Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Análise de Protocolos de Roteamento *Unicast* em Redes *Ad Hoc* Móveis Baseada em um Modelo Realístico de Bateria

José Athayde Torres Costa Neto

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Redes de Computadores e Sistemas Distribuídos

Marco Aurélio Spohn
(Orientador)

Campina Grande, Paraíba, Brasil
©José Athayde Torres Costa Neto, 06/05/2011

Resumo

As redes *ad hoc* são caracterizadas por seus dispositivos móveis operarem sem a presença de qualquer controle centralizado, o que permite a comunicação em ambientes de difícil acesso, onde é inviável a presença de uma infraestrutura fixa. A maioria dos protocolos de roteamento *unicast* para redes *ad hoc* móveis foram avaliados via simulação. No entanto, a maior parte dos simuladores utilizam um modelo de bateria linear, que produz resultados incorretos, por não levar em consideração o efeito de recuperação da capacidade da bateria. Esse trabalho visa reavaliar três representativos protocolos de roteamento *unicast* (i.e., AODV, DSR e OLSR) contemplando o modelo realístico de bateria de Rakhmatov-Vrudhula, permitindo assim, compreender detalhes mais precisos desses protocolos. Para isso, utilizou-se o simulador de redes *Network Simulator* (NS-2). Os cenários de simulação foram configurados variando-se os padrões de tráfego e mobilidade da rede. As métricas consumo de energia, tempo de vida da rede, quantidade de nós sobreviventes, fração de entrega de pacotes, carga de roteamento normalizada e atraso médio fim a fim foram utilizadas na avaliação dos protocolos. Os resultados mostram que nas simulações baseadas no modelo de Rakhmatov-Vrudhula, além dos dispositivos da rede sobreviverem por mais tempo em todos os protocolos, por haver menor consumo de energia, a entrega de pacotes foi superior em comparação ao modelo linear.

Abstract

Mobile Ad hoc Networks (MANETs) are characterized by their devices to operate without presence of any centralized control. Most unicast routing protocols for mobile ad hoc networks were evaluated through simulation. However, most network simulators adopt a linear battery model, which produces incorrect results, because it does not take into account the battery capacity recovery effect. This work aims to re-evaluate three representative unicast routing protocols (i.e., AODV, DSR and OLSR) based on the more realistic Rakhmatov-Vrudhula battery model; thus, providing us a better insight into these protocols performance. We used the Network Simulator (NS-2). The scenarios simulation were configured by varying traffic patterns and mobility. The metrics of energy consumption, network lifetime, number of alive nodes, packet delivery fraction, normalized routing load and average delay end to end were used in the evaluation of protocols. The results show energy consumption minor and that the protocols' lifetime and packet delivery rate increase significantly when simulations are based on the Rakhmatov-Vrudhula battery model.

Agradecimentos

Agradeço, primeiramente, à Deus por estar sempre me iluminando e protegendo em todos os caminhos da minha vida.

Aos meus pais, José Ataide e Vânia Maria pelo amor incondicional e tudo que fizeram para garantir minha educação. Agradeço às minhas irmãs Olivia, Larissa e Liana pelo apoio e carinho que sempre tiveram comigo e, à todos os meus familiares que me apoiaram.

Em especial, agradeço à minha namorada Raquel Galvão, minha fonte de inspiração, que sempre esteve ao meu lado nesta conquista e superou comigo todas as dificuldades. Obrigado por tudo!

Ao meu orientador Marco Aurélio Spohn pela oportunidade que me concedeu na realização deste trabalho, pela orientação e valiosos ensinamentos dados durante a pesquisa.

Agradeço aos professores do mestrado, que proporcionaram discussões interessantes nas disciplinas, e também, aos funcionarios da COPIN pelo apoio.

Agradeço aos meus colegas do Laboratório de Telecomunicação (Latec) que contribuiram bastante com o meu trabalho: Thiago, Elmano, Rafael, Ruan.

Aos amigos que fiz em Campina Grande, nos quais tive o prazer em conhecê-los. Agradeço à todos pelos bons momentos de diversão e aprendizado.

Agradeço aos meus amigos do PoP-PI/RNP, por tudo que aprendi com eles; e aos meus amigos, colegas e professores do DIE/UFPI que também contribuiram para minha formação.

Por fim, agradeço ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro.

Conteúdo

1 Introdução **1**

1.1 Motivação 4

1.2 Objetivos 5

1.3 Relevância 6

1.4 Estrutura da Dissertação 6

2 Protocolos de Roteamento em Redes *Ad hoc* Móveis **8**

2.1 Protocolos Proativos 10

2.1.1 *Optimized Link State Routing Protocol* 10

2.2 Protocolos Reativos 13

2.2.1 *Ad Hoc On Demand Distance Vector Routing* 13

2.2.2 *Dynamic Source Routing* 16

2.3 Protocolos Híbridos 18

3 Modelos de Bateria **20**

3.1 Modelo Linear 22

3.2 Modelo de Rakhmatov-Vrudhula 22

4 Preparação e Execução dos Experimentos **26**

4.1 Ambiente de Simulação 27

4.2 Configuração do Consumo de Energia 28

4.3 Descrição dos Cenários 29

4.4 Execução dos Experimentos 32

4.5 Métricas de Desempenho 33

5 Análise dos Resultados **35**

5.1 Relação Parâmetro x Métrica x Modelo de Bateria 36

5.1.1 Consumo de Energia 36

5.1.2 Tempo de Vida da Rede 42

5.1.3 Quantidade de Nós Sobreviventes 45

5.1.4 Fração de Entrega de Pacotes 46

5.1.5 Carga de Roteamento Normalizada 50

5.1.6 Atraso Médio Fim a Fim 52

5.2 Relação Entre Métricas 54

6 Conclusões e Trabalhos Futuros **58**

6.1 Trabalhos Futuros 60

A *Script Tcl* **67**

B *Script Awk* **72**

Lista de Símbolos

AM - *Atraso Médio Fim a Fim*

AODV - *Ad hoc On Demand Distance Vector*

C - *Capacidade*

CBR - *Constant Bit Rate*

CE - *Consumo de Energia*

CRN - *Carga de Roteamento Normalizada*

CTS - *Clear To Send*

DSR - *Dynamic Source Routing*

FEP - *Fração de Entrega de Pacotes*

FIFO - *First-In Fisrt-Out*

IC - *Intervalo de Confiança*

MANETs - *Mobile Ad Hoc Networks*

MAC - *Media Access Control*

NS-2 - *Network Simulator 2*

MPR - *Multipoint Relays*

NAM - *Network Animator*

OLSR - *Optimized Link State Routing Protocol*

OTCL - *Object-oriented Tool Command Language*

QoS - *Quality of Service*

RSSFs - *Redes de Sensores Sem Fio*

RREP - *Route Reply*

RREQ - *Route Request*

RERR - *Route Error*

RTS - *Request To Send*

TC - *Topology Control*

TVR - *Tempo de Vida da Rede*

ZRP - *Zone Routing Protocol*

Lista de Figuras

1.1 Rede Sem Fio *Ad Hoc* e Infraestruturada 1

2.1 OLSR sem e com o uso do MPR 12
2.2 Descoberta de Rota no protocolo AODV 15
2.3 Descoberta de Rota no protocolo DSR 17

3.1 Estados de operação da bateria 21
3.2 Capacidade da Bateria em Função do Perfil de Descarga 23
3.3 Modelo Linear (a) versus Modelo de Rakhmatov-Vrudhula (b) para um perfil de descarga de 91 segundos 24

4.1 Movimento de um nó no modelo de mobilidade *Random Waypoint* 29
4.2 Trechos do padrão de movimento de um nó da rede 30

5.1 Consumo médio de energia dos nós, variando-se o tempo de pausa, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula 36
5.2 Consumo médio de energia dos nós, variando-se o número de nós fontes, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula 38
5.3 Consumo médio de energia dos nós quando o primeiro nó da rede tem sua bateria esgotada, variando-se o tempo de pausa, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula 40
5.4 Porcentagem de consumo de energia, segundo (a) estado de operação (TX, RX e Idle) e (b) tipo de pacote (CBR, MAC e Roteamento) 41
5.5 Tempo de vida da rede, variando-se o tempo de pausa, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula 42

5.6 Tempo de vida da rede, variando-se o número de nós fontes, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula 43
5.7 Tempo de vida da rede (quando 20% dos nós morrem), variando-se o número de nós fontes, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula 44
5.8 Tempo de vida da rede (quando 50% dos nós morrem), variando-se o número de nós fontes, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula 45
5.9 Quantidade de nós sobreviventes, variando-se o número de nós fontes, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula 46
5.10 Fração de entrega de pacotes, variando-se o tempo de pausa, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula 47
5.11 Fração de entrega de pacotes, variando-se o número de nós fontes, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula 47
5.12 Carga de roteamento normalizada, variando-se o tempo de pausa, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula 51
5.13 Carga de roteamento normalizada, variando-se o número de nós fontes, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula 51
5.14 Atraso médio fim a fim, variando-se o tempo de pausa, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula 53
5.15 Atraso médio fim a fim, variando-se o número de nós fontes, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula 53

Lista de Tabelas

4.1 Consumo de energia da interface *Wireless WaveLan* 11 Mbps da *Lucent*. . . 29

4.2 Parâmetros dos experimentos. 31

5.1 Consumo de energia, variando-se o tempo de pausa 37

5.2 Consumo de energia, variando-se o número de nós fontes 39

5.3 Fração de entrega de pacotes, variando-se o número de nós fontes 48

5.4 Média do total de pacotes enviados na simulação, variando-se o número de nós fontes 49

5.5 Matriz de correlação entre métricas para o modelo linear de bateria 55

5.6 Matriz de correlação entre métricas para o modelo de Rakhmatov-Vrudhula 56

6.1 Protocolos que obtiveram melhor desempenho em cada métrica. 60

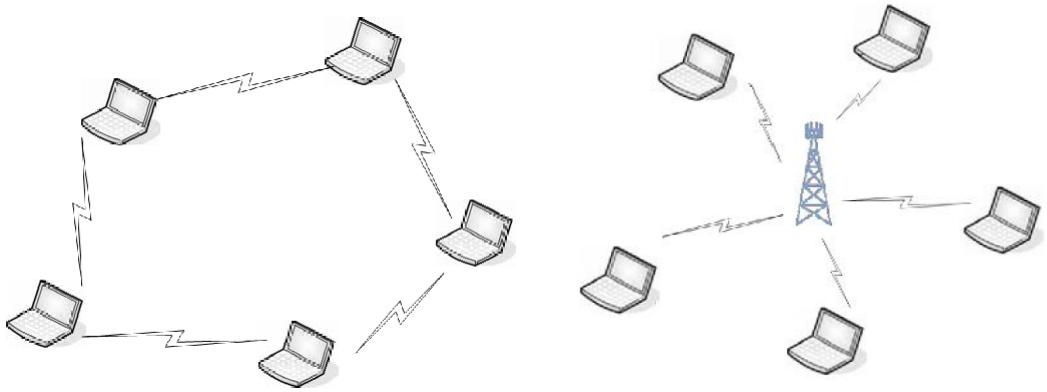
Lista de Códigos Fonte

A.1 Script Tcl 67
B.1 Script Awk 72

Capítulo 1

Introdução

Nos últimos anos, as redes sem fio destacaram-se pelo importante papel no campo tecnológico e pela sua notável expansão. A necessidade das pessoas de estarem conectadas entre si, o aumento considerável do número de computadores e de outros dispositivos móveis, bem como a consolidação de padrões de tecnologia sem fio (e.g., IEEE 802.11) contribuíram certamente para este crescimento. Comparadas às redes cabeadas, as redes sem fio apresentam baixo custo de instalação e maior flexibilidade, por isso são cada vez mais utilizadas.



(a) *Ad Hoc* (b) Infraestruturada

Figura 1.1: Rede Sem Fio *Ad Hoc* e Infraestruturada

As redes sem fio são classificadas em duas categorias: infraestruturada e *ad hoc* (Figura 1.1) [47]. As redes infraestruturadas (e.g., redes de telefonia celular) constituem-se de um parte fixa (i.e., ponto de acesso ou estação base) que é responsável por toda a comunicação da rede, enquanto que nas redes *ad hoc*, os nós móveis se comunicam diretamente, sem a

necessidade de um ponto de acesso fixo.

As redes *ad hoc* móveis (em inglês, *Mobile Ad hoc NETwork* - MANETs) foram desenvolvidas nos Estados Unidos, na década de 70, através de um projeto da Agência de Pesquisa em Projetos Avançados (DARPA). Essas redes são compostas por dispositivos móveis conectados ponto a ponto por alguma tecnologia de comunicação sem fio (e.g., radiofrequência), operando sem a presença de qualquer controle centralizado ou infraestrutura fixa definida. A utilização dessas redes vêm crescendo, principalmente pela simplicidade de implementação, baixo custo, e por permitir a comunicação em ambientes de difícil acesso, onde é inviável a presença de uma infraestrutura fixa [47].

O termo "*ad hoc*" deriva do latim e significa "para este propósito". Assim, uma rede *ad hoc* representa uma idéia de rede com propósito definido e de carácter temporário, ou seja, alguns dispositivos da rede podem fazer parte desta apenas enquanto estão em alcance do restante da rede ou durante o tempo da sessão de comunicação. Um fator relevante para pesquisas em redes *ad hoc* é a diversidade em aplicações, como exemplos:

- Envio de informações em campos de batalha (militar);
- Envio de informações em áreas de resgate;
- Redes veiculares em ambientes urbanos;
- Redes de sensores em ambientes industriais;
- Redes em ambientes corporativos e residenciais.

As MANETs envolvem geralmente uma heterogeneidade de dispositivos móveis (e.g., celulares, *palms*, *laptops*, sensores), cada um funcionando como um roteador, podendo encaminhar informações para outros nós, mesmo estes não estando ao seu alcance direto. Para isso, cada nó executa algum protocolo de roteamento que permite descobrir caminhos *multihop* para qualquer outro nó da rede, isto é, permite descobrir caminhos a vários saltos de distância do nó origem até o destino. Porém, a presença de mobilidade em uma rede *ad hoc* implica que conexões sejam estabelecidas e desfeitas de forma não determinística, resultando em mudança dinâmica da topologia. Uma rede *ad hoc* deve também ser auto-organizável, determinando automaticamente seus parâmetros de configuração (e.g., endereçamento, roteamento e controle de potência) [14].

O roteamento em redes *ad hoc* é sem dúvida um problema complexo que precisa ser estudado e avaliado, no sentido de determinar um comportamento ideal para esse tipo de rede. Assim, características essenciais como simplicidade, robustez, escalabilidade, adaptabilidade, Qualidade de Serviço (*Quality of Service* - QoS), segurança, limitação de largura de banda e consumo de energia, devem ser consideradas na projeção dos protocolos de roteamento.

Dispositivos móveis são equipados com baterias capazes de fornecer energia para seu funcionamento. Constituindo uma rede *ad hoc*, a partir desses dispositivos, torna-se essencial que esta funcione o máximo de tempo possível. Assim, a falha de um dispositivo que seja peça chave de uma rede, poderá comprometer todo o funcionamento desta e interromper o transporte de informações. Nesse caso, a bateria é um recurso que precisa ser considerado, pois seu consumo influencia diretamente no tempo de vida da rede. Assim, técnicas para minimizar o consumo de energia e aumentar a vida útil da rede são importantes e devem ser introduzidas nos protocolos de roteamento.

Uma classe particular de redes *ad hoc* são as Redes de Sensores Sem Fio (RSSFs) [48] que, comparadas às redes móveis sem fio em geral, apresentam certas limitações como, por exemplo, baixo alcance de comunicação, baixo poder de processamento e baixa capacidade de bateria. Nesse caso, esse tipo de redes requer ainda mais atenção quanto ao consumo de energia e tempo de vida da rede, devido a baixa capacidade de bateria dos sensores.

Pesquisas referentes à avaliação de desempenho de protocolos de redes móveis *ad hoc* seguem basicamente dois métodos [2]: experimentação e simulação. O método de experimentação trata da realização e monitoração de experimentos em sistemas reais (*testbeds*) e, por isso, é a que proporciona maior fidelidade dos dados obtidos. Contudo, podem acarretar elevado custo e tempo de análise [41].

O método de simulação trata da construção de um modelo que simule o funcionamento do sistema a ser avaliado [41]. Em trabalhos referentes à avaliação de protocolos, esta abordagem é, sem dúvida, a mais utilizada, devido à riqueza de detalhes que ela explora e à simplicidade na construção de cenários mais complexos. Todavia, o tempo necessário para realizá-la cresce em proporção direta a sua complexidade.

Uma das principais técnicas de avaliação de desempenho via simulação baseia-se na utilização de eventos discretos. Uma ferramenta bastante conhecida na comunidade acadêmica,

e que se enquadra nessa técnica, é o simulador de rede *Network Simulator* (NS-2) [21]. Outros simuladores de rede também muito usados na avaliação de desempenho de protocolos de redes são o *GloMoSim* [35] e o *Qualnet* [49].

