhoc Routing Agent</i>
OBD	<i>On-board Diagnostic</i>
OSI	<i>Open System Interconnection</i>
PAGCH	<i>Packet Access Grant Channel</i>
PACCH	<i>Packet Associated Control Channel</i>
PARC	<i>Palo Alto Research Center</i>
PBCH	<i>Packet Broadcast Channel</i>
PCC	<i>Packet Control Channels</i>
PCCCH	<i>Packet Common Control Channel</i>
PDP	<i>Packet Data Protocol</i>
PDN	<i>Packet Data Network</i>
PDCHs	<i>Packet Data Channels</i>
PDU	<i>Packet Data Units</i>
PLL	<i>Physical Link Layer</i>
PTC	<i>Packet Traffic Channels</i>
PRACH	<i>Packet Random Access Channel</i>
PTMSI	<i>Packet Temporary Mobile Subscriber Identity</i>
PPCH	<i>Packet Paging Channel</i>
PNCH	<i>Packet Notification Channel</i>
PBCCCH	<i>Packet Broadcast Control Channel</i>
PTCCH	<i>Packet Timing advance Control Channel</i>
PDTCH	<i>Packet Data Traffic Channel</i>
PT_IAT	<i>Pacote de Transmissão de Informações Avançadas de Telemetria</i>
PT_IBT	<i>Pacote de Transmissão de Informações Básicas de Telemetria</i>
QoS	<i>Quality of Service</i>
RA	<i>Routing Areas</i>
RFL	<i>Physical RF Layer</i>
RLC	<i>Radio Link Control</i>
RTO	<i>Retransmission Timeout</i>
SCCP	<i>Signaling Connection Control Part</i>
SAPI	<i>Service Access Point Interface</i>
SFQ	<i>Stochastic Fair Queuing</i>
SIM	<i>Subscriber Identity Module</i>
SIT	<i>Sistema para Inteligência no Transporte</i>
SGSN	<i>Serving GPRS Support Nodes</i>
SNDCP	<i>SubNetwork Dependent Convergence Protocol</i>
SS#7	<i>Signaling System no. 7</i>
SSTHRESH	<i>Slow Start Threshold</i>
TBF	<i>Temporary Block Flow</i>
TCAP	<i>Transactions Capabilities Application Part</i>
TCP	<i>Transfer Control Protocol</i>



TDMA	<i>Time Division Multiple Access</i>
TFI	<i>Temporary Flow Identity</i>
TLLI	<i>Temporary Logical Link ID</i>
TU	<i>Typical Urban Channel</i>
UDP	<i>User Datagram Protocol</i>
USF	<i>Uplink State Flag</i>
VANET	<i>Vehicular Adhoc Network</i>
VITP	<i>Vehicle Identification Transfer Protocol</i>
VLR	<i>Visitor Location Register</i>
WS	<i>Window Size</i>

## RESUMO

Devido ao avanço dos sistemas voltados à inteligência no transporte, novos protocolos e algoritmos estão sendo desenvolvidos para suprir a crescente demanda de tráfego de informações geradas por esses sistemas. O desenvolvimento de novas técnicas de comunicação tem aumentado o uso de dispositivos GPS/GSM em sistemas de monitoramento de veículos. Este tipo de tecnologia pode transmitir informações sobre rastreamento de veículos para centros de controle através de comunicação sem fio sobre redes GPRS. O objetivo deste trabalho é propor e avaliar um protocolo de transmissão de informações de rastreamento veicular sobre redes GPRS. Para isto, pacotes de informações de rastreamento veicular são descritos neste trabalho, métricas referentes à transmissão destes pacotes são investigadas em diversos cenários de comunicação utilizando o simulador de redes NS-2, tendo seus resultados analisados.

**Palavras-chave:** sistemas de rastreamento veicular; GSM GPRS; *network simulator*.

## **ABSTRACT**

Due to the advances in transport intelligence systems, new protocols and algorithms are being developed to supply the increasing bandwidth required by them. In addition, new communication techniques have increased the use of GPS/GPRS equipments in vehicle monitoring systems. This kind of technology is able to transmit vehicle tracking data to control centers using wireless communication over GPRS networks. The goal of the present work is to propose and evaluate a vehicle tracking protocol that works over GPRS. In order to achieve this objective, packages of vehicular tracking data are described in this study, and metrics regarding the transmission of these packages are analyzed in several scenarios of communication using the network simulator NS-2.

**Key words:** vehicular monitoring systems; GSM GPRS; network simulator.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
<b>2</b>	<b>GPRS</b>	<b>15</b>
2.1	GSM	15
2.1.1	Estação Móvel (MS – <i>Mobile Station</i> )	15
2.1.2	Sistema de Estação Base (BSS - <i>Base Station System</i> )	16
2.1.3	Subsistema de Rede (NS - <i>Network Subsystem</i> )	16
2.2	REDE GPRS	17
2.3	GERENCIAMENTO DE MOBILIDADE, DE SESSÃO E DE ROTEAMENTO	18
2.3.1	Procedimento GPRS <i>attach</i> e <i>detach</i>	19
2.3.2	Contexto PDP	19
2.3.3	Estados de Operação de uma MS	19
2.4	CLASSES DE MOBILIDADE GPRS/GSM	20
2.5	PROTOCOLOS DO GPRS	21
2.5.1	Plano de Transmissão	21
2.5.2	Plano de Sinalização	23
2.6	TRANSMITINDO OS PDUs ( <i>PACKET DATA UNITS</i> )	24
2.6.1	Transmissão no GGSN	25
2.6.2	Transmissão no SGSN	26
2.6.3	Transmissão na BSS	26
2.6.4	Transmissão através da Interface Aérea	27
2.7	INTERFACE AÉREA	27
2.7.1	Canais Físicos	28
2.7.2	Canais Lógicos	29
2.7.3	Mapeamento de Canais Lógicos sobre Canais Físicos	30
2.7.4	Cenários de Codificação	32
2.8	OPERAÇÕES DE TRANSFERÊNCIA DE PACOTES DE DADOS	33
2.8.1	Transferência no Sentido <i>Uplink</i>	33
2.8.2	Alocação de Blocos de Rádio	33
2.8.2.1	Dinâmica	34
2.8.2.2	Fixa	34
2.8.3	Transferência de Blocos de Dados RLC	34
2.8.4	Transferência no Sentido <i>Downlink</i>	35
2.9	CONSIDERAÇÕES FINAIS	35
<b>3</b>	<b>PROPOSTA DE PACOTES DE INFORMAÇÃO DE RASTREAMENTO VEICULAR</b>	<b>37</b>

3.1	VISÃO GERAL DO SISTEMA .....	37
3.2	FORMATO DOS PACOTES DE COMUNICAÇÃO .....	38
3.2.1	<b>PT_IBT: Pacote de Transmissão de Informações Básicas de Telemetria</b> .....	39
3.2.2	<b>PT_IAT: Pacote de Transmissão de Informações Avançadas de Telemetria</b> .....	41
3.3	PROCEDIMENTOS EM AUSÊNCIA DE SINAL GSM/GPS .....	45
3.4	PROTOCOLOS DE TRANSPORTE.....	45
3.4.1	<b>TCP – Transmission Control Protocol</b> .....	45
3.4.2	<b>UDP – User Datagram Protocol</b> .....	47
3.5	TRABALHOS RELACIONADOS .....	48
3.6	CONSIDERAÇÕES FINAIS .....	51
<b>4</b>	<b>GPRS NO NETWORK SIMULATOR .....</b>	<b>52</b>
4.1	NETWORK SIMULATOR .....	52
4.1.1	<b>Estrutura do NS</b> .....	<b>53</b>
4.1.2	<b>O Modelo Wireless no NS</b> .....	<b>55</b>
4.1.2.1	A pilha de rede em um nó wireless no NS .....	57
4.1.2.2	Agentes de roteamento .....	58
4.1.2.3	Cenários de rede fixa sem fio .....	58
4.1.2.4	Um agente de roteamento não ad-hoc: NOAH .....	58
4.2	MÓDULO GPRS .....	59
4.2.1	<b>Estrutura do Módulo</b> .....	<b>59</b>
4.2.1.1	<i>Link Layer (LL)</i> .....	61
4.2.1.2	<i>Radio Link Control (RLC)</i> .....	62
4.2.1.3	<i>Medium Access Control (MAC)</i> .....	62
4.2.1.3.1	Canais .....	62
4.2.1.3.2	Estrutura dos slots TDMA.....	63
4.2.1.3.3	Alocação de canais .....	63
4.2.1.3.4	Modelo de erros.....	63
4.2.2	<b>Dificuldades Iniciais</b> .....	<b>64</b>
4.2.3	<b>Limitações e Problemas do Módulo</b> .....	<b>64</b>
4.3	MODIFICAÇÕES REALIZADAS.....	66
4.3.1	<b>Correção da Vazão</b> .....	<b>66</b>
4.3.2	<b>Robustez no Mecanismo de Alocação de Canais</b> .....	<b>66</b>
4.3.3	<b>Eliminação de Vazamento de Memória</b> .....	<b>68</b>
4.3.4	<b>Melhoramento no Modelo de Erros</b> .....	<b>69</b>
4.3.5	<b>Separação entre MS e BS</b> .....	<b>70</b>
4.3.6	<b>Divisão da Fila de Pacotes da BS</b> .....	<b>71</b>
4.3.7	<b>Transmissão dos Blocos RLC de Acordo com a Especificação</b> .....	<b>71</b>
4.3.8	<b>Correção do Modo com Confirmação da Camada RLC</b> .....	<b>71</b>
4.3.9	<b>Limitação dos Buffers</b> .....	<b>72</b>
4.3.10	<b>Redução do Uso de Timers</b> .....	<b>73</b>
4.4	ADIÇÕES REALIZADAS .....	73
4.4.1	<b>Implementação dos Cenários de Codificação</b> .....	<b>74</b>
4.4.2	<b>Implementação de Multislots</b> .....	<b>74</b>
4.5	CONSIDERAÇÕES FINAIS .....	74
<b>5</b>	<b>AVALIAÇÃO DE DESEMPENHO .....</b>	<b>76</b>
5.1	CONFIGURAÇÕES DE SIMULAÇÃO .....	76
5.2	RESULTADOS .....	78
5.2.1	<b>Transmissão de Informações de Rastreamento Veicular sobre GPRS Utilizando o Protocolo TCP</b> .....	<b>78</b>

5.2.1.1	Tráfego de Retaguarda ( <i>Background</i> ) .....	79
5.2.1.2	C/S Adequado para cada C/I .....	79
5.2.1.3	Tipo de Pacotes de Rastreamento Veicular sobre GPRS .....	81
5.2.1.4	Tempo de Recuperação do Histórico de Rastreamento no Restabelecimento do Sinal GPS/GPRS .....	83
5.2.1.5	Número de Estações Móveis (Veículos) .....	84
5.2.1.6	<i>Multislot</i> .....	85
5.2.1.7	Modo de Transmissão RLC.....	86
<b>5.2.2</b>	<b>Transmissão de Informações de Rastreamento Veicular sobre GPRS/UDP87</b>	
5.2.2.1	Numero de Estações Móveis (veículos) .....	88
5.2.2.2	Variação na Relação Sinal / Ruído.....	89
5.2.2.3	TCP x UDP para os Cenários de Codificação .....	90
<b>5.3</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>91</b>
<b>6</b>	<b>CONCLUSÃO .....</b>	<b>93</b>
	<b>REFERÊNCIAS.....</b>	<b>95</b>
	<b>APÊNDICE A – ALGORITMO DE COMPACTAÇÃO DE DATA E HORA NA LINGUAGEM DE PROGRAMAÇÃO “C” .....</b>	<b>98</b>
	<b>APÊNDICE B – ALGORITMO DE VERIFICAÇÃO E CONTROLE DE ERROS NA LINGUAGEM DE PROGRAMAÇÃO “C” .....</b>	<b>101</b>

## 1 INTRODUÇÃO

O volume de veículos nas ruas vem crescendo aceleradamente ao longo das últimas décadas, assim como os congestionamentos, número de mortes por acidentes de trânsito e o número de veículos furtados, um dado que preocupa diversas empresas e organizações como transportadoras, seguradoras, órgãos de fiscalização, policiamento e gerência de tráfego.

Sistemas para inteligência no transporte (SIT) visam aplicar tecnologia da informação, comunicação e rastreamento aliadas à Internet para melhorar a segurança, confiabilidade, conveniência e mobilidade nos sistemas voltados à área de transporte. Sistemas SIT incluem uma gama variada de aplicações e serviços como sistemas inteligentes de controle de tráfego, tecnologias *in-vehicle*, localização e segurança no tráfego, sistemas de auxílio em viagens, entre outros [1].

Devido ao avanço deste tipo de sistemas, novos protocolos e algoritmos estão sendo desenvolvidos para suprir a crescente demanda de tráfego de informações geradas por esses sistemas, principalmente para extração de informações de transporte usando receptores de sistemas de posicionamento global (GPS) integrados com tecnologias de telefonia móvel celular (GSM/GPRS) e o uso da Internet. É no método ou modo de comunicação que se concentra uma grande área de pesquisa neste tipo de sistema.

O desenvolvimento de técnicas de comunicação móvel tem aumentado o uso de dispositivos GPS/GSM em sistemas de monitoramento de veículos. Este tipo de tecnologia pode transmitir informações GPS para centros de controle através de comunicação sem fio sobre redes GPRS. GPRS é caracterizado como uma rede “sempre conectada” (uma vez conectado, sempre *online*), com baixa taxa (medido pelo tráfego de dados), e fácil conexão (conecta diretamente com a Internet) constituindo em uma rede cuja velocidade pode chegar a 171.2kbps [2].

Simulação é uma ferramenta importante para que de uma forma rápida e com pouco custo seja possível explorar o comportamento de novos protocolos de comunicação e transmissão de dados.

O objetivo geral deste trabalho é avaliar a transmissão de informações de rastreamento veicular sobre redes GPRS através de simulações. Para isto, são definidos formatos de pacotes de dados contendo informações sobre posicionamento de veículos equipados com dispositivos GPS, seu comportamento sobre a rede GPRS é estudado utilizando como base os protocolos de transporte TCP e UDP, bastante difundidos na Internet.

Os protocolos, TCP, UDP e GPRS apresentam algoritmos e comportamentos complexos que, quando utilizados em conjunto, podem gerar resultados inesperados tornando-se, muitas vezes, difícil analisar os seus resultados. Assim, é necessário explorar diferentes cenários com intuito de detectar e estudar casos deste tipo.

Vários cenários distintos são avaliados neste trabalho, incluindo diferentes taxas de erros do meio físico de transmissão, tamanho de pacotes, quantidade de usuários e de *timeslots* utilizados por usuário no sentido *uplink*. Além disso, um novo modelo de erros do meio físico [3] é introduzido no pacote de comunicação GPRS, tornando os resultados das simulações em cima deste modelo mais condizentes com a realidade.

O restante desta dissertação encontra-se organizado da seguinte forma. No capítulo 2 são descritos as principais características e os conceitos relacionados ao GPRS, tais como a rede GSM e os componentes adicionados a esta para formar a rede GPRS. Por último, os protocolos e a interface aérea do GPRS são descritos e algumas considerações apresentadas.

No Capítulo 3 são apresentados os trabalhos relacionados com o estudo e interação dos protocolos TCP, UDP e GPRS no meio de comunicação sem fio, bem como outros trabalhos que estudam a troca de informações entre veículos (*in vehicle*) trafegadas sobre redes sem fio utilizando este tipo de tecnologia.

No Capítulo 4, os pacotes de rastreamento veicular utilizados como base para simulação da transferência de informações sobre redes GPRS são descritos, sendo detalhado seu formato e as informações contidas dentro de cada pacote. Uma breve descrição sobre os protocolos de transporte TCP e UDP são apresentados, protocolos sobre o qual os pacotes de rastreamento serão transportados sobre a rede GPRS.

O Capítulo 5 apresenta o ambiente de simulação utilizado neste trabalho, o *NetWork Simulator*, descrevendo suas principais características e conceitos. O módulo GPRS, implementado e instalado à parte, é descrito detalhadamente, sendo apresentado seus problemas e limitações, bem como as modificações realizadas para superá-los.



No Capítulo 6, os aspectos relacionados às simulações são apresentados, sendo descrita a topologia, os parâmetros utilizados nas simulações e as métricas coletadas. Os resultados obtidos são apresentados e discutidos detalhadamente.

Por último, as conclusões, contribuições e trabalhos futuros são apresentados no Capítulo 7.

## 2 GPRS

Esse capítulo descreve as principais características e conceitos relacionados ao GPRS (*General Packet Radio Service*) [7], tais como a rede GSM (*Global System for Mobile communication*) e os componentes adicionados a esta para formar a rede GPRS. Também são descritos os protocolos, a interface aérea, as operações de transferência dos pacotes de dados, e as vantagens e desvantagens do GPRS.

### 2.1 GSM

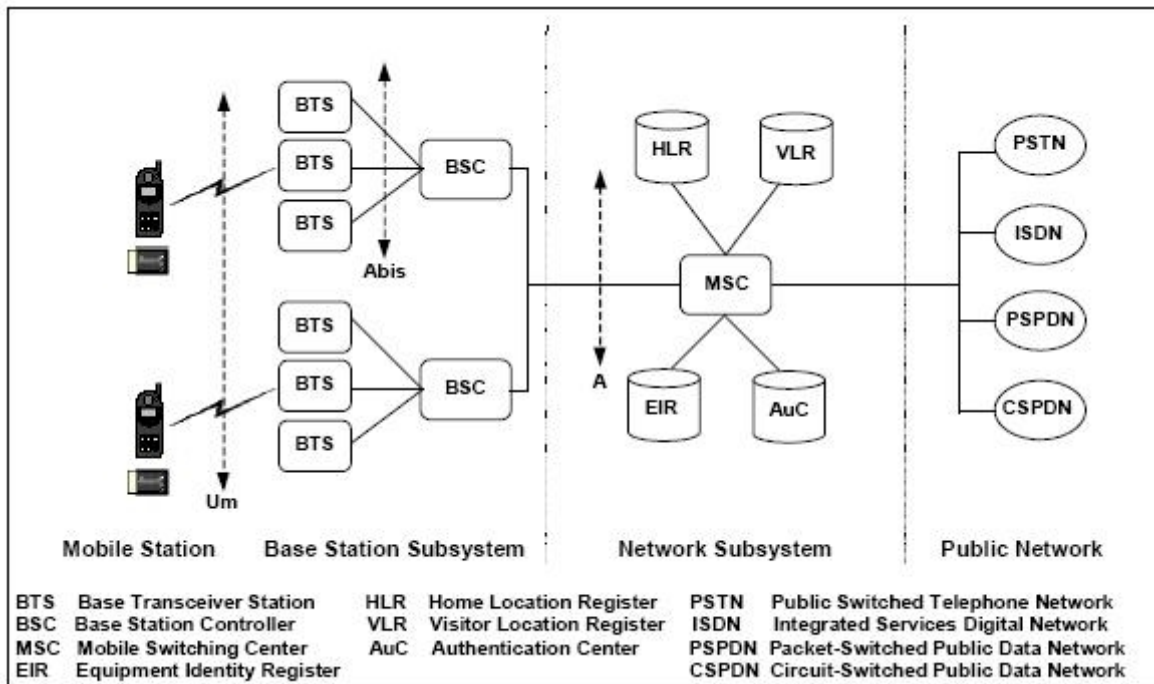
O GSM [5] consiste em um padrão desenvolvido pela ETSI (*European Telecommunications Standards Institute*) na tentativa de uniformizar os sistemas celulares europeus existentes, os quais divergiam bastante, impossibilitando a compatibilidade entre os mesmos. O GSM provê transferência de voz e de dados através de comutação por circuito. Para tanto, a rede GSM é composta por várias entidades, as quais estão agrupadas em três subsistemas: MS (*Mobile Station*) ou estação móvel, BSS (*Base Station System*) e NS (*Network Subsystem*). Na Figura 1 pode ser observada uma rede GSM, cujos elementos serão descritos resumidamente a seguir.

#### 2.1.1 Estação Móvel (MS – *Mobile Station*)

A MS consiste em um equipamento móvel (ou terminal) e um *smart card* chamado SIM (*Subscriber Identity Module*). O SIM provê ao usuário acesso a serviços específicos, de acordo com o perfil do usuário, independente do terminal utilizado. Em outras palavras, para fazer chamadas e ter acesso aos serviços disponíveis, o usuário precisa apenas inserir seu cartão em um terminal. Para isso, o SIM armazena, entre outras informações, uma identidade

chamada IMSI (*International Mobile Subscriber Identity*), que identifica o usuário no sistema, e uma chave secreta para autenticação (ou certificado).

Para identificar de forma única o terminal, há uma identidade denominada IMEI (*International Mobile Equipment Identity*).



**Figura 1: Rede GSM**

### 2.1.2 Sistema de Estação Base (BSS - *Base Station System*)

O BSS é formado por dois tipos de componentes: BSC (*Base Station Controller*) e BTS (*Base Transceiver Station*). A BTS contém os transceptores que definem uma célula e controla os protocolos de links de rádio com a MS. O BSC gerencia os recursos de rádio para uma ou mais BTSs, sendo responsável pelo estabelecimento de chamadas, saltos de frequência (*frequency hopping*) e *handovers*. O BSC interconecta a estação móvel (MS) e o MSC (*Mobile Switching Center*).

### 2.1.3 Subsistema de Rede (NS - *Network Subsystem*)

O principal componente do NS é o MSC (*Mobile Switching Center*). Este funciona como um *switch*, decidindo qual caminho o tráfego de informações deve seguir, e provê

funções necessárias para a “manipulação” de um usuário, como registro, autenticação e atualização de localização. Para prover tais funções, o MSC trabalha em conjunto com outros componentes: HLR, VLR, AuC e EIR.

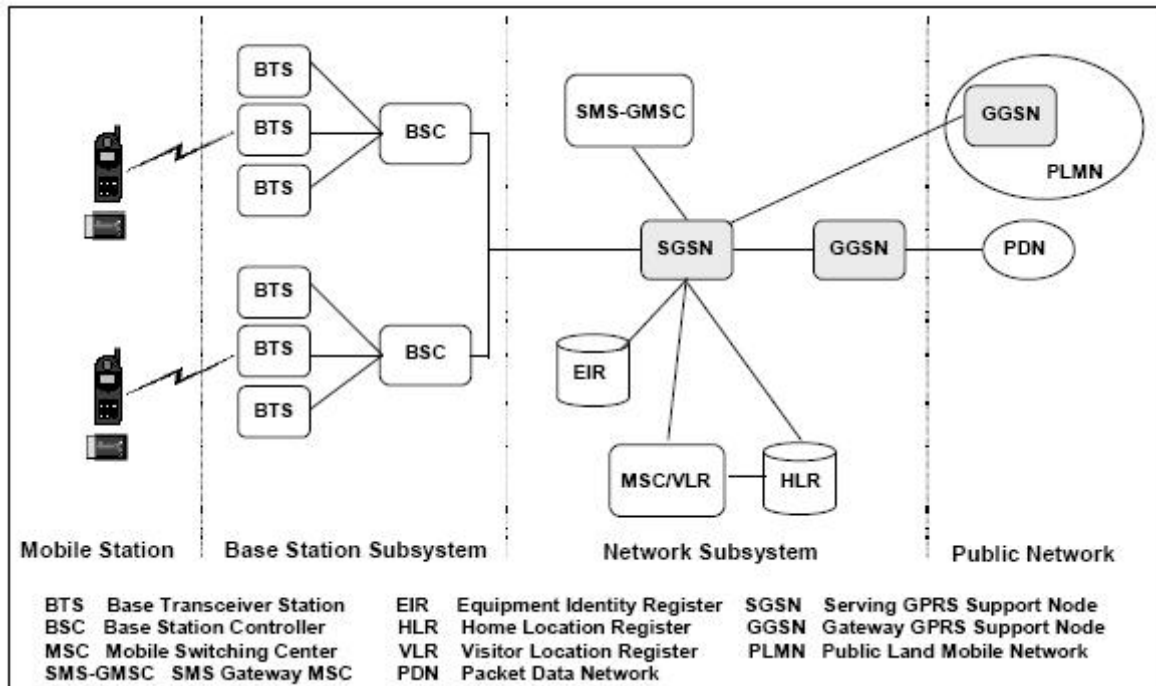
O HLR (*Home Location Register*) armazena dados permanentes e temporários (voláteis) sobre o usuário. Os dados permanentes consistem em informações sobre o perfil do usuário, que inclui desde dados pessoais até serviços aos quais o assinante tem direito, enquanto os dados temporários são informações sobre a localização do usuário. O VLR (*Visitor Location Register*) contém dados necessários para o controle de chamadas e provisão de serviços ao usuário que se encontra na área geográfica controlada por ele. Tais dados incluem parte dos dados permanentes contidos no HLR bem como dados locais, como uma identificação temporária.

O AuC (*Authentication Center*) guarda dados relacionados à segurança, como as chaves utilizadas na criptografia e na autenticação (certificados). O EIR (*Equipment Identity Register*) armazena dados sobre o equipamento móvel. Contém uma lista de todos os equipamentos válidos na rede, na qual cada estação móvel é identificada pela sua IMEI.

## 2.2 REDE GPRS

Além dos elementos que compõem a rede GSM, a arquitetura da rede GPRS (Figura 2) apresenta novos componentes: os GPRS *Support Nodes* (GSNs). Esses componentes são responsáveis pela entrega e roteamento de pacotes de dados entre as estações móveis e as redes de pacotes de dados (PDN – *Packet Data Network*) externas.

Há dois tipos de GSNs, *Serving GPRS Support Nodes* (SGSN) e *Gateway GPRS Support Nodes* (GGSN). O primeiro é responsável pela entrega de pacotes de e para as estações móveis dentro de sua área de serviço. Suas tarefas incluem roteamento e transferência de pacotes, gerenciamento de mobilidade, gerenciamento de link lógico, e funções de autenticação e cobrança. O segundo, GGSN, funciona como uma interface entre a rede backbone GPRS e as redes de pacotes de dados externas. O GGSN converte os pacotes GPRS vindos do SGSN no formato apropriado do protocolo de pacotes de dados (PDP – *Packet Data Protocol*) – por exemplo, IP – e manda os pacotes para a rede externa correspondente. Analogamente, na direção inversa endereços PDP de pacotes de dados que chegam são convertidos para o endereço GSM do destinatário. Os pacotes “convertidos” são mandados para o SGSN responsável. Assim, o GGSN armazena em seu registro de localização o endereço do SGSN corrente e o perfil do usuário.



**Figura 2 - Rede GPRS**

Em geral, há relação muitos-para-muitos entre os SGSNs e os GGSNs: um GGSN é a interface para redes de pacotes de dados externas para vários SGSNs; um SGSN pode rotear seus pacotes por diferentes GGSNs para alcançar diferentes redes de pacotes de dados.

### 2.3 GERENCIAMENTO DE MOBILIDADE, DE SESSÃO E DE ROTEAMENTO

Conhecer a localização da estação móvel é bastante importante para a rede GPRS, para que esta possa encaminhar pacotes destinados à MS. Assim, torna-se necessário gerenciar a mobilidade dos terminais, mantendo atualizada a localização destes. Contudo, se a cada “passo” a MS informar sua localização, haverá muita sinalização. Para equilibrar essa relação (isto é, manter a localização da MS atualizada sem que haja excesso de sinalização), uma rede GSM é dividida em áreas de localização (LA – *Location Areas*). Estas são divididas em áreas de roteamento (RA – *Routing Areas*), que, por sua vez, encontram-se divididas em células.

