



Pós-Graduação em Ciência da Computação

**“ANÁLISE DE DESEMPENHO DE TCP SOBRE  
GPRS EM UM AMBIENTE FIM A FIM”**

**Por**

***Jeísa Pereira de Oliveira***

**Dissertação de Mestrado**



Universidade Federal de Pernambuco  
posgraduacao@cin.ufpe.br  
www.cin.ufpe.br/~posgraduacao

RECIFE, FEVEREIRO/2004



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE INFORMÁTICA  
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

JEÍSA PEREIRA DE OLIVEIRA

“ANÁLISE DE DESEMPENHO DE TCP SOBRE  
GPRS EM UM AMBIENTE FIM A FIM”

*ESTE TRABALHO FOI APRESENTADO À PÓS-GRADUAÇÃO EM  
CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA  
UNIVERSIDADE FEDERAL DE PERNAMBUCO COMO REQUISITO  
PARCIAL PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA DA  
COMPUTAÇÃO.*

ORIENTADOR(A): JUDITH KELNER  
CO-ORIENTADOR(A): CARLOS ALBERTO KAMIENSKI

RECIFE, FEVEREIRO/2004

Aos meus pais Juarez e Margareth  
e à minha irmã Jeanne.

# Agradecimentos

Agradeço primeiramente a Deus por tudo o que Ele fez, tem feito e ainda fará em minha vida.

A meus pais, Juarez e Margareth, por seu apoio e amor, fundamentais para a minha vinda e estabelecimento em Recife.

À minha irmã, Jeanne, por seu amor, amizade, companheirismo e encorajamento.

A todos os que compõem o Grupo de Pesquisa em Redes e Telecomunicações (GPRT) pela amizade, pelos momentos de descontração e por me ajudarem direta e indiretamente a superar os momentos difíceis. Em especial a Joseane Fidalgo, Suzana Dantas, Arthur Callado, Thiago Souto e Igor Cananéa por terem ajudado de forma mais direta.

Aos professores Judith Kelner e Djamel Sadok pela orientação, pela amizade e pelo apoio.

A Carlos Kamienski por ter se engajado no trabalho, compartilhando sua amizade, seu tempo e seus conhecimentos, sem os quais este trabalho não teria sido concluído.

À família SEINF, especialmente a David Gadelha, por ter me incentivado a ingressar no mestrado.

A todos os amigos, antigos e recentes, cuja amizade foi essencial para a continuidade deste trabalho. Em especial à Cláudia Brito por ter compartilhado minhas angústias, alegrias, problemas e soluções desde a graduação até o mestrado. E a Marco Domingues, João Rocha, Rossam Silva e Eduardo Souto pelo apoio incondicional.

# Sumário

<b>Abreviações e Acrônimos</b>	<b>ix</b>
<b>Resumo</b>	<b>xii</b>
<b>Abstract</b>	<b>xiii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação.....	1
1.2 Objetivo .....	2
1.3 Gerações dos Sistemas Celulares.....	2
1.4 Estrutura da Dissertação.....	4
<b>2 GPRS</b>	<b>5</b>
2.1 GSM.....	5
2.2 Rede GPRS.....	7
2.3 Gerenciamento de Mobilidade, de Sessão e de Roteamento .....	8
2.3.1 Procedimentos GPRS <i>attach</i> e <i>detach</i> .....	9
2.3.2 Contexto PDP .....	9
2.3.3 Estados de Operação de uma MS.....	9
2.4 Classes de Mobilidade GPRS/GSM .....	10
2.5 Protocolos do GPRS.....	11
2.5.1 Plano de Transmissão .....	11
2.5.2 Plano de Sinalização .....	13
2.6 Transmitindo os PDUs ( <i>Packet Data Units</i> ).....	14
2.6.1 Transmissão no GGSN .....	15
2.6.2 Transmissão no SGSN.....	16
2.6.3 Transmissão na BSS .....	16
2.6.4 Transmissão através da Interface Aérea.....	17
2.7 Interface Aérea .....	17
2.7.1 Canais Físicos .....	18
2.7.2 Canais Lógicos.....	19
2.7.3 Mapeamento de Canais Lógicos sobre Canais Físicos.....	20
2.7.4 Cenários de Codificação .....	22
2.8 Operações de Transferência de Pacotes de Dados.....	23
2.8.1 Transferência no Sentido <i>Uplink</i> .....	23
2.8.2 Alocação de Blocos de Rádio .....	23
2.8.3 Transferência de Blocos de Dados RLC.....	24
2.8.4 Transferência no Sentido <i>Downlink</i> .....	25
2.9 Vantagens e Desvantagens do GPRS .....	26
2.10 Considerações Finais.....	27
<b>3 TCP sobre GPRS</b>	<b>29</b>
3.1 TCP.....	29
3.1.1 TCP Reno .....	30

3.1.2	TCP New-Reno.....	31
3.1.3	TCP SACK.....	31
3.1.4	TCP Westwood.....	32
3.2	TCP sobre Redes sem Fio.....	33
3.3	TCP sobre GPRS.....	35
3.3.1	Trabalhos Relacionados.....	35
3.3.2	Problemas do TCP sobre GPRS.....	37
3.4	Considerações Finais.....	38
<b>4</b>	<b>GPRS no ns-2</b> .....	<b>39</b>
4.1	<i>Network Simulator</i> .....	39
4.2	Módulo GPRS.....	40
4.2.1	Estrutura do Módulo.....	40
4.2.2	Dificuldades Iniciais.....	45
4.2.3	Limitações e Problemas do Módulo.....	46
4.3	Modificações Realizadas.....	47
4.3.1	Correção da Vazão.....	48
4.3.2	Robustez no Mecanismo de Alocação de Canais.....	48
4.3.3	Eliminação de Vazamento de Memória.....	50
4.3.4	Melhoramento no Modelo de Erros.....	51
4.3.5	Separação entre MS e BS.....	52
4.3.6	Divisão da Fila de Pacotes da BS.....	53
4.3.7	Transmissão dos Blocos RLC de Acordo com a Especificação.....	53
4.3.8	Implementação do Modo com Confirmação da Camada RLC de Acordo com a Especificação.....	53
4.3.9	Limitação dos <i>Buffers</i> .....	55
4.3.10	Redução do Uso de <i>Timers</i> .....	55
4.4	Adições Realizadas.....	56
4.4.1	Implementação dos Cenários de Codificação.....	56
4.4.2	Implementação do Modelo Adaptativo.....	56
4.4.3	Implementação de <i>Multislots</i> .....	57
4.5	Considerações Finais.....	58
<b>5</b>	<b>Avaliação de Desempenho</b> .....	<b>59</b>
5.1	Configurações de Simulação.....	59
5.2	Resultados.....	61
5.2.1	TCP Reno sobre GPRS.....	61
5.2.2	TCP x UDP.....	69
5.2.3	Modelo Adaptativo.....	70
5.2.4	Versões do TCP.....	75
5.3	Considerações Finais.....	78
<b>6</b>	<b>Conclusão</b> .....	<b>80</b>
6.1	Contribuições.....	80
6.2	Trabalhos Futuros.....	82
6.3	Considerações Finais.....	84
<b>7</b>	<b>Referências</b> .....	<b>85</b>

# Lista de Figuras

<b>Figura 2.1</b> Rede GSM.....	6
<b>Figura 2.2</b> Rede GPRS.....	8
<b>Figura 2.3</b> Estados da Estação Móvel.....	10
<b>Figura 2.4</b> Plano de Transmissão .....	11
<b>Figura 2.5</b> Plano de Sinalização: MS-SGSN .....	13
<b>Figura 2.6</b> Plano de Sinalização: SGSN-HLR, SGSN-EIR e SGSN-MSC/VLR.....	14
<b>Figura 2.7</b> Transferindo Pacotes de Dados Fim a Fim em uma Rede GPRS [24].....	15
<b>Figura 2.8</b> Portadoras de Frequência GSM, Duplexação e Quadros TDMA.....	18
<b>Figura 2.9</b> Divisão da Interface Aérea GPRS em Canais Físicos .....	19
<b>Figura 2.10</b> Mapeamento de Canais Lógicos em Canais Físicos.....	21
<b>Figura 2.11</b> Codificação dos Pacotes de Dados GPRS.....	22
<b>Figura 4.1</b> Pilha de Protocolos de Rede em um Nó Móvel no ns [47] .....	41
<b>Figura 4.2</b> Pilha Modificada de Protocolos de Rede em um Nó Móvel GPRS [47].....	42
<b>Figura 4.3</b> Valores dos Parâmetros de Transição entre os Cenários de Codificação [51].....	57
<b>Figura 5.1</b> Topologia Utilizada nas Simulações .....	59
<b>Figura 5.2</b> Vazão do TCP Reno para os quatro Cenários de Codificação .....	62
<b>Figura 5.3</b> Percentual de Blocos Descartados para o TCP Reno para os quatro CSs .....	63
<b>Figura 5.4</b> Vazão do TCP Reno para Diferentes Tamanhos de Pacote .....	64
<b>Figura 5.5</b> Vazão do TCP Reno para Números Diferentes de Estações Móveis .....	66
<b>Figura 5.6</b> Vazão do TCP Reno com Múltiplos <i>Timeslots</i> .....	67
<b>Figura 5.7</b> Vazão do TCP Reno Utilizando o Modo RLC sem Confirmação (UNACK) .....	68
<b>Figura 5.8</b> Vazão do TCP Reno Variando o Modo de Transmissão RLC.....	69
<b>Figura 5.9</b> Vazão do TCP Reno e do UDP (CBR) para CS-1 e CS-4.....	70
<b>Figura 5.10</b> Vazão do TCP Reno Utilizando o Modelo Adaptativo para CS .....	71
<b>Figura 5.11</b> Variação do Cenário de Codificação Utilizando o Modelo Adaptativo.....	72
<b>Figura 5.12</b> Pacotes Transmitidos Utilizando o Modelo Adaptativo.....	73
<b>Figura 5.13</b> Novos Valores para os Parâmetros de Transição entre os CSs .....	74
<b>Figura 5.14</b> Resultados Utilizando os Novos Valores para os Parâmetros de Transição.....	75
<b>Figura 5.15</b> Vazão do TCP Reno, New-Reno, SACK e Westwood para Todos os CSs.....	77

# Lista de Tabelas

<b>Tabela 2.1</b> Cenários de Codificação .....	23
<b>Tabela 4.1</b> BLER [51] .....	51
<b>Tabela 5.1</b> Parâmetros de Simulação.....	60



# Abreviações e Acrônimos

1G	Primeira geração de sistemas celulares
2G	Segunda geração de sistemas celulares
2,5G	Geração intermediária de sistemas celulares, entre 2G e 3G
3G	Terceira geração de sistemas celulares
4G	Quarta geração de sistemas celulares
8PSK	eight-phase-shift keying
ABSE	Adaptive Bandwidth Share Estimation
ACK	Acknowledgement
AMPS	Advanced Mobile Phone Services
ARQ	Automatic Repeat reQuest
AuC	Authentication Center
BCH	Broadcast Channel
BCS	Block Check Sequence
BE	Bandwidth Estimation
BLER	Block Error Rate
BS	Base Station
BSC	Base Station Controller
BSN	Block Sequence Number
BSS	Base Station Subsystem
BSSAP	Base Station System Application Part
BSSGP	Base Station System GPRS Protocol
BTS	Base Transceiver Station
C/I	carrier-to-interface
CS	Coding Scheme
CWND	Congestion Window
EDGE	Enhanced Data Rate for GSM Evolution
EIR	Equipment Identity Register
ELN	Explicit Loss Notification
ETSI	European Telecommunications Standards Institute
FCS	Frame Check Sequence
FDMA	Frequency Division Multiple Access
GGSN	Gateway GPRS Support Nodes
GMM/SM	GPRS Mobility Management and Session Management
GMSK	Gaussian minimum-shift keying
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
GSN	GPRS Support Node
GTP	GPRS Tunneling Protocol

HLR	Home Location Register
IFQ	Interface Queue
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IMT-2000	International Mobile Telecommunications - year 2000
IP	Internet Protocol
I-TCP	Indirect TCP
ITU	International Telecommunications Union
LA	Location Area
MAC	Medium Access Control
MAP	Mobile Application Part
MS	Mobile Station
MSC	Mobile Switching Center
MTP2	Message Transfer Part Level 2
MTP3	Message Transfer Part Level 3
MTU	Maximum Transmission Unit
NetIF	Network Interface
NOAH	Non Adhoc Routing Agent
N-PDU	Network Packet Data Unit
NS	Network Subsystem
ns	Network Simulator
PACCH	Packet Associated Control Channel
PAGCH	Packet Access Grant Channel
PBCCH	Packet Broadcast Control Channel
PBCH	Packet Broadcast Channel
PCCCH	Packet Common Control Channel
PDCH	Packet Data Channel
PDN	Packet Data Network
PDP	Packet Data Protocol
PDTCH	Packet Data Traffic Channel
PDU	Packet Data Unit
PHL	Physical Layer
PLL	Physical Link Layer
PNCH	Packet Notification Channel
PPCH	Packet Paging Channel
PRACH	Packet Random Access Channel
PTCCH	Packet Timing advance Control Channel

PTCH	Packet Traffic Channel
PTM	Point-To-Multipoint
P-TMSI	Packet Temporary Mobile Subscriber Identity
QoS	Quality of Service
RA	Routing Area
RE	Rate Estimation
RED	Random Early Detection
RFC	Request For Comments
RFL	Physical RF Layer
RFP	Radio Fixed Part
RLC	Radio Link Control
RTO	Retransmission Timeout
RTT	round-trip time
SACK	Selective ACKnowledgement
SAPI	Service Access Point Interface
SCCP	Signaling Connection Control Part
SGSN	Serving GPRS Support Node
SIM	Subscriber Identity Module
SNDCP	SubNetwork Dependent Convergence Protocol
SS#7	Signaling System no. 7
SSTHRESH	Slow Start Threshold
TBF	Temporary Block Flow
TCAP	Transactions Capabilities Application Part
TCP	Transmission Control Protocol
TCPW	TCP Westwood
TCPW BR	TCP Westwood with Bulk Repeat
TCPW CBR	TCP Westwood with Combined Bandwidth and Rate estimation
TDMA	Time Division Multiple Access
TFI	Temporary Flow Identity
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
USF	Uplink State Flag
VLR	Visitor Location Register

# Resumo

O GPRS (*General Packet Radio Service*) é uma extensão do GSM (*Global System for Mobile communications*) para a transmissão de dados baseada em pacotes e surgiu com o objetivo de atender à demanda por acesso sem fio à Internet. Por aproveitar toda a estrutura da rede GSM, tecnologia mais utilizada na área de comunicação celular, o GPRS vem sendo bastante adotado pelas operadoras celulares, inclusive no Brasil.

Redes sem fio apresentam grandes taxas de perdas de pacotes devido a erros na sua interface aérea. Esta característica causa problemas de desempenho para o protocolo TCP (*Transmission Control Protocol*), que é utilizado pela maioria dos serviços de aplicação oferecidos atualmente na Internet. A perda de pacotes consiste em uma notificação implícita de congestionamento na rede para o transmissor TCP, que diminui sua taxa de transmissão. Em uma rede sem fio, como GPRS, o TCP não possui informação suficiente para distinguir entre perdas de pacote devido a erros na interface aérea e perdas causadas por descartes nas filas dos roteadores. Assim, é necessário que o seu comportamento sobre o GPRS seja cuidadosamente analisado.

Alguns estudos têm sido realizados nos últimos anos sobre esse assunto, entretanto todos possuem um escopo bastante limitado. Este trabalho apresenta uma análise mais abrangente do desempenho do TCP sobre GPRS, baseada em simulação. Vários cenários distintos são avaliados, incluindo diferentes taxas de erros no meio físico de transmissão, tamanhos de pacotes e quantidades de usuários e de *timeslots* utilizados por usuário no sentido *downlink*. Além disso, um aspecto inovador é analisado: o uso de um modelo adaptativo para configurar dinamicamente os cenários de codificação (CS) do GPRS.

Dentre as contribuições deste trabalho, destacam-se ainda os vários ajustes e acréscimos ao código do simulador utilizado visando tornar os resultados mais confiáveis.

**Palavras-chave:** GPRS, TCP, análise de desempenho

# Abstract

*The General Packet Radio Service (GPRS) represents a packet-switched network extension for GSM (Global System for Mobile communications) and aims to provide wireless access to the Internet. For sharing the network structure with GSM, the world's leading technology in cellular communications system, it has been adopted by many cellular operators, including Brazilian ones.*

*Wireless networks suffer from high packet loss levels due to air interface errors. This results in performance problems for TCP (Transmission Control Protocol), which is used by most application services currently available in the Internet. Packet losses represent an implicit notification of network congestion for a TCP transmitter, which decreases its transmission rate. When using TCP over GPRS, TCP has not enough information to distinguish between packet losses occurred in the wireless link and due to discards in routers queues. Therefore, it is necessary to study carefully TCP behavior over GPRS.*

*Although there have recently been a number of publications on this topic, they present a limited scope in their analysis. This work presents, via simulation results, wider performance issues related to TCP over GPRS. Many scenarios are studied, including different error rates, packet sizes, number of users and downlink timeslots per user. In addition, an innovating aspect is analysed: the use of an adaptive model that configures dynamically GPRS coding schemes.*

*Amongst the contributions of this work, it is worth emphasizing the many adjustments and additions made to the used simulator code so it could generate more reliable and correct results.*

**Keywords:** GPRS, TCP, performance analysis

# 1

# Introdução

Nos últimos anos, observou-se um contínuo crescimento tanto na área de comunicação celular quanto na Internet, o que gerou a necessidade de acesso sem fio à Internet. O GPRS (*General Packet Radio Service*) [1] surgiu para atender a essa demanda, estendendo a tecnologia mais utilizada na área: GSM (*Global System for Mobile communications*) [2]. Enquanto o GSM utiliza comutação por circuitos, o GPRS utiliza comutação por pacotes para transmitir dados, permitindo comunicação entre as estações móveis e redes externas de pacotes de dados, incluindo a Internet. O GPRS vem sendo bastante adotado pelas operadoras por apresentar um baixo custo de implantação uma vez que aproveita toda a estrutura da rede GSM. No Brasil, até a presente data, duas operadoras já utilizam o GPRS [3]: TIM[4] e Oi (TLN PCS)[5].

