

NGWA: Esquema de Controle de Congestionamento para TCP Baseado na Banda Disponível

Marcos Talau e Emilio Carlos Gomes Wille

Resumo—O controle de congestionamento padrão do TCP apresenta vários problemas, como a redução enganosa da taxa de transmissão, e a necessidade contínua de informação de perdas de pacotes como indicativo de congestionamento da rede. Neste trabalho é apresentado o *new general window advertising (NGWA)*, que é um novo esquema de controle de congestionamento para o TCP. O *NGWA* traz informações da banda disponível da infraestrutura de rede para os pontos finais da conexão TCP. Seu desempenho foi comparado com TCP New Reno e o TCP padrão. O novo método demonstrou ser superior aos comparados, sendo compatível na ausência de congestionamento, justo com diferentes fluxos, além de ser ausente de perdas.

Palavras-Chave—Controle de Congestionamento, TCP, Produto Banda-Atraso, GWA.

Abstract—The TCP congestion control mechanism in standard implementations presents several problems, for example, a misleading reduce of the transmission rate, and the need of packet losses as an indication of network congestion. This paper presents the *new general window advertising (NGWA)*, which is a new congestion control scheme for TCP. The *NGWA* provides information considering the available bandwidth of the network infrastructure to the endpoints of the TCP connection. Results obtained by the *NGWA* approach were compared with those obtained by TCP New Reno and standard TCP. The new method proved to be superior when compared with the traditional approaches. It ensures fairness, zero packet losses and backward compatibility.

Keywords—Congestion Control, TCP, Bandwidth-Delay Product, GWA.

I. INTRODUÇÃO

O protocolo de controle de transmissão (*transmission control protocol* - TCP) é o protocolo de transporte mais importante para a Internet [1] [2] [3]. Ele é utilizado em uma grande quantidade de redes de computadores, pois traz uma forma simples de entrega de dados e contém estratégias de retransmissão e adaptação que são ocultas para as camadas superiores, isso é suficiente à maior parte das aplicações [4]. Em torno de 80 a 90 por cento de todo o tráfego de dados que circula pela Internet é transmitido pelo protocolo TCP [5]. Segundo [6] pode haver perda de pacotes e falhas de transmissões no nível físico de uma rede. Os dados podem ser perdidos, duplicados ou interrompidos sem chegar ao seu destino. O TCP assume que todas as perdas de transmissão são

causadas pela ocorrência de um congestionamento, ou seja, se ocorrer uma perda de pacotes devido a um problema no enlace (por exemplo, sem fio) o TCP irá reagir como se isto fosse um congestionamento. Um congestionamento real ocorre quando existem mais dados em uma rede do que ela pode suportar; a solução lógica para resolver este problema corresponde à diminuição da taxa de transmissão de dados [7]. Além disso, o protocolo TCP apresenta outros problemas, como a redução enganosa da taxa de transmissão, e a necessidade contínua de informação de perdas de pacotes como indicativo de congestionamento da rede.

Neste trabalho é proposto um novo esquema de controle de congestionamento para o TCP, denominado *new general window advertising (NGWA)*. O *NGWA* traz informações da banda disponível da infraestrutura de rede para os pontos finais da conexão TCP. O trabalho está organizado da seguinte forma: A Seção II apresenta o modelo de transmissão do TCP e problemas relacionados. Na Seção III é introduzido o novo esquema e são descritos detalhes de implementação do *NGWA*. Resultados de simulação obtidos com o *software NS-3* [8] são analisados na Seção IV. As conclusões são apresentadas na Seção V.

II. TCP – PRINCÍPIOS E PROBLEMAS

O protocolo TCP utiliza dois níveis para determinar o tamanho da rajada de dados que irá transmitir: o campo janela (*advertised window, W_a*) do cabeçalho TCP e a janela de congestionamento (*congestion window, W_c*), usados para o controle de fluxo e de congestionamento, respectivamente. O controle de fluxo tem a tarefa de não enviar dados além do que o receptor é capaz processar, levando em consideração o valor do campo janela do segmento recebido. O controle de congestionamento é uma medida que tenta prever a quantidade de dados que a rede irá suportar, e é utilizada pelos dois pontos da conexão. Cada um deles mantém uma janela de congestionamento, baseada em cálculos que tentam prever o estado do meio (rede) que os liga. O W_c surgiu no trabalho de Jacobson como parte do algoritmo *slow-start* [9]. Implementações TCP utilizam a Equação (1) para determinar a quantidade de dados a ser transmitida (*allowed window, W_t*); com isso tenta-se evitar a sobrecarga do receptor e da rede. O W_u é a quantidade de dados já enviados, porém ainda sem confirmação de recebimento.