A frequência e a precisão com que a MS informa sua localização depende do estado em que se encontra, os quais serão explicados a seguir (Seção 2.3.3). No estado *idle*, a estação móvel é inacessível, isto é, não é possível realizar *paging* para essa unidade. No estado *ready*, o terminal informa ao SGSN sempre que muda de célula; enquanto no estado *standby* a MS informa apenas sua RA. Ao receber um pacote destinado à estação, a rede realiza um *paging* para saber a célula em que a MS se encontra.

### 2.3.1 Procedimento GPRS *attach* e *detach*

Para ter acesso aos serviços GPRS, uma estação móvel deve antes se registrar em um SGSN de uma rede GPRS. Isso é feito através de um procedimento chamado GPRS *attach*. Nesse procedimento, a rede checa se o usuário é autorizado, copia o perfil do usuário do HLR para o SGSN e determina uma identidade temporária (PTMSI – *Packet Temporary Mobile Subscriber Identity*) ao usuário. O móvel informa se deseja ser registrado como um dispositivo GPRS, um dispositivo GSM ou ambos.

O procedimento de desconexão de uma rede GPRS é denominado GPRS *detach* e pode ser iniciado pelo móvel ou pela rede (SGSN ou HLR).

### 2.3.2 Contexto PDP

Um procedimento de *attach* não habilita a estação a transmitir e receber dados. Para isso, a MS precisa ativar uma sessão de comunicação através de um contexto PDP (*Packet Data Protocol*). Em outras palavras, a estação móvel necessita de um ou mais endereços usados na rede de pacotes de dados externa com a qual ela deseja se comunicar, para que seja possível a troca de pacotes.

Um contexto PDP é formado pelo tipo do PDP (por exemplo, IPv4), o endereço PDP associado à estação móvel (por exemplo, 149.158.176.42), a qualidade de serviço (QoS) requerida e o endereço do GGSN que serve de ponto de acesso à rede de dados (PDN). Para cada sessão é criado um contexto PDP, o qual é armazenado na MS, no SGSN e no GGSN [6].

### 2.3.3 Estados de Operação de uma MS

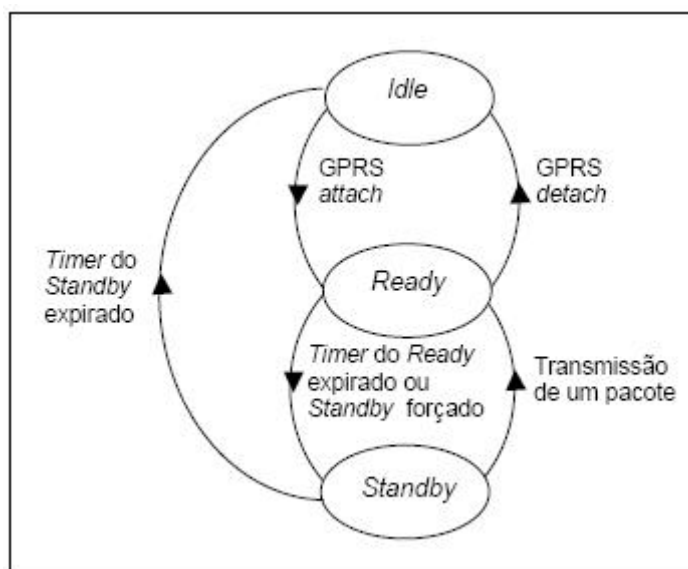
Em um determinado momento, uma estação móvel pode estar em um dos seguintes estados:

GPRS *idle* – nesse estado, a MS não está conectada a nenhuma rede GPRS; para transmitir e/ou receber dados é preciso realizar um *attach* a uma rede GPRS, mudando para o estado *ready*.

GPRS *ready* – o terminal passa para esse estado quando realiza um *attach*, registrando-se a uma rede GPRS. Nesse estado, a MS pode estabelecer um contexto PDP, a fim de se comunicar com uma rede externa (PDN). Quando um contexto é ativado, recursos

são reservados para a sessão até que a transferência de dados pare por um determinado tempo, então os recursos são liberados, todos os contextos PDP são apagados e a MS entra no estado *standby*.

GPRS *standby* – nesse estado, o móvel continua registrado a uma rede GPRS, mas não possui contexto PDP, logo não é possível a transferência de dados. Quando a rede recebe um novo pacote destinado à MS, é realizado um *paging* para localizar a estação móvel e esta ativa um novo contexto PDP, voltando ao estado *ready*.



**Figura 3 - Estados de uma Estação Móvel**

A Figura 3 ilustra resumidamente a mudança entre estados de uma MS.

#### 2.4 CLASSES DE MOBILIDADE GPRS/GSM

Em redes híbridas GSM/GPRS, os dispositivos móveis podem ser distribuídos em três classes [4]:

Classe A (GSM/GPRS): o dispositivo móvel pode estar conectado a uma rede GPRS e a uma GSM ao mesmo tempo, podendo fazer e receber chamadas GPRS e GSM ao mesmo tempo.

Classe B (GSM/GPRS): essa classe é similar à classe A, mas nesse caso o móvel não suportará chamadas simultâneas, isto é, se estiver realizando uma chamada GPRS, não poderá fazer/receber uma chamada GSM e vice-versa.

Classe C (GSM ou GPRS): o dispositivo móvel nessa classe possui ambas funcionalidades – GSM e GPRS – mas só poderá se vincular a uma rede por vez.

## 2.5 PROTOCOLOS DO GPRS

A especificação do GPRS define protocolos para o plano de transmissão e para o plano de sinalização.

### 2.5.1 Plano de Transmissão

Os protocolos que constituem o plano de transmissão podem ser observados na Figura 4. Começando pela direita, temos a interface Gi, a qual possibilita a comunicação entre redes GPRS e redes IP. Para uma rede IP externa, a rede GPRS parece uma sub-rede IP comum.

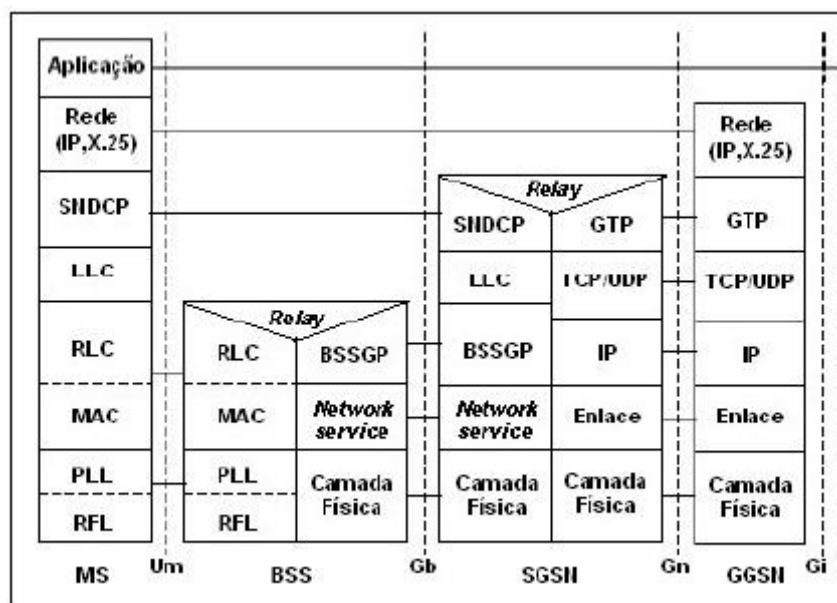


Figura 4 - Plano de Transmissão

A comunicação SGSN-GGSN é realizada através da interface Gn, a qual possui os seguintes protocolos:

O GTP (GPRS *Tunneling Protocol*) é responsável pelos pacotes de dados do usuário e as informações de sinalização relacionadas a estes entre os GSNs. Em outras palavras, o GTP recebe datagramas IP e cria túneis (rotas válidas apenas na rede GPRS, invisíveis a redes



externas) entre os GSNs. Vale ressaltar que esse protocolo atua não apenas na interface Gn (entre SGSN e GGSN) mas também na interface Gp (entre GGSNs).

Os protocolos TCP (*Transmission Control Protocol*) e UDP (*User Datagram Protocol*) transportam os pacotes GTP dentro do *backbone* da rede. O TCP é utilizado quando se necessita de confiabilidade na transmissão, isto é, quando é importante que não haja perdas de dados. Utiliza-se o UDP quando a perda de pacotes não consiste em um grande problema. Assim, o TCP é usado para dados do usuário que precisam de confiabilidade e o UDP, para sinalização e dados que não precisam de confiabilidade (como aplicações multimídia).

O IP (*Internet Protocol*) é empregado na camada de rede, para prover roteamento dos pacotes entre os GSNs.

A interface Gb promove a comunicação SGSN-BSS e SGSN-MS utilizando os protocolos abaixo:

O SNDCP (*SubNetwork Dependent Convergence Protocol*) é usado para transferir pacotes de dados entre o SGSN e a MS. Esse protocolo converte pacotes da camada de rede (N-PDUs – *Network layer Packet Data Unit*) em um formato adequado à arquitetura de rede GPRS.

O LLC (*Logical Link Control*) provê um link lógico bastante confiável entre a MS e o SGSN associado a ela. Suas funcionalidades incluem controle de seqüência, entrega em ordem, detecção e correção de erros, e retransmissão (ARQ – *Automatic Repeat reQuest*). Possui os modos de transmissão de quadros com e sem confirmação (*acknowledged* e *unacknowledged*, respectivamente).

O BSSGP (*Base Station System GPRS Protocol*) transporta entre o SGSN e o BSS informações relacionadas ao roteamento e a QoS. Esse protocolo opera sobre *Frame Relay*.

O *Network Service* mapeia os serviços solicitados pelo BSSGP para serviços *Frame Relay* apropriados.

A interface Um é responsável pela comunicação entre a MS e o BSS e é formada pelos seguintes protocolos:

O RLC (*Radio Link Control*) tem como principal objetivo estabelecer um link confiável entre a MS e o BSS. Entre suas funções encontram-se fragmentação e desfragmentação dos quadros LLC em blocos de dados RLC, e correção de erros através de um mecanismo de retransmissão seletiva desses blocos.

O MAC (*Medium Access Control*) controla o acesso aos canais de rádio, ou seja, determina como os pares frequência/*timeslot* são compartilhados pelas MSs. Baseado no *slotted ALOHA* [7], o MAC realiza resolução de contenção entre as tentativas de acesso ao

canal, multiplexação de vários usuários em um mesmo canal lógico (PDTCH), e escalonamento e priorização baseado no QoS negociado.

A camada física divide-se em duas subcamadas: PLL (*Physical Link Layer*) e RFL (*Physical RF Layer*). A PLL provê um canal físico entre a MS e o BSS. Entre suas funções estão codificação dos dados e detecção e correção de erros de transmissão no meio físico. A RFL realiza modulação/demodulação das ondas físicas.

### 2.5.2 Plano de Sinalização

O plano de sinalização engloba protocolos para controle e suporte das funções do plano de transmissão, como os procedimentos de *attach* e *detach*.

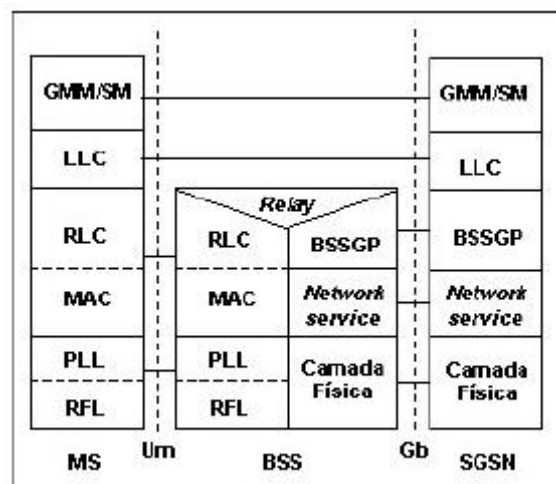
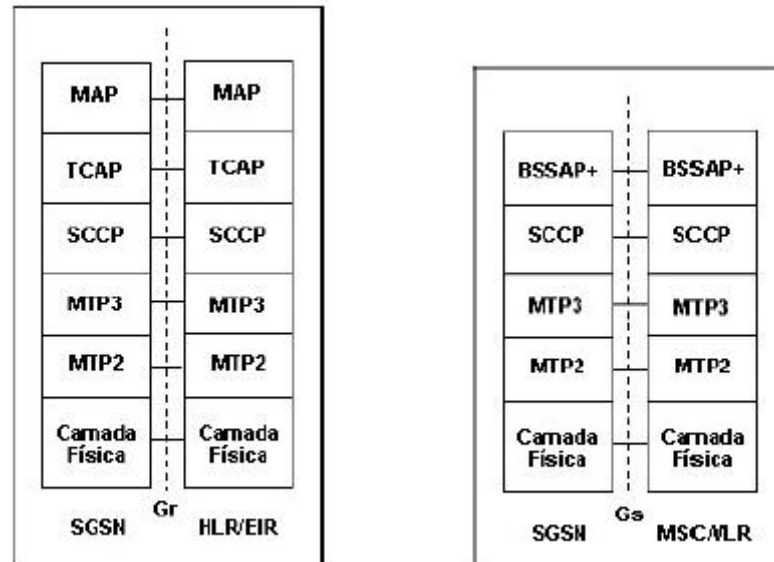


Figura 5 - Plano de Sinalização: MS – SGSN

Entre a MS e o SGSN (Figura 5), o protocolo GMM/SM (*GPRS Mobility Management and Session Management*) provê suporte ao gerenciamento de mobilidade e de sessão na realização de funções como *GPRS attach/detach* e ativação do contexto PDP.

A arquitetura de sinalização entre o SGSN e as bases de dados HLR, VLR e EIR, ilustrada na Figura 6, usa os mesmos protocolos do GSM e os incrementa com funcionalidades específicas do GPRS. Entre o SGSN e o HLR, assim como entre o SGSN e o EIR, um MAP (*Mobile Application Part*) aperfeiçoado é empregado. O MAP transporta informações de sinalização relacionadas a atualizações de localização, informações de roteamento, perfis de usuários e *handovers*. A troca de mensagens MAP é realizada sobre o TCAP (*Transactions Capabilities Application Part*) e o SCCP (*Signaling*

*Connection Control Part*), os quais fazem parte da pilha de protocolos do SS#7 (*Signaling System no. 7*) [8] juntamente com o MTP3 (*Message Transfer Part Level 3*) e MTP2 (*Message Transfer Part Level 2*).

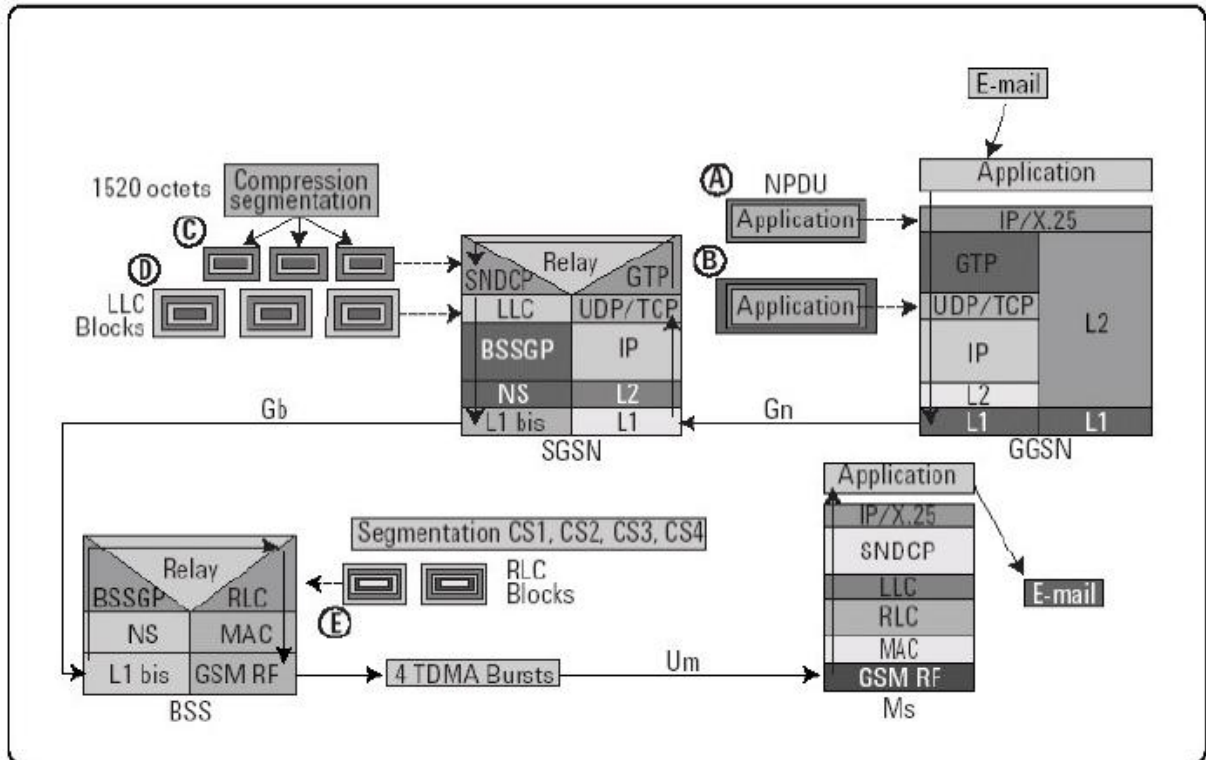


**Figura 6 - Plano de Sinalização: SGSN-HLR, SGSN-EIR e SGSN-MSC/VLR**

O BSSAP+ (*Base Station System Application Part*) abrange as funções descritas na especificação do GSM e é usado para transferir informações de sinalização entre o SGSN e o VLR. Isso inclui sinalização de gerenciamento de mobilidade quando é necessária a coordenação entre funções GPRS e funções convencionais GSM (por exemplo, *paging* de uma estação móvel via GPRS para uma chamada GSM recebida).

## 2.6 TRANSMITINDO OS PDUs (*PACKET DATA UNITS*)

Uma vez definidos os protocolos utilizados na transmissão de dados, será mostrado agora um cenário de um pacote sendo transmitido fim a fim em uma rede GPRS.



**Figura 7 - Transferindo Pacotes de Dados Fim a Fim em uma Rede GPRS [24]**

A Figura 7 ilustra o caminho percorrido por um “short e-mail” enviado de um computador para um telefone móvel GPRS. O dado tem que “viajar” pela Internet até chegar no GGSN, prosseguindo para o SGSN, depois para a BSS e finalmente chegando à MS. Em todos os elementos pelos quais o dado passa, a cada camada ele sofre um processamento, que normalmente envolve segmentação e inclusão de novo cabeçalho. Para simplificar, apenas alguns desses processamentos são mostrados na figura, sendo indicados pelas letras de A até E.

### 2.6.1 Transmissão no GGSN

A camada de aplicação (isto é, o e-mail do computador) gera um datagrama IP e o envia através da rede de comunicação externa (IP) para o GGSN. Ao chegar no GGSN, o datagrama IP é chamado N-PDU (*Network Packet Data Unit*) e recebe um endereço IMSI ou IP.

Na camada IP/X.25, o NPDU recebe um cabeçalho IP, como ilustrado na letra A da Figura 7. Em seguida, o dado é passado através da camada GTP, que adiciona um cabeçalho GTP para diferenciar o N-PDU de uma mensagem GTP. O dado segue adiante para a camada

UDP/TCP, a qual insere seu próprio cabeçalho – UDP ou TCP – baseado na classe QoS (letra B no diagrama). Esse cabeçalho contém o endereço da porta de origem e de destino, informações sobre roteamento, e (no caso do TCP) controle de fluxo.

Encaminhado para a camada IP, o N-PDU recebe o endereço da origem e do GSN de destino (no caso, SGSN). Dependendo do tamanho do N-PDU e da unidade máxima de transmissão (MTU – *Maximum Transmission Unit*) da camada IP, é necessária a fragmentação do N-PDU.

Finalmente, o N-PDU, com todos os cabeçalhos adicionados, é transportado através da camada física da interface Gn para o SGSN.

### 2.6.2 Transmissão no SGSN

No SGSN, os cabeçalhos são retirados e o N-PDU é conduzido para a camada SNDCP. Esta irá comprimir (opcionalmente) e segmentar o pacote para atingir os 1520 octetos requeridos (MTU) para a transmissão através da camada de serviços de rede *Frame Relay* na interface Gb. Primeiro o SNDCP classificará o N-PDU como SN-DATA PDU orientado à conexão ou SN-UNITDATA PDU não orientado à conexão. Depois é adicionado um cabeçalho SNDCP contendo informações sobre compressão e segmentação, como visualizado na letra C da Figura 7.

Na camada LLC, o SN-DATA PDU ou o SN-DATAUNIT PDU é encapsulado em um quadro LLC, denominado bloco LLC, com seu próprio cabeçalho (letra D do diagrama). Este contém informações de controle (usados no modo de transmissão com confirmação), *frame check sequence* (FCS) e valores de SAPI (*Service Access Point Interface*).

O bloco LLC é enviado para a camada BSSGP, a qual provê informações de roteamento para a camada de serviços de rede. O BSSGP também adiciona um cabeçalho, que contém informações essenciais às camadas RLC/MAC da interface aérea - relacionadas à transmissão de bloco, como prioridade, TLLI (*Temporary Logical Link ID*), etc.

### 2.6.3 Transmissão na BSS

O dado é enviado através da interface Gb entre o SGSN e o BSS. No BSS, o BSSGP envia todas as informações para a RLC. Esta segmenta os blocos LLC em blocos menores, denominados blocos RLC. A letra E da Figura 7 ilustra esse processo. Um grupo de blocos LLC, agora segmentados em blocos menores, é conhecido como um TBF (*Temporary Block*

*Flow*). Para cada TBF, recursos são alocados na interface aérea em um ou mais PDTCHs. Um TBF é temporário e mantido apenas enquanto durar a transferência de dados. Cada TBF possui um TFI (*Temporary Flow Identity*) e a camada RLC adiciona um cabeçalho contendo o TFI, o número de seqüência do bloco RLC (BSN – *Block Sequence Number*), a direção (*uplink/downlink*) e outras informações.

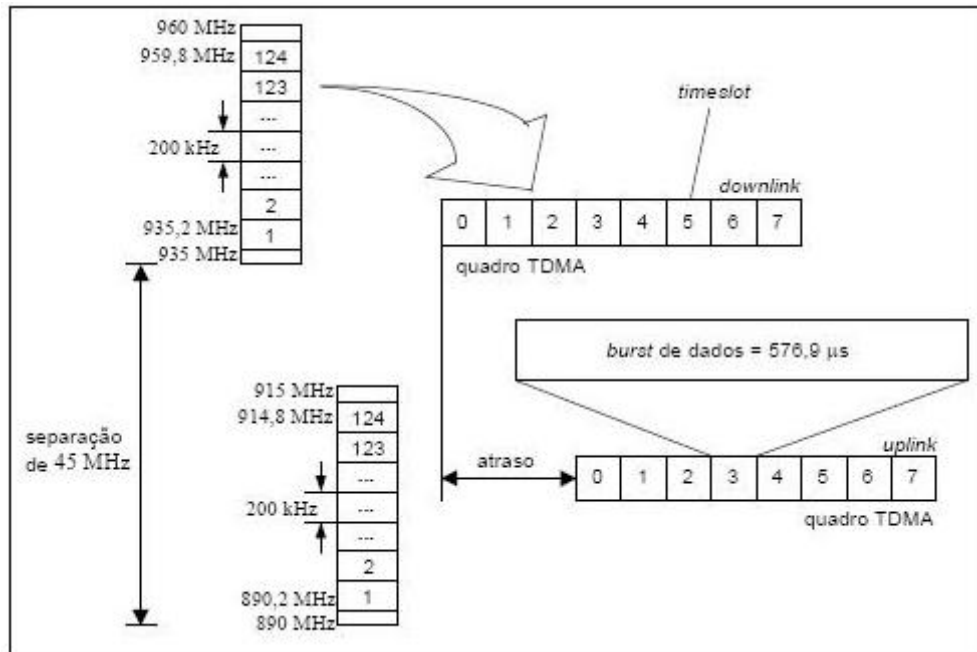
Para a segmentação dos blocos LLC em blocos de dados RLC, precisa ser definida a quantidade de dados úteis a serem transmitidos. Tal quantidade varia de acordo com o cenário de codificação (CS) escolhido. Os possíveis valores são 22, 32, 38 e 52 octetos de dados, correspondendo a CS-1, CS-2, CS-3 e CS-4, respectivamente. O CS é determinado de acordo com a relação vazão/confiabilidade desejada (vide Seção 2.7.4).

#### **2.6.4 Transmissão através da Interface Aérea**

Após a segmentação dos dados e inserção do cabeçalho RLC, os blocos RLC são passados à próxima camada, MAC. Esta adiciona seu próprio cabeçalho e encaminha os blocos RLC para a camada física (GSM RF), que os transmite através da interface aérea para a estação móvel. Na MS, os blocos são conduzidos através das camadas, nas quais os respectivos cabeçalhos são retirados e os segmentos remontados. Finalmente, a mensagem de e-mail original é recebida na camada de aplicação pelo usuário do móvel.

### **2.7 INTERFACE AÉREA**

A interface aérea do GPRS utiliza a estrutura do GSM. Na camada física, o GSM usa uma combinação de TDMA com FDMA. O espectro reservado para o GSM, que varia de acordo com o país, é dividido em duas bandas de frequência de 25 MHz, separados por 45 MHz. Na Europa (Figura 7), são usadas as bandas 890 – 915 MHz para transmissão *downlink* (isto é, no sentido BS-MS) e 935 – 960 MHz para transmissão *uplink* (ou seja, no sentido MS-BS). Cada banda é dividida em 124 portadoras de canais (de frequências) com 200 kHz de largura. Um determinado número dessas portadoras é reservado para uma BTS (isto é, para uma célula).



**Figura 8 - Portadoras de Frequência GSM, Duplexação e Quadros TDMA**

Cada portadora é dividida em quadros TDMA, os quais são compostos por oito *timeslots*, ou seja, oito canais. Cada *timeslot* dura 576,9 ms, logo um quadro dura 4,615 ms. A repetição de um *timeslot* particular define um canal físico.

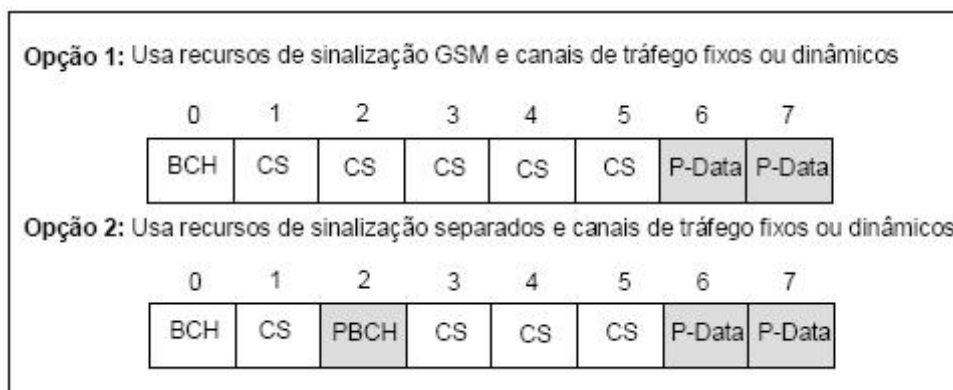
Como pode ser observado na Figura 8, um quadro TDMA no sentido *uplink* é iniciado com um atraso de três *timeslots* do início de um quadro TDMA no sentido *downlink*.

