

INSTITUTO MILITAR DE ENGENHARIA

MÁRCIO BARROSO TOSCANO DANTAS

**MELHORIA DE DESEMPENHO DO PROTOCOLO TCP EM
CANAIS DE HF VIA ESCOLHA DE PARÂMETROS E
EMPREGO DE TÉCNICAS DE CONTROLE DE ERROS**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Engenharia Elétrica do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Ciências em Engenharia Elétrica.

Orientador: Prof. Juraci Ferreira Galdino - D. C.

Co-orientador: Prof. Ernesto Leite Pinto - D. C.

Rio de Janeiro

2006

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

c2006

INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80-Praia Vermelha
Rio de Janeiro-RJ CEP 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmear ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

D192m Dantas, Márcio Barroso Toscano

Melhoria de desempenho do protocolo TCP em canais de HF via escolha de parâmetros e emprego de técnicas de controle de erros / Márcio Barroso Toscano Dantas. - Rio de Janeiro : Instituto Militar de Engenharia, 2006.
128 p.: il, graf., tab.

Dissertação (mestrado) - Instituto Militar de Engenharia- Rio de Janeiro, 2006.

1. Comunicações Digitais. 2. Protocolo TCP. 3. Canal HF. I. Contribuições à avaliação de desempenho do TCP em canais de HF. II. Instituto Militar de Engenharia.

CDD 384.5

INSTITUTO MILITAR DE ENGENHARIA

MÁRCIO BARROSO TOSCANO DANTAS

**MELHORIA DE DESEMPENHO DO PROTOCOLO TCP EM CANAIS
DE HF VIA ESCOLHA DE PARÂMETROS E EMPREGO DE TÉCNICAS
DE CONTROLE DE ERROS**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Engenharia Elétrica do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Ciências em Engenharia Elétrica.

Orientador: Prof. Juraci Ferreira Galdino - D. C.

Co-orientador: Prof. Ernesto Leite Pinto - D. C.

Aprovada em 3 de fevereiro de 2006 pela seguinte Banca Examinadora:

Prof. Juraci Ferreira Galdino - D. C. do IME - Presidente

Prof. Ernesto Leite Pinto - D. C. do IME

Prof. Marco Antonio Grivet Mattoso Maia - PhD da PUC

Prof. Ronaldo Moreira Salles - PhD do IME

Prof. Márcio Nascimento Bispo - M. C. do IME

Rio de Janeiro
2006

Dedico este trabalho a Deus, nosso Mestre maior e à Roberta, pela cumplicidade, pelo apoio e incentivo em todos os momentos.

AGRADECIMENTOS

A Deus pela oportunidade de evoluir profissionalmente.

A minha amada esposa Roberta, pelo apoio, incentivo e compreensão, estando sempre ao meu lado nos momentos mais difíceis.

Aos familiares, pela força e carinho depositados.

Ao meu orientador, Maj Galdino, pelos conhecimentos transmitidos e paciência no esclarecimento de todas as dúvidas, pela dedicação e confiança depositados e por todo trabalho de orientação envolvendo ética, dedicação, qualidade e amizade.

Ao meu Co-orientador, Professor Ernesto, pela amizade conquistada, pela dedicação, ensinamentos e paciência, que foram fundamentais no desenvolvimento deste trabalho.

Aos Professores Marco Antonio Grivet Mattoso Maia, Ronaldo Moreira Salles e Márcio Nascimento Bispo, integrantes da banca examinadora, pela colaboração e sugestões que enriqueceram de forma significativa este trabalho.

Aos companheiros do IME que compartilharam os momentos de luta. Em especial, ao amigo Arthur Henrique pela amizade e ajuda prestada no momento necessário.

Ao Exército Brasileiro pela oportunidade de realizar este curso.

Por fim, a todos os professores, alunos e funcionários do Departamento de Engenharia Elétrica do IME, que de alguma forma contribuíram na realização deste trabalho.

SUMÁRIO

LISTA DE ILUSTRAÇÕES	11
LISTA DE TABELAS	14
LISTA DE ABREVIATURAS E SÍMBOLOS	15
1 INTRODUÇÃO	19
1.1 Motivação	19
1.2 Objetivos	21
1.3 Revisão Bibliográfica	21
1.4 Organização da dissertação	23
2 PROTOCOLOS DE COMUNICAÇÃO	24
2.1 Introdução	24
2.2 Arquitetura TCP/IP	24
2.3 Protocolo de Aplicação: FTP	27
2.4 Protocolo TCP	28
2.4.1 Formato do Segmento TCP	29
2.4.2 Abertura e fechamento de conexão	30
2.4.3 Controle de erros e/ou perdas de segmento	31
2.4.4 Controle de fluxo	38
2.4.5 Controle de congestionamento	39
2.5 Protocolo IP	42
2.6 Protocolos de enlace	43
2.7 Resumo	43
3 VAZÃO DO PROTOCOLO TCP EM REDES SEM FIO E ES- TRATÉGIAS PARA SUA MELHORIA	45
3.1 Introdução	45
3.2 Baixa vazão do protocolo TCP em redes sem fio	45
3.3 Estratégias de Melhoria da vazão do protocolos TCP em redes sem fio	46
3.3.1 Modificação e inclusão de novos algoritmos no protocolo TCP	47
3.3.1.1 Estratégias que atuem em resposta à fonte de perda de pacote	47

3.3.1.2	Desenvolvimento de novas versões do protocolo TCP	50
3.3.1.3	Otimização dos parâmetros do protocolo TCP	56
3.3.2	Estratégias que modificam a semântica do TCP	57
3.3.2.1	<i>TCP Spoofing</i>	57
3.3.2.2	<i>Splitting TCP Connections</i>	58
3.3.3	Soluções focadas nos Protocolos da camada de enlace	58
3.3.3.1	Repetição Seletiva	59
3.3.3.2	Persistência do protocolo	60
3.3.3.3	Uso da Correção de Erros - FEC	61
3.3.3.4	Protocolos de enlace para HF	61
3.4	Resumo	66
4	CARACTERÍSTICAS DAS COMUNICAÇÕES EM CANAL HF ..	67
4.1	Introdução	67
4.2	Propagação em HF	67
4.3	Transmissão em HF	69
4.4	Modelagem do processo de erros de bits em HF	70
4.4.1	Erros correlacionados: Modelo HMM	72
4.4.2	Modelo de erros independentes	74
4.5	Resumo	75
5	AMBIENTE DE SIMULAÇÃO	76
5.1	Introdução	76
5.2	Ferramenta de simulação: <i>Network Simulator</i>	77
5.3	Topologia utilizada nas simulações	78
5.4	Cenário de simulação carregado no <i>ns-2</i>	78
5.4.1	Considerações sobre o cenário de simulação	78
5.5	Simulação do processo de erros no canal HF	80
5.5.1	Geração do processo de erros correlacionados de bits a partir do modelo HMM	80
5.5.2	Simulação no <i>ns-2</i> do Modelo de erros correlacionados	82
5.5.3	Simulação no <i>ns-2</i> do Modelo de erros independentes	83
5.6	Simulação do protocolo de enlace com mecanismo ARQ-SR	84
5.7	Modelo de Simulação do mecanismo FEC	86
5.8	Medidas de desempenho	88

5.9	Resumo	89
6	RESULTADOS	90
6.1	Comparação de algumas das estratégias de melhoria da vazão do protocolo TCP em canal HF	92
6.2	Modelagem de erros em canal HF e sua influência no desempenho do TCP ..	96
6.2.1	Modelo HMM: Taxa de perda de pacotes vs Tamanho do pacote	96
6.2.2	Influência do modelo de erros no desempenho do protocolo TCP	97
6.3	Desempenho do TCP em canal HF	102
6.3.1	Vazão das versões do protocolo TCP em canal HF	102
6.3.2	Influência do tamanho do arquivo a ser transferido no desempenho do protocolo TCP	105
6.4	Desempenho das versões do protocolo TCP em presença de protocolo de enlace com FEC	108
6.5	Desempenho do TCP em presença de protocolo de enlace com ARQ-SR em canal HF	109
6.5.1	Otimização dos parâmetros do protocolo TCP	110
6.5.1.1	Influência do <i>timeout</i> mínimo	110
6.5.1.2	Influência do tamanho da janela de congestionamento inicial	112
6.5.2	Otimização dos parâmetros do protocolo de enlace com ARQ-SR em canal HF	113
6.5.2.1	Influência do tamanho do quadro	113
6.5.2.2	Influência da persistência (δ)	115
6.5.2.3	Influência da temporizador NACK (<i>Retransmit Timer</i>)	115
6.5.2.4	Influência do tamanho do cabeçalho do quadro no desempenho dos protocolos em canal HF	116
6.5.3	Vazão do protocolo TCP em presença do protocolo de enlace com ARQ-SR com parâmetros criteriosamente escolhidos	117
6.6	Resumo	118
7	CONCLUSÕES E TRABALHOS FUTUROS	120
7.1	Principais conclusões	120
7.2	Trabalhos Futuros	123

8	REFERÊNCIAS BIBLIOGRÁFICAS	125
---	----------------------------------	-----

LISTA DE ILUSTRAÇÕES

FIG.2.1	center	25
FIG.2.2	Formação das estruturas de dados de cada camada.	26
FIG.2.3	Princípio de funcionamento do protocolo FTP.	28
FIG.2.4	Formato do segmento TCP.	30
FIG.2.5	Procedimentos de abertura e fechamento de conexão.	31
FIG.2.6	Estratégia de reconhecimento <i>Stop-and-Wait</i>	32
FIG.2.7	Exemplo de três pacotes transmitidos usando a Janela Deslizante.	33
FIG.2.8	center	33
FIG.2.9	center	35
FIG.2.10	Exemplo de perda de um pacote e a geração de reconhecimentos duplicados.	36
FIG.2.11	center	41
FIG.3.1	<i>knee point</i> da curva de taxa de transmissão versus vazão	48
FIG.3.2	Exemplo de reconhecimento parcial	51
FIG.3.3	Técnica <i>Spoofing</i>	58
FIG.3.4	Técnica de <i>Splitting</i>	59
FIG.3.5	Diagrama em blocos simplificado de um Codificador de canal Binário.	61
FIG.3.6	Formato do quadro AX.25	64
FIG.3.7	Estrutura do campo PID do quadro AX.25.	65
FIG.4.1	Propagação da onda eletromagnética em HF	68
FIG.4.2	Camadas da ionosfera	69
FIG.4.3	Trajetórias da onda eletromagnética na ionosfera	70
FIG.4.4	Diagrama de estados do Modelo HMM	73
FIG.5.1	Topologia utilizada nas simulações	78
FIG.5.2	Cenário de simulação carregado no <i>ns-2</i>	79
FIG.5.3	Diagrama de estados do Modelo HMM	81
FIG.5.4	Diagrama em blocos do ambiente de simulação usado na geração do modelo HMM e sua inclusão no <i>ns-2</i>	82

FIG.5.5	Diagrama em blocos do ambiente de simulação usado para geração do processo de erros de bits independentes e sua inclusão no <i>ns-2</i>	83
FIG.5.6	Mecanismo ARQ-SR do protocolo de enlace implementado.	84
FIG.5.7	Uso dos <i>Idle frames</i> na detecção do quadro perdido.	85
FIG.5.8	Diagrama em blocos do codificador FEC especificado em MIL-STD-188-110B.	88
FIG.6.1	Comparação das estratégias de melhoria do TCP em canal HF empregando os parâmetros da TAB. 6.1.	93
FIG.6.2	Comparação das estratégias de melhoria do TCP em canal HF empregando os parâmetros da TAB. 6.2.	94
FIG.6.3	Tamanho de <i>cwnd</i> do TCP sem protocolo de enlace.	96
FIG.6.4	Tamanho de <i>cwnd</i> do TCP com protocolo de enlace.	97
FIG.6.5	Taxa de perda de pacotes em função do tamanho do pacote IP com camada física modelado por HMM.	98
FIG.6.6	Tamanho do surto de erros das sequências geradas segundo o modelo HMM com o conjunto de parâmetros 1	99
FIG.6.7	Tamanho do surto de erros das sequências geradas segundo o modelo HMM com o conjunto de parâmetros 2	99
FIG.6.8	Influência das probabilidades de erros de cada estado do Modelo HMM no desempenho dos protocolos TCP e de camada de enlace em canal modelados com P_{alvo} igual a 10^{-3}	100
FIG.6.9	Desempenho do protocolo TCP em presença de erros correlacionados e erros estatisticamente independentes em canal modelado com P_{alvo} igual a 10^{-3}	101
FIG.6.10	Desempenho do protocolo TCP em presença de erros correlacionados e erros estatisticamente independentes modelados em canal modelado com P_{alvo} igual a 10^{-4}	102
FIG.6.11	Desempenho do protocolo TCP em presença de erros correlacionados e erros estatisticamente independentes em canal modelado com P_{alvo} igual a 10^{-5}	103
FIG.6.12	Desempenho das versões do TCP sem a presença de protocolo de enlace.	104

FIG.6.13	Vazão do TCP em canal HF com P_{alvo} de 10^{-3} em função do tamanho do pacote IP, considerando a transferência de arquivos de 10000 e 100000 bytes.	105
FIG.6.14	Vazão do TCP em canal HF com P_{alvo} de 10^{-4} em função do tamanho do pacote IP, considerando a transferência de arquivos de 10000 e 100000 bytes.	106
FIG.6.15	Evolução de <i>cwnd</i> do TCP em função do tempo em canal modelado com P_{alvo} igual a 10^{-3} submetido a transferência de arquivo de tamanho de 10000 bytes usando FTP.	107
FIG.6.16	Evolução de <i>cwnd</i> do TCP em função do tempo em canal HF modelado com P_{alvo} igual a 10^{-3} submetido a transferência de arquivo de tamanho de 100000 bytes usando FTP.	108
FIG.6.17	Evolução de <i>cwnd</i> do TCP em função do tempo em canal modelado com P_{alvo} igual a 10^{-4} submetido a transferência de arquivo de tamanho de 10000 bytes via FTP.	109
FIG.6.18	Evolução de <i>cwnd</i> do TCP em função do tempo em canal modelado com P_{alvo} igual a 10^{-4} submetido a transferência de arquivo de tamanho de 100000 bytes via FTP.	110
FIG.6.19	Desempenho das versões do TCP em presença do protocolo de enlace com FEC Taxa 1/2.	111
FIG.6.20	Influência do <i>timeout</i> mínimo no desempenho do protocolo TCP em canal HF.	112
FIG.6.21	Influência da janela de congestionamento inicial do TCP na vazão.	113
FIG.6.22	Influência do tamanho do quadro na vazão do protocolo TCP em canal HF.	114
FIG.6.23	Influência da persistência do protocolo de enlace na vazão do protocolo TCP em canal HF.	115
FIG.6.24	Influência do valor do temporizador NACK do protocolo de enlace no desempenho dos protocolos em canal HF.	116
FIG.6.25	Influência do tamanho do cabeçalho do quadro no desempenho dos protocolos em canal HF.	117
FIG.6.26	center	118

LISTA DE TABELAS

TAB.3.1	Parâmetros de forma de onda do protocolo HDL+.	63
TAB.5.1	Parâmetros do Modelo HMM.	81
TAB.5.2	Códigos corretores de erros para MODEMs HF com modo de operação em frequência fixa, segundo padrão MIL-STD-188-110B	86
TAB.5.3	Desempenho mínimo exigido para MODEMs seriais (MIL-STD-188-110B).	87
TAB.6.1	Parâmetros da simulação utilizados na comparação das estratégias do protocolo TCP em canal HF.	90
TAB.6.2	Parâmetros criteriosamente escolhidos via simulação computacional do protocolo de enlace com mecanismo ARQ-SR.	93
TAB.6.3	Conjuntos de parâmetros do Modelo HMM utilizados para modelar um canal com P_{alvo} igual a 10^{-3}	98
TAB.6.4	Tamanho médio de surto das seqüências geradas pelo modelo HMM com base no conjunto de parâmetros 1 e 2 nos critérios 3, 5 e 7	98
TAB.6.5	Vazão do protocolo TCP em presença protocolo de enlace com SR em função do tamanho do quadro do canal reverso	114

LISTA DE ABREVIATURAS E SÍMBOLOS

ABREVIATURAS

- iid* - independente e identicamente distribuído
- ns-2* - *Network Simulator* versão 2
- 2G* - Segunda Geração
- 3G* - Terceira Geração

SÍMBOLOS

- δ - Persistência do protocolo de camada de enlace
- M - Matriz transição de estados do modelo HMM
- p_i - Probabilidade de erros de bit do estado i da Cadeia de Markov escondida
- Δ_T - Retardo Total
- Δ_p - Retardo de propagação
- N - Tamanho do pacote IP ou quadro
- R - Taxa de transmissão
- \Rightarrow - se, então
- \Leftrightarrow - se, somente se

LISTA DE SIGLAS

ACK	<i>Acknowledgment</i>
ACS	<i>Automatic Channel Selection</i>
ALE	<i>Automatic Link Establishment</i>
ARQ	<i>Automatic Repeat Request</i>
ARQ-SR	<i>Automatic Repeat Request - Seletive Repeat</i>
BER	<i>Bit Error Rate</i>
BLOS	<i>Beyond Line of Sight</i>
FEC	<i>Foward Error Correction</i>
FTP	<i>File Transport Protocol</i>
GBN	<i>Go-Back-N</i>
HDL	<i>High-latency Data Link</i>
HDLC	<i>High-level Data Link Control</i>
HF	<i>High Frequency</i>
HMM	<i>Hidden Markov Model</i>
HMTP	<i>HF Mail Transport Protocol</i>
HTTP	<i>Hiper Text Transport Protocol</i>
ICMP	<i>Internet Control Message Protocol</i>
IP	<i>Internet Protocol</i>
LDL	<i>Low-latency Data Link</i>
MMHS	<i>Military Message Handling Systems</i>
MODEM	Modulador e Demodulador
NACK	<i>Not Acknowledgment</i>
PSK	<i>Phase Shift Keying</i>
REJ	<i>Reject</i>
RLP	<i>Radio Link Protocol</i>
RTT	<i>Round Trip Time</i>
SMTP	<i>Simple Mail Transport Protocol</i>
SREJ	<i>Seletive Reject</i>
STANAG	<i>Standardization Agreement</i>
TCP	<i>Transmission Control Protocol</i>
UDP	<i>User Datagram Protocol</i>

RESUMO

É amplamente conhecido que o protocolo TCP apresenta uma baixa vazão em redes sem fio. Nos últimos anos, com o advento dos novos sistemas de transmissão em HF com capacidade de integração com as redes convencionais baseadas na arquitetura TCP/IP, evidenciou-se a necessidade do desenvolvimento de pesquisas que têm por objetivo melhorar o desempenho do TCP neste cenário de comunicação.

Em recentes trabalhos foram propostas e investigadas alternativas para melhoria de desempenho do TCP no contexto das comunicações em HF, entre as quais destaca-se o uso de protocolos de camada de enlace dotados de mecanismos de correção de erro ARQ com repetição seletiva (ARQ-SR).

O objetivo principal do presente trabalho é avaliar, por meio de simulações computacionais, o desempenho do protocolo TCP em canal de HF típico, considerando o uso de estratégias que vêm sendo propostas para melhorar o desempenho desse protocolo em canais sem fio. Dentre essas estratégias, além do ajuste de parâmetros do supramencionado protocolo, são aqui também investigados o emprego de mecanismo de correção de erro ARQ-SR na camada de enlace e a estratégia de codificação FEC na camada física.

ABSTRACT

It is well known that the TCP protocol produces low information throughput in wireless networks. In the last years, the advent of new HF transmission systems potentially feasible for integration with conventional networks based on TCP/IP architecture has highlighted the necessity of researches focused on the investigation and improvement of the TCP performance in this communication scenario.

Some recent works have proposed and investigated alternatives for TCP performance improvement in the context of HF communications, among which the use of the link-layer protocols ARQ with selective repeating (SR) has been shown to provide significant advantages.

The main objective of the present work is to evaluate by computer simulation the TCP performance in typical HF links, and the improvements that can be obtained by employing a ARQ-SR protocol in the link layer and/or a FEC coding strategy in the physical layer. Besides, the influence of protocol parameters of different layers on the TCP performance is also addressed.

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Desde 1895, quando o físico italiano Guglielmo Marconi realizou a primeira transmissão de sinal de rádio através do oceano Atlântico, as comunicações em *High Frequency* (HF) têm se mostrado uma alternativa viável para o estabelecimento de um enlace de longa distância. Contudo, tendo em vista a limitação de largura de banda (3 KHz) inerente aos canais HF, a partir da década de 1970 as comunicações de longa distância por satélite tornaram-se o meio mais utilizado para transmissão de dados e vídeo em longa distância.

Com o surgimento de novos sistemas em HF que permitem a integração com as redes convencionais baseadas na arquitetura de protocolos *Transmission Control Protocol/Internet Protocol* (TCP/IP), o canal HF voltou a ser objeto de estudos que objetivam a melhoria deste tipo de integração. Neste contexto, as sub-redes HF constituem um alternativo ambiente a ser considerado, pois representam um meio de prover comunicações sem fio em enlaces de longa distância sem a presença de repetidores. Além disso, os sistemas em HF têm se mostrado uma alternativa de baixo custo e de menor vulnerabilidade em relação às redes satélite.

O uso de enlaces HF como parte integrante das redes convencionais TCP/IP torna-se justificável em face dos seguintes cenários:

- comunicação com plataformas móveis localizadas além da linha de visada;
- comunicações de dados envolvendo localizações remotas, onde a instalação da rede convencional usando meios confinados é inviável por questões técnicas ou econômicas;
- em caso de acidentes, bombardeios ou desastres naturais (as redes HF poderiam ser rapidamente instaladas, para substituir as redes danificadas nas áreas de emergência);
- durante uma operação militar com emprego de tropas, em ambientes hostis;

- inclusão na rede TCP/IP de usuários localizados em aviões, navios, barcos, etc., os quais encontram-se dentro dos requisitos de alcance, cobertura e propagação em HF.

O grande interesse pela utilização da pilha de protocolos TCP/IP como elo de interoperação de redes heterogêneas deve-se à sua padronização consolidada, à popularização da Internet, à disponibilidade de diversas soluções em tecnologia da informação, bem como à grande variedade de aplicações disponíveis (HTTP, SMTP, FTP, VoIP, Bancos de dados, etc.) para diversos ambientes corporativos, domésticos, acadêmicos, governamentais e militares, contribuindo para convergência de aplicações, infra-estrutura e gerência de redes.

A convergência em torno da arquitetura TCP/IP busca também, a integração de serviços para diferentes tipos de terminais, em diferentes ambientes sem fio.

Porém, o protocolo de transporte de dados, *Transmission Control Protocol* (TCP), responsável por noventa por cento do tráfego das aplicações na Internet (BARMAN, 2003), foi desenvolvido originalmente para redes cujas camadas físicas propiciam baixas taxas de erro e nas quais o principal distúrbio é o congestionamento. Assim sendo, em ambientes sem fio que são caracterizados por apresentarem elevadas taxas de erros da camada física, recentes pesquisas indicam que a vazão do protocolo TCP é, em geral, baixa.

O canal HF, sendo o ambiente sem fio a ser considerado, apresenta elevada taxa de erros, bem como erros em surtos, em função do fenômeno de desvanecimento por multipercurso inerente ao mecanismo de propagação produzido por refrações nas camadas ionosféricas. Além disso, as transmissões digitais em HF estão sujeitas a variações das condições de propagação em diversas escalas de tempo e, em geral, se caracterizam por altos retardos e pequena largura de banda (3KHz) contribuindo para as baixas taxas de bits obtidas ($\leq 10\text{Kbps}$). Diante das características do canal HF mencionadas, supõe-se que o desempenho TCP neste tipo de canal seja ainda mais pobre.

Diante do acima exposto, é de grande interesse realizar avaliações de desempenho de protocolos da arquitetura TCP/IP em canais HF e, principalmente, de estratégias que visam a obtenção de melhoria de vazão do TCP nestes ambientes de comunicação.

Dentre as estratégias de melhoria da vazão do TCP em canais sem fio propostas em recentes trabalhos, serão abordados no presente trabalho as seguintes: emprego de diversas versões do TCP recentemente propostas para canais sem fio, o aumento do

tamanho inicial da janela de congestionamento e do *timeout* mínimo do TCP, o uso dos protocolos de camada de enlace com mecanismos *Automatic Repeat Request* (ARQ) e a otimização de alguns dos seus parâmetros e o uso de camada física com *Forward Error Correction* (FEC) .

1.2 OBJETIVOS

A presente dissertação tem por objetivo o estudo e a análise de desempenho do protocolos TCP e de camada de enlace com ARQ em canais HF quando se emprega na camada de aplicação o protocolo FTP, bem como o estabelecimento de estratégias de camada de transporte, enlace e FEC na camada física a serem adotadas na busca da melhoria da vazão em face das dificuldades de entrega confiável de dados nesses ambientes de comunicação.

Especificamente esta dissertação possui os seguintes objetivos:

- Avaliar a influência dos modelos de erros correlacionados, tipicamente gerados por canais HF, no desempenho dos protocolos TCP e de camada de enlace com ARQ;
- Avaliar a influência do tamanho do arquivo a ser transferido via FTP no desempenho dos protocolos TCP em canal HF;
- Avaliar o desempenho de algumas versões do TCP em presença de camada física com FEC;
- Avaliar o desempenho dos protocolos TCP e de camada de enlace com ARQ, sob condição de variação de importantes parâmetros, tais como: *timeout* mínimo, tamanho inicial da janela de congestionamento, persistência, temporizador NAK e tamanho dos quadros.

1.3 REVISÃO BIBLIOGRÁFICA

Em virtude da importância das comunicações em HF nas operações militares, boa parte das pesquisas envolvendo o uso de HF tem caráter sigiloso, dificultando desta forma o acesso às publicações neste assunto.

Em relação à integração da arquitetura TCP/IP e sistemas em HF, na referência (JOHNSON, 2001) foi investigado o desempenho dos protocolos de enlace STANAG

5066, 3G HDL ARQ e 3G LDL ARQ em canais HF submetidos à aplicação SMTP/TCP, desconsiderando os efeitos de retransmissão do TCP, bem como os efeitos dos ajustes de *timeout* e tamanho do segmento TCP, e do protocolo ARQ. Além disso, quando não se conhece a priori as condições do canal HF, o protocolo 3G ARQ produziu melhores resultados de desempenho do que o STANAG 5066.

Em (JOHNSON, 2002) avaliou-se o impacto no desempenho de alguns modos de operação dos protocolos de enlace com mecanismo ARQ submetidos à aplicação SMTP. Neste artigo, com base nos protocolos de enlace STANAG 5066, 3G HDL ARQ e 3G LDL ARQ, os modos de operação avaliados foram: repetição seletiva, *stop-and-wait*, grau de precisão no sincronismo da transmissão dos quadros e o uso de técnicas de *Code combining* ou *Stand Alone*.

Com base em dados reais obtidos em transmissões em HF com protocolo de enlace STANAG 5066, no artigo (KALLGREN, 2001) foi apresentada uma discussão sucinta do desempenho do TCP usando a aplicação FTP, porém o mesmo não investigou a influência dos parâmetros e técnicas da camada de transporte e enlace com ARQ no desempenho.

Em (JODALEN, 2004) foram investigados o desempenho dos protocolos de enlace 5066 ARQ, HDL 3G e HDL+ submetidos à aplicação MMHS e UDP. Os resultados mostraram que para tamanhos de mensagens longas (100Kb) o protocolo HDL teve menor vazão que os protocolos 5066 e HDL+. Para tamanho médio ou pequeno de mensagem (10Kb-1 Kb), o protocolo HDL+ produziu maior vazão do que as outras.

No artigo (CHAMBERLAIN, 2003) foi avaliado o desempenho do protocolo de enlace HDL+ e comparou-se com o desempenho do protocolo HDL, para aplicações baseadas em UDP. Os resultados apresentados neste artigo indicam que o desempenho do protocolo HDL+ é melhor do que o do HDL. Tal fato se deve à capacidade de adaptação do HDL+ face a diferentes condições do canal.

Sobre as estratégias de melhoria do TCP em redes sem fio, a referência (ZORZI, 1999) sugere que a mais promissora estratégia deve ser focada nos protocolos da camada de enlace com ARQ utilizando a capacidade de adaptação de seus parâmetros em relação à camada física.

A avaliação da latência do TCP em presença de protocolo de enlace com mecanismos ARQ e camada física com FEC foi considerada por (ABDELMOUMEN, 2004), que mostrou que a latência do TCP sempre melhora com o aumento da persistência

do protocolo de enlace ARQ, exceto para os casos extremos onde o retardo é grande, os arquivos FTP são pequenos e a taxa de perda de pacotes é baixa.

1.4 ORGANIZAÇÃO DA DISSERTAÇÃO

O trabalho está dividido em 7 capítulos.

No Capítulo 2 serão apresentadas as características básicas da arquitetura TCP/IP. Posteriormente, será abordado o protocolo FTP adotado para as avaliações de desempenho dos protocolos TCP e de camada de enlace com ARQ em canal HF. Em seguida, será descrito o protocolo TCP que será objeto de avaliação neste trabalho. Por fim as características gerais dos protocolos IP e de camada de enlace serão apresentadas.

No Capítulo 3 enfoca-se o problema do baixo desempenho do protocolo TCP em canal sem fio. Também são apresentadas algumas das estratégias que visam a melhoria do TCP em ambientes sem fio propostas em recentes trabalhos.

No Capítulo 4 são abordados, inicialmente, algumas características da propagação de sinal em meios ionosféricos. Também discutiremos alguns modelos para descrever os erros que ocorrem nas camadas físicas de sistemas de comunicações que operam em canais de HF. Em seguida, serão descritos os modelos utilizados na presente dissertação para geração de um processo de erros de bits correlacionados e independentes, para fins de avaliação da influência do modelo de erros no desempenho dos protocolos TCP e de camada de enlace em canais HF.

No Capítulo 5 descreve-se o ambiente de simulação utilizado para as avaliações de desempenho dos protocolos TCP.

No capítulo 6 serão apresentados os resultados obtidos referente às avaliações de desempenho supra mencionadas.

No capítulo 7 são apresentadas as principais conclusões desta dissertação e algumas sugestões para realização de trabalhos futuros de forma a estender e complementar os resultados aqui obtidos.

2 PROTOCOLOS DE COMUNICAÇÃO

2.1 INTRODUÇÃO

Este capítulo aborda os principais conceitos básicos da arquitetura TCP/IP. Inicialmente, será apresentado o protocolo de aplicação FTP adotado para as avaliações de desempenho dos protocolos TCP e de camada de enlace com ARQ em canal HF. Em seguida, será descrito o protocolo TCP que será objeto de avaliação neste trabalho. E por fim serão apresentadas as características gerais dos protocolos IP e de camada de enlace.

2.2 ARQUITETURA TCP/IP

No final da década de 1960, o Departamento de Defesa Americano, por meio do projeto DARPA (*Defense Advanced Reserch Projects Agency*), visualizou a importância e o potencial da tecnologia de interconexão em redes (*internetworking*) para tornar as comunicações de dados redundantes e mais robustas, de forma a responder dinamicamente a falhas e a outras situações anormais. Por essa razão, foi iniciado o desenvolvimento de um arquitetura de protocolos para interoperar com diversos computadores e comutadores de diferentes fabricantes em diferentes ambientes de *software* e *hardware*, tornando possível a interconexão global das redes, origem da atual Internet. Entre os anos de 1977 e 1979, esta arquitetura de protocolos foi concluída e denominada de *TCP/IP Internet Protocol Suite*, comumente referida como TCP/IP.

A arquitetura TCP/IP é organizada em 5 (cinco) camadas: camada de aplicação, camada de transporte, camada de rede, camada de enlace e camada física conforme ilustrado na FIG. 2.1. Cada camada possui protocolos que executam serviços para as camadas superiores.

O protocolo empregado em cada camada contém detalhes referentes a:

- formatação da mensagem (estrutura de dados);
- descrição do serviço da respectiva camada a ser feito;
- descrição do endereçamento;

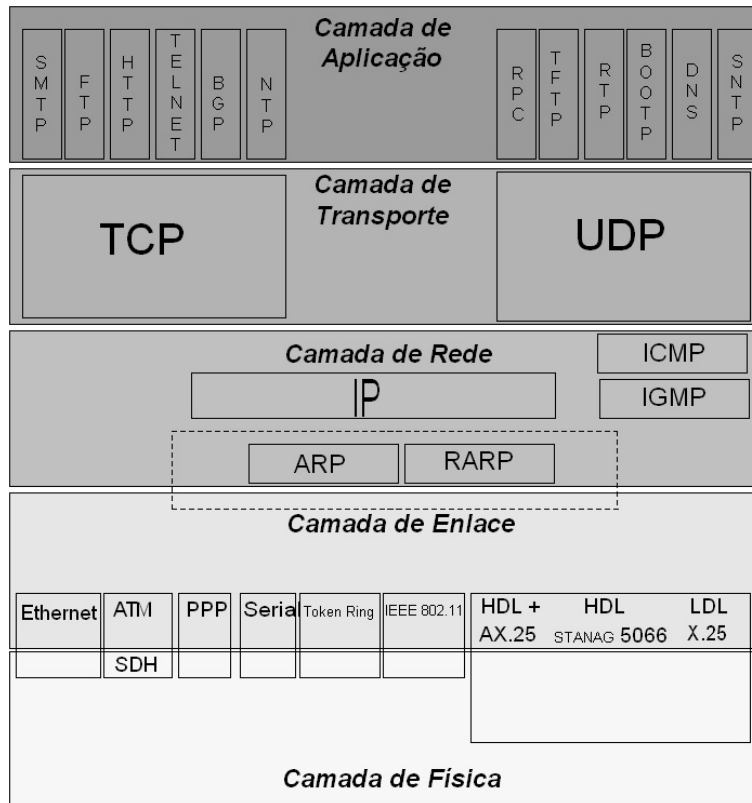


FIG. 2.1: Organização em camadas dos protocolos da arquitetura TCP/IP.

- ação em caso de erros e/ou condições anormais.

Os serviços de cada camada são realizados através das informações contidas nos campos do cabeçalho da respectiva camada.

A formação da estrutura de dados de cada camada, também chamada de PDU (*Protocol Data Unit*), é feita adicionando o cabeçalho da respectiva camada e enviando a PDU assim formada para a camada imediatamente inferior.

