

# UMA FERRAMENTA PARA SUPORTE AO ENSINO DO PROTOCOLO AODV

DOI: 10.15552/2236-0158/abenge.v34n2p71-81

Karine S. de Almeida,<sup>1</sup> Tassalon F. da Silva,<sup>2</sup> Daniel G. Costa<sup>3</sup>

## RESUMO

Este artigo apresenta o desenvolvimento de um *software* para o ensino do protocolo de roteamento AODV (*Ad Hoc On-Demand Distance Vector*). O AODV é muito utilizado em diversas aplicações de redes de telecomunicações sem fio, a fim de possibilitar o encaminhamento de dados entre os dispositivos de origem e destino, quando uma infraestrutura de comunicação não está disponível. Devido à importância desse protocolo no cenário atual das telecomunicações, torna-se necessário o uso de ferramentas que possam contribuir para o ensino dos conceitos relacionados a essa tecnologia. Assim, cursos como o de engenharia de computação e engenharia de telecomunicações, por exemplo, poderão abordar os conceitos do protocolo AODV de forma diferenciada com o uso do *software* apresentado neste trabalho. Isso permitirá que alunos possam não apenas aprender, mas também simular, realizar análises e comparações e obter conclusões mais detalhadas sobre essa tecnologia.

**Palavras-chave:** Ferramenta de ensino; protocolo de roteamento; AODV.

## ABSTRACT

### A TOOL TO SUPPORT TEACHING OF PROTOCOL AODV

This article presents the development of a software tool for teaching a routing protocol, Ad Hoc On-Demand Distance Vector (AODV). AODV is broadly used in many applications of wireless telecommunications networks in order to allow routing of data between source and destination devices, when a communication infrastructure is not available. Due to the importance of this protocol in the current telecom scenario, it becomes necessary to use tools that can contribute to teaching the concepts related to this technology. Thus, courses such as computing and telecommunications engineering, for instance, can treat the concepts of AODV protocol in a different way using the software showed in this paper. This will allow students to not only learn, but also to simulate, to perform analysis and comparisons and to get detailed conclusions about this technology.

**Keywords:** Teaching tool; routing protocol; AODV.

1 Mestranda no Programa de Pós-Graduação em Computação Aplicada da Universidade Estadual de Feira de Santana (UEFS); Engenheira de Computação; karine.almeidaa@gmail.com

2 Mestrando no Programa de Pós-Graduação em Computação Aplicada da Universidade Estadual de Feira de Santana (UEFS); Engenheiro de Computação; tassalonfs@gmail.com

3 Doutor em Engenharia Elétrica pela Universidade Federal do Rio Grande do Norte (UFRN); Professor Adjunto da Universidade Estadual de Feira de Santana (UEFS); danielgcosta@uefs.br

## INTRODUÇÃO

Atualmente, presencia-se um notório crescimento na área de redes sem fio. Essas redes possibilitam a mobilidade de dispositivos, flexibilidade, diminuição de custos de infraestrutura, instalações em áreas de difícil cabeamento, entre outras características (KUROSE; ROSS, 2010). Esses aspectos permitem que a tecnologia móvel ofereça aos usuários a possibilidade de acesso a serviços e o compartilhamento de informações a qualquer instante e em diferentes lugares, propiciando, assim, seu uso em diversas aplicações, como, por exemplo, redes de sensores sem fio e redes veiculares (VIEIRA; CELESTINO JÚNIOR, 2001; CAO; LU, 2010).

Uma rede sem infraestrutura definida, mais conhecida como rede *ad-hoc*, corresponde a um conjunto de nós móveis, sem fio, com a capacidade de auto-organização em uma topologia arbitrária e temporária, com a inexistência de uma entidade central (KUROSE; ROSS, 2010). Um nó de uma rede *ad-hoc* pode se comunicar diretamente com outros nós que estiverem no seu raio de alcance, sendo chamados de nós vizinhos. Caso a distância entre dois nós seja maior do que o raio de cobertura, é possível estabelecer uma rota entre os dois, tornando a comunicação fim-a-fim possível. Isso é feito através de nós móveis intermediários, que agem como roteadores, encaminhando pacotes de dados para outros nós na rede.

Nesse contexto, este artigo apresenta o desenvolvimento de um *software* para suporte ao ensino de um protocolo de roteamento usado em redes *ad hoc*, denominado *Ad Hoc On-Demand Distance Vector* (AODV), o qual estabelece formas de envio de pacotes entre dois nós quaisquer, em uma rede sem fio. O *software* desenvolvido apresenta as principais características do protocolo AODV, por meio da simulação de seu funcionamento, sendo possível, também, fazer análises de desempenho através de gráficos, permitindo que o usuário possa compreender as vantagens e limitações desse protocolo de comunicação.

Este artigo está organizado da seguinte forma. Na Seção 2, é apresentada uma revisão da literatura sobre o protocolo AODV, com enfoque nas suas principais características. Na Seção 3, descrevemos

em detalhes como foi desenvolvido o *software* para ensino do AODV e explicamos o seu funcionamento, bem como as possibilidades de análises de desempenho que ele oferece. Na Seção 4, apresentamos algumas simulações realizadas pelo *software* para demonstrar seu potencial didático e exemplificamos como ele pode ser usado para gerar resultados e fazer análises de desempenho. Por fim, na Seção 5, são feitas as considerações finais, resumindo os pontos mais relevantes deste trabalho, seguidas pelas referências.

## PROTOCOLO AODV

O AODV é um protocolo de roteamento que foi desenvolvido para utilização em redes *ad-hoc* com dezenas ou mesmo milhares de nós móveis. Seu principal objetivo é se adaptar de forma rápida e dinâmica às variações da rede (PERKIN; ROYER, 1999). Dessa maneira, o AODV possui um mecanismo de descoberta de rotas para otimizar o uso do canal de transmissão, ao mesmo tempo em que diminui o uso de memória e do processamento.

