

Estudo Comparativo de Diferentes Protocolos de Roteamento Aplicados a Redes de Sensores Sem Fio

Jonathan J. P. Moura, J. M. R. de Souza Neto, *Member IEEE*, Fabrício B. S. de Carvalho *Senior Member IEEE*

Resumo—As Redes *Ad Hoc* Sem Fio estão cada vez mais rigorosas no que diz respeito à confiabilidade durante trocas de dados. Um fator que está diretamente ligado a isto é o método adotado pelo protocolo de roteamento na rede, que pode influenciar positivamente ou não dependendo da aplicação. Desta maneira, este artigo apresenta os resultados de estudos realizados com três protocolos que utilizam métodos distintos de roteamento de rede com o intuito de avaliar o comportamento dos mesmos observando-os em aplicações de alguns cenários a fim de levantar dados comparativos.

Palavras-Chave—Roteamento, Redes *Ad Hoc* Sem Fio.

Abstract—Wireless *Ad Hoc* Networks are increasingly stringent when it comes to reliability during data exchange. A factor that is directly connected with that is the method assumed by the network routing protocol, which can influence positively or not depending on the application. In this way, this article presents the results of studies carried out with three protocols which use different methods of network routing with the intention of evaluating their behavior by observing them in application of some scenarios, in order to set up comparative data.

Keywords—Routing, Wireless *Ad Hoc* Networks.

I. INTRODUÇÃO

A utilização de dispositivos para troca de informações sem fio apresenta cada vez mais a necessidade de uma comunicação segura e rápida, em que não só a eficiência na troca de dados como a economia de energia dos dispositivos devem ser levadas em consideração [1]. As redes de sensores sem fio (RSSF) podem ser vistas como um tipo especial de redes *ad hoc* móveis (*Mobile Ad Hoc Network - MANET*). Uma rede *ad hoc* não possui uma administração central, ou seja, nesta rede cada um dos dispositivos funciona como um roteador com alcance limitado de transmissão [2]. Por outro lado MANETs caracterizam-se como um sistema de nós sem fio dinamicamente auto-organizado em termos de configuração da rede, dos nós roteadores e das estações remotas (*hosts*) [3]. A depender do tipo de protocolo adotado para um ambiente específico, a qualidade do sistema (QoS) pode ser melhorada, o que é sempre desejado.

Este trabalho teve como objetivo realizar o estudo comparativo, bem como a implementação de três tipos distintos de protocolos de roteamento, de modo que estes protocolos serão

Jonathan Jefferson Pereira Moura, José Maurício Ramos de Souza Neto, Fabrício Braga Soares de Carvalho, Departamento de Engenharia Elétrica, Centro de Energias Alternativas e Renováveis, Universidade Federal da Paraíba - Paraíba, Brazil, E-mail: {jonathan.moura,mauricio,fabricio}@cear.ufpb.br, os autores agradecem ao Departamento de Engenharia Elétrica da UFPB

avaliados em alguns cenários com o intuito de observar qual apresenta um melhor desempenho.

II. REDES DE SENSORES SEM FIO (RSSF)

Ao longo dos anos, a crescente demanda por uma sofisticação do monitoramento e controle de sistemas com múltiplos sensores gerou um grande interesse no desenvolvimento das RSSFs. Estes tipos de redes proporcionam um acesso distribuído à rede para sensores, atuadores e processadores embutidos em uma variedade de equipamentos, instalações e ambientes, o que representa uma melhoria significativa sobre os sensores tradicionais [4].

De forma geral, as RSSFs são compostas por sistemas embarcados (sistemas computacionais de uso dedicado), com capacidade de comunicação e dotados de algum elemento para realizar sensoriamento e atuação, de acordo com a aplicação alvo [5]. Sua implantação é feita através da disposição de uma determinada quantidade de sensores que trabalham em conjunto em um ambiente com a finalidade de realizar a observação de um determinado fenômeno que está ocorrendo ou a detecção de um fenômeno que venha a ocorrer, de modo que essa rede de sensores possa interagir com os acontecimentos que a mesma esteja observando, podendo modificar/modelar o fenômeno observado ou não [1].

III. PROTOCOLOS DE ROTEAMENTO

A principal função da camada de redes é rotear pacotes de uma máquina origem para uma ou mais máquinas destino. Um algoritmo de roteamento é a parte do *software* da camada de redes responsável pela decisão sobre a linha de saída a ser usada na transmissão do pacote de entrada [6].