No lado de recepção, cada camada retira o seu cabeçalho, acessa às informações nele contidas e, após a realização dos serviços, a PDU restante é passada para camada imediatamente superior, conforme descrito na FIG. 2.2.

As estruturas de dados de cada camada (PDU) recebem denominações específicas de acordo com a camada. Em particular a PDU é denominada de segmento, pacote e quadro nas camadas de transporte, de rede e de enlace, respectivamente, conforme ilustrado na FIG. 2.2.

Os protocolos da camada de aplicação têm como função prover a interação necessária com os usuários por meio de uma arquitetura cliente-servidor.

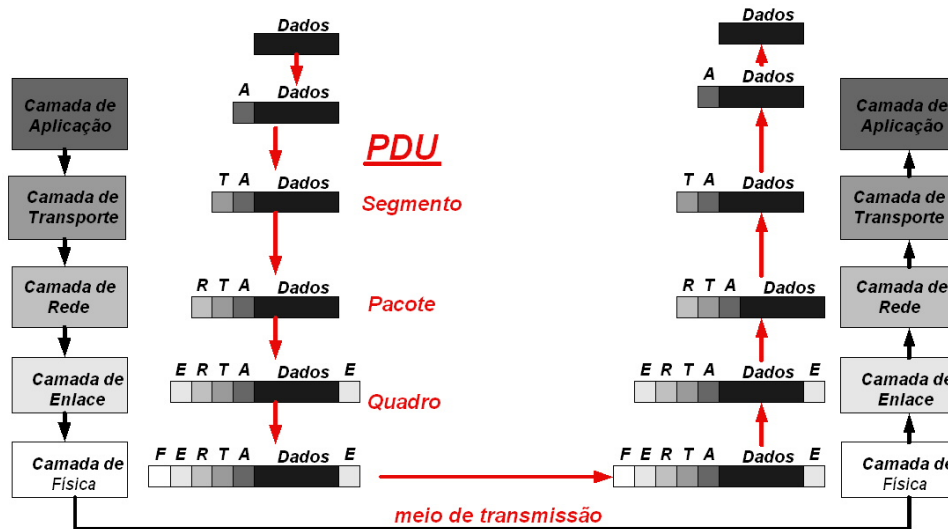


FIG. 2.2: Formação das estruturas de dados de cada camada.

A arquitetura TCP/IP especifica os protocolos UDP (*User Datagram Protocol*) e TCP (*Transmission Control Protocol*) para as funções de camada de transporte. Estes protocolos aparecem como os primeiros protocolos fim-a-fim, sendo, portanto, críticos para a questão de interoperação das redes convencionais e, em especial, das redes sem fio.

O UDP (POSTEL, 1980) tem como função principal a multiplexação das diversas aplicações, diferenciando as mesmas por meio do conceito de portas UDP padronizadas. Este protocolo não dispõe de quaisquer mecanismos de garantia de entrega dos dados, tampouco de correção de erros, sendo, portanto, um protocolo não orientado à conexão (*connectionless*). A estrutura de dados desta camada é o datagrama UDP.

Em contrapartida, o TCP (POSTEL, 1981b) possui mecanismos para prover garantia de entrega de dados, controle de fluxo e congestionamento na rede. Além do mais, este protocolo realiza a função de multiplexação das diversas aplicações através do uso de identificadores padronizados denominados portas TCP.

A camada de Rede tem por finalidade prover o roteamento dos pacotes e realizar o envio de mensagem de erros de roteamento. Os serviços realizados pelo protocolo *Internet Protocol* (POSTEL, 1981a) são caracterizados por serem não orientados à conexão e sem garantia de entrega dos dados.

Os protocolos da camada de enlace têm por finalidade prover as funções de detecção e/ou correção de erros, de controle de fluxo, de endereçamento físico de enlace, de acesso ao meio e de controle da conexão lógica de enlace. Dependendo do tipo de meio de transmissão a ser considerado, o protocolo de enlace poderá implementar parte

ou totalidade dos serviços citados.

O protocolo UDP não será abordado no presente capítulo, tendo em vista que se encontra fora do escopo desta dissertação. Contudo, nas próximas seções serão detalhados o protocolo de aplicação FTP (*File Transport Protocol*), o protocolo de transporte TCP e as características gerais dos protocolos de enlace, tópicos estes que são fundamentais para o bom entendimento deste trabalho.

2.3 PROTOCOLO DE APLICAÇÃO: FTP

Como mostrado na FIG. 2.1, existem muitos protocolos de camada de aplicação e foge ao escopo deste trabalho discutí-los, apenas, de forma bastante sumária, o *File Transport Protocol* (FTP) que foi o escolhido para realizar as avaliações de desempenho dos protocolos de comunicação aqui investigadas. Esse protocolo foi escolhido tendo em vista a pequena quantidade de troca de mensagens entre os servidores e os clientes, diferente, por exemplo, da aplicação HTTP.

A quantidade de troca de mensagem dos protocolos de aplicação é especialmente crítica em termos de desempenho, sobretudo, em canais com grandes retardos, caso típico de HF.

O FTP é o protocolo de acesso e transferência de arquivos da Internet que utiliza o TCP para o transporte confiável de dados. Esse protocolo de aplicação vem sendo utilizado durante décadas, para permitir o acesso remoto de usuários cadastrados a servidores¹ FTP. Através do uso desse protocolo, quaisquer usuários da Internet munidos de login e senha (clientes FTP) podem copiar, escrever e modificar arquivos disponíveis nos servidores.

Este tipo de aplicação se caracteriza pela alta assimetria entre os fluxos de tráfego cliente-servidor e o servidor-cliente. Por exemplo, após as fases de abertura de conexão, fornecimento de login e senha, o cliente FTP envia o comando curto de copiar (baixo tráfego) ao servidor FTP tendo como resposta o atendimento de enviar o arquivo ao cliente (alto tráfego), conforme descrito na FIG. 2.3.

¹São os ativos de rede responsáveis pela disponibilização de serviços para os clientes. Em determinados protocolos da camada de aplicação há uma constante troca das funções de clientes e servidor.



FIG. 2.3: Princípio de funcionamento do protocolo FTP.

2.4 PROTOCOLO TCP

O protocolo TCP foi especialmente projetado para prover um serviço confiável de transporte de dados sobre uma rede que apresenta vários distúrbios que podem acarretar erros durante a transmissão da informação.

Uma grande rede, tal como a Internet, possui alta complexidade, tendo em vista que cada uma das diversas sub-redes que a integram apresenta diferentes topologias, larguras de banda, retardos, tamanhos de pacote, quantidades de usuários, dentre outros parâmetros. Assim, o TCP atua no sentido de se adaptar dinamicamente às alterações dos parâmetros da rede bem como no de responder ou solucionar eventuais falhas ocorridas durante a transmissão.

De maneira resumida, o TCP tem as seguintes funções:

- formatar o segmento TCP;
- realizar a abertura (*three way handshaking*) e fechamento de conexão;
- efetuar o controle de erros e/ou perdas de segmento;
- controlar o fluxo de dados;
- realizar o controle de congestionamento;
- prover segmentação, agrupamento e seqüenciamento dos segmentos;

- multiplexar as diversas aplicações.

Embora o TCP se encontre presente como parte integrante da arquitetura TCP/IP, seu funcionamento é independente dos protocolos da camada de rede. Assim sendo, o protocolo opera de forma independente sobre qualquer outro serviço de entrega de pacotes, como por exemplo sobre o protocolo AX.25.

Serão descritas na Seção 3.3.1.2 algumas versões do TCP, destinadas a melhorar a vazão do protocolo em redes sem fio. Dentre as versões, o TCP Reno é a mais utilizada na Internet e considerada como padrão de fato. Assim sendo, serão detalhados nas seções subsequentes os conceitos e as funcionalidades do TCP Reno.

2.4.1 FORMATO DO SEGMENTO TCP

A primeira importante funcionalidade do TCP diz respeito a segmentação dos dados oriundos da aplicação. Na fase de abertura de conexão, os terminais definem o tamanho máximo de dados (*payload*) que cada segmento pode transportar, com base no campo do cabeçalho TCP chamado Tamanho Máximo de Segmento (*Maximum Segment Size* - MSS).

O cabeçalho TCP, apresentado na FIG.2.4, possui tamanho padrão de 20 bytes e o segmento TCP utilizado na Internet de 556 bytes, ou seja, 536 bytes de *payload*. Adicionando este segmento padrão ao cabeçalho IP de 20 bytes, resulta um pacote IP padrão de 576 bytes.

Na FIG. 2.4 é mostrada a configuração do cabeçalho do TCP, cujos campos são especificados abaixo:

- **Source port** e **Destination port** - São os identificadores das aplicações de origem e destino envolvidos na conexão TCP.
- **Sequence number** - É o número que identifica um segmento TCP.
- **Acknowledgment number** - Especifica o próximo byte esperado pelo receptor.
- **TCP header length** - Tamanho do cabeçalho TCP em múltiplos de 32 bits.
- **Flags**: Composto de seis bits que possuem as seguintes funções:
 - **URG** - Campo que indica um segmento urgente.

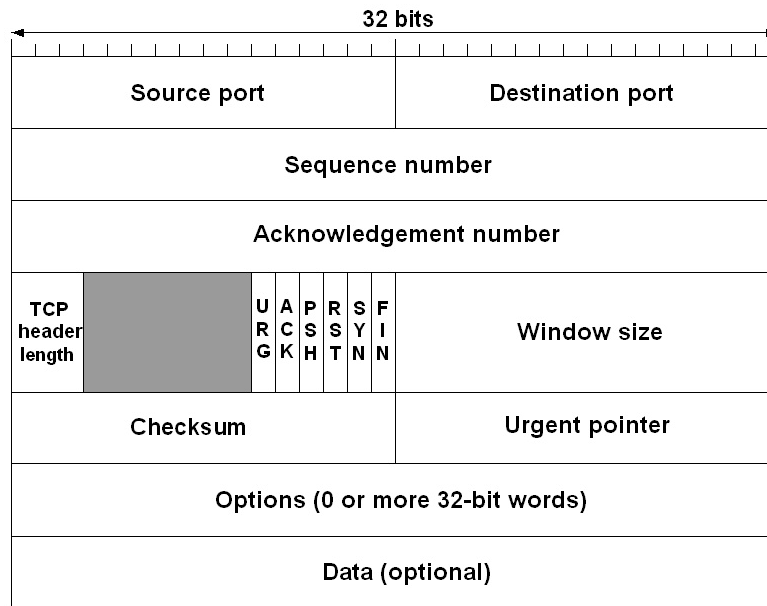


FIG. 2.4: Formato do segmento TCP.

- **ACK** - campo que indica um segmento de reconhecimento positivo.
 - **PSH** - informa ao sistema operacional de destino para proceder a entrega imediata para a aplicação.
 - **RST** - reinicialização da conexão.
 - **SYN** - utilizado no estabelecimento da conexão.
 - **FIN** - utilizado no fechamento da conexão.
- **Window size** - Indica o tamanho da janela do receptor. Em outras palavras, informa à fonte TCP, por meio da indicação presente no reconhecimento positivo, quantos bytes poderão ser enviados.
 - **Urgent pointer** - Este campo permite que o destinatário saiba quantos dados urgentes serão enviados.
 - **Options** - Utilizado para serviços extras, como, por exemplo, para aumentar o tamanho máximo de *payload* permitido.

2.4.2 ABERTURA E FECHAMENTO DE CONEXÃO

A abertura de uma conexão TCP é efetuada empregando-se uma técnica chamada de *three way handshaking*. Nesta técnica, cujo principio de funcionamento é

apresentado na FIG. 2.5, o computador A envia um segmento de abertura de conexão (SYN) ao computador B. Este, por seu turno, responde com um segmento de reconhecimento positivo (*Acknowledgment segment - ACK*) e o número de seqüência Y . Finalmente, o computador A retorna para o computador B um pacote de reconhecimento positivo ($ACK\ X + 1$) que confirma a chegada de um pacote de abertura de conexão SYN.

Os números de seqüência inicial X e Y são gerados aleatoriamente em ambos os computadores e utilizados durante a conexão TCP como identificadores dos segmentos.

Por outro lado, o fechamento da conexão TCP emprega 4 (quatro) segmentos, como ilustrado na FIG. 2.5. Neste caso o segmento FIN significa a intenção de terminar a conexão, logo para que a conexão seja terminada, ambos os computadores devem explicitar esta intenção, enviando o segmento FIN e a respectiva confirmação do recebimento através do ACK, gerando, portanto, a troca de 4 segmentos.

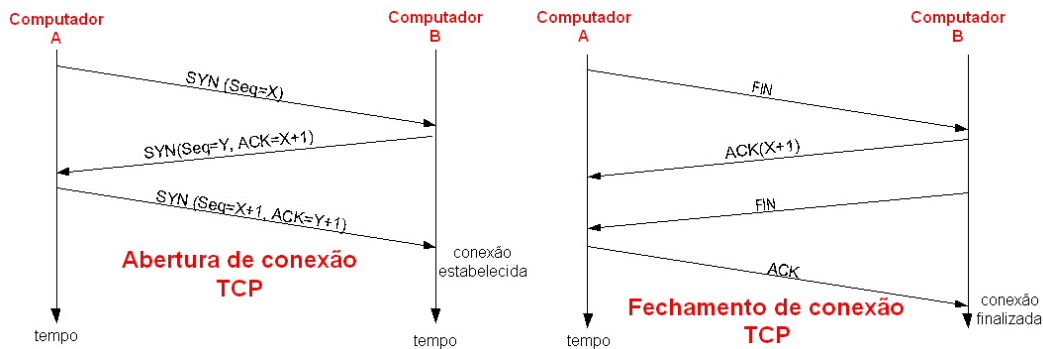


FIG. 2.5: Procedimentos de abertura e fechamento de conexão.

Vale ressaltar que a conexão estabelecida consiste de um circuito virtual entre dois programas de aplicação, de forma que cada terminal é associado a um par de identificadores: endereço IP e a porta TCP. Portanto, a conexão realizada entre os computadores A e B é unicamente especificada pelos 2 (dois) pares de endereços IP e portas TCP desses computadores.

2.4.3 CONTROLE DE ERROS E/OU PERDAS DE SEGMENTO

Nesta seção são apresentados alguns mecanismos que podem ser adotados para realizar controle de erro ou perda de segmento em uma rede TCP.

- a) Uso de reconhecimento positivo com retransmissão.

O serviço de transporte é considerado confiável quando a entrega dos dados ocorre sem duplicação ou erros. A maioria dos protocolos de transporte usa uma técnica

simples chamada de reconhecimento positivo com retransmissão (*positive acknowledgment with retransmission*) como mecanismo para conferir confiabilidade na transmissão.

De forma genérica, existem três estratégias de reconhecimento que os protocolos implementam: *Stop-and-Wait*, janela deslizante (*sliding-window*) e *Go-Back-N*.

Na estratégia *Stop-and-Wait*, ilustrada na FIG. 2.6, a fonte mantém gravadas todas as PDUs² enviadas e estabelece que somente será transmitida uma nova PDU após o reconhecimento positivo (ACK).

A cada transmissão a fonte inicia um temporizador para estabelecer um tempo máximo de espera da mensagem de reconhecimento (*timeout*). Caso esse tempo seja transcorrido sem que a PDU seja reconhecida, a fonte interpreta esse evento como uma perda e imediatamente retransmite a PDU supostamente perdida. Na FIG. 2.6 mostra-se que para este mecanismo de controle de erro o canal permanece ocioso na maior parte do tempo, pois somente ocorre transmissão em uma única direção, por conta disso, esta estratégia é ineficiente na utilização do canal, principalmente, naqueles com alto retardo, os quais determinam a necessidade de empregar elevados valores de *timeout* para evitar a retransmissão de pacotes recebidos com sucesso.

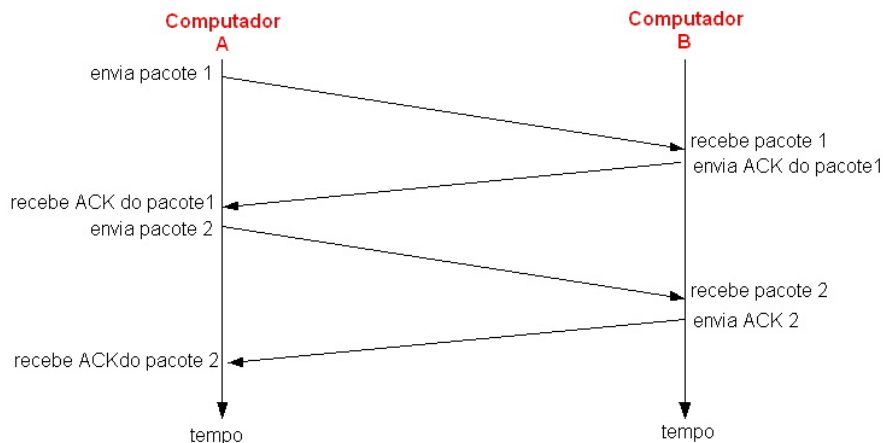


FIG. 2.6: Estratégia de reconhecimento *Stop-and-Wait*.

A estratégia de Janela Deslizante, diferentemente da *Stop-and-wait*, permite a transmissão de pacotes que estão incluídos dentro de uma janela sem a necessidade do recebimento de mensagens de reconhecimento, possibilitando dessa forma reduzir o tempo de ociosidade do canal.

²O termo PDU (*Protocol Data Unit*) foi escolhido no sentido de caracterizar o emprego de tais estratégias de reconhecimento tanto na camada de transporte quanto na camada de enlace.

O princípio de funcionamento da estratégia de Janela Deslizante é ilustrado nas Figuras 2.7 e 2.8. Quando um terminal contém uma seqüência de dados a ser transmitida, o protocolo aloca uma janela de tamanho fixo sobre a seqüência e transmite todas as PDUs nela contida. O número máximo de PDUs por transmissão é denominado de janela de transmissão, cuja representação é ilustrada na FIG. 2.8.

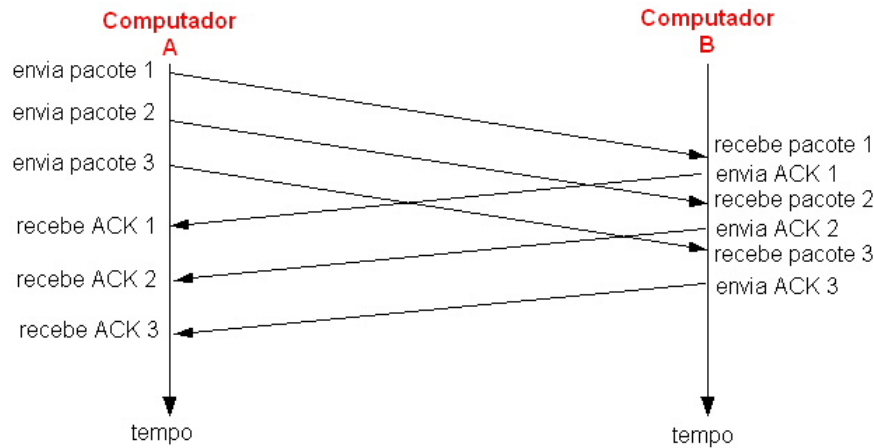


FIG. 2.7: Exemplo de três pacotes transmitidos usando a Janela Deslizante.



FIG. 2.8: Operação com a Janela Deslizante.

Em particular no TCP, a cada reconhecimento positivo recebido pela fonte a janela é deslocada ou "deslizada" para incluir em seu interior um novo bloco de dados. Vale mencionar que à esquerda da Janela Deslizante encontram-se os blocos de dados

enviados com sucesso, à sua direita aqueles que estão na iminência de serem enviados e no seu interior são representados os blocos de dados que já foram enviados e cujas mensagens de reconhecimento ainda não foram recebidas.

O desempenho de protocolos que utilizam o conceito de janela deslizante depende do tamanho da janela e da taxa de envio das PDUs. Por exemplo, quando o tamanho da janela deslizante é igual a uma PDU, esta estratégia de transmissão é igual à *stop-and-wait* que, conforme mencionado anteriormente, determina uma baixa utilização do canal. À medida que se aumenta o tamanho da janela de transmissão, diminui-se o tempo de ociosidade do canal. No entanto, se este aumento for muito grande, corre-se o risco de se enviar um número de PDUs maior do que a capacidade da rede, ocasionando o fenômeno de congestionamento. Este, por seu turno, provoca perdas de pacotes determinando, em última análise, um uso inadequado da rede.

Diante do acima exposto, o tamanho da janela deslizante deve ser escolhido com critério, de tal sorte que em estado estacionário a fonte seja capaz de transmitir PDUs tão rápido quanto a rede pode suportar, sem ocasionar congestionamento. Vale mencionar também, como será discutido posteriormente, que a manipulação do tamanho da janela de transmissão é uma ferramenta poderosa de que o protocolo TCP dispõe para diminuir o congestionamento na rede.

Outra estratégia para conferir confiabilidade na transmissão e que é adotada em alguns protocolos é a *Go-Back-N*. Nesta abordagem quando um pacote de número de seqüência n é perdido, o receptor solicita a retransmissão deste pacote e dos posteriores a ele que estão dentro da janela de transmissão, ou seja, que a transmissão seja re-iniciada a partir do pacote perdido. Na FIG.2.9 é apresentado o princípio de funcionamento do protocolo *Go-Back-N* para uma janela de transmissão de tamanho 5 quando o pacote 2 é perdido.

b) Detecção de perdas de pacotes.

O TCP utiliza duas fontes de sinalização de perda de pacotes: a ocorrência de *timeout* e o recebimento dos chamados ACKs duplicados.

O processo de reconhecimento do TCP é acumulativo. Ou seja, o ACK informa à fonte o número do próximo segmento (*sequence number*) a ser recebido, aqui denotado por n , indicando que todos os segmentos com numeração igual ou inferior a $n - 1$ foram recebidos corretamente.

Assim, quando ocorre o recebimento do segmento com número de seqüência

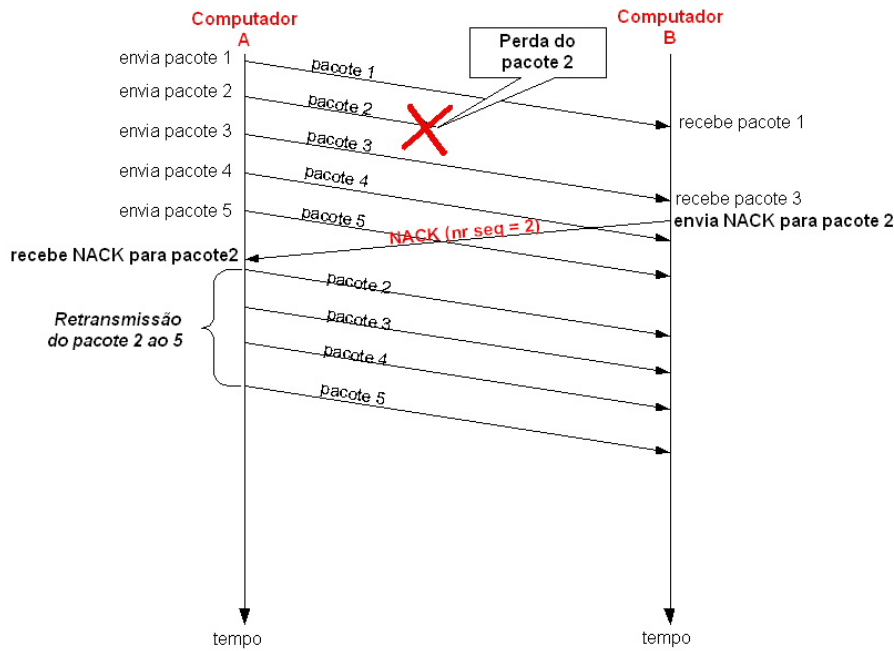


FIG. 2.9: Estratégia de reconhecimento *Go-Back-N*.

maior do que n , um novo ACK solicitando o segmento n será enviado, configurando um ACK duplicado. Portanto, a chegada de segmentos fora de ordem geram ACKs duplicados na fonte.

Na verdade, a recepção de um pacote fora de ordem e, a correspondente geração do ACK duplicado, são normais em redes com roteamento dinâmico onde pacotes trafegam em rotas diferentes que apresentam retardos diferentes. Por outro lado, em condições anormais, outro motivo para o recebimento de pacotes fora de ordem é a perda de pacotes. Logo, o aumento da geração de ACK duplicados configura um indício de anormalidade na entrega dos pacotes.

Desta forma, o recebimento de 3 (três) ACKs duplicados é usado como um indicativo para a fonte TCP que ocorreu uma perda de pacote, sendo o mesmo naturalmente retransmitido, conforme ilustrado na FIG. 2.10, sem aguardar a ocorrência do *timeout*.

O outro procedimento empregado pelo TCP para realizar a detecção de perda de pacote é o uso do *timeout*. O valor do temporizador *timeout* é um parâmetro fundamental para o desempenho do TCP em canais sem fio, pois influi diretamente na agilidade do protocolo na detecção do pacote perdido.

Pode-se dizer que a detecção de perda de pacote via *timeout* sugere a ocorrência

de perdas em surto oriundas do congestionamento severo ou das condições de propagação em canais sem fio, ou seja, um cenário ruim para o tráfego de pacotes. Entretanto, para o caso de detecção de perda de pacotes por meio dos ACKs duplicados sugere uma condição de ocorrência moderada de perdas de pacotes.

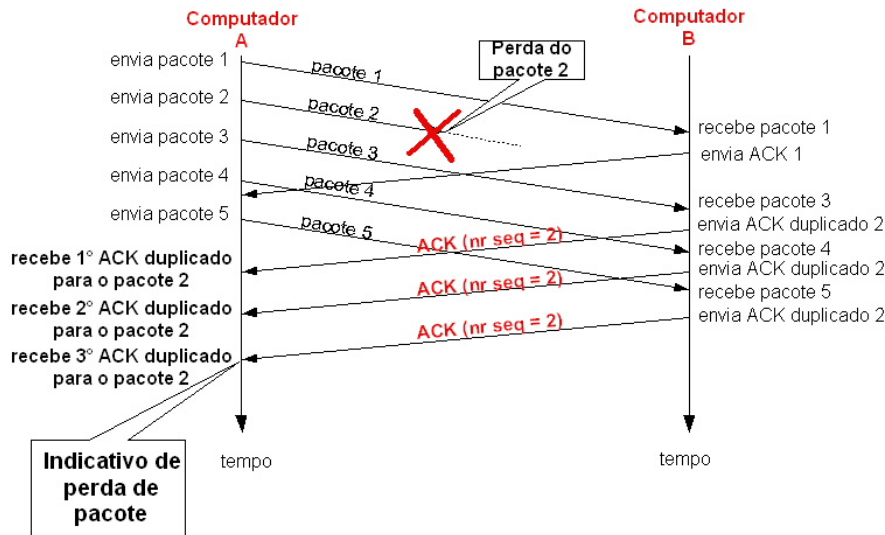


FIG. 2.10: Exemplo de perda de um pacote e a geração de reconhecimentos duplicados.

c) Estimativa de *Timeout* e o algoritmo de Karn:

Um dos mais complexos e importantes conceitos do TCP é a manipulação dos *timeouts* e o correspondente algoritmo de retransmissão. A complexidade reside no ambiente de Internet para o qual o TCP foi projetado. Neste ambiente, as transmissões de pacotes entre dois pontos terminais da rede, chamados de *hosts*, estão sujeitos a variações de retardo, em virtude da possibilidade de que cada pacote transite por caminhos diferentes que podem apresentar tanto retardos quanto capacidades de transmissão diferentes. Além disso, ao longo de cada caminho estão os roteadores³ que possuem uma unidade de memória temporária, chamada de *buffer*, utilizada na serialização dos pacotes oriundas de diversas sub-redes, e que podem introduzir, em função dos tráfegos de tais sub-redes, uma variação do retardo (*jitter*) considerável.

Assim sendo, é impossível o conhecimento *a priori* dos instantes de recepção dos reconhecimentos positivos. Na tentativa de considerar tais variações de retardo da Internet, o TCP utiliza o algoritmo de retransmissão adaptativa (*adaptive retransmission*

³Nó ativo utilizado na interligação de diversas redes e responsável pelo roteamento dos pacotes de uma rede para outra.

algorithm) que estabelece um valor razoável do *timeout* através do monitoramento contínuo da variável RTT (*Round Trip Time*) de cada conexão. O RTT é a estimativa do tempo entre a transmissão do pacote e o recebimento de seu reconhecimento positivo. Denotando por RTT_{ant} o valor do parâmetro RTT anteriormente estimado e por RTS (*Round trip Sample*) como a medição atual feita pelo TCP do tempo entre a transmissão do pacote e o recebimento de seu reconhecimento positivo, o RTT pode ser estimado de acordo com a EQ. 2.1

$$RTT = \alpha \times RTT_{ant} + (1 - \alpha) \times RTS, \quad (2.1)$$

na qual α é o fator de atenuação ou fator da média ponderada. Para valores pequenos de α , resultará em grande variabilidade a cada cálculo do RTT estimado. Ao passo que para valores pequenos de α , a contribuição na variação do RTT é pequena a cada nova estimativa. Um valor típico de α corresponde a 7/8 (TANEMBAUN, 1996).

Com base no RTT estimado o *timeout* é obtido pela EQ.2.2:

$$Timeout = \beta \times RTT, \quad (2.2)$$

onde β é o fator variacional do atraso. Conforme mostrado na referência (TANEMBAUN, 1996) um valor típico para este parâmetro é 2 (dois).

Foi apresentada em (JACOBSON, 1988) uma nova proposta de ajuste do *timeout*, pois na abordagem anterior, utilizando valores fixos de β , o TCP não consegue se adaptar às altas flutuações do RTT, causando retransmissões desnecessárias. Nesta nova abordagem, a estimativa do *timeout* foram considerados a estimativa do RTT e sua variância, em substituição ao parâmetro β do algoritmo precedente. Desta forma, denotando a variável *Erro* como a diferença entre o último valor medido e a estimativa corrente e RTT_s como o parâmetro RTT corrigido das bruscas variações, tem-se que

$$\begin{aligned} Erro &= RTS - RTT_{ant}, \\ RTT_s &= RTT_{ant} + \gamma \times Erro, \end{aligned} \quad (2.3)$$

onde γ representa um fator entre 0 e 1 com valor típico de 1/8 (COMER, 1995). Sendo DEV_s e DEV_{ant} correspondendo, respectivamente, ao desvio da medida atual e ao desvio de medida na iteração anterior, obtêm-se o valor de DEV_s , dado por

$$DEV = RTT_{ant} + \rho \times (|Erro| - DEV_{ant}), \quad (2.4)$$

onde ρ é um fator entre 0 e 1, que representa o grau de efeito das novas amostras no desvio médio. Em (COMER, 1995) mostra um valor adequado para este parâmetro é 1/4.

Assim, deriva-se o valor do temporizador *timeout* através da EQ. 2.6.

$$Timeout = RTT_s + \eta \times DEV, \quad (2.5)$$

sendo η o grau de efeito das novas amostras no *timeout*. Em (COMER, 1995) mostra o valor típico de $\eta = 4$.

Portanto, em ambas as implementações mencionadas, o valor do temporizador *timeout* é adaptativo, sendo esta característica fundamental para o uso do TCP em canais com alto retardo, caso típico de HF.

No caso de recebimento de ACK no cenário em que tanto a transmissão original quanto a retransmissão do pacote são perdidos ocorre uma situação de ambiguidade, pois o ACK recebido poderia se referir tanto à transmissão original quanto a retransmissão. Neste caso a estimativa de *timeout* torna-se incerta. Para resolver esta ambiguidade foi proposto o algoritmo de Karn que em caso de perda de pacote o valor corrente do *timeout* é dobrado, conforme a EQ.2.6:

$$Timeout = Timeout_corrente \times 2. \quad (2.6)$$

2.4.4 CONTROLE DE FLUXO

O TCP incorpora mecanismos de controle de fluxo para garantir às comunicações entre *hosts* com diferentes taxas de transmissão. Em outras palavras, a fonte TCP somente poderá enviar segmentos a uma taxa compatível com a capacidade do receptor TCP, sendo essa característica fundamental para lidar com a heterogeneidade da Internet.

O controle de fluxo é feito através da largura da janela de transmissão que pode ser definida como o mínimo entre a janela de congestionamento (*cwnd*) e a janela do receptor TCP, também chamada de *advertisement window* (*awnd*).

$$\text{Janela de transmissão} = \text{Mínimo}(cwnd, awnd). \quad (2.7)$$

A janela de transmissão corresponde ao tamanho da janela deslizante efetivamente utilizado pelo TCP, descrito na Seção 2.4.3. Logo, pode-se dizer que o tamanho máximo da janela deslizante utilizada pelo TCP é limitado de acordo com a expressão 2.7.

De fato, na recepção de cada ACK, o campo "*Window Size*" do cabeçalho do segmento TCP (FIG. 2.4) contém a quantidade de bytes que o receptor está preparado para receber, ou seja, o parâmetro *awnd*. Além disso, durante a fase de abertura de conexão TCP ambos os terminais trocam informações mútuas a respeito do *awnd*.

A janela de congestionamento é a variável manipulada pelo TCP para responder aos eventuais congestionamentos na rede. Os mecanismos de controle de congestionamento serão detalhados na próxima seção.

2.4.5 CONTROLE DE CONGESTIONAMENTO

Conforme mencionado na Seção 2.4.3, as fontes de congestionamento resultam do acúmulo de pacotes nos *buffers* dos roteadores ao longo do caminho. O congestionamento tem como principais conseqüências a ocorrência de *timeout* e a perda de pacotes, em caso de estouro da capacidade de *buffer* em algum dos roteadores.

O TCP realiza o controle de congestionamento através de quatro algoritmos: *slow start*, *congestion avoidance*, *fast retransmit* e *fast recovery*, os quais serão discutidos a seguir.

a) *Slow Start*

Inicialmente, a fonte TCP não possui qualquer informação atinente ao estado da rede. Em razão disso, o algoritmo *Slow Start* inicia a conexão com uma janela de congestionamento pequena, sendo aumentada exponencialmente no decorrer da transmissão. O TCP utiliza o *Slow Start* no início de cada conexão ou no caso de reinício em função da ocorrência de *timeout*. O algoritmo é resumido a seguir:

1. Início da conexão ou após o *timeout* $\implies cwnd = 1$;
2. Cada recebimento de ACK $\implies cwnd = cwnd + 1$.

A ocorrência de *timeout* é interpretada pelo TCP como um indício de alto

grau de congestionamento da rede. Desta forma, o algoritmo reduz bruscamente a janela de congestionamento e, conseqüentemente, a taxa de envio dos pacotes.

