UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANA
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE ESPECIALIZACÃO EM TELEINFORMÁTICA E REDES DE COMPUTADORES

EDER VICENTIN MESSIAS

ESTUDO DO COMPORTAMENTO DO PROTOCOLO TCP EM REDES SEM FIO

MONOGRAFIA DE ESPECIALIZAÇÃO

CURITIBA
2014

EDER VICENTIN MESSIAS

ESTUDO DO COMPORTAMENTO DO PROTOCOLO TCP EM REDES SEM FIO

Projeto de Especialização apresentada ao Programa de Pós-Graduação em Teleinformática e Redes de Computadores da Universidade Tecnológica Federal do Paraná Campus Curitiba, como requisito parcial à obtenção do título de Especialista em Teleinformática e Redes de Computadores - Área de Concentração: Informática.

Orientador: Christian Carlos Souza Mendes

CURITIBA
2014

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Diretoria de Pesquisa e Pós-Graduação do Campus Curitiba
Departamento Acadêmico de Eletrônica
Especialização em Teleinformática e Redes de Computadores

TERMO DE APROVAÇÃO

Título da Monografia (Estudo do Comportamento do Protocolo TCP em Redes sem Fio)

por

Eder Vicentin Messias

Esta monografia foi apresentada às..... do dia ..... de ..... de 2014 como requisito parcial para a obtenção do título de ESPECIALISTA EM TELEINFORMÁTICA E REDES DE COMPUTADORES, Universidade Tecnológica Federal do Paraná. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado com a nota.....

_____

Prof. Christian Carlos Souza Mendes
(UTFPR)

_____

Visto da Coordenação

_____

Prof. Augusto Foronda
Coordenador do Curso

DEDICATÓRIA

Dedico esse trabalho à todas as pessoas que me ajudaram, seja pelos comentários, pelas observações e críticas que foram de grande valia e ajudaram a construir e melhorar este trabalho. Dedicação especial à minha esposa Andréa e minha filha Laura pelo apoio, tolerância e compreensão, pois entenderam o objetivo e abraçaram a causa junto comigo para que esta etapa da vida fosse superada.

AGRADECIMENTOS

Agradeço primeiramente a DEUS pela inteligência e permitir que eu pudesse realizar mais um sonho. Mais um passo foi dado em direção ao conhecimento.

Agradeço aos amigos com quem compartilhei todo o conhecimento neste período de estudo, aos professores que permitiram agregar novas ideias e desenvolver o pensamento crítico.

Não poderia deixar de agradecer minha querida família pelo incentivo e esforço ao proporcionar um ambiente saudável para o estudo e compreensão dos ensinamentos adquiridos.

À Universidade Tecnológica Federal do Paraná que proporcionou infraestrutura adequada, dinamismo que o curso e os alunos necessitam.

Ao meu orientador Prof. Christian Carlos Souza Mendes pela orientação no desenvolvimento do trabalho, dicas e sugestões de melhoria.

EPÍGRAFE

“O primeiro passo em direção ao sucesso é o conhecimento.”

(Nicola Tesla)

RESUMO

MESSIAS, Eder Vicentin. **Estudo do comportamento do protocolo TCP em redes sem fio**. 2014. 42p. Monografia (Especialização em Teleinformática e Redes de Computadores) – Programa de Pós-Graduação em Teleinformática, Universidade Tecnológica Federal do Paraná. Curitiba, 2014.

As redes sem fio têm características diferentes das redes cabeadas. As principais versões do protocolo TCP apresentam desempenhos distintos frente às adversidades comumente encontradas em redes sem fio como interferências, mobilidade dos nós e colisões frequentes. Baseado nos estudos realizados há alguns pontos falhos destes algoritmos no que se relaciona com as redes sem fio, motivando o estudo de novos algoritmos que melhor se adéquam a essas redes. Em seguida, são apresentadas algumas das versões do protocolo TCP e outras desenvolvidas especificamente para redes sem fio, e são analisados os seus funcionamentos, ressaltando que vantagens teriam sobre os algoritmos tradicionais. Finalmente, conclui-se que nenhum dos protocolos apresentados tem um desempenho totalmente satisfatório ao lidar com todas as características das redes sem fio, sendo um tópico em potencial para novas pesquisas. Para tanto, serão mostradas também algumas novas abordagens que estão em desenvolvimento que relatam diferentes maneiras de controle de congestionamento no TCP.

Palavras-Chaves: protocolo, redes sem fio, algoritmos, desempenho.

ABSTRACT

MESSIAS, Eder Vicentin. **Estudo do comportamento do protocolo TCP em redes sem fio**. 2014. 50 p. Monografia (Especialização em Teleinformática e Redes de Computadores) – Programa de Pós-Graduação em Teleinformática, Universidade Tecnológica Federal do Paraná. Curitiba, 2014.

Wireless networks have different characteristics than wired networks. The traditional TCP versions have a worse performance due to problems commonly found in wireless networks, as interferences, node mobility and frequent collisions. Based on studies, there are some weak points of these algorithms as it relates to wireless networks. Next, some TCP versions developed specifically for wireless networks are studied and their advantages over the traditional TCP algorithms are highlighted. At the end, is concluded that none of the protocols presented has a satisfactory performance dealing with all the wireless networks characteristics, suggesting it's a potential subject for new researches.

Keywords: protocol, wireless networks, algorithms, performance.

LISTA DE SIGLAS

SMTP	*Simple Mail Transfer Protocol*
FTP	*File Transfer Protocol*
HTTP	*Hypertext Transfer Protocol*
Wi-Fi	*Wireless Fidelity*
TCP	*Transmission Control Protocol*
ACK	*Acknowledgment*
OSI	*Open System Interconnect*
IP	*Internet Protocol*
ISP	*Internet Service Provider*
SS	*Slow Start*
SSTRESH	*Slow Start Threshold*
MSS	*Maximum Size Segment*
RTT	*Round-Trip Time*
RFC	*Request For Comment*
CWND	*Congestion Windows*
AWND	*Allowed Windows*
MAC	*Medium Access Control*
PHY	*Physical*
MIMO	*Multiple-Input Multiple-Output*
DCF	*Distributed Coordination Function*
CSMA/CA	*Carrier Sense Multiple Access with Collision Avoidance*
DIFS	*Distributed Inter Frame Space*
RTS	*Request to Send*
CTS	*Clear to Send*
ZWA	*Zero Window Advertisements*

LISTA DE FIGURAS

SUMÁRIO

1. INTRODUÇÃO

Com o advento da expansão dos recursos tecnológicos, aumento do número de usuários acessando as redes de computadores (foco em alta disponibilidade e instantaneidade) e a busca por maiores desempenhos provido pelos ISPs, um dos grandes problemas enfrentados atualmente são os colapsos de congestionamento de dados. Este fenômeno contribui negativamente para o aumento do tempo de entrega de pacotes e diminuição da vazão/fluxo dos dados (ROESLER, 2004).

O protocolo IP apenas fornece conectividade entre dois sistemas finais (também chamados de hospedeiros), porém não provê garantia de entrega dos pacotes, o que é especificamente a função do protocolo TCP.

O protocolo TCP está contido na camada de transporte da pilha de protocolos do modelo TCP/IP (fig.1), que roda sobre o IP e que fornece transferência confiável de dados sobre uma rede não confiável. O TCP foi desenvolvido para dinamicamente adaptar-se às condições da rede fornecendo controle de congestionamento (ROESLER, 2004).

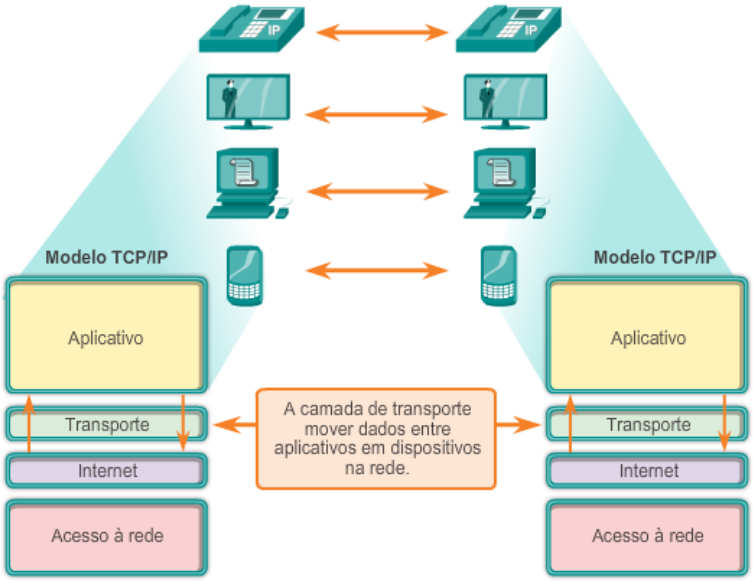

Figura 1 - Modelo TCP/IP
Fonte: http://www.cisco.ct.utfpr.edu.br/

Existem dois mecanismos de controle de congestionamento. O controle fim-a-fim e o assistido pela rede. No assistido pela rede, os roteadores que compõem a rede fornecem realimentação para os remetentes sobre o estado de congestionamento.

Já no mecanismo fim-a-fim, a camada de rede não fornece qualquer informação de congestionamento. O congestionamento da rede deve ser detectado de alguma forma pelos sistemas finais (onde está alocado o protocolo TCP). O TCP utiliza o método fim-a-fim na maioria dos casos.

Congestionamentos na rede são detectados pelo protocolo da camada de transporte através de perdas ou atrasos de pacotes. O controle do congestionamento é feito através de uma janela de congestionamento semelhante esta da figura 2 (KUROSE; ROSS, 2010).

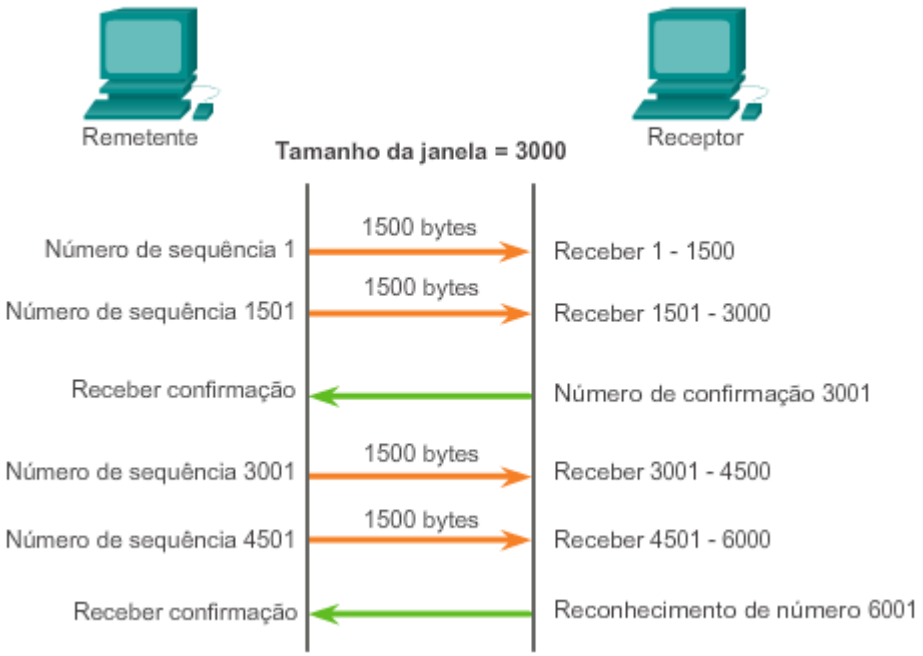

Figura 2 – Exemplo de janela de congestionamento
Fonte: http://www.cisco.ct.utfpr.edu.br/material

Veja no exemplo acima que o tamanho da janela inicial para uma sessão TCP representada é definido em 3000 bytes. Quando o remetente tiver transmitido 3000 bytes, ele espera por uma confirmação desses bytes antes de transmitir mais segmentos nesta sessão. Quanto o remetente tiver recebido essa confirmação do receptor, o remetente poderá transmitir mais 3000 bytes.