1.1 Motivação

A maioria dos protocolos de roteamento de redes *ad hoc* móveis propostos até o momento foi avaliada utilizando algum tipo de simulador de redes. No entanto, estes podem produzir resultados incorretos, caso os modelos utilizados não forem suficientemente acurados. Margi e Obraczka [31] afirmam que os principais simuladores (e.g., NS-2, *GloMoSim*, *Qualnet*) utilizados na avaliação de desempenho de protocolos, apresentam modelos de consumo de energia imprecisos, pois não consideram todos os estados do rádio, os diferentes níveis de energia que eles consomem, e os efeitos não lineares da bateria. Desse modo, quanto mais realísticos forem os modelos adotados nos simuladores, melhor será a análise baseada nos resultados.

O problema evidenciado neste trabalho relaciona-se ao modelo convencional de descarga de bateria, utilizado como padrão em grande parte desses simuladores de rede. No modelo linear, a bateria é tratada como um fornecedor de corrente, em que sua capacidade é consumida de forma linear, após cada operação do nó, e não são considerados efeitos reais existentes em uma bateria. Portanto, o seu uso pode ocasionar interpretações equivocadas do consumo de energia dos dispositivos e na avaliação de protocolos, especialmente quando se leva em consideração o tempo de vida da rede.

A literatura apresenta uma variedade de trabalhos relacionados à avaliação de protocolos de roteamento de redes *ad hoc* móveis. Contudo, poucos destacam o modelo de bateria utilizado nas simulações. Trabalhos clássicos [38][10][3][4] apresentam uma avaliação de desempenho, via simulação, dos principais protocolos de roteamento *unicast* em redes *ad hoc*. Tais avaliações foram realizadas com protocolos proativos e reativos (ver Capítulo 2), sob cenários com diferentes padrões de tráfego e mobilidade, considerando-se métricas como fração de entrega de pacotes, atraso fim-a-fim e sobrecarga de roteamento. Na maior parte dos cenários, os resultados mostram menor sobrecarga para os protocolos reativos, porém, os proativos obtiveram melhor desempenho no atraso e entrega de pacotes [38][10]. Nesses

artigos, os autores não consideram o modelo de bateria e o consumo de energia dos nós.

Heidemann *et al.* [17] abordam que é preciso dosar a quantidade certa de detalhes em uma simulação de redes sem fio, para garantir resultados satisfatórios. Reforçam ainda que é preciso realizar a escolha apropriada do modelo de bateria a ser utilizado e que vários fatores como entrega e recebimento de pacotes, retransmissão de CTS/RTS na camada MAC (*Media Access Control*), nós em estado de escuta, propriedades internas da bateria (i.e., não-linearidade, memória, temperatura) e uso de outros sistemas (e.g., *display*, CPU, disco), podem incorrer no consumo de energia dos nós de uma rede móvel.

Em virtude desses problemas, surge a necessidade da utilização de um modelo realístico de bateria na avaliação de protocolos de roteamento de redes *ad hoc* móveis. Com isso, poderá ser possível compreender detalhes até então não observados de cada um dos protocolos avaliados, devido ao uso do modelo linear de bateria na maioria dos simuladores de rede, além de uma melhor representação dos resultados. Rakhmatov e Vrudhula [43] apresentam um modelo de bateria analítico, que considera os efeitos de taxa de capacidade e de relaxação, possibilitando a recuperação da capacidade da bateria quando submetida a baixas correntes. Estes efeitos estão presentes em diferentes tipos de baterias (e.g., alcalinas, íons de lítio).

Trabalhos com simulações envolvendo redes *ad hoc* naturalmente apresentam um grande custo de recursos computacionais, pois geralmente estão envolvidos diversos dispositivos movimentando-se e enviando pacotes a todo instante. Por outro lado, quando essas simulações envolvem modelos de bateria mais realísticos (i.e., modelo de Rakhmatov-Vrudhula[43]), esse esforço computacional aumenta consideravelmente. Por isso, muitos trabalhos desconsideram a utilização de modelos de bateria realistas na avaliação de protocolos.

1.2 Objetivos

Este trabalho objetiva reavaliar e comparar o desempenho de três protocolos de roteamento *unicast* de redes *ad hoc* móveis - *Ad Hoc On Demand Distance Vector Routing* (AODV) [39], *Dynamic Source Routing* (DSR) [26] e *Optimized Link State Routing Protocol* (OLSR) [23] - utilizando um modelo realístico de descarga de bateria (i.e., modelo de Rakhmatov-

Vrudhula). As simulações foram realizadas por meio do simulador de redes NS-2, ferramenta consolidada pela comunidade científica. A partir dessa avaliação, será possível compreender detalhes até então não observados de cada um dos protocolos, devido ao uso do modelo linear de bateria na maioria dos simuladores de rede, além de uma melhor representação dos resultados.

Pretende-se atingir os seguintes objetivos específicos:

- Estender o NS-2 para trabalhar com o modelo realista de bateria de Rakhmatov-Vrudhula.
- Analisar o desempenho dos protocolos de roteamento *unicast* quando submetidos aos modelos de bateria realista e linear.
- Comparar os resultados baseados no modelo não linear com os resultados baseados no modelo linear.

1.3 Relevância

Este trabalho é relevante porque permite uma nova avaliação de protocolos de roteamento *unicast* de redes *ad hoc* móveis sob uma visão mais realista dos resultados, quando comparados aos do modelo linear de bateria. Tais resultados (relacionados ao consumo de energia dos nós, tempo de vida da rede, entrega de pacotes, entre outros) poderão auxiliar pesquisadores no desenvolvimento e aperfeiçoamento de protocolos com atenção para o consumo de energia dos dispositivos.

1.4 Estrutura da Dissertação

Esta dissertação está estruturada em seis capítulos que abrangem conteúdos da seguinte forma: o Capítulo 2 apresenta uma revisão bibliográfica dos protocolos de roteamento de redes *ad hoc* utilizados nesta pesquisa; o Capítulo 3 revisa os modelos de bateria linear e de Rakhmatov-Vrudhula e as diferenças de comportamento entre eles. O Capítulo 4 enfoca a metodologia utilizada para preparação e execução dos experimentos. No Capítulo 5, os resultados dos experimentos são analisados, a fim de avaliar o desempenho dos protocolos

de roteamento. Por fim, no Capítulo 6, são apontadas as considerações finais deste trabalho, bem como sugestões de trabalhos futuros.

Capítulo 2

Protocolos de Roteamento em Redes *Ad hoc* Móveis

Para que os nós de uma rede *ad hoc* móvel se conectem de forma dinâmica e arbitrária, eles devem se comportar como roteadores e assumirem o trabalho de descoberta e manutenção das rotas para os demais nós da rede. Esse trabalho é realizado pelo protocolo de roteamento, que tem como objetivo planejar um esquema para transferir os pacotes de um nó para outro. Esse esquema pode ser planejado com base em diversos critérios, como quantidade de saltos, latência, largura de banda, energia de transmissão, entre outros [18]. Como a topologia das redes *ad hoc* é dinâmica, os protocolos de roteamento tradicionais, como *Routing Information Protocol* (RIP) [30] e *Open Shortest Path First* (OSPF) [32], apresentam problemas para convergir nestas redes, pois são adequados para redes cabeadas, em que a topologia geralmente permanece estática.

Considerando a complexidade do roteamento em redes *ad hoc*, determinadas características são essenciais nestes protocolos, tais como [47]:

- **Simplicidade**: o protocolo deve oferecer os serviços com a mínima quantidade de processamento;
- **Robustez**: o protocolo deve funcionar sem falhas o máximo de tempo possível;
- **Escalabilidade**: a rede precisa se adaptar ao aumento considerável, mas suportável, do número de nós;

- **Adaptabilidade**: o protocolo deve ser capaz de trabalhar com mudanças frequentes na topologia da rede;
- **Qualidade de Serviço** (*Quality of Service* - QoS): dependendo do tipo de tráfego, o protocolo deve dar suporte à QoS, garantindo perda de pacotes e atrasos mínimos;
- **Segurança**: o protocolo deve prover segurança dos dados trafegados.
- **Heterogeneidade**: o protocolo deve trabalhar considerando as diferenças entre os dispositivos.

Alguns desafios a serem superados na projeção dos protocolos de roteamento em MANETs [9] são listados abaixo:

- **Mobilidade**: A rede precisa se adaptar rapidamente às mudanças na sua topologia, devido à mobilidade dos nós, que tornam obsoletas as informações de localização, além de provocarem altas taxas de erros;
- **Recursos limitados**: A bateria dos dispositivos é um recurso limitado, que precisa ser melhor gerenciado pelos protocolos de roteamento, a fim de conter o gasto excessivo de energia. A largura de banda também pode apresentar limitações.

O roteamento de pacotes em redes *ad hoc* pode ocorrer nas seguintes modalidades: *unicast*, *multicast* e *broadcast* [47]. No roteamento *multicast*, a entrega é feita para múltiplos destinatários simultaneamente, enquanto que no roteamento *broadcast*, a entrega ocorre para todos os pontos de uma rede.

Este trabalho aborda o roteamento *unicast*, no qual a entrega dos dados ocorre de maneira simples, ponto-a-ponto, sendo ideal para redes em que as topologias mudam constantemente e cujos recursos são limitados. Em geral, os protocolos de roteamento *unicast* são classificados em: proativos ou orientado por tabela (*table-driven*); reativos ou sob demanda (*on-demand*) [46][1]; e híbridos, que reunem características das duas categorias.

No presente trabalho, foram avaliados os desempenhos de três protocolos *unicast*, utilizando-se um modelo realístico de bateria. Estes protocolos foram escolhidos com base nos seguintes critérios:

- Representatividade do protocolo na comunidade científica;

- Quantidade considerável de artigos relacionados;
- Bom desempenho quando comparados com outros protocolos;
- Tenha implementação para a versão mais recente do simulador de redes *Network Simulator* (NS-2, versão 2.34).

Optou-se por escolher protocolos proativos e reativos, bem como uma quantidade que não sobrecarregasse tanto a avaliação dos mesmos. Conforme tais critérios, foram escolhidos os protocolos de abordagem reativa, *Ad Hoc On Demand Distance Vector Routing* (AODV) [39] e *Dynamic Source Routing* (DSR) [26]), e o protocolo de abordagem proativa, *Optimized Link State Routing Protocol* (OLSR) [23], que serão descritos a seguir.

2.1 Protocolos Proativos

O roteamento proativo ou orientado por tabela (*table-driven*), caracteriza-se por realizar continuamente a determinação de rotas, independente da necessidade da rota e do tráfego existente na rede [46]. Utilizam-se tabelas de roteamento para armazenar e manter as rotas atualizadas. A principal vantagem desta classe de protocolos é a disponibilidade imediata de rotas, reduzindo assim o atraso inicial de transmissão dos pacotes. Também apresenta vantagens nas situações em que existem muitos pares comunicantes. Por outro lado, a mobilidade pode provocar intervalos elevados de perda de conectividade, durante a descoberta de uma nova rota válida. Além disso, estes protocolos tendem a gerar uma carga elevada de mensagens de roteamento, fator crítico em um ambiente com restrições de largura de banda. São exemplos de protocolos de roteamento proativos: OLSR [23], DSDV [40], WRP [33], CGSR [6].

2.1.1 *Optimized Link State Routing Protocol*

O *Optimized Link State Routing Protocol* (OLSR) [23] é um protocolo *table-driven*, que armazena informações de topologia e roteamento da rede em tabelas, atualizadas através de mensagens pré-definidas enviadas periodicamente. Trata-se de uma otimização dos protocolos de estado de enlace (*link-state*) puros, adotando um mecanismo de redução de trans-

missões redundantes em operações que requerem o *broadcasting* de mensagens de controle, diminuindo muito a utilização de banda pelo protocolo.

Analisando-se uma rede sem o uso do protocolo OLSR, quando um dispositivo deseja enviar uma mensagem, este transmite, via *broadcast*, pacotes de controle aos seus nós vizinhos, os quais enviam para os próximos vizinhos, até que todos os nós recebam os pacotes, ocorrendo uma "inundação" (*flooding*) de informações. Porém, cada um deles receberá o mesmo pacote várias vezes de diferentes vizinhos, gerando portanto um *overhead*; ou seja, a rede funcionará sobrecarregada de informações redundantes.

Diferentemente, com o protocolo OLSR em funcionamento, a mensagens *broadcast* serão propagadas, porém somente alguns nós selecionados irão retransmití-las aos vizinhos, o que torna o processo mais eficiente, e diminui a sobrecarga de informações e a utilização de banda. Esses nós são denominados de *Multipoint Relays* (MPR) e escolhidos com o consentimento dos seus vizinhos, de tal modo que, através destes, o nó de origem consiga alcançar outro nó da rede a dois saltos de distância.

A principal função dos nós MPR é diminuir a quantidade de mensagens *broadcast* repetidas na rede, uma vez que somente eles podem retransmitir pacotes. Além disso, estes dispositivos são capazes de capturar informações para o cálculo de rotas mais curtas e mais rápidas até um destino. Todos os nós da rede mantêm uma lista com os MPRs designados para retransmitir seus pacotes enviados. As informação contidas nesta lista são anunciadas periodicamente na rede, o que ajuda os dispositivos a identificarem a topologia da rede. Com o objetivo de evitar que mensagens duplicadas sejam processadas por um mesmo nó, uma outra lista também é armazenada, contendo informações das mensagens recebidas pelo nó [18].

Quando uma mensagem precisa ser atualizada na rede, esta é enviada para os nós vizinhos, e quando recebida por um MPR, é retransmitida por ele para os próximos vizinhos até atingir toda rede. Neste processo, cada nó receberá a informação apenas uma vez, evitando-se o *overhead*. A Figura 2.1, mostra na primeira imagem como ocorre o *broadcast* via *flooding* sem o uso do MPR, e na segunda imagem, com o uso de MPR.

Com base neste esquema, no *flooding* sem o uso de MPR, todos os nós intermediários (de cor azul) retransmitem pacotes, fazendo com que recebam mensagem duplicadas. Na segunda imagem, quando os nós MPR (de cor azul) são utilizados, apenas estes podem

retransmitir pacotes, evitando que um dispositivo receba a mesma informação mais de uma vez. Esse mecanismo não é viável em redes com muitos nós, porque se torna mais custoso difundir informação sobre a topologia da rede, à medida em que se aumenta o número de nós, principalmente em cenários de grande mobilidade.

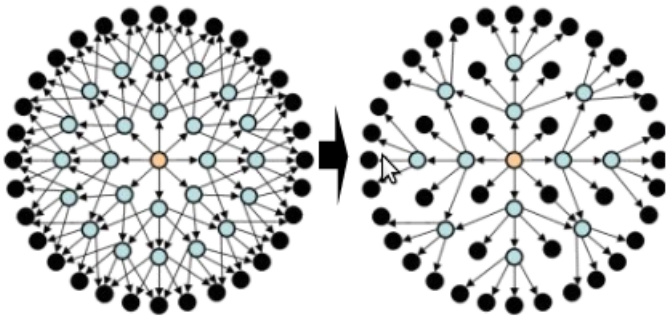


Figura 2.1: OLSR sem e com o uso do MPR [1].

Mensagens de Controle

Cada mensagem OLSR enviada, carrega em seu cabeçalho as seguintes informações: tipo de mensagem (e.g., *HELLO*, *Topology Control* - TC), tamanho da mensagem em bytes, endereço da origem, TTL (i.e., número máximo de saltos que a mensagem pode percorrer, sendo decrementado em 1 a cada salto), contador de saltos (i.e., quantidade de saltos percorrido pela mensagem) e número de sequência da mensagem (i.e., identificador único recebido a cada envio de mensagem) [7].

Basicamente, no OLSR são trocados dois tipos de mensagens de controle: *HELLO* e *Topology Control* (TC). As mensagens *HELLO* são trocadas periodicamente, a fim de descobrirem os nós vizinhos a dois saltos e também selecionar os nós que funcionarão como MPR. Em contrapartida, as mensagens TC são trocados periodicamente em todos os nós, para obterem informações da topologia da rede e do conjunto de nós MPR [19].

Todas as informações obtidas pela troca dessas mensagens de controle são armazenadas para posterior computação da tabela de roteamento. Assim, para poder encaminhar mensagens a qualquer nó da rede, cada dispositivo mantém atualizada sua tabela de roteamento,

[1]Disponível em `http://wiki.uni.lu/secan-lab/graphics/olsr02.gif`. Acesso em abril de 2010.

que contém os parâmetros a seguir: endereço do destino, quantidade de saltos até o destino e endereço do próximo nó por onde a rota deve passar.

A troca periódica de mensagens de controle no OLSR, pode evitar, por exemplo, a perda de pacotes provocadas por problemas de transmissão ou colisão. Cada mensagem de controle carrega consigo um número de sequência que é incrementado cada vez que é enviada, o que permite identificar facilmente se a informação é mais recente ou não.

2.2 Protocolos Reativos

O roteamento reativo caracteriza-se por atuar sob demanda, criando e mantendo rotas somente quando requisitados por um determinado nó [1]. Apresenta como vantagem a otimização do uso de recursos da rede, evitando propagações desnecessárias de informações de roteamento. No entanto, estes protocolos impõem certo retardo inicial no roteamento de pacotes para um destino, cuja rota não esteja disponível e deva ser computada. Apesar disso, a mobilidade dos nós provoca apenas pequenos intervalos de perda de conectividade, pois quando uma rota fica inválida, a abordagem sob demanda ativa rapidamente a criação de uma nova rota válida. São exemplos de protocolos de roteamento reativos: AODV [39], DSR [26], TORA [36].