Uma estação móvel GSM usa o mesmo *timeslot* nos dois sentidos: *downlink* e *uplink* [8]. GPRS apresenta um mecanismo de alocação de canais mais flexível: mais de um *timeslot* (operação *multislot*) pode ser reservado para uma mesma MS. Além disso, a alocação *downlink* e *uplink* pode ser assimétrica, pois é realizada separadamente.

Em um sistema GSM, um canal é alocado permanentemente durante todo o período de uma chamada, havendo ou não transmissão de dados. Em GPRS, o canal é alocado apenas quando há transmissão de pacotes, beneficiando o usuário, que não precisa se preocupar com o tempo de conexão.

### 2.7.1 Canais Físicos

Em um sistema GPRS, a interface aérea é dividida em canais físicos para tráfego GPRS e canais físicos para GSM.



**Figura 9 - Divisão da Interface Aérea GPRS em Canais Físicos**

Há duas opções para estabelecer essa divisão. Nos dois casos, um canal – chamado BCH (*Broadcast Channel*) – é reservado para a sinalização GSM. Existem canais reservados para serviços por comutação de circuito, ou seja, serviços GSM, e canais reservados para tráfego de dados GPRS, denominados PDCHs (*Packet Data Channels*). Estes podem ser fixos ou alocados dinamicamente. A diferença entre as duas opções, que podem ser visualizadas na Figura 9, consiste na separação ou compartilhamento de um canal para sinalização. Na primeira opção, o GPRS utiliza o canal de sinalização do GSM (BCH). Enquanto na segunda opção, o GPRS possui um canal particular para sua sinalização (PBCH – *Packet Broadcast Channel*). Ou seja, nesse caso os recursos GSM e GPRS são totalmente separados.

### 2.7.2 Canais Lógicos

Sobre os canais físicos, são definidos canais lógicos para realizar várias funções, como sinalização, *broadcast*, *paging* e transporte de dados (*payload*).

Os canais lógicos GPRS podem ser classificados em dois grupos: canais de controle (*Packet Control Channels*) e canais de tráfego (*Packet Traffic Channels*). Os canais de controle podem ainda ser divididos em canais de controle compartilhados, canais de controle dedicados e canais de controle de *broadcast*. Uma breve descrição de cada canal é realizada a seguir.

PCCCH (*Packet Common Control Channel*) – consiste em um conjunto de canais lógicos utilizados para sinalização comum entre o BSS e a MS. É composto por quatro subcanais:



- PRACH (*Packet Random Access Channel*) – usado pela MS para solicitar um ou mais canais para transmissão de dados (PDTCH) ou responder mensagens de *paging*.
- PAGCH (*Packet Access Grant Channel*) – utilizado para responder mensagens de reserva de recursos (PDTCHs) para uma estação móvel antes da transferência de pacotes.
- PPCH (*Packet Paging Channel*) – usado pelo BSS para localizar uma MS (isto é, realizar um *paging*) antes da transmissão de pacotes *downlink*.
- PNCH (*Packet Notification Channel*) – utilizado para informar um grupo de estações móveis sobre a chegada de mensagens PTM-M (*Point-To-Multipoint-Multicast*).

PBCCH (*Packet Broadcast Control Channel*) – utilizado pelo BSS para transmitir informações sobre o sistema para todas as estações móveis GPRS na célula.

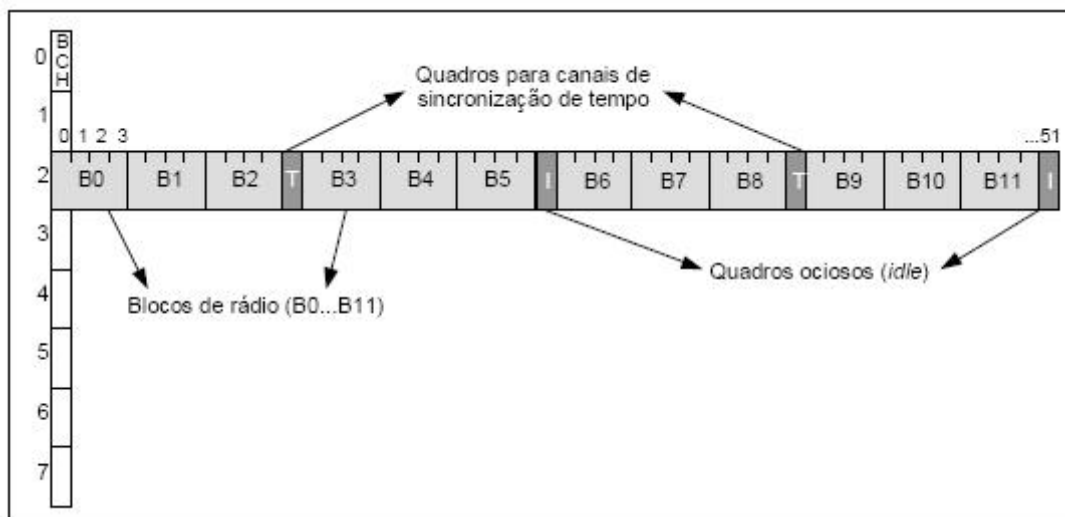
Os canais de controle dedicados são:

- PACCH (*Packet Associated Control Channel*) – provê informações de sinalização sobre uma determinada MS (por exemplo, controle de potência). Um PACCH é associado a um ou mais PDTCHs atribuídos a uma estação móvel.
- PTCCH (*Packet Timing advance Control Channel*) – usado para sincronização.

PDTCH (*Packet Data Traffic Channel*) – transmite dados dos usuários. É alocado para uma estação ou para um grupo de estações (para aplicações *multicast*). Uma MS pode usar vários PDTCHs ao mesmo tempo para operação *multislot*.

### 2.7.3 Mapeamento de Canais Lógicos sobre Canais Físicos

Os canais lógicos definidos no item anterior não necessariamente requerem recursos físicos separados, sendo mapeados nos canais físicos através de uma técnica de *multiframeing*. Um *multiframe* é um conjunto de N quadros (*frames*) TDMA que juntos apresentam uma funcionalidade [9].



**Figura 10 - Mapeamento de Canais Lógicos em Canais Físicos**

Um *multiframe* GPRS é composto por 52 quadros (*frames*) e pode ser visualizado na Figura 10. Em cada quadro, um *timeslot* é considerado para formar essa estrutura, ou seja, após 52 quadros, juntando os *timeslots* de número 2, por exemplo, uma estrutura como essa terá sido gerada. Os 52 quadros são divididos em 12 blocos, B0 a B11, que são constituídos por 4 quadros. Dos 4 quadros restantes (uma vez que 12 blocos de 4 quadros cada resultam em 48 quadros), 2 são usados para canais de sincronização de tempo (PTCCH) e 2 são deixados ociosos. Também são definidos *multiframes* com 51 quadros (*frames*), os quais são usados para carregar apenas os canais PCCCH e PBCCH.

Há três combinações possíveis para o mapeamento de canais lógicos em um único canal físico (usando o *multiframe* com 52 quadros) [10]:

A primeira combinação permite que todos os canais lógicos sejam mapeados em um canal físico. Uma vez que existem apenas 12 blocos para canais lógicos em um *multiframe*, essa opção reduzirá o número de canais disponíveis para cada categoria.

A segunda combinação permite que todos os canais exceto o PBCCH sejam mapeados em um canal físico. Isso aumenta a capacidade de PCCCH e PDTCH. Mas essa opção só pode ser usada se os canais PBCCH e PCCH existirem em outro canal físico ou se os canais BCCH e CCCH do GSM puderem ser usados.

A terceira e última combinação é para tráfego e sinalização dedicada apenas, o que exclui PBCCH e PCCCH. Essa combinação provê mais PDTCHs em um canal físico e pode ser utilizada apenas se esses canais existirem em outros canais físicos ou se os canais BCCH e CCCH do GSM puderem ser usados.

O conceito mestre-escravo pode ser usado em algumas instâncias nas quais um canal físico é configurado como mestre e carrega todos os tipos de canais (isto é, utiliza a primeira opção descrita acima), e os outros canais físicos são configurados como escravos e usados apenas para transferência de dados (terceira opção). Quando a demanda de transferência de dados é baixa, esses outros canais físicos podem ser usados para serviços de comutação por circuito.

#### 2.7.4 Cenários de Codificação

A codificação de canais é usada para proteger pacotes de dados transmitidos contra erros. Há quatro cenários de codificação no GPRS, os quais são utilizados de acordo com as condições do canal (ruído). Quanto piores as condições do ambiente de transmissão, maior deve ser a proteção aos dados. E quanto maior a proteção, menor a quantidade de dados úteis (isto é, dados do usuário).



**Figura 11 - Codificação dos Pacotes de Dados GPRS**

A Figura 11 mostra o processo de codificação. Primeiramente um codificador de blocos é utilizado para adicionar os bits de paridade (BCS – *Block Check Sequence*). Depois os bits da USF (*Uplink State Flag*) são codificados. Os *tail bits*, necessários para a terminação da codificação de convolução, são adicionados em seguida. Um codificador “convolucional” é usado na seqüência. Finalmente, alguns *bits* de proteção são eliminados (se necessário) através de um mecanismo denominado *puncturing*, resultando em uma *codeword* de 456 bits. Para o CS-4, o processo pára no segundo bloco (codificação da USF).

**Tabela 1 - Cenários de Codificação**

Cenário de Codificação	Dados e cabeçalho RLC	Cabeçalho e dados RLC/MAC	USF	BCS	Tail bits	Saída codificador convolução	Punctured bits	Taxa de codificação	Taxa de dados
CS-1	22 octetos	181 bits	3	40	4	456 bits	0	1/2	9,05 kbps
CS-2	32 octetos	268 bits	6	16	4	588 bits	132	≈ 2/3	13,4 kbps
CS-3	38 octetos	312 bits	6	16	4	676 bits	220	≈ 3/4	15,6 kbps
CS-4	52 octetos	428 bits	12	16	0	456 bits	0	1	21,4 kbps

A Tabela 1 apresenta os valores envolvidos nos cenários de codificação.

## 2.8 OPERAÇÕES DE TRANSFERÊNCIA DE PACOTES DE DADOS

Para iniciar uma transferência de pacotes, uma estação móvel GPRS deve primeiro se conectar a uma rede GPRS e realizar a ativação de um contexto PDP (vide Seção 2.3.2). Depois a MS pode acessar a rede, solicitar recursos, enviar dados, entrar no modo *standby* caso não esteja transmitindo dados e repetir o processo todo novamente.

### 2.8.1 Transferência no Sentido *Uplink*

Uma MS precisa solicitar recursos ao BSS antes de transferir os dados. Esse processo começa com a requisição de um canal de dados através do PRACH (ou RACH, caso a sinalização seja dividida com a rede GSM – vide Seção 2.7.1). A rede responde através do PDCH (ou AGCH). A requisição pode ser feita em uma ou duas fases.

Na requisição em uma fase, a rede responde à solicitação da MS reservando recursos em um ou mais PDCHs para a transferência *uplink* de um determinado número de blocos de rádio. Uma requisição em duas fases permite que a estação móvel forneça uma descrição completa dos recursos requisitados para a transferência *uplink*.

Se a rede não responde a uma requisição dentro de um certo período, a MS tenta novamente após um tempo de *back-off*.

### 2.8.2 Alocação de Blocos de Rádio

O próximo passo na transferência de pacotes de dados é a alocação de blocos de rádio. Esta pode ser dinâmica ou fixa.

### 2.8.2.1 Dinâmica

Na alocação dinâmica de blocos de rádio, a MS é informada sobre os blocos disponíveis bloco-a-bloco. A rede manda uma mensagem para a estação móvel com a lista de PDCHs associados e o correspondente valor da USF (*Uplink Status Flag*). A MS monitora as USFs nos PDCHs alocados e transmite os blocos de rádio nos canais que possuem o valor USF reservado para ela.

A USF atribui um ou quatro blocos a uma MS. Assim, quando um móvel detecta sua USF em um *downlink*, ele transmite um bloco ou uma seqüência de quatro blocos RLC/MAC. A MS inclui na transmissão do bloco *uplink* o número de blocos que faltam serem transmitidos. A rede continuará reservando blocos para a estação até que esta indique que não há mais blocos a serem transmitidos.

### 2.8.2.2 Fixa

Na alocação fixa, a rede reserva todos os blocos requisitados pela MS. A mensagem de reserva inclui os parâmetros frequência, *timeslots*, um mapa de 128 bits e um *start frame number*.

Ao receber essa informação, a estação móvel começará a transmitir o dado *uplink* nos blocos correspondentes, começando com o *start frame number* e usando a informação contida no mapa de bits (0 = bloco não reservado; 1 = bloco reservado).

## 2.8.3 Transferência de Blocos de Dados RLC

Os blocos de dados RLC podem ser transferidos utilizando o modo com ou sem confirmação RLC.

No modo sem confirmação (*unacknowledged*), os blocos são transmitidos para a camada superior na ordem em que chegam. Blocos perdidos são substituídos por bits 0, e a camada superior responsabiliza-se pela detecção/correção de erros [11].

No modo com confirmação (*acknowledged*), os blocos são numerados, sendo entregues à camada LLC apenas quando todos os blocos RLC que compõem um bloco LLC são recebidos. Isso é possível através da retransmissão seletiva de blocos (ARQ – *Automatic Repeat Request*).

O RLC é um protocolo de janela deslizante, isto é, permite que o transmissor envie um determinado número de blocos sem receber ACKs (*ACKnowledgement*). E ao receber um ACK, a janela “desliza” permitindo que mais blocos sejam enviados. O tamanho da janela (*WS – Window Size*) para o GPRS é 64, ou seja, um transmissor pode enviar até 64 blocos RLC sem receber nenhum ACK.

A confirmação da camada RLC é realizada através dos pacotes ACK/NACK (*acknowledged/not acknowledged*). Estes contêm um mapa de bits com 128 elementos em que cada bit indica o estado de cada bloco transmitido (0 = bloco não recebido corretamente; 1 = bloco recebido corretamente).

Quando a janela fica completa, o transmissor pára de enviar blocos novos e começa a retransmitir os blocos não reconhecidos, avisando que a janela está cheia e que necessita de um pacote ACK/NACK.

De acordo com a especificação [11], o receptor envia pacotes ACK/NACK “quando necessário”. Os casos explicitamente necessários são quando solicitado pelo transmissor e ao receber o último bloco de uma seqüência.

### **2.8.4 Transferência no Sentido *Downlink***

A rede inicia a transferência de dados no sentido *downlink* enviando uma mensagem para a MS destino (que se encontra no estado *ready*), associando a ela um canal. Se a MS estiver no estado *standby*, a rede realiza primeiramente um *paging*.

A rede transmite blocos RLC/MAC pertencentes a um fluxo temporário de blocos (*TBF – Temporary Block Flow*) no canal *downlink* associado à estação móvel. Geralmente mais de uma estação é multiplexada em um PDCH em um determinado momento. Cada estação identifica seu TBF através do TFI (*Temporary Flow Identity*) contido no cabeçalho RLC.

Após enviar o último bloco RLC, a rede inicia o processo de liberação de recursos. Ao receber da MS o pacote ACK/NACK final, a rede inicializa um timer e, quando este expira, o TFI e todos os recursos alocados para a MS são liberados.

## **2.9 CONSIDERAÇÕES FINAIS**

Nesse capítulo foi apresentada uma visão geral do GPRS, mostrando características e conceitos relacionados a essa tecnologia.

Primeiramente o GSM, que constitui a base do GPRS, foi apresentado de forma resumida. Mostrou-se a rede GSM com seus subsistemas e, em seguida, os novos componentes, adicionados pela rede GPRS foram descritos.

O modo como o GPRS gerencia a mobilidade de suas estações móveis foi exposto, descrevendo os procedimentos utilizados para isso.

Os protocolos que constituem os planos de transmissão e de sinalização do GPRS (por exemplo, GTP, RLC/MAC, BSSAP+) foram apresentados, seguidos de um exemplo de um cenário ilustrando a transmissão fim a fim em uma rede GPRS.

Também foi descrita a interface aérea, com seus canais lógicos e físicos, bem como o mapeamento entre eles. Os cenários de codificação utilizados para transmitir os dados através da interface aérea foram detalhados, sendo explicado inclusive o processo de codificação.

A transferência de pacotes de dados, tanto no sentido *downlink* quanto no *uplink*, foi exposta. Também foram apresentadas as formas de alocação de blocos de rádio, além da transferência de blocos de dados RLC.

O estudo sobre a especificação da rede GPRS e seus protocolos é importante para que se possa criar um ambiente de simulação de acordo com a especificação, obtendo resultados muito próximos da realidade deste tipo de ambiente.

### **3 PROPOSTA DE PACOTES DE INFORMAÇÃO DE RASTREAMENTO VEICULAR**

Baseado no estudo realizado nos Trabalhos Individuais I e II, foram identificadas as informações necessárias para o rastreamento e monitoramento de veículos através de dispositivos GPS e redes sem fio.

A identificação das informações necessárias e a definição do formato dos pacotes contendo informações sobre o rastreamento de veículos servem como base para definição do tamanho do pacote de dados que será trafegado nos diversos cenários gerados no simulador de redes NS, buscando resultados cada vez mais condizentes com a realidade deste tipo de sistema [16] [17] [18].

Este capítulo inicia com uma visão alto nível do sistema de rastreamento veicular, sobre o qual os pacotes de comunicação irão trafegar, logo após, esses pacotes são detalhados. Por último, são abordados os protocolos de transporte TCP e UDP sobre o qual os pacotes de informações de rastreamento veicular serão transportados.

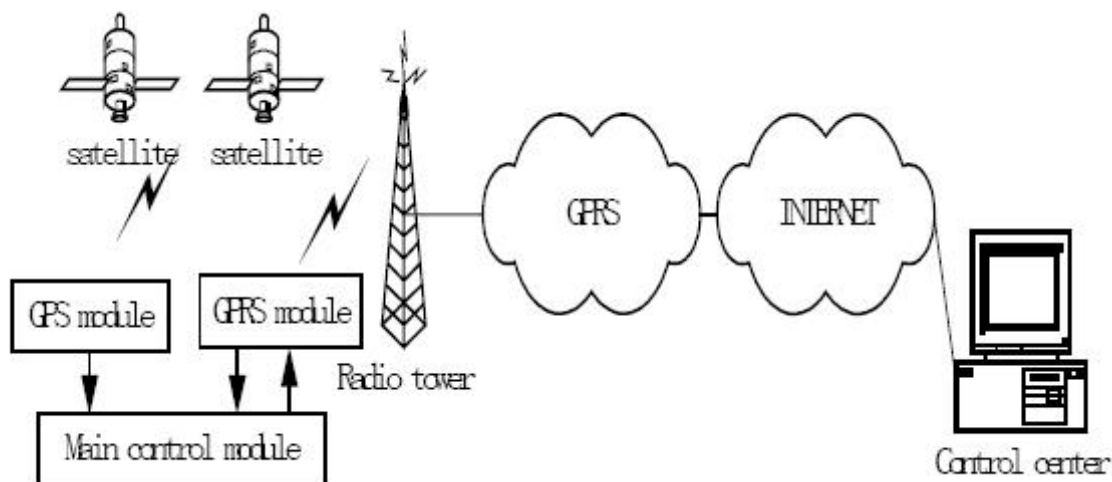
Através das informações levantadas, foram definidos dois pacotes utilizados no envio (sentido veículo servidor central) de informações em um sistema de rastreamento veicular, (1) pacote de informações básicas de telemetria e (2) pacote de informações avançadas de telemetria. O formato destes pacotes é detalhado ao longo deste capítulo.

#### **3.1 VISÃO GERAL DO SISTEMA**

Para facilitar o entendimento do fluxo de informações (pacotes de dados) trocados pelo sistema, devemos analisar o *framework* sobre o qual o protocolo proposto irá funcionar. O sistema de rastreamento veicular é composto de um componente de veículo móvel, um centro de monitoramento (servidor central) e uma rede de comunicação.



O componente do veículo móvel inclui um receptor GPS, um módulo GPRS e um módulo de controle central. A estrutura do *framework* do sistema pode ser visualizada na Figura 12.



**Figura 12 – Visão Geral do sistema**

Os componentes do sistema já foram detalhados nos capítulos anteriores deste trabalho (vide Capítulo 2), esta seção tem como objetivo fornecer uma visão macro do funcionamento do sistema para facilitar a compreensão do fluxo de informações realizado pelo sistema.

### 3.2 FORMATO DOS PACOTES DE COMUNICAÇÃO

Esta seção descreve o formato dos pacotes de comunicação que serão utilizados para simular o comportamento da rede e do sistema no envio de informações de rastreamento veicular, serão detalhadas as informações contidas dentro de cada pacote de dados bem como as técnicas de compactação destes dados, utilizadas com intuito de diminuir o custo necessário para realizar o envio e recebimento destes dados.

Baseado nas necessidades de comunicação do sistema de rastreamento veicular foram identificados dois pacotes de transmissão de informações, (1) Pacote de transmissão de informações básicas de telemetria, (2) Pacote de transmissão de informações avançadas de telemetria.

Estes pacotes são detalhados nas próximas seções deste trabalho.

### 3.2.1 PT\_IBT: Pacote de Transmissão de Informações Básicas de Telemetria

O pacote PT\_IBT contém as informações básicas de telemetria possíveis geradas pelo veículo. Este pacote é transmitido do veículo para o servidor central em determinados casos.

Este pacote possui 41 bytes de tamanho e as informações estão dispostas no pacote conforme Tabela 2:

**Tabela 2 – Posição e descrição dos bytes do pacote de informações básicas**

Posição do Byte	Descrição
1	Tipo do evento
2	codigo_sim_card[0]
3	codigo_sim_card[1]
4	codigo_sim_card[2]
5	codigo_sim_card[3]
6	codigo_sim_card[4]
7	codigo_sim_card[5]
8	codigo_sim_card[6]
9	codigo_sim_card[7]
10	codigo_sim_card[8]
11	codigo_sim_card[9]
12	codigo_sim_card[10]
13	codigo_sim_card[11]
14	codigo_sim_card[12]
15	codigo_sim_card[13]
16	codigo_sim_card[14]
17	codigo_sim_card[15]
18	codigo_sim_card[16]
19	codigo_sim_card[17]
20	codigo_sim_card[18]
21	codigo_sim_card[19]
22	data_hora[1]
23	data_hora[2]
24	data_hora[3]
25	data_hora[4]
26	data_hora[5]
27	data_hora[6]
28	data_hora[7]
29	data_hora[8]
30	Lath
31	Latm
32	Latl
33	Lonh
34	Lonm
35	Lonl
36	Lonlat
37	Direção
38	Velocidade
39	Sinalgsm
40	Sensordig
41	CRC

A posição 1 identifica o tipo de pacote enviado. Tipo B identifica pacote de informações básicas.

As posições 2 a 21 identificam o código do *sim card* do *chip* GSM utilizado como referência única ao veículo monitorado.

As posições 22 a 29 informam a data e hora do evento de forma compactada, mais detalhes sobre o algoritmo de compactação da data pode ser visualizada no Apêndice A deste trabalho.

As posições 30 a 36 identificam a coordenada geográfica do veículo identificado em graus e minutos com o padrão *Datum* WGS-84. A posição 30 representa o byte mais significativo da latitude (graus, variando de 00..90) , a posição 31 como o byte intermediário da latitude (minutos variando de 00..59), a posição 32 como o byte menos significativo da latitude (décimos de minutos, variando de 00..99) mais o quadrante (o bit mais significativo do byte latl contem o quadrante global onde o estado do bit igual a 1 representa sul e o estado do bit igual a 0 representa o norte) , a posição 33 como o byte mais significativo da longitude (graus, variando de 00..180), a posição 34 como o byte intermediário da longitude (minutos variando de 00..59), a posição 35 como o byte menos significativo de longitude (décimos de minutos, variando de 00..99) mais o quadrante (o bit mais significativo do byte lonl contem o quadrante global onde o estado do bit igual a 1 representa o oeste e o estado do bit igual a 0 representa o leste) e a posição 36 como uma possível informação complementar de latitude e longitude.

A posição 37 identifica a direção em que o veículo esta se locomovendo, variando seu valor de 0 .. 35, representando 0° a 350° com referência ao norte.

A posição 38 identifica o pico de velocidade atingida a partir do último evento originado do GPS, variando seu valor de 0km a 250km/h.

A posição 39 é referente ao nível de sinal GSM atual. Seus valores possíveis estão descritos na Tabela 3.

**Tabela 3 – Valores possíveis referente ao sinal GSM**

Valores Possíveis	Descrição
0	113 dBm ou menos
1	111 dBm
2..30	51 dBm ou mais
99	Sem sinal ou não detectável

O byte de posição 40 indica uma seqüência de 8 sinais (8 bits) digitais que representam alguns avisos possíveis sobre o veículo, onde o estado do bit = 0 indica sensor acionado e estado do bit = 1 indica condição normal, conforme a Tabela 4:

**Tabela 4 – Representação de cada bit da posição 31**

Bit	Descrição
0	Botão de pânico
1	Botão de chamada
2	Sensor de porta
3	Sensor de porta do carona
4	Sensor de Bagageiro
5	Sensor Baú
6	Sensor Ignição
7	Sensor de 5º roda

A posição 41 refere-se a um byte de verificação cíclica de redundância, para controle de erro ou sujeira na transmissão do pacote, o algoritmo para cálculo e verificação de erro pode ser analisado no Apêndice B deste trabalho.

### **3.2.2 PT\_IAT: Pacote de Transmissão de Informações Avançadas de Telemetria**

O pacote PT\_IAT contém todas informações de telemetria possíveis geradas pelo veículo. Este pacote é transmitido do veículo para o servidor central em determinados casos.

Este pacote possui 66 bytes de tamanho e as informações estão dispostas no pacote conforme Tabela 5:

**Tabela 5 – Posição e descrição dos bytes do pacote de informações avançadas**

Posição do Byte	Descrição
1	Tipo do evento
2	Codigo_sim_card[0]
3	Codigo_sim_card[1]
4	Codigo_sim_card[2]
5	Codigo_sim_card[3]
6	Codigo_sim_card[4]
7	Codigo_sim_card[5]
8	Codigo_sim_card[6]
9	Codigo_sim_card[7]
10	Codigo_sim_card[8]
11	Codigo_sim_card[9]
12	Codigo_sim_card[10]
13	Codigo_sim_card[11]
14	Codigo_sim_card[12]
15	Codigo_sim_card[13]
16	Codigo_sim_card[14]
17	Codigo_sim_card[15]
18	Codigo_sim_card[16]
19	Codigo_sim_card[17]
20.	Codigo_sim_card[18]
21	cod Equipamento[1]
22	cod Equipamento[2]
23	cod Equipamento[3]
24	cod Equipamento[4]
25	cod Equipamento[5]
26	cod Equipamento[6]
27	cod Equipamento[7]
28	cod Equipamento[8]
29	cod Equipamento[9]
30	cod Equipamento[10]
31	cod Equipamento[11]
32	cod Equipamento[12]
33	cod Equipamento[13]
34	cod Equipamento[14]
35	cod Equipamento[15]
36	cod Equipamento[16]
37	data_hora[1]
38	data_hora[2]
39	data_hora[3]
40	data_hora[4]
41	data_hora[5]
42	data_hora[6]
43	data_hora[7]
44	data_hora[8]
45	Lath
46	Latm
47	Latl
48	Lonh
49	Lonm
50	Lonl
51	Lonlat
52	Direção
53	Velocidade
54	Hodometro
55	Giro

**Tabela 5 – Continuação descrição pacote de informações avançadas**

56	Sinalgsm
57	Sensordig
58	Reservado(1)
59	Reservado(2)
60	Reservado(3)
61	Reservado(4)
62	Reservado(5)
63	Reservado(6)
64	Reservado(7)
65	Reservado(8)
66	CRC

A posição 1 identifica o tipo de pacote enviado. Tipo C identifica pacote de informações completas.

