JONATHAN JEFFERSON PEREIRA MOURA

ESTUDO COMPARATIVO DE PROTOCOLOS DE ROTEAMENTO APLICADOS AS REDES DE SENSORES SEM FIO

João Pessoa

2016

JONATHAN JEFFERSON PEREIRA MOURA

ESTUDO COMPARATIVO DE PROTOCOLOS DE ROTEAMENTO APLICADOS AS REDES DE SENSORES SEM FIO

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Elétrica da UFPB como requisito parcial para obtenção do título de Engenheiro Eletricista

UNIVERSIDADE FEDERAL DA PARAÍBA - UFPB

Orientador: José Maurício Ramos de Souza Neto

João Pessoa

2016

JONATHAN JEFFERSON PEREIRA MOURA

ESTUDO COMPARATIVO DE PROTOCOLOS DE ROTEAMENTO APLICADOS AS REDES DE SENSORES SEM FIO

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Elétrica da UFPB como requisito parcial para obtenção do título de Engenheiro Eletricista

Orientador: Prof. José Maurício Ramos de Souza Neto

Banca Examinadora

Prof. José Maurício Ramos de Souza Neto
Orientador

Prof. Fabrício Braga Soares de Carvalho
Prof. Convidado 1

Prof. Waslon Terllizie Araújo Lopes
Prof. Convidado 2

João Pessoa
2016

RESUMO

Nesta monografia são realizadas implementações dos protocolos de roteamento AODV (*Ad-Hoc On-demand Distance Vector*), AOMDV (*Ad-Hoc On-demand Multipath Distance Vector*) e DSR (*Dynamic Source Routing*) utilizados em redes *Ad Hoc* sem fio com o intuito de avaliar o comportamento dos mesmos observando-os em aplicações de alguns cenários a fim de levantar dados comparativos. Utilizando PDR (*Packet Delivery Ratio*), o atraso médio para chegada dos pacotes de dados e a razão de perda de pacotes como métricas de avaliação foi possível realizar simulações e determinar qual protocolo possui um melhor desempenho em cada cenário implementado. Esses dados são de relevante importância visto que um uso inadequado de protocolos de roteamento pode acarretar em perdas de pacotes em uma rede sem fio, levando a eventuais sobrecargas ou à utilização de um protocolo desnecessário para uma aplicação em particular entre outros problemas que poderiam ser evitados com o uso correto do mecanismo de roteamento utilizado.

Palavras-chaves: Roteamento. Redes *ad hoc*. Padrão IEEE 802.11. Redes de Sensores Sem Fio.

ABSTRACT

In this monograph are performed implementations of the routing protocols AODV (Ad-Hoc On-demand Distance Vector), AOMDV (*Ad-Hoc On-demand Multipath Distance Vector*) and DSR (*Dynamic Source Routing*) used in Ad Hoc wireless networks with the intention to evaluate their behavior by observing them in applications of some scenarios in order to set up comparative data. Using PDR (*Packet Delivery Ratio*), the average delay for arrival of data packets and the rate of packet loss as evaluation metrics was possible perform simulations and to determine which protocol has a better performs in each scenario implemented. These data are great importance because an improper use of routing protocols can result in packet loss in a wireless network, leading to work with any overloads or to the use of unnecessary protocol for a particular application and other problems that could be avoided with the correct use of the routing mechanism used.

Keywords: Routing. *Ad hoc* networks. IEEE 802.11 standard. Wireless Sensor Networks.

AGRADECIMENTOS

Aos meus pais José Moura e Marluce por sempre me guiarem pelo melhor caminho, nunca duvidarem da minha capacidade de ir sempre mais longe e me apoiarem incondicionalmente.

À minha namorada Amanda por estar ao meu lado e sempre acreditar em mim.

Aos meus amigos Kaique, Higo, João, Anderson, Ramilo, Gabriel, Gilvan, George dentre outros que de alguma forma me ajudaram durante o curso.

Ao professor José Maurício por me guiar no decorrer deste trabalho.

Ao professor Antônio Augusto pelos seus ensinamentos que um dia serão de grande valia na minha vida profissional.

Ao meu tio Valdecir Pereira e ao seu filho Vinícius por me acolherem em momentos de dificuldade.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A utilização de dispositivos para troca de informações sem fio apresenta cada vez mais a necessidade de uma comunicação mais segura e rápida, em que não só a eficiência na troca de dados como a economia de energia dos dispositivos devem ser levados em consideração. 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 [1]. Por outro lado redes MANET's 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*) [2].

Ao realizar um projeto de implementação de uma RSSF o projetista deve ter em mente que esses tipos de redes são normalmente desenvolvidas para aplicações em que é necessário um consumo de energia muito baixo, uma vez que os nós sensores que farão parte da rede eventualmente poderão ser dispostos em localizações de difícil acesso ou até mesmo onde não é possível realizar a manutenção desses equipamentos. Por considerar estes fatores o projeto de uma RSSF se torna muitas vezes um trabalho que requer uma atenção considerável.

Devido a estes fatores, levando em consideração que os nós sensores terão uma fonte energética limitada, que é o projeto de uma RSSF se torna muitas vezes um trabalho não trivial.

Por outro lado ao considerar a área de atuação da RSSF em muitos casos é necessário a implementação de mais de dois nós sensores, de modo que os nós adicionais servem como roteadores que recebem uma mensagem e as retransmitem adiante até que a mesma chegue ao seu nó de destino. Isso é necessário pelo fato da potência necessária para transmissão de um sinal ser inversamente proporcional ao quadrado do do raio de alcance (R^2). A Equação 1.1 descreve a relação entre potência de transmissão e o seu respectivo raio de alcance.

$$P_R \propto \frac{P_T}{R^2} \tag{1.1}$$

Em que:

- P_R corresponde à potência recebida;
- P_T corresponde à potência transmitida;
- R indica o raio de alcance do sinal.

Tendo em vista essa relação, múltiplas mensagens em curtas distâncias requerem menor potência que uma única mensagem a longa distância [3].

Como foi dito, para se economizar energia nas RSSF é preciso a utilização de nós roteadores, mas além disso é necessário levar em consideração o tipo de protocolo que deverá ser implementado para realizar o processo de roteamento da rede. A depender do tipo de protocolo adotado para um ambiente específico a qualidade do sistema (QoS) pode ser melhorada, o que é sempre desejado.

1.1 Motivação

Por observar uma discrepância muito relevante na eficiência, seja ela energética ou com respeito a QoS de uma RSSF, a depender do uso adequado do protocolo de roteamento implementado em uma aplicação, se vê necessário um estudo mais aprofundado de como cada protocolo de roteamento se comportar em cenários específicos para que a partir destes dados seja possível determinar qual protocolo se encaixa melhor, a depender dos requisitos essenciais da rede, para uma aplicação. Desta forma os item citados acima são tomados como base motivacional para realização deste trabalho.

1.2 Objetivos

Este trabalho tem como objetivo realizar o estudo comparativo bem como a implementação de três tipos distintos de protocolos de roteamento, AODV (*Ad-Hoc On-demand Distance Vector*), AOMDV (*Ad-Hoc On-demand Multipath Distance Vector*) e DSR (*Dynamic Source Routing*), de modo que estes protocolos serão avaliados em alguns cenários teóricos e reais com o intuito de observar qual protocolo apresenta um melhor desempenho levando em consideração métricas de comparação pré-estabelecidas.

1.3 Estrutura do Trabalho

Este Trabalho de Conclusão de Curso possui um desenvolvimento dividido em quatro capítulos: Capítulo I - Desenvolvimento Teórico, Capítulo II - Materiais e Métodos, Capítulo III - Resultados e Capítulo IV - Conclusões. O primeiro capítulo diz respeito à abordagem acerca da teoria sobre RSSF, protocolos de roteamento e o padrão de comunicação sem fio IEEE 802.11, sua arquitetura e suas características de funcionamento.

Ainda no primeiro capítulo do trabalho são introduzidos nas Seções 2.4, 2.5 e 2.6 os protocolos de roteamento utilizados, onde seus respectivos funcionamentos e características são abordadas mais a fundo. Em seguida temos o segundo capítulo, iniciado na Seção 3.1, onde são mostradas todas a ferramentas necessárias para implementação do trabalho,

nas Seções 3.2, 3.3 e 3.4 são descritas as análises para obtenção dos dados desejados, as métricas utilizadas para à analise qualitativa e todo o procedimento estatístico para validação dos resultados das simulações, respectivamente. Ao final do segundo capítulo são mostrados os cenários utilizados para estudos de caso na Seção 3.5.

O terceiro capítulo inicia-se com a apresentação dos resultados das simulações para todos os protocolos nos cenários 1, 2 e 3 descritos nas Seções 4.1, 4.2 e 4.3, respectivamente. Por fim o terceiro capítulo do trabalho é finalizada nas Seções 4.4 e 4.5, em que são apresentados os resultados da implementação computacional dos protocolos baseados em cenários reais.

O trabalho é finalizado com a apresentação das conclusões tomadas a partir dos dados obtidos por meio das simulações, seguido por citações dos trabalho futuros.

2 DESENVOLVIMENTO TEÓRICO

2.1 Redes de Sensores Sem Fio (RSSF's)

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 RSSF's. 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 RSSF's 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.

As aplicações para esses tipos de redes são inúmeras e vão desde um simples monitoramento da temperatura para prédios domiciliares até a utilização para fins militares, em que o reconhecimento de uma área pode ser feito através da disposição de vários nós sensores, prezando a integridade dos soldados ou fins ambientais, tais como detecção do alastramento de um incêndio em uma floresta. A Figura 1 mostra um exemplo de uma RSSF.

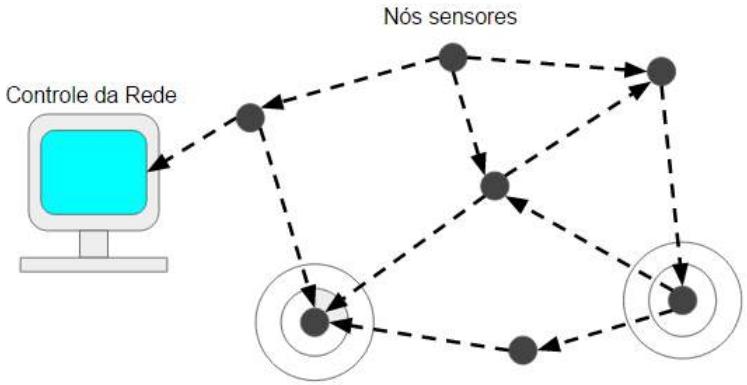

Figura 1 – Exemplo de uma RSSF típica.

Alguns parâmetros devem ser considerados ao se implantar uma RSSF, onde o

método como o fenômeno estudado e os nós sensores irão se dispor, os tipos de dados que serão utilizados nas comunicações e acima de tudo a vida útil dos sensores afetarão na escolha do tipo de protocolo ou sensor a ser utilizado. A seguir é mostrado um pouco sobre alguns parâmetros que devem ser levados em consideração no projeto de uma RSSF [3].

- **Dinâmica da rede:** ao se falar desta especificação os nós podem ser do tipo: dinâmico ou estático, que dizem respeito à movimentação dos nós da rede. Uma aplicação dinâmica é utilizada quando se necessita um monitoramento de um fenômeno móvel, como deslocamento de um veículo de transporte de valores. Por sua vez, uma rede estática, implica-se em uma rede que não necessite de um deslocamento por parte dos nós sensores, como exemplo é possível citar o caso da detecção e propagação de um incêndio florestal;

- **Arranjo dos nós:** o arranjo como os nós se encontram determina se os nós sensores da RSSF estarão dispostos de maneira aleatória ou de maneira previamente determinada. Isso influenciará diretamente na eficiência energética da rede, pois se os nós por acaso estiverem dispostos de forma a gerar um gargalo, isso acarretará em uma maior carga sobre alguns nós, o que afetará de maneira considerável a sua vida útil visto que seu esforço será maior;

- **Modelo de envio de dados:** o modelo de envio pode ser contínuo, dirigido a eventos, por solicitação ou híbrido. Ele diz repeito à estratégia de envio de dados, uma vez que o fenômeno monitorado ou um conjunto desses seja computado, independente de como seja a dinâmica do próprio fenômeno;

- **Capacidade dos nós:** diz respeito à capacidade de funcionamento dos nós de uma RSSF, visto que um nó pode ser utilizado de diferentes maneiras onde cada uma delas irá refletir em um consumo energético específico. Por exemplo um nó que atuar apenas como roteador terá um vida útil maior do que um nó que além de rotear ainda efetua um sensoriamento da temperatura ambiente.

2.2 O Padrão IEEE 802.11

O padrão IEEE 802.11 é utilizado para definir as especificações do controle de acesso ao meio, que diz respeito a camada MAC (*Medium Access Control*), além das especificações da camada física de uma rede *Wireless* LAN [6]. A Figura 2 ilustra o modelo da pilha do padrão IEEE 802.11 onde as camadas superiores, que correspondem à aplicação, acessam a camada MAC diretamente ou de maneira indireta por meio das subcamadas 802.2 LLC (*Logical Link Control*) e SSCS (*Service Specific Convergence Sublayer*), que correspondem as camadas de controle lógico de enlace e o serviço de subcamada de convergência específica, respectivamente.

