

# TCP em Redes *Ad Hoc*: A Influência do Tamanho de Pacotes e das Filas

Marcos Talau, Mauro Fonseca, e Emilio C. G. Wille

**Resumo**—O uso do TCP em redes sem fio *ad hoc* não funciona de forma adequada devido as frequentes perdas que ocorrem no meio físico, isso invoca de forma equivocada os mecanismos de controle de congestionamento, que reduzem indevidamente a taxa de transmissão. Uma possibilidade para minimizar tais problemas consiste no uso do *early congestion control* (ECC), um novo método que atua como um gerenciador de filas que ajusta o campo janela do cabeçalho TCP de acordo com o uso da fila do roteador. O método se mostrou promissor, trazendo ganhos ao TCP. Este artigo busca responder a duas perguntas: o quanto o tamanho dos pacotes influencia o comportamento do ECC? e qual o efeito do tamanho da fila em seu desempenho? Em simulações realizadas com o ECC, Droptail e RED, resultados indicaram que a redução do tamanho do pacote TCP produz mais congestionamento, e que com o aumento do tamanho da fila o congestionamento é reduzido. Com o aumento/diminuição também foi verificado que o ECC se comportou de maneira semelhante aos outros métodos, porém, reduziu significativamente 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**—The use of TCP in ad hoc wireless networks does not work properly due to frequent losses that occur in the physical medium, it invokes wrongly congestion control mechanisms, that unduly reduce the transmission rate. One possibility to minimize these problems is the use of the early congestion control (ECC), a new method that acts as an active queue manager that adjusts the TCP header window field according to the use of the router queue. The approach has shown promise, bringing gains to TCP. This paper seeks to answer two questions: how the size of the packages influence the ECC behavior? and what effect of the queue size in your performance? In simulations with the ECC, DropTail, and RED, results indicated that reducing the size of the TCP packet produces more congestion, and with increasing length of the queue, congestion is reduced. With the increase/decrease is also observed that the ECC behaved similarly to the other methods, however, significantly reduced the number of delays and losses.

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

## I. INTRODUÇÃO

Um dos protocolos de transporte mais utilizados na internet é o protocolo de controle de transmissão (TCP). Ele tem como principais características a transferência confiável de dados e controles de transmissão. Os controles de transmissão buscam evitar sobrecarga no receptor e o congestionamento da rede. Um receptor TCP consegue receber dados até um nível, se o transmissor enviar mais dados que a capacidade do receptor, naturalmente haverá perda de dados, pois o

receptor não conseguirá processar todos os dados; para evitar que tal problema ocorra, a implementação padrão do TCP tem um recurso chamado de controle de fluxo, onde o receptor sempre informa ao transmissor a quantidade de dados que ele consegue receber. No caminho transmissor-receptor haverá equipamentos de rede, como roteadores, que também conseguem receber uma quantidade limitada de dados. O TCP não tem informações das capacidades deste meio, mas consegue prever quando algum equipamento está sobrecarregado, isso é feito com o uso de marcadores de tempo, se ele não receber a confirmação de algum dado enviado, ele realiza a retransmissão e subentende que a rede está congestionada, reduzindo a sua taxa de transmissão; este comportamento é chamado de controle de congestionamento [1].

O TCP foi projetado para operar com redes cabeadas, onde falhas no nível físico são difíceis de ocorrer, então quando o TCP detecta uma perda de dados é bem provável que ela tenha ocorrido devido a rede estar congestionada, e o uso do controle de congestionamento para reduzir a taxa de transmissão é a solução adequada [2]. Porém, em redes onde o meio físico é propício a falhas, o controle de congestionamento é utilizado de forma indevida, pois a perda costuma não ser causada por uma sobrecarga de rede, e ele age como fosse, reduzindo indevidamente a taxa de transmissão [3] [4].

Nas redes sem fio os erros no nível físico são frequentes. Visando otimizar o controle de congestionamento nestas redes, diversos estudos já foram realizados, como em [4] [5]. Naturalmente, se o número de erros for maior, o desempenho do TCP piora ainda mais, isso é o que ocorre em redes sem fio *ad hoc*. Nelas existem ainda outros problemas, como: contenção em canais e problemas de roteamento [6].