As posições 2 a 20 identificam o código do *sim card* do *chip* GSM utilizado como referência única ao veículo monitorado.

As posições 21 a 36 identificam o código do aparelho instalado no veículo.

As posições 37 a 44 informam a data e hora do evento de forma compactada, mais detalhes sobre o algoritmo de compactação da data pode ser visualizada no Apêndice A deste trabalho.

As posições 45 a 51 identificam a coordenada geográfica do veículo identificado em graus e minutos com o padrão Datum WGS-84. A posição 35 representa o byte mais significativo da latitude (graus, variando de 00..90), a posição 36 como o byte intermediário da latitude (minutos variando de 00..59), a posição 37 como o byte menos significativo da latitude (décimos de minutos, variando de 00..99) mais o quadrante (o bit mais significativo do byte *latl* contem o quadrante global onde o estado do bit igual a 1 representa sul e o estado do bit igual a 0 representa o norte), a posição 38 como o byte mais significativo da longitude (graus, variando de 00..180), a posição 39 como o byte intermediário da longitude (minutos variando de 00..59), a posição 40 como o byte menos significativo de longitude (décimos de minutos, variando de 00..99) mais o quadrante (o bit mais significativo do byte *lonl* contem o quadrante global onde o estado do bit igual a 1 representa o oeste e o estado do bit igual a 0 representa o leste) e a posição 41 como uma possível informação complementar de latitude e longitude.

A posição 52 identifica a direção em que o veículo esta se locomovendo, variando seu valor de 0 .. 35, representando 0° a 350° com referência ao norte.

A posição 53 identifica o pico de velocidade atingida a partir do último evento originado do GPS, variando seu valor de 0km a 250km/h.

A posição 54 identifica o hodômetro que corresponde à distância percorrida pelo veículo a partir do último envio originado do GPS.

A posição 55 indica picos de giros do motor a partir do último evento originado do GPS.

A posição 56 é referente ao nível de sinal GSM atual. Seus valores possíveis estão descritos na Tabela 6.

**Tabela 6 – Valores possíveis referente ao sinal GSM**

Valores Possíveis	Descrição
0	113 dBm ou menos
1	111 dBm
2..30	51 dBm ou mais
99	Sem sinal ou não detectável

O byte de posição 57 indica uma seqüência de 8 sinais (8 bits) digitais que representam alguns avisos possíveis sobre o veículo, onde o estado do bit = 0 indica sensor acionado e estado do bit = 1 indica condição normal, conforme a Tabela 7:

**Tabela 7 – Representação de cada bit da posição 31**

Bit	Descrição
0	Botão de pânico
1	Botão de chamada
2	Sensor de porta
3	Sensor de porta do carona
4	Sensor de Bagageiro
5	Sensor Baú
6	Sensor Ignição
7	Sensor de 5° roda

As posições 58 a 64 são reservadas no pacote de informações avançadas para uso em casos específicos de cargas especiais.

A posição 65 refere-se a um byte de verificação cíclica de redundância, para controle de erro ou sujeira na transmissão do pacote, o algoritmo para cálculo e verificação de erro pode ser analisado no Apêndice B deste trabalho.

### 3.3 PROCEDIMENTOS EM AUSÊNCIA DE SINAL GSM/GPS

Sistemas que trafegam dados sobre redes sem fio estão sujeitas a uma alta interferência de ruídos provenientes do meio físico. Tratando-se de sistemas que trafegam dados sobre redes móveis e utilizam o meio físico de redes de celulares, por exemplo GSM, este tipo de interferência ocorre ainda em maior escala.

Além da perda de pacotes provenientes do meio físico, existe a questão de que muitas vezes, o sinal GPS possa sofrer a mesma interferência do meio, pois se trata de um ambiente que utiliza um canal de conexão sem fio, só que neste caso, comunicação via satélite.

Sistemas para rastreamento de veículos, que trabalham com estes tipos de variáveis, devem estar preparados para armazenar as informações no momento de ausência de sinal GSM/GPS [24]. Para simular este comportamento, supõe-se que os dispositivos GPS/GSM simulados nos cenários deste trabalho possuem uma memória de massa, com capacidade de 10 Mbytes (capaz de armazenar em torno de 32 horas de informações de rastreamento utilizando o pacote PT\_IAT com intervalo de envio de 1 segundo), que armazena os pacotes de informações, tanto os pacotes PT\_IBT quanto PT\_IAT, transferindo estes pacotes no momento de restabelecimento do sinal ou melhoria do canal de comunicação.

Devido a esta característica, será simulada a vazão máxima de envio de pacotes nos diversos cenários propostos pelo trabalho, para os variados tamanhos de pacotes, condições de sinal e número de estações móveis. O resultado destas simulações será detalhado no capítulo 6 deste trabalho.

### 3.4 PROTOCOLOS DE TRANSPORTE

Os protocolos TCP (protocolo de controle de transferência) e UDP (protocolo de dados do usuário) são protocolos de transporte, de baixo nível, que são utilizados em grande escala sobre redes GPRS é sobre estes protocolos que os pacotes de informações de rastreamento veicular serão transportados.

#### 3.4.1 TCP – *Transmission Control Protocol*

O TCP (*Transmission Control Protocol*) [37] é um protocolo da camada de transporte, orientado à conexão, desenvolvido para oferecer funções de comunicação confiável e controle de congestionamento fim a fim. Para prover confiabilidade, o TCP utiliza o mecanismo



ARQ (*Automatic Repeat Request*), que retransmite todos os segmentos que não são explicitamente confirmados. Para evitar o congestionamento da rede e a sobrecarga dos sistemas finais, o TCP realiza controle de tráfego (controle de fluxo e de congestionamento) através de um mecanismo de janela deslizante [27]. Esse mecanismo permite que o transmissor envie um determinado número de segmentos sem receber ACKs (*ACKnowledgements* - confirmações), e, ao receber um ACK, a janela “desliza” permitindo que mais segmentos sejam enviados. O tamanho da janela, indicado pela variável CWND (*Congestion Window*), varia dinamicamente, baseado nos ACKs recebidos.

Quatro algoritmos básicos governam o controle de congestionamento do TCP: *Slow-Start*, *Congestion Avoidance*, *Fast Retransmit* e *Fast Recovery* [35]. O *Slow Start* consiste no aumento gradual da quantidade de dados inseridos na rede, através do aumento do tamanho da janela. É usado no início da conexão TCP e após a detecção de congestionamento. O algoritmo começa inicializando CWND com um valor pequeno (normalmente igual a 1) e, a cada ACK recebido, incrementa esse valor em um segmento, causando um crescimento exponencial da janela. O *Slow Start* continua até que seja atingido um limiar, denominado SSTHRESH (*Slow Start Threshold*), a partir do qual é utilizado o *Congestion Avoidance*. No *Congestion Avoidance*, o valor do CWND é maior ou igual a SSTHRESH. Este algoritmo incrementa CWND a uma taxa menor que o *Slow Start*. Para cada ACK recebido, a janela é incrementada em  $1/CWND$ , produzindo um crescimento linear da janela.

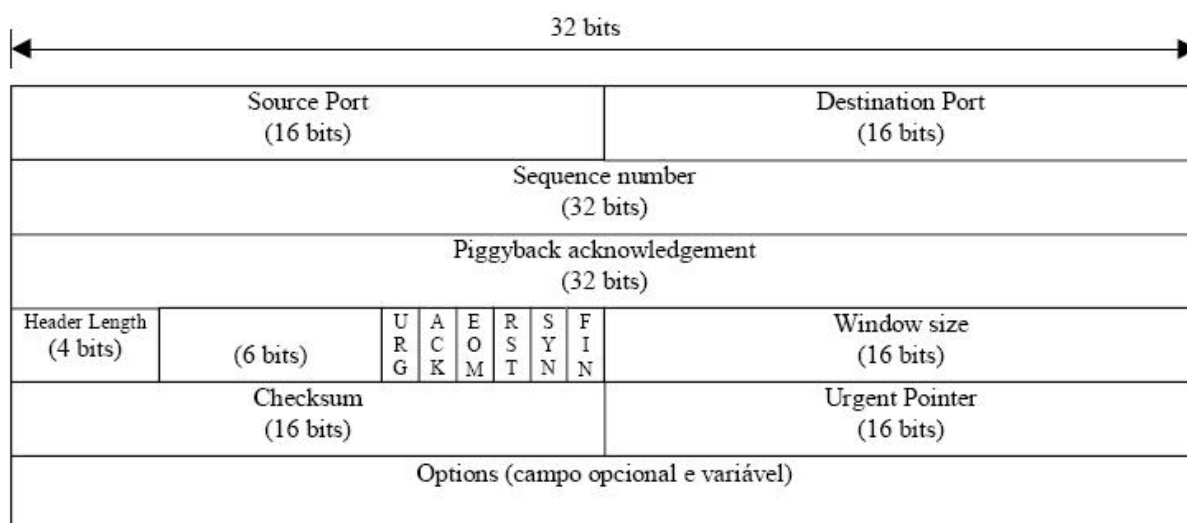
Os algoritmos *Fast Retransmit* e *Fast Recovery* permitem que o TCP detecte e corrija a perda de segmentos antes que o RTO (*Retransmission Timeout*) expire. RTO consiste em um *timer* que é inicializado sempre que um segmento é enviado. Se um ACK não for recebido após esse tempo, considera-se que o segmento foi perdido. Assim, o segmento é retransmitido e o TCP volta ao *Slow Start*. O *Fast Retransmit* assume que um segmento foi perdido após receber três ACKs duplicados. Nesse caso, o TCP retransmite o segmento perdido e retorna ao *Slow Start*. O *Fast Recovery* pode ser utilizado após a retransmissão do segmento no *Fast Retransmit*, evitando que o TCP volte ao *Slow Start*. O *Fast Recovery* reduz a janela à metade, incrementando-a, em seguida, de forma linear.

Embora seu funcionamento básico esteja padronizado, o TCP apresenta algumas variações e melhoramentos visando cada vez mais um melhor aproveitamento da rede. Os algoritmos que implementam e a forma como estes são implementados diferenciam as versões do TCP. Há, por exemplo, versões que não apresentam o *Fast Recovery* (TCP Tahoe) e outras que o implementam de forma diferente (TCP New-Reno).

Como não é objetivo deste trabalho o aprofundamento sobre as variações do protocolo TCP e sim a interação deste protocolo com o GPRS, foi assumido a versão TCP Reno para as simulações envolvendo este protocolo.

O TCP Reno [23] implementa os quatro algoritmos básicos do controle de congestionamento sem modificações, ou seja, da forma descrita em [35]. Em outras palavras, inicializa a CWND com um segmento, incrementando a janela de forma exponencial (*Slow Start*) até atingir o Ssthresh, quando o crescimento passa a ser linear (*Congestion Avoidance*). Ao receber três ACKs duplicados de um segmento, retransmite esse segmento (*Fast Retransmit*) e reduz a janela à metade (*Fast Recovery*). No caso de ocorrer um *timeout*, volta ao *Slow Start*.

Entidades TCP trocam dados na forma de segmentos, que consiste num cabeçalho de 20 bytes (fixo) mais uma parte opcional, seguida de zero ou mais bytes de dados. O formato do cabeçalho TCP pode ser visualizado na Figura 13.



**Figura 13 – Formato do Cabeçalho TCP [26]**

### 3.4.2 UDP – User Datagram Protocol

O UDP é um protocolo de nível de transporte orientado à transmissão de mensagens sem o estabelecimento de uma conexão entre máquina fonte e destino, fornecendo uma comunicação menos confiável que o TCP. Ele envia as mensagens (sem conexão) e não oferece nenhuma garantia de entrega ou seqüência. O formato do cabeçalho do UDP é mostrado na Figura 14.

Source Port (16 bits)	Destination Port (16 bits)
Length (16 bits)	Checksum (16 bits)

**Figura 14 – Formato Cabeçalho UDP**

O campo *length* dá o tamanho do cabeçalho mais o campo de dados. O campo de *checksum* também é para o cabeçalho mais dados.

Pode-se notar pela Figura 14 que o tamanho do cabeçalho UDP é de 8 bytes.

### 3.5 TRABALHOS RELACIONADOS

Nos últimos anos têm sido realizados estudos avaliando o comportamento de redes GPRS. No entanto, cada um deles em geral avalia apenas poucos aspectos possíveis em um cenário que suporta protocolos de transporte, principalmente TCP, sobre GPRS.

Em [12] (*A Performance Evaluation of Internet Access via the General Packet Radio Service of GSM*) são apresentados pela primeira vez resultados de simulações para tráfego Web sobre GPRS considerando todas as camadas de protocolos envolvidas. Tais resultados foram obtidos através de um simulador GPRS apresentado no mesmo trabalho. O simulador desenvolvido pelos autores é baseado no paradigma de simulações orientadas a eventos, e foi implementado utilizando a ferramenta *Cadence Bones (Block Oriented Network Simulator)*, que suporta um sistema de modelagem hierárquico orientado por blocos [39]. Os autores realizam uma análise fim a fim, avaliando a vazão geral do sistema (no sentido *downlink*) para 30 usuários móveis. O simulador utiliza o TCP *Tahoe*, com segmentos de 1460 bytes. Para o GPRS, foi utilizada a configuração 4+1 (4 *timeslots* no sentido *downlink* e 1 no *uplink*) para oito canais de dados GPRS (PDCH), variando o cenário de codificação (CS-1 a CS-4) e o C/I (3 a 18 dB). O trabalho conclui que o GPRS é bastante apropriado para tráfego TCP/IP.

Em [3] (*Analysis of the General Packet Radio Service GPRS of GSM as Access to the Internet*), os mesmos autores analisam, agora, a vazão total para o CS-2, com C/I de 12db, variando o número de usuários (de 1 a 40), o número de *timeslots* utilizados no sentido *downlink* (1, 2 e 4) e o número de canais de dados (1, 4 e 8). Utilizando o tamanho do segmento TCP padrão (536 bytes). Todas as medidas de vazão são analisadas no nível de aplicação, ou seja, todo o *overhead* ocasionado pelos cabeçalhos dos protocolos, recuperação de erros e controle de fluxo (ex. TCP *slow start*) estão incluídos. A vazão é definida pelo

tamanho do pacote de um objeto WWW dividido pelo seu tempo de transmissão. Esta forma de cálculo é utilizada com objetivo de comparar os resultados para transmissão de objetos de diferentes tamanhos. Foi demonstrado também pelos autores, como um operador pode aumentar o desempenho de um sistema GPRS aumentando o número de canais de dados. Os resultados confirmam as conclusões anteriores de que o GPRS é bastante apropriado para tráfego de informações de aplicações Internet.

Utilizando o simulador desenvolvido em [12], Meyer apresenta em [13] (*TCP Performance over GPRS*) uma avaliação da harmonia entre o mecanismo ARQ (*Automatic Repeat Request*) do TCP e do GPRS (RLC). Apenas um usuário transferindo arquivos longos foi estudado. São utilizados 4 PDCHs e 4 *timeslots* no sentido *downlink*, variando o CS (de 1 a 4) e o C/I (de 6 a 30). O autor conclui que o ARQ da camada RLC é suficientemente rápido para retransmitir segmentos perdidos antes que ocorra um *timeout* no TCP. Assim, usando o CS apropriado para as condições do ambiente, não há perdas de segmentos TCP.

Aspectos relacionados à vazão geral do sistema e à taxa de pacotes de dados são avaliados em [14] (*Wireless Internet Access Based on GPRS*), utilizando ainda o mesmo simulador. Tal trabalho possui dois objetivos. O primeiro é avaliar a capacidade do sistema GPRS em termos de vazão sob diferentes condições de carga, variando o número de usuários (de 10 a 60) e o número de *timeslots* (no sentido *downlink*) por usuário (1, 2 e 4), o C/I (de 1 a 30 dB) e o cenário de codificação (CS-1 a CS-4). O segundo é analisar a perspectiva do usuário ao utilizar GPRS como meio de acesso a web, observando a taxa de pacotes de dados recebida pelo usuário. Para as simulações apresentadas neste trabalho os autores utilizaram um modelo de tráfego na web. Este modelo é dividido em duas partes, o modelo do cliente e o modelo do servidor. Enquanto o modelo do cliente determina o seu comportamento o modelo do servidor é responsável pela geração de tráfego. O modelo de comportamento do usuário é composto de ações ativas (requisição de uma página) e ociosidade (ex. lendo informações de uma página), determinando o comportamento do usuário. O Modelo do servidor leva em consideração o número de objetos (textos, ícones, figuras) de uma página *web*, e o tamanho de cada objeto. Os autores concluem que, comparado à comutação por circuito (GSM), o GPRS é significativamente superior.

Em [1], o desempenho do SACK e *Timestamps* em um sistema GPRS é estudado através da simulação da transferência de um arquivo de 512 Kbytes. Para isto o TCP-Reno é utilizado. Para simular um ambiente GPRS o autor utiliza o simulador baseado em eventos *Cadence Bones* [39]. Este modela a internet no simulador como um objeto com um *delay* e perda de pacotes cujos valores são definidos de forma randômica. O *Gateway GPRS Support*

*Node* (GGSN) é representado com um roteador, o *Serving GPRS Support Node* (SGSN) é modelado como um objeto com *delay* fixo, que representa o tempo de processamento desta camada. A comunicação GGSN-SGSN e SGSN-BSS (*Base Station Server*) são modelados com atrasos fixos, que levam em consideração a capacidade dos *links* (2Mbps e 64Kbps respectivamente). A estação móvel é modelada como um *host* TCP/IP com as mesmas características de um *host* fixo. Os parâmetros utilizados são: 4 canais de dados, CS-4, C/I de 24 dB e 15 usuários, sendo o segmento TCP de 431 bytes. O trabalho conclui que a utilização do TCP-Reno com SACK resulta em melhor desempenho e que, apesar de melhorar o desempenho do TCP-Reno, o uso de TCP-Reno com *Timestamps* atrapalha a *performance* do SACK.

Em [15] um protocolo de transferência de informações para computação veicular (VITP) é apresentado. O VITP é um protocolo que visa oferecer serviços através da comunicação entre veículos conectados através de redes sem fio, como condições das estradas, condições de tráfego, alertas de tráfego (acidentes de trânsito ou veículos danificados bloqueando tráfego), serviços de páginas amarelas (localização e lista de preços de postos de gasolina, localização e visualização de cardápios de restaurantes) utilizando muitas vezes informações de navegadores GPS no auxílio ao sistema de navegação de bordo. O protocolo proposto pelos autores possui uma especificação semântica e sintática de mensagens entre veículos, caracterizando um componente de software como proposta para realização de determinados serviços.

Um observador (VTIP PEER) armazena informações através de sensores localizados no veículo. O observador estabelece uma dinâmica de grupo *ad-hoc* que coleta, comunica, e mesclam informações do sensor a bordo de diferentes veículos com objetivo de resolver uma solicitação encaminhada. VITP é inspirado pelo protocolo de transferência de hipertexto *world wide web*.

Logo após, os autores apresentam um cenário motivador, introduzindo o modelo de serviços do protocolo. Em seguida, as especificações do protocolo são apresentadas. Por último, uma simulação, através do simulador de redes NS-2 é apresentada, utilizando-se um gerador de tráfego veicular para investigar a *performance* do protocolo em um cenário realístico. Os resultados das simulações executadas pelo autor concluem que o protocolo VITP é viável para utilização em redes VANET devido a forma como este foi definido.

### 3.6 CONSIDERAÇÕES FINAIS

Neste capítulo foi fornecida uma visão geral do sistema que serve como base para geração dos cenários que serão simulados nos próximos capítulos deste trabalho.

Além de uma visão geral do sistema, foram descritos o pacote de transmissão de informações básicas de telemetria e o pacote de transmissão de informações avançadas de telemetria, tendo sido detalhado as informações contidas em cada pacote. Esses pacotes serão trafegados sobre os diversos cenários gerados no simulador de redes NS-2. O funcionamento do sistema na ausência de sinal GSM/GPS também foi descrito. Foram abordados os protocolos de transporte TCP e UDP, sobre o qual os pacotes de transmissão de informações de rastreamento veicular serão transportados pela rede GPRS.

Por último, foram descritos alguns trabalhos relacionados, seus objetivos e suas limitações, servindo como material de base para a pesquisa realizada nesta dissertação.

No Capítulo 5, o ambiente de simulação utilizado para validação do modelo descrito será apresentado.

## 4 GPRS NO NETWORK SIMULATOR

Este capítulo descreve as principais características e conceitos relacionados ao ambiente de simulação utilizado neste trabalho, o *NetWork Simulator*. O módulo GPRS, implementado e instalado à parte, é descrito detalhadamente, sendo apresentados seus problemas e limitações, bem como as modificações realizadas para superá-los.

### 4.1 NETWORK SIMULATOR

O NS [39] é um simulador que tem sido utilizado com grande frequência em pesquisas em redes de computadores. Atualmente o seu desenvolvimento é suportado pelo DARPA (*Defense Advanced Research Projects Agency*, EUA) através do projeto SAMAN [30] e pela NSF (*National Science Foundation*, EUA) através do projeto CONSER [31], em colaboração com outros pesquisadores como o centro ICIR [32]. O simulador já recebeu apoio do *Lawrence Berkeley National Laboratory*, do Xerox PARC (*Palo Alto Research Center*), da Universidade da Califórnia em Berkeley, *Sun Microsystems* e também agrega diversos módulos contribuídos por pesquisadores independentes. É um software de código livre e fornecido gratuitamente [33]. Uma lista de discussão é mantida pelos desenvolvedores, onde os pesquisadores de diversas partes do mundo podem trocar idéias e experiências, e também 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 e também Brasil, contribuem para o valor desta ferramenta.

O NS é um simulador de eventos discretos, focado para o desenvolvimento de pesquisas em redes de computadores. Ele prevê suporte a TCP e variantes do protocolo, *multicast*, redes sem fio (*wireless*), roteamento e satélite. Tem facilidades de *tracing*, que é a coleta e registro de dados de cada evento da simulação para análise posterior. Possui um visualizador gráfico para animações da simulação (nam – *network animator*), *timers* e

escalonadores, modelos para controle de erros e algumas ferramentas matemáticas como gerador de números aleatórios e integrais para cálculos estatísticos. Inclui também uma ferramenta de geração de gráficos, o xgraph, e vários tipos de geradores de tráfego.

O NS foi desenvolvido na linguagem orientada a objetos C++, de forma modular. O uso desta linguagem nos módulos confere velocidade e mais praticidade na implementação de protocolos e modificação de classes. A interface com o usuário, configuração, estabelecimento de parâmetros e manipulação de objetos e classes é feita em modo texto, através da linguagem interpretada Otcl [29], que também é orientada a objetos.

#### 4.1.1 Estrutura do NS

A interface entre o usuário e o NS dá-se através da linguagem script Otcl. Segundo os desenvolvedores, a divisão em duas linguagens (Otcl e C++) objetiva dar ao simulador tanto velocidade e poder, quanto flexibilidade e facilidade de mudança de parâmetros. O núcleo do simulador é escrito em C++, conferindo velocidade, mas esta linguagem torna-se lenta para manipulação constante ou mudança de parâmetros. Otcl, por ser interpretada, é bem mais lenta, porém pode ser facilmente alterada. Além do mais, os objetos compilados são disponibilizados para o interpretador Otcl por *linkagem*, o que virtualmente cria um objeto Otcl para cada objeto C++, e que podem ser manipulados através das facilidades da Otcl.

A Figura 15 mostra a construção geral do NS. Um usuário comum atua no perímetro “tcl”, escrevendo scripts em Otcl e executando simulações. Os escalonadores de eventos e os componentes de rede são implementados em C++ e disponibilizados ao interpretador Otcl através de uma replicação feita pela camada tccl, que recria os objetos C++ em objetos Otcl, e que podem finalmente ser manipulados por esta última (processo denominado *linkage*). Todo o conjunto constitui-se no NS, que é um interpretador de Otcl com bibliotecas de simulação para redes de computadores.





**Figura 15 – Arquitetura do NS**

De forma a realizar uma simulação no NS é necessário, primeiramente, escrever um *script* em *Otcl*. Este *script* contém as seguintes partes básicas:

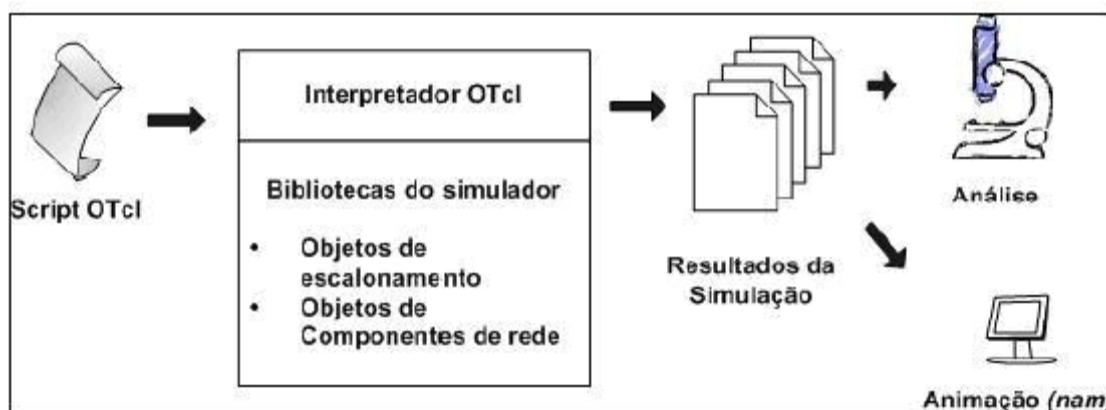
- criação do objeto Simulador;
- abertura de arquivos para *tracing* e análise posterior;
- criação da topologia de rede;
- criação de nós ou nodos;
- conexão dos nós entre si (*links*);
- criação das filas de saída;
- criação dos agentes de 4ª. Camada e conexão com *hosts*;
- criação dos geradores de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

O processo de simulação pode ser assim resumido (Figura 16):

- confecção do *script* (arquivo texto comum);
- execução do *script* com o comando `ns nomedascript.tcl`;
- arquivos de *tracing* serão gerados com registro de cada evento simulado.

Após concluída a simulação:

- imprimir estatísticas calculadas no *script*;
- visualizar os eventos com o *nam*;
- analisar resultados através dos arquivos de *tracing* com apoio de ferramentas apropriadas (*awk*).



**Figura 16 – Esquema de Utilização do NS**

Um ponto importante a observar é que o NS não fornece estatísticas de simulação de modo automático; estas devem ser obtidas através de procedimentos matemáticos no script ou pela manipulação de objetos especiais chamados monitores. Pode-se, ainda, usar ferramentas para análise dos arquivos de *tracing* gerados durante a simulação, que são os verdadeiros resultados da simulação, estes arquivos com formatação específica registram cada evento gerado pelos escalonadores. As ferramentas para análise dos arquivos de *tracing* devem então ser capazes de ler os dados gravados nestes arquivos e efetuar os cálculos desejados. Uma destas ferramentas, muito utilizada, é o *awk* [28], uma linguagem projetada para buscar padrões dentro de um arquivo e efetuar ações programadas. O animador *nam* pode também ser usado para analisar visualmente a simulação e obter algumas estatísticas, mas ele não é apropriado para análises mais profundas. Se nada for feito, o simulador apenas rodará o *script*, gerará os arquivos de saída (*tracing*) e encerrará, sem nada mostrar ao usuário. Em primeiro momento, esta característica um tanto não-amigável e não imediatista do NS pode frustrar o usuário iniciante. Entretanto, essa é a realidade de bom número dos simuladores existentes na categoria do NS, que não foram feitos com ótica didática.

