

UNIVERSIDADE FEDERAL FLUMINENSE

CESAR HENRIQUE PEREIRA RIBEIRO

Adaptação do Mecanismo de Controle de Congestionamento TFRC do
Protocolo de Transporte DCCP para Redes em Malha sem Fio

NITERÓI

2009

Livros Grátis

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

Milhares de livros grátis para download.

Cesar Henrique Pereira Ribeiro

Adaptação do Mecanismo de Controle de Congestionamento TFRC do
Protocolo de Transporte DCCP para Redes em Malha sem Fio

Dissertação de Mestrado
submetida ao Programa de
Mestrado em Engenharia de
Telecomunicações da
Universidade Federal Fluminense
como requisito parcial para
obtenção do título de Mestre.
Área de concentração: Sistemas
de Telecomunicações.

Orientadora: Profa. Débora Christina Muchalut Saade

Niterói

2009

Cesar Henrique Pereira Ribeiro

Adaptação do Mecanismo de Controle de Congestionamento TFRC do
Protocolo de Transporte DCCP para Redes em Malha sem Fio

Dissertação de Mestrado
submetida ao Programa de
Mestrado em Engenharia de
Telecomunicações da
Universidade Federal Fluminense
como requisito parcial para
obtenção do título de Mestre.
Área de concentração: Sistemas
de Telecomunicações.

Aprovada em 2009:

Profa. Dra. Débora Christina Muchaluat Saade
Universidade Federal Fluminense
(Orientadora)

Prof. Dr. Miguel Elias Mitre Campista
Universidade Federal Fluminense

Prof. Dr. Marcelo Gonçalves Rubinstein
Universidade do Estado do Rio de Janeiro

Niterói, dezembro de 2009.

Dedicatória

*À minha mãe, **Abigail**, por sempre acreditar em mim e
por ter sempre lutado pelas realizações
e felicidade de seus filhos.*

*Ao meu pai, **Geraldo**, minha irmã **Cristina**, meu irmão **Henrique**,
minhas tias **Eliane** e **Edna**, e minhas avós **Rita** e **Ruth**
por sua preocupação, carinho e incentivo.*

*À minha amada esposa **Ana Paula**,
por todo amor, incentivo, apoio, compreensão e
por sempre estar ao meu lado nos momentos mais difíceis.*

Agradecimentos

A Deus, pela dádiva da vida e por me permitir tantas realizações nesta existência. Obrigado por me permitir errar, aprender e crescer, por Sua eterna compreensão e tolerância, por Seu infinito amor e principalmente por ter me cercado de pessoas tão especiais, enfim, obrigado por tudo.

À Prof^a Débora Christina Muchaluat Saade, pela confiança que teve em mim. Sou grato pela orientação, competência, profissionalismo e dedicação. Obrigado por acreditar em mim e por me guiar neste que foi um dos maiores desafios da minha vida.

À minha amada esposa Ana Paula, por todo amor, carinho, compreensão e apoio em tantos momentos difíceis desta caminhada. Obrigado por permanecer ao meu lado, mesmo sem os carinhos rotineiros, sem a atenção devida e depois de tantos momentos de lazer perdidos. Obrigado pelo presente de cada dia, pelo seu sorriso e por me fazer feliz.

À minha família deixo um agradecimento especial, por todas as lições de amor, companheirismo, amizade, caridade, dedicação e caráter. Sinto-me orgulhoso e privilegiado por ter uma família tão especial.

Por fim, a todos aqueles que contribuíram, direta ou indiretamente, para a realização desta dissertação, o meu sincero agradecimento.

As ondas atendem ao meu mandar:

“Sossegai !”

(Cantor Cristão - Hino 328)

Resumo

Este trabalho propõe uma adaptação do mecanismo de controle de congestionamento *TCP-Friendly Rate Control* (TFRC) utilizado pelo protocolo de transporte *Datagram Congestion Control Protocol* (DCCP). O DCCP é um protocolo não confiável cuja proposta é substituir o uso do protocolo de transporte UDP principalmente por aplicações multimídia, que são sensíveis ao atraso e overhead provocado pelo protocolo de transporte TCP. O DCCP implementa controle de congestionamento sem o aumento do atraso, de forma a evitar que aplicações com grande demanda de largura de banda consumam todos os recursos de uma rede, causando degradação do serviço a outros fluxos concorrentes. Entretanto, o mecanismo de controle de congestionamento TFRC foi projetado para redes cabeadas e não possui um desempenho adequado para redes sem fio com múltiplos saltos (redes em malha sem fio ou redes mesh).

Especificamente, os mecanismos de acesso ao meio da camada MAC do padrão IEEE 802.11, como retransmissão e backoff exponencial, atrapalham o mecanismo de controle de congestionamento TFRC, resultando no cálculo de uma taxa de transmissão inadequada. Esta dissertação demonstra como o mecanismo TFRC padrão utilizado pelo DCCP sobrecarrega a camada MAC em redes sem fio com múltiplos saltos e apresenta uma nova proposta, chamada M-TFRC, cujos resultados de simulação indicam melhoria do desempenho do protocolo.

Palavras-chave: DCCP, redes mesh sem fio, TFRC, controle de congestionamento

Abstract

This work proposes an adaptation of the TCP-Friendly Rate Control (TFRC) congestion control mechanism used by the Datagram Congestion Control Protocol (DCCP) transport protocol. DCCP is an unreliable transport protocol proposed to replace the use of the UDP transport protocol, mainly for multimedia applications, which are affected by the delay and overhead imposed by the TCP transport protocol. DCCP implements congestion control without increasing delay, avoiding applications with high bandwidth demand from consuming all network resources, which causes degradation in concurrent flows. However, as the TFRC congestion control mechanism was designed for wired networks, it does not perform well in multihop wireless networks (mesh networks).

Specifically, IEEE 802.11 MAC layer medium access mechanisms, such as retransmission and exponential backoff, mislead the TFRC congestion control mechanism, resulting in an inaccurate sending rate adjustment. This dissertation shows how the default DCCP TFRC mechanism overloads the MAC layer in multihop wireless mesh networks and proposes a new mechanism, called M-TFRC, whose simulation results in significant performance improvements.

Keywords: DCCP, wireless mesh networks, TFRC, congestion control.

Índice

1	Introdução.....	1
1.1	Estrutura da Dissertação	2
2	Protocolo de Transporte DCCP	4
2.1	Estrutura do Segmento DCCP	5
2.2	Estabelecimento de Sessão e Negociação de Parâmetros	7
2.3	Comparação do DCCP com o TCP e UDP	11
3	Controle de Congestionamento	13
3.1	ECN - Explicit Congestion Notification	14
3.2	Controle de Congestionamento do Protocolo de Transporte TCP	15
3.3	Controle de Congestionamento DCCP CCID2 – TCP LIKE Congestion Control... ..	18
3.4	Controle de Congestionamento DCCP CCID3 – TFRC Congestion Control	21
4	Redes Mesh sem Fio com Múltiplos Saltos	25
4.1	Redes IEEE 802.11	25
4.2	Redes Mesh Sem Fio Com Múltiplos Saltos	30
4.3	Principais Desafios do Controle de Congestionamento TFRC em Redes Mesh	32
4.4	Trabalhos Relacionados	33
5	Adaptação do Mecanismo de Controle de Congestionamento TFRC do Protocolo de Transporte DCCP para Redes Mesh	36
5.1	Estudo do Comportamento do Protocolo DCCP Padrão	36
5.2	Adaptação do Controle do congestionamento CCID 3.....	43
5.3	Testes com Fluxos Concorrentes	49
5.4	Testes com Topologia Mesh com Múltiplos Saltos Densa.....	58
5.5	Conclusão do Capítulo.....	60

6 Conclusões.....	61
Referências Bibliográficas	62
Anexo 01 – Testes Variando os Parâmetros K1 e K2.....	65

Lista de Figuras

Figura 2.1. Cabeçalho genérico DCCP (Estendido).....	5
Figura 2.2. Cabeçalho genérico DCCP (Simplificado).....	5
Figura 2.3. Ciclo de Vida Conexão DCCP.....	8
Figura 2.4. Sessão DCCP.....	9
Figura 3.1. Exemplos de Intervalos de Perdas.....	22
Figura 4.1. IEEE 802.11 DCF.....	27
Figura 4.2. Problema do Terminal Escondido.....	28
Figura 4.3. Problema do Terminal Exposto.....	28
Figura 4.4. Mecanismo RTS/CTS do padrão IEEE 802.11.....	29
Figura 4.5. Rede Mesh.....	31
Figura 5.1. Topologia usada nas Simulações.....	37
Figura 5.2. Comparação entre a Vazão Média DCCP TFRC Padrão Medida e a Taxa de Envio de Dados da Aplicação para topologia com 6 nós.....	39
Figura 5.3. Atraso Médio DCCP TFRC Padrão para topologia com 6 nós.....	39
Figura 5.4. Variação do Atraso Médio DCCP TFRC Padrão para topologia com 6 nós.....	40
Figura 5.5. Comparação entre a Vazão Média DCCP TFRC Padrão Medida e a Taxa de Envio de Dados da Aplicação para topologia com 7 nós.....	42
Figura 5.6. Atraso Médio DCCP TFRC Padrão para topologia com 7 nós.....	42
Figura 5.7. Variação do Atraso Médio DCCP TFRC Padrão para topologia com 7 nós.....	43
Figura 5.8. Comparação Vazão Média DCCP M-TFRC e DCCP TFRC Padrão.....	48
Figura 5.9. Comparação Atraso Médio DCCP M-TFRC e DCCP TFRC Padrão.....	49
Figura 5.10. Vazão Instantânea Fluxos DCCP TFRC Padrão Concorrentes.....	50
Figura 5.11. Vazão Instantânea Fluxos DCCP M-TFRC Concorrentes.....	50
Figura 5.12. Vazão Média Fluxos DCCP M-TFRC Concorrentes e fluxos TFRC Padrão Concorrentes.....	51
Figura 5.13. Atraso Médio Fluxos DCCP M-TFRC Concorrentes e fluxos TFRC Padrão Concorrentes.....	51
Figura 5.14. Vazão Instantânea Fluxos TCP e DCCP TFRC Padrão Concorrentes (1 Salto).....	52
Figura 5.15. Vazão Instantânea Fluxos TCP e DCCP M-TFRC Concorrentes (1 Salto).....	53

Figura 5.16. Vazão Média Fluxos DCCP M-TFRC e TCP Concorrentes e Fluxos DCCP TFRC Padrão e TCP Concorrentes (1 Salto).....	53
Figura 5.17. Atraso Médio Fluxos DCCP M-TFRC e TCP Concorrentes e Fluxos DCCP TFRC Padrão e TCP Concorrentes (1 Salto).....	54
Figura 5.18. Vazão Instantânea Fluxos TCP e DCCP TFRC Padrão Concorrentes (2 Saltos).....	54
Figura 5.19. Vazão Instantânea Fluxos TCP e DCCP M-TFRC Concorrentes (2 Saltos).....	55
Figura 5.20. Vazão Média Fluxos DCCP M-TFRC e TCP Concorrentes e Fluxos DCCP TFRC Padrão e TCP Concorrentes (2 Saltos).....	55
Figura 5.21. Atraso Médio Fluxos DCCP M-TFRC e TCP Concorrentes e Fluxos DCCP TFRC Padrão e TCP Concorrentes (2 Saltos).....	56
Figura 5.22. Vazão Instantânea Fluxos TCP e DCCP TFRC Padrão Concorrentes (3 Saltos).....	56
Figura 5.23. Vazão Instantânea Fluxos TCP e DCCP M-TFRC Concorrentes (3 Saltos).....	57
Figura 5.24. Vazão Média Fluxos DCCP M-TFRC e TCP Concorrentes e Fluxos DCCP TFRC Padrão e TCP Concorrentes (3 Saltos).....	57
Figura 5.25. Atraso Médio Fluxos DCCP M-TFRC e TCP Concorrentes e Fluxos DCCP TFRC Padrão e TCP Concorrentes (3 Saltos).....	58
Figura 5.26. Topologia Mesh com Múltiplos Saltos Densa.....	59
Figura 5.27. Comparação de Vazão Média DCCP M-TFRC e DCCP TFRC.....	59
Figura 5.28. Comparação de Atraso Médio DCCP M-TFRC e DCCP TFRC.....	60

Lista de Tabelas

Tabela 2.1. Campos do cabeçalho genérico DCCP.....	6
Tabela 2.2. Tipos de Segmentos DCCP.....	7
Tabela 2.3. Regras de Reconciliação de Parâmetros.....	11
Tabela 2.4. Características dos Protocolos UDP, TCP e DCCP.....	12
Tabela 5.1. Resultados Obtidos Topologia com 6 Nós.....	38
Tabela 5.2. Resultados Obtidos Topologia com 7 Nós.....	41
Tabela 5.3. M-TFRC Resultados Obtidos Topologia com 6 Nós.....	46
Tabela 5.4. M-TFRC Resultados Obtidos Topologia com 7 Nós.....	47
Tabela 5.5. M-TFRC Resultados Obtidos Topologia com N Saltos.....	48

Lista de Abreviaturas

AODV	<i>Ad hoc On-demand Distance Vector</i>
CBR	<i>Constant Bit Rate</i>
CCID	<i>Congestion Control Identifier</i>
CE	<i>Congestion Experienced</i>
CSMA/CA	<i>Carrier sense multiple access with collision avoidance</i>
CTS	<i>Clear to Send</i>
DCCP	<i>Datagram Congestion Control Protocol</i>
DCF	<i>Distributed Coordination Function</i>
DIFS	<i>DCF Interframe Space</i>
ECN	<i>Explicit Congestion Notification</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IETF	<i>Internet Engineering Task Force</i>
IP	<i>Internet Protocol</i>
IPTV	<i>Internet Protocol Television</i>
LAN	<i>Local Area Network</i>
MAC	<i>Medium Access Control</i>
MANET	<i>Mobile Ad-hoc Networks</i>
MSS	<i>Maximum Segment Size</i>
M-TFRC	<i>Mesh-TFRC</i>
NAT	<i>Network Address Translation</i>
NAV	<i>Network Allocation Vector</i>
NN	<i>Non-Negotiable</i>
OSI	<i>Open Systems Interconnection</i>
RFC	<i>Request for Comments</i>
RTS	<i>Request to Send</i>
SACK	<i>Selective Acknowledgment</i>
SIFS	<i>Short Interframe Space</i>
SP	<i>Server-Priority</i>
TCP	<i>Transmission Control Protocol</i>
TFRC	<i>TCP Friendly Rate Control</i>
UDP	<i>User Datagram Protocol</i>

VoIP

Voice over IP

1 Introdução

Congestionamento em uma rede comutada por pacotes é a situação na qual existe uma quantidade de pacotes a serem transmitidos cujo número é maior do que a rede é capaz de suportar. O congestionamento ocasiona um aumento no retardo e nas perdas de pacotes. Se nenhum mecanismo de controle de congestionamento for adotado, a rede pode entrar em colapso.

Comunicações sensíveis a atraso são transportadas de forma muito mais eficiente sobre protocolos de transporte não confiáveis, pois o atraso gerado pelos protocolos confiáveis para detectar perdas e retransmitir segmentos faz com que a comunicação tenha um desempenho inferior do que se apenas ignorasse os segmentos perdidos. Segmentos com atraso superior a um determinado limite não possuem utilidade nenhuma para a aplicação receptora.

Com o rápido crescimento de aplicações como videoconferência e IPTV, que têm como característica o consumo contínuo e prolongado de grande largura de banda, o uso do UDP (*User Datagram Protocol*) como protocolo de transporte leva ao risco de colapso nas redes, pois diferente do TCP (*Transmission Control Protocol*), o UDP não possui nenhum tipo de controle de congestionamento.

O controle de congestionamento pode ser feito na camada de aplicação, acima do UDP, mas sua implementação é complexa e o fato de cada programador implementar o seu tipo de controle de congestionamento pode gerar comportamento instável na rede.

Redes em malha sem fio são redes cada vez mais populares em um mundo onde a necessidade de estar sempre conectado à Internet é cada vez maior. Juntando a crescente demanda por acesso à diminuição do custo de equipamentos como antenas e roteadores sem fio, assim como laptops e aparelhos portáteis com acesso *wi-fi* temos um cenário ideal para a proliferação de redes em malha sem fio de múltiplos saltos.

O controle de congestionamento é ainda mais crítico em redes sem fio com múltiplos saltos (redes mesh), pois os recursos são mais escassos e a comunicação mais sujeita a erros do que em redes cabeadas.

O protocolo de transporte DCCP (*Datagram Congestion Control Protocol*) foi proposto para a substituição do protocolo UDP para aplicações multimídia, pois implementa controle de congestionamento sem ser confiável, logo não há o aumento do atraso gerado pelas retransmissões de segmentos perdidos.

O problema é que o controle de congestionamento do DCCP, assim como o do TCP [Fu et al, 2003], [Balakrishnan et al, 1996], [Gerla et al, 1999a] e [Gerla et al, 1999b], não atua corretamente em redes sem fio com múltiplos saltos por causa dos mecanismos de acesso ao meio da camada MAC do padrão IEEE 802.11.

A principal motivação deste trabalho diz respeito à necessidade de adaptação do controle de congestionamento do protocolo de transporte DCCP para o uso em redes sem fio com múltiplos saltos.

Um dos mecanismos de congestionamento propostos para o protocolo de transporte DCCP é chamado TFRC – *TCP Friendly Rate Control* – (CCID 3). O TFRC é um mecanismo de controle de congestionamento baseado em taxa, onde o transmissor ajusta sua taxa de transmissão baseado em informações recebidas do receptor. Em redes sem fio, além do congestionamento, há também o problema da saturação da camada MAC causado pela sobrecarga de transmissões e os mecanismos de contenção do padrão IEEE 802.11. O mecanismo TFRC ignora os efeitos da saturação e aumenta a taxa máxima de transmissão acima do que a rede é capaz de suportar, sobrecarregando-a e, conseqüentemente, degradando o desempenho da rede.

O objetivo principal desta dissertação é a proposta de uma adaptação ao mecanismo de controle de congestionamento TFRC, que permita a detecção do início da sobrecarga da camada MAC do padrão IEEE 802.11 e reduza a vazão máxima do fluxo DCCP de maneira a evitar a degradação da rede em malha sem fio. Esta proposta é chamada de M-TFRC (Mesh-TFRC) e foi avaliada através de simulações utilizando o software NS-2, considerando diferentes cenários de redes em malha sem fio.

1.1 Estrutura da Dissertação

O Capítulo 2 apresenta o protocolo de transporte DCCP, informando suas principais características, tais como ser um protocolo orientado à conexão não confiável, a estrutura de seu cabeçalho e seu mecanismo de negociação de parâmetros. Também é apresentada uma comparação entre os protocolos de transporte DCCP, UDP e TCP.

O Capítulo 3 fundamenta os conhecimentos relativos ao controle de congestionamento e apresenta os mecanismos de controle de congestionamento existentes na implementação atual do protocolo de transporte DCCP.

O Capítulo 4 apresenta o padrão IEEE 802.11, as redes mesh com múltiplos saltos, os principais desafios do controle de congestionamento TFRC em redes em malha sem fio e os trabalhos relacionados a esta dissertação.

O Capítulo 5 mostra o estudo do desempenho do protocolo DCCP padrão em redes em malha sem fio, usado como base para o trabalho realizado nesta dissertação. O capítulo também apresenta a nova proposta de controle de congestionamento M-TFRC do protocolo de transporte DCCP e a avaliação de seu desempenho através de resultados obtidos de simulações de cenários de redes em malha sem fio que comparam a proposta com a implementação padrão do mecanismo TFRC.

O Capítulo 6 trata das considerações finais, enumerando as principais contribuições da dissertação e apontando trabalhos futuros.

2 Protocolo de Transporte DCCP

O *Datagram Congestion Control Protocol* (DCCP) é um protocolo da camada de transporte projetado para prover controle de congestionamento em conexões não confiáveis, proposto para substituição ao UDP, devido ao fato do UDP não competir de forma justa ao compartilhar largura de banda com outros fluxos concorrentes .

O DCCP, definido na RFC 4340 [Kohler et al, 2006], foi projetado para ser tão simples quanto possível, acrescentando apenas o mínimo necessário para o controle de congestionamento em tráfegos não confiáveis. Suas principais características serão apresentadas ao longo deste capítulo.

O DCCP possui um mecanismo que permite que o cliente e o servidor negociem os parâmetros da sessão, entre eles o mecanismo de controle de congestionamento utilizado. Ele também possui baixo *overhead*, o que é fundamental para aplicações como voz sobre IP (VoIP) e jogos online, pois diminui a latência e o tempo de resposta. O DCCP possui suporte ao *Explicit Congestion Control* (ECN). O ECN permite que os roteadores notifiquem aos receptores que existe congestionamento no caminho percorrido pelo segmento.

Outra característica importante do DCCP é ser um protocolo não confiável orientado à conexão e que provê informação sobre sequência dos pacotes transmitidos. Para implementar o controle de congestionamento, é necessária a utilização de números de sequência nos datagramas DCCP para detectar as perdas ocorridas na rede. Também é necessário usar um canal de retorno (*feedback*), para fornecer informação sobre o congestionamento da rede para o transmissor e também para o suporte ao ECN.

Todas as funcionalidades que poderiam ser realizadas por protocolos das camadas superiores sem comprometimento de sua eficiência não foram implementadas pelo DCCP, para tornar o protocolo o mais simples possível. Além das funcionalidades básicas como multiplexação/demultiplexação, detecção de erro e controle de congestionamento, o DCCP suporta mobilidade e segurança [Kohler et al, 2006].

2.1 Estrutura do Segmento DCCP

O tamanho do cabeçalho DCCP pode variar de 12 a 1020 bytes. Os 12 primeiros bytes do cabeçalho possuem a mesma semântica para todos os tipos de segmentos. São seguidos por campos de tamanho fixo definidos pelo tipo do segmento, e por campos de tamanhos variáveis dependendo do uso da lista de opções. Ao fim do cabeçalho encontra-se a área de dados de aplicação. Alguns tipos de segmentos DCCP não carregam dados de aplicações.

A forma do cabeçalho DCCP depende do valor do campo *X*, denominado *Extended Sequence Numbers* bit. Se $X = 1$, o campo *Sequence Number* possui 48 bits e o cabeçalho DCCP genérico possui 16 bytes. Se $X = 0$, o campo *Sequence Number* possui 24 bits e o cabeçalho DCCP genérico possui 12 bytes. As Figuras 2.1 e 2.2, retiradas da RFC 4340 [Kohler et al, 2006], ilustram o cabeçalho genérico DCCP para $X = 1$ e $X = 0$, respectivamente, e seus campos são definidos na Tabela 2.1.

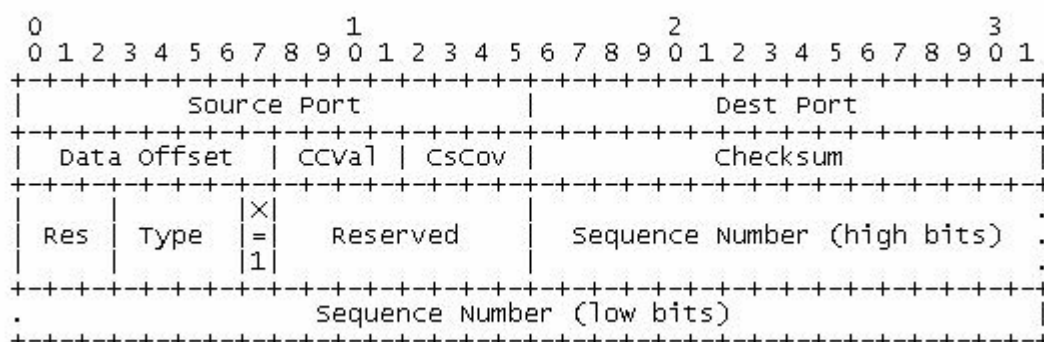


Figura 2.1. Cabeçalho genérico DCCP (Estendido)

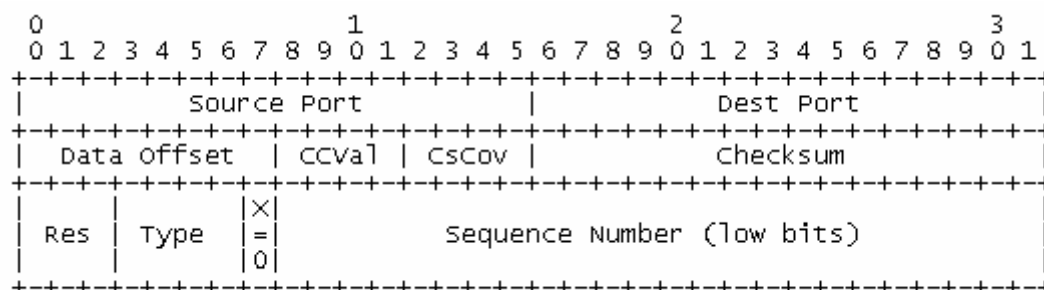


Figura 2.2. Cabeçalho genérico DCCP (Simplificado)

Tabela 2.1. Campos do cabeçalho genérico DCCP

Campo	nº bits	Descrição
<i>Source Port</i>	16	Identifica a porta de origem da conexão de forma similar ao campo correspondente no TCP e UDP.
<i>Destination Port</i>	16	Identifica a porta de destino da conexão de forma similar ao campo correspondente no TCP e UDP.
<i>Data Offset</i>	8	Determina o tamanho do cabeçalho DCCP, indicando o deslocamento, em múltiplos de 32 bits, a partir do início do segmento DCCP em que se encontra o início dos dados de aplicação.
<i>CCVal</i>	4	Indica o tamanho da janela que será usada pelo receptor para determinar quando múltiplas perdas pertencem a um único evento de perdas.
<i>Checksum Coverage (CsCov)</i>	4	Determina que partes do segmentos são cobertas pelo campo <i>checksum</i> . Ele sempre inclui o cabeçalho e as opções, mas alguns ou todos os dados de aplicação podem ser excluídos do <i>checksum</i> . Isto aumenta o desempenho de enlaces com ruídos para aplicações com tolerância a dados corrompidos
<i>Checksum</i>	16	Informa o <i>checksum</i> do segmento DCCP.
<i>Reserved (Res)</i>	3	Campo reservado.
<i>Type</i>	4	Este campo especifica o tipo do segmento DCCP, listados na Tabela 2.2.
<i>Extended Sequence Numbers (X)</i>	1	Se $X = 0$, o tamanho dos campos <i>Sequence</i> e <i>Acknowledgement</i> é 24 bits. Se $X = 1$, o tamanho é 48 bits. Os datagramas DCCP-Data, DCCP-DataAck e DCCP-Ack podem ter o campo $X = 0$ ou $X = 1$. Os datagramas DCCP-Request, DCCP-Response, DCCP-CloseReq, DCCP-Close, DCCP-Reset, DCCP-Sync e DCCP-SyncAck sempre têm $X = 1$.
<i>Sequence Number</i>	48 ou 24	Identifica unicamente um segmento DCCP. O número de sequência incrementa para todo o segmento enviado, incluindo <i>acks</i> sem nenhum dado de aplicação.
<i>Reserved</i>	16 ou 8	Campo reservado
<i>Acknowledgement Number</i>	48 ou 24	Usado pelo receptor para informar ao transmissor o número de sequência do segmento recebido.