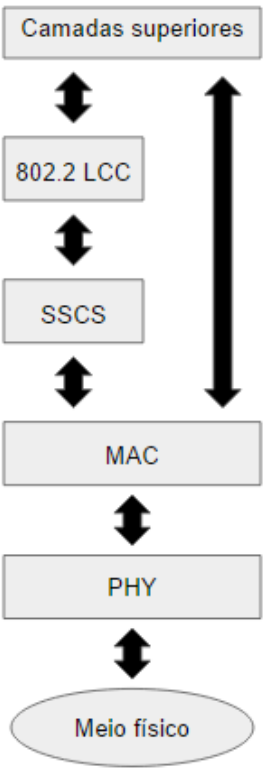

Figura 2 – Diagrama de pilha para padrão IEEE 802.11.

2.2.1 Camada Física do 802.11

Na camada física, o protocolo 802.11 define uma série de padrões de transmissão e codificação para comunicações sem fio, sendo os mais comuns: FHSS (*Frequency Hopping Spread Spectrum*), DSSS (*Direct Sequence Spread Spectrum*) e OFDM (*Orthogonal Frequency Division Multiplexing*). As principais funções da camada física são listadas a seguir e os padrões de rede sem fio para essa camada são ilustrados na Tabela 1 [7].

- Codificação e decodificação de sinais;
- Geração/remoção de parâmetros (cabeçalho) para sincronização;
- Recepção e transmissão de bits;
- Incluir especificações do meio de transmissão em que a rede de sensores será utilizada de modo que essas informações podem auxiliar em uma melhor comunicação entre transmissores e receptores.

2.2.2 Camada de Enlace do 802.11

A camada de enlace de dados para o padrão 802 é dividida em duas subcamadas: o LLC (*Logical Link Control*) e o MAC (*Medium Access Control*). A camada MAC é

802.11	A taxa de transmissão original deste padrão era de 2 Mbit/s com a utilização do FHSS e 2.4 GHz como frequência de operação. Entretanto, sob condições não ideais uma taxa de 1 Mbit/s era utilizada para transmissão.
802.11b	A criação do IEEE 802.11b possibilitou uma padronização de uma camada física que fosse capaz de efetuar transmissões com uma taxa mais elevada. Sendo capaz de transmitir à taxas de 5.5 e 11 Mbit/s usando a mesma frequência de operação este padrão utiliza a DSSS para conseguir esse alto desempenho nas taxas de transmissão. Apesar de altas taxas de transmissão os valores utilizados em aplicações não ideais são 5.5 Mbit/s, 2 Mbit/s ou 1 Mbit/s.
802.11a	Apesar de ser o primeiro a ser padronizado o 802.11a só se tornou mais utilizado a pouco tempo, operando com uma taxa de 54 Mbit/s e com uma frequência de operação de 5 GHz. Esse padrão utiliza-se do OFDM permitindo transmissão à grandes taxas enviando dados por sub-frequências. Essa tecnologia habilita a rede a transmitir dados de vídeo e áudio e como não opera em frequências utilizadas no 802.11b não possui problemas com interferências de outros equipamento que a usem. Outras velocidades também pode ser obtidas (48, 36, 24, 18, 12 e 6 Mbit/s).
802.11g	Este padrão trabalha em uma taxa de 54 Mbit/s com frequência de operação de 2.4 GHz e modulação OFDM. O 802.11g possui compatibilidade com o padrão 802.11b podendo operar nas taxas de transmissão que o 802.11b trabalha utilizando a modulação DSSS. O 802.11g fornece uma opção de migração para redes 802.11b por apresentar a mesma faixa de frequência de operação com uma taxa de transmissão mais elevada.

Tabela 1 – Padrões de rede sem fio 802.11.

responsável por reunir dados dentro de um pacote com endereços e campos detecção de erro, abrir pacotes, executar reconhecimento de endereços além de detectar erros e controlar o acesso ao meio de transmissão LAN.

A subcamada LLC é voltada para gerar uma comunicação (interface) com as camadas superiores assim como executar o controle de fluxo e erro de pacotes. A Figura 3 ilustra uma mensagem normalmente utilizada no 802.11, onde sua composição é feita por um cabeçalho, o corpo da mensagem e o campo FCS (*Frame Check Sequence*), responsável por determinar se houve erro durante a transmissão. A numeração acima de cada quadro representa o número de *bytes* utilizados em cada campo. A seguir são feitas as descrições de cada um deles.

O Campo de Controle do Quadro (*Frame Control Field*) é responsável por determinar qual o tipo de Quadro MAC 802.11 é utilizado na transmissão. Dentre os elementos que compõem o FCF estão o de versão do protocolo que indica o protocolo utilizado na

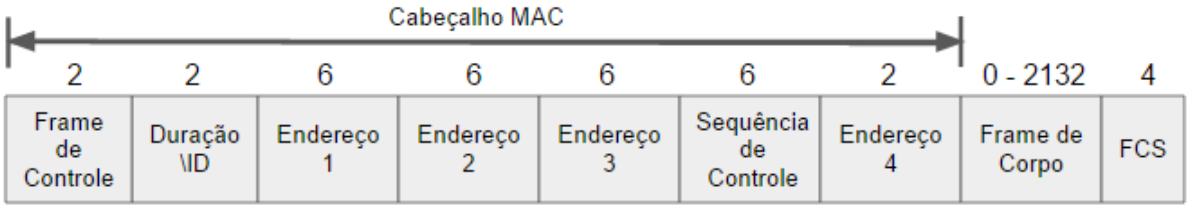

Figura 3 – Mensagem MAC de uma rede 802.11.

rede e o campo *Retry* que indica se a mensagem (informação) está sendo retransmitida ou não.

A mensagem MAC possui também um campo responsável por determinar o tempo restante necessário para receber a próxima transmissão, chamado de Duração/ID Campo (*Duration/ID Field*).

Um campo que possui bastante relevância é o de endereço (*Address Field*), que a depender da transmissão na RSSF pode conter combinações dos tipos de endereço descritos abaixo.

- Identificador BSS: o BSSID (*Basic Service Set Identifier*) serve unicamente para identificar cada BSS, em que uma BSS nada mais é do que um simples *Access Point* (AP) que suporta um ou mais clientes sem fio. Essa rede é também conhecida como *Infrastructure Wireless Network* (Rede Infra-estrutura) [8].
- Endereço Destino: o DA (*Destination Address*) determina qual o endereço MAC do nó destino para uma dada transmissão.
- Endereço da Fonte: determina o endereço do nó do qual partiu inicialmente a transmissão de uma mensagem.
- Endereço do Receptor: indica o endereço do próximo salto até o nó destino.
- Endereço do Transmissor: indica o endereço do nó que transmitiu uma mensagem na RSSF.

Logo após os campos de endereço está o campo de Controle de Sequência (*Sequence Control*) que é subdivido em dois campos: o Número de Fragmento e o Número de Sequência. O Número de Fragmento indica o número de sequência de cada quadro. Esse número é sempre o mesmo para cada quadro enviado para o caso de um quadro fragmentado. Para o próximo quadro não fragmentado, o número é incrementado até atingir 4095 e então retorna para o valor zero novamente. O segundo subcampo do *Sequence Control* é

responsável por indicar o número para cada fragmento do quadro enviado, de modo que o valor inicial é zero e é incrementado para cada fragmento [8].

Os dois últimos campos da mensagem são o Corpo do Quadro (*Frame Body*), em que é possível encontrar as informações de dados de uma transmissão ou de um gerenciamento de transmissão, e a Sequência de Verificação do Quadro (*Frame Check Sequence*) em que o transmissor gera um número específico que corresponde a uma mensagem enviada e o deposita no campo FCS. Ao chegar no receptor o mesmo reconstrói o FCS a partir dos dados recebido e o compara com a sequência enviada na transmissão a fim de detectar algum erro na comunicação.

2.3 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 [9].

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 [10]. As principais características desejadas de um algoritmo de roteamento são listadas a seguir [9]:

- Simplicidade: por se tratar de RSSF, um protocolo de roteamento deve realizar tudo o que se propõe a fazer com uma quantidade mínima de processamento;
- Adaptabilidade: como esses tipos de protocolos são geralmente utilizados em redes móveis, é desejável que o algoritmo seja capaz de se adaptar a variações na rede;
- Robustez: o algoritmo deve gerar respostas aceitáveis e funcionar corretamente com o mínimo de manutenção e falhas possível.

Pensando em protocolos que atendessem aos critérios citados acima foi decidido então trabalhar com o AODV (*Ad-Hoc On-demand Distance Vector*), AOMDV (*Ad-Hoc On-demand Multipath Distance Vector*) e DSR (*Dynamic Source Routing*). Optou-se por esses três protocolos por apresentarem métodos diferentes de realizar suas descobertas de rotas, assim sendo possível determinar qual modo é mais adequado e apresenta maior

eficiência quando aplicados aos cenários propostos neste trabalho. As próximas seções apresentam de forma mais detalhada cada um do protocolos analisados.

2.4 *Ad-Hoc On-demand Distance Vector* (AODV)

O algoritmo de roteamento *Ad-hoc On-demand Distance Vector* é um protocolo de roteamento 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. Adicionalmente, este protocolo forma árvores o qual conectam os membros do grupo *multicast*, essas árvores por sua vez são compostas por membros do grupo *multicast* e os nós necessários para interliga-los, chamados de nós roteadores. Um outro artifício que o protocolo AODV possui é a utilização de sequência de números para garantir o quão recente uma rota é, sendo assim imune a *loops* [11].

Este protocolo possui duas etapas distintas: descoberta de rotas e manutenção de rotas. Para tais etapas são utilizados os quatro tipos de pacotes de dados descritos a seguir:

- RREQ (*Route Request*): o requerimento de rota é um sinal emitido em *broadcast* pelo nó fonte sempre que necessita-se de uma nova rota;
- RREP (*Route Reply*): esta mensagem é simplesmente uma mensagem em resposta, de forma *unicast*, a um RREQ;
- HELLO: um pacote deste tipo é emitido periodicamente com o intuito de detectar erros na rota;
- ERROR: pacote que indica ao nó fonte um erro de comunicação em uma rota.

Cada etapa do protocolo possui seus pacotes específicos, que em conjunto fazem com que as comunicações entre um nó fonte e um nó destino sejam em sua maioria isentas de erros e se comportem de forma dinâmica com relação às modificações na rede.

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, esses vetores indicam a distância entre o nó emissor e os nós que receberam o seu RREQ. Para estes vetores são relevantes no momento de decidir qual rota tomar: a quantidade de saltos e o próximo salto em direção ao destino [3].

2.4.1 Descoberta de Rotas

A Figura 4 ilustra o procedimento de descoberta de uma rota, em que o nó fonte X necessita de uma rota para transmissão de dados para o nó Y de modo que ainda não existe uma rota pré-definida entre eles.

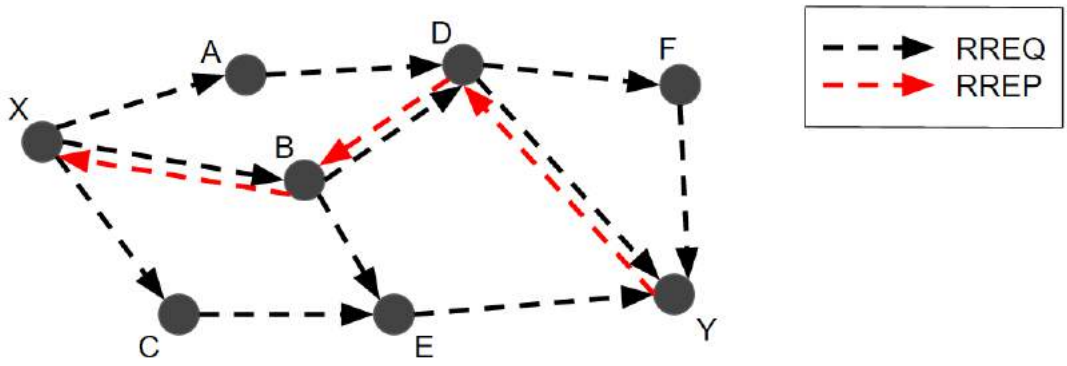

Figura 4 – Descoberta de rota para o protocolo AODV.

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 utilizada para transmissão de dados. Assim que o RREQ chega ao nó destino, é iniciado o envio de pacotes RREP 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*). O fluxograma mostrado na Figura 5 é realizado pelos nós roteadores na fase de descobertas:

Caso possua dados da rota desejada, o nó roteador irá simplesmente enviar o RREP para o nó fonte, caso contrário o mesmo adicionará dados da rota reversa na sua tabela de roteamento (endereço IP do nó destino, endereço IP do nó fonte, identificador do pacote RREQ, tempo de expiração da rota e número de sequência do nó fonte). O identificador do pacote RREQ (BID) é um dado de suma importância nesta etapa, uma vez que o mesmo armazena dados IP do nó fonte e dos nós que retransmitiram um RREQ na rede. Caso um nó receba um conjunto BID que já esteja salvo em sua tabela de roteamento, ele descartará esse requerimento de rota, impedindo assim a formação de *loops*.