2.2.1 *Ad Hoc On Demand Distance Vector Routing*

O *Ad Hoc On Demand Distance Vector Routing* (AODV) [39] é um protocolo de roteamento reativo que utiliza, para cada nó, uma tabela de roteamento tradicional, em que há somente uma entrada para cada destino, referente ao menor caminho. Com isso, o nó registra o próximo salto para se chegar ao destino e a distância total em número de saltos. Cada dispositivo mantém em sua tabela de roteamento as seguintes informações: destino, próximo salto, número de saltos, número de sequência do destino, nós vizinhos ativos para a rota, e tempo de duração para a rota. Um número de sequência crescente é usado com o objetivo de evitar laços e verificar se uma informação de rota é recente ou não [20].

Para que a tabela de roteamento dos dispositivos esteja sempre atualizada, o AODV dispõe de dois mecanismos, um de descoberta e outro de manutenção das rotas, descritos a seguir.

Descoberta de Rotas

Quando um nó (fonte) precisa se comunicar com outro nó (destino), e o caminho não está disponível na tabela de roteamento, inicia-se um processo de descoberta de rota. Este mecanismo consiste no envio de mensagens *Route Request* (RREQ) via *broadcast* para todos os nós vizinhos, sobre a necessidade de uma rota até o destino desejado [46]. Cada pacote RREQ contém os campos subsequentes: identificador do *broadcast*, endereço da fonte, endereço do destino, número de sequência da fonte, número de sequência do destino e contador de saltos.

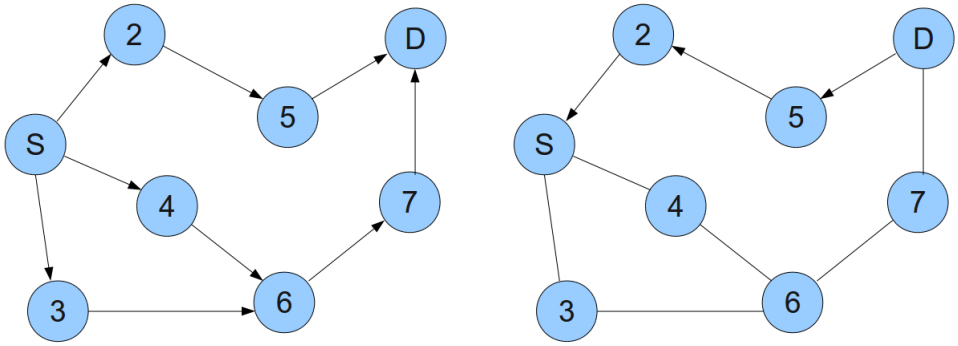
Cada vez que o nó fonte envia uma nova mensagem RREQ o identificador do *broadcast* é incrementado. Logo, se um nó intermediário entre a fonte e o destino receber um RREQ e posteriormente receber outro com o mesmo identificador e endereço da fonte, o último RREQ será descartado. Enquanto a rota para o destino não for encontrada, os pacotes RREQ são propagados novamente para os próximos vizinhos, sempre incrementando o contador de saltos e guardando entradas temporárias em suas tabelas, que contêm as seguintes informações: endereço da fonte e do destino, identificador do *broadcast*, número de sequência da fonte e tempo limite da rota com o caminho de volta para fonte (i.e., rota reversa). O processo continua até que o nó destino seja encontrado ou que algum nó saiba a rota para tal. O número de sequência da fonte é usado para verificar o quão recente são as informações da rota reversa, e o número de sequência do destino mantém mais recentes as informações da rota do nó fonte para o destino.

Caso o pacote RREQ encontre um nó intermediário que possua uma rota até o destino, ele inicialmente verificará se esta rota é mais atual, a partir da comparação do número de sequência de destino carregado pelo RREQ, com o número de sequência de destino existente no nó. Se o número de sequência de destino do RREQ for maior, então a rota contida no nó intermediário não será usada e o RREQ é retransmitido para os vizinhos. Caso contrário, um pacote *Route Reply* (RREP) via *unicast* é enviado de volta para a fonte, seguindo a rota reversa. O mesmo processo é realizado quando o próprio nó destino é encontrado.

O pacote RREP contém informações como: endereço da fonte e do destino, número de sequência do destino, contador de saltos e o tempo de duração da rota. Cada nó por onde o RREP passa, guarda em sua tabela de roteamento uma entrada com o próximo salto e a quantidade destes até o destino, e também atualiza informações do tempo de duração da rota

e o último número de sequência para o destino. Apenas uma entrada para cada destino é registrada [46].

A Figura 2.2(a) mostra o caminho percorrido pelo pacote RREQ no processo de descobrimento de rota do AODV, partindo da origem S para o destino D. Da mesma forma, a Figura 2.2(b) mostra o caminho percorrido pelo pacote RREP, depois que o destino D é alcançado pelo pacote de requisição de rota. Neste caso, mais de uma rota foi identificada pelo destino D, porém apenas a mais curta foi enviada para ser utilizada pela origem S. Na tabela de roteamento do nó S será registrada a rota para D, sendo o nó 2 o próximo salto.



(a) *Route Request* (b) *Route Reply*

Figura 2.2: Descoberta de Rota no protocolo AODV

Manutenção de Rotas

Quebras de enlaces ocorrem constantemente, devido à mobilidade dos nós. Isto faz com que rotas presentes nas tabelas de roteamento fiquem inválidas. Para a solução de tal problema, o protocolo AODV dispõe de um mecanismo de manutenção das rotas, que verifica a validade dessas rotas [37].

Para assegurar a conectividade local de um nó e detectar falhas de conexão, o mecanismo de manutenção de rotas utiliza mensagens *HELLO* que são enviadas periodicamente para seus vizinhos via *broadcast*. Se um nó não recebe esta mensagem do vizinho, considera-se uma quebra de enlace, invalidando a rota. As mensagens *HELLO* contêm um identificador e o número de sequência da fonte. A informação de queda de enlace é enviada, através de um pacote sinalizador de erro, *Route Error* (RERR), de volta para o nó de origem e para todos os nós que usam este enlace em alguma rota recente de sua tabela. Então, essas rotas são

invalidadas e removidas da tabela conforme seus números de sequência.

Apesar de a manutenção periódica das rotas ativas ser uma vantagem para o protocolo AODV, o envio de mensagens *HELLO* aumenta consideravelmente a carga na rede.

2.2.2 *Dynamic Source Routing*

O *Dynamic Source Routing* (DSR) [26] é um protocolo de roteamento sob demanda do tipo *source routing*, ou seja, o nó fonte é quem determina e constrói toda a sequência de nós em que passará o pacote até chegar ao destino. Cada nó armazena todas as rotas de que necessita em sua memória (*cache*), e sempre que descobre novas rotas, a *cache* é atualizada. Diferente do protocolo AODV, que armazena apenas a rota mais recente em sua tabela, com o próximo salto pelo qual o pacote deverá passar, o DSR pode armazenar múltiplas rotas para um só destino. Nesse caso, a rota mais curta terá prioridade sobre as demais.

No DSR, quando um nó fonte precisa enviar um pacote de dados, este primeiramente verifica em seu *cache* se possui uma rota para o destino. Caso possua, o nó fonte envia o pacote para o primeiro nó identificado na rota. Se este nó não for o destino, o pacote é enviado para o próximo salto identificado no cabeçalho do pacote, e tal processo continua até que o pacote chegue ao destino. Caso o nó fonte não possua a rota para o destino, inicia-se o mecanismo de descoberta de rotas, também utilizado no AODV e descrito a seguir.

Descoberta de Rotas

O nó fonte inicia o mecanismo de descoberta de rota com o envio de mensagens *Route Request* (RREQ), via *broadcast*, para seus vizinhos. Cada pacote carrega as seguintes informações: endereço da origem e do destino, identificador determinado pela origem e uma lista de endereços por onde o pacote tenha passado até então. Para que as requisições de rotas duplicadas sejam identificadas, cada nó mantém uma lista com o endereço da origem e o identificador do RREQ recebido [46].

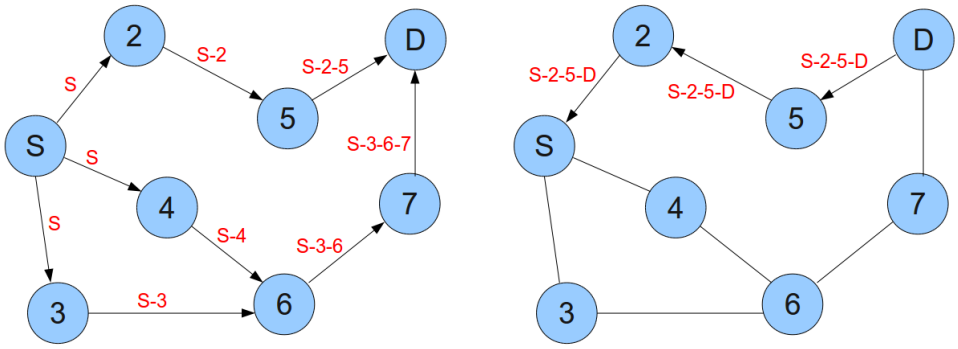
Ao receber o RREQ, o nó intermediário verifica se ele é o destino, e neste caso, envia para a origem um pacote *Route Reply* (RREP), contendo a lista da sequência de todos os nós acumulados pelo RREQ até o destino. O RREP é enviado via *unicast* seguindo a rota reversa do RREQ. Se não for o destino, o nó confere se esta requisição de rota foi recebida

anteriormente e, se verdadeiro, o pacote é descartado. Se não, o nó verifica se possui em sua *cache* uma rota para o destino. Caso exista, um pacote RREP é enviado até a origem, contendo o caminho de todos os nós até o destino acumulado pelo RREQ.

Por último, se o nó intermediário não conhecer a rota para o destino, então este envia novos pacotes RREQ para seus próximos vizinhos, sempre inserindo seu endereço no registro de rota do pacote. Assim, esse processo continua até que esta requisição de rota seja recebida pelo nó destino ou por um nó intermediário que conheça a rota.

O nó destino pode receber mais de uma requisição de rota com o mesmo identificador do RREQ e endereço de origem. Nestas condições, o nó destino responderá todos os RREQ, enviando os pacotes RREP à fonte. Isso assegura rotas alternativas da fonte para o destino, caso alguma delas falhe. Uma vez que o nó fonte possui uma rota para o destino, os pacotes de dados são enviados seguindo o caminho especificado.

A Figura 2.3(a) mostra o caminho percorrido pelo pacote RREQ no processo de descobrimento de rota do DSR, partindo da origem S para o destino D. O pacote adiciona no cabeçalho os endereços dos nós por onde passa, até chegar ao destino. Já a Figura 2.3(b) mostra o caminho percorrido pelo pacote RREP, depois que o destino D foi alcançado pelo RREQ. As duas rotas identificadas, $S-2-5-D$ e $S-3-6-7-D$, são enviadas à origem, porém a mais curta terá prioridade. Então, o nó S guarda em seu *cache* as rotas descobertas.



(a) *Route Request* (b) *Route Reply*

Figura 2.3: Descoberta de Rota no protocolo DSR

Manutenção de Rotas

Por ser um protocolo reativo, a alta mobilidade dos dispositivos faz com que as rotas armazenadas na *cache* se tornem inválidas mais rapidamente. Assim, da mesma forma que o AODV, o DSR também dispõe de um mecanismo de manutenção de rotas para prover o conserto das rotas com quebras de enlace ou problemas de transmissão.

O DSR não utiliza mensagens *HELLO* para reconhecimento da vizinhança, como utilizadas no AODV, o que reduz a largura de banda disponível e a energia das baterias dos dispositivos. Para monitorar a conectividade dos nós, o protocolo utiliza um procedimento que monitora a operação da rota, de forma que cada nó fica responsável por verificar o recebimento de pacotes do vizinho. Essa verificação é feita através das confirmações recebidas pela camada de enlace. Caso o nó seguinte não envie a confirmação de recebimento de pacote, é identificada queda de enlace. Então, um pacote *unicast Route Error* (RERR) é enviado ao nó fonte pelo caminho reverso, contendo o endereço do nó que identificou o erro e do nó que receberia o pacote. Ao receber o pacote de erro, o nó remove de sua *cache* todas as rotas que contêm o endereço para o enlace defeituoso [46].

Uma vantagem do DSR é que este consegue operar em modo promíscuo, que permite aos nós observarem os pacotes que estão trafegando pela rede e acumularem os caminhos que estão contidos neles sem que participem, necessariamente, do processo de descoberta. Uma desvantagem do protocolo é a ausência de mecanismos que descartem rotas obsoletas.

Diferenças entre os dois protocolos (AODV e DSR) fazem com que o DSR comporte-se melhor em cenários com baixa mobilidade e número de nós, enquanto que o AODV apresenta o melhor desempenho em cenários com maior mobilidade e número de nós [1].

2.3 Protocolos Híbridos

Os protocolos híbridos combinam as características proativa e sob demanda [47]. Um exemplo de protocolo híbrido é o *Zone Routing Protocol* (ZRP) [15]. Este protocolo foi proposto com o objetivo de reduzir a sobrecarga de pacotes de controle dos protocolos proativos e diminuir a latência causada pela descoberta de rotas dos protocolos reativos.

O ZRP limita sua tabela de roteamento utilizando a estratégia proativa (roteamento intrazona) para destinos mais próximos, ou seja, para uma determinada parte da rede, na zona de

roteamento, o que evita o congestionamento de banda. Este protocolo também adota a estratégia sob demanda (roteamento interzona) quando deseja enviar pacotes para os destinos fora da zona de roteamento, o que diminui os atrasos na rede.

Capítulo 3

Modelos de Bateria

Os modelos matemáticos de bateria representam características reais de uma bateria com a finalidade de prever seu desempenho. São úteis para projetos de sistemas alimentados por baterias, pois permitem a análise do seu comportamento de descarga [48].

Existem vários tipos de modelos com diferentes características e complexidades [29][27]: modelos analíticos, modelos baseados em circuitos elétricos, modelos estocásticos e modelos eletroquímicos. Entre estes, os modelos analíticos destacam-se por serem simples e flexíveis na implementação, sendo facilmente configurados para diversos tipos de baterias.

Uma bateria geralmente consiste de uma matriz de uma ou mais células eletroquímicas. Cada célula é formada por dois eletrodos (i.e., condutor metálico por onde a corrente entra e sai de um sistema): um ânodo de polaridade negativa e um cátodo de polaridade positiva, separados por um eletrólito (i.e., condutor de eletricidade). As reações eletroquímicas que ocorrem na bateria dão origem às *espécies eletroativas* ou *elétrons*, que são liberados pelo ânodo para fornecerem corrente elétrica a um circuito externo, o que denomina-se fase de descarga da bateria [43].

O comportamento de uma bateria é representado por dois principais efeitos químicos, que devem ser considerados para determinar as propriedades de descarga e a capacidade da bateria (i.e., quantidade de materiais ativos presentes na bateria) [25], de modo que sua capacidade real nunca poderá exceder sua capacidade teórica (i.e., quantidade máxima de energia que pode ser extraida da bateria). Os efeitos químicos estão a seguir:

- *Efeito de taxa de capacidade*: depende da condição de que, quanto maior a corrente de descarga, menor será a capacidade disponível da bateria, geralmente medida em *mA-h*

ou *mA-ms*.

- *Efeito de recuperação da capacidade (ou de relaxação)*: a bateria recupera parte de sua capacidade quando encontra-se em períodos ociosos ou relaxada, isto é, quando a corrente fornecida pela bateria é reduzida significativamente [29]. Essa recuperação possibilita o aumento da vida útil da bateria.

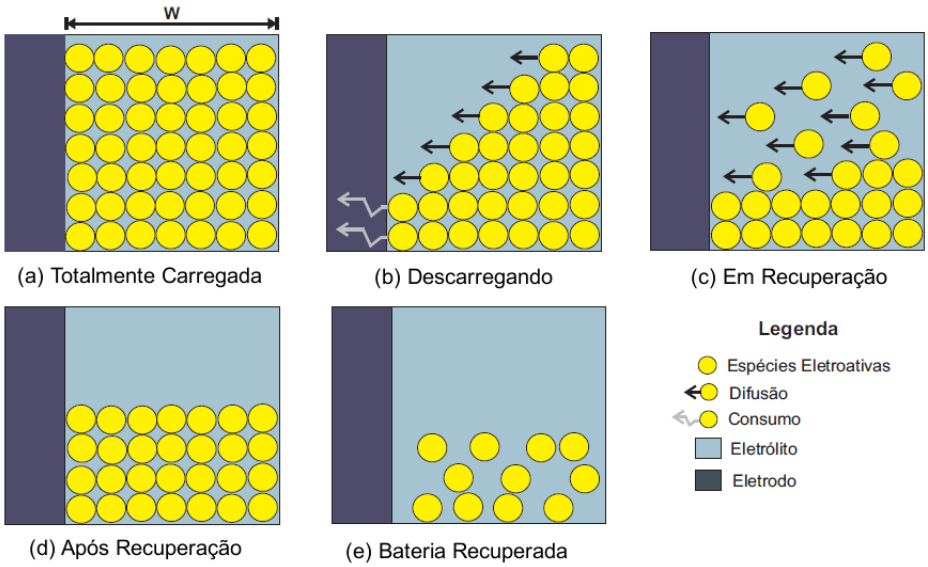


Figura 3.1: Estados de operação da bateria [48].