O mecanismo de descoberta de rotas do AODV atua sob demanda, de forma que não é preciso que um nó conheça uma rota até outro nó a todo instante, mas apenas quando um nó deseja se comunicar com outro nó da rede (PERKIN *et al.*, 2003). O processo de descoberta de rotas do AODV é iniciado quando um nó fonte envia para os seus vizinhos uma mensagem *route request* (RREQ) e cada vizinho encaminha essa mensagem para os seus respectivos vizinhos, e assim sucessivamente (MURTHY; MANOJ, 2004). Essa etapa da descoberta de rotas termina quando a mensagem alcança um nó que possui uma rota para o destino ou quando alcança o próprio nó destino. Além disso, é importante ressaltar que, quando um nó recebe um RREQ, ele insere uma entrada na tabela de roteamento referente ao nó que encaminhou a mensagem (PERKIN *et al.*, 2003).

Após alcançar o destino, ou algum nó intermediário que conhece uma rota para o destino, ocorre a segunda etapa do processo de descoberta de rotas, que diz respeito a uma mensagem de resposta denominada *route response* (RREP), enviada via *unicast*, através do caminho reverso configurado na primeira etapa (PERKIN; ROYER, 1999). Por se tratar de

uma rede de nós móveis, o protocolo tenta garantir que as rotas criadas sejam sempre válidas. Para isso, o AODV realiza um mecanismo conhecido como manutenção de rotas. A manutenção de rotas é acionada quando um nó detecta a queda de um enlace (MURTHY; MANOJ, 2004). Então, uma mensagem informando essa queda de enlace, o *route error* (RERR), é enviada e propagada através dos nós que pertencem ao caminho, sendo disparado um novo processo de descoberta de rota, atualizando, assim, as tabelas de roteamento. O reconhecimento da queda de um enlace é feito através da detecção de vizinhança, que atua enviando periodicamente mensagens de *broadcast* (mensagens HELLO) que servem para confirmar a conectividade entre vizinhos (PERKIN; ROYER, 1999). Quando um vizinho não recebe uma mensagem durante certo período de tempo, então é considerada a queda de um enlace (MURTHY; MANOJ, 2004).

O AODV tem uma política de fazer com que os nós intermediários de um caminho introduzam dinamicamente entradas em suas tabelas de roteamento para cada destino (PERKIN *et al.*, 2003). Entretanto, seus pacotes não precisam possuir todo o caminho do nó fonte até o nó destino, o que diminui a sobrecarga da rede. Em adição a isso, esse protocolo de roteamento é baseado na combinação de outros dois algoritmos, o *Dynamic Source Routing* (DSR) e o *Destination-Sequenced Distance-Vector* (DSDV) (TOMAR *et al.*, 2011). O AODV adota os mecanismos de descoberta e manutenção de rotas usados no DSR e utiliza uma perspectiva melhorada do DSDV no sentido de tratar erros. Outra diferença entre o AODV e o DSDV é que AODV busca eliminar a necessidade de um *broadcast* global para o roteamento de informações, diferentemente do DSDV, que possui essa característica, limitando, assim, a sua escalabilidade. Sem o *broadcast* global, o AODV minimiza a latência quando novos nós são incluídos à rede (TOMAR *et al.*, 2011).

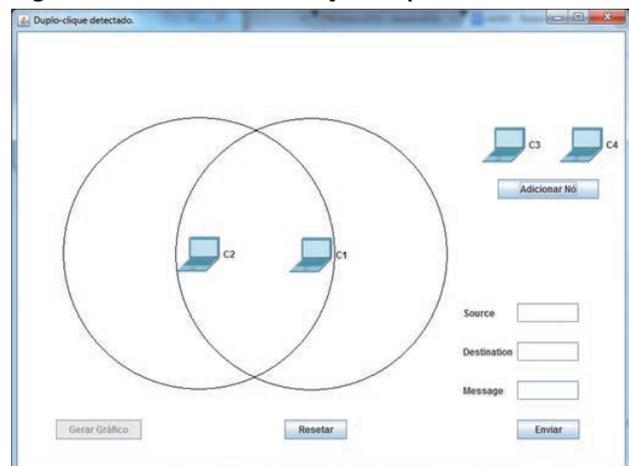
Em geral, alguns pontos que devem ser observados ao avaliar quantitativamente o desempenho de protocolos de roteamento são: *throughput* de dados fim-a-fim e o atraso dos pacotes; tempo de descobrimento da rota, principalmente nos algoritmos que operam sob demanda; e eficiência com relação à quantidade de dados de controle que precisa ser

trafegada na rede (TOMAR *et al.*, 2011). Entre os critérios de avaliação citados, o *software* desenvolvido é capaz de avaliar a sobrecarga de informações de controle no protocolo AODV, possibilitando ao usuário analisar quanto de dados de controle é necessário para transmitir dados úteis pela rede, levando em consideração a quantidade de nós móveis da rede.

## Ferramenta desenvolvida

O *software* para suporte ao ensino do protocolo AODV foi desenvolvido usando a linguagem de programação JAVA, com auxílio da biblioteca *jfreechart*, para a criação dos gráficos. O primeiro passo para o desenvolvimento do simulador foi criar um ambiente propício para o teste do AODV. Tal ambiente iria, então, permitir aos usuários a inserção de dispositivos móveis, também chamados de *hosts*, ou simplesmente de nós. Campos para especificar os *hosts* de origem e destino, bem como a mensagem a ser transmitidas entre eles, fazem parte da interface de simulação. A Figura 1 ilustra a interface desenvolvida para a simulação do protocolo AODV.

Figura 1: Interface de simulação do protocolo AODV.