O TCP usa tamanhos de janela para tentar gerenciar a taxa de transmissão até o fluxo máximo que a rede e o dispositivo destino podem suportar, enquanto minimiza perdas e retransmissões. O TCP também fornece mecanismos de controle de fluxo, uma vez que auxilia na manutenção da confiabilidade da transmissão TCP definindo a taxa de fluxo entre a origem e o destino em uma determinada sessão. Esse controle de fluxo é percebido quando é limitada a quantidade de dados dos segmentos enviados ao mesmo tempo e exigindo a recepção de um aviso do destinatário antes de enviar mais segmentos.

Para realizar o controle de fluxo, a primeira coisa que o TCP faz é determinar a quantidade de dados de segmento que o dispositivo destino pode aceitar. O cabeçalho TCP inclui um campo de 16 bits chamado de tamanho de janela (fig. 3) e é esse o número de bytes que o dispositivo destino pode aceitar e processar ao mesmo tempo (LEUNG, 2006).

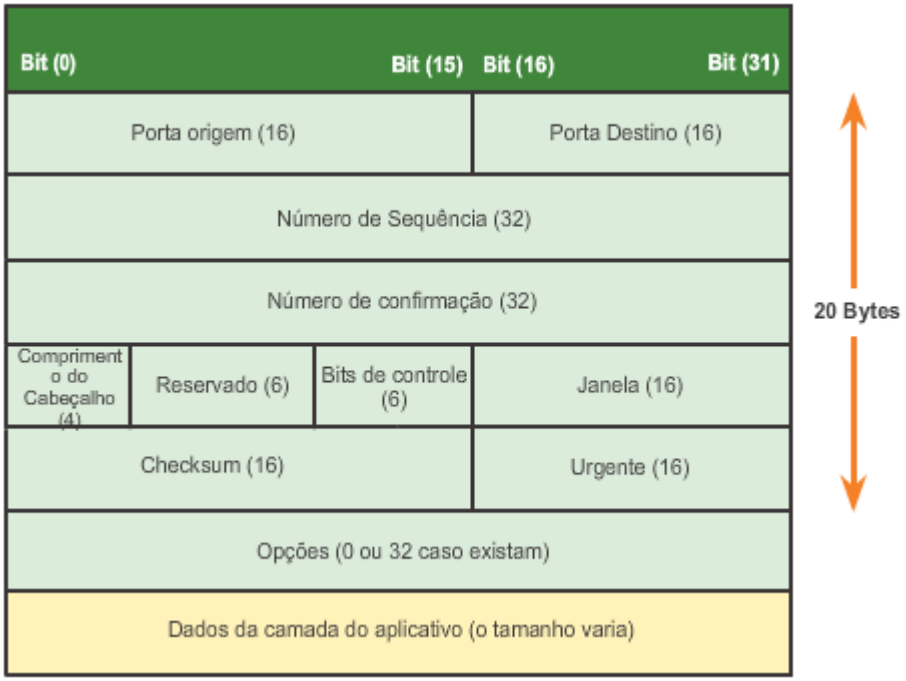

Figura 3 - Cabeçalho TCP (PDU)
Fonte: http://www.cisco.ct.utfpr.edu.br/material

O tamanho da janela inicial é acordado durante a inicialização da sessão por meio do *handshake* triplo entre a origem e o destino. Uma vez acordado, o

dispositivo origem deve limitar a quantidade de dados dos segmentos enviados ao destino com base no tamanho da janela.

Somente depois que o dispositivo origem receber uma confirmação de que os segmentos de dados foram recebidos, ele poderá continuar a enviar mais dados para a sessão. Até receber a confirmação, o emissor não enviará quaisquer segmentos adicionais. Em períodos em que a rede está congestionada ou os recursos do host destino estão extenuados, o atraso pode aumentar. À medida que este atraso aumenta, a taxa de transmissão efetiva dos dados para esta sessão diminui. A redução da transmissão de dados de cada sessão ajuda a reduzir o conflito de recursos do dispositivo de rede e destino enquanto as várias sessões estão em execução.

Esta janela define a negociação entre remetente e destinatário, ou seja, o quanto o remetente está apto a enviar ao destinatário e quanto este, por sua vez, pode receber (volume de tráfego em bits por segundo). O tamanho da janela é variável, aumentando na medida em que os segmentos enviados são reconhecidos.

Por outro lado, a janela também pode diminuir ao ocorrer um evento de perda de pacote, pois o TCP interpreta que a perda do pacote tenha ocorrido por descarte numa fila cheia, como consequência de pacotes sem reconhecimento, corrompidos ou atrasados (LEUNG, 2006).

Em redes cabeadas, uma perda de pacote geralmente significa que há congestionamento e o protocolo TCP deve diminuir o fluxo de dados. Já em uma rede sem fio, uma perda de pacote pode significar que simplesmente houve uma colisão ou interferência. Por este motivo, o comportamento padrão do TCP em redes sem fio não é o mais eficiente, pois uma colisão de um único pacote pode ser erroneamente interpretada como congestionamento e diminuir drasticamente a taxa de transmissão (vazão/fluxo de dados).

O TCP possui diferentes versões com diferentes abordagens de controle e detecção de congestionamento. Com isso, as diferentes versões do protocolo TCP podem apresentar diferenças de desempenho em redes sem fio.

1.1. OBJETIVOS

1.1.1. Objetivo Geral

Este trabalho tem o objetivo de estudar o protocolo TCP, analisar o seu comportamento em redes sem fio padrão 802.11 e discutir as alternativas existentes na literatura.

Serão analisadas as diferenças de desempenho das versões mais comuns do TCP em redes sem fio, e sugerir a adoção de implementações já disponíveis.

1.1.2. Objetivos Específicos

Prover um estudo comparativo de algumas das principais versões do protocolo TCP num ambiente de rede sem fio, lidando com todas as características deste cenário que é muito comum em ambientes corporativos, campus de universidades e outros. Detalhar vantagens e desvantagens destas versões no tratamento de congestionamento de rede, atrasados, perdas de pacote e outros fenômenos.

1.2. JUSTIFICATIVA

Este trabalho está centrado no estudo e análise do comportamento das versões do protocolo TCP mais utilizadas em redes sem fio. Várias implementações do protocolo surgiram pela necessidade de se ter melhor aproveitamento do enlace.

Como o protocolo tem as funções de detecção de erros, ordenação dos pacotes, controle de fluxo e controle de congestionamento, ele necessita de algoritmos que tem o objetivo de otimizar a vazão, a fim de manter a alta eficiência

de utilização do canal e evitar que os diversos fluxos concorrentes causem congestionamento na rede.

É de grande interesse para quem atua no segmento de gerenciamento e implementação de serviços de acesso Wi-Fi, controlar da melhor maneira, devido à grande necessidade de informação em tempo real, alta disponibilidade de acesso e troca de informações.

Posicionada entre as camadas de aplicação e de rede, a camada de transporte é uma peça central da arquitetura de rede em camadas. Ela desempenha o papel fundamental de fornecer serviços de comunicação diretamente aos processos de aplicação que rodam nos sistemas finais comunicantes.

Se fossemos listar os dez maiores problemas fundamentalmente importantes para o trabalho em redes, o da transferência confiável de dados seria o candidato número 1.

Mobilidade dos nós é um fator importantíssimo na escolha desse tipo de acesso (Wi-Fi) e, garantir níveis de qualidade de serviço, é primordial no que diz respeito à experiência dos clientes/usuários.

Portanto, este trabalho monográfico poderá auxiliar pessoas engajadas no ramo de novas tecnologias de acesso sem fio a entender melhor o protocolo de camada de transporte TCP e suas aplicações.

2. PROTOCOLO TCP

O TCP/IP é uma combinação de dois protocolos individuais. O IP opera na camada 3 do modelo OSI e é um protocolo sem conexão, que oferece um serviço de entrega de melhor esforço (*Best Effort)* em uma rede. O TCP opera na camada 4 do mesmo modelo e é um serviço orientado à conexão, ou seja, ele necessita estabelecer uma sessão entre os hosts antes de efetivamente trocar pacotes (COMER, 1998).

Esses protocolos (TCP e IP) juntos fornecem uma ampla variedade de serviços e são a base de todo um conjunto de protocolos chamado TCP/IP. A internet foi construída com base nesse conjunto de protocolos. O modelo TCP/IP tem as seguintes camadas (fig.4).

- Camada de Aplicação
- Camada de Transporte
- Camada de Internet
- Camada de acesso à rede

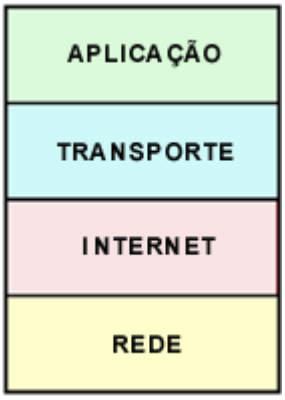

Figura 4 - Camada da Arquitetura TCP/IP
Fonte: http://www.ccuec.unicamp.br/treinamento_int2004/tcpip/03.html

A camada de transporte lida com questões de qualidade de serviços de confiabilidade, controle de fluxo e correções de erros. Um de seus protocolos, o TCP, fornece formas flexíveis de se desenvolver comunicação de rede confiável com baixa taxa de erro e bom fluxo.

Ele mantém um diálogo entre a origem e o destino enquanto empacota informações da camada de aplicação em unidades chamadas segmentos. O termo orientado à conexão não quer dizer que existe um circuito entre os computadores comunicantes; significa que segmentos da camada 4 do modelo OSI trafegam entre dois hosts para confirmar que a conexão existe logicamente durante um certo período.

O propósito da camada de internet é dividir os segmentos TCP em pacotes e enviá-los a partir de qualquer rede. Os pacotes chegam à rede de destino independente do caminho; a determinação do melhor caminho e a comutação de pacotes ocorrem na camada de Internet do modelo TCP/IP. Pode-se imaginar que o IP aponta o caminho para os pacotes, enquanto que o TCP proporciona um transporte confiável.

O significado do nome da camada de acesso à rede é muito amplo. Esta camada lida com todos os componentes, tanto físico como lógico, que são necessários para fazer um link físico (COMER, 1998).

Desenvolvido na década de 70, o TCP é destinado a prover transporte confiável de dados de aplicações a exemplo de SMTP, FTP, HTTP, Telnet, conforme mostra a figura 5 abaixo:

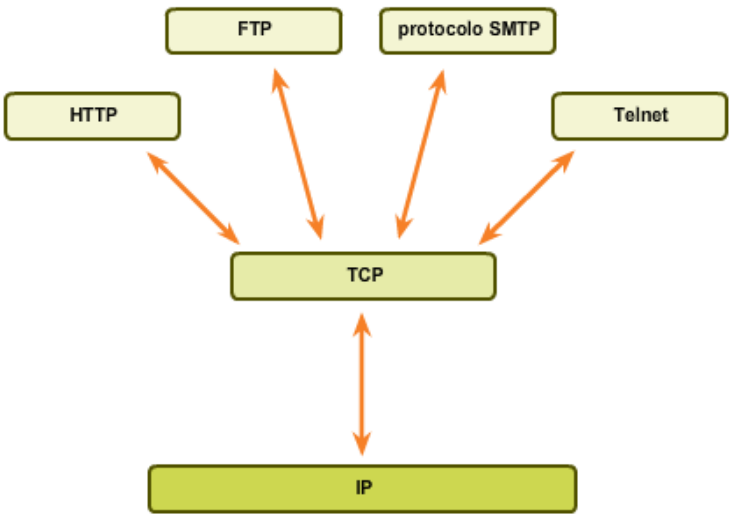

Figura 5 - Alguns exemplos de aplicativos que usam TCP
Fonte: http://www.cisco.ct.utfpr.edu.br/

As primeiras implementações do TCP não possuíam mecanismos de controle de congestionamento devido ao tráfego de Internet naquela época não ser tão intenso como o atual.