O cabeçalho do protocolo DCCP apresenta um campo denominado tipo do pacote. Este campo determina que informação está contida em um determinado pacote DCCP. Na Tabela 2.2 são apresentados os possíveis valores desse campo, nome do pacote e descrição.

Tabela 2.2. Tipos de Segmentos DCCP

Tipo	Função	Descrição
0	DCCP-Request	Enviado pelo cliente para iniciar a conexão (a primeira parte do <i>three-way handshake</i>).
1	DCCP-Response	Enviado pelo servidor em resposta a um DCCP-Request (a segunda parte do <i>three-way handshake</i>).
2	DCCP-Data	Usado para transmitir dados de aplicação
3	DCCP-Ack	Usado para transmitir somente <i>acks</i> .
4	DCCP-DataAck	Usado para transmitir dados de aplicações e <i>ack</i> no mesmo datagrama (<i>piggybacked acknowledgement</i>).
5	DCCP-CloseReq	Enviado pelo servidor para solicitar que o cliente encerre a conexão.
6	DCCP-Close	Usado pelo cliente ou servidor para encerrar a conexão. É esperado um segmento DCCP-Reset após o seu envio.
7	DCCP-Reset	Usado para terminar a conexão.
8	DCCP-Sync	Usados para sincronizar os números de sequência após perdas em rajadas.
9	DCCP-SyncAck	
10-15	Reserved	Reservado

2.2 Estabelecimento de Sessão e Negociação de Parâmetros

O DCCP é um protocolo *unicast* orientado à conexão, isto é, para haver comunicação de dados entre um sistema final A e um sistema final B, é necessário que haja um procedimento explícito de início de sessão entre eles.

O estabelecimento de uma conexão DCCP é ilustrado na Figura 2.3, adaptada da RFC 4340 [Kohler et al, 2006]. O processo inicia quando o cliente envia para o servidor um pacote

do tipo DCCP-Request e o servidor responde com um pacote DCCP-Response. Ao receber o DCCP-Response, o cliente envia para o servidor um pacote que confirma o recebimento do DCCP-Response e a partir desse momento a conexão é efetivamente estabelecida. Após o estabelecimento da conexão, os dois sistemas trocam dados entre si através dos pacotes DCCP-Data ou DCCP-DataAck enquanto ocorrer a conexão. A conexão é finalizada quando o servidor envia um pacote do tipo DCCP-Closereq ou o cliente envia um pacote DCCP-Close. Ao receber do cliente a confirmação de recebimento do DCCP-Closereq transmitido, o servidor envia para o cliente um pacote do tipo DCCP-Reset para informar que a conexão está finalizada. Nesse ponto, o cliente permanece no estado de *timewait*, que serve para receber eventuais pacotes da conexão que ainda estejam em trânsito na rede.

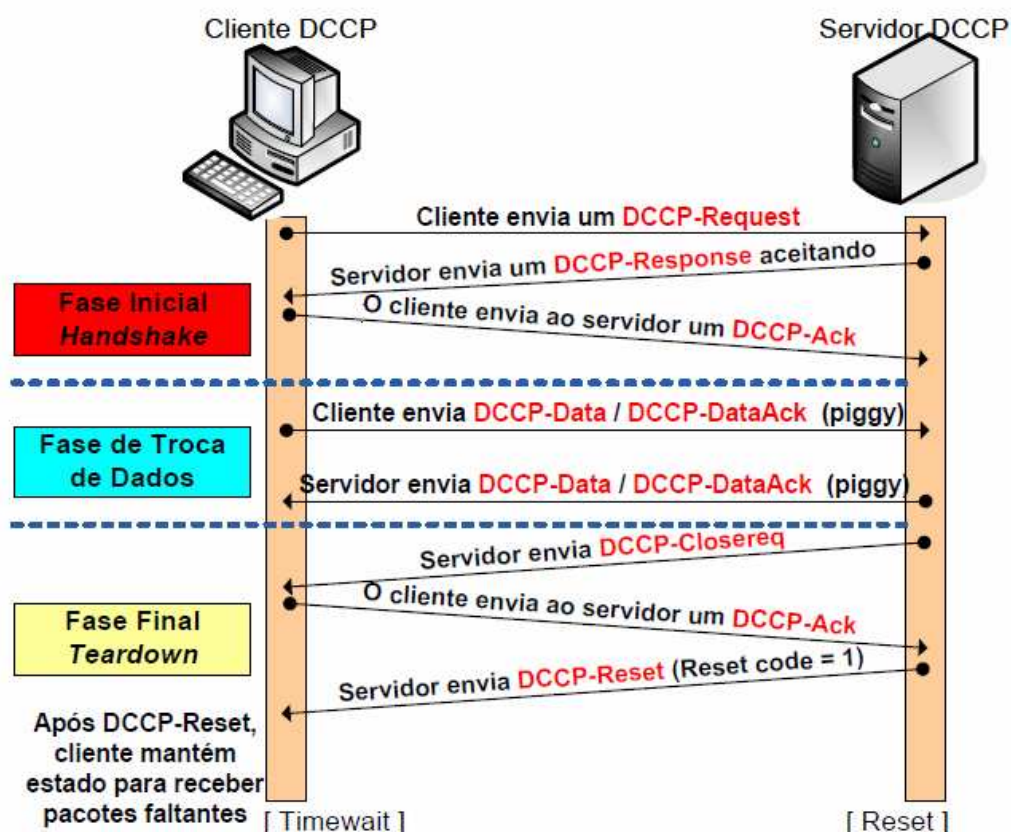


Figura 2.3. Ciclo de Vida Conexão DCCP

As conexões DCCP são bidirecionais, ou seja, dados podem ser originados por qualquer um dos sistemas finais. Dados e reconhecimentos (*acks*) podem ser transmitidos em ambos os sentidos simultaneamente. Dessa forma, a conexão DCCP comporta-se como duas conexões unidirecionais separadas, denominadas *half-connections*. Cada *half-connection*

consiste dos dados de aplicação enviados por um sistema final e os *acks* correspondentes enviados pelo outro sistema final.

Em uma conexão entre A e B, duas *half-connections* são criadas: uma para enviar dados de A para B e *acks* de B para A e outra no sentido inverso. A negociação dos parâmetros das duas *half-connections* são completamente independentes e podem ocorrer simultaneamente. Por exemplo, as duas *half-connections* podem usar mecanismos de controle de congestionamento diferentes. Os parâmetros também podem ser renegociados e alterados durante a sessão.

A Figura 2.4, adaptada da RFC 4340 [Kohler et al, 2006], ilustra uma comunicação bidirecional entre os sistemas finais A e B. A *half-connection* de A para B é composta de um canal de envio de dados de A para B, indicado pelo número 1 e um canal de confirmação (*feedback*) indicado pelo número 2. Da mesma forma, os números 3 e 4 indicam o canal de dados e confirmação, respectivamente, da *half-connection* de B para A.

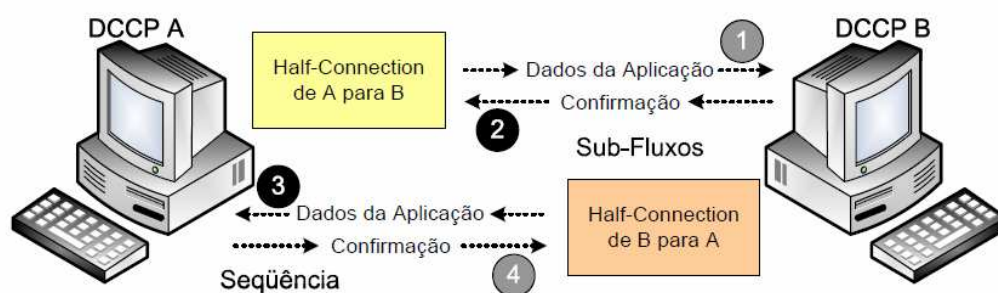


Figura 2.4. Sessão DCCP

Embora sejam logicamente distintas, na prática as conexões se sobrepõem; um segmento DCCP-DataAck pode conter dados de aplicação de uma *half-connection* e *acks* de outra *half-connection*. Ainda no contexto das *half-connections*, o protocolo DCCP não permite apenas uma delas, ou seja, o protocolo ao finalizar uma conexão termina as duas *half-connections*, que portanto, são tratadas como uma única entidade.

No início da comunicação DCCP, os sistemas finais precisam negociar os parâmetros que serão utilizados durante a sessão DCCP, entre eles o mecanismo de controle de congestionamento que será utilizado. Esta negociação acontece através do uso de opções sinalizadas no cabeçalho de um pacote DCCP.

Quatro opções DCCP são usadas para negociação de parâmetros: *Change L*, *Confirm L*, *Change R* e *Confirm R*. Uma opção *Change* inicia a negociação e uma opção *Confirm* completa a negociação. O “L” significa parâmetro local e o “R” significa parâmetro remoto.

Todas as quatro opções possuem o mesmo formato. O primeiro byte é o número do parâmetro e os bytes subsequentes guardam um ou mais valores para o parâmetro indicado, de acordo com a ordem de preferência do transmissor para o valor do parâmetro.

Change L e *Change R* iniciam a negociação do parâmetro. A opção depende da localização do parâmetro a ser alterado. Na comunicação entre A e B, se A deseja alterar um parâmetro próprio, ele envia um datagrama *change L* para B. Se A deseja alterar um parâmetro de B, ele envia um datagrama *change R*. *Confirm L* e *Confirm R* completam a negociação do parâmetro e são enviados em resposta a um *Change R* e *Change L*, respectivamente.

Existem regras de conciliação que são usadas quando o cliente e o servidor discordam quanto ao valor de um determinado parâmetro. Existem duas regras de reconciliação: *server-priority* (SP) e *non-negotiable* (NN). Para cada parâmetro a regra de conciliação é definida de acordo com a Tabela 2.3.

Na regra *server-priority*, cada opção *Change* contém uma lista de valores ordenados de acordo com a preferência de quem está solicitando a alteração. Cada opção *Confirm* contém o valor confirmado, seguido por uma lista com os valores de preferência de quem está confirmando. É escolhido o primeiro valor na lista do servidor que também está contido na lista do cliente. Se não houver nenhum valor coincidente nas duas listas, o valor não é alterado e a opção *Confirm* indicará que o valor continua o mesmo.

Na regra *non-negotiable*, cada opção contém exatamente um valor. O sistema final onde o parâmetro reside informa o valor novo do parâmetro através de uma opção *Change L*. O sistema final remoto aceita qualquer valor válido, respondendo com um *Confirm R* confirmando o novo valor do parâmetro. Se o valor solicitado for inválido, o sistema final remoto envia um *Confirm R* com valor zerado, indicando que o valor não foi alterado

Tabela 2.3. Regras de Reconciliação de Parâmetros

Número	Significado	Regra de Conciliação	Valor Inicial
0	Reservado		
1	<i>Congestion Control ID (CCID)</i>	SP	2
2	<i>Allow short seqnos</i>	SP	0
3	<i>Sequence Window</i>	NN	100
4	<i>ECN Incapable</i>	SP	0
5	<i>Ack Ratio</i>	NN	2
6	<i>Send Ack Vector</i>	SP	0
7	<i>Send NDP Count</i>	SP	0
8	<i>Minimum Checksum Coverage</i>	SP	0
9	<i>Check Data Checksum</i>	SP	0
10-127	Reservado		
128-255	CCID – <i>Specific Features</i>		

2.3 Comparação do DCCP com o TCP e UDP

Na Tabela 2.4 são apresentadas de forma resumida as características do protocolo DCCP apresentadas neste capítulo. Também são apresentadas as suas semelhanças e diferenças em relação aos protocolos UDP [Postel, 1980] e TCP [RFC793, 1981], [Braden, 1989], [Jacobson et al, 1992], [Mathis et al, 1996], [Allman et al, 1999], [Allman et al, 2002].

Tabela 2.4. Características dos Protocolos UDP, TCP e DCCP

Característica	UDP	TCP	DCCP
Tamanho do cabeçalho Genérico	8 bytes	20 bytes	12 ou 16 bytes
Numeração de Porta	Sim	Sim	Sim
Detecção de Erro nos Dados	Opcional	Sim	Opcional
Garantia de Entrega de Dados	Não	Sim	Não
Número de Sequência	Não	Sim	Sim
Ordenação	Não	Sim	Não
Controle de Erros	Não	Sim	Não
Controle de Fluxo	Não	Sim	Sim
Controle de Congestionamento	Não	Sim	Sim
Suporte a ECN	Não	Sim	Sim

3 Controle de Congestionamento

No protocolo de transporte DCCP, a aplicação pode escolher o mecanismo de controle de congestionamento que será usado durante a conexão. A escolha é feita através do *Congestion Control ID* (CCID), que é um campo de 8 bits negociado de forma independente no estabelecimento de cada *half-connection* DCCP.

Os CCIDs (*Congestion Control Identifier*) são módulos independentes do restante do protocolo e responsáveis por realizar o controle de congestionamento durante o ciclo de vida de uma conexão DCCP. Eles descrevem como um sistema que utiliza DCCP limita a taxa de transmissão de pacotes na rede e os valores iniciais de parâmetros da conexão, por exemplo, o tamanho inicial da janela de transmissão (controle de congestionamento baseado em janela) ou como e com qual frequência o receptor envia informações de congestionamento para o transmissor (controle de congestionamento baseado em taxa).

Além de serem negociados no estabelecimento da conexão DCCP, os CCIDs podem ser alterados durante o ciclo de vida da conexão, sendo possível a execução de um CCID em um sentido e um outro CCID no sentido contrário. Esta flexibilidade na utilização dos algoritmos de controle de congestionamento em uma conexão DCCP é importante, uma vez que a característica de tráfego em um sentido de uma conexão pode ser totalmente diferente se comparada ao tráfego no sentido oposto.

A principal justificativa para prover uma forma modular para gerenciar os CCIDs é que um determinado algoritmo pode ser mais apropriado para um tipo de aplicação, sendo possível adicionar novos CCIDs ou removê-los de forma independente do núcleo do protocolo. Por exemplo, as aplicações de jogos na Internet podem fazer uso de qualquer largura de banda disponível na rede, pois muitas delas utilizam técnicas de diferença de quadros, onde são enviadas apenas as diferenças entre uma cena do jogo e a outra. Por outro lado, as aplicações de voz sobre IP transmitem continuamente pequenos pacotes de voz ou transmitem rajadas de pequenos pacotes em um curto espaço de tempo (quando um dos interlocutores fala), sendo as rajadas separadas por períodos de silêncio entre palavras, entre frases, ou quando um interlocutor para de falar para dar a vez ao outro.

Atualmente, o IETF provê dois CCIDs para mecanismos de controle de congestionamento no DCCP [Floyd e Kohler, 2006], [Floyd et al, 2006a]. São eles:

- CCID 2 – *TCP Like*: Mecanismo similar ao utilizado pelo TCP. É indicado para aplicações que geram tráfego em rajada, onde deve ser enviada a maior

quantidade de dados possível em um curto intervalo de tempo. É definido na RFC 4341 [Floyd e Kohler, 2006].

- CCID 3 – *TCP-Friendly Rate Control* (TRFC): Mecanismo baseado em taxa. É indicado para aplicações que geram tráfego continuamente. É definido na RFC 4342 [Floyd et al, 2006a].

A seguir será feita uma descrição sobre os principais mecanismos de controle de congestionamento do DCCP, começando por uma visão geral sobre *Explicit Congestion Notification* (ECN) e sobre o controle de congestionamento usado no TCP, necessários para o entendimento dos mecanismos propostos para o DCCP.

3.1 ECN - Explicit Congestion Notification

Existem diversos mecanismos utilizados para que um sistema final tenha conhecimento do estado da rede em termos de congestionamento. Alguns mecanismos estão implícitos através da própria transmissão, como o aumento no tempo de resposta e a não confirmação de recepção de pacotes, pelos quais pode-se inferir que o sistema receptor não está recebendo os dados transmitidos.

Porém, existe um mecanismo explícito utilizado para notificar que a rede está congestionada, também conhecido por ECN (*Explicit Congestion Notification*) [Ramakrishnan et al, 2001], [Spring et al, 2003]. O ECN considera o descarte de pacotes pelos roteadores, o que pode acontecer de forma aleatória e depende da capacidade de processamento do roteador. Em vez de fazer descarte de pacotes, o roteador utiliza determinados pacotes para marcá-los com uma sinalização de que a rede está congestionada, também conhecida por sinalização CE (*Congestion Experienced*) realizada através de marcações do campo ECN (os dois bits mais significativos do campo *Differentiated Services* do cabeçalho IP). O objetivo dessa sinalização é notificar o transmissor que a rede está congestionada (ou na iminência de congestionar) para que ele reduza sua taxa de transmissão.

Com o uso de ECN, é possível diminuir as perdas de pacotes quando o congestionamento é incipiente, reduzindo as retransmissões e o tráfego na rede. Como o ECN evita perdas desnecessárias de pacotes, as aplicações com pouca troca de dados ou que sejam sensíveis ao atraso podem se beneficiar com isso [Fonseca, 2004]. O mecanismo de ECN para IP está especificado na RFC 3168 [Ramakrishnan et al, 2001] e tanto o TCP quanto o DCCP

suportam esse mecanismo. O ECN é um dos pontos discutidos em [Floyd et al, 2006b], que apresenta diversas justificativas para criação do protocolo DCCP.

3.2 Controle de Congestionamento do Protocolo de Transporte TCP

O protocolo de transporte TCP está definido nas RFCs 793 [RFC793, 1981], 1122 [Braden, 1989], 1323 [Jacobson et al, 1992], 2018 [Mathis et al, 1996], 2581 [Allman et al, 1999] e 3390 [Allman et al, 2002] e oferece as funções de controle de erros, controle de fluxo e controle de congestionamento.

O controle de fluxo do TCP tem por objetivo eliminar a possibilidade do remetente saturar o *buffer* do receptor. O receptor TCP informa ao transmissor o tamanho de sua janela de recepção e o número do próximo byte que ele espera receber. Em uma conexão *full-duplex*, o remetente de cada sentido da conexão mantém uma janela de recepção distinta. A janela de recepção é dinâmica, isto é, muda durante o tempo de vida útil da conexão.

O receptor possui as seguintes variáveis para calcular e informar ao transmissor o tamanho de sua janela de recepção:

- *LastByteRead* = o número do último byte lido no buffer do TCP pelo processo de aplicação no receptor;
- *LastByteRcvd* = o número do último byte recebido do transmissor pela janela de recepção no receptor;

A janela de recepção, denominada *RcvWindow*, é então calculada através da fórmula:

$$RcvWindow = RcvBuffer - [LastByteRcvd - LastByteRead]$$

O transmissor, por sua vez, mantém duas variáveis, *LastByteSent* e *LastByteAcked*. A diferença entre essas duas variáveis, *LastByteSent - LastByteAcked*, é a quantidade de dados não reconhecidos que o transmissor enviou para o receptor. Assim, o transmissor certifica-se durante toda a duração da conexão de que:

$$LastByteSent - LastByteAcked \leq RcvWindow$$

No controle de congestionamento, uma conexão TCP controla sua taxa de transmissão limitando sua quantidade de dados transmitidos e ainda não reconhecidos. Essa quantidade é denominada janela de congestionamento e é ajustada dinamicamente pelo TCP.

A janela de congestionamento, chamada de *CongWin*, impõe uma limitação adicional à quantidade de tráfego que um sistema final pode transmitir durante uma conexão. Especificamente, a quantidade de dados não reconhecidos que um transmissor pode ter em uma conexão TCP não pode exceder o mínimo de *CongWin* e *RcvWindow*, ou seja:

$$LastByteSent - LastByteAcked \leq \min \{ CongWin, RcvWindow \}$$

Uma conexão TCP começa com um pequeno valor de janela de congestionamento e estima a existência de largura de banda disponível no trajeto fim-a-fim aumentando pouco a pouco este valor até que ocorra a perda de um segmento (detectada por temporização ou por recebimento de reconhecimento duplicado). Quando isto ocorre, o TCP reduz o tamanho da janela de congestionamento para um nível seguro e, então, recomeça o processo de estimar a existência de largura de banda disponível.

Assim que a conexão é estabelecida entre dois sistemas finais TCP, o processo de aplicação do remetente escreve bytes para o *buffer* de envio do TCP remetente. O TCP encapsula os dados do *buffer* de envio em segmentos de tamanho igual ao tamanho máximo do segmento (*MSS - Maximum Segment Size*) e passa o segmento para a camada de rede para transmissão ao sistema final de destino. A janela de congestionamento do TCP regula os tempos nos quais os segmentos são entregues à camada de rede.

Inicialmente, a janela de congestionamento é igual a um *MSS*. O TCP envia o primeiro segmento e espera por um reconhecimento. Se esse segmento for reconhecido antes que seus temporizadores de retransmissão se esgotem, após um *RTT (round trip time)*, o remetente dobrará a janela de congestionamento e enviará dois segmentos de tamanho máximo. Esse procedimento continua enquanto a janela de congestionamento estiver abaixo do limite e os reconhecimentos chegarem antes de suas temporizações correspondentes. Esta fase do algoritmo é chamada de partida lenta (*slow start*).

A fase de partida lenta termina quando o tamanho da janela ultrapassa o valor limite, representado por uma variável chamada *threshold*. Quando a janela de congestionamento fica maior que o valor atual de *threshold*, ela deixa de ter um crescimento exponencial e passa a crescer linearmente, ou seja, o valor da janela de congestionamento aumenta um *MSS* para cada *RTT* para o qual cheguem reconhecimentos correspondentes a uma janela inteira. Essa fase é chamada prevenção de congestionamento (*congestion avoidance*).

Quando ocorre a perda de um segmento, isto é, quando o reconhecimento do segmento não chega antes do término da temporização, o valor da variável *threshold* é ajustado para metade do valor corrente da janela de congestionamento e o valor da janela de congestionamento é ajustada para um MSS. O remetente então novamente passa a aumentar a janela de congestionamento exponencialmente, até que a janela de congestionamento atinja o limite estabelecido pela variável *threshold*.

O modelo explicado acima é um dos primeiros algoritmos de controle de congestionamento do TCP, conhecido por *Tahoe* [RFC793, 1981], que diminui incondicionalmente o tamanho da janela de congestionamento para o tamanho de um segmento e entra no estágio de partida lenta após qualquer um dos tipos de eventos de perda citados (ou por três *acks* duplicados ou por esgotar o tempo de espera por confirmação).

A versão seguinte ao TCP *Tahoe* é o TCP *Reno* [Allman et al, 1999], que cancela o processo de partida lenta após detectar um evento de perda por receber três *acks* duplicados. O motivo de não entrar na fase de partida lenta é baseado na constatação de que mesmo se um pacote tenha sido perdido, a chegada de três confirmações posteriores indica que alguns segmentos foram recebidos no remetente e, portanto, a rede ainda é capaz de entregar alguns pacotes, mesmo perdendo outros pacotes devido ao congestionamento. Essa nova fase adicionada ao TCP *Reno* é chamada de recuperação rápida (*fast recovery*). No entanto, quando ocorrem eventos de perda de pacote por esgotamento do tempo de espera por confirmação, o TCP *Reno* não entra na fase de recuperação rápida, e sim na de fase de partida lenta. Essa decisão está relacionada com a idéia de que a rede não tem capacidade de entregar nenhum pacote e que provavelmente todos estão sendo descartados em algum ponto da rede. Para que o transmissor receba um pacote de confirmação, o pacote transmitido deve primeiro ser recebido pelo receptor. Como os eventos de perda de pacotes ocorrem por esgotamento do tempo de espera por confirmação, isto significa que nenhum pacote transmitido chegou no destino e portanto a rede está descartando todos os pacotes.

Se ignorarmos a fase de partida lenta, vemos que o TCP essencialmente incrementa o tamanho de sua janela para cada RTT (aumentando, assim, sua taxa de transmissão de forma aditiva) quando o caminho pela rede não está congestionado e diminui o tamanho de sua janela de um fator de 2 para cada RTT quando o caminho está congestionado. Por esta razão, o TCP é frequentemente chamado de algoritmo de aumento-aditivo, diminuição-multiplicativa (*additive-increase, multiplicative-decrease - AIMD*) [Kurose e Ross, 2004].

3.3 Controle de Congestionamento DCCP CCID2 – TCP LIKE Congestion Control

O CCID 2 [Floyd e Kohler, 2006] é um mecanismo de controle de congestionamento do DCCP similar ao controle de congestionamento do TCP. Ele também utiliza os mecanismos de partida lenta e AIMD, porém, como o DCCP é um protocolo não confiável, existem algumas diferenças importantes entre a implementação desses mecanismos no CCID 2 e no TCP.

Como no TCP, o CCID 2 também usa uma janela de congestionamento no transmissor para limitar a quantidade de segmentos não reconhecidos sendo enviados, mas ele não pode usar *acks* cumulativos, pois essa abordagem não faz sentido para um protocolo não confiável. Por isso, outro mecanismo é necessário para garantir que quando segmentos são perdidos, o transmissor diminua sua taxa de transmissão de forma apropriada.