$$W_t = \min[W_a, W_c] - W_u \quad (1)$$

Muitos problemas ainda impactam o desempenho do protocolo TCP como mostram os parágrafos seguintes.

Segundo Morris [10] a habilidade do TCP em prover um serviço eficiente em um ponto central de tráfego (*bottleneck*) diminui de acordo com o crescimento do número de fluxos. Seu trabalho afirma que para se resolver tal problema deve-se fazer com que o TCP seja menos agressivo e mais adaptativo quando a janela de congestionamento for pequena.

O TCP não funciona bem em redes que apresentam erros frequentes no meio físico (exemplo, redes sem fio), pois ao detectar uma perda de pacotes o protocolo acredita que isso foi devido a um congestionamento, ativando mecanismos para resolver o problema. Com a ativação de mecanismos, como o *slow-start*, haverá uma redução desnecessária de envio de dados [3] [11] [12]. Uma forma bastante pesquisada na literatura para tentar resolver este problema é a utilização do TCP *Snoop*. O *Snoop* teve sua origem no trabalho de Balakrishnan *et al.* [11] e foi pesquisado e expandido em diversos trabalhos [3] [12] [13]. Simplificadamente, a ideia consiste na instalação de um serviço no dispositivo que faz a ligação das redes sem/com fio. Este serviço tem por objetivo armazenar temporariamente pacotes TCP e gerenciar as mensagens TCP do receptor, com isto, atuando no reenvio de segmentos perdidos e na ocultação de segmentos de resposta desnecessários.

Um fator importante que deve ser mantido em dispositivos que trabalham com diversas conexões TCP é a justiça (*fairness*). Conexões com grande RTT (*round-trip time*) que passam por um ponto de tráfego elevado tendem a possuir uma velocidade de transmissão inferior a outras conexões [14]. Em [15] foi proposto um protocolo para controlar o tamanho da janela do TCP, baseando-se no tempo total, porém é necessária a intervenção do usuário no seu ajuste. O esquema *bandwidth aware TCP* (BA-TCP) [16] foi criado para ser usado em enlaces de satélites, e possibilita o envio da informação de RTT na camada de rede (campo do protocolo IPv6). O resultados demonstraram que a implementação fez uma alocação justa de largura de banda, e a vazão (*throughput*) foi três vezes maior que outras implementações TCP.

A adoção de técnicas de gerenciamento ativo de filas (*active queue management* – AQM) visam detectar uma situação de congestionamento antes que a fila do roteador fique cheia, além de prover o repasse da informação da possível ocorrência de congestionamento aos envolvidos na conexão. A notificação explícita de congestionamento (*explicit congestion notification* – ECN), e a detecção aleatória antecipada (*random early detection* – RED) foram criadas com o objetivo de serem usadas com os protocolos TCP/IP [17] [18] [19]. Floyd descreveu os benefícios e problemas na adoção do ECN no TCP; o uso (com o mecanismo DECBIT) permite o conhecimento de uma situação de congestionamento sem chegar ao ponto de ter que descartar pacotes; o ECN apresentou dois problemas: (1) mensagens ECN podem não chegar ao destino, por serem descartadas pela rede, evitando o recebimento da notificação de congestionamento, e (2) problemas de compatibilidade de dispositivos sem

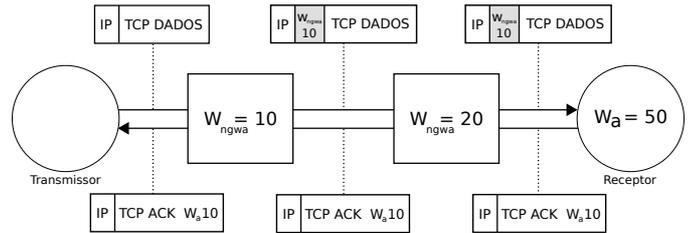


Fig. 1. Mecanismo base do algoritmo NGWA.

suporte a ECN com dispositivos com suporte [20]. O RED tenta evitar possíveis congestionamentos com base no tamanho médio de fila no roteador. A notificação do congestionamento pode ser feita com a marcação de segmentos ou com o seu descarte. Uma desvantagem do RED é a necessidade do ajuste fino de seus parâmetros [17].

