

Controle de congestionamento do protocolo TCP em predição de perda de Pacotes

Taila de França Santos, Henrique Alves Dinarte da Silva, Ricardo Ataíde de Lima e Edison de Queiroz Albuquerque

Resumo— O controle de congestionamento tem sido foco de muitas pesquisas há décadas e diversas versões do protocolo TCP fizeram abordagem desta problemática em suas implementações. O objetivo principal deste artigo é implementar o mecanismo de aceleração da taxa de transmissão dos pacotes, no protocolo TCP POLI [1], em redes com riscos de congestionamento. Para tanto, comportamento da rede foi analisado na ferramenta de simulação NS2, onde foi criado um cenário de rede em que diversas versões do protocolo TCP compartilham do mesmo canal em diferentes áreas de abrangência (LAN, MAN e WAN). Os resultados obtidos mostraram que é possível melhorar o desempenho da rede em relação às versões atuais do protocolo TCP com apenas algumas alterações na base dos algoritmos das versões existentes. Principalmente em relação a parâmetros como: latência, vazão e perda de pacotes.

Palavras-Chave— Controle de Congestionamento, Protocolo TCP, Otimização.

Abstract— The Congestion control has been a focus of researches for decades and a diversity of version of the TCP protocol has done its approach in their implementations. This article's main focus has been to implement the acceleration of the packages transmission rate, on the TCP POLI protocol, in networks without traffic risk. So, network behavior has been analyzed on the simulator NS2, where a network scenario has been created, in which a diversity of versions of TCP protocol share the same channel in different coverage areas (LAN, MAN and WAN). The result showed that it is possible to improve the network performance comparing to the current versions of the TCP protocol with little alterations on the algorithms basis of existing versions. Especially relating to parameters such as: latency, void and package lost.

Keywords— Congestion Control, TCP Protocol, Optimization.

I. INTRODUÇÃO

A internet tornou-se para a população um dos maiores meios de comunicação. Diante desta realidade, é necessário aprimorá-la cada vez mais para que sejam atendidas as necessidades dos usuários de forma eficiente. Entretanto, as redes, na maioria dos casos, apresentam uma quantidade restrita de recursos, como, limitação de largura de banda e do comprimento das filas em que os pacotes são armazenados na espera da transmissão [2].

O controle de congestionamento do protocolo TCP (*Transmission Control Protocol*) possui um papel fundamental para o bom desempenho deste cenário [3]. O mesmo, limita a taxa de transmissão dos pacotes quando a rede for considerada

congestionada [1]. Diante disso, diversas versões do protocolo TCP fizeram abordagem desta problemática em suas implementações. Inicialmente foi criado o TCP versão *Tahoe*, por *Jacobson*, que utiliza da técnica de partida lenta (*Slow Start*) [4]. Quando um pacote na rede é perdido, sua taxa de retransmissão é reduzida, acarretando na baixa utilização do canal.

No intuito de melhorar o TCP *Tahoe*, surgiu a versão TCP *Reno*. A principal diferença entre as versões do protocolo é que o TCP *Reno* trabalha com o *Fast Recovery* que ao detectar a perda retransmite o pacote perdido [5]. Quando um segmento é perdido, o mesmo será retransmitido, e sua confirmação reconhecerá todos os segmentos pertencentes à janela [6]. Porém, quando múltiplos pacotes são perdidos na mesma janela, a confirmação do segmento retransmitido reconhece todos os pacotes transmitidos até a próxima perda [3].

O TCP *New Reno* surgiu como uma atualização do TCP *Reno*. Nesta nova versão, a *Fast Recovery* tem a capacidade de recuperar os dados perdidos sem a necessidade de aguardar o timeout expirar, retransmitindo um segmento a cada *rtt* (*Round Trip Time*) [7].

Dentre as diferentes técnicas explanadas percebe-se uma característica em comum, todas utilizam a perda de pacote como indicador de congestionamento [8]. Nesse contexto surge o TCP *Vegas* que detecta o congestionamento nos roteadores antes de ocorrer uma perda, ajustando a janela de congestionamento de acordo com o *rtt* do pacote [9]. Caso o *rtt* aumente, a rede pode estar congestionando, em consequência a janela de congestionamento é reduzida, de forma a diminuir a transmissão de pacotes antes que haja a perda. Apesar das grandes melhorias, o TCP *Vegas* apresenta algumas limitações em redes que apresentam constantes congestionamentos, muitas alterações no roteamento, dificuldade em trabalhar com outras versões do TCP [9].

