

# Reduzindo a sobrecarga da descoberta de rotas no roteamento em redes ad hoc móveis

Adrian Boaventura da Silva, Clederson Bahl e Dotti e Luiz Carlos Pessoa Albini

**Resumo**—Redes ad hoc são redes de dispositivos móveis que se comunicam por meio de enlaces sem fio e sem utilizar qualquer infra-estrutura. Para construir rotas nestas redes, são utilizados protocolos de roteamento completamente distribuídos e que respondam rapidamente a quebras de enlaces devido à mobilidade. Um dos principais protocolos de roteamento para redes ad hoc é o *Ad Hoc On-Demand Distance Vector (AODV)*. Este artigo apresenta duas modificações extremamente fáceis de ser incorporadas ao AODV para reduzir a sua sobrecarga, com um impacto mínimo sobre a taxa de entrega. A redução da sobrecarga implica em uma economia da energia, pois os nós da rede passam a ter menos mensagens para transmitir, processar e armazenar. Em alguns casos, é possível um aumento na taxa de entrega e uma redução do atraso na construção de rotas, pois reduzindo-se a sobrecarga libera-se banda para a transmissão de mensagens de dados.

**Palavras-Chave**—Redes Ad Hoc, Roteamento, Sobrecarga.

## I. INTRODUÇÃO

O uso de notebooks, celulares e PDAs, além de outros dispositivos, cresceu significativamente nos últimos anos, incorporando as redes sem fio ao dia-a-dia das pessoas. Dentre as tecnologias de redes sem fio, as redes ad hoc móveis (MANETs - Mobile Ad Hoc Networks) estão entre as que mais se destacam. Essas redes não possuem uma infraestrutura pré-existente ou centralizada e o seu funcionamento deve ser mantido pelos próprios nós, de forma autônoma, adaptativa e auto-organizada [1]. Devido às suas características e flexibilidade, essas redes são alternativas atrativas para ambientes em que a infraestrutura de rede é muito cara ou possa estar comprometida. Além disso, estudos indicam que a próxima geração da comunicação sem fio unirá os conceitos e tecnologias das redes sem fio convencionais e das redes sem fio sem infraestrutura [2]. Os termos "unidades" e "nós" são usados como sinônimos neste artigo.

MANETs formam um sistema de cooperação distribuído. Como o raio de alcance das transmissões sem fio é limitado, as comunicações são geralmente multi-saltos, sendo que os nós devem funcionar como roteadores para mensagens de outros nós. Para garantir a entrega das mensagens, as MANETs utilizam protocolos de roteamento distribuídos que devem se adaptar rapidamente a mudanças na topologia da rede devido à mobilidade [3], [4]. Os protocolos de roteamento para MANETs podem ser classificados em proativos, reativos e híbridos. Os protocolos proativos [5] mantêm uma tabela de roteamento atualizada para chegar em qualquer nó da rede. A latência destes protocolos é muito baixa, mas dependendo

da velocidade de movimentação dos nós, a sobrecarga pode ser alta. Os protocolos reativos [6], [3], por outro lado, não mantêm informações atualizadas. Quando um nó deseja enviar uma mensagem a outro nó da rede, ele deve primeiramente construir uma rota para o destino e só, então, enviar a mensagem. Embora eles possuam uma latência alta, a sobrecarga é muito baixa, pois não precisam manter nenhuma informação atualizada. Por fim, os protocolos híbridos [7], [4], [8] misturam características dos proativos e dos reativos. Geralmente, a parte proativa é feita em áreas pequenas e concentradas da rede e a parte reativa é usada para alcançar o resto da rede. Dentre os protocolos de roteamento para MANETs, um dos principais é o *Ad Hoc On-Demand Distance Vector (AODV)* [3].

Um ponto crucial para as redes sem fio é o consumo de energia, pois todos os nós usam baterias como fontes de energia. Este trabalho apresenta duas melhorias extremamente fáceis de ser incorporadas ao protocolo AODV para reduzir a sua sobrecarga. As duas melhorias consistem em (i) utilizar inundações para múltiplos destinos e (ii) descartar aleatoriamente mensagens de descoberta de rotas. É importante salientar que a redução da sobrecarga implica em uma economia da energia, pois os nós da rede tem menos mensagens para transmitir, processar e armazenar. Em alguns casos, é possível inclusive aumentar a taxa de entrega, pois reduzindo-se a sobrecarga, reduz-se também a quantidade de mensagens descartadas por estouro de filas. Além disso, com a redução da sobrecarga também é possível, em casos específicos, a redução do atraso na construção de rotas, pois elimina-se a transmissão de diversas mensagens de controle, liberando-se banda para a transmissão de mensagens de dados.