Alguns trabalhos tentam resolver mais de um dos problemas acima [16] [21] [22] [23]. Em [21] foi proposto um esquema, chamado *general window advertising* (GWA), para auxiliar no controle de congestionamento. O método faz uso do modelo *end-host-network*, onde o congestionamento é notificado da rede para os pontos finais. Essa notificação é feita pelo repasse da informação de banda disponível para o cabeçalho IPv6 no sentido transmissor-receptor. Caso o valor seja menor que o atual, o receptor irá extrair a informação do protocolo IP e inserir no campo janela do TCP. O GWA demonstrou ser mais eficaz em manter a estabilidade e justiça da rede comparado ao TCP com RED, e o ECN.

III. NEW GENERAL WINDOW ADVERTISING

A implementação TCP por padrão utiliza o campo janela do cabeçalho TCP recebido como parâmetro da rajada de dados a ser transmitida. O NGWA busca trazer até o receptor a quantidade de banda disponível na rede. O número total de *bytes* disponíveis na fila do nó ¹ é armazenado na variável W_{ngwa} , sendo inserida na camada de rede; a variável é armazenada no campo *options* do cabeçalho IPv4. Desta forma é mantida a compatibilidade com o protocolo IP. A atualização da variável é realizada pelos nós da rota TCP. Cada nó pelo qual os pacotes trafeguem irá realizar a atualização da variável; caso $W_{ngwa}(i)$ seja menor que $W_{ngwa}(i-1)$, $W_{ngwa} = W_{ngwa}(i)$. A Figura 1 ilustra de maneira simples o processo. Internamente, quando um pacote chega ao receptor ele é processado normalmente, a variável W_{ngwa} é extraída do cabeçalho IP e depositada na memória. Durante a criação de um segmento ACK a ser enviado ao transmissor, W_{ngwa} é extraída da memória, podendo ser suavizada, sendo inserida no campo janela do segmento TCP. Com isso o receptor TCP saberá a quantidade máxima de *bytes* que a rede suporta naquele instante.

Utilizando o mecanismo base acima, o método proposto neste artigo foi dividido em oito modos (Tabela I). Estes são constituídos de uma combinação das características:

¹ Este termo é genérico fazendo referência a um dispositivo com capacidades de roteamento.

uso da janela de congestionamento, equação de suavização, e divisão por fluxos (DPF).

TABELA I
MODOS DO *NGWA*

	Jan. de Congest.	Eq. Suavização	Div. por Fluxos
1	X	-	-
2	X	X	-
3	-	-	-
4	-	X	-
5	X	-	X
6	X	X	X
7	-	-	X
8	-	X	X

A. Janela de Congestionamento

Quando esta característica estiver ativa uma implementação TCP com controle de congestionamento é utilizada pelo transmissor, neste trabalho foi usado o TCP New Reno. Senão o TCP padrão (RFC 793) [24] é utilizado. A Equação abaixo define a quantidade de dados à transmitir em ambos os casos.

$$W_t = \begin{cases} \min[W_a, W_c] - W_u, & \text{opção ativa.} \\ W_a - W_u, & \text{caso contrário.} \end{cases} \quad (2)$$

A variável W_{ngwa} é utilizada pelo lado receptor, sendo indiretamente aplicada ao transmissor pela utilização do parâmetro W_a . Ele é gerado pelo receptor com o uso da Equação 3.

$$W_a = \begin{cases} \min[W_b, W_{max}, W_{ngwa}], & \text{com } NGWA. \\ \min[W_b, W_{max}], & \text{sem } NGWA. \end{cases} \quad (3)$$

onde, W_{max} corresponde ao tamanho máximo da janela e W_b a quantidade de memória disponível na fila do receptor TCP.