Diante do cenário apresentado foi criada a versão TCP *Poli* com soluções eficientes para a problemática do congestionamento. Uma das ferramentas matemáticas utilizadas na nova versão foi a média móvel ponderada que permitiu a mensuração do valor limite inicial da perda de pacote e baseado nessa informação a janela de congestionamento é desacelerada [3]. Porém, assim como as outras versões o TCP *Poli* tem suas limitações. A vazão da rede, por exemplo, não apresentou resultados satisfatórios quando comparada às versões anteriores. Baseado nessas limitações, este artigo se propõe a otimizar o controle de congestionamento no protocolo TCP, realizando ajustes no

TCP *Poli*. Pretende-se criar um mecanismo de aceleração da janela de congestionamento quando a rede for considerada com baixos níveis de congestionamento, assim, parâmetros como a vazão da rede podem ter resultados mais eficientes.

II. TCP POLI

A ideia principal do TCP *POLI* é classificar os níveis de congestionamento da rede e a partir desta análise desacelerar ou acelerar a taxa de transmissão (janela de congestionamento) para que a perda de pacotes não ocorra. O parâmetro utilizado como referência na classificação é a variável *rel* que relaciona a média móvel ponderada dos *rtts* mais recentes (*prtt*) com o *rtt* anterior (*prtt-1*), como pode ser observado na Equação 1.

Segundo estudos realizados por Silva [1], a variável *rel* é considerada, no TCP *Poli*, um indicativo do nível de congestionamento da rede. No caso do primeiro segmento recebido a média móvel atual é igual ao primeiro *rtt* e a média móvel anterior não existe. Diante disso, o algoritmo TCP *Poli* faz este valor ser igual ao primeiro *rtt*, resultando em um valor de *rel* igual a 1, que seria o indicativo de uma rede com o nível de congestionamento constante, valores acima de 1 indicariam uma rede com aumento de congestionamento e valores abaixo, uma rede com redução de congestionamento.

A partir desta análise preditiva serão feitos os ajustes necessários na janela de congestionamento. Se a indicação for rede em congestionamento, a taxa de transmissão é reduzida. Caso contrário, a rede é considerada disponível, não sendo necessária a redução da janela de congestionamento.

$$rel = \frac{prtt}{prtt \text{ anterior}} \quad (1)$$

III. CENÁRIO E AMBIENTE DE SIMULAÇÃO

A. Cenário

O cenário selecionado para realização dos testes foi criado no NS-2 (um simulador de eventos discretos, amplamente utilizado e reconhecido na comunidade acadêmica, capaz de suportar a simulação de tecnologias de rede baseadas no protocolo TCP/IP) [10], sendo caracterizado por apresentar alta concorrência em suas conexões. Nele, foram inseridos 12 transmissores que enviam pacotes a outros 12 receptores compartilhando do mesmo link de rede (caminho e roteadores), como pode ser visto na Figura 1.

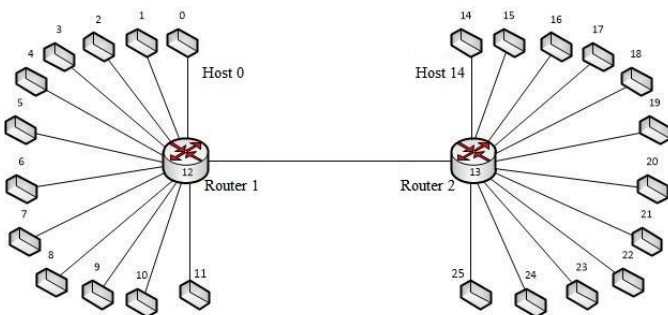


Fig. 1. Cenário de Simulação.

Os computadores conectados ao roteador 1 (*Router 1*) realizam uma transferência de arquivo (FTP – *File Transfer Protocol*) para os hosts do roteador 2 (*Router 2*) com uma duração média de 60 segundos. No intuito de criar um

ambiente com um tráfego de dados elevado e alta concorrência, a largura de banda entre os roteadores utilizada representa apenas 70% da banda necessária para o tráfego [1]. A versão do protocolo em análise (proposta) foi implementada entre o Host0 e o Host14. Nas demais conexões, utilizadas como *Background*, foram configuradas outras versões do protocolo TCP, vide Tabela 1. Assim, realizou-se uma análise comparativa entre o comportamento da configuração C4 com outras versões do protocolo TCP, identificando quais os ambientes em que a atualização do protocolo TCP *Poli* apresenta melhores resultados em relação a performance da rede.