Resultados de simulação demonstram as vantagens das duas abordagens comparando-se os seus resultados com o AODV original nos mesmos cenários. É importante salientar que as vantagens das duas modificações dependem do tráfego da rede. No primeiro caso, usando *floodings* para múltiplos destinos, é possível reduzir o atraso e a sobrecarga e, ao mesmo tempo, aumentar a taxa de entrega, desde que as origens tenham tráfego para diversos destinos simultaneamente. Já no segundo caso, usando-se o descarte aleatório, é possível aumentar a taxa de entrega e reduzir a sobrecarga desde que o tráfego entre a origem e o destino seja alto.

O restante deste artigo está organizado da seguinte forma: a seção 2 apresenta o AODV; a seção 3 detalha as duas modificações; a seção 4 contém os resultados de simulação e a a seção 5 apresenta as conclusões e os trabalhos futuros.

## II. AD HOC ON-DEMAND DISTANCE VECTOR

Os criadores do AODV [3] classificam o protocolo como um sistema de aquisição de rotas puramente sob demanda, porque as unidades não dependem de rotas ativas nem trocam informações de roteamento periodicamente. O AODV usa números de sequência para garantir que as rotas não possuam laços e para assegurar que as unidades possuam as informações de roteamento mais recentes. Todas as unidades devem manter uma tabela de roteamento contendo a próxima unidade para a qual a mensagem deve ser enviada para chegar ao destino. Por exemplo, quando  $i$  quer enviar uma mensagem para  $j$ , ele verifica na sua tabela de roteamento para quem deve enviar a mensagem, para que a mensagem chegue ao destino,  $k$ . Quando  $k$  recebe a mensagem, ela repete o processo, verificando na sua tabela para quem deve mandar a mensagem para que ela chegue no destino,  $l$ , e envia a mensagem para  $l$ . Este processo é repetido até que a mensagem chegue ao destino. Caso a origem não saiba como chegar ao destino, ela precisa procurar por uma rota até o destino.

Para procurar uma rota, a origem envia uma mensagem de *Requisição de Rota* (RREQ), inundando a rede. Caso uma unidade receba o RREQ mais de uma vez, ela simplesmente descarta o RREQ duplicado. Os RREQs carregam um identificador que, em conjunto com o endereço da origem, permite que eles possam ser unicamente identificados por todas as unidades. A rota do destino para a origem é estabelecida pelo RREQ à medida que ele atravessa a rede. Quando uma unidade recebe um RREQ, ela guarda na sua tabela de roteamento o endereço da unidade que transmitiu o RREQ para ela. Esta informação serve também para informar para a unidade atual o caminho entre ela e a origem do RREQ. À medida que o RREQ atravessa a rede ele constrói a rota de todas as unidades pelas quais ele passa até a origem do RREQ.

A rota da origem para o destino é construída pela mensagem de *Resposta de Rota* (RREP). Uma unidade manda uma mensagem de RREP em resposta a um RREQ quando ela é o destino do RREQ ou quando ela possui uma rota suficientemente nova para o destino do RREQ. O RREP é enviado através do caminho construído pelo RREQ. Enquanto o RREP segue o caminho do destino para a origem, ele vai estabelecendo o caminho da origem para o destino. Quando uma unidade recebe um RREP, ela guarda na sua tabela o endereço da unidade da qual ela recebeu o RREP, construindo o caminho entre a origem e o destino da mesma forma que foi feito no RREQ. Quando o RREP chega ao seu destino, as duas rotas estão estabelecidas e pode-se iniciar a transmissão dos dados.

Durante a transmissão dos dados, o movimento dos nós pode causar a quebra da rota estabelecida. Caso seja o movimento da origem que cause a quebra da rota, ela precisa reiniciar o processo de descoberta de rota. Se a quebra da rota for causada pelo movimento de qualquer outra unidade, o seu predecessor na rota detecta a quebra do enlace e deve propagar uma notificação de *Quebra de Rota* (RERR) para os seus predecessores. Este processo é repetido até que a notificação de quebra de rota chegue na origem dos dados, que deve, então, reiniciar o procedimento de descoberta de rota.