Portanto, a cada ACK recebido a variável *cwnd* é incrementada de uma unidade de tamanho de segmento⁴. O efeito desta lógica é o aumento exponencial de *cwnd* levando rapidamente ao *timeout*.

b) *Congestion Avoidance*

Após a ocorrência de *timeout*, provocado pelo *Slow Start*, o algoritmo *Congestion Avoidance* é acionado, visando incrementar o *cwnd* de maneira mais conservadora. Este algoritmo é apresentado a seguir.

1. Cada recebimento de ACK $\implies cwnd = cwnd + \frac{1}{cwnd}$.

c) *Slow Start e Congestion Avoidance*

Na realidade, os algoritmos *Slow Start* e *Congestion Avoidance* são implementados em conjunto, sob o controle de uma variável denominada *ssthresh* (*Slow Start threshold*) da seguinte maneira:

1. Início da conexão $\implies cwnd = 1$ e *ssthresh* = 0 : *Slow Start*;
2. Ocorreu *timeout* $\implies ssthresh = \frac{cwnd}{2}$;
3. *cwnd* = 1 : reinicia *Slow Start*;
4. Se *cwnd* < *ssthresh* $\implies cwnd$ manipulada por *Slow Start*;
5. Se *cwnd* \geq *ssthresh* $\implies cwnd$ manipulada por *Congestion Avoidance*;

A FIG. 2.11 ilustra o emprego da variável *ssthresh* no controle dos algoritmos *Slow-Start* e *Congestion Avoidance*:

d) *Fast Retransmit*

A finalidade do *Fast Retransmit* é tornar mais eficiente o processo de retransmissão de pacotes. Este algoritmo estabelece que o recebimento de 3 (três) ACKs duplicados indica uma perda de pacote, devendo a fonte retransmiti-lo imediatamente.

Desta forma, evita-se a longa espera pelo estouro do temporizador (*timeout*), agilizando a retransmissão.

⁴O mecanismo de janela deslizante do TCP opera a nível de bytes (octeto).

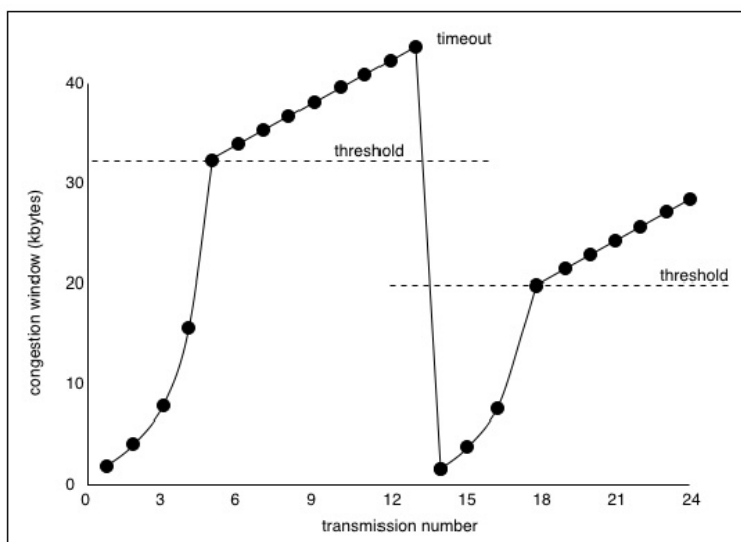


FIG. 2.11: Exemplo da evolução do controle de congestionamento

e) *Fast Recovery*

O *Fast Recovery* é implementado em conjunto com o *Fast Retransmit*, visando a manipulação de *cwnd* em caso de congestionamento moderado. A descrição do algoritmo é resumida a seguir:

1. Ao receber 3 ACKs duplicados consecutivos ou ocorrer *timeout* $\implies ssthresh = \frac{cwnd}{2}$;
2. Retransmite o pacote supostamente perdido : *Fast Retransmit*;
3. $cwnd = ssthresh + 3$;
4. A cada ACK duplicado recebido $\implies cwnd = cwnd + 1$;
5. Ao receber o novo ACK do pacote retransmitido $\implies cwnd = ssthresh$.

Conforme mencionado na Seção 2.4.3 a perda de pacote é detectada de duas maneiras: *timeout* ou recepção de 3 ACKs duplicados. No primeiro caso, como não se pode inferir sobre as condições da rede, é adotada uma medida conservadora por meio do algoritmo *Slow Start* com o retorno do tamanho da janela de congestionamento para um segmento. No segundo caso, pode-se inferir a ocorrência de um congestionamento moderado, supondo que o pacote ainda esteja circulando na rede, sendo adotado os algoritmos *Fast Retransmit/Fast Recovery*.

2.5 PROTOCOLO IP

O protocolo IP, pertencente a camada de rede da arquitetura TCP/IP, define os mecanismos de entrega dos pacotes aos destinatários. Os serviços executados por este protocolo são caracterizados por serem não confiáveis e não orientados à conexão e, portanto, são serviços sem garantia de entrega. Contudo, o IP utiliza todos os recursos e ferramentas disponíveis para encaminhar os pacotes ao destino. Em decorrência disso, o serviço IP é chamado de "Melhor esforço" ou *Best Effort*.

De maneira geral, este protocolo não possui requisitos rígidos relacionado aos instantes de entrega dos pacotes, sendo esta característica fundamental para interoperação com redes sem fio de grande retardo. Porém, em determinadas situações onde se deseja garantir tráfego de aplicações com requisitos rígidos de qualidade de serviço⁵ (QoS), o IP implementa mecanismos de reserva de recursos e diferenciação das classes de serviços.

O IP é responsável pelas seguintes funções:

- Especificar o formato dos pacotes IP;
- Encaminhar os pacotes com base nos algoritmos de roteamento;
- Estabelecer os procedimentos no processamento dos pacotes nos *hosts* e nos roteadores;
- Gerar mensagens de erros de roteamento;
- Estabelecer procedimentos quanto ao descarte de pacotes nos roteadores;
- Executar as políticas de qualidade de serviço (*Quality of Service - QoS*);
- Definir o endereçamento lógico da rede (endereço IP);
- Realizar a fragmentação dos pacotes, em caso de necessidade.

Portanto, em virtude das funções apresentadas acima, percebe-se que este protocolo representa parte fundamental da arquitetura TCP/IP.

⁵Define-se requisitos de *QoS* como sendo os requisitos da aplicação, ou seja, o que ela requisita da rede a fim de que funcione adequadamente. Estes requisitos são traduzidos em parâmetros indicadores do desempenho da rede como, por exemplo, atraso máximo, *jitter*, taxa máxima de perda de pacotes permitida, etc.

2.6 PROTOCOLOS DE ENLACE

Os protocolos de camada de enlace são responsáveis pela comunicação entre dois pontos sucessivos de uma mesma rede. A união de tais pontos da rede formam as chamadas Redes Locais (*Local Area Network - LAN*) e as Redes Metropolitanas (*Metropolitan Area Network - MAN*).

De maneira geral, os protocolos da camada de enlace executam as seguintes funções:

- Prover a interface com o protocolo de camada superior (protocolo IP, caso geral) e com o protocolo da camada física;
- Estabelecer o formato do quadro;
- Executar os mecanismos de controle de acesso ao meio (*Medium Access Control - MAC*);
- Estabelecer o endereçamento físico (endereço de enlace);
- Prover mecanismos para melhorar a confiabilidade do enlace por meio de detecção e/ou correção de erros e de mecanismo de reconhecimento local (*Automatic Repeat Request - ARQ*) dos quadros;

A camada de enlace é formada por um ou mais protocolos que implementam o conjunto ou parte as funções supracitadas. Como exemplo de tais protocolos tem-se: as especificações IEEE 802.2 e 802.3 (*Ethernet*), IEEE 802.11 a e b (*Wireless LAN*), ATM (*Asynchronous Transfer Mode*), *Frame Relay*, X.25, AX.25, HDLC (*High-level Data Link Control*), PPP (*Point to Point Protocol*), RLP (*Radio Link Protocol*), STANAG 5066, LDL ARQ (*Low-latency Data Link protocol*), HDL ARQ (*High-latency Data Link protocol*), HDL+, etc (ver FIG.2.1).

No próximo capítulo serão detalhados alguns protocolos de enlace que foram usados nas simulações a serem apresentadas no CAP. 6, bem como aqueles especificamente estabelecidos como padrões para sub-redes HF.

2.7 RESUMO

Neste capítulo foram apresentados alguns dos aspectos mais importantes da arquitetura TCP/IP, dos protocolos FTP, IP e de protocolos de enlace, assim como a

fundamentação teórica mais detalhada do protocolo TCP convencional e seus mecanismos de controle de congestionamento, de erros e de fluxo, em redes baseada na arquitetura TCP/IP.

No próximo capítulo serão discutidas os problemas de vazão apresentados pelo TCP em redes sem fio, bem como as possíveis estratégias para sua melhoria.

3 VAZÃO DO PROTOCOLO TCP EM REDES SEM FIO E ESTRATÉGIAS PARA SUA MELHORIA

3.1 INTRODUÇÃO

No presente capítulo, inicialmente, serão apresentados os problemas de baixo desempenho do protocolo TCP em redes sem fio. Posteriormente, serão discutidas algumas estratégias recentemente propostas em trabalhos centrados na melhoria da vazão do protocolo TCP em ambientes sem fio.

3.2 BAIXA VAZÃO DO PROTOCOLO TCP EM REDES SEM FIO

O protocolo TCP convencional foi projetado para operar em redes cujos canais de comunicação são confinados, os quais, em geral, se caracterizam por apresentar baixas taxas de erros. Em razão disso, o TCP apresenta baixo desempenho em canais com altas taxas de erro.

Isto se deve ao fato de que o protocolo TCP considera quaisquer tipos de erros como fonte ou indício de congestionamento na rede. Assim, quando ocorre a perda de pacote, o TCP utiliza os algoritmos de controle de congestionamento descritos na Seção 2.4.5 que, de forma conjunta, ativam mecanismos de resposta ao suposto congestionamento, resultando na redução da janela de congestionamento do emissor e, conseqüentemente, da taxa de envio de pacotes. Assim sendo, a alta incidência de perdas de pacotes provoca a diminuição da vazão do protocolo TCP.

A premissa de que quaisquer perdas de pacotes advêm de congestionamento na rede se aplica bem aos meios convencionais guiados, porém, é inadequada para redes sem fio, onde considerável quantidade de perdas pode ser provocada por erros induzidos pelo canal de comunicação (perda por canal). Além disso, as camadas físicas dessas redes se caracterizam por apresentar erros em surto, podendo comprometer a recepção de pacotes dentro de uma mesma janela de transmissão, aumentando as chances de acionamento do algoritmo *Slow Start* e, conseqüentemente, provocando uma drástica redução da taxa de transmissão.

Após a detecção de um pacote perdido, ocorre a retransmissão do mesmo e um

aumento na estimativa de *timeout*, conforme a EQ. 2.6. Em um cenário com perdas de pacotes em surto pode ocasionar sucessivos aumentos do temporizador *timeout*, levando o protocolo, a esperar um tempo relativamente grande na detecção da próxima perda de pacote, deteriorando, desta forma, a vazão.

Portanto, em virtude dos mecanismos de redução da janela de congestionamento, do aumento do temporizador *timeout* e do acionamento do *Slow Start*, o TCP apresenta baixa vazão em canais sem fio.

3.3 ESTRATÉGIAS DE MELHORIA DA VAZÃO DO PROTOCOLOS TCP EM REDES SEM FIO

No sentido de prover uma melhoria na vazão do protocolo TCP convencional em redes sem fio, foram propostas diversas soluções, as quais podem ser classificadas em três estratégias ⁶ básicas (BALAKRISHNAN, 1997):

- Modificação e inclusão de novos algoritmos no protocolo TCP;
- Estratégias que modificam a semântica fim-a-fim do TCP;
- Soluções focadas nos protocolos da camada de enlace com ARQ.

As estratégias de modificação do TCP propõem refinamentos dos algoritmos existentes na versão convencional do TCP e implementações de novos algoritmos, com intuito de melhorar vazão do protocolo da camada de transporte em ambientes sem fio.

Por seu turno, as propostas de modificação na camada de enlace visam proporcionar maior confiabilidade, de forma a ocultar para o TCP as perdas devido ao canal. Como apresentado no capítulo anterior, os protocolos de enlace realizam a segmentação e remontagem do pacote IP em quadros, para implementação dos mecanismos de reconhecimento local automático chamado de "*Automatic Repeat Request*" (ARQ), detecção e correção de erros, controle de fluxo e técnicas adaptativas de camada física.

Por fim, as estratégias que modificam a semântica fim-a-fim do TCP requerem uma estação tradutora, também chamada de *proxy* ou *gateway*, entre os canais sem fio

⁶Estão fora do escopo desta dissertação as estratégias que levam em consideração falhas de roteamento, falhas de enlace e dimensionamento de buffers em redes do tipo *Ad hoc*. Tais redes se caracterizam por enlaces sem fio sucessivos, cujos terminais assumem funções de comutadores, aumentando a complexidade da análise do cenário.

e os canais guiados. Esta estação tem por finalidade ocultar as perdas do emissor TCP, melhorando a taxa de envio dos pacotes.

3.3.1 MODIFICAÇÃO E INCLUSÃO DE NOVOS ALGORITMOS NO PROTOCOLO TCP

As estratégias que envolvem a modificação e inclusão de novos algoritmos no protocolo TCP podem ser subdivididos em:

- Estratégias que atuem em resposta à fonte de perda de pacote;
- Desenvolvimento de novas versões do protocolo TCP;
- Otimização dos parâmetros do protocolo TCP.

3.3.1.1 ESTRATÉGIAS QUE ATUEM EM RESPOSTA À FONTE DE PERDA DE PACOTE

Busca-se neste caso a inclusão de mecanismos de tal forma que o TCP atue diferentemente para perda por canal e perda por congestionamento. Assim sendo, as estratégias deste tipo envolvem dois problemas distintos: primeiro, a classificação do motivo da perda de pacotes, ou seja, se as perdas são devidas ao canal ou ao congestionamento; e segundo, atuar em resposta à fonte de perda de pacotes.

Inicialmente, serão apresentados a seguir alguns métodos relativos a estes dois problemas.

a) Classificação do tipo de perda de pacotes

A classificação do tipo de perda de pacotes pode ser obtida por avaliação indireta ou implícita e avaliação explícita.

O método de classificação que emprega a avaliação indireta de perdas devido ao canal tem por base a estimação de parâmetros referentes à camada de rede e/ou à camada física. A desvantagem deste método é a possibilidade de imprecisão na classificação.

Por exemplo, em (SAMARAWEEERA, 1999) é proposto um método de avaliação indireta, chamado de algoritmo *Non-congestion Packet Loss Detection* (NCPLD). No referido algoritmo, faz-se o uso do conceito de *knee point* da rede, definido como o ponto máximo no gráfico de taxa de transmissão versus vazão, ilustrado na FIG. 3.1. Associado a este *knee point* calcula-se o limiar de retardo (*delay threshold*) de ida e volta

do segmento. Desta forma, após a medição do *RTS*, se o RTT estiver abaixo do limiar, as perdas de pacotes detectadas serão consideradas como devidas a perdas por canal. Caso contrário, serão consideradas como provocadas por congestionamento na rede. A referência (SAMARAWEEERA, 1999) apresenta maiores detalhes para obtenção do limiar de retardo.

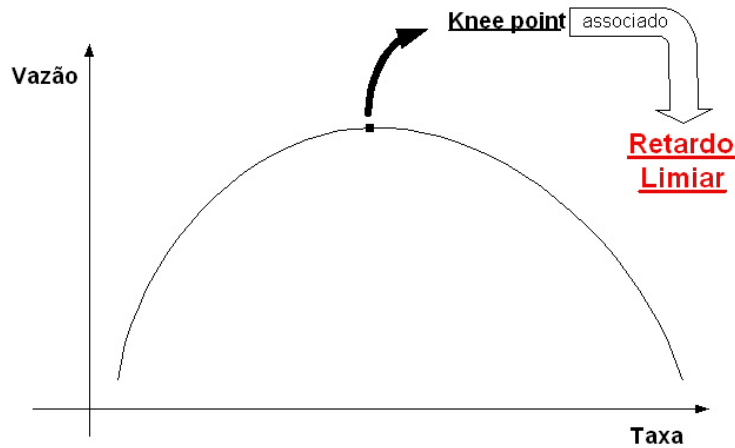


FIG. 3.1: *knee point* da curva de taxa de transmissão versus vazão

A referência (BANSAL, 1999) descreve outro método de determinação implícita da causa de perda de pacote, aplicável somente à versão TCP Tahoe. Esta antiga versão do TCP não emprega o algoritmo *Fast Recovery*, ou seja, após a execução do *Fast Retransmit*, o parâmetro *cwnd* é reduzido a um segmento, reiniciando o algoritmo Slow Start, conforme descrito no Seção 2.4.5.

Este método utiliza a variável auxiliar *ssthresh* usada nos algoritmos de controle de congestionamento do TCP. Com base nos valores dessa variável no instante da perda de pacote, assim como no tempo decorrido entre sucessivas perdas, é realizada a classificação do tipo de perda de pacotes. Denotando por S_i o valor de *ssthresh* após a i ésima perda de pacote e por N_i denotando o tempo entre a i ésima e a $(i + 1)$ ésima perda de pacote, tem-se que se:

- $S_i > S_{i-1}$ e $N_i < N_{i-1} \implies$ a perda é devida ao congestionamento;
- $S_i > S_{i-1}$ e $N_i > N_{i-1} \implies$ a perda é devida ao canal;
- $S_i < S_{i-1}$ e $N_i < N_{i-1} \implies$ a perda é devida ao congestionamento;
- $S_i < S_{i-1}$ e $N_i > N_{i-1} \implies$ a perda é devida ao canal.

É sabido que $cwnd$ aumenta exponencialmente até que o valor de $ssthresh$ seja atingido, de acordo com o mecanismo *Slow Start*. O acima algoritmo supõe que se S_i é maior do que S_{i-1} , a taxa de envio de pacotes durante o período entre i_{esima} e $(i+1)_{esima}$ perda de pacote é maior do que a taxa de envio de pacotes durante o período entre $(i-1)_{esima}$ e i_{esima} perda de pacote. Portanto, neste caso é maior a probabilidade de ocorrência de congestionamento, produzindo perdas sucessivas, ou seja, tempos menores entre as perdas ($N_i < N_{i-1}$).

Os métodos de classificação que empregam a avaliação indireta têm a vantagem de somente requerer códigos adicionais no emissor, não requerendo ajustes na rede e/ou no receptor. Este é o método menos impactante em termos de adição de códigos, porém é menos preciso. Outros métodos de classificação do tipo de perda de pacote fazem uso de notificações explícitas.

Um destes métodos é chamado Notificação Explícita de Perda por Canal (*Explicit Wireless Loss Notification - EWLN*). Neste caso, as notificações podem ser enviadas por uma estação rádio presente na rede ou pelo receptor, para o emissor TCP, informando-o explicitamente a ocorrência de perdas por canal. O método EWLN apresentado em (BANSAL, 1999) pode ser implementado por meio do uso de mensagens *Internet Control Message Protocol* (ICMP) ou por meio do cabeçalho TCP. O protocolo ICMP, presente na camada de rede da arquitetura TCP/IP, especifica um conjunto de mensagens padronizadas geradas pelos roteadores para informar ao emissor a ocorrência de problemas de roteamento de algum pacote enviado. A implementação por meio do cabeçalho TCP utiliza o campo *options*, descrito na Seção 2.4.1, para notificação.

Outro método de classificação por meio de notificações é o chamado Notificação Explícita de Congestionamento (*Explicit Congestion Notification - ECN*), pelo qual o roteador sinaliza explicitamente para o emissor a ocorrência de congestionamento.

A proposta de adicionar ECN na arquitetura TCP/IP apresentada em (RAMAKRISHNAN, 2000) estabelece que, se os *buffers* dos roteadores estiverem na iminência de estourar, eles informarão a ocorrência deste fato ao emissor, por meio do cabeçalho IP, com o bit do campo *Congestion Experienced* setado. Este método foi adaptado em (FEI, 2000) para redes sem fio, sendo denominado *Wireless-ECN* (WECN). Este mecanismo de classificação da perda especifica que a falta da notificação WECN indica a ocorrência de perda devida ao canal. Em caso contrário, a perda é computada como devida ao congestionamento, invocando os mecanismos usuais de controle de congestionamento.

A vantagem das implementações que utilizam as notificações explícitas é a maior precisão na classificação, porém a desvantagem destes métodos é a necessidade de alteração dos códigos nos emissores e estações rádio e, em alguns casos, em todos os roteadores, tornando-os bastante onerosos.

b) Controle do tamanho da Janela de Congestionamento em caso de perda por canal:

Uma vez classificadas o tipo de perda de pacote, serão considerados os procedimentos para melhorar a vazão do protocolo TCP.

A estratégia natural para melhoria face a uma perda devido ao canal é realizar a retransmissão sem acionar os algoritmos de controle de congestionamento, de forma a manter o tamanho janela de congestionamento constante. Esta estratégia, em conjunto com um dos métodos de classificação do tipo de perda de pacote, são adotadas nas referências (SAMARAWEERA, 1999; BALAKRISHNAN, 1997; GOEL, 1998; FEI, 2000).

3.3.1.2 DESENVOLVIMENTO DE NOVAS VERSÕES DO PROTOCOLO TCP

O avanço das pesquisas na área dos protocolos de rede e a crescente demanda pela interoperação da arquitetura TCP/IP em ambientes sem fio levaram ao desenvolvimento de diferentes versões do protocolo TCP. Tais versões agregaram evoluções dos algoritmos de controle de congestionamento, bem como o uso de reconhecimentos seletivos.

Serão descritas as seguintes versões do protocolo TCP: Tahoe, Reno, New Reno, Sack, Fack e Westwood.

a) TCP Tahoe

É a versão mais antiga e mais rudimentar do TCP, e se caracteriza pela utilização dos seguintes algoritmos: *Slow Start*, *Congestion Avoidance* e *Fast Retransmit*.

A ausência do algoritmo *Fast Recovery* causa o principal problema do TCP Tahoe, pois em situação de congestionamento moderado, configurado pelo recebimento de ACKs duplicados, esta versão aciona o algoritmo *Slow Start*, que abruptamente reduz a taxa de envio dos pacotes.

b) TCP Reno

Como uma evolução do Tahoe, o TCP Reno, descrito na Seção 2.4, incorpora o algoritmo *Fast Recovery* ao *Fast Retransmit*.

Conforme apresentado na Seção 2.4.5, a condição de saída do algoritmo *Fast Recovery* é o recebimento de um ACK do pacote retransmitido. Se houver somente uma perda de pacote na janela, o recebimento do ACK do pacote retransmitido reconhecerá todos os pacotes pertencentes à janela. Entretanto, se houver mais de uma perda de pacotes na mesma janela, o ACK do pacote retransmitido reconhecerá todos os pacotes recebidos até a próxima perda, como ilustrado na FIG. 3.2. Este reconhecimento, que não confirma todos os pacotes enviados pertencentes à mesma janela é denominado reconhecimento parcial.

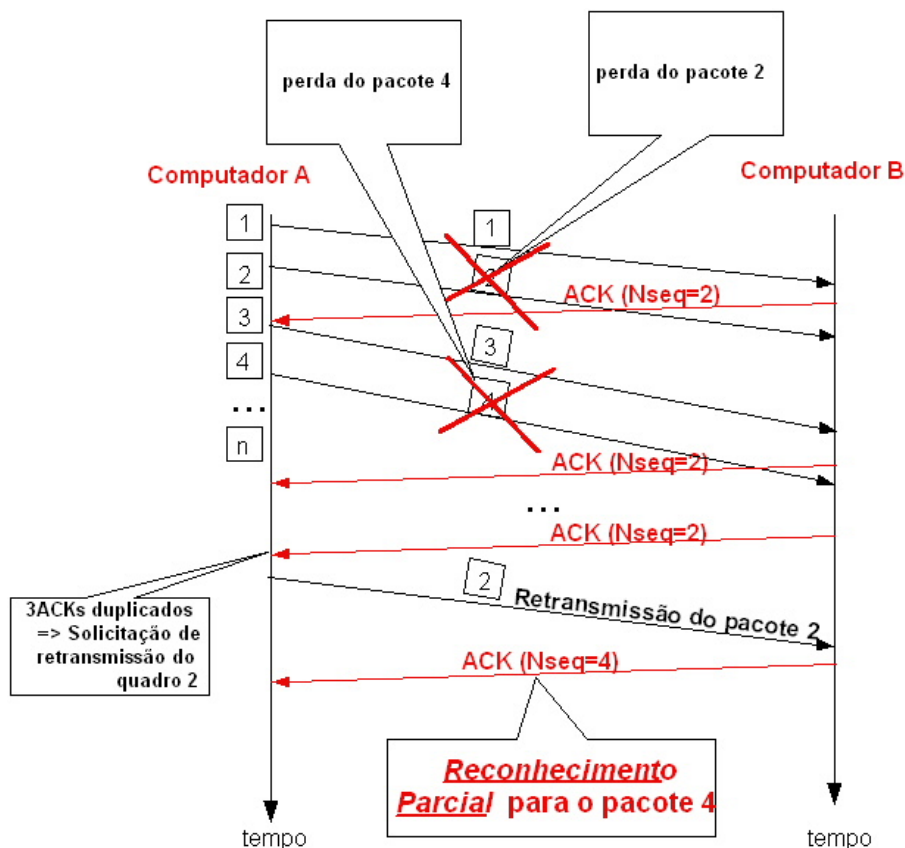


FIG. 3.2: Exemplo de reconhecimento parcial

A cada reconhecimento, o TCP Reno aciona o algoritmo *Fast Recovery*, que divide a janela de congestionamento pela metade. Portanto, em presença de perdas de pacotes em surto, o TCP Reno apresenta desempenho insatisfatório, em função das sucessivas divisões de *cwnd*. Em outras palavras, o TCP Reno não diferencia os reconheci-

mentos parciais recebidos, de forma que, sucessivamente, o protocolo executa o término e o reinício do algoritmo *Fast Recovery*.

Por se tratar da versão do TCP mais difundida e considerada como padrão de fato, os desempenhos das estratégias de melhoria do TCP em canais sem fio serão comparados com o do TCP Reno através de simulações computacionais, cujos resultados serão apresentados no CAP. 6.

c) TCP New Reno

Visando a melhoria de desempenho, o TCP New Reno não permite que ocorram sucessivas divisões da janela de congestionamento, mesmo na presença de várias perdas. Isto se deve ao fato de que esta versão diferencia os reconhecimentos parciais, tratando as múltiplas perdas de pacotes dentro da mesma janela de transmissão na mesma interação do algoritmo *Fast Retransmit*, evitando, assim, os sucessivos términos e reinícios do algoritmo, que geravam as sucessivas reduções da janela de congestionamento.

Implementando o reconhecimento parcial, o algoritmo *Fast Retransmit* do New Reno utiliza a variável *recover* para armazenar o maior número de seqüência dos segmentos transmitidos, de forma que esta versão do TCP somente se retira do *Fast Retransmit* quando for recebido o ACK do pacote retransmitido de maior número de seqüência.

O algoritmo *Fast Retransmit* do New Reno é descrito a seguir:

1. Ao receber 3 ACKs duplicados consecutivos $\implies ssthresh = \frac{cwnd}{2}$;
2. Armazena na variável *recover* o maior número de seqüência dos segmentos transmitidos;
3. Retransmite o pacote supostamente perdido : *Fast Retransmit*;
4. $cwnd = ssthresh + 3$;
5. Se ocorre timeout, $ssthresh = \frac{cwnd}{2}$;
6. Cada ACK duplicado $\implies cwnd = cwnd + 1$;
7. Ao receber o novo ACK do pacote retransmitido uma das ações das duas ações que se segue pode ocorrer:

se o ACK inclui o dado cujo número de seqüência é armazenado em *recover* atribui-se a variável *ssthresh* a variável *cwnd* e termina-se a execução do algoritmo *Fast Recovery*.

se o ACK não reconhece todos os pacotes enviados, o que configura-se um ACK parcial, evita-se a redução de *cwnd* e o término de execução do algoritmo *Fast Recovery*.

d) TCP SACK

Em um cenário de múltiplas perdas de pacotes em uma mesma janela de transmissão, o emissor, após constatar o primeiro pacote perdido através de ACKs duplicados, deverá proceder a retransmissão do mesmo. Após a retransmissão do pacote perdido, o emissor TCP deverá esperar por novos ACKs duplicados para detectar a próxima perda de pacote. Portanto, o protocolo somente pode recuperar um pacote perdido a cada RTT, aumentando consideravelmente o retardo na recuperação dos erros, reduzindo a vazão do protocolo, sobretudo em enlaces com RTT elevados.

Na tentativa de agilizar a recuperação do pacote perdido, foi implementada uma versão do TCP, chamada de TCP SACK, que utiliza um mecanismo de reconhecimento seletivo

No esquema apresentado em (MATHIS, 1996b), as mensagens de reconhecimento possuem tamanhos limitados e podem notificar um conjunto de blocos de dados não contíguos. A referência prevê a utilização de dois tipos de mensagens para a implementação do mecanismo proposto. A primeira delas é a mensagem de habilitação (permissão), denominada "*SACK permitted*", que pode ser enviada na fase de abertura de conexão, e serve para indicar que o reconhecimento seletivo (SACK) pode ser usado durante a fase de transmissão da informação. Nesta fase os terminais concordam ou não com o uso da estratégia de reconhecimento seletivo, de forma que os usuários que não dispõem do mecanismo de reconhecimento seletivo não são prejudicados quando compartilham recursos da rede com conexões que usam esta funcionalidade. Caso a conexão a ser estabelecida tenha permissão para adotar o reconhecimento seletivo, um segundo formato de mensagem é utilizado. Trata-se da mensagem SACK, que é enviada pelo receptor dos dados para relatar o recebimento de pacotes que chegam fora de ordem e, pode ser utilizada enquanto durar a conexão.

A mensagem "*SACK permitted*" é formada por dois bytes. Já a opção SACK possui tamanho variável, sendo em geral mais longa do que a primeira. Vale salientar que, enquanto o receptor não detectar a presença de blocos de dados descontinuados, o reconhecimento dos pacotes é realizado através do mecanismo padrão de reconhecimento acumulativo.

A opção SACK contém uma lista de blocos recebidos. Cada bloco é delimitado por dois números inteiros, sem sinal, de 32 bits (*sequence number* da seqüência de dados). Uma mensagem SACK que especifica N blocos de dados não contíguos possui tamanho

igual a $8 \times N + 2$ bytes, onde 2 bytes representa o início e o fim dos blocos não recebidos. Por questões relacionadas com o tamanho da mensagem de reconhecimento, propõe-se incluir na listagem até 3 blocos de dados. O primeiro bloco de dados da mensagem deve reportar o bloco que contém o pacote que originou a mensagem de reconhecimento, ou seja, o último pacote que chegou com sucesso no receptor, independentemente do seu número de seqüência. Os demais blocos da mensagem são listados de forma arbitrária. É importante ressaltar que, devido à limitação imposta na quantidade de blocos, alguns blocos não contíguos contidos na fila do receptor podem não ser listados. Na realidade, isto não representa um problema sério, devido à redundância criada pela freqüência de geração dessas mensagens, o que ocorre mesmo se forem implementados mecanismos de reconhecimentos atrasados (ALLAN, 1998).

A vantagem do uso de reconhecimento seletivo em enlaces sem fio é que este reconhecimento permite o emissor corrigir as múltiplas perdas de pacotes dentro de uma mesma janela de transmissão. Esta melhoria é mais evidente em canais que apresentam erros em surtos (alta correlação dos erros), como ocorre tipicamente em canais HF.

e) TCP FACK

Apesar do reconhecimento seletivo ser considerado um esquema para recuperar dados, é possível empregá-lo para auxiliar no controle de congestionamento. Tal abordagem foi adotada em (MATHIS, 1996a), onde propôs-se um algoritmo denominado FACK, do termo em inglês *Forward Acknowledment*. O FACK explora a informação fornecida pela opção SACK para promover um melhor controle da injeção de pacotes na rede durante a fase de recuperação dos dados.

Diferentemente do TCP Reno, que tenta estimar a quantidade de dados na rede mediante a contagem de reconhecimentos duplicados, o FACK implementa um procedimento mais elaborado, baseado nas informações contidas na mensagem de reconhecimento seletivo.

Na implementação do FACK, duas variáveis de estado são importantes: *snd.fack* e *retran_data*. A variável de estado *snd.fack* contém o número do seqüência do dado mais avançado recebido com sucesso. Durante as fases de execução do *Slow Start* ou do *Congestion Avoidance*, a variável *snd.fack* possui o mesmo valor da variável de estado *snd.una*, que é utilizada na versão convencional do TCP para indicar o número de seqüência do primeiro byte do pacote enviado e ainda não reconhecido. No entanto,

na fase de recuperação dos pacotes perdidos, essas variáveis passam a conter diferentes números de seqüência. Em particular, a variável de estado *snd.fack* é atualizada com as informações contidas nas mensagens de reconhecimento seletivo.

A estimativa da quantidade de dados em trânsito na rede (*est_dados*) é obtida pelo FACK da seguinte maneira:

$$est_dados = snd.nxt - snd.fack + retran_data, \quad (3.1)$$

sendo que *retran_data* representa uma estimativa da quantidade de dados retransmitidos que ainda não foram reconhecidos e a variável *snd.nxt* contém o *sequence number* do próximo dado a ser transmitido. Cada vez que um pacote é reenviado, o *retran_data* é incrementado pelo tamanho do pacote. Quando se detecta o reconhecimento de um pacote que foi retransmitido, o valor da variável é reduzido pelo tamanho do pacote. Dessa forma, estima-se a quantidade de dados retransmitidos que estão ainda em circulação na rede.

Os algoritmos adotados pelo TCP Reno para identificar e reenviar pacotes perdidos podem produzir um retardo excessivo se vários pacotes forem perdidos em uma janela de dados. O FACK baseia a identificação do pacote perdido no seguinte critério:

$$snd.fack - snd.una > 3 \times MSS \text{ ou } dup_ack = 3, \quad (3.2)$$

sendo que *dup_ack* representa a quantidade de reconhecimentos duplicados, MSS é o tamanho máximo do pacote (COMER, 1995) e *snd_una* é a variável de estado que contém o *sequence number* do primeiro byte de dado ainda não reconhecido.