## 1.1 Motivação

Ao mesmo tempo em que permitem uma grande flexibilidade de comunicação e acesso à Internet ao usuário (pela mobilidade proporcionada em grandes áreas geográficas), redes sem fio como a GPRS apresentam grandes taxas de erros na sua interface aérea, que variam significativamente de acordo com as condições do ambiente. Este aspecto causa problemas de desempenho no protocolo TCP (*Transmission Control Protocol*) [6], o qual é utilizado pela maioria dos serviços de aplicação oferecidos atualmente na Internet. Em situações de congestionamento aparente, o TCP reduz a sua janela de congestionamento, causando a diminuição da taxa de transmissão. Como não possui nenhum mecanismo de notificação explícita da ocorrência de congestionamento em algum ponto da rede, o transmissor TCP o infere (implicitamente) através da perda de pacotes. Esse mecanismo foi desenvolvido para redes cabeadas, nas quais raramente ocorrem perdas de pacotes devido a problemas no meio físico. Em uma rede sem fio, como a GPRS, na ocorrência de perda de pacotes na interface aérea, o TCP não possui informação suficiente para distinguir entre perdas de pacote na rede sem fio e perdas devido a descartes nas filas dos roteadores. Por este motivo, é importante

que o seu comportamento sobre o GPRS seja cuidadosamente analisado.

Alguns estudos sobre esse assunto têm sido realizados nos últimos anos [7]-[13]. No entanto, cada um deles em geral avalia apenas poucos aspectos possíveis em um cenário que suporta TCP com GPRS. Faz-se necessário, portanto, um estudo mais abrangente sobre essa interação.

## 1.2 Objetivo

Este trabalho visa analisar o desempenho do TCP sobre o GPRS através de simulações. Ambos os protocolos, TCP e GPRS, apresentam um comportamento complexo e algoritmos que, ao interagirem, interferem entre si. Utilizar em conjunto os dois protocolos pode gerar resultados inesperados, tornando-se muitas vezes difícil identificar o motivo desses resultados. Assim, é necessário explorar diferentes cenários a fim de detectar e estudar casos desse tipo.

Vários cenários distintos são avaliados neste trabalho, incluindo diferentes taxas de erros no meio físico de transmissão, tamanhos de pacotes e quantidades de usuários e de *timeslots* utilizados por usuário no sentido *downlink*. Além disso, um aspecto é avaliado pela primeira vez: o uso de um modelo adaptativo para configurar cenários de codificação (CSs) da camada RLC (*Radio Link Control*) do GPRS. Tal modelo muda dinamicamente o CS utilizado baseado na taxa de blocos RLC perdidos por janela de transmissão, permitindo uma melhor utilização dos recursos e, conseqüentemente, uma maior taxa de transmissão de dados.

A fim de estabelecer o contexto no qual o GPRS se encontra, as gerações dos sistemas celulares serão descritas de forma sucinta a seguir.

## 1.3 Gerações dos Sistemas Celulares

Os primeiros sistemas celulares comerciais surgiram no início da década de 80 [14]. Eram baseados na tecnologia analógica e ofereciam apenas transmissão de voz com baixa qualidade de serviço e baixa capacidade no canal de comunicação. Esses sistemas formam a primeira geração de sistemas celulares (1G) e um típico exemplo dessa geração é o AMPS (*Advanced Mobile Phone Services*) [15]. A segunda geração (2G) trouxe melhoras na qualidade de transmissão e capacidade do sistema. Utilizando tecnologia digital, essa geração permite transmissão

de dados via canal de sinalização. O principal representante dessa geração é o GSM [2], que em 1999 já havia se tornado a tecnologia sem fio móvel de maior sucesso no mundo [14] e continua sendo a tecnologia dominante, utilizada em 190 países por (estimado) 1 bilhão de usuários [3].

As duas primeiras gerações apresentam um grande número de padrões e tecnologias incompatíveis, o que motivou a ITU (*International Telecommunications Union*) a integrar as tecnologias da próxima geração (3G) em uma única família de padrões compatíveis. Estes poderão ser usados em todo o mundo, para todas as aplicações móveis, com suporte à transmissão de dados por comutação de pacotes ou de circuitos, e com taxas de transmissão de até 2Mbps. Essa integração é denominada IMT-2000 (*International Mobile Telecommunications - year 2000*) [16].

Entre a segunda e a terceira geração, há uma geração intermediária denominada 2,5G, que visa facilitar a migração entre essas duas gerações. O GPRS faz parte dessa geração intermediária que tem como principal característica oferecer serviços de melhor qualidade e maiores taxas de transmissão que a segunda geração utilizando a infra-estrutura desta. Uma evolução importante introduzida pela 2,5G consiste na comutação por pacotes. Os sistemas 2G utilizam apenas comutação por circuito, enquanto os sistemas 2,5G utilizam os dois tipos, uma vez que consistem em redes *overlay* sobre as redes da segunda geração. Embora algumas operadoras já ofereçam serviços 3G (como a NTT DoCoMo no Japão [17]), os sistemas 2G ainda são os mais utilizados (em especial o GSM) e os sistemas 2,5G encontram-se em expansão mundial.

No processo de evolução dos sistemas celulares, o GPRS evoluirá para o EDGE (*Enhanced Data Rate for GSM Evolution*) [18] – outra tecnologia 2,5G com taxa de transmissão mais altas, próximas das alcançadas pela terceira geração – e, em seguida, para o UMTS (*Universal Mobile Telecommunications System*) [19] – padrão 3G europeu.

Há ainda a quarta geração (4G), que pretende aumentar consideravelmente a taxa de transmissão da 3G. Os sistemas 4G prometem taxas acima de 20 Mbps [20]. Apesar de já existirem estudos e *testbeds* sobre essa geração [21], sua implantação está prevista para 2010.



## 1.4 Estrutura da Dissertação

O restante desta dissertação encontra-se organizado da seguinte forma. No Capítulo 2 são descritas 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; os protocolos e a interface aérea do GPRS; e as vantagens e desvantagens dessa tecnologia.

No Capítulo 3, uma breve descrição sobre o TCP e seu controle de congestionamento é realizada, abordando algumas de suas versões. O comportamento do TCP sobre redes sem fio é exposto, seguido dos mecanismos utilizados para melhorar seu desempenho nesse tipo de rede. Também são listados os trabalhos que estudam a interação entre o TCP e o GPRS e descritos os problemas relacionados a essa interação.

O Capítulo 4 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 apresentados seus problemas e limitações, bem como as modificações realizadas para superá-los.

No Capítulo 5, os aspectos relacionados às simulações são apresentados, sendo descritos a topologia, os parâmetros utilizados nas simulações e as métricas coletadas. Os resultados obtidos são apresentados e discutidos detalhadamente.

Finalmente, as conclusões, incluindo a enumeração das contribuições e dos trabalhos futuros, são apresentadas no Capítulo 6.

# 2

# GPRS

Esse capítulo descreve as principais características e conceitos relacionados ao GPRS (*General Packet Radio Service*) [1], tais como a rede GSM (*Global System for Mobile communications*) e os componentes adicionados a ela 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 [2] 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 eles. Considerado uma tecnologia 2G, 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 Subsystem*) e NS (*Network Subsystem*). Na Figura 2.1 pode ser observada uma rede GSM, cujos elementos serão descritos resumidamente a seguir.

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

A MS consiste em um equipamento móvel (o 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*).

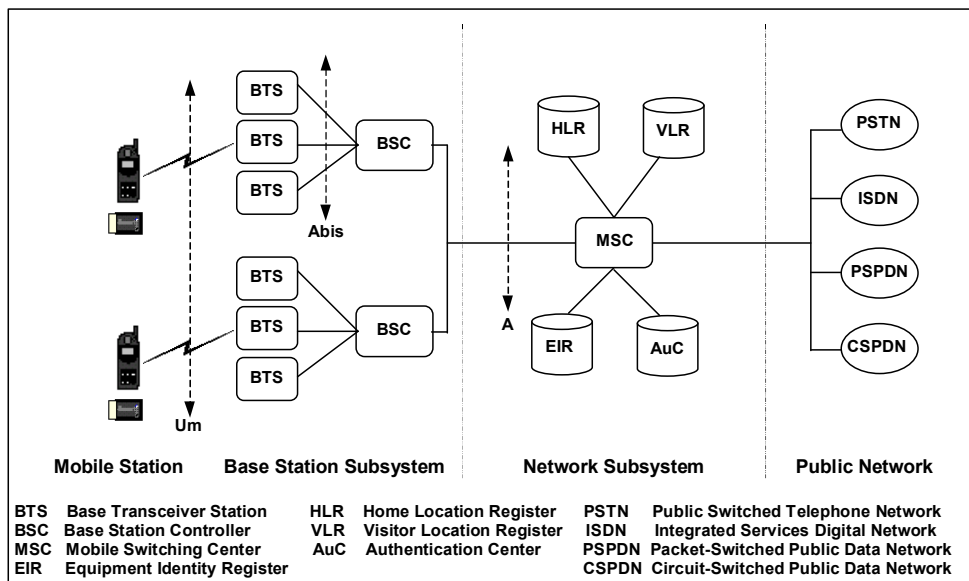


Figura 2.1 Rede GSM

### BSS (*Base Station Subsystem*)

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*).

### 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 2.2) apresenta novos nós: os GPRS *Support Nodes* (GSNs). Esses nós 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, age como uma interface entre a rede *backbone* GPRS e as redes de pacotes de dados externas. Ele 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 “reendereçados” 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.

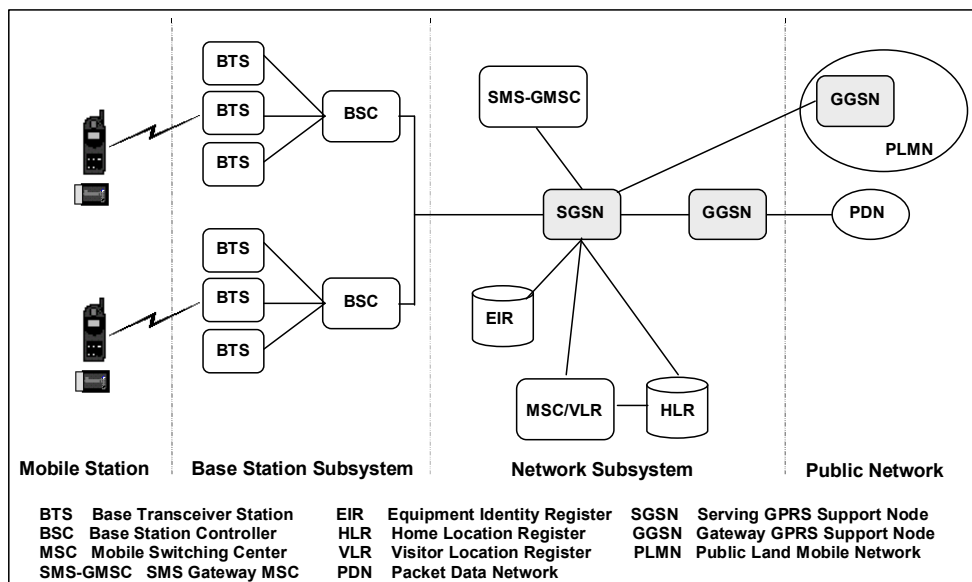


Figura 2.2 Rede GPRS

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 ela 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 Procedimentos 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 (P-TMSI – *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 [22].

### 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 reser-

vados 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*.

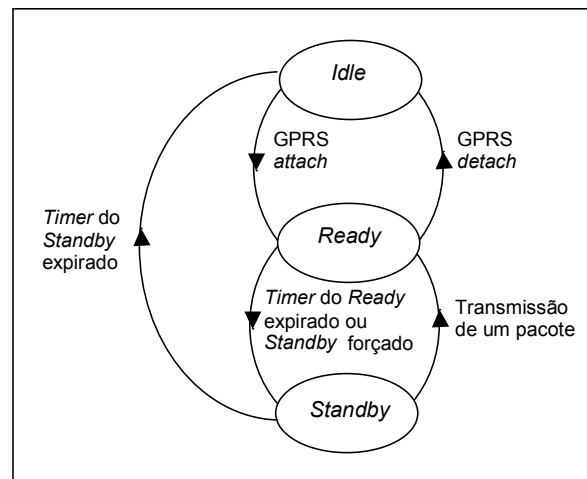


Figura 2.3 Estados da Estação Móvel

A Figura 2.3 ilustra resumidamente a mudança entre os estados da MS.

## 2.4 Classes de Mobilidade GPRS/GSM

Em redes híbridas GSM/GPRS, os móveis podem ser distribuídos em três classes [1]:

- Classe A (GSM/GPRS): o 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 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 2.4. 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.

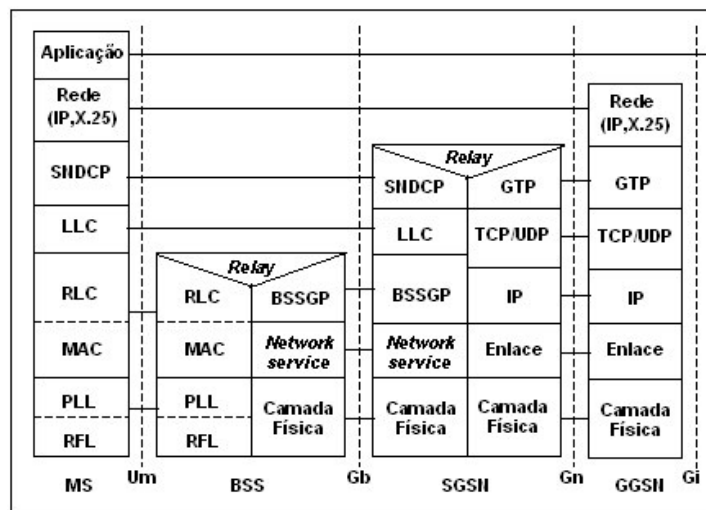


Figura 2.4 Plano de Transmissão

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

- O GTP (*GPRS Tunneling Protocol*) é responsável por “tunelar” os 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* [23], 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*.

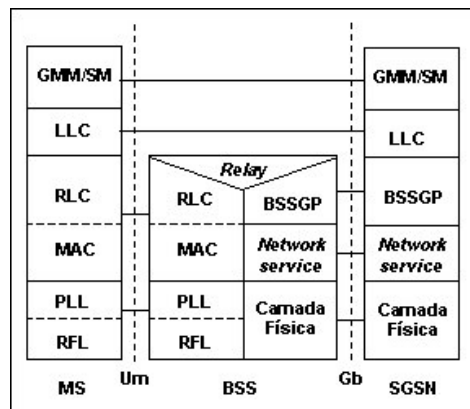


Figura 2.5 Plano de Sinalização: MS-SGSN

Entre a MS e o SGSN (Figura 2.5), 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 2.6, 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*) [15] juntamente com o MTP3 (*Message Transfer Part Level 3*) e MTP2 (*Message Transfer Part Level 2*).

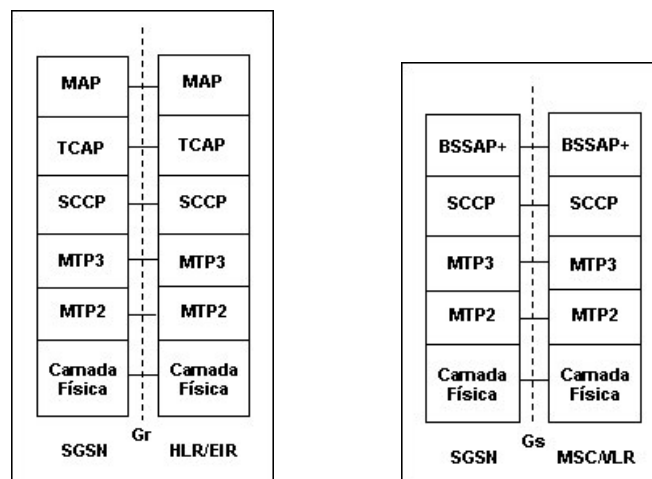
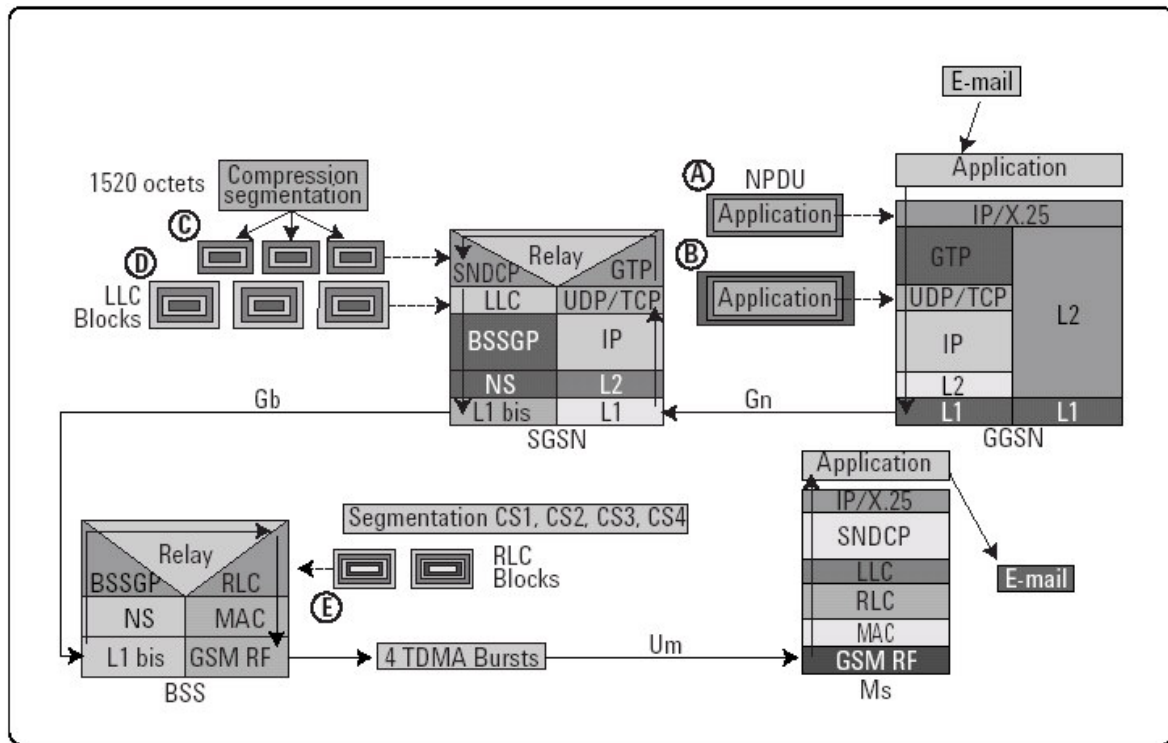


Figura 2.6 Plano de Sinalização: SGSN-HLR, SGSN-EIR e SGSN-MSC/VLR

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 2.7** Transferindo Pacotes de Dados Fim a Fim em uma Rede GPRS [24]

A Figura 2.7 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 a 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 2.7. 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 SNDTCP. 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 SNDTCP classificará o N-PDU como SN-DATA PDU orientado à conexão ou SN-UNITDATA PDU não orientado à conexão. Depois é adicionado um cabeçalho SNDTCP contendo informações sobre compressão e segmentação, como visualizado na letra C da Figura 2.7.