Muitos modelos analíticos de baterias conseguem representar o efeito de taxa de capacidade e o efeito de recuperação. A Figura 3.1, ilustra esses efeitos em uma bateria. Ela apresenta os diferentes estados de operação de uma bateria de forma simplificada. Observa-se na Figura 3.1(a), o estado da bateria totalmente carregada, com uma distribuição uniforme das espécies eletroativas em toda a região de comprimento *w* do eletrólito. O processo de descarga, representado na Figura 3.1(b), ocorre devido ao fluxo externo de *elétrons* entre os eletrodos, causando redução das *espécies eletroativas* da bateria. Quando a corrente é reduzida significativamente, ocorre um processo de difusão, de forma a reequilibrar a concentração das espécies na bateria (Figura 3.1(c)). Tal processo explica a recuperação da capacidade da bateria [42]. A Figura 3.1(d) mostra o estado da bateria após a difusão, com sua capacidade já recuperada. Quando a concentração das espécies atinge um certo limite, a bateria pára de fornecer carga ao sistema, ficando indisponível (Figura 3.1(e)).

3.1 Modelo Linear

O modelo analítico mais simples é o modelo linear, em que a bateria é descarregada de forma linear, e nenhum efeito (e.g., taxa de capacidade e relaxação) é representado por ele. A Equação 3.1 permite calcular a capacidade restante, C, de uma bateria, sempre que a corrente de descarga, I, mudar, sendo C' a capacidade no início da operação e td, o tempo de duração da operação [48][50].

$$C = C' - I.td. \tag{3.1}$$

A maioria dos simuladores de rede utiliza o modelo linear de descarga de bateria. Esse modelo pode induzir a interpretações equivocadas na análise de protocolos, pois o cálculo de forma errada da capacidade disponível da bateria pode levar a uma falsa estimação do consumo de energia, o que influencia diretamente no tempo de vida dos dispositivos de uma rede *ad hoc*. Por isso, neste trabalho é utilizado o modelo de Rakhmatov-Vrudhula [43], que considera o efeito de relaxação da bateria e calcula a sua capacidade de forma mais real, obtendo informações mais precisas do consumo de energia.

3.2 Modelo de Rakhmatov-Vrudhula

Rakhmatov e Vrudhula [43] propõem um modelo de bateria analítico que representa os efeitos de taxa de capacidade e de relaxação, presentes em diferentes tipos de baterias (e.g., alcalinas, íons de lítio). Estes efeitos consideram que, quanto maior a corrente de descarga, menor a capacidade disponível na bateria (efeito de taxa de capacidade), e que esta pode recuperar sua capacidade quando estiver em períodos ociosos (i.e., relaxada) e a corrente submetida for reduzida significativamente. A recuperação possibilita o aumento da vida útil da bateria e, consequentemente, o tempo de vida da rede. Outros trabalhos [2] [28] [11] também apresentam modelos realísticos de bateria, projetados para redes *ad hoc* e de sensores.

O modelo de Rakhmatov-Vrudhula [43], mais realista que o modelo linear, procura estimar o tempo de vida de uma bateria de íons de lítio, a partir da Equação 3.2, usando dois parâmetros específicos: o parâmetro α, que está relacionado à capacidade da bateria; e o parâmetro β, que está relacionado ao comportamento não linear da bateria durante os períodos

de carga e descarga.

$$\alpha = \sum_{k=1}^{n} 2I_{k-1}A(L, t_k, t_{k-1}, \beta). \tag{3.2}$$

Na Equação 3.2, I_{k-1} é a corrente de descarga durante o período $k-1$. A função A calcula o impacto do comportamento não linear na descarga da bateria, L é o tempo de vida da bateria, t_k é o tempo de duração do período k e t_{k-1} é o tempo de duração para o período $k-1$. Rakhmatov e Vrudhula [43] apresentam mais detalhes sobre esta equação e avaliam a qualidade desse modelo simulando uma bateria de íons de lítio.

Margi e Obraczka [31] afirmam que os principais simuladores de redes de computadores (e.g., NS-2 [21], GloMoSim [35], Qualnet [49]) utilizados na avaliação de desempenho de protocolos, possuem modelos de consumo de energia imprecisos, pois não consideram todos os estados do rádio, os diferentes níveis de energia que eles consomem e os efeitos não lineares da bateria.

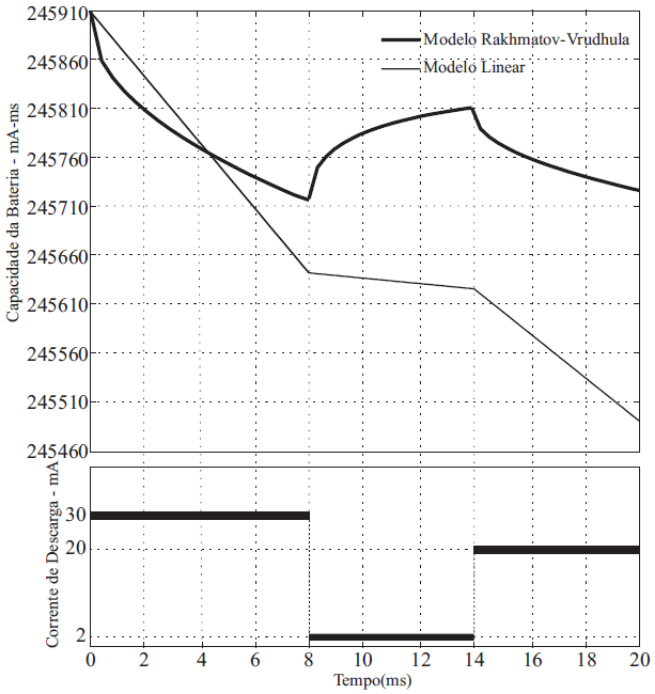
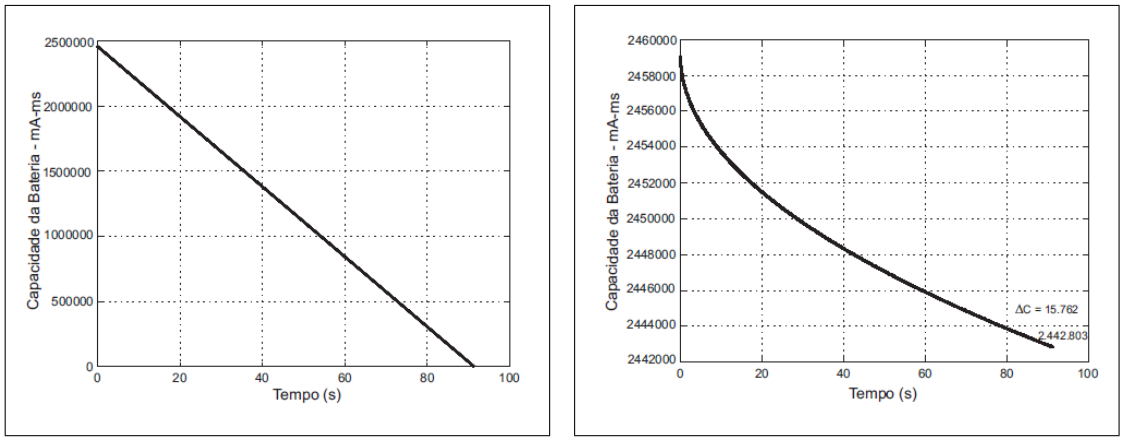


Figura 3.2: Capacidade da Bateria em Função do Perfil de Descarga [48].

Handy e Timmermann [16] apresentam um algoritmo para integração do modelo de Rakhmatov-Vrudhula em ambientes de simulação para redes sem fio, como o NS-2, que

realiza uma estimativa, de forma interativa, do tempo de vida de dispositivos, levando em conta os efeitos não lineares da bateria. Esta implementação foi adaptada ao NS-2 por Sausen [48], considerando baterias alcalinas para redes de sensores, e adotando α=4918200 e β=4034. Utilizou-se a implementação de Sausen neste trabalho, mas ajustada para baterias de íons de lítio, adotando α=35220 e β=0,637, mesmos valores utilizados por Handy e Timmermann [16].

Na Figura 3.2, Sausen [48] compara os modelos linear e de Rakhmatov-Vrudhula, e mostra a variação da capacidade da bateria (medida em *mA-ms*) para cada modelo, em um mesmo perfil de descarga. O gráfico considera um perfil de descarga variando a corrente em 30, 2 e 20 mA para os intervalos de tempo de 8, 6 e 6 ms, respectivamente. Nota-se que no intervalo de tempo entre 8 e 14 ms a corrente de descarga foi reduzida de 30 para 2 mA, caracterizando um repouso da bateria. Nesse instante, observou-se a recuperação da capacidade da bateria, no modelo de Rakhmatov-Vrudhula. Em relação ao modelo linear, para o mesmo intervalo de tempo, o gráfico mostra apenas um consumo linear da capacidade da bateria, sem nenhum efeito agindo sobre esta. Ao final dos 20 ms, percebe-se que a variação da capacidade da bateria representada pelo modelo linear foi consideravelmente maior que a do modelo de Rakhmatov-Vrudhula.



(a) Modelo Linear (b) Modelo de Rakhmatov-Vrudhula

Figura 3.3: Modelo Linear (a) versus Modelo de Rakhmatov-Vrudhula (b) para um perfil de descarga de 91 segundos [48].

A Figura 3.3, ilustra a diferença nos resultados obtidos com os dois modelos de bateria, para um mesmo perfil de descarga, com tempo de simulação de 91 segundos e capacidade

inicial de bateria de 2500000 *mA-ms*. Neste tempo, a simulação envolvendo o modelo linear foi encerrada, devido ao esgotamento da capacidade da bateria. Para o mesmo intervalo de tempo, considerando o modelo de Rakhmatov-Vrudhula, o consumo da capacidade da bateria (ΔC) não superou 15762 *mA-ms*. Assim, enquanto para o modelo linear, a bateria descarregou 100% o que representou um tempo de vida de 91 segundos, para o modelo realista, ela descarregou apenas $0,65\%$, tendo capacidade suficiente para fazer funcionar um dispositivo por mais tempo. Isso ocorre em função do modelo linear não conseguir capturar o comportamento não linear das baterias (i.e., efeito de taxa de capacidade e o efeito de recuperação).

Capítulo 4

Preparação e Execução dos Experimentos

A avaliação de desempenho de sistemas computacionais permite, através de métricas, identificar o comportamento dos sistemas e comparar a eficiência e rendimento destes. Para isso, existem métodos a serem seguidos, que se diferenciam na forma como os resultados são obtidos. Pesquisas referentes à avaliação de desempenho de protocolos de redes móveis *ad hoc* seguem basicamente os métodos de experimentação e simulação [2]. O método de experimentação trata da realização e monitoração de experimentos em sistemas reais (*testbeds*) e, por isso, é a que proporciona maior fidelidade dos dados obtidos. Contudo, podem acarretar em elevado custo e tempo de análise [41].

O método de simulação trata da construção de um modelo que simule o funcionamento do sistema a ser avaliado [41]. A simulação foi utilizada neste trabalho, e é o método mais empregado nas avaliações de protocolos em redes *ad hoc*, o que justifica-se pela riqueza de detalhes que esta explora e à facilidade na construção de cenários mais complexos. Todavia, o tempo necessário para realizá-la cresce proporcionamente à sua complexidade.

No presente trabalho, foram avaliados os desempenhos de três protocolos *unicast*, utilizando-se um modelo realístico de bateria. Foi avaliado um protocolo seguindo a abordagem proativa (*Optimized Link State Routing Protocol* (OLSR) [23]) e dois protocolos de abordagem reativa (*Ad Hoc On Demand Distance Vector Routing* (AODV) [39] e *Dynamic Source Routing* (DSR) [26]), já descritos no Capítulo 2.

4.1 Ambiente de Simulação

As simulações foram realizadas por meio do simulador de rede *Network Simulator* (NS-2) [21] na versão 2.34, disponibilizada em 2009. O NS-2 é um simulador de eventos discretos, de uso livre e com código fonte aberto, o que permite ao usuário realizar os ajustes que julgar necessário. É uma ferramenta com foco em pesquisas na área de redes de computadores, que oferece suporte a diversas tecnologias de rede, tanto guiadas como não guiadas (i.e., sem fio).

A estrutura básica do simulador é composta de um escalonador de eventos e dos componentes de rede (e.g., protocolos, agentes), desenvolvidos na linguagem orientada a objetos C++, o que garante mais praticidade na implementação desses recursos. O uso como *frontend* (i.e., interface com o usuário, manipulação de objetos e configuração de parâmetros) é feito através de *scripts* escritos na linguagem interpretada OTCL (*Object-oriented Tool Command Language*). O uso de uma linguagem interpretada facilita a alteração dos *scripts* sem a necessidade de recompilar. Assim, um usuário escreve suas simulações em *scripts* OTCL, utilizando-se dos componentes disponíveis na ferramenta, como os objetos C++ gerados na compilação do núcleo NS-2 [22].

Nos *scripts* de simulação, o usuário inicia o escalonador de eventos e depois configura os parâmetros da rede, determinando sua topologia, padrão de movimento, modelo de transmissão, modelo de bateria, padrão de tráfego, protocolos e agentes envolvidos. A execução desses *scripts* resulta em arquivos de saída, contendo informações detalhadas dessas simulações, que podem ser utilizadas nas análises. O NS-2 também oferece um gerador de topologias e um visualizador de simulações (i.e., Network Animator (NAM)), que mostra uma animação gráfica da topologia da rede.

Diversas implementações de protocolos de roteamento para redes *ad hoc* móveis podem ser encontradas para o NS-2. Os protocolos AODV e DSR, já vêm instalados na versão adotada do simulador. O protocolo OLSR foi instalado, utilizando-se da implementação UM-OLSR [45]. Do mesmo modo, como o modelo linear já vem padronizado na instalação do NS-2, o modelo de bateria de Rakhmatov-Vrudhula também foi adicionado ao simulador. Utilizou-se a implementação de Sausen [48] do modelo de Rakhmatov-Vrudhula, adaptada neste trabalho para baterias de íons de lítio. A escolha da utilização de baterias de íons de

lítio é devido a maioria dos dispositivos móveis na atualidade utilizarem esse tipo de bateria.

O modelo de transmissão utilizado nas simulações baseou-se nas características da interface de rede *Wireless WaveLan* da Lucent. Esta adota para a camada MAC (*Media Access Control*) o padrão 802.11 de redes *Wireless LANs*, que utiliza pacotes de controle RTS (*Request To Send*) e CTS (*Clear To Send*) para transmissões aos nós vizinhos. Todos os pacotes são armazenados em filas de transmissão com tamanho máximo de 50 pacotes e baseada no esquema FIFO (i.e., *First-In Fisrt-Out*), até que possam ser transmitidos pela camada MAC. Nestas filas, os pacotes de roteamento tem prioridade maior sobre os pacotes de dados, e para gerenciamento da fila, adota-se a política *DropTail*, em que os pacotes que chegam são descartados, caso a fila atinja o seu limite.

A interface de rede foi configurada para operar com frequência de $2,4\ GHz$, taxa nominal de transmissão de $11\ Mbps$, alcance de rádio de 250 metros e antenas omnidirecionais. O modelo de propagação de ondas de rádio, baseou-se no *TwoRayGround*, em que a potência do sinal diminui com o quadrado da distância percorrida.

4.2 Configuração do Consumo de Energia

Na implementação do modelo de Rakhmatov-Vrudhula foram utilizados a corrente em *miliampere* (mA) e o tempo em *milisegundos* (ms) para o cálculo da descarga da bateria. Por padrão, o NS-2 utiliza para o cálculo da descarga, a potência em *Watts* (W), o tempo em *segundos* (s) e a unidade de energia em *Joules* (J). Por isso, foram realizadas conversões dessas unidades e, ao final, o resultado do consumo de energia da bateria foi apresentado em *Joules*.

O consumo de energia dos dispositivos varia conforme o estado em que estes se encontram, transmitindo (TX), recebendo (RX) ou ocioso (IDLE). Utilizou-se os valores de corrente de descarga e potência da Tabela 4.1, para configurar o consumo de energia dos dispositivos da rede, utilizando-se uma voltagem de $4,74\ V$. Esses valores foram medidos por Feeney [13], de acordo com as especificações da interface *Wireless WaveLan* da *Lucent*, operando a $11\ Mbps$. Parte dessa energia é consumida pelo rádio da antena da placa, que foi configurado com a potência de $281,8\ mW$ para proporcionar o alcance ideal de 250 metros.

Quanto à bateria, esta foi configurada para operar com capacidade inicial de 40 *Joules*,

Tabela 4.1: Consumo de energia da interface *Wireless WaveLan* 11 Mbps da *Lucent*.

Estado	**Corrente**	**Potência**
IDLE	$156\ mA$	$0,740\ W$
RX	$190\ mA$	$0,900\ W$
TX	$284\ mA$	$1,350\ W$

nos experimentos. Trata-se de um valor relativamente baixo para uma bateria. No entanto, foi utilizado com a finalidade de forçar o gasto total da energia dos dispositivos e, consequentemente, o desligamento dos mesmos, o que permite verificar o tempo de vida da rede.

