

# Early Congestion Control: Melhorando o Desempenho do TCP em Redes Sem Fio

Marcos Talau, Mauro Fonseca e Anelise Munaretto

**Resumo**—Diversas pesquisas vêm sendo realizadas buscando melhorar o desempenho do protocolo TCP em redes sem fio adhoc. Vários destes estudos propõem alterações no TCP, que levam a alteração nos pontos finais, o que pode ser difícil de implementar. Outras propostas buscam ocultar do TCP problemas no enlace, mas que já se mostraram não serem eficazes. Com o objetivo de melhorar o desempenho do TCP nestas redes dinâmicas, este artigo apresenta um mecanismo *cross-layer*, implementado na forma de gerenciamento ativo de filas (*active queue management - AQM*). O mecanismo proposto altera dinamicamente o valor do campo janela dos cabeçalhos TCP de acordo com a taxa de utilização da fila do roteador. O método é compatível com qualquer implementação TCP e não impõe alterações no protocolo TCP. Para avaliar seu desempenho foram realizados testes comparativos com o Droptail e com o RED. Resultados em diferentes ambientes de simulação indicam que o método proposto aumentou a vazão e reduziu o atraso e o número de perdas.

**Palavras-Chave**—Redes sem fio com múltiplos saltos, protocolo de controle de transporte, gerenciador ativo de filas.

**Abstract**—There have been many solutions for improving performance of TCP protocol over adhoc wireless networks. Several of these studies suggest changes in TCP, leading to changes in end points, which may be difficult to implement. Other solutions seek to hide the problems in the TCP link, but which have proven to be ineffective. In order to improve TCP performance in these dynamic networks, this paper presents a *cross-layer* approach, implemented as an active queue management - AQM. The proposed method dynamically changes the value of window field of TCP headers according to the rate of the router queue. The approach is TCP-Friendly. To assess the performance were conducted comparative tests with DropTail and the RED. The simulation results show that the new method improves throughput, delay and number of losses.

**Keywords**—Wireless multihop networks, transport control protocol, active queue management.

## I. INTRODUÇÃO

Já é bastante conhecido que o controle de congestionamento presente no protocolo de transporte TCP não funciona adequadamente em redes onde o meio físico está sujeito a falhas, como é o caso em redes sem fio [1] [2] [3]. Nestas redes, o principal problema é que ele não consegue diferenciar entre perdas ocorridas por congestionamento daquelas causadas por problemas no enlace, e reage a ambas diminuindo a taxa de transmissão.

Este problema é ainda maior em redes adhoc com múltiplos nós, como já foi verificado em diversos estudos [4] [5] [6]. Thangam e Kirubakaran apresentaram em [7] o estado da arte na avaliação de desempenho do TCP em redes

adhoc, identificando três principais problemas: 1) O TCP não é capaz de diferenciar entre perdas por falhas na rota e congestionamento de rede. 2) O TCP passa por frequentes problemas de roteamento. 3) Contenção no canal sem fio.

Buscando adaptar o TCP para uma rede sem fio adhoc, diversos trabalhos criaram técnicas objetivando ignorar perdas e mudanças no enlace, não reduzindo desta forma a taxa de transmissão do TCP. Os trabalhos apresentados em [8] e [4] sugerem que tal procedimento pode não ser eficiente. Este comportamento agressivo pode prejudicar o desempenho do protocolo, induzindo a mais perdas. Um dos principais problemas do TCP sobre redes sem fio é o número excessivo de acesso ao meio pelo TCP, isto é causado não apenas por ACKs que competem com pacotes de dados, mas por retransmissões causadas por perdas [9].

O objetivo desse artigo é apresentar uma solução ao baixo desempenho do protocolo TCP em redes sem fio com múltiplos saltos (*multihops*). Para isso, algumas premissas serão consideradas, como, por exemplo, informações entre camadas (*cross-layer*), para possibilitar a diferenciação entre congestionamento nas filas dos roteadores ou nas filas dos enlaces de saídas de acesso ao meio.

O artigo está dividido em seis seções, como apresentado a seguir. A Seção 2 apresenta os trabalhos relacionados na busca da melhoria do desempenho do TCP em redes sem fio adhoc. A Seção 3 apresenta a proposta deste trabalho, denominada ECC (*Early Congestion Control*), um mecanismo AQM em formato *cross-layer* que altera a janela do TCP buscando melhorar o seu desempenho. Em seguida, a Seção 4 apresenta a avaliação da proposta em dois cenários. Por último, são apresentadas as Seções 5 e 6, com análise dos resultados e conclusões.