#### 4.1.2 O Modelo *Wireless* no NS

O modelo *wireless* consiste essencialmente no *MobileNode* (*mobilenode.h*), com características adicionais que permite simulações de redes *ad-hoc multi-hop*, LANs *wireless*



No NS, os protocolos principais, os elementos de topologia, as ações de processamento de pacotes, entre outros, são escritos em arquivos separados na linguagem C++. Estes arquivos são agrupados através da linguagem Otcl na construção de um simulador.

Um nó sem fio é criado de maneira diferente de um nó fixo. Este possui uma estrutura em camadas e uma pilha de rede, como mostra a Figura 16. Este tem a capacidade de se mover em uma determinada topologia de rede, além de enviar e receber dados de um canal sem fio. As características de mobilidade de um nó incluem sua movimentação (baseado em duas posições dimensionais), alteração periódica de sua posição e gerenciamento de limites. Estas características são implementadas em arquivos C++ (ns/mobilenode.{cc,h}). Cada uma dessas camadas é implementada individualmente em arquivos C++ . A pilha é agrupada através da linguagem Otcl (ns/tcl/lib/ns-mobilenode.tcl). Por padrão, cada nó é configurado como um nó *ad-hoc* e pode enviar ou receber informações de qualquer outro nó móvel que esteja a seu alcance.

#### 4.1.2.1 A pilha de rede em um nó wireless no NS

A pilha de rede de um nó *wireless* é descrita a seguir:

*Link Layer* (LL) é responsável por simular o protocolo da camada de conexão dos dados. Esta fixa o endereço MAC do destino no cabeçalho Mac do pacote. Atualmente, esta camada simplesmente envia e recebe os pacote Mac. Esta possui também um módulo ARP que transforma um endereço IP para um endereço de maquina (Mac address).

*Address Resolution Protocol* (ARP). Se o ARP possui o endereço de maquina do destino é gravado no cabeçalho Mac do pacote, senão este realiza um pedido de resolução de endereço e armazena o pacote temporariamente. Para cada destino não encontrado, existe um espaço em memória onde apenas um pacote é armazenado. Se algum outro pacote com o mesmo destino for recebido à memória é sobrescrita.

*Interface Queue* (IFQ). Esta camada implementa uma fila de prioridades, que fornece prioridade para pacotes de roteamento do protocolo na fila e pode de forma seletiva remover pacotes de um certo destino.

*Medium Access Control* (MAC) *Layer*. Atualmente implementado no NS existem os padrões 802.11 e 802.3, CSMA e *multihop* (IEEE).

*Network Interface* (NETIF) é usada pelo nó móvel para obter acesso a um canal de dados. A interface *wireless* compartilhada é implementada como Phy/WirelessPhy. Esta interface é sujeita a colisões e recebe pacotes através do modelo de propagação via rádio. O

cabeçalho do pacote possui informações relacionadas à interface de transmissão, como a corrente (força) utilizada na transmissão, amplitude da onda, etc. Esta informação é utilizada pelo modelo de propagação para determinar se um pacote possui força suficiente para ser recebido. O modelo aproxima-se da interface da radio DSSS (Lucent WaveLan Direct-Sequence Spread Spectrum).

*Radio Propagation Model.* Este usa  $1/r^2$  para distâncias pequenas e uma aproximação do modelo *two-ray-ground* ( $1/r^4$ ) para distâncias longas.

#### 4.2.2.2 Agentes de roteamento

Um nó móvel é criado de acordo com o protocolo de roteamento *ad-hoc* que este irá usar. Este é configurado para ter as tabelas e estruturas de dados apropriadas. O simulador NS suporta DSR, DSDV, AODV e TORA. Estes, por padrão, podem se comunicar com todos outros nó móveis que tenham alcance.

#### 4.1.2.3 Cenários de rede fixa sem fio

Se um nó móvel deseja comunicar-se com outro nó de uma rede fixa, este pode realizar a comunicação conectando-se a uma estação base. Uma estação base é implementada como um nó móvel no NS mas com protocolos de roteamento de redes fixas e mobilidade desativada. Isto é feito no nível de programação Otcl em ns/tcl/lib/ns-bsnode.tcl. Roteamento hierárquico deve ser utilizado com partes fixas e sem fio dispostas em domínios diferentes.

Dois ou mais nós móveis ainda podem se comunicar, porém será necessário o uso de uma estação base para se comunicarem com um nó fixo. Este processo funciona perfeito com o padrão Mac 802.11 mas não para o GPRS.

#### 4.1.2.4 Um agente de roteamento não ad-hoc: NOAH

Para um cenário de comunicação GPRS, é necessário que cada estação móvel (MS) converse apenas com sua estação base (BS), e não com qualquer outra MS. Isto implica que não poderá ser utilizado um configuração *ad-hoc* padrão para nós móveis fornecida pelo NS. Ao invés disso, o agente de roteamento não *ad-hoc* (NOAH) construído por Joerg Widmer [6] foi utilizado. Este cria um nó móvel simples (sem tabelas de roteamento, sinalização, normalmente criado pelos nodos dsr / dsdv) que pode se comunicar apenas com sua estação

base (BS). O NOAH envia e recebe todos os pacotes através de sua BS. De forma inerente ele necessita o uso do roteamento hierárquico do NS. Deste modo, um nó NOAH pode ser utilizado também em um cenário fixo / sem fio. (ns/noah/noah.{cc,h})

No próximo capítulo será descrito o Modulo GPRS, extensão utilizada no NS, para tornar possível a simulação do protocolo sobre a rede GPRS.

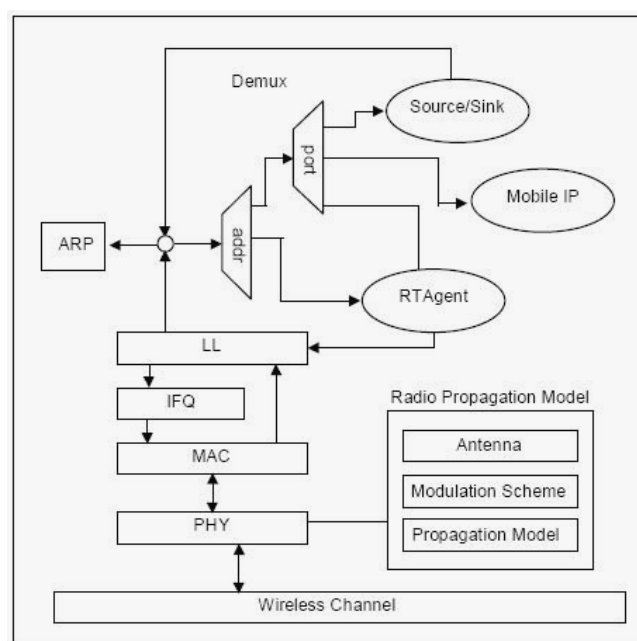
## 4.2 MÓDULO GPRS

O módulo do GPRS foi desenvolvido por Richa Jain [36] no *Indian Institute of Technology - Bombay*. Sua implementação focaliza a interação MS-BS e abrange as camadas referentes à interface aérea da especificação.

### 4.2.1 Estrutura do Módulo

Um nó móvel no NS apresenta a pilha de protocolos de rede ilustrada na Figura 18. Os componentes dessa pilha são brevemente descritos a seguir.

- *Link Layer (LL)*: responsável pela simulação dos protocolos de link de dados, esse componente altera o endereço MAC do destino no cabeçalho MAC do pacote. Atualmente, ele simplesmente passa pacotes de/para a MAC. Em um nó móvel, apresenta um módulo ARP conectado a ele que resolve as conversões de endereços IP em endereços MAC.



**Figura 18 – Pilha de Protocolos de Rede em um Nó Móvel no NS**

- *Address Resolution Protocol (ARP)*: este componente recebe consultas da camada LL. Se o endereço MAC já constar na sua tabela (*ARP table*), ele escreve o endereço no cabeçalho MAC do pacote. Caso contrário, ele envia um *ARP Request* e armazena temporariamente o pacote. Para cada destino desconhecido, há espaço para armazenar apenas um pacote. Assim, ao chegar um novo pacote para o mesmo destino, o pacote anterior é sobrescrito. Ao receber a resposta com o endereço MAC do destino, o pacote é inserido na fila (IFQ).

- *Interface Queue (IFQ)*: implementa uma fila de prioridades, a qual dá prioridade a pacotes de protocolos de roteamento, inserindo-os no início da fila. Também permite o uso de filtros sobre os pacotes na fila e remove aqueles que possuem um destino específico.

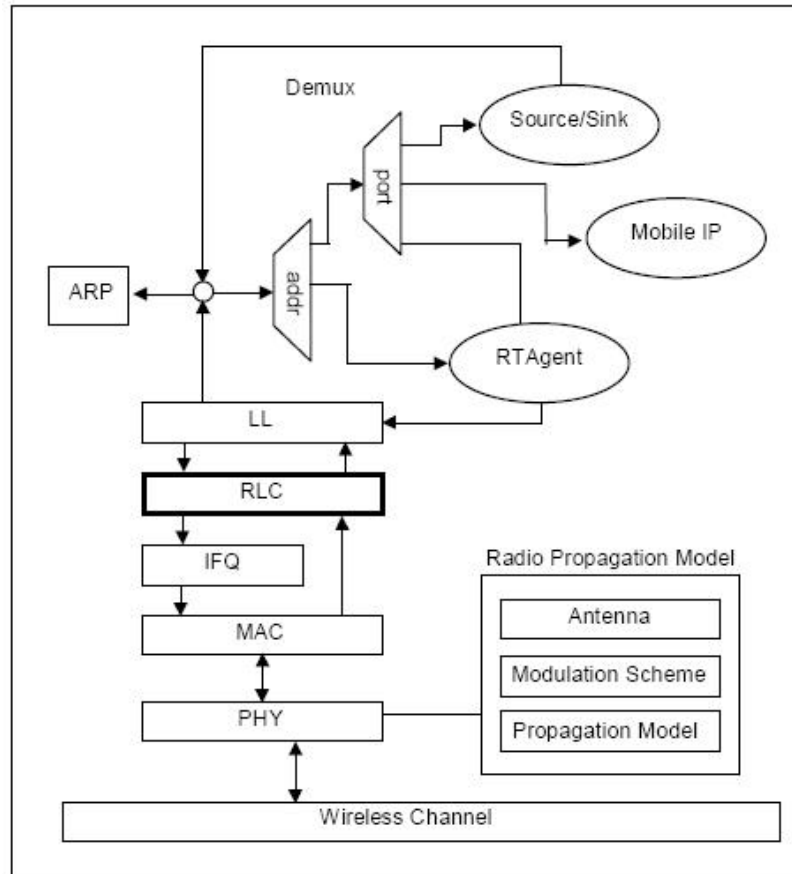
- *Medium Access Control Layer (MAC)*: atualmente implementa IEEE 802.11 e 802.3, CSMA (*Carrier Sense Multiple Access*) e *multihop*.

- *Network Interface (NetIF)*: é utilizado pelo nó móvel para acessar o canal. Essa interface é sujeita a colisões e recebe pacotes através do modelo de propagação de rádio.

- *Radio Propagation Model*: implementa o modelo  $1/r^2$  para distâncias pequenas e  $1/r^4$  (modelo two-ray-ground) para distâncias longas.

- *Wireless Channel*: simula a transmissão real do pacote na camada física. Implementa um meio compartilhado com suporte a mecanismos de contenção. Se mais de uma transmissão ocorre ao mesmo tempo, é sinalizada uma colisão, que pode ser tratada pela camada MAC.

O módulo GPRS incrementa a camada LL, adiciona a camada RLC (*Radio Link Control*) à pilha e cria uma nova versão para a MAC, específica para o GPRS. A pilha modificada pode ser visualizada na Figura 19. As modificações, assim como o novo componente, são descritas a seguir.



**Figura 19 – Pilha de Protocolos de Rede Modificada para um Nó Móvel GPRS**

#### 4.2.1.1 *Link Layer (LL)*

A camada LL do ns foi modificada pelo módulo GPRS, sendo incluídos fragmentação de pacotes e modo com confirmação.

No modo com fragmentação, os pacotes recebidos do agente transmissor são segmentados antes de serem passados para a RLC e são reagrupados no outro lado, antes de serem entregues ao agente receptor.

O modo com confirmação implementa um mecanismo de retransmissão *stop-and-wait*. Assim, se um pacote (LLC PDU) é perdido ou recebido com erro, o receptor “para e espera” que esse pacote seja reenviado, mandando um ACK para sinalizar qual o pacote esperado. Qualquer outro pacote recebido é ignorado.



#### 4.2.1.2 *Radio Link Control (RLC)*

Este componente implementa a camada de link de rádio e, assim como a LL, inclui fragmentação de pacotes e modo com confirmação.

No modo com fragmentação, os LLC PDUs são segmentados em RLC PDUs, sendo transmitido um RLC PDU por quadro.

O modo com confirmação implementa um mecanismo ARQ (*Automatic Repeat Request*).

Se um fragmento é perdido, um ACK duplicado do último fragmento recebido corretamente é enviado e os fragmentos seguintes são armazenados. Ao chegar o fragmento que estava faltando, este é colocado na ordem (isto é, na frente dos fragmentos que o sucedem). Os fragmentos são passados adiante apenas quando todos os fragmentos que compõem um LLC PDU chegarem.

No modo sem confirmação, se faltar um fragmento do LLC PDU, nenhum fragmento é passado adiante. Nesse caso, a LL se encarrega de retransmitir o LLC PDU (se estiver em modo com confirmação) ou deixa sob a responsabilidade da camada superior (por exemplo, TCP).

#### 4.2.1.3 *Medium Access Control (MAC)*

A MacGprs implementa a divisão da interface aérea em canais, a alocação desses canais, a transmissão de pacotes (fragmentos RLC) e de mensagens de sinalização nos sentidos *downlink* e *uplink*, e um modelo de erros randômico.

##### 4.2.1.3.1 Canais

Para formar os canais, a interface aérea é dividida em frequências FDMA e cada frequência dividida em quadros TDMA, compostos por oito *timeslots*. Um canal consiste em um par frequência/*timeslot*.

O número de frequências pode ser estabelecido pelo usuário e determina a quantidade de canais *uplink* e *downlink* a serem criados. O slot 0 de cada frequência (nos dois sentidos) é reservado para sinalização e *broadcasts*. O usuário também pode decidir quantos slots por frequência (PDCHs) são reservados para serviços GPRS. O restante fica disponível para serviços GSM.

#### 4.2.1.3.2 Estrutura dos slots TDMA

Cada quadro TDMA é composto por oito *timeslots*, nos quais cada *slot* tem duração de 577 microssegundos. O início e o fim de cada slot é marcado por um timer, havendo um timer distinto para cada sentido (*uplink* e *downlink*). Como determinado pela especificação (vide Seção 2.7), um quadro TDMA no sentido *uplink* é iniciado com um atraso de três *timeslots* do início de um quadro TDMA no sentido *downlink*.

Os quadros TDMA, em ambas as direções, deveriam ser sincronizados na MS e na BS. Mas um atraso finito de propagação entre a MS e a BS causa um desencontro entre eles, impedindo a sincronização. Assim, o relógio da MS deve ser adiantado o tempo desse atraso para que a sincronização ocorra. Contudo, para simplificar o código nessa implementação, esse atraso foi considerado nulo.

#### 4.2.1.3.3 Alocação de canais

Baseada no mecanismo ALOHA, a MAC é responsável pela alocação de canais, na qual cada MS pode alocar apenas um canal (ou seja, não foi implementado *multislot*). A alocação nos sentidos *uplink* e *downlink* é simétrica, sendo realizada através da troca de mensagens.

Ao receber da sua fila (IFQ) um pacote para transmitir, a MS solicita um canal enviando uma *Resource Request*. Quando recebe essa mensagem, a BS aloca um canal (se houver algum disponível) e retorna uma *Resource Reply*, informando à MS qual o canal alocado. Tanto a BS quanto a MS armazenam essa informação para uso posterior.

A BS pode alocar um canal para uma MS sem que esta tenha solicitado. Isso ocorre quando a BS recebe da sua fila um pacote destinado a uma MS que não tem canal alocado. Nesse caso, a BS aloca o canal e envia uma *Resource Reply* para a MS.

Após um determinado período de ociosidade (isto é, sem transmissão em qualquer direção), a MS solicita a liberação do canal através da mensagem *Resource Release*.

#### 4.2.1.3.4 Modelo de erros

A MAC também implementa um modelo de erros para validar o modo com confirmação da camada RLC. O modelo marca randomicamente um pacote (isto é, um fragmento RLC) como errado, resultando no seu descarte ao chegar ao destino.

### 4.2.2 Dificuldades Iniciais

A utilização do módulo GPRS apresenta algumas dificuldades iniciais. A primeira dificuldade consiste na sua instalação. O pacote de instalação do módulo não possui nenhum arquivo que explique como instalá-lo. Além disso, faltam no pacote arquivos de um outro módulo, denominado NOAH (*Non Ad-hoc Routing Agent*), que é utilizado pelo módulo GPRS. Assim, faz-se necessário primeiro instalar o módulo NOAH, que apresenta um manual de instalação passo a passo [34].

A segunda dificuldade encontrada se deve à ausência de transparência de ambos os módulos. A instalação deles modifica de forma inconseqüente arquivos que são utilizados por outras classes do NS. Um módulo bem estruturado deveria ser transparente para o resto do código. Contudo, isto não acontece. Após a instalação dos módulos GPRS e NOAH, à parte *wireless* do NS não funciona mais. Assim, foi necessário o isolamento dos módulos. Em outras palavras, todos os arquivos modificados por esses módulos que são compartilhados por outros tiveram que ser alterados de forma que o ns voltasse a funcionar exatamente como antes da instalação desses módulos.

Outra dificuldade é a falta de documentação detalhada. O módulo contém um resumido manual em formato HTML que o descreve de forma complexa. Várias leituras ao manual são necessárias para que se possa entender o funcionamento do módulo. Há também a dissertação de mestrado de Richa Jain [36]. O módulo GPRS é resultado de seu trabalho de mestrado. Porém a dissertação não acrescenta muita informação ao conteúdo do manual. Deste modo, o entendimento do código, pouco comentado, torna-se custoso.

### 4.2.3 Limitações e Problemas do Módulo

O módulo GPRS propriamente dito apresenta vários problemas e limitações. Entre suas limitações, podem ser citadas:

- Implementação restrita à interface aérea: apenas a interação entre o nó móvel (MS) e a estação rádio-base (BSS) é tratada. Os outros elementos da rede GPRS, como SGSN, GGSN e HLR, não estão implementados. Assim, mecanismos de *handovers* e *QoS* não podem ser estudados.

- Restringe a rede a uma única célula: por apresentar variáveis estáticas, apenas uma BS pode ser criada.

- Não permite a criação de mais de uma rede GPRS: uma vez que as variáveis da BS são estáticas, se outra rede (e conseqüentemente outra BS) for criada, essas variáveis serão compartilhadas pelas duas, havendo sobreposição de informações.

- O modo com confirmação limita o número de MSs: tanto na camada RLC como na LLC, o módulo com confirmação restringe o número de estações móveis a 1.

- Não implementa *multislot* : cada MS pode alocar apenas um canal.

- Desenvolvido para uma versão antiga do ns: a versão 2.1b7a.

Algumas dessas limitações foram superadas no trabalho atual e serão discutidas na próxima seção.

Além das limitações, descritas no próprio manual do módulo, foram encontrados vários problemas no código, os quais envolvem erros lógicos de implementação e falta de concordância com a especificação do GPRS. Tais problemas produziam resultados errados e, portanto, tiveram que ser resolvidos para que o módulo pudesse ser utilizado. A seguir são listados os problemas resolvidos no presente trabalho cuja solução será descrita na próxima seção:

- vazão de chegada maior que a de saída;
- a combinação de certos parâmetros tornava a vazão nula;
- mecanismo de alocação de canais com falhas;
- vazamento de memória;
- modelo de erros só aplicado no sentido *downlink*;
- MS e BS não eram separadas;
- fila única da BS compartilhada por todas as MSs;
- a transmissão dos blocos RLC não estava de acordo com a especificação;
- o modo com confirmação da camada RLC não estava de acordo com a especificação;
- *buffers* ilimitados (pequeno vazamento de memória);
- *timers* desnecessários utilizados (problema de desempenho).

A seguir, serão descritos detalhadamente cada um dos problemas citados acima e as soluções encontradas.

### 4.3 MODIFICAÇÕES REALIZADAS

Esta seção descreverá as modificações realizadas no módulo GPRS a fim de torná-lo mais confiável e mais próximo da especificação.

Para uma melhor compreensão, cada problema citado na seção anterior será detalhado, seguido da solução aplicada para repará-lo.

#### 4.3.1 Correção da Vazão

Utilizando os mesmos parâmetros contidos no *script* exemplo que acompanha o módulo, foram realizados testes para verificar a vazão alcançada pelo módulo GPRS. Anexando uma fonte CBR a uma estação móvel, detectou-se que o destino recebia uma quantidade de *bytes* maior que a quantidade enviada. O problema consistia na inclusão indevida de um cabeçalho IP realizada pelo módulo NOAH. Retirando essa linha de código, a quantidade de bytes enviada tornou-se igual à quantidade recebida.

Para entender melhor o funcionamento do módulo, os parâmetros das camadas RLC e LL foram variados. Descobriu-se que determinadas combinações de parâmetros tornavam a vazão nula. Isso ocorria devido a falhas no código que se tornavam visíveis apenas em algumas situações. Um exemplo é a alocação de canais que, em alguns casos, libera o canal errado mas, dependendo da combinação de parâmetros utilizada, consegue transmitir sem que esse erro seja percebido observando-se apenas a vazão. Esse e outros exemplos serão detalhados adiante.

#### 4.3.2 Robustez no Mecanismo de Alocação de Canais

A alocação de canais demonstrou instabilidade devido a alguns casos não tratados. O ponto principal, motivo que acarreta nos outros problemas, consiste na “liberação por partes”. Tanto a MS quanto a BS possuem variáveis que armazenam o canal alocado. Ao solicitar a liberação do canal (enviando uma mensagem *Resource Release*), a MS já considera seu canal liberado, reinicializando sua variável. Mas a BS só liberará o canal quando receber a *Resource Release*. Assim, o período compreendido entre o envio da mensagem (pela MS) e o seu recebimento (pela BS) apresentam inconsistência, uma vez que, durante esse intervalo de tempo, a MS considera não ter mais canal reservado para ela, enquanto a BS ainda considera o canal alocado. Essa brecha permite o surgimento de alguns problemas.

O primeiro problema ocorre quando o modelo de erros é usado. Se a mensagem *Resource Release* fosse marcada como errada e, conseqüentemente, descartada, a BS não liberava o canal. Isso causava dois comportamentos indesejados. Em um caso, a BS recebia pacotes para a MS e enviava no canal que ainda considerava alocado para a mesma. Contudo, como a MS já havia liberado seu canal, ela não recebia o pacote (pois são recebidos apenas mensagens de sinalização e broadcast ou pacotes que se destinam a ela e que foram transmitidos no canal reservado). Em outro caso, a MS recebia um pacote para transmitir e, como não possuía mais canal alocado, solicitava um novo canal. Ao receber tal solicitação (*Resource Request*), a BS reservava um novo canal para a MS, ficando, então, dois canais reservados, mas apenas um sendo utilizado. Quando a taxa de erros era muito grande, isso ocorria com bastante freqüência, ficando vários canais ociosos alocados para uma mesma MS. Por simplicidade, a solução adotada foi o não descarte de mensagens de sinalização, ou seja, mensagens de sinalização (que incluem o *Resource Release*) nunca são marcadas pelo modelo de erros.

O segundo problema identificado foi à liberação de canais errados. Ao receber um *Resource Release*, a BS liberava o primeiro canal alocado para a MS, pois este deveria ser o único. Contudo, como já foi descrito acima, havia situações em que durante um curto espaço de tempo a MS possuía dois (ou mais) canais reservados, apesar de estar ciente de apenas um deles. O algoritmo de alocação de canal é o mais simples possível: aloca-se o primeiro canal disponível. Inicialmente, canais eram alocados na seqüência. Porém, após um determinado tempo de simulação, ocorria à liberação de canais. Logo, surgiam “buracos” na seqüência, ou seja, existiam canais livres intercalados com canais reservados. Assim, aconteciam situações em que, havendo um canal reservado para uma MS (por ainda não ter recebido sua mensagem de liberação), a BS reservava um novo canal para a MS, onde o novo canal encontrava-se antes do antigo na ordem de precedência (por exemplo, o novo canal era o par freqüência 1/slot 5 e o antigo, o par freqüência 3/slot 2). Então a *Resource Release* chegava à BS e esta liberava o primeiro canal alocado para a MS, que, neste caso, era o canal novo, isto é, o canal que estava sendo utilizado. Apesar de o conceito empregado estar errado (uma vez que a origem transmitia em um canal e o destino recebia em outro), em algumas combinações de parâmetros essa situação funcionava, ou melhor, havia transferência normal de pacotes. Um exemplo seria quando a MS estivesse transmitindo para a BS com as duas camadas de link (RLC e LL) no modo sem confirmação. Mas se a transmissão fosse no sentido contrário ou uma das duas camadas estivesse em modo com confirmação, a vazão tornava-se nula. Como, mesmo no caso em que funcionava, a lógica estava errada, a solução

encontrada para esse problema foi incluir na mensagem de *Resource Release* os dados do canal a ser liberado (isto é, frequência/slot), evitando, assim, a liberação do canal errado.

Outro problema não tratado era a falta de canais disponíveis. Dependendo do número de frequências utilizadas e da quantidade de MSs, ocorria de não haver canais livres para serem alocados. Quando a BS não encontrava nenhum canal disponível, ela simplesmente não alocava nenhum canal, o que estava correto. Contudo, uma mensagem *Resource Reply*, com valores aleatórios nos campos “freq” e “slot”, era enviada. Além de ocorrerem casos em que uma MS estava usando o canal reservado para outra, havia também casos em que os valores aleatórios estavam fora do intervalo válido, o que gerava acesso indevido de memória e conseqüente “falha de segmentação”. A solução adotada consiste em enviar uma *Resource Reply* apenas quando um canal for alocado.

### 4.3.3 Eliminação de Vazamento de Memória

O módulo GPRS apresentou um grande vazamento de memória, o qual se encontrava na camada MAC e foi herdado da MacTDMA, na qual a MacGPRS se baseia.

A interface aérea do NS, implementada pela camada *Channel*, é compartilhada por todas as redes sem fio. Ao receber um pacote, essa camada manda uma cópia deste para cada nó móvel. Cabe ao nó identificar se o pacote é destinado a ele e, em caso negativo, descartá-lo. Assim como a MacTdma, a MacGprs não descartava os pacotes que não lhe pertenciam. Esses pacotes eram simplesmente ignorados. Dessa forma, quanto mais nós móveis houvessem em uma simulação e quanto maior fosse o tráfego gerado, maior era a quantidade de memória desperdiçada, pois para cada pacote enviado, eram geradas  $n$  cópias (onde  $n$  representa o número de MSs) e  $n-1$  destas eram ignoradas, acumulando memória. Esse vazamento foi solucionado liberando essas cópias.