Na interface desenvolvida, os nós (representados como computadores portáteis) podem ser arrastados e reposicionados em qualquer local do cenário, possibilitando simular os efeitos da mobilidade dos nós. A qualquer momento, novos nós podem ser adicionados ao cenário, não havendo limites para a quantidade de *hosts* que podem ser inseridos. O botão **Resetar** reinicia a interface, retirando os nós que foram inseridos e permitindo ao usuário realizar outra simulação. O botão **Enviar**

dá início ao processo de execução do protocolo de roteamento AODV, fazendo com que o *host* de origem consiga entregar uma mensagem para o *host* de destino.

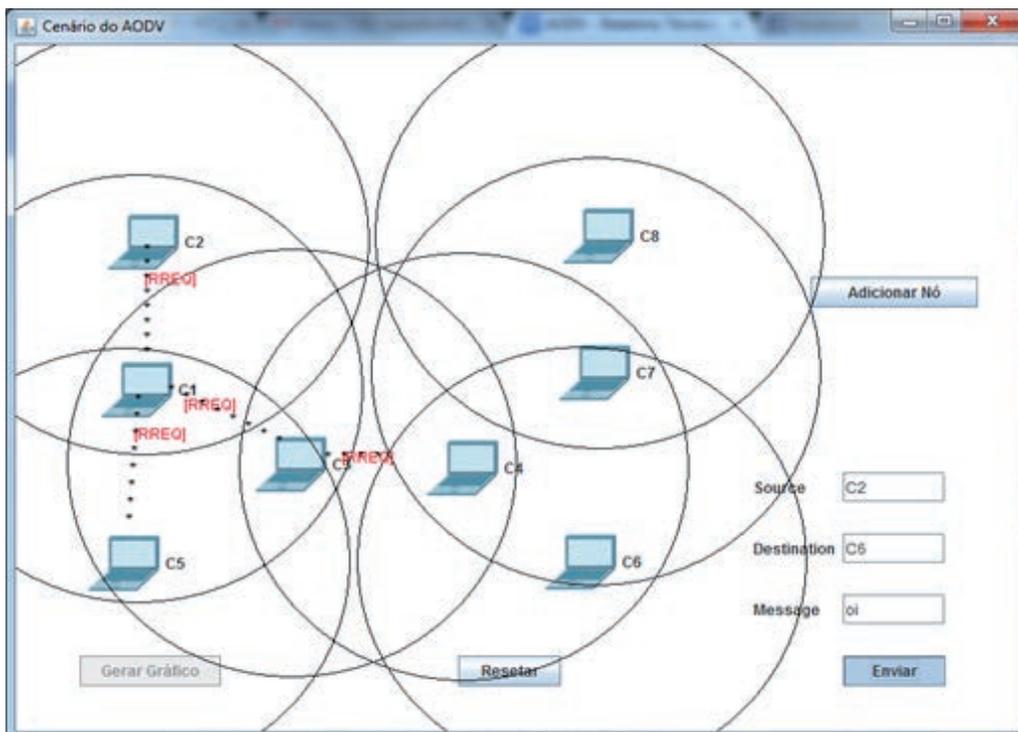
O círculo desenhado em volta de cada *host* inserido no cenário representa o seu raio de alcance, numa transmissão sem fio. A distância do alcance dos *hosts* é pré-fixada no programa, e é igual para todos os dispositivos. É a partir desse raio de alcance que os *hosts* detectam seus vizinhos e criam suas tabelas de vizinhos, necessárias para o posterior descobrimento de rotas para *hosts* remotos (não vizinhos). O raio de alcance de cada *host* é exibido após um duplo clique sobre o *host* posicionado no cenário. Em alguns momentos, durante a execução do protocolo AODV, o raio de alcance de todos os *hosts* é exibido simultaneamente, para facilitar a compreensão do seu funcionamento.

Quando um *host* está dentro do raio de alcance de outro *host*, eles são considerados vizinhos. Para implementar a detecção de vizinhos, calcula-se a distância entre dois nós e se verifica se o raio de alcance é maior que essa distância entre eles. Caso o raio de alcance seja maior do que a distância entre

eles, então, os nós envolvidos são considerados vizinhos e essa informação deve ser armazenada em cada *host*, em sua respectiva tabela de vizinhos.

Após adicionar *hosts* no cenário e definir quem será o nó de origem e o de destino, a mensagem pode ser enviada clicando-se no botão **Enviar**. Como o protocolo de roteamento AODV é um protocolo sob demanda, as rotas para os *hosts* são obtidas apenas quando uma mensagem precisa ser enviada. Quando um *host* precisa enviar uma mensagem e não conhece uma rota para o destino, ele envia um *route request* (RREQ) em *broadcast*. Os seus vizinhos recebem esse *broadcast* e verificam se conhecem uma rota para o destino da mensagem. Caso um dos vizinhos conheça uma rota para o destino ou caso seja o próprio destino da mensagem, ele envia então um *route reply* (RREP) *unicast* para a origem, informando a rota. Caso nenhum dos vizinhos conheça uma rota para o destino, então, eles repassam o RREQ até alcançar o nó de destino da mensagem ou um nó que tenha uma rota para o destino. A Figura 2 mostra o *host* C2 enviando um RREQ para descobrir uma rota para o *host* C6.

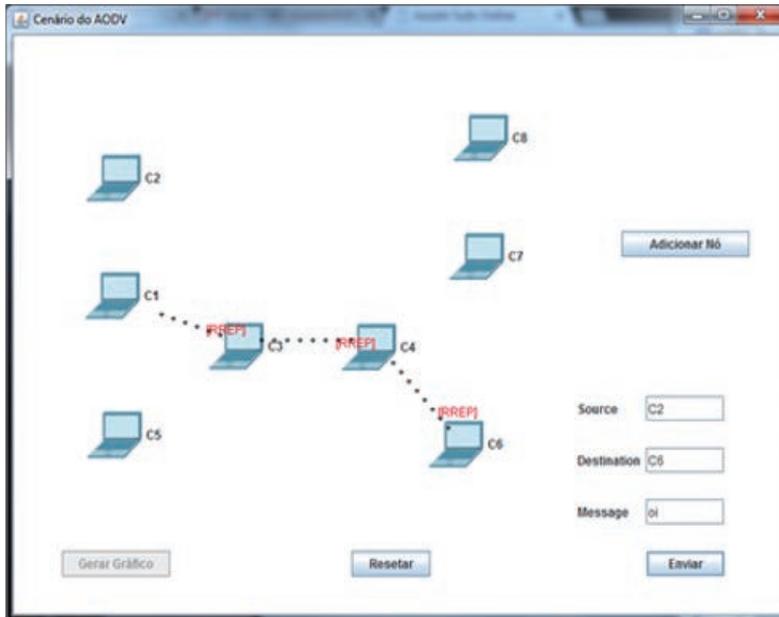
Figura 2: Envio de RREQs em busca de uma rota para o destino da mensagem.