## II. TRABALHOS RELACIONADOS

Existem diversos estudos sobre o desempenho do TCP em redes sem fio adhoc, onde inúmeras simulações foram feitas para obter um melhor entendimento do seu comportamento e buscar novas formas para se aumentar o seu desempenho.

Nithya, Mala e Sivasankar criaram um método denominado *fibonacci based contention congestion control* (FBCCC), dividido em duas partes, uma para operar com o problema de contenção da camada de enlace e outra para determinar o tamanho da janela de congestionamento durante a fase *slow start*. Simulações indicaram melhoras em vários pontos, como na taxa de colisão e atrasos [10]. Em [11] também foi criado um método *cross-layer*; ele utiliza um mecanismo próprio de controle de congestionamento que opera quando a taxa de ocupação do canal e o nível de sinal recebido é menor que um

limite definido. O mecanismo de controle de congestionamento estima o atraso e a largura de banda disponível. Resultados indicaram melhoras em termos de vazão, taxa de entrega e atraso. No trabalho de [12] os autores também utilizaram informações do canal sem fio, eles criaram um esquema para coletar informações da camada de enlace para estimar a utilização do canal da estação; as informações são utilizadas para determinar valor de janela de contenção, este valor é repassado para a camada de transporte, que ajusta a taxa de vazão de acordo. O método melhorou a justiça e sensivelmente a vazão.

Quando a codificação de rede (*Network Coding - NC*) é utilizada em enlaces sem fio, os segmentos de diferentes fluxos são combinados para se tentar reduzir o número de transmissões [13]. Vu, Boukhatem e Nguyen reformularam essa codificação de rede para que o TCP destino informe quantos pacotes não vistos existem na janela de combinação (*coding window*). Os resultados indicaram melhoras na vazão e no tempo de entrega [14]. Em [13] foram exploradas as possibilidades da codificação de rede para encapsular segmentos TCP ACK junto a combinação padrão. Resultados obtidos pelo simulador de rede ns-3 mostraram que o novo encapsulamento melhorou em 15% o desempenho em relação a codificação de rede padrão.

Vários métodos da literatura realizam modificações no protocolo TCP, o que induz a um problema de escalabilidade de implementação. Uma alternativa interessante para tratar do problema pode ser o uso de gerenciadores ativos de fila (AQMs), proposta em alguns trabalhos. Em [15] foi proposto um esquema similar ao RED [16], conhecido como NRED (*Neighborhood RED*), que busca aumentar a justiça do TCP observando informações do canal para realizar o ajuste das probabilidades de descarte. Assim como o RED, este método requer o ajuste fino dos parâmetros. Para aliviar este problema, os autores em [17] propõem o NDEM (*Neighborhood Diffusion Early Marking*), que faz uso de propriedades estatísticas de redes adhoc.

### III. PROPOSTA DO TRABALHO: *Early Congestion Control* (ECC)

Levando em conta que uma das causas de um congestionamento (em um tempo  $t$ ) é a transmissão de dados em quantidade superior às capacidades (no instante  $t$ ) da rede, neste trabalho é proposto o método *cross-layer* denominado ECC (*early congestion control*). O esquema é um AQM derivado (*fork*) do Droptail, possuindo todas as suas características e comportamentos. O que o diferencia do Droptail é a criação de um recurso: a alteração do valor do campo janela do cabeçalho TCP de segmentos que estiverem saindo da fila. A atualização da janela é feita com o uso da Equação 1.

$$f(w) = \begin{cases} \frac{Q-q}{Q} \cdot w & q > Q \cdot t \\ w & \text{caso contrário} \end{cases} \quad (1)$$

onde  $Q$  representa o tamanho total (em *bytes*) da fila do roteador;  $q \subset Q$  e indica a quantidade de *bytes* da fila que estão sendo utilizados;  $t$  é uma constante entre  $[0, 1]$ ; e  $w$

corresponde ao valor, em *bytes*, da janela do cabeçalho TCP. Quando a quantidade de *bytes* utilizados na fila,  $q$ , for superior ao limiar  $Q \cdot t$ , o valor do campo janela do cabeçalho TCP do segmento é reduzido proporcionalmente à porcentagem  $(\frac{Q-q}{Q})$  de *bytes* disponíveis (livres) na fila; caso isto não ocorra o valor do campo janela permanece o mesmo.