Ao se usar TCP é necessário estabelecer antes uma conexão de troca de dados para que os sistemas finais saibam da existência um do outro e para que troquem informações como porta de origem e destino, números de sequência iniciais, janelas iniciais, tamanho máximo de segmento, tipo de implementação para controle de congestionamento, dentre outros (fig. 6).

> O TCP é orientado para conexão porque, antes que um processo de aplicação possa começar a enviar dados a outro, os dois processos precisam primeiramente se 'apresentar' – isto é, devem enviar alguns segmentos preliminares um ao outro para estabelecer os parâmetros da transferência de dados em questão. Como parte do estabelecimento da conexão TCP, ambos os lados da conexão iniciarão muitas "variáveis de estado" associadas com a conexão TCP (KUROSE;ROSS, 2010, p.174).

A janela limita quantos segmentos podem ser enviados simultaneamente. Conforme os segmentos já enviados são reconhecidos, a janela move-se e aumenta, permitindo que novos segmentos de dados sejam enviados.

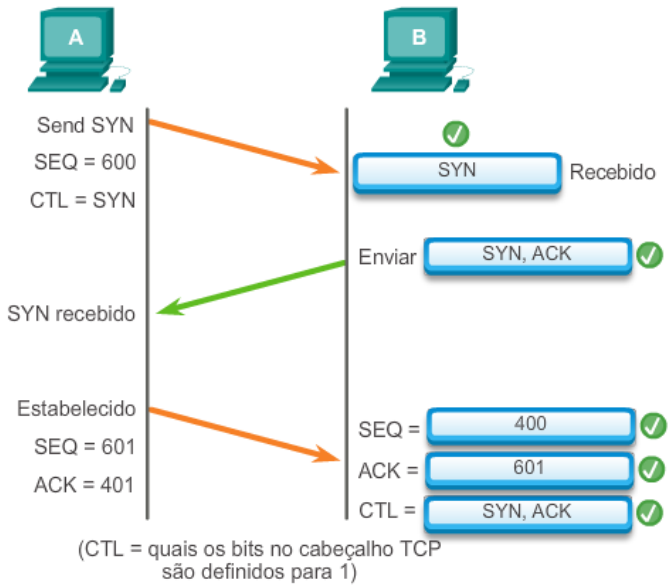

Figura 6 - Handshake tripo da sessão de estabelecimento de TCP
Fonte: http://www.cisco.ct.utfpr.edu.br/material/CCNA5.0/

Durante o estabelecimento de uma conexão, são especificados os valores iniciais para AWND e CWND. O valor de AWND é anunciado pelo transmissor e pelo receptor no campo janela do cabeçalho TCP. Não há um campo para CWND. Por

esta razão, esta variável é criada pelo algoritmo chamado *Slow Start* com o valor inicial de 1*MSS.

Depois do estabelecimento da conexão, ou seja, durante a troca de dados, a AWND ainda continua sendo fornecida dinamicamente pelos servidores terminais. Já no caso da CWND, são usados vários algoritmos para a sua regulação.

Ao longo dos tempos foram feitas alterações no protocolo TCP, resultando em várias versões com diferentes características. Basicamente estas diferenças consistem no modo em que a janela irá aumentar ou diminuir e como reagir a perdas de pacotes. Isto faz com que estas versões tenham diferentes desempenhos em situações distintas. O TCP está definido nos RFCs 793, 1122, 1323,2018 e 2581 (KUROSE;ROSS, 2010).

Estas versões coexistem nas redes atualmente. A seguir, apresentamos o funcionamento das versões mais importantes encontradas do TCP.

2.1. TCP Tahoe

É a primeira versão do protocolo TCP desenvolvida e possui um funcionamento relativamente simples e inclui o controle de congestionamento. Ele usa os algoritmos de *Slow Start*, *Congestion Avoidance* e *Fast Retransmit* juntamente com modificações no tempo RTT. A retransmissão era feita de acordo com a estratégia "*go-back",* ou seja, todos os segmentos após o segmento dado por perdido eram retransmitidos. Não havia nenhum tipo de busca de realimentação de informações que indicasse a situação da rede. Além disso, os temporizadores do TCP eram diferentes no cálculo da estimativa do RTT, o que provocava problemas nas retransmissões.

Na fase inicial do TCP Tahoe, a janela de congestionamento é definida inicialmente com 1*MSS e aumenta exponencialmente (fig. 7).

A cada ACK recebido, a janela aumenta em 1 MSS. Em um RTT, que é o tempo gasto de ida e volta do pacote para cada conexão, a janela terá dobrado de tamanho. A janela aumenta até atingir o sstresh, ou seja, limiar da partida lenta. Ao chegar neste limiar, inicia-se o processo de prevenção de congestionamento (JACOBSON, 1988).

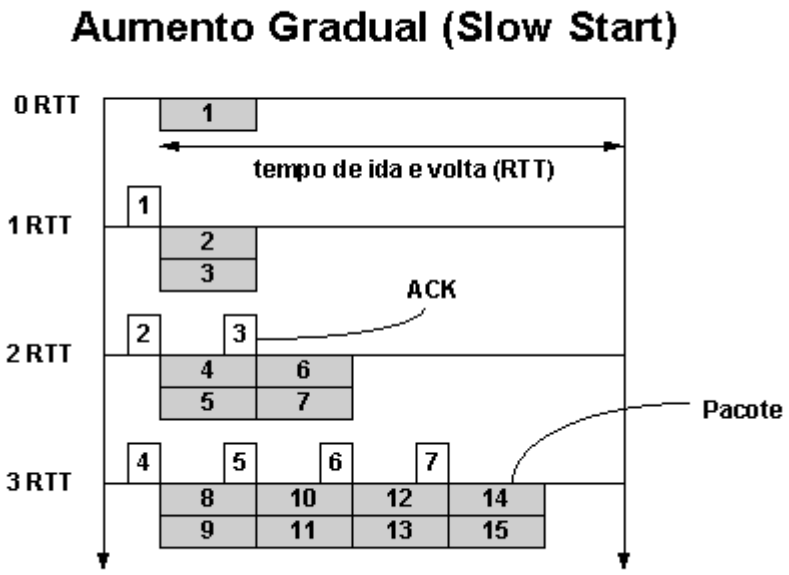

Figura 7 - Comportamento Janela de Congestionamento do Tahoe
Fonte: http://memoria.rnp.br/newsgen/9909/tcp_atm.html

A fase de prevenção de congestionamento, ou *Collision Avoidance*, também é conhecida por AIMD.

A cada segmento reconhecido a janela aumenta de 1*MSS em 1*MSS CWND. Isto significa que quando todos os segmentos da janela forem reconhecidos, a janela terá aumentado em 1MSS. Portanto, nesta fase, a janela aumenta linearmente. Ao ocorrer um evento de perda, o sstresh é diminuído pela metade e inicia-se novamente partida lenta.

O TCP Tahoe não considera ACK duplicados, logo sempre que um segmento for perdido a fase de partida lenta irá iniciar novamente após o *timeout* (fig. 8).

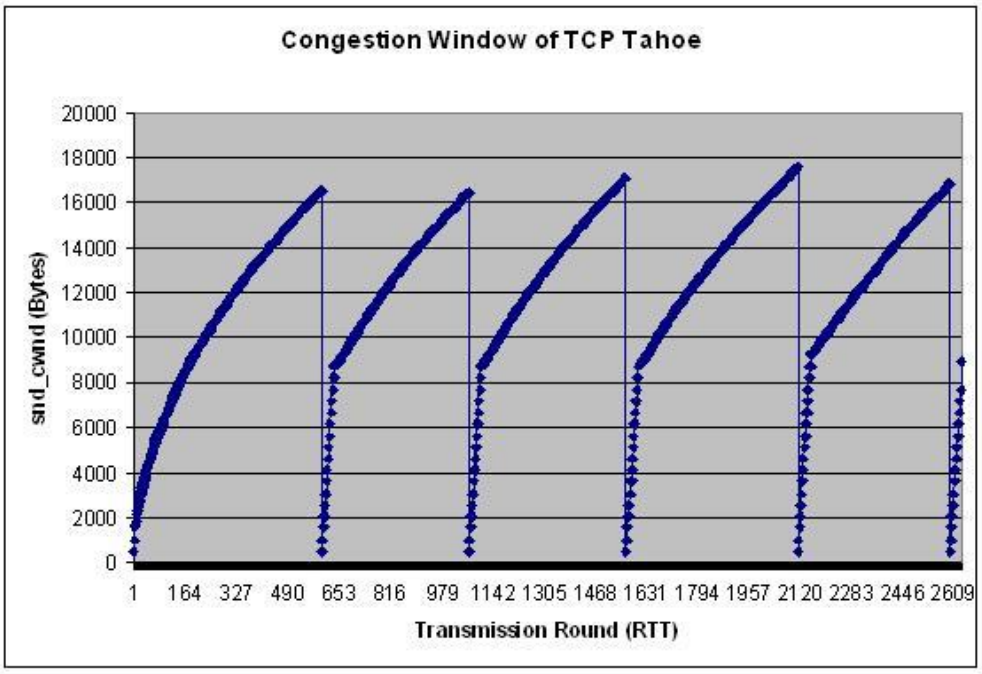

Figura 8 - Janela de Congestionamento do TCP Tahoe
https://www.google.com.br/search?q=grafico+tahoe

A desvantagem é que esta versão dispara várias vezes o algoritmo de *Slow Start,* diminuindo o desempenho da rede (JACOBSON, 1988).

2.2. TCP RENO

O *Reno* é um melhoramento do *Tahoe*, adicionando os algoritmos de *Fast Retransmit* e *Fast Recovery*, sendo utilizado na maioria dos dispositivos ligados à Internet. O TCP Reno funciona de um modo parecido com o TCP Tahoe, diferencia-se apenas na resposta a ACKs duplicados Tanto o *Reno* quanto o *Tahoe* atribuem um segmento para a janela de congestionamento até um *timeout*.

No caso do *Reno*, que utiliza o *Fast Retransmit*, a transmissão de um segmento perdido é disparada e executada depois que três reconhecimentos duplicados são recebidos antes do *timeout* ser alcançado (fig. 9).

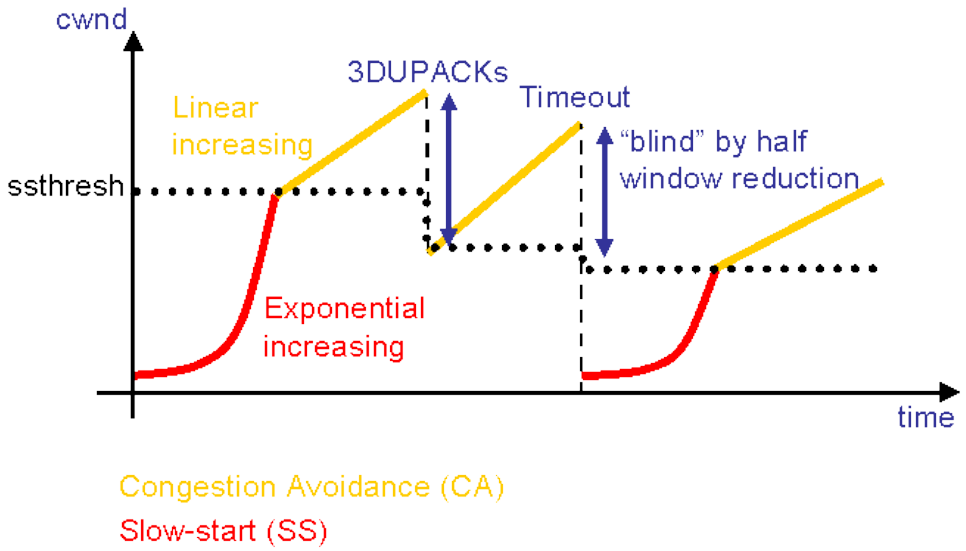

Figura 9 - Janela de Prevenção de Congestionamento TCP Reno
Fonte: https://www.google.com.br/search?q=grafico+reno

O *Fast Recovery* faz com que a janela de congestionamento aumente de acordo com o número de reconhecimentos duplicados anteriores a um novo reconhecimento.