Se apenas um pacote for perdido, este critério e o adotado pelo TCP Reno operam da mesma forma, ou seja, retransmitem quando *dup_ack* for igual a três. Do contrário, o FACK detecta o pacote perdido com maior rapidez.

Os resultados de simulação computacional obtidos em MATHIS (1996a) indicam que o FACK proporciona uma sensível melhora no desempenho da conexão em relação a versão TCP Reno. Isto é obtido evitando-se retransmissões acionadas pelo *timeout* e pelo rígido controle dos dados em circulação na rede durante a fase de recuperação de pacotes perdidos. Apesar dos bons resultados, os autores são cautelosos e ressaltam a importância de realizar medidas com tráfegos concorrentes.

Obviamente, em relação ao mecanismo convencional (reconhecimento cumulativo), o reconhecimento seletivo aumenta a quantidade de dados em circulação na rede,

em função da intensificação do fluxo no enlace reverso. Além disso, a implementação do esquema de reconhecimento seletivo exige que adaptações sejam feitas tanto no lado de transmissão quanto no de recepção. Tais considerações devem ser ponderadas a fim de estabelecer os reais benefícios proporcionados por esta técnica.

f) TCP Westwood

A grande dificuldade de diversas versões do TCP é a determinação dos parâmetros de controle de congestionamento, após a detecção da perda de pacote, que maximizam a vazão na recuperação das perdas. Por exemplo, conforme apresentado anteriormente, após o recebimento de 3 (três) ACKs duplicados, o TCP Reno reduz pela metade o *cwnd*, podendo esta redução ser insuficiente ou excessiva como resposta ao suposto congestionamento.

Em virtude disso, o TCP Westwood, também chamado de TCPW, propõe selecionar os parâmetros de controle de congestionamento com base na contínua monitoração da taxa de retorno dos ACKs, utilizada na estimação da largura de banda efetiva da conexão. Após cada detecção de perda de pacote esta estimativa é utilizada para a determinação de *cwnd* e *ssthresh* ótimos, iniciando o algoritmo chamado *Faster Recovery*. Este algoritmo é similar ao *Fast Recovery*, porém é mais eficiente na recuperação das perdas de pacotes, devido à maior precisão dos novos valores de *cwnd* e *ssthresh* obtidos com base na estimação da largura de banda da conexão.

Em (CASETTI, 2002) é apresentada uma análise de desempenho do TCPW em redes com canais confinados, em redes sem fio e em redes híbridas. Os resultados deste artigo mostram que em determinadas situações o TCPW pode produzir uma melhoria de vazão de até 550% em relação ao TCP Reno.

3.3.1.3 OTIMIZAÇÃO DOS PARÂMETROS DO PROTOCOLO TCP

Por fim, a última estratégia de melhoria focado no protocolo TCP em ambientes sem fio diz respeito a refinamentos nos parâmetros TCP tais como: tamanho do segmento TCP e *cwnd* inicial.

Conforme mencionado anteriormente, o TCP foi criado para redes com canais confinados que apresenta características de baixas taxas de erro de bit. Nestas condições, pode-se adotar tamanhos de pacotes IP grandes com baixos valores de perda de pacote IP. Por exemplo, o tamanho padrão do pacote IP na Internet é 576 bytes.

Por outro lado, em canais sem fio que possuem altas taxas de erros, a escolha do tamanho do pacote IP torna-se relevante, tendo em vista que pacotes IP grandes podem produzir elevados valores de perda de pacotes, por outro lado tamanhos muito pequenos elevam o *overhead* do pacote. Portanto, a escolha criteriosa do tamanho do pacote é fundamental para melhoria do TCP em ambientes sem fio.

Por questões discutidas previamente, no início de uma conexão, o TCP adota uma janela de transmissão composta por apenas um segmento ($cwnd = 1$). Isto determina a utilização de uma taxa de transmissão baixa, que tende a ser aumentada ao longo da transmissão de acordo com a operação do algoritmo *Slow Start*. Alguns trabalhos envolvendo canais com longos retardos propõem iniciar uma conexão TCP com uma janela de transmissão constituída de uma quantidade maior de segmentos. Isto permite reduzir o tempo de execução do algoritmo *Slow Start* e, por conseguinte, o desperdício ocorrido neste período.

3.3.2 ESTRATÉGIAS QUE MODIFICAM A SEMÂNTICA DO TCP

Conforme apresentado no Seção 2.4, o TCP é o protocolo especialmente projetado para comunicação fim-a-fim, ou seja, as trocas de informações são feitas do terminal de origem para o terminal de destino. No entanto, com objetivo de obter melhoria da vazão em canais sem fio, foram propostas duas técnicas que modificam esta semântica fim-a-fim: *TCP Spoofing* e *Splitting TCP Connections*.

3.3.2.1 TCP SPOOFING

A técnica utilizada para aumentar a eficiência da transmissão de dados através de enlaces sem fio denominada "*TCP Spoofing*" é discutida em (ETAL., 1995). De acordo com esta técnica, o roteador que antecede o enlace sem fio, correspondendo ao *Gateway* entre a rede convencional e o canal sem fio, armazena e reconhece os pacotes recebidos, como se fora o destinatário da informação. Essa estratégia, do ponto de vista do transmissor dos dados, cria a ilusão de uma conexão com um RTT pequeno. Como consequência a taxa de transmissão aumenta rapidamente, conforme ilustrado na FIG. 3.3.

Este procedimento, apesar de permitir o empacotamento dos dados de forma mais adequada para a transmissão através do enlace sem fio, possui alguns inconvenientes. Em primeiro lugar, aumenta-se sobremaneira a quantidade de processamento no roteador, além dos requisitos de armazenamento de dados. Em segundo lugar, o roteador deve se

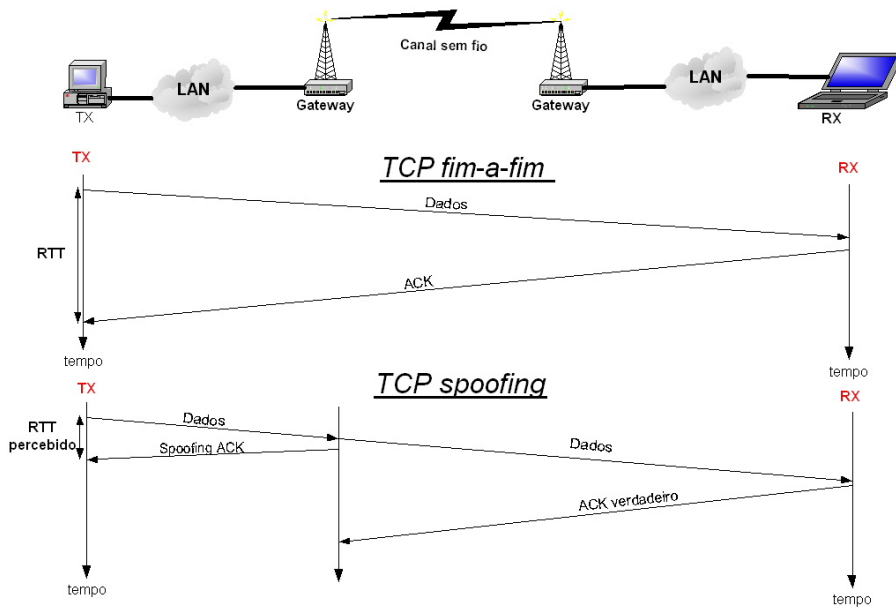


FIG. 3.3: Técnica *Spoofing*

encarregar de realizar retransmissões dos pacotes recebidos com problemas ao seu destino. Por fim, este esquema é bastante susceptível a falhas. Caso ocorra algum problema que implique a perda irrecuperável dos dados, existe um problema sério, pois, do ponto de vista do transmissor, a transmissão foi bem sucedida.

As simulações apresentadas em (BALAKRISHNAN, 1997) mostraram que, a despeito do alto *overhead* requerido pelo processamento de dois protocolos nos *gateways*, esta estratégia resulta em um desempenho pior do que o obtido com o uso dos protocolos de enlace.

3.3.2.2 SPLITTING TCP CONNECTIONS

Esta técnica consiste na divisão de uma conexão fim-a-fim em duas ou três conexões TCP intermediárias. Assim, cada conexão é, por si só, uma conexão TCP fim-a-fim. A seqüência de dados é encaminhada de uma conexão a outra, sendo armazenada nos nós intermediários se necessário. O *Splitting TCP* é, em geral, mais aplicado em redes satélites, sendo ilustrado na FIG. 3.4.

3.3.3 SOLUÇÕES FOCADAS NOS PROTOCOLOS DA CAMADA DE ENLACE

As soluções que visam otimizar os protocolos da camada de enlace têm por objetivo prover a confiabilidade necessária por meio do emprego dos seguintes mecanismos:

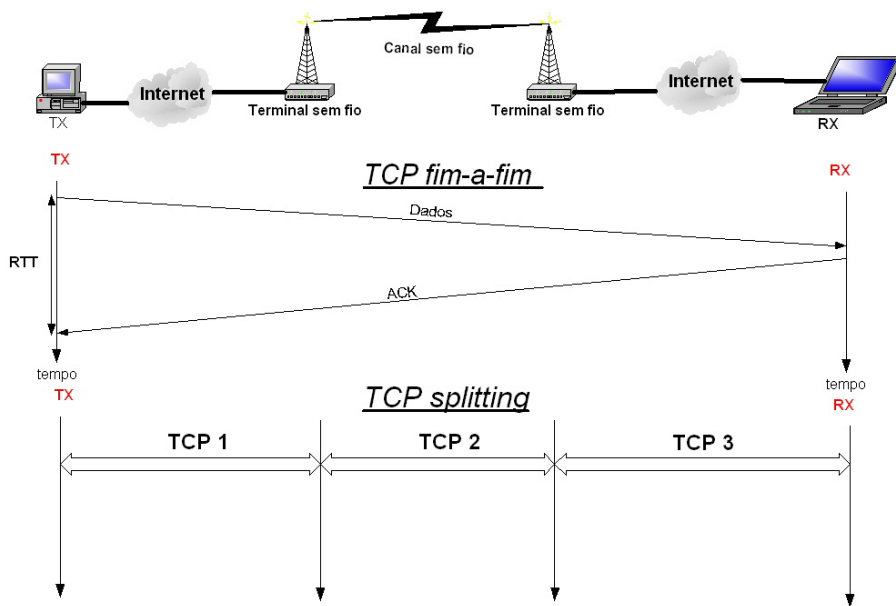


FIG. 3.4: Técnica de *Splitting*

- reconhecimento local automático dos quadros chamado de *Automatic Repeat Request* - ARQ;
- detecção de erros;
- controle de fluxo e o
- uso de técnicas adaptativas que associam as condições da camada física aos parâmetros do protocolo de enlace.

Desta forma, tais mecanismos podem ocultar para o TCP as perdas de pacotes devido a erros no canal. Esta estratégia de melhoria é interessante, na medida que não requer códigos adicionais em relação à pilha de protocolos TCP/IP nos terminais e/ou nos roteadores.

3.3.3.1 REPETIÇÃO SELETIVA

Os protocolos de enlace utilizados em canais com alta taxa de erros de bits implementam, com o uso de técnicas de codificação de canal, mecanismos ARQ para garantir a entrega dos quadros. Basicamente, tais mecanismos são subdivididos em *Go-Back-N* (GBN), repetição seletiva (*Selective Repeat* - SR) e mecanismos híbridos com GBN e repetição seletiva.

O mecanismo de GBN foi descrito na Seção 2.4.3. A repetição seletiva é implementado da seguinte maneira:

1. O receptor realiza a detecção de erros ou perda do quadro;
2. O receptor utiliza os quadros NACK (*Not Acknowledgment*) para informar o quadro com problemas;
3. O transmissor ao receber o NAK, imediatamente, retransmite o quadro indicado pelo NAK;
4. O procedimento é repedido até que o quadro seja recebido adequadamente. Se número de retransmissões do mesmo quadro for superior ao limite da persistência, o protocolo repassa a tarefa de retransmissão do pacote inteiro para camada de transporte.

O mecanismo ARQ-SR é mais eficiente, pois evita as retransmissões desnecessárias observados, por exemplo, no mecanismo ARQ-GBN. A desvantagem do uso do ARQ-SR é o acúmulo de retransmissões do protocolo de enlace e do TCP, causados por temporizadores incompatíveis nas referidas camadas. De fato, o requisito mínimo para obtenção de desempenho satisfatório do TCP em presença de protocolo de enlace com ARQ é $timeout_{TCP} > timeout_{enlace}$.

Portanto, a presença de ARQ-SR causa aumento do RTT, que pode deteriorar a vazão TCP. Por outro lado, a recuperação local dos quadros evita as perdas de pacote e o acionamento dos usuais mecanismos de controle de janelas. Desta forma, o ARQ-SR permite ao TCP manter o *cwnd* com valores maiores, melhorando a vazão.

3.3.3.2 PERSISTÊNCIA DO PROTOCOLO

As retransmissões de um mesmo quadro usando o mecanismo ARQ-SR são realizadas por um número máximo de tentativas chamado de persistência do ARQ-SR e denotado por δ . Ou seja, para $\delta = 3$ significa que, após 3 (três) retransmissões do mesmo quadro, o ARQ-SR assumirá que o quadro não pode ser recuperado localmente, passando a tarefa de retransmitir o segmento inteiro para o TCP. Para $\delta = 0$ significa que os quadros com problemas não serão retransmitidos.

Em (ABDELMOUMEN, 2004) mostra-se que o desempenho do TCP melhora com o aumento da persistência do ARQ-SR, exceto para alguns casos extremos onde os canais possuem baixas taxa de perda de pacote e grande retardo submetidos a transfe-

rência de arquivos pequenos via FTP.

3.3.3.3 USO DA CORREÇÃO DE ERROS - FEC

As estratégias que utilizam FEC fazem uso do acréscimo de redundância à informação a ser transmitida, para detectar e corrigir os erros ocorridos durante a transmissão.

Um código comumente aplicado em comunicações digitais é o chamado código convolucional binário apresentado (VITERBI, 1971) com taxa k/n , onde k é o quantidade de informação em bits a ser alimentada no codificador e n é o tamanho da palavra código em bits gerada pelo codificador (FIG. 3.5), resultando em uma redundância de $n - k$ bits. Cada código convolucional binário de taxa k/n está associado um parâmetro chamado de distância livre, d_{free} , o qual está relacionado com a capacidade de detecção e correção de erro do código.

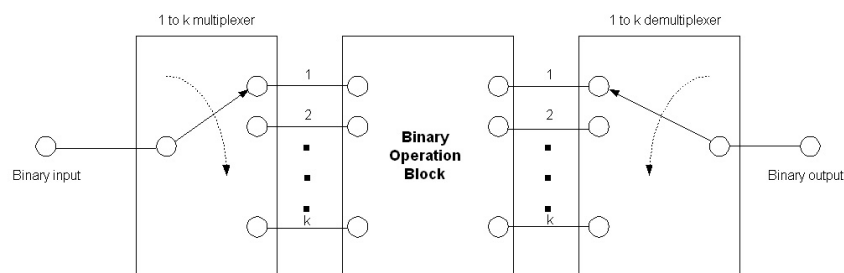


FIG. 3.5: Diagrama em blocos simplificado de um Codificador de canal Binário.

O FEC é implementado na camada física, porém alguns padrões de camada de enlace fazem uso de mecanismos de controle FEC adaptativos, nos quais os números de bits redundantes, ou a taxa de código $\frac{k}{n}$, variam de acordo com o estado da camada física, conforme ilustrado na TAB. 3.1.

A vantagem do uso da estratégia do FEC é melhorar a eficiência na detecção e recuperação de erros de bits, evitando, em alguns casos, a retransmissão do quadro correspondente. A desvantagem desta estratégia é o consumo extra de largura de banda para transmitir informações redundantes.

3.3.3.4 PROTOCOLOS DE ENLACE PARA HF

Nesta seção apresentaremos uma descrição de duas arquiteturas de protocolos de enlace, comumente, utilizados em comunicações HF: os padrões STANAG (e MIL-STD)

e o protocolo AX.25.

a) Padrões STANAG

A Organização do Tratado do Atlântico Norte (OTAN) desenvolveu famílias de padrões de camada de enlace e física para equipamentos rádio de emprego militar dentro de um conceito chamado HF integrado (*HF-House*). Vários padrões da OTAN chamados de STANAG possuem correspondentes americanos, chamados U.S. MIL-STD, como por exemplo, o STANAG 4538 tem como correspondente o MIL-STD-188-141B (HF 3G).

O conceito HF-House abrange duas famílias de tecnologia: a 2ª Geração de sistemas HF (HF 2G) e a 3ª Geração de sistemas HF (HF 3G). Ambas estabelecem procedimentos automáticos na camada de enlace e física, tipos de MODEMs apropriados à camada física e interfaces para rede convencional TCP/IP. Encontra-se em operação, há alguns anos, a tecnologia 2G, porém a mesma não deve ser considerada obsoleta diante da tecnologia 3G. Além disso, ambas as tecnologias deverão coexistir durante um certo tempo, pois podem ter diferentes aplicações em sub-redes HF.

As configurações operacionais comuns de um sistema HF 2G são baseadas nos seguintes padrões:

- STANAG 5066 para protocolo de camada de enlace, incluindo mecanismo ARQ;
- MIL-STD-188-110B⁷ (MIL-STD-188-110B, 2000) ou STANAG 4539 para MODEMs.

O protocolo da camada de enlace STANAG 5066 utiliza as técnicas de repetição seletiva e fluxo duplex de dados.

Por outro lado, a tecnologia HF 3G inclui os seguintes padrões:

- HDL+ ARQ e LDL ARQ como protocolos de camada de enlace, com mecanismos ARQ;
- MIL-STD-141B Apêndice C ou STANAG 4538 para MODEMs.

Os protocolos de enlace HDL+ ARQ e LDL ARQ são conhecidos como XDL. O 3G HDL+ utiliza técnicas relacionadas com repetição seletiva, integradas com mecanismos

⁷É um documento emitido pelo Departamento de Defesa americano que estabelece padrões técnicos necessários para assegurar a interoperabilidade e definir exigências de desempenho mínimo para MODEMs de dados em HF

TAB. 3.1: Parâmetros de forma de onda do protocolo HDL+.

<i>Waveform Options</i>	Constelação de sinais	Taxa de FEC
0	2-PSK	R 1/2
1	2-PSK	R 1/2 (<i>Punctured</i>)
2	4-PSK	R 3/4 (<i>Punctured</i>)
3	8-PSK	R 3/4 (<i>Punctured</i>)
4	16-QAM	R 3/4 (<i>Punctured</i>)
5	64-QAM	R 3/4 (<i>Punctured</i>)
6	64-QAM	R 1/2 (<i>Alt. Phases</i>)
7	Reservado	Reservado

adaptativos de taxa de transmissão, bits de sincronização e FEC. No entanto, o protocolo 3G LDL não inclui a técnica de repetição seletiva.

Vale ressaltar que as famílias de tecnologias 2G e 3G são incompatíveis e, portanto, não interoperáveis.

Recentes investigações integraram as técnicas avançadas de *Code Combining*⁸ dos protocolos XDL (STANAG 4638) e a família de MODEMs de altas taxas de bits baseado no STANAG 4539, produzindo um novo protocolo denominado de HDL+ (CHAMBERLAIN, 2003).

Este protocolo também apresenta um mecanismo adaptativo que integra a técnicas de ARQ-SR, FEC e *Code Combining*, capaz de entregar dados sem erro a uma taxa superior a 10 Kbps em canal de 3 KHz (CHAMBERLAIN, 2003). Contudo, não inclui procedimentos de abertura de conexão, sendo os parâmetros de taxa de FEC e constelação de sinais definidos durante a transmissão dos quadros. A TAB. 3.1 apresenta um exemplo da capacidade de adaptação do protocolo em relação ao estado do canal, especificando as opções do campo "*waveform options*" do cabeçalho do quadro:

b) Protocolo AX.25

A comunidade internacional de rádio-amadorismo sentiu a necessidade e o desejo de definir um protocolo com capacidade de entregar dados de forma confiável entre

⁸Refere-se a técnica de controle de erros adaptativa de camada física que melhora a detecção e correção de erros. Nesta técnica, as várias recepções de um mesmo bloco de dados codificado são conjuntamente decodificados ou "combinados" na estação receptora. Quanto maior o número a repetição dos blocos, maior é a capacidade de detecção e correção dos erros. De maneira adaptativa, o transmissor ajusta o número de repetições de cada bloco em função das informações contidas nos quadros de reconhecimento.

terminais em uma rede de comutação de pacotes via rádio HF. Com este fim, o AX.25, que corresponde a uma variação do protocolo X.25, foi padronizada pela *American Radio Relay League* (ARRL). Este protocolo é embutido em um dispositivo chamado *Terminal Node Control* (TNC).

O AX.25 especifica três tipos de quadros que são informados no campo "Control" do cabeçalho do quadro (FIG. 3.6):

- Quadro de informação (*Information frame - I frame*);
- Quadro de Supervisão (*Supervisory frame - S frame*);
- Quadro Não Numerável (*Unnumbered frame - U frame*).

Os *I-frames* são utilizados para o transporte dos dados do usuário e seu formato é mostrado na FIG. 3.6, onde os campos são especificados abaixo:

Flag	Address	Control	PID	Info	FCS	Flag
01111110	112/224 Bits	8/16 Bits	8 Bits	N*8 Bits	16 Bits	01111110

FIG. 3.6: Formato do quadro AX.25

- **Flag** - campo de 1 byte responsável em delimitar o quadro AX.25.
- **Campo de Endereço** - Identifica o fonte e o destino do quadro.
- **Campo de controle** - identifica os tipos de quadros e controla todos os atributos do AX.25.
- **Campo PID** (*Protocol Identification*) - identifica os protocolos de camadas superiores, conforme a estrutura apresentada na FIG. 3.7.
- **Campo de Informação** - Contêm os dados do usuário.
- **Frame Check Sequence - FCS** - Seqüência de bits utilizado pelo CRC-16 ("*Cyclic Redundancy Check*") na detecção de erros de bits no quadro.

A função de prover a confiabilidade e o controle do enlace HF são realizados pelos *S-frames* através de mensagens de reconhecimento (*Receive Ready - RR*), não reconhecimento e solicitação de retransmissão (*Reject - REJ* ou *Selective Reject - SREJ*).

HEX	M S B	L S B	Translation
**	yy01	yyyy	AX.25 layer 3 implemented.
**	yy10	yyyy	AX.25 layer 3 implemented.
0x01	00000001		ISO 8208/CCITT X.25 PLP
0x06	00000110		Compressed TCP/IP packet. Van Jacobson (RFC 1144)
0x07	00000111		Uncompressed TCP/IP packet. Van Jacobson (RFC 1144)
0x08	00001000		Segmentation fragment
0xC3	11000011		TEXNET datagram protocol
0xC4	11000100		Link Quality Protocol
0xCA	11001010		Appletalk
0xCB	11001011		Appletalk ARP
0xCC	11001100		ARPA Internet Protocol
0xCD	11001101		ARPA Address resolution
0xCE	11001110		FlexNet
0xCF	11001111		NET/ROM
0xF0	11110000		No layer 3 protocol implemented.
0xFF	11111111		Escape character. Next octet contains more Level 3 protocol information.
Escape character. Next octet contains more Level 3 protocol information.	00001000		

FIG. 3.7: Estrutura do campo PID do quadro AX.25.

Os *U-frames* são utilizados em transmissões não orientados a conexão sendo responsáveis pelos seguintes mecanismos de controle e gerência da conexão na camada de enlace: abertura e término de conexão, negociação de parâmetros e os reconhecimentos necessários para prover tais mecanismos. Este tipo de quadro também é utilizado para transmissões do tipo *multicast* e *broadcast*. Por exemplo, nas operações chamadas de "*round-table*", onde vários rádio-amadores compartilham os mesmos quadros.

Diferentemente dos protocolos de enlace mencionados anteriormente, o AX.25 pode exercer funções referentes às camadas de enlace e de rede conjuntamente, conforme especificado nas duas primeiras linhas da FIG. 3.7. Desta forma, as funções da camada de rede são traduzidas no grande cabeçalho AX.25, que possui 20 bytes (FIG. 3.6), em função do campo de endereçamento de 14 bytes que pode ser utilizado nas operações de repetidor ("*digipear*").

Resumindo, o protocolo de enlace AX.25 possui as seguintes características:

- realiza os mecanismos ARQ-SR, ARQ-GBN e ARQ híbrido SR e GBN;

- emprega o mecanismo ARQ-GBN, através dos quadros REJ;
- emprega o mecanismo ARQ-SR, através da recepção dos quadros SREJ;
- emprego da janela de transmissão e de recepção constantes;
- especifica procedimentos de transmissão tanto orientados à conexão quanto não orientado a conexão;
- especifica dois modos de transmissão: half-duplex e full-duplex;
- permite o estabelecimento de múltiplos enlaces, de acordo com a recomendação CCITT Q.921 (LAPD), que são distinguíveis por meio do campo de endereço.

A especificação não inclui as técnicas adaptativas de *Code Combining*.

Conforme mencionado na Seção 2.3, vários protocolos usuais da camada de aplicação podem ter baixo desempenho em canais HF. Por esta razão, a comunidade HF tem definido alguns protocolos similares inteiramente voltadas para o canal HF, resultando em desempenhos melhores em relação às estratégias focadas no TCP e na camada de enlace. Por exemplo, as aplicações HMTP, que são variantes do protocolo *Simple Mail Transport Protocol* (SMTP) da arquitetura TCP/IP, utilizam diretamente os protocolos de camada de enlace. No entanto, esta estratégia encontra-se fora do escopo desta dissertação, levando-se em consideração a necessidade de interoperação das sub-redes HF com as redes convencionais baseadas na pilha de protocolos TCP/IP, de forma a compartilhar as mesmas aplicações.

3.4 RESUMO

Neste capítulo caracterizamos a baixa vazão do protocolo TCP em canais sem fio. Também apresentamos diversas estratégias para obtenção de melhorias da vazão do TCP em tais canais, resumido em 3 (três) estratégias principais: modificações e inclusão de novos algoritmos no protocolo TCP, estratégias que modificam a semântica fim-a-fim do TCP, e uso e otimização dos protocolos da camada de enlace.

No Capítulo 4 apresentaremos as principais características das comunicações em canal HF, tendo em vista ser este o canal utilizado para as avaliações de desempenho do TCP na presente dissertação.

4 CARACTERÍSTICAS DAS COMUNICAÇÕES EM CANAL HF

4.1 INTRODUÇÃO

As transmissões de rádio nas frequências entre 3 e 30 MHz são chamadas de comunicações em HF. A banda HF é largamente utilizada para comunicações de longas distâncias, também chamadas de comunicações do tipo "*Beyond Line-of-Sight*" (BLOS), onde o mecanismo de propagação intrínseco utiliza a refração de ondas em camadas ionosféricas.

As camadas ionosféricas apresentam uma grande variação das condições de propagação, resultando em um canal com altas taxas de perdas de bits. Além disso, limitação de banda (3kHz, tipicamente) inerente aos sistemas em HF leva a comunicações com baixas taxas de transmissão ($\leq 10Kbps$).

Neste capítulo serão descritas algumas características da propagação de sinal em meios ionosféricos. Discutiremos também alguns modelos para descrever os erros que ocorrem nas camadas físicas de sistemas de comunicações que operam em canais de HF.

4.2 PROPAGAÇÃO EM HF

As comunicações BLOS em HF são possíveis devido à existência da camada ionosférica, que se estende aproximadamente de 65 a 1000 Km acima da superfície da Terra e se caracteriza pela ocorrência da ionização causada pela radiação solar.

Nas comunicações via ionosfera as ondas de rádio penetram nas camadas ionizadas e, dependendo da frequência, algumas são absorvidas, outras atravessam a ionosfera, e outras ainda são refratadas e retornam à superfície da Terra, conforme ilustrado na FIG. 4.1. Vale ressaltar que, sob condições adequadas, a onda eletromagnética retorna a outro ponto da superfície como se fosse refletida pela ionosfera, mas na verdade ocorre um processo de refração (ver FIG. 4.1).

Na ionosfera, há quatro camadas de ionização ilustradas na FIG. 4.2. As camadas ionosféricas caracterizam-se basicamente pela altura e pela densidade de elétrons livres oriunda da ionização. Como a ionização é causada pela radiação solar, as camadas mais altas da ionosfera tendem a ser mais densas, enquanto as camadas inferiores, prote-

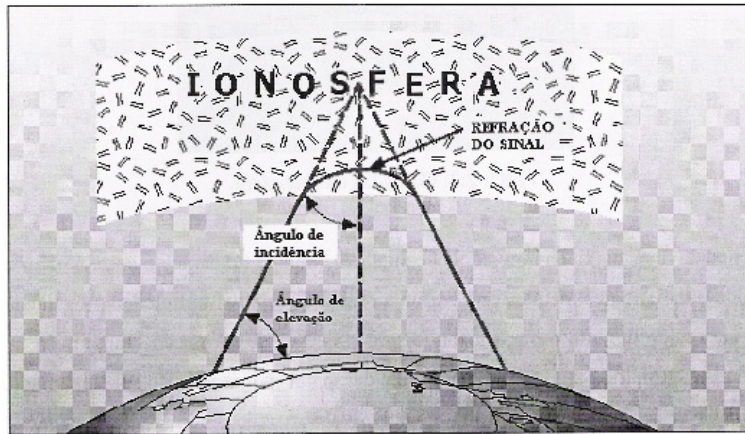


FIG. 4.1: Propagação da onda eletromagnética em HF

gidas pelas superiores, recebem menos ionização. A primeira camada foi descoberta em 1920 por Appleton, que usou a letra E para defini-la, simbolizando o vetor campo elétrico. Posteriormente, as camadas D e F foram descobertas.

A camada mais baixa da ionosfera, a D, situada entre 65 Km e 90 Km de altitude para latitudes médias, e atua na propagação ionosférica de forma negativa, absorvendo energia eletromagnética sem contribuir para o retorno dos sinais à superfície da Terra.

A camada E, normalmente situada entre 90 e 140 Km, é a primeira útil para propagação das ondas de rádio. Sua intensidade varia conforme a posição do sol, atinge o ápice ao meio-dia, apresenta rápida queda no pôr do sol e atinge a mínima atividade à meia-noite.

A camada F situa-se tipicamente entre 140 e 1000 Km. É a camada que apresenta maiores taxas de ionização. Durante o dia, no verão, nos picos de ionização, a camada F pode se subdividir em duas outras. Uma das camadas, a F1, aparece normalmente de 140 até 200 Km de altitude (BOITHIAS, 1987). De 200 até 1000 Km, tem-se a camada F2.

As melhores condições de propagação são obtidas através da camada F. Esta camada apresenta uma grande estabilidade, devido à lenta recombinação dos elétrons na região, tornando-a razoavelmente insensível às variações da radiação solar no decorrer do dia, inclusive mantendo sua eficiência durante a noite. Essa camada é responsável pelas maiores distâncias de propagação, de forma que com apenas um processo de reflexão (reflexão simples) pode-se obter comunicações entre distâncias de até aproximadamente

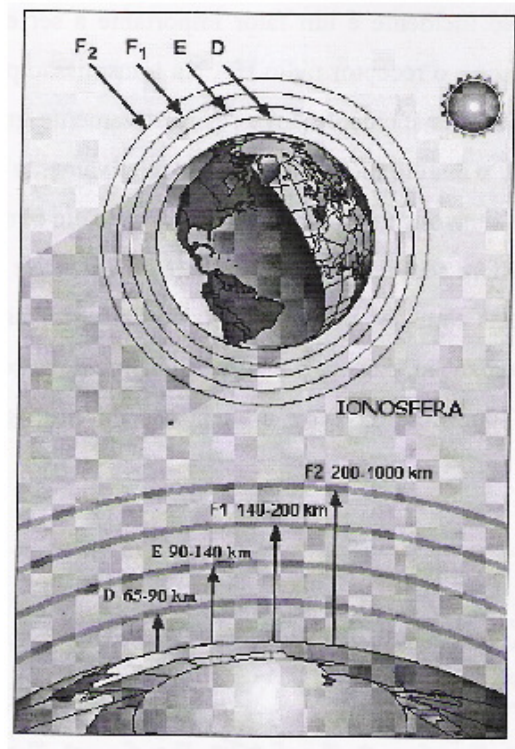


FIG. 4.2: Camadas da ionosfera

4000 Km.

Além das camadas ionosféricas, a própria superfície da Terra pode refletir ondas eletromagnéticas provenientes da camada ionosférica, de forma que uma onda refletida de volta a partir do ponto de sua incidência na Terra pode sofrer uma segunda reflexão ionosférica, e assim por diante. A FIG. 4.3 ilustra algumas situações possíveis de múltiplas reflexões.

A ionosfera, como meio de propagação de ondas eletromagnéticas, é dispersivo, ruidoso, dinâmico e aleatório. O caráter dispersivo deve-se ao efeito de multipercurso e sua característica dinâmica resulta no espalhamento Doppler. Em função disso, o canal HF em geral produz elevada taxa de erros, bem como erros em surtos ("*bursty*").

4.3 TRANSMISSÃO EM HF

As transmissões digitais em HF estão sujeitas a variações das condições de propagação em diversas escalas de tempo e, em geral, se caracterizam pelas baixas taxas de transmissão que, em presença de canais com altas taxas de erros, produzem enlaces com altos retardos.

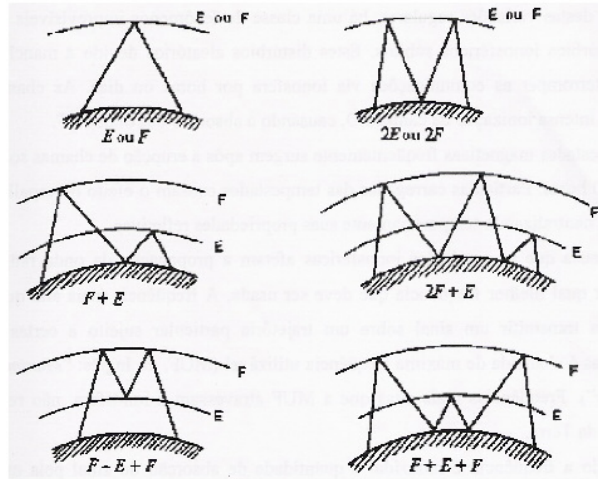


FIG. 4.3: Trajetórias da onda eletromagnética na ionosfera

O avanço nas pesquisas em torno da propagação ionosférica e o rápido desenvolvimento do processamento digital de sinais durante a última década, tornaram possível uma substancial melhora nas comunicações em canal HF, permitindo, por exemplo, que modernos MODEMS, com mecanismos adaptativos em relação à camada física, transmitam a uma taxa de até 9600 bps em canais de 3KHz.