O objetivo do ECC é a realização de um controle adicional da vazão dos fluxos TCP para evitar que a fila do roteador fique cheia, e assim inicie um congestionamento. Ele é compatível com qualquer implementação TCP, e para utilizá-lo é somente necessário a sua instalação em um roteador da rede.

Em uma rede adhoc de múltiplos saltos os pacotes podem passar por vários nós até chegar ao destino final, para isto ser possível cada nó da rede tem capacidades de roteamento, isto é, ele deve ter no mínimo uma fila de pacotes e uma tabela de roteamento. O roteamento atua na camada de rede, o ECC é executado no momento do roteamento — na saída dos pacotes da fila, e manipula a camada de transporte, por isto o consideramos como *cross-layer*.

Apesar do nome ser semelhante, é importante destacar que o ECC tem um comportamento diferente do AQM RED (*random early detection*). O RED utiliza um mecanismo para monitorar a média do tamanho da fila realizando um descarte probabilístico de segmentos, enquanto o ECC funciona da mesma forma que o Droptail, ou seja, realiza o descarte de pacotes apenas quando a fila estiver cheia, além de possuir a característica própria de reduzir o valor do campo janela do cabeçalho TCP proporcionalmente a quantidade de *bytes* disponíveis na fila.

O ECC tem um comportamento de redução do valor do campo janela dos segmentos TCP de forma proporcional ao nível de utilização da fila do roteador, quanto mais a fila cresce, mais as janelas são reduzidas. Este comportamento foi criado para tentar evitar que a fila fique cheia. Nas Figuras 1 e 2 são exibidas as taxas de utilização da fila de um roteador durante uma simulação; na Figura 2 o método ECC foi utilizado, e na outra o AQM padrão (Droptail). Pode se observar que o ECC manteve o uso da fila mais controlado, apresentando uma baixa variação.

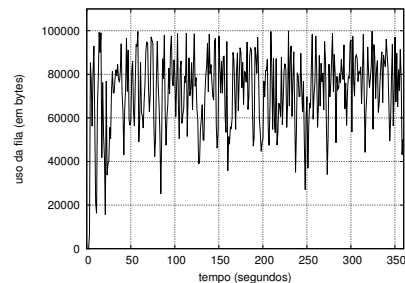


Fig. 1. Taxa de utilização da fila do roteador com o uso do AQM Droptail.

Exemplo de funcionamento da proposta: Considere que um roteador tem uma fila com capacidade de 100 KB. O nível de utilização do ECC foi definido para 40% da fila ( $t = 0.4$ ), logo, quando o nível de utilização da fila atingir esta porcentagem o valor do campo janela dos segmentos TCP começará a ser alterado. Após um intervalo de tempo

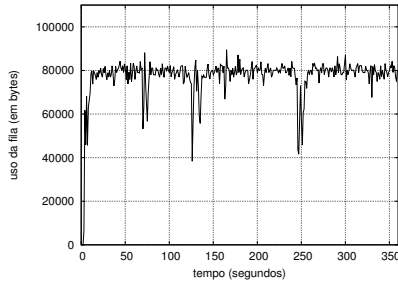


Fig. 2. Taxa de utilização da fila do roteador com o uso do AQM ECC.

o roteador passa a ter 65 KB na fila, este valor, que equivale a 65% da fila, é maior que o nível de utilização definido, logo, quando um segmento (com janela TCP = 30000) for sair da fila a sua janela será atualizada para 10500, ou seja, o campo janela foi reduzido de acordo com a porcentagem de *bytes* disponíveis na fila do roteador.

#### IV. AVALIAÇÃO DA PROPOSTA

Para avaliar a proposta foram criados dois cenários. O primeiro é composto por apenas três grupos de nós; este cenário reduzido pode existir em diversas redes e também pode avaliar a eficiência do AQM sem grandes interferências. O segundo cenário é composto por uma topologia grid 4x4.

##### A. Cenário 1

Os nós foram posicionados para existir um nó central com um conjunto de  $n$  nós a esquerda, e  $n$  nós a direita. A rede *adhoc* foi configurada para utilizar o padrão IEEE 802.11a, com método de transmissão OFDM a 6 Mbps<sup>1</sup>. Para o nó central ser utilizado como roteador na comunicação entre os nós da esquerda/direita foi utilizado o protocolo de roteamento *on-demand distance vector* (AODV).