Outro artifício utilizado pelo protocolo AODV durante a descoberta de rotas é o tempo de expiração de uma rota, que define o quão recente é uma rota, de modo que esse parâmetro é utilizado para apagar rotas em desuso.

2.4.2 Manutenção de Rotas

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

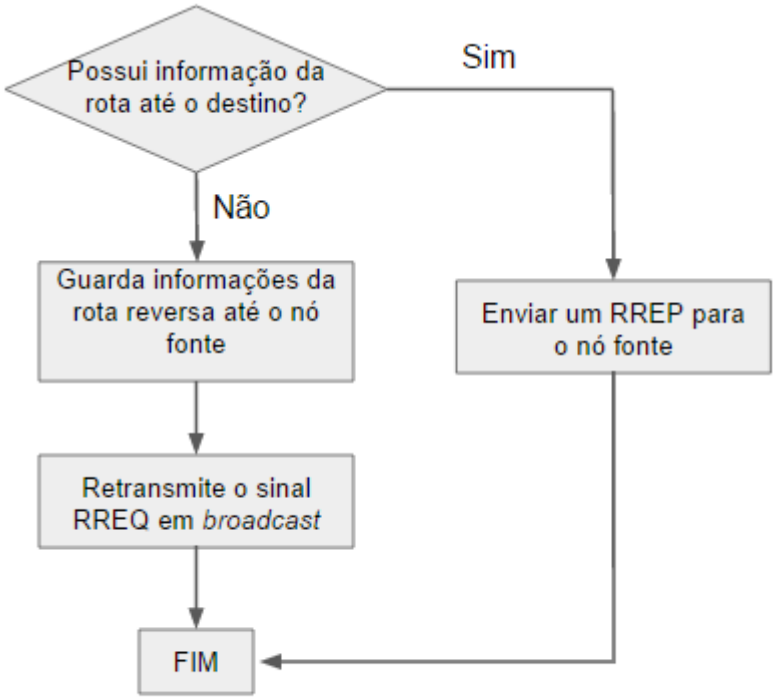

Figura 5 – Fluxograma executado pelos nós roteadores na fase de descoberta de rotas.

manutenção de rotas é realizada através de uma emissão periódica de um sinal HELLO por *broadcast* através dos nós ativos, em que um nó é dito ativo quando envia ou recebe dados antes que o tempo de expiração de rota seja atingido, ou seja, só ocorre monitoramento de rotas que estejam ativas.

A detecção de erro em uma rota ocorre quando um sinal ACK deixa de ser emitido por um nó. Quando um evento como este 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.

2.5 *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 [12].

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 [13] é divido em duas etapas: descoberta e manutenção de rotas e assim como o AODV é um protocolo reativo.

2.5.1 Descoberta de Rotas

O processo de descoberta de rotas é iniciado quando um nó fonte transmite em *broadcast* pacotes RREQ's 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 RREQ's atrasados, deste modo armazenando mais de uma rota de transmissão de pacotes de dados.

Assim ao receber RREQ's 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 6 ilustra o processo de descoberta de rotas:

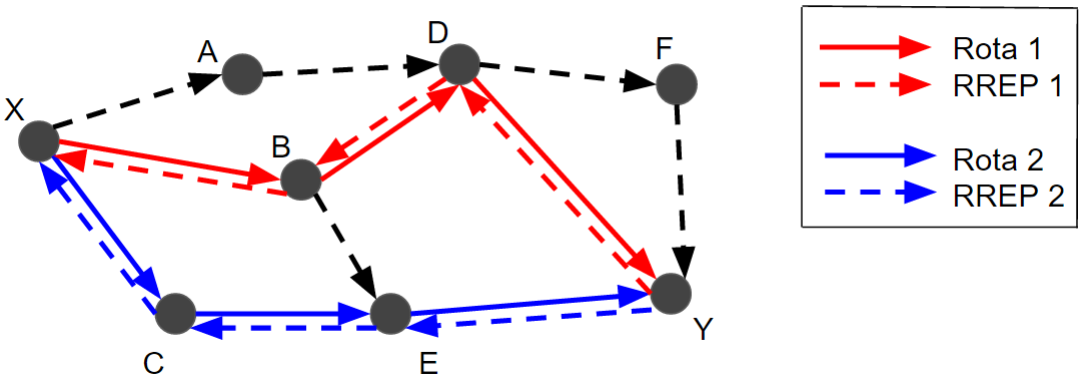

Figura 6 – Descoberta de rotas para o protocolo AOMDV.

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.

2.5.2 Manutenção de Rotas

A manutenção de rotas deste protocolo utiliza-se do envio periódico de sinais HELLO transmitido da fonte ao destino e aguardando se há a ocorrência de uma falha de resposta por parte de algum nó da rota principal. Caso um erro seja detectado, um pacote ERROR é enviado ao nó fonte informando o nó defeituoso e aquela rota é desativada, o que faz com que o protocolo de roteamento automaticamente mude a transmissão de dados, que antes estavam sendo feitas pela rota principal para uma rota alternativa. Este processo se repete até que todas as rotas alternativas apresentem algum erro, o que faz com o que o algoritmo de roteamento inicie uma nova etapa de descoberta de rotas.

No AOMDV, condições necessárias para a manutenção de múltiplos caminhos são empregadas, como a regra do anúncio de rotas, em que nunca se anuncia uma rota mais curta que uma já anunciada; e a regra de aceitação de rotas, em que nunca se aceita uma rota mais longa que uma já anunciada [12].

2.6 *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 nas subseções a seguir.

2.6.1 Descoberta de Rotas

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

A Figura 7 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 possuem 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 nó B. Feito isto todo o processo de descoberta de rotas é encerrado e a transmissão é iniciada.

Nó caso representado no Exemplo 3, o próprio nó A possui em sua *cache* a tabela de roteamento com todos os endereços do 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.

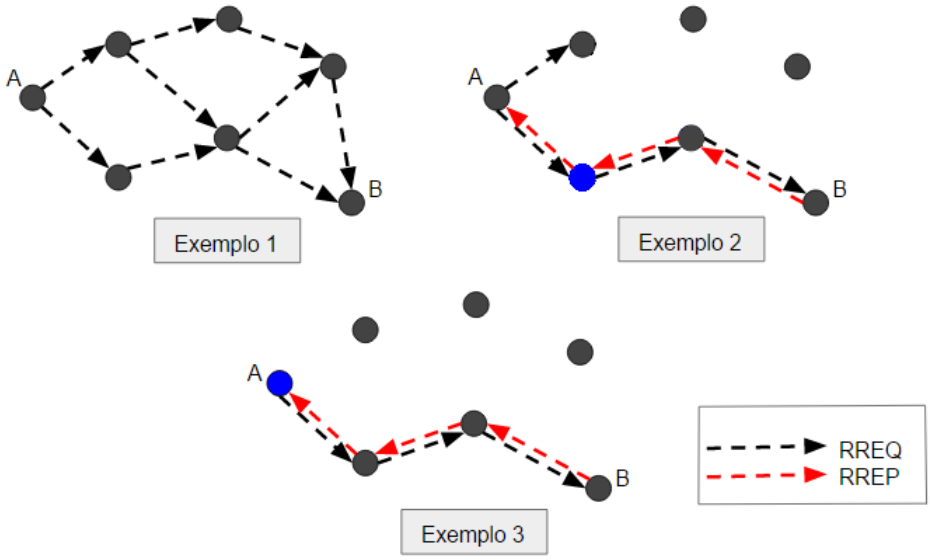

Figura 7 – Descoberta de rotas em um protocolo DSR.

2.6.2 Manutenção de Rotas

No que diz respeito à manutenção das rotas no DSR o procedimento é o mesmo descrito para os protocolos anteriormente estudados. O nó de origem transmite um sinal de verificação (HELLO) através da rota até o destino e aguarda o recebimento de um pacote de dados de confirmação. Caso algum nó não responda ao sinais de verificação isto é considerado erro na comunicação. O erro é reportado ao nó fonte através de um pacote ERROR transmitido para nó fonte contendo o endereço do nó que identificou o erro e o nó que não respondeu ao sinal HELLO, fazendo com que inicie-se um novo processo de descoberta de rotas.

2.7 Visão Geral do Protocolos

Após a apresentação de cada protocolo, a Tabela 2 exibe as principais características do AODV, AOMDV e DSR, respectivamente.

AODV	
Tipo de Protocolo	Reativo
Descoberta de rotas	Através de *broadcast* entre os nós roteadores sem interrupções até que o sinal de requerimento de rota (RREQ) chegue ao nó de destino. Possui apenas uma rota direta e uma rota reversa, que ao primeiro sinal de erro gera uma nova descoberta de rotas.
Manutenção de rotas	Utiliza-se do envio periódico de um pacote de detecção de erro (ERROR) a procura de erros na rota de transmissão de pacotes.
AOMDV	
Tipo de Protocolo	Reativo
Descoberta de rotas	Envia pacotes de requerimento de rota através da rede salvando múltiplas rotas diretas e reversas entre o nó destino e o nó origem. O processo de descoberta de rotas só é novamente inicializado após todas as rotas salvas serem descartadas por detecção de erro.
Manutenção de rotas	Utiliza-se do envio periódico de um pacote de detecção de erro (ERROR) à procura de erros na rota de transmissão de pacotes.
DSR	
Tipo de Protocolo	Reativo
Descoberta de rotas	Envia em *broadcast* sinais de requisição de rota até o nó destino, porém a cada recebimento de um RREQ cada nó analisa se existem rotas válidas entre o nó fonte e nó destino em suas memórias. Caso existam, o processo de descoberta é parado e é dado início à transmissão dos dados. Assim como o AODV possui apenas uma rota direta e uma rota reversa.
Manutenção de rotas	Utiliza-se do envio periódico de um pacote de detecção de erro (ERROR) a procura de erros na rota de transmissão de pacotes.

Tabela 2 – Visão geral dos protocolos estudados.

3 MATERIAIS E MÉTODOS

Tendo adquirido todo o conhecimento teórico acerca dos protocolos de roteamento a serem analisados, apresenta-se agora todo o desenvolvimento de como se deram as simulações e aquisições dos dados que eram relevantes como métricas de avaliação.

3.1 *Network Simulator* (NS-2)

O *Network Simulator* (NS-2) [14] é a segunda versão de um simulador de eventos discretos resultante de um projeto conhecido como VINT (*Virtual InterNetwork Testbed*). Dentre outros, compõem o projeto de desenvolvimento do NS-2 a DARPA, USC/ISI, Xerox PARC, LABNL e a Universidade de Berkeley. Uma grande vantagem do NS-2 reside no fato que ele ser totalmente gratuito e com código fonte aberto, o que permite ao usuário proceder os ajustes que julgar necessários [15].

Este *software* dá a possibilidade de se realizar simulações de redes com e sem fio, utilização de protocolos TCP ou 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).

3.1.1 Ambiente de Simulação

Os blocos básicos para gerar uma simulação no *Network Simulator* são apresentados na Figura 8.

Inicialmente é definido um preâmbulo para configuração dos parâmetros que caracterizarão a rede que será simulada. Nesta etapa, os principais elementos a serem configurados são descritos abaixo:

- Protocolo: define o tipo de protocolo que a rede utilizara para realizar o seu roteamento;
- Antena: determina de que forma se dará a propagação dos sinais na rede, para os casos estudados é utilizado o tipo Omnidirecional (propagação homogênea em todas as direções);

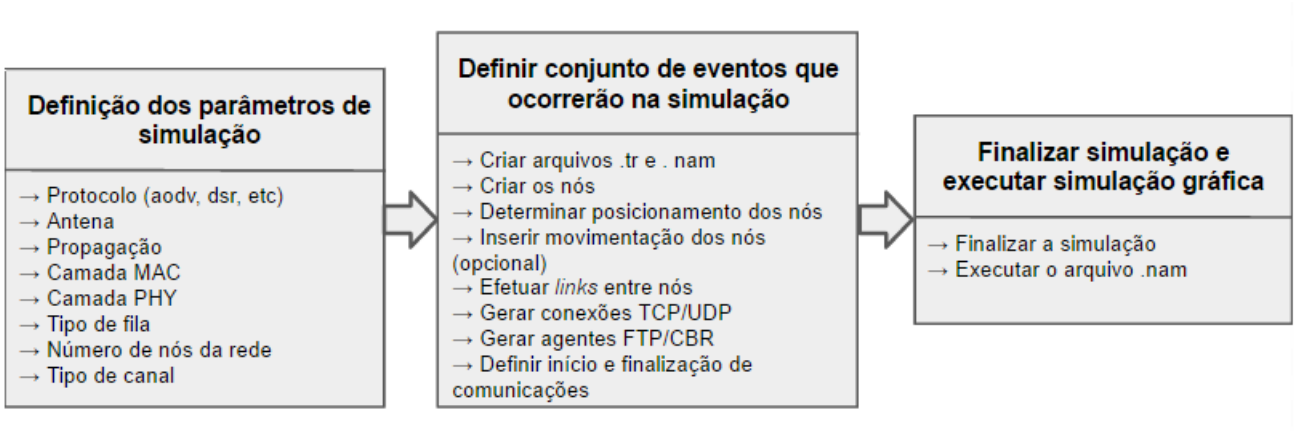

Figura 8 – Fluxograma para desenvolvimento de uma simualação de RSSF no NS-2.