O mecanismo usado pelo CCID 2 é uma variante do *selective acknowledgements* (SACK) [Mathis et al, 1996], [Allman et al, 1999], [Allman et al, 2002], [Blanton et al, 2003] e a comunicação confiável do *ack* enviado pelo receptor ao transmissor, usando um vetor de *acks* (*ack vector*) e confirmação dos *acks*. O receptor continua informando ao transmissor que o segmento k foi recebido até que o transmissor envie uma mensagem com o vetor de *acks* informando que recebeu a confirmação de que o segmento k foi recebido. O vetor de *acks* descreve exatamente para quais segmentos o transmissor recebeu o *ack* do receptor e quais tiveram marcação ECN da rede.

O transmissor mantém 3 parâmetros medidos em segmentos da camada de transporte:

- *cwnd* – Indica o tamanho da janela de congestionamento, isto é, o número máximo de segmentos de dados (DCCP-Data, DCCP-DataAck, DCCP-Request e DCCP-Response) não reconhecidos permitidos na rede.
- *ssthresh* – Indica o limite da fase *slow start* no algoritmo de controle de congestionamento.
- *pipe* – Indica o cálculo do transmissor do número de segmentos de dados pendentes na rede.

Esses parâmetros são manipulados e seus valores iniciais são calculados de acordo com a implementação do TCP baseado em SACK [Mathis et al, 1996], [Allman et al, 1999], [Allman et al, 2002], [Blanton et al, 2003]. A única diferença é que eles são medidos em segmentos e não em bytes.

O transmissor tem permissão para enviar um segmento somente se *pipe* for menor que *cwnd*. Cada segmento de dados enviados incrementa *pipe* de 1 segmento. O transmissor decrementa *pipe* toda vez que detecta que um segmento de dados deixou a rede, tanto por ter sido reconhecido quanto por ter sido descartado ao longo do caminho. Para os segmentos reconhecidos, o transmissor decrementa a variável *pipe* para cada *ack* recebido. Para os segmentos descartados, o transmissor decrementa a variável *pipe* para cada segmento cujo descarte foi detectado através de acks duplicados, mecanismo DCCP semelhante ao utilizado pelo TCP [RFC793, 1981]. O parâmetro *nundupack* indica o número de *acks* necessários para caracterizar uma perda. O valor do parâmetro *nundupack* é três, assim como no TCP. O segmento S é considerado perdido quando pelo menos *nundupack* segmentos transmitidos após S forem reconhecidos pelo receptor.

O último caso é o dos segmentos perdidos por temporização. O transmissor mantém temporizadores similares aos do TCP para o caso em que uma janela inteira de segmentos é perdida. O transmissor estima o RTT pelo menos uma vez por janela de dados e usa os mesmos algoritmos do TCP [Paxson e Allman, 2000] para manter este valor coerente com as condições da rede. De forma semelhante ao TCP, quando um segmento é perdido por temporização, o transmissor ajusta a variável *pipe* para 0. O transmissor não decrementa o valor de *pipe* quando detecta a perda de segmentos que não contenham dados, como por exemplo DCCP-Ack .

Um evento de congestionamento na rede faz com que o CCID 2 reduza sua janela de congestionamento. Ele é caracterizado por pelo menos um segmento perdido ou marcado. Como no TCP, dois segmentos perdidos ou marcados são considerados parte do mesmo evento de congestionamento se o segundo segmento foi transmitido antes da perda ou marca do primeiro segmento ter sido detectada.

Para cada evento de congestionamento, o valor de *ssthresh* é reduzido para metade da janela de congestionamento ($cwnd/2$) e a janela de congestionamento é reduzida para 1 segmento. O valor de *cwnd* e *ssthresh* nunca são inferiores a 1 e 2, respectivamente.

Enquanto *cwnd* for menor que *ssthresh*, o transmissor está na fase de partida lenta e a janela de congestionamento começa sendo incrementada de um segmento para cada dois novos *acks* recebidos (sem marcação ECN), até um máximo de $[(Ack\ Ratio)/2]$ segmentos por

ack. Esta forma permite que a janela de congestionamento do CCID 2 cresça tão agressivamente quanto a do TCP.

O transmissor CCID 2 usa o parâmetro *Ack Ratio* para influenciar a taxa com que os receptores geram *acks* para os segmentos recebidos, controlando assim o congestionamento no caminho inverso. O valor padrão para a taxa de *acks* é 2, isto é, envia 1 *ack* para cada 2 segmentos. Desta forma, o CCID 2 tem o mesmo comportamento do TCP *delayed acks* [Kohler et al, 2006].

Quando *cwnd* for maior ou igual a *ssthresh*, a janela de congestionamento é incrementada de 1 segmento para cada janela de dados reconhecidos sem perda ou marcação ECN.

Um segundo vetor pode, opcionalmente, ser usado para controle de fluxo, informando à aplicação no transmissor que segmentos foram descartados devido ao *buffer* do receptor estar cheio. O controle de fluxo de um protocolo não confiável deve ser implementado de forma diferente em relação ao TCP. Por exemplo, se o segmento *n* foi recebido, mas ainda não foi lido pela aplicação e o *buffer* de recepção estiver cheio, caso chegue um novo segmento, o DCCP deve descartar o segmento *n* do *buffer* e armazenar o segmento mais recente.

Uma das limitações do TCP é a ausência de controle de congestionamento para os *acks* enviados do receptor para o transmissor. O controle de congestionamento no caminho reverso é particularmente importante em redes com largura de banda assimétrica como redes sem fio. No DCCP, esse controle é feito e o receptor ajusta a taxa de envio de *acks* baseado no congestionamento do caminho reverso.

Outra diferença importante é que no DCCP o tamanho da janela de congestionamento e demais parâmetros são medidos em número de segmentos da camada de transporte e não em bytes.

O CCID 2 é indicado para aplicações que necessitem enviar o máximo de dados possível em um curto intervalo de tempo (rajadas) aproveitando o máximo da banda disponível e que suportem variações abruptas nas taxas de transmissão. Um exemplo de aplicação usando o CCID 2 seria um *streaming* de vídeo no qual o receptor armazena uma grande quantidade de informação em seu *buffer* antes de iniciar sua execução.

3.4 Controle de Congestionamento DCCP CCID3 – TFRC Congestion Control

O mecanismo de controle de congestionamento usado no CCID 3 [Floyd et al, 2006a] segue o mecanismo *TCP-Friendly Rate Control* (TFRC) padronizado pelo IETF [Handley et al, 2003], com algumas poucas alterações.

O TFRC é um mecanismo de controle de congestionamento baseado no receptor (*receiver-based*) projetado para ser “justo” ao competir por largura de banda com fluxos TCP. Ele utiliza uma abordagem diferente em relação ao CCID 2 para o controle de congestionamento. Ao invés de operar com uma janela de congestionamento, o transmissor TFRC ajusta sua taxa de transmissão em resposta às informações que recebe do receptor. O receptor limita a taxa de envio de pacotes do transmissor através do envio de segmentos de *feedback* contendo informações do estado da conexão, como a taxa de recepção, intervalos de perda e o tempo em que um pacote permanece na fila de recepção (*buffers* de recepção) até que seja confirmado como recebido pelo transmissor. O transmissor usa todas essas informações para ajustar a sua taxa de transmissão.

O TFRC deve ser usado apenas por aplicações que se beneficiam da lenta alteração da largura de banda, evitando o comportamento do TCP e do CCID 2 que dividem a taxa de transmissão ao meio após a perda de um único segmento. No TFRC a taxa de transmissão somente é dividida ao meio quando o transmissor não recebe nenhum segmento de *feedback* do receptor em um intervalo igual a 4 RTTs. Esta característica o torna indicado para o uso em aplicações de *streaming* que não armazenam muitas informações em *buffers* no receptor (por exemplo, videoconferência) e em telefonia IP. A desvantagem de ter uma variação menor da vazão em relação ao TCP é que o TFRC responde mais lentamente às alterações na largura de banda também quando há banda disponível.

Assim como o CCID 2, o CCID 3 foi projetado para aplicações que alteram sua taxa de envio de segmentos em resposta ao congestionamento da rede e não para aplicações que variam o tamanho do segmento.

Se o transmissor não receber *feedback* antes do *nofeedback timer* expirar, ele diminui à metade sua taxa de transmissão. A taxa de transmissão nunca é menor do que 1 segmento a

cada 64 segundos. Todos os *acks* são considerados *feedbacks* e devem conter as opções *Loss Intervals*, *Elapsed Time* e *Receive Rate*. O receptor anuncia o intervalo de perda usando a opção *Loss Interval*, que deve ser enviada pelo receptor em todos os *acks*. A opção informa até 28 intervalos de perdas, embora o TFRC atualmente use os 9 intervalos de perdas mais recentes. Isto permite ao transmissor calcular a taxa de evento de perda. O *ack vector* não é necessário no CCID 3, pois a opção *Loss Interval* fornece toda a informação necessária para gerenciar a taxa de transmissão.

A operação básica do CCID 3 é baseada no cálculo da taxa de evento de perdas, que é o número de eventos de perdas dividido pelo número de segmentos transmitidos, com maior ponderação para os últimos intervalos de perdas. A taxa de evento de erro, o RTT estimado e a média do tamanho do segmento são usadas na equação de vazão do TCP, como especificado em [Handley et al, 2003]. O resultado é uma taxa de transmissão muito próxima à do TCP nas mesmas condições. Os transmissores CCID 3 são limitados a esta taxa.

O transmissor inicia na fase *slow start* e dobra sua taxa de transmissão em cada RTT. A fase *slow start* termina quando o receptor informa segmentos perdidos ou marcados. Após a fase *slow start*, o transmissor usa a taxa de evento de perdas para calcular sua taxa de transmissão permitida.

Como descrito em [Handley et al, 2003], um intervalo de perdas inicia-se com um segmento perdido ou marcado com ECN e continua até um RTT depois do término de segmentos perdidos ou marcados. A Figura 3.1 ilustra dois exemplos de intervalos de perdas.

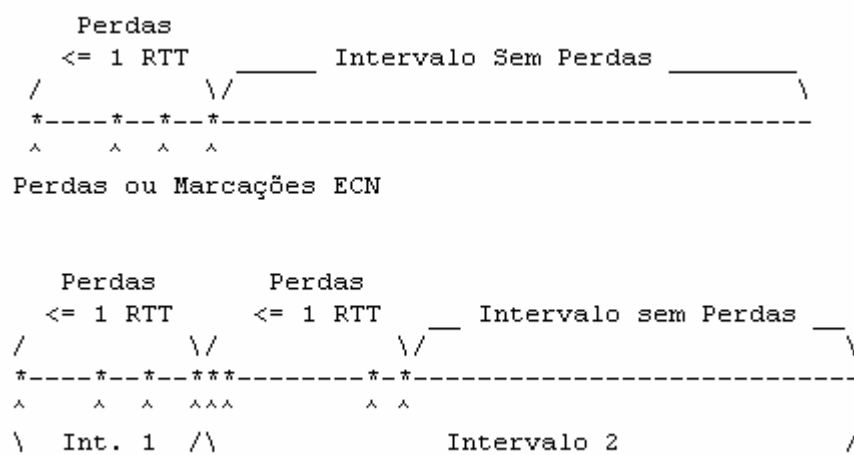


Figura 3.1. Exemplos de Intervalos de Perdas

Segmentos que não transportam dados (DCCP-Ack, DCCP-Close, DCCP-CloseReq, DCCP-Reset, DCCP-Sync e DCCP-SyncAck) que são perdidos ou marcados não afetam um intervalo de perdas.

De forma resumida, o controle de congestionamento TFRC funciona da seguinte maneira:

1. O transmissor transmite segmentos de dados (DCCP-Data). Cada segmento DCCP-Data possui um número de sequência e um campo CCVal que indica o tamanho da janela que será usada pelo receptor para determinar quando múltiplas perdas pertencem a um único evento de perdas.

2. O receptor envia segmentos DCCP-Ack reconhecendo os segmentos de dados a uma taxa de, no mínimo, uma vez por RTT, a menos que o transmissor transmita a uma taxa menor do que um segmento por RTT. Cada segmento DCCP-Ack usa um número de sequência, identificando os segmentos mais recentes recebidos do transmissor e incluem *feedbacks* sobre os intervalos de perdas recentes acontecidos no receptor.

3. O transmissor continua a transmitir segmentos DCCP-Data controlados pela taxa de transmissão permitida. Enquanto continuar recebendo segmentos DCCP-Acks, o transmissor atualiza sua taxa de transmissão como especificado em [Handley et al, 2003]. Essa atualização é baseada na taxa de eventos de perdas, calculada pelo transmissor baseado no *feedback* fornecido pelo receptor.

4. O transmissor calcula o RTT e o parâmetro *nofeedback timer*, que indica o tempo em que, se nenhum *feedback* for recebido (no mínimo 4 RTTs), o transmissor reduz ao meio sua taxa de transmissão.

A equação de cálculo da taxa usada nas implementações atuais do TFRC é uma versão simplificada de equação de vazão (*throughput*) do TCP *Reno*. A equação para o TFRC é:

$$X = \frac{s}{\left(RTT \cdot \sqrt{\frac{2 \cdot b \cdot p}{3}} + 4 \cdot RTT \cdot \left(3 \cdot \sqrt{\frac{3 \cdot b \cdot p}{8}} \right) \cdot p \cdot (1 + 32 \cdot p^2) \right)} \quad (1)$$

Onde X é a taxa de transmissão em bytes/segundos; s é o tamanho do segmento em bytes; RTT é o tempo de ida e volta (*round-trip time*) em segundos; p é a taxa de eventos de perdas, entre 0 e 1, calculada como a fração do número de eventos de perdas sobre o número de segmentos transmitidos e b é o número de segmentos reconhecidos por *ack*. Na prática, os

parâmetros s (tamanho do segmento), p (taxa de evento de perdas) e RTT devem ser medidos pelo TFRC. O parâmetro b é um valor fixo e igual a 1.

4 Redes Mesh sem Fio com Múltiplos Saltos

Neste capítulo é realizada uma breve introdução às redes locais sem fio IEEE 802.11 e às redes sem fio com múltiplos saltos (redes mesh). Em seguida, são abordados os principais desafios encontrados para o controle de congestionamento TFRC em redes mesh e, por último, são apresentados trabalhos relacionados à melhoria de desempenho do controle de congestionamento nestas redes.

4.1 Redes IEEE 802.11

O padrão IEEE 802.11 compõe o conjunto de padrões da família 802.x e oferece um conjunto de especificações descritas em diversos documentos disponibilizados pelo IEEE (*Institute of Electrical and Electronics Engineers*). Este padrão é subdividido em diversas especificações, sendo as mais conhecidas as 802.11a, 802.11b, 802.11g e 802.11e. O padrão inicial foi aceito pelo IEEE em 1997. Em 2007, o IEEE agregou o padrão inicial e todas as extensões existentes a uma única especificação denominada IEEE 802.11-2007 [IEEE, 2007].

A família de padrões 802.x refere-se às camadas física e de enlace do modelo ISO/OSI [ISO, 1994]. O padrão 802.11 especifica o mecanismo de controle de acesso ao meio (*Medium Access Control* - MAC) para redes sem fio e a camada física (PHY) para prover conectividade entre dispositivos fixos, portáteis e móveis dentro de uma rede local (LAN – *Local Area Network*).

As especificações do padrão 802.11 podem ser descritas da seguinte forma:

- 802.11 é o primeiro padrão para redes sem fio da IEEE. Permite conexões a uma taxa de até 2 Mbits/s na frequência de 2.4 GHz. Atualmente não são mais fabricados produtos que seguem esta especificação e a mesma foi estendida para o padrão 802.11b;
- 802.11a é também uma extensão do padrão 802.11. Foi criada na mesma época do 802.11b e provê transmissão a 54 Mbits/s na frequência de 5 GHz.
- 802.11b também chamado de 802.11 *High Rate* ou Wi-Fi, é uma extensão do 802.11 e oferece transmissão a uma taxa de até 11 Mbits/s na frequência de 2.4

GHz. Não funciona com equipamentos que seguem o padrão 802.11a, pois cada um dos padrões funciona em uma frequência diferente;

- 802.11g oferece transmissão a uma taxa de até 54 Mbits/s na frequência de 2.4 GHz. Foi criado para substituir e ser compatível com o padrão 802.11b.
- 802.11e provê um mecanismo para transmissão de dados baseado na qualidade de serviço da aplicação. É uma adaptação voltada para as aplicações que transmitem dados multimídia em redes sem fio, como as aplicações VoIP (Voz sobre IP).

Existem dois modos de operação suportados pelo padrão: infraestruturado (no qual os nós se comunicam com um ponto de acesso fixo) e ad hoc (no qual os nós se comunicam entre si, sem a necessidade de controle centralizado). A seguir será apresentada uma descrição do mecanismo de acesso ao meio obrigatório, ou seja, da subcamada MAC para os modos de operação ad hoc [IEEE, 2007] e infraestruturado, sendo o único disponível no modo ad hoc.

O método de acesso utilizado pelo 802.11 no modo ad hoc é através de uma função de coordenação distribuída (DCF – *Distributed Coordination Function*). Essa função é baseada numa versão do CSMA/CA, cujo mecanismo funciona da seguinte maneira:

1. Se o meio está ocioso por um intervalo de tempo chamado DIFS (*DCF Interframe Space*), o nó acessa o meio para iniciar a transmissão. Logo, pode-se afirmar que em situações de carga leve, o atraso de acesso ao canal é igual a DIFS.

2. Caso o meio esteja ocupado após a duração de DIFS, o nó espera um intervalo de tempo aleatório chamado tempo de *backoff*, cujo valor é escolhido dentro de um intervalo chamado de janela de contenção (CW), que varia de CWmin até CWmax. Inicialmente, a CW tem duração aleatória escolhida entre 0 e CWmin. Pequenas CWs implicam em intervalos de *backoff* muito próximos um dos outros, fazendo com que a probabilidade de colisão seja alta. Logo, para cada colisão que ocorre, a CW tem a sua duração duplicada, sendo essa duração limitada por CWmax. Quando uma transmissão é efetuada com sucesso, a duração da CW é reiniciada, ou seja, volta a ser um valor aleatório escolhido entre 0 e CWmin.

3. Após a escolha do valor de *backoff*, um contador entra em ação e vai sendo decrementado até que chegue em zero, quando o nó ganha o acesso ao meio e transmite seu dado. Neste momento, todos os nós que ouvirem a transmissão atualizam seus respectivos NAVs (*Network Allocation Vector*), os quais determinam o instante de tempo mais próximo em que um nó pode “escutar” o meio para iniciar uma transmissão.

4. Durante o procedimento de *backoff*, se o nó detecta o meio ocupado, o contador de *backoff* é paralisado e este só volta a ser decrementado quando o nó detecta o meio ocioso por um intervalo de tempo igual a DIFS.

5. Cada um dos nós executa o procedimento de *backoff* pelo menos uma vez após toda transmissão bem sucedida.

É importante notar que, nesse mecanismo, nós que já efetuaram uma transmissão com sucesso possuem uma CW menor que os nós que sofreram colisão. Consequentemente, o tempo de *backoff* para os nós que já conseguiram transmitir é menor.

Logo, tal mecanismo não garante a equidade no acesso ao meio. Em outras palavras, o mecanismo de acesso ao meio do 802.11 favorece os terminais que já executaram uma transmissão com sucesso, em detrimento daqueles que sofreram colisão. Numa situação ideal, desejar-se-ia que os nós que esperaram mais para transmitir devido a colisões tivessem prioridade maior.

Outro mecanismo importante no 802.11 é a transmissão de quadros de confirmação de recebimento (*ack*). Tais quadros precisam ser enviados após a recepção de um dado para garantir que sua entrega tenha sido bem sucedida. Após a recepção do dado, o nó receptor espera um intervalo de tempo chamado de SIFS (*Short Interframe Space*) e envia um quadro *ack*. Os outros nós devem então esperar por DIFS mais o tempo de *backoff*, reduzindo a probabilidade de colisão, já que a prioridade de um quadro *ack* é maior, pois o SIFS é menor que o DIFS. Caso não haja a recepção de um quadro de *ack* por parte do nó origem, este tenta retransmitir o dado até que, ou receba um *ack*, ou o número de tentativas de retransmissão chegue ao limite e a camada MAC informe a camada superior que a transmissão falhou. A Figura 4.1, adaptada da especificação IEEE 802.11-2007 [IEEE, 2007] mostra um esquema completo de funcionamento da DCF.

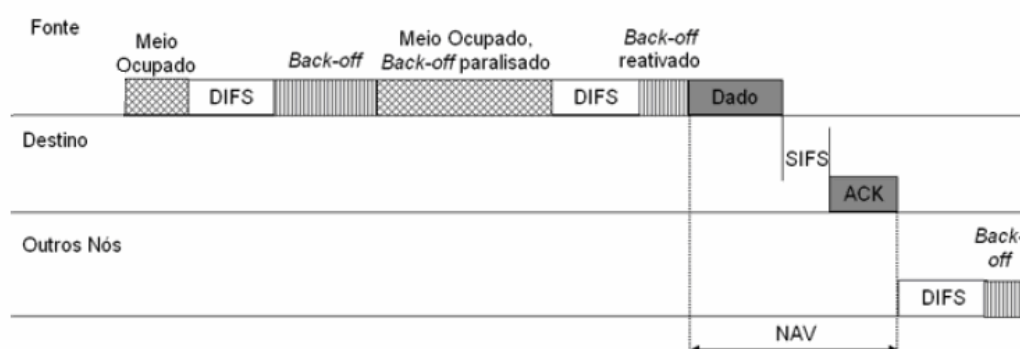


Figura 4.1. IEEE 802.11 DCF

Dois problemas clássicos de redes sem fio são conhecidos como problema do terminal escondido e problema do terminal exposto [Conceição, 2006].

O problema do terminal escondido se refere à situação em que ocorre colisão de quadros num nó receptor devido à transmissão simultânea de dois ou mais nós que estão no alcance do receptor, porém estão fora de alcance entre si. Esta situação é mostrada na Figura 4.2.

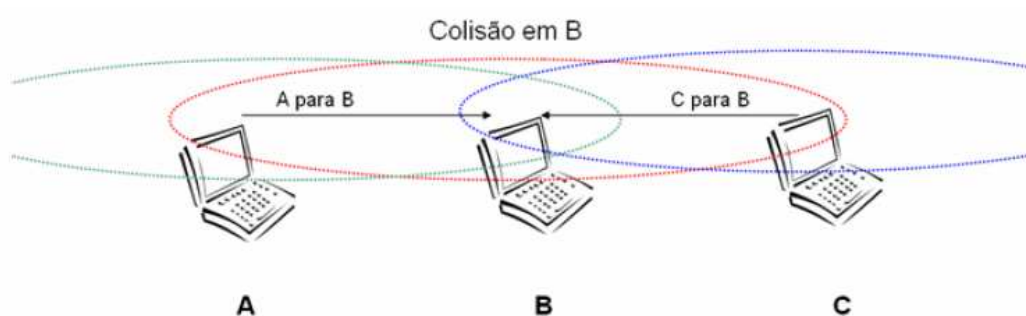


Figura 4.2. Problema do Terminal Escondido

Já o problema de terminal exposto se refere à incapacidade de um nó de transmitir sua informação devido ao bloqueio no acesso ao meio causado por uma transmissão próxima (dentro de seu alcance), porém para um destinatário diferente e fora do alcance do nó que transmite. Esta situação é mostrada na Figura 4.3.

É desejável que o protocolo MAC tenha algum tipo de mecanismo a fim de evitar ambos os problemas que podem causar queda significativa na vazão da rede.

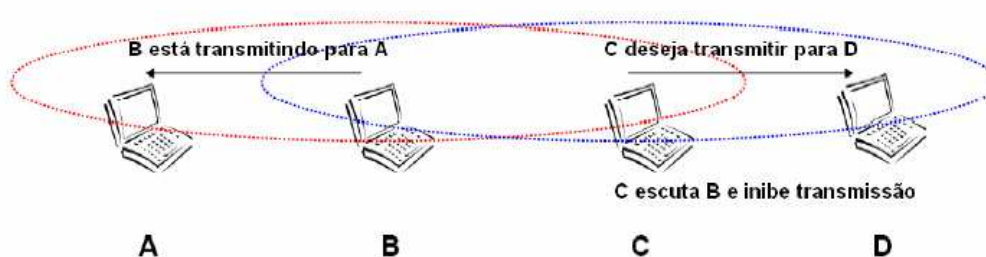


Figura 4.3. Problema do Terminal Exposto

Para tentar minimizar o problema de terminal escondido, o 802.11 possui um mecanismo opcional de controle RTS/CTS descrito a seguir e ilustrado na Figura 4.4, adaptada da especificação IEEE 802.11-2007 [IEEE, 2007]:

1. O transmissor envia um quadro RTS (*Request to Send*) para o receptor. O pacote inclui informação sobre o destinatário do próximo pacote de dados a ser transmitido e o tempo total da transmissão do dado. O RTS é recebido por todos os nós dentro do raio de alcance do transmissor.

2. Cada um dos nós que recebeu o RTS atualiza seu NAV.

3. Depois de esperar SIFS, o receptor envia um quadro CTS (*Clear to Send*) se estiver apto a receber o dado. O CTS contém informações sobre a duração da transmissão e é recebido por todos os nós e dentro do raio de alcance do receptor.

4. Todos os nós que receberam o CTS atualizam seus NAVs. Como o conjunto de nós que recebeu o RTS não é necessariamente o mesmo que recebeu o CTS, fica evidente a existência de alguns terminais escondidos.

5. Depois que o CTS é recebido pelo transmissor, todos os nós no alcance do transmissor e do receptor ficam cientes que o meio está reservado para apenas um transmissor. O transmissor então espera SIFS e inicia a transmissão do dado.

6. O receptor, depois de receber o dado, espera SIFS e envia um *ack*.

7. Depois da conclusão da transmissão, o NAV de todos os nós envolvidos no processo indica que o meio está livre (exceto no caso que algum desses nós tenha recebido algum outro RTS ou CTS durante o processo).