Nas simulações os nós da esquerda foram configurados para serem fontes TCP ( $[F1, Fn]$ ), e os da direita receptores TCP ( $[R1, Rn]$ ). Durante a simulação cada fonte fez uma conexão com um receptor, e dados foram transmitidos até o término da simulação. As fontes foram configuradas para utilizar o gerador de tráfego *BulkSendApplication* (ns-3), que gera dados a uma taxa constante.

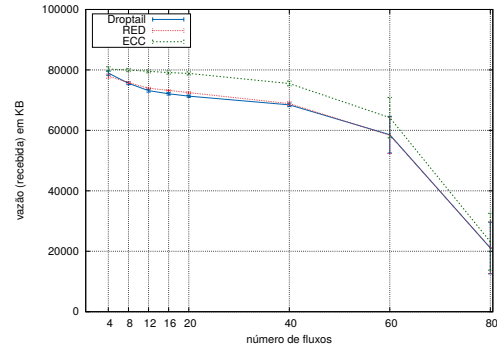
Demais configurações: o AQM foi configurado apenas no nó central; tamanho da fila: 97 KB; RED: os parâmetros foram ajustados conforme o procedimento presente em [18], ficando: *minth*: 3 KB, *maxth*: 48.5 KB,  $Q_w/w_q$ : 0.002, tamanho médio dos segmentos: 530 *bytes*; tamanho dos segmentos TCP: 1000 *bytes*; algoritmo de controle de congestionamento: TCP *New Reno*.

O ECC foi comparado ao Droptail e ao RED através de simulações no ns-3. Cada simulação foi executada durante o tempo de seis minutos. As simulações foram executadas com 4, 8, 12, 16, 20, 40, 60 e 80 fluxos; os nós da esquerda foram utilizados como fontes TCP, e os da direita como receptores, cada nó fonte/receptor criou apenas um

<sup>1</sup>Atualmente no ns-3 não são suportadas maiores velocidades para redes *adhoc*.

fluxo, logo para se aumentar o número de fluxos, mais nós fontes/receptores foram criados. Cada simulação foi executada 50 vezes, com diferentes sementes de números aleatórios. Nelas foram coletados dados referentes a vazão (recebida), perdas, e atraso e justiça. Os dados que são apresentados nas próximas subseções são referentes a média das 50 rodadas com um intervalo de confiança de 95%.

- 1) *Vazão*: Foi coletada a vazão (*throughput*) recebida pelos receptores  $[R1, Rn]$ . A Figura 3 exibe a vazão total (somatória de  $R1$  até  $Rn$ ).

Fig. 3. Vazão recebida pelos receptores  $[R1, Rn]$ .

Com a mudança do número de fluxos o método proposto teve pouca variação da vazão. No Droptail/RED a vazão foi bastante reduzida em função do número de fluxos; e comparado ao ECC seu desempenho foi em média 7,31% menor.

O uso do ECC trouxe uma maior vazão, até 20 fluxos a vazão pouco variou. Com 40 e 60 fluxos a vazão foi reduzida, mas o novo método manteve uma vazão superior aos demais. No geral o ECC obteve uma vazão 8,28% maior.

- 2) *Perdas*: As perdas registradas foram do nó central onde o AQM estava presente. Elas ocorreram quando a fila ficou cheia (Droptail/ECC), ou quando o AQM produziu (RED). A Figura 4 exibe o número total de perdas.

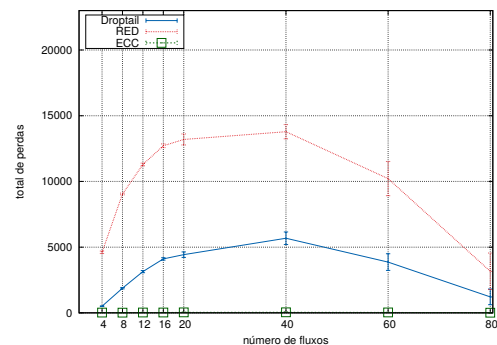


Fig. 4. Número total de perdas (o ECC registrou poucas perdas (média de 25) por isso quase não aparece no gráfico).