Em um estudo recente [7] foi criado o *early congestion control* (ECC), um novo método que busca melhorar a performance do TCP em redes sem fio. O método consiste em um gerenciador de filas que tem por objetivo reduzir o congestionamento em redes *ad hoc*; isto é feito com a alteração do campo janela do cabeçalho TCP conforme o uso da fila do roteador. Em testes nestas redes foi verificado que o método se mostrou promissor, trazendo ganhos ao TCP. Este artigo busca responder a duas perguntas: o quanto o tamanho dos pacotes influencia o comportamento deste método? e qual o efeito do tamanho da fila no seu desempenho?

O artigo está organizado em seções como seguem. Na Seção II é apresentada uma revisão da literatura sobre o problema do TCP em redes sem fio. A Seção III detalha o ECC. A Seção IV apresenta detalhes do modelo de simulação. Na sequência, a Seção V apresenta os resultados e discussões. Por fim são apresentadas as conclusões.

Marcos Talau, Mauro Fonseca, e Emilio C. G. Wille, Universidade Tecnológica Federal do Paraná (UTFPR) E-mails: {mtalau,maurofonseca,ewille}@utfpr.edu.br

## II. TRABALHOS RELACIONADOS

Estudos da literatura já exploraram diversas técnicas visando otimizar o TCP em redes sem fio *ad hoc*, como por exemplo: *cross-layer* entre as camadas de enlace/transporte [8] [9], *network coding* (NC) [10] [11], uso de protocolos de roteamento [12] [13] e alterações no protocolo TCP [14] [15] [16].

Wang e Perkins criaram um algoritmo *cross-layer* objetivando aumentar a vazão de dados em redes sem fio *multihop*. O método utiliza dados da camada de enlace para calcular o nível de contenção aplicado a um nó, além de obter o uso da fila do nó adjacente. Com estas duas informações cada nó realiza o ajuste da transmissão de dados. Os resultados apresentados indicaram um aumento da vazão em relação ao protocolo TCP [8]. Em [9] os autores afirmam que a contenção de mídia é um dos principais problemas do TCP em redes *ad hoc*, e criaram um protocolo chamado de *wireless congestion control protocol* (WCCP), que faz uso de dois componentes: o primeiro situado na camada de transporte, que substitui o algoritmo de ajuste de janela do TCP por um algoritmo de controle de taxa fim a fim, e o segundo entre as camadas de rede e enlace, realizando uma alocação justa de recursos entre os nós de um caminho. Os resultados indicaram que o esquema superou o TCP em termos de utilização do canal, atraso fim a fim e justiça no compartilhamento de recursos.

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 [11]. 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 [10]. Em [11] foram exploradas as possibilidades da codificação de rede para encapsular segmentos TCP ACK. 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.

Em [12] foi proposto um método, chamado *explicit link failure notification* (ELFN), para prevenir a perda de desempenho do TCP por causa de alterações na rota. Com o uso desta técnica, quando ocorre uma falha no enlace é enviada uma mensagem para o nó de origem para que ele pare de transmitir dados, e quando uma nova rota é descoberta, a origem é novamente notificada. Utilizando o método ELFN, o trabalho [13] apresentou dois outros mecanismos: o *early packet loss notification* (EPLN) que busca informar as fontes TCP sobre os pacotes perdidos, e o *best-effort ACK delivery* (BEAD), que busca retransmitir os *acknowledgements* (ACKs) perdidos pelos nós intermediários ou pelo TCP emissor. Os resultados indicaram melhoras sobre o ELFN, e o autor conclui que uma possível solução para o baixo desempenho do TCP em redes sem fio é a combinação de dados de diferentes camadas da rede.