B. Equação de Suavização

A janela do TCP tende a variar constantemente [13]. Desta forma, uma equação para suavização de W_{ngwa} (Eq. 4) foi utilizada, onde $0 < \alpha < 1$.

$$\bar{W}_{ngwa} = (1 - \alpha) \cdot \bar{W}_{ngwa} + \alpha \cdot W_{ngwa} \quad (4)$$

C. Divisão por Fluxos

Por padrão o *NGWA* armazena na variável W_{ngwa} a quantidade total de *bytes* disponíveis na fila do roteador. Este comportamento tende a trazer injustiças, pois um fluxo pode consumir todos os recursos do nó.

A divisão por fluxos (DPF) realiza o compartilhamento dos *bytes* livres na fila do nó pela quantidade de fluxos/conexões ativas. A detecção de fluxos TCP/UDP (*user datagram protocol*) é feita através do registro de origem/destino de endereços IP e portas. Quando esta opção for selecionada, um processo será executado para criar e manter o registro de conexões ativas que trafeguem pelo nó. Desta forma o divisor de banda é atualizado dinamicamente.

IV. RESULTADOS

Os testes executados tiveram por objetivo avaliar o desempenho da proposta. Foram executados testes de taxa de transmissão de fluxos e justiça (*fairness*) de uso de rede, variação da janela de transmissão, e taxa de pacotes perdidos. Duas topologias foram consideradas. A primeira delas (Figura 2) foi baseada no trabalho de [25]. A segunda topologia (Figura 3) é clássica e presente em muitos trabalhos relacionados ([10] [15] [16] [20] [22]). Nesta segunda topologia aplicou-se um tráfego intenso; cada nó do lado esquerdo estabeleceu uma conexão TCP com um nó do outro lado. Transmissões paralelas foram executadas até o término da simulação.



Fig. 2. Topologia I.

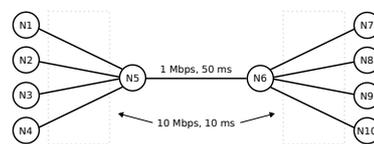


Fig. 3. Topologia II.

Para realizar uma comparação com a proposta foram utilizadas as implementações TCP New Reno e a TCP padrão [24]. Todos os modos do *NGWA* foram avaliados. Os seguintes valores foram considerados: $W_{max} = 64$ *Kbits*; tamanho dos segmentos TCP: 1000 bytes; fila: *drop tail*; capacidade da fila dos roteadores: 97 *KBytes*; constante $\alpha = 0,3$; tempo de simulação: dez minutos. O tráfego da simulação foi gerado com a utilização do modelo *OnOffApplication* do *NS-3*. Este modelo foi configurado para transmitir dados constantemente até o final da simulação a uma taxa de 500 *Kbps*.

1) *Variação da Janela de Transmissão*: Foi avaliada a variação da janela de transmissão para as duas topologias. Foram executados testes com o TCP padrão, TCP New Reno e com o *NGWA*.

Os resultados obtidos das topologias são bem diferentes. A primeira topologia, devido a sua configuração, não deve levar a uma situação de congestionamento; com isso foi testado o comportamento do *NGWA* em uma rede estável. Na Figura 4 pode-se observar a taxa constante da janela de transmissão para as implementações testadas, o que representa uma ausência de perdas.

A segunda topologia possui mais nós e fluxos, com os dados trafegando por um *backbone* limitado a 1 *Mbps*. Foi observado que após os 50 segundos iniciais de simulação não houve grandes mudanças na janela, por isto na Figura 5 é apresentada a transmissão até este ponto. Neste segundo teste, também foi comparada a reação ao encerramento de dois dos quatro fluxos em atividade; este evento ocorreu no instante de tempo de 25 segundos.

Na primeira metade da Figura 5 pode-se observar que o TCP New Reno inicia a janela de transmissão com um

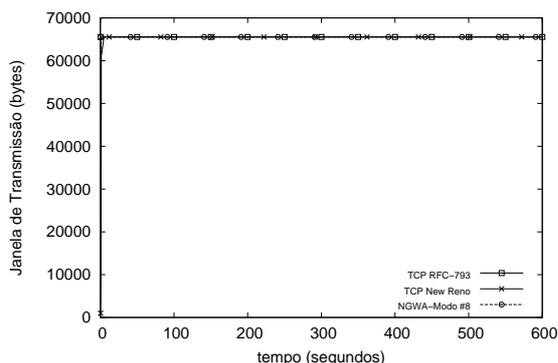


Fig. 4. Variação da janela de transmissão da topologia I.