4.4 MODELAGEM DO PROCESSO DE ERROS DE BITS EM HF

Estudos mostram que o desempenho de vários protocolos de camada de enlace e na camada de transporte, depende do modelo de erro adotado para camada física. Em (ZORZI, 1999) é apresentada uma compilação de recentes trabalhos envolvendo modelagem de erro, revelando que as características estatísticas de alta ordem dos erros de um canal tem significativo impacto no desempenho de protocolos de comunicação e que correlações de erros de canais podem e devem ser consideradas nessas avaliações de desempenho.

A simples especificação da taxa de erro médio não fornece informação suficiente para avaliação do comportamento do protocolo. Por exemplo, foi demonstrado em (DIETRICH, 1996) que a probabilidade de um bloco de bits ser recebido com sucesso pode dobrar de acordo com auto-correlação do processo de erros, para a mesma taxa média de erros. Isto nos leva a concluir que o entendimento estatístico e o uso de uma modelagem precisa são fundamentais para a realização de um estudo completo sobre desempenho.

Uma ferramenta matemática muito usada na modelagem de processos com

memória é o processo de Markov, devido ao fato de combinar a facilidade de análise matemática com a capacidade de reproduzir a correlação desses processos. Em (ZORZI, 2003) é mostrado que um modelo de Markov com 2 (dois) estados pode ser empregado para representar as perdas de pacotes devido ao estouro de *buffer* em sistemas com recursos limitados. Embora esses modelos sejam mais completos do que o modelo de erros independentes e identicamente distribuídos (*iid*), recentes estudos (ZORZI, 1999) têm mostrado que eles ainda são inadequados para representar alguns canais variantes no tempo e para o estudo de algumas métricas de desempenho.

Modelos de Markov mais elaborados empregam uma quantidade maior de estados. Eles permitem, em geral, descrever com maior precisão os processos, contudo, se caracterizam por possuírem maior complexidade de análise e de ajuste dos seus parâmetros. Vale mencionar que em determinadas aplicações o modelo com 2 ou 3 estados têm sido muito empregado por conferir um bom compromisso entre esses dois aspectos conflitantes.

Fritchman (FRITCHMAN, 1967) investigou um modelo de cadeia de Markov com vários estados particionados em dois grupos, correspondendo aos estados "livres de erros" e estados com a presença de erros. Os modelos de Fritchman têm sido aplicados por vários pesquisadores para representar as seqüências de erros obtidas em sistemas de transmissão que operam sobre canais que apresentam o efeito de desvanecimento.

Particularizando para camadas físicas em canal HF, conforme descrito nas seções anteriores, a propriedade da variação das condições de propagação ionosféricas nos leva a buscar um modelo de erros que represente não somente a taxa de erros desejada, mas, também, a estrutura de correlação do canal HF, de forma a melhor representar os erros em surtos.

Em função disso e das assertivas acima mencionadas, o modelo de erros adotado nos sistemas de comunicação em canal HF para avaliação do desempenho dos protocolos de comunicação investigados é o Modelo de Markov Escondido (*Hidden Markov Model* - HMM) com 3 (três) estados. Nesta modelagem os erros do canal HF⁹ são regidos por um processo de erros de bits correlacionados com uma determinada probabilidade de erros alvo. O modelo HMM será melhor detalhado na seção subsequente.

Por outro lado, muitos trabalhos (ABDELMOUMEN, 2004) avaliam o de-

⁹Nesta dissertação, os erros do canal HF referem-se, na verdade, a erros do sistema de comunicação em canal HF gerados de acordo com os modelos apresentados.

sempenho de protocolos de comunicações em sistemas que geram erros em surto usando modelos de erros independentes. Assim sendo, para fins de análise da influência da modelagem de erros no desempenho dos protocolos das camadas superiores em canais HF, será apresentado também o modelo de erros independentes, apesar desse modelo não ser apropriado para descrever os erros gerados nos sistemas de comunicação em canal HF. A descrição do modelo será feita nas próximas seções.

4.4.1 ERROS CORRELACIONADOS: MODELO HMM

O Modelo HMM é formado por um conjunto finito de estados, em cada qual associa-se uma distribuição de probabilidade que governa a geração do processo que se deseja modelar com o HMM. Assim sendo, para cada particular estado, um resultado é gerado de acordo com a distribuição de probabilidade a ele associado. As transições entre os estados do modelo são governadas por um conjunto de probabilidades chamadas de probabilidades de transição de estado.

O termo "escondido" advém do fato que, a evolução dos estados não determina a função amostra do processo modelado.

Para definir completamente o modelo HMM com n estados empregado, os seguintes elementos são necessários:

- a) Número de estados do modelo: n
- b) Número de resultados gerados em cada estado.

Em cada estado são gerados um número de resultados N com uma determinada distribuição de probabilidade. Por exemplo, no processo de erros assume-se instantaneamente 2 resultados possíveis para designar a presença de erro e a sua ausência.

- c) Conjunto das probabilidades de transição de cada estado.

Denotado por p_{ij} define a probabilidade de se estar no estado i no instante t e no estado j no instante $t + 1$ representado matematicamente pela EQ. 4.2.

$$p_{ij} = P\{est_{t+1} = j | est_t = i\}, \quad 1 \leq i, j \leq n. \quad (4.1)$$

As probabilidades de transição p_{ij} devem satisfazer:

$$p_{ij} \geq 0, \quad 1 \leq i, j \leq n \quad (4.2)$$

e

$$\sum_{j=1}^n p_{ij} = 1, 1 \leq i \leq n. \quad (4.3)$$

Agrupando as probabilidades de transição em uma matriz resulta na chamada Matriz de probabilidades de transição de estados \mathbf{M} dada por:

$$\mathbf{M} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & \dots & p_{1j} & \dots & p_{1n} \\ p_{21} & p_{22} & p_{23} & \dots & p_{2j} & \dots & p_{2n} \\ p_{31} & p_{32} & p_{33} & \dots & p_{3j} & \dots & p_{3n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ p_{i1} & p_{i2} & p_{i3} & \dots & p_{ij} & \dots & p_{in} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ p_{n1} & p_{n2} & p_{n3} & \dots & p_{nj} & \dots & p_{nn} \end{pmatrix} \quad (4.4)$$

na qual, i ésima linha estão registrados todas as possíveis transições a partir do estado i , e na j ésima coluna as transições para o estado j .

d) Distribuição de probabilidade de cada estado.

Em cada estado é gerado um processo aleatório de resultados governada por uma determinada função densidade de probabilidade. Como o processo a ser modelado é a ocorrência de erros, logo, associado a cada estado i , encontra-se a probabilidade de erro do processo denotado por p_i .

Portanto, as probabilidades dos erros de cada estado denotado por $p_1, p_2, p_3, \dots, p_i, \dots, p_n$, e a matriz das probabilidades de transição de estados \mathbf{M} são parâmetros fundamentais do modelo HMM e definem o diagrama de estados da cadeia de Markov ilustrada na FIG. 4.4:

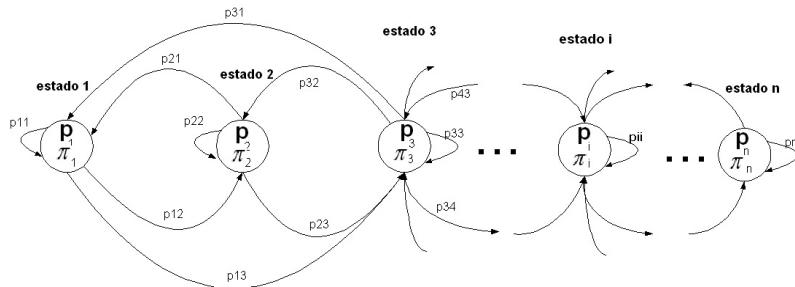


FIG. 4.4: Diagrama de estados do Modelo HMM

e) Estado inicial

A escolha do estado inicial é o último dos elementos de definição do modelo HMM.

Supondo que a cadeia de Markov é estacionária, podemos empregar a seguinte equação matricial:

$$\pi = \pi \cdot \mathbf{M}, \quad (4.5)$$

onde $\pi = (\pi_1, \pi_2, \dots, \pi_i, \dots, \pi_n)$ é o vetor probabilidade em regime estacionário, cujos elementos descrevem a probabilidade dos estados e satisfazem a seguinte equação:

$$\sum_{i=1}^n \pi_i = 1. \quad (4.6)$$

Com as definições das probabilidades de erro associada a cada estado e as probabilidades dos estados, pode-se obter a probabilidade de erro a ser atingido pelo processo gerado pelo modelo (Probabilidade de erro alvo - P_{alvo}) da seguinte maneira:

$$P_{alvo} = p_1 \cdot \pi_1 + p_2 \cdot \pi_2 + p_3 \cdot \pi_3 + \dots + p_i \cdot \pi_i + \dots + p_n \cdot \pi_n. \quad (4.7)$$

Portanto, a utilização da Matriz das probabilidades de transição dos estados que corresponde à alternância, ao longo do tempo, dos processos de erros resultam em um modelo de erros com memória e representativo dos erros em surtos.

O modelo adotado nesta dissertação para modelar os erros de bit ocorridas na camada física foi o HMM com 3 estados, onde para cada um dos estados, derivamos uma seqüência de variáveis aleatórias de Bernoulli independentes e limitados ao conjunto $\{0, 1\}$, indicando, respectivamente, os eventos: bit sem erro e bit com erro.

O detalhamento e os parâmetros utilizados na implementação do Modelo HMM serão abordados no próximo capítulo.

4.4.2 MODELO DE ERROS INDEPENDENTES

Neste modelo, as seqüências binárias indicando a presença ou não de erro de bit são geradas com base na variável aleatória de Bernoulli. Desta forma, a ocorrência

de erro em um determinado intervalo de bit não altera a probabilidade de erro para o próximo intervalo de bit, ou seja, o processo de erro é sem memória, ao contrário dos modelos que geram erros em surto.

Assim sendo, o processo aleatório *iid* gerado apresenta uma probabilidade de erro em k bits do processo dada por:

$$p(k) = \sum_{i=1}^N \binom{N}{k} p^k (1-p)^{N-k}, \quad (4.8)$$

onde N denota o tamanho do blocos de bits gerado e p é a probabilidade de ocorrência de um bit errado.

4.5 RESUMO

Neste capítulo apresentamos algumas das principais características dos canais HF, enfocando os aspectos de propagação nas camadas ionosféricas.

Em seguida, descrevemos o modelo HMM utilizado para representar as sequências de erros, bem como os erros em surto que caracterizam os sistemas de transmissão em canal HF.

Com o objetivo de avaliar a influência do modelo de erros no desempenho dos protocolos das camadas superiores em canais HF, descrevemos também o modelo de erros independentes e identicamente distribuídos.

No próximo capítulo serão abordados o ambiente de simulação adotado para avaliação do desempenho TCP em canais HF.

5 AMBIENTE DE SIMULAÇÃO

5.1 INTRODUÇÃO

Conforme mencionado no CAP. 1, o objetivo da presente dissertação é avaliar o desempenho de algumas estratégias de melhoria da vazão do protocolo TCP em canais HF. As estratégias a serem avaliadas são:

- uso da camada física com mecanismo FEC, denotado por FEC;
- uso do protocolo da camada de enlace com mecanismo ARQ com repetição seletiva, denotado por ARQ-SR e
- uso do protocolo da camada de enlace com mecanismo ARQ com repetição seletiva em presença da camada física com FEC, denotado por ARQ-SR-FEC.

De forma complementar, serão avaliados os desempenhos em canais HF das seguintes versões do TCP: Tahoe, Reno, New Reno, Sack, Fack e Westwood. Também serão avaliados a influência dos seguintes parâmetros no desempenho do TCP e do protocolo de enlace:

- Tamanho do arquivo FTP;
- Persistência do protocolo da camada de enlace;
- Tamanho da PDU da camada de transporte (segmento) e da camada de enlace (quadro);
- Valor do temporizador *NACK* do protocolo da camada de enlace ("*Retransmit Timer*");
- Valor de *timeout* mínimo e
- Tamanho de *cwnd* no início da conexão.

Por último, serão avaliados a influência dos modelos de erros correlacionados e independentes no desempenho dos protocolos TCP e de camada de enlace em canais HF.

De posse dos objetivos, adotamos o recurso computacional com uma abordagem de simulação por eventos discretos, onde cada etapa de manipulação do pacote ou quadro pelo simulador (ver Seção 5.2) configura um evento. A vantagem desta abordagem é a rapidez, flexibilidade e facilidade, quando comparada outras estratégias de simulação, tal como a simulação de Monte Carlo.

Neste capítulo serão abordados alguns aspectos do ambiente de simulação através da ferramenta *Network Simulator* e feitas considerações sobre os modelos de erros, ao protocolo de enlace simulado, ao mecanismo FEC e aproximações empregadas nas simulações.

5.2 FERRAMENTA DE SIMULAÇÃO: *NETWORK SIMULATOR*

O *Network Simulator* versão 2 (*ns-2*) é um simulador por eventos discretos específico para redes TCP/IP baseado em *software* livre e mantido pelo projeto VINT (*Virtual InterNetwork Testbed*) e desenvolvidos por pesquisadores de instituições como Universidade de *Berkeley*, USC/ISI, LBL e Xerox PARC. Essa ferramenta é largamente utilizada pela comunidade acadêmica, pois permite a adição de novos módulos ou alteração dos existentes abrindo perspectivas amplas para pesquisas nas áreas de redes e protocolos de comunicação.

O *ns-2* utiliza duas linguagens de programação baseado em orientação ao objeto: linguagem OTcl e C++. O interpretador OTcl é a linguagem Tcl com extensão a orientação ao objeto, utilizada como interface para o usuário na definição dos *scripts* (eventos) de simulação e na utilização das bibliotecas dos objetos componentes de rede a serem usadas durante a simulação. Por outro lado, a linguagem C++ é utilizada na implementação dos agendadores de eventos e dos objetos componentes básicos de rede. Os objetos compilados em C++ são disponíveis para o interpretador OTcl através de um recurso chamado de "*OTcl Linkage*" que cria um vínculo entre os objetos em C++ e os objetos em OTcl, de forma que as funções de controle e as variáveis configuráveis especificadas nos objetos em C++ atuem como funções membro e variáveis membro do objeto em OTcl correspondente. Ambas as linguagens de programação contribuem para formar um *software* de simulação de processamento rápido, devido ao C++, e de fácil interação com usuário, devido ao OTcl.

No *ns-2* é implementado uma grande variedade de protocolos e outros mecanismos de redes baseadas na arquitetura TCP/IP. As implementações contidas no *ns-2*

que foram utilizadas no presente trabalho são: as versões do protocolo TCP descritas na Seção 3.3.1.2 e o protocolo de aplicação FTP. Originalmente, o módulo do protocolo de enlace com repetição seletiva não consta no *ns-2*. Contudo, foi utilizado um módulo do protocolo de enlace implementado para Sistemas CDMA2000 que foi modificado, conforme será descrito na Seção 5.6, compilado e instalado no simulador.

5.3 TOPOLOGIA UTILIZADA NAS SIMULAÇÕES

A FIG. 5.1 mostra a topologia que foi utilizada em todas as simulações. Nesta topologia encontram-se um servidor FTP conectado a uma rede local (LAN), um terminal conectado a uma rede local (LAN) e dois rádios HF com a pilha TCP/IP integrada (*Gateway HF*) distanciados de 2900 Km.

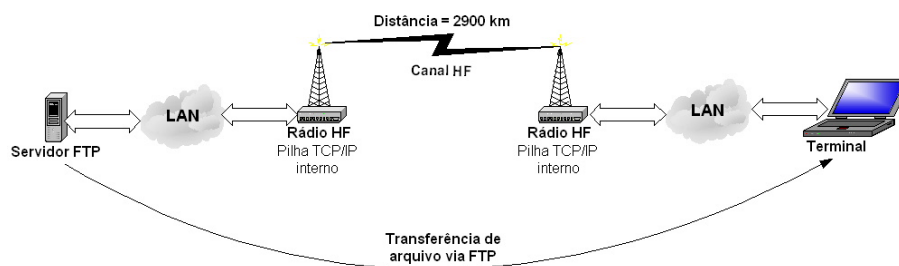


FIG. 5.1: Topologia utilizada nas simulações

Em todas as simulações, considera-se finalizada a etapa de acesso e autenticação no servidor FTP, mediante login e senha, permitindo ao usuário fazer a cópia de um determinado arquivo. Desta forma, a avaliação de desempenho será feita com base na transferência do arquivo FTP do servidor para o terminal.

5.4 CENÁRIO DE SIMULAÇÃO CARREGADO NO *NS-2*

A topologia ilustrada na FIG. 5.1 foi traduzida e carregada no *ns-2*, resultando no cenário de simulação descrito na FIG. 5.2.

5.4.1 CONSIDERAÇÕES SOBRE O CENÁRIO DE SIMULAÇÃO

Sobre cenário de simulação (FIG. 5.2) são feitas a seguir algumas considerações. Primeiramente, a taxa de transmissão adotada em ambos os MODEMS HF é de 4800

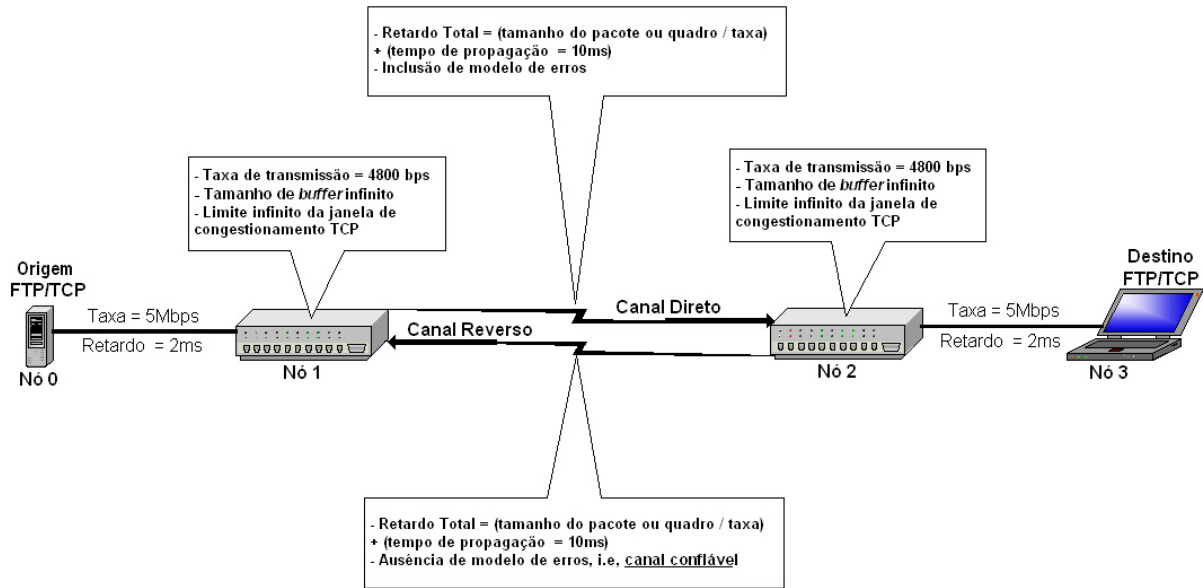


FIG. 5.2: Cenário de simulação carregado no *ns-2*

bps. Este valor encontra-se especificado no documento MIL-STD-188-110B (MIL-STD-188-110B, 2000) para MODEMs com modo de operação em frequência fixa (ausência do mecanismo de salto de frequência), utilizando modulação *Phase Shift Keying* (PSK) e sem codificador FEC.

Admitiu-se que a distância entre os terminais é de 2900 Km e que o mecanismo de propagação em HF é uma reflexão total (ver FIG. 4.1) na camada F2 com altura aproximada de 380 Km. Supondo ser esta uma propagação em espaço livre, o retardo de propagação resultante é de aproximadamente 10ms.

Denotando por Δ_T o retardo total do canal HF, por Δ_p o retardo de propagação, por R a taxa de transmissão e por N como sendo o tamanho do pacote/quadro, o retardo total é dado por:

$$\Delta_T = \frac{N}{R} + \Delta_p. \quad (5.1)$$

No cenário apresentado o canal HF utilizado é inteiramente dedicado, em outras palavras, assumimos que neste cenário não há compartilhamento do canal com outros usuários, logo não há mecanismo de controle de acesso ao meio (MAC).

Em relação ao protocolo TCP, as janelas de recepção (*awnd*) ou *buffers* dos *gateways* HF são considerados de tamanho infinito, para que as perdas fossem provocadas apenas por erros devido ao canal. Assim sendo, conforme descrito na EQ 2.7, a janela de

transmissão será igual à janela de congestionamento ($cwnd$), de tal forma que a manipulação de $cwnd$ influirá diretamente na taxa de envio dos segmentos. Além disso, a janela de congestionamento pode assumir qualquer valor, ou seja, o valor de $cwnd$ é ilimitado.

Também assumimos que o canal reverso é inteiramente confiável, ou seja, ocorre erro de transmissão apenas no canal direto, conforme ilustrado na FIG. 5.2, tendo em vista o pequeno tamanho dos pacotes IP do canal reverso se comparado com os pacotes IP no canal direto. Trata-se de uma suposição razoável que tem sido empregada em diversos trabalhos que versam sobre desempenho de protocolos de comunicação em canais sem fio.

5.5 SIMULAÇÃO DO PROCESSO DE ERROS NO CANAL HF

Os processos de erros gerados para canal HF, descritos nas Seções 4.4.1 e 4.4.2, são gerados e carregados no *ns-2* para avaliação do desempenho do TCP. O detalhamento deste processo será feita nas seções subseqüentes.

5.5.1 GERAÇÃO DO PROCESSO DE ERROS CORRELACIONADOS DE BITS A PARTIR DO MODELO HMM

Conforme descrito na Seção 4.4.1, a partir do modelo HMM, o processo de erros correlacionados de bits gerado apresenta uma probabilidade de erros de bits (P_{alvo}) alvo governada pela EQ. 4.7. O modelo tem como parâmetros de entrada as probabilidades de erros de bit p_1 , p_2 e p_3 de cada estado da cadeia de Markov Escondida e a matriz de probabilidades de transição dos estados.

A referência (TSAI, 1969) propõe uma matriz de probabilidades de transição de estados M , cujos parâmetros modelam as transições entre os estados em um canal HF. A matriz M é dada por:

$$M = \begin{pmatrix} 0.99911 & 0 & 0.00089 \\ 0 & 0.73644 & 0.26356 \\ 0.36258 & 0.58510 & 0.05232 \end{pmatrix} \quad (5.2)$$

O diagrama de estados da cadeia de Markov correspondente é mostrada na FIG. 5.3.

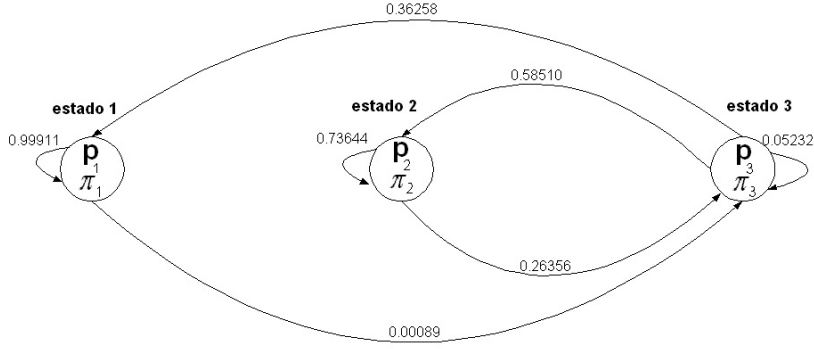


FIG. 5.3: Diagrama de estados do Modelo HMM

TAB. 5.1: Parâmetros do Modelo HMM.

P_{alvo}	p_1	p_2	p_3
10^{-3}	0.0005	0.048161	0.1
10^{-3}	0.0001	0.03147	0.3
10^{-4}	0.00001	0.003147	0.03
10^{-5}	0.000001	0.000315	0.003

No trabalho de Tsai os coeficientes da referida matriz foram estimados com base nos dados coletados em um enlace HF entre Chicago e San Diego realizados pelo *ITT Research Institute* para o *US.Naval Eletronics Laboratory Center*.

Por outro lado, os valores de p_1 , p_2 e p_3 foram adotados nesta dissertação no sentido de garantir a P_{alvo} dada pela EQ. 4.7. As probabilidade de erros adotadas para cada estado resulta em uma cadeia de Markov Escondida com cada estado representando, respectivamente, um canal de boa qualidade, um canal de qualidade média e um canal de qualidade ruim. Admitiu-se assim que $p_1 < p_2 < p_3$.

Além disso, procurou-se considerar nas simulações o fato de que as estações HF mais modernas possuem a técnica de estabelecimento automático do enlace denominado de *Automatic Link Establishment* (ALE), a qual inclui um mecanismo, conhecido como *Automatic Channel Selection* (ACS), que realiza uma prévia varredura nos canais HF e seleciona aqueles que se encontram em melhores condições de transmissão. Para incluir a funcionalidade ACS no ambiente de simulação considerado, o estado inicial do modelo HMM foi escolhido aleatoriamente entre o estado bom (estado 1) e o estado médio (estado 2).

As probabilidade de erros de cada estado do modelo HMM e a respectiva P_{alvo} são mostrados na TAB. 5.1.

5.5.2 SIMULAÇÃO NO *NS-2* DO MODELO DE ERROS CORRELACIONADOS

A FIG. 5.4 apresenta o diagrama em blocos do ambiente de simulação utilizado para a geração do processo de erros correlacionados de bits, bem como sua inclusão no *ns-2*.

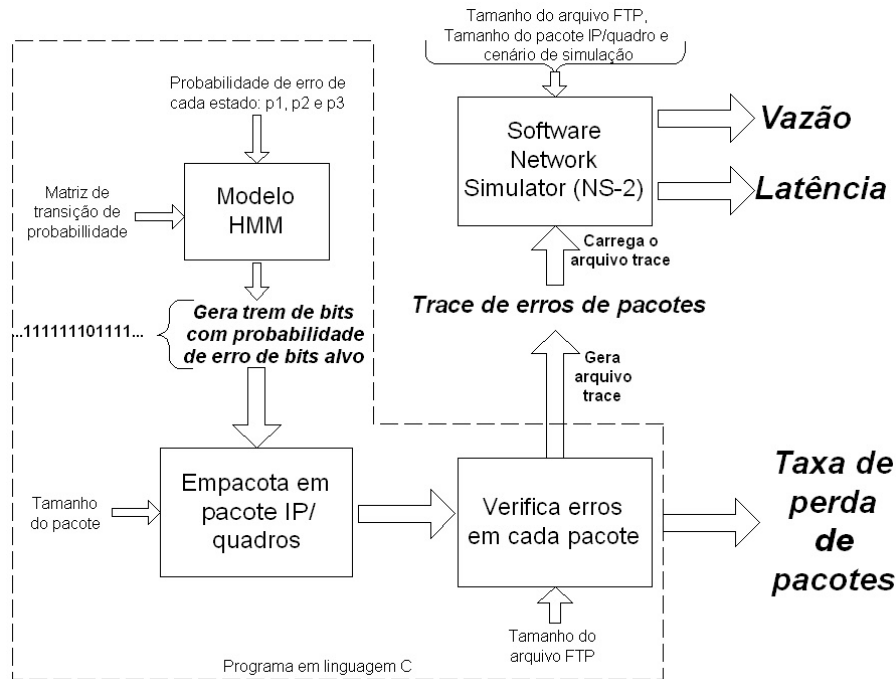


FIG. 5.4: Diagrama em blocos do ambiente de simulação usado na geração do modelo HMM e sua inclusão no *ns-2*.

As seqüências binárias geradas pelo modelo HMM devem ser transformadas em seqüências que representam os pacotes livres de erros e os pacotes com erros. Esta transformação é feita tomando-se, ao longo da seqüência, blocos de bits de tamanho igual ao pacote IP (ou ao quadro da camada de enlace, quando for o caso). Após a etapa de empacotamento em blocos, uma varredura ao longo do pacote é feita, no sentido de verificar a presença de erros de bits. Posteriormente, é criada uma seqüência ordenada dos pacotes que estão com erros de bits. Esta seqüência ordenada funciona como um "trace" de sinalização para realizar o descarte de pacotes. Todo este processo de geração do modelo HMM, empacotamento em blocos e geração dos "traces" foram implementados através de programação em linguagem C.

Vale ressaltar ainda que o critério para caracterizar um pacote ou um quadro como errado e, conseqüentemente descartá-lo, é a existência de um único bit errado dentro

do bloco. Este é o critério padrão usado pelo TCP e pela maioria dos protocolos ARQ para considerar um pacote corrompido e requisitar sua retransmissão.

De posse da seqüência ordenada de pacotes ou quadros com erros e, mediante o uso de uma função membro de uma classe específica para módulo de erros do *ns-2*, o "*trace*" é carregado no *ns-2* e os pacotes indicados serão considerados pacotes com erros pelo simulador. Desta forma, as perdas de pacotes ocorridas durante a simulação são geradas pelo modelo HMM com parâmetros que caracterizam o canal de HF.

5.5.3 SIMULAÇÃO NO *NS-2* DO MODELO DE ERROS INDEPENDENTES

De forma similar ao modelo HMM, o diagrama em blocos da FIG. 5.5 ilustra o ambiente de simulação usado para geração e inclusão no *ns-2* de um processo de erros de bits independente.

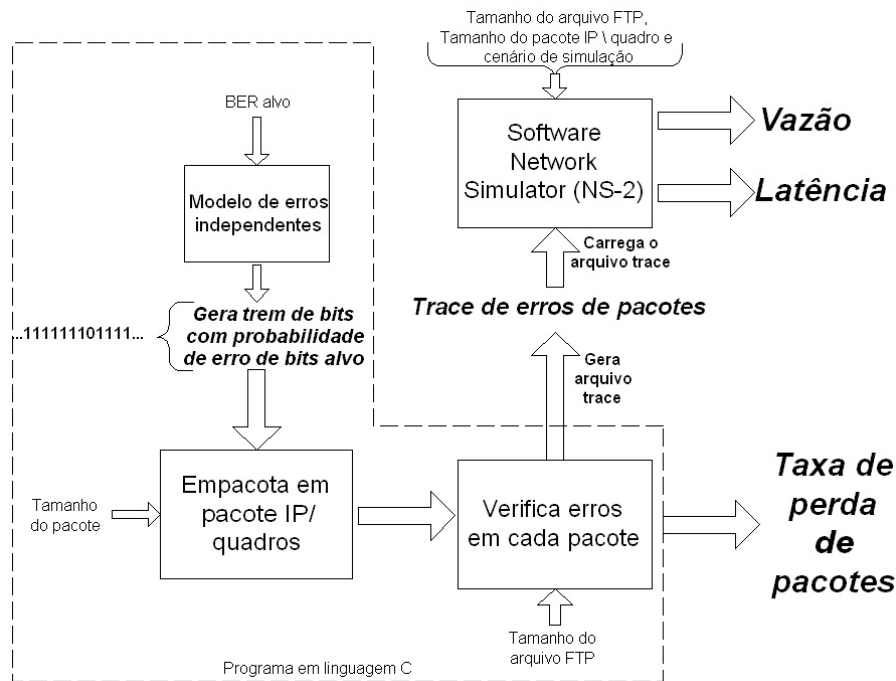


FIG. 5.5: Diagrama em blocos do ambiente de simulação usado para geração do processo de erros de bits independentes e sua inclusão no *ns-2*

Para fins de comparação do desempenho de TCP em canais HF modelados com processo de erros correlacionados (modelo HMM), a ser apresentada no CAP. 6, foram implementados processos de erros independentes com P_{alvo} de 10^{-3} , 10^{-4} e 10^{-5} .

5.6 SIMULAÇÃO DO PROTOCOLO DE ENLACE COM MECANISMO ARQ-SR

O protocolo da camada de enlace com mecanismo ARQ adotado nas simulações foi baseado em modificações feitas no módulo chamado de *Radio Link Protocol* voltado para sistemas CDMA2000. Tais modificações foram realizadas de tal forma que seus mecanismos e parâmetros fossem aproximados aos do AX.25.

No *ns-2*, o protocolo de enlace implementado realiza a fragmentação e remontagem do pacote IP em quadros com tamanhos e cabeçalhos configuráveis. O mecanismo ARQ é do tipo de repetição seletiva (ARQ-SR), de forma semelhante ao AX.25. Contudo, diferentemente do AX.25, este protocolo não utiliza esquemas de janela de transmissão, transmitindo todos os quadros sequencialmente. Outra diferença entre esses protocolos é que o algoritmo implementado não reconhece os quadros recebidos corretamente. Em caso de perda ou detecção de erros no quadro, o protocolo realiza a recuperação através do uso dos *NAK frames*, conforme ilustrado na FIG. 5.6.

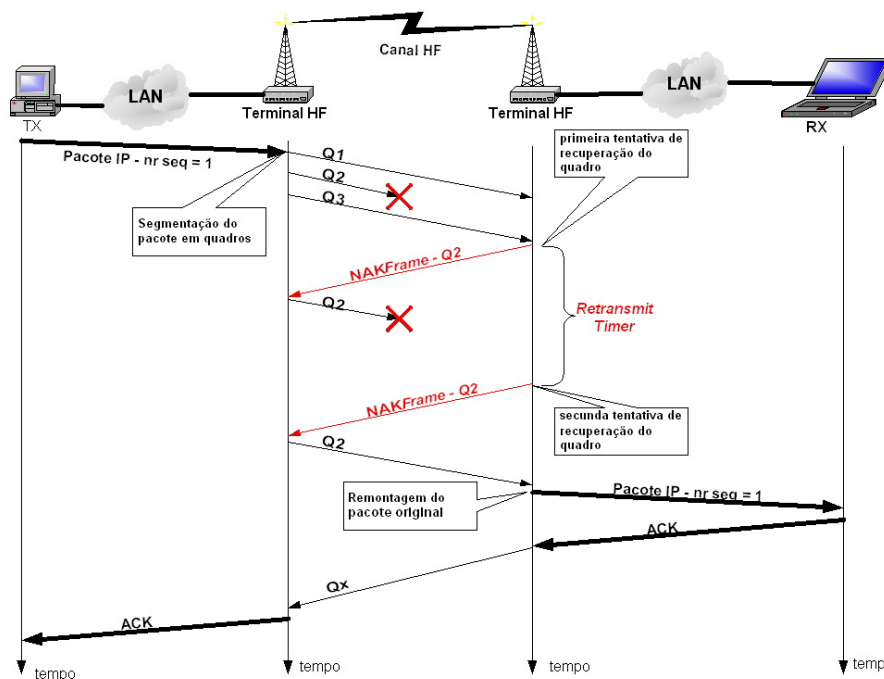


FIG. 5.6: Mecanismo ARQ-SR do protocolo de enlace implementado.