Diversos trabalhos realizaram modificações no TCP para melhorar o desempenho do TCP em redes *ad hoc*. Em [14], os autores apresentaram uma proposta denominada *TCP-dynamic adaptive acknowledgment* (TCP-DAA), o método foca no lado

TCP receptor, embora utilize técnicas no lado transmissor para minimizar retransmissões por *timeout*. No lado transmissor são feitas duas mudanças: 1) O número de ACKs duplos que ativam o algoritmo *fast retransmit* passa de três para dois. 2) Aumento do *retransmission timeout* (RTO) em cinco vezes. No lado TCP receptor sempre é realizado o agrupamento de quatro ACKs, ou seja, para cada quatro pacotes de dados recebidos o receptor envia somente um ACK. Os resultados indicaram que o método melhorou a vazão e também o consumo de energia quando comparado com o TCP. No estudo de [15] foi proposto o *sudden recovery algorithm* (SRA), que é aplicado na implementação TCP do transmissor. O algoritmo atua na fase de *fast retransmit*, quando ocorrem múltiplas perdas de pacotes recentemente enviados. Resultados indicaram melhoras em termos de vazão e taxa de perdas. E em [16] foi criado o *adaptive congestion window* (ACW), que estima o estado de contenção da rede e dependendo dele a janela de congestionamento do TCP é ajustada, o trabalho também apresentou uma forma de estimar o *round-trip time* (RTT) utilizando informações de roteamento. O método aumentou a vazão da rede.

## III. EARLY CONGESTION CONTROL (ECC)

O método criado em [7] visa mitigar o congestionamento em redes sem fio *ad hoc*, para fazer isto ele trabalha com a alteração do campo janela do cabeçalho TCP de pacotes que passam por um roteador, a alteração é feita de acordo com o seu uso da fila. O ECC é um gerenciador de filas tradicional, em uma memória (*buffer*), insere/remove pacotes utilizando a política *first-in-first-out* (FIFO), e quando a memória fica cheia, novos pacotes são descartados. Para realizar um controle de congestionamento adicional ao TCP o método altera o campo janela presente no cabeçalho TCP com base no índice de utilização da fila do roteador. Esta alteração só é feita se a fila atinge uma porcentagem de uso, de acordo com a Equação 1.

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

em que,  $Q$  é o tamanho total (*bytes*) da fila;  $q \subset Q$  é a quantidade de *bytes* que estão sendo utilizados na fila;  $l$  é uma constante entre  $[0, 1]$  e é utilizada para definir quando o método começa a alterar as janelas; e  $w$  indica o valor (*bytes*) da janela do TCP.

*Analisando a equação:* O tamanho da janela,  $w$ , de um segmento é atualizada se o índice de uso da fila,  $q$ , é maior que uma quantidade determinada por  $Q \cdot l$  (ponto de início), quando isto é verdade a janela  $w$  é alterada para  $\frac{Q-q}{Q}$  por cento do seu valor, sofrendo assim uma redução de  $Q - \frac{Q-q}{Q}$  por cento. Se  $q$  não for maior, a janela não é alterada.

*Exemplificando:* Considere um roteador com uma fila com tamanho total de 100 KB ( $Q = 100$ ) e  $l$  tendo o valor de 0,6. Imagine que a fila atualmente está sendo utilizada em 70% ( $q = 70$ ), como  $q$  é maior que  $Q \cdot l$ , qualquer pacote que passar pela fila será alterado. Se chegar um pacote com janela de tamanho 1000 *bytes* ( $w = 1000$ ), a janela terá seu valor alterado para 300, pois  $\frac{100-70}{100} \cdot 1000$  é igual a 300.

TABELA I  
CONFIGURAÇÕES DE REDE DA GRID.

Parâmetro	Valor
Enlace	IEEE 802.11a
Modelo de Propagação	Yans (ns-3)
Transmissão	OFDM 6 Mbps
Algoritmo de Roteamento	DSR
Implementação TCP	New Reno
Tipo de Tráfego	CBR BulkSendApplication (ns-3)

Por que a redução do valor da janela do TCP pode ajudar no controle de congestionamento? Isto tem forte relação com o controle de fluxo. A janela do TCP é usada apenas neste controle, e funciona assim: na construção do cabeçalho TCP o emissor calcula a sua capacidade de recepção e registra este valor no campo janela, o pacote é transmitido, quando ele chega ao ponto final este valor é salvo e utilizado como parâmetro para definir a quantidade de dados a se transmitir. Se o valor da janela for superior à capacidade da rede, existe grande probabilidade de ocorrer um congestionamento. A alteração da janela feita pelo método tenta evitar que isso ocorra.