4.3 Descrição dos Cenários

Os cenários de simulação foram criados por meio da ferramenta "*Setdest*", disponível no NS-2 [21]. Esta ferramenta é capaz de gerar arquivos de *trace* contendo todas as informações de movimentação dos dispositivos na rede, com base no modelo aleatório de mobilidade *Random Waypoint*.

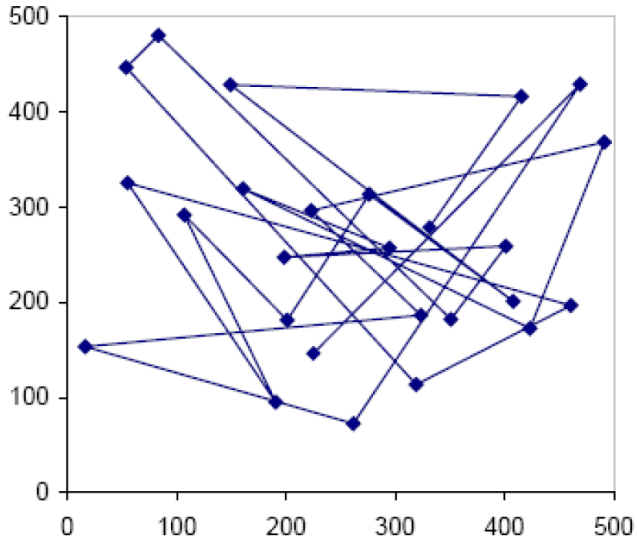


Figura 4.1: Movimento de um nó no modelo de mobilidade *Random Waypoint* [5]

O modelo de mobilidade *Random Waypoint* determina que cada dispositivo móvel sele-

cione randomicamente um ponto de destino $(x; y)$, dentro de uma área de tamanho determinado, e uma velocidade constante na qual o nó deverá se mover até alcançar o ponto de destino. Em seguida, o nó poderá ficar parado por um certo tempo e depois recomeçar o processo, selecionando aleatoriamente um novo ponto de destino e velocidade (Figura 4.1).

Para a configuração do padrão de mobilidade através do "*Setdest*", alguns parâmetros de entrada são especificados: número de nós, velocidade mínima ($\frac{m}{s}$), velocidade máxima ($\frac{m}{s}$), comprimento do terreno (m), largura do terreno (m), tempo de simulação (s) e tempo de pausa (s). O tempo de pausa representa o tempo da simulação em que os dispositivos ficam parados. Assim, podem ser criados cenários com diferentes perfis de mobilidade: pouca mobilidade (i.e., velocidade baixa e tempo de pausa alto) ou mobilidade intensa (i.e., velocidade alta e tempo de pausa baixo).

A Figura 4.2 mostra trechos de um exemplo de arquivo gerado pelo "*Setdest*", representando o padrão de movimento de um dispositivo. Nas duas primeiras linhas, o nó "15" é configurado com a posição inicial aleatória $x = 1131,618429994358$ e $y = 16,571359207584$, e em seguida, na terceira linha, no instante $0,000000000000$ segundos, o dispositivo inicia o movimento para a posição $(x; y) = (1208,569533397086; 150,589909502934)$ com velocidade de $18,364398870569$ $\frac{m}{s}$. As demais linhas mostram as movimentações seguintes.

```
$node_(15) set X_ 1131.618429994358
$node_(15) set Y_ 16.571359207584
$ns_ at 0.000000000000 "$node_(15) setdest 1208.569533397086 150.589909502934 18.364398870569"
$ns_ at 8.415165630022 "$node_(15) setdest 379.229767685708 114.477909289162 3.616976913142"
$ns_ at 237.923295492757 "$node_(15) setdest 1019.407439636241 107.526726314230 18.309108207071"
$ns_ at 272.890340435440 "$node_(15) setdest 262.359333357481 163.010997862156 5.942036229173"
```

Figura 4.2: Trechos do padrão de movimento de um nó da rede

Para as simulações deste trabalho, os cenários foram configurados com 50 nós distribuídos aleatoriamente em uma área de tamanho 1500 x 300 metros, movimentando-se com velocidade constante durante cada percurso, variando-a no intervalo de $]0, 20]$ $\frac{m}{s}$ (i.e., velocidade máxima até 20 $\frac{m}{s}$ não incluindo o zero). Com um tempo de simulação de 300 segundos, os cenários foram montados, variando-se o tempo de pausa dos nós em 0, 100, 200 e 300 segundos, o que representa diferentes perfis de mobilidade na rede. Com tempo de pausa 0 segundos, por exemplo, os nós se movimentam a todo instante, e com tempo de pausa 300 segundos, os nós ficam sempre parados. Os parâmetros relativos à configuração

dos experimentos de simulação estão disponíveis na Tabela 4.2.

Uma vez que a geração dos cenários ocorre de forma randômica, existem várias possibilidades de topologia. Diante disso, para alcançar resultados estatisticamente mais coerentes, deve-se executar uma quantidade satisfatória de experimentos, considerando um determinado intervalo de confiança [24]. Assim, determinou-se uma amostra de 10 execuções para 95% de confiança. Então, cada cenário de simulação foi executado 10 vezes, variando-se a semente de aleatoriedade, o que garante diferentes cenários.

Tabela 4.2: Parâmetros dos experimentos.

Tamanho do terreno	1500 x 300 m
Tempo de simulação	300 s
Número de nós	50
Velocidade dos nós	$]0, 20]$ $\frac{m}{s}$
Tempos de pausa	0, 100, 200 e 300 s
Quantidade de amostras	10
Tipo de pacote	CBR
Tamanho do pacote	512 *bytes*
Taxa de envio	4 $\frac{pacotes}{s}$
Número de nós fontes	5, 10, 15 e 20
Capacidade da bateria	40 *Joules*

Em relação ao tráfego de pacotes dos dispositivos, este foi configurado por meio da ferramenta CBRGEN [21], a qual gera arquivos especificando tráfegos do tipo *Constant Bit Rate* (CBR). A aplicação CBR presente em cada dispositivo fonte de tráfego, é capaz de gerar um fluxo de dados com uma taxa de envio constante. Neste trabalho, utilizou-se esta aplicação, com pacotes de tamanho 512 *bytes*, a uma taxa de envio de 4 pacotes por segundo. O número de nós fontes de tráfego variou em 5, 10, 15 e 20, com o objetivo de aumentar o envio de pacotes na rede.

Depois de gerados, os arquivos de padrões de tráfego e de mobilidade são combinados e chamados por um *script* OTCL (Apêndice A). Essa combinação resulta em um conjunto de experimentos de simulação, realizados para cada um dos três protocolos de roteamento

(AODV, DSR e OLSR). Quanto as configurações dos protocolos no simulador, estas foram mantidas padrões. Cada experimento também foi realizado duas vezes, uma utilizando-se o modelo linear de bateria, e outra com o modelo de Rakhmatov-Vrudhula.

4.4 Execução dos Experimentos

Depois de preparados e configurados, todos os experimentos de simulação foram executados em diversas máquinas, visto que eles consumiam bastante processamento das mesmas e levavam um grande tempo para serem concluídos. Tal demora ocorreu principalmente com os experimentos envolvendo o modelo de Rakhmatov-Vrudhula, pois este apresenta um elevado esforço computacional. Esse é o motivo pelo qual foi utilizado o tempo de simulação de 300 segundos e não um tempo maior, pois isso aumentaria consideravelmente o tempo de execução dos experimentos. Vale considerar que o tempo de simulação utilizado ainda é significativo para análise. Os recursos computacionais de *hardware* e sistema operacional utilizadas nas etapas de preparação, execução e análise dos experimentos são apresentados abaixo com as respectivas configurações.

- *Notebook*

  Sistema Operacional: Linux Ubuntu Desktop 10.04 64-bits

  Processador: Intel Core 2 Duo 2.53GHz

  Memória: 4GB

  Armazenamento: 500GB

  Etapa de Utilização: Preparação e Análise

- *Desktop*

  Sistema Operacional: Linux Ubuntu Desktop 10.04 64-bits

  Processador: Intel Core 2 Duo 3.0GHz

  Memória: 4GB

  Armazenamento: 320GB

  Etapa de Utilização: Preparação, Execução e Análise

- *Cluster*

  8 nodos com 8 núcleos cada

  Sistema Operacional: Linux Ubuntu Server 10.04 64-bits

  Processador: Intel Xeon 2.67GHz

  Memória: 8GB

  Armazenamento: 1TB

  Etapa de Utilização: Execução

4.5 Métricas de Desempenho

A execução das simulações resultam em arquivos de *trace* contendo todas as informações do que ocorreu na rede durante o período de simulação. Exemplos dessas informações são: em um certo instante um determinado nó envia um pacote, em outro instante um certo nó recebe um pacote, em outro, a capacidade restante da bateria de um dispositivo é calculada. Diante desses dados, desenvolveu-se um *script* na linguagem *awk* [44], capaz de ler os *traces* e extrair algumas métricas de desempenho para avaliação dos protocolos de roteamento. O *script awk* desenvolvido será apresentado no Apêndice B. Dentre as métricas coletadas, o consumo médio de energia dos dispositivos e o tempo de vida da rede são essenciais, particularmente em ambientes com recursos limitados de bateria. As métricas utilizadas são descritas a seguir:

- **Consumo Médio de Energia:** O consumo de energia foi definido como a média da energia gasta por todos os nós juntos, ou seja, o consumo médio de energia da rede, e não de um nó específico.

- **Tempo de Vida da Rede:** Não existe um concesso entre autores em relação à definição exata de tempo de vida da rede. Dietrich e Dressler [12] apresentam diferentes abordagens, com base no número de nós vivos (quando o primeiro nó esgotar sua bateria ou quando uma certa quantidade de nós esgotarem suas baterias), na cobertura dos nós, na conectividade, entre outras. No presente trabalho, considera-se o tempo de vida quando o primeiro nó da rede ficar inoperante por esgotamento da bateria.

Mesmo depois que o primeiro nó da rede fica inoperante, a simulação continua sendo executada, e esta só encerra quando o tempo de simulação se esgota. Por isso, também foi considerado na análise o tempo de vida da rede quando uma certa porcentagem dos nós (e.g., 20% e 50%) ficaram inoperantes por esgotarem suas baterias.

- **Quantidade de Nós Sobreviventes:** É a quantidade de dispositivos com energia remanescente, após o final da simulação.

- **Fração de Entrega de Pacotes:** É a razão entre o número de pacotes recebidos pelo destino e o número de pacotes enviados pela camada de aplicação CBR da origem. A quantidade de pacotes perdidos também pode ser obtida a partir dessa métrica.

- **Carga de Roteamento Normalizada:** É a quantidade de pacotes de roteamento necessários, contando com as retransmissões dos nós intermediários, para que cada pacote de dados CBR seja entregue ao destino.

- **Atraso Médio Fim a Fim:** É o atraso médio de entrega de pacotes, considerando-se o tempo decorrido desde a geração do pacote na fonte até o seu recebimento no destino. Este tempo inclui atrasos relacionados à propagação do pacote e espera na fila de transmissão, ocasionada principalmente por congestionamento na rede ou inexistência de rotas.

Capítulo 5

Análise dos Resultados

Os resultados das simulações são apresentados através de tabelas e gráficos, comparando-se o desempenho dos três protocolos de roteamento (AODV, DSR e OLSR) para os dois modelos de bateria (Linear e Rakhmatov-Vrudhula). Cada valor da tabela ou ponto do gráfico representa a média e o intervalo de confiança das 10 amostras rodadas para cada cenário de simulação, considerando-se um nível de confiança de 95%.

Em alguns pontos dos gráficos, as barras referentes ao intervalo de confiança podem não ser perfeitamente visualizadas, pois apresentaram valores pequenos. Os intervalos do eixo das ordenadas foram ajustados em cada gráfico, para melhor vizualização. Neste caso, procurou-se, quando possível, deixar os gráficos de cada métrica com a mesma escala e, assim, facilitar a comparação de um gráfico com outro (e.g., comparar o desempenho dos protocolos no gráfico do modelo linear com o gráfico do modelo de Rakhmatov-Vrudhula).

A seguir são apresentados os recursos computacionais de *software* e versões utilizados na etapa de análise dos experimentos:

- *OpenOffice.org Calc*

  Versão: 3.2.0

  Descrição: Utilizado na construção dos gráficos referentes as métricas de desempenho.

- *SPSS Statictics* [8]

  Versão: 18.0.0

Descrição: Utilizado na construção das tabelas de correlação entre as métricas.

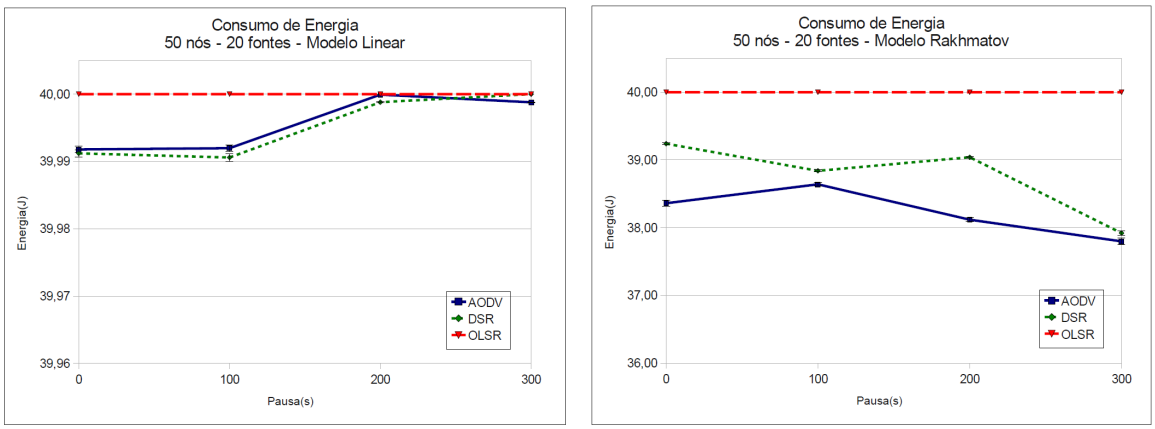
A análise dos resultados contempla as seguintes relações: Parâmetro *versus* Métrica *versus* Modelo de Bateria; e Métrica *versus* Métrica.

5.1 Relação Parâmetro x Métrica x Modelo de Bateria

A Relação Parâmetro *versus* Métrica *versus* Modelo de Bateria visa analisar o comportamento de cada protocolo (i.e., AODV, DSR, OLSR) através do impacto que cada parâmetro (e.g., tempo de pausa, número de fontes) causa nas métricas utilizadas (i.e., consumo de energia, tempo de vida da rede, fração de entrega de pacotes, quantidade de nós sobreviventes, carga de roteamento normalizada e atraso médio fim a fim) para cada modelo de bateria (i.e., linear e Rakhmatov-Vrudhula). Todas as possíveis relações foram analisadas para cada métrica relacionada.

5.1.1 Consumo de Energia

A Figura 5.1 mostra os gráficos que representam os resultados quanto ao consumo médio de energia dos nós, para os modelos linear, Figura 5.1(a), e de Rakhmatov-Vrudhula, Figura 5.1(b), variando-se o parâmetro tempo de pausa e considerando-se um cenário com 20 nós fontes.



(a) (b)

Figura 5.1: Consumo médio de energia dos nós, variando-se o tempo de pausa, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula

A Tabela 5.1, que expressa melhor as diferenças nos valores dos modelos linear e realista, apresenta, em números, os mesmos resultados dos gráficos da Figura 5.1, com os intervalos de confiança dos consumos médios de energia dos nós, quantidade de 20 nós fontes e o tempo de pausa variável. A Tabela também mostra em números percentuais o ganho no consumo de energia na comparação do modelo de Rakhmatov-Vrudhula com o modelo linear. Outro objetivo do ganho nessa Tabela é comparar o desempenho dos protocolos, observando qual deles obteve maior ganho.