O receptor monitora duas variáveis: $L_V(N)$ e $L_V(R)$. A primeira delas registra o número de seqüência do quadro necessário para entrega ordenada à camada superior. A segunda designa o número de seqüência do próximo quadro esperado pelo receptor. Assim, o receptor identifica a perda de um quadro de duas maneiras: pela

recepção de um quadro com número de seqüência L_Seq maior do que $L_V(R)$ (ver FIG. 5.6), ou pela recepção do quadro chamado de *Idle frame* com L_Seq maior do que $L_V(R)$ (ver FIG. 5.7).

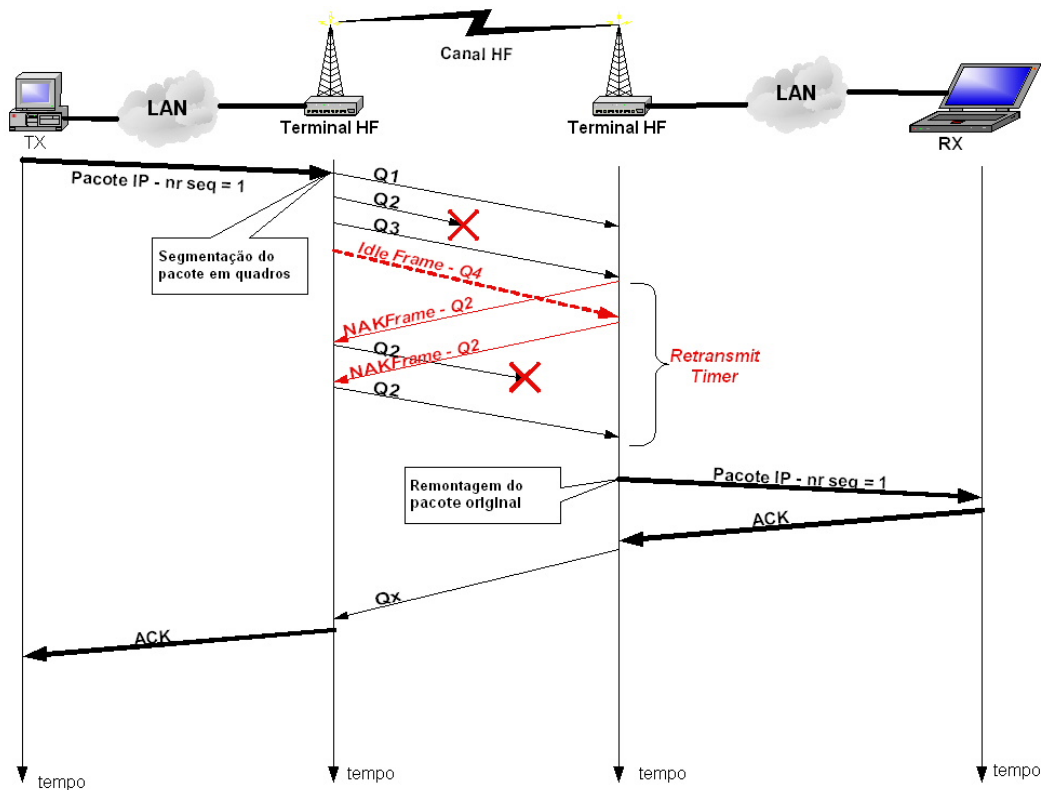


FIG. 5.7: Uso dos *Idle frames* na detecção do quadro perdido.

Em caso de detecção de perda, o receptor grava o L_Seq do quadro perdido em uma lista chamada de *NAK LIST*. Posteriormente, envia um *NAK frame* contendo o L_Seq do quadro perdido, solicitando a retransmissão do mesmo, e inicia um temporizador chamado *Retransmit Timer*. Este temporizador é decrementado a cada envio de *Idle frame*, sendo que, ao expirar, um novo *NAK frame* é enviado (FIG. 5.6), gerando uma nova retransmissão. O número de tentativas de recuperação do quadro perdido é limitado pela persistência do protocolo (δ).

Os *idle frames* são quadros sem conteúdo de informação que são enviados, periodicamente, sempre que o transmissor ou receptor não tiver dados a enviar. Desta forma, o uso de *Idle frames* agiliza a detecção e recuperação dos quadros perdidos, conforme descrito na FIG. 5.7.

Na situação de recebimento dos quadros retransmitidos, o protocolo atualiza o *NAK LIST*, realiza o ordenamento e a remontagem do pacote IP.

5.7 MODELO DE SIMULAÇÃO DO MECANISMO FEC

O padrão MIL-STD-188-110B especifica que os códigos corretores de erros para MODEMs com modo de operação em frequência fixa devem estar de acordo com a TAB. 5.2.

TAB. 5.2: Códigos corretores de erros para MODEMs HF com modo de operação em frequência fixa, segundo padrão MIL-STD-188-110B

Taxa de dados (bps)	Taxa de código efetiva	Método para atingir a taxa de código
4800	–	–
2400	1/2	Rate 1/2
1200	1/2	Rate 1/2 code
600	1/2	Rate 1/2 code
300	1/4	Rate 1/2 code repeated 2 times
150	1/8	Rate 1/2 code repeated 4 times
75	1/2	Rate 1/2

Por outro lado, a TAB. 5.3 apresenta a probabilidade de erro de bit máxima exigida no padrão MIL-STD-188-110B para MODEM seriais. Também é especificado neste padrão que a verificação de desempenho seja testada utilizando-se um simulador de canal HF baseado no modelo de Waterson (WATTERSON, 1970) com 2 raios de mesma variância, sendo os demais parâmetros do modelo de canal também especificados na mesma tabela.

Tais requisitos podem ser atendidos com o uso de códigos convolucionais de distância livre adequada. A FIG. 5.8 mostra o diagrama em blocos do codificador convolucional de comprimento 7, especificado em MIL-STD-188-110B, cujos polinômios geradores, $T_1(x)$ e $T_2(x)$ são dados por:

$$T_1(x) = X^6 + X^4 + X^3 + X + 1 \text{ e } T_2(x) = X^6 + X^5 + X^4 + X^3 + 1. \quad (5.3)$$

O modelo de simulação adotado para camada física considerando o uso de FEC com taxa 1/2, utiliza as informações da TAB. 5.3 para gerar, através do modelo HMM, um canal com taxa de erro de bits codificada igual a 10^{-5} e de taxa de transmissão de informação reduzida pela metade, ou seja, de 2400 bps ($\frac{4800}{2}$). Resumindo, o código convolucional não foi implementado, apenas foi considerado suas características no que

TAB. 5.3: Desempenho mínimo exigido para MODEMs seriais (MIL-STD-188-110B).

Taxa de bits (bps)	Percursos do canal	Espalhamento de retardo (ms)	Espalhamento de Doppler (Hz)	SNR (dB)	BER codificada
4800	1 fixo	–	–	17	10^{-3}
4800	2 c/ desvanecimento	2	0,5	27	10^{-3}
2400	1 fixo	–	–	10	10^{-5}
2400	2 c/ desvanecimento	2	1	18	10^{-5}
2400	2 c/ desvanecimento	2	5	30	10^{-3}
2400	2 c/ desvanecimento	5	1	30	10^{-5}
1200	2 c/ desvanecimento	2	1	11	10^{-5}
600	2 c/ desvanecimento	2	1	7	10^{-5}
300	2 c/ desvanecimento	5	5	7	10^{-5}
150	2 c/ desvanecimento	5	5	5	10^{-5}
75	2 c/ desvanecimento	5	5	2	10^{-5}

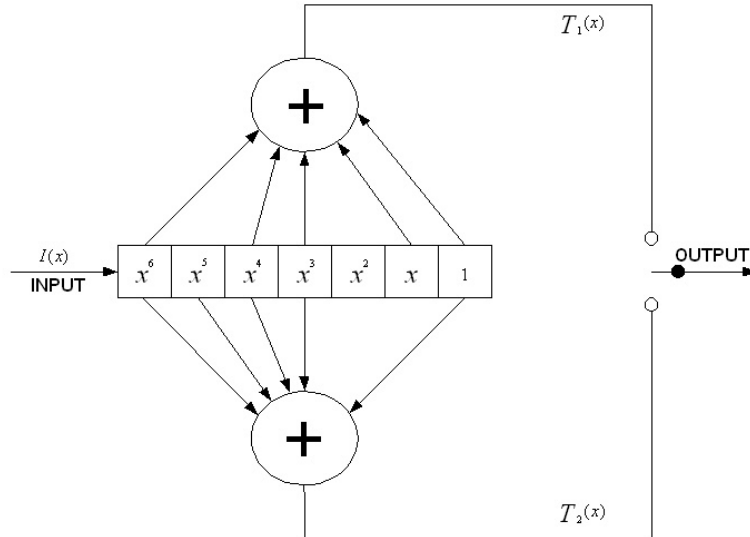


FIG. 5.8: Diagrama em blocos do codificador FEC especificado em MIL-STD-188-110B.

se refere a melhoria de desempenho ($P_{alvo} = 10^{-5}$) e redução na taxa útil de transmissão, em razão do uso de redundâncias (no caso, considera-se 2400 bps).

O código convolucional especificado pelo padrão MIL-STD-188-110B possui distância livre igual a 10 (WICKER, 1995). Assim sendo, é razoável admitir que um código código convolucional de taxa $2/3$ e distância livre 10, tem o mesmo desempenho do código anterior e, portanto, produzirá também um canal com taxa de erro de bits codificada igual a 10^{-5} . Em (CHANG, 1997) é apresentado um código convolucional com estas características.

De forma similar, o protocolo de enlace na presença de FEC com taxa $2/3$ será simulado através da geração do canal com BER igual a 10^{-5} e utilização de taxa de transmissão de 3200 bps ($\frac{2 \times 4800}{3}$). Novamente, o código convolucional não foi implementado, sendo o modelo de FEC de taxa $2/3$ adotado como sendo P_{alvo} da camada física igual a 10^{-5} e taxa de transmissão de 3200 bps.

5.8 MEDIDAS DE DESEMPENHO

A figura de mérito utilizada nesta dissertação para avaliar o desempenho dos protocolos simulados é a Vazão (*Throughput*). Sendo a vazão representado por B , F e T denotando, respectivamente, o tamanho do arquivo FTP a ser transmitido e o tempo total para entrega do arquivo FTP ao destinatário (latência), a vazão, expressa em termos de bits por segundo, é matematicamente dada pela EQ. 5.5:

$$B = \frac{F}{T} \quad (5.4)$$

Outra medida de desempenho a ser considerada como análise quantitativa é a Latência, definida como o tempo total para entrega do arquivo FTP ao destinatário. A medida da latência adotada para as transmissões em HF é o segundo, minutos ou horas dependendo do caso.

5.9 RESUMO

Neste capítulo foi apresentado o ambiente de simulação adotado para avaliação de desempenho dos protocolos de camada de enlace e de transporte (TCP) em canais HF. Decreveu-se, inicialmente, a ferramenta de simulação *ns-2*, a topologia e o cenário de simulação adotados. Posteriormente, foi apresentado o processo de geração dos modelos de erros correlacionados e independentes, bem como sua inclusão no simulador *ns-2*.

Com objetivo de avaliar o desempenho do TCP na presença de protocolos de camada de enlace com mecanismo ARQ e de camada física com FEC, foram descritos o protocolo de enlace com mecanismo ARQ e o modelo de simulação para os esquemas FEC adotados na presente dissertação, que tiveram como base as especificações do padrão MIL-STD-188-110-B para MODEMs HF de portadora única.

6 RESULTADOS

Os resultados apresentados a seguir foram dispostos em tópicos de acordo com os objetivos da presente dissertação mencionados no CAP. 1, da seguinte forma:

- Comparações das estratégias de melhoria da vazão do protocolo TCP em canal HF;
- Modelagem de erros em canal HF e sua influência no desempenho do TCP;
- Desempenho do TCP em canal HF;
- Desempenho das versões do TCP em presença de camada física com FEC-Taxa 1/2 e
- Desempenho do TCP em presença de protocolo de enlace com ARQ-SR.

TAB. 6.1: Parâmetros da simulação utilizados na comparação das estratégias do protocolo TCP em canal HF.

Parâmetro	Valor
Tamanho do arquivo FTP	10000 bytes
Tamanho do pacote IP	400 bytes
Versão do TCP	Reno
Cabeçalho do pacote TCP/IP	40 bytes
<i>Timeout</i> mínimo	20ms
<i>cwnd</i> inicial	1
Tamanho do quadro	50 bytes
Tamanho do cabeçalho do quadro	6 bytes
Persistência	3
Temporizador NACK do protocolo de enlace	10
Modelo de erros	Modelo HMM
<i>P_{alvo}</i>	10^{-3}

Além do cenário de simulação descrito na Seção 5.3, para fins de investigação inicial da comparação de desempenho dos protocolos de TCP e de camada de enlace, foram consideradas os parâmetros apresentados na TAB. 6.1. Os valores dos tamanhos de pacote IP e quadro, persistência e temporizador NACK foram escolhidos em uma investigação preliminar.

Posteriormente, com intuito de promover uma criteriosa escolha de parâmetros do protocolos, alguns dos valores contidos da TAB. 6.1 terão avaliadas suas influências nos desempenhos do TCP e do protocolo de enlace. Em particular, serão considerados os efeitos do tamanho do arquivo FTP e do valor do Temporizador NACK do protocolo de enlace, dentre outros.

Como se mostra na TAB. 6.1, o valor da P_{alvo} escolhido como padrão nas simulações foi de 10^{-3} , pois o mesmo encontra-se dentro da faixa de valores de BER¹⁰ encontrados em canal HF.

O tamanho do cabeçalho do quadro do protocolo de enlace adotado foi de 6 bytes, por representar o tamanho do cabeçalho do AX.25 (20 bytes) sem o campo de endereço (14 bytes). Este valor foi adotado em virtude de tratar-se de um enlace ponto-a-ponto, sendo, neste caso, considerado dispensável o campo de endereço. Além do mais, este valor encontra-se em consonância com outros protocolos de camada de enlace como, por exemplo, o quadro RLP, empregado em sistemas CDMA2000, o qual utiliza um cabeçalho de 5 bytes. No entanto, serão também apresentados nas seções subsequentes, alguns resultados de vazão dos protocolos TCP em presença de protocolos de enlace utilizando quadros com cabeçalho de 20 bytes, de acordo com a especificação AX.25.

Vale ressaltar ainda que em todos cenários foram realizados 1800 simulações independentes, sendo a medida de desempenho estimada pela média empírica dos valores obtidos nessas simulações. Conforme mencionado na Seção 5.8, a medida de desempenho adotada nas simulações foi a Vazão (B) obtida da seguinte forma:

$$B = \frac{F \times 8}{T} \text{ bps}, \quad (6.1)$$

onde F e T corresponde, respectivamente, ao tamanho do arquivo FTP (em bytes) a ser transmitido e ao tempo total para entrega do arquivo FTP ao destinatário.

¹⁰O *Bit Error Rate* refere-se a um parâmetro de qualidade em sistemas de comunicação medido por equipamentos específicos. Neste trabalho emprega-se a notação P_{alvo} para enfatizar que os parâmetros dos modelos HMM adotados para os processos de erros na camada física foram ajustados para atender a este valor de probabilidade de erro.

6.1 COMPARAÇÃO DE ALGUMAS DAS ESTRATÉGIAS DE MELHORIA DA VAZÃO DO PROTOCOLO TCP EM CANAL HF

As Figuras 6.1 e 6.2 apresentam uma comparação de desempenho das estratégias para melhoria da vazão do protocolo TCP em canal HF. As estratégias avaliadas são:

- Protocolo TCP em canal HF¹¹, denotado por TCP;
- Protocolo TCP em presença de camada física com mecanismo FEC com Taxa 1/2, denotado por TCP-LL-FEC-Taxa 1/2;
- Protocolo TCP em presença de camada física com mecanismo FEC com Taxa 2/3, denotado por TCP-LL-FEC-Taxa 2/3;
- Protocolo TCP em presença de protocolo da camada de enlace com mecanismo ARQ-SR, denotado por TCP-LL-ARQ-SR;
- Protocolo TCP em presença de protocolo da camada de enlace com mecanismo ARQ-SR e camada física com FEC com Taxa 1/2, denotado por TCP-LL-ARQ-SR-FEC-Taxa 1/2;
- Protocolo TCP em presença de protocolo da camada de enlace com mecanismos ARQ-SR e camada física com FEC com Taxa 2/3, denotado por TCP-LL-ARQ-SR-FEC-Taxa 2/3.

Os resultados da FIG. 6.1 foram obtidos empregando os parâmetros especificados na TAB. 6.1, ao passo que os resultados de simulação apresentado na FIG. 6.2 foram obtidos considerando as modificações mostrados na TAB. 6.2. Os valores desta última tabela foram obtidos com base nos resultados de avaliação de parâmetros associados ao protocolo de camada de enlace que serão apresentados nas seções subseqüentes.

¹¹Na realidade, nas redes convencionais o uso dos protocolos de camada de enlace é imprescindível, pois, em geral, esses protocolos estabelecem o endereçamento físico e os procedimentos de conexão de enlace, além de desempenhar mecanismos de detecção de erros, controle de acesso ao meio e outros, caso típico da especificação IEEE 802.3 Ethernet. Contudo, neste trabalho, o termo "TCP em canal HF" será utilizado para referir-se à ausência de um protocolo de enlace com mecanismos ARQ e de camada física com FEC.

TAB. 6.2: Parâmetros criteriosamente escolhidos via simulação computacional do protocolo de enlace com mecanismo ARQ-SR.

Parâmetro	Valor
Tamanho do quadro	25 bytes
Tamanho do cabeçalho do quadro	6 bytes
Persistência	3
Temporizador NACK do protocolo de enlace	5

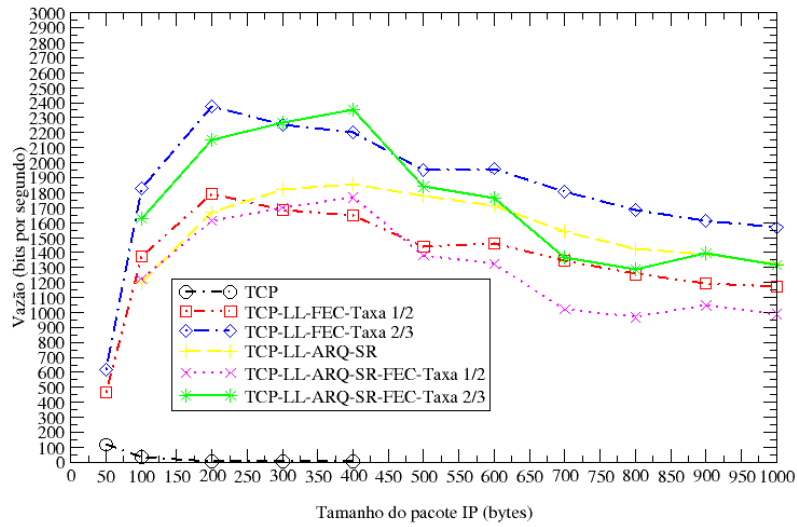


FIG. 6.1: Comparação das estratégias de melhoria do TCP em canal HF empregando os parâmetros da TAB. 6.1.

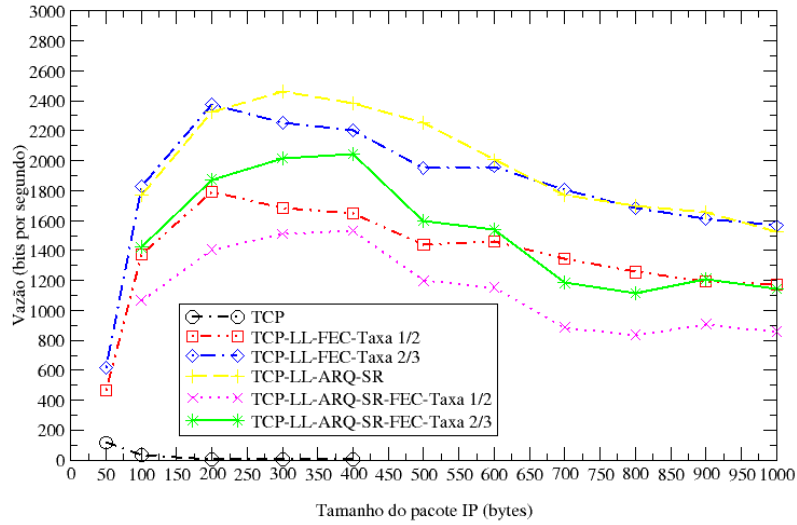


FIG. 6.2: Comparação das estratégias de melhoria do TCP em canal HF empregando os parâmetros da TAB. 6.2.

Ambas as figuras evidenciam a enorme diferença de desempenho entre o TCP em presença de protocolo de camada de enlace ARQ e o TCP sem a camada de enlace ARQ, o que nos leva a concluir que o uso do mecanismo ARQ se mostra sempre vantajoso em relação ao TCP sem a camada de enlace com ARQ. De fato, este resultado era esperado devido à capacidade dos protocolos de enlace ARQ de prover maior grau de confiabilidade no canal, permitindo ao TCP utilizar seu mecanismo de janela deslizante de forma plena, bem como evitar os sucessivos incrementos do temporizador *timeout*, melhorando, desta forma, consideravelmente a vazão. Para confirmar as conclusões previamente apresentadas, foram obtidas amostras da evolução de *cwnd* em função do tempo para os casos do TCP com tamanho de pacote IP de 100 bytes sem a presença do protocolo de camada de enlace com ARQ e com presença do protocolo de enlace com ARQ com valores da TAB. 6.1, ilustradas, respectivamente, nas Figuras 6.3 e 6.4.

Nas figuras mencionadas, percebe-se, inicialmente, que no caso do TCP Reno com a presença do protocolo de enlace ARQ, o valor de *cwnd* médio = 5,01 corresponde a 3 vezes o valor *cwnd* médio = 1,67 obtido no caso do TCP sem a presença do protocolo de enlace ARQ, confirmando a melhor utilização da janela de congestionamento e menor ocorrência de *timeout*.

Outro fato importante mostrado na FIG. 6.4 é a redução inicial de *cwnd* para uma unidade, que foi provocado pela ocorrência de um *timeout* no início da conexão. Em geral, no início da conexão o valor do temporizador *timeout* é pequeno, pois foi atualizado (EQ. 2.6) com base na abertura de conexão feita com um pacote de 40 bytes. Logo, ao se transmitir um pacote IP de 100 bytes que serão segmentados em 3 quadros de 50 bytes ($\frac{100}{50-6}$) e remontados posteriormente, ocorre um considerável aumento do RTT e, conseqüentemente, o *timeout*. A ocorrência dos *timeouts* no início da conexão foi relatado em (KALLGREN, 2001). Este efeito torna-se mais freqüente à medida que aumenta-se o tamanho do pacote IP, sendo que uma possível solução para evitar a incidência dos mesmos é aumentar o valor do *timeout* mínimo, solução esta que será avaliada na Seção 6.5.1.1.

Além disso, observa-se, nas Figuras 6.1 e 6.2, o comportamento similar das estratégias que envolvem o mecanismo FEC com taxas 1/2 e 2/3. Este comportamento é explicado com base no modelo de FEC adotado, segundo, o qual, são empregados taxas diferentes submetido a um canal com mesma probabilidade de erro, logo o comportamento será regido pela taxa de transmissão, dando origem a curvas de desempenho paralelas, a que apresenta melhor desempenho o associado ao código com taxa maior.

Utilizando os parâmetros da TAB. 6.1, a FIG. 6.1 mostra que os desempenhos das estratégias envolvendo camada física com FEC Taxa 2/3 e com ARQ-SR-FEC Taxa 2/3 foram semelhantes e superiores aos das demais estratégias.

A FIG. 6.2 apresenta os resultados de desempenho das estratégias utilizando um protocolo de enlace ARQ cujos parâmetros são apresentados na TAB. 6.2. Nesta figura verifica-se que para a faixa de 200 a 600 bytes de tamanhos de pacote IP, a estratégia ARQ-SR produziu melhores resultados do que as demais combinações de protocolos investigados. Observa-se também a melhoria de desempenho da estratégia ARQ-SR em relação aos resultados da FIG. 6.1, fato que será melhor evidenciado e descrito na Seção 6.5.3.

Entretanto, as estratégias que envolvem conjuntamente os mecanismos ARQ-FEC produziram valores mais baixos de vazão frente aos alcançados na FIG. 6.1. Na realidade, o FEC provê uma redução drástica da quantidade de erros no canal. Logo, em um cenário que apresenta uma baixa taxa de erros, em virtude do mecanismo FEC, à medida que se reduz o tamanho do quadro, aumenta-se o retardo de transmissão devido ao maior número de quadros a serem transmitidos e ao maior *overhead*, tendo como

conseqüência a diminuição da vazão.

Os resultados aqui apresentados evidenciam a importância de contar com eficientes camadas de enlace para reduzir a necessidade de retransmissão acionadas pelo protocolo TCP. No entanto, esses resultados não são definitivos no que se refere a adotar as estratégias FEC, ARQ ou FEC e ARQ.

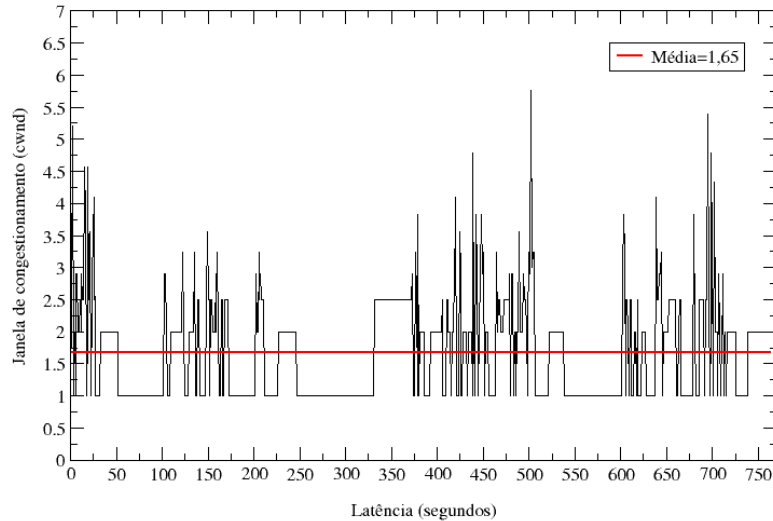


FIG. 6.3: Tamanho de *cwnd* do TCP sem protocolo de enlace.

6.2 MODELAGEM DE ERROS EM CANAL HF E SUA INFLUÊNCIA NO DESEMPENHO DO TCP

6.2.1 MODELO HMM: TAXA DE PERDA DE PACOTES VS TAMANHO DO PACOTE

A FIG. 6.5 mostra a taxa de perda de pacotes resultante do processo de erro gerado pelo modelo HMM e pelo mecanismo de empacotamento e detecção de erro de bits, descritos na Seção 5.5.2. Os valores de taxa de perda de pacotes foram obtidos de tal forma que, para cada tamanho de bloco, foi gerada uma seqüência de 10^6 bits.

Os gráficos da FIG. 6.5 mostram a forte dependência da taxa de perda de pacotes com o tamanho do bloco, de forma que, independentemente do valor de P_{alvo} utilizado, quanto maior o tamanho do pacote, maior será a probabilidade de perda. Este

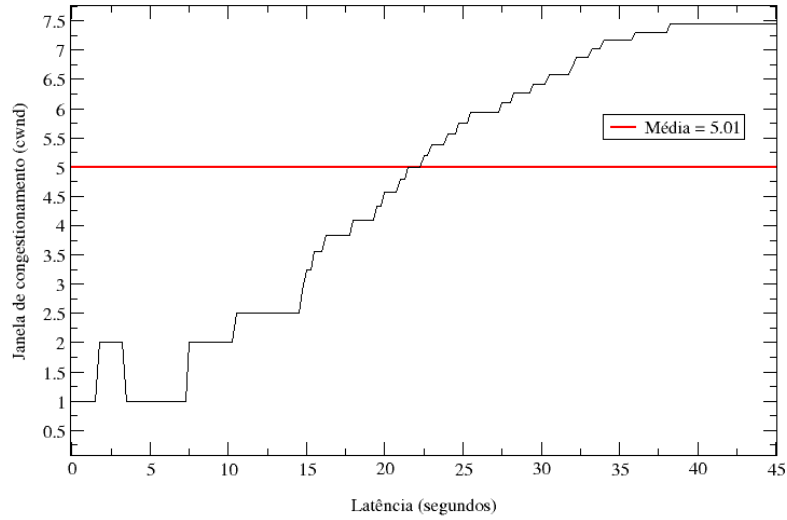


FIG. 6.4: Tamanho de *cwnd* do TCP com protocolo de enlace.

resultado é coerente, pois à medida que se aumenta o tamanho do pacote IP, aumenta-se a probabilidade de que pelo menos 1 bit desse pacote seja detectado com erro, ocasionando o seu descarte. Vale mencionar que as perdas tornam-se críticas, sobretudo em canal modelado com P_{alvo} igual a 10^{-3} , onde se observa que, para tamanhos de pacote maiores do que 400 bytes, a taxa de perda se aproxima de 100%.

Esses resultados serão freqüentemente utilizados para ajudar na análise do comportamento do desempenho do TCP e dos protocolos de camada de enlace, nas avaliações que se seguem.

6.2.2 INFLUÊNCIA DO MODELO DE ERROS NO DESEMPENHO DO PROTOCOLO TCP

Nesta seção será apresentada uma comparação de resultados de desempenho do TCP Reno com camada física gerando erros *iid* ou em surto.

A TAB. 6.3 mostra dois conjuntos de parâmetros do HMM utilizados em conjunto com a Matriz \mathbf{M} especificada pela EQ. 5.2 para modelar um canal com P_{alvo} igual a 10^{-3} .

A FIG. 6.8 apresenta as comparações de desempenho do TCP com o protocolo de enlace ARQ com base nos parâmetros da TAB. 6.2 na presença de erros correlacionados

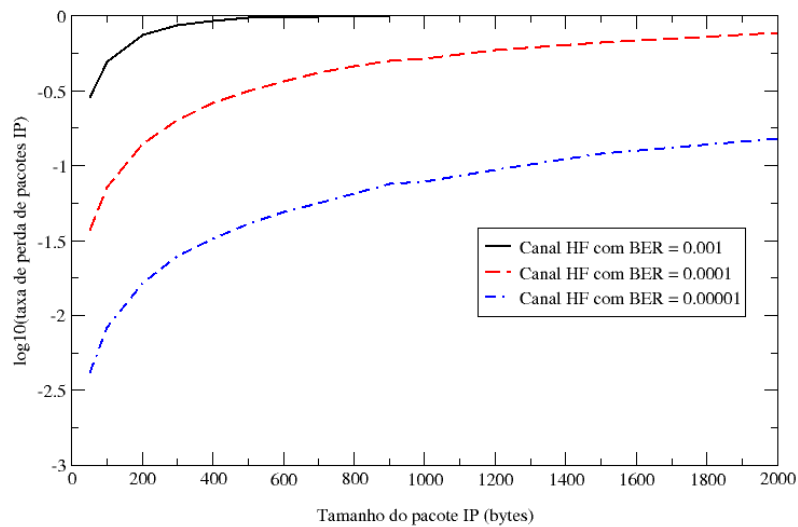


FIG. 6.5: Taxa de perda de pacotes em função do tamanho do pacote IP com camada física modelado por HMM.

TAB. 6.3: Conjuntos de parâmetros do Modelo HMM utilizados para modelar um canal com P_{alvo} igual a 10^{-3}

Conjunto de parâmetros	p_1	p_2	p_3
Conjunto de parâmetros 1	0.0005	0.048161	0.1
Conjunto de parâmetros 2	0.0001	0.03147	0.3

TAB. 6.4: Tamanho médio de surto das seqüências geradas pelo modelo HMM com base no conjunto de parâmetros 1 e 2 nos critérios 3, 5 e 7

	Conjunto de parâmetros 1	Conjunto de parâmetros 2
Critério 3	1,1298	1,3812
Critério 5	1,2942	1,8651
Critério 7	1,4648	2,2656

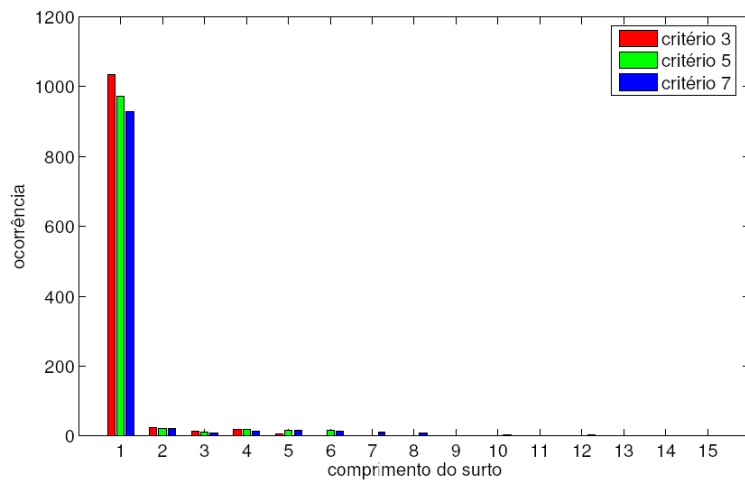


FIG. 6.6: Tamanho do surto de erros das sequências geradas segundo o modelo HMM com o conjunto de parâmetros 1

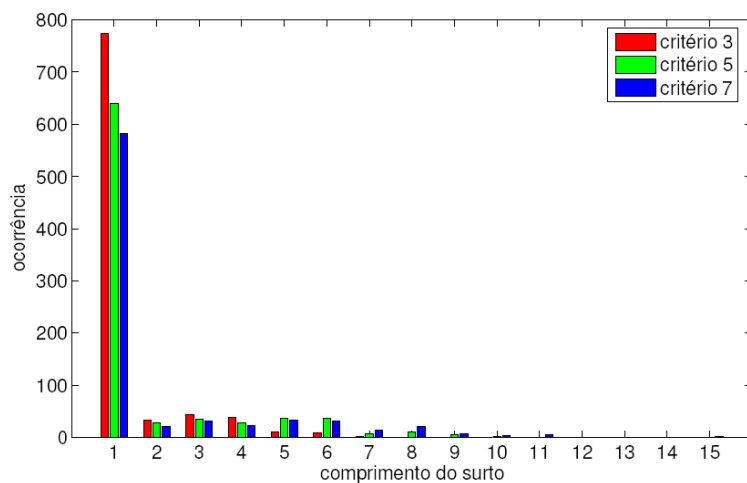


FIG. 6.7: Tamanho do surto de erros das sequências geradas segundo o modelo HMM com o conjunto de parâmetros 2

modelados pelo HMM com parâmetros dos conjunto 1 e conjunto 2, bem como pelo modelo de erros independentes.