Além desse vazamento principal, haviam outros vazamentos menos significativos. Em vários pontos do código, nas três camadas, pacotes que não eram mais utilizados eram simplesmente ignorados, acumulando memória por não serem liberados. Como exemplo, podem ser citados os pacotes de sinalização da MAC e os ACKs das camadas LL e RLC, os quais são criados no transmissor e não eram liberados no receptor. Cada caso desses foi identificado e o problema solucionado, ou seja, todos os pacotes não mais utilizados agora são liberados.

#### 4.3.4 Melhoria no Modelo de Erros

O modelo de erros estava sendo aplicado apenas no sentido *downlink*. A solução seria simplesmente aplicar o modelo também no sentido *uplink*. Porém, uma vez que o modelo de erros era muito simples, optou-se por mudá-lo, tornando-o mais próximo do real.

O modelo de erros atualmente implementado consiste na utilização de uma tabela com valores de *block error rate* (BLER) pré-calculados, fornecida pela Ericsson e adotada em [3], [25], [12] e [22]. O cenário assumido para a geração da tabela foi um canal urbano (*Typical Urban Channel - TU*) com usuários s imentando a 50 km/h.

**Tabela 8 – BLER**

C/I	CS-1	CS-2	CS-3	CS-4
0	0,78	0,9	0,95	0,9998
1	0,6636	0,8681	0,9393	0,9961
2	0,5612	0,824	0,919	0,9938
3	0,4714	0,77	0,8896	0,992
4	0,357	0,6753	0,8303	0,9889
5	0,2624	0,5708	0,755	0,983
6	0,1837	0,4641	0,6662	0,9715
7	0,122	0,3626	0,5672	0,9503
8	0,0781	0,2728	0,4624	0,9161
9	0,0478	0,1968	0,361	0,8701
10	0,0266	0,1355	0,2717	0,8143
11	0,0132	0,0888	0,1966	0,7471
12	0,0067	0,0558	0,1359	0,6681
13	0,0039	0,0338	0,0896	0,5827
14	0,002	0,0198	0,0571	0,4972
15	0,0006	0,0114	0,0355	0,415
16	0,00025	0,0061	0,0214	0,3384
17	0,0001	0,0031	0,0127	0,2689
18	0	0,0016	0,0075	0,2079
19	0	0,0007	0,0038	0,1574
20	0	0,00025	0,0018	0,1185
21	0	0,00009	0,0009	0,088
22	0	0	0,0004	0,0626
23	0	0	0,0002	0,0429
24	0	0	0,00007	0,0296
25	0	0	0	0,0205
26	0	0	0	0,0135
27	0	0	0	0,0084
28	0	0	0	0,0053
29	0	0	0	0,0036
30	0	0	0	0,0024

Fonte: ERICSON



A Tabela 8 mostra os valores de BLER utilizados. Esses valores estão em função da taxa de *carrier-to-interface* (C/I) e do cenário de codificação. A taxa C/I indica as condições do ambiente, isto é, indica se a estação móvel está sob alta, média ou baixa interferência. Como pode ser observado na tabela, quanto menor o valor de C/I, maior a taxa de erros, ou seja, maior a interferência.

O modelo funciona da seguinte forma. Cada nó possui um CS e um C/I, os quais são determinados no início da simulação. Ao receber um pacote, o nó sorteia um valor aleatório entre 0 e 1, utilizando uma distribuição Normal. Esse valor é comparado então ao valor da tabela BLER que é obtido utilizando o CS e o C/I do nó. Se o valor sorteado for menor ou igual ao BLER coletado, o pacote é marcado como errado e, posteriormente, descartado.

#### **4.3.5 Separação entre MS e BS**

Um problema que gerou vários outros consiste na ausência de separação entre a BS e a MS. As duas deveriam apresentar estruturas diferentes, pois a BS possui vários atributos que não se aplicam à MS, ou seja, deveriam ser classes diferentes. Contudo, as duas contêm os mesmos atributos e a diferenciação entre elas é feita utilizando variáveis estáticas, sendo estas utilizadas apenas pela BS. Além de ferir o conceito de orientação a objetos, essa abordagem trouxe outros problemas, como a limitação do número de BSs por rede e de redes GPRS por cenário: apenas uma rede com uma única BS pode ser utilizada em um cenário. Isso ocorre porque, dado que as variáveis são estáticas, é criada apenas uma variável por classe. Logo, se fossem criadas duas redes GPRS, as BSs compartilhariam os atributos estáticos, havendo sobreposição de dados nesses atributos. Este problema foi resolvido eliminando as variáveis estáticas e separando parcialmente as estruturas da BS e da MS.

O código atual permite a criação de mais de uma rede GPRS. E, apesar de não haver mais limitações no código com relação ao número de BSs por rede (isto é, poderiam ser utilizados mais de um objeto BS sem maiores dificuldades), apenas uma BS continua sendo utilizada por não haver ainda a implementação do BSC (*Base Station Controller*), responsável por gerenciar mais de uma BS, tratando, por exemplo, do mecanismo de *handovers*.

### 4.3.6 Divisão da Fila de Pacotes da BS

Como visto no tópico anterior, o módulo do GPRS não fazia distinção clara entre a BS e a MS. Assim, o comportamento das duas em muitos casos era semelhante. Um desses casos era o tratamento dos pacotes a serem transmitidos.

Entre as camadas RLC e MAC, há uma fila de prioridades que mantém os pacotes vindos da RLC. A MAC recebe da fila e processa apenas um pacote por vez, recebendo o próximo pacote após enviar o pacote atual. Em um nó normal, isto é, uma MS, este comportamento é adequado. Contudo, em uma BS não. Porque isso significa que a BS só processará e transmitirá um pacote por vez, quando, na prática, pode transmitir em várias frequências ao mesmo tempo. Assim, decidiu-se criar filas separadas por canais (par frequência/slot), permitindo que a BS possa transmitir em todas as frequências simultaneamente.

### 4.3.7 Transmissão dos Blocos RLC de Acordo com a Especificação

Na implementação original do módulo, um fragmento RLC era transmitido em um único quadro. De acordo com a especificação [11], um bloco RLC deve ser transmitido em quatro quadros consecutivos. Então, considerando que um fragmento corresponde a um bloco RLC, um fragmento deveria ser transmitido em quatro quadros.

Por default, um fragmento possui 50 bytes. O número de bytes de dados úteis de um bloco RLC depende do cenário de codificação (CS) utilizado (vide Seção 2.7.4). O CS-4 permite a transmissão de 51 bytes de dados por bloco. Assim, pode-se considerar que, na implementação original, a taxa de dados máxima atingida deveria ser próxima da taxa do CS-4: 21,4 kbps. Contudo, uma vez que um bloco era transmitido em um único quadro, a taxa alcançada era aproximadamente 4 vezes maior que a taxa do CS-4.

Esse erro foi corrigido e a implementação atual encontra-se em conformidade com a especificação, ou seja, a transmissão de um bloco RLC dura quatro quadros.

### 4.3.8 Correção do Modo com Confirmação da Camada RLC

No módulo GPRS original, o modo com confirmação da camada RLC funcionava da seguinte forma. Um pacote (fragmento) era enviado e guardado em um buffer, para uma possível retransmissão. Cada pacote era numerado para tornar possível a preservação da ordem dos pacotes no destino. Ao receber um pacote, o destino enviava um ACK do último

pacote recebido corretamente, exceto se o pacote já tivesse sido recebido (isto é, se seu número fosse menor que o número do pacote esperado). Se algum fragmento fosse perdido ou recebido com erros, os fragmentos seguintes eram armazenados em um buffer até que o segmento que estava faltando fosse recebido. O transmissor podia enviar até 64 segmentos sem receber nenhum ACK. Após este valor, ele parava de transmitir e esperava receber um ACK. Passado um determinado tempo sem receber nenhum ACK, o transmissor reenviava o último pacote transmitido. Após três reenvios, o transmissor desistia de enviar pacotes e permanecia esperando por um ACK. O mesmo acontecia no receptor quando este não recebia nenhum pacote durante um certo tempo. O último ACK era retransmitido três vezes e, depois disso, o receptor ficava parado esperando um pacote. Pode-se perceber que ocorriam casos em que os dois lados paravam, tornando a vazão nula.

Inicialmente, trabalhou-se sobre este modo da forma como estava implementado, consertando os casos em que a vazão era reduzida a zero. Contudo, depois foi observado que algumas partes dessa implementação não estavam de acordo com a especificação.

A especificação determina que um ACK consiste em um mapa de bits com 128 elementos, onde cada bit indica o estado de cada bloco transmitido (0 = bloco não recebido corretamente; 1 = bloco recebido corretamente) na janela. Além disso, um ACK deve ser enviado quando solicitado pelo transmissor (por exemplo, quando a janela está completa) e ao receber o último bloco de uma seqüência. Quando a janela fica completa, o transmissor deve parar de enviar blocos novos e começar a retransmitir os blocos não reconhecidos, avisando que a janela está cheia e que necessita de um ACK. Comparando com a implementação original do módulo, não é difícil notar que este não seguia a especificação: um ACK continha apenas o número do último pacote recebido corretamente; era enviado um ACK cada vez que um pacote novo era recebido e, ao completar a janela, apenas o último pacote transmitido era reenviado, e apenas três vezes.

Assim, o modo com confirmação da camada RLC foi remodelado, e, atualmente, encontra-se de acordo com a especificação.

#### **4.3.9 Limitação dos *Buffers***

A RLC armazena os pacotes (fragmentos) a serem enviados em um *buffer*. Depois percorre esse buffer, passando para a IFQ os pacotes enquanto houver espaço na janela. Se a janela estiver completa, os pacotes seguintes permanecem no buffer, até que haja espaço na janela para transmiti-los.

Na implementação original do módulo, o *buffer* não possuía limite de tamanho. Assim, quando a taxa de transmissão da origem era muito superior à alcançada pelo GPRS (ou seja, RLC recebia muito mais pacotes do que podia transmitir), o *buffer* continuava crescendo de forma ilimitada. Isso gerava um consumo exagerado de memória durante a simulação. Além disso, considerando que uma estação móvel possui, na realidade, uma pequena quantidade de memória, não faz sentido armazenar milhares de pacotes. Decidiu-se então limitar o *buffer* para armazenar dois LLC PDUs, eliminando o uso desnecessário de memória.

A camada LL também armazena os pacotes a serem transmitidos. Apesar de não ter sido observado armazenamento excessivo de pacotes, como no caso da RLC, optou-se por prevenir tal situação, limitando também seu *buffer* para armazenar apenas dois pacotes recebidos da camada superior.

#### 4.3.10 Redução do Uso de *Timers*

O início e o fim de cada slot é marcado por um *slotTimer*. Os slots no sentido *downlink* são delimitados pelo *downslotTimer* e no sentido *uplink* é usado o *upslotTimer*.

Cada nó precisa estar ciente do *slotTimer* no sentido no qual transmite, uma vez que a transmissão é realizada durante o slot que possui o canal apropriado para a transmissão. Contudo, a recepção não está vinculada ao *slotTimer* no sentido oposto. Em outras palavras, o receptor não verifica o *slot* atual para saber se deve ou não receber o pacote. Logo, não é necessário que um nó tenha conhecimento do *slotTimer* no sentido no qual recebe.

Na implementação original do módulo, cada nó possuía os dois *slotTimers*. Visto que um nó não utiliza os dois, na implementação atual o *slotTimer* desnecessário foi eliminado, havendo ganho no desempenho. Assim, cada nó possui apenas o *slotTimer* no sentido no qual transmite. Isto é, a BS apresenta apenas o *downslotTimer* enquanto a MS possui apenas o *upslotTimer*.

#### 4.4 ADIÇÕES REALIZADAS

Além das modificações realizadas no módulo original, algumas características foram adicionadas ao módulo GPRS visando torná-lo mais próximo da especificação. As principais adições realizadas serão descritas a seguir.

#### 4.4.1 Implementação dos Cenários de Codificação

O módulo do GPRS não apresentava nenhum cenário de codificação na camada RLC. O manual do módulo diz que foi utilizada uma carga útil de dados de 50 bytes por fragmento RLC, simulando o tamanho médio de dados enviados pela RLC. Como visto anteriormente (Seção 4.3.7), essa implementação estava errada, pois gerava uma taxa de transmissão irreal, aproximadamente quatro vezes maior que a taxa obtida com o CS-4.

Buscando tornar o módulo mais próximo do real, foram implementados os quatro cenários de codificação. Cada fragmento representa um bloco RLC, transmitido em quatro quadros consecutivos, e a quantidade de dados (bytes) enviados por bloco é determinada pelo CS escolhido. Os valores podem ser observados na Tabela 1, na Seção 2.7.4.

#### 4.4.2 Implementação de *Multislots*

O GPRS permite que um usuário use mais de um *timeslot* em paralelo nos sentidos downlink e *uplink*. Quanto mais *slots* o usuário puder usar ao mesmo tempo, maior será sua taxa de transmissão. Teoricamente podem ser utilizados os oito *timeslots* em paralelo, contudo na prática são usados no máximo cinco, sendo as configurações indicadas 1+1 (1 canal downlink e 1 *uplink*), 2+1, 4+1, 1+2, 1+4 [19].

O módulo do GPRS original implementa apenas a configuração 1+1, isto é, um *timeslot* por usuário nos dois sentidos. Essa configuração limita a taxa máxima a aproximadamente 21,4 kbps utilizando CS-4. Visto que, na prática, pode-se utilizar mais de um *timeslot* no sentido downlink / *uplink*, decidiu-se implementar o conceito de *multislot*, ou seja, agora é possível escolher o número de slots por usuários.

#### 4.5 CONSIDERAÇÕES FINAIS

Esse capítulo descreveu o ambiente de simulação utilizado e modificado nesse trabalho.

Inicialmente, foi realizada uma breve descrição do *Network Simulator* (NS), abrangendo sua origem e suas características. Entre estas, encontra-se o fato de o NS ser código aberto, o que permite sua modificação e possibilita que outros pesquisadores contribuam com melhoramentos no código existente e desenvolvimento de novos módulos.

O módulo GPRS, que representa uma dessas contribuições, foi apresentado, contendo uma descrição detalhada de sua estrutura. Em seguida, foram expostas suas limitações e seus principais problemas.

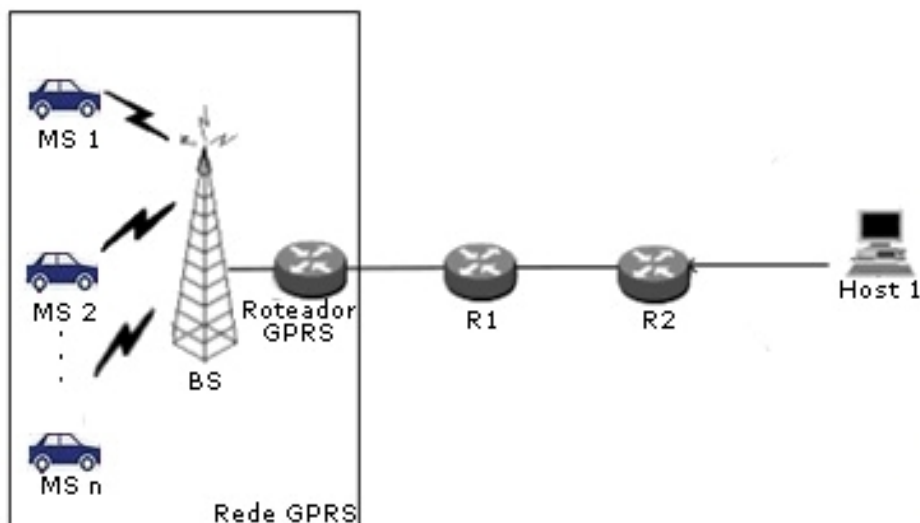
Finalmente, as modificações realizadas no módulo GPRS a fim de superar as limitações e problemas apresentados por ele foram descritas. O próximo capítulo apresentará os cenários utilizados e os resultados obtidos com o uso desse módulo modificado.

## 5 AVALIAÇÃO DE DESEMPENHO

Esse capítulo apresenta os aspectos relacionados às simulações que foram realizadas com o objetivo de estudar o envio de informações de rastreamento veicular sobre os protocolos TCP e UDP em uma rede GPRS. São descritas as topologias, os parâmetros utilizados e as métricas coletadas nas simulações. Os resultados obtidos são apresentados e discutidos detalhadamente.

### 5.1 CONFIGURAÇÕES DE SIMULAÇÃO

O ambiente de simulação utilizado foi o NS-2, descrito no Capítulo 5. A Figura 21 ilustra a topologia utilizada nas simulações: ela é formada por uma rede GPRS, dois roteadores intermediários (R1 e R2) e um sistema final (*host*). A rede GPRS é constituída por um roteador GPRS, que une as funções do GGSN e do SGSN, uma estação rádio-base (BS) e  $n$  estações móveis (MSs). Os enlaces fixos apresentam capacidade de 1Mbps e atraso de 10ms.



**Figura 21 – Topologia Utilizada nas Simulações**

O número de sistemas finais é igual a um, ou seja, existe um servidor central que coleta os dados de todos os MSs. A transmissão de dados ocorre no sentido MS - *host* (isto é, no sentido *uplink*), simulando um veículo enviando suas informações de rastreamento.

A topologia escolhida é simples visando facilitar a interpretação dos resultados. Em outras palavras, optou-se por uma topologia que permitisse estudar os aspectos desejados com relação ao comportamento da transferência de pacotes de informação de rastreamento sobre os protocolos de transporte TCP e UDP sobre a rede GPRS sem, contudo, encobrir ou influenciar esse comportamento.

Os parâmetros utilizados na simulação são apresentados na Tabela 9. Foi analisado a versão Reno do TCP. Também foi avaliado o protocolo UDP, para fins de comparação, por permitir a utilização de aplicações que geram taxas de transmissão constantes (CBR – *Constant Bit Rate*) e não ser baseado em confirmações. Foram utilizados três tamanhos diferentes de pacotes TCP (61, 85 e 1500 bytes) e UDP.

**Tabela 9 – Parâmetros de Simulação**

Parâmetro	Valor
Protocolo de transporte	TCP, UDP
Tamanho do pacote TCP	61, 85, 1500 bytes
Número de fontes/usuários	1, 5, 50
Taxa de erros (C/I médio)	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30 dB
Variância da taxa de erros	2 dB
Cenário de codificação	1, 2, 3, 4
Modo RLC	com confirmação (ACK), sem confirmação (UNACK)
Multislot – uplink	1, 2, 4 TS
Multislot – downlink	1 TS
Número de PDCH	5
Tempo de simulação	100 segundos

Todos os experimentos foram realizados com 1, 5 e 50 fontes de dados móveis (simulando veículos). A taxa de erros da interface aérea (C/I médio) variou de 2 a 30 dB, com intervalo de 2 dB entre elas. Os experimentos incluíram os quatro cenários de codificação (CS-1, CS-2, CS-3 e CS-4), que foram configurados estaticamente. Foi utilizado um *timeslot* no sentido da BS para a MS (*downlink*) e um, dois e quatro *timeslots* no sentido *uplink* (configurações 1+1, 1+2 e 1+4). Para todas as simulações foi utilizado um número de frequências suficiente para todas as estações transmitirem e, em cada frequência, cinco canais de dados (PDCH – *Packet Data Channels*) foram permitidos.

As simulações tiveram duração de 100 segundos, que é um tempo suficiente para que os efeitos esperados pudessem ser observados. Para cada experimento, foram realizadas 100



replicações, variando os valores escolhidos. Esse número de replicações foi escolhido por representar um compromisso aceitável entre o custo computacional e a confiança estatística nos resultados.

Durante a execução dos experimentos, foram coletadas várias métricas, tanto relacionadas ao tamanho dos pacotes de informações de rastreamento veicular quanto aos protocolos de transporte (TCP e UDP) e quanto ao GPRS. As principais métricas TCP/UDP coletadas são vazão, tamanho da janela de congestionamento, número de segmentos transmitidos e número de segmentos retransmitidos. No que se refere ao GPRS, foram coletados dados da camada RLC, como número de blocos transmitidos e descartados, C/I e CS.

## 5.2 RESULTADOS

Essa seção apresenta os resultados mais significativos dos vários experimentos de simulação realizados utilizando o cenário ilustrado na Figura 20, variando os parâmetros exibidos na Tabela 9.

### 5.2.1 Transmissão de Informações de Rastreamento Veicular sobre GPRS Utilizando o Protocolo TCP

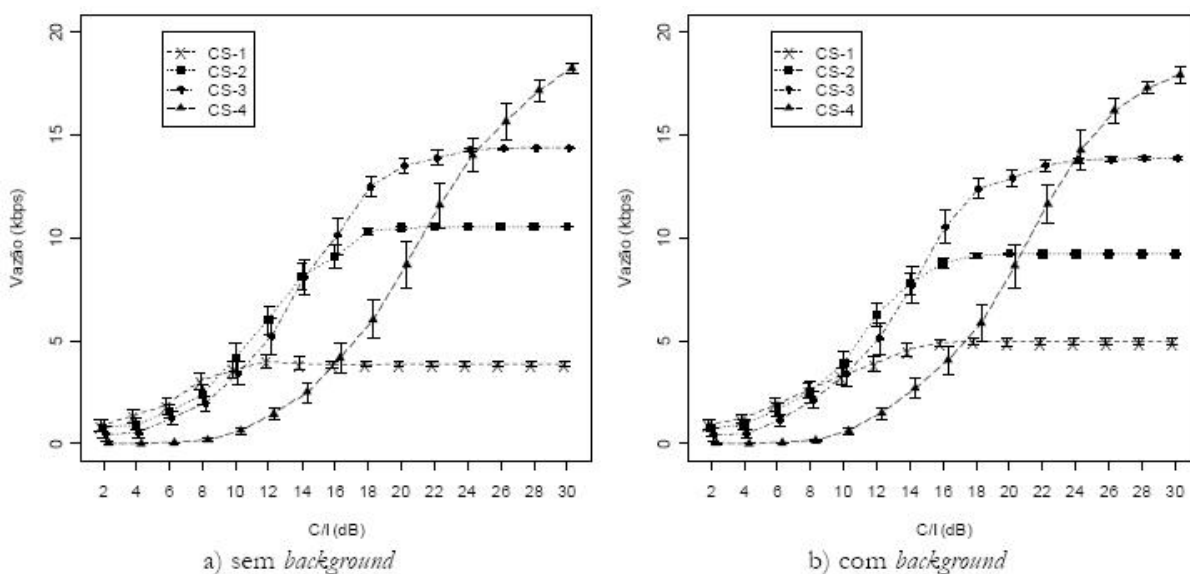
Os primeiros experimentos estudam o comportamento da transmissão de informações de rastreamento veicular sobre redes GPRS utilizando o protocolo de transporte TCP. Para efeitos de simulação, considerou-se que um veículo envia um pacote de informação de rastreamento, seja ele no formato básico (PT\_IBT) ou avançado (PT\_IAT) a cada 1 segundo. Este tipo de transferência sobre a rede GPRS não apresenta nenhum tipo de gargalo no cenário criado, devido ao fato do tamanho dos pacotes transferidos (em média 70 bytes) serem muito inferiores à vazão da rede GPRS para o cenário proposto.

Optou-se por testar, principalmente, o cenário de ausência de sinal GPS/GPRS, onde um veículo teve suas informações de rastreamento gravadas de forma local, na memória de massa do aparelho. Considerando este caso, o veículo deve enviar todos os pacotes de rastreamento armazenados localmente no momento de restabelecimento de sinal, aumentando consideravelmente o volume de dados a serem trafegados pela rede GPRS.

A Figura 22 mostra a vazão alcançada por uma estação móvel submetida a diferentes condições de ambiente. Foi utilizada a configuração 1+1 (um *timeslot* no sentido *downlink* e um no sentido *uplink*) e pacotes de 1500 bytes.

### 5.2.1.1 Tráfego de Retaguarda (*Background*)

Nestes experimentos, foram gerados tráfegos de retaguarda (*background*) entre sistemas finais conectados entre os roteadores intermediários (R1 e R2) para observar o impacto resultante no tráfego GPRS. Na Figura 22 são apresentados os resultados com e sem esse tráfego. Comparando os gráficos, pode-se observar que a presença do tráfego de retaguarda não influencia de forma significativa os resultados, uma vez que estes estão bastante parecidos. Assim, como não houve impacto significativo no tráfego GPRS, os próximos resultados não apresentarão mais tráfego de retaguarda.



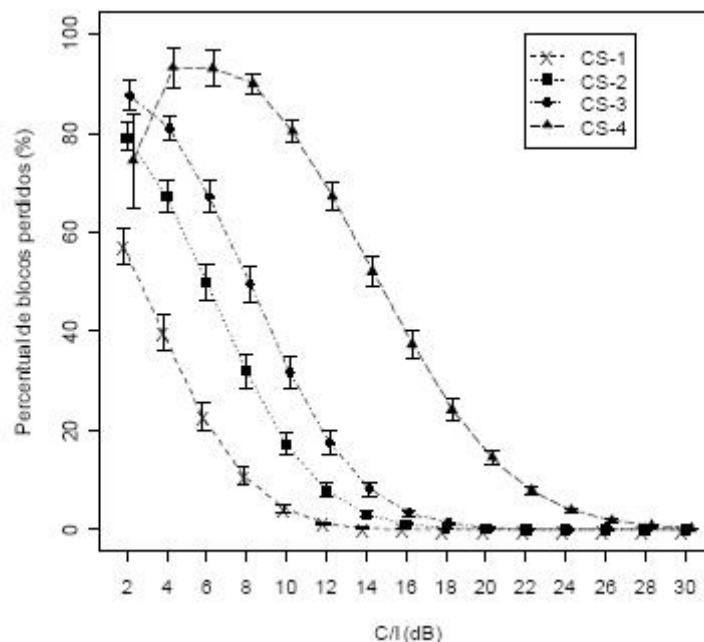
**Figura 22 – Vazão do Sistema nos Quatro Cenários de Codificação**

### 5.2.1.2 C/S Adequado para cada C/I

Observando o gráfico da Figura 22a, pode-se notar que há um CS adequado para cada C/I. Abaixo de 9 dB, o CS-1 obtém a maior vazão. Entre 9 e 14 dB, o CS-2 é capaz de transmitir maior quantidade de dados. O CS-3 apresenta melhor desempenho para C/Is de 15 a 24 dB, enquanto o CS-4 possui maior vazão a partir de 25 dB. Com esses resultados pode-se concluir que o uso dos CSs extremos (CS-1 e CS-4) só é justificado em condições ambientais

extremas: o CS-1 para ambientes com muito ruído (C/I abaixo de 9 dB) e o CS-4 para ambientes com pouco ruído (acima de 25 dB). Assim, os CSs mais apropriados são o CS-2 e o CS-3, que abrangem a faixa de 9 a 24 dB. Esta conclusão está de acordo com a experiência das operadoras, que geralmente utilizam o CS-2 [21].