Sendo assim, às vezes é necessária a utilização de nós que sirvam de roteadores para deslocar um pacote de dados de um nó fonte até um nó destino. Estes nós roteadores, com base nas informações adquiridas pela rede, decidem então o caminho a tomar para entregar dados aos seus respectivos destinos.

Com o intuito de atingir esse objetivo, os roteadores trocam informações entre si, na tentativa de obter um conhecimento parcial ou total da rede, e dessa forma selecionar a melhor rota [7].

Pensando em protocolos que atendessem aos critérios citados acima, decidiu-se então trabalhar com o AODV (*Ad-Hoc On-demand Distance Vector*), AOMDV (*Ad-Hoc On-demand Multipath Distance Vector*) e DSR (*Dynamic Source Routing*).

A. Ad-Hoc On-demand Distance Vector (AODV)

O protocolo de roteamento *Ad-hoc On-demand Distance Vector* é projetado para redes *ad hoc* móveis. O AODV é capaz de efetuar roteamentos tanto em *unicast* quanto em *multicast*, sendo um algoritmo por demanda, o que significa que há construção de rotas entre nós apenas quando solicitado por um nó fonte, o que o classifica como um protocolo reativo. O AODV possui duas etapas distintas: descoberta de rotas e manutenção de rotas.

Quando uma nova rota é requerida, são feitos dois caminhos entre o nó fonte e o nó de destino: uma rota direta (da fonte para o destino) e uma rota reversa (do destino para a fonte). O AODV utiliza vetores de distância no roteamento que indicam a distância entre o nó emissor e os nós que receberam o seu RREQ (*Route Request*). Para estes vetores, são relevantes, no momento de decidir qual rota tomar: a quantidade de saltos e o próximo salto em direção ao destino [8]. A Figura 1 ilustra o procedimento de descoberta entre o nó X (fonte) e o nó Y (destino).

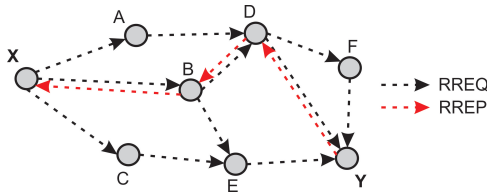


Fig. 1. Descoberta de rota no protocolo AODV.

Inicialmente, o nó X emite um sinal RREQ em *broadcast* para os nós vizinhos que os retransmitem até que o sinal chegue ao nó Y, estabelecendo uma rota direta entre a fonte e o destino com o menor número de saltos utilizado para transmissão de dados. Assim que o RREQ chega ao nó destino, é iniciado o envio de pacotes RREP (*Route Reply*) utilizando os nós que formam a rota direta entre o nó fonte e o nó destino para criar uma rota reversa, utilizada para transmitir sinais de ACK (*Acknowledgement*).

A etapa de manutenção de rotas é mais simples do que a descoberta de rotas, de modo que nesta etapa são realizados dois procedimentos: manutenção e detecção. A manutenção de rotas é realizada através de uma emissão periódica de um sinal HELLO por *broadcast* através dos nós ativos. A detecção de erro em uma rota ocorre quando um sinal ACK deixa de ser emitido por um nó. Quando isto ocorre, o nó que detecta o erro armazena o endereço do nó que gerou o erro e o seu próprio endereço e os envia através de um pacote ERROR para o nó fonte, que por sua vez inicia um novo processo de descoberta de rotas.

B. Ad-Hoc On-demand Multipath Distance Vector (AOMDV)

Protocolos *On-demand Multipath* descobrem múltiplos caminhos entre a origem e o destino em uma simples descoberta de rota. Portanto, a descoberta de uma nova rota é necessária somente quando todos os caminhos existentes falharem [9].

Por terem acesso a várias rotas alternativas, o AOMDV possui uma maior eficiência quando comparado a protocolos que

têm um caminho único entre o nó fonte e o nó destino (*single path*), já que possuem uma menor interrupção no tráfego de dados, reduzindo a sobrecarga do roteamento. Assim como o AODV, o AOMDV [10] é dividido em duas etapas: descoberta e manutenção de rotas e, assim como o AODV, é um protocolo reativo.