Uma desvantagem do TCP *Reno* é que ele não retorna ao algoritmo de *Slow Start* quando múltiplos segmentos são perdidos e dispara o algoritmo de *Congestion Avoidance*. Consequentemente, a redução da janela de congestionamento à metade de seu valor ocorre várias vezes e o algoritmo *Slow Start* é utilizado somente quando o *timeout* expirar.

A retransmissão rápida consiste no reenvido automático de um segmento perdido ao receber três ACKs duplicados.

Deste modo, não é necessário aguardar pelo *timeout* para perceber que o segmento foi perdido. Depois da retransmissão rápida, apesar da perda do pacote, o TCP não volta para a partida lenta. O TCP Reno considera que se o remetente recebeu três ACKs duplicados, apenas um segmento foi perdido e os outros da sequência chegaram ao destino, logo é provável que não houve congestionamento na rede. O ssthresh é definido como metade do tamanho da janela de congestionamento e o cwnd é definido para o mesmo valor de ssthresh. Esta é a recuperação rápida.

O TCP Reno apresenta um problema se múltiplos pacotes são perdidos. Para

cada pacote perdido o TCP Reno irá entrar na recuperação rápida, diminuir a janela de congestionamento e retornar para o modo normal. Logo, se houver duas perdas de pacotes na mesma janela, o TCP Reno entrará no modo de recuperação rápida duas vezes a janela será reduzida 2 vezes. Outra limitação deste protocolo ocorre quando a janela é muito pequena.

Neste caso, se houver uma perda, o protocolo não recebe 3 ACKs duplicados e só detecta a perda depois do *timeout* (University of California, 2002).

2.3. TCP NEW-RENO

O TCP New-Reno é uma otimização do Reno para o caso em que múltiplas perdas acontecem em uma única janela de transmissão. Ele inclui a modificação no algoritmo de *Fast Recovery,* eliminando a necessidade de esperar por um estouro do temporizador no caso dos múltiplos descartes (LEUNG, 2006).

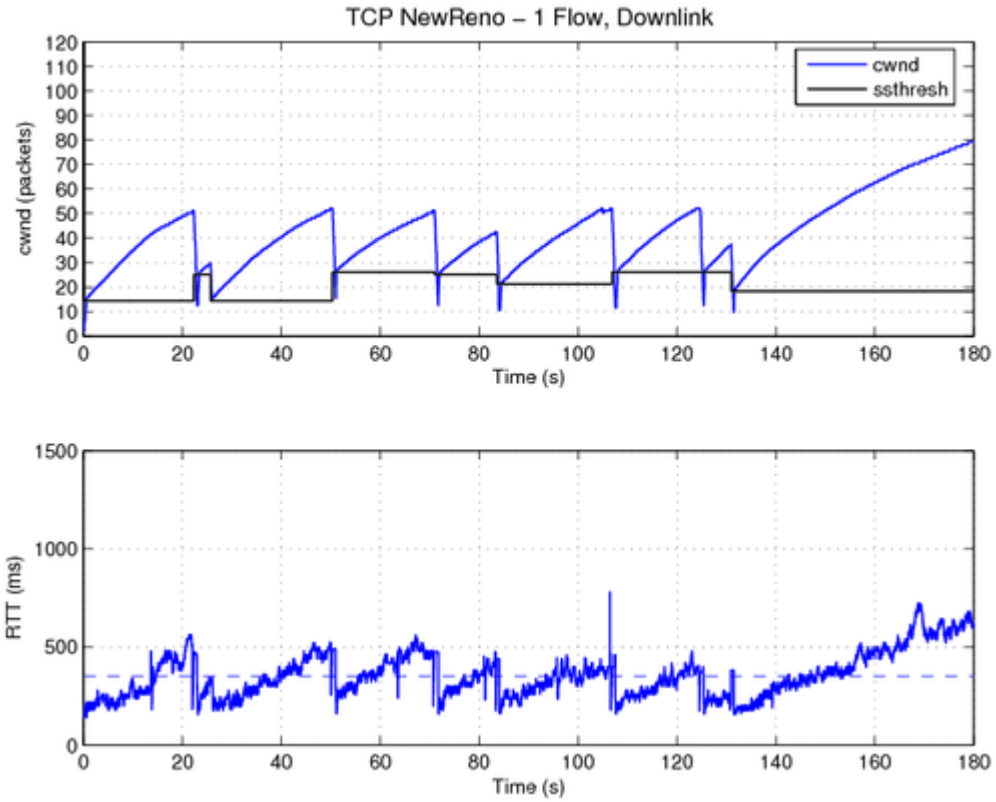

Figura 10 - Janela de Congestionamento TCP New Reno
Fonte: Fonte: https://www.google.com.br/search?q=grafico+reno

O algoritmo New Reno (fig.10) utiliza o conceito de reconhecimento parcial. Deve-se notar que quando o TCP percebe que ocorreu a perda de pacote através de recebimento de um determinado número de reconhecimentos duplicados, a próxima informação nova para o emissor só será dada com a chegada do reconhecimento para aquele pacote retransmitido. No caso de uma única perda, esse reconhecimento irá confirmar o recebimento de todos os pacotes transmitidos antes do início do *fast retransmit.* Se ocorrer a perda de múltiplos pacotes, entretanto, o reconhecimento irá confirmar alguns segmentos, mais precisamente, todos os reconhecimentos até a próxima perda. Esse reconhecimento que não confirma todos os pacotes enviados antes do início do *fast retransmit* é chamado de reconhecimento parcial (LEUNG, 2006).

Pode-se concluir então que reconhecimentos parciais duplicados, da mesma forma que os reconhecimentos duplicados iniciais, indicam muito provavelmente que um outro pacote dentro da mesma janela foi perdido ao longo do caminho.

Nesta variante, o algoritmo de *Fast Recovery* é interrompida ao ser recebida a confirmação da chegada de todos os segmentos pendentes, inclusive o que foi recuperado. Sua desvantagem é que a retransmissão de mais de um segmento a cada RTT provoca atrasos no envio dos próximos segmentos. Ele realiza reconhecimentos parciais e se mantém no algoritmo de *Fast Retransmit*, evitando múltiplas reduções na janela de congestionamento. Uma outra desvantagem do *New-Reno* é o fato de demorar um RTT para detectar que mais de um pacote foi perdido, pois somente quando o reconhecimento do primeiro segmento retransmitido é recebido que é detectada a perda de outro segmento.

Quando ocorre perda de um pacote, a próxima informação será enviada após o reconhecimento do pacote retransmitido. Caso aconteça uma única perda, o reconhecimento confirmará a chegada de todos os pacotes, antes do algoritmo de *Fast Retransmit* e, se acontecerem múltiplas perdas, o reconhecimento confirmará os segmentos até a próxima perda (LEUNG, 2006).

O *New-Reno* consegue detectar melhor perdas de segmentos, contornando os problemas do TCP Reno. O TCP *New-Reno* aguarda todos os pacotes já enviados serem reconhecidos antes de sair da retransmissão rápida, evitando diminuir várias vezes a janela (University of Califórnia, 2002).

2.4. TCP SACK

O TCP tradicional realiza um esquema de reconhecimentos cumulativos, em que ACK(n) indica que todos os segmentos até n-1 foram recebidos com sucesso e o transmissor pode mandar o segmento (n).

Isto impõem uma grave limitação de desempenho em situações onde há mais de uma perda na mesma janela de transmissão, porque o transmissor tem que esperar um RTT para identificar cada segmento perdido. Isto ocorre porque caso haja diversas perdas, os reconhecimentos são enviados referenciando-se ao primeiro pacote perdido, não dando maiores informações sobre as perdas as demais perdas ao emissor (T. HENDERSON, 1999).

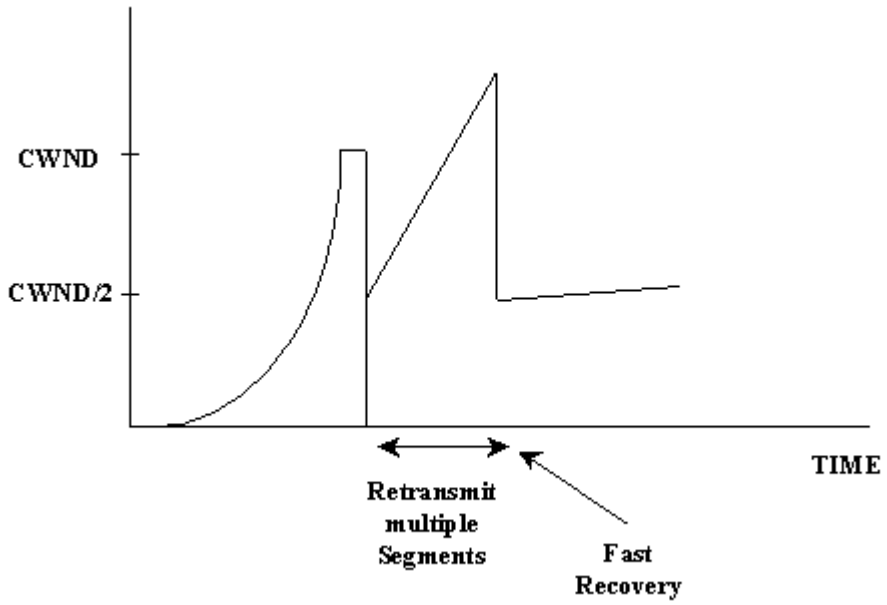

Figura 11 - Janela de Congestionamento do TCP SACK
Fonte: http://www.cse.wustl.edu/

Assim, o emissor reenvia apenas o pacote referenciado pelos ACKs duplicados, tendo que aguardar por um RTT a chegada do reconhecimento do pacote retransmitido, que estará indicando o próximo pacote perdido. Desta forma apenas um segmento perdido pode ser notado a cada RTT. A alternativa para esta limitação seria retransmitir mais do que o necessário, como por exemplo todos os pacotes após aquele primeiro que foi perdido. Medidas assim, entretanto, além de

ineficientes podem acabar agravando uma situação de congestionamento existente na rede (T. HENDERSON, 1999).

Uma solução para este problema seria o uso de reconhecimentos seletivos por parte do TCP. Com esta modificação, o emissor tem mais informações sobre quais segmentos foram recebidos corretamente e quais não foram, podendo portanto retransmitir somente os pacotes necessários, aumentando a eficiência na recuperação de perdas.

Esta modificação usa o campo *options* do cabeçalho TCP para incluir as informações de reconhecimento seletivo. Existem dois campos opcionais que são utilizados pelo TCP SACK: um para estabelecer se a opção de SACK será ou não usada (enviada ao iniciar uma conexão), e o SACK propriamente dito, que passa as informações sobre os segmentos reconhecidos.