TABELA I. CONEXÕES DE *BACKGROUND*

Conexões	Versão do Protocolo TCP
Host 1 e Host 15	TCP Vegas
Host 2 e Host 16	TCP Reno
Host 3 e Host 17	TCP Reno
Host 4 e Host 18	TCP Reno
Host 5 e Host 19	TCP Reno
Host 6 e Host 20	TCP Reno
Host 7 e Host 21	TCP New Reno
Host 8 e Host 22	TCP New Reno
Host 9 e Host 23	TCP New Reno
Host 10 e Host 24	TCP New Reno
Host 11 e Host 25	TCP New Reno

B. Ambiente

As redes de computadores podem ser dispostas de acordo com a sua área de abrangência. Uma rede LAN (*Local Area Network*), por exemplo, apresenta uma cobertura limitada, ou seja a distância entre os hosts e os roteadores chega em média a 100m [11]. A medida que essa distância entre os dispositivos de rede vai aumentando, uma nova classificação é feita. No intuito de realizar uma análise em diversos ambientes de rede foram testados os cenários LAN, MAN (*Metropolitan Area Network*), WAN (*Wide Area Network*) [12].

O produto entre o atraso de propagação e a velocidade de transmissão dos sinais resulta no alcance da rede. A velocidade de transmissão é conhecida e tem o seu valor próximo ao da velocidade da luz. Assim o atraso de propagação é o parâmetro que ao ser ajustado determina as diferentes distâncias entre os dispositivos da rede [3].

$$\text{LAN} - d = (3 \cdot 10^8) \cdot (0.003 \cdot 10^3) = 300m$$

$$\text{MAN} - d = (3 \cdot 10^8) \cdot (1 \cdot 10^3) = 300Km \quad (2)$$

$$\text{WAN} - d = (3 \cdot 10^8) \cdot (50 \cdot 10^3) = 15000Km$$

IV. PROPOSTA

A implementação da aceleração da janela de congestionamento foi realizada para valores com $rel < 1$. Assim como no TCP *Poli*, os ajustes corretivos respeitaram o limite de 30%. Pois, quando trata-se de análise de performance em redes de computadores, valores acima de 30% podem causar degradação da comunicação[13]. Com isso, aumentamos o valor da janela de congestionamento de acordo com três intervalos de valores da variável *rel*: $0.85 < rel < 1$; $0.7 < rel < 0.85$; e $rel < 0.7$, os quais indicam, respectivamente, os níveis de congestionamento como: pouco baixo, baixo e muito baixo.

TABELA II. CRESCIMENTO DA JANELA DE CONGESTIONAMENTO

Limiar de crescimento da janela	0.85 < rel < 1	0.7 < rel < 0.85	rel < 0.7
C1	cwnd = cwnd * 1.05	cwnd = cwnd * 1.1	cwnd = cwnd * 1.2
C2	cwnd = cwnd * 1.05	cwnd = cwnd * 1.1	cwnd = cwnd * 1.15
C3	cwnd = cwnd * 1.1	cwnd = cwnd * 1.2	cwnd = cwnd * 1.3
C4	cwnd = cwnd * 1.1	cwnd = cwnd * 1.3	cwnd = cwnd * 1.5
C5	cwnd = cwnd * 1.1	cwnd = cwnd * 1.6	cwnd = cwnd * 1.7
C6	cwnd = cwnd * 1.5	cwnd = cwnd * 1.7	cwnd = cwnd * 1.9

Após detectar os intervalos, o valor ideal para o crescimento da janela de congestionamento é determinado a partir de testes com variados limiares de crescimento percentual da janela, como pode ser visto na Tabela 2. Após analisar parâmetros como vazão, perda de pacote e latência, verificou-se que o limiar representado pela configuração C4 obteve um melhor desempenho em relação aos demais.

V. RESULTADOS

A seguir serão mostrados os resultados da análise comparativa das diferentes versões do protocolo nos ambientes de rede citados anteriormente.

A. Ambiente LAN

TABELA III. AMBIENTE LAN

Indicador	Tahoe	Reno	New Reno	Vegas	Poli	C4	Melhor
Vazão Média(kbps)	636,19	912,93	672,81	581	521	897	C4
Latência Média(ms)	2,92	2,41	2,81	2,97	8,91	2,24	C4
Pacotes Enviados	542	759	566	511	418	730	New Reno
Pacotes Perdidos	40	41	33	33	9	26	Poli
Perda Percentual (%)	7,38	5,4	5,83	6,45	2,15	3,56	Poli

Os resultados mostram, na Tabela 3, que a configuração C4 adquiriu valores de Latência e Vazão melhores do que as demais versões, sendo a diferença da vazão bastante considerável em relação aos outros algoritmos. Em termos de perda percentual, o desempenho obtido foi ligeiramente próximo ao TCP *Poli* que se apresenta melhor do que os demais neste parâmetro.