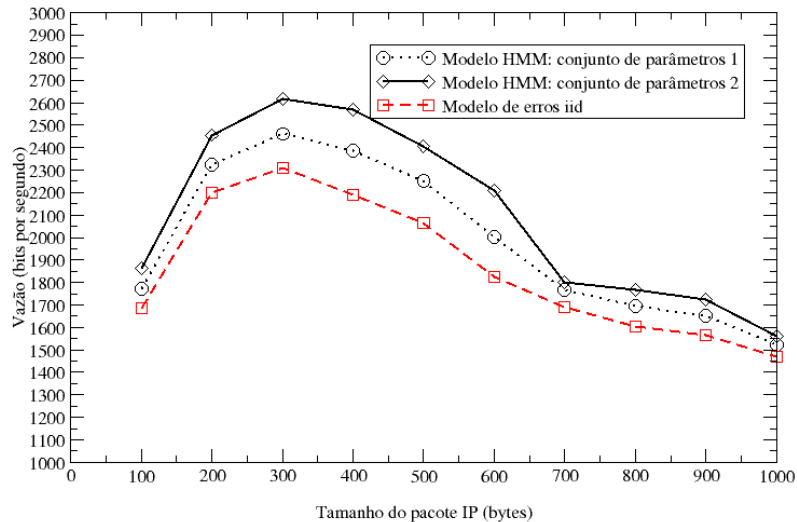


FIG. 6.8: Influência das probabilidades de erros de cada estado do Modelo HMM no desempenho dos protocolos TCP e de camada de enlace em canal modelados com P_{alvo} igual a 10^{-3} .

O gráfico da FIG. 6.8 evidencia que as vazões alcançadas pelo TCP em presença de erros correlacionados (conjunto de parâmetros 1 e 2) foram superiores ao desempenho do protocolo em presença de erros estatisticamente independentes. Os resultados mostram, por exemplo, que para tamanho do pacote IP igual a 400 bytes encontra-se uma redução no desempenho a partir do HMM com parâmetros do conjunto 2 em relação ao modelo de erros independentes de aproximadamente 15%. Este comportamento é esperado devido à característica de agrupamento dos erros correlacionados, ocasionando que um pacote contenha maior número de erros de bits em série (erros em surto) e outros pacotes não, resultando em uma menor taxa de perda de pacotes e, conseqüentemente, um melhor desempenho (maior vazão). Por outro lado, os erros *iid* são mais distribuídos ao longo da seqüência, resultando em um número maior de pacotes com pelo menos um erro para uma mesma BER.

Os processos de erros de bits se caracterizam por apresentarem diferentes tamanhos de surto. Um surto é definido pelo CCITT (CCITT) como sendo um grupo de bits no qual dois bits errados sucessivos estejam sempre separados por menos do que

um dado número L de bits corretos. O valor L é chamado de critério de comprimento de surto.

De acordo com a TAB. 6.4 e com as Figuras 6.6 e 6.7, percebe-se que as seqüência derivadas do modelo HMM segundo o conjunto de parâmetros 2 possuem um tamanho médio de surto maior para critérios 3, 5 e 7 do que o conjunto 1. Este fato explica a melhoria de desempenho obtida pelo conjunto 2 em relação ao conjunto 1 (FIG. 6.8), pois, conforme mencionado anteriormente, quanto maior o grau de concentração dos erros, menor a taxa de perda de pacotes, para uma mesma BER.

Corroborando com resultados anteriores, nas Figuras 6.9, 6.10 e 6.11 são apresentadas as comparações de desempenho do TCP em canais modelados por HMM (conjunto de parâmetros 1) e pelo modelo de erros independentes, com P_{alvo} igual a 10^{-3} , 10^{-4} e 10^{-5} , respectivamente.

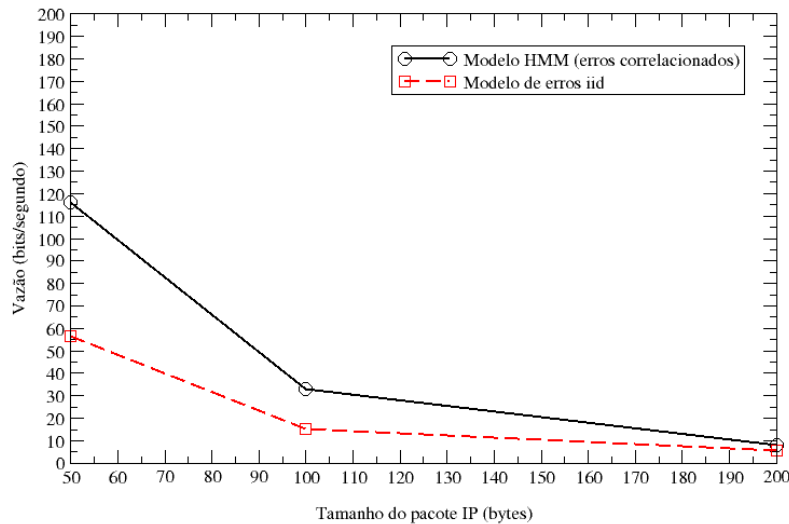


FIG. 6.9: Desempenho do protocolo TCP em presença de erros correlacionados e erros estatisticamente independentes em canal modelado com P_{alvo} igual a 10^{-3} .

Os resultados destas Figuras mostram que na medida que a P_{alvo} diminui, a diferença de desempenho do TCP em presença de erros correlacionados e erros *iid* também diminui. Na FIG. 6.10 vê-se que a vazão do TCP em presença de erros correlacionados é ligeiramente superior à vazão do TCP em presença de erros do tipo *iid* em canal modelado com P_{alvo} igual a 10^{-4} , ao passo que, na FIG. 6.11, a diferença de desempenho do TCP para canal modelado com P_{alvo} igual a 10^{-5} com erros correlacionados e erros estatisticamente

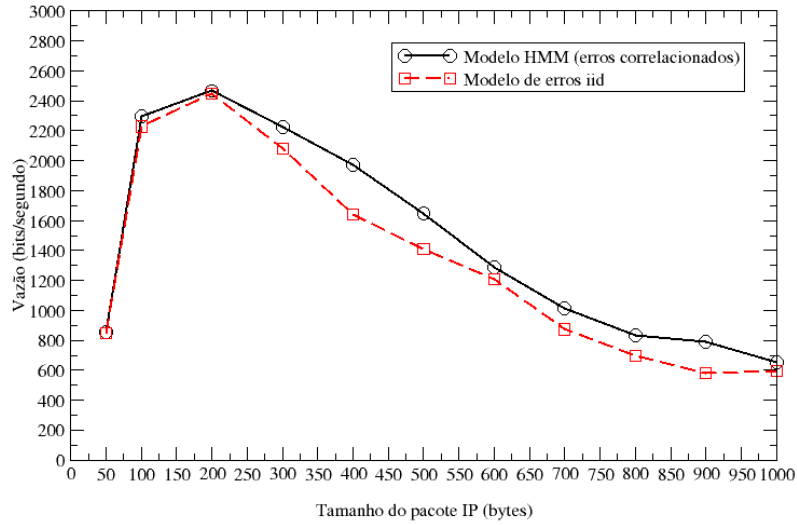


FIG. 6.10: Desempenho do protocolo TCP em presença de erros correlacionados e erros estatisticamente independentes modelados em canal modelado com P_{alvo} igual a 10^{-4} .

independentes foi praticamente nulo.

Em suma, esses resultados indicam que quanto menor a BER propiciada pela camada física do sistema de comunicação, menos importante é a correlação exibida pelos erros. Obviamente, esse resultados se referem ao ambiente simulado, que desconsidera os efeitos do congestionamento.

Portanto, conforme previsto em (ZORZI, 1999), tais resultados sugerem a importância da estrutura de correlação do processo de erros no desempenho dos protocolos, sobretudo em cenário com altas taxas de erros de bits, quando o modelo de erros independentes e identicamente distribuídos torna-se inadequado e pessimista na modelagem dos erros devido ao canal.

6.3 DESEMPENHO DO TCP EM CANAL HF

6.3.1 VAZÃO DAS VERSÕES DO PROTOCOLO TCP EM CANAL HF

A FIG. 6.12 ilustra o desempenho das versões do TCP em canal HF modelados com P_{alvo} igual a 10^{-3} .

Os resultados mostram que, mesmo com todas as melhorias implementadas desde o TCP Tahoe, o desempenho das versões do TCP é, em geral, baixo, e que as

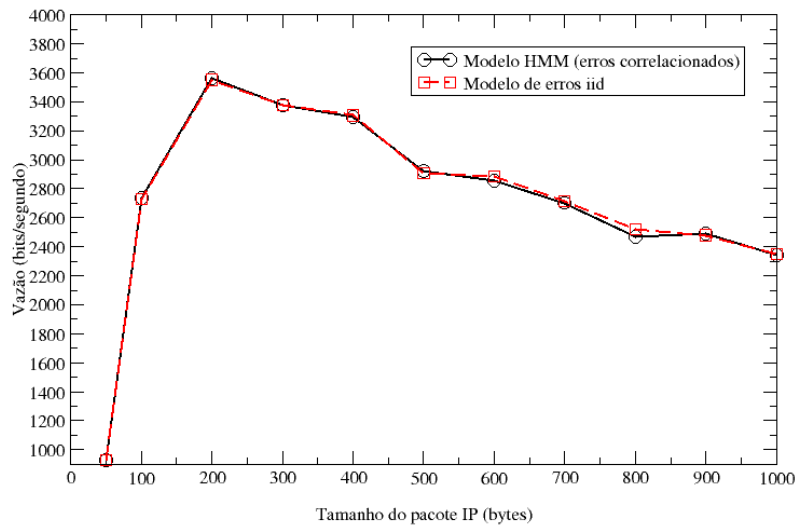


FIG. 6.11: Desempenho do protocolo TCP em presença de erros correlacionados e erros estatisticamente independentes em canal modelado com P_{alvo} igual a 10^{-5} .

versões apresentam resultados de vazão muito similares, exceção feita para o TCP FACK, que obteve desempenho ainda abaixo dos demais. Além disso, cabe notar que os melhores resultados foram obtidos para pacotes de 50 bytes, o que se deve à menor taxa de perda de pacotes (em torno de 28,5%) observada na FIG. 6.5.

O baixo desempenho das versões do TCP em canal HF decorre principalmente do aumento do temporizador *timeout* a cada perda de pacotes, de acordo com o algoritmo de Karn, abordado na Seção 2.4.3. Desta forma, em canais que apresentam um taxa de perda de pacote superior a 28 %, este aumento do temporizador torna-se elevado, levando o protocolo a esperar por um tempo excessivo na detecção da próxima perda de pacote, o que acarreta uma baixa vazão.

Um fator secundário que colabora para redução da vazão é a política de redução da janela de congestionamento para cada perda de pacote. Por exemplo, para tamanho de pacote de 50 bytes, os TCPs New Reno e Westwood, que usam mecanismos de controle do *cwnd* após a detecção da perda de pacotes, obtiveram resultados ligeiramente superiores ao TCP Reno, confirmando que o controle de *cwnd*, neste canal com altas taxas de perdas, contribui pouco para a melhoria da vazão.

O desempenho alcançado pelos TCPs que implementam o reconhecimento seletivo, tal como SACK e FACK, evidencia que os mecanismos de detecção rápida de

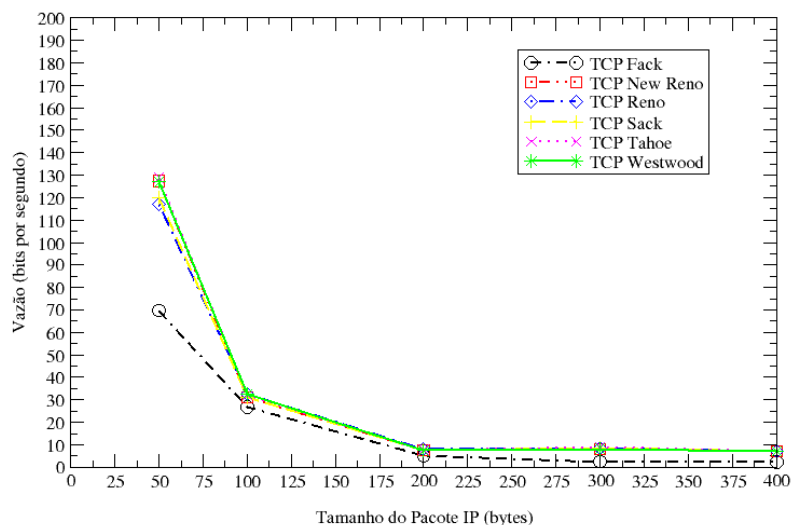


FIG. 6.12: Desempenho das versões do TCP sem a presença de protocolo de enlace.

múltiplas perdas de pacote não produziram o efeito esperado em canal HF. Neste caso, a limitação imposta no reconhecimento de até 3 blocos não contíguos presentes na mensagem de reconhecimento SACK e a alta taxa de perda de pacotes deste canal contribuíram para o baixo desempenho observado.

Outro aspecto a ser notado, é que o rudimentar TCP Tahoe produziu melhor vazão do que o TCP Reno, fato já apresentado em (ZORZI, 2000). Isso significa que a ausência do algoritmo *Fast Recovery* no TCP Tahoe foi determinante para o ganho de desempenho em relação ao Reno.

De posse desses resultados, conclui-se que as estratégias que envolvem as versões do TCP em canal HF apresentam desempenhos similares e baixa vazão, ocasionados pelo aumento excessivo do temporizador *timeout* e pela política de redução de *cwnd* a cada perda de pacotes.

Vale ressaltar ainda que os resultados aqui apresentados diferem dos reportados em outros trabalhos que mostram o ganho de desempenho das versões simuladas, o que pode ser explicado pelo fato de que os mesmos foram implementadas para o tráfego na Internet, o qual, em geral, apresenta congestionamentos e baixas taxas de perdas de pacote devido ao canal, ou ainda para canais sem fio, com taxas de perdas moderadas.

6.3.2 INFLUÊNCIA DO TAMANHO DO ARQUIVO A SER TRANSFERIDO NO DESEMPENHO DO PROTOCOLO TCP

Nesta seção são apresentadas as comparações de desempenho do TCP em função do tamanho do arquivo a ser transferido usando o protocolo de aplicação FTP. As Figuras 6.13 e 6.14 mostram o desempenho do TCP em canais modelados com P_{alvo} de 10^{-3} e 10^{-4} , respectivamente, quando são transmitidos arquivos de tamanho 10000 bytes e 100000 bytes.

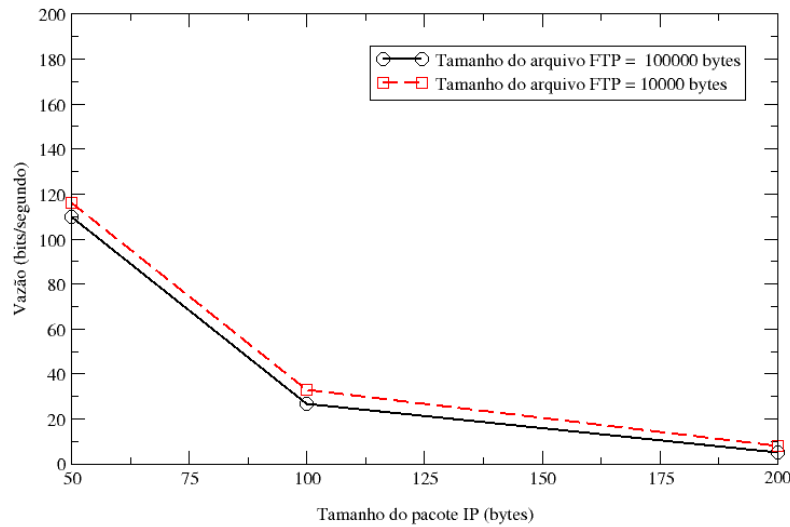


FIG. 6.13: Vazão do TCP em canal HF com P_{alvo} de 10^{-3} em função do tamanho do pacote IP, considerando a transferência de arquivos de 10000 e 100000 bytes.

Nos dois casos, o desempenho do TCP com arquivos de 10000 bytes foi superior ao obtido com arquivos de 100000 bytes.

De fato, em um canal que apresenta alta taxa de perda, a transferência de um arquivo de 100000 bytes produzirá uma quantidade de perdas de pacote superior ao caso de transferência de 10000 bytes. Além disso, a cada perda de pacote, o TCP Reno reduz a janela de congestionamento, devido aos algoritmos *Slow Start* e *Congestion Avoidance*, e aumenta o temporizador *timeout*.

Para avaliar a influência da redução de *cwnd* na diferença de desempenho do TCP nos arquivos de tamanho diferentes, são apresentadas nas Figuras 6.15 e 6.16 funções amostra da evolução de *cwnd* ao longo da transferência de arquivo em canal modelado

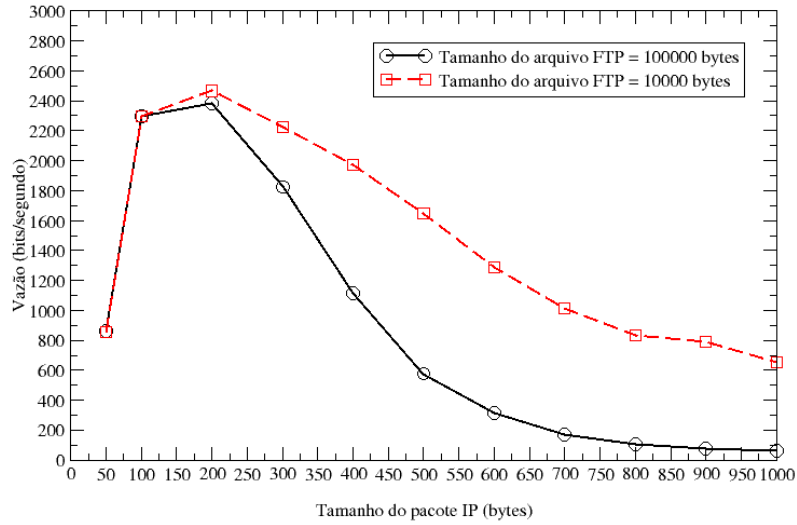


FIG. 6.14: Vazão do TCP em canal HF com P_{alvo} de 10^{-4} em função do tamanho do pacote IP, considerando a transferência de arquivos de 10000 e 100000 bytes.

com P_{alvo} igual a 10^{-3} . Estas curvas foram obtidas considerando tamanho de pacote IP igual a 50 bytes e transferência de arquivos de tamanho 10000 bytes (FIG. 6.15) e 100000 bytes (FIG. 6.16).

A evolução de $cwnd$ ao longo do tempo para transferência de um arquivo de tamanho de 10000 bytes, mostrada na FIG. 6.15, resulta num valor médio de 1.65. Para o caso de transferência de 100000 bytes de dados mostrada na FIG. 6.16, o $cwnd$ médio é de 1.74. A pequena diferença observada indica que $cwnd$ médio mantém-se razoavelmente constante independentemente do tamanho do arquivo FTP.

Assim sendo, o elevado aumento do $timeout$ é uma justificativa plausível que explica a diferença de desempenho apresentada nas Figuras 6.13 e 6.14. Em outras palavras, tendo em vista que a transferência de 100000 bytes produz maiores quantidades de perdas de pacote, o valor de $timeout$ será dobrado em um número maior de vezes, levando o protocolo TCP a esperar por um longo tempo na constatação da próxima perda de pacote. Ou seja, o protocolo TCP se torna mais ocioso em um cenário de transferência de 100000 bytes do que 10000 bytes, resultando na diferença de desempenho mencionada.

Além disso, a FIG. 6.14 mostra que a partir do tamanho do pacote IP de 300 bytes a diferença de desempenho se intensifica. Isto se deve ao fato de que a taxa de

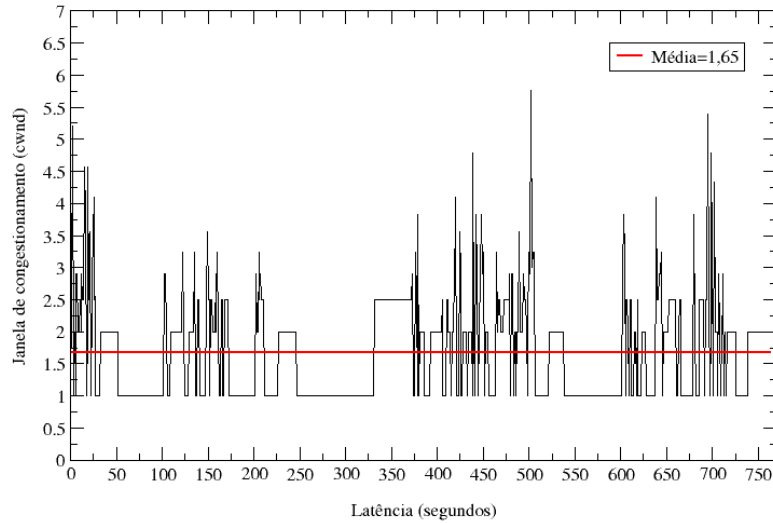


FIG. 6.15: Evolução de $cwnd$ do TCP em função do tempo em canal modelado com P_{alvo} igual a 10^{-3} submetido a transferência de arquivo de tamanho de 10000 bytes usando FTP.

perda de pacotes aumenta com o tamanho do pacote IP, conforme mostrado na FIG. 6.5. Assim, este aumento na diferença de desempenho do TCP se deve a dois fatores: a maior quantidade de perdas de pacote na transferência de 100000 bytes de dados e a maior incidência de perdas de pacote em função do aumento do tamanho do pacote IP. Tais fatores levam a um aumento maior do temporizador *timeout* na transferência de 100000 bytes em relação ao caso do arquivo de 10000 bytes, gerando o maior espaçamento entre as curvas de desempenho da FIG. 6.14 a partir de tamanho do pacote IP de 300 bytes.

Contudo, observa-se na mesma figura que os desempenhos foram similares para tamanhos de pacotes de até 100 bytes. Tal comportamento se explica pelo efeito conjunto de dois fatores: a menor incidência de perda de pacotes para tamanhos menores de blocos (ver FIG. 6.5), diminuindo o efeito nocivo do aumento do temporizador *timeout* e o maior valor de $cwnd$ médio em transferência de 100000 bytes em relação às transferências de 10000 bytes, conforme as funções-amostras das Figuras 6.17 e 6.18, que foram simulados com tamanho de pacote IP igual a 50 bytes.

As Figuras 6.17 e 6.18, indicam um aumento de aproximadamente 20% do $cwnd$ médio para o caso de transferência de 100000 bytes, em relação ao caso de 10000 bytes.

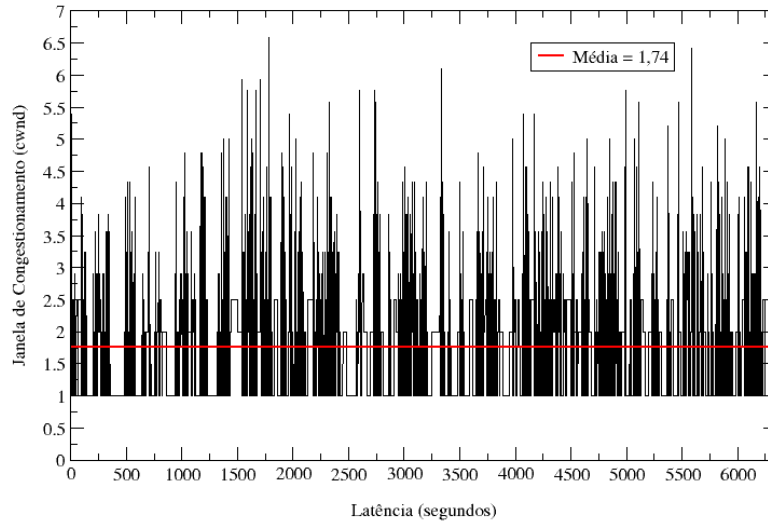


FIG. 6.16: Evolução de $cwnd$ do TCP em função do tempo em canal HF modelado com P_{alvo} igual a 10^{-3} submetido a transferência de arquivo de tamanho de 100000 bytes usando FTP.

A avaliação da influência do tamanho do arquivo FTP no desempenho do TCP para canais modelados com P_{alvo} igual a 10^{-5} não foi realizado, tendo em vista que tais valores de BER não correspondem à realidade dos canais HF, principalmente, porque nesta seção não se leva em conta a presença de códigos corretores de erro do tipo FEC e/ou ARQ na camada de enlace.

6.4 DESEMPENHO DAS VERSÕES DO PROTOCOLO TCP EM PRESENÇA DE PROTOCOLO DE ENLACE COM FEC

Na presente seção será apresentado o desempenho das versões do TCP em presença de camada física com mecanismo FEC com taxa 1/2 em canal HF. As simulações que envolvem o mecanismo FEC foram realizadas segundo o modelo proposto na Seção 5.7.

Os resultados obtidos da FIG.6.19 indicam que, excetuando-se os TCPs SACK e FACK, as vazões alcançadas pelas demais versões do TCP investigadas, em função do tamanho do pacote IP, mostraram-se bastante semelhantes.

Os TCPs SACK e FACK, que implementam o reconhecimento seletivo, obtiveram desempenho superior para vários tamanhos de pacotes diferentes, indicando que

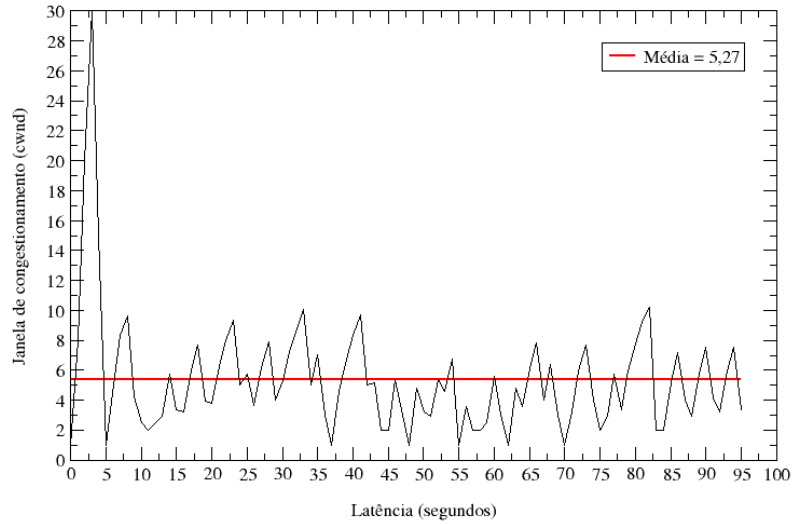


FIG. 6.17: Evolução de $cwnd$ do TCP em função do tempo em canal modelado com P_{alvo} igual a 10^{-4} submetido a transferência de arquivo de tamanho de 10000 bytes via FTP.

a baixa ocorrência de perda de pacotes, devido ao mecanismo FEC, e a presença de correlação entre os erros, contribuíram para o pleno funcionamento do mecanismo de reconhecimento seletivo. Em outras palavras, a presença de poucas perdas de pacote, em função do FEC, bem como a ocorrência de perdas em surto, permitiram que na mensagem SACK estivessem especificados até 3 blocos de dados não contíguos, agilizando a recuperação de múltiplas perdas de pacote dentro de uma mesma janela de transmissão.

6.5 DESEMPENHO DO TCP EM PRESENÇA DE PROTOCOLO DE ENLACE COM ARQ-SR EM CANAL HF

Nesta seção serão apresentadas os resultados das simulações que visam buscar valores dos parâmetros do TCP e do protocolo de camada de enlace com mecanismo ARQ-SR que propiciem melhores resultados de desempenho. Assim sendo, foi estabelecido um conjunto de valores padrão, especificados na TAB.6.1, que foram utilizados para as avaliações da influência de cada parâmetro no desempenho do TCP em presença de protocolo de enlace ARQ.

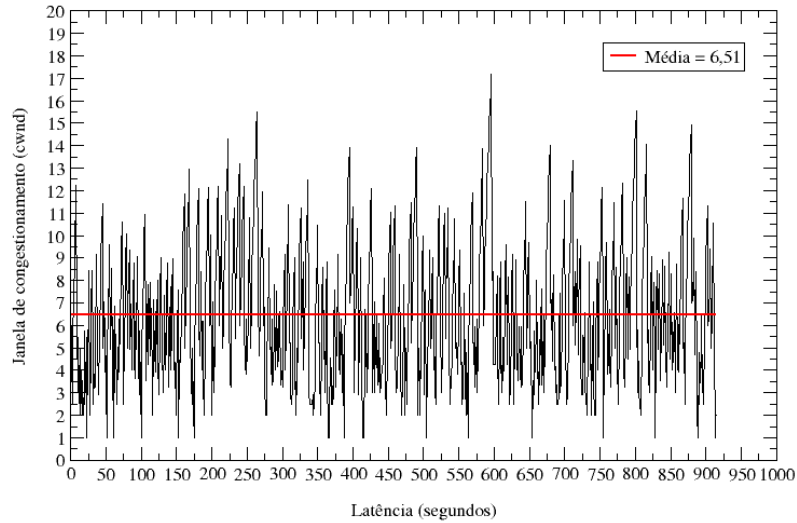


FIG. 6.18: Evolução de *cwnd* do TCP em função do tempo em canal modelado com P_{alvo} igual a 10^{-4} submetido a transferência de arquivo de tamanho de 100000 bytes via FTP.

6.5.1 OTIMIZAÇÃO DOS PARÂMETROS DO PROTOCOLO TCP

6.5.1.1 INFLUÊNCIA DO *TIMEOUT* MÍNIMO

Conforme mencionado na Seção 6.1, no início de cada conexão TCP, o valor do temporizador *timeout* é obtido com base no RTT estimado da abertura de conexão, estabelecido por um pacote de 40 bytes. Quando o transporte de dados se inicia com tamanhos maiores de pacote IP, ocorrem sucessivos *timeouts* devido ao retardo de segmentação e remontagem do pacote IP, serialização dos quadros na fila de transmissão e aos erros e retransmissões necessárias na camada de enlace. Esta anormalidade se estende até que o temporizador *timeout* alcance um valor suficiente para sobrepôr as fontes de retardo mencionadas.

Em (ABDELMOUMEN, 2004) é proposto uma maneira de se aumentar o valor do *timeout* mínimo, que será denotado por $timeout_{min}$, para evitar os sucessivos estouros de *timeouts* e, conseqüentemente, melhorar o desempenho do TCP. Assim, com o objetivo de avaliar a influência do $timeout_{min}$ no desempenho, foram feitas simulações variando este parâmetro a partir do valor de 20ms. Os resultados obtidos são mostrados

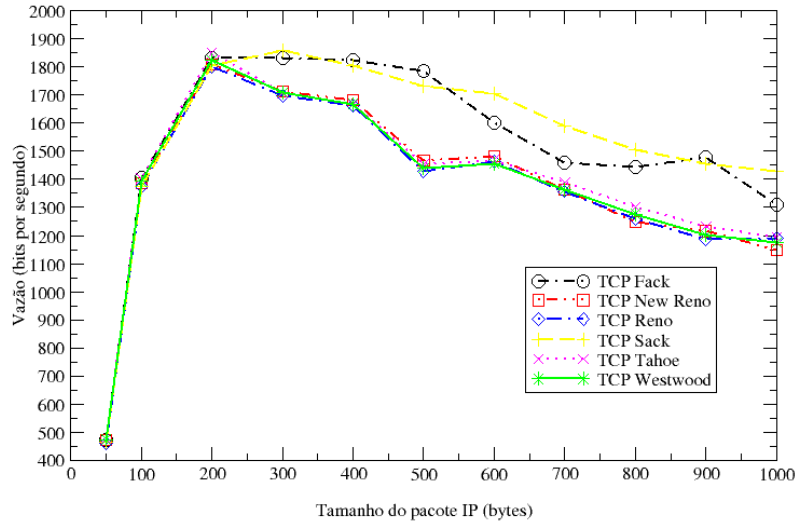


FIG. 6.19: Desempenho das versões do TCP em presença do protocolo de enlace com FEC Taxa 1/2.

na FIG. 6.20.

De acordo com os resultados aqui apresentados (FIG. 6.20) o valor de $timeout_{min}$ que propiciou as melhores vazões foi de 5 segundos.

Observa-se ainda uma lenta queda de desempenho para valores de $timeout_{min}$ superiores ao de vazão máxima. Por exemplo, o valor da vazão obtida em $timeout_{min} = 20ms$ somente é repetida em $timeout_{min} = 25,5$ segundos. Este comportamento sugere que, em geral, a camada de enlace, de maneira eficiente, oculta para o TCP as perdas de canal, de forma que o aumento do $timeout_{min}$ a partir do ponto máximo resulta em uma pequena degradação de desempenho. Em outras palavras, o valor de persistência igual a 3, permite que a maior parte dos erros no canal seja recuperada pela camada de enlace, reduzindo a probabilidade de ocorrência de *timeouts*. Pode-se dizer que a queda de desempenho a partir do ponto máximo é devida aos poucos casos em que a camada de enlace não consegue recuperar o quadro com erros, repassando a tarefa de retransmissão do pacote inteiro para camada de transporte.