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, deno-

minados blocos RLC. A letra E da Figura 2.7 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.

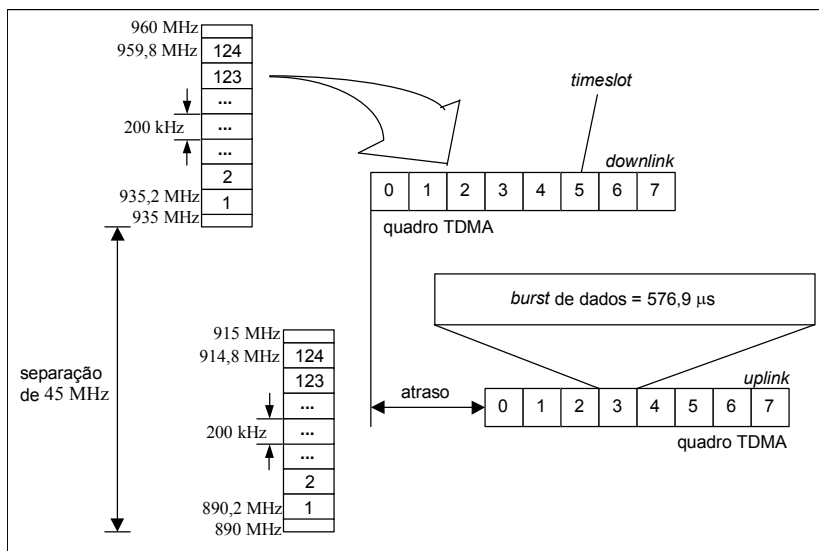
Para a segmentação dos blocos LLC em blocos de dados RLC, precisa ser definida a quantidade de dados úteis a serem transmitidos. Tal quantidade varia de acordo com o cenário de codificação (CS) escolhido. Os possíveis valores são 22, 32, 38 e 52 octetos de dados, correspondendo a CS-1, CS-2, CS-3 e CS-4, 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 2.8), 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 2.8** 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 μs, logo um quadro dura 4,615 ms. A repetição de um *timeslot* particular define um canal físico.

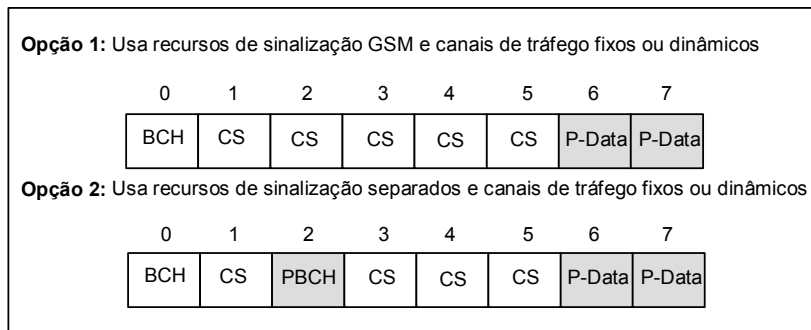
Como pode ser observado na Figura 2.8, um quadro TDMA no sentido *uplink* é iniciado com um atraso de três *timeslots* do início de um quadro TDMA no sentido *downlink*.

Uma estação móvel GSM usa o mesmo *timeslot* nos dois sentidos: *downlink* e *uplink* [22]. 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 2.9** 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 2.9, 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.

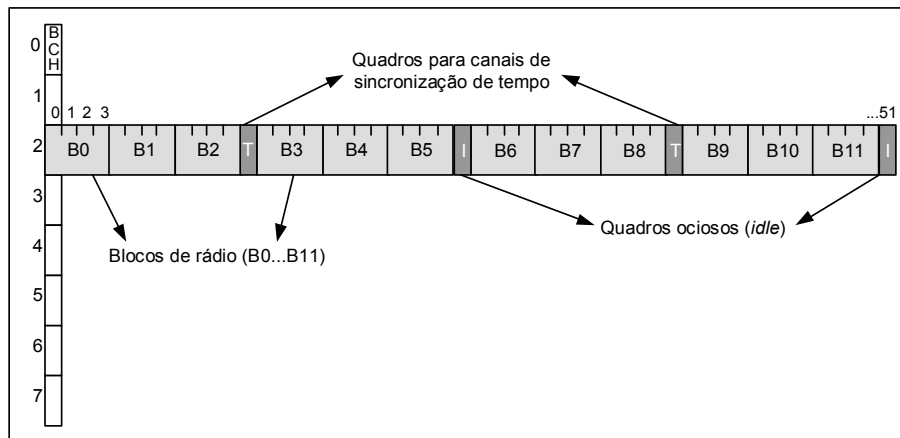
- PCCCH (*Packet Common Control Channel*) – consiste em um conjunto de canais lógicos utilizados para sinalização comum entre o BSS e a MS. É composto por quatro sub-canais:
  - PRACH (*Packet Random Access Channel*) – usado pela MS para solicitar um ou mais canais para transmissão de dados (PDTCH) ou 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 *multiframe*. Um *multiframe* é um conjunto de N quadros (*frames*) TDMA que juntos apresentam uma funcionalidade [24].



**Figura 2.10** Mapeamento de Canais Lógicos em Canais Físicos

Um *multiframe* GPRS é composto por 52 quadros (*frames*) e pode ser visualizado na Figura 2.10. 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) [25]:

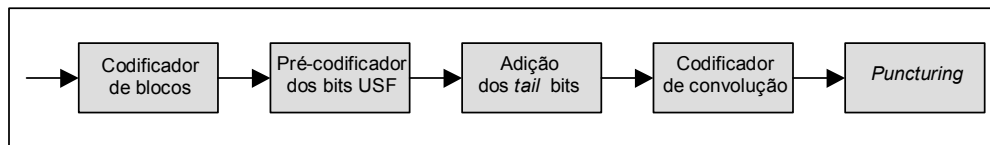
- A primeira combinação permite que todos os canais lógicos sejam mapeados em um canal físico. Uma vez que existem apenas 12 blocos para canais lógicos em um *multiframe*, essa opção reduzirá o número de canais disponíveis para cada categoria.
- A segunda combinação permite que todos os canais exceto 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 2.11** Codificação dos Pacotes de Dados GPRS

A Figura 2.11 mostra o processo de codificação. Primeiramente um codificador de bloco 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 2.1** 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 2.1 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.

## 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 ser transmitidos. A rede continuará reservando blocos para a estação até que esta indique que não há mais blocos a serem transmitidos.

## 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 [26].

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 [26], 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 Vantagens e Desvantagens do GPRS

Como qualquer tecnologia, o GPRS apresenta vantagens e desvantagens.

### Vantagens

As principais vantagens do GPRS estão relacionadas às melhorias quando comparado ao GSM e a outras tecnologias 2G.

Do ponto de vista dos usuários, as principais vantagens incluem:

- altas taxas de transmissão: teoricamente, um usuário pode obter 171,2 kbps utilizando oito *timeslots* ao mesmo tempo (mais de dez vezes a velocidade atingida pelo GSM).
- conectividade: o GPRS facilita a conexão instantânea através da qual a informação pode ser enviada ou recebida assim que necessário, sem precisar estabelecer conexão por modem discada (*dial-up*).
- custo: ao contrário dos sistemas de comutação por circuito (como o GSM), o usuário encontra-se constantemente conectado (serviço geralmente referido como “*always on*”), mas não paga por tempo de conexão, e sim por quantidade de dados transferida.

As principais vantagens para a rede são:

- custo de implantação: uma vez que o GPRS aproveita a infra-estrutura do GSM (isto é, o GPRS apresenta uma rede *overlay* sobre a rede GSM), sua implantação apresenta custos moderados.
- eficiência do espectro: com a comutação por pacotes, os recursos de rádio são alocados apenas quando há transferência de dados, permitindo um maior aproveitamento dos recursos.
- compatibilidade: o GPRS permite fácil conexão com a Internet ou *intranets* baseadas em IP.

## Desvantagens

As desvantagens do GPRS consistem em:

- capacidade limitada: o GPRS compartilha os mesmos recursos de rádio do GSM, o que implica em limitação de usuários GPRS. A quantidade de usuários que podem ser atendidos vai depender do número de *timeslots* reservados para uso exclusivo do GPRS.
- velocidade real menor: apesar de, teoricamente, o GPRS poder alcançar uma taxa 171,2 kbps, na prática isso não ocorre. Pois isso significaria utilizar os oito *timeslots* para um único usuário sem qualquer proteção contra erros. Além de ser inconcebível, essa configuração não pode ser obtida na prática, pois estudos mostram que um único transceptor mal suporta cinco *timeslots*. Segundo [27], os sistemas devem operar em 1+1 (1 canal *downlink* e 1 *uplink*), 2+1 ou 4+1.
- modulação: a técnica de modulação utilizada pelo GPRS, denominada *Gaussian minimum-shift keying* (GMSK), apresenta uma taxa de bits muito menor que a técnica utilizada pelo EDGE (*Enhanced Data Rate for GSM Evolution*), denominada 8PSK (*eight-phase-shift keying*). Como 8PSK será a técnica utilizada pelo UMTS (*Universal Mobile Telecommunications System*), os operadores de rede terão que incorporá-la ao sistema para migrar para a terceira geração.
- ausência de *store and forward*: ao contrário do SMS (*Short Message Service*), o GPRS não apresenta mecanismo de armazenamento incorporado no padrão.

## 2.10 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. Finalmente, as vantagens e desvantagens do GPRS foram apresentadas.

O capítulo seguinte descreverá o TCP e sua interação com o GPRS, sendo apresentados os trabalhos que estudam essa interação e os problemas relacionados a ela.

# 3

## TCP sobre GPRS

Esse capítulo discorre sobre o TCP e sua interação com o GPRS. Uma breve descrição sobre o TCP e seu controle de congestionamento é realizada, abordando algumas de suas versões. Também é exposto o comportamento do TCP sobre redes sem fio, bem como os mecanismos utilizados para melhorar seu desempenho nesse tipo de rede, e são relacionados os trabalhos que estudam a interação entre o TCP e o GPRS, sendo descritos os problemas relacionados a essa interação.

### 3.1 TCP

O TCP (*Transmission Control Protocol*) [6] 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 [28]. 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* [29]. 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  $1/\text{CWND}$ , 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* (se a versão do TCP não apresentar *Fast Recovery*, como o TCP Tahoe). 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). A seguir, serão descritas brevemente as versões utilizadas nesse trabalho.

### 3.1.1 TCP Reno

O TCP Reno [30] implementa os quatro algoritmos básicos do controle de congestionamento sem modificações, ou seja, da forma descrita em [29]. 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*.

O TCP Reno apresenta uma melhora significativa sobre o comportamento do Tahoe (que não implementa o *Fast Recovery*), mas não tem um bom desempenho quando vários segmentos são perdidos em uma mesma janela.

### 3.1.2 TCP New-Reno

O New-Reno [31] é uma versão melhorada do TCP Reno que visa resolver o problema de desempenho relacionado à perda de múltiplos segmentos em uma mesma janela. A mudança refere-se ao comportamento do transmissor durante o *Fast Recovery* ao receber um ACK parcial que confirma o recebimento de alguns mas não todos os segmentos que estavam pendentes no início do período do *Fast Recovery*. No Reno, ACKs parciais fazem com que o TCP saia do algoritmo *Fast Recovery* (pois um segmento só é considerado perdido após o recebimento de três ACKs ou após a ocorrência de um *timeout*). O New-Reno considera ACKs parciais, recebidos durante o *Fast Recovery*, como um indicativo de que o segmento imediatamente após o confirmado pelo ACK (na seqüência) foi perdido e deve ser retransmitido. Assim, quando múltiplos segmentos são perdidos em uma única janela, o New-Reno consegue se recuperar sem que haja *timeout*, retransmitindo um segmento perdido por *round-trip time* ( $RTT^1$ ) até que todos os segmentos perdidos da janela tenham sido retransmitidos. O New-Reno permanece no *Fast Recovery* até que todos os segmentos que estavam pendentes ao ser iniciado esse algoritmo sejam confirmados.

### 3.1.3 TCP SACK

O TCP SACK (*Selective ACKnowledgement*) [32] permite confirmar, de forma não-sequencial, mais de um segmento de uma vez. Utilizando SACK, o transmissor pode saber exatamente quais segmentos foram perdidos, evitando atrasos e retransmissões desnecessárias.

O SACK também modifica o algoritmo *Fast Recovery*. Durante o período em que esse algoritmo está sendo utilizado, o TCP usa uma variável denominada PIPE que representa o número estimado de segmentos “no caminho” (ou seja, segmentos que já foram transmitidos mas ainda não chegaram ao destino). Essa variável é incrementada a cada segmento enviado e decrementada de acordo com os ACKs recebidos, e o transmissor só envia segmentos (novos ou retransmitidos) quando o valor de PIPE é menor que o de CWND.

---

<sup>1</sup> RTT (*Round-Trip Time*) é o intervalo entre o envio de um pacote e o recebimento de sua confirmação. A medição de seu valor é realizada durante toda a conexão, utilizando o Algoritmo de Karn [33], e é utilizada no cálculo do RTO.

Assim como o New-Reno, o TCP SACK permanece no *Fast Recovery* até que todos os segmentos que estavam pendentes ao ser iniciado esse algoritmo sejam confirmados.

### 3.1.4 TCP Westwood

O TCPW [34] consiste em uma versão melhorada do TCP Reno/New-Reno, que modifica o esquema de controle da janela de congestionamento (CWND) do lado do transmissor usando uma estimativa da largura de banda. O transmissor TCPW estima constantemente a taxa de pacotes da conexão através do monitoramento da taxa de recepção de ACKs. Essa estimativa é usada para calcular os valores de CWND e Ssthresh que podem ser utilizados após a detecção de congestionamento (isto é, após o recebimento de três ACKs duplicados ou após um *timeout*). A idéia principal consiste em escolher valores de um modo “mais informado”, consistente com a taxa de conexão efetiva no tempo em que ocorre a perda de segmento(s), ao invés de reduzir a janela à metade ou a um segmento de forma “cega”, como fazem o Reno e o New-Reno. Em casos de perda randômica de pacotes (que ocorre com certa frequência em redes sem fio), essa nova estratégia traz um grande ganho de desempenho, uma vez que evita a redução desnecessária da janela. O valor do Ssthresh também é calculado baseado na estimativa, permitindo que a taxa de transmissão cresça exponencialmente e atinja a largura de banda (*bandwidth*) disponível mais rapidamente.

Apesar de ser uma versão recente, o TCP Westwood já apresenta variações. O TCPW BE (*Bandwidth Estimation*) é a versão original descrita em [34] e apresenta problemas de “*friendliness*” (isto é, sua atitude “agressiva” não é amigável com as outras versões de TCP). A versão seguinte, proposta em [35], tenta melhorar a estimativa da largura de banda compartilhada a fim de superar esse problema. O TCPW CBR (*Combined Bandwidth and Rate estimation*) combina a estratégia de estimação da largura de banda utilizada na primeira versão com uma nova estratégia de estimação da taxa. Estudos mostram que RE (*Rate Estimation*) funciona melhor com perda de segmentos devido a congestionamento, enquanto BE apresenta um melhor desempenho com perdas relacionadas a erros de *link*. Assim, CBR infere a causa predominante da perda e utiliza a estratégia apropriada. Apesar de resolver o problema de *friendliness*, o TCPW CBR apresenta um problema de degradação de eficiência em ambientes que geram muitas perdas. Para contornar isso, um novo método, denominado *Adaptive Bandwidth Share Estimation* (ABSE), é incorporado ao TCPW CBR em [36]. Esse método adapta o tamanho do intervalo de observação utilizado para obter amostras da largura de

banda. Depois as amostras coletadas são suavizadas através de um filtro cuja agilidade é também adaptada para medir a instabilidade da rede. Explorando esses dois níveis de adaptação, o ABSE produz uma estimativa mais precisa da largura de banda compartilhada.

A última versão apresentada chama-se *TCPW with Bulk Repeat (BR)* [37]. O TCPW BR traz modificações para melhorar o desempenho do TCPW em ambientes com muito erro. Quando erros randômicos são freqüentes o bastante para haver múltiplos erros em uma janela, a performance do New-Reno (e, conseqüentemente, do TCPW) decai, pois apenas um segmento é retransmitido por RTT, subutilizando a largura de banda disponível. No TCPW, os erros freqüentes vão impedir que o mecanismo de estimativa da largura de banda aumente a janela até o tamanho alvo. Isto ocorre porque cada vez que uma perda é detectada, a janela é reduzida ou, pelo menos, não é incrementada. Além disso, quando ocorrem sucessivos *timeouts*, qualquer versão do TCP incrementa o valor de RTO até 64 vezes o valor inicial [37], reduzindo a vazão do TCP a praticamente zero. O TCPW BR contorna esses problemas através de três modificações: (1) retransmite todos os pacotes imediatamente após a detecção da segunda perda em uma mesma janela; (2) utiliza um valor fixo para o RTO; e (3) aumenta o valor de CWND mesmo havendo perdas de segmentos.

Há também o TCP Westwood+ [38], que corrige uma deficiência do algoritmo de estimativa da largura de banda original, o qual não funcionava de maneira apropriada na presença de compressão de ACKs.

### 3.2 TCP sobre Redes sem Fio

O protocolo TCP não apresenta um bom desempenho quando é utilizado em redes sem fio. Por ter sido desenvolvido para redes fixas, o TCP deduz que perda de pacotes implica em congestionamento. Contudo, em redes sem fio, perda de pacotes pode significar outras coisas, como ruído na interface aérea, *handovers* ou desconexões não planejadas. Assim, as versões mais comuns do TCP, como Tahoe e Reno, não apresentam funcionamento adequado para redes sem fio.