#### IV. AVALIAÇÃO DO MÉTODO

Para avaliar o ECC com relação a quanto o tamanho dos pacotes influencia o seu comportamento e qual o efeito do tamanho da fila em seu desempenho, foram realizadas simulações em um cenário composto de uma topologia grid 4x4 (Figura 1). A topologia é formada por computadores que são conectados através de uma rede sem fio *ad hoc*. Na grid, cada computador é representado por um nó, distante a 100 metros do nó mais próximo. Esta topologia foi escolhida pois já foi utilizada em outros trabalhos da literatura, como em [17]. As configurações de rede usadas no cenário estão descritas na Tabela I.

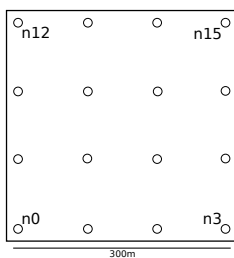


Fig. 1. Topologia grid 4x4 utilizada nas simulações.

*Sobre as configurações:* A taxa de transmissão foi definida em apenas 6 Mbps pois o ns-3 não suporta velocidades maiores em redes *ad hoc*, para isto não afetar as simulações outros pontos foram ajustados para que ocorresse congestionamento, como, por exemplo, a redução do tamanho da fila dos roteadores. O algoritmo de roteamento *dynamic source routing* (DSR) foi escolhido por ele ter sido adotado em outros trabalhos que utilizaram a topologia grid, como em [17].

Utilizando o cenário grid com as configurações anteriormente descritas, simulações foram executadas no

TABELA II  
RESULTADOS COM DIFERENTES TAMANHOS DE PACOTE COM O USO DO DROPTAIL.

Tam. Pacote	Vol. Rece. (MB)	Atraso (seg.)	Perdas (pacote)
1000	23,01 ± 0,31	62,02 ± 3,31	2367,2 ± 279,58
1500	24,42 ± 1,06	57,99 ± 4,90	1294,2 ± 197,89
2000	25,28 ± 2,10	53,11 ± 11,79	919,5 ± 208,32

TABELA III  
RESULTADOS COM DIFERENTES TAMANHOS DE PACOTE COM O USO DO RED.

Tam. Pacote	Vol. Rece. (MB)	Atraso (seg.)	Perdas (pacote)
1000	22,68 ± 1,31	59,2 ± 2,45	2804,8 ± 255,66
1500	23,46 ± 2,38	55,63 ± 4,36	1791,6 ± 74,24
2000	26,77 ± 0,36	50,35 ± 5,39	1249,6 ± 169,54

simulador de rede ns-3 [18] durante seis minutos. Ao início da simulação são estabelecidas 20 conexões TCP de  $n0$  para  $n15$  e outras 20 de  $n3$  para  $n12$ . Os nós  $n12/n15$  atuam como receptores TCP e os nós  $n0/n3$  como transmissores. Imediatamente após o estabelecimento da conexão TCP os transmissores enviam dados a uma taxa constante utilizando o modelo *BulkSendApplication* (ns-3), neste modelo os dados são gerados e enviados de forma constante e sem atraso.

Para comparar com o ECC, simulações também foram executadas com os *active queue managements* (AQMs) Droptail e o *random early detection* (RED) (parâmetros do RED ajustados conforme [19]).

#### V. RESULTADOS E DISCUSSÃO

Para realizar uma comparação entre os métodos Droptail, RED, e ECC, foram coletados dados referentes ao volume total recebido, atraso e perdas de pacotes. O primeiro corresponde à quantidade total de dados recebidos pelos receptores TCP, o atraso é definido pelo tempo de um pacote ir da origem até o destino, e as perdas indicam o número total de pacotes perdidos na rede. Os resultados são formados por uma média de dez rodadas (com sementes iniciais distintas), porém, cada rodada é composta de uma média dos resultados dos 20 fluxos TCP.