## III. REDUZINDO A SOBRECARGA

Este artigo propõe duas modificações ao protocolo AODV: (i) o uso de inundação para múltiplos destinos; (ii) o descarte aleatório de mensagens de RREQ. Ambas serão detalhadas nas próximas seções.

### A. Inundação para Múltiplos Destinos

A técnica de *Inundação para Múltiplos Destinos* foi inicialmente proposta em [4]. Ela consiste em alterar a mensagem de construção de rotas para procurar diversas rotas para destinos diferentes simultaneamente, ao invés de enviar uma mensagem para cada destino procurado. Esta modificação é feita no pacote de *Requisição de Rota* (RREQ). Quando um nó envia um RREQ, ele procura em toda a sua fila de mensagens a ser enviadas e envia um único RREQ para todos os destinos de pacotes de dados para os quais ainda não possui uma rota.

Ao receber este novo RREQ, uma unidade continua verificando se é a segunda vez que este pacote passa por ela. Se a resposta for afirmativa, a mensagem é descartada. A unidade também verifica se ela é um dos destinos apontados no RREQ. Em caso afirmativo, ela envia para a origem um RREP da mesma forma que no AODV original.

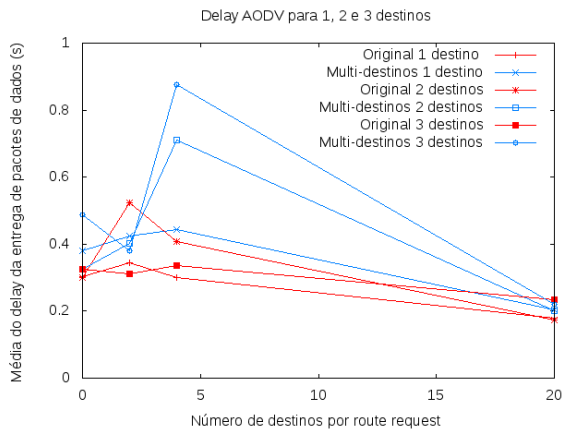
Outra modificação que o uso dos múltiplos destinos obriga é decidir o que fazer com o RREQ após recebê-lo e tratá-lo. No protocolo original, se uma unidade recebe um RREQ e ela é o destino da mensagem, o RREQ não tem mais utilidade e é descartado. Usando múltiplos destinos, mesmo que a unidade seja um dos destinos listados no RREQ, ela deve repassar a mensagem, pois ele ainda deve chegar aos outros destinos. Um nó que não é listado como um destino do RREQ simplesmente repassa a mensagem. Todo o restante do protocolo é idêntico ao AODV original.

### B. Descarte aleatório

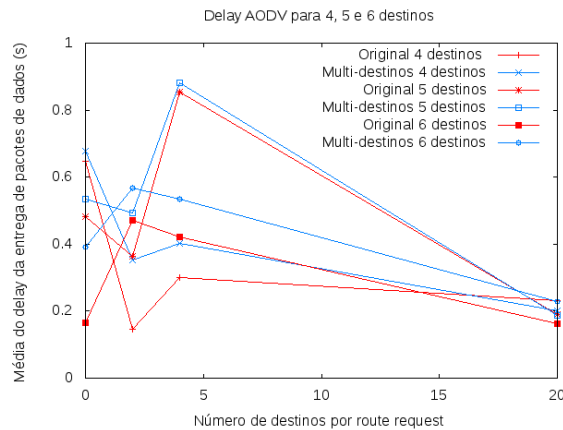
A técnica de *Descartes Aleatórios* consiste em cada unidade decidir se deve processar ou descartar cada RREQ recebido. Em outras palavras, quando uma unidade recebe um RREQ pela primeira vez, ela joga uma moeda. Se der cara, a mensagem é descartada; se der coroa, a mensagem é processada. Todo o restante do protocolo é idêntico ao original.

Com o uso dos descartes aleatórios, é possível reduzir o descarte das mensagens de dados por *buffer overflow*. Segundo [4], o descarte das mensagens de dados por *buffer overflow* é umas das principais causas da redução da taxa de entrega. Além disso, um dos principais causadores dos *buffer overflows* é o tráfego das requisições de rotas em broadcast. Como esta modificação reduz o tráfego das requisições de rotas, ela pode ter um impacto significativo na taxa de entrega do protocolo dependendo do tráfego da rede.