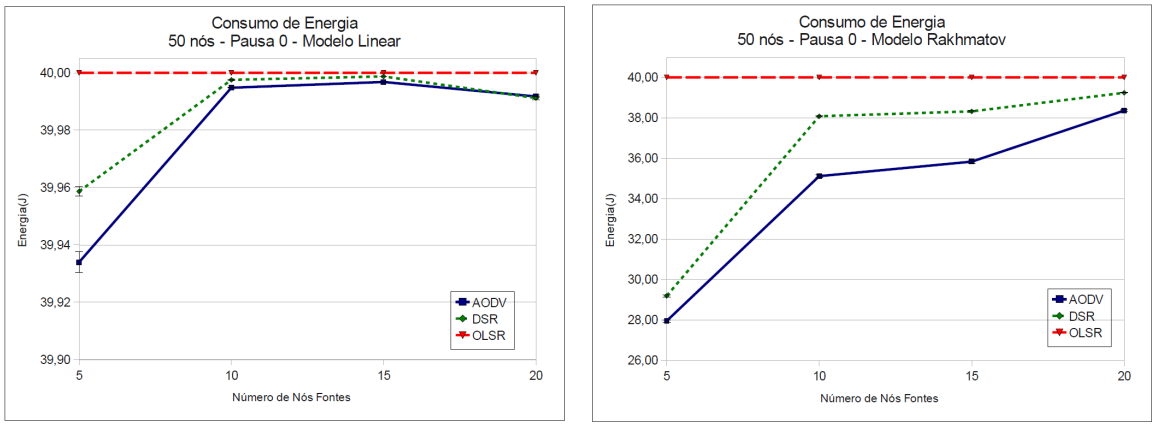
Tabela 5.1: Consumo de energia, variando-se o tempo de pausa

AODV			
Pausa	Linear (J)	R-V (J)	Ganho (%)
0	$39,992 \pm 0,001$	$38,359 \pm 0,044$	$4,08$
100	$39,992 \pm 0,001$	$38,639 \pm 0,025$	$3,38$
200	$39,999 \pm 0,000$	$38,118 \pm 0,030$	$4,70$
300	$39,999 \pm 0,000$	$37,798 \pm 0,046$	$5,50$
DSR			
Pausa	Linear (J)	R-V (J)	Ganho (%)
0	$39,991 \pm 0,001$	$39,240 \pm 0,012$	$1,88$
100	$39,991 \pm 0,001$	$38,839 \pm 0,017$	$2,88$
200	$39,999 \pm 0,000$	$39,039 \pm 0,015$	$2,40$
300	$40,000 \pm 0,000$	$37,918 \pm 0,029$	$5,21$
OLSR			
Pausa	Linear (J)	R-V (J)	Ganho (%)
0	$40,000 \pm 0,000$	$40,000 \pm 0,000$	$0,00$
100	$40,000 \pm 0,000$	$40,000 \pm 0,000$	$0,00$
200	$40,000 \pm 0,000$	$40,000 \pm 0,000$	$0,00$
300	$40,000 \pm 0,000$	$40,000 \pm 0,000$	$0,00$

Dentre os resultados, vale destacar o desempenho do protocolo OLSR, que manteve o mesmo consumo de 40 *Joules* (i.e., capacidade total da bateria) em todos os cenários, com tempos de pausa de 0, 100, 200 e 300 segundos. Logo, esgotou a bateria de todos os nós da rede, na totalidade dos experimentos realizados, para os dois modelos de bateria. O cálculo mais acurado da capacidade da bateria e informações mais precisas sobre consumo de energia dos dispositivos não foram suficientes para mostrar alterações de desempenho do OLSR nos dois modelos de bateria. A característica proativa desse protocolo pode ter ocasionado esse consumo excessivo de energia, pois são enviadas uma quantidade maior de

pacotes de controle para a manutenção das rotas.

Na análise comparativa dos modelos de bateria, para os protocolos reativos, nota-se menor consumo de energia no modelo realista, principalmente em cenários com menor mobilidade dos dispositivos. O maior ganho para o consumo de energia foi de aproximadamente 5%, em cenários com tempo de pausa de 300 segundos (sem movimentação dos dispositivos). Em contrapartida, quando a análise é feita para cada modelo de bateria, os protocolos AODV e DSR obtiveram desempenho semelhante para o modelo linear, ou ainda, a diferença entre eles não foi significativa. Diferentemente, no modelo realista, o AODV apresentou certa vantagem, pois obteve valores significativamente menores de consumo de energia em comparação com o DSR. Os ganhos para o consumo de energia do AODV também foram superiores aos ganhos do DSR (Tabela 5.1).



(a) (b)

Figura 5.2: Consumo médio de energia dos nós, variando-se o número de nós fontes, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula

A Figura 5.2 apresenta os graficos relativos ao consumo médio de energia dos nós, para os modelos linear, Figura 5.2(a), e de Rakhmatov-Vrudhula, Figura 5.2(b), variando-se o parâmetro número de nós fontes em 5, 10, 15 e 20, em um cenário com tempo de pausa igual a 0 segundos; isto é, os dispositivos movimentando-se a todo instante. A Tabela 5.2 representa os mesmos resultados destes gráficos, com o consumo médio de energia dos nós e a porcentagem do ganho obtido comparando-se os dois modelos de bateria.

O protocolo OLSR manteve o desempenho dos resultados anteriores; ou seja, gasto total da bateria (40 *Joules*) de todos os nós da rede, na totalidade dos experimentos realizados. As

informações mais precisa sobre o consumo de energia dos dispositivos não foram suficiente para que alterasse o desempenho do OLSR, quando comparados os dois modelos de bateria e, portanto, não obteve nenhum ganho. Esse comportamento do OLSR já era esperado e justifica-se por sua característica proativa.

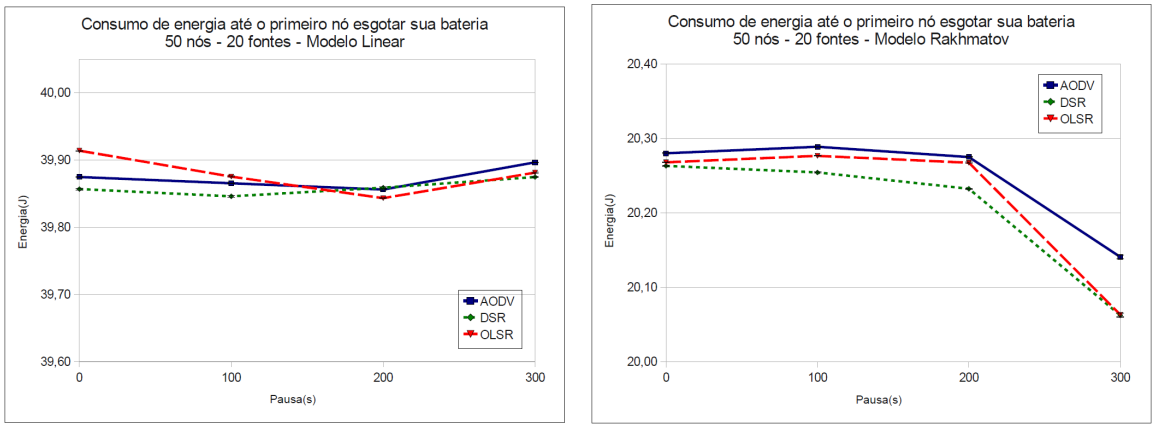
Tabela 5.2: Consumo de energia, variando-se o número de nós fontes

AODV			
Fontes	Linear (J)	R-V (J)	Ganho (%)
5	$39,934 \pm 0,004$	$27,951 \pm 0,057$	$30,00$
10	$39,995 \pm 0,000$	$35,116 \pm 0,068$	$12,20$
15	$39,997 \pm 0,000$	$35,836 \pm 0,079$	$10,40$
20	$39,992 \pm 0,001$	$38,359 \pm 0,044$	$4,08$
DSR			
Fontes	Linear (J)	R-V (J)	Ganho (%)
5	$39,959 \pm 0,002$	$29,193 \pm 0,056$	$26,94$
10	$39,998 \pm 0,000$	$38,079 \pm 0,033$	$4,80$
15	$39,999 \pm 0,000$	$38,319 \pm 0,037$	$4,20$
20	$39,991 \pm 0,001$	$39,240 \pm 0,012$	$1,88$
OLSR			
Fontes	Linear (J)	R-V (J)	Ganho (%)
5	$40,000 \pm 0,000$	$40,000 \pm 0,000$	$0,00$
10	$40,000 \pm 0,000$	$40,000 \pm 0,000$	$0,00$
15	$40,000 \pm 0,000$	$40,000 \pm 0,000$	$0,00$
20	$40,000 \pm 0,000$	$40,000 \pm 0,000$	$0,00$

Para os protocolos AODV e DSR, o consumo de energia dos dispositivos cresceu com o aumento do número de fontes de dados na rede. No entanto, o AODV apresentou menor consumo que o DSR, tanto para o modelo linear quanto para o modelo de Rakhmatov-Vrudhula. Apenas no cenário com 20 fontes, a diferença entre os dois portocolos não foi significativa. O AODV também apresentou ganhos superiores no consumo de energia, comparado-se os dois modelos de bateria, atingindo 30% para os cenários com 5 fontes. Tal diferença, pode ser explicada pelo fato do DSR capturar e armazenar em seu *cache* mais informações dos pacotes que trafegam na rede, quando este opera em modo promíscuo, o que acarreta um maior consumo de energia em relação ao AODV.

No protocolo OLSR, todos os dispositivos esgotaram suas baterias ao final das simulações e, por isso, não foi possível observar a variação do seu consumo médio de energia na

presença dos diferentes padrões de tráfego e mobilidade. Diante disso, a Figura 5.3 mostra os resultados referentes ao consumo médio de energia em função da mobilidade, calculado no momento em que o primeiro nó da rede morre. Assim, foi possível observar a variação do consumo de energia no OLSR.



(a) (b)

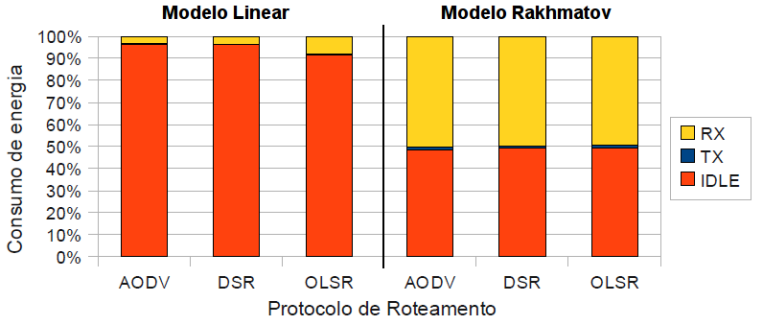
Figura 5.3: Consumo médio de energia dos nós quando o primeiro nó da rede tem sua bateria esgotada, variando-se o tempo de pausa, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula

Com base no modelo linear, percebe-se uma certa proximidade no desempenho dos três protocolos, mas nos cenários com maior mobilidade o OLSR permaneceu com maior consumo médio e o DSR menor. Para o modelo realista, os consumos de energia dos protocolos aumentaram com a mobilidade na rede, atingindo valores pouco maiores que 20 *Joules*. Dessa vez, o maior consumo foi apresentado pelo AODV, causado principalmente pelo seu processo de manutenção de rotas, que sobrecarrega a rede com o envio de mensagens *HELLO*. Já no OLSR a manutenção de rotas ocorreu com menor frequência até os dispositivos começarem a esgotar suas baterias e, por isso, consumiu menos energia que o AODV.

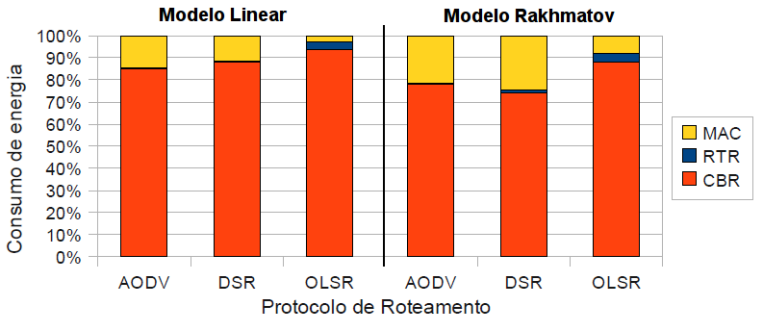
A Figura 5.4(a) ilustra a porcentagem da energia consumida de acordo com o estado de operação dos dispositivos: transmitindo (TX), recebendo (RX) e ocioso (*Idle*). Nestes gráficos, considerou-se um cenário básico com 20 fontes de tráfego e tempo de pausa de 0 segundos (i.e., mobilidade máxima). Nos três protocolos, quanto ao modelo linear, a maior parte da energia (i.e.., valor superior a 90%) é consumida quando os nós estão ociosos, pois

estes ficam a maior parte do tempo sob escuta dos pacotes que passam pela rede.

No modelo realista, o consumo de energia é maior quando os dispositivos se apresentam nos estados RX e ocioso, nos diferentes protocolos. Observou-se também um aumento do consumo de energia nos estados TX e RX, comparado-se com o modelo linear. Isso justifica-se pela recuperação da capacidade da bateria dos dispositivos quando ficam em estado ocioso, o que resulta em mais capacidade de bateria para o funcionamento destes, e assim, poderem entrar mais vezes nos estados de transmissão e recebimento de dados.



(a)



(b)

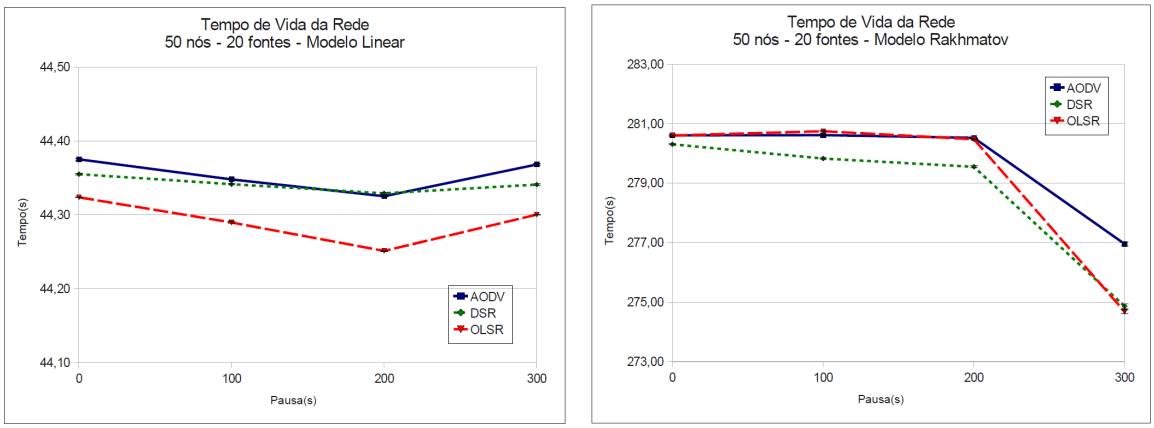
Figura 5.4: Porcentagem de consumo de energia, segundo (a) estado de operação (TX, RX e Idle) e (b) tipo de pacote (CBR, MAC e Roteamento)

A Figura 5.4(b) mostra a porcentagem da energia consumida em função do tipo de pacote envolvido: dados (CBR), MAC e roteamento; para 20 fontes e pausa de 0. A energia gasta com o tráfego de pacotes de dados afeta significativamente o consumo total, ultrapassando os 70% para os dois modelos de bateria, mas com uma porcentagem menor de consumo para o modelo de Rakhmatov-Vrudhula. O protocolo OLSR apresentou maiores porcentagens de consumo de energia para os pacotes de roteamento em relação aos protocolos reativos, devido à sua característica proativa, que aumenta o tráfego de pacotes de controle na rede.

O OLSR também apresentou maiores porcentagens de consumo de energia para os pacotes CBR, mas isso não indica, necessariamente, maior quantidade de envios.

5.1.2 Tempo de Vida da Rede

Os gráficos da Figura 5.5 apresentam os resultados referentes ao tempo de vida da rede, em função da mobilidade, para os dois modelos de bateria. Nesses gráficos, o tempo de vida representa o momento em que o primeiro nó da rede esgota sua bateria. Vale ressaltar a diferença importante desta métrica entre os dois modelos de bateria. Enquanto que no modelo linear o primeiro nó da rede morre entre $44,2$ e $44,4$ segundos, no modelo realista, o primeiro nó morre entre 274 e 281 segundos dentro dos 300 segundos de simulação. A Figura 5.6 mostra o tempo de vida da rede para os dois modelos de bateria, em função da carga aplicada na rede, e do mesmo modo, a diferença entre os modelos de bateria é bastante considerável.



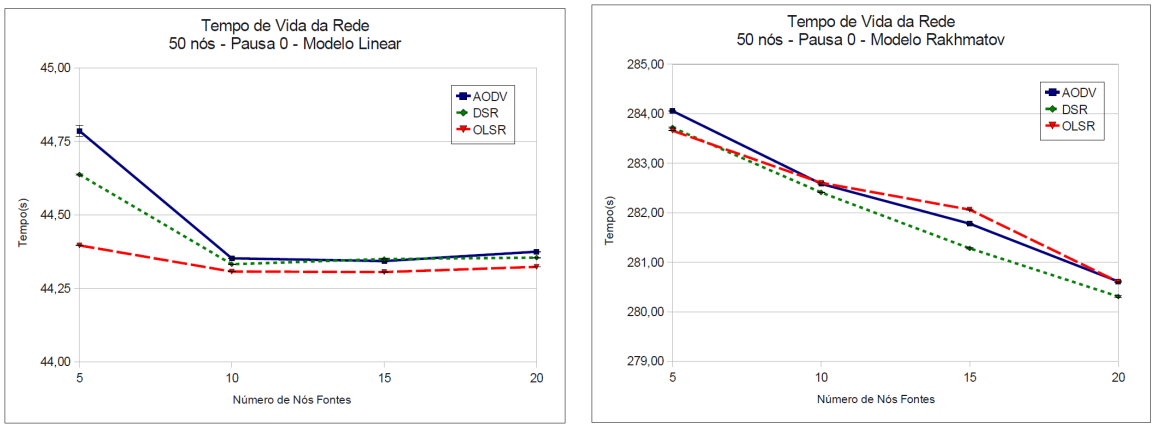
(a) (b)

Figura 5.5: Tempo de vida da rede, variando-se o tempo de pausa, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula

A diferença entre os modelos de bateria representa um ganho expressivo de aproximadamente 630% no tempo de vida da rede quando o primeiro nó morre, para o modelo de Rakhmatov-Vrudhula, e é explicada pelo efeito de relaxação capturado pelo modelo realista. Este efeito faz com que a bateria dos dispositivos recupere sua capacidade, demorando mais tempo para a energia ser consumida, em comparação ao modelo linear. Os dispositivos tem a possibilidade de realizar mais ações na rede, como transmitir, rotear e receber pacotes,

porque sobrevivem por mais tempo durante a simulação.