O TCP com reconhecimento seletivo é uma extensão do TCP Reno e elimina os problemas do *Reno* e *New-Reno* de perdas de múltiplos pacotes e retransmissões de mais de um pacote perdido por RTT. Nesta implementação, os segmentos são reconhecidos seletivamente e não cumulativamente. Assim, o remetente consegue saber precisamente quais segmentos chegaram ao destino e quais precisam ser retransmitidos. Reconhecimento seletivo é uma opção do TCP habilitada ao se definir um determinado bit no primeiro segmento (SYN) ao iniciar a conexão. Se ambos os lados suportarem esta característica, reconhecimentos seletivos serão usados na comunicação (T. HENDERSON, 1999).

2.5. TCP VEGAS

O TCP Vegas apresenta algumas modificações mais profundas com o objetivo de melhorar o desempenho do *Reno*. As alterações do *Vegas* acontecem somente do lado transmissor. Do lado do receptor, a política de emissão de reconhecimentos é a mesma do TCP *Reno.* A idéia básica do *Vegas* é detectar o congestionamento nos roteadores entre a fonte e o destino antes de ocorrer a perda do pacote e reduzir a taxa linearmente quando a iminente perda do pacote é detectada.

Considerando que o *Tahoe* e o *Reno* reagem ao congestionamento, o TCP Vegas é uma abordagem proativa de controle de congestionamento. Esta implementação detecta perdas de segmentos através da estimativa do RTT. Se o reconhecimento de segmentos demorarem mais do que o RTT estimado, é considerado que ocorreu uma perda (HENGARTNER, 2000).

Na fase de prevenção de congestionamento, o TCP Vegas não detecta congestionamento por perdas de pacotes e sim por uma redução na taxa de envio comparada com a taxa esperada. Da mesma forma, o TCP Vegas aumenta a taxa se estiver abaixo da taxa esperada. Ele tem um melhor controle de congestionamento, evitando saturar a banda disponível para depois diminuir a janela.

O TCP Vegas também introduz modificações na slow-start, para evitar gerar congestionamento na rede aumentando demais a janela de congestionamento. A janela de congestionamento só aumenta exponencialmente a cada RTT. Entre RTTs, o TCP Vegas calcula a taxa real e compara com a esperada. Quando a diferença for maior que certo limite inicia-se a fase de prevenção de congestionamento (HENGARTNER, 2000).

2.6. CONSIDERAÇÕES

Segue abaixo uma tabela que representa as diferentes implementações do TCP explicadas até aqui para que se possa observar as diferenças entre elas; isso com relação aos mecanismos de controle de congestionamento e à política de reconhecimento empregada.

Tabela 1 - Comparação das diferentes implementações do TCP

Implementação TCP	Slow Start	Congestion Avoidance	Fast Retransmit	Fast Recovery	Política de Reconhecimento
Tahoe	Sim	Sim	Sim	Não	Cumulativo
Reno	Sim	Sim	Sim	Sim	Cumulativo
New Reno	Sim	Sim	Sim	Sim	Cumulativo, difrencia os ACKs parciais
Vegas	Próprio	Próprio	Não	Não	Cumulativo
Sack	Sim	Sim	Sim	Sim	Cumulativo com Seletividade

O TCP Vegas se mostrou extremamente eficiente quanto a perda de pacotes, chegando muitas vezes a zero de perda. Ele foi melhor que o TCP *Tahoe* porque é muito mais robusto perante perdas de pacotes. Ele pode detectar e retransmitir pacotes perdidos muito mais cedo que o *timeout* do TCP *Tahoe*. Também tem menos retransmissões e é melhor em prevenção de congestionamentos.

Em comparação com o protocolo *Reno,* o Vegas não tem que aguardar por 3 reconhecimentos duplicados, assim ele pode retransmitir mais cedo e não reduz a janela de congestionamento prematuramente (University of California, 2002).

3. COMPORTAMENTO

3.1 INTERFERÊNCIA

Apesar da camada de enlace e camada fisica serem diferentes, o protocolo TCP é o mesmo e os erros introduzidos simulam interferencias na rede, como ocorreria em redes sem fio. Temos abaixo gráficos sem interferencia e com interferencia que causam 0.5%, 1% e 2% de perda de pacote. Cada ambiente tem seu próprio nivel de ruido, podendo estar dentro desta faixa ou muito acima dela. Último gráfico temos um comparativo das versão aqui estudadas com 1% de interferência (BRAKMO, 1995).

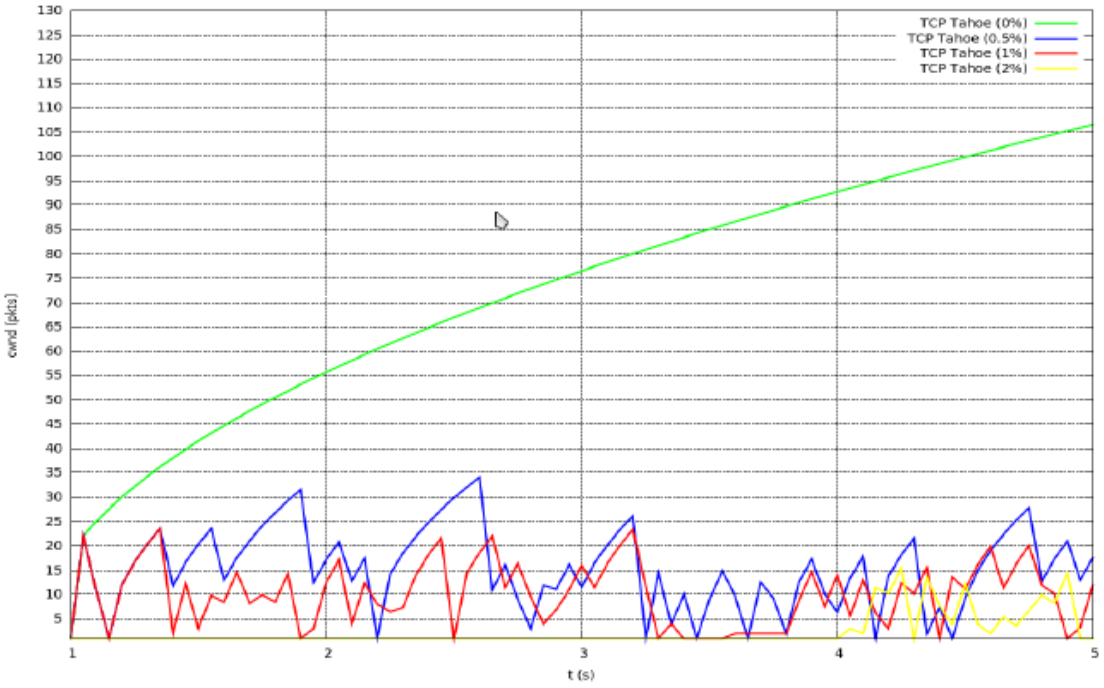

Figura 12 - Desempenho do TCP Tahoe frente à interferência
Fonte: http://www.soi.wide.ad.jp/class/20070044/slides/02/index_44.html

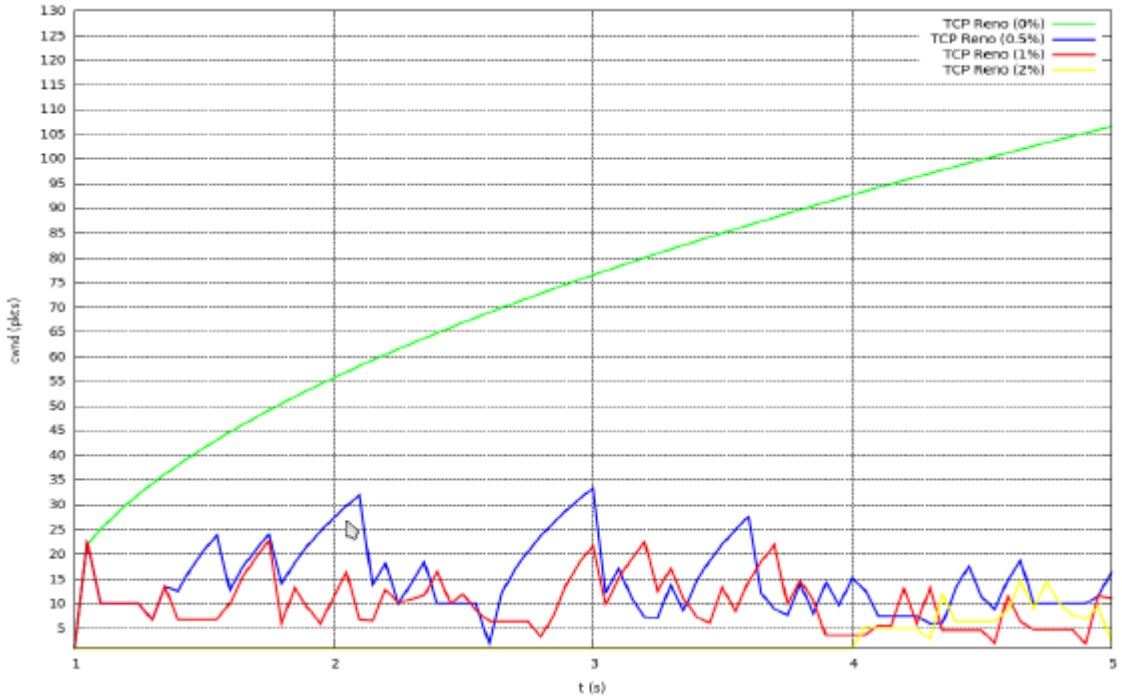

Figura 13 - Desempenho do TCP Reno frente à interferência
Fonte: http://www.soi.wide.ad.jp/class/20070044/slides/02/index_44.html

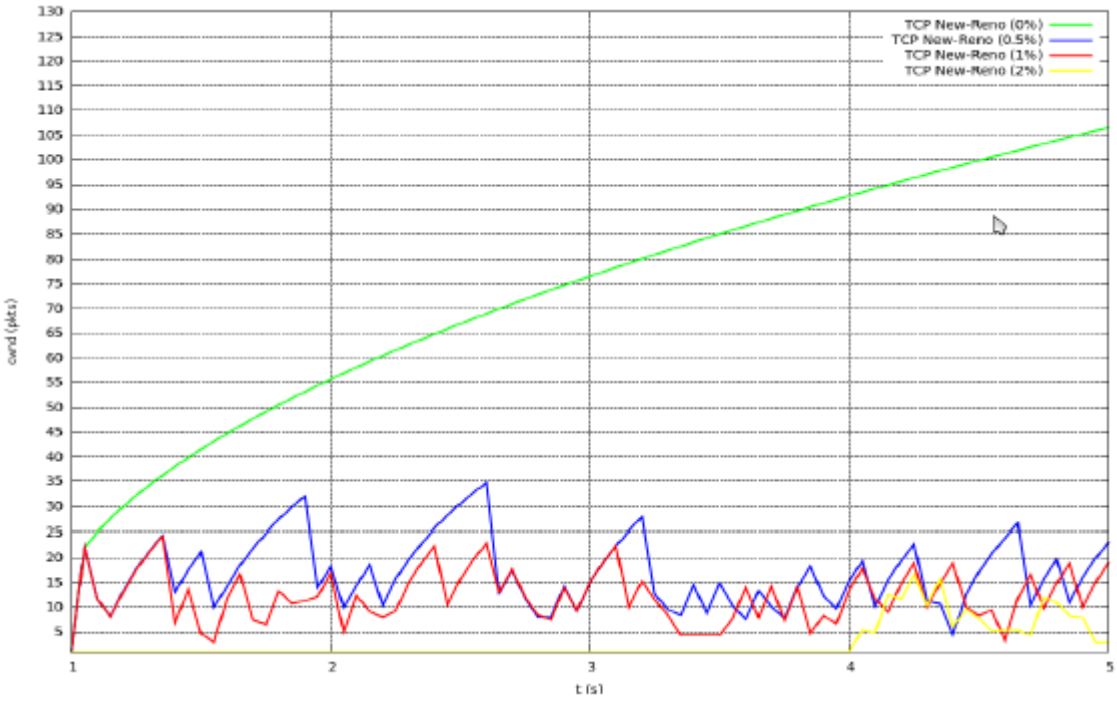

Figura 14 - Desempenho do TCP New-Reno frente à interferência
Fonte: http://www.soi.wide.ad.jp/class/20070044/slides/02/index_44.html