O uso do ECC levou a poucas (média  $26,55 \pm 10,95$ ) perdas no roteador, por isto na Figura 4 os dados referentes ao método quase não aparecem. O baixo número de perdas com a utilização do ECC era previsto,

pois o comportamento de reduzir a janela dos segmentos de acordo com o nível de utilização da fila leva a uma redução da taxa de transmissão das fontes. O Droptail e o RED tiveram o número de perdas em função do número de fluxos, comparado ao ECC, o número de perdas foi muito maior, porém, nestes métodos as perdas são esperadas e necessárias.

As perdas na rede ativam mecanismos de controle de congestionamento das fontes TCP, um outro fator também considerado na ativação é o recebimento de três ACKs duplos (*triple dupacks*). A Figura 5 exibe o número total de *triple dupacks* registrados nas simulações. Comparado ao Droptail e ao RED, o ECC registrou um número bastante reduzido de *triple dupacks*. Com o menor registro de *triple dupacks* pode-se dizer que a rede ficou menos congestionada.

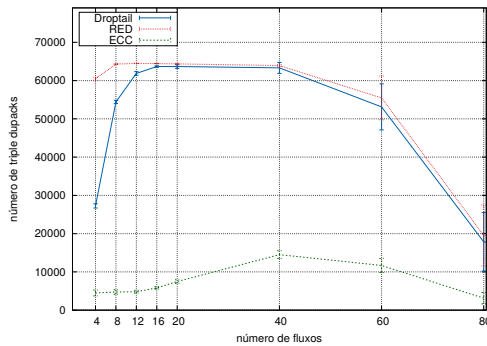


Fig. 5. Número total de *Triple dupacks*.

- 3) *Atraso e Justiça*: O atraso refere-se ao tempo utilizado para um pacote ir da fonte até o destino final. A Figura 6 apresenta o atraso médio.

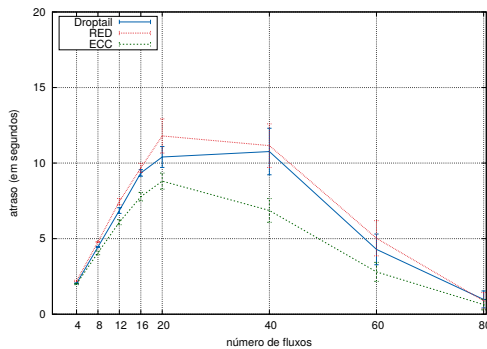


Fig. 6. Atraso médio registrado.

Pode-se observar que, até 20, o atraso aumentou de acordo com o número de fluxos. Com 40 fluxos o ECC reduziu o atraso, e o Droptail aumentou, com 60/80 fluxos o atraso de todos os métodos diminuiu. Comparando, o ECC teve um atraso médio de 28,40% a menos que o Droptail, e 36,21% a menos que o RED. Analisando isoladamente os AQMs Droptail e ECC: o algoritmo do Droptail é mais rápido que o do ECC pois não realiza a atualização de segmentos TCP, porém na prática o ECC causou uma redução de atrasos na rede,

pois o seu uso trouxe redução de congestionamento de rede.

A diferença de atrasos entre os fluxos foi utilizada para definir a justiça. Para gerar este dado: em cada rodada foi calculado o desvio padrão dos atrasos, após isso foi calculado a média e o intervalo de confiança do desvio padrão dos atrasos de todas as rodadas. A Figura 7 apresenta o resultado. O desvio padrão dos atrasos cresceu com o aumento do número de fluxos, chegando ao limite com 40 fluxos. No geral o ECC obteve o menor valor em cada teste, o que caracteriza uma maior justiça entre os fluxos.

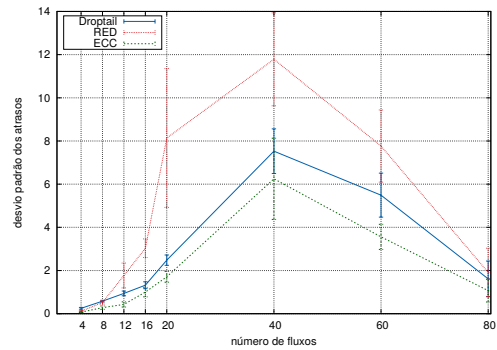


Fig. 7. Desvio padrão dos atrasos.

**B. Cenário 2**

Neste cenário utilizamos uma topologia grid 4x4. A configuração da rede foi a mesma do Cenário 1, com exceção do tamanho da fila ajustada para 25 pacotes e do protocolo de roteamento, onde foi utilizado o protocolo *dynamic source routing* (DSR), a escolha deu-se por ele ter sido adotado em outros trabalhos que utilizaram a topologia grid, como em [19].