Durante o processo de envio de RREQ, as tabelas de roteamento dos nós intermediários vão sendo montadas. Através dos RREQs, eles aprendem como alcançar o *host* de origem da mensagem. Quando algum *host* tem uma rota para o destino ou

é o próprio destino da mensagem, ele responde ao RREQ com uma resposta RREP, informando a rota para alcançar o destino. A Figura 3 mostra o RREP sendo encaminhado para o *host* de origem (que nesse exemplo é o C2).

Figura 3: Envio de RREP para informar para origem (C2) de uma rota para o destino (C6).



O pacote RREP é encaminhado pelos nós até alcançar o *host* de origem da mensagem. Com isso, o nó de origem aprende uma rota para o destino e, a partir desse momento, ele pode enviar a mensagem para o destinatário. Durante o encaminhamento dos RREP, os nós intermediários aprendem novas rotas

para adicionar na tabela de roteamento. A tabela de roteamento de um *host* pode ser visualizada dando um duplo clique sobre o *host*. A Figura 4 ilustra a tabela de roteamento de vários *hosts* do cenário exibido na Figura 3, após C2 receber o RREP.

Figura 4: Tabela de roteamento dos *hosts* C2, C1, C3, C4, C6 e C5.

Tabela de Rotas de C2			
Destination	Hops	Next Hop	Precursors
C1	1	C1	
C6	4	C1	

Tabela de Rotas de C1			
Destination	Hops	Next Hop	Precursors
C2	1	C2	C3,
C3	1	C3	
C6	3	C3	C2,

Tabela de Rotas de C3			
Destination	Hops	Next Hop	Precursors
C1	1	C1	
C2	2	C1	C4,
C4	1	C4	
C6	2	C4	C1,

Tabela de Rotas de C4			
Destination	Hops	Next Hop	Precursors
C3	1	C3	
C2	3	C3	C6,
C6	1	C6	C3,

Tabela de Rotas de C6			
Destination	Hops	Next Hop	Precursors
C4	1	C4	
C2	4	C4	

Tabela de Rotas de C5			
Destination	Hops	Next Hop	Precursors
C1	1	C1	
C2	2	C1	
C3	1	C3	

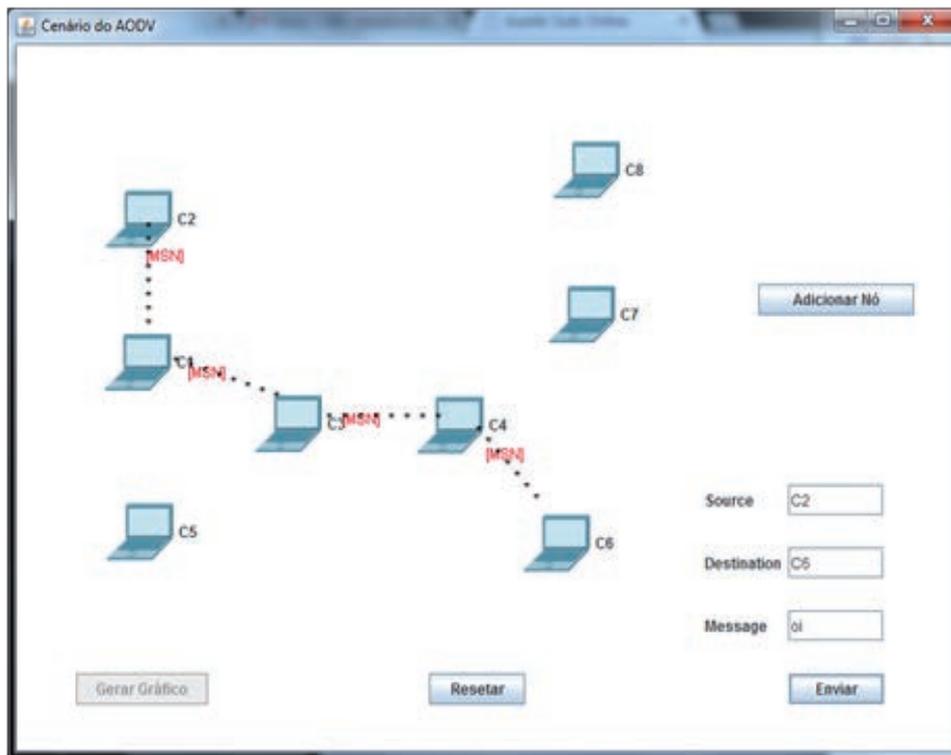
Cada rota da tabela de roteamento implementada é composta pelos campos:

- **Destination** – Campo responsável por identificar qual *host* é acessível com as informações dessa rota.
- **Hops** – Campo que informa por quantos *hosts* o pacote terá que passar para chegar ao destino.
- **NextHop** – Campo que determina qual é o próximo *host* a que deve ser enviado o pacote para se alcançar o destino.
- **Precursors** – Campo responsável em determinar os *hosts* que também usam essa rota para encaminhar pacotes. Esse campo é útil para informar aos *hosts* quando uma rota se torna inalcançável.