Por outro lado, verifica-se uma degradação severa de desempenho adotando-se valores de $timeout_{min}$ inferiores ao valor ótimo. Isto se deve às retransmissões desnecessárias do TCP no início da conexão, em função dos sucessivos estouros de *timeouts*, provocado pelo acréscimo de retardo devido ao processo de segmentação e re-

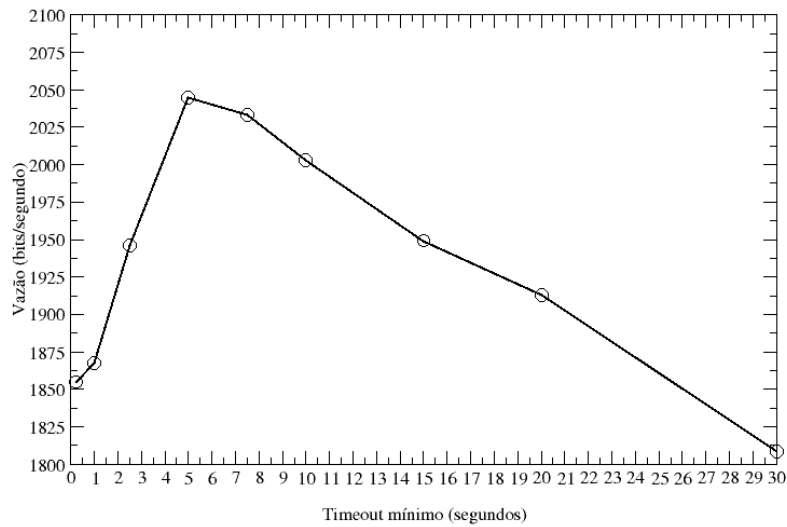


FIG. 6.20: Influência do *timeout* mínimo no desempenho do protocolo TCP em canal HF.

montagem dos quadros, pela serialização dos quadros na fila e pelas retransmissões dos quadros na camada de enlace.

6.5.1.2 INFLUÊNCIA DO TAMANHO DA JANELA DE CONGESTIONAMENTO INICIAL

A FIG. 6.21 ilustra o desempenho do TCP em presença de protocolo de enlace ARQ-SR em função do tamanho inicial da janela de congestionamento com $timeout_{min} = 20ms$ e $timeout_{min} = 20s$.

Os resultados desta figura mostram que com $timeout_{min} = 20ms$ o aumento do tamanho inicial de *cwnd* produz uma considerável redução da vazão. Na verdade, em virtude do mecanismo da janela que permite injetar um número maior de pacotes IP no início da conexão, o efeito dos *timeouts* sucessivos durante o início da conexão, descrito na seção anterior, é intensificado, sendo a razão principal do comportamento mostrado na FIG. 6.21.

Para avaliar completamente esta estratégia, foi adotada novamente a solução de aumentar o valor do *timeout* mínimo para 20 segundos, obtendo os resultados também contidos na FIG. 6.21. Como esperado, mantendo $timeout_{min} = 20s$, o aumento do tamanho inicial de *cwnd* resulta na melhoria da vazão, pois evita o efeito de sucessivos

estouros de *timeouts* e aumenta a taxa de utilização do canal em função do maior número de pacotes IP enviados no início da conexão.

É válido mencionar mais uma vez que as vazões alcançadas foram obtidas desconsiderando os efeitos do congestionamento. Nesses casos, as conclusões feitas podem ser outras.

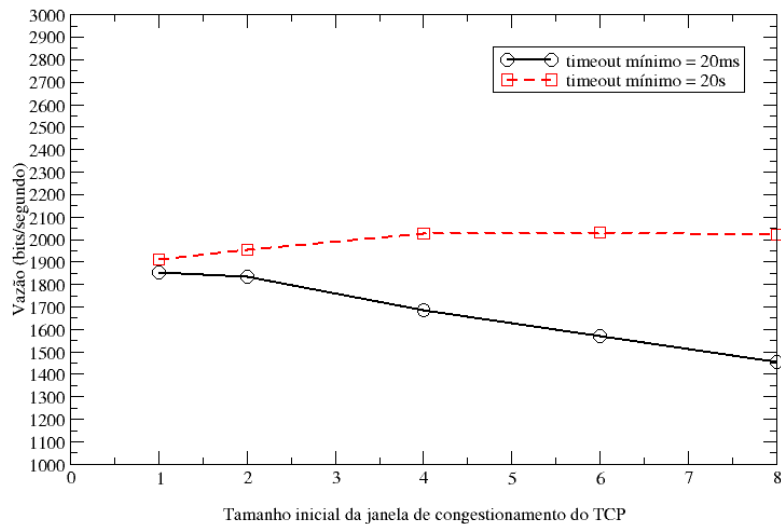


FIG. 6.21: Influência da janela de congestionamento inicial do TCP na vazão.

6.5.2 OTIMIZAÇÃO DOS PARÂMETROS DO PROTOCOLO DE ENLACE COM ARQ-SR EM CANAL HF

6.5.2.1 INFLUÊNCIA DO TAMANHO DO QUADRO

A FIG. 6.22 ilustra a influência do tamanho do quadro no desempenho do TCP em presença de protocolo de enlace com SR. Observa-se que à medida que o tamanho do quadro aumenta, o desempenho piora consideravelmente, de acordo com o apresentado na Seção 6.2.1. Por exemplo, comparando-se os tamanhos de quadro de 25 e 100 bytes, a redução da vazão chega a 82,4%, mesmo com o maior *overhead* no primeiro caso em relação ao segundo.

Em face destes resultados, pode-se concluir que o desempenho do protocolo TCP em presença de protocolo de enlace com SR é extremamente sensível ao tamanho do quadro.

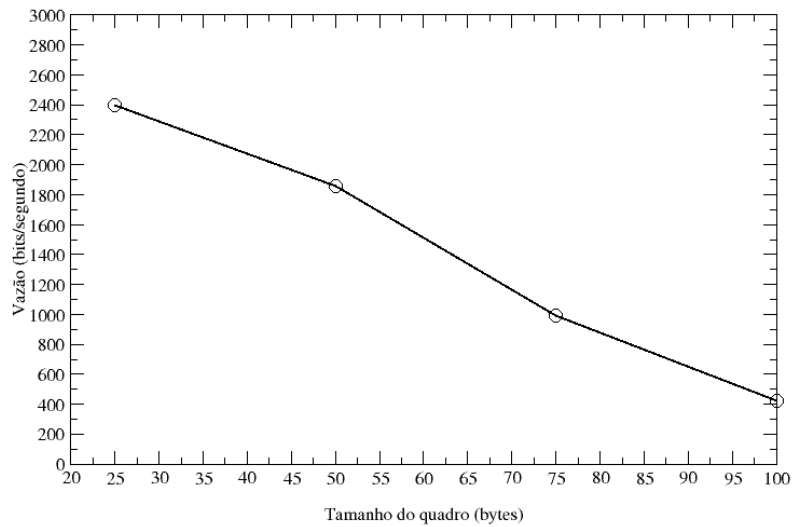


FIG. 6.22: Influência do tamanho do quadro na vazão do protocolo TCP em canal HF.

Em uma segunda avaliação foi abordada a influência no desempenho do tamanho do quadro de reconhecimento (*idle frame* ou quadro NACK) na camada de enlace, obtendo-se os resultados mostrados na TAB. 6.5

TAB. 6.5: Vazão do protocolo TCP em presença protocolo de enlace com SR em função do tamanho do quadro do canal reverso

Tamanho do quadro no canal direto (" <i>forward</i> ") (bytes)	Tamanho do quadro no canal reverso (" <i>reverse</i> ") (bytes)	Vazão (bits/segundo)
50	50	1854.92
50	25	1860.44
75	75	994.31
75	25	1025.71
100	100	421.684
100	25	428.296

Verifica-se uma nesta tabela uma ligeira melhoria do desempenho para tamanhos menores de quadros no canal de retorno em relação a quadros de tamanhos iguais no canal direto e no reverso. De fato, a melhoria não é considerável, pois nos casos do tamanhos de quadro de retorno igual a 25 bytes, a cada pacote ACK do TCP composto de 40 bytes necessita de 3 quadros de 25 bytes ($\frac{40}{(25-6)} = 2,1$) para o transporte no canal

de retorno, aumentando o retardo no processo de reconhecimento do TCP.

6.5.2.2 INFLUÊNCIA DA PERSISTÊNCIA (δ)

O desempenho do TCP em presença do protocolo de enlace ARQ-SR em função da persistência (δ) é mostrado na FIG. 6.23. Este resultado demonstra que a medida que δ é incrementado observa-se uma melhoria considerável na vazão, fato evidenciado em (ABDELMOUMEN, 2004). De fato, o parâmetro δ pode ser encarado como a medida do grau de confiabilidade proporcionado pela camada de enlace, que oculta para o TCP as perdas devidas ao canal, evitando os sucessivos aumentos do temporizador *timeout* e as drásticas reduções de *cwnd*, permitindo assim o aumento da taxa de envio dos pacotes IP.

Foram apresentados na FIG. 6.23 os resultados com δ limite de 3^{12} .

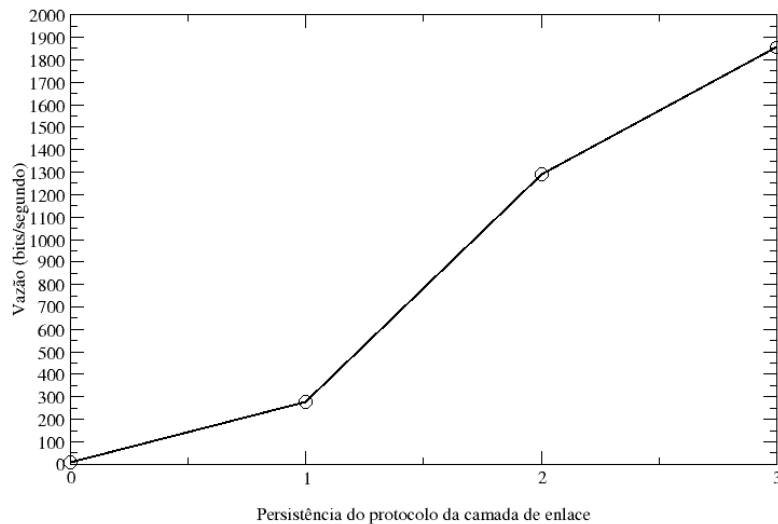


FIG. 6.23: Influência da persistência do protocolo de enlace na vazão do protocolo TCP em canal HF.

6.5.2.3 INFLUÊNCIA DA TEMPORIZADOR NACK (*RETRANSMIT TIMER*)

Como apresentado na Seção 5.6, o temporizador NACK (*Retransmit Timer*), que será aqui denotado por t_{nak} , influi diretamente na rapidez de recuperação dos quadros

¹²Porque foram constatadas alguns erros no módulo implementado quando se tentava simular com $\delta > 3$.

com erros ou perdidos. Em particular, na implementação do protocolo de enlace adotado neste trabalho, a unidade do *Retransmit Timer* é o número de transmissões sucessivas de *idle frames* a partir do qual será retransmitido o quadro NAK.

A FIG. 6.24 apresenta a influência do temporizador NACK utilizado pelo protocolo de enlace no desempenho dos protocolos. Os resultados mostram que para $t_{nak} = 5$ a vazão obtida alcançou o ponto máximo. Para valores de $t_{nak} > 5$, o aumento do retardo na recuperação dos quadros com erros foi fundamental para a redução da vazão. Para valores de $t_{nak} < 5$, a redução do desempenho foi motivada pelo rápido reenvio do quadro NAK por parte do receptor, ocasionando mais de uma retransmissão de um mesmo quadro, aumentando desta forma o retardo na recuperação dos quadros com erros e, conseqüentemente, diminuindo a vazão.

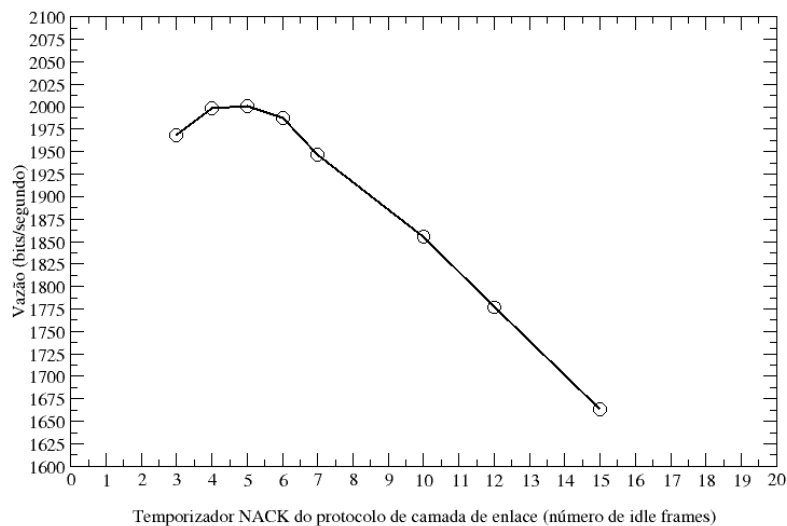


FIG. 6.24: Influência do valor do temporizador NACK do protocolo de enlace no desempenho dos protocolos em canal HF.

6.5.2.4 INFLUÊNCIA DO TAMANHO DO CABEÇALHO DO QUADRO NO DESEMPENHO DOS PROTOCOLOS EM CANAL HF

Esta seção apresenta resultados das simulações realizadas com base na alteração do tamanho do cabeçalho do quadro. A FIG. 6.25 ilustra a comparação de desempenho em face dos tamanhos do cabeçalho do quadro de 6 e 20 bytes. Vale ressaltar que, conforme mencionado no início do presente capítulo, o valor de 20 bytes foi adotado com base

nos quadros especificados no AX.25 (ver FIG. 3.6).

Como esperado, observa-se, na referida figura, a redução de desempenho com o aumento do cabeçalho do quadro, bem como o comportamento similar em ambos os gráficos da vazão em função do tamanho do pacote IP.

Considerando o quadro de tamanho 50 bytes e aumentando-se o tamanho do cabeçalho de 6 para 20 bytes há uma redução na capacidade de transportar dados (*payload*) por quadro de aproximadamente 31,8%, devido ao maior *overhead* para o quadro com cabeçalho maior. Para um tamanho de pacote IP igual a 400 bytes, verifica-se uma redução de desempenho, para os casos mencionados, de aproximadamente 31%, mostrando a coerência dos resultados obtidos.

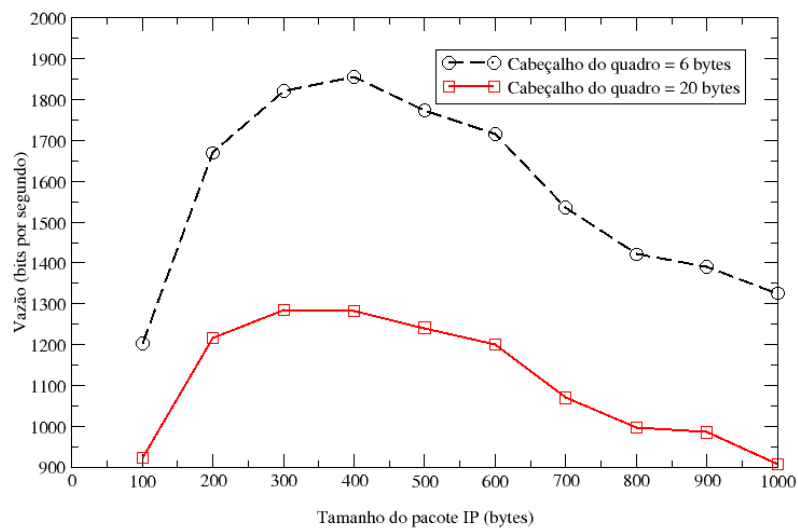


FIG. 6.25: Influência do tamanho do cabeçalho do quadro no desempenho dos protocolos em canal HF.

6.5.3 VAZÃO DO PROTOCOLO TCP EM PRESENÇA DO PROTOCOLO DE ENLACE COM ARQ-SR COM PARÂMETROS CRITERIOSAMENTE ESCOLHIDOS

Esta seção pretende mostrar a importância de utilizar um protocolo de camada de enlace ARQ com parâmetros criteriosamente escolhidos para cada tipo de cenário.

Após as avaliações das influências dos parâmetros do protocolo de enlace ARQ realizadas nas seções anteriores, foram estabelecidos os novos valores de parâmetros

mostrados na TAB. 6.2. Assim sendo, a FIG. 6.26 apresenta a comparação de desempenho do protocolo TCP em presença de protocolos de enlace ARQ configurados de acordo com as Tabelas 6.1 e 6.2.

Esta figura evidencia a melhoria do desempenho resultante da utilização do protocolo de camada de enlace ARQ com base nos parâmetros da TAB. 6.2. Percebe-se que, para um tamanho de pacote IP igual a 300 bytes, a melhoria na vazão corresponde a aproximadamente 26%. Comparando-se os parâmetros das duas tabelas mencionadas acima, verifica-se que essa sensível melhora de desempenho foi obtida alterando apenas o tamanho do quadro e o temporizador NACK.

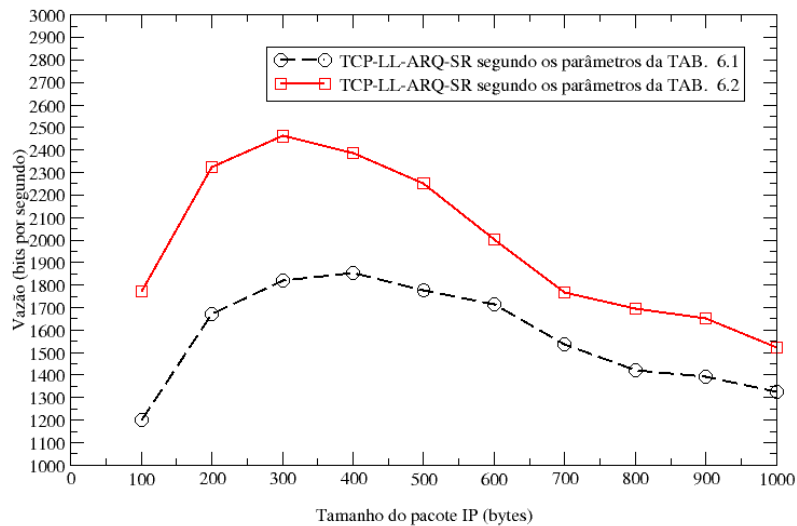


FIG. 6.26: Comparação de desempenho dos protocolos de enlace otimizados segundo TAB. 6.1 e não otimizados segundo TAB. 6.2.

6.6 RESUMO

Neste capítulo, foram apresentadas comparações de desempenho de algumas das estratégias de melhoria do protocolo TCP em canal HF, para o ambiente de simulação proposto no Capítulo 5.

Foram inicialmente avaliadas a taxa de perda de pacotes (usando o modelo HMM) em função do tamanho do pacote IP, e a influência dos modelos de erros independentes e correlacionados no desempenho dos protocolos de camadas superiores.

Em seguida foram feitas avaliações de desempenho das versões do TCP em canal HF e, posteriormente, com a presença de camada física com FEC. A influência do tamanho do arquivo a ser transferido por meio do FTP no desempenho do TCP também foi avaliada.

Por fim, na busca dos parâmetros e técnicas que proporcionem melhores resultados de vazão, foram avaliadas as influências do *timeout* mínimo e tamanho inicial de *cwnd* do TCP, bem como dos seguintes parâmetros do protocolo de enlace ARQ: tamanho do quadro, persistência, temporizador NACK e tamanho do cabeçalho do quadro. A partir dos resultados dos parâmetros da camada de enlace obtidos, foi realizada a comparação de desempenho do TCP em presença de protocolo de enlace com ARQ-SR para 2 conjuntos de parâmetros, sendo um desses conjuntos considerado otimizado.

Os resultados obtidos mostram a significativa superioridade de desempenho do TCP em presença de protocolo de camada de enlace com mecanismo ARQ e/ou camada física com FEC em relação ao TCP em canal HF. Esses resultados também evidenciam que as versões do TCP obtiveram desempenhos semelhantes, exceção feita aos TCPs SACK e FACK na presença de camada física com taxa $1/2$, que obtiveram resultados de vazão superiores ao demais. Verificou-se ainda a importância de contar com uma camada de enlace ARQ com parâmetros criteriosamente escolhidos para conferir melhor desempenho ao TCP.

Em relação aos resultados obtidos envolvendo os modelos de erros, observou-se a melhoria de desempenho prevista para modelos de erros correlacionados, em relação aos modelos de erros *iid*.

No capítulo seguinte, são apresentadas as conclusões gerais do trabalho.

7 CONCLUSÕES E TRABALHOS FUTUROS

7.1 PRINCIPAIS CONCLUSÕES

Nesta dissertação, foi realizado um estudo com o objetivo de avaliar o desempenho do TCP em canal HF, bem como algumas estratégias envolvendo TCP com mecanismos ARQ da camada de enlace e FEC da camada física. Este estudo contemplou:

- a implementação de modelos de erros correlacionados e independentes que se aproximassem do comportamento do canal HF e análise de sua influência no desempenho dos protocolos das camadas de enlace e de transporte;
- a avaliação do desempenho das versões do TCP em canal HF;
- a análise da influência do tamanho do arquivo FTP no desempenho do TCP em canal HF;
- a avaliação do desempenho das versões do TCP em presença de protocolo de camada de enlace com mecanismo FEC com Taxa 1/2;
- a avaliação das seguintes estratégias de melhoria da vazão do TCP em canal HF: TCP-LL-FEC-Taxa 1/2, TCP-LL-FEC-Taxa 2/3, TCP-LL-ARQ-SR, TCP-LL-ARQ-SR-FEC-Taxa 1/2 e TCP-LL-ARQ-SR-FEC-Taxa 2/3;
- a análise da influência dos parâmetros *timeout* mínimo e *cwnd* inicial do protocolo TCP na vazão;
- a análise da influência dos parâmetros de camada de enlace ARQ na vazão do TCP.

Todas as avaliações de desempenho aqui apresentadas foram obtidas considerando um cenário (discutido no CAP. 5) com as seguintes características:

- uso da topologia ponto-a-ponto;
- canal dedicado, ou seja, ausência de mecanismo de acesso ao meio;
- ausência de congestionamento;

- presença de canal com altas taxas de erro;
- presença de erros correlacionados na camada física;
- baixas taxas de transmissão;
- retardos moderados de propagação e
- ausência de erros no canal de retorno.

Com base neste cenário apresentado e nos modelos adotados no CAP.5, chegou-se às seguintes conclusões relativas aos processos de erros:

- De acordo com a sequências binárias geradas pelo Modelo HMM, a taxa de perdas de pacotes aumenta consideravelmente com o aumento do tamanho do blocos de dados (pacote IP ou quadro);
- Modelando o canal com P_{alvo} igual a 10^{-3} , o desempenho dos protocolos TCP e de camada de enlace ARQ em presença de erros correlacionados foi sempre superior se comparado com a presença de erros independentes.
- Para canais modelados com P_{alvo} igual a 10^{-4} , o desempenho do protocolo TCP em presença de erros correlacionados foi ligeiramente superior se comparado com a presença dos erros independentes;
- Por outro lado, em canais modelados com P_{alvo} igual a 10^{-5} , o desempenho do protocolo TCP em presença de erros correlacionados foi muito similar ao obtido na presença dos erros independentes.

Os resultados referentes ao desempenho das versões do TCP em canal HF mostraram que todas as versões simuladas na presente dissertação apresentaram baixo desempenho, bem como não se observou significativas diferenças de desempenho entre as mesmas. Tal fato nos leva a concluir que, em canais com altas taxas de perdas, o efeito nocivo, decorrente do aumento considerável do temporizador *timeout*, tornando o protocolo ocioso na detecção da próxima perda de pacote e das sucessivas reduções de *cwnd*, prevaleceu sobre as técnicas implementadas por tais versões. Desta forma, a viabilidade de se empregar aplicações por meio do TCP em canal HF fica condicionada à

presença de protocolo de camada de enlace com mecanismo ARQ e/ou FEC na camada física.

Em relação às estratégias de melhoria do TCP em canal HF, os resultados mostraram que o desempenho obtido pelo TCP em presença de protocolos de enlace com mecanismos ARQ e/ou de camada física com FEC foi sempre superior ao obtido pelas versões do TCP sem camada de enlace com ARQ. Além do mais, a estratégia TCP-LL-ARQ-SR obteve melhores resultados em relação às demais estratégias.

Considerando a influência do tamanho do arquivo FTP no desempenho do TCP em canal HF, pode-se concluir que, considerando canais com P_{alvo} correspondente a 10^{-3} e a 10^{-4} , o desempenho do TCP submetido à transferência de menores arquivos, em geral, foi superior ao obtido na transferência de arquivos maiores, exceção feita para o caso com P_{alvo} de 10^{-4} , onde para tamanhos de pacote IP menores os desempenhos para ambos os tamanhos de arquivos foram semelhantes.

As avaliações de desempenho das versões do TCP em presença de camada física com mecanismo FEC resultaram nas seguintes conclusões:

- Excetuando-se os TCPs SACK e FACK, as demais versões apresentaram desempenho similares;
- Em virtude da baixa ocorrência de perda de pacotes, devido ao emprego do código corretor de erro avante (FEC), as técnicas envolvendo reconhecimento seletivo dos TCPs SACK e FACK proporcionaram melhor desempenho em relação às demais versões do TCP que não empregam reconhecimento seletivo. Isso foi verificado para vários tamanhos de pacotes.

A utilização de valores maiores do *timeout* mínimo resultou em significativa melhora no desempenho do TCP em presença de protocolo de enlace ARQ, pois evitou as retransmissões desnecessárias dos pacotes IP decorrentes do aumento de retardo introduzido pela segmentação e remontagem desses pacotes, serialização do quadros na fila de transmissão e das retransmissões dos quadros com erros.

A estratégia de melhoria do TCP em presença de protocolo de camada de enlace ARQ referente ao aumento do tamanho inicial de *cwnd* mostrou-se promissora somente quando o valor do *timeout* mínimo foi superior ao aumento do retardo introduzido pela camada de enlace. Para *timeouts* mínimos menores, a referida estratégia mostrou-se ineficiente.

Analisando a influência dos parâmetros da camada de enlace ARQ no desempenho dos protocolos das camadas superiores, pode-se concluir que:

- A medida que o tamanho do quadro aumenta, a vazão é reduzida consideravelmente, mostrando que o desempenho é extremamente sensível a este parâmetro.
- O aumento da persistência de 1 a 3 resultou em uma visível melhoria de desempenho.

7.2 TRABALHOS FUTUROS

A partir dos resultados e conclusões apresentadas sugerimos a seguir futuros estudos que poderão ser realizados.

1. Obtenção de dados reais em transmissões HF, para realizar treinamento do Modelo HMM levando em consideração parâmetros da camada física, tais como de técnica de correção de erro, RSR, espalhamento Doppler, etc.

2. Analisar o desempenho das versões TCP em presença de protocolo de camada de enlace com ARQ-SR em canal HF.

3. Avaliar o desempenho do TCP em presença de protocolo de camada de enlace com repetição seletiva em canal HF, considerando os aspectos de congestionamento.

4. Implementar no *ns-2* um protocolo de enlace com repetição seletiva e parâmetros de persistência superior a 3, para avaliar em melhores condições o efeito desse parâmetro na vazão do TCP.

5. Implementar no *ns-2* um protocolo de enlace cujos parâmetros, tais como taxa do código corretor de erros variem dinamicamente de acordo com a qualidade do canal.

6. Avaliar o desempenho da estratégia *TCP Spoofing* em canal HF considerando os seguintes casos: a presença e ausência de protocolo de camada de enlace com mecanismo ARQ.

7. Realizar avaliação de desempenho considerando a presença de erros no canal reverso.

8. Avaliar o desempenho das estratégias baseado em UDP em presença de protocolo de camada de enlace ARQ e de camada física com FEC submetidos à aplicação TFTP (*Trivial File Transport Protocol*).

9. Em aplicações militares o uso freqüente de sinais de reconhecimento não é adequado, pois as emissões de sinais eletromagnéticos contribuem por denunciar a posição de uma tropa em território inimigo. Desta forma, uma sugestão de trabalho futuro seria avaliar

o desempenho das estratégias baseado em UDP em presença de camada física com FEC submetidos à aplicação TFTP (*Trivial File Transport Protocol*).

8 REFERÊNCIAS BIBLIOGRÁFICAS

- ABDELMOUMEN, R., MALLI, M. e BARAKAT, C. Analysis of TCP Latency over Wireless Links Supporting FEC/ARQ-SR for Error Recovery. *IEEE International Conference on Communications (ICC) and Personal Communications Magazine*, 27(1):3994–3998, June 2004.
- ALLAN, M. On the Generation and Use of TCP Acknowledgments. *ACM Computer Communication Review*, October 1998.
- BALAKRISHNAN, H., PADMANABHAN, S., SESHAN, S. e KATZ, R. A Comparison of Mechanisms for Improving TCP Performance over Wireless Links. *IEEE/ACM Transactions on Networking*, 5(6), December 1997.
- BANSAL, D., CHANDRA, A. e SHOREY, R. An extension of the TCP flow control algorithm for Wireless Networks. *Proceedings of the IEEE International Conference on Personal Wireless Communication*, February 1999.
- BARMAN, D., MATTA, I., ALTAMAN, E. e AZOUZI, R. TCP Optimization through FEC, ARQ and Transmission Power Tradeoffs. Technical report, Springs Labs and Motorola Labs and NSF, 2003.
- BOITHIAS, L. *Radio Waves Propagations*. McGraw-Hill, 1987.
- CASETTI, C., GERLA, M., MASCOLO, S., SANADIDI, M. e WANG, R. TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks. *Kluwer Academic Publishers*, 2002.
- CCITT. Terms and definitions. *fasc.I.3, rec.M.60 n.34, rec.Q.9*, 1(0222).
- CHAMBERLAIN, M. e FURMAN, W. HF Data Link Protocol Enhancements based on STANAG 4538 and STANAG 4539, Providing Greater than 10kb/s Throughput over 3 khz channels. *IEE Conf. Pub.*, IT-13 (493):64–68, April 2003.
- CHANG, J., D., H. e LIN, M. Some Extended Results on the Search for Good Convolutional Codes. *IEEE Trans. on Information Theory*, 43(5), Setember 1997.

- COMER, D. E. *Internetworking with TCP/IP volume I*. Prentice-Hall., 1995.
- DIETRICH, P., M. e RAO, R. Second Order Bounds on Block Error Probabilities in Stationary, Time-Varying Channels. *CISS*, March 1996.
- ET AL., Z. L. Evaluation of TCP Vegas: Emulation and Experiment. *ACM SIGCOMM'95*, págs. 24–35, August 1995.
- FEI, P., SHIDUAN, C. e JIAN, M. An Effective Way to Improve TCP Performance in Wireless/Mobile Networks. *Proceedings of the IEEE/AFCEA EUROCOMM 2000, Information Systems for Enhanced Public Safety and Security, 2000*, 2000.
- FRITCHMAN, B. A Binary Channel Characterization using partitioned Markov Chains. *IEEE Trans. on Information Theory*, IT-13, April 1967.
- GOEL, S. e SANGHI, D. Improving TCP Performance **over wireless links**. *Proceedings of the TENCON '98, 1998 IEEE Region 10 International Conference on Global Connectivity in Energy, Computer, Communication and Control*, 2, December 1998.
- JACOBSON, V. Congestion avoidance and control. *ACM SIGCOMM '88*, August 1988.
- JODALEN, V., BJÖRN, S. e GRONNERUD, O. Military Messaging in IP Networks Using HF Links. *IEEE Personal Communications Magazine*, págs. 98–104, November 2004.
- JOHNSON, E. Interoperability and Performance Issues in HF E-mail. *IEEE*, 2001.
- JOHNSON, E. ARQ Protocol Features For HF Radio Links. *IEEE*, 2002.
- KALLGREN, D. e SMAAL, J. IP unicast/multicast operation over STANAG 5066. *Military Communications Conference*, 1(28-31):501–505, October 2001.
- MATHIS, M. e MAHDAVI, J. Forward Acknowledgment: Refining TCP Congestion Control. *ACM SIGCOMM*, October 1996a.
- MATHIS, M., MAHDAVI, J., FLOYD, S. e ROMANOW, A. Selective Acknowledgement Options. Technical Report RFC 2018, Internet Engineering

- Task Force and Internet Engineering Steering Group, October 1996b. URL <http://www.rfc-editor.org>.
- MIL-STD-188-110B. Interoperability and Performance Standards for Data Modems. Technical report, Department of Defense U.S., April 2000.
- POSTEL, J. User Datagram Protocol. Technical Report RFC 768, Internet Engineering Task Force and Internet Engineering Steering Group, August 1980. URL <http://www.rfc-editor.org>.
- POSTEL, J. Internet Protocol. Technical Report RFC 791, Internet Engineering Task Force and Internet Engineering Steering Group, September 1981a. URL <http://www.rfc-editor.org>.
- POSTEL, J. Transmission Control Protocol. Technical Report RFC 793, Internet Engineering Task Force and Internet Engineering Steering Group, September 1981b. URL <http://www.rfc-editor.org>.
- RAMAKRISHNAN, K., FLOYD, S. e BLACK, D. The Addition of Explicit Congestion Notification (ECN) to IP. *Internet draft draft-ietf-tsvwg-ecn-00.txt*, November 2000.
- SAMARAWEERA, N. Non-congestion Packet Loss Detection for TCP Error Recovery using Wireless Links. *IEEE Proceedings - Communications*, 146 (4), August 1999.
- TANEMBAUN, A. *Computer Networks*. Prentice-Hall., 1996.
- TSAI, S. Markov Characterization of the HF Channel. *IEEE Trans. on Communication Technology*, COM-17(1):24–32, February 1969.
- VITERBI, A. J. Convolutional Codes and their Performance in Communication Systems. *IEEE Trans. on Communication Technology*, COM-19: 751–772, October 1971.
- WATTERSON, C. C.AND JURUSHEK, J. Experimental Confirmation of HF Channel. *IEEE Trans. on Communication Technology*, COM-18(6): 792–803, December 1970.
- WICKER, S. *Error Control Systems for Digital Communication and Storage*. Prentice-Hall., 1995.

- ZORZI, M. Packet Dropping Statistics of a Data-Link Protocol for Wireless Local Communications. *IEEE Trans. on Vehicular Technology*, 52(1), January 2003.
- ZORZI, M. e RAO, R. Perspectives on the Impact of Error Statics on Protocols for Wireless Networks. *Personal Communications Magazine*, October 1999.
- ZORZI, M. E. A. Throughput Analysis of TCP on Channels with Memory. *IEEE Journal on Selected Areas in Communications*, 18(7):1289–1300, July 2000.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)