Há várias propostas para reduzir os efeitos de perdas de pacotes não relacionadas a congestionamento em redes sem fio, as quais estão divididas em cinco grupos [39]: meca-

nismos fim a fim, conexão dividida, protocolo TCP Snoop, sinalização entre camadas e propostas na camada de enlace. Segue abaixo uma breve descrição de cada grupo.

### **Mecanismos Fim a Fim**

Os mecanismos fim a fim não realizam mudanças em nós intermediários e podem modificar tanto a parte fixa quanto a parte móvel. Algumas RFCs como SACK [32] e *Timestamps* [40] encontram-se nesse grupo. Por não ferir a semântica fim a fim do TCP e por modificar apenas o receptor e o transmissor, esta é a opção mais desejada e de mais fácil aplicação. Contudo, ainda não foi desenvolvida uma solução que obtivesse melhor desempenho que todas as outras propostas. O TCP Westwood também é um mecanismo fim a fim e tem alcançado ótimos resultados comparado ao SACK, até então a melhor proposta desse grupo [34]-[37].

### **Conexão Dividida**

A idéia dessa abordagem é dividir a conexão TCP entre o receptor e o transmissor em duas conexões diferentes na estação base: uma entre a parte móvel e a estação base (BS) e outra entre a BS e a parte fixa. O *Indirect* TCP (I-TCP), descrito em [41], pertence a esse grupo. Entre os principais problemas desse grupo estão a quebra da semântica fim a fim do TCP e a sobrecarga (*overhead*) causada pelo duplo processamento do TCP na BS (comparado a nenhum processamento para abordagens sem divisão de conexão).

### **Protocolo TCP Snoop**

O TCP Snoop [42] consiste em ter um agente (*proxy*) instalado na estação base que realiza retransmissões locais na parte sem fio dependendo do tipo de ACK (recebido da estação móvel) e dos *timers* locais. O Snoop esconde da parte fixa as perdas no *link* sem fio. Quando o agente detecta uma perda, ele retransmite o segmento TCP perdido para a estação móvel (MS), espera pelo ACK correspondente e o envia para a parte fixa antes que esta descubra que houve perda de segmentos. Estudos como [43] mostram ser essa a melhor proposta até o momento. A principal desvantagem dessa abordagem é a necessidade de um agente na estação base. Além disso, caso o transmissor seja a MS, o TCP precisa ser modificado para responder a notificações explícitas de perda (*Explicit Loss Notification* – ELN) da BS.

## Sinalização entre Camadas

Nessa abordagem, as camadas de enlace (*link*) e de rede nos nós intermediários ou na MS informam o transmissor TCP na parte fixa sobre o estado do *link* sem fio (perda de pacotes, *handovers* etc.). Em [44], por exemplo, o transmissor TCP recebe informação sobre *handovers* para iniciar o algoritmo *Fast Retransmit* e evitar a espera ociosa pela retransmissão por *timeout*.

## Propostas na Camada de Enlace

Nesse tipo de proposta, a estação base realiza retransmissões na camada de *link* de rádio, não havendo troca de informações com camadas superiores. A solução empregada depende do protocolo de *link* de rádio utilizado para cada rede sem fio.

Além dos problemas (e propostas para superá-los) relacionados ao TCP que são comuns às redes sem fio, há problemas específicos de cada tecnologia. A seguir serão apresentados os problemas particulares do GPRS.

## 3.3 TCP sobre GPRS

Nos últimos anos têm sido realizados estudos avaliando o comportamento do TCP sobre o GPRS. No entanto, cada um deles em geral avalia apenas poucos aspectos possíveis em um cenário que suporta TCP com GPRS. Abaixo são apresentados alguns desses trabalhos.

### 3.3.1 Trabalhos Relacionados

Em [7] 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. 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 [8], os mesmos autores analisam, agora, a vazão total para o CS-2, com C/I de 12 db, 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), os resultados confirmam as conclusões anteriores.

Utilizando o simulador desenvolvido em [7], Meyer apresenta em [9] 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 [10], 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 à Web, observando a taxa de pacotes de dados recebida pelo usuário. Os autores concluem que, comparado à comutação por circuito (GSM), o GPRS é significativamente superior.

Em [11], o desempenho do SACK e *Timestamps* em um sistema GPRS é estudado através da simulação da transferência de um arquivo de 512 Kbytes. 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 o SACK resulta em melhor desempenho e que, apesar de melhorar o desempenho do Reno, o uso de *Timestamps* atrapalha a performance do SACK.

Nos últimos dois anos, medições reais vêm sendo realizadas na Universidade de Cambridge. Utilizando um *Testbed*, foram detectados problemas na interação entre o TCP e o GPRS (descritos na próxima seção). Tais problemas são apresentados em [12], seguidos de uma proposta para superá-los. O trabalho propõe a inserção de um *proxy* no GGSN. Esse *proxy* divide a conexão em duas partes e na parte sem fio uma versão modificada do TCP é utilizada. Em [13] os problemas são mais detalhados e uma nova proposta é realizada: o uso de dois *proxies*, um em cada extremidade do *link* GPRS. Entre os *proxies*, um protocolo “per-

sonalizado” é empregado para reduzir o volume de tráfego sobre o *link* GPRS e tentar amenizar os efeitos do alto RTT gerado pelo GPRS.

### 3.3.2 Problemas do TCP sobre GPRS

Os problemas específicos encontrados na interação entre o TCP e o GPRS, observados no sentido *downlink*, apresentados em [12] são:

**Slow Start muito lento:** O primeiro problema consiste em uma fase *Slow Start* muito demorada devido aos altos valores alcançados pelo RTT (1000 milissegundos ou mais). Nos experimentos realizados em [12], sob boas condições de rádio, o TCP gasta mais de 6 segundos para que a janela de congestionamento (CWND) atinja um tamanho que permita utilizar a largura de banda disponível. Assim, fluxos curtos (como o *download* de uma página Web) apresentam tempos de transferência bastante altos, pois não chegam a sair do período de *Slow Start*.

**Compressão de ACKs:** Apesar de serem gerados ACKs de forma moderada (isto é, à medida em que segmentos são recebidos), eles são enviados em grupos de quatro. Esse efeito é observado com frequência e ocorre quando a estação móvel possui dados para enviar e para receber de forma concorrente. O agrupamento de ACKs prejudica o cálculo do RTO, uma vez que este é baseado na medição de RTT e os ACKs não são enviados ao serem gerados.

**Enfileiramento Excessivo no GGSN:** Devido à sua baixa largura de banda, o *link* GPRS é normalmente o gargalo de uma conexão TCP. Desta forma, pacotes (segmentos) destinados ao *downlink* são enfileirados no GGSN. Esse problema gera outros, entre eles: injustiça entre outros fluxos existentes e novos; grande probabilidade de *timeout* na requisição inicial para estabelecer nova conexão; recuperação lenta do TCP após *timeout* (RTO) devido ao tempo gasto para liberar os pacotes que ficaram na fila.

Neste trabalho de dissertação, um novo problema, relacionado ao tamanho do segmento TCP e ao tamanho da janela de transmissão do GPRS (RLC), é identificado, sendo apresentado no Capítulo 5.

### 3.4 Considerações Finais

Esse capítulo apresentou características relacionadas ao TCP e à sua interação com o GPRS.

Inicialmente foi realizada uma breve descrição sobre o TCP e seu controle de congestionamento. Foram descritos os quatro algoritmos que dirigem o controle de congestionamento – *Slow Start*, *Congestion Avoidance*, *Fast Retransmit* e *Fast Recovery* – e algumas versões do TCP geradas pelas variações desses algoritmos – Reno, New-Reno, SACK e Westwood. O comportamento do TCP sobre redes sem fio, bem como os mecanismos utilizados para melhorar seu desempenho nesse tipo de rede, foram expostos em seguida.

Também foram listados e descritos de forma sucinta alguns trabalhos que estudam a interação entre o TCP e o GPRS, e os problemas relacionados a ela. No Capítulo 4, o ambiente de simulação utilizado para estudar tal interação neste trabalho será descrito.

# 4

## GPRS no ns-2

Esse 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 *Network Simulator – version 2* (ns-2) [45] é um simulador de eventos discretos cujo objetivo é auxiliar em pesquisa sobre redes de comunicação. Desenvolvido no *Lawrence Berkeley National Laboratory*, o ns começou como uma variante do *REAL Network Simulator*, sendo depois apoiado pelo projeto VINT (*Virtual Internetwork Testbed*) da DARPA (*Defense Advanced Research Projects Agency*). Atualmente, o ns é desenvolvido sob o apoio da DARPA, através do projeto SAMAN (*Simulation Augmented by Measurement and Analysis for Networks*), e da NSF (*National Science Foundation*), com o projeto CONSER (*Collaboration Simulations for Education and Research*), e recebe contribuições de outros pesquisadores, como a ACIRI (*AT&T Center for Internet Research at ICSI*).

Orientado a objetos, o ns foi desenvolvido utilizando duas linguagens: C++ e OTcl. Apresenta duas hierarquias de classes, uma hierarquia compilada, escrita em C++, e uma hierarquia interpretada, escrita em OTcl. Essas hierarquias estão relacionadas de tal forma que, do ponto de vista do usuário, há uma correspondência de um-para-um entre uma classe da hierarquia interpretada e uma da compilada (isto é, cada objeto em C++ possui um correspondente em OTcl).

O uso de duas linguagens permite, de um lado, implementações detalhadas de protocolos (com manipulação de pacotes, bytes e cabeçalhos de maneira eficiente) utilizando C++ e,

do outro, configurações de simulações (através da variação de parâmetros e mudança de cenários) usando OTcl.

O ns implementa protocolos de rede, como TCP e UDP; comportamento de fonte de tráfego, como FTP (*File Transfer Protocol*), Telnet, Web, CBR (*Constant Bit Rate*) e VBR (*Variable Bit Rate*); mecanismos de gerenciamento de fila de roteador, como *Drop Tail*, RED (*Random Early Detection*) e CBQ (*Class Based Queue*); e algoritmos de roteamento, como o algoritmo de Dijkstra. Também implementa *multicasting* e alguns protocolos da camada MAC para simulação de LANs (*Local Area Networks*).

O ns possui o código aberto e, por isso, seus usuários podem modificá-lo e estendê-lo para atender melhor às suas necessidades. Algumas dessas extensões e alterações – que variam de implementações de novos protocolos (isto é, protocolos não inclusos na distribuição do ns) a correções e melhoramento de módulos já implementados – encontram-se disponibilizadas em [46]. Entre essas contribuições está o módulo do GPRS, descrito a seguir.

## 4.2 Módulo GPRS

O módulo do GPRS foi desenvolvido por Richa Jain 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 4.1. 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.

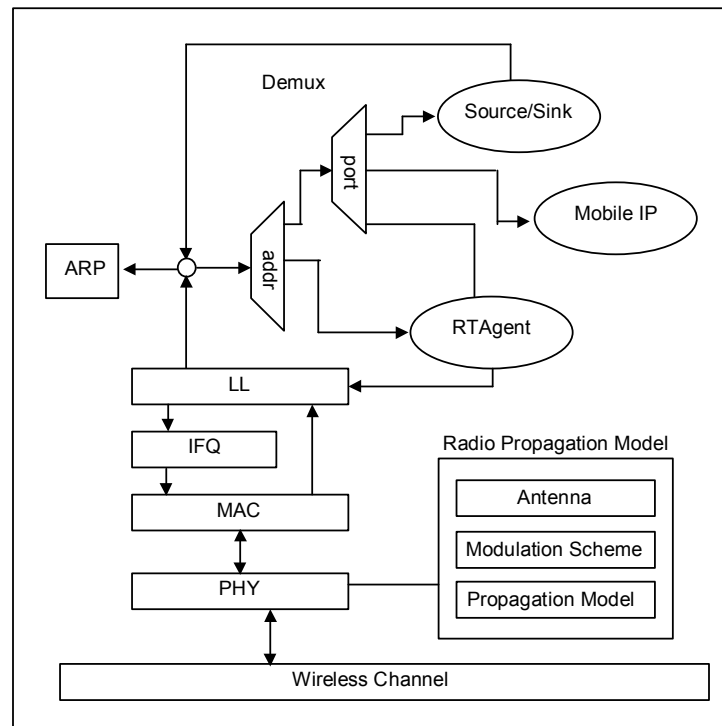


Figura 4.1 Pilha de Protocolos de Rede em um Nó Móvel no ns [47]

- **Address Resolution Protocol (ARP):** esse 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 4.2. As modificações, assim como o novo componente, são descritas a seguir.

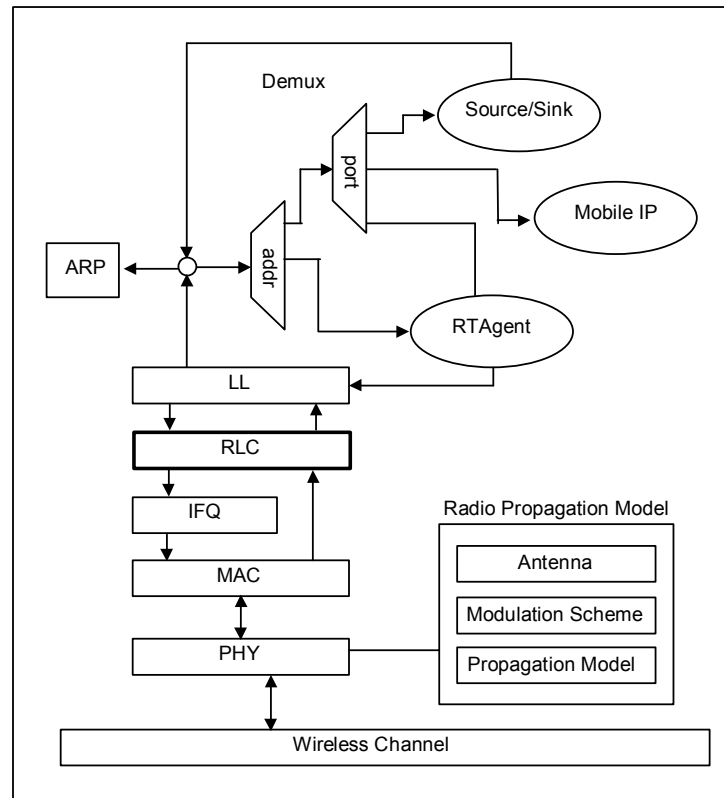


Figura 4.2 Pilha Modificada de Protocolos de Rede em um Nó Móvel GPRS [47]

### ***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.

### ***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).



## ***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.

### **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.

### **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.

### **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*.

### **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 Adhoc 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 [48].

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, a 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 resumo do 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 [47]. 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 um 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 con-

cordância com a especificação do GPRS. Tais problemas produziam resultados errados e, portanto, tiveram que ser resolvidos para que o módulo pudesse ser utilizado. Abaixo segue uma lista de 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 *uplink*;
- 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 próximo da especificação.

Para uma melhor compreensão, cada problema citado na seção anterior será detalhado, seguido da solução aplicada para repará-lo. Convém ressaltar que a primeira modificação realizada consistiu em migrar o módulo para a versão até então mais atual do ns: a versão 2.1b9a.

### 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) apresenta 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, como será explicado a seguir.

O primeiro problema ocorria 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 frequência, 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 a 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 a 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 frequência 1/*slot* 5 e o antigo, o par frequência 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 *uplink*. Assim, quando as MSs recebiam dados (situação mais comum no mundo real), nenhum pacote era perdido. A solução seria simplesmente aplicar o modelo também no sentido *downlink*. 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 [8], [49], [50] e [51]. O cenário assumido para a geração da tabela foi um canal urbano (*Typical Urban Channel - TU*) com usuários se movimentando a 50 km/h.

Tabela 4.1 BLER [51]

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



A Tabela 4.1 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, ela 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 [26], 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 Implementação do Modo com Confirmação da Camada RLC de Acordo com a Especificação

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 ela 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, a 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 checa 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 2.1, na Seção 2.7.4.

### 4.4.2 Implementação do Modelo Adaptativo

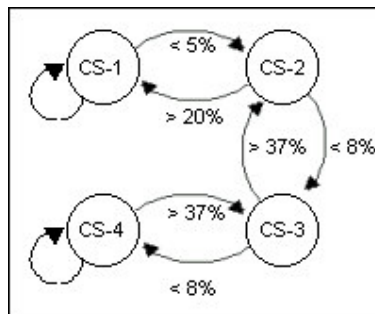
O cenário de codificação é escolhido de acordo com as condições do canal (ruído). Quanto piores as condições do ambiente de transmissão, maior deve ser a proteção dos dados. E quanto maior a proteção, menor a quantidade de dados úteis (isto é, dados do usuário). Assim, se não for utilizado o cenário adequado, haverá perda de dados ou subutilização do canal. Se, por exemplo, o CS-4 for utilizado em um ambiente com muito ruído, vários pacotes serão perdidos, pois este cenário não apresenta proteção aos dados. Por outro lado, em um ambiente bom, com pouco ruído, pacotes praticamente não são perdidos, não havendo necessidade de redundância (proteção) de dados. Se em um ambiente como esse, em que o uso do CS-4 seria apropriado, o CS-1 for escolhido, a taxa de dados transmitidos será bastante inferior a que poderia ser alcançada, uma vez que serão enviados menos dados úteis. Logo, se houvesse uma forma de mudar o CS utilizado de acordo com as condições do ambiente, a quantidade de dados transmitidos seria maior.

Em [51], um modelo adaptativo para os cenários de codificação é apresentado. Tal modelo modifica o cenário de codificação baseado no percentual de blocos descartados du-

rante a transmissão da janela. Isto é, para cada bloco enviado, um BLER é coletado, indicando o descarte ou não do bloco. Ao término da transmissão de uma janela (ou seja, após a transmissão de 64 blocos), calcula-se o percentual de blocos descartados (%BD), dividindo-se o total de blocos descartados (BD) pelo total de blocos transmitidos (BT):

$$\%BD = BD / BT * 100$$

De acordo com o valor desse percentual, o cenário de codificação pode ser mudado. Também é apresentado um estudo que determina os valores para a transição entre os cenários. Esse valores podem ser observados na Figura 4.3.



**Figura 4.3** Valores dos Parâmetros de Transição entre os Cenários de Codificação [51]

Atualmente, esse modelo adaptativo encontra-se implementado no módulo GPRS.