O processo de descoberta de rotas é iniciado quando um nó fonte transmite em *broadcast* pacotes RREQs através da rede até identificar o nó destino. Até então isso se assemelha ao AODV, porém no caso do AOMDV os nós roteadores podem reconhecer mais de um RREQ em sua tabela de roteamento sem que seja necessário o descarte dos RREQs atrasados, deste modo armazenando mais de uma rota de transmissão de pacotes de dados.

Assim, ao receber RREQs o nó roteador analisa se suas rotas até a fonte são distintas para então decidir se os dois caminhos são válidos ou não. A Figura 2 ilustra o processo de descoberta de rotas:

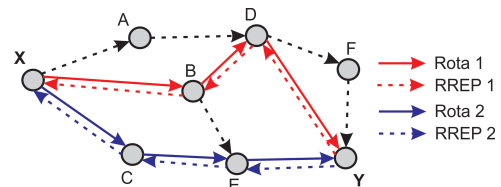


Fig. 2. Descoberta de rotas para o protocolo AOMDV.

Como pode ser visto, o nó X envia vários pacotes de requisição de rotas até o nó Y, obtendo uma rota direta principal ($X \rightarrow B \rightarrow D \rightarrow Y$) e uma rota direta alternativa ($X \rightarrow C \rightarrow E \rightarrow Y$) de modo que a rota alternativa só será utilizada caso ocorra a detecção de algum erro de transmissão na rota principal, sem que seja necessária uma nova busca de rotas. O protocolo só iniciará uma nova busca por rotas caso todas as rotas alternativas na rede apresentem erro de transmissão. A manutenção de rotas é idêntica à descrita para o protocolo AODV.

C. Dynamic Source Routing (DSR)

O *Dynamic Source Routing* é um algoritmo de roteamento para RSSF baseado em um método conhecido como roteamento por fonte. É semelhante ao AODV no que diz respeito a ser um protocolo reativo, gerando um anúncio de rotas apenas quando demandado [11]. Suas etapas de roteamento são descritas a seguir.

O processo de descoberta de rotas inicia-se quando um nó origem analisa se em sua *cache* existe alguma rota salva que forneça uma comunicação entre ele e o nó de destino. Caso não possua uma rota salva o mesmo emite, em *broadcast*, um sinal de RREQ para os nós vizinhos.

Caso algum nó roteador possua informações em *cache* da rota até o destino, ele envia ao nó origem uma mensagem contendo o endereço de todos os nós até o destino e o processo de busca por rotas é então finalizado, dando início à transmissão de dados. Caso contrário, o mesmo salva seu endereço na tabela de roteamento e retransmite o sinal *broadcast* de requerimento de rotas até que seja encontrado um roteador com

informações sobre a rota ao destino ou o próprio nó destino receba o pacote RREQ.

A Figura 3 ilustra exemplos das três possíveis situações que podem vir a ocorrer em uma etapa de descoberta de rotas de um protocolo DSR. O Exemplo 1 ilustra o caso em quem o nó A necessita efetuar uma transmissão até o nó B, porém nenhum dos nós possui dados salvos em suas tabelas de roteamento que ligue uma rota entre o nó A e B.

O caso mostrado no Exemplo 2 mostra um exemplo em que um nó roteador (azul), ao receber um RREQ e analisar sua tabela de roteamento, percebe que possui uma rota válida entre o nó A e o nó B. Feito isto, todo o processo de descoberta de rotas é encerrado e a transmissão é iniciada.

No caso representado no Exemplo 3, o próprio nó A possui em sua *cache* a tabela de roteamento com todos os endereços dos nós que criam uma rota entre ele e o nó destino B. Neste caso, a etapa de descoberta de rotas não efetua nenhuma transmissão em *broadcast* e já inicia a transmissão de dados.

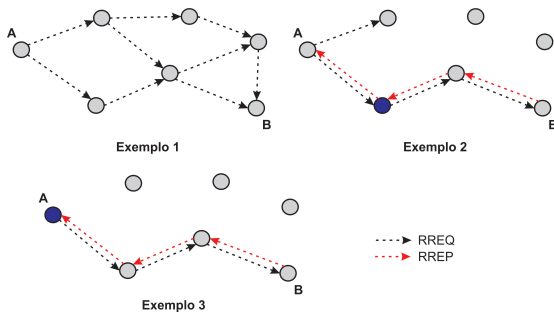


Fig. 3. Descoberta de rotas em um protocolo DSR.