- Propagação: utilizada para determinar os efeitos de propagação entre um *host* e outro. É utilizado o modelo *Two Ray Ground* no desenvolvimento deste trabalho. No modelo *Two Ray Ground*, a terra é assumida como sendo plana e as alturas de transmissor e receptor de antena são pequenas em comparação com a sua distância de separação. Desta forma, a onda de campo elétrica recebida tem duas componentes: uma onda direta e uma onda refletida do terreno;
- Camada MAC: define as regras de comunicação MAC dentro da RSSF simulada;
- Camada PHY: determina a série de padrões de transmissão e codificação para a comunicação utilizada na simulação;
- Tipo de fila: geralmente utiliza-se uma fila de prioridade (*Queue/DropTail/PriQueue*), em que *Queue* está relacionado com interfaces de fila, *DropTail* é considera uma das políticas de administração de lista mais simples, onde todos os pacotes de entrada são descartados após o *buffer* ficar cheio e *PriQueue* vem de *prioritized queues*, ou seja, fila de prioridade. Essa classe deriva da classe *DropTail*. Filas de prioridade operam de forma semelhante a filas *DropTail* mas elas colocam pacotes de alta prioridade e de baixa prioridade no topo e no fim da final, respectivamente [14];
- Número de nós da rede: configura a quantidade de nós da rede;
- Tipo de canal: define o meio físico que vai ser utilizado. Para redes sem fio utiliza-se o tipo de canal *Channel/WirelessChannel*.

O corpo do programa deve conter inicialmente a criação dos arquivos *.trace* e .nam para armazenar os eventos e para gerar a simulação visual, respectivamente. Dando procedimento ao corpo é necessário definir as coordenadas iniciais dos nós que irão compor a rede, sendo possível configura valores nos eixos X, Y e Z, também existe a opção de

gerar movimentação nos nós onde definem-se o ponto inicial, o ponto final e a velocidade com que o nó deverá se deslocar.

Para que se possa realizar a implementação de protocolos de transporte, TCP (*Transport Control Protocol*) e UDP (*User Datagram Protocol*), é necessária a criação de agentes, que são componentes da arquitetura NS-2 responsáveis pela simulação destes protocolos. Os agentes criam um canal de comunicação entre os nós transmissor e receptor [16]. Fixado o protocolo de transporte é necessário gerar os agentes que se encarregam do tráfego de dados, no que diz respeito a esses agentes o *Network Simulator* utiliza agentes CBR (*Constant Bit Rate*) e FTP (*File Transfer Protocol*) para efetuar essas atividades.

Por último, são determinados os instantes de início e término da comunicações entre os nós fonte e destino. Finalizada a simulação como um todo, é então necessário executar o arquivo .nam para que se possa observar a simulação de uma forma visual, mostrando os nós em si além das transferências de pacotes de dados, tornando assim mais compreensível a visualização da simulação para o usuário, devido ao NS-2 em si gera apenas arquivos de texto contendo os eventos ocorridos durante a simulação.

Ao término de uma simulação, uma janela semelhante à mostrada na Figura 9 é aberta para que o usuário possa acompanhar de forma visual toda a simulação.

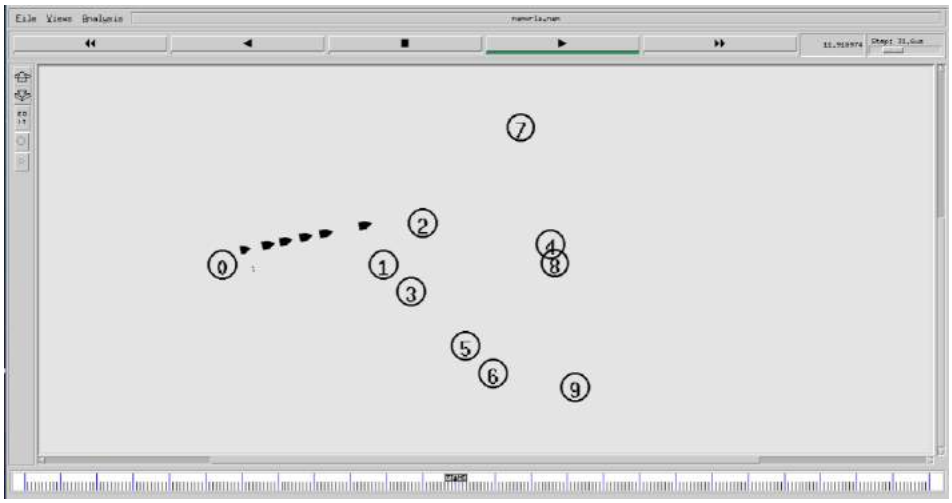

Figura 9 – Tela de simulação do NAM.

Executando-se uma simulação de um arquivo OTCL através do comando *"ns nome-do-arquivo.tcl"* inserido no *prompt* de comando do computador, o compilador do NS-2 realiza todo o processamento dos parâmetros de simulação e os executa, simultaneamente armazena todos os eventos ocorridos em um arquivo *.trace*. A Figura 10 ilustra uma linha de um arquivo *trace* que corresponde a um evento. Suas respectivas funções são descritas em seguida.

- *Event* (Evento): corresponde a quatro tipos de eventos diferentes que um pacote

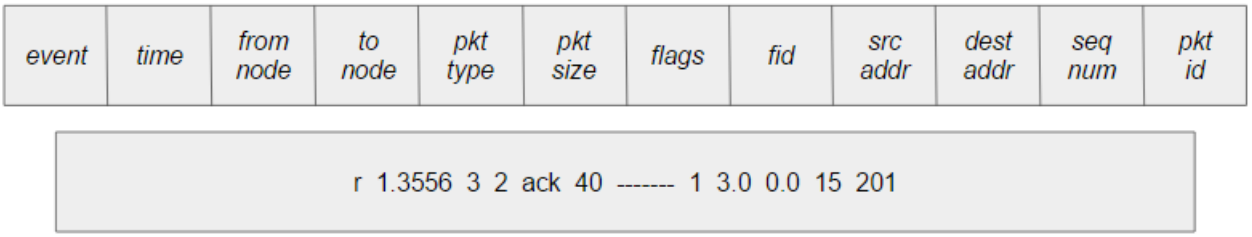

Figura 10 – Código gerado por um arquivo *trace*.

pode ter, são eles 'r' para recebimento, '+' entrada de pacote na fila, '-' saída de pacote na fila e 'd' perda de pacote;

- *Time* (Tempo): inicia-se de zero indo até o valor pré-definido para o final da simulação, denotando o tempo em que um determinado evento ocorreu;
- *From node* e *To node* (Nó origem e Nó destino): nós de onde partiu um pacote e nó onde o pacote chegou durante um evento qualquer;
- *Packet type* (Tipo de pacote): indica o tipo de pacote no *link*, podendo ser CBR's, ACK's, TCP's, etc.;
- *Packet size* (Tamanho de pacote): define o tamanho do pacote;
- *Flags*: indica qualquer comportamento anormal. Uma saída do tipo '- - - - - - -' denota que não existem *flags* de comportamentos irregulares;
- *Flow identifier* (Identificador de fluxo): identificador de fluxo dos pacotes durante a transmissão;
- *Source address* e *Destination addres* (Endereço da fonte e Endereço do destino): marcam os endereços da fonte e do destino;
- *Sequence number* (Número de sequência): usado para montagem de pacotes no destino de forma correta;
- *Packet identifier* (Identificador de pacotes): mantem o controle de todos os pacotes.

3.2 Obtenção dos 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. Tendo em vista essa filtragem dos momentos onde ocorrem eventos relevantes para os estudos das métricas, que serão descritos na Seção 3.3, é utilizado um código na linguagem AWK que é uma linguagem de programação interpretada que utiliza comandos

C/C++, geralmente usada para deixar os *scripts* mais poderosos e com mais recursos sendo capaz de processar dados de texto.

Para os estudos em questão, o fato dos arquivos .awk processarem dados de texto é de suma importância, pois assim é possível tratar o arquivo *trace* proveniente da simulação no *Network Simulator*, que não deixa de ser um arquivo de texto.

Para efeitos de análise de dados é vital que um número mínimo de repetições das simulações sejam feitas para obtenção de uma análise estatística (descrita na Seção 3.4) confiável. Portanto, para realizar essa repetição de eventos de forma aleatória são implementados códigos para gerar variáveis aleatórias dentro do próprio código OTCL. Assim, a cada nova simulação um novo nó fonte e um novo nó destino são gerados aleatoriamente. Então são feitas simulações no NS-2 seguida pela execução do arquivo .awk para observar os resultados das métricas qualitativas.

Porém, efetuar um número tão elevado de repetições de um mesmo *script* demanda muito tempo do programador, tendo em vista que deve ser executado o arquivo no NS-2, e em seguida o programa para extrair os dados relevantes para cada análise de formas separadas e manualmente. Pensando nisto, 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. Uma ilustração do funcionamento dos *scripts* .awk e *bash* é mostrada na Figura 11:

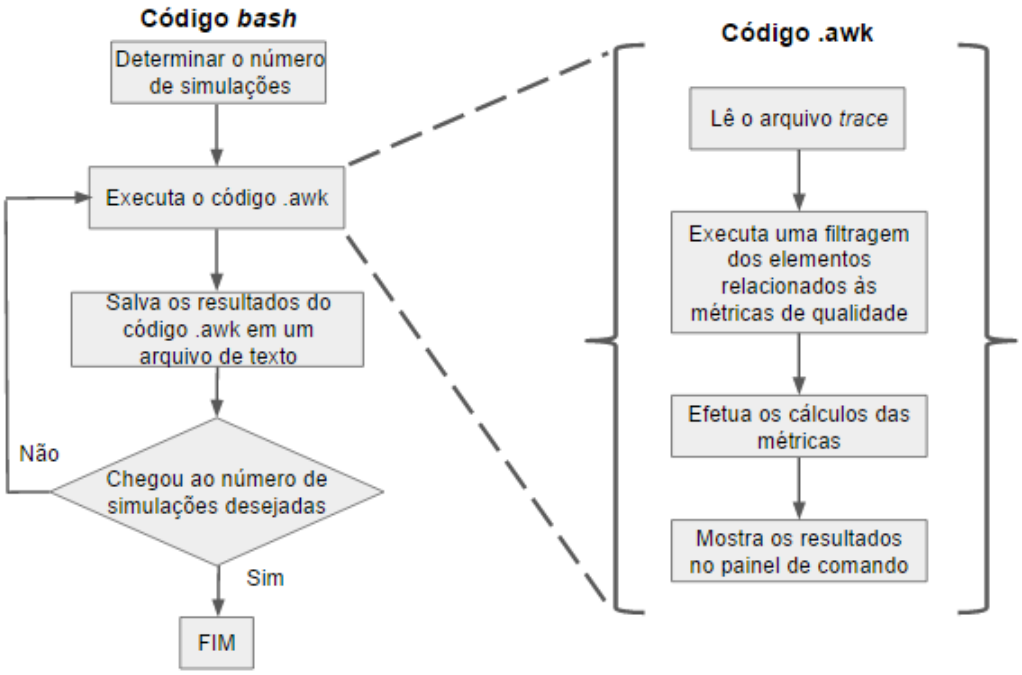

Figura 11 – Funcionamento dos códigos .awk e *bash*.

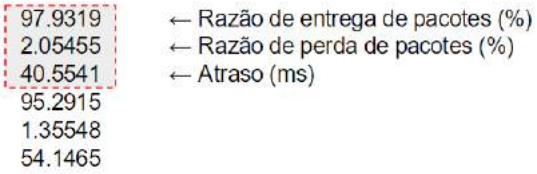

Figura 12 – Formato dos arquivos .txt.

Desta forma, todo o esforço a ser realizado pleo usuário é substituído por esforço puramente computacional, permitindo que um número muito elevado de simulações seja feito sem muitos problemas, requerendo apenas tempo para que todas as simulações sejam efetuadas. Os resultados gerados em arquivo de texto possuem uma sequência específica, como é descrita na Figura 12, onde são exemplificadas duas sequências:

3.3 Métricas de Análise Qualitativa

Para efetuar o processo de análise qualitativa entre os três diferentes protocolos descritos no capítulo anterior foram adotados três parâmetros: razão de entrega de pacotes (*Packet Delivery Ratio* - PDR), razão de perdas de pacotes (*Packet Drop Rate*) e o atraso médio para a chegada dos pacotes de dados. São tomadas essas métricas para avaliar a eficiência, confiabilidade e velocidade com que os protocolos de roteamento realizam suas atividades, posto que essas características pesam na análise geral de um protocolo.

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 [17]. Matematicamente, pode ser definida por:

$$PDR = \frac{S_R}{S_T} \tag{3.1}$$

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 visto 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 3.2.

$$Packet\ Drop\ Rate = \frac{S_P}{S_T} \tag{3.2}$$

Similarmente à Equação 3.1 S_T continua sendo o número de pacotes enviados por cada fonte, S_P denota o número de pacotes perdidos durante toda a transmissão de dados.