O tamanho do pacote influencia a eficiência dos AQMs? e o tamanho da fila tem qual efeito? O novo método, ECC, se mantém eficiente com estas variações? Para tentar responder a essas perguntas foram realizadas simulações apresentadas a seguir.

##### A. Tamanho dos Pacotes

Nestes testes foi alterado o tamanho fundamental dos pacotes com o objetivo de verificar a influência do tamanho no comportamento dos métodos. Foram utilizados três tamanhos de pacotes, 1000, 1500 e 2000 bytes. Não foram feitas simulações com pacotes de maior tamanho por limitações do *maximum transmission unit* (MTU) da rede. As Tabelas II, III e IV apresentam os resultados.

TABELA IV

RESULTADOS COM DIFERENTES TAMANHOS DE PACOTE COM O USO DO ECC.

Tam. Pacote	Vol. Rece. (MB)	Atraso (seg.)	Perdas (pacote)
1000	23,30 ± 0,48	52,65 ± 7,42	2138 ± 258,19
1500	23,71 ± 1,43	46,27 ± 8,85	1090,8 ± 50,53
2000	26,76 ± 0,98	48,17 ± 9,92	829,8 ± 201,95

TABELA V

RESULTADOS COM DIFERENTES TAMANHOS DE FILA COM O USO DO DROPTAIL.

Tam. Fila	Vol. Recebido (MB)	Atraso (seg.)	Perdas (pacote)
25	23,01 ± 0,31	62,02 ± 3,31	2367,2 ± 279,58
50	23,35 ± 1,25	58,16 ± 4,09	1142,6 ± 166,69
75	23,85 ± 0,50	67,26 ± 5,77	558,2 ± 81,10

*Resultados do Droptail:* Com o aumento do tamanho dos pacotes houve um aumento do volume de dados recebidos, assim como um aumento do desvio padrão. O atraso também variou conforme o tamanho dos pacotes, porém ele foi sendo reduzido; as perdas tiveram o mesmo comportamento.

*Resultados do RED:* O RED teve um comportamento semelhante ao Droptail, o com o crescimento do tamanho dos pacotes o volume de dados aumentou, o atraso foi reduzido e as perdas foram diminuindo.

*Resultados do ECC:* A mudança do tamanho dos pacotes 1000 para 1500 pouco afetou o volume dos dados recebidos, de 1500 para 2000 houve um aumento semelhante ao do RED. O atraso reduziu e aumentou, apresentando um desvio padrão elevado, isto pode ser explicado pela alteração da janela de diferentes fluxos e a sua falta de sincronia. Já as perdas foram reduzidas com o aumento do tamanho dos pacotes.

*Comparação entre os métodos:* Somando o volume recebido obtido com 1000/1500/2000 o método que registrou o maior valor foi o ECC (1,17% superior ao RED). Em todos os casos o ECC teve um menor atraso, sendo 17,7% menor que o Droptail e 12,30% que o RED. Em todos os métodos as perdas diminuíram com o aumento do tamanho do pacote, isto é relacionado a quantidade de pacotes que trafegam pela rede, se o seu tamanho for menor, haverá mais pacotes independentes na rede, e com mais pacotes o congestionamento é maior (pois haverá mais acessos ao meio sem fio), logo, o desempenho geral é reduzido. O ECC registrou um menor número de perdas em todos os casos, 12,87% sobre o Droptail e 44,04% sobre o RED.

### B. Tamanho da Fila

O tamanho da fila foi alterado para tentar verificar o seu efeito no desempenho do método. Foram realizados testes com uma fila de tamanho total de 25, 50, e 75 pacotes. A alteração deste tamanho foi replicada para todos os computadores da grid. As Tabelas V, VI e VII apresentam os resultados.

*Resultados do Droptail:* O aumento da fila pouco afetou o volume de dados recebidos. O atraso sofreu um aumento

TABELA VI

RESULTADOS COM DIFERENTES TAMANHOS DE FILA COM O USO DO RED.