#### 4.4.3 Implementação de *Multislots*

O GPRS permite que um usuário use mais de um *timeslot* em paralelo no sentido *downlink*. 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 quatro, sendo as configurações indicadas 1+1 (1 canal *downlink* e 1 *uplink*), 2+1 e 4+1 [27].

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, utiliza-se mais de um *timeslot* no sentido *downlink*, 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-2), 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 a interação entre os protocolos TCP e GPRS. São descritos a topologia, 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 4. A Figura 5.1 ilustra a topologia utilizada nas simulações: ela é formada por uma rede GPRS, dois roteadores intermediários e um conjunto de sistemas finais (*hosts*). 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.

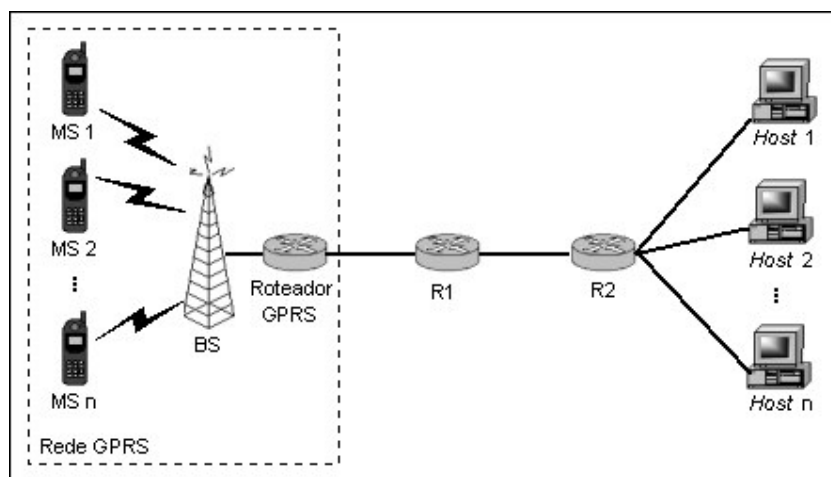


Figura 5.1 Topologia Utilizada nas Simulações



O número de sistemas finais é igual ao número de MSs, havendo uma correspondência de um-para-um entre as MSs e os *hosts* de mesmo índice. Ou seja, o *host i* se comunica com a MS *i*. A transmissão de dados ocorre no sentido *host* → MS (isto é, no sentido *downlink*), simulando um usuário realizando *download* de arquivos.

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 do TCP sobre o GPRS sem, contudo, encobrir ou influenciar esse comportamento.

Os parâmetros utilizados na simulação são apresentados na Tabela 5.1. O objetivo principal deste trabalho é avaliar o desempenho do protocolo TCP quando utilizado sobre o GPRS. Para tanto, foram analisadas as versões Reno, New-Reno, SACK e Westwood. 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 (536, 1000 e 1500 bytes) e apenas um tamanho (200 bytes) de pacote UDP.

**Tabela 5.1** Parâmetros de Simulação

Parâmetro	Valor
Protocolo de transporte	TCP, UDP
Versão do TCP	Reno, New-Reno, SACK, Westwood
Tamanho do pacote TCP	536, 1000, 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	3 dB
Modelo adaptativo	sim, não
Cenário de codificação	1, 2, 3, 4
Modo RLC	com confirmação (ACK), sem confirmação (UNACK)
<i>Multislot - uplink</i>	1 TS
<i>Multislot - downlink</i>	1, 2, 4 TS
Número de PDCH	5
Tempo de simulação	100 segundos

Todos os experimentos foram realizados com 1, 5 e 50 fontes de dados (em usuários móveis). 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, além de modificados dinamicamente

através do modelo adaptativo descrito na Seção 4.4.2. Foi utilizado um *timeslot* no sentido da MS para a BS (*uplink*) e um, dois e quatro *timeslots* no sentido contrário (configurações 1+1, 2+1 e 4+1). 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 as sementes com valores cuidadosamente escolhidos (sementes consideradas “boas” pelo ns-2). Esse número de replicações foi escolhido por representar um compromisso aceitável entre o custo computacional e a confiança estatística nos resultados. Para todos os resultados, foi calculado um intervalo de confiança assintótico ao nível de 99%, que é representado através de barras horizontais nos gráficos da Seção 5.2.

Durante a execução dos experimentos, foram coletadas várias métricas, tanto relacionadas aos protocolos de transporte (TCP e UDP), quanto ao GPRS. As principais métricas TCP 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 5.1, variando os parâmetros exibidos na Tabela 5.1.

### 5.2.1 TCP Reno sobre GPRS

Os primeiros experimentos estudam o comportamento do TCP Reno sobre o GPRS. A Figura 5.2 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.

## 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 5.2 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.

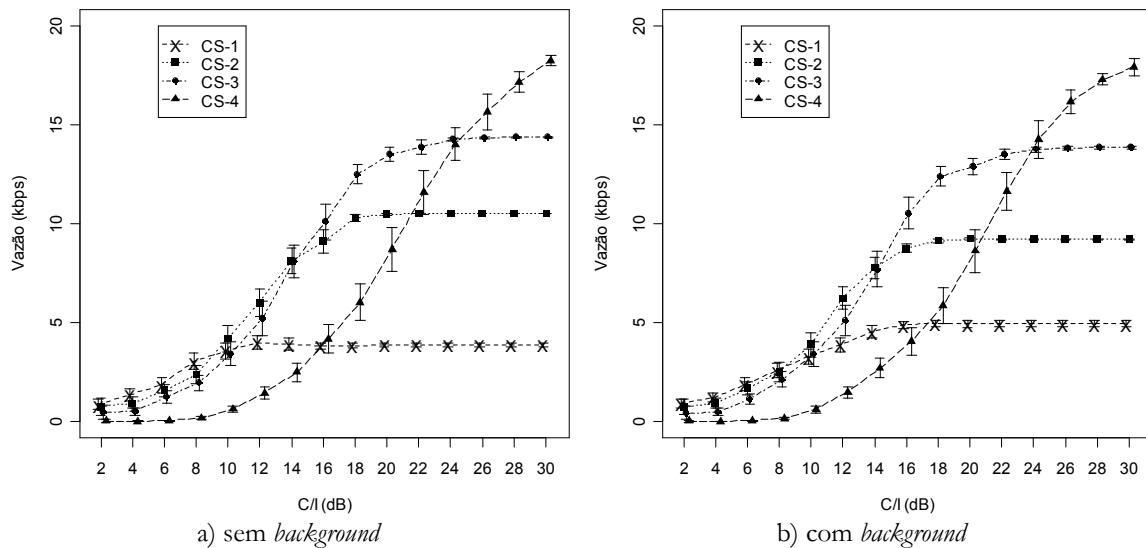
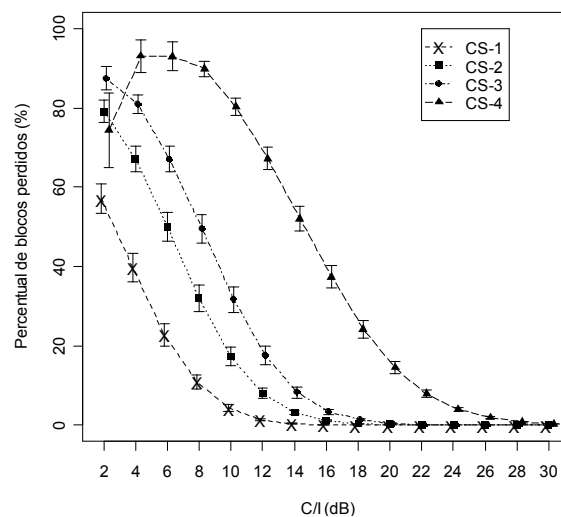


Figura 5.2 Vazão do TCP Reno para os quatro Cenários de Codificação

## CS Adequado para Cada C/I

Observando o gráfico da Figura 5.2a, 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 [12].

O percentual de blocos descartados, considerando os mesmos parâmetros, é exibido na Figura 5.3. 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.

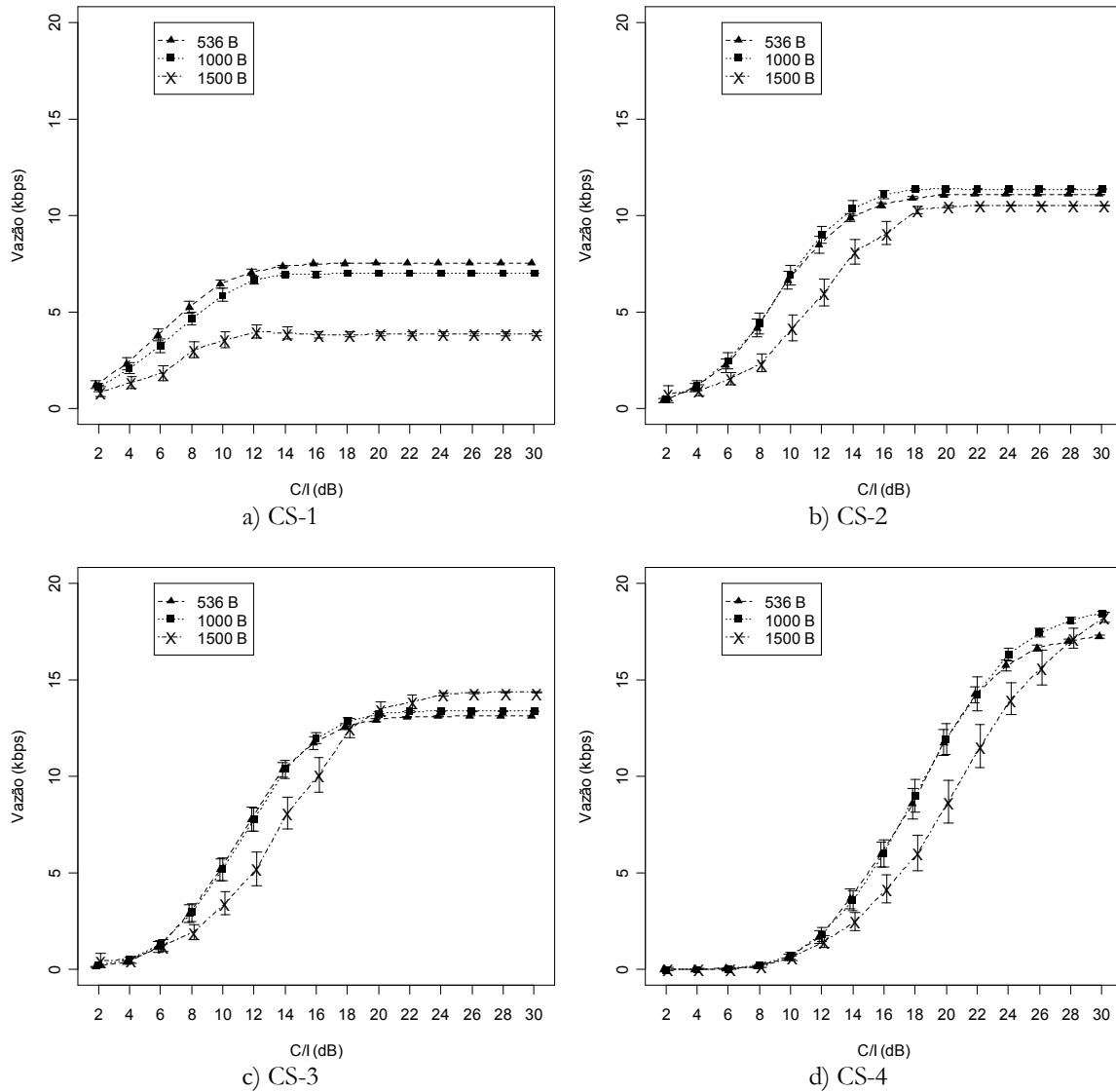


**Figura 5.3** Percentual de Blocos Descartados para o TCP Reno para os quatro CSs

### Tamanho do Pacote (Segmento) TCP

Todos os estudos relacionados (Seção 3.3.1) utilizam tamanho fixo de pacotes (segmentos) TCP, sendo esse tamanho diferente entre os estudos. A fim de verificar se o tamanho do pacote influencia na interação entre o TCP e o GPRS, esse tamanho foi variado, assumindo três valores diferentes: 536, por ser o valor padrão do TCP; 1500, por ser um tamanho comum, observado com frequência em transferências de arquivos grandes na Inter-

net; e 1000, por ser um valor intermediário. Os resultados obtidos para cada cenário de codificação são ilustrados na Figura 5.4.



**Figura 5.4** Vazão do TCP Reno para Diferentes Tamanhos de Pacote

Os valores apresentados nos quatro gráficos foram gerados utilizando 1 MS, com configuração 1+1. Para o CS-1 (Figura 5.4a), 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 [26], 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 5.4a. Com os demais CSs (Figuras 5.4b a 5.4d) 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 acima, o tamanho de 1500 bytes não ser o mais apropriado para o uso do TCP sobre o GPRS, este será o tamanho utilizado nos resultados seguintes, pois é o valor mais comum observado na transferência de arquivos.

### **Número de Fontes**

A Figura 5.5 ilustra a vazão do TCP Reno para os quatro CSs variando o número de estações móveis utilizadas nas simulações. Os valores utilizados foram 1, 5 e 50 usuários. 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 consideram apenas uma estação móvel.

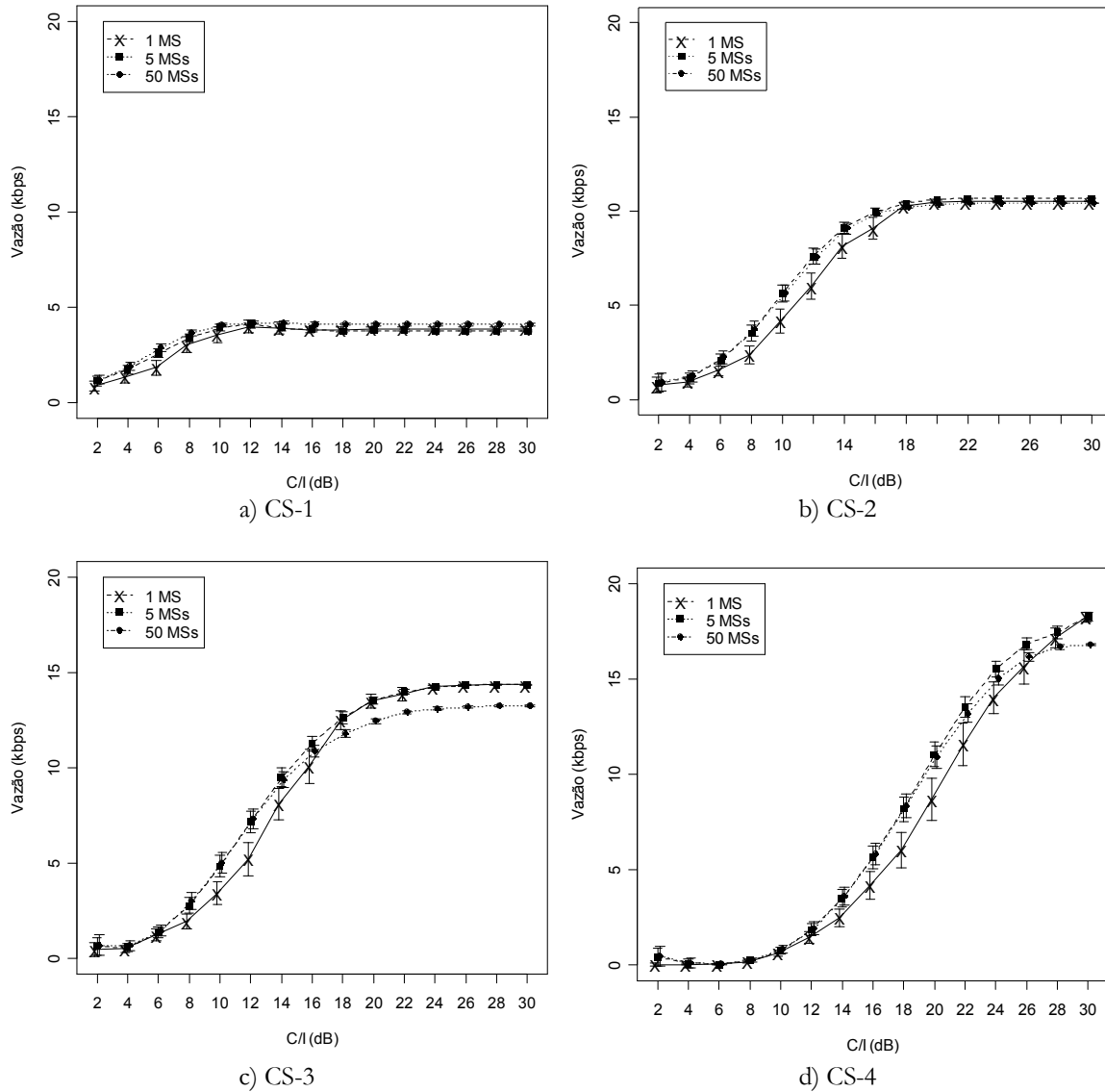


Figura 5.5 Vazão do TCP Reno para Números Diferentes de Estações Móveis

### Multislot

A Figura 5.6 ilustra as vazões atingidas variando o número de *timeslots* utilizados por 1 usuário no sentido *downlink*. Além da configuração *singleslot* (1+1) utilizada para os demais resultados, foram utilizadas as configurações *multislot* 2+1 e 4+1. Comparando as Figuras 5.5a e 5.5b, observa-se que a vazão praticamente dobra, o que é absolutamente coerente visto que o número de *slots* utilizados é o dobro. Os resultados obtidos para a configuração 1+1 utilizando os mesmos parâmetros podem ser visualizados na Figura 5.2a. Como esperado, a vazão obtida é aproximadamente igual à metade da vazão alcançada com 2 TSs. Como o

comportamento do TCP não muda de forma significativa ao aumentar o número de *timeslots*, os resultados seguintes levam em consideração apenas a configuração *singleslot* (1+1).