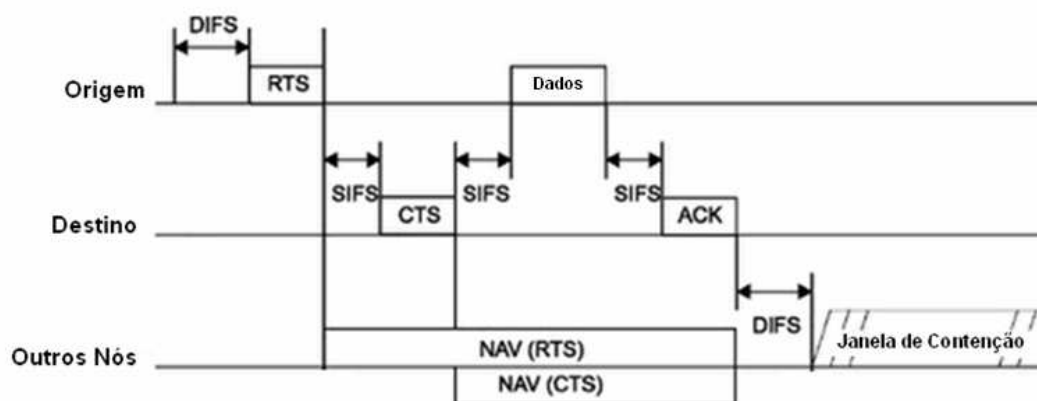


Figura 4.4. Mecanismo RTS/CTS do Padrão IEEE 802.11

É importante notar que quadros RTS e CTS são como quaisquer outros quadros e podem ocorrer colisões envolvendo qualquer um dos dois. Também nota-se que existe a inclusão de um *overhead* não desprezível ao se usar esse mecanismo. O que se faz é adotar um limiar baseado no tamanho do quadro de dados para habilitar ou desabilitar o mecanismo.

Se o tamanho do quadro for maior que o limiar, o mecanismo é habilitado; caso contrário, é desabilitado.

O padrão IEEE 802.11 só oferece comunicações em redes ad hoc de um único salto. Redes sem fio de múltiplos saltos podem ser construídas através de roteamento nível 3 de redes IEEE 802.11 (conforme será comentado na próxima seção). Uma nova extensão ao padrão está sendo proposta para comunicação em múltiplos saltos na camada MAC (802.11s) [Muchaluat-Saade et al, 2008].

4.2 Redes Mesh Sem Fio Com Múltiplos Saltos

De acordo com [Abelém et al, 2007], redes mesh (redes em malha sem fio) são redes com topologia dinâmica, variável e de crescimento orgânico, constituídas por nós cuja comunicação, no nível físico, é feita através de variantes dos padrões IEEE 802.11 e 802.16, e cujo roteamento é dinâmico.

Mesh não é propriamente uma tecnologia, mas sim um conceito. Uma rede mesh caracteriza-se por nós sem fio que se comunicam diretamente com um ou mais nós sem a necessidade de um ponto de acesso central. Cada nó opera não apenas como um *host* da rede mas como um roteador, encaminhando pacotes para outros nós mesmo que estes últimos não estejam necessariamente em contato direto com o destino dos pacotes. Desta forma, uma rede mesh é composta de vários nós/roteadores, constituindo um único backbone, possibilitando que o cliente se conecte a qualquer um destes nós através de acessos com ou sem fio. Os nós têm a função de roteadores, regenerando o sinal recebido e encaminhando para o próximo nó, fornecendo uma área de cobertura ampla através de saltos menores. Cada nó está conectado a um ou mais nós. Desta maneira é possível transmitir mensagens de um nó a outro por diferentes caminhos, computados através dos protocolos de roteamento dinâmico, garantindo assim a robustez do sistema. A Figura 4.5, retirada de [Abelém et al, 2007], ilustra uma topologia de rede mesh.

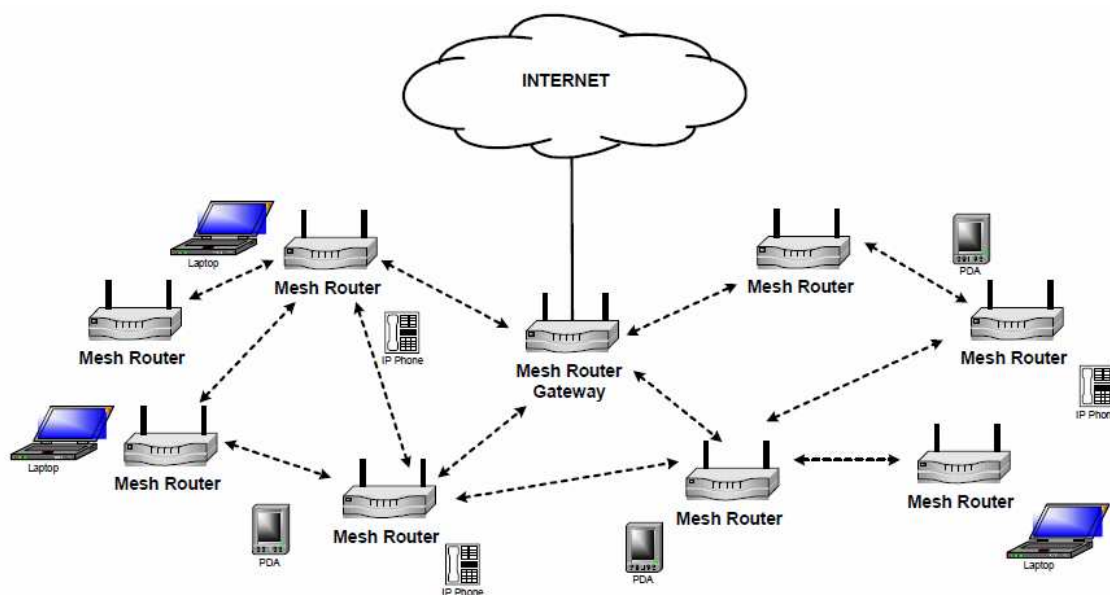


Figura 4.5. Rede Mesh

Uma rede mesh caracteriza-se também pela sua capacidade de organização. Os próprios nós se autoconfiguram e se adaptam às mudanças de topologia. A rede deve ser capaz de se adaptar a mudanças causadas pela adição, retirada ou falha de determinados nós. Quanto mais densa for a rede (quanto maior o número de roteadores sem fio), maior é a confiabilidade da rede, pois aumentam as probabilidades de múltiplos caminhos entre a origem e o destino da comunicação. Como existem múltiplos caminhos, é necessário o uso de um protocolo de roteamento para a escolha da melhor rota disponível. Os principais protocolos de roteamento utilizados para redes mesh são apresentados em [Abelém et al, 2007].

Nas redes em malha sem fio, os nós que compõem o *backbone* são geralmente estáticos, podendo ser alimentados diretamente pela rede elétrica. Do ponto de vista do roteamento, não há problemas com economia de energia ou mobilidade. Por outro lado, a transmissão sem fio se traduz em enlaces com altas taxas de erro por bit além de qualidade do enlace bastante variável no tempo. Diferentes fontes de interferência podem influenciar na qualidade do enlace, como equipamentos de rede, outros equipamentos elétricos como motores, telefones, fornos de microondas, etc.

Recentemente, estas redes têm sido alvo de estudo de grandes empresas fornecedoras de equipamentos de rede e operadoras de telecomunicações e de instituições acadêmicas. Dentre as aplicações implementadas, estão o fornecimento de acesso banda larga dentro dos

campi das universidades e para as comunidades próximas e a criação de cidades digitais, permitindo também a inclusão digital. Exemplos de projeto piloto de redes de acesso sem fio do tipo mesh são o ReMesh na UFF em Niterói/RJ [ReMesh, 2007], RoofNet no MIT [Aguayo et al, 2004], Google Mesh na Califórnia [Google, 2007], VMesh na Grécia [Tsarmopoulos, 2005], MeshNet na UCSB [Ho et al, 2004], Microsoft Mesh [Draves et al, 2004], entre outros.

As redes mesh oferecem um menor custo de infraestrutura, pois requerem menor número de pontos de acesso às redes cabeadas (Internet), proporcionando conectividade a uma área muito maior que as redes sem fio infraestruturadas.

4.3 Principais Desafios do Controle de Congestionamento TFRC em Redes Mesh

O *TCP-Friendly Rate Control* (TFRC) é um mecanismo de controle de congestionamento baseado em taxa projetado para oferecer variação suave na taxa de transmissão, baixo atraso (*delay*) e uma transmissão "amigável" a fluxos TCP. O TFRC foi projetado para redes cabeadas e testes comprovaram que seu uso não é adequado em redes sem fio [Navaratnam et al, 2006], [Ware et al, 2001], [Ray et al, 2003], [Abelém et al, 2007], [Li et al, 2004]. O principal motivo para o baixo desempenho do TFRC em redes sem fio é o funcionamento da camada MAC no meio sem fio, que utiliza procedimentos de retransmissão de quadros e *backoff* exponencial, levando o mecanismo de controle de congestionamento do TFRC a ajustar sua taxa de transmissão de forma equivocada. A forma padrão do TFRC degrada o desempenho do DCCP em redes sem fio com múltiplos saltos, aumentando consideravelmente o atraso dos pacotes, prejudicando assim o seu uso para aplicações multimídia.

O grande desafio do TFRC em redes sem fio está relacionado ao funcionamento da camada MAC do padrão IEEE 802.11, que usa o mecanismo CSMA/CA para evitar colisões. A saturação na camada MAC 802.11, os atrasos de contenção, as retransmissões e o *overhead* causado pelo mecanismo RTS/CTS são os principais degradantes do desempenho do TFRC. Este comportamento é referido como *RTS/CTS jamming* [Ware et al, 2001] e congestionamento induzido pelo RTS/CTS [Ray et al, 2003]. Mesmo sem uso de RTS/CTS, o TFRC ignora a saturação da camada MAC. Com isso, ele aumenta sua taxa máxima de transmissão acima do que a rede é capaz de suportar, levando-a ao colapso. Uma carga que é

maior do que a camada MAC pode sustentar é gerada pelo DCCP, causando múltiplas retransmissões e aumento do atraso. Embora o TFRC eventualmente receba algumas notificações de perda devido às retransmissões de quadros, esses *feedbacks* chegam muito tarde para que o TFRC ajuste a sua taxa de forma a evitar a saturação da camada MAC. Em redes mesh, esse problema se agrava devido às transmissões sem fio em múltiplos enlaces adjacentes, onde os transmissores estão no alcance dos demais.

4.4 Trabalhos Relacionados

Existe uma série de trabalhos sobre controle de congestionamento em redes sem fio com múltiplos saltos. Em sua maioria, abordam propostas para otimizar o desempenho do TCP, que também tem seu funcionamento prejudicado pelos mecanismos de contenção da camada MAC. Além disso, encontram-se também propostas de alteração do TFRC e diversos testes sobre o assunto.

Em [Navaratnam et al, 2006], há o estudo do desempenho do controle de congestionamento TFRC do protocolo de transporte DCCP através de testes em redes mesh sem fio com cenários variados. Um dos testes verifica como o aumento do número de saltos (*hops*) prejudica a vazão do TFRC e compara o resultado com o obtido pelo TCP. Os testes demonstraram que embora o TFRC sofra com os efeitos da saturação da camada MAC, seu resultado é ligeiramente superior ao TCP. Em testes subsequentes, foi verificado como o TFRC afeta os fluxos concorrentes TCP, UDP e DCCP-TFRC. Conforme esperado, o UDP sempre ocupa toda a largura de banda disponível, ou seja, compete de forma injusta com outros fluxos TCP ou DCCP-TFRC. Nos testes com fluxos concorrentes DCCP-TFRC e TCP, foi verificado que o DCCP-TFRC compete de forma mais justa com o TCP, fazendo com que aplicações multimídia utilizando o DCCP-TFRC não ocupem toda a largura da banda da rede.

Em [Li et al, 2004], os autores propõem uma melhoria no algoritmo TFRC do protocolo DCCP chamada RE TFRC (*Rate Estimation TFRC*). O RE TFRC estima o ponto de saturação da camada MAC. O protocolo cria um modelo que calcula o RTT máximo em uma rede sem fio na qual ainda não exista saturação na camada MAC. A partir deste valor de RTT máximo teórico é derivada uma taxa de evento de perdas que reflete o nível de congestionamento da camada MAC. Com a taxa de eventos de perdas derivada do RTT máximo antes da saturação e o RTT atual, é calculada a taxa de transmissão do RE TFRC. Além disso, também é calculada a taxa de transmissão TFRC normal. A nova taxa de

transmissão será a menor entre a taxa calculada pelo TFRC e a taxa calculada pelo RE TFRC. Os resultados apresentaram redução de 5% a 43% nos valores de RTT. Embora os resultados tenham sido significativos na redução do RTT, a metodologia utilizada para a modelagem do RTT máximo antes da saturação da camada MAC é baseada na topologia física da rede (distância entre nós, número de saltos, etc.) e parâmetros da camada física (tipo de modulação utilizada, tamanho dos quadros de controle, etc.) tornando sua implementação difícil em redes reais.

Em [Zhou et al, 2007], os autores propõem uma melhoria no controle de congestionamento TFRC do DCCP alterando a forma como a taxa de evento de perdas é calculada. A vazão do TFRC é calculada baseada na equação do TCP Reno, que utiliza a taxa de evento de perdas para o cálculo da vazão máxima permitida. A forma como a taxa de eventos de perdas é calculada no mecanismo padrão TFRC do DCCP considera que qualquer perda ocorrida é decorrente de congestionamento na rede. Na adaptação proposta pelos autores, os eventos de perdas são divididos em perdas devido ao congestionamento e outras perdas. As perdas que não são devidas ao congestionamento têm peso muito menor no cálculo da taxa de evento de perdas do que as perdas devido ao congestionamento. A diferenciação entre as perdas é feita através do parâmetro N , utilizado para determinar o aumento do atraso devido a filas no roteador. O parâmetro N é calculado utilizando a taxa de transmissão atual (X), o RTT atual (RTT) e o menor RTT (RTT_{min}) calculado pelo TFRC ($N = X \cdot (RTT - RTT_{min})$). Na implementação dos autores, somente quando o parâmetro N é maior que 1, as perdas são relativas ao congestionamento na rede. Em sua adaptação, o transmissor TFRC calcula o valor de N , se N for maior do que 1, a taxa de eventos de perdas é calculada normalmente, conforme implementação padrão do TFRC. Caso contrário, a taxa de eventos de perdas é dividida por 3, valor obtido comparando a equação de vazão do TCP Reno utilizada pela implementação padrão do TFRC com a equação da vazão do TCP Veno [Peng e Liew, 2003], de onde o mecanismo de diferenciação de perdas foi originalmente proposto. Desta forma, as perdas que não são devidas ao congestionamento influenciam menos a redução vazão do TFRC do que as perdas ocasionadas pelo congestionamento da rede. A adaptação proposta pelos autores funciona melhor em redes sem fio com maiores taxas de erro. Nos testes realizados, houve aumento de até 70% na vazão do TFRC.

Já em [Fu, 2003], é analisado o impacto de redes sem fio de múltiplos saltos na vazão e perdas do TCP. O estudo mostra que, dada uma determinada topologia e padrão de fluxos, existe uma janela de transmissão W^* na qual o TCP consegue a melhor vazão. Contudo, o TCP não limita sua janela de transmissão em um valor próximo a W^* , e tipicamente aumenta

sua janela de transmissão para valores muito maiores que W^* . Uma janela de transmissão maior que W^* resulta no aumento da saturação da camada MAC e conseqüentemente aumento das perdas e degradação da rede. Para limitar a janela de transmissão em torno de W^* , os autores propõem duas técnicas chamadas *link RED* e *adaptive pacing*, com as quais conseguiram um aumento de 5% a 30% na vazão TCP em testes com topologias variadas. A técnica *link RED* monitora o parâmetro “número médio de quadros retransmitidos na camada MAC” e informa ao TCP sobre o estado da camada MAC através de marcações ECN em fluxos TCP quando esse parâmetro supera um valor limite. A técnica *adaptive pacing* melhora o reuso do canal, distribuindo o tráfego através dos nós intermediários de forma mais balanceada. Ela resolve o problema do terminal exposto devido a falta de coordenação entre nós que estão a 2 saltos de distância um do outro, coordenando o encaminhamento de quadros ao longo do caminho de forma mais eficiente. Os dois algoritmos trabalham juntos, sendo que o *adaptive pacing* é ativado pelo *link RED*. Quando um nó verifica que o número médio de retransmissões está abaixo do valor limite, ele calcula o intervalo de *backoff* de forma usual. Quando o número de retransmissões está acima do valor limite, o *adaptive pacing* é habilitado e o intervalo de *backoff* é incrementado de um valor equivalente ao tempo de transmissão do último quadro de dados enviado. Desta forma, uma melhor coordenação entre nós é alcançada considerando diferentes cargas na rede.

Também em [Mascolo et al, 2001], é proposta uma adaptação do controle de congestionamento utilizada no TCP denominada TCP *Westwood* (TCPW). O TCPW é uma modificação do algoritmo do TCP *NewReno* realizada apenas no transmissor para melhorar seu desempenho em redes sem fio. O TCPW adapta a taxa de transmissão ao nível de congestionamento dinâmico da rede. Na técnica TCPW, a largura de banda é estimada pelo transmissor baseada nas informações recebidas nos *acks* e na frequência com que são recebidos. Estes dados passam por uma série de filtros até que a estimativa da largura de banda é calculada. Após a detecção da perda de pacotes, que pode ser por congestionamento ou erros no enlace, o transmissor usa a largura de banda estimada para ajustar as variáveis *CWin* e *SSTRESH*, relativas ao tamanho da janela de congestionamento e ao limite da fase de partida lenta respectivamente. Ao invés de reduzir à metade a janela de congestionamento após uma perda, o TCPW ajusta os parâmetros de forma consistente com os valores calculados no período anterior à perda. Os autores denominam este mecanismo como “recuperação rápida”.

5 Adaptação do Mecanismo de Controle de Congestionamento TFRC do Protocolo de Transporte DCCP para Redes Mesh

Neste capítulo são apresentados os testes realizados com o protocolo DCCP padrão que serviram para analisar o comportamento do protocolo durante a fase de saturação da camada MAC do padrão IEEE 802.11 em redes em malha sem fio. Também é apresentada uma nova proposta para o mecanismo de controle de congestionamento denominada M-TFRC (Mesh-TFRC) [Ribeiro e Muchaluat-Saade, 2009]. Por último, é apresentada a comparação entre o TFRC e o M-TFRC em testes com diversos cenários de redes mesh.

5.1 Estudo do Comportamento do Protocolo DCCP Padrão

Nesta seção será analisado o desempenho do mecanismo de controle de congestionamento CCID 3 (TFRC) padrão do DCCP em redes mesh sem fio, através de simulações usando a ferramenta NS-2 [NS2, 2009]. A topologia mesh em cadeia (ou em linha) foi utilizada, pois é uma topologia simples que utiliza múltiplos saltos e que permite a análise da camada de transporte com mínima interferência de outros fatores. Para avaliar o desempenho do CCID 3 em redes mesh foram utilizadas topologias com 6 e 7 nós em cadeia.

O cenário consiste de 6 e 7 nós afastados entre si de 200 metros usando o protocolo IEEE 802.11 nas camadas MAC e física, com o mecanismo RTS/CTS desabilitado e utilizando o protocolo de roteamento dinâmico AODV (*Ad hoc On-demand Distance Vector*) [Perkins et al, 2003]. Todos os nós se comunicam através de conexões de rádio *half-duplex* idênticas, com alcance de transmissão nominal de 250 metros, distância de interferência de 550 metros e taxa de transmissão de 2 Mbps (valores *defaults* da implementação do IEEE 802.11 no NS-2), com a origem e o destino dos fluxos situados nos extremos da topologia. A Figura 5.1 ilustra a topologia utilizada.

Para os testes, foi utilizada uma aplicação CBR (*Constant Bit Rate*) com tamanho de pacote fixo em 1400 bytes, simulando o envio de um fluxo de vídeo. A vazão fim-a-fim e o atraso médio de ida foram medidos na camada de aplicação. O atraso médio de ida foi utilizado na análise ao invés do RTT (atraso de ida e volta), pois a implementação do protocolo DCCP no NS-2 não disponibiliza diretamente os valores de RTT de cada pacote e

não existe informação nos arquivos de *trace* que permitam relacionar os *acks* com os pacotes correspondentes.

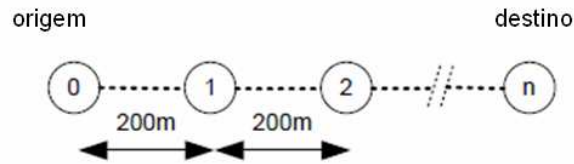


Figura 5.1. Topologia usada nas simulações

Em todas as simulações foram usados os valores default do NS-2 para os parâmetros MAC e PHY do 802.11, entre eles $CW_{Min} = 31$ e $CW_{Max} = 1023$.

Para reproduzir o comportamento do CCID 3 levando à saturação da camada MAC do padrão IEEE 802.11 em um ambiente de múltiplos saltos, foram realizados testes em que a taxa da aplicação CBR foi incrementada de 5 em 5 Kbps, iniciando em 100 Kbps até 200 Kbps. Os testes tiveram duração de 300 segundos, com o início da transmissão em $t = 10s$ e fim em $t = 300s$. A Tabela 5.1 mostra os resultados para uma topologia com 6 nós (5 saltos), apresentando a comparação entre a vazão média medida e a vazão solicitada pela aplicação CBR. As Figuras 5.2, 5.3 e 5.4 são gráficos gerados a partir dos valores da tabela, nas quais os valores do eixo x denominados “pontos de medição” servem como referência em relação à Tabela 5.1. Para cada ponto de medição foram realizadas 10 simulações utilizando o *ns-random* do NS-2, sendo calculados o valor médio e o intervalo de confiança de 95%. As colunas variação da vazão e variação do atraso representam a comparação do ponto de medição atual com o ponto de medição anterior, de acordo com as Equações 2 e 3.

$$Variação do Atraso = \frac{[AtrasoAtual - AtrasoAnterior]}{(AtrasoAnterior)} \quad (2)$$

$$Variação da Vazão = \frac{[VazãoAtual - VazãoAnterior]}{(VazãoAnterior)} \quad (3)$$

Tabela 5.1. Resultados Obtidos para Topologia com 6 Nós

Ponto de Medição	Taxa da Aplicação [kbps]	Vazão Média [kbps]	Atraso Médio [ms]	Varição da Vazão	Varição do Atraso
1	100	100,04	66,61	0,00%	0,00%
2	105	105,04	66,61	4,76%	0,01%
3	110	110,04	66,62	4,54%	0,01%
4	115	115,04	66,75	4,35%	0,19%
5	120	120,04	66,95	4,16%	0,31%
6	125	125,04	67,06	4,00%	0,16%
7	130	130,04	67,90	3,84%	1,24%
8	135	135,04	69,43	3,70%	2,20%
9	140	139,99	71,36	3,54%	2,70%
10	145	144,87	74,08	3,37%	3,68%
11	150	149,93	77,23	3,37%	4,08%
12	155	154,86	79,72	3,18%	3,12%
13	160	159,73	80,68	3,05%	1,19%
14	165	163,17	97,21	2,11%	17,00%
15	170	165,49	123,18	1,40%	21,09%
16	175	167,62	145,98	1,27%	15,61%
17	180	168,98	167,60	0,80%	12,90%
18	185	170,91	179,14	1,13%	6,44%
19	190	171,55	195,32	0,38%	8,29%
20	195	171,71	198,94	0,09%	1,82%
21	200	171,63	200,34	-0,05%	0,70%

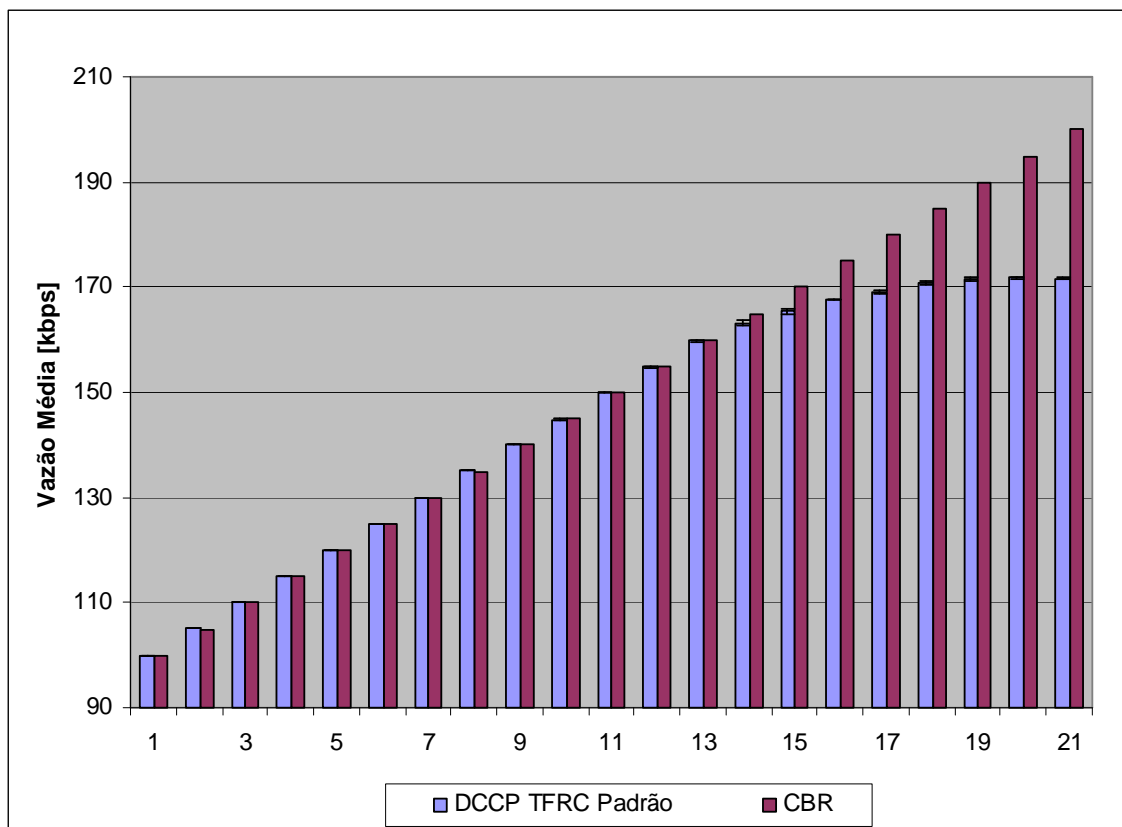


Figura 5.2. Comparação entre a Vazão Média DCCP TFRC Padrão Medida e a Taxa de Envio de Dados da Aplicação para topologia com 6 nós