No que diz respeito à manutenção das rotas no DSR, o procedimento é o mesmo descrito para os protocolos anteriormente apresentados.

IV. METODOLOGIA EMPREGADA

A. Network Simulator (NS-2)

Este *software* dá a possibilidade de se realizar simulações de redes com e sem fio, utilização de protocolos de controle de transporte (*Transport Control Protocol* - TCP) ou protocolo de datagrama do usuário (*User Datagram Protocol* - UDP) e muito mais. São utilizados dois tipos de linguagens nas simulações no NS-2: linguagem C++ e linguagem OTCL (*Object-oriented Tool Command Language*), que é uma linguagem orientada a objeto. A linguagem C++ é utilizada no desenvolvimento de blocos básicos aplicados nas simulações (protocolos, agentes, etc). Já a linguagem OTCL é usada no desenvolvimento das redes em si (definição de parâmetros de simulação, disposição dos nós, início e finalização de comunicações, etc). Executando-se uma simulação de um arquivo OTCL através do *prompt* de comando o compilador do NS-2 realiza todo o processamento dos parâmetros de simulação e os executa, simultaneamente armazenando todos os eventos ocorridos em um arquivo *.trace*.

B. Obtenção de Dados

Como o arquivo *trace* gera uma quantidade muito grande de informações que não estão no escopo das métricas de análise

qualitativa, faz-se necessária uma "filtragem" dos dados, é utilizado um código na linguagem AWK para filtrar os arquivos.

Para efeitos de análise de dados é vital que um número mínimo de repetições das simulações sejam feitas para obtenção de uma análise estatística confiável. Portanto, para realizar essa repetição de eventos de forma aleatória são implementados códigos para gerar variáveis aleatórias dentro do próprio código OTCL. Então são feitas simulações no NS-2 seguida pela execução do arquivo *.awk* para observar os resultados das métricas qualitativas.

Foi criando um código utilizando um implementador de comandos *bash*, que permite a execução de sequências de comandos diretamente do *prompt* de comando ou escritos em arquivos de texto, para implementação de um *loop* que execute em sequência o código OTCL (proveniente do NS-2) e o código para avaliação das métricas, no caso o código em *.awk*.

C. Métricas de Análise Qualitativa

Para efetuar o processo de análise qualitativa entre os três diferentes protocolos descritos anteriormente foram adotados três parâmetros: razão de entrega de pacotes (*Packet Delivery Ratio* - PDR), razão de perda de pacotes (*Packet Drop Rate*) e o atraso médio para a chegada dos pacotes de dados.

Packet Delivery Ratio é definida como a razão entre os pacotes de dados recebidos pelo destino em relação aos pacotes que foram enviados pela fonte [12]. Matematicamente, pode ser definida por:

$$PDR = \frac{S_R}{S_T}, \quad (1)$$

em que S_R é dado como a soma dos pacotes recebidos por cada destino e S_T é a soma do pacotes gerados por cada fonte. Já a razão de perda de pacotes é vista como o oposto do *Packet Delivery Ratio*, sendo assim definida como a razão entre a quantidade de pacotes perdidos durante uma transmissão e o número de pacotes enviados pela fonte. É expressa matematicamente pela equação:

$$Packet\ Drop\ Rate = \frac{S_P}{S_T} \quad (2)$$

Similarmente à Equação 1 S_T , continua sendo o número de pacotes enviados pelas fonte e S_P denota o número de pacotes perdidos durante toda a transmissão de dados.

Como último critério de avaliação do protocolo, tem-se o atraso médio para chegada de pacotes, que é simplesmente uma média de todos os atrasos que venham a ocorrer desde a geração de um pacote de dados em um nó fonte até o seu recebimento em um nó destino.

D. Método Estatístico para Análise dos Resultados

As análises estatísticas dos fatores de qualidade são feitas através de uma mescla entre a análise por quartis e distribuição z .

Segundo [13], não é possível obter uma estimativa perfeita da média de uma população a partir de qualquer número finito de um conjunto finito de amostras. Uma alternativa para

se contornar este problema é a definição de limites probabilísticos, onde chega-se a dois valores limites que possuem uma alta probabilidade, $1 - \Delta$, de que a média da população esteja dentro do intervalo.