Esta modificação também reduz o tempo que as mensagens ficam nos *buffers* esperando para ser transmitidas. Isso porque reduzindo-se o tráfego, reduz-se a quantidade de mensagens nestes *buffers* e as mensagens são transmitidas mais rapidamente.

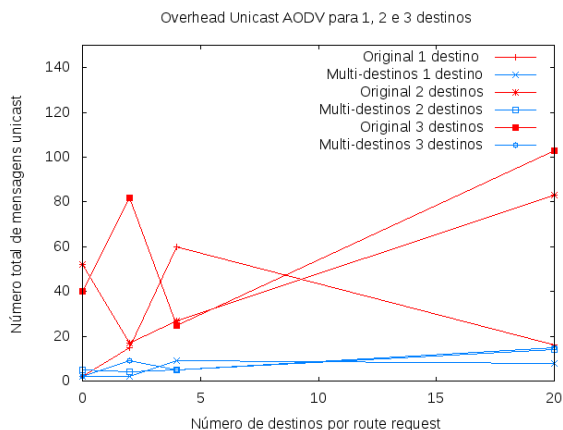


(a) 1, 2 e 3 múltiplos destinos.

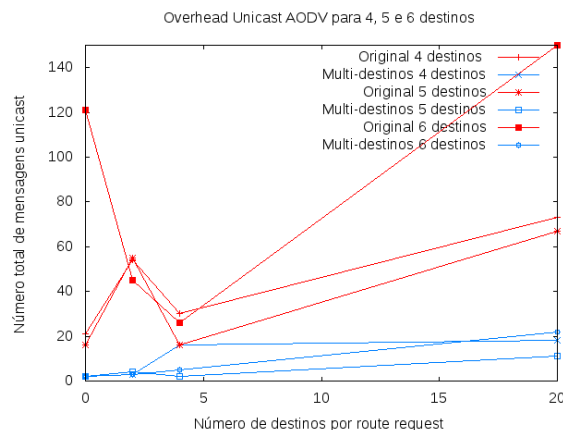


(b) 4, 5 e 6 múltiplos destinos.

Fig. 1. Atraso múltiplos destinos.



(a) 1, 2 e 3 múltiplos destinos.



(b) 4, 5 e 6 múltiplos destinos.

Fig. 2. Sobrecarga em unicast.

IV. RESULTADOS

O simulador NS-2 foi utilizado para testar o desempenho das duas modificações propostas para o AODV. A versão do software usada foi a 2.34 no sistema operacional Ubuntu 10.04 em um computador Intel Dual Core 1.95 Ghz com 2GB de memória RAM. Todos os resultados apresentados são médias de 35 simulações com 95% de intervalo de confiança. A tabela I contém os parâmetros comuns as duas modificações usados nas simulações.

TABELA I  
PARÂMETROS DE SIMULAÇÃO.

Simulador	ns-2.34
Dimensão dos cenários (m)	1000x1000
Quantidade de nós	50
Raio de transmissão (m)	120
Distribuição dos nós	Uniforme
Mobilidade	Random Waypoint
Protocolos MAC	802.11
Tempo de simulação (s)	300 segundos
Velocidade dos nós (m/s)	0 a 20

Os resultados para atraso (*delay*) apresentados são as médias de atraso em milissegundos para os pacotes de dados saírem das origens e chegarem nos destinos. A sobrecarga (*overhead*)

é o total de mensagens de controle usadas pelo protocolo de roteamento durante toda a simulação.

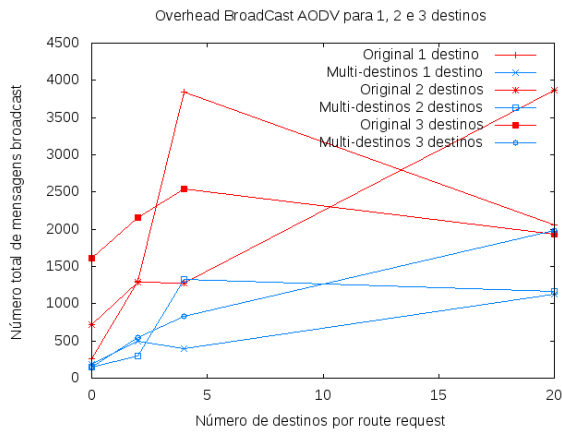
A. Múltiplos Destinos

A tabela II contém os parâmetros específicos usados nas simulações do uso dos *Inundação para Múltiplos Destinos*. As figuras 1, 2, 3 e 4 comparam o atraso, a sobrecarga do tráfego unicast e broadcast e a taxa de entrega (*delivery ratio*) entre o AODV original e usando-se múltiplos destinos para 2 à 6 destinos simultâneos.