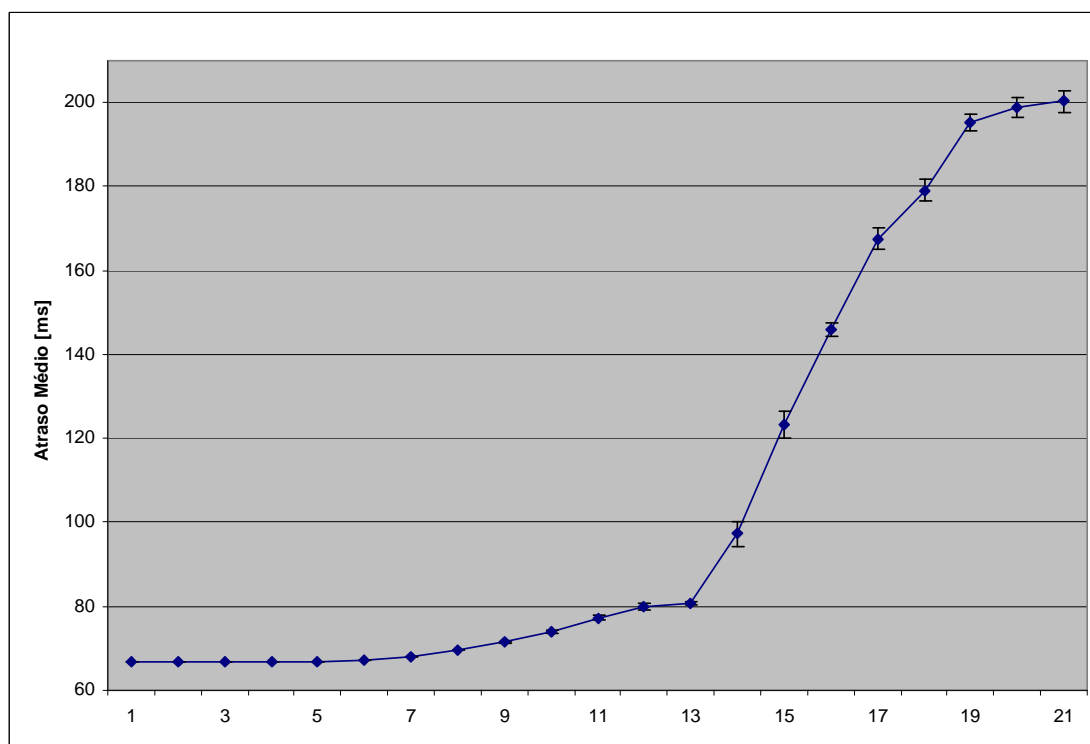


Figura 5.3. Atraso Médio DCCP TFRC Padrão para topologia com 6 nós

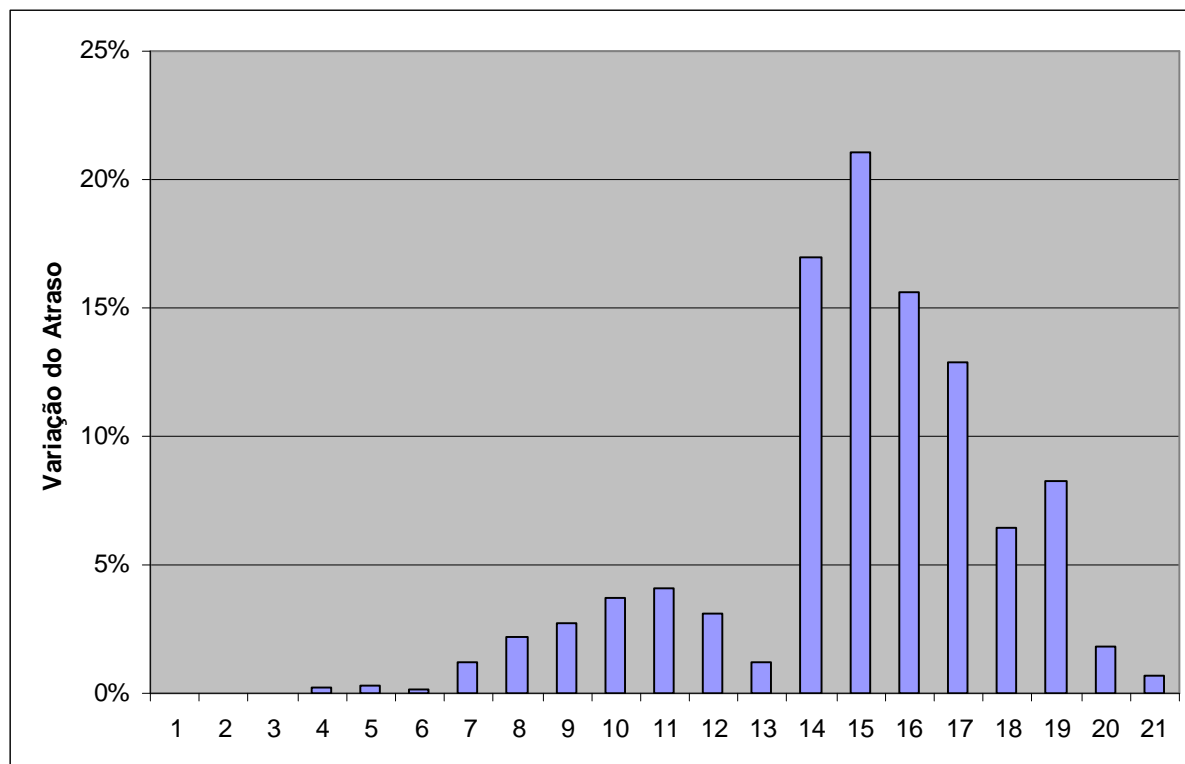


Figura 5.4. Variação do Atraso Médio DCCP TFRC Padrão para topologia com 6 nós

Analisando as Figuras 5.2, 5.3 e 5.4, verifica-se que até o ponto de medição de número 13, o aumento da taxa de envio de dados da aplicação CBR corresponde a um aumento de mesma intensidade no valor da vazão e a um pequeno incremento no atraso médio, ambos medidos na camada de aplicação.

Nos pontos de medição 14 ao 21, verificamos que o aumento da taxa de envio de dados da aplicação CBR não é acompanhado por um aumento de mesma intensidade na vazão DCCP. Por outro lado, o atraso médio aumenta bastante a cada novo aumento da taxa de envio de dados da aplicação CBR. A diferença entre a taxa de envio de dados da aplicação CBR e a vazão medida na camada de aplicação indica a quantidade de pacotes perdidos na rede. O aumento elevado do atraso médio indica o início da saturação da camada MAC do padrão IEEE 802.11.

Também foram realizadas simulações para uma topologia em cadeia com 7 nós (6 saltos) conforme apresentado na Tabela 5.2 e Figuras 5.5, 5.6 e 5.7.

Tabela 5.2. Resultados Obtidos para Topologia com 7 Nós

Ponto de Medição	Taxa da Aplicação [kbps]	Vazão Média [kbps]	Atraso Médio [ms]	Varição da Vazão	Varição do Atraso
1	100	100,04	80,22	0,00%	0,00%
2	105	105,04	80,33	4,76%	0,13%
3	110	110,04	81,65	4,54%	1,62%
4	115	115,04	83,03	4,35%	1,67%
5	120	120,04	84,62	4,16%	1,88%
6	125	125,03	86,96	3,99%	2,68%
7	130	129,96	88,89	3,80%	2,18%
8	135	134,97	90,04	3,71%	1,28%
9	140	140,01	89,92	3,59%	-0,13%
10	145	144,78	91,18	3,30%	1,38%
11	150	149,71	92,52	3,29%	1,45%
12	155	154,46	95,23	3,08%	2,84%
13	160	158,65	103,31	2,64%	7,82%
14	165	162,12	119,83	2,13%	13,79%
15	170	161,99	161,10	-0,08%	25,62%
16	175	163,61	182,13	0,99%	11,55%
17	180	164,01	203,18	0,24%	10,36%
18	185	163,00	208,86	-0,62%	2,72%
19	190	163,78	210,96	0,48%	1,00%
20	195	163,90	210,84	0,07%	-0,06%
21	200	163,75	210,39	-0,09%	-0,21%

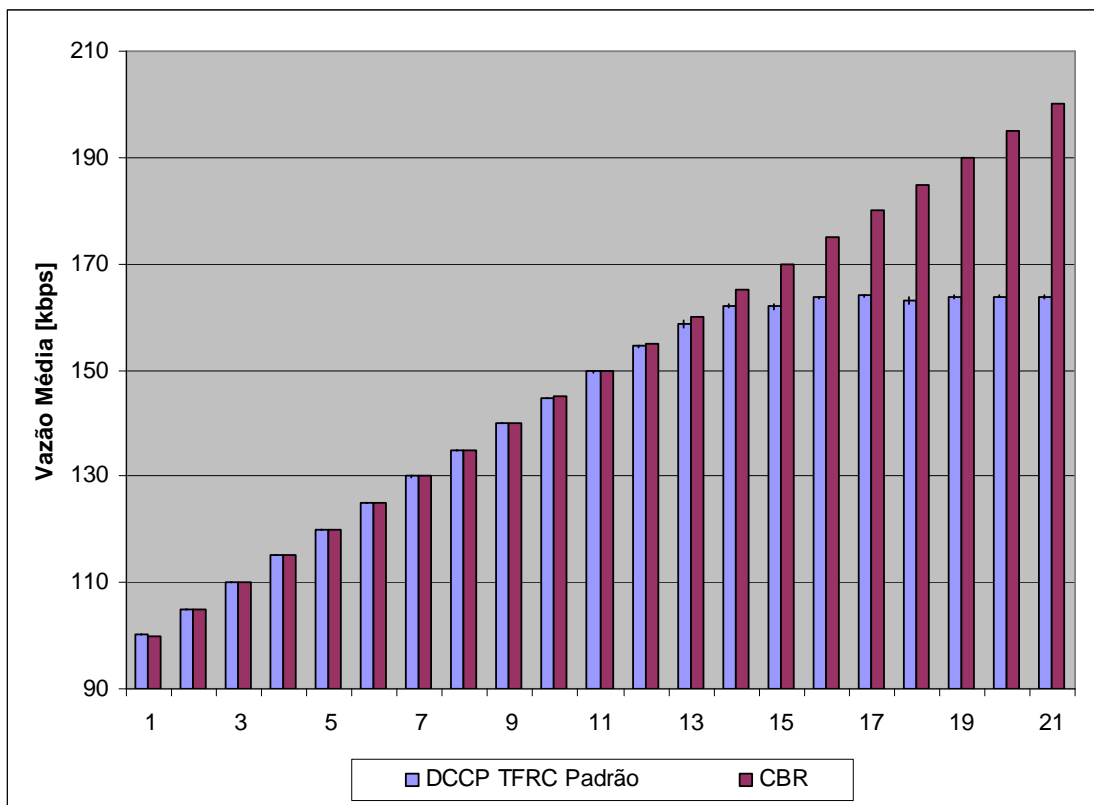


Figura 5.5. Comparação entre a Vazão Média DCCP TFRC Padrão Medida e a Taxa de Envio de Dados da Aplicação para topologia com 7 nós

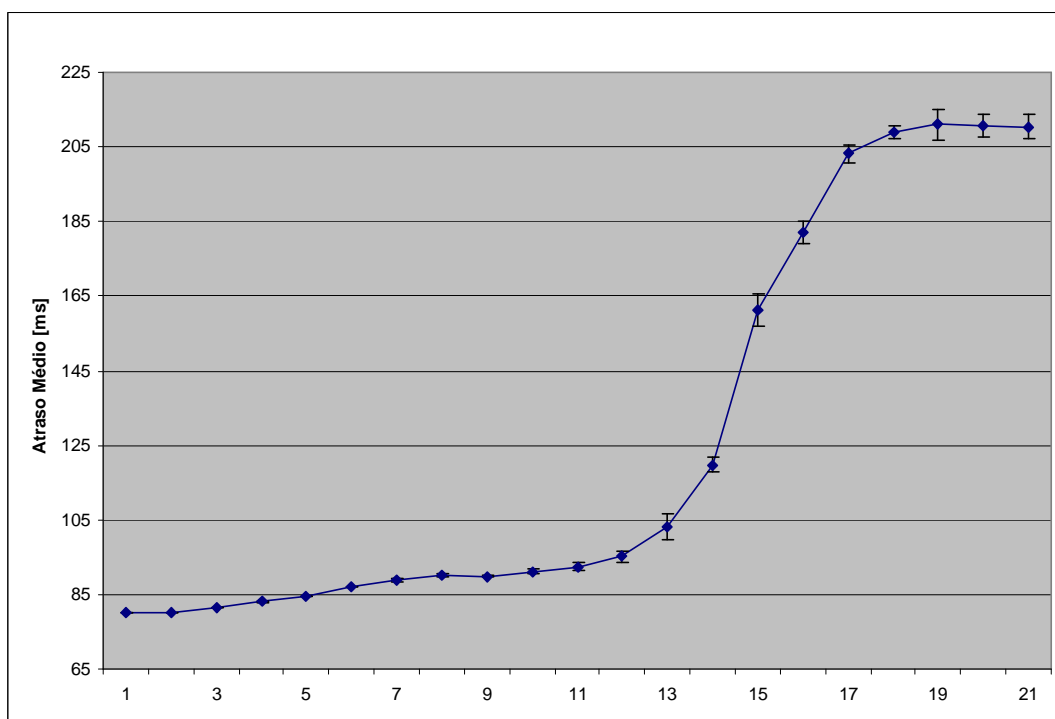


Figura 5.6. Atraso Médio DCCP TFRC Padrão para topologia com 7 nós

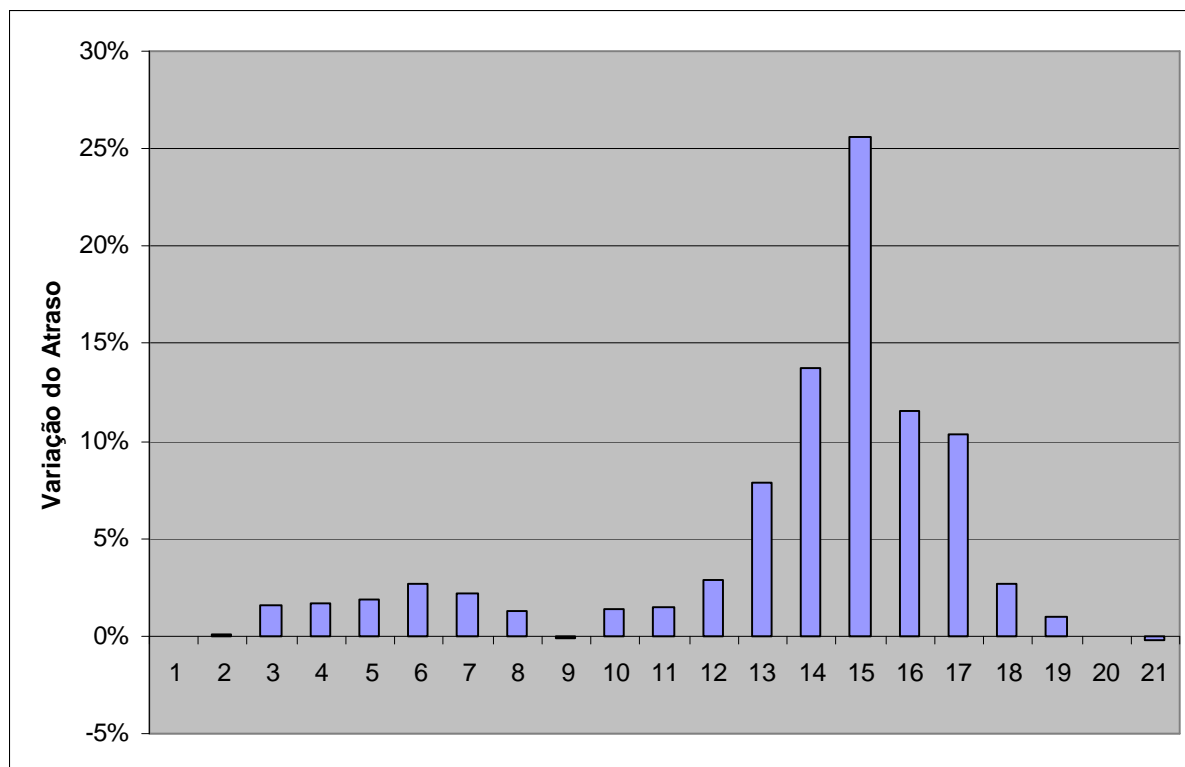


Figura 5.7. Variação do Atraso Médio DCCP TFRC Padrão para topologia com 7 nós

Os resultados apresentados na topologia com 7 nós foram semelhantes aos da topologia com 6 nós e serviram para modelar os efeitos da camada MAC do padrão IEEE 802.11 em redes mesh sem fio com múltiplos saltos.

Baseados nas informações dos testes realizados, observamos que o CCID 3 ignora a fase de saturação da camada MAC. Desta forma, ele é incapaz de detectá-la, o que impede que ele ajuste sua taxa de transmissão em um ponto ótimo. A camada MAC então sofre múltiplas retransmissões de quadros, o que faz aumentar o atraso médio na rede.

5.2 Adaptação do Controle do congestionamento CCID 3

Durante os testes analisados na Seção 5.1, foi verificado que o início da saturação da rede em malha sem fio, mais precisamente o início do colapso da camada MAC, é marcado por um aumento considerável no atraso médio, e conseqüentemente, no RTT. Desta forma, uma maneira adequada de identificar o início da fase de saturação da camada MAC no nó transmissor DCCP é comparar o RTT atual com o último valor de RTT calculado. Se a

diferença percentual entre eles for maior do que um determinado limite, então é detectado o início da fase de saturação.

O transmissor CCID 3 calcula o RTT baseado nos *acks* recebidos do receptor. Desta maneira, o RTT é calculado uma vez a cada intervalo de tempo igual ao RTT. Durante os testes realizados, foi verificado que o valor limite que identifica o início da fase de saturação é um aumento de 5% em relação ao atraso médio anterior, conforme pode ser observado nas Tabelas 5.1 e 5.2.

Como o protocolo não armazena valores anteriores de RTT, o mecanismo de adaptação proposto neste trabalho, chamado de M-TFRC (Mesh-TFRC), armazena os valores dos últimos RTTs medidos. O cálculo da vazão é realizado pela fórmula padrão do TFRC. Após o cálculo padrão da vazão, é realizada uma verificação para saber se há saturação na camada MAC considerando os últimos valores de RTT. Se, entre dois RTTs consecutivos, houver um aumento superior a 5% no valor do RTT, então a vazão é reduzida de forma a evitar a saturação da camada MAC.

Nesta proposta, a ideia é detectar a saturação na camada MAC logo no início e reduzir a vazão de maneira que a taxa máxima de transmissão permitida se situe em uma faixa estreita próxima ao ponto ótimo no limiar da saturação da camada MAC. Após diversos testes, que estão documentados no Anexo 1, foi verificado que os melhores resultados ocorreram quando a taxa máxima de transmissão é reduzida a 85% da taxa calculada após a detecção da fase de saturação.

Essa redução em 15% da vazão permite que o controle de congestionamento permaneça a maior parte do tempo próximo ao valor ótimo da vazão e, como a fase de saturação é detectada e evitada logo no início, não há riscos de que a taxa de transmissão seja reduzida à metade caso nenhum *ack* seja recebido em um intervalo de tempo igual ao parâmetro *nofeedback timer* [Floyd et al, 2006a] devido ao crescente número de perdas de pacotes na rede.

O algoritmo a seguir demonstra o princípio de funcionamento do M-TFRC, proposto nesta dissertação. Ao receber um *ack*, o transmissor M-TFRC atualiza a taxa de transmissão (X) de acordo com o algoritmo.

```

if (p > 0) { // Se não está na fase de partida lenta

// Calcula a vazão usando a equação (1)
x_calc = calcX(s,RTT,p);

/* Conforme [Floyd et al, 2006a], garante que a vazão não será
* maior que duas vezes a taxa de recepção
* informada pelo receptor no último ack
* recebido e nem menor que a taxa mínima
* permitida.
*/
X = max(min(X_calc, 2*X_recv), s/t_mbi);

// Verifica se existe saturação na camada MAC.
// K1 = 1.05
if ((RTT/RTT_anterior) > K1) {

/* Reduz a vazão em 15% de forma a limitar
* os efeitos da saturação da camada MAC.
* K2 = 0.85
*/
X = K2*X;
}
} else {
// Implementa a partida lenta conforme [Floyd et al, 2006a]
if (t_now - tld >= R)
X = max(min(2*X, 2*X_recv),s/R);
tld = t_now;
}
// Armazena o valor do RTT para comparar com o próximo
RTT_anterior = RTT;
}

```

As Tabelas 5.3 e 5.4 apresentam os resultados obtidos para os mesmos cenários das Tabelas 5.1 e 5.2 anteriores, utilizando o mecanismo de controle de congestionamento M-TFRC proposto.

Tabela 5.3. M-TFRC – Resultados Obtidos para Topologia com 6 Nós

Ponto de Medição	Taxa da Aplicação [kbps]	Vazão Média [kbps]	Atraso Médio [ms]	Varição da Vazão	Varição do Atraso
1	100	100,04	66,61	0,00%	0,00%
2	105	105,04	66,61	5,00%	0,00%
3	110	110,04	66,62	4,76%	0,01%
4	115	115,04	66,74	4,54%	0,19%
5	120	120,04	66,95	4,35%	0,31%
6	125	125,04	67,08	4,16%	0,20%
7	130	130,04	67,93	4,00%	1,27%
8	135	135,04	69,44	3,85%	2,21%
9	140	140,04	71,14	3,70%	2,45%
10	145	145,02	73,91	3,56%	3,89%
11	150	149,98	76,40	3,42%	3,38%
12	155	154,90	79,03	3,28%	3,44%
13	160	159,77	81,36	3,14%	2,94%
14	165	164,09	90,25	2,70%	10,93%
15	170	166,38	117,81	1,40%	30,55%
16	175	168,37	141,60	1,20%	20,19%
17	180	169,06	154,15	0,41%	8,86%
18	185	168,84	163,02	-0,13%	5,76%
19	190	168,00	156,56	-0,49%	-3,96%
20	195	167,07	162,25	-0,56%	3,63%
21	200	168,59	157,23	0,91%	-3,09%

Tabela 5.4. M-TFRC – Resultados Obtidos para Topologia com 7 Nós

Ponto de Medição	Taxa da Aplicação [kbps]	Vazão Média [kbps]	Atraso Médio [ms]	Variação da Vazão	Variação do Atraso
1	100	100,04	80,23	0,00%	0,00%
2	105	105,04	80,32	4,76%	0,12%
3	110	110,04	81,64	4,54%	1,61%
4	115	115,04	83,11	4,35%	1,77%
5	120	120,03	84,58	4,16%	1,73%
6	125	125,04	86,91	4,00%	2,68%
7	130	130,04	88,46	3,85%	1,76%
8	135	135,03	89,44	3,70%	1,10%
9	140	139,99	89,86	3,55%	0,46%
10	145	144,88	90,70	3,37%	0,93%
11	150	149,83	91,31	3,30%	0,66%
12	155	154,80	92,17	3,21%	0,93%
13	160	159,27	98,39	2,81%	6,32%
14	165	162,36	118,61	1,90%	17,05%
15	170	162,86	153,19	0,30%	22,58%
16	175	160,73	174,83	-1,32%	12,38%
17	180	161,15	170,83	0,26%	-2,34%
18	185	163,11	171,81	1,20%	0,57%
19	190	160,95	176,99	-1,34%	2,93%
20	195	160,39	182,91	-0,35%	3,24%
21	200	160,09	173,48	-0,19%	-5,44%

Baseado nos valores das Tabelas 5.3 e 5.4, verifica-se que o algoritmo de controle de congestionamento M-TFRC, embora não tenha convergido para o ponto ótimo no limiar da fase de saturação da camada MAC, apresentou uma redução de aproximadamente 22% do atraso médio em uma topologia em cadeia com 6 nós (5 saltos) e 18% em uma topologia em cadeia com 7 nós (6 saltos). O fato do protocolo não convergir próximo ao ponto ótimo deve-se a degradação muito rápida do RTT logo no início da fase de saturação devido aos mecanismos da camada MAC como retransmissão de quadros e *backoff* exponencial. Como o protocolo não evita o início da saturação da camada MAC, esses fatores elevam o atraso médio acima do ponto ótimo no limiar da saturação.

Também foram realizados testes adicionais variando o número de nós entre 4 e 15 (3 a 14 saltos) e verificados a vazão e o atraso médio para uma aplicação CBR enviando dados a uma taxa igual a 1000 Kbps, bem acima do que a rede é capaz de suportar, de forma a comparar o desempenho dos dois mecanismos em situação de saturação da camada MAC. Para cada número de saltos, foram realizadas 10 simulações e utilizado um intervalo de

confiança de 95%. Os resultados tanto para a versão padrão do DCCP TFRC quanto para o mecanismo DCCP M-TFRC proposto são apresentados na Tabela 5.5 e Figuras 5.8 e 5.9.