Como último critério de avaliação de 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.

3.4 Método Estatístico para Análise dos Resultados

Para determinar os resultados finais apresentados neste trabalho, foi necessária uma análise estatística utilizando as simulações realizadas para cada estudo de caso, dado que um único evento não pode representar de forma fidedigna um cenário específico.

As análises estatísticas dos fatores de qualidade são feita através de uma mescla entre a análise por quartis e análise por distribuição z. As Seções 3.4.1 e 3.4.2 a seguir definem mais detalhadamente cada técnica.

3.4.1 Distribuição z

Segundo [18], 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 são observados um conjunto de amostras $\{x_1, x_2, ..., x_n\}$ 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 $\delta/\sqrt{n}$ [18].

$$\bar{x} \approx N\left(\mu, \frac{\delta}{\sqrt{n}}\right) \tag{3.3}$$

Usando o teorema do limite central, o intervalo de confiança para a média de uma população é dado pela Equação 3.4:

$$\left(\mu - z_{1-\alpha/2} \times \frac{\delta}{\sqrt{n}}, \mu + z_{1-\alpha/2} \times \frac{\delta}{\sqrt{n}}\right) \tag{3.4}$$

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 [18].

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 [3].

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 chegam em resultados aceitáveis de confiabilidade. Assim, para um $\Delta = 5\%$ o intervalo de confiança é definido por:

$$(\mu(1 - 0.05), \mu(1 + 0.05)) \tag{3.5}$$

Utilizando as Equações 3.4 e 3.5, 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:

$$n_{min} = \left(\frac{1.96 \times \delta}{0.05 \times \mu}\right)^2 \tag{3.6}$$

Inicialmente são feitas para cada estudo de caso um total de 50 simulações para que através destas sejam concebidas médias μ e desvios padrões δ. Com esses parâmetros e sabendo que deseja-se uma confiabilidade de 95% ($\Delta = 0.05$), é permitido realizar com a utilização da Equação 3.6 o cálculo do número mínimo de simulações para se alcançar os resultados com a confiabilidade desejada em cada caso estudado.

3.4.2 Análise por Quartis

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. Por exemplo, em algumas simulações os resultados provenientes de algumas das métricas se mostrava visivelmente distante da média, fossem eles maiores (gerando resultados muito elevados) ou menores (criando resultados nulos).

Assim os dados obtidos através dos métodos descritos na Seção 3.2 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 ordenado de forma crescente e então os quartis das extremidades foram retirados, de maneira que foram feitos estudos apenas com os quartis centrais. A Figura 13 ilustra o procedimento para uma matriz com quatro valores.

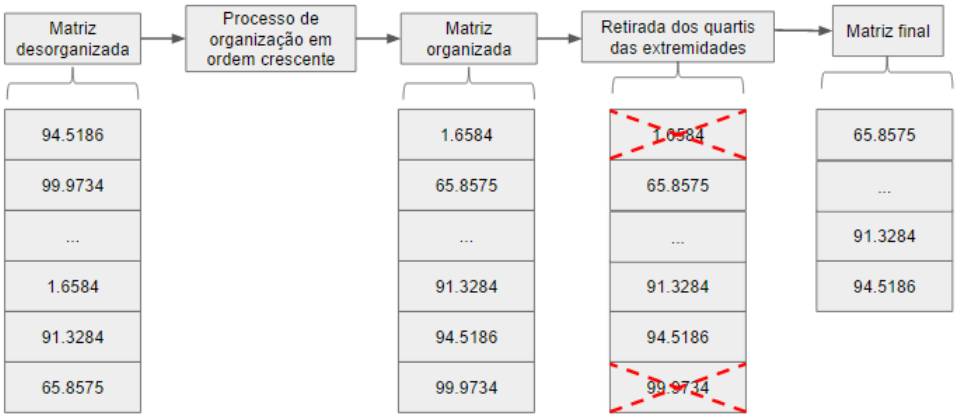

Figura 13 – Separação dos resultados em quartis.

A partir das matrizes obtidas após a separação por quartis é viável efetuar os estudos por distribuição z descritos na Seção 3.4.1 sem que ocorram erros graves.

3.5 Estudos de Caso

Como a proposta deste trabalho é uma avaliação da qualidade e confiabilidade de diferentes protocolos de roteamento, é necessária a realização de simulações em diferentes cenários que venha a definir uma possível situação real de aplicação dos protocolos. Para tal foram propostos ao todo cinco tipos diferentes de cenários: três cenários hipotéticos e dois cenários reais.

3.5.1 Cenário 1

O primeiro cenário é apresentado na Figura 14:

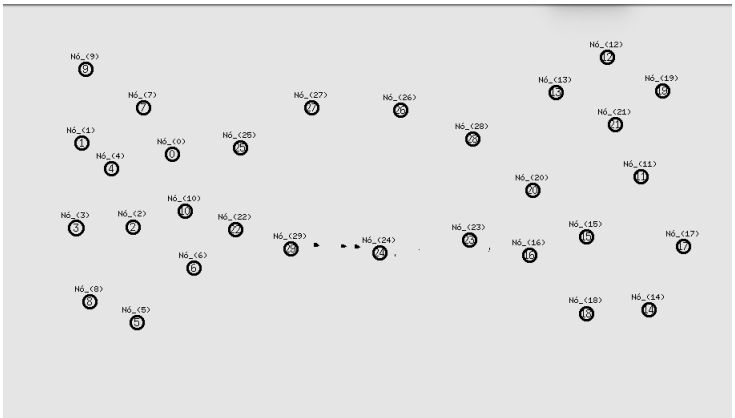

Figura 14 – Cenário hipotético 1 para estudo de caso.

Este cenário emula uma situação em que um dos nós que se encontra no conjunto da esquerda necessita efetuar envio de dados para algum dos nós que se encontram no

conjunto da direita, de modo que existem dois caminhos distinto por onde os pacotes de dados podem ser transmitidos. Para tal cenário foi gerado um nó fonte aleatório dentre os nós da esquerda e um nó de destino, também aleatório, no conjunto da direita, em que a forma com que o nó define sua rota (por cima ou por baixo) se dá por meio do protocolo de roteamento.

3.5.2 Cenário 2

O cenário 2, assim como o cenário 1, possui uma distribuição de nós situados mais à esquerda e outro conjunto de nós situado à direita. O diferencial entre o cenário anterior e este se encontra na disposição dos nós que interligam os dois conjuntos, onde agora existe apenas um único caminho entre os dois lados. Esse cenário pode ser utilizado para avaliar se os protocolos produziram mais atrasos, provenientes de *loops* momentâneos na "ponte"que liga os dois lados. A ilustração do cenário 2 é mostrada na Figura 15 abaixo:

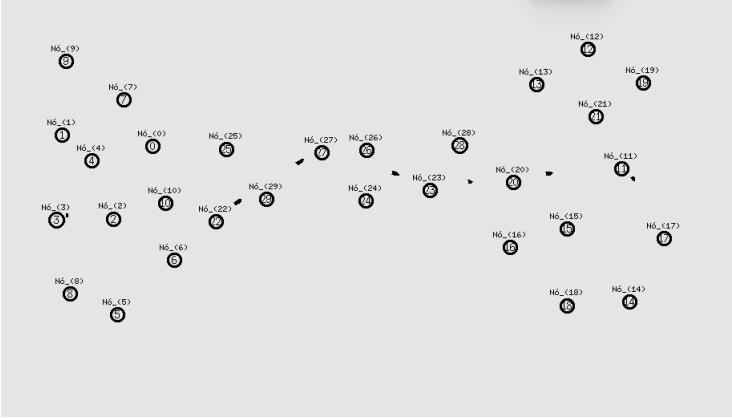

Figura 15 – Cenário hipotético 2 para estudo de caso.

3.5.3 Cenário 3

Enquanto os cenários 1 e 2 lidavam com situações em que existem limitações, representadas por gargalos localizados no centro das redes, o cenário 3 deixa os protocolos com um grau bem maior de liberdade para escolhas de rotas. Para este caso foram feitas simulações com transmissões entre nós localizados mais abaixo na rede se comportando como fonte e nós localizados mais acima da rede atuando como destinos, assim como anteriormente as transmissões foram executadas de forma aleatória. Os nós foram distribuídos como é apresentado na Figura 16.

3.5.4 Cenário 4

O quarto cenário é voltado para a implementação dos protocolos de roteamento em ambientes reais, onde é simulado neste cenário o segundo andar do prédio do Centro

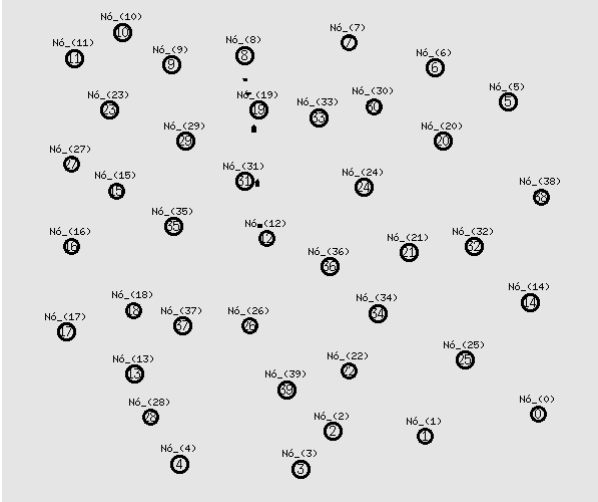

Figura 16 – Cenário hipotético 3 para estudo de caso.

de Energias Alternativas e Renováveis (CEAR), que ainda está em fase de construção. Este andar foi escolhido por, após análises, ser o andar que mais demanda nós. A Figura 17 demonstra a disposição dos nós sensores/roteadores (representados por pontos pretos, vermelhos e azuis na planta).

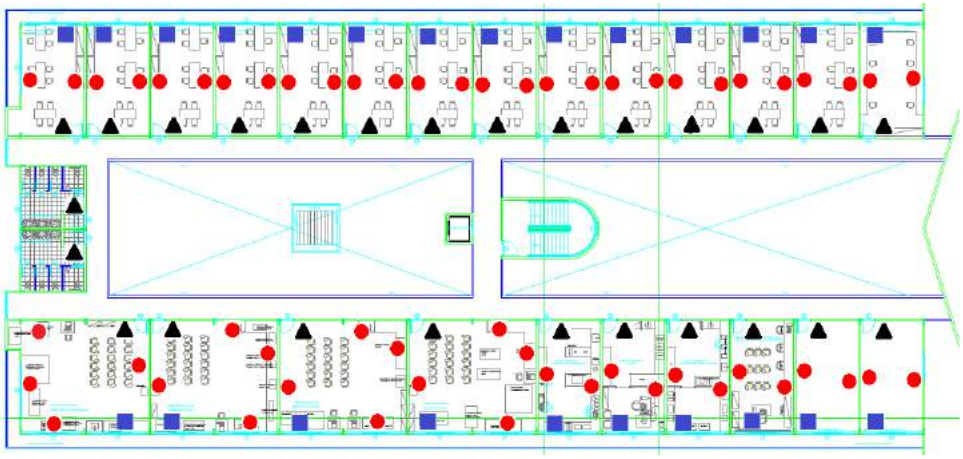

Figura 17 – Cenário real 1 para estudo de caso.

Basicamente são utilizados, nesta aplicação, nós de três tipos:

- Nós de iluminação (triângulos): responsáveis pelo controle de iluminação das salas;
- Nós para ar-condicionado (quadrados): utilizados para controlar o funcionamento dos ares-condicionados;
- Nós auxiliares (círculos): disponível em salas que venham a necessitar de mais terminais, como salas de professores ou laboratórios.

3.5.5 Cenário 5

Igualmente ao cenário 4, o cenário 5 propõe-se à análise da eficiência dos protocolos de roteamento em outro dos andares do prédio do CEAR. Ao contrário do cenário 5, este foi selecionado por apresentar a menor disposição de nós e, assim como o cenário real anterior, possui os mesmos três tipos de nós. As disposições dos nós são apresentadas na Figura 18.

Através destas duas simulações, é possível observar por meio dos resultados obtidos se um protocolo que se mostra eficiente em um cenário com poucos nós pode baixar drasticamente sua qualidade pelo simples fato de se aumentar o número de nós.

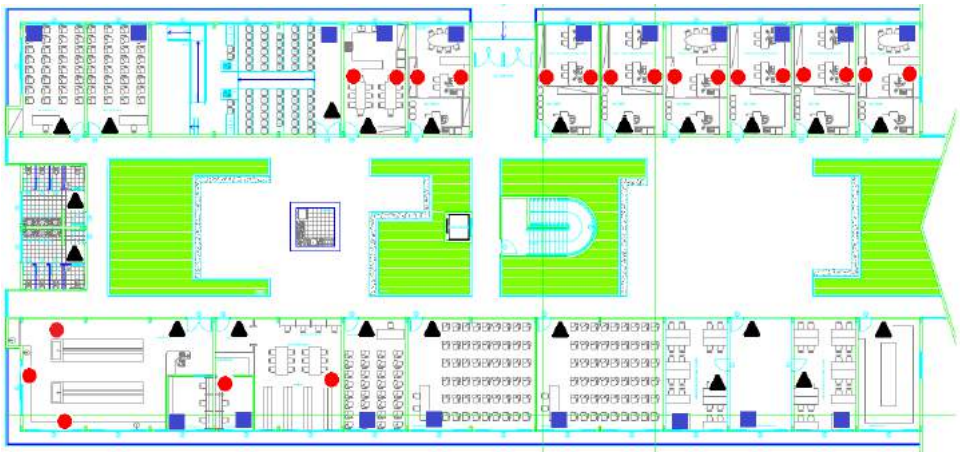

Figura 18 – Cenário real 2 para estudo de caso.