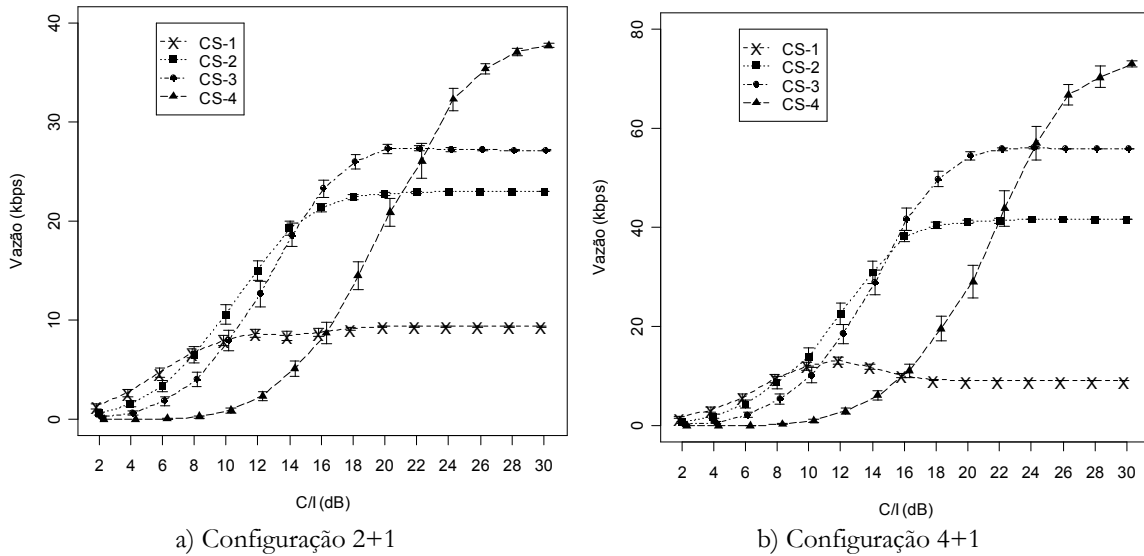


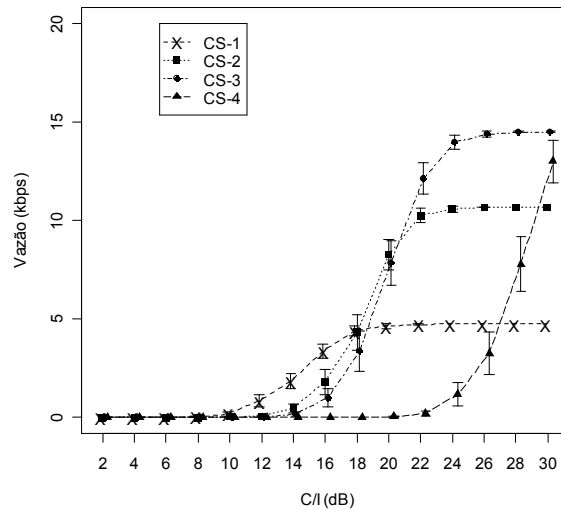
Figura 5.6 Vazão do TCP Reno com Múltiplos *Timeslots*

### 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, tanto neste trabalho como nos trabalhos relacionados, utilizaram apenas o modo com confirmação. Portanto, decidiu-se verificar o desempenho do TCP sobre o GPRS utilizando o modo sem confirmação.

A Figura 5.7 ilustra a vazão obtida por uma MS submetida a diferentes condições de ambiente, utilizando a configuração 1+1 e pacotes de 1500 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 5.7** Vazão do TCP Reno Utilizando o Modo RLC sem Confirmação (UNACK)

Os gráficos ilustrados na Figura 5.8 comparam o desempenho do TCP utilizando os dois modos da camada RLC. A Figura 5.8a 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 5.8b, 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.

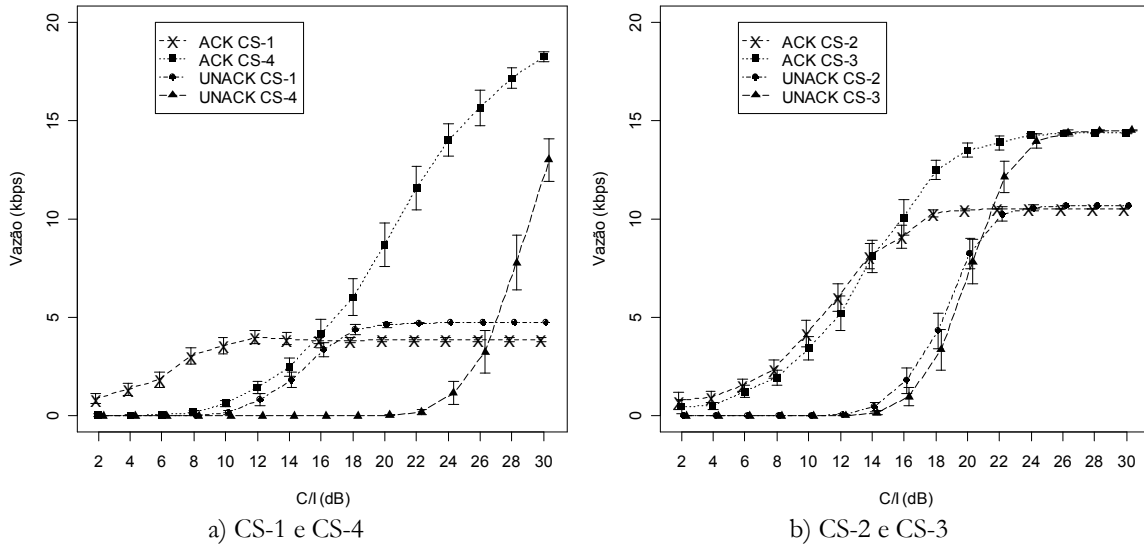
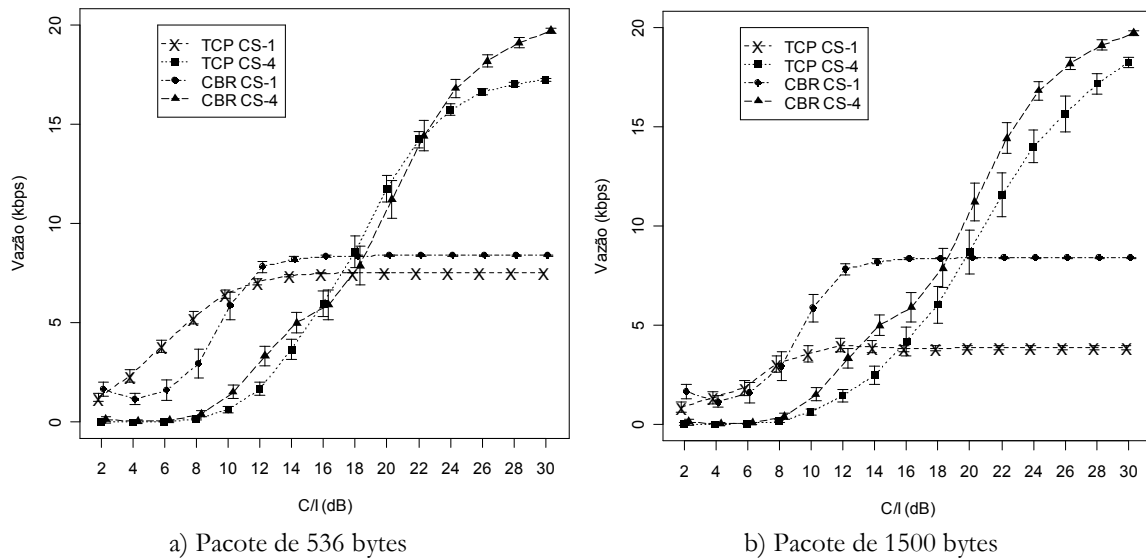


Figura 5.8 Vazão do TCP Reno Variando o Modo de Transmissão RLC

## 5.2.2 TCP x UDP

Para fins de comparação, os mesmos experimentos foram também realizados para tráfego UDP. 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 5.9 apresenta os dados obtidos para o CS-1 e o CS-4 utilizando TCP Reno e UDP. Na Figura 5.9a o tamanho de pacote é 536 bytes. Pode-se perceber que, neste caso, o TCP apresenta um desempenho próximo do UDP. Contudo, na Figura 5.9b (na qual o tamanho do pacote é 1500 bytes), quando a perda de pacotes torna-se desprezível para o CS-1 (acima de 12 dB), a vazão alcançada pelo TCP ( $\approx 4$  kbps) é praticamente a metade da vazão do UDP ( $\approx 9$  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 TCP – Seção 5.2.1).

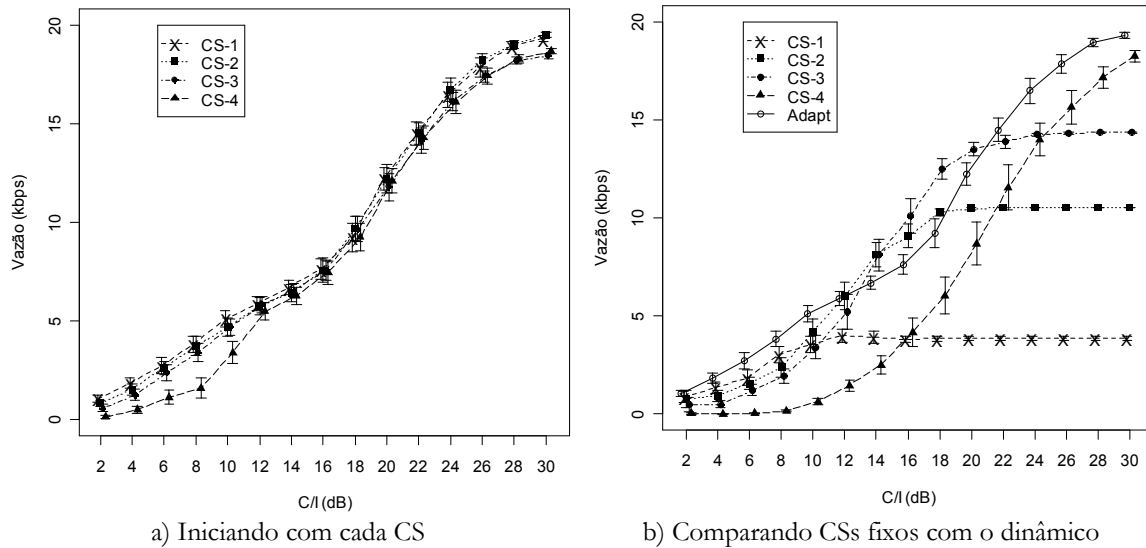


**Figura 5.9** Vazão do TCP Reno e do UDP (CBR) para CS-1 e CS-4

Novamente os resultados mostram que o tamanho do pacote influencia o desempenho do TCP 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 5.9a, 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).

### 5.2.3 Modelo Adaptativo

Todos os experimentos realizados com CS fixo foram também realizados com o modelo adaptativo (Seção 4.4.2). A Figura 5.10a ilustra a vazão alcançada por uma MS utilizando o modelo adaptativo, variando o CS inicial. Como pode ser observado, independente do CS inicial, o comportamento do modelo é semelhante, ou seja, migra para o mesmo CS final. Por isso, a Figura 5.10b mostra a vazão obtida pelo modelo adaptativo (com CS-1 como cenário inicial) comparando-o ao desempenho de cada CS fixo (já apresentados na Figura 5.2).

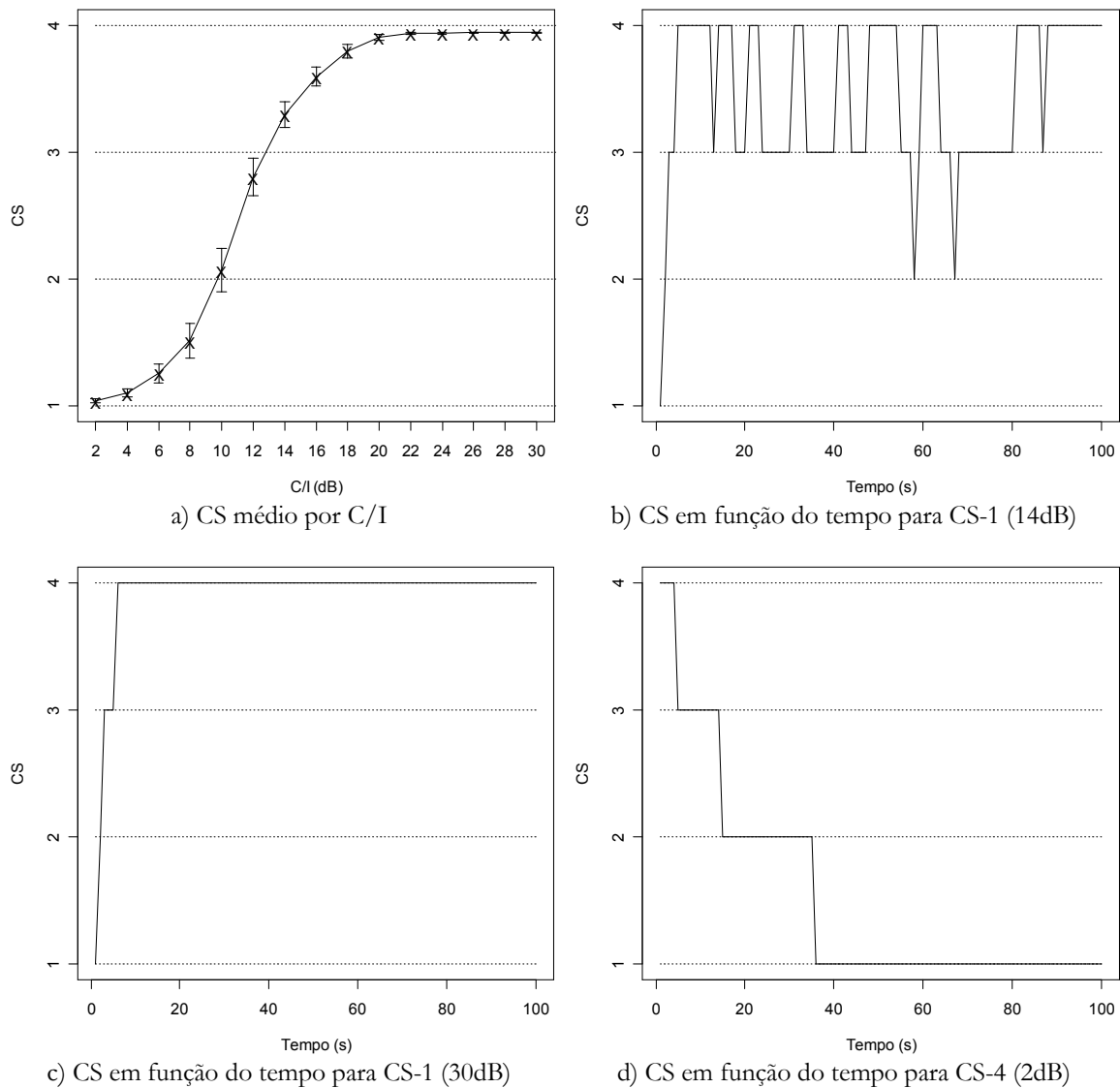


**Figura 5.10** Vazão do TCP Reno Utilizando o Modelo Adaptativo para CS

Analisando o gráfico, pode-se notar que o modelo adaptativo consegue, na maioria dos casos, obter uma vazão maior do que a alcançada utilizando CSs fixos. Em outras palavras, o modelo consegue fazer com que o cenário de codificação mais adequado ao ambiente seja utilizado, alcançando a máxima vazão possível na maioria dos casos. Pode-se perceber também que o modelo não está devidamente ajustado, ou seja, os valores utilizados para a transição entre os CSs (Figura 4.3) não são adequados para o cenário em estudo. O modelo tende a migrar rapidamente para um CS maior, como pode ser observado na Figura 5.11a, que mostra o CS médio para cada C/I, tendo o CS-1 como cenário inicial. Considerando as conclusões retiradas da Figura 5.2, na faixa de 9 a 14 dB o cenário de codificação mais adequado é o CS-2, enquanto o CS-3 é mais apropriado entre 15 e 24 dB. Na Figura 5.11a verifica-se que o modelo migra para o CS-2 no intervalo de 9 a 11 dB, e entre 12 e 16 dB o cenário final é o CS-3. A partir de 17 dB, quando o CS-3 ainda é o cenário mais propício, o modelo migra para o CS-4.

### Variação do cenário de codificação

O gráfico da Figura 5.11b exibe a variação do CS durante uma replicação, na qual o cenário inicial é o CS-1 e o C/I é de 14 dB. A curva mostra a atividade do modelo adaptativo, que muda de cenário de acordo com o percentual de blocos descartados por janela. Também pode-se perceber que o modelo está migrando bastante para o CS-4 quando deveria permanecer mais tempo no CS-3, que é o cenário mais adequado para esse ambiente.

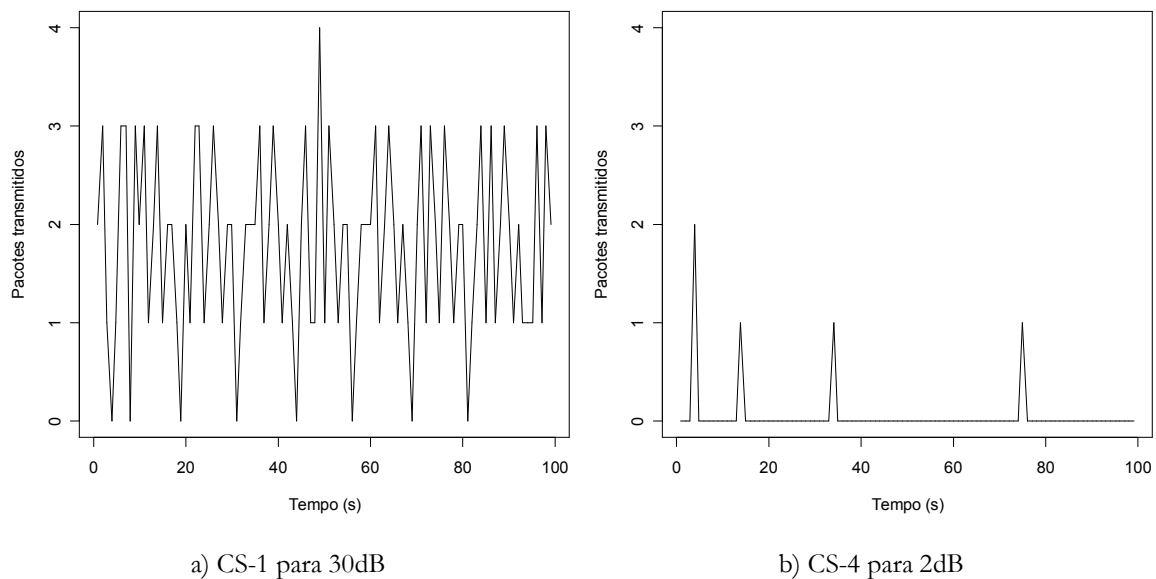


**Figura 5.11** Variação do Cenário de Codificação Utilizando o Modelo Adaptativo

A Figura 5.11c mostra o modelo agindo em um ambiente cujas condições são muito boas (30 dB) mas o cenário utilizado (inicialmente) é o CS-1. Nesse caso, o modelo migra sucessivamente para um cenário maior (isto é, com maior taxa de dados e menos codificação) até conseguir o mais adequado, o CS-4. Dessa forma, em uma situação em que os recursos seriam subutilizados com uma vazão de 3,88 kbps, utilizando o modelo adaptativo a vazão atinge 19,31 kbps. O outro extremo, com CS-4 como inicial em um ambiente com péssimas condições (2 dB), pode ser visualizado na Figura 5.11d. Nesse caso, o modelo migra sucessivamente para um CS menor, que ofereça maior proteção aos dados, chegando ao CS-1. O modelo permite que a MS, que antes não conseguia transmitir nenhum dado (vazão nula), consiga uma vazão de 0,14 kbps. O aumento não é significativo devido ao tempo gas-

to pelo modelo para migrar entre os CSs. Enquanto no primeiro caso o modelo migrou para o CS-4 em 6 segundos, nesse caso o modelo gastou 36 segundos para migrar para o CS-1 (mais de 30% do tempo de simulação). Esse grande intervalo de tempo ocorre devido à quantidade de pacotes transmitidos. O modelo adaptativo é acionado sempre que a camada RLC recebe um pacote da camada superior (LLC), assim, a quantidade e frequência com que os pacotes são enviados, entre outros fatores, determina a velocidade com que o modelo migra entre os cenários de codificação.