Além disso, há diversos outros campos que o protocolo AODV original determina para compor uma rota da tabela de roteamento, porém, eles não foram usados nessa implementação por questões de simplificação no desenvolvimento do simulador. Os campos usados são suficientes para simular as principais funcionalidades do protocolo, sem muitos prejuízos.

Quando o *host* de origem recebe um RREP informando uma rota para o destino da mensagem, ele pode de fato enviar a mensagem que deseja. Caso o *host* de origem não receba um RREP, ele considera que não há uma rota para o destino e, então, a mensagem é descartada. A Figura 5 mostra o envio da mensagem do *host* C2 até o *host* de destino C6.

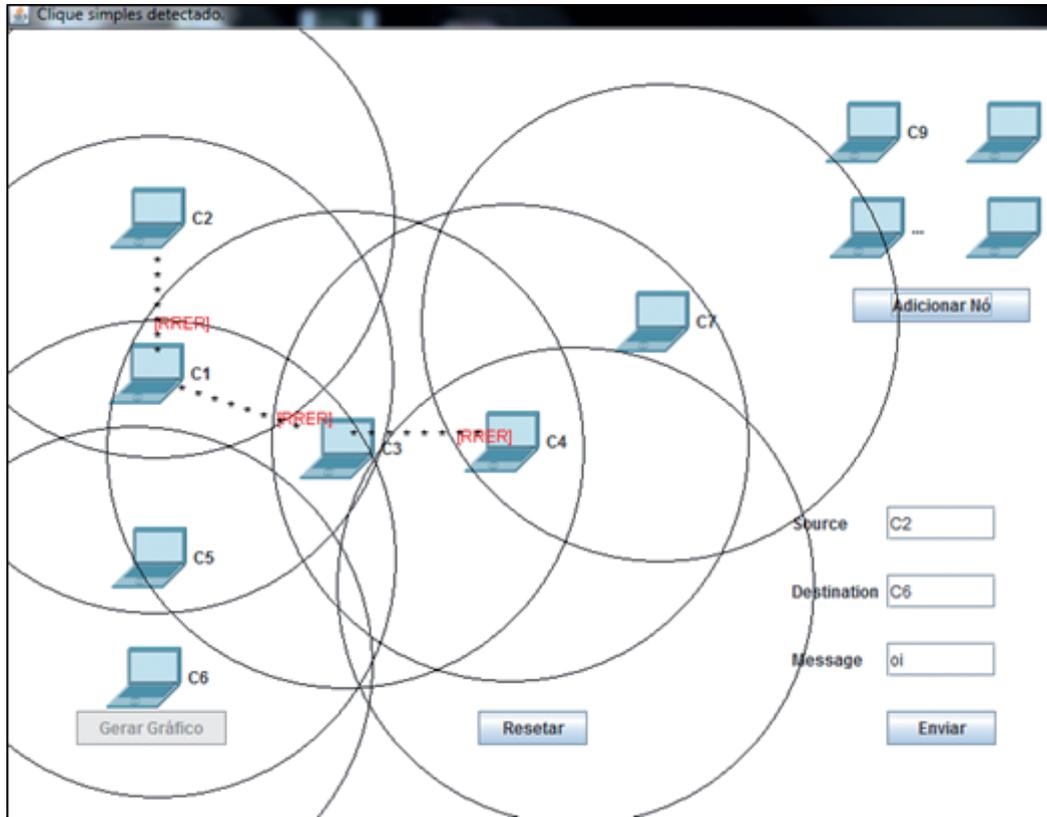
Figura 5: Envio da mensagem de C2 para C6, após a origem aprender uma rota para destino.



Pacotes *route error* (RERR) são enviados quando uma rota se torna inativa. Isso pode acontecer pela movimentação dos nós. Quando um *host* percebe que um nó vizinho se tornou inalcançável, ele envia um pacote RERR para todos os *hosts* que tinham uma rota para esse nó, informando que a rota não está mais disponível. Isso é possível porque o *host* que percebeu a ausência do vizinho armazena uma tabela de precursores para a rota que foi desativa-

tiva. Então, os *hosts* removem de suas tabelas de roteamento a rota para o destino que se tornou inalcançável. Caso o *host* que se movimentou tenha se tornado vizinho de outro *host* no cenário, será possível descobrir uma nova rota para ele através do envio de novos pacotes de requisição RREQ. A Figura 6 mostra o encaminhamento de mensagens RERR após a mudança de posição do nó C6.

Figura 6: Envio de RERR quando o *host* 6 muda de posição no cenário.



Por fim, foi desenvolvido um mecanismo para a geração de gráficos, com o objetivo de avaliar o desempenho e a sobrecarga da rede sob a operação do protocolo de roteamento AODV. Assim, as mensagens transmitidas pela rede são registradas e usadas para gerar gráficos que relacionam a quantidade de dados úteis (mensagens) transmitidos com a quantidade de pacotes de controle (RREQ, RREP e RERR) necessários para tornar a comunicação possível. Para gerar um gráfico, basta clicar sobre o botão **Gerar Gráfico**. Esse botão só se torna ativo após o envio de pelo menos três mensagens. Os gráficos obtidos são dinâmicos e irão variar de acordo com o cenário de teste utilizado, ou seja, a quantidade de mensagens transmitidas, a quantidade de *hosts* inseridos no cenário e a movimentação desses *hosts* irão influenciar diretamente no gráfico resultante.