Tam. Fila	Vol. Recebido (MB)	Atraso (seg.)	Perdas (pacote)
25	22,68 ± 1,31	59,2 ± 2,45	2804,8 ± 255,66
50	23,73 ± 0,34	57,12 ± 3,74	1748,5 ± 211,21
75	24,15 ± 0,46	59,13 ± 5,29	1423,2 ± 275,59

TABELA VII

RESULTADOS COM DIFERENTES TAMANHOS DE FILA COM O USO DO ECC.

Tam. Fila	Vol. Recebido (MB)	Atraso (seg.)	Perdas (pacote)
25	23,30 ± 0,48	52,65 ± 7,42	2138 ± 258,19
50	23,46 ± 0,53	44,66 ± 5,15	880,0 ± 143,36
75	23,53 ± 0,69	51,29 ± 4,71	429,4 ± 113,73

significativo quando a fila estava com 75 pacotes. As perdas foram reduzidas com o aumento da fila.

*Resultados do RED:* Quanto ao volume de dados, os resultados foram semelhantes aos do Droptail. Com a alteração do tamanho da fila o atraso manteve-se na faixa dos 58 segundos, já as perdas foram decaindo com o aumento da fila, porém continuaram altas, como era esperado, pois os RED busca causar perdas na tentativa de reduzir o congestionamento.

*Resultados do ECC:* O volume dos dados também foi pouco alterado com o aumento do tamanho da fila. O menor atraso foi registrado com uma fila de tamanho 50, já o número de perdas também foi reduzido com o aumento da fila.

*Comparação entre os métodos:* O maior volume registrado foi do método RED, com uma diferença de 0,24% para o ECC. O atraso do ECC foi menor, sendo 26,14% menor que o Droptail e 18,17% menor que o RED. O número de perdas do ECC foi 18% menor que o Droptail e 73,36% menor que o RED.

### C. Mais Congestionamento

A configuração de pacote/fila que produziu um maior congestionamento foi com pacote 1000 e fila 25. A representação gráfica destes resultados está na Figura 2. Analisando, pode se observar que o volume de dados foi aproximado, que o atraso foi bastante reduzido com o uso do ECC, assim como o número de perdas de pacotes.

### D. Discussão

Considerando que com o aumento do tamanho do pacote todos os métodos tiveram um maior volume de dados e um menor registro de perdas de pacotes, pode-se concluir que a redução do tamanho do pacote TCP produz mais congestionamento.

O aumento da fila levou a um pequeno aumento do volume dos dados e uma redução significativa de perdas, comportamento verificado em todos os métodos testados. Uma fila reduzida, por dedução, leva a mais congestionamento e ocasiona mais perdas, comportamento também comprovado no ECC.

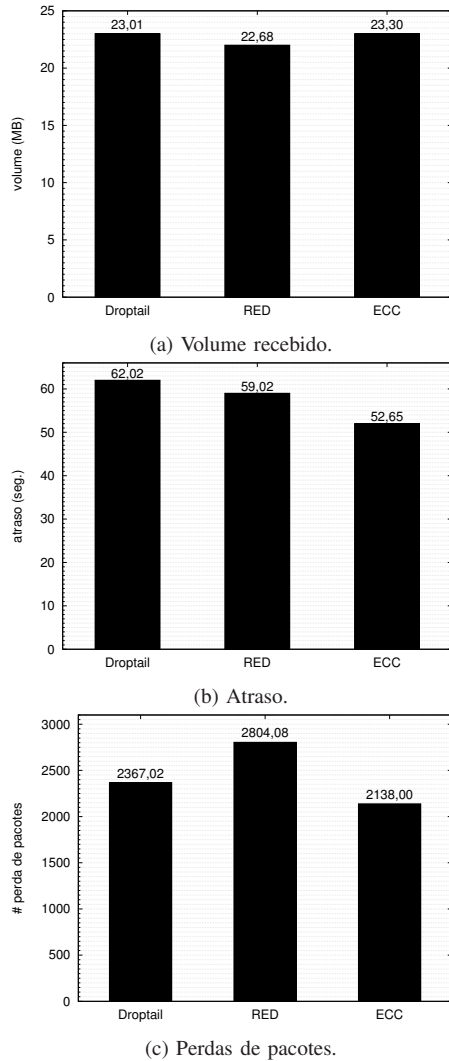


Fig. 2. Representação gráfica dos resultados de onde houve mais congestionamento (tamanho de pacote = 1000 e fila = 25).