Tabela 5.5. M-TFRC – Resultados Obtidos para Topologia com N saltos

Saltos	CBR[kbps]	DCCP M-TFRC		DCCP TFRC Padrão	
		Vazão Média [kbps]	Atraso Médio [ms]	Vazão Média [kbps]	Atraso Médio [ms]
3	1000	273,37	719,46	276,03	1051,25
4	1000	186,91	130,19	191,40	163,42
5	1000	167,74	170,92	171,43	234,53
6	1000	158,54	183,14	162,19	253,37
7	1000	155,06	208,96	158,98	257,98
8	1000	149,36	231,56	155,21	275,41
9	1000	149,97	221,13	155,37	270,47
10	1000	149,13	244,63	154,88	274,56
11	1000	146,28	267,04	154,43	311,60
12	1000	147,90	274,12	155,15	300,33
13	1000	146,88	278,08	154,35	336,85
14	1000	148,63	273,22	154,77	342,54

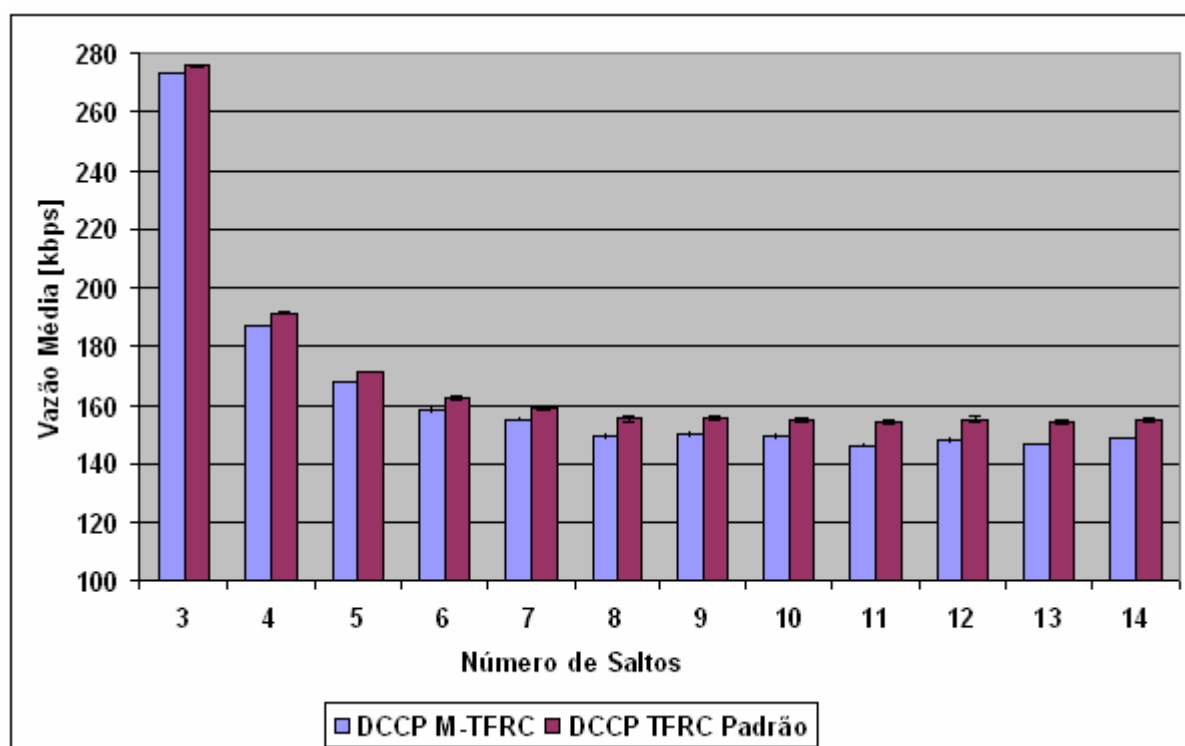


Figura 5.8. Comparação Vazão Média DCCP M-TFRC e DCCP TFRC Padrão

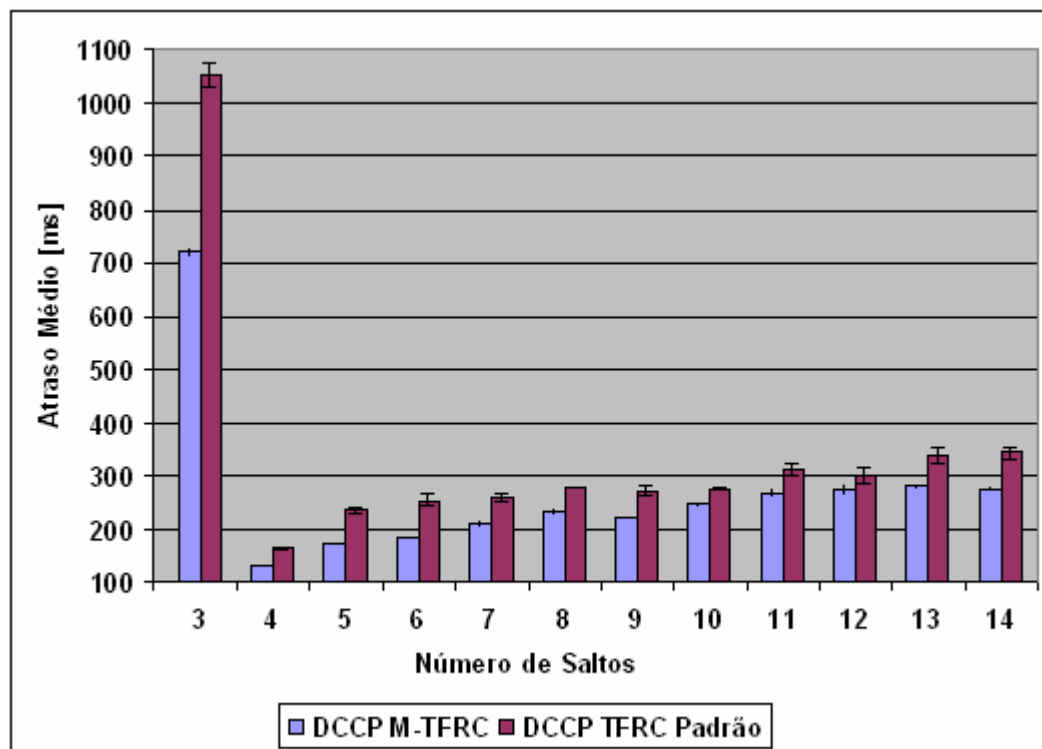


Figura 5.9. Comparação Atraso Médio DCCP M-TFRC e DCCP TFRC Padrão

De acordo com os resultados obtidos na Tabela 5.5 e Figuras 5.8 e 5.9, a proposta de adaptação do DCCP M-TFRC apresentada obteve melhora significativa no atraso médio para todos os testes realizados, sem diminuição significativa da vazão média da rede. O atraso médio foi reduzido em média 20% em relação ao mecanismo TFRC padrão, sendo o melhor caso a topologia com 3 saltos, com uma redução de 32% no atraso médio.

5.3 Testes com Fluxos Concorrentes

Nesta seção são apresentados os resultados dos testes realizados para cenários com fluxos concorrentes usando o controle de congestionamento DCCP M-TFRC e comparando-os com os resultados dos testes realizados para o DCCP TFRC padrão, com o objetivo de verificar qual dos dois protocolos é mais amigável ao TCP (*TCP-Friendly*). Para cada cenário, foram realizadas 10 simulações.

O primeiro cenário consiste de uma topologia com 3 nós (2 saltos) e duas aplicações CBR com taxa de 250 kbps sobre DCCP. A comunicação ocorre entre os nós situados nas extremidades da topologia. O fluxo 1 é iniciado em $t = 10s$ e o fluxo 2 em $t = 60s$ e ambos são finalizados em $t=300s$.

As Figuras 5.10 a 5.13 apresentam os resultados dos testes que permitem a comparação do comportamento da rede (vazão e atraso) quando são utilizados dois fluxos DCCP TFRC padrão concorrentes com os resultados obtidos quando utilizados dois fluxos DCCP M-TFRC concorrentes.

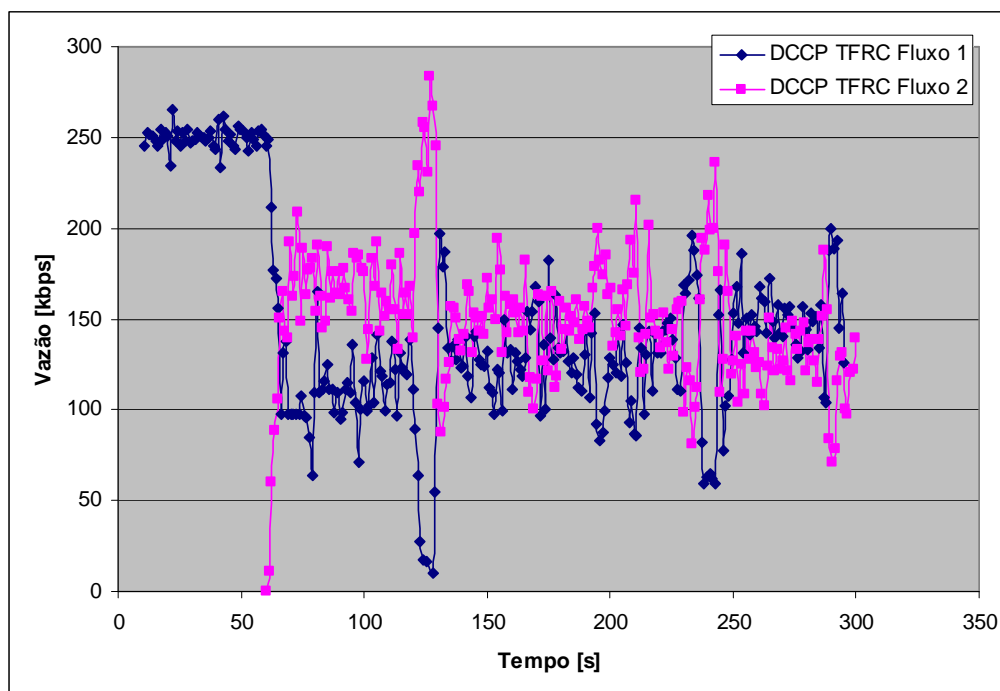


Figura 5.10. Vazão Instantânea de Fluxos DCCP TFRC Padrão Concorrentes

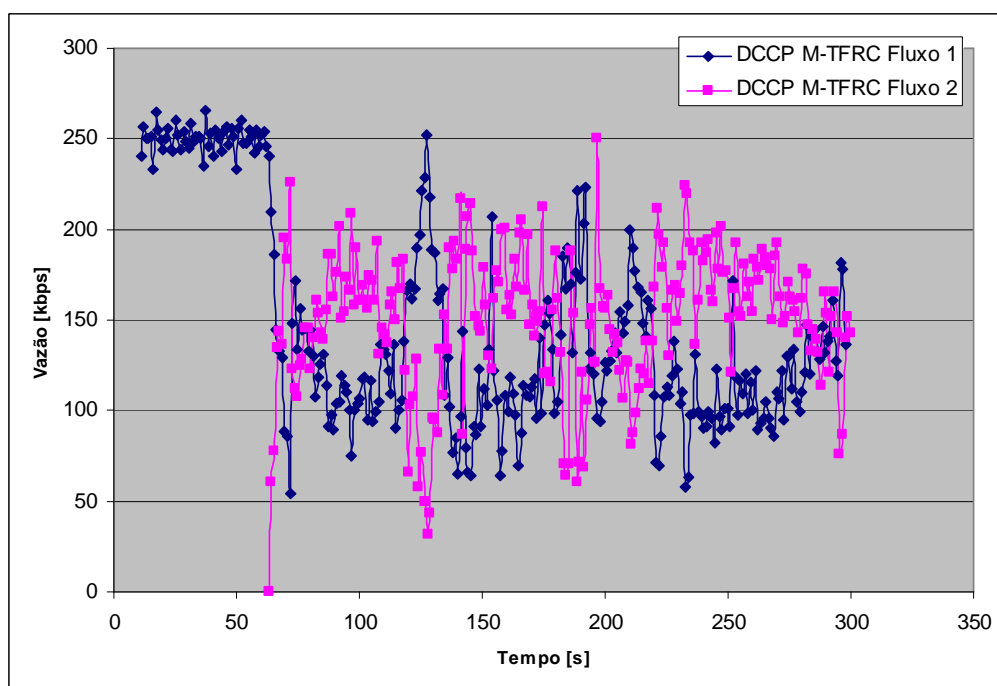


Figura 5.11. Vazão Instantânea de Fluxos DCCP M-TFRC Concorrentes

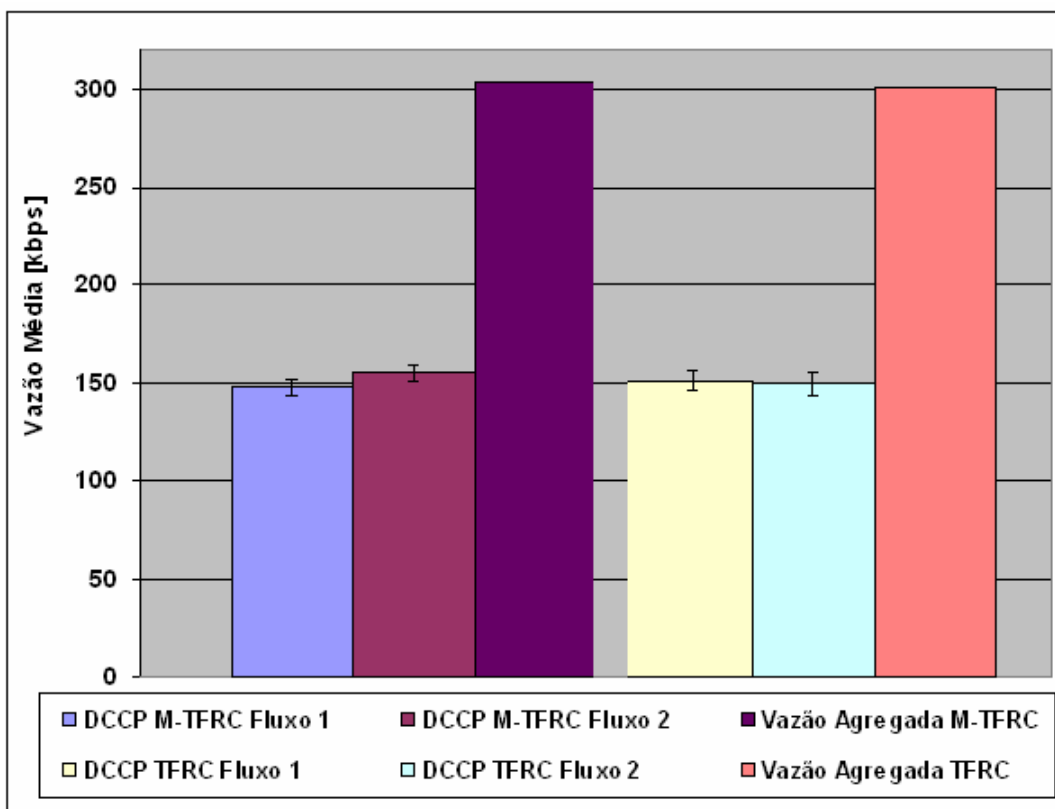


Figura 5.12. Vazão Média de Fluxos DCCP M-TFRC Concorrentes e Fluxos TFRC Padrão Concorrentes

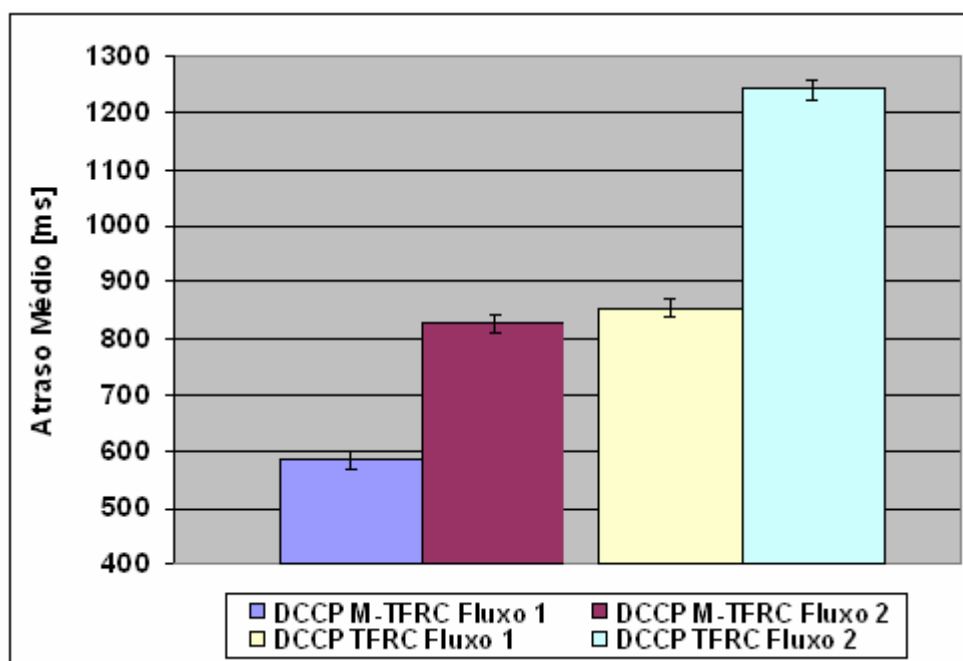


Figura 5.13. Atraso Médio de Fluxos DCCP M-TFRC Concorrentes e Fluxos TFRC Padrão Concorrentes

A vazão instantânea foi calculada de forma similar à utilizada para o cálculo da vazão média, porém ao invés de ser calculada como o volume de dados dividido pelo tempo total da sessão, é calculada como o volume de dados a cada segundo, durante toda a sessão de dados.

A Figura 5.12 mostra que não há diferença significativa na vazão média entre dois fluxos DCCP M-TFRC comparada com dois fluxos DCCP TFRC. Já a Figura 5.13 apresenta a diminuição considerável (em média 33 %) do atraso médio dos fluxos DCCP M-TFRC em comparação aos fluxos DCCP TFRC. Esta diminuição significativa no atraso médio utilizando o DCCP M-TFRC comprova a diminuição dos efeitos da saturação da camada MAC do padrão IEEE 802.11 também para o cenário com fluxos concorrentes.

Como segundo cenário, foram realizados testes comparando os resultados de fluxos DCCP M-TFRC e TCP concorrentes com os resultados de fluxos DCCP TFRC e TCP concorrentes.

Inicialmente, foram realizados testes em uma topologia com 2 nós (1 salto) com uma aplicação CBR com taxa de 500 Kbps sobre DCCP competindo com uma aplicação FTP sobre TCP. O fluxo TCP é iniciado em $t = 10s$ e o DCCP em $t = 60s$ e ambos são finalizados em $t=300s$. As Figuras 5.14 a 5.17 apresentam os resultados dos testes para este cenário, considerando os valores médios de 10 simulações.

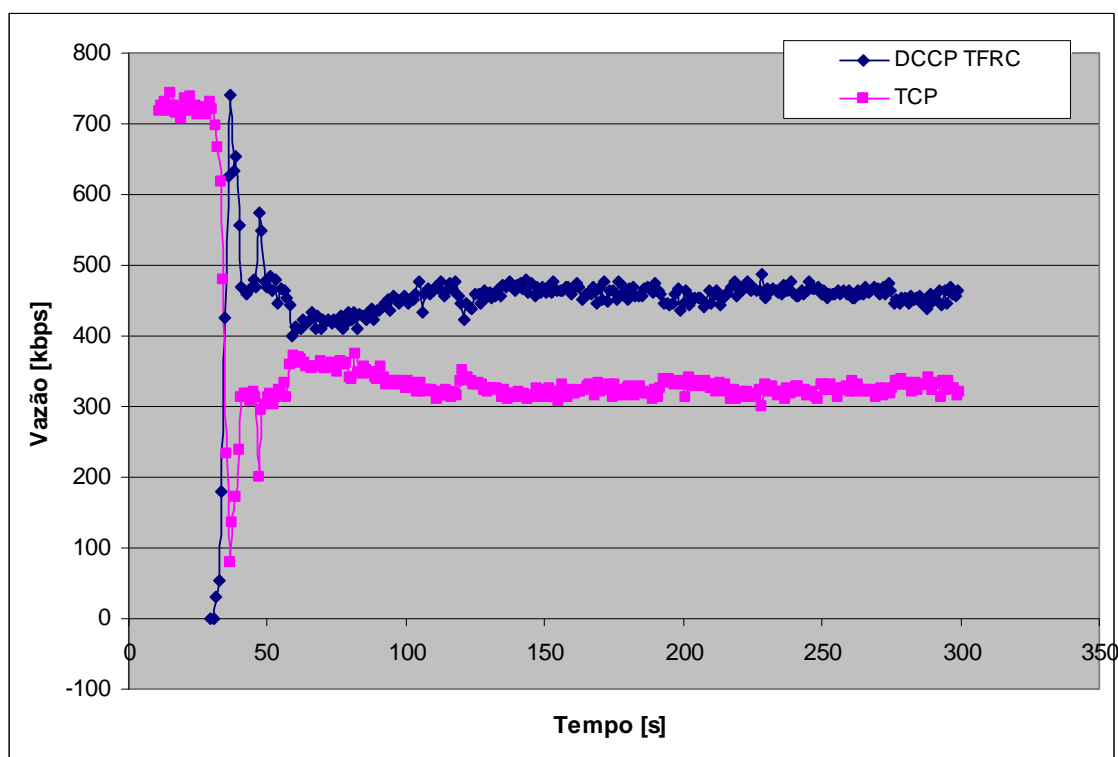


Figura 5.14. Vazão Instantânea de Fluxos TCP e DCCP TFRC Padrão Concorrentes (1 Salto)

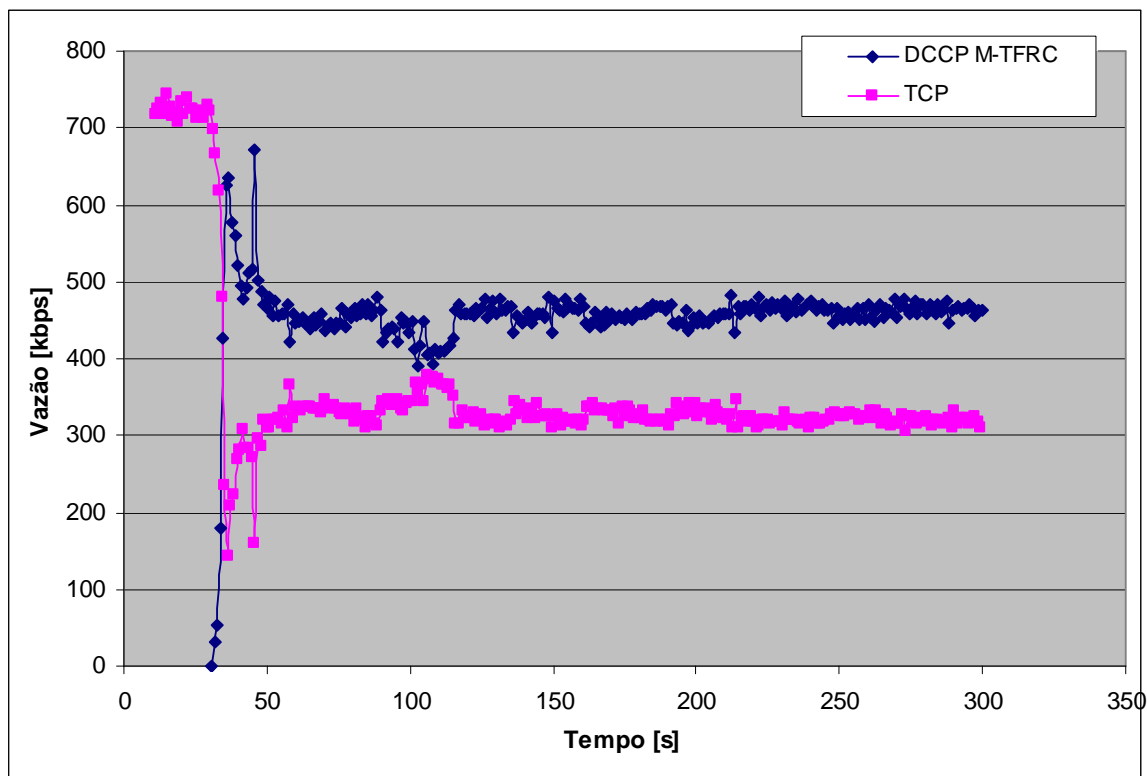


Figura 5.15. Vazão Instantânea de Fluxos TCP e DCCP M-TFRC Concorrentes (1 Salto)

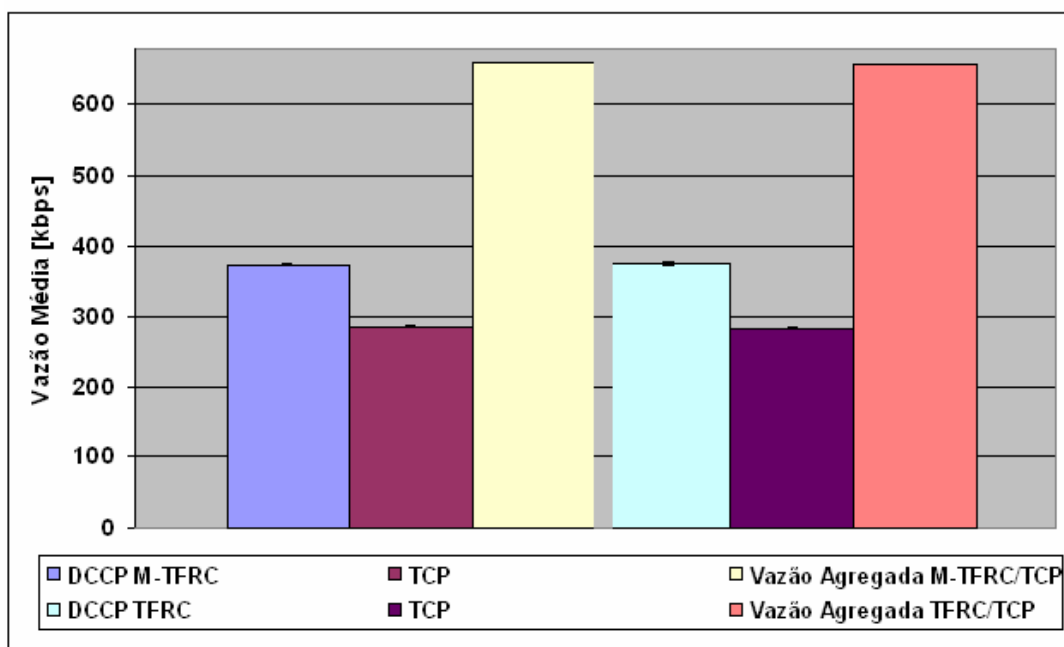


Figura 5.16. Vazão Média de Fluxos DCCP M-TFRC e TCP Concorrentes e Fluxos DCCP TFRC Padrão e TCP Concorrentes (1 Salto)

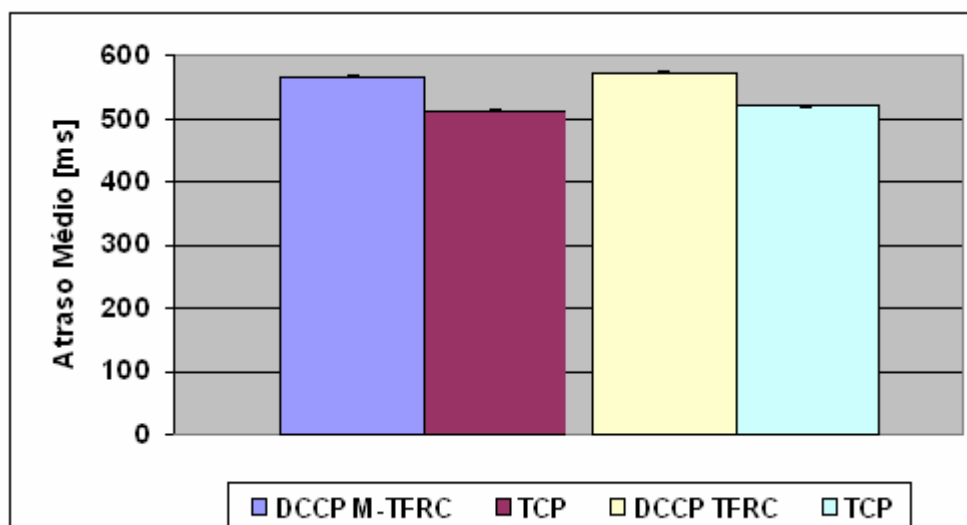


Figura 5.17. Atraso Médio de Fluxos DCCP M-TFRC e TCP Concorrentes e Fluxos DCCP TFRC Padrão e TCP Concorrentes (1 Salto)

Depois, foram realizados testes com uma topologia com 3 nós (2 saltos). As Figuras 5.18 a 5.21 apresentam os resultados do teste para uma aplicação CBR com taxa de 250 Kbps sobre DCCP com fluxo iniciado no primeiro nó (N0) e finalizado no último nó (N2) competindo com uma aplicação FTP sobre TCP com fluxo iniciado no segundo nó (N1) e finalizado no último nó (N2). O fluxo TCP é iniciado em $t = 10s$ e o DCCP em $t = 40s$ e ambos são finalizados em $t=300s$. Foram realizadas 10 simulações para este cenário.