Os gráficos podem ser usados para avaliar diversas situações, como, por exemplo, a influência da quantidade de *hosts* para a sobrecarga da rede, bem como a diferença na sobrecarga da rede com a variação da quantidade de mensagens transmitidas. A sobrecarga ocasionada com a movimentação dos *hosts*

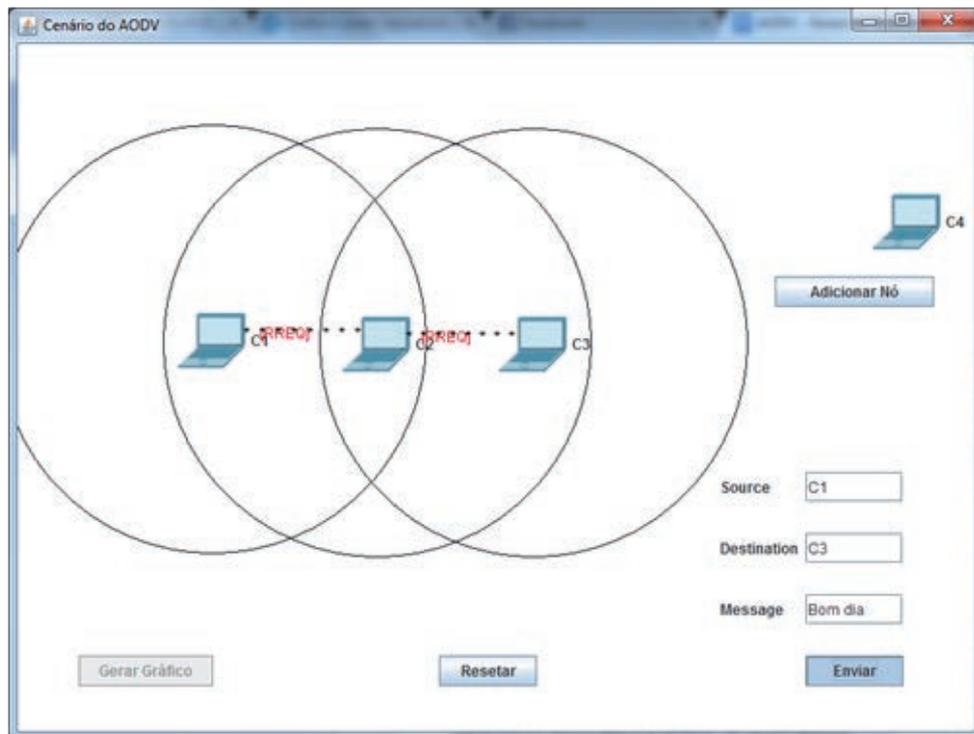
durante o envio de mensagens também é mais uma das diversas possibilidades de análise que o mecanismo de geração de gráficos desenvolvido permite realizar.<sup>1</sup>

## ESTUDO DE CASO

Nesta seção, iremos apresentar alguns gráficos gerados pelo próprio simulador, durante a realização de testes sobre o protocolo de roteamento AODV, para diversas situações. O teste realizado tem o objetivo de avaliar a influência da quantidade de *hosts* para a sobrecarga da rede no pior caso. O pior caso acontece quando *hosts* das extremidades da rede tentam se comunicar, pois, nessa situação, os dados de controle precisam passar por toda a rede para alcançar os destinatários. Para realizar o teste, inicialmente, foram colocados três nós no cenário, como mostra a Figura 7, e enviadas dez mensagens entre os nós das extremidades.

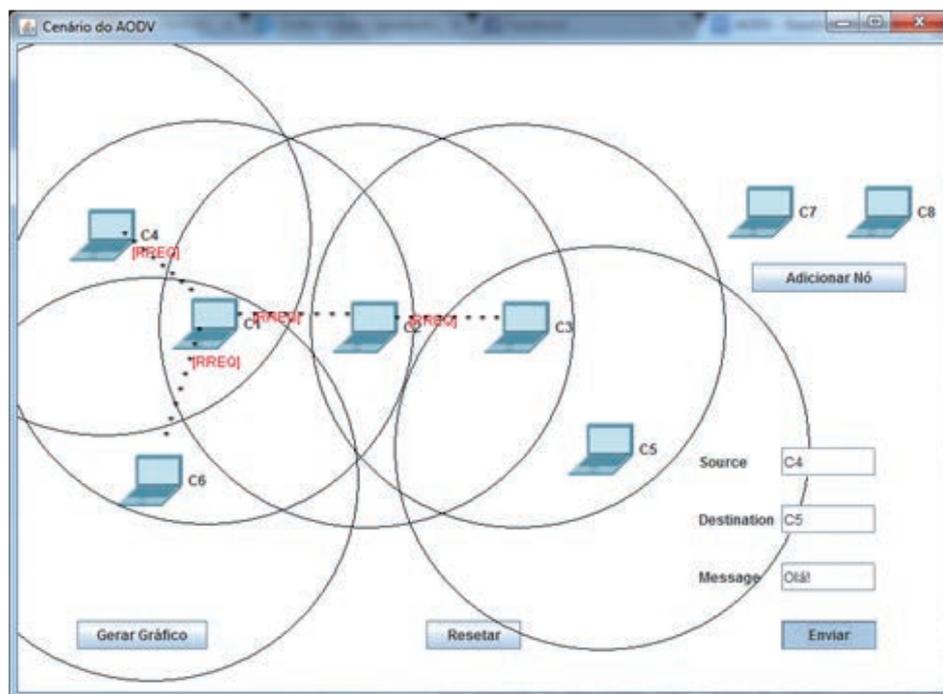
<sup>1</sup> A ferramenta pode ser encontrada para download no endereço: <[www2.uefs.br/netmedia/downloads.html](http://www2.uefs.br/netmedia/downloads.html)>.

Figura 7: Teste realizado com três *hosts* enviando mensagens entre C1 e C3.