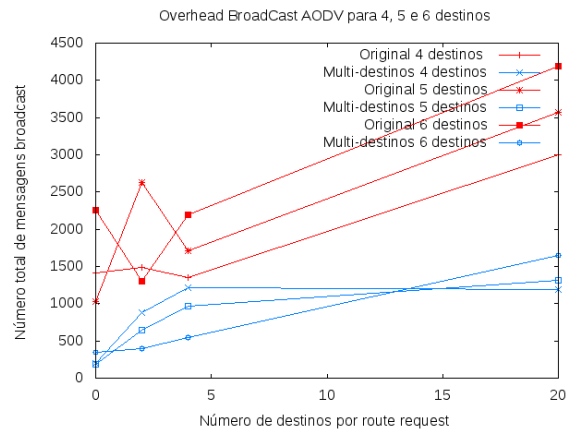
TABELA II  
PARÂMETROS DE SIMULAÇÃO MÚLTIPLOS DESTINOS

Tráfego	HTTP
Quantidade de origens / destinos	20
Múltiplos Destinos simultâneos	1 a 6

É possível notar pela figura 3 que o uso dos múltiplos destinos reduz a sobrecarga do tráfego em *broadcast* do protocolo. Além disso, pode-se verificar pelas figuras 1 e 4 que o uso dos múltiplos destinos pode reduzir o atraso da construção de rotas e aumentar a taxa de entrega. Entretanto, estes resultados

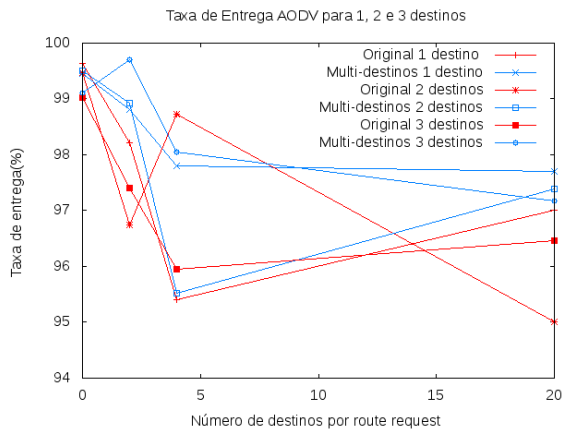


(a) 1, 2 e 3 múltiplos destinos.

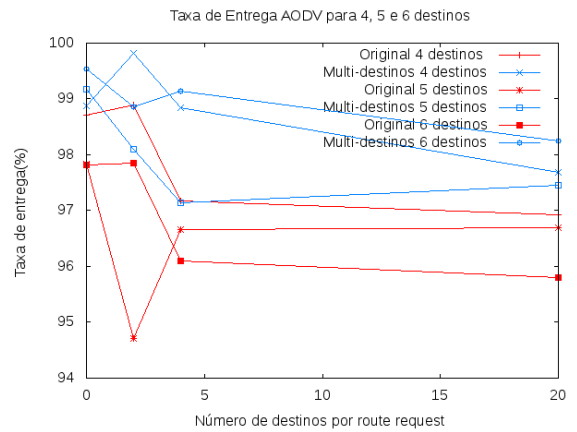


(b) 4, 5 e 6 múltiplos destinos.

Fig. 3. Sobrecarga em broadcast.



(a) 1, 2 e 3 múltiplos destinos.



(b) 1, 2 e 3 múltiplos destinos.

Fig. 4. Taxa de Entrega.

dependem do tipo de tráfego na rede. Se a origem precisar se comunicar com um único destino, esta modificação implica em resultados piores que a versão original.

**B. Descarte Aleatório**

A tabela III contém os parâmetros específicos usados nas simulações do uso do *Descarte aleatório*. O descarte aleatório implementado consiste em descartar aleatoriamente 50% das mensagens de construção de rotas antes de processá-las. Note que foram realizados testes com o descarte variando de 10% a 90%, entretanto neste artigo somente os resultados com 50% são apresentados devido a limitação de espaço e porque serem os melhores resultados obtidos.