A determinação dos limites de confiança parte do Teorema do Limite Central, que afirma que se é observado um conjunto de amostras $\{x_1, x_2, \dots, x_n\}$ independentes oriundas de uma mesma população que possui uma média μ e um desvio padrão δ , então a média, definida por (\bar{x}) , adquirida de um número maior de amostragens possui uma distribuição normal que tende a μ e um desvio padrão tendendo a δ/\sqrt{n} [13].

$$\bar{x} \approx N\left(\mu, \frac{\delta}{\sqrt{n}}\right) \quad (3)$$

Usando o Teorema do Limite Central, o intervalo de confiança para a média de uma população é dado pela Equação 4:

$$\left(\mu - z_{1-\alpha/2} \times \frac{\delta}{\sqrt{n}}, \mu + z_{1-\alpha/2} \times \frac{\delta}{\sqrt{n}}\right), \quad (4)$$

em que $z_{1-\alpha/2}$ possui valor igual a 1,96 para $1 - \Delta = 95\%$, de acordo com a Tabela A.2, do Apêndice A, da referência [13]. Assim, a partir de um conjunto de amostras provenientes de um número n de simulações independentes, tem-se um intervalo de confiança em que a amostra está contida em 95% dos casos, quando o conjunto de amostras é expandido [8].

Para que os resultados sejam confiáveis é necessário um número bastante elevado de amostras, porém existe uma forma de se determinar o número mínimo (n_{min}) de simulações que levam a resultados aceitáveis de confiabilidade. Assim, para um $\Delta = 5\%$ o intervalo de confiança é definido por:

$$(\mu(1 - 0,05), \mu(1 + 0,05)) \quad (5)$$

Utilizando as Equações 4 e 5, resolvendo para n chega-se a uma expressão que define o número mínimo de simulações necessárias para se obter uma confiança de 95%, dada por:

$$n_{min} = \left(\frac{1,96 \times \delta}{0,05 \times \mu}\right)^2 \quad (6)$$

Inicialmente são feitas, para cada estudo de caso, um total de 50 simulações, para que por meio destas sejam concebidas médias μ e desvios padrões δ .

A análise em quartis se mostrou essencial para separar alguns resultados muito discrepantes que ocasionalmente viessem a ocorrer durante as simulações dos resultados que realmente são relevantes. Assim, os dados foram separados com a utilização do *software* MATLAB em três matrizes: matriz de razões de entregas de pacotes, matriz de razões de perdas de pacotes e matriz de atrasos. Cada matriz foi então ordenada de forma crescente e os quartis das extremidades foram retirados, de maneira que foram feitos estudos apenas com os quartis centrais.

E. Cenários

Os dados foram simulados em 5 cenários distintos. O cenário 1 ilustra uma situação em que existem dois grupos de nós roteadores separados um do outro e é necessário efetuar uma transmissão de um grupo para o outro, de modo que existem dois caminhos possíveis pra efetuar a comunicação. O cenário 2, assim como o cenário 1, apresenta dois grupos de nós separados, sendo que o diferencial em relação ao cenário 1 se dá no fato de existir apenas um caminho possível para comunicação entre os dois grupos. Enquanto os cenários 1 e 2 lidavam com situações em que existem limitações, representadas por gargalos localizados no centro das redes, o cenário 3 dispõe os nós de uma forma mais ou menos uniforme. Para este caso foram feitas simulações com transmissões entre nós localizados nos extremos da rede.

Por outro lado, os cenários 4 e 5 foram definidos com base em ambientes reais, onde são realizadas simulações utilizando plantas do recém construído bloco do Centro de Energias Alternativas e Renováveis (CEAR) da Universidade Federal da Paraíba (UFPB).

V. RESULTADOS

São apresentados agora os resultados correspondentes a cada um dos cenários implementados.

A. Cenário 1

A partir dos dados da Tabela I, nota-se que o protocolo DSR apresenta, de fato, um excelente desempenho ao se levar em consideração apenas a razão de entrega da pacotes, já que ao se ocorrer um erro de transmissão a reconstrução da rota é mais fácil por existirem poucos caminhos possíveis entre um grupo de nós e o outro grupo, porém a sua utilização mostra-se indesejável quando o seu atraso é comparado com os dos outros protocolos. Para este cenário, o protocolo que se mostra mais balanceado quando são analisadas todas as métricas de qualidade é o protocolo AOMDV.