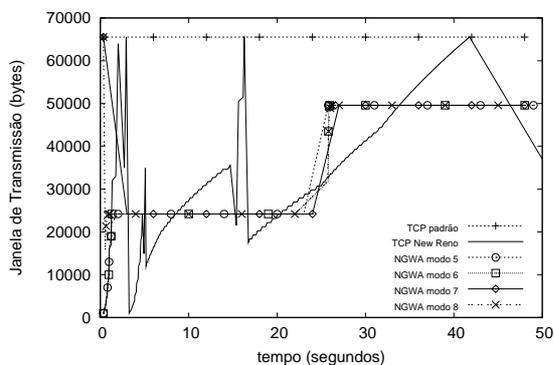


Fig. 5. Variação da janela de transmissão da topologia II. O gráfico apresenta dados para um dos fluxos presentes na topologia II, os demais fluxos apresentaram um comportamento semelhante.

valor baixo, crescendo constantemente até atingir um pico, reduzindo a taxa, permanecendo em oscilação constante. O NGWA modos 5 e 6 seguiram o comportamento inicial do Reno, enquanto os modos 7 e 8 iniciaram com um valor de janela alto; ambos os modos chegaram rapidamente a uma taxa transmissão estabilizada. Após os 25 segundos, com a presença de dois fluxos ativos, o New Reno manteve o comportamento oscilante, porém com uma média de transmissão maior. Os modos do NGWA rapidamente elevaram a taxa de transmissão, mantendo-a constante.

Nota-se que, em geral, o NGWA obteve uma taxa de transmissão estável, com grande justiça (*fairness*) entre os fluxos. O New Reno apresentou injustiça (*unfairness*) entre os fluxos e alta oscilação.

2) *Taxa de Transmissão por Fluxos / Justiça de Uso de Rede:* Neste teste foi comparada a quantidade total de bytes transmitidos por cada fluxo. Com a análise deste teste é possível verificar a justiça do uso de rede.

A topologia I possui apenas um fluxo, por isso não foi utilizada. A Figura 6 exibe a taxa de transmissão dos quatro fluxos da topologia II. Foram testados o TCP padrão, TCP New Reno, e todos os modos do NGWA.

Analisando os resultados percebe-se que o TCP padrão, naturalmente, foi o mais injusto - os fluxos 2 e 4 consumiram toda a banda de rede, deixando os fluxos 1 e 3 com uma baixa transmissão. O TCP New Reno manteve

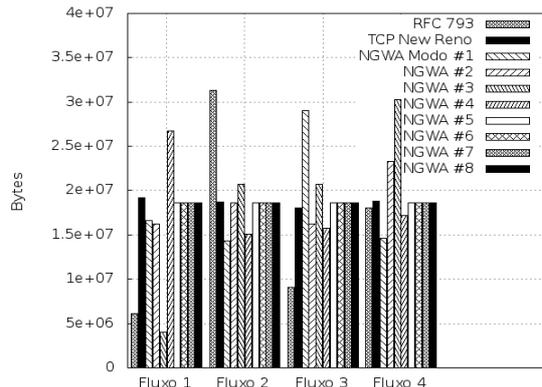


Fig. 6. Taxa total de transmissão dos quatro fluxos da topologia II.

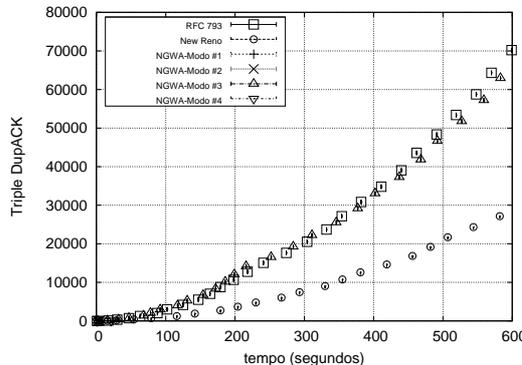


Fig. 7. Acumulo de *triple dupacks* durante a transmissão. Os modos 1, 2 e 4 não são visíveis, pois seus valores foram baixos, ocorrendo no início da transmissão.

os fluxos com uma boa justiça. Os modos 1 e 3 do NGWA foram bastante injustos. Ao contrário, os modos 5, 6, 7 e 8 foram altamente justos, chegando a superar o TCP New Reno. A equação de suavização teve efeito mais positivo nos modos 2 e 4, reduzindo a injustiça presente nos modos 1 e 3.