TABELA III  
PARÂMETROS DE SIMULAÇÃO DESCARTE ALEATÓRIO.

Tráfego	CBR
Quantidade de origens / destinos	10 a 40
Mensagens de dados por segundo	2

Pode-se notar nas figuras 5, 6 e 7 que os ganhos no uso

do descarte aleatório é diretamente proporcional ao tráfego da rede. Quanto maior o tráfego da rede, maior o ganho. Isso se deve pelo fato de que aumentando o tráfego aumenta-se o descarte de mensagens de dados por *buffer overflow* na versão original do protocolo. Com o uso do descarte aleatório reduz-se os descartes por *buffer overflows*, aumentando a taxa de entrega e reduzindo a sobrecarga do protocolo. Além disso, o uso desta técnica também tem impacto no atraso do protocolo, figura 5. Isso porque reduzindo-se o tráfego, reduz-se o tamanho das filas para envio de mensagens e, com isso, as mensagens são enviadas mais rapidamente.

**V. CONCLUSÕES**

Este trabalho propõem duas melhorias ao protocolo de roteamento *Ad Hoc On-Demand Distance Vector* (AODV). A primeira consiste em alterar as mensagens de construção de rotas para procurarem diversos caminhos simultaneamente. A segunda consiste em descartar aleatoriamente mensagens de construção de rotas para reduzir a quantidade de mensagens descartadas por *buffer overflows*.

Conforme foi mostrado, as duas técnicas podem trazer ganhos consideráveis à rede, tanto no aumento da taxa de entrega

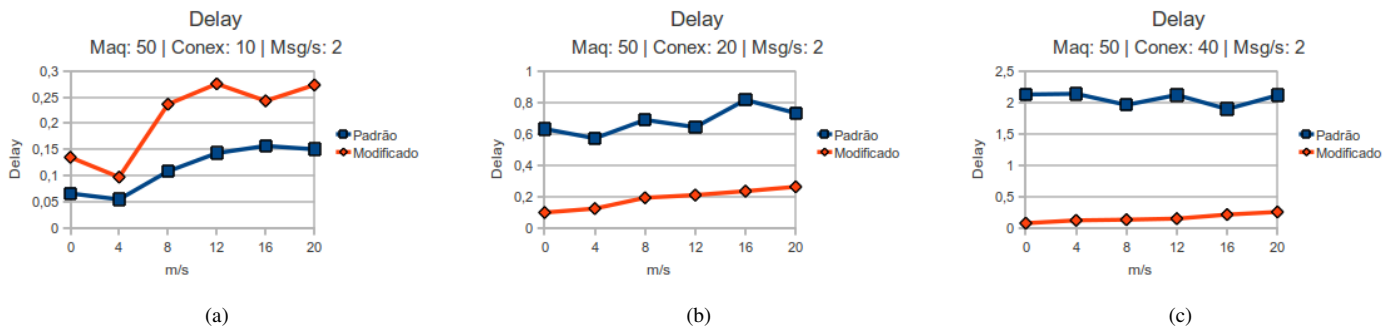


Fig. 5. Atraso usando descartes aleatórios.

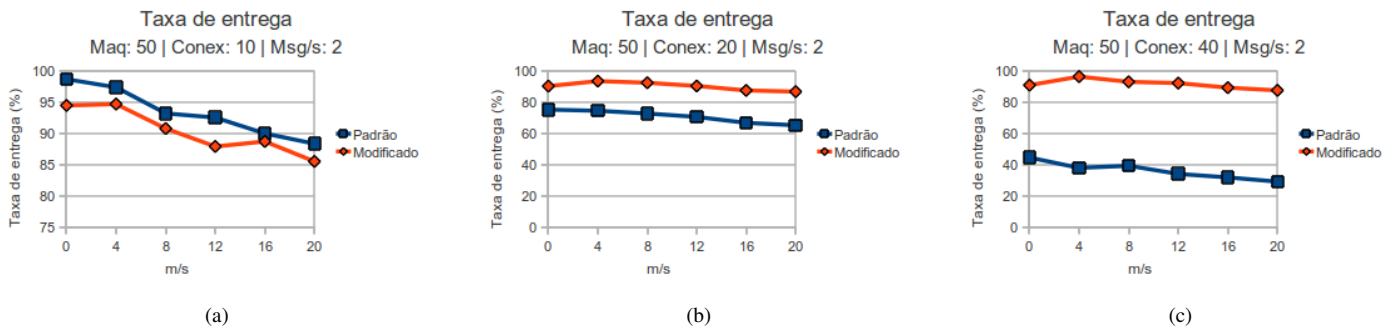


Fig. 6. Taxa de Entrega usando descartes aleatórios.