A partir de uma análise detalhada dos *traces* gerados nos experimentos, constatou-se que os primeiros dispositivos a morrerem, em geral, estão diretamente ligados ao tráfego da rede, participando como fonte ou destino de pacotes de dados. Assim, estes dispositivos entram com mais frequência no estado TX ou RX, que consomem mais energia e, consequentemente, esgotam sua bateria antecipadamente.



(a) (b)

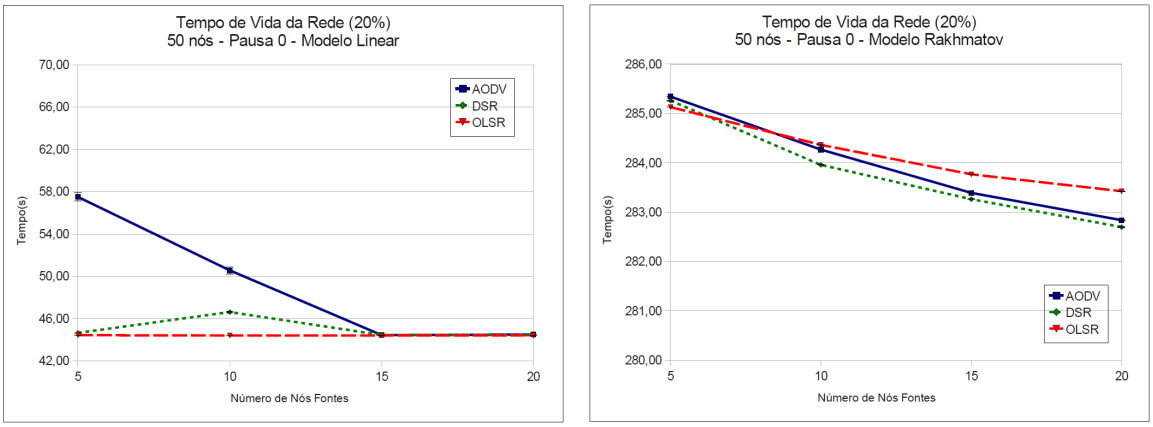
Figura 5.6: Tempo de vida da rede, variando-se o número de nós fontes, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula

Para o modelo linear, os três protocolos obtiveram desempenhos semelhantes quanto ao tempo de vida da rede, mas com uma pequena vantagem para o AODV. Nos cenários com tempo de pausa de 200 segundos (Figura 5.5) e 15 fontes (Figura 5.6), por exemplo, essa vantagem não chega a ser significativa em relação ao DSR. Os protocolos também apresentaram uma certa estabilidade no comportamento, pois os resultados se mantiveram próximos quando a mobilidade e a carga na rede foram alteradas. Observou-se maior diferença apenas no cenário com 5 nós fontes, em que o AODV apresentou superioridade em relação aos demais protocolos.

Para o modelo realista, a intensificação do tráfego na rede determinou que os dispositivos morressem mais cedo, já que o consumo de energia cresce com a elevação de carga na rede. Por outro lado, o aumento da mobilidade resultou em pequeno acréscimo do tempo de vida da rede, visto que os nós tendem a permanecer em estado ocioso por mais tempo, devido à maior frequência de quebras de enlace. Com as baterias relaxadas por mais tempo, os

dispositivos tendem a consumir energia mais lentamente. No entanto, estes provavelmente entregarão uma quantidade menor de pacotes.

Em geral, para o modelo realista, os três protocolos também obtiveram valores próximos para o tempo de vida da rede. Porém, os protocolos AODV e OLSR apresentaram uma pequena vantagem em relação ao DSR, principalmente nos cenários com maior mobilidade e quantidade de nós fontes. Apesar de que, no protocolo OLSR, as baterias de todos os nós esgotaram-se ao final das simulações baseadas no modelo de Rakhmatov-Vrudhula, como denotou-se na análise para o consumo de energia, o tempo de vida da rede mostrou-se similar ou superior aos demais, em alguns pontos (e.g., tempos de pausa 0 e 100, com 15 ou 20 fontes). Este comportamento significa uma vantagem para o OLSR, pois, apesar de um desempenho pior no modelo linear, com um modelo de bateria mais realista, a rede passou a funcionar por mais tempo que os demais protocolos, com maior disponibilidade para o envio de dados.



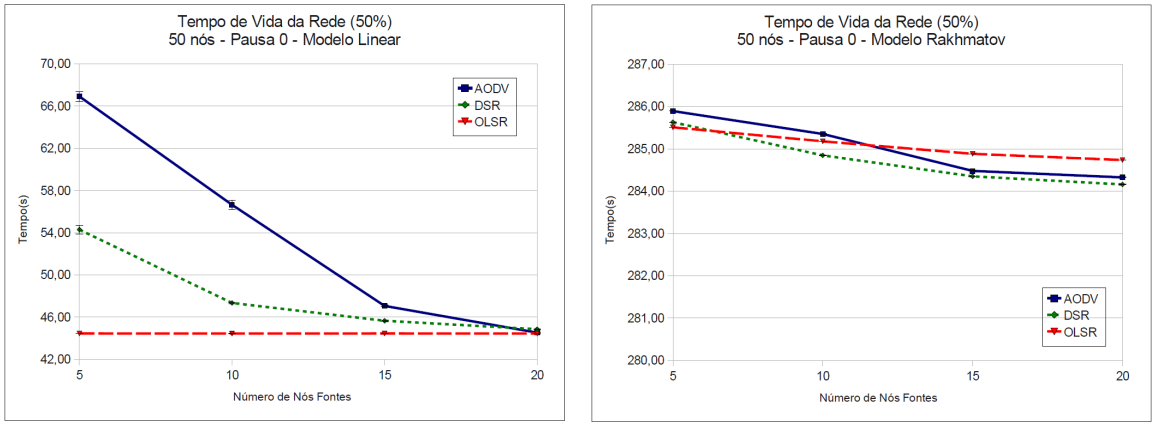
(a) (b)

Figura 5.7: Tempo de vida da rede (quando 20% dos nós morrem), variando-se o número de nós fontes, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula

Visto que, após o primeiro nó esgotar sua bateria, a simulação não é interrompida e a rede continua parcialmente funcional, consideraram-se outras abordagens para o cálculo do tempo de vida da rede, quando 20% e 50% dos dispositivos ficam inoperantes. As Figuras 5.7 e 5.8 mostram, respectivamente, o tempo de vida da rede quando 20% e 50% dos dispositivos esgotam suas baterias, de acordo com o número de nós fontes.

Para o modelo linear, visualiza-se nas Figuras 5.7(a) e 5.8(a) que o tempo de vida dos

protocolos reativos variou bastante, principalmente nos cenários com menor quantidades de nós fontes, comparando-se com os resultados da Figura 5.6(a). Por outro lado, o OLSR manteve-se quase estável. A variação observada deve-se à condição de que a maioria dos dispositivos diretamente ligados ao tráfego de dados estavam entre os 20% ou 50% dos nós inoperantes. Então, os demais dispositivos continuarão operando em estado ocioso, fazendo a rede sobreviver por mais tempo. Em contrapartida, a rede não estará realizando sua principal função, que é o envio de dados, pois os nós fontes estarão inoperantes.



(a) (b)

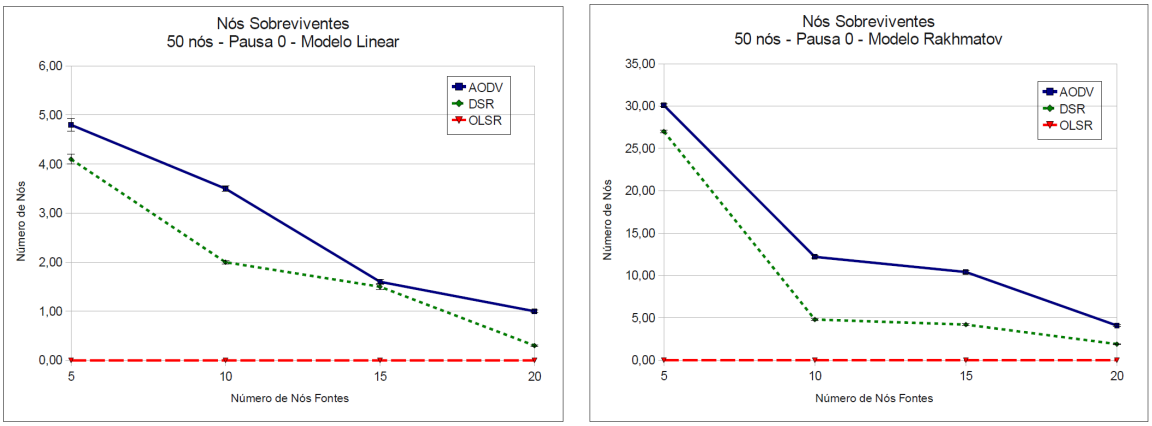
Figura 5.8: Tempo de vida da rede (quando 50% dos nós morrem), variando-se o número de nós fontes, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula

Para o modelo realista, nas Figuras 5.7(b) e 5.8(b), os protocolos apresentaram comportamento similar, mas com uma pequena vantagem para OLSR, que obteve um maior tempo de vida da rede nos cenários com mais fontes (i.e., 15 e 20 fontes).

5.1.3 Quantidade de Nós Sobreviventes

Para diferenciar melhor o desempenho dos protocolos, analisou-se também a quantidade de nós sobreviventes ao final da simulação, em função do padrão de tráfego na rede (Figura 5.9). Como cada ponto do gráfico representa uma média de 10 amostras, os resultados não estão representados em números inteiros, como determina a métrica, devendo-se considerar uma aproximação.

No AODV e DSR, percebe-se uma diferença significativa do modelo linear em relação ao modelo de Rakhmatov-Vrudhula. Para 5 fontes, apenas 4 ou 5 nós sobrevivem para o modelo



(a) (b)

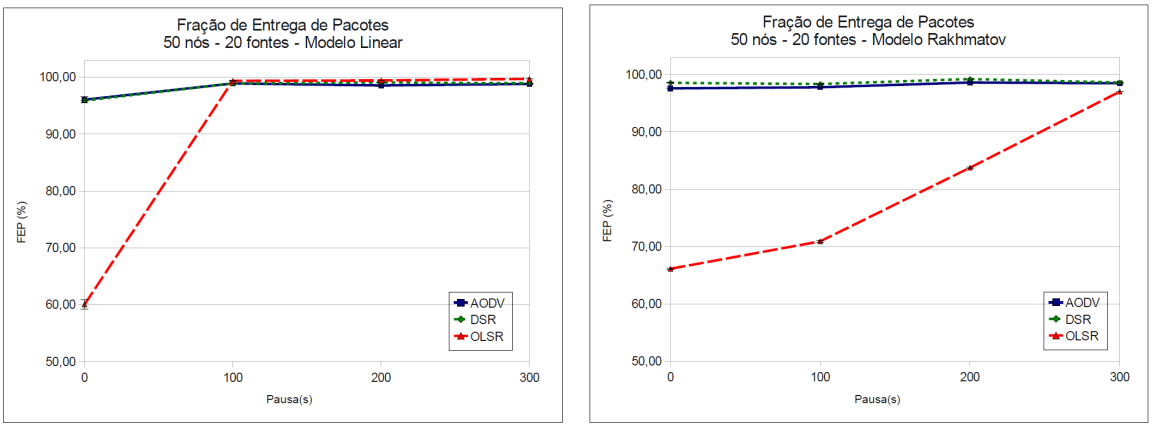
Figura 5.9: Quantidade de nós sobreviventes, variando-se o número de nós fontes, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula

linear, enquanto que, no modelo realista, esse número ultrapassa 25 nós, o que representa um ganho superior a 500%. Observou-se também que, no modelo linear, os poucos nós que permaneceram ativos ao final da simulação, praticamente não tinham capacidade de bateria. Para os dois modelos de bateria, com o aumento do número de fontes, a quantidade de nós sobreviventes cai significativamente nos protocolos reativos, mas com uma certa vantagem do AODV em relação ao DSR. Como o DSR obteve um consumo de energia maior que o AODV, sua quantidade de dispositivos sobreviventes ao final da simulação tende a ser menor, pois o DSR pode capturar e armazenar em seu *cache* mais informações dos pacotes que trafegam na rede, quando este opera em modo promíscuo. Isto resulta em maior consumo de energia e menor número de nós sobreviventes. O OLSR obteve o pior desempenho nos dois modelos de bateria, pois, ao final da simulação todos os nós da rede morreram, logo, apresentou valores maiores de consumo de energia em relação aos demais.

5.1.4 Fração de Entrega de Pacotes

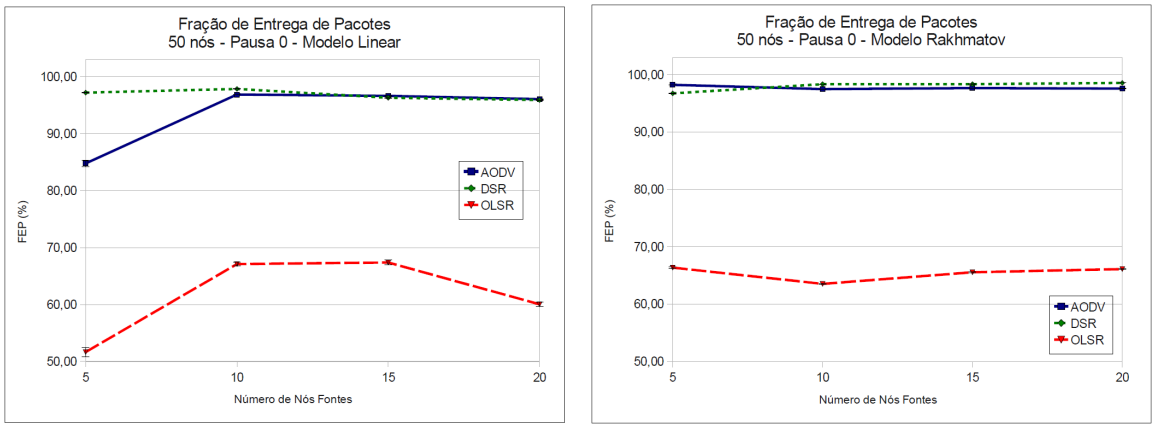
Os resultados referentes à fração de entrega de pacotes (i.e., razão da taxa de pacotes recebidos pelos pacotes enviados), de acordo com o tempo de pausa, para os dois modelos de bateria, estão representados na Figura 5.10. Com base no modelo linear, para um tempo de pausa de 0 segundos, a taxa de entrega do protocolo OLSR caiu significativamente, enquanto nos outros tempos de pausa, a taxa mostrou-se satisfatória, com rendimento próximo ao dos

demais protocolos (i.e., taxas altas, superior a 90%). Por outro lado, com base no modelo de Rakhmatov-Vrudhula, o aumento da mobilidade na rede fez com que a taxa de entrega para o OLSR caísse consideravelmente, em comparação com o AODV e DSR, que mantiveram as taxas altas. Neste caso, nos protocolos reativos, os problemas de falhas de conexão causados pela movimentação dos nós são superados rapidamente, por isso se adaptam melhor aos diferentes níveis de mobilidade na rede, não projudicando tanto a entrega de pacotes.



(a) (b)

Figura 5.10: Fração de entrega de pacotes, variando-se o tempo de pausa, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula



(a) (b)

Figura 5.11: Fração de entrega de pacotes, variando-se o número de nós fontes, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula

Diante desses resultados, nota-se um desempenho bastante inferior do OLSR em relação aos protocolos reativos, pois mostrou-se sensível ao aumento da mobilidade na rede, o que já

era esperado. O alto consumo de energia apresentado por este protocolo favoreceu a redução na taxa de entrega de pacotes, uma vez que os dispositivos selecionados para receberem estes pacotes estariam inacessíveis por esgotamento da capacidade da bateria. No entanto, o OLSR obteve um ganho superior a 10% na entrega de pacotes, na comparação entre os modelos de bateria, para um cenário com 20 nós fontes e mobilidade máxima.

A Figura 5.11 também mostra os resultados da fração de entrega, mas variando-se a quantidade de nós fontes, com base nos modelos de bateria. A Tabela 5.3 exibe os mesmos resultados juntamente com o ganho adquirido com a utilização do modelo de Rakhmatov-Vrudhula. As taxas para os protocolos AODV e DSR mantiveram-se altas, superior a 90%, mas apenas no cenário com 5 fontes o AODV obteve uma entrega menor. Constatou-se novamente um desempenho inferior do OLSR em comparação com os protocolos reativos, isso devido ao alto nível de mobilidade (i.e., tempo de pausa 0 segundos) utlizado neste experimento. Também percebe-se um comportamento instável do OLSR com o aumento do tráfego na rede, ora aumentando, ora diminuindo a taxa de entrega. Essa instabilidade também ocorre com os protocolos reativos, como pode ser visto com mais precisão na Tabela 5.3.