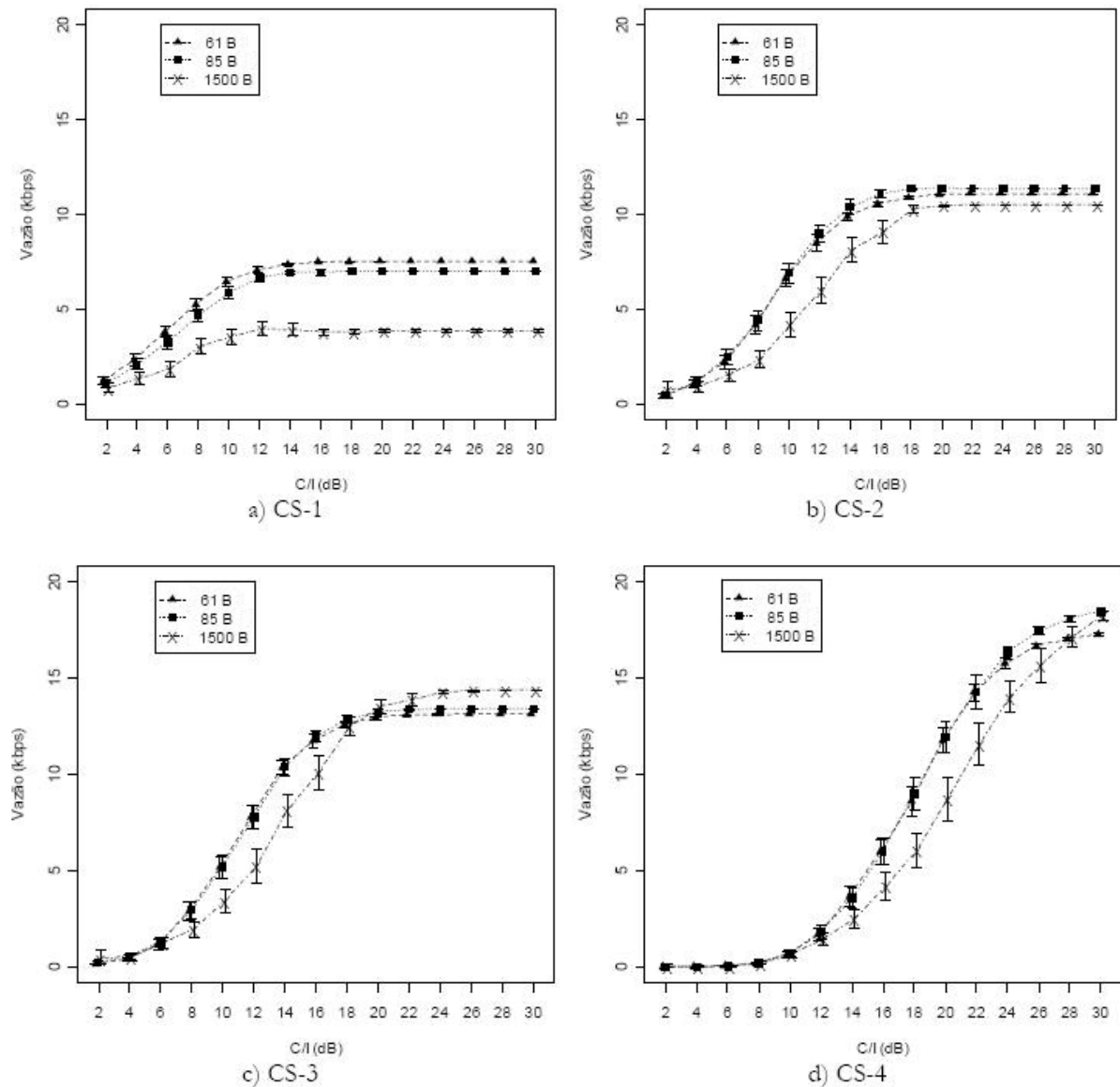
O percentual de blocos descartados, considerando os mesmos parâmetros, é exibido na Figura 23. Como esperado, para todos os cenários de codificação o percentual diminui à medida que as condições do ambiente melhoram (isto é, o valor de C/I aumenta). O gráfico mostra também o ponto em que cada cenário torna-se estável, ou seja, o ponto em que a taxa de erros não prejudica mais a taxa de transmissão, porque os blocos que apresentam erros podem ser recuperados rapidamente através da retransmissão ou da recuperação dos dados (utilizando a codificação oferecida pelo cenário). O CS-1 torna-se estável a partir de 16 dB, enquanto o CS-2 estabiliza em 20 dB e o CS-3, a partir de 24 dB. O CS-4 não alcança a estabilidade: aos 30 dB, seu percentual de blocos descartados ainda não atingiu o valor zero e, uma vez que esse cenário não oferece proteção aos dados, estes não podem ser recuperados. Esse resultado comprova que o CS-4 não é um cenário indicado por exigir condições ótimas de ambiente para que não haja perda de blocos.



**Figura 23 – Percentual de Blocos Descartados na Transferência de Informações para os Quatro CS**

### 5.2.1.3 Tipo de Pacotes de Rastreamento Veicular sobre GPRS

Com intuito de verificar se o tipo do pacote de informação de rastreamento veicular influencia na interação entre o TCP e o GPRS, esse tamanho foi variado, assumindo três valores diferentes: 61 por ser o valor do pacote de informações básicas de telemetria acrescido do cabeçalho TCP (41 + 20 bytes), 85 bytes por ser o tamanho do pacote de informações avançadas de telemetria acrescido do cabeçalho TCP (65 + 20 bytes) e 1500 bytes, por ser um tamanho comum, observado com frequência em transferências de arquivos grandes na Internet. Os resultados obtidos para cada cenário de codificação são ilustrados na Figura 24.



**Figura 24 – Vazão para Diferentes Tipos de Pacotes de Rastreamento Veicular**

Os valores apresentados nos quatro gráficos foram gerados utilizando 1 estação móvel, com configuração 1+1. Para o CS-1 (Figura 24a), o tamanho do pacote influencia a vazão: quanto menor o tamanho do pacote, maior a vazão. O esperado seria que a vazão fosse diretamente proporcional ao tamanho do pacote, uma vez que, quanto menor o pacote, menos dados podem ser transmitidos por segmento, sendo necessário o uso de mais pacotes e, conseqüentemente, mais cabeçalhos e maior sobrecarga (*overhead*). A diferença ocorre devido ao modo com confirmação (ACK) usado na camada RLC, que utiliza o mecanismo de janela deslizante com tamanho fixo de 64 blocos. A quantidade de dados úteis por bloco no CS-1 é reduzida (21 bytes, sem incluir cabeçalhos RLC e MAC). Assim, para transmitir um pacote de 1500 bytes (somando ainda cabeçalhos das camadas superiores), são necessários 74 blocos RLC. Mas a janela de transmissão da camada RLC só permite que sejam enviados 64 blocos sem que seja recebida uma confirmação (ACK). De acordo com a especificação [5], o receptor envia um ACK apenas quando recebe o último pacote da seqüência, ou quando o transmissor requisita uma confirmação. Como o último bloco dessa seqüência não será transmitido dentro da janela, o receptor só enviará um ACK quando for solicitado. O transmissor solicitará um ACK quando a janela ficar completa, retransmitindo os blocos cujo recebimento ainda não foi confirmado. O tempo gasto desde a retransmissão do primeiro bloco não confirmado (com solicitação de confirmação) até a recepção do ACK e a liberação da janela para novas transmissões causa a diminuição da taxa de transmissão observada no gráfico da Figura 24a. Com os demais CSs (Figuras 24b a 24d) isso não é tão perceptível, porque, para os tamanhos de pacotes utilizados nos experimentos, o número de blocos gerados por pacote pode ser enviado em uma única janela de transmissão. Dessa forma, a janela causa limitação apenas quando a taxa de erros é alta, provocando a perda de muitos blocos e, conseqüentemente, impedindo o “deslizamento” da janela.

Apesar de, segundo os resultados obtidos, o tamanho de 1500 bytes não ser o mais apropriado para o uso do TCP sobre o GPRS, este é o valor comum, utilizado na transferência de arquivos pela Internet. Para o resto das simulações optou-se por fixar o tamanho do pacote TCP em 85 bytes, visto ser o tamanho do pacote de transmissão de informações avançadas de telemetria (PT\_IAT) além de ser um valor intermediário nas simulações executadas.

#### 5.2.1.4 Tempo de Recuperação do Histórico de Rastreamento no Restabelecimento do Sinal GPS/GPRS

Observando a vazão do sistema para pacotes de rastreamento veicular (Figura 24) sobre as diversas codificações do cenário GPRS avaliado, podemos prever o tempo de restabelecimento do histórico de rastreamento no momento de restabelecimento do sinal GSM, após um período de armazenamento local das informações.

Para esta avaliação, tomou-se como parâmetro um veículo que tenha permanecido o tempo de 6 horas em um ambiente sem a presença de sinal GSM, tendo como configuração de rastreamento o envio de pacotes de informações avançadas de telemetria (85 bytes) com intervalo de 1 segundo no envio de cada pacote.

Neste cenário, após as 6 horas de armazenamento local, o veículo tem armazenado na memória de massa do equipamento um montante de pacotes ocupando um espaço de aproximadamente 1793 Kbytes.

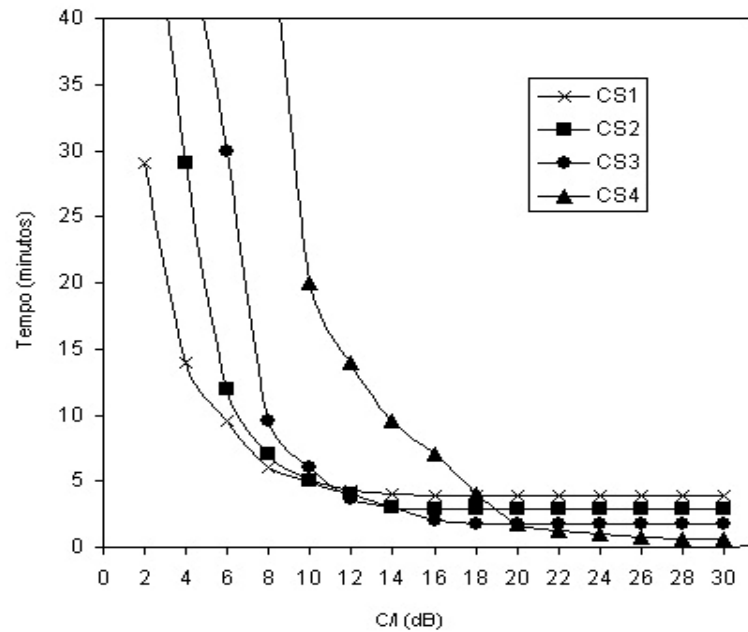
A Figura 25 apresenta o tempo necessário para que o veículo realize o envio dos pacotes armazenados localmente para o servidor central, levando em consideração as diferentes codificações do cenário.

Pode-se notar que o CS-1 consegue recuperar o histórico de rastreamento de modo mais eficiente para C/Is entre 2 e 7, perdendo eficácia na medida que as condições do ambiente (ruído) melhoram. O tempo de envio máximo para o CS-1 fica em torno de 5 minutos para C/I acima de 12 Db.

O CS-2 possui um desempenho muito inferior que o CS-2 para C/Is abaixo de 4. Porém consegue completar o envio das informações em menor tempo no intervalo de C/Is de 8 a 14, tendo a possibilidade de enviar o histórico completo em aproximadamente 4 minutos para condições de C/I acima de 14 Db.

O CS-3 tem desempenho superior aos outros para o intervalo de C/I entre 16 e 20 Db, tendo um péssimo desempenho para C/Is abaixo 6. Este consegue realizar o envio completo do histórico de informações em aproximadamente 2 minutos para C/Is acima de 20.

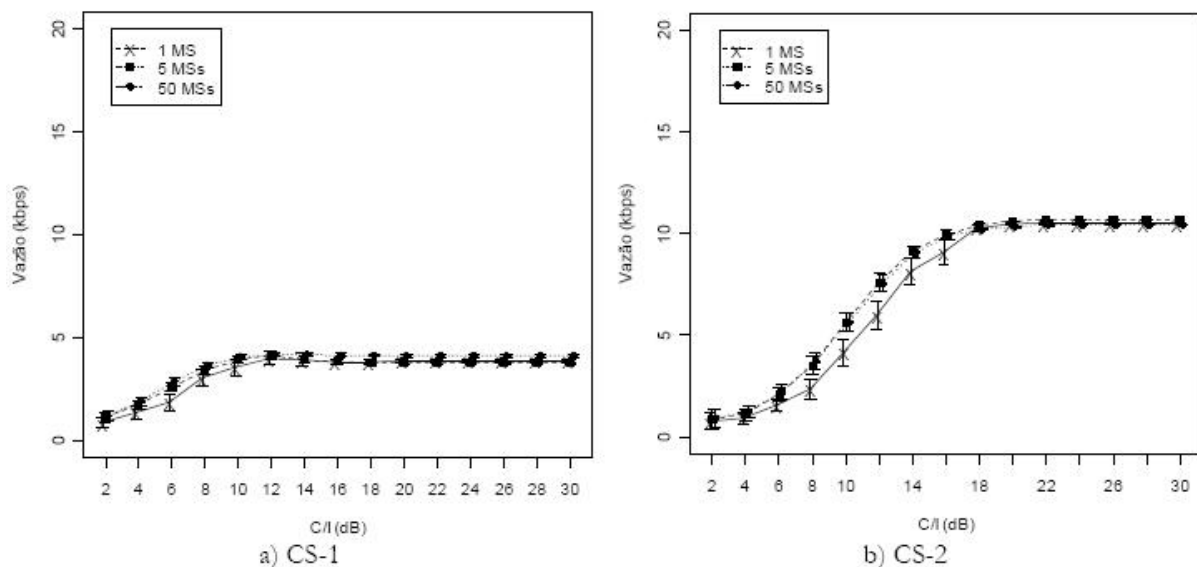
Por último, o CS-4 não consegue completar o envio do histórico de pacotes de rastreamento para C/Is abaixo de 10, tendo desempenho superior para o intervalos de C/I acima de 20, conseguindo realizar o envio completo do histórico em menos de 1 minuto, para o cenários com C/I acima de 24.



**Figura 25 - Tempo de Recuperação do Histórico de Rastreamento no Restabelecimento do Sinal GPRS**

#### 5.2.1.5 Número de Estações Móveis (Veículos)

A Figura 26 ilustra a vazão de dados da rede GPRS utilizando o protocolo TCP para os quatro CSs, variando o número de estações móveis (veículos) utilizados nas simulações. Os valores utilizados foram 1, 5 e 50 MS (veículos). Como pode ser visto nos gráficos, a quantidade de MSs praticamente não influenciou os resultados. Em outras palavras, o tráfego gerado por uma estação móvel não prejudicou as outras estações. Por este motivo, os demais resultados apresentados neste trabalho considerarão apenas uma estação móvel.



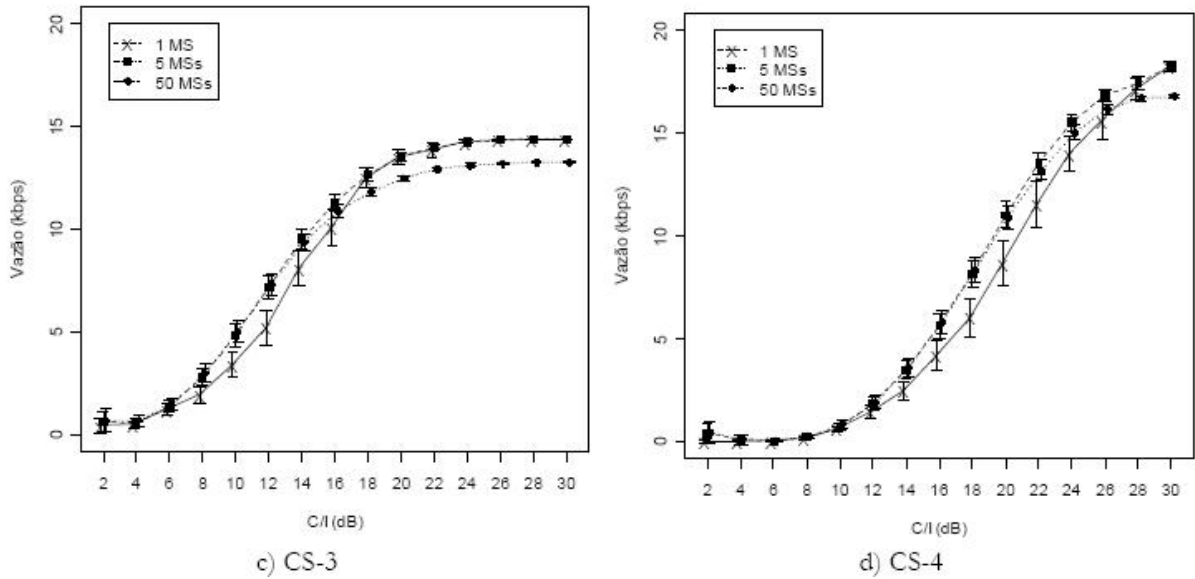


Figura 26 – Vazão para Diferentes Números de Estações Móveis

### 5.2.1.6 Multislot

A Figura 27 ilustra as vazões atingidas variando o número de *timeslots* utilizados por 1 usuário no sentido *uplink*. Além da configuração *singleslot* (1+1) utilizada para os demais resultados, foram utilizadas as configurações *multislot* 1+2 e 1+4. Comparando as Figuras 27a e 27b, observa-se que a vazão praticamente dobra, o que é absolutamente coerente visto que o número de slots utilizados é o dobro. Como o comportamento do sistema não muda de forma significativa ao aumentar o número de *timeslots*, os resultados seguintes levam em consideração apenas à configuração *singleslot* (1+1).

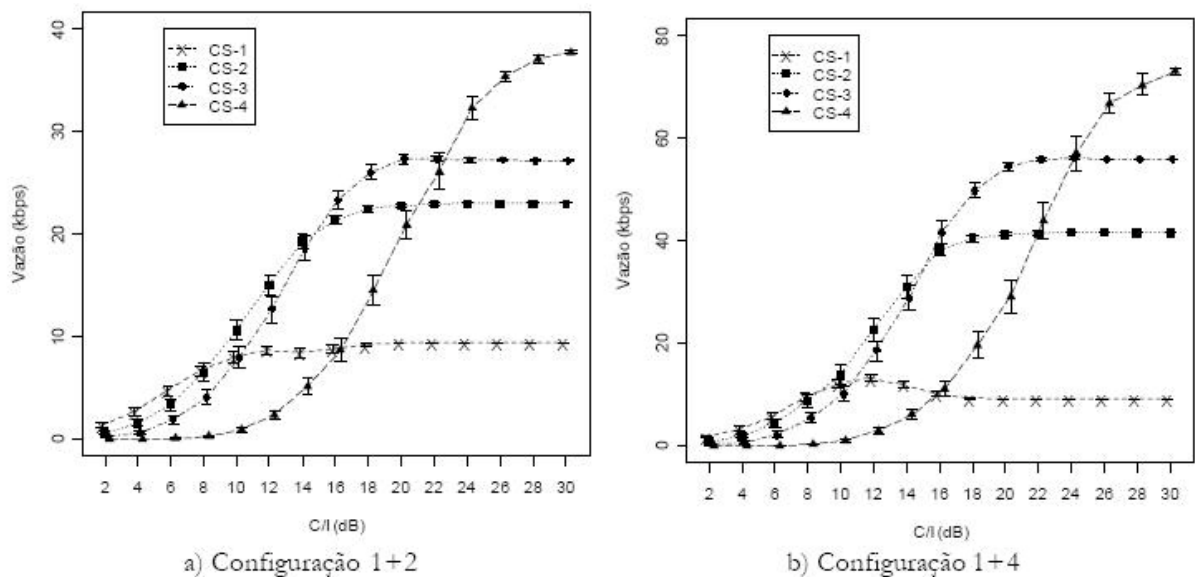


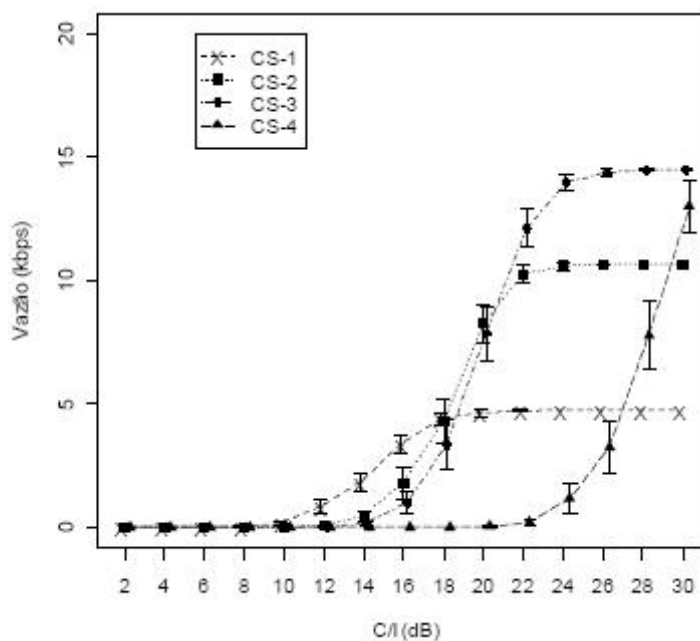
Figura 27 – Vazão para Múltiplos *Timeslots*



### 5.2.1.7 Modo de Transmissão RLC

A camada RLC do GPRS pode operar nos modos com ou sem confirmação (Seção 2.8.3). Os resultados apresentados até então, utilizaram apenas o modo com confirmação, portanto, decidiu-se verificar o desempenho do sistema sobre o GPRS utilizando o modo sem confirmação.

A Figura 28 ilustra a vazão obtida por uma MS submetida a diferentes condições de ambiente, utilizando a configuração 1+1 e pacotes de 85 bytes. Analisando o gráfico, pode-se perceber que em ambientes que apresentam muito ruído (C/I abaixo de 10 dB) a vazão é nula para todos os CSs. Esse resultado é bastante coerente uma vez que não ocorrem retransmissões de blocos perdidos e os erros têm que ser corrigidos utilizando a codificação oferecida pelo cenário. Como esperado, o CS-1 é o cenário que apresenta melhor desempenho por oferecer maior proteção aos dados. O CS-4, que não proporciona proteção aos dados, apresenta um desempenho muito ruim.

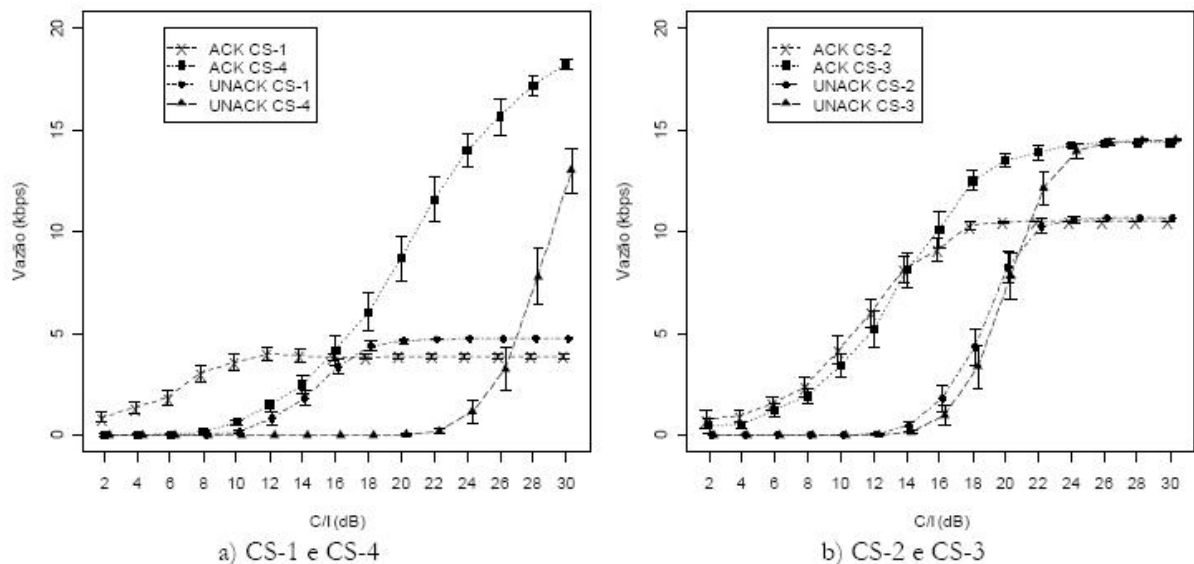


**Figura 28 – Vazão Utilizando Modo RLC sem Confirmação (UNACK)**

Os gráficos ilustrados na Figura 29 comparam o desempenho do sistema utilizando os dois modos da camada RLC. A Figura 29a mostra a vazão atingida pelos cenários extremos: CS-1 e CS-4. O CS-1, que já consegue transmitir alguns dados a partir de 10 dB, consegue obter uma vazão ainda maior que o modo com confirmação quando todos os erros podem ser recuperados (isto é, em condições muito boas de ambiente). Isso ocorre devido à limitação

imposta pela janela de transmissão utilizada no modo com confirmação (como explicado anteriormente). Enquanto o CS-4, por não oferecer proteção aos dados, apenas consegue transmitir dados em condições muito boas (depois de 22 dB), alcançando taxas bem abaixo do valor máximo teórico mesmo em ambientes com pouco ruído (30 dB).

Ilustrados na Figura 29b, o CS-2 e o CS-3 transmitem alguns dados em ambientes com boas condições (a partir de 14 dB) e atingem a mesma vazão obtida pelo modo com confirmação. Dessa forma, os resultados demonstram que o modo sem confirmação deve ser utilizado apenas em ambientes com condições muito boas e que o CS-4 deve ser evitado, visto que não apresenta proteção aos dados.



**Figura 29 – Vazão Variando o Modo de Transmissão RLC**

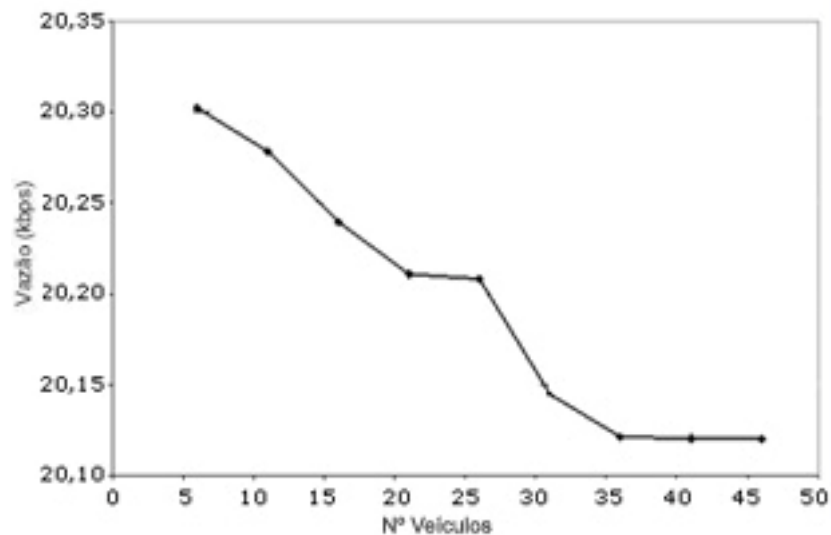
### 5.2.2 Transmissão de Informações de Rastreamento Veicular sobre GPRS/UDP

Para fins de comparação, os mesmos experimentos foram também realizados para tráfego UDP. Inicialmente, foram realizadas simulações variando-se o número de estações móveis (veículos) para a transferência de informações de rastreamento veicular utilizando o protocolo de transporte UDP sobre um canal de comunicação com C/I igual a 30Db (baixo ruído). Logo após, foram realizadas simulações variando o tamanho do pacote UDP e o nível de sinal / ruído. Por último, é realizada uma comparação da vazão de dados utilizando o protocolo de transporte TCP e UDP para os cenários propostos.