A média do volume de dados recebido pelos métodos foram similares. Com relação ao atraso e as perdas, o ECC teve um ganho considerável em relação ao Droptail e o RED, principalmente no cenário de maior congestionamento (Figura 2). Isso indica que a alteração da janela do TCP em relação ao uso da fila se mostrou eficaz no controle de congestionamento, mesmo com a variação do tamanho dos pacotes e com diferentes tamanhos de fila.

## VI. CONCLUSÕES

Neste artigo foi avaliado o desempenho do TCP em redes sem fio *ad hoc* com a utilização do método ECC em relação ao Droptail e o RED, buscando responder às seguintes perguntas: o quanto o tamanho dos pacotes influencia o comportamento do método? e qual o efeito do tamanho da fila no seu desempenho?

Resultados obtidos em simulações com o uso de uma topologia grid indicaram que a redução do tamanho dos pacotes TCP tende a produzir mais congestionamento. E que com o aumento do tamanho da fila o congestionamento é reduzido (mas existe um limite, que quando é atingido

gera o fenômeno *bufferbloat*). No geral o ECC se comportou de maneira semelhante aos outros métodos com o aumento/diminuição do tamanho dos pacotes e das filas, porém, comparado aos outros métodos, reduziu significativamente o atraso e o número de perdas.

No futuro pretende-se realizar testes mais abrangentes buscando explorar os limites extremos e seus resultados com relação ao aumento do tamanho da fila.

## REFERÊNCIAS

- [1] J. Kurose and K. Ross, *Computer Networking: A Top-Down Approach*. Pearson Education, 2012.
- [2] R. W. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*. Reading: Addison Wesley, 1994.
- [3] V. Bhanumathi and R. Dhanasekaran, "TCP variants - A comparative analysis for high bandwidth - delay product in mobile adhoc network," in *ICCAE*, vol. 2, Feb 2010, pp. 600–604.
- [4] 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.
- [5] H. Balakrishnan, S. Seshan, and R. H. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," *Wireless Networks*, vol. 1, no. 4, pp. 469–481, 1995.
- [6] 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.
- [7] M. Talau, M. Fonseca, and A. Munaretto, "Early Congestion Control: Melhorando o Desempenho do TCP em Redes Sem Fio," in *XXXIII Simpósio Brasileiro de Telecomunicações (SBRT)*, Juiz de Fora, Brasil, set. 2015.
- [8] X. Wang and D. Perkins, "Cross-Layer Hop-by-Hop Congestion Control in Mobile Ad Hoc Networks," in *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, March 2008, pp. 2456–2461.
- [9] H. Zhai, X. Chen, and Y. Fang, "Improving Transport Layer Performance in Multihop Ad Hoc Networks by Exploiting MAC Layer Information," *Wireless Communications, IEEE Transactions on*, vol. 6, no. 5, pp. 1692–1701, May 2007.
- [10] 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.
- [11] D. Gomez, R. Agüero, 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.
- [12] G. Holland and N. Vaidya, "Analysis of tcp performance over mobile ad hoc networks," *Wireless Networks*, vol. 8, no. 2-3, pp. 275–288, 2002.
- [13] X. Yu, "Improving TCP Performance over Mobile Ad Hoc Networks by Exploiting Cross-layer Information Awareness," in *MobiCom*, ser. *MobiCom '04*. New York, NY, USA: ACM, 2004, pp. 231–244.
- [14] 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.
- [15] S. Prasanthi, M. Lee, and S.-H. Chung, "A sender side algorithm for handling retransmission timeouts of tcp newreno over multi-hop wireless networks," in *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on*, March 2013, pp. 661–665.
- [16] M. Patil and A. Patil, "Enhancing tcp performance in multihop ad hoc networks," in *Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on*, July 2013, pp. 1–6.
- [17] 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.
- [18] ns-3, "ns-3," <http://www.nsnam.org>, 2013, acesso em: 12 abril. 2016, 22:17.
- [19] 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.