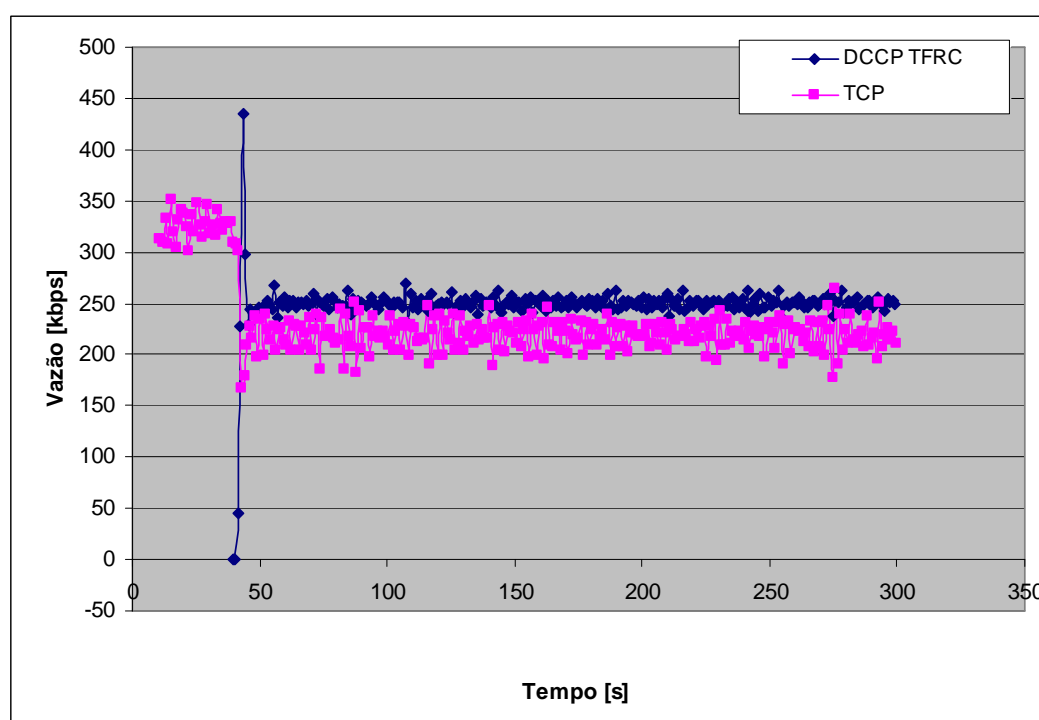


Figura 5.18. Vazão Instantânea de Fluxos TCP e DCCP TFRC Padrão Concorrentes (2 Saltos)

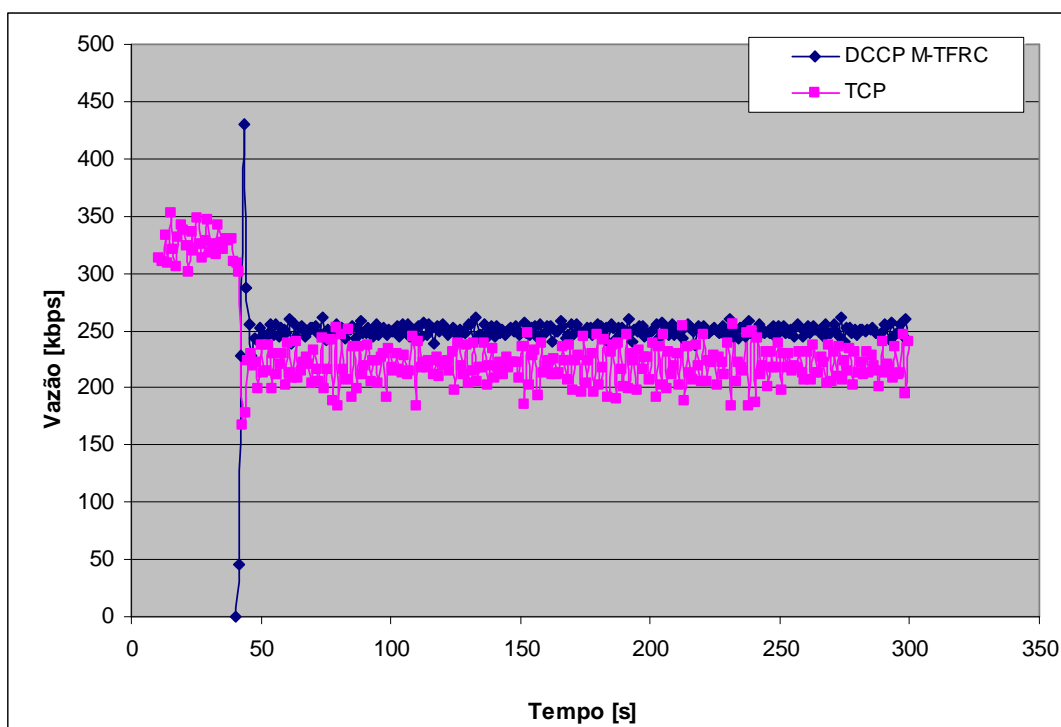


Figura 5.19. Vazão Instantânea Fluxos TCP e DCCP M-TFRC Concorrentes (2 Saltos)

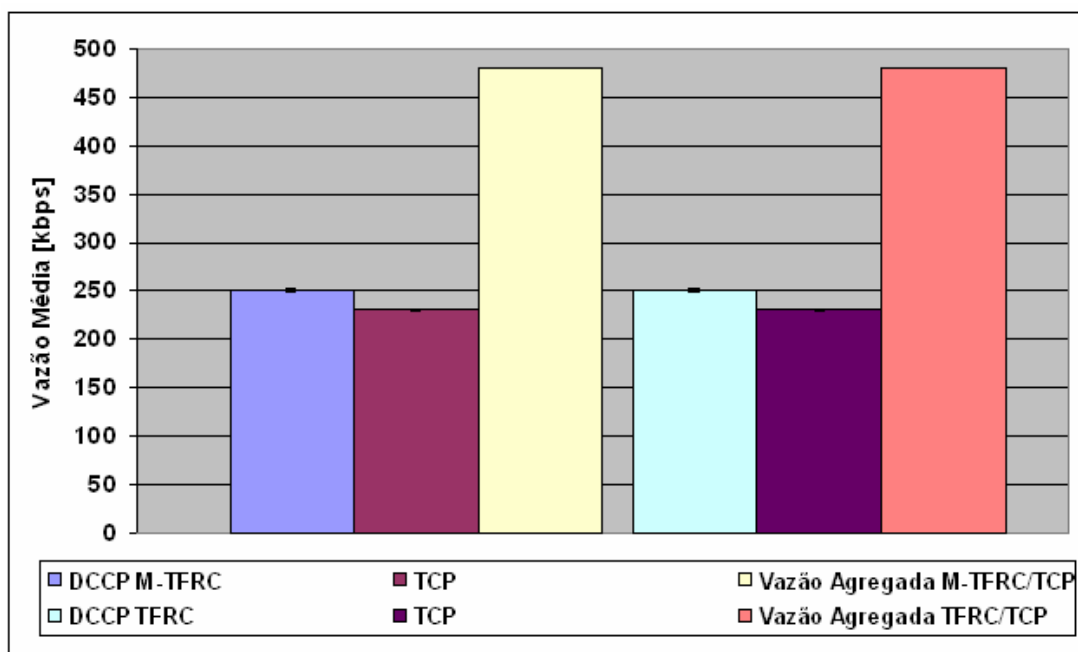


Figura 5.20. Vazão Média Fluxos DCCP M-TFRC e TCP Concorrentes e Fluxos DCCP TFRC Padrão e TCP Concorrentes (2 Saltos)

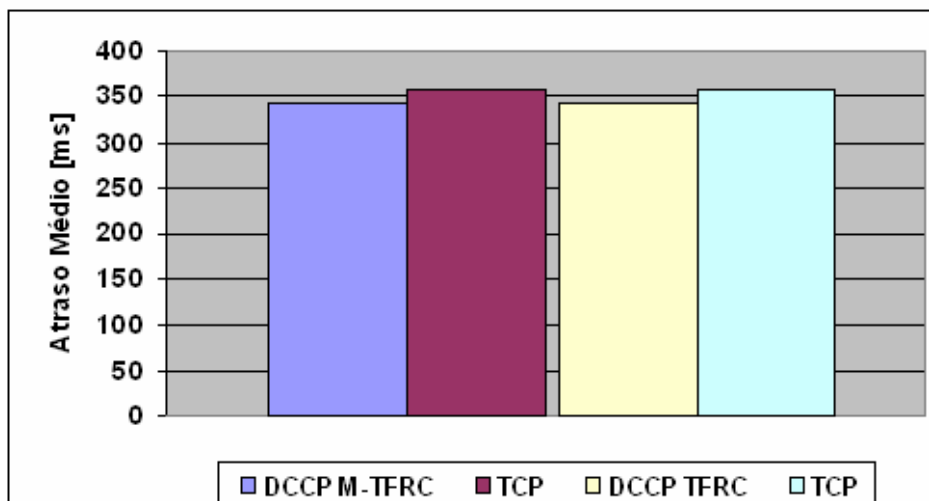


Figura 5.21. Atraso Médio Fluxos DCCP M-TFRC e TCP Concorrentes e Fluxos DCCP TFRC Padrão e TCP Concorrentes (2 Saltos)

Por último, foram realizados testes com uma topologia com 4 nós (3 saltos) para uma aplicação CBR com taxa de 250 Kbps sobre DCCP com fluxo iniciado no primeiro nó (N0) e finalizado no último nó (N3) competindo com uma aplicação FTP sobre TCP com fluxo iniciado no segundo nó (N1) e finalizado no último nó (N3). O fluxo TCP é iniciado em $t = 10s$ e o DCCP em $t = 40s$ e ambos são finalizados em $t=300s$. As Figuras 5.22 a 5.25 ilustram os resultados dos testes, considerando os valores médios de 10 simulações.

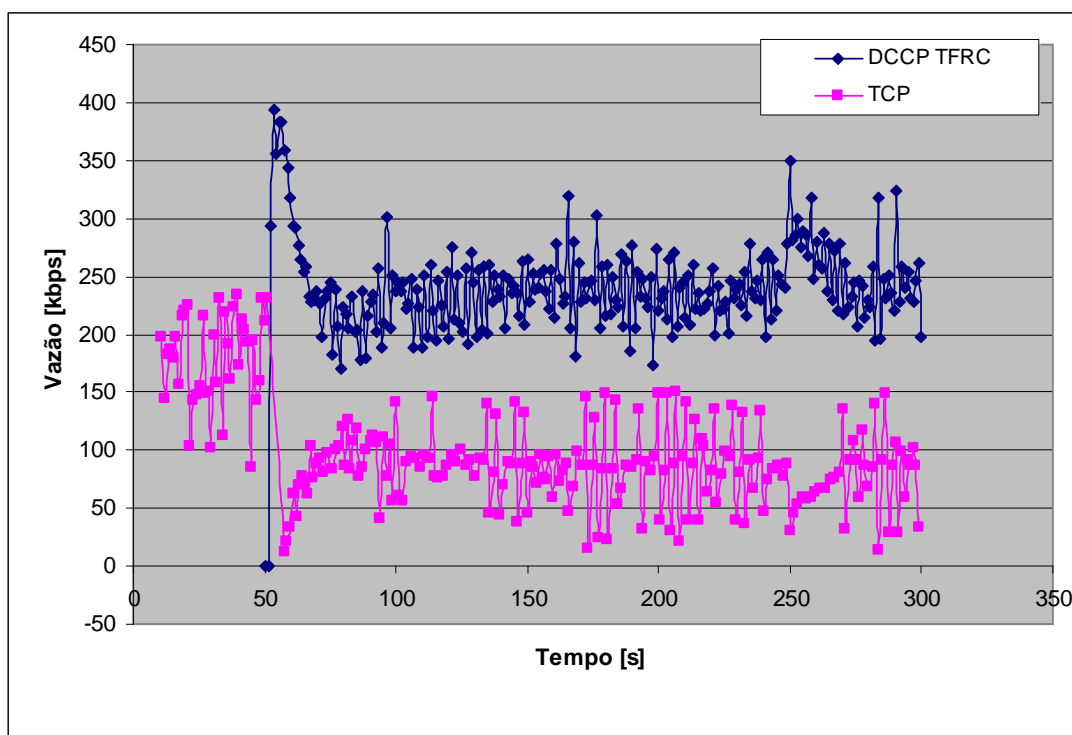


Figura 5.22. Vazão Instantânea de Fluxos TCP e DCCP TFRC Padrão Concorrentes (3 Saltos)

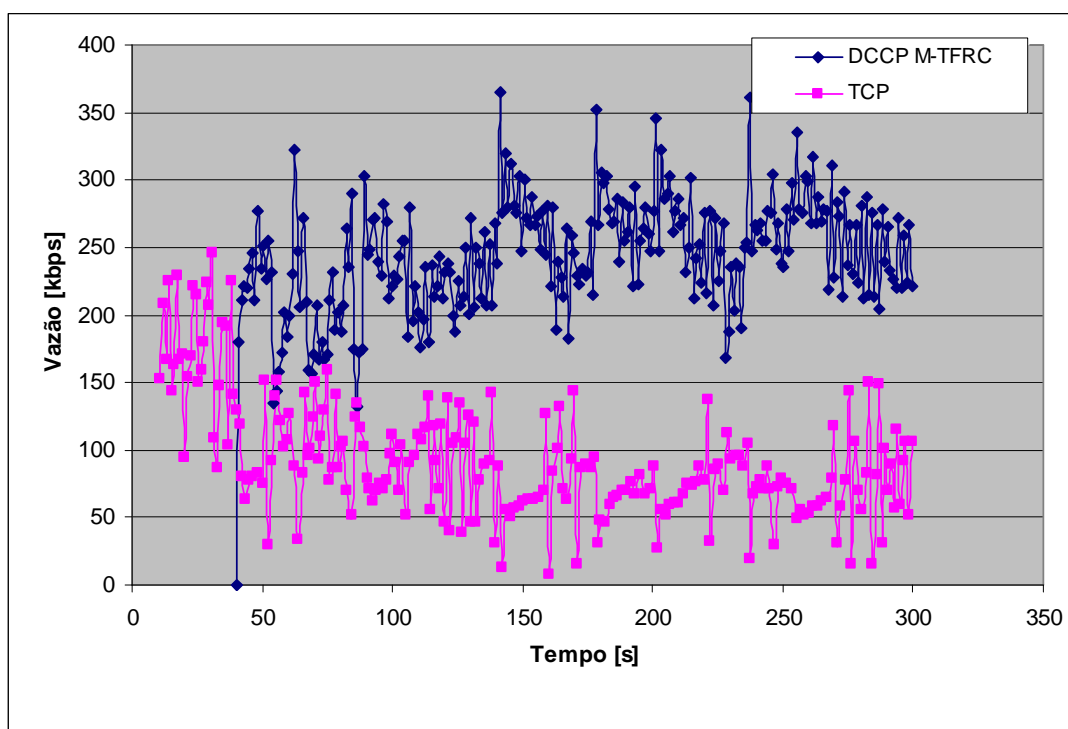


Figura 5.23. Vazão Instantânea de Fluxos TCP e DCCP M-TFRC Concorrentes (3 Saltos)

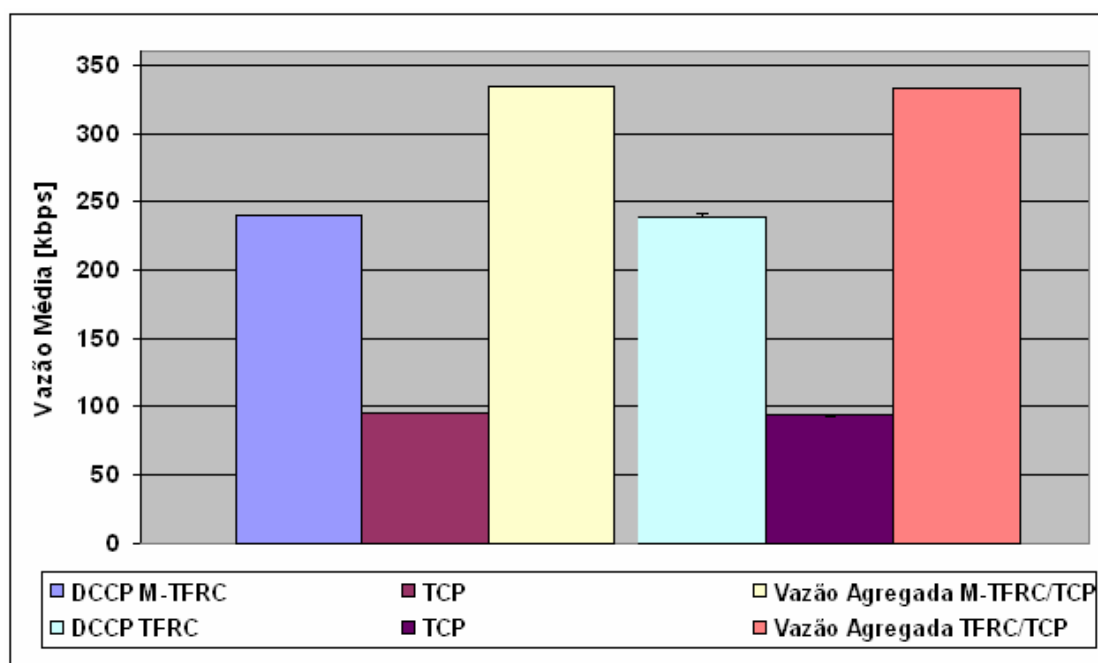


Figura 5.24. Vazão Média de Fluxos DCCP M-TFRC e TCP Concorrentes e Fluxos DCCP TFRC Padrão e TCP Concorrentes (3 Saltos)

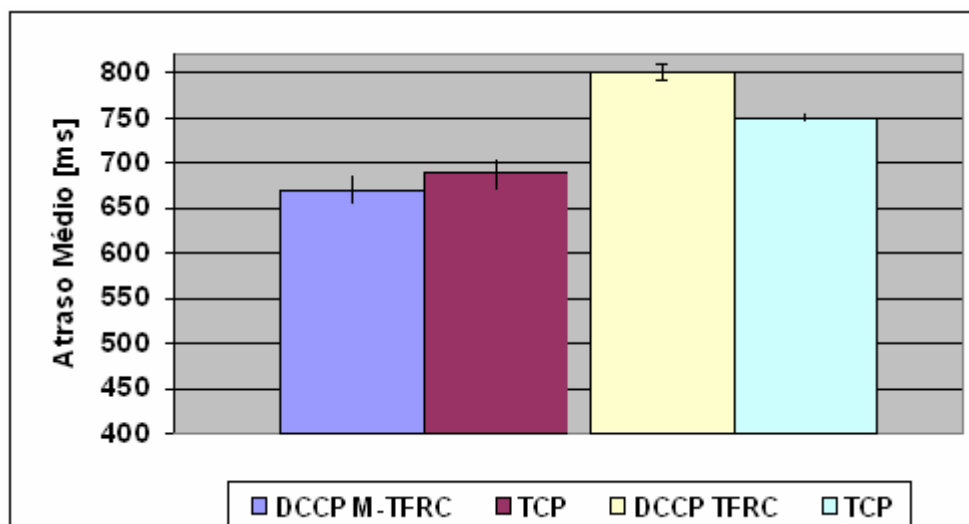


Figura 5.25. Atraso Médio de Fluxos DCCP M-TFRC e TCP Concorrentes e Fluxos DCCP TFRC Padrão e TCP Concorrentes (3 Saltos)

Dos testes apresentados, verificamos que o M-TFRC é tão *TCP-Friendly* quanto o TFRC. Quanto a vazão média, o desempenho do mecanismo M-TFRC ao competir com fluxos TCP foi semelhante ao do mecanismo TFRC para todos os cenários. Já para o atraso médio, na topologia com 3 saltos, o desempenho do M-TFRC foi 16% melhor que o do TFRC, enquanto o atraso médio do TCP ao competir com o M-TFRC foi 10% menor do que ao competir com o TFRC. Para os testes com 1 salto e 2 saltos, o desempenho do M-TFRC, quanto ao atraso médio, foi semelhante ao do mecanismo TFRC.

5.4 Testes com Topologia Mesh com Múltiplos Saltos Densa

Nesta seção, são apresentados os resultados realizados para testes utilizando como cenário uma rede mesh com múltiplos saltos simulando uma topologia diferente da topologia em cadeia utilizada nos testes anteriores, havendo vários caminhos possíveis entre a origem e o destino, destacados em vermelho na Figura 5.26.

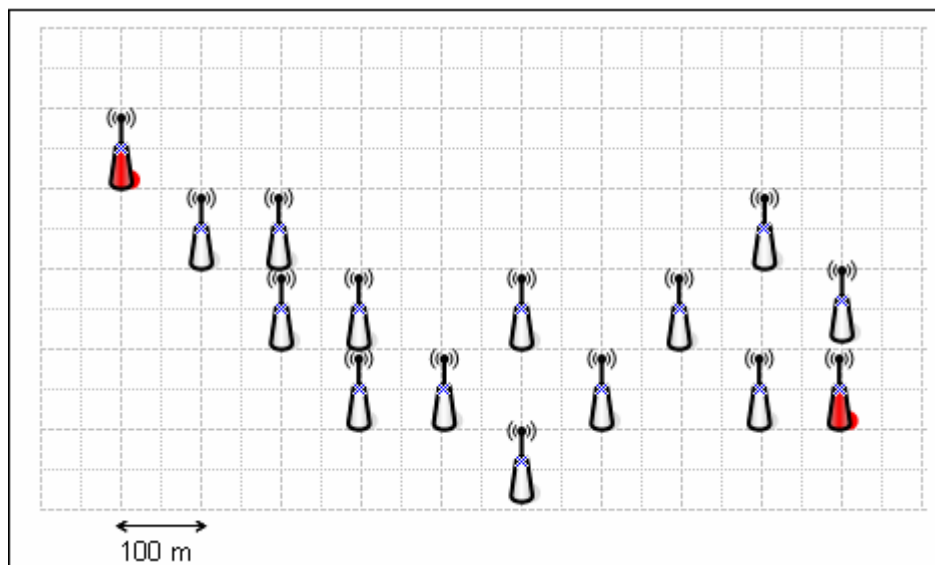


Figura 5.26. Topologia Mesh com Múltiplos Saltos Densa

Neste novo cenário, é utilizado um fluxo CBR com 1000 Kbps entre os dois nós marcados com a cor vermelha na Figura 5.26. Devido a maior complexidade da topologia, neste cenário foram realizadas 30 simulações. As Figuras 5.27 e 5.28 ilustram a comparação do resultados de vazão e atraso médios obtidos com uso do DCCP TFRC Padrão e do DCCP M-TFRC.