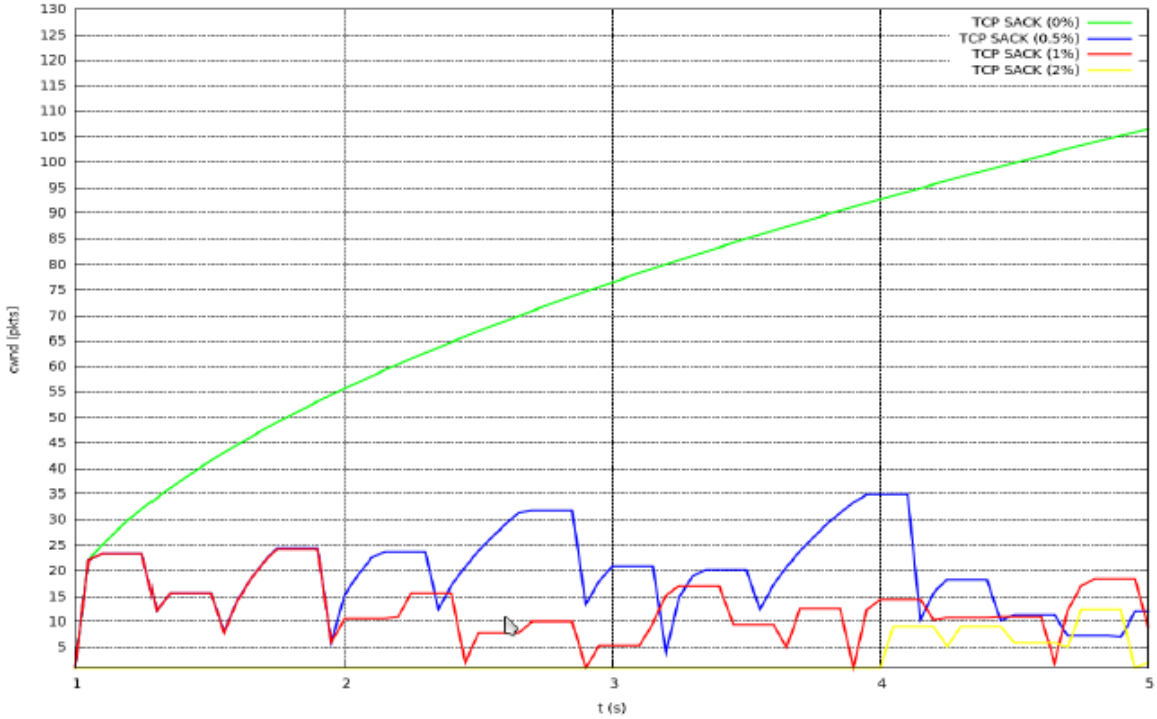

Figura 15 - Desempenho do TCP SACK frente à interferência
Fonte: http://www.soi.wide.ad.jp/class/20070044/slides/02/index_44.html

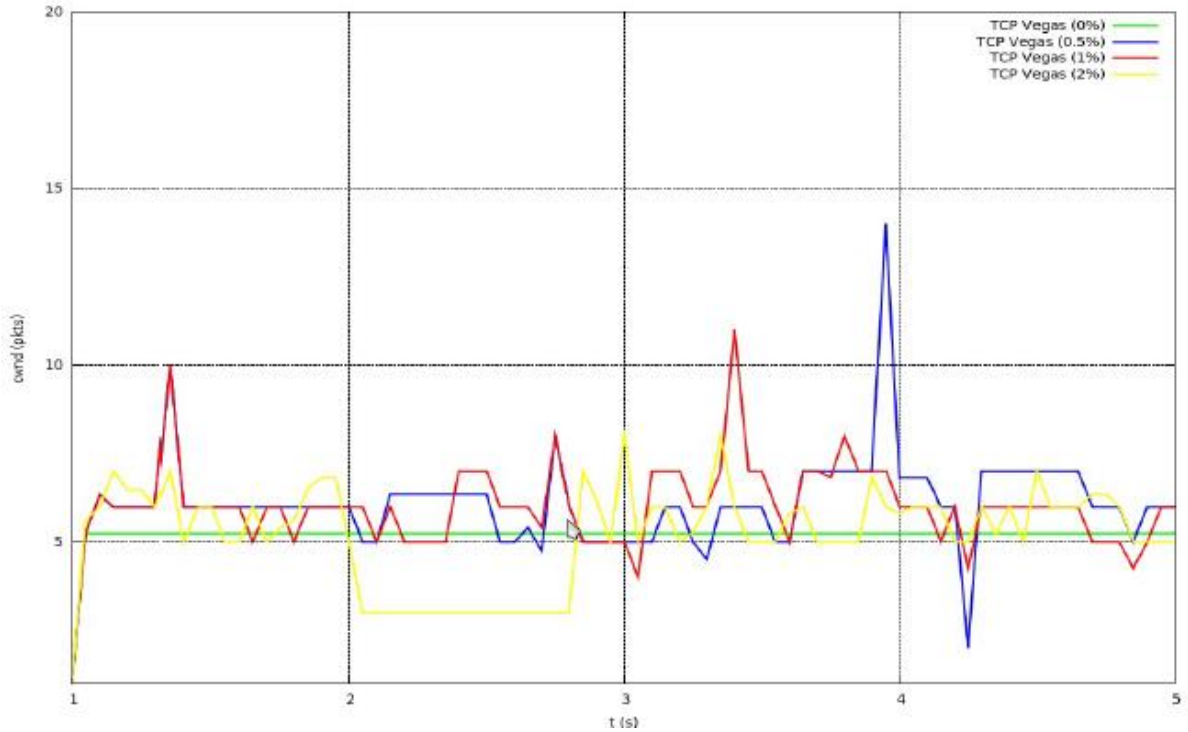

Figura 16 - Desempenho do TCP Vegas frente à interferência
Fonte: http://www.soi.wide.ad.jp/class/20070044/slides/02/index_44.html

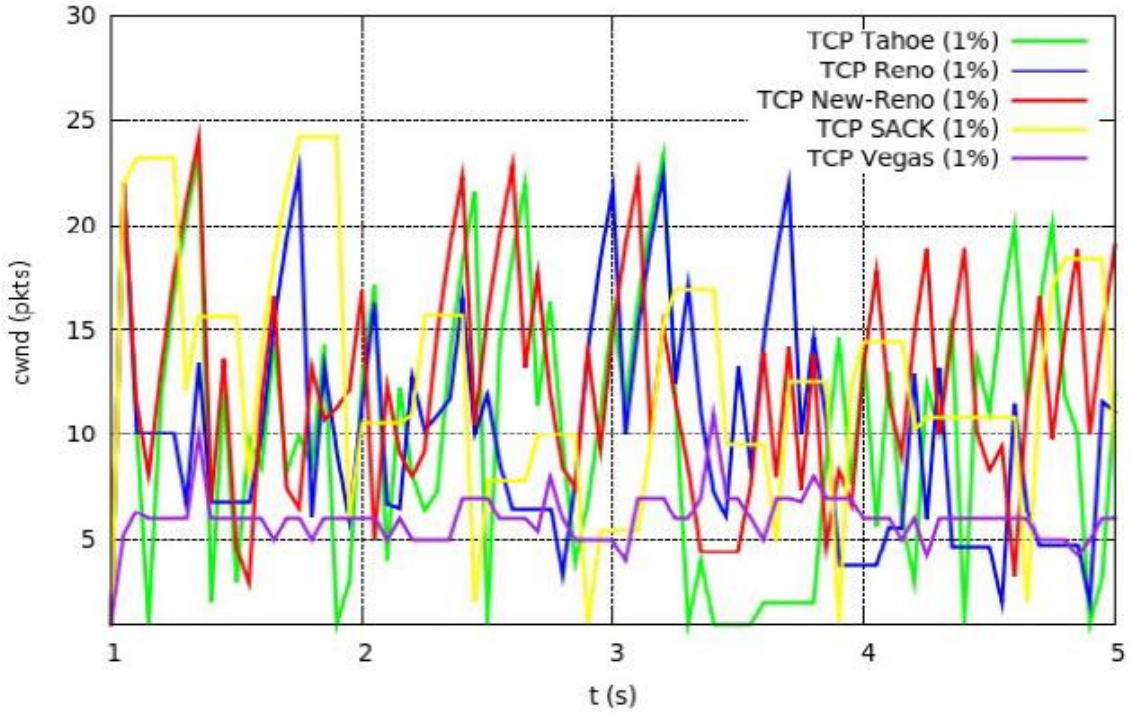

Figura 17 - Comparação de todos os algorítmos a 1% de interferência
Fonte: http://www.soi.wide.ad.jp/class/20070044/slides/02/index_44.html

Observamos uma queda de desempenho quando são introduzidos ruídos no enlace, indicando que todas as abordagens apresentam falhas ao lidar com perdas aleatórias de pacotes, mesmo com 0.5% de perdas (BRAKMO, 1995).

O melhor desempenho foi apresentado pelo TCP SACK. Porém, por diversas vezes, a janela cai a 1 e inicia-se novamente a partida lenta. O TCP Vegas apresentou uma janela relativamente constante mesmo com uma taxa de erros de 2%, apesar de manter a janela pequena. Isto deve-se ao fato do TCP Vegas não utilizar perdas de pacotes como sinalizador de congestionamento.

Estes gráficos demonstram que o TCP pode ter seu desempenho prejudicado em redes sem fio, pois nestes cenários, ruídos são comuns (BRAKMO, 1995).

4. REDES SEM FIO

Redes de comunicação sem fio ou *Wireless Networks* estão sendo cada vez mais empregadas em razão de sua excelente mobilidade, pois em determinados ambientes, as redes cabeadas convencionais não é a melhor opção a ser escolhido devido há limitações físicas existentes. O padrão IEEE 802.11 é conhecido popularmente como Wi-Fi (KUROSE;ROSS, 2010).

A comunicação é feita por ondas de rádio. Tanto notebooks quanto computadores de mesa (desktops) podem ter adaptadores de rede Wi-Fi, formando uma rede sem fio. Inúmeros equipamentos estão disponíveis para este tipo de rede.

Dois dos grandes pilares da tecnologia de redes sem fio são representados pela portabilidade e a praticidade; garantindo menores custos de operação e implantação, pois proporcionam mais facilidade em sua operação e menos tempo em sua implantação (KUROSE;ROSS, 2010).

Redes sem fio são consideradas extensões de redes cabeadas. Para que isso ocorra de maneira eficaz é necessário que haja padronização. Sendo assim faz-se uma perfeita integração com redes cabeadas convencionais possibilitando conectividade entre as redes sem alterar drasticamente a infraestrutura já implementada.

Existem dois tipos de redes sem fio. As redes de infraestrutura e redes ad-hoc. Nas redes de infraestrutura, tem-se um *Access Point* (AP) que concentra todos os pacotes. Qualquer nó da rede irá sempre encaminhar os pacotes para o AP, que irá fazer o roteamento. Já nas redes ad-hoc, cada nó faz o seu próprio roteamento, permitindo que os nós se comuniquem entre si sem intervenção de um AP (KUROSE;ROSS, 2010).

4.1. Características das redes sem fio

As redes sem fio apresentam características diferentes das redes cabeadas. Estas características devem ser consideradas no desenvolvimento do protocolo TCP, pois influenciam no desempenho da comunicação de dados (KUROSE; ROSS, 2010).

4.1.1. Colisões

Apesar da utilização do protocolo CSMA/CA, que tem como objetivo evitar colisões, terminais ocultos podem gerar essas colisões (KUROSE; ROSS, 2010).