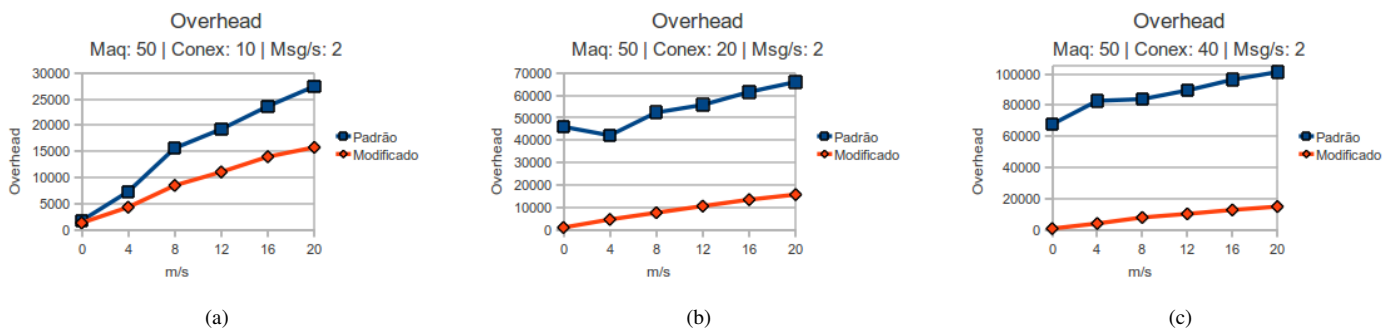


Fig. 7. Sobrecarga usando descartes aleatórios.

como na redução do atraso e da sobrecarga do protocolo. Entretanto, deve-se notar que estes ganhos são dependentes do cenário de uso da rede, e o uso ou não das melhorias propostas neste artigo deve ser analisado pelo administrador da rede considerando-se o tipo de tráfego da mesma.

Trabalhos futuros incluem o teste destas mesmas modificações em outros protocolos de roteamento para verificar se é possível obter os mesmos ganhos. Além disso, eles também incluem testes do uso das duas modificações simultaneamente.

REFERENCIAS

[1] I. Chlamtac, M. Conti, and J. J.-N. Liu, "Mobile ad hoc networking: imperatives and challenges," *Ad Hoc Networks*, vol. 1, no. 1, pp. 13–64, 2003.

[2] J. G. Jayanthi, S. A. Rabara, and A. R. M. Arokiajaraj, "Ipv6 manet: An essential technology for future pervasive computing," *Communication Software and Networks, International Conference on*, vol. 0, pp. 466–470, 2010.

[3] C. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector (AODV) routing," in *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 1999)*, 1999, pp. 90–100.

[4] L. Albini, A. Caruso, S. Chessa, and P. Maestrini, "Reliable routing in wireless ad hoc networks: The virtual routing protocol," *Journal of Network and Systems Management*, vol. 14, no. 3, pp. 335–358, Setembro 2006.

[5] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proceedings of the ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM 1994)*, 1994, pp. 234–244. [Online]. Available: [citeseer.nj.nec.com/perkins94highly.html](http://citeseer.nj.nec.com/perkins94highly.html)

[6] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, Imielinski and Korth, Eds. Kluwer Academic Publishers, 1996, vol. 353. [Online]. Available: [citeseer.ist.psu.edu/johnson96dynamic.html](http://citeseer.ist.psu.edu/johnson96dynamic.html)

[7] Z. Haas and M. Pearlman, "The performance of query control schemes for the zone routing protocol," *IEEE/ACM Transactions on Networking*, vol. 9, no. 4, pp. 427–438, 2001.

[8] A. Robba and P. Maestrini, "Routing in mobile ad-hoc networks: The virtual distance vector protocol," in *Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2007)*, 2007, pp. 1–9.