A Figura 5.12 ilustra os pacotes (segmentos TCP) transmitidos nos dois casos referidos acima (CS-1 para 30 dB e CS-2 para 2 dB). No caso do CS-1 como cenário inicial em um C/I de 30 dB (Figura 5.12a), o transmissor envia pacotes (normalmente três) com grande frequência, acionando o modelo adaptativo (sempre que 64 blocos são transmitidos) e mudando o CS (quando necessário). No segundo caso (CS-4 como inicial em um C/I de 2 dB), visualizado na Figura 5.12b, o transmissor envia os dois primeiros pacotes (quando o cenário muda para o CS-3) e aguarda o ACK. Por serem péssimas as condições do ambiente, a MS não recebe praticamente nenhum bloco, logo, não envia um ACK. Assim, o pacote é retransmitido aos 15 e aos 35 segundos, acionando o modelo e migrando para o CS-2 e o CS-1, respectivamente.



**Figura 5.12** Pacotes Transmitidos Utilizando o Modelo Adaptativo

### Parâmetros de transição

Como uma tentativa de tornar o modelo adaptativo mais preciso (isto é, ajustá-lo de forma que ele migre sempre para o CS mais apropriado), novos valores para os parâmetros de transição foram testados. A Figura 5.13 ilustra os novos valores utilizados. Estes foram obtidos a partir da observação do percentual de blocos descartados próximo aos pontos (C/Is) em que um cenário torna-se melhor que o anterior. Tais pontos são as extremidades dos intervalos em que cada CS alcança a maior vazão (Figura 5.2a). Por exemplo, o CS-2 torna-se mais apropriado que o CS-1 aos 9dB. Então, foram coletados os percentuais de blocos descartados para 8 e 10 dB, utilizando CS-1 e CS-2. Para o CS-1, são descartados em média 8% dos blocos transmitidos aos 8 dB e 5% aos 10dB. Assim, se estando no CS-1, o transmissor perder menos de 7% dos blocos transmitidos, o CS-2 pode ser utilizado. Da mesma forma, considerando que, para o CS-2 são perdidos em média 18% dos blocos aos 8 dB e 11% aos 10 dB, deve-se migrar para o CS-1 se o percentual de blocos perdidos for maior 12%.

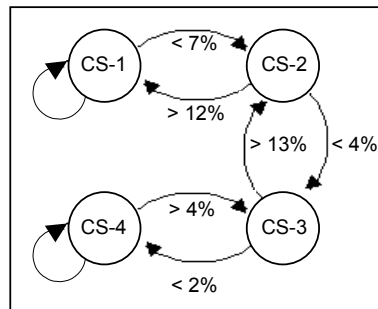
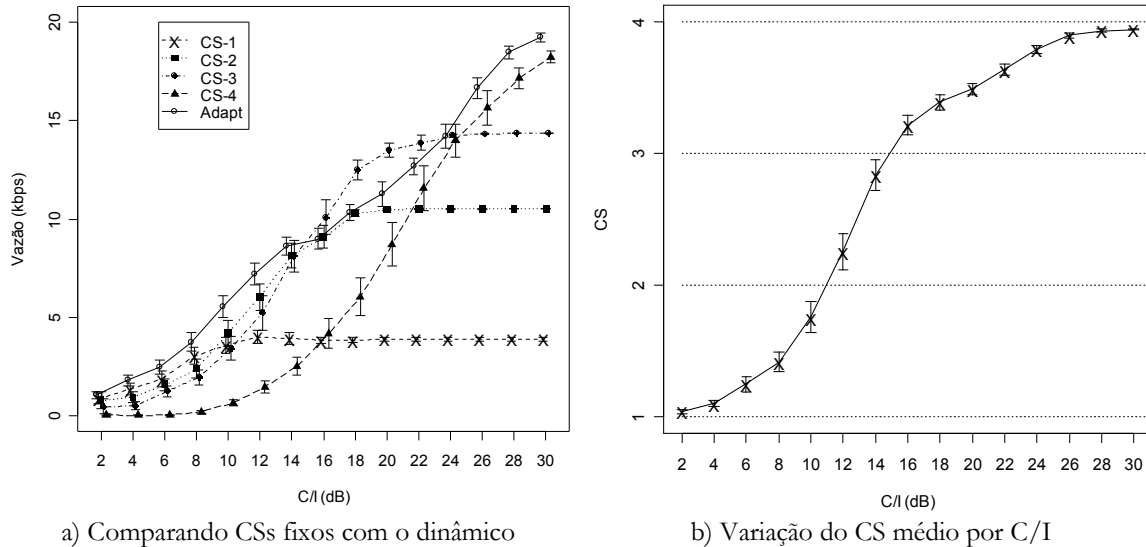


Figura 5.13 Novos Valores para os Parâmetros de Transição entre os CSs

Na Figura 5.14 podem ser observados os resultados obtidos com os novos parâmetros. Comparando o gráfico da Figura 5.14a com o da Figura 5.10b, percebe-se que houve uma melhora nos intervalos em que o CS-2 e o CS-3 são mais apropriados (9 a 14 dB e 15 a 24 dB, respectivamente). Pode-se observar que no intervalo de 10 a 18 dB houve uma melhora na vazão, que, na maioria dos casos, ultrapassa a vazão máxima alcançada sem o modelo. Isso ocorre porque a tendência do modelo a migrar para CSs altos diminuiu, como pode ser observado comparando a Figura 5.14b com a Figura 5.11a. Antes, a partir de 16 dB o modelo já migrava para o CS-4. Com os novos valores, o modelo permanece no CS-3 até 20 dB, oscilando entre o CS-3 e o CS-4 para 22 e 24 dB.



**Figura 5.14** Resultados Utilizando os Novos Valores para os Parâmetros de Transição

Analisando ainda a Figura 5.14a, percebe-se que entre 20 e 28 dB a vazão diminuiu (quando comparada à Figura 5.10b). Isso mostra quão difícil é encontrar valores ótimos para os parâmetros de transição. Em [52], é realizado um estudo sobre um algoritmo de gerenciamento de fila denominado RED (*Random Early Detection*). Esse algoritmo possui dois parâmetros cujos valores são essenciais para seu bom funcionamento. O estudo mostra que valores estáticos para esses parâmetros não conseguem ser considerados ótimos para todos os casos, e sim para determinados tipos de cenários. Analogamente, é demasiadamente difícil encontrar valores ótimos para os parâmetros de transição do modelo adaptativo que sirvam para todos os cenários. Modificando o cenário, os valores que para o cenário atual são ótimos, tornam-se inadequados, precisando ser novamente ajustados.

## 5.2.4 Versões do TCP

O comportamento do TCP varia de acordo com os algoritmos do controle de congestionamento que implementa. Esses algoritmos são constantemente modificados para melhorarem o desempenho do TCP sobre as redes com as quais ele interage, gerando novas versões do TCP.

Os trabalhos relacionados (Seção 3.3.1) limitam-se a estudar o comportamento de apenas uma versão do TCP sobre redes GPRS, com exceção do [11], que analisa o SACK e o uso de *Timestamps*. Assim, esta dissertação realiza um estudo do desempenho de quatro ver-

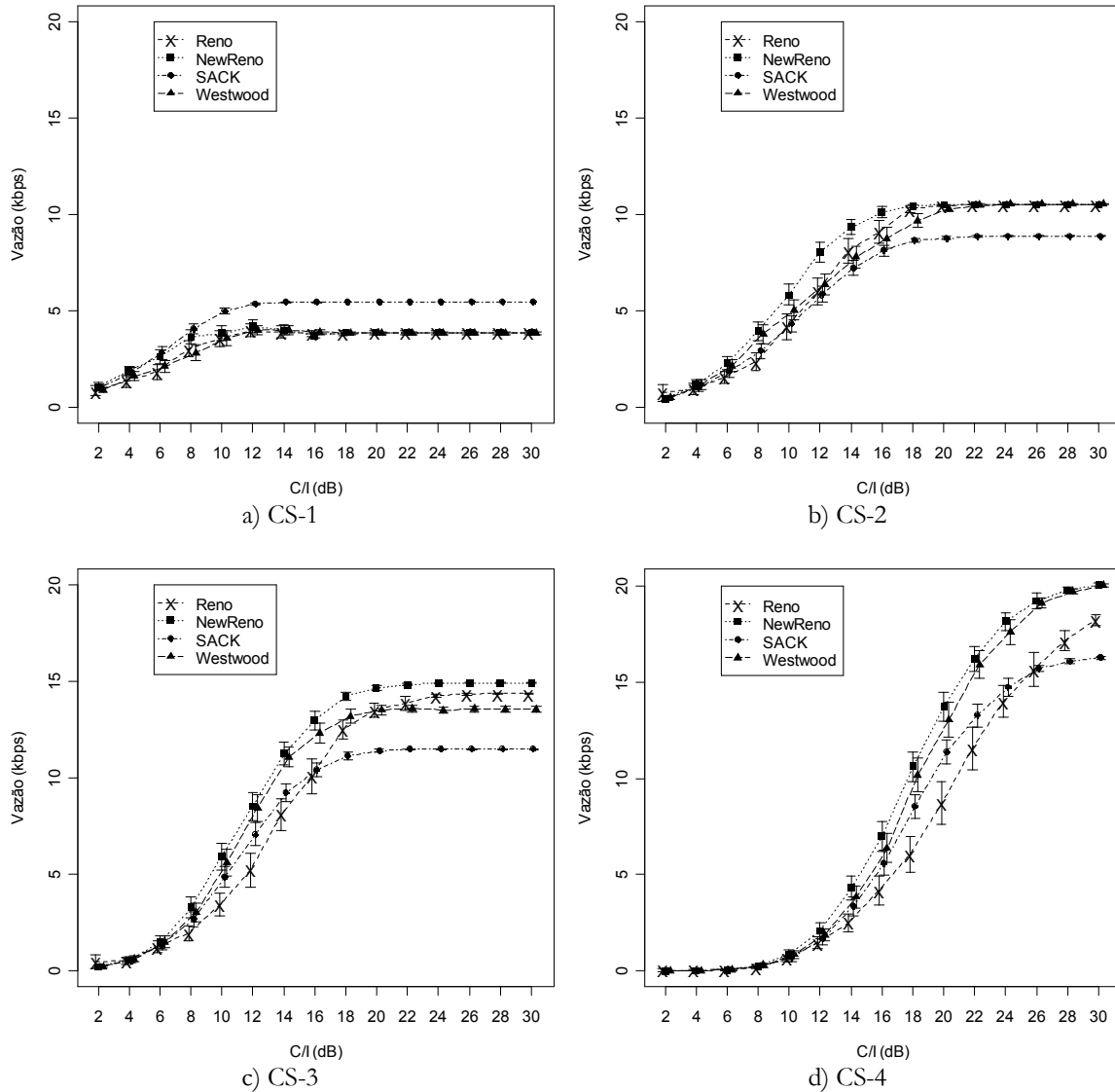


sões do TCP sobre o GPRS: Reno, New-Reno, SACK e Westwood. As versões Reno e New-Reno são estudadas por serem as mais utilizadas na Internet. SACK foi escolhido por apresentar um bom desempenho sobre redes sem fio em geral e, segundo [11], melhor performance em redes GPRS que o Reno. O TCP Westwood têm apresentado um ótimo desempenho em redes sem fio, sendo consideravelmente melhor do que outras versões do TCP, incluindo o TCP New-Reno, no qual o Westwood é baseado [36]. Contudo, nenhum estudo referente ao seu desempenho sobre redes GPRS foi ainda apresentado. Por isso, o interesse em estudá-lo nesse trabalho.

A Figura 5.15 ilustra a vazão alcançada por uma estação móvel, considerando as quatro versões estudadas, para os quatro cenários de codificação, utilizando a configuração 1+1 e pacotes (segmentos) de 1500 bytes.

Para o CS-1 (Figura 5.15a), pode-se notar que o SACK apresenta o melhor desempenho. As outras versões apresentam um desempenho semelhante. Para o CS-2 (Figura 5.15b) e o CS-3 (Figura 5.15c), Reno, New-Reno e Westwood continuam com desempenhos semelhantes, contudo, surpreendentemente, o SACK mostra uma vazão inferior a dessas versões. Observa-se que, utilizando CS-3, o New-Reno alcança uma taxa um pouco maior que o Westwood para condições boas de ambiente (acima de 20 dB). Para o CS-4, o New-Reno e o Westwood conseguem melhor desempenho que as versões SACK e Reno.

Esses resultados são surpreendentes. Esperava-se que o SACK apresentasse sempre um desempenho melhor que o TCP Reno (no qual é baseado), como afirma [11], uma vez que sempre realiza menos retransmissões. Os experimentos realizados não forneceram informações suficientes para explicar esses resultados, sendo necessárias novas investigações.



**Figura 5.15** Vazão do TCP Reno, New-Reno, SACK e Westwood para Todos os CSs

Também era esperado que o TCP Westwood apresentasse uma melhora significativa em relação às outras versões. Contudo, além de, na maioria dos casos, conseguir uma vazão bastante próxima da alcançada pelo New-Reno (versão na qual o TCPW é baseado), em alguns casos, como na Figura 5.15c (CS-3), sua vazão é menor que a do New-Reno. Um provável motivo é a baixa taxa de transmissão do GPRS. O algoritmo que atualiza o Ssthresh durante o *Fast Recovery* no Westwood é baseado na largura de banda disponível (estimada), como mostrado abaixo:

$$\text{Ssthresh} = \text{BWE} * \text{RTT}_{\min} / \text{Seg\_size}, \quad (1)$$

onde BWE é a largura de banda estimada,  $RTT_{min}$  é o valor mínimo do *round-trip time* e  $Seg\_size$  é o número de bits do segmento TCP. Quando esse valor calculado é menor que 2, Ssthresh recebe o valor 2. A janela de congestionamento (CWND) também é baseada nesse valor:

$$\begin{aligned} &\text{se forem recebidos ACKs duplicados,} \\ &\quad CWND = Ssthresh \end{aligned} \tag{2}$$

$$\begin{aligned} &\text{se expirar o } timeout, \\ &\quad CWND = 1 \end{aligned} \tag{3}$$

Considerando o melhor cenário utilizado neste estudo (CS-4 com configuração 4+1), a taxa de transmissão máxima possível para o GPRS é de 85,60 kbps (sendo a alcançada pelo simulador um pouco menor). E, de acordo com a nossa topologia (Figura 5.1), o  $RTT$  mínimo é de 80 ms (ida e volta em 4 *links* com atraso de 10 ms). Então, utilizando a fórmula (1), para o pacote de 1500 bytes, temos:

$$Ssthresh = 85,60 * 80 / (1500 * 8) = 0,57$$

Como o valor calculado é menor que 2, o valor de Ssthresh será 2. Assim, a janela de congestionamento não ultrapassará o valor 2 durante o *Fast Recovery*, limitando a taxa de transmissão. Para confirmar esses resultados, serão realizados novos estudos, com novos cenários.

### 5.3 Considerações Finais

Nesse 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 ao comportamento do TCP Reno sobre o GPRS foram expostos, através dos quais pôde-se concluir que tráfegos de retaguarda (*background*) entre sistemas finais localizados entre roteadores intermediários não influenciam de forma significativa no tráfego GPRS. Também 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, foi possível notar que quando o CS-1 é utilizado, o tamanho influencia na vazão, sendo esses valores inversamente proporcionais. O número de fontes (usuários) também foi variado, porém o tráfego gerado por uma estação móvel não prejudicou as outras estações, ou seja, os resultados obtidos foram praticamente os mesmos independente do número de MSs. Da mesma forma, o comportamento do TCP não muda significativamente ao aumentar o número de *timeslots* por usuário. Experimentos utilizando o modo sem confirmação da camada RLC demonstram que esse modo deve ser utilizado apenas em ambientes com condições muito boas e que o CS-4 deve ser evitado nesse modo, visto que não apresenta proteção aos dados.

Em seguida, foi realizada uma comparação entre o desempenho dos protocolos TCP e UDP. Os resultados evidenciam mais uma vez que o tamanho do pacote influencia no desempenho do TCP sobre GPRS. Para o tamanho padrão (536 bytes), o TCP apresentou uma vazão próxima da alcançada pelo UDP, enquanto com pacotes de 1500 bytes, a vazão atingida utilizando o CS-1 é menor que a metade da obtida pelo UDP.

Os resultados obtidos utilizando o modelo adaptativo para os cenários de codificação também foram expostos. Estes mostram que o modelo traz ganho na vazão, migrando para um CS mais adequado de acordo com as condições do ambiente (C/I).

Finalmente foram apresentados os resultados referentes à comparação de quatro versões do TCP: Reno, New-Reno, SACK e Westwood. Os resultados mostram de forma surpreendente que, para o CS-1, o SACK apresenta melhor desempenho que as outras versões; contudo, nos outros cenários sua vazão é inferior. O Westwood, que consegue um ganho de até 500% na vazão em redes sem fio segundo [36], apresentou vazões semelhantes ao New-Reno, sendo pior em alguns casos.