4.1.2. Desvanecimento do sinal

O sinal transmitido pode ter sua intensidade diminuída por obstáculos, pela distância percorrida e até mesmo por reflexões. O desvanecimento de sinal pode causar perdas de pacotes pois o sinal pode não ser reconhecido pelo receptor.

4.1.3. Mobilidade

Em redes 802.11 com mais de um AP, o usuário pode movimentar-se entre diferentes áreas de cobertura. A mudança do controle de um AP para outro se chama *handoff*. Pacotes podem ser perdidos ou entregues fora de ordem neste procedimento.

4.1.4. Energia

Equipamentos sem fio precisam ser eficientes no uso da energia para maximizar a duração da bateria (devido ao consumo de processamento do equipamento concentrador). Para que isto não ocorra, devem ser evitadas retransmissões desnecessárias de pacotes.

4.1.5. Half Duplex

As redes 802.11 são *half duplex*, ou seja, enquanto um nó transmite não pode ao mesmo tempo receber dados. Nas redes cabeadas a transmissão e recepção simultâneas são possíveis.

4.2. Perdas de pacote em redes sem fio

As implementações do TCP apresentadas no capítulo 2 assumem que as perdas de pacotes são decorrentes exclusivamente de congestionamentos na rede. Em redes cabeadas, normalmente este é o caso, porém em redes sem fio as perdas de pacotes podem ocorrer por diferentes motivos e de diferentes formas, como explicado a seguir.

É normal em redes Wi-Fi a perda de pacotes aleatórios devido a colisões e interferências. Já as perdas em rajada podem ocorrer durante interferências prolongadas, causando a perda de vários pacotes sem que tenha havido congestionamento. Como explicado anteriormente, pacotes podem pegar caminhos diferentes devido à *handoffs*. Logo, é normal que os pacotes cheguem ao seu destino fora de ordem (FILIPPETTI, 2008).

4.3. Protocolo CSMA/CA

Em redes padrão 802.11 é usado o protocolo CSMA/CA como protocolo MAC. CSMA/CA significa acesso múltiplo com detecção de portadora e prevenção de colisão. Cada estação “sonda” o canal antes de transmitir e não transmite quando percebe que o mesmo está ocupado (KUROSE; ROSS, 2010).

Diferentemente do CSMA/CD, este protocolo tem como objetivo evitar as colisões ao invés de só detectá-las. Isto se deve às razões a seguir:

- Para detectar colisões, os terminais devem ter a capacidade de enviar o seu próprio sinal e detectar se alguma estação está transmitindo no momento. Como o sinal enviado é normalmente muito mais forte que o recebido, é caro construir um equipamento para detectar colisões.
- Os nós podem não detectar colisões devido a terminais escondidos, como mostrado na figura 14 abaixo:

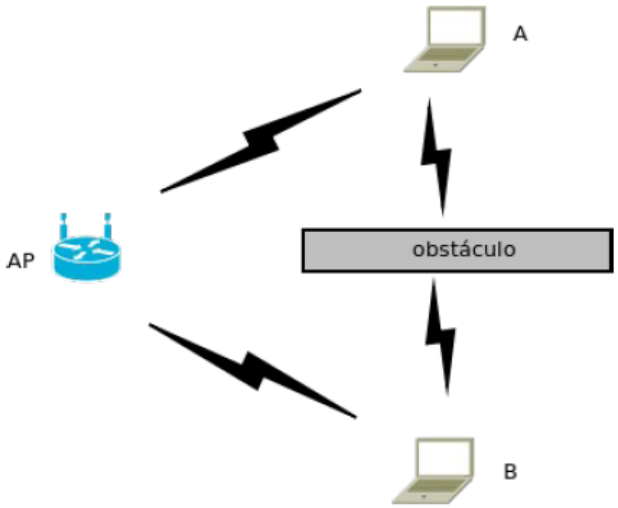

Figura 14 – Obstáculo entre dois nós (terminais ocultos)
Fonte: http://www.cisco.ct.utfpr.edu.br/material/CCNA5.0/

- O nó A não consegue receber sinal do nó B e vice-versa. Neste caso, é impossível para ambos os nós detectarem qualquer tipo de colisão.
- Devido à distancia entre dois nos, é possível que o sinal de um nó A que está transmitindo chegue ao AP mas não chegue a outro nó B mais distante. Assim, é possível que o nó B não detecte o sinal de A e transmita simultaneamente. É aí que entra o sistema RTS/CTS que vemos a seguir.

4.3.1. RTS/CTS

O padrão 802.11 possui um sistema de reserva de canal opcional que ajuda a evitar colisões mesmo na presença de terminais ocultos e desvanecimento de sinal.

Uma estação antes de efetivamente transmitir o quadro de dados, transmite um quadro de controle RTS, que carrega uma estimativa de duração de tempo da futura transmissão do quadro de dados (fig. 8).

A estação de destino em resposta ao quadro de RTS envia um quadro de controle CTS avisando que está pronta para receber o quadro de dados. Só então a estação transmissora envia o quadro de dados, que deve ser respondido com um reconhecimento (ACK) enviado pela estação receptora.

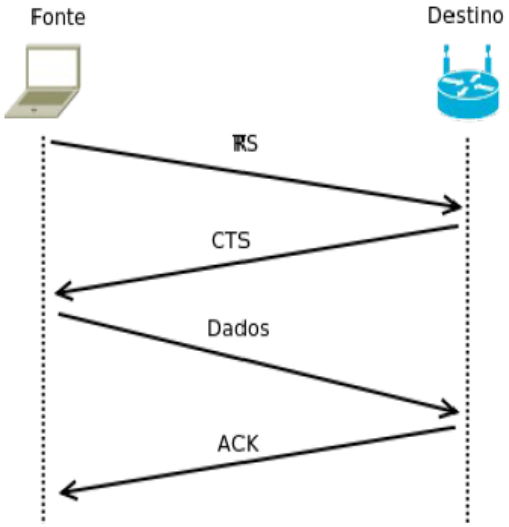

Figura 15 - RTS/CTS
Fonte: sshttp://www.cisco.ct.utfpr.edu.br/material/CCNA5.0

O quadro RTS basicamente possui as funcionalidades de reservar o meio de transmissão do quadro de dados e de verificar se a estação de destino está pronta para receber o quadro, sendo esta última funcionalidade devido à possibilidade da estação de destino estar operando no modo de economia de energia.

Os quadros RTS e CTS só são utilizados quando quadros longos vão ser transmitidos, diminuindo o *overhead* que causaria se fossem usados para todos os pacotes. O problema de terminal oculto é atenuado visto que um quadro longo é transmitido somente quando o canal é reservado (KUROSE; ROSS 2010).

5. O PADRÃO 802.11

Em 1997, foi formulado o primeiro padrão IEEE 802.11, atuando na faixa de 2,4 GHz e com taxas de 1 a 2 Mbps. Devido a reclamações constantes por parte dos usuários de que o padrão era muito lento, em 1999 começaram a surgir variações do padrão, como a proposta do 802.11a. Atingia a velocidade de 54 Mbps sob a frequência de 5 GHz.

Desenvolveu-se o padrão 802.11b, mudando sua técnica de modulação, e foi atingido velocidade de 11 Mbps teóricos sob a frequência de 2,4 GHz. Surgiu outra variação que utilizava a mesma técnica de modulação do 802.11a e a mesma frequência do 802.11b, alcançando 54 Mbps, denominado 802.11g.

Também em 1999, definiu-se uma norma denominada "*Wireless LAN Medium Access Control and Physical Layer Specifications*". O padrão 802.11, assim como todos os padrões da família, determina as camadas PHY e MAC. Padrão que continua em desenvolvimento constante (FILIPPETTI, 2014)

Em 2009, com o objetivo de atingir velocidades superiores as redes cabeadas de 100 Mbps, surgiu à proposta do padrão 802.11n. Para atingir a velocidade planejada, o padrão 802.11n utiliza a tecnologia MIMO juntamente com melhoras em algoritmos de transmissão. Sendo assim, a velocidade teórica passa a ser de 300 Mbps (FILIPPETTI, 2014).

5.1. Controle de acesso ao meio

Para que possam existir redes sem fio com vários computadores, foi desenvolvido um protocolo de controle de acesso ao meio. Responsável por evitar colisões de pacotes entre máquinas que estejam utilizando o mesmo canal, é implementado na camada MAC.

Nas comunicações sem fio, o mecanismo de acesso ao meio é chamado de DCF. Esse mecanismo é baseado no CSMA/CA. O método de acesso ao meio pode ser descrito da seguinte forma: um dispositivo que deseja transmitir escuta o meio, se estiver livre a transmissão ainda sim pode ser rejeitada. O dispositivo só poderá

transmitir quando o meio estiver livre por um intervalo de tempo DIFS, então transmite o pacote. Se o meio estiver ocupado o dispositivo terá que esperar um tempo de DIFIs e entrar numa fase de contenção. É escolhido um *backoff time* aleatório dentro de uma janela de contenção, então tenta acessar novamente o meio depois desse tempo aleatório. Se o meio estiver ocupado novamente, terá que esperar mais um tempo DIFS, em que o meio esteja livre, e repetir o processo.

Após todo processo para garantir disponibilidade do meio, se o meio estiver livre, primeiramente é transmitida uma solicitação para transmissão RTS. Após recebimento do RTS, o receptor envia um CTS (Clear to Send), permitindo o remetente enviar o pacote. Com o recebimento do CTS o transmissor espera um tempo de SIFS (*Short Inter Frame Space*) e envia o pacote. O receptor verifica se o pacote esta correto e sem erros, aguarda um tempo SIFS e envia um pacote ACK. Sendo assim o transmissor, ao receber o ACK, sabe que o pacote foi transmitido com sucesso. Se o ACK não for recebido o transmissor enviará novamente o pacote.

6. NOVAS ABORDAGENS

A seguir são apresentadas algumas abordagens diferentes de controle de congestionamento no TCP.

O *Freeze*-TCP e o ILC-TCP são abordagens de suspensão de estado, ou seja, detectam o estado da rede para decidir quando a comunicação deve ser suspensa e quando deve ser retomada. Já o TCP *Westwood* e o TCP-*Peach* utilizam a abordagem de detecção de colisão, medem as condições da rede para determinar se ocorreu congestionamento ou não. Uma terceira abordagem e chamada de abordagem de atraso de resposta, na qual o cliente TCP atrasa a ativação de controle de tráfego para aliviar os problemas em redes sem fio.

Já o ATCP utiliza uma abordagem híbrida, que envolve as três abordagens apresentadas (LEUNG, 2006).

6.1. Freeze-TCP

É uma solução de suspensão de estado implementada no lado do destino e não do remetente. O nó móvel monitora o sinal recebido para detectar um *handoff* iminente. Neste momento, ele envia ZWA para forçar o remetente a entrar no "modo persistente" aproximadamente um RTT antes de ocorrer o *handoff*. Em seguida, são enviados ZWPs para o destino. Quando os ZWPs são respondidos com uma *advertisement window* positiva, o recebedor sai do modo persistente e retorna a operação normal (LEUNG, 2006).

O *Freeze*-TCP tem alguns problemas:

- Para comunicação entre camadas (potência do sinal para camada de transporte), são necessárias alterações na pilha de protocolos.
- O nó móvel precisa prever quando uma desconexão pode ocorrer.
- Falha ao identificar perda de segmento(s) aleatório(s).
- Não há garantias de que a largura de banda disponível no novo link após o *handoff* seja o mesmo do link anterior. Se for menor, poderá gerar congestionamento.

6.2. ILC-TCP

É uma solução do lado do servidor para prevenir deterioração de desempenho devido a desconexões temporárias. As decisões são definidas de acordo com um gerenciador de estados. Este gerenciador guarda as informações da rede passadas através das camadas.