4 RESULTADOS

Após implementações de inúmeras simulações são agora mostrados todos os resultados encontrados para cada cenário a partir das simulações efetuadas no *Network Simulator*, seleções de dados através dos códigos awk e *bash* e tratamento estatístico do MATLAB.

4.1 Cenário 1

Ao se observar a distribuição dos nós na rede retratada na Seção 3.5.1, era de se esperar que o protocolo de roteamento utilizando o DSR obtivesse um melhor desempenho em comparação com os outros dois protocolos. Isso se dá devido ao método de descoberta de rotas do DSR, que possui a capacidade de armazenar rotas não utilizadas por um período de tempo, e como o número de rotas possíveis é bastante limitado isso representaria uma vantagem para este protocolo. Os histogramas para os três protocolos são apresentados a seguir nas Figuras 19, 20 e 21.

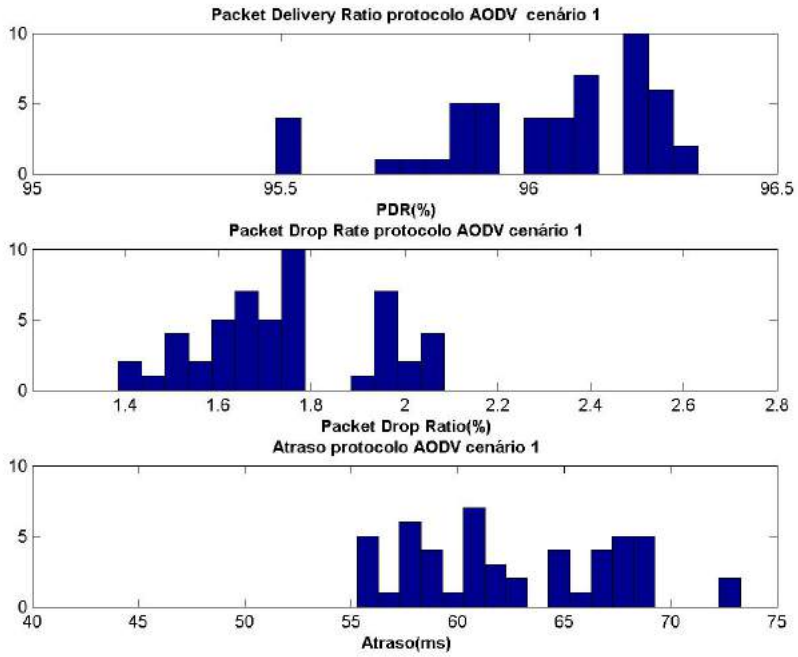

Figura 19 – Resultados do cenário 1 para o protocolo AODV.

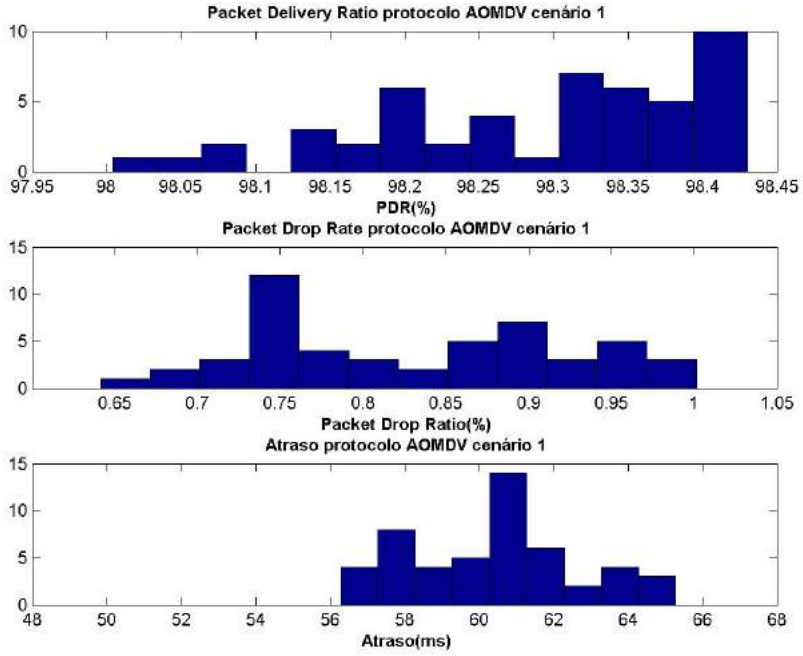

Figura 20 – Resultados do cenário 1 para o protocolo AOMDV.

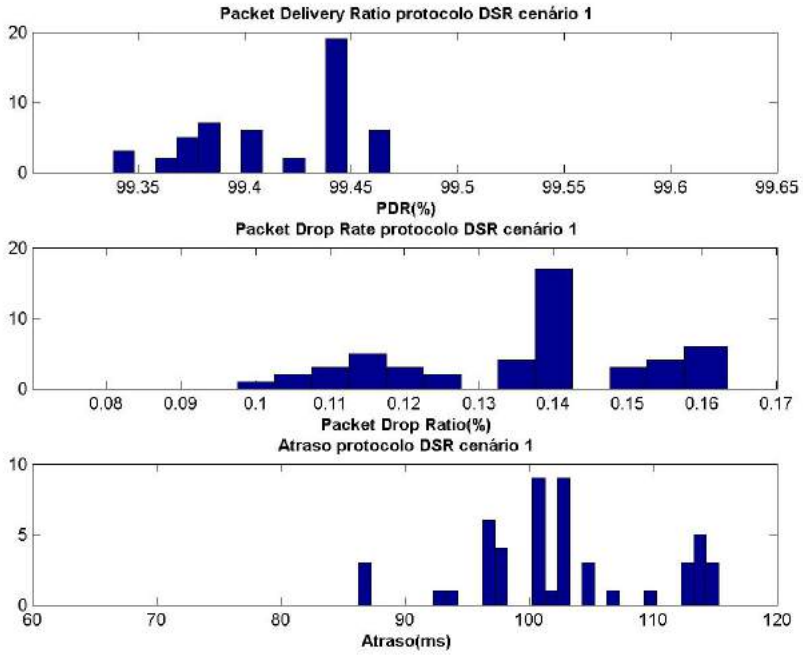

Figura 21 – Resultados do cenário 1 para o protocolo DSR.

Ao se observar os resultados apresentados acima, nota-se que o protocolo DSR apresenta, de fato, um excelente desempenho ao se levar em consideração apenas a razão de entrega da pacotes e razão de perda de pacotes, 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 a métricas de qualidade é o protocolo AOMDV. Uma tabela com as médias da métricas qualitativas e o número (n_{min}) de simulações referente a cada protocolo é apresentada na Tabela 3.

Métrica	**AODV**	**AOMDV**	**DSR**
PDR (%)	96.0402	98.2874	99.4132
Razão de perda de pacotes (%)	1.7469	0.8288	0.1358
Atraso (ms)	62.7082	60.3768	102.7323
n_{min}	17.0221 $\approx$ 18	19.9118 $\approx$ 20	24.3337 $\approx$ 25

Tabela 3 – Médias das métricas qualitativas para o cenário 1.

4.2 Cenário 2

Aplicando-se os métodos de análise de qualidade dos protocolos ao cenário 2, era de se esperar que uma equiparação entre os protocolos DSR e AOMDV fosse alcançada, já que este cenário propõe a existência de apenas um caminho de ligação entre as os nós mais à esquerda e os nós mais à direita. Os resultados são apresentados nas Figuras 22, 23 e 24.

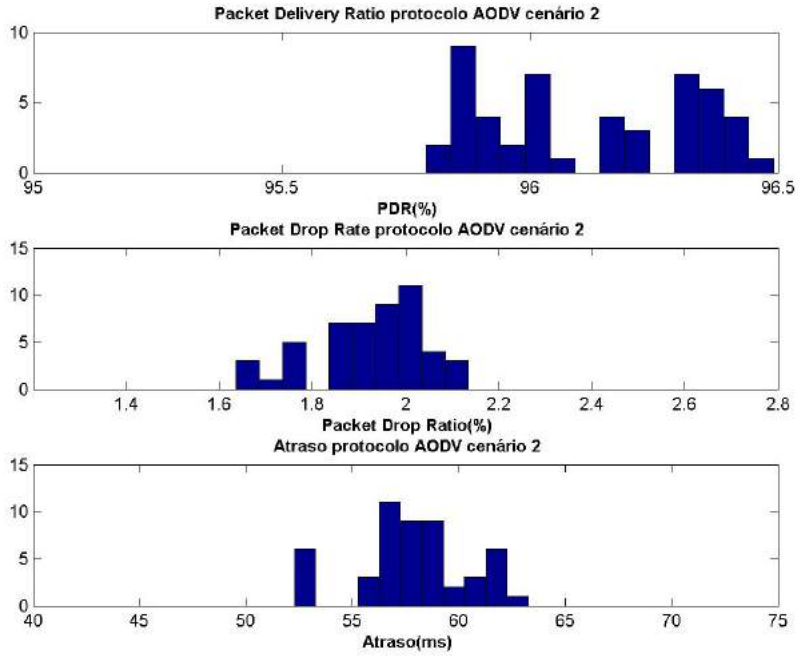

Figura 22 – Resultados do cenário 2 para o protocolo AODV.

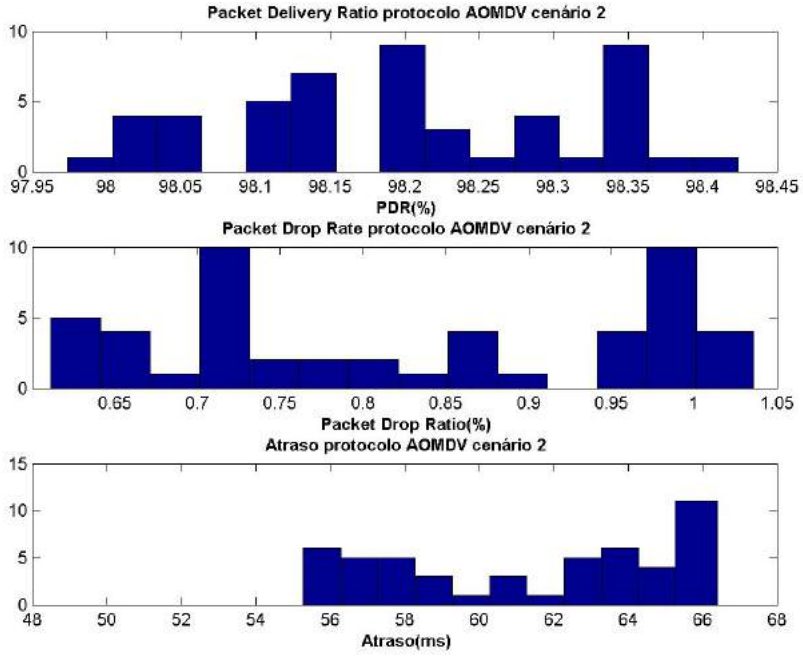

Figura 23 – Resultados do cenário 2 para o protocolo AOMDV.

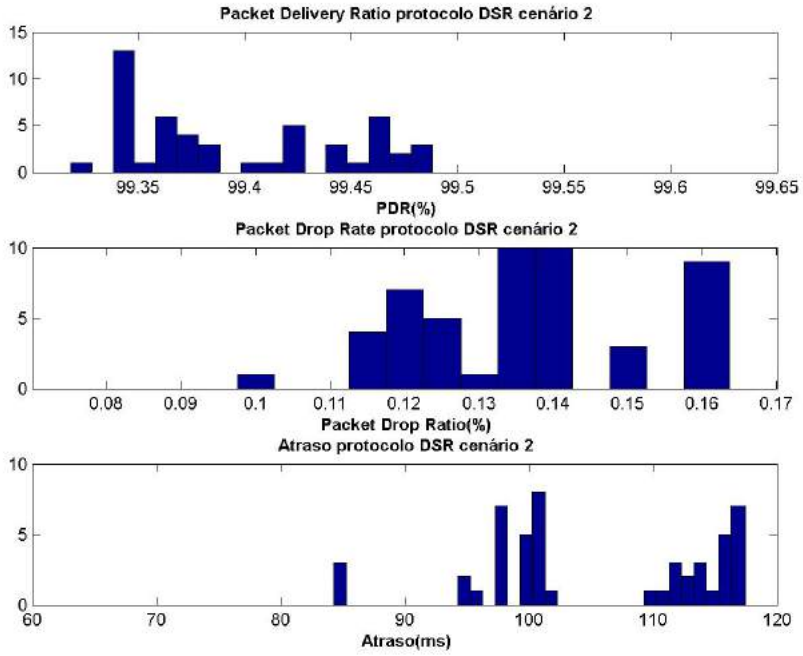

Figura 24 – Resultados do cenário 2 para o protocolo DSR.