O capítulo seguinte apresentará as conclusões e os trabalhos futuros relacionados a este trabalho que visam um melhor entendimento sobre os resultados até então obtidos.

# 6

## Conclusão

Este trabalho teve como objetivo analisar o desempenho do TCP sobre o GPRS através de simulações, variando parâmetros referentes aos dois protocolos. Vários cenários foram avaliados, incluindo diferentes taxas de erros (C/I) no meio físico de transmissão, tamanhos de pacotes e quantidades de usuários e de *timeslots* utilizados por usuário no sentido *downlink*.

A seguir, são apresentadas as principais contribuições e os possíveis trabalhos futuros.

### 6.1 Contribuições

São três as principais contribuições deste trabalho:

#### **Estudo abrangente do comportamento do TCP sobre o GPRS**

A primeira contribuição deste trabalho consiste no estudo abrangente realizado sobre o comportamento do TCP interagindo com o GPRS. Alguns estudos vêm sendo realizados nessa área, mas cada um avalia apenas poucos aspectos possíveis em um cenário que suporta TCP com GPRS. Este trabalho estudou os seguintes aspectos:

- o impacto de tráfego de retaguarda no tráfego GPRS: foram inseridos tráfegos de retaguarda (*background*) entre a rede GPRS e o transmissor 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 TCP para cada cenário de codificação da camada RLC (GPRS) e, através dessa 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 segmento TCP no seu desempenho sobre o GPRS: variando o tamanho do pacote (segmento) TCP, pôde-se verificar que este influencia na vazão alcançada (quando o CS-1 é utilizado), sendo esta inversamente proporcional ao tamanho do segmento. Isso ocorre devido à limitação imposta pelo tamanho da janela de transmissão da camada RLC (Seção 5.2.1). Considerando que ainda não havia sido relatado esse problema relacionado ao tamanho do segmento TCP, vale ressaltar que a identificação de tal problema também é uma contribuição deste trabalho;
- o efeito causado sobre cada usuário pela quantidade de usuários no sistema GPRS: o número de usuários no sistema GPRS foi variado a fim de detectar o impacto causado ao incrementar usuários ao sistema. Os resultados obtidos foram praticamente os mesmos independente do número de MSs, mostrando que o tráfego gerado por uma estação móvel não prejudica as outras estações;
- a influência do número de *timeslots* por usuário no sentido *downlink*: foram analisadas três configurações *multislot* (1+1, 2+1 e 4+1). O comportamento do TCP permaneceu praticamente o mesmo para as três configurações, diferenciando apenas na vazão alcançada que dobrou à medida que o número de *timeslots* duplicava;
- o comportamento do TCP sobre o modo sem confirmação da camada RLC: o modo sem confirmação da camada RLC (não estudado em conjunto com o TCP até então) foi utilizado nos estudos e pôde-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 do 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 536 bytes, o TCP apresentou uma vazão próxima da alcançada pelo UDP, enquanto com segmentos de 1500 bytes, a vazão atingida utilizando o CS-1 é menor que a metade da obtida pelo UDP. Esses resultados comprovam a influência do tamanho do segmento TCP no seu desempenho sobre o GPRS;
- o desempenho de quatro versões do TCP: as versões Reno, New-Reno, SACK e Westwood foram avaliadas interagindo com o GPRS. Os resultados foram inesperados em se tratando do SACK e do Westwood. Apesar de apresentar melhor desempenho que

as outras versões para o CS-1, o SACK apresenta a menor vazão nos outros cenários. O Westwood, que consegue um ganho de até 500% na vazão em redes sem fio segundo [36], apresentou vazões semelhantes ao New-Reno, sendo pior em alguns casos.

### **Módulo mais estável do GPRS para o ns-2**

Outra importante contribuição foi o aprimoramento do módulo GPRS para o ns-2. Após várias alterações, esse módulo apresenta-se agora mais estável e de acordo com a especificação, podendo ser facilmente estendido e utilizado pelo vasto conjunto de usuários do ns-2. Entre as principais modificações (Seção 4.3), ressaltam-se o modelo de erros, que utiliza uma tabela BLER bastante adotada no meio acadêmico, e o modelo adaptativo para cenários de codificação, que migra entre os CSs de acordo com a percentagem de blocos RLC descartados por janela de transmissão.

### **Estudo do modelo adaptativo com o TCP**

O estudo do comportamento do TCP utilizando o modelo adaptativo foi também uma importante contribuição. No trabalho em que o modelo é proposto [51], foi analisado apenas o comportamento do modelo na rede GPRS. O presente trabalho analisou o modelo em um ambiente fim a fim, utilizando o TCP e avaliando seu desempenho ao variar tamanho do segmento TCP, número de usuários no sistema GPRS, número de *timeslots* por usuário no sentido *downlink* (configuração *multislot*), modo da camada RLC (com e sem confirmação) e versão do TCP. Em todos os cenários o modelo apresentou ganho na vazão, migrando para um CS mais adequado de acordo com as condições do ambiente (C/I). Também pôde-se perceber a enorme dificuldade em se encontrar os valores ótimos para os parâmetros de transição entre os CSs, o que, algumas vezes causou a migração do modelo para um CS mais apropriado que o corrente, contudo, não o CS mais apropriado de todos.

## **6.2 Trabalhos Futuros**

Há duas linhas principais de trabalhos futuros que podem ser realizados. A primeira refere-se à extensão do módulo GPRS e a segunda diz respeito à continuação dos estudos sobre a interação do TCP com o GPRS.

Com relação ao módulo, são sugeridos os seguintes trabalhos futuros:

- Incluir gerenciamento de mobilidade e gerenciamento de sessão, implementando os procedimentos *attach* e *detach* e permitindo que o usuário tenha contexto PDP e mude de estado (*idle*, *ready* e *standby*);
- Permitir mobilidade aos usuários, adicionando novas células e implementando procedimentos de gerenciamento de mobilidade;
- Estender o GPRS para o EDGE (próximo passo na evolução dos sistemas celulares), adicionando novos cenários de codificação, o que permitiria taxa de transmissão bem mais altas (de até 473,6 kbps, teoricamente);
- Modificar o modelo adaptativo para que este migre de forma direta entre CSs não vizinhos, permitindo uma mudança mais rápida para o cenário adequado;
- Permitir que um usuário seja classe A, isto é, que ele possa utilizar serviços GPRS e GSM simultaneamente.

Continuando os estudos sobre a interação entre o TCP e o GPRS, podem ser realizados os trabalhos sugeridos abaixo:

- Estudar de forma mais aprofundada o comportamento das versões SACK e Westwood, coletando novas métricas e utilizando diferentes cenários, para verificar o motivo do baixo desempenho inesperado;
- Analisar o modo com confirmação da camada LLC, estudando a relação entre o uso deste modo nessa camada e na RLC (isto é, verificando quando o uso desse modo em uma camada torna desnecessário o uso na outra; e em que condições é mais apropriado usá-lo em cada camada);
- Estudar uma possível mudança no tamanho da janela de transmissão da camada RLC, a fim de que o segmento de 1500 bytes, mais utilizado na Internet, obtenha bom desempenho;



- Avaliar a combinação entre o uso do modo com confirmação e o CS utilizado na camada RLC, estudando quando a presença de um torna o outro redundante e quando o uso de cada um é mais adequado;
- Estudar a variação do C/I durante a simulação (nos estudos apresentados o C/I permanece o mesmo durante todo o tempo de simulação), avaliando o desempenho do modelo adaptativo com a mudança das condições do ambiente;
- Estudar o comportamento do TCP sobre o EDGE, analisando se as altas taxas de transmissão influenciarão no seu desempenho (em especial no Westwood, tendo em vista a probabilidade da baixa taxa de transmissão do GPRS ter causado seu mau desempenho).

### 6.3 Considerações Finais

Este trabalho apresentou um estudo aprofundado do desempenho do TCP sobre o GPRS, utilizando simulações. Através da variação de vários parâmetros referentes aos dois protocolos, foi possível estudar bastantes cenários, analisando vários aspectos relacionados a essa interação.

Vários resultados foram apresentados e discutidos. Dentre eles, destacaram-se a influência do tamanho do segmento TCP no seu desempenho sobre o GPRS – devido ao tamanho da janela de transmissão da camada RLC – e o estudo do modelo adaptativo para cenários de codificação em diferentes cenários.

Para realizar o estudo apresentado foi necessário modificar o módulo GPRS do ns-2. Tal módulo apresentava vários problemas, que incluíam desde erros lógicos a incompatibilidades com a especificação. Atualmente o módulo encontra-se mais estável, de acordo com as normas e com características adicionais, como o modelo de erros utilizando a tabela BLER, o modelo adaptativo para cenários de codificação e a possibilidade de configuração *multislot*.

Tanto o módulo GPRS quanto o estudo da interação entre o TCP e o GPRS podem ser incrementados, tendo surgido possibilidades de trabalhos futuros.

# 7

## Referências


- [1] “Digital Cellular Telecommunications System (Phase 2+); General Packet Radio Service (GPRS); Service Description; Stage 2”, GSM 03.60 v 7.4.1, ETSI, out. 2000
- [2] M. Mouly, M.-B. Pautet, “The GSM System for Mobile Communications”, 1992
- [3] “GSM World - GSM Technology”, <http://www.gsmworld.com/technology/gsm.shtml>
- [4] “TIM Brasil”, <http://www.timbrasil.com.br/>
- [5] “Oi”, <http://www.oi.com.br/>
- [6] J. Postel, “Transmission Control Protocol”, RFC-793, USC/Information Sciences Institute, set. 1981.
- [7] S. Hoff, M. Meyer, A. Schieder, “A Performance Evaluation of Internet Access via the General Packet Radio Service of GSM”, In: Proc. IEEE 48th VTC'98, 1998. p.1760-1764.
- [8] S. Hoff, M. Meyer, J. Sachs, “Analysis of the General Packet Radio Service (GPRS) of GSM as access to the Internet”, In: Proc. IEEE ICUPC'98, 1998. p.415-419.
- [9] M. Meyer, “TCP performance over GPRS”, In: Proc. IEEE Wireless Communications and Networking Conference (WCNC), set. 1999.
- [10] R. Kalden, I. Meirick, M. Meyer, “Wireless Internet Access based on GPRS”, IEEE Personal Communications, 7(2), abr. 2000.
- [11] J. Rendón, F. Casadevall, J.L. Faner, “Wireline TCP Options Behavior in the GPRS Network”, In: Proc. IST Mobile Communications Summit, Sitges-Barcelona, Spain, set. 2001.
- [12] R. Chakravorty, J. Cartwright, I. Pratt, “Practical Experience with TCP over GPRS”, In: Proc. IEEE Global Communications Conference (IEEE GLOBECOM 2002), Taipei, Taiwan, nov. 2002.
- [13] R. Chakravorty, A. Clark, I. Pratt, “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. Disponível em: <http://www.cl.cam.ac.uk/users/rc277/gprsweb.ps>


- [14] A. Selian, "3G Mobile Licensing Policy: GSM Case Study", ITU, 2001. Disponível em: <http://www.itu.int/osg/spu/ni/3G/casestudies/GSM-FINAL.pdf>
- [15] T. Rappaport, "Wireless Communications: Principles & Practice", Prentice Hall PTR, New Jersey, 1996.
- [16] Tektronix, "Universal Mobile Telecommunications System (UMTS) Protocols and Testing". Disponível em: <http://www.iec.org/online/tutorials/umts/topic01.html>
- [17] "NTT DoCoMo: The 3G Pioneer",  
<http://www.nttdocomo.com/corebiz/foma/3g/index.html>
- [18] A. Furuskär, S. Mazur, F. Müller, H. Olofsson, "EDGE: Enhanced data rates for GSM and TDMA/136 evolution", IEEE personal Communications Magazine, jun. 1999. p.56-66.
- [19] A. Samukik, "UMTS Universal Mobile Telecommunications System: Development of Standards for the Third Generation", IEEE Transaction on Vehicular Technology, v.47, n.4, nov. 1998
- [20] J.-Z. Sun, J. Sauvola, D. Howie, "Features in Future: 4G Visions From a Technical Perspective", In: Proc. IEEE Global Communications Conference, San Antonio, TX, nov. 2001. p.3533 -3537.
- [21] Y. Miyashita, "NTT DoCoMo Successfully Completes 4G Mobile-Communications Experiment Including 100-Mbps Transmission", nG Japan, v.1, n.11, nov. 2002.
- [22] C. Bettstetter, H.-J. Vogel, J. Eberspacher. "GSM Phase 2+; General Packet Radio Service GPRS: Architecture, Protocols and Air Interface", IEEE Communications Surveys, 2(3), 1999.
- [23] N. Abramson, "The Aloha System – Another Alternative for Computer Communication", In: AFIPS Proc. Fall Joint Computer Conference, 1970. p.281-285.
- [24] Agilent, "Understanding General Packet Radio Service (GPRS)", Application Note 1377.
- [25] "Digital Cellular Telecommunications System (Phase 2+); Multiplexing and multiple access on the radio path", GSM 05.02 v 8.5.1, ETSI, nov. 2000.
- [26] "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", GSM 04.60 v 8.5.0, ETSI, out. 2000.
- [27] D. Marsh, "Direct Conversion Receivers Battle Superherts for GSM Design Wins", EDN Europe, out. 2000.
- [28] M. Allman, V. Paxson, W. Stevens, "TCP Congestion Control", RFC-2581, abr. 1999.


- [29] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", RFC-2001, jan. 1997
- [30] K. Fall, S. Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP, ACM Computer Communications Review", v.26, n.3, jul. 1996. p.5-21.
- [31] J. Hoe, "Improving the Start-up Behavior of a Congestion Control Scheme for TCP", In: Proc. ACM SIGCOMM, aug. 1996.
- [32] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, "TCP Selective Acknowledgement options", RFC-2018, out. 1996.
- [33] P. Karn, C. Partridge, "Improving RoundTrip Time Estimates in Reliable Transport Protocols", In: ACM SIGCOMM, ago. 1987. p. 2-7.
- [34] C. Casetti, M. Gerla, S. Mascolo, M.Y. Sanadidi, R. Wang, "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links", In: Proc. ACM Mobicom, jul. 2001.
- [35] R. Wang, M. Valla, M.Y. Sanadidi, B. Ng, M. Gerla, "Efficiency/Friendliness Tradeoffs in TCP Westwood", IEEE Symposium on Computers and Communications, Taormina, Italy, jul. 2002.
- [36] R. Wang, M. Valla, M.Y. Sanadidi, M. Gerla, "Adaptive Bandwidth Share Estimation in TCP Westwood", In: Proc. IEEE Globecom, nov. 2002.
- [37] G. Yang, R. Wang, F. Wang, M. Y. Sanadidi, M. Gerla, "TCP Westwood with Bulk Repeat for Heavy Loss Environments", UCLA Technical Report #020023, 2002.
- [38] A. Dell'Aera, L.A. Grieco, S. Mascolo, "Linux 2.4 Implementation of Westwood+ TCP with rate-halving: A Performance Evaluation over the Internet", Tech. Rep. #08/03/S, Dipartimento di Elettrotecnica ed Elettronica, Politecnico di Bari.
- [39] J. Redón, F. Casadevall, J. Carrasco, "Wireless TCP Proposals with Proxy Servers in the GPRS Network", In: Proc. PIMRC, 2002.
- [40] N. Brownlee, "Traffic Flow Measurement: Experiences with NeTraMet", RFC-2123, mar. 1997.
- [41] A. Bakre, B. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts". In: Proc. 15th International Conf. on Distributed Computing Systems (ICDCS), mai. 1995.
- [42] H. Balakrishnan, S. Seshan, R. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks", ACM Wireless Networks, 1(4), dez. 1995.
- [43] H. Balakrishnan, V. Padmanabhan, S. Seshan, R. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links", In: Proc. ACM SIGCOMM, aug. 1996.

- [44] R. Caceres, L. Iftode, “Improving the Performance of Reliable Transport Protocols in Mobile Computing Environment”, IEEE JSAC, jun. 1995.
- [45] “ns-2 The Network Simulator”, <http://www.isi.edu/nsnam/ns>
- [46] “The Network Simulator: Contributed Code”,  
<http://www.isi.edu/nsnam/ns/ns-contributed.html>
- [47] R. Jain, “GPRS Simulations Using ns-Network Simulator”, Dissertação de Mestrado, Department of Electrical Engineering, Indian Institute of Technology, Bombay, jun. 2001.
- [48] “Extensions to the ns Network Simulator”,  
<http://www.icsi.berkeley.edu/~widmer/mnav/ns-extension/>
- [49] M. Adams, M. Meyer, J. Sachs, “Quality of Service Evaluation for Mobile Internet Access via the General Packet Radio Service (GPRS)”, In: Proc. 5<sup>th</sup> Intl. Workshop on Mobile Multimedia Communication MoMuc, Berlin, out. 1998.
- [50] S. Hoff, M. Meyer, A. Schieder, “A Performance Evaluation of Internet Access via the General Packet Radio Service of GSM”, In: Proc. IEEE VTC, Canadá, mai. 1998.
- [51] S. Dantas, “Avaliação de Serviços de Dados em Redes GPRS”, Dissertação de Mestrado, Centro de Informática, UFPE, Recife, 2001.
- [52] M. May, J. Bolot, C. Diot, and B. Lyles, “Reasons not to deploy RED”, In: Proc. IW-QoS '99, mar. 1999. pp.260-262.

Dissertação de Mestrado apresentada por **Jeisa Pereira de Oliveira** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título, "Análise de Desempenho de TCP sobre GPRS em um Ambiente P2P", orientada pela **Profa. Judith Kelner** e aprovada pela Banca Examinadora formada pelos professores:

  
\_\_\_\_\_  
Prof. Nelson Souto Rosa  
Centro de Informática / UFPE

  
\_\_\_\_\_  
Profa. Rossaria Maria de Castro Andrade  
Departamento de Computação / UFC

  
\_\_\_\_\_  
Profa. Judith Kelner  
Centro de Informática / UFPE

Visto e permitida a impressão.  
Recife, 6 de fevereiro de 2004.

  
\_\_\_\_\_  
**Prof. Jaelson Freire Brelaz de Castro**  
Coordenador da Pós-Graduação em Ciência da Computação do  
Centro de Informática da Universidade Federal de Pernambuco.