Especificamente, a camada de enlace reporta ao gerenciador de estados o estado de um link quando ele muda. Um estado ruim indica que um *handoff* é iminente. No *timeout* de um segmento, o remetente primeiro verifica com o gerenciador de estados se a conexão está estável. Se estiver, ele assume que a rede está congestionada e ativa os mecanismos de controle de congestionamento padrões do TCP. Caso contrário, é considerado que houve uma desconexão temporária e a conexão é congelada.

Quando a conexão estiver estável novamente, a conexão é recuperada (LEUNG, 2006).

6.3. TCP Peach/TCP-Peach+

O TCP-Peach foi desenvolvido para redes de satélites com longos *delays* e alta taxa de erros. São introduzidos dois novos algoritmos, *sudden start* e *rapid recovery*. Segmentos *dummy* são usados como sonda da largura de banda disponível. Um segmento *dummy* recebido com sucesso significa que ainda há recursos disponíveis na rede. Bits não usados no cabeçalho do TCP são usados para marcar os segmentos como *dummy* (LEUNG, 2006).

Quando o remetente recebe o ACK, ele incrementa o valor de cwnd em um se wsnd for zero; wsnd é um contador que controla se a janela de congestionamento deve aumentar ao receber um ACK. A *sudden start* substitui a partida lenta. Seu objetivo é abrir a janela de congestionamento muito mais rapidamente. A *sudden start* inicia com cwnd igual a 1 e wdsn igual 0.

Depois que o primeiro segmento é enviado, é transmitido um segmento dummy a cada awnd ate awnd-*1* segmentos *dummy* terem sido enviados e o RTT

estimado da conexão e awnd é o tamanho da advertisement window em segmentos. Deste modo, cwnd pode aumentar até o valor máximo em um RTT. Após este ponto, inicia-se a prevenção de congestionamento.

A *rapid recovery* substitui a recuperação rápida. Na perda de um segmento, cwnd é dividido por 2. WDSN é definido como w/2, sendo que w é o valor de cwnd antes da perda. Quando um ACK chega, a fonte envia dois segmentos *dummy* até o total de w segmentos *dummy* terem sido enviados. Ao receber um ACK para um segmento *dummy*, wdsn é decrementada até chegar a 0. Para qualquer ACK que chegar depois que wdsn=0, cwdn é incrementada. Uma desvantagem deste algoritmo é que os segmentos *dummy* não carregam dados úteis (LEUNG, 2006).

Para melhorar o uso da rede, foi desenvolvido o TCP Peach+. Neste algoritmo, os segmentos *dummy* são substituídos por segmentos *NIL*, que carregam dados ainda não reconhecidos. *Sudden start* e *rapid recovery* são substituídos por *jump start* e *quick recovery*. *Jump start* é como *sudden start*, exceto que segmentos NIL são usados ao invés de segmentos *dummy*.

O *quick recovery* usa SACK (*selective* ACK) para ajudar na retransmissão de segmentos perdidos. Ao receber um ACK na fase de *quick recovery*, um segmento NIL só pode ser enviado depois que não há mais segmentos a serem enviados. A fase de *quick recovery* termina quando um ACK reconhece todos os segmentos enviados antes da fase de recuperação iniciar. Este algoritmo precisa de alterações nos roteadores para descartar primeiro os segmentos *dummy* ou NIL (LEUNG, 2006).

6.4. TCP Westwood

Em outras abordagens, a janela de congestionamento é ajustada sem levar em consideração o nível de congestionamento da rede. O TCP *Westwood*, implementado no lado do remetente, ajusta a janela de congestionamento monitorando a taxa de pacotes reconhecidos. A cada ACK recebido, é feito uma estimativa da largura de banda disponível baseada na quantidade de dados reconhecidos.

Quando é recebido um ACKn no tempo tn, a largura de banda para aquele ACK e calculada por:

$$b_n = \frac{L_n}{t_n - t_{n-1}}$$

O TCP *Westwood* apresenta um problema quando ocorre compressão de ACKs. Compressão de ACKs ocorre quando os ACKs chegam ao destino com um espaçamento menor do que foram enviados devido a enfileiramentos nos roteadores. Isto faz com que o TCP *Westwood* calcule erroneamente a largura de banda disponível. Para resolver este problema, foi desenvolvido o TCP *Westwood*+, que ao invés de estimar a largura de banda a cada ACK recebido, faz isso a cada RTT (LEUNG, 2006).

6.5. ATCP

A idéia do ATCP é introduzir uma camada entre o TCP e o IP. Assim, o ATCP pode monitorar o TCP e mudar de estado quando necessário. Apesar de ser um algoritmo voltado para redes ad hoc, é o único que tenta resolver todos os problemas de redes sem fio, e vale a pena ser estudado. Há quatro estados: normal, congestionado, perda e desconectado. Inicia-se no estado normal. Quando ocorre congestionamento, o roteador define a *flag* ECN nos pacotes. Também uma mensagem ICMP (*source quench*) pode ser enviada (fig.19).

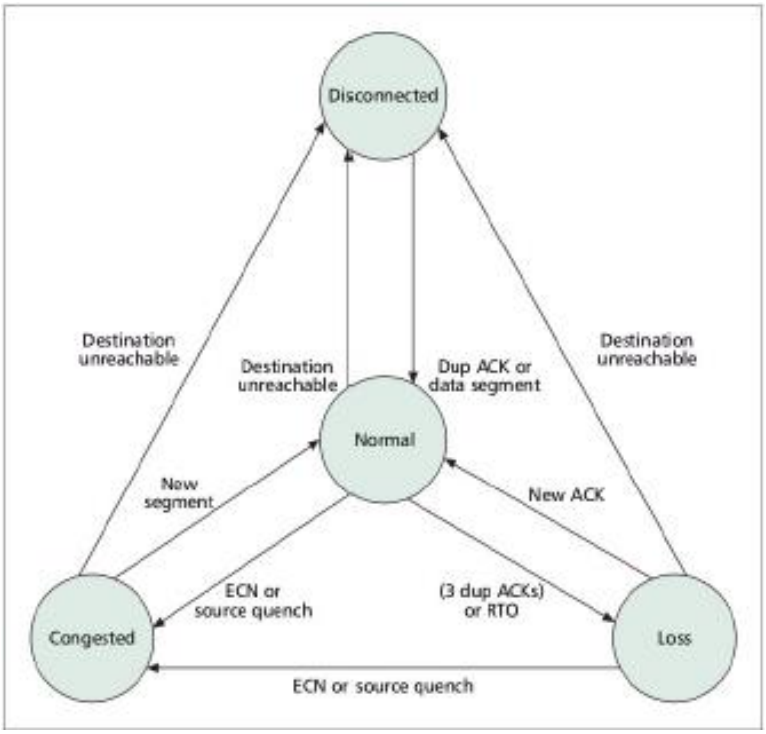

Figura 18 - Diagrama de funcionamento do ATCP
Fonte: http://www.soi.wide.ad.jp/class/20070044/slides/02

Quando o TCP recebe uma dessas duas mensagens, muda para o estado de congestionamento e não interfere no comportamento do TCP. Ele volta ao estado normal depois que um novo segmento é enviado. Quando há 3 ACKs duplicados ou o *timer* expira, a rede está com perdas ou há reordenação de pacotes. Neste caso, o ATCP entra no modo persistente e no estado de perda. São enviados segmentos não reconhecidos. Quando um novo ACK é recebido, o ATCP volta para o estado normal.

O ATCP entra no modo persistente e no estado desconectado quando recebe uma mensagem ICMP *destination unreachable* (destino inalcançável). Isso indica que a rede está instável devido à mobilidade. A fonte gera pacotes de sonda e ao receber um ACK, é retornado para o estado normal. A partida lenta é ativada pois a largura de banda pode ter mudado (LEUNG, 2006).

7. CONCLUSÃO

Neste trabalho foi analisado o desempenho das versões do TCP mais utilizadas em redes sem fio. Primeiramente foi feito um estudo teórico do TCP e os seus possíveis pontos falhos. Em seguida, foi mostrado o comportamento destas diferentes versões.

No comportamento relacionado à interferência, percebeu-se que estes algoritmos tem dificuldades de tratar estes eventos comuns em redes sem fio.

O melhor desempenho apresentado foi o TCP Vegas, que mostrou-se um protocolo mais conservador. O TCP Vegas conseguiu manter a janela estável por mais tempo do que as outras abordagens, apesar de nao ter conseguido aumenta-la para mais do que 5 MSS.

Tambem foi verificada uma grande quantidade de colisões numa rede sem fio e uma tendência para aumento exponencial destas colisões conforme aumenta-se a rede. Conclui-se que nenhum protocolo teve um desempenho satisfatorio nas redes sem fio, motivando o desenvolvimento de algumas alternativas. Estas novas versões do TCP foram especialmente desenvolvidas para sanar os pontos falhos dos TCPs tradicionais em redes sem fio.

Percebeu-se que a maioria destes novos algorítmos são muito específicos, tendo um bom desempenho considerando apenas alguns dos problemas das redes sem fio, com excecao do ATCP. Apesar do ATCP ter como foco as redes *ad hoc*, é o único que visa tratar todas as falhas introduzidas por redes sem fio, sendo um bom ponto de referência para o desenvolvimento de um protocolo igualmente completo para redes de infraestrutura. Com isso, pode-se concluir que ainda não há nenhum algoritmo que trate todos os eventos de rede sem fio, sendo uma área em potencial para o desenvolvimento de novas pesquisas.

REFERÊNCIAS BIBLIOGRÁFICAS

LEUNG, Ka-Cheong; LI, Victor O. K., 2006, **Transmission Control Protocol (TCP) in wireless networks: issues, approaches, and challenges**, IEEE communications Survey.

JACOBSON, Van. **Congestion avoidance and control,** ACM Computer Communication Review, Stanford: University of California: 1988.

KUROSE, James; ROSS, Keith. **Redes de Computadores e a Internet: uma abordagem Top-Down**, 5ª edição, São Paulo: Addison Wesley, 2010.

COMER, Douglas E. **Interligação em rede com TCP/IP**, 1ª edição, Rio de Janeiro: Editora Campus, 1998.

TANENBAUM, Andrew S. **Redes de computadores**, 4ª edição, Rio de Janeiro: Editora Campus, 2003

COMER, Douglas E. **Redes de computadores e internet**, 4ª edição, Porto Alegre: Bookman, 2007.

FILIPPETTI, Marco Aurélio. **CCNA 5.0 Guia Completo de Estudo**, 1ª edição, São Paulo: Editora Visual Books, 2014

ITU Telecommunication Standardization Sector. Disponível em: http://www.itu.int/ITU-D/asp/Events/ITU-BSNL-India/presentations/17-1-Internet%20I.pdf. Acesso em : 20/04/2014

JUN, Takei, 2008, 802.11 specifications, disponivel em <http://www.soi.wide.ad.jp/class/20070044/slides/02/index_44.html>, acesso em 03/09/2013

University of California, 2002, A Comparative Analysis of TCP Tahoe, Reno, New-Reno, SACK and Vegas, disponível em http://inst.eecs.berkeley.edu/~ee122/fa05/projects/Project2/SACKRENEVEGAS.pdf, acesso em 23/05/2014.

ROSA JUNIOR, Laerte Claudemir da. **“Monografia: Funcionamento e implantação do protocolo TCP**”. Campus Curitiba: UTFPR, 2007.

ROESLER, Valter. **Redes de Computadores** – modelo OSI e TCP/IP. São Leopoldo: Universidade do Vale do Rio dos Sinos, 2004.

SOUZA, Lindeberg de. **Projetos e Implementação de redes: fundamentos, soluções, arquiteturas e planejamento**, 1ª edição, São Paulo: Érica, 2007.

.