Como se pode observar nos histogramas e na Tabela 4 com as 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 e por consequência baixas taxas de perda de pacotes, algo desejável em um protocolo

Métrica	**AODV**	**AOMDV**	**DSR**
PDR (%)	96.1277	98.1972	99.3961
Razão de perda de pacotes (%)	1.9211	0.8244	0.1363
Atraso (ms)	57.8604	61.3600	105.3615
n_{min}	5.6951 $\approx$ 6	44.6391 $\approx$ 45	20.1308 $\approx$ 21

Tabela 4 – Médias das métricas qualitativas para o cenário 2.

de roteamento eficiente. Para este cenário o protocolo AOMDV novamente se mostrou o mais competente quando observadas todas as métricas em conjunto.

4.3 Cenário 3

Os resultados para todos os protocolos neste cenário se mostraram melhores considerando-se uma análise global em comparação com os cenários 1 e 2, pois apesar de ocorrer algumas perdas de qualidade em certos parâmetros qualitativos em alguns protocolos, outros parâmetros tiveram aumentos bem mais significativos e com maiores contribuições positivas para o desempenho do protocolo como um todo. As médias das métricas são apresentadas na Tabela 5.

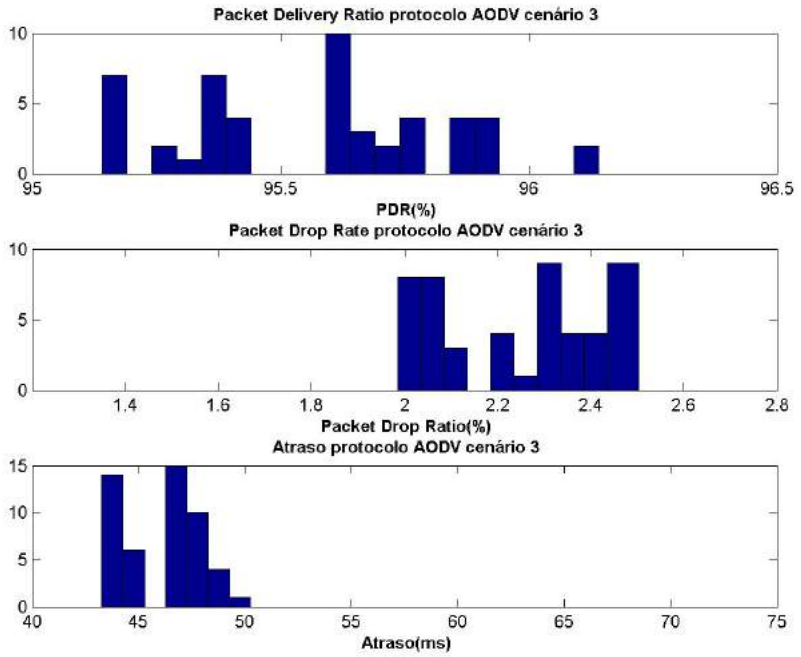

Figura 25 – Resultados do cenário 3 para o protocolo AODV.

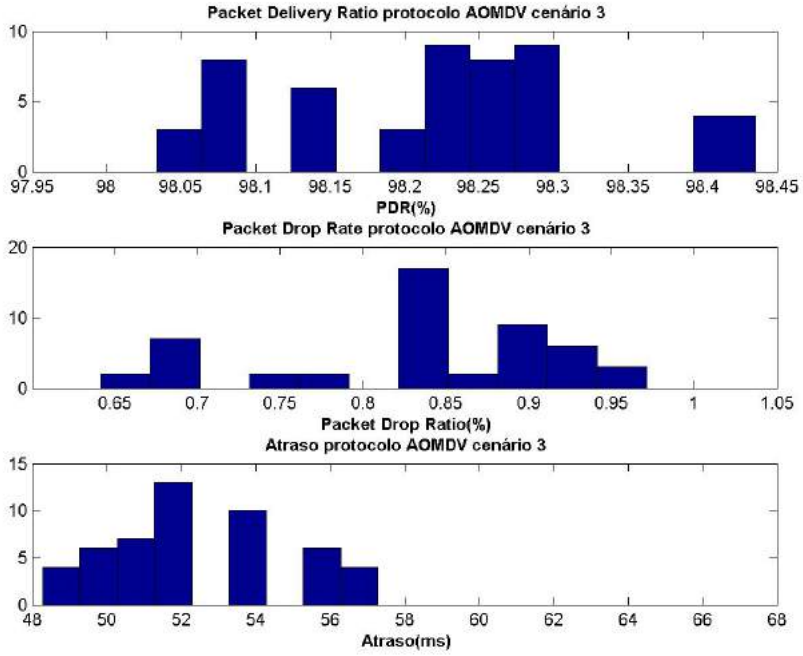

Figura 26 – Resultados do cenário 3 para o protocolo AOMDV.

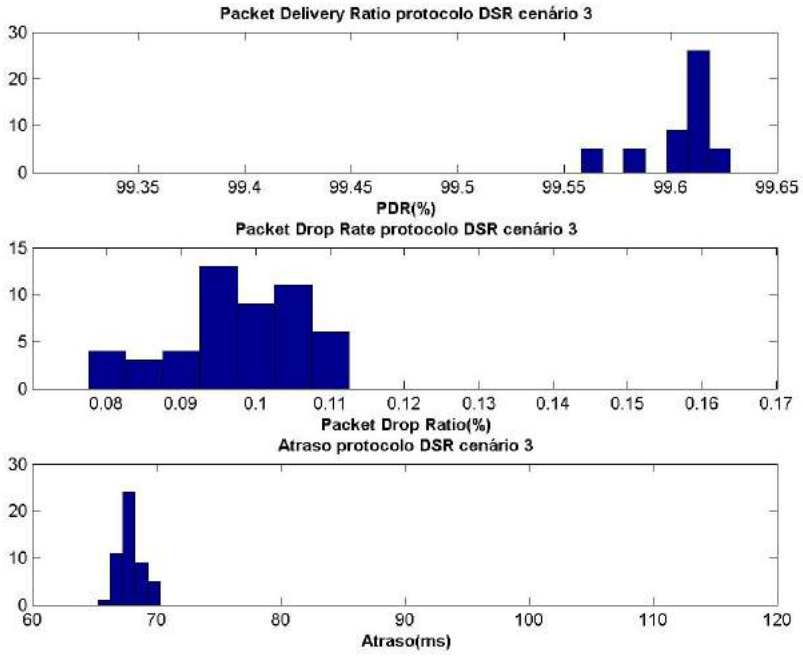

Figura 27 – Resultados do cenário 3 para o protocolo DSR.

Através dos resultados obtidos vê-se que agora o protocolo DSR apresentou uma redução significativa no seu atraso, o que faz com que ele apresente os melhores resultados, pois apesar de ainda possuir um maior atraso se comparado com os outros dois protocolos, um atraso médio de 67.9610 ms é aceitável para muitas aplicações em RSSF visto que

Métrica	**AODV**	**AOMDV**	**DSR**
PDR (%)	95.5628	98.2170	99.6055
Razão de perda de pacotes (%)	2.2490	0.8318	0.0974
Atraso (ms)	46.2218	52.5582	67.9610
n_{min}	9.5291 $\approx$ 10	15.9576 $\approx$ 16	12.8610 $\approx$ 13

Tabela 5 – Médias das métricas qualitativas para o cenário 3.

essa latência não afeta de forma relevante a rede quanto ao seu funcionamento.

4.4 Cenário 4

Agora apresentam-se os resultados obtidos a partir da análise de casos baseados em cenários reais. O cenário 4 descrito na Seção 3.5.4 ilustra o andar do prédio do CEAR com a maior demanda de nós sensores/roteadores, o que reduz as expectativas de que o protocolo DSR possua um desempenho melhor dentre os três protocolos. Os resultados dos protocolo AODV, AOMDV e DSR são expressos graficamente nas Figuras 28, 29 e 30,respectivamente. A Tabela 6 apresenta os valores médios das métricas de avaliação.

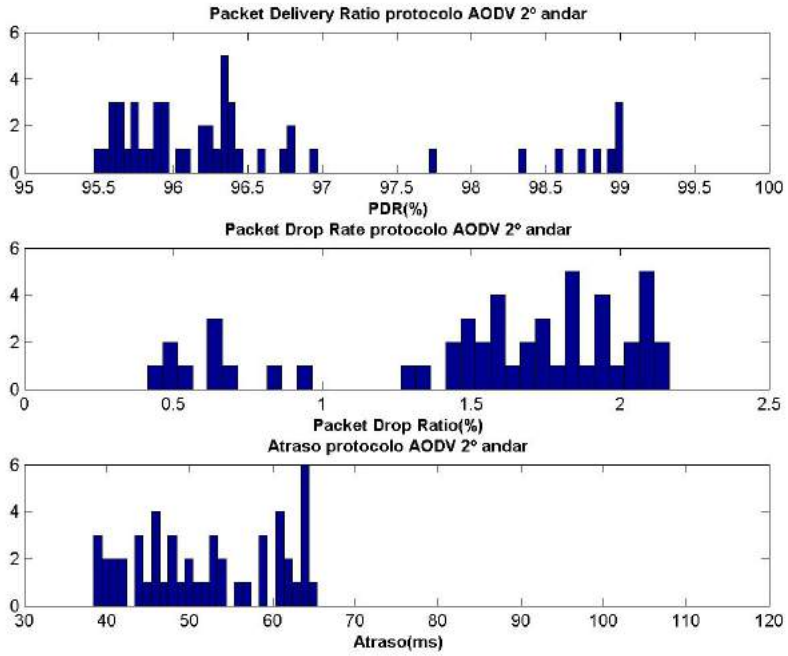

Figura 28 – Resultados do cenário 4 para o protocolo AODV.

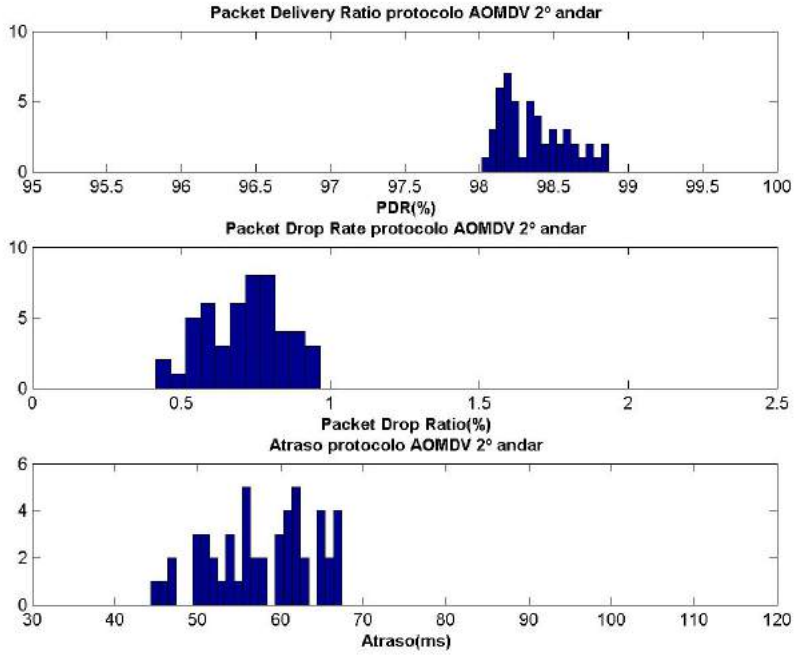

Figura 29 – Resultados do cenário 4 para o protocolo AOMDV.

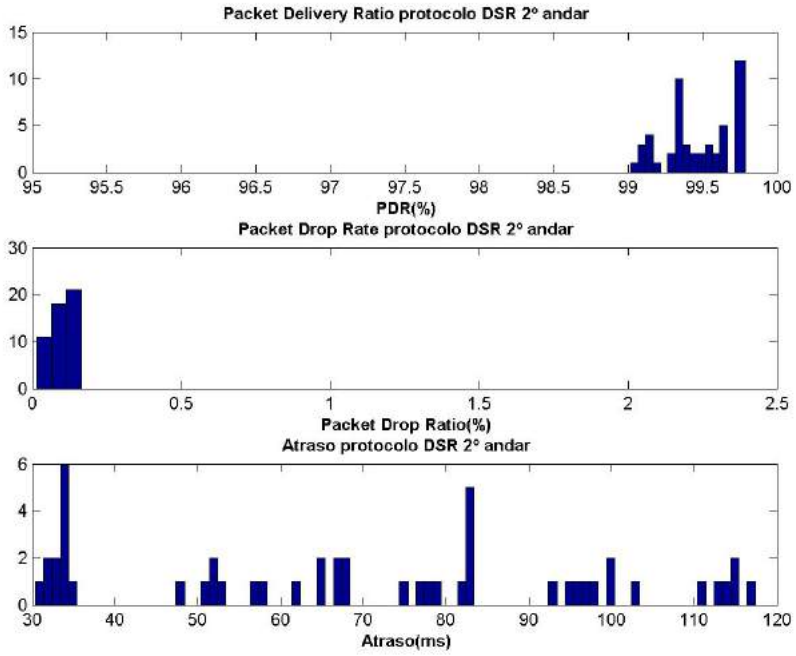

Figura 30 – Resultados do cenário 4 para o protocolo DSR.