TABELA I
MÉDIA DAS MÉTRICAS QUALITATIVAS PARA O CENÁRIO 1.

| | PDR (%) | Perdas (%) | Atraso (ms) | n_{min} |
|-------|---------|------------|-------------|-----------|
| AODV | 96,0402 | 1,7469 | 62,7082 | 18 |
| AOMDV | 98,2874 | 0,8288 | 60,3768 | 20 |
| DSR | 99,4132 | 0,1358 | 102,7323 | 25 |

B. Cenário 2

Como se pode observar na Tabela II a partir das médias das métricas de qualidade, o protocolo DSR ainda mostra uma deficiência na velocidade com que entrega os pacotes durante as transmissões de dados, apesar de ainda apresentar altas taxas de PDR (neste cenário existe apenas um caminho entre os dois grupos de nós), algo desejável em um protocolo de roteamento eficiente. Para este cenário o protocolo AOMDV novamente se mostrou o mais eficiente por conciliar um atraso admissível para aplicações de RSSF com uma pequena taxa de perda de pacotes.

TABELA II
MÉDIA DAS MÉTRICAS QUALITATIVAS PARA O CENÁRIO 2.

| | PDR (%) | Perdas (%) | Atraso (ms) | n_{min} |
|-------|---------|------------|-------------|-----------|
| AODV | 96,1277 | 1,9211 | 57,8604 | 6 |
| AOMDV | 98,1972 | 0,8244 | 61,3600 | 45 |
| DSR | 99,3961 | 0,1363 | 105,3615 | 21 |

C. Cenário 3

Como o terceiro cenário apresenta uma dispersão dos nós um pouco maior que os cenários já citados, a probabilidade de se encontrar um nó que possua em sua memória dados da rota desejada durante a fase de descoberta de rotas é maior, o que reduziu os atrasos do protocolo DSR para um valor médio de 67,9610 ms. Este valor é aceitável para muitas aplicações em RSSF visto que essa latência não afeta de forma relevante a rede quanto ao seu funcionamento.

TABELA III
MÉDIA DAS MÉTRICAS QUALITATIVAS PARA O CENÁRIO 3.

| | PDR (%) | Perdas (%) | Atraso (ms) | n_{min} |
|-------|---------|------------|-------------|-----------|
| AODV | 95,5628 | 2,2490 | 46,2218 | 10 |
| AOMDV | 98,2170 | 0,8318 | 52,5582 | 16 |
| DSR | 99,6055 | 0,0974 | 67,6910 | 13 |

D. Cenário 4

Agora apresentam-se os resultados obtidos a partir da análise de casos baseados em cenários reais. Analisando os resultados apresentados, observa-se que o protocolo AOMDV apresentou-se mais estável nos seus resultados, tendo uma variação menor do que as observadas nos demais protocolos. Novamente é notado que o protocolo DSR é ineficiente no seu atraso acumulado na entrega de pacotes, e apesar de possuir os melhores resultados de PDR e razão de perda de pacotes, torna-se ineficiente por apresentar um atraso muito instável (tendo variações que vão de aproximadamente 30 ms a valores da ordem de 119 ms).

TABELA IV
MÉDIA DAS MÉTRICAS QUALITATIVAS PARA O CENÁRIO 4.

| | PDR (%) | Perdas (%) | Atraso (ms) | n_{min} |
|-------|---------|------------|-------------|-----------|
| AODV | 96,5504 | 1,5491 | 52,0440 | 168 |
| AOMDV | 98,3704 | 0,7127 | 57,6548 | 54 |
| DSR | 99,4680 | 0,1025 | 70,0023 | 241 |

E. Cenário 5

Conforme esperado, os protocolos tiveram melhoras significativas nas suas métricas de atraso, mantendo valores aproximados de PDR e razão de perda de pacotes quando comparados ao cenário 4. Para estes testes é visível uma superioridade do protocolo DSR sobre os outros, tendo um PDR elevado e conseqüentemente uma razão de perda de pacotes mínima. Ainda assim, possui um atraso parelho com os obtidos para os protocolos AODV e AOMDV.