### 5.2.2.1 Numero de Estações Móveis (veículos)

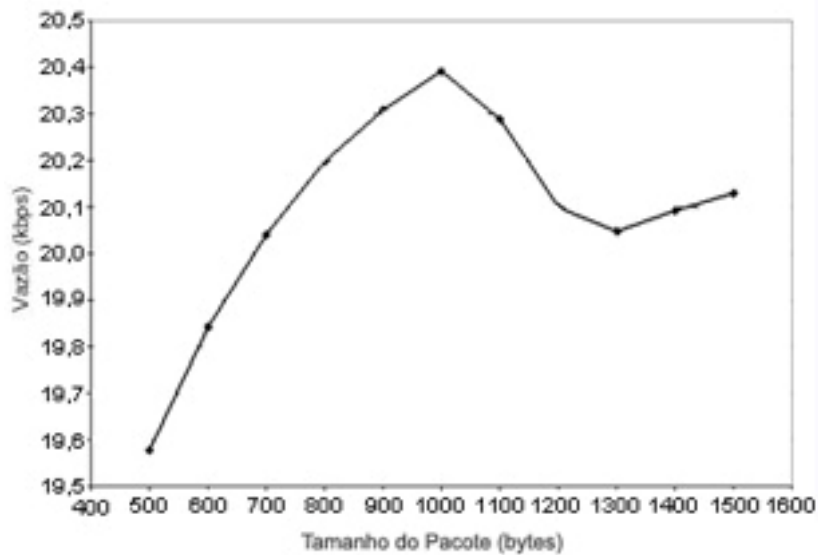
A Figura 30 mostra um gráfico obtido a partir da vazão alcançada em relação à variação do número de veículos trafegando pacotes UDP com tamanho de 800 bytes utilizando um canal de comunicação com C/I igual a 30Db (baixo ruído).

Como visto nas seções anteriores deste trabalho, os dados sobre a rede GPRS são enviados através de múltiplos *slots* de tempo (comutação de pacotes), que são alocados conforme a demanda dos pacotes enviados ou recebidos. Com o aumento da demanda por *slots* de tempo na rede GPRS, observou-se uma pequena diminuição na vazão de dados utilizando o protocolo UDP.



**Figura 30 – Vazão para Diferentes Números de Estações Móveis  
Variação do Tamanho do Pacote**

A Figura 31 mostra um gráfico obtido através dos resultados da vazão do sistema em relação ao tamanho do pacote de dados utilizando o protocolo de transporte UDP, trafegado sobre o sistema, os resultados tiveram como objetivo variar o comportamento do sistema frente a este tipo de variação.



**Figura 31 – Vazão com Diferentes Tamanhos de Pacotes de Dados sobre UDP**

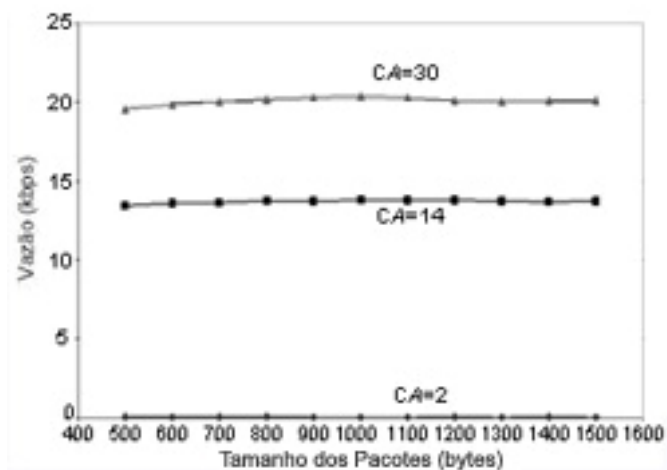
Como pode ser observado na Figura 31, houve um aumento gradativo na vazão até o pacote alcançar o tamanho de 1000 bytes, que é o tamanho do MTU (*Maximum Transmission Unit*) da rede, com isso os pacotes maiores que esse valor tiveram um decremento da vazão.

Isso se dá pelo fato de que os pacotes maiores de 1000 bytes são fragmentados na camada de enlace (por exemplo, pacotes de 1100 bytes são fragmentados em um pacote de 1000 bytes e outro de 100 bytes).

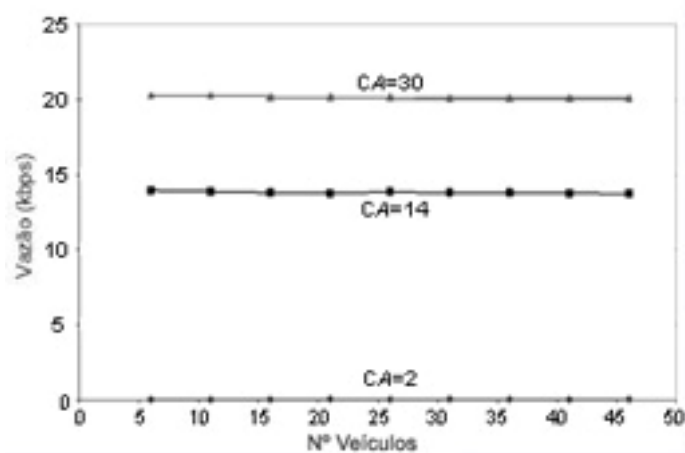
Na medida em que o tamanho dos pacotes com mais de 1000 bytes vai aumentando, a vazão volta novamente a crescer.

#### 5.2.2.2 Variação na Relação Sinal / Ruído

Para esta simulação, foram escolhidos três valores de C/I (2, 14, 30), que correspondem respectivamente a alto nível de ruído na rede, ruído médio e boas condições do sinal.



**Figura 32 – Variação da Taxa de Erros, Vazão e Tamanho dos Pacotes**



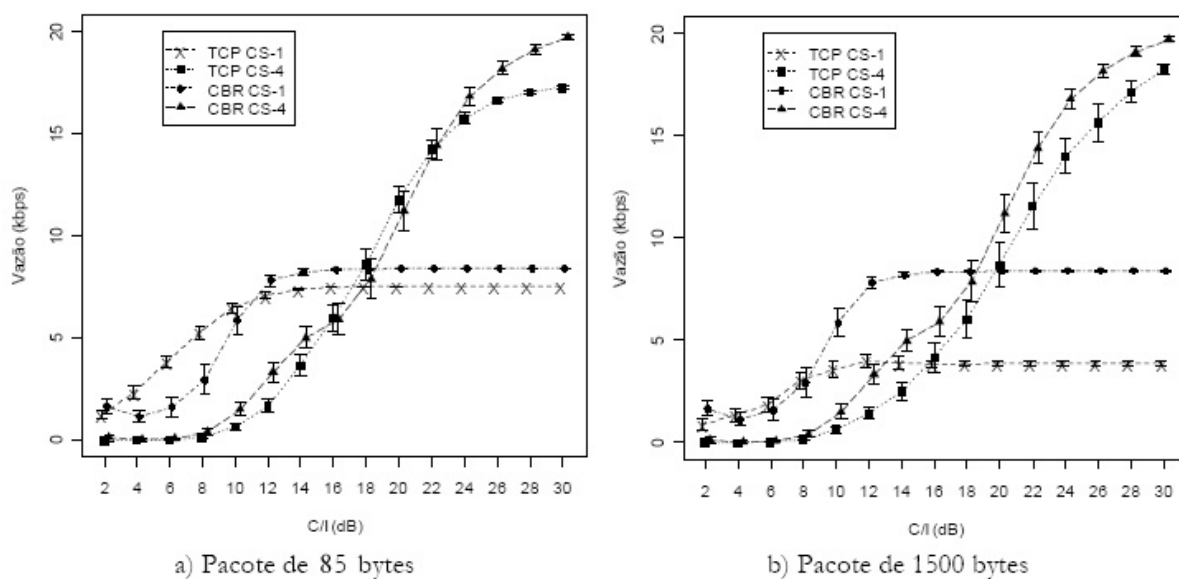
**Figura 33 – Variação da Taxa de Erros, Vazão e Número de Usuários**

Como pode ser observado nas figuras 32 e 33, com C/I igual a dois a vazão alcançada foi muito baixa em relação à taxa transmitida (cerca de 0,5 Kbps), o que compromete a recepção dos dados pelo receptor.

Com C/I igual a trinta obteve-se uma melhor vazão dos dados em relação à taxa transmitida e com C/I igual a quatorze, uma vazão intermediária.

#### 5.2.2.4 TCP x UDP para os Cenários de Codificação

Utilizando uma aplicação geradora de tráfego CBR com uma taxa de transmissão de 24 kbps, é possível observar a vazão máxima atingida pelo GPRS para cada CS sob os diferentes valores de C/I. A Figura 33 apresenta os dados obtidos para o CS-1 e o CS-4 utilizando como protocolo de transporte TCP e UDP. Na Figura 33a o tamanho de pacote TCP é 85 bytes e o tamanho do pacote UDP de 800 bytes. Pode-se perceber que, neste caso, o TCP apresenta um desempenho próximo do UDP. Contudo, na Figura 33b (na qual o tamanho do pacote TCP é de 1500 bytes), quando a perda de pacotes torna-se desprezível para o CS-1 (acima de 12 dB), a vazão alcançada pelo TCP (~ 4 kbps) é praticamente a metade da vazão do UDP (~ 9 kbps). Isso se deve ao modo com confirmação da RLC e à quantidade de blocos RLC gerados nesse cenário de codificação (como explicado nos resultados relacionados à variação do tamanho do pacote de informações de rastreamento veicular – Seção 6.2.1).



**Figura 34– Vazão TCP x UDP para CS-1 e CS-4**

Novamente os resultados mostram que o tamanho do pacote de informação de rastreamento veicular influencia o desempenho do sistema sobre o GPRS. Ressalta-se que, através do UDP, pôde-se validar o módulo GPRS utilizado nos experimentos, pois as taxas de transmissão alcançadas foram bastante próximas das taxas apresentadas na Tabela 2.1. Observando-se a Figura 33a, o CS-1 alcançou 8,38 kbps para um valor máximo teórico de 9,05 kbps, enquanto o CS-4 atingiu 19,7 kbps, para um máximo de 21,4 kbps. Em um ambiente ideal (isto é, sem o modelo de erros), o CS-4 obtém uma taxa mais próxima da esperada (20,24 kbps).

### 5.3 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados os aspectos relacionados às simulações. Inicialmente foram expostas as configurações da simulação. A topologia e os parâmetros utilizados foram descritos de forma detalhada. Em seguida os resultados mais significativos dos diversos experimentos de simulação realizados foram apresentados e discutidos.

Resultados relacionados à transmissão de informações de rastreamento veicular sobre redes GPRS utilizando os protocolos de transporte TCP e UDP foram expostos, através dos quais pode-se concluir que tráfegos de retaguarda (*background*) não influenciam de forma significativa no tráfego GPRS. Através das simulações foi possível obter faixas de valores de C/Is apropriados para o uso de cada cenário de codificação (CS), sendo observado que os cenários mais adequados são o CS-2 e o CS-3 por abrangerem uma faixa maior de C/Is no

qual apresentam melhor desempenho. Variando o tamanho do pacote trafegado, foi possível notar que no cenário CS-1, o tamanho do pacote influencia na vazão, sendo o tamanho do pacote inversamente proporcional à vazão obtida. O número de estações móveis (veículos) também foi variado, mostrando que o tráfego gerado por uma estação móvel não influencia as outras estações, ou seja, os resultados obtidos foram praticamente os mesmos independentes do número de MSs. Experimentos utilizando o modo sem confirmação da camada RLC demonstram que esse modo deve ser utilizado apenas em ambientes com condições muito boas de sinal e que o esquema de codificação CS-4 deve ser evitado neste modo, visto que não apresenta proteção aos dados.

Por último, foi realizada uma comparação entre o desempenho do sistema utilizando os protocolos de transporte TCP e UDP. Os resultados mostram, novamente, que o tamanho do pacote influencia no desempenho do sistema quando utilizando sobre o protocolo TCP no GPRS. Para o tamanho de 85 bytes, o TCP apresentou uma vazão próxima da alcançada pelo UDP, enquanto que com pacotes de 1500 bytes, a vazão atingida utilizando o CS-1 é menor que a metade da obtida pelo UDP.

O capítulo seguinte apresentará as conclusões e os trabalhos futuros relacionados a essa pesquisa.

## 6 CONCLUSÃO

Este trabalho teve como objetivo avaliar a transmissão de informações de rastreamento veicular sobre redes GPRS através de simulações. Vários cenários foram avaliados, incluindo diferentes taxas de erros (C/I) no meio físico de transmissão, tamanho de pacotes e quantidades de usuários e de *timeslots* utilizados por usuário no sentido *uplink*.

A primeira contribuição deste trabalho consiste no estudo abrangente realizado sobre a transferência de informações de rastreamento veicular sobre redes GPRS. Foram estudados os seguintes aspectos:

- Impacto do tráfego de retaguarda no tráfego de dados sobre GPRS: foram inseridos tráfegos de retaguarda (*background*) entre a rede GPRS e o transmissor de dados (no caso TCP), sendo constatado que não interferem no tráfego GPRS.
- O CS apropriado para cada taxa de erros (C/I): variando a taxa de erros do ambiente (C/I), foi avaliado o desempenho do sistema para cada cenário de codificação da camada RLC, e através desta análise, pôde-se determinar o intervalo em que cada CS produz maior taxa de transmissão. Verificou-se também que o CS-2 e o CS-3 são mais indicados por abrangerem uma faixa maior (9 a 25 dB);
- A influência do tamanho do pacote de informação de rastreamento veicular sobre GPRS: variando o tamanho do pacote (quando utilizando sobre o protocolo TCP), pode-se verificar que este influencia na vazão alcançada (principalmente quando CS-1 é utilizado), sendo inversamente proporcional ao tamanho do pacote. Isso ocorre devido à limitação imposta pelo tamanho da janela de transmissão da camada RLC;
- Quantidade de usuários no sistema GPRS: o número de estações móveis (veículos) no sistema GPRS foi variado a fim de detectar o impacto causado ao



incrementar o número de usuários no sistema. Os resultados obtidos foram praticamente os mesmos, independente do número de MS, mostrando que o tráfego gerado por uma estação móvel não prejudica as outras estações;

- A influência do número de *timeslots* por usuário no sentido *uplink*: foram analisadas duas configurações *multislot* (2+1,4+1). O comportamento do sistema permaneceu praticamente o mesmo para as duas configurações, diferenciando apenas na vazão alcançada que dobrou à medida que os números de *timeslots* duplicava;
- O comportamento do TCP sobre o modo sem confirmação da camada RLC: o modo sem confirmação da camada RLC foi utilizado, e pode-se concluir que este deve ser utilizado apenas em ambientes com condições muito boas e que, por não apresentar proteção aos dados, o CS-4 deve ser evitado nesse modo;
- O desempenho do TCP comparado ao UDP: foram realizados experimentos com o protocolo UDP a fim de comparar seus resultados com os obtidos utilizando o TCP. Observou-se que para segmentos de 85 bytes, os pacotes transmitidos utilizando o protocolo TCP apresentaram uma vazão próxima da alcançada pelo UDP, enquanto que com segmentos de 1500 bytes, a vazão atingida utilizando o CS-1 é menor que a metade obtida pelo UDP.

Outra importante contribuição foi o aprimoramento do módulo GPRS para o NS-2. Após as alterações realizadas, este módulo apresenta-se mais estável e está mais de acordo com a especificação do GPRS. Entre as principais modificações está a melhoria do modelo de erros, que atualmente utiliza uma tabela BLER bastante adotada no meio acadêmico.

Quanto a trabalhos futuros, uma interessante linha de pesquisa baseia-se na extensão do módulo GPRS para que este trabalhe com múltiplas células, incluindo gerenciamento de mobilidade, *handovers* e QoS. Outra possibilidade é a adaptação dos cenários do NS para simulação de redes veiculares (VANET), permitindo a troca de informações direta entre veículos, simulando assim, um nova gama de sistemas de auxílio em viagens e de gerenciamento de congestionamentos.

## REFERÊNCIAS

- [1] RENDÓN, J.; CASADEVALL, F.; FANER, J. L. **Wireline TCP Options Behavior in the GPRS Network**. In: Proc. 1<sup>ST</sup> Mobile Communications Summit, Sitges-Barcelona, Spain. Polytechnic University of Catalonia, Sep. 2001. pp.18.
- [2] BATES, R. J. **GPRS technology and its application**. China Mail Press, apr.2004. pag.2.
- [3] HOFF, S.; MEYER, M.; SACHS, J. **Analysis of the General Packet Radio Service (GPRS) of GSM as access to the Internet**. In: Proc. IEEE Universal Personal Communications, Oct.1998. Florence, Italy. p.415-419.
- [4] ETSI GSM 03.60. **Digital Cellular Telecommunications System (Phase 2+); General Packet Radio Service (GPRS); Service Description; Stage 2**. ver.6.6.0. Oct. 2000.
- [5] MOULY, M.; PAUTET, B. **The GSM System for Mobile Communications**. Telecom Publishing. Apr. 1992. pp.701.
- [6] BETTSTETTER, C.; VOGEL, H. J.; EBERSPACHER, J. **GSM Phase 2+; General Packet Radio Service GPRS: Architecture, Protocols and Air Interface**. IEEE Communications Surveys Journal, vol.2(3). Jun. 1999. pp. 63.
- [7] ABRAMSON, N. **The Aloha System – Another Alternative for Computer Communication**. Hawaii University of Honolulu. In: American Federation of Information Processing Societies Proc. Fall Joint Computer Conference. Apr. 1970. pp. 22.
- [8] RAPPAPORT, T. **Wireless Communications: Principles & Practice**. Prentice Hall PTR, New Jersey. Jun. 1996. p. 281-285.
- [9] AGILENT. **Understanding General Packet Radio Service (GPRS)**. Application Note. Jun. 2001. pp. 32.
- [10] ETSI GSM 05.02. **Digital Cellular Telecommunications System (Phase 2+); Multiplexing and multiple access on the radio path**. ver. 8.5.1. Nov. 2000.
- [11] ETSI GSM 04.60. **Digital Cellular Telecommunications System (Phase 2+); General Packet Radio Service (GPRS); Mobile Station (MS) - Base Station System (BSS) interface; Radio Link Control/Medium Access Control (RLC/MAC) protocol**. ver. 8.5.0, Oct. 2000.

- [12] HOFF, S.; MEYER, M.; SCHIEDER, A. **A Performance Evaluation of Internet Access via the General Packet Radio Service of GSM.** In: Proc. IEEE Vehicular Technology Society, Canada, May. 1998. p. 1760-1764.
- [13] MEYER, M. **TCP performance over GPRS.** In: Proceedings IEEE. Wireless Communications and Networking Conference, Sep. 1999. p. 1248 – 1252.
- [14] KALDEN, R.; MEIRICK, I.; MEYER, M. **Wireless Internet Access based on GPRS.** IEEE Personal Communications, vol. 7(2), Apr. 2000. p. 8 – 18.
- [15] DIKAIKOS, M. D.; NADEEM, T. N.; IQBAL, S. L. **VIPT: An Information Transfer Protocol for Vehicular Computing.** In: Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks. Cologne, Germany. May. 2005. p. 30 – 39.
- [16] QUIROGA, C.A.; BULLOCK, D. **A Data Model for Managing GPS Travel Time Data.** In: Applications of Advanced Technologies in Transportation: Proceedings of the 5th International Conference (eds: Chris T. Hendrickson and Stephen G. Ritchie), American Society of Civil Engineers. Newport Beach, California, 1998a. pp. 403-410.
- [17] TURNER, S. M. **Advanced Techniques for Travel Time Data Collection.** Transportation Research Record 1551, Transportation Research Board, Washington, D.C. 1996, pp. 51-58.
- [18] GALLAGHER, J. **Travel Time Data Collection Using GPS.** In: National Traffic Data Acquisition Conference (NATDAC '96) Proceedings, Volume I, 56 Report No. NM-NATDAC-96, Alliance for Transportation Research, Albuquerque, New Mexico. 1996. pp. 147-161.
- [19] MARSH, D. **Direct Conversion Receivers Battle Superherts for GSM Design Wins.** EDN Europe. Oct. 2000. pp. 523.
- [20] CHAKRAVORTY, R.; CARTWRIGHT, J.; PRATT, I. **Practical Experience with TCP over GPRS.** In: Proc. IEEE Global Communications Conference (IEEE GLOBECOM 2002), Taipei, Taiwan, Nov. 2002. p. 1678 – 1681.
- [21] TELE-MEDIOS Corp. **Redes Móveis de 2G+ e a Capacidade Espectral das Operadoras Móveis.** Télé-Médios Corp. e Sygnals Telecom. Mar. 2005. pp. 61.
- [22] DANTAS, S.; **Avaliação de Serviços de Dados em Redes GPRS.** Dissertação de Mestrado, Centro de Informática, UFPE, Recife. Dec. 2001. pp. 127.
- [23] FALL, K.; FLOYD, S. **Simulation-based Comparisons of Tahoe, Reno, and SACK TCP.** ACM Computer Communications Review. vol. 26, n.3. Berkeley, CA. Jul. 1996. p.5-21.
- [24] MURAKAMI, E; WAGNER, D. P. **Can Using Global Positioning System (GPS) Improve Trip Reporting?** Transportation Research Part C. vol. 7. Apr. 1999. pp. 149-165.

- [25] ADAMS, M.; MEYER, M.; SACHS, J. **Quality of Service Evaluation for Mobile Internet Access via the General Packet Radio Service (GPRS)**. In: Proc. 5th Intl. Workshop on Mobile Multimedia Communication MoMuc, Berlim. Oct. 1998. p. 173 – 179.
- [26] TANENBAUM, A. C. **Redes de Computadores**. 3a edição. Ed. Campus, Rio de Janeiro, 1997. pp. 903.
- [27] ALLMAN, M.; PAXSON, V.; STEVENS, W. **TCP Congestion Control**. Request For Comments – 2581. Network Working Group. Apr. 1999. pp. 14.
- [28] BRIAN, D. **Introduction to awk** [online]. <http://allman.rhon.itam.mx/dcomp/awk.html>. Acessado em 10/02/2007.
- [29] **Object Tool Command Language** [online]. <http://otcl-tclcl.sourceforge.net/otcl/>. Acessado em 11/06/2007.
- [30] **Simulation Augmented by Measurement and Analysis for Networks** [online]. <http://www.isi.edu/saman/index.html>. Acessado em 01/07/2007.
- [31] **Collaborative Simulation for Education and Research** [online]. <http://www.isi.edu/conser/index.html> Cited 20 April 2003. Acessado em 20/04/2007.
- [32] **The International Computer Science Institute Center for Internet Research** [online]. <http://www.icir.org/>. Acessado em 10/03/2007.
- [33] **The NetWork Simulator** [online]. <http://www.isi.edu/nsnam>. Acessado em 16/08/2006.
- [34] **Extensions to the ns Network Simulator** [online]. <http://www.icsi.berkeley.edu/~widmer/mnav/ns-extension/>. Acessado em 10/03/2007.
- [35] STEVENS, W. **TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms**. Request For Comments – 2001. Network Working Group. Jan. 1997. pp. 6.
- [36] JAIN, R. **GPRS Simulations Using ns-Network Simulator**. Dissertação de Mestrado, Department of Electrical Engineering, Indian Institute of Technology, Bombay, Jun. 2001.
- [37] POSTEL, J. **Transmission Control Protocol**. Request For Comments - 793, USC/Information Sciences Institute. Sep. 1981. pp. 85.
- [38] CHALRAVORTY R.; CLARK A.; PRATT I. **GPRSWeb: Optimizing the Web for GPRS Links**. In: Proc. ACM/USENIX First International Conference on Mobile Systems, Applications and Services (ACM/USENIX MOBISYS), San Francisco, USA, 2003. Acessado em: <http://www.cl.cam>.
- [39] Cadence.com [online]. <http://www.cadence.com/alta/products/bonesdat.html>. Acessado em 10/03/2007.

**APÊNDICE A – ALGORITMO DE COMPACTAÇÃO DE DATA E HORA  
NA LINGUAGEM DE PROGRAMAÇÃO “C”**

```
Unsigned long int codifica_data_hora(unsigned char p_ano,  
unsigned char p_mês, unsigned char p_dia, unsigned char  
p_hora, unsigned char p_min, unsigned char p_seg)  
{  
  
    unsigned long int soma, auxint;  
    unsigned char meses, x;  
  
    soma = p_ano;  
    auxint = soma / 4; // Pega o numero de anos bisextos  
  
    if ((soma % 4) != 0) //Verifica se o ano corrente é  
bisexto  
        auxint++; // Ano zero é bisexto  
  
    auxint = (auxint * 366) + ((soma - auxint) * 365); //  
Soma dos dias dos anos completos  
  
    meses = p_mês;
```

```

for (x = 1; x <= meses - 1; x++) //meses - 1 representa o
número de meses completos que já se
{
    // passaram no ano corrente
    if (x == 2) // Se for fevereiro
    {
        if ((soma % 4) == 0) // Verifica se o ano é
bisexto ou não
        {
            auxint = auxint + 29;
        }
        else
        {
            auxint = auxint + 28;
        }
    }
    else if (x < 8) //Se o mês for menor que agosto
    {
        if ((x % 2) == 1)
            auxint = auxint + 31; //mes impar tem 31
dias
        else
            auxint = auxint + 30; //mês par tem 30
dias
    }
    else // se o mês for maior que Julho
    {
        if ((x % 2) == 1)
            auxint = auxint + 30; //mês impar tem 30
dias
        else
            auxint = auxint + 31; //mês par tem 31
dias
    }
}

// Neste ponto a variavel auxint possui o numero de dias que
se passaram até o mês anterior da data
// corrente

soma = p_dia;
soma--;
soma = soma + auxint; // Acrescenta os dias completos do mês
corrente
soma = soma X 86400 // Soma recebe o numero de segundos de
todos os dias que se passaram

auxint = p_hora;
soma = soma + (auxint * 3600); //Soma acrescenta o numero de
segundos das horas completas do dia

```

```
//corrente  
  
    auxint = p_seg;  
    soma = soma + auxint; //Soma acrescenta o numero de  
segundos do minuto corrente  
  
    return (soma);  
}
```

**APÊNDICE B – ALGORITMO DE VERIFICAÇÃO E CONTROLE DE ERROS NA LINGUAGEM DE PROGRAMAÇÃO “C”**

```
// polinômio gerador = x^8 + x^5 + x^4 + 1

unsigned char far bfrso[TAMANHO_BUFFER]; //vetor que contem o
pacote de dados

unsigned char crc_frame (unsigned char tamanho_frame)
{
    unsigned char x, aux_valor = 0;

    tamanho_frame--; //Decrementa o tamanho do frame, pois o
ultimo byte do pacote é CRC e não entra
    // no cálculo

    for (x = 0; x < tamanho_frame; x++) //Laço percorre todos
os bytes do frame menos o ultimo byte
    {
        aux_valor = tabela_crc(aux_valor ^ bfrso[x]);
//aux_valor acumula o calculo do CRC
    }

    return (aux_valor); //Valor calculado é retornado
}
```



```
unsigned char tabela_crc (unsigned char newbyte)
{
    unsigned char x, aux_calculo;

    aux_calculo = 0;

    for (x = 8; x > 0; x--)
    {
        if ((aux_calculo & 1) != (newbyte & 1))
        {
            if (aux_calculo & 16)
                aux_calculo = aux_calculo & 239;
            else
                aux_calculo = aux_calculo | 16;

            if (aux_calculo & 8)
                aux_calculo = aux_calculo & 247;
            else
                aux_calculo = aux_calculo | 8;

            aux_calculo = aux_calculo >> 1;
            aux_calculo = aux_calculo + 128;
        }
        else
            aux_calculo = aux_calculo >> 1;

        newbyte = newbyte >> 1;
    }

    return (aux_calculo);
}
```