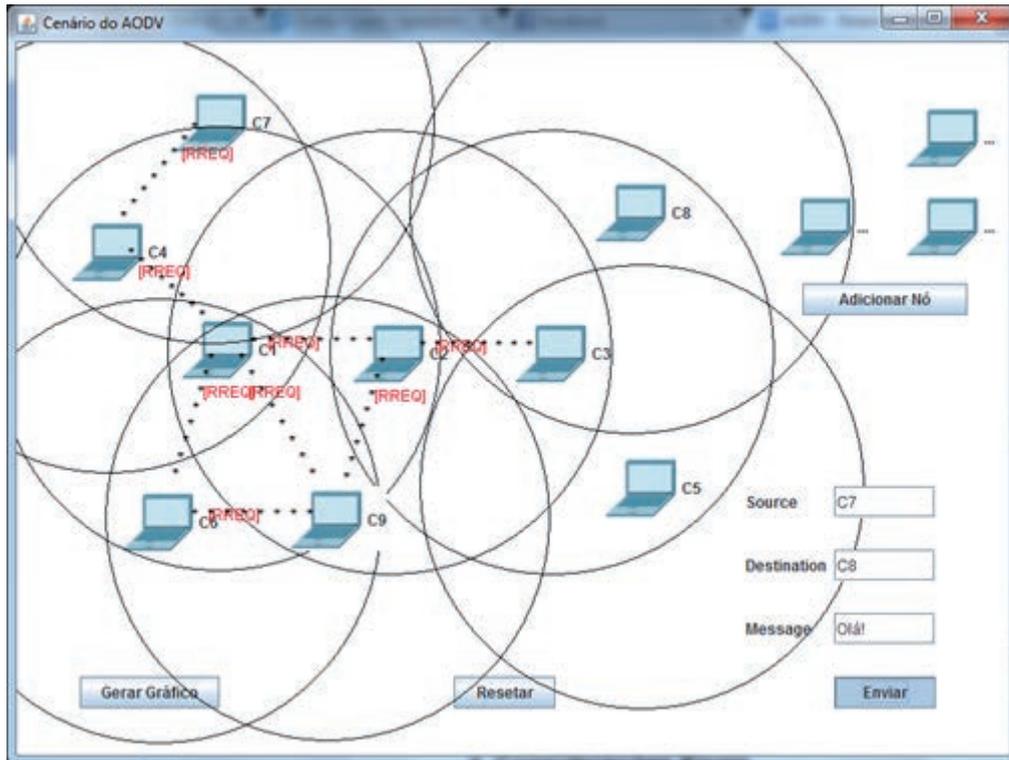
Em seguida, foram adicionados mais três *hosts* ao cenário, conforme mostra a Figura 8, e enviado outros dez pacotes entre os nós das extremidades.

Figura 8: Teste realizado com seis *hosts* enviando mensagens entre C4, C5 e C6.



Para finalizar o teste, foram adicionados mais três *hosts* ao cenário, totalizando nove *hosts*, conforme mostra a Figura 9, e foram enviados outros dez pacotes entre os nós das extremidades.

Figura 9: Teste realizado com 6 *hosts* enviando mensagens entre C7, C8 e C9.



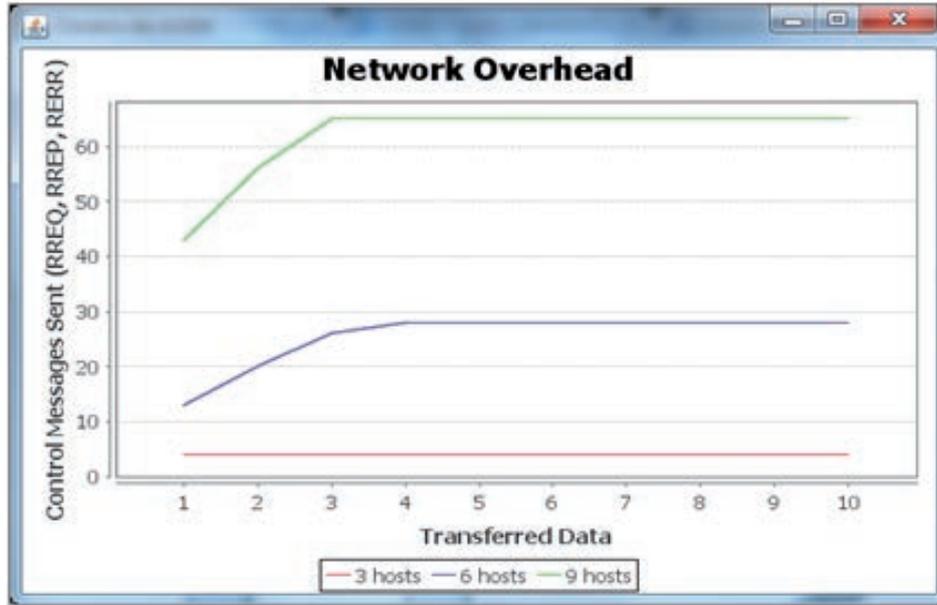
Após o envio das mensagens, o gráfico foi gerado e os resultados obtidos podem ser conferidos na Figura 10. Percebe-se um aumento considerável no envio de mensagens de controle à medida que mais *hosts* são adicionados na rede. Isso mostra que a sobrecarga da rede para enviar dados úteis (mensagens) aumenta diretamente com a quantidade de *hosts* que a compõem.

É importante notar que o aumento da quantidade de mensagens de controle enviadas reflete também no tempo gasto para transmitir as mensagens pela rede. Se mais mensagens de controle precisam ser enviadas para descobrir uma rota até o destino, consequentemente, o tempo gasto para enviar um pacote de dados entre os *hosts* também aumenta.

Porém, após o período de aprendizado das rotas, cada *host* já conhece uma rota para todos os

outros *hosts*, não sendo mais necessário o envio de mais mensagens de controle. A partir desse momento, a sobrecarga provocada pelas mensagens de controle é eliminada e o desempenho da rede melhora no envio de novas mensagens. Isso pode ser notado, na Figura 10, após o envio da quarta mensagem. A figura mostra que, para enviar cinco ou dez pacotes para as diferentes quantidades de *hosts*, a quantidade total de pacotes de controle necessários é a mesma. Em outras palavras, o tempo necessário para enviar o 5º pacote é o mesmo tempo gasto para enviar o 6º, o 7º ou o 10º pacote, pois não há necessidade de enviar mais mensagens de controle, já que os *hosts* de origem já conhecem as rotas para os destinos das mensagens. Nesse caso, novas mensagens de controle apenas são necessárias quando uma rota se torna inativa.