TABELA V
MÉDIA DAS MÉTRICAS QUALITATIVAS PARA O CENÁRIO 5.

| | PDR (%) | Perdas (%) | Atraso (ms) | n_{min} |
|-------|---------|------------|-------------|-----------|
| AODV | 97,0299 | 1,3411 | 31,6439 | 1450 |
| AOMDV | 98,9281 | 0,4795 | 30,6935 | 623 |
| DSR | 99,7541 | 0,0502 | 31,1407 | 230 |

VI. CONCLUSÕES

Os resultados observados ao final do trabalho indicam que, como se esperava, o protocolo AODV apresentou uma menor eficiência na entrega de pacotes de dados quando comparado aos demais protocolos por possuir o método de descoberta de rotas mais simplificado, mas ainda assim mostrou uma boa qualidade em relação ao atraso na entrega de seus pacotes, possuindo uma média de atraso de cerca de 50,0957 ms considerando todos os cenários estudados. O DSR demonstrou a capacidade de entregar seus dados em segurança e com o mínimo de perdas possível ao seu destino, sendo sempre o protocolo com menor razão de perda de dados, com uma média de 0,1044 % para os 5 cenários. Porém, essa eficiência vem acompanhada de um atraso em geral muito instável em certos cenários estudados. Por fim, o algoritmo AOMDV se mostrou o mais estável, com resultados satisfatórios em todos os testes apesar de ser menos eficiente, porém ainda assim aceitável, do que o DSR em alguns cenários.

REFERÊNCIAS

- [1] J. J. P. Moura, *Estudo Comparativo de Diferentes Protocolos de Roteamento Aplicados a Redes de Sensores Sem Fio*. Trabalho de Conclusão de Curso - Universidade Federal da Paraíba, 2016.
- [2] A. J. G. Bessa, *Uma Extensão ao Protocolo OLSR para Roteamento de Dados em Tempo Real Utilizando Múltiplos Caminhos*. Trabalho de Conclusão de Curso - Universidade Estadual do Ceará, 2010.
- [3] N. P. Mahlink, *Sensor Networks and Configuration*. [S.l.]: Springer, 2007.
- [4] L. B. Ruiz, *MANÁ: Uma Arquitetura para Gerenciamento de Redes de Sensores Sem Fio*. Universidade Federal de Minas Gerais, Dezembro 2003.
- [5] I. Muller, *Redes de Sensores Sem Fio*. X Escola de Microeletrônica do Nordeste, Outubro 2015.
- [6] S. A. Tanenbaum, *Computer Networks*. Prentice, 1996.
- [7] G. V. Brignori, *Estudo de Protocolos de Roteamento em Redes Ad Hoc*. Universidade Federal de Santa Catarina, Julho 2005.
- [8] C. W. B. Araujo, *Protocolo de Roteamento Multi-Métrico para Redes de Sensores Sem Fio - CA-AODV*. Universidade Federal de Campina Grande, Novembro 2009.
- [9] F. L. Figueiredo, M. C. de Castro e M. A. de Siqueira, *Análise de desempenho de protocolos de roteamento para redes Ad Hoc sem fio*. Cad. CPqD Tecnologia, v. 2, pp. 61-70, Julho/Dezembro 2006.
- [10] M. K. Marina e R. S. Das, *Ad Hoc On-Demand Multipath Distance Vector routing*. Wireless Communications and Mobile Computing, v. 6, pp. 969-988, Dezembro 2006.
- [11] S. A. Ade e P. A. Tijare, *Performance Comparison of AODV, DSDV, OLSR and DSR Routing Protocols in Mobile Ad Hoc Networks*. International Journal of Information Technology and Knowledge Management, v. 2, pp. 545-548, Dezembro 2010.
- [12] P. Rohal, R. Dahiya e P. Dahiya, *Study and Analysis of Throughput, Delay and Packet Delivery Ratio in MANET for Topology Based Routing Protocols (AODV, DSR and DSDV)*. International Journal for Advance Research in Engineering and Technology, v. 1, n. 2, pp. 54-58, Março 2013.
- [13] R. Jain, *Art of Computer Systems Performance Analysis Techniques For Experimental Design Measurements Simulation And Modeling*. Wiley Computer Publishing, Jhon Wiley and Sons, Inc, Janeiro 1991.