3) *Taxa de Pacotes Perdidos:* A quantidade de pacotes enviados e não entregues é um importante fator para verificação da qualidade do método de controle de congestionamento, pois, pacotes retransmitidos podem aumentar o congestionamento. Assim como um *timeout*, o recebimento de três ACKs duplos (*triple dupacks*) é considerado um sinal de congestionamento na rede [26].

Durante as transmissões foram coletadas a quantidade total de *triple dupacks* e o número de perdas de todos os nós. Os resultados aqui apresentados são da topologia II. O acumulo de *triple dupacks* ao longo da transmissão é apresentado na Figura 7.

O número de *triple dupacks* do TCP padrão e do NGWA modo 3 foram os mais altos da simulação. Seguidos pelo TCP New Reno (28935). Os modos do NGWA tiveram diferentes desempenhos. O pior caso foi com o modo 3, atingindo 65910 *triple dupacks* recebidos. Seguido pelo modo 4 (382), 1 (30) e 2 (14). A equação de suavização novamente teve um excelente efeito para os modos 2 e 4, tendo um resultado muito superior aos modos 1 e 3.

Os modos 5/6/7/8 não aparecem no gráfico, pois não receberam *triple dupacks*.

Também foi coletada a quantidade de pacotes perdidos (Tabela II). O TCP padrão produziu a maior quantidade de perdas (4001), seguido por: NGWA modo 3 (1674), New Reno (328), modo 4 (308), e modo 1 (7). Os modos 2/5/6/7/8 não produziram perdas.

TABELA II
NÚMERO DE PACOTES PERDIDOS

	Topologia I	Topologia II
RFC 793	0	4001
New Reno	0	328
NGWA Modo #1	0	7
NGWA Modo #2	0	0
NGWA Modo #3	0	1674
NGWA Modo #4	0	308
NGWA Modo #5	0	0
NGWA Modo #6	0	0
NGWA Modo #7	0	0
NGWA Modo #8	0	0

V. CONCLUSÕES

Neste trabalho foi apresentado um novo esquema para controle de congestionamento do protocolo TCP. O esquema denominado *new general window advertising (NGWA)* traz informações do uso de banda da rede para os pontos finais da conexão TCP. O NGWA foi subdividido em oito modos, cada qual com um conjunto de características diferentes. Foram feitos testes de taxa de transmissão e justiça (*fairness*) de uso de rede, variação da janela de transmissão, e taxa de pacotes perdidos.

Os resultados foram comparados com o TCP New Reno e o TCP padrão. O NGWA demonstrou ser compatível na ausência de congestionamento, justo com diferentes fluxos, além de ser ausente de perdas. Comparado ao TCP New Reno e o TCP padrão, o NGWA obteve um desempenho superior em todos os testes, com destaque para o número de perdas, onde os modos 5/6/7/8 não registraram o recebimento de *triple dupacks* e perdas.

O emprego da suavização melhorou o desempenho dos modos 1 e 3. A divisão por fluxos (DPF) produziu resultados extremamente superiores ao demais. Os modos 7 e 8 são os que menos utilizam recursos computacionais, pois não necessitam de controle de congestionamento, como o TCP New Reno, TCP Vegas, ou o TCP Cube. Desta forma o mecanismo de controle de congestionamento pode ser retirado do TCP, como já foi sugerido por [1].

Um problema do TCP ainda não resolvido, envolve as redes sem fio, onde as perdas no meio físico fazem a ativação desnecessária dos mecanismos de controle de congestionamento do TCP. Já pesquisado em diversos trabalhos, como em [3] [11] [12], o problema ainda é investigado. Pode-se aplicar o NGWA para buscar resolver tal problema. Também como trabalho futuro, é possível de se realizar a comparação de resultados obtidos pelo NGWA com métodos *active queue management (AQM)*, como o RED.