Na simulação foram criados 20 fluxos TCP (10 de [1, 1] para [4, 4] e outros 10 de [1, 4] para [4, 1]) executados por seis minutos a uma taxa de transmissão constante.

A Tabela I apresenta os resultados de uma média de 50 rodadas, que são analisados a seguir.

TABELA I  
RESULTADOS DA SIMULAÇÃO II

	Vazão	Atraso	Perdas
Droptail	23.75 KB ± 1.12 KB	38.4 s ± 4.1 s	1387 ± 0232
RED	22.67 KB ± 1.19 KB	38.4 s ± 8.3 s	5593 ± 1387
ECC	23.80 KB ± 1.40 KB	30.5 s ± 5.0 s	1176 ± 0175

- 1) *Vazão*: A vazão total recebida pelos receptores TCP foi considerada. O novo método obteve uma vazão em torno de 5% superior aos demais.
- 2) *Atraso*: O atraso considerado foi o tempo que um pacote levou para chegar até o receptor TCP. O ECC reduziu consideravelmente o atraso, sendo 20% menor.
- 3) *Perdas*: As perdas registradas foram de todos os nós da grid, e ocorreram quando o a fila ficou cheia, e no caso do RED, também quando o AQM produziu. Comparado ao Droptail o ECC diminuiu as perdas em 15%.

## V. ANÁLISE DOS RESULTADOS

Verificando os dados das simulações realizadas pode-se concluir que o ECC foi eficaz em reduzir o congestionamento de rede, e consequentemente aumentar o desempenho do protocolo TCP. O uso de seu mecanismo de atualização de janelas, em função do nível de utilização da fila do roteador, levou a uma maior vazão, poucas perdas, e a um menor atraso.

Com o crescimento do número de fluxos a disputa por recursos de um roteador aumenta, se o aumento não for controlado a fila irá ficar cheia, resultando em perdas de pacotes. As fontes TCP conseguem detectar estas perdas, e quando isso acontece é feita uma redução da taxa de transmissão (de acordo com o algoritmo de controle de congestionamento que ela utiliza). Então, quando o número de fluxos aumenta, se as técnicas de controle de congestionamento utilizadas não regularem adequadamente os recursos que a rede possui, a vazão total (soma de todos os fluxos) irá ser reduzida. A utilização do ECC conseguiu manter a vazão constante, independentemente do número de fluxos.

As perdas na fila do roteador ocorrem quando ela fica cheia. O mecanismo do ECC foi eficiente em evitar que ela ficasse cheia; poucas perdas foram registradas com o seu uso. A utilização do ECC também reduziu bastante o número de ACKs triplos, e um pouco do atraso, o que indica que a rede ficou menos congestionada.

A aplicação do ECC em uma topologia grid pouco aumentou a vazão, porém o congestionamento foi melhor controlado, pois o atraso foi bastante reduzido e as perdas também foram menores.

## VI. CONCLUSÕES

Este trabalho apresentou um método (chamado de ECC), que atua como AQM buscando melhorar o desempenho do protocolo TCP em redes sem fio adhoc. O ECC baseia-se no ajuste do campo janela dos segmentos TCP de acordo com a taxa de utilização da fila do roteador. O método foi comparado com o Droptail e ao RED em dois cenários de simulação.

No primeiro cenário o novo método produziu uma maior vazão, o número de perdas foi bastante reduzido, assim como o número de ACKs triplos. O atraso também foi menor, e a justiça entre fluxos foi maior. No ambiente grid a vazão pouco aumentou, porém o congestionamento foi melhor controlado, pois o atraso foi bastante reduzido e as perdas também foram menores.

No geral, os resultados indicaram que o método melhorou o desempenho do protocolo TCP. No futuro podem ser feitas comparações com técnicas produzidas em estudos semelhantes, como o TCP-DAA [9], além da aplicação do método em novas simulações, por exemplo, com fluxos UDP, ou com o uso de diferentes algoritmos de controle de congestionamento nas fontes.

## REFERÊNCIAS

- [1] V. Bhanumathi and R. Dhanasekaran, "TCP variants - A comparative analysis for high bandwidth - delay product in mobile adhoc network," in *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*, vol. 2, Feb 2010, pp. 600–604.