Analisando os resultados apresentados em cada histograma, 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 peca no seu atraso acumulado na entrega de pacotes, e apesar de possuir os melhores

Métrica	**AODV**	**AOMDV**	**DSR**
PDR (%)	96.5504	98.3704	99.4680
Razão de perda de pacotes (%)	1.5491	0.7127	0.1025
Atraso (ms)	52.0440	57.6548	70.0023
n_{min}	167.1440 $\approx$ 168	53.1015 $\approx$ 54	240.7575 $\approx$ 241

Tabela 6 – Médias das métricas qualitativas para o cenário 4.

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.

4.5 Cenário 5

Finalizando as análises dos resultados chega-se ao cenário 5, que é uma representação de um dos andares também do prédio do CEAR que desta vez possui a característica de ser o andar com menor demanda de nós. Esse tipo de cenário deve proporcionar uma melhoria no desempenho de todos os protocolos, pois ao se reduzir o número de nós os erros de redundâncias que devem ser corrigidos nos processos de descobertas ocorrem com maior dificuldade, deixando assim o protocolo mais rápido. As representações dos resultados são mostradas nas Figura 31, 32 e 33, em seguida são descritos os valores médios das métricas de avaliação na Tabela 7.

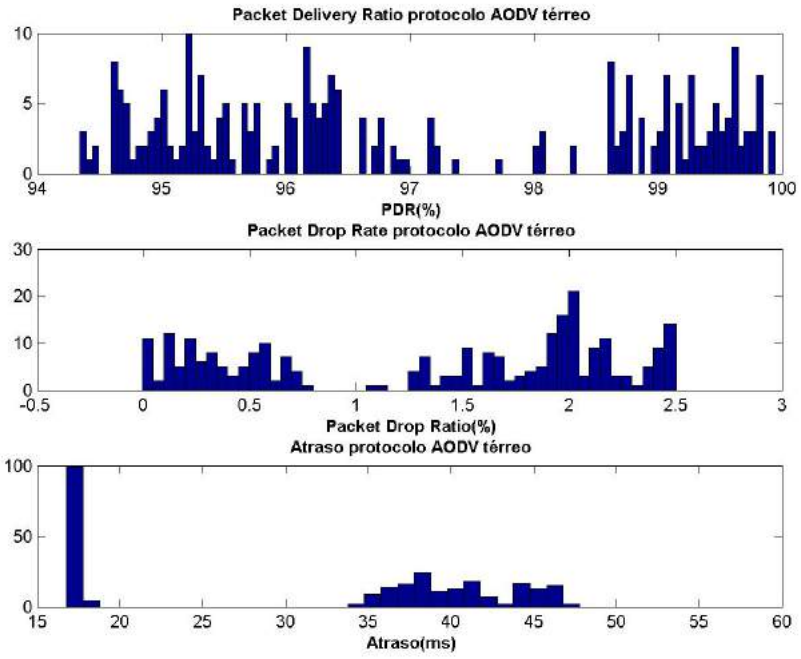

Figura 31 – Resultados do cenário 5 para o protocolo AODV.

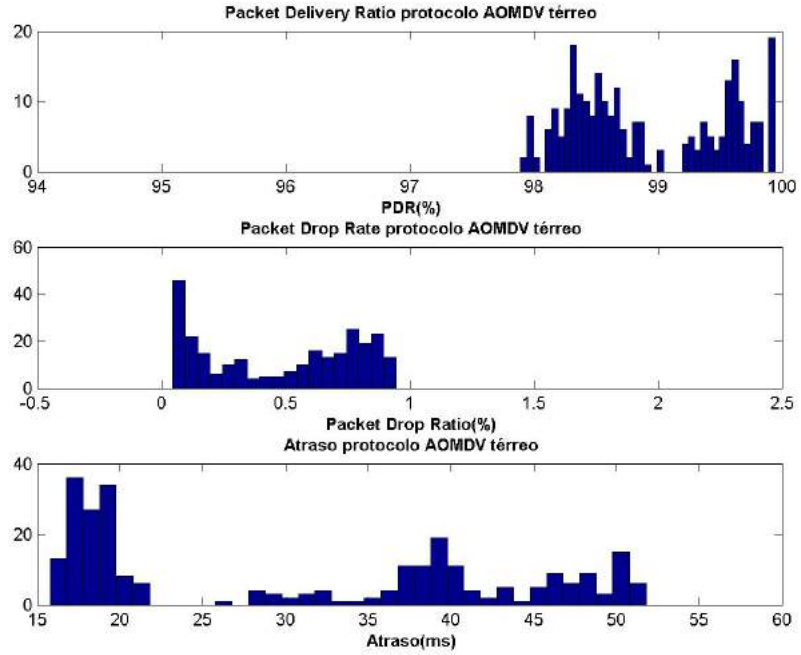

Figura 32 – Resultados do cenário 5 para o protocolo AOMDV.

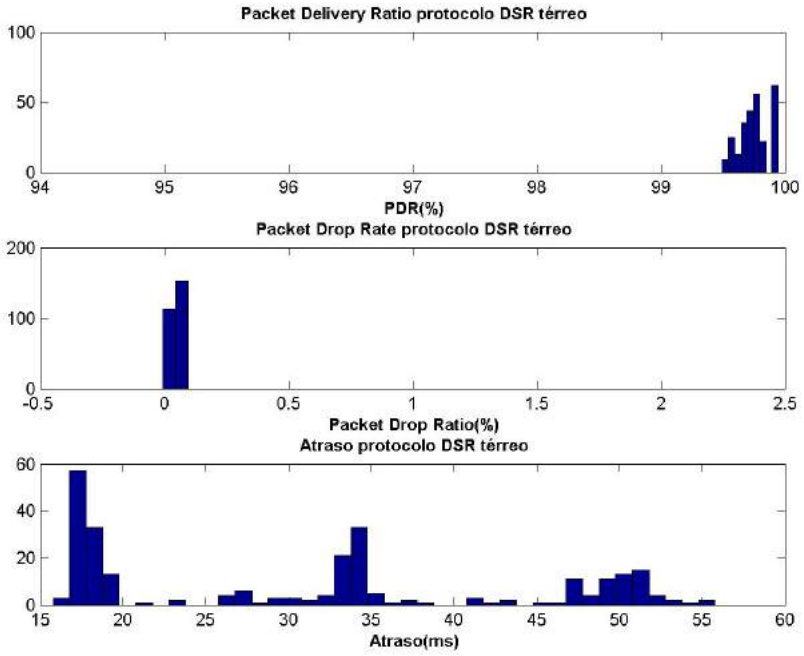

Figura 33 – Resultados do cenário 5 para o protocolo DSR.

Como esperado, os protocolos tiveram melhoras bastante 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 consequentemente uma razão de perda de

Métrica	**AODV**	**AOMDV**	**DSR**
PDR (%)	97.0299	98.9281	99.7541
Razão de perda de pacotes (%)	1.3411	0.4795	0.0502
Atraso (ms)	31.6439	30.6935	31.1407
n_{min}	$1449.4 \approx 1450$	$622.2544 \approx 623$	$229.5662 \approx 230$

Tabela 7 – Médias das métricas qualitativas para o cenário 5.

pacotes mínima e ainda assim possui um atraso parelho com os obtidos para os protocolos AODV e AOMDV.

5 CONCLUSÃO

Neste trabalho foram estudados os comportamentos dos protocolos de roteamento AODV, AOMDV e DSR com o propósito de comparar a eficiência dos mesmos em diferentes ambientes de aplicação, tanto hipotéticos quanto reais. A escolha dos três protocolos estudados foi pensada de modo a desenvolver uma análise de diferentes métodos de descobertas de rotas a fim de determinar qual deles apresentaria um melhor desempenho levando-se em consideração os cenários de aplicações.

Para tal comparação, foi utilizado o *Network Simulator* por apresentar bastante flexibilidade e por ser um *software* aberto, com possibilidade de modificação de código para melhor adequação às necessidades de cada estudo. Partindo dos resultados de simulação obtidos no NS-2 foi então efetuado um tratamento do dados para selecionar apenas o que era de interesse, no caso as métricas de análise: razão de entrega de pacotes, razão de perda de pacotes e atraso para chegada de dados ao destino. Foram então aplicadas análises estatísticas descritas nas Seções 3.4.1 e 3.4.2, para que assim os resultados possuíssem um certo nível de confiança.

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 cerda de 50.0957 ms considerando todos os cenários estudados. O DSR mostrou ser um protocolo focado em 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.

Conclui-se, desta maneira que os objetivos almejados por este trabalho foram alcançados, uma vez que o dados resultantes destas comparações de protocolos facilitarão o estudo bem como a implementação de futuras RSSF's.

5.1 Trabalhos Futuros

Após os estudos realizados, as propostas abaixo podem ser utilizadas em trabalhos futuros:

- Efetuar a implementação dos protocolos estudados aplicando-os em módulos *XBee* nos cenários estudados, para que comparações sejam feitas entre os dados gerados a partir de simulações e os dados obtidos experimentalmente;
- Analisar uma qualidade mais ampla de protocolos, aumentando assim a literatura voltada a este assunto;
- Desenvolver protocolos de roteamento aprimorados tomando os protocolos estudados como base.

REFERÊNCIAS

1 BESSA, A. J. G. *Uma Extensão ao Protocolo OLSR para Roteamento de Dados em Tempo Real Utilizando Múltiplos Caminhos*. 2010. Trabalho de Conclusão de Curso - Universidade Estadual do Ceará.

2 MAHLINK, N. P. *Sensor Networks and Configuration*. [S.l.]: Springer, 2007.

3 ARAUJO, C. W. B. *Protocolo de Roteamento Multi-Métrico para Redes de Sensores Sem Fio - CA-AODV*. Dissertação (Mestrado) — Universidade Federal de Campina Grande, Novembro 2009.

4 RUIZ, L. B. *MANÁ: Uma Arquitetura para Gerenciamento de Redes de Sensores Sem Fio*. Tese (Doutorado) — Universidade Federal de Minas Gerais, Dezembro 2003.

5 MÜLLER, I. *Redes de Sensores Sem Fio*. [S.l.], 2015.

6 IEEE STANDARD 802.11. *Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: High-speed physical layer in the 5 GHz band*. [S.l.], 1999.

7 BULHMAN, H. J.; CABIANCA, L. A. *Redes LAN/MAN Wireless II: Protocolo 802.11*. 2016. <http://www.teleco.com.br/tutoriais/tutorialrwlanman2/pagina_4.asp>. Acessado em 03-10-2016.

8 BULHMAN, H. J.; CABIANCA, L. A. *Redes LAN/MAN Wireless II: Protocolo 802.11*. 2016. <http://www.teleco.com.br/tutoriais/tutorialrwlanman2/pagina_2.asp>. Acessado em 04-10-2016.

9 TANENBAUM, S. A. *Computer Networks*. Hall PTR, Ney Jersey: Prentice, 1996.

10 BRIGNORI, G. V. *Estudo de Protocolos de Roteamento em Redes Ad Hoc*. 2005. Trabalho de Conclusão de Curso - Universidade Federal de Santa Catarina.

11 ADE, S. A.; TIJARE, P. A. 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, p. 545–548, Dezembro 2010.

12 FIQUEIREDO, F. L.; CASTRO, M. C. de; SIQUEIRA, M. A. de. Análise de desempenho de protocolos de rotamento para redes ad hoc sem fio. *Cad. CPqD Tecnologia*, Campinas, v. 2, p. 61–70, Julho/Dezembro 2006.

13 MARINA, M. K.; DAS, R. S. Ad hoc on-demand multipath distance vector routing. *Wireless Communications and Mobile Computing*, v. 6, p. 969–988, Novembro 2006. Disponível em (<http://onlinelibrary.wiley.com/doi/10.1002/wcm.432/pdf>). Acessado em 06-10-2016.

14 ISSARIYAKUL, T.; HOSSAIN, E. *Introduction to Network Simulator NS2*. [S.l.]: Springer Science & Business Media, 2011.

15 COUTINHO, M. M. *Network Simulator - Guia Básio para Iniciantes*. [S.l.], 2003.

16 GONÇALVES, L. C.; CORRÊA, M. E. de O. *Tutorial de NS-2*. [S.l.], 2005.

17 ROHAL, P.; DAHIYA, R.; DAHIYA, P. Study and analysus 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, p. 54–58, Março 2013. Disponível em (<http://www.ijaret.org/1.2.html>). Acessado em 11-10-2016.

18 JAIN, R. *Art of Computer Systems Performance Analysis Techniques For Experimental Design Measurements Simulation And Modeling*. [S.l.]: Wiley Computer Publishing, Jhon Wiley and Sons, Inc, 1991.