REFERÊNCIAS

- [1] S. Jiang, Q. Zuo, and G. Wei, "Decoupling congestion control from TCP for multi-hop wireless networks: semi-TCP," in *CHANTS '09: Proc. of the 4th ACM workshop on Challenged networks*. New York, NY, USA: ACM, 2009, pp. 27–34.
- [2] E. Weigle and W. chun Feng, "Dynamic right-sizing: a simulation study," 2001, pp. 152–158.
- [3] S. Vangala and M. Labrador, "The TCP SACK-aware snoop protocol for TCP over wireless networks," vol. 4, out. 2003, pp. 2624–2628 Vol.4.
- [4] M. Mathis, J. Heffner, and R. Reddy, "Web100: extended TCP instrumentation for research, education and diagnosis," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 3, pp. 69–79, 2003.
- [5] A. Veres and M. Boda, "The chaotic nature of TCP congestion control," vol. 3, mar. 2000, pp. 1715–1723 vol.3.
- [6] D. Comer, "Internetworking with TCP/IP vol I: principles, protocols, and architecture," *Prentice Hall*, 2000.
- [7] A. Tanenbaum, "Redes de Computadores," *Editora Campus*, 1997.
- [8] The Network Simulator, "NS-3," disponível em <http://www.nsnam.org>.
- [9] V. Jacobson, "Congestion avoidance and control," in *SIGCOMM '88: Symposium proc. on Communications architectures and protocols*. New York, NY, USA: ACM, 1988, pp. 314–329.
- [10] R. Morris, "TCP behavior with many flows," in *ICNP*. IEEE Computer Society, 1997, pp. 205–211.
- [11] H. Balakrishnan, S. Seshan, and R. H. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," *Wirel. Netw.*, vol. 1, no. 4, pp. 469–481, 1995.
- [12] K.-F. Leung and K. Yeung, "G-Snoop: enhancing TCP performance over wireless networks," vol. 1, jun. 2004, pp. 545–550 Vol.1.
- [13] J.-C. Moon and B. G. Lee, "Rate-adaptive snoop: a TCP enhancement scheme over rate-controlled lossy links," *IEEE/ACM Trans. Netw.*, vol. 14, no. 3, pp. 603–615, 2006.
- [14] S. Floyd, "Connections with multiple congested gateways in packet-switched networks part 1: one-way traffic," *SIGCOMM Comput. Commun. Rev.*, vol. 21, no. 5, pp. 30–47, 1991.
- [15] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 556–567, 2000.
- [16] M. Gerla, W. Weng, and R. Lo Cigno, "BA-TCP: a bandwidth aware TCP for satellite networks," 1999, pp. 204–207.
- [17] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Netw.*, vol. 1, no. 4, pp. 397–413, 1993.
- [18] K. Ramakrishnan and S. Floyd, "RFC 2481: A Proposal to add Explicit Congestion Notification (ECN) to IP," Jan. 1999.
- [19] K. Ramakrishnan, S. Floyd, and D. Black, "RFC 3168: The addition of Explicit Congestion Notification (ECN) to IP," 2001.
- [20] S. Floyd, "TCP and explicit congestion notification," *SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 5, pp. 8–23, 1994.
- [21] M. Gerla, R. L. Cigno, and W. Weng, "Generalized Window Advertising for TCP Congestion Control," 2002.
- [22] M. Gerla, W. Weng, and R. Lo Cigno, "Bandwidth feedback control of TCP and real time sources in the Internet," vol. 1, 2000, pp. 561–565 vol.1.
- [23] M. Gerla, M. Sanadidi, R. Wang, A. Zanella, C. Casetti, and S. Mascolo, "TCP Westwood: congestion window control using bandwidth estimation," vol. 3, 2001, pp. 1698–1702 vol.3.
- [24] J. B. Postel, "Transmission Control Protocol," September 1981, RFC 793.
- [25] K. Fall and S. Floyd, "Simulation-based comparisons of Tahoe, Reno and SACK TCP," *SIGCOMM Comput. Commun. Rev.*, vol. 26, pp. 5–21, jul. 1996.
- [26] W. Stevens, "RFC 2001: TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," Jan. 1997.