- [2] Y. Zhang, J. Hu, and G. Feng, "SNOOP-based TCP Enhancements with FDA in wireless cellular networks: A comparative study," in *Communications, Circuits and Systems, 2008. ICCCAS 2008. International Conference on*. Fujian Province, China: IEEE Press, maio 2008, pp. 181–185.
- [3] Y. Zhang and G. Feng, "A new method to improve the TCP performance in wireless cellular networks," in *Communications, Circuits and Systems, 2009. ICCAS 2009. International Conference on*. Milpitas, California, USA: IEEE Press, jul. 2009, pp. 246–250.
- [4] K. Chen, Y. Xue, and K. Nahrstedt, "On setting TCP's congestion window limit in mobile ad hoc networks," in *Communications, 2003. ICC '03. IEEE International Conference on*, vol. 2, May 2003, pp. 1080–1084 vol.2.
- [5] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on TCP throughput and loss," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3, March 2003, pp. 1744–1753 vol.3.
- [6] H. Zhai, X. Chen, and Y. Fang, "Rate-based transport control for mobile ad hoc networks," in *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 4, March 2005, pp. 2264–2269 Vol. 4.
- [7] S. Thangam and E. Kirubakaran, "A Survey on Cross-Layer Based Approach for Improving TCP Performance in Multi Hop Mobile Adhoc Networks," in *Education Technology and Computer, 2009. ICETC '09. International Conference on*, April 2009, pp. 294–298.
- [8] J. Li, C. Blake, D. S. De Couto, H. I. Lee, and R. Morris, "Capacity of Ad Hoc Wireless Networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '01. New York, NY, USA: ACM, 2001, pp. 61–69. [Online]. Available: <http://doi.acm.org/10.1145/381677.381684>
- [9] R. De Oliveira and T. Braun, "A Smart TCP Acknowledgment Approach for Multihop Wireless Networks," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 2, pp. 192–205, Feb 2007.
- [10] B. Nithya, C. Mala, and E. Sivasankar, "A novel cross layer approach to enhance qos performance in multihop adhoc networks," in *Network-Based Information Systems (NBIS), 2014 17th International Conference on*, Sept 2014, pp. 229–236.
- [11] D. Sunitha, A. Nagaraju, and G. Narsimha, "A cross-layer approach for congestion control in multi hop mobile ad hoc networks," in *Computing for Sustainable Global Development (INDIACom), 2014 International Conference on*, March 2014, pp. 54–60.
- [12] P. T. Giang and P. M. Vi, "Cross layer design to enhance tcp performance in multi-hop ad hoc networks," in *Advanced Technologies for Communications (ATC), 2013 International Conference on*, Oct 2013, pp. 642–647.
- [13] D. Gomez, R. Aguero, M. Garcia-Arranz, and D. Ros, "Tcp acknowledgement encapsulation in coded multi-hop wireless networks," in *Vehicular Technology Conference (VTC Spring), 2014 IEEE 79th*, May 2014, pp. 1–5.
- [14] T. V. Vu, N. Boukhatem, T. M. T. Nguyen, and G. Pujolle, "Dynamic coding for tcp transmission reliability in multi-hop wireless networks," in *A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on*, June 2014, pp. 1–6.
- [15] K. Xu, M. Gerla, L. Qi, and Y. Shu, "Enhancing TCP Fairness in Ad Hoc Wireless Networks Using Neighborhood RED," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '03. New York, NY, USA: ACM, 2003, pp. 16–28. [Online]. Available: <http://doi.acm.org/10.1145/938985.938988>
- [16] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking (TON)*, vol. 1, no. 4, pp. 397–413, 1993.
- [17] A. Medina, G. Arce, and B. Sadler, "Statistical Approach to Neighborhood Congestion Control in Ad Hoc Wireless Networks," in *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, Nov 2007, pp. 764–768.
- [18] E. C. G. Wille, M. Mellia, E. Leonardi, and M. Ajmone-Marsan, "Design and Analysis of IP Networks with End-to-End QoS Guarantees," in *XXI Simpósio Brasileiro de Telecomunicações*. Belém-Pará, Brasil: Sociedade Brasileira de Telecomunicações, set. 2004.
- [19] S. Gajjar and H. Gupta, "Improving performance of adhoc TCP in Mobile Adhoc Networks," in *India Conference, 2008. INDICON 2008. Annual IEEE*, vol. 1, Dec 2008, pp. 144–147.