Figura 10: Gráfico comparando a sobrecarga sobre a rede em relação à quantidade de *hosts*.



## CONCLUSÕES

Na elaboração do *software* que simula o funcionamento do AODV, buscou-se focar nos aspectos didáticos para fazer da ferramenta não apenas um simples simulador, mas um facilitador do ensino de uma tecnologia cada vez mais comum no mundo real, como é o protocolo de roteamento AODV. Entre os aspectos didáticos desenvolvidos, podemos citar a interface com o usuário, que permite que qualquer indivíduo possa posicionar os *hosts* em qualquer local, simplesmente arrastando-o para a posição desejada. Além disso, o usuário pode adicionar quantos nós desejar para fazer desde simulações simples até as mais complexas.

Outro fator facilitador do entendimento do protocolo é tornar visível o raio de alcance de cada *host*, para que o usuário possa entender que a transmissão de dados só é possível quando os dispositivos estão dentro do raio de alcance dos demais, seja diretamente ou indiretamente, através de vizinhos. Durante a simulação da transmissão de dados, o *software* também informa ao usuário qual tipo de pacote está sendo transmitido (se são pacotes de controle, e quais são eles, ou de dados), o que facilita ainda mais o entendimento de todos os passos necessários para a construção das rotas e o encaminhamento correto dos dados. A possibilidade de o usuário também analisar a tabela de roteamento

de qualquer *host*, em qualquer instante da simulação, permite sanar quaisquer dúvidas e oferecer uma completa compreensão do funcionamento do AODV.

Por fim, a possibilidade de gerar gráficos de desempenho que possam levar o usuário a tirar conclusões pertinentes acerca do funcionamento do protocolo é de grande importância para completar o entendimento das vantagens e desvantagens do protocolo estudado. Através da geração dos gráficos, o usuário pode analisar a quantidade de requisições necessárias para realizar a comunicação entre diferentes dispositivos, levando em consideração a quantidade de *hosts* entre eles, e perceber, na prática, como a quantidade de nós influencia no desempenho da rede.

Contudo, é importante salientar que não foram desenvolvidos todos os detalhes do protocolo AODV. A implementação desenvolvida é o suficiente para simular uma comunicação sem fio típica entre *hosts* móveis. Apesar de não ser uma implementação fiel do protocolo AODV, o simulador é capaz de mostrar as principais funcionalidades do protocolo e de obter diversos resultados que contribuem para uma análise e um melhor entendimento da essência do protocolo, bem como as suas vantagens e desvantagens. Para projetos futuros, pode-se utilizar o mesmo ambiente já desenvolvido para simular outros protocolos de comunicação sem fio, e,

assim, permitir que o usuário possa testar diferentes protocolos sobre o mesmo cenário e comparar o desempenho entre eles.

## REFERÊNCIAS

- KUROSE, J. F.; ROSS, K. W. **Redes de computadores e a Internet**: uma abordagem top-down. 5. ed. São Paulo: Pearson, 2010.
- CAO, Z.; LU, G. **S-AODV**: sink routing table over AODV routing protocol for 6LoWPAN. Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010 Second International Conference on, p. 340-343, 2010.
- VIEIRA, A. S.; CELESTINO JÚNIOR, S. J. **Lógica Fuzzy para melhorar o desempenho do protocolo AODV em redes ad hoc veiculares**. Anais do InfoBrasil, 2001. Disponível em: <<http://www.infobrasil.inf.br/pagina/anais-2011>>. Acessado em: 13 jun. 2014.
- PERKINS, C. E.; ROYER, E. M. **Ad Hoc On-Demand Distance Vector**. Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on, pp.90, 100, 25-26 Feb 1999.
- PERKINS, C. E.; BELDING-ROYER, E.; DAS, S. **Ad Hoc On-Demand Distance Vector (AODV) Routing**, RFC 3561, July 2003. Disponível em: <<http://www.ietf.org/rfc/rfc3561.txt>>. Acessado em: 3 nov. 2014.
- MURTHY, C. S. R.; MANOJ, B. S. **Ad Hoc wireless networks**: architectures and protocols. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2004.
- TOMAR, G. S.; SHARMA, T.; BHATTACHARYYA, D.; TAI-HOON, K. **Performance comparison of AODV, DSR and DSDV under various network conditions**: a survey. Ubiquitous Computing and Multimedia Applications (UCMA), 2011 International Conference on, p. 3-7, 13-15 April 2011.

---

## DADOS DOS AUTORES



**Karine Souza de Almeida** – Engenheira de Computação Universidade Estadual de Feira de Santana (UEFS, 2013). No período da graduação, desenvolveu projetos de Iniciação Científica direcionados aos sistemas de controle e sistemas digitais. Atualmente, é mestranda em Computação Aplicada, pela UEFS, elaborando projeto na área de telecomunicações.,



**Tassalon Ferreira da Silva** – Engenheiro de Computação pela Universidade Estadual de Feira de Santana (UEFS, 2013). No período da graduação, desenvolveu projetos de pesquisa relacionados a sistemas de controle, automação e desenvolvimento de jogos digitais educativos. Atualmente, é mestrando em Computação Aplicada pela UEFS, desenvolvendo projetos de pesquisa na área de redes de sensores visuais sem fio.



**Daniel Gouveia Costa** – Possui Doutorado em Engenharia Elétrica e de Computação pela Universidade Federal do Rio Grande do Norte (UFRN, 2013), mestrado em Engenharia Elétrica pela Universidade Federal do Rio Grande do Norte (2006), graduação em Engenharia de Computação pela Universidade Federal do Rio Grande do Norte (2005) e graduação em Tecnologia em Informática pelo Centro Federal de Educação Tecnológica do RN (2003). É pesquisador na área de redes de computadores e professor da Universidade Estadual de Feira de Santana (UEFS).