B. Ambiente MAN

TABELA IV. AMBIENTE MAN

Indicador	Tahoe	Reno	New Reno	Vegas	Poli	C4	Melhor
Vazão Média(kbps)	854	768	491	856	711	906	C4
Latência Média(ms)	2,11	2,34	3,44	2,02	2,62	2,41	Vegas
Pacotes Enviados	700	670	428	718	590	755	C4
Pacotes Perdidos	29	64	40	26	30	28	Vegas
Perda Percentual (%)	4,12	9,55	9,34	3,62	5,08	3,71	Vegas

Os resultados mostrados na Tabela 4 demonstram que a configuração C4 obteve um desempenho de Vazão melhor do que as demais versões, sendo a diferença considerável em relação aos demais algoritmos. Em termos de perda percentual e latência, o desempenho obtido foi ligeiramente próximo ao TCP *Vegas* que se apresenta melhor do que as demais versões do protocolo.

C. Ambiente WAN

TABELA V. AMBIENTE WAN

Indicador	Tahoe	Reno	New Reno	Vegas	Poli	C4	Melhor
Vazão Média(kbps)	578	613,56	456,04	498,36	549,34	690,09	C4
Latência Média(ms)	7	6,19	8,21	3,69	8,42	7,33	Vegas
Pacotes Enviados	485	533	386	415	446	569	C4
Pacotes Perdidos	31	32	29	18	13	19	Poli
Perda Percentual (%)	6,39	6	7,51	4,33	2,91	3,34	Poli

A análise da Tabela 5 demonstra que a configuração C4 apresentou melhores resultados de Vazão em relação demais versões. Em termos de perda percentual, o desempenho obtido foi ligeiramente próximo ao TCP *Poli* que se apresenta melhor do que os demais.

Nesse ambiente, também foi possível verificar um resultado não tão satisfatório para a latência da rede que apesar de apresentar melhores resultados em relação ao TCP *Poli* e o *New Reno*, obteve resultados mais elevados em relação as demais versões testadas.

VI. CONCLUSÃO E TRABALHOS FUTUROS

Com todas as simulações realizadas, foi possível concluir que a possibilidade de melhoria no desempenho das redes de computadores, a partir de novas configurações e propostas de controle de congestionamento, é possível com algumas modificações no algoritmo base dos protocolos existentes.

Portanto, investir em pesquisas que tratem do congestionamento de redes é essencial, para que junto aos conceitos do cenário de redes atual seja definido qual melhor se adequa ao seu cenário redes, ou de que forma uma versão do protocolo pode ser complemento da outra. Também foi percebido que o desempenho das versões do TCP depende diretamente do ambiente em que a rede está inserida.

Diante disso, trabalhos futuros podem visar as limitações presentes nas versões atuais do protocolo TCP. Outra proposta seria modificar a função preditiva utilizada no TCP *Poli* (média móvel ponderada). Duas opções podem ser sugeridas: as redes neurais e cadeias de *Markov*.

REFERÊNCIAS

- [1] Silva, H., Otimização do mecanismo de controle de congestionamento do protocolo TCP utilizando média móvel ponderada, Mestrado em Engenharia de Sistemas, Universidade de Pernambuco, 2016.
- [2] James F. Kurose e Keith W. Ross, Redes de Computadores e a Internet: Uma Abordagem Top-Down, 2010.
- [3] SOARES, A., SANTOS, J., SILVA F., Redes SDN X Redes tradicionais Estudo comparativo em ambientes VOIP, Trabalho de conclusão do Curso de Tecnologia de Redes de Computadores, UNIBRATEC, 2015.
- [4] Stevens, W., STCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms, vol. 1, pp. 1-10, 2001.
- [5] Lima, J., Análise comparativa dos algoritmos de controle de congestionamento do tcp, Mestrado em Engenharia da Computação, Universidade de Pernambuco, 2015.
- [6] Floyd, S. e H. T., The New Reno Modification to TCP's Fast Recovery Algorithm. IETF Request for Comments (RFC) 2582, vol. 1, 1999.
- [7] Podlesny, M., Williamso, C., Providing Fairness Between TCP NewReno and TCP Vegas with RD Network Services, 2010.
- [8] Hong, Z., Amoakoh, G., e Zhongwei, Z., Performance of STTVegas in Heterogeneous Wired and Wireless Networks, 2006.

- [9] Brakmo, L. e Peterson, L., TCP Vegas: End to End Congestion Avoidance on a Global Internet, vol. 25, pp. 69-86, 1995.
- [10] Issariyakul T. e Hossain E., Introduction to Network Simulator NS2, E., ISBN:978-1-4614-1405-6, Springer, 2nd Edition, 2012.
- [11] A. S. Tanenbaum e D. J. Wetherall, Computer Networks, Pearson Education, 2011.
- [12] E. Q. Albuquerque, Qualidade de Serviço em Redes de Computadores, Elsevier Publishers, 2013.
- [13] D. A. Manasse e V. A. F. Almeida, Capacity Planning for Web Performance: Metrics Models and Methods. Prentice Hall Books, 2002.