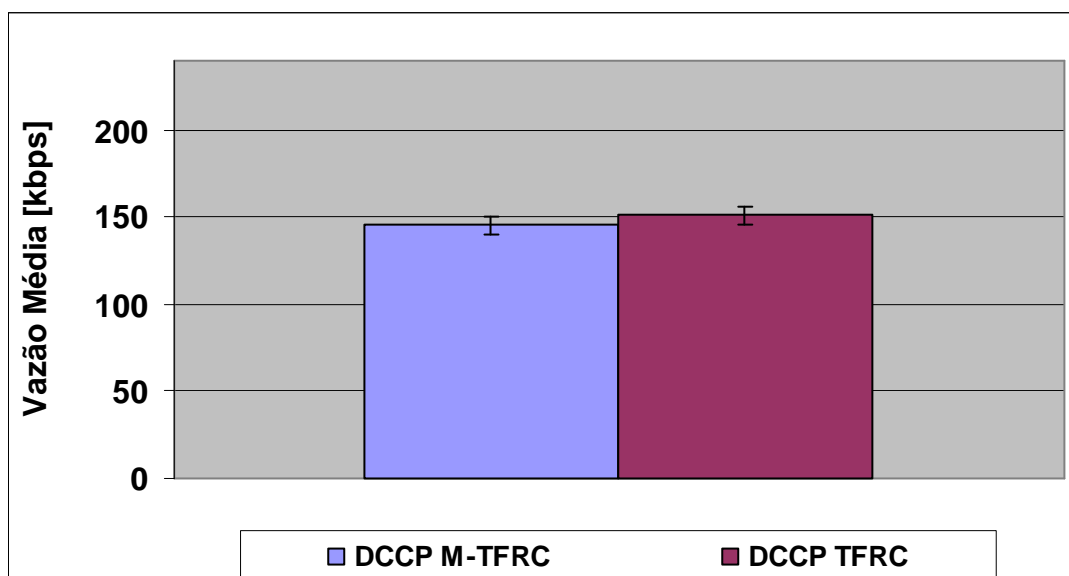


Figura 5.27. Comparação de Vazão Média DCCP M-TFRC e DCCP TFRC

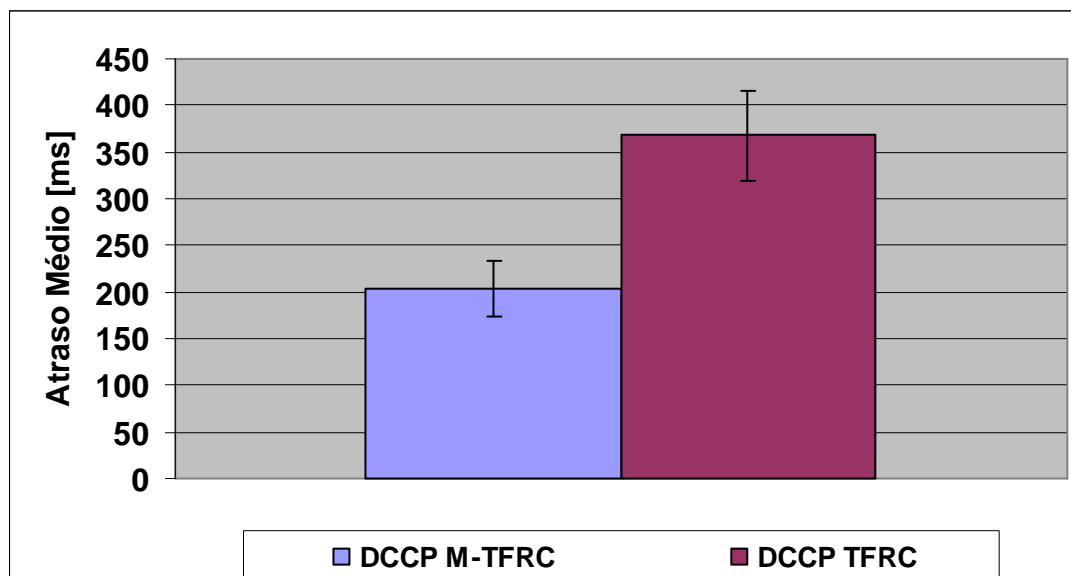


Figura 5.28. Comparação de Atraso Médio DCCP M-TFRC e DCCP TFRC

Novamente os resultados dos testes foram mais favoráveis ao DCCP M-TFRC, pois com uma redução de apenas 4% da vazão média, o atraso médio diminuiu 44% em relação ao DCCP TFRC.

5.5 Conclusão do Capítulo

O mecanismo M-TFRC proposto neste trabalho conseguiu melhorar o desempenho do protocolo DCCP em redes sem fio com múltiplos saltos, com uma proposta de implementação mais simples do que os mecanismos utilizados nos trabalhos relacionados comentados no Capítulo 4, utilizando um mecanismo de adaptação implementado puramente pela camada de transporte. O M-TFRC não possui a complexidade de implementação de comunicação entre a camada de enlace e a camada de transporte e nem a necessidade de conhecimento sobre a topologia da rede e parâmetros de nível físico para adaptar a taxa usada pelo transmissor DCCP.

Uma observação importante é que o M-TFRC possui os mesmos resultados do TFRC em condições de não saturação da camada MAC do padrão IEEE 802.11. Em condições de saturação, o M-TFRC reduziu o atraso médio em todos os cenários, sem comprometer a vazão média agregada dos fluxos, conforme testes apresentados neste capítulo.

6 Conclusões

Este trabalho propôs uma adaptação do algoritmo de controle de congestionamento TFRC (CCID 3) do protocolo de transporte DCCP, para o uso em redes sem fio com múltiplos saltos, denominada M-TFRC. A adaptação proposta reduz os efeitos da saturação da camada MAC do padrão IEEE 802.11, melhorando a eficiência do protocolo DCCP em redes mesh, reduzindo o atraso médio e, conseqüentemente, evitando a degradação no desempenho das aplicações multimídia. A adaptação consiste em detectar a fase de saturação da camada MAC através da medição e comparação dos valores de RTTs subsequentes. Toda vez que um RTT for maior que o anterior acima de um limite percentual, a taxa de transmissão é reduzida de forma a não sobrecarregar a camada MAC.

A proposta de adaptação M-TFRC foi implementada e avaliada usando a ferramenta de simulação NS-2. Os testes apresentados confirmam que a adaptação reduz o RTT em até 44%, enquanto fornece uma vazão muito semelhante à oferecida pela implementação padrão do mecanismo de controle de congestionamento TFRC.

Além disso, durante os testes foi verificado que o mecanismo M-TFRC proposto é mais justo ao competir com outros fluxos M-TFRC do que o mecanismo TFRC ao competir com outros fluxos TFRC. Além disso, testes comprovaram que o M-TFRC é tão *TCP-Friendly* quanto o TFRC. Fluxos TCP ao concorrerem com fluxos M-TFRC obtiveram melhora de até 10% no atraso médio que ao concorrerem com fluxos TFRC.

As principais contribuições desta dissertação são o estudo sobre o protocolo de transporte DCCP, o estudo sobre os mecanismos de controle de congestionamento na camada de transporte, a revisão da literatura sobre propostas para a melhoria de desempenho de controle de congestionamento em redes em malha sem fio com múltiplos saltos, a proposta de uma nova técnica para este controle e sua avaliação através de simulações.

Como trabalho futuro, planeja-se a implementação em Linux da adaptação do controle de congestionamento M-TFRC como um novo CCID do protocolo de transporte DCCP e a realização de testes de desempenho em redes em malha sem fio reais.

Referências Bibliográficas

[Abelém et al, 2007] Antônio Jorge Gomes Abelém, Célio Vinicius Neves Albuquerque, Débora Christina Muchaluat Saade, Elisangela Santana Aguiar, Jairo Lino Duarte, José Eduardo Mendonça da Fonseca e Luiz Claudio Schara Magalhães, “Redes Mesh: Mobilidade, Qualidade de Serviço e Comunicação em Grupo”, SBRC 2007, Belém, PA.

[Allman et al, 1999] M. Allman, V. Paxson, and W. Stevens, “TCP Congestion Control”, RFC 2581, IETF, Abril, 1999.

[Allman et al, 2002] M. Allman, S. Floyd, and C. Partridge, “Increasing TCP’s Initial Window”, RFC 3390, IETF, Outubro, 2002.

[Balakrishnan et al, 1996] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, “A comparison of mechanisms for improving TCP performance over wireless links,” ACM SIGCOMM’96, Agosto, 1996.

[Blanton et al, 2003] E. Blanton, M. Allman, K. Fall and L. Wang, “A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm for TCP”, RFC 3517, IETF, Abril, 2003.

[Braden, 1989] R. Braden, “Requirements for Internet Hosts - Communication Layers”, RFC 1122, IETF, Outubro, 1989.

[Conceição, 2006] Da Conceição, A. F, “Voz e Vídeo Sobre Redes Sem Fio IEEE 802.11”, Dissertação de Doutorado apresentada ao Instituto de Matemática e Estatística da Universidade de São Paulo, Maio, 2006.

[Floyd e Kohler, 2006] S. Floyd and E. Kohler, “Profile for Datagram Congestion Control Protocol (DCCP) - Congestion Control ID 2: TCP-like Congestion Control”, RFC 4341, IETF, Março, 2006.

[Floyd et al, 2006a] S. Floyd, E. Kohler and J. Padhye, “Profile for Datagram Congestion Control Protocol (DCCP) - Congestion Control ID 3: TCP-Friendly Rate Control (TFRC)”, RFC 4342, IETF, Março, 2006.

[Floyd et al, 2006b] Sally Floyd, Mark Handley, and Eddie Kohler, “Problem Statement for the datagram congestion control protocol (DCCP)”, RFC 4336, IETF, Março, 2006.

[Fonseca, 2004] Nelson Luís Saldanha da Fonseca. Investigação da Efetividade de Explicit Congestion Notification, 10 2004. <http://www.gta.ufrj.br/quaresma/atividades/13.htm>. Acesso em Maio de 2009.

[Fu et al, 2003] Zhenghua Fu, Petros Zerfos, Haiyun Luo, Songwu Lu, Lixia Zhang and Mario Gerla, “The Impact of Multihop Wireless Channel on TCP Throughput and Loss,” INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, vol.3 pp. 1744-1753, Abril, 2003.

[Gerla et al, 1999a] M. Gerla, R. Bagrodia, L. Zhang, K. Tang, and L. Wang, "TCP over wireless multihop protocols: simulation and experiments," IEEE ICC'99, Vancouver, Canadá, Junho, 1999.

[Gerla et al, 1999b] M. Gerla, K. Tang, and R. Bagrodia, "TCP performance in wireless multihop networks," IEEE WMCSA'99, New Orleans, LA, Fevereiro. 1999.

[Handley et al, 2003] M. Handley, S. Floyd, J. Padhye and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 3448, IETF, 2003.

[IEEE, 2007] IEEE The Institute of Electrical and Electronics Engineers. IEEE 802.11-2007, IEEE Standard for Information Technology, Março, 2007.

[ISO, 1994] ISO International Organization for Standardization. ISO/IEC standard 7498-1:1994, Novembro, 1994.

[Jacobson et al, 1992] V. Jacobson, R. Braden and D. Borman, "TCP Extensions for High Performance", RFC 1323, IETF, Maio, 1992.

[Kohler et al, 2006] E. Kohler, M. Handley and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, IETF, Março, 2006.

[Kurose e Ross, 2004] Kurose, James F and Ross, Keith W., "Computer Networking - A Top-Down Approach", Editora: ADDISON WESLEY, 3ª Edição, 2004.

[Li et al, 2004] Mingzhe Li, Choong-Soo Lee, Emmanuel Agu, Mark Claypool, and Robert Kinicki, "Performance Enhancement of TFRC in Wireless Ad Hoc Networks", 10th International Conference on Distributed Multimedia Systems (DMS), San Francisco, California, USA Setembro 8 - 10, 2004.

[Mascolo et al, 2001] Saverio Mascolo, Claudio Casetti, Mario Gerla, M. Y. Sanadidi and Ren Wang, "TCP westwood: Bandwidth estimation for enhanced transport over wireless links," Proceedings of the 7th annual international conference on Mobile computing and networking table of contents, Roma, Itália, Pags: 287 – 297, 2001.

[Mathis et al, 1996] VM. Mathis, J. Mahdavi, S. Floyd and A. Romanow, "TCP Selective Acknowledgment Options", RFC 2018, IETF, Outubro, 1996.

[Muchaluat-Saade et al, 2008] Muchaluat-Saade, D.C., Gomes, A.G., Carrano, R.C., Magalhães, L.C.S., Albuquerque, C.V.N., Tarouco, L.R. Multihop MAC: Desvendando o Padrão 802.11s, Capítulo 1 do Livro de Minicursos do Simpósio Brasileiro de Redes de Computadores, SBRC 2008, Rio de Janeiro, 2008.

[Navaratnam et al, 2006] P. Navaratnam, N. Akhtar and R. Tafazolli, "On the Performance of DCCP in Wireless Mesh Networks", MobiWAC'06, Malaga, Spain, Outubro, 2, 2006.

[NS2, 2009] NS2, <http://www.isi.edu/nsnam/ns/>, acesso em setembro de 2009.

[Paxson e Allman, 2000] V. Paxson and M. Allman, "Computing TCP's Retransmission Timer", RFC 2988, IETF, Novembro, 2000.

[Peng e Liew, 2003] Cheng Peng Fu and Soung C. Liew, "TCP Veno: TCP Enhancement for Transmission Over Wireless Access Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 21, NO. 2, Fevereiro, 2003.

[Perkins et al, 2003] Perkins, C. E.; Belding-Royer, E. M.; Das, S. R.; "Ad Hoc On-Demand Distance Vector Routing", RFC 3561, IETF, Julho, 2003.

[Postel, 1980] J. Postel, "User Datagram Protocol", RFC 768, IETF, Agosto, 1980.

[Ramakrishnan et al, 2001] K. Ramakrishnan, S. Floyd, and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, IETF, Setembro, 2001.

[Ray et al, 2003] Saikat Ray, Jeffrey Carruthers, and David Starobinski, "RTS/CTS-induced Congestion in Ad-Hoc Wireless LANs," in *Proceedings of IEEE WCNC*, Março, 2003, pp. 1516–1521.

[RFC793, 1981] "Transmission Control Protocol", RFC 793. IETF, Setembro. 1981.

[Ribeiro e Muchaluat-Saade, 2009] Ribeiro, Cesar H.P., Muchaluat-Saade, D.C., "M-TFRC: Adaptação de Mecanismo de Congestionamento do Protocolo de Transporte DCCP para Uso em Redes Mesh sem Fio", *I2TS*, Dezembro, 2009.

[Spring et al, 2003] N. Spring, D. Wetherall and D. Ely, "Robust Explicit Congestion Notification (ECN) Signaling with Nonces", RFC 3540, IETF, Junho, 2003.

[Ware et al, 2001] C. Ware, T. Wysocki, and J.F. Chicharo, "On the Hidden Terminal Jamming Problem in IEEE 802.11 Mobile Ad Hoc Networks," in *Proceedings of IEEE ICC*, 2001.

[Zhou et al, 2007] Bin Zhou, Cheng Peng Fu, Chiew Tong Lau and Chuan Heng Foh, "An Enhancement of TFRC over Wireless Networks", *IEEE WCNC 2007*, Março, 2007.

Anexo 1 – Testes para Ajuste dos Parâmetros K1 e K2

Neste anexo, são apresentados os resultados das simulações realizadas alterando-se os valores dos parâmetros K1 (aumento do RTT) e K2 (diminuição da taxa), do algoritmo do M-TFRC, para ajuste da taxa de transmissão X:

```
if ((RTT/RTT_anterior) > K1) {
    X = K2*X;
}
```

Foram realizadas 10 simulações para as topologias em cadeia com 6 e 7 nós. Foi utilizada uma aplicação CBR com taxa de 200 kbps. Foram testadas as seguintes combinações de valores dos parâmetros K1 e K2:

- M-TFRC: K1 = 1,05 e K2 = 0,85, opção escolhida;
- K1 = 1,10 e K2 = 0,90;
- K1 = 1,20 e K2 = 0,85;
- K1 = 1,10 e K2 = 0,85;

A opção que gerou melhores resultados considerando a diminuição do atraso médio foi a que reduziu a taxa de transmissão em 15%, após um aumento de 5% no RTT, por isso, estes valores foram adotados no mecanismo M-TFRC.

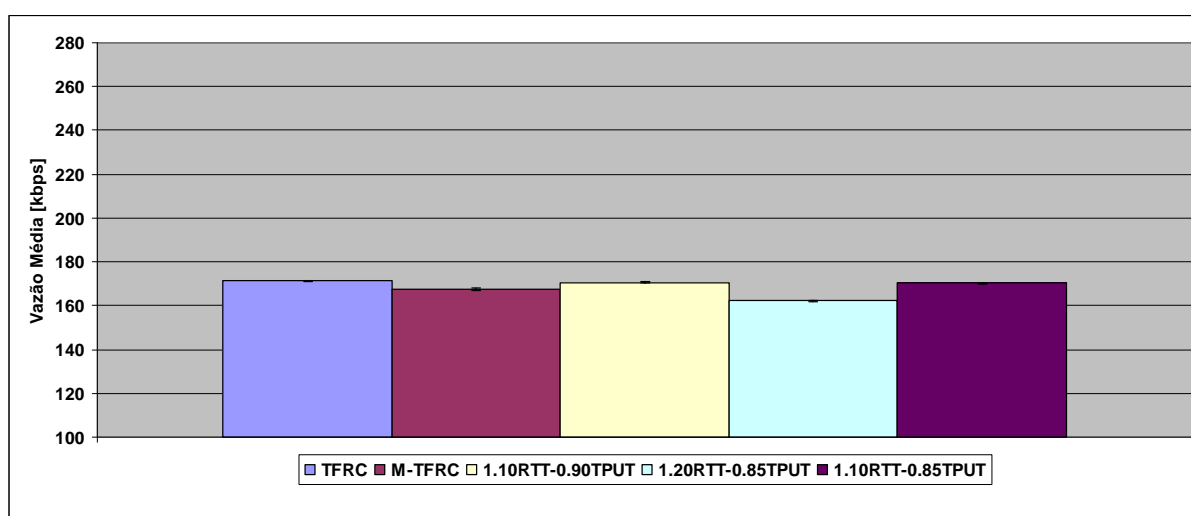


Figura A1.1. Comparação de Vazão Média Topologia com 6 Nós

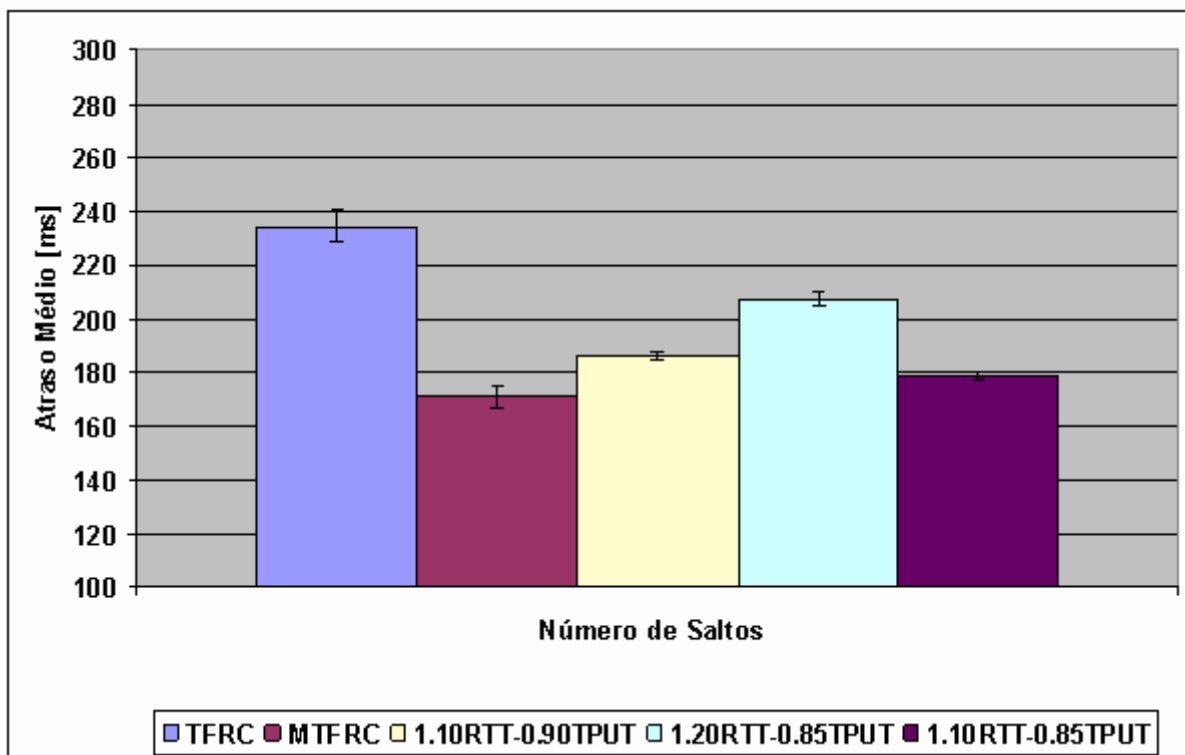


Figura A1.2. Comparação de Atraso Médio Topologia com 6 Nós

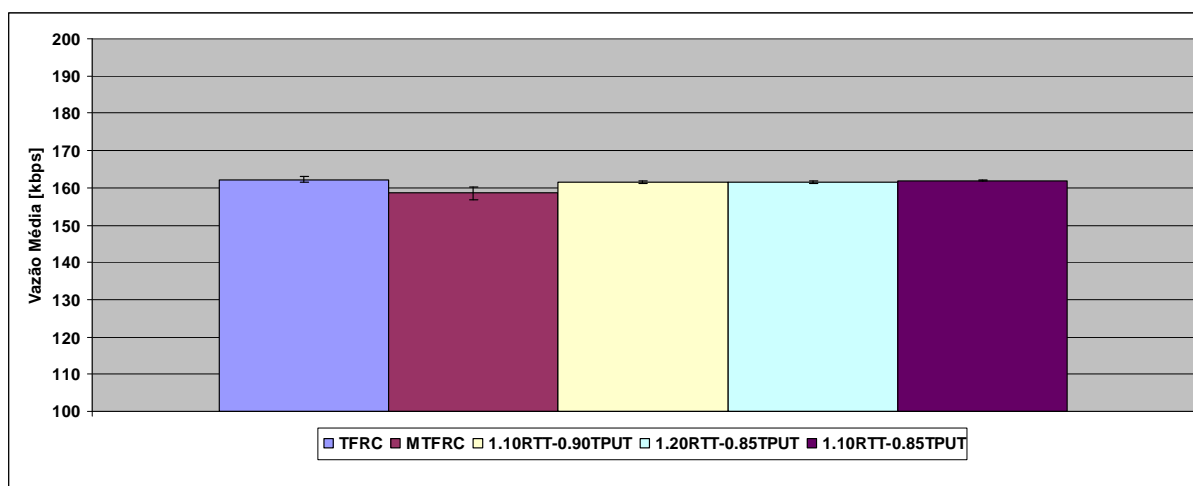


Figura A1.3. Comparação de Vazão Média Topologia com 7 Nós

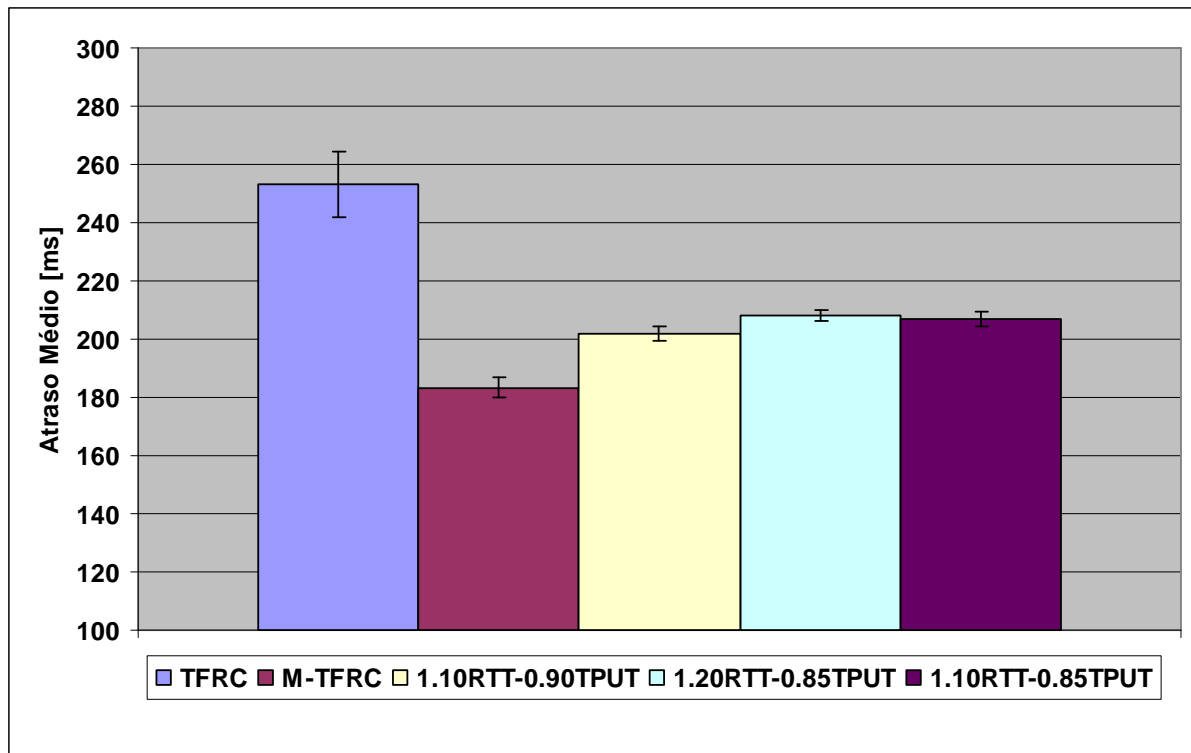


Figura A1.4. Comparação de Atraso Médio Topologia com 7 Nós

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](#)