Tabela 5.3: Fração de entrega de pacotes, variando-se o número de nós fontes

AODV			
Fontes	Linear (%)	R-V (%)	Ganho (%)
5	$84,769 \pm 0,580$	$98,211 \pm 0,010$	$15,86$
10	$96,846 \pm 0,035$	$97,473 \pm 0,012$	$0,65$
15	$96,612 \pm 0,054$	$97,651 \pm 0,006$	$1,08$
20	$96,036 \pm 0,028$	$97,571 \pm 0,010$	$1,60$
DSR			
Fontes	Linear (%)	R-V (%)	Ganho (%)
5	$97,188 \pm 0,037$	$96,727 \pm 0,042$	$-0,47$
10	$97,835 \pm 0,035$	$98,324 \pm 0,010$	$0,50$
15	$96,283 \pm 0,081$	$98,331 \pm 0,011$	$2,13$
20	$95,885 \pm 0,053$	$98,552 \pm 0,013$	$2,78$
OLSR			
Fontes	Linear (%)	R-V (%)	Ganho (%)
5	$51,656 \pm 0,801$	$66,376 \pm 0,146$	$28,50$
10	$67,093 \pm 0,329$	$63,543 \pm 0,130$	$-5,29$
15	$67,378 \pm 0,428$	$65,566 \pm 0,081$	$-2,69$
20	$60,033 \pm 0,365$	$66,120 \pm 0,115$	$10,14$

O aumento de carga na rede deveria proporcionar um maior envio de pacotes na rede, mas não necessariamente uma maior taxa de entrega. No modelo linear, observa-se que as maiores taxas de entrega ocorrem nos cenários com 10 e 15 fontes e no cenário com 20 fontes essa métrica sofre uma redução. Isso ocorre porque com 20 fontes os dispositivos ficam inoperantes mais cedo por esgotarem sua bateria e consequentemente tendem a enviar uma menor quantidade de pacotes, que pode ser constatada na Tabela 5.4.

Tabela 5.4: Média do total de pacotes enviados na simulação, variando-se o número de nós fontes

AODV		
Fontes	Linear	R-V
5	$35,80 \pm 0,075$	$3713,00 \pm 0,302$
10	$475,20 \pm 0,103$	$9041,50 \pm 0,824$
15	$468,90 \pm 0,091$	$13647,20 \pm 0,858$
20	$272,80 \pm 0,111$	$15118,10 \pm 1,320$
DSR		
Fontes	Linear	R-V
5	$35,30 \pm 0,021$	$3709,00 \pm 0,521$
10	$472,10 \pm 0,152$	$9013,90 \pm 0,624$
15	$467,80 \pm 0,115$	$13648,00 \pm 0,831$
20	$272,20 \pm 0,079$	$15075,00 \pm 1,373$
OLSR		
Fontes	Linear	R-V
5	$33,10 \pm 0,026$	$3692,90 \pm 0,438$
10	$474,60 \pm 0,163$	$9039,00 \pm 0,521$
15	$470,60 \pm 0,135$	$13647,30 \pm 0,503$
20	$269,60 \pm 0,046$	$15154,20 \pm 0,989$

As altas taxas de entregas não significam desempenhos satisfatórios, pois como representam frações (i.e., pacotes enviados dividido por pacotes recebidos), o envio de pacotes pode ter sido pequeno. Portanto, também é importante analisar a quantidade de pacotes enviados para demonstrar se o desempenho em questão foi favorável ou não. A Tabela 5.4 mostra os resultados dos intervalos de confiança da quantidade média de pacotes enviados pelos três protocolos, variando-se o número de fontes na rede. Verificou-se que os experimentos baseados no modelo linear, apesar de apresentarem uma taxa de entrega equiparada à do modelo realista, obtiveram pequenas quantidade de pacotes enviados, já que no modelo linear, as baterias dos dispositivos descarregam mais rapidamente que no modelo realista,

impossibilitando o envio e recebimento de mais pacotes. Considerando-se a taxa de envio de 4 pacotes por segundo, percebe-se que no modelo linear a quantidade de pacotes enviados foi muito abaixo do esperado, por isso, é importante considerar as mortes dos dispositivos atuantes como fontes de envio. Vale lembrar que após os 44 segundos de simulação, os nós já começam a ficar inoperantes. Nos experimentos com 5 nós fontes, por exemplo, para os três protocolos, apenas um dispositivo consegue enviar pacotes de dados, o que justifica sua baixa quantidade de pacotes enviados.

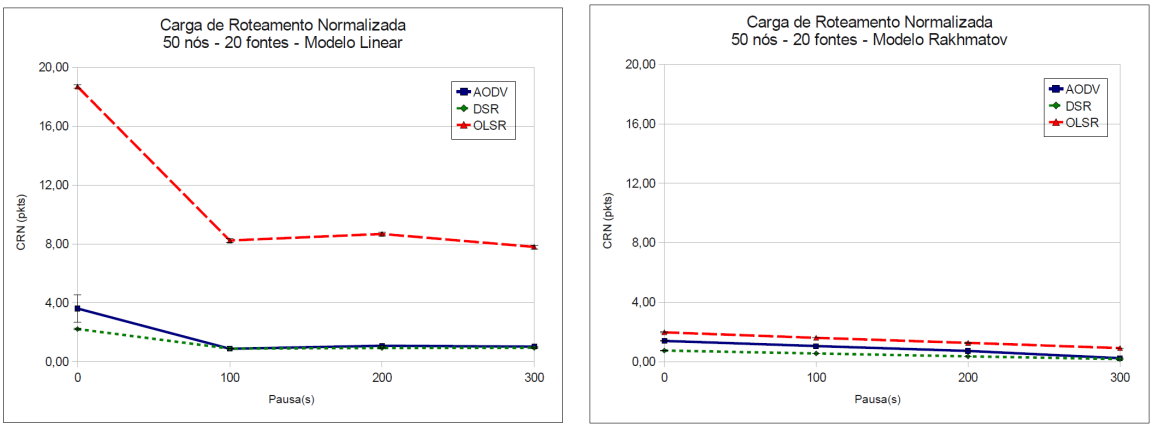
Para o modelo de Rakhmatov-Vrudhula, o envio de pacotes não foi tão afetado pela inacessibilidade dos dispositivos, pois a média de pacotes enviados para os três protocolos cresceu com o aumento do tráfego na rede. Em contrapartida, para o modelo linear, a inacessibilidade dos dispositivos interferiu bastante no envio de pacotes na rede, pois este comportou-se de forma irregular com o aumento do número de nós fontes. Já em relação à entrega de pacotes, o protocolo OLSR foi o mais prejudicado, tanto no modelo linear quanto no realista. Em termos gerais, o DSR apresentou as maiores taxa de entrega de pacotes, mesmo considerando-se que sua diferença em relação ao AODV não foi significativa em alguns cenários.

5.1.5 Carga de Roteamento Normalizada

A métrica carga de roteamento normalizada representa a quantidade de pacotes de roteamento necessários para que cada pacote de dados CBR seja entregue ao destino. A Figura 5.12 mostra a carga de roteamento para os dois modelos de bateria, variando-se o tempo de pausa. Nota-se que a métrica cresce juntamente com a mobilidade, pois a movimentação dos dispositivos torna a rede mais propícia a erros de rota, forçando um maior envio de pacotes de controle.

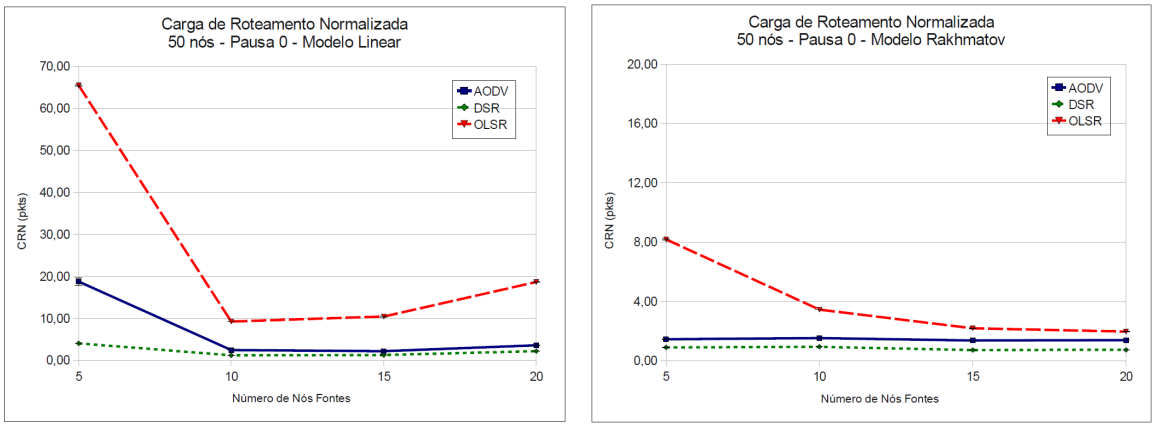
Observa-se na Figura 5.12(a) que o OLSR foi o protocolo que apresentou os maiores valores de carga de roteamento para o modelo linear, devido à sua caracteristica proativa, que naturalmente tende a enviar uma maior quantidade de pacotes de controle (e.g., *HELLO*, *Topology Control* (TC)), e consequentemente sua carga de roteamento será maior. No modelo de Rakhmatov-Vrudhula (Figura 5.12(b)), o OLSR continuou com uma carga de roteamento superior aos protocolos reativos, pelo mesmo motivo retratado no modelo linear. O DSR, por sua vez, apresentou o melhor desempenho, pela não utilização de pacotes *HELLO*. Com

o aumento do tempo de pausa, a sua diferença em relação ao AODV tornou-se menos significativa.



(a) (b)

Figura 5.12: Carga de roteamento normalizada, variando-se o tempo de pausa, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula



(a) (b)

Figura 5.13: Carga de roteamento normalizada, variando-se o número de nós fontes, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula

A Figura 5.13 expõe os resultados em relação à carga de roteamento, variando-se a quantidade de fontes. Do mesmo modo, o modelo linear apresentou valores maiores de carga de roteamento que o modelo realista. Os piores desempenhos ocorreram nos cenários com 5 fontes, pois como a quantidade de pacotes enviados é menor, a quantidade de pacotes recebidos também tende a ser baixa, fazendo a carga de roteamento aumentar. O OLSR,

por exemplo, enviou aproximadamente 2200 pacotes de roteamento para uma quantidade de apenas 33 pacotes recebidos, o que resultou em altíssima carga de roteamento normalizada (superior a 65). O mesmo ocorreu com o AODV, quase 20 pacotes de roteamento por pacote de dados recebido. Por outro lado, o DSR apresentou a menor carga de roteamento para todas as variações de tráfego.

Constatou-se no gráfico da Figura 5.13(a) uma irregularidade no comportamento dos protocolos à proporção que o tráfego de dados aumenta. No cenário com 20 fontes, por exemplo, a carga de roteamento aumentou, em comparação aos cenários com 10 e 15 fontes, porque os dispositivos ficaram inoperantes mais cedo por esgotamento de bateria, enviando e recebendo uma menor quantidade de pacotes, o que elevou a carga de roteamento. Isso ocorre para os três protocolos, porém com menos intensidade para os reativos.

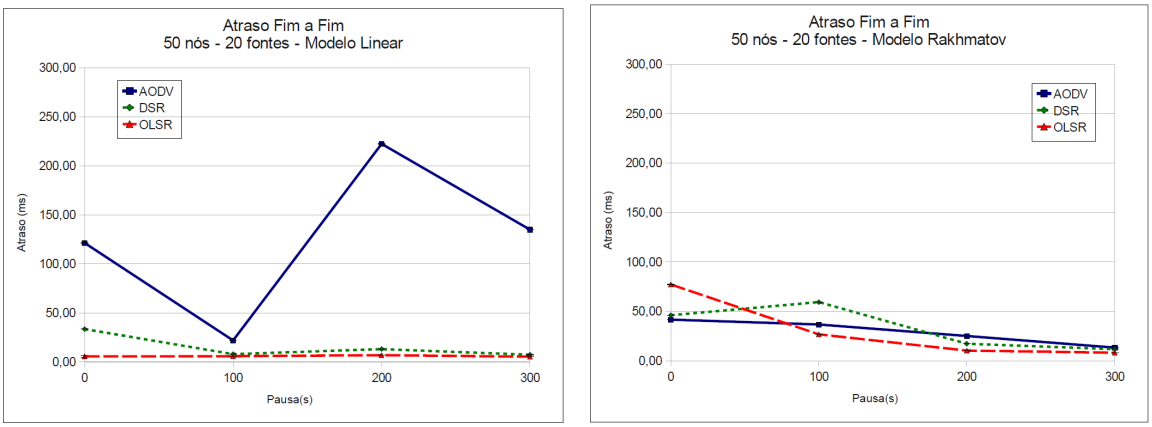
Para o modelo de Rakhmatov-Vrudhula, a recuperação da capacidade da bateria dos dispositivos impede que estes morram mais rápido, mesmo com um tráfego de dados mais intenso. Assim, o recebimento de pacotes tende a ser maior, reduzindo a carga de roteamento normalizada. Por enviar uma menor quantidade de pacotes de controle, o DSR obteve o melhor desempenho entre os protocolos.

5.1.6 Atraso Médio Fim a Fim

Os resultados referentes ao atraso médio fim a fim da rede (i.e., tempo médio necessário para que um pacote de dados seja enviado da origem e chegue ao destino), variando-se o padrão de mobilidade e de tráfego na rede, respectivamente, são evidenciados nas Figuras 5.14 e 5.15.

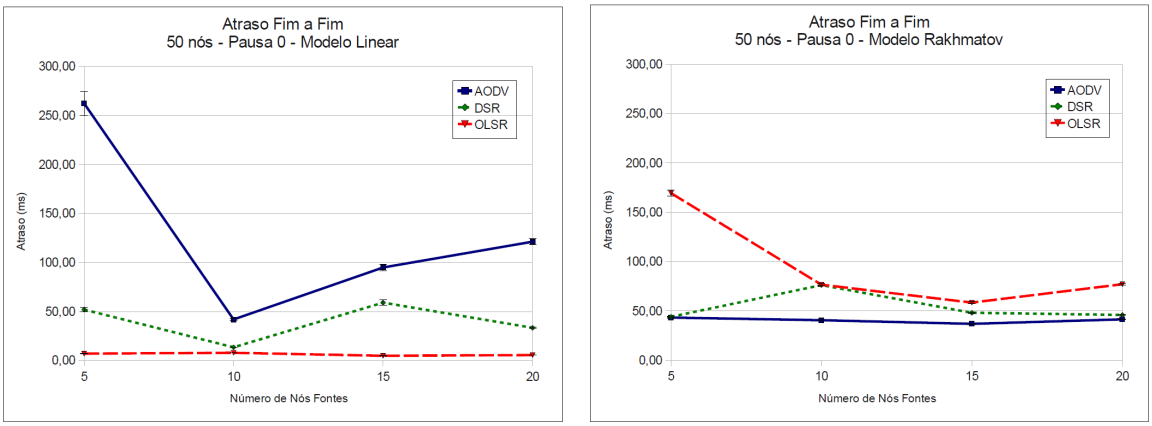
Na comparação dos dois modelos de bateria, verificam-se grandes variações relativas no desempenho dos protocolos. O AODV apresentou valores elevados de atraso para o modelo linear e desempenho satisfatório para o modelo realista, comparado-se com os demais protocolos. Inversamente, o OLSR apresentou baixos valores de atraso médio fim a fim no modelo linear, enquanto que no modelo de Rakhmatov-Vrudhula, seu desempenho foi ruim. Entretanto, no modelo linear, o OLSR deveria ter apresentado maiores valores de atraso que o AODV, pois ele esgota as baterias dos dispositivos mais rapidamente. Uma explicação para este resultado é que, no modelo linear, o esgotamento da bateria não interfe tanto no resultado do atraso médio do OLSR, pois o tempo entre a morte do primeiro nó e a do último

é bem menor comparado ao AODV. Então, o baixo desempenho do AODV resultou da perca de tempo no processo de descoberta de uma nova rota para o destino.



(a) (b)

Figura 5.14: Atraso médio fim a fim, variando-se o tempo de pausa, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula



(a) (b)

Figura 5.15: Atraso médio fim a fim, variando-se o número de nós fontes, para os modelos (a) Linear e (b) Rakhmatov-Vrudhula

A variação da mobilidade e do tráfego na rede também provocou certa instabilidade no desempenho dos protocolos reativos (i.e., atraso ora aumentando, ora diminuindo) para o modelo linear, particulamente no AODV, devido as replicações de pacotes na rede. Neste caso, além do processo de descoberta de rotas, o esgotamento da bateria dos dispositivos também interferiu nos resultados do atraso, pois o tempo entre a morte do primeiro e do

último nó no AODV foi considerável. Comprova-se isto visualizando os resultados de tempo de vida da rede, quando o primeiro nó morre, e 50% dos nós morrem. O mesmo deveria ter acontecido com o DSR, porém nele os dispositivos morreram mais cedo que no AODV e o tempo entre a morte do primeiro e do último nó foi menor, assim, a inacessibilidade dos dispositivos não interferiu tanto nos resultados do atraso.

Com base no modelo de Rakhmatov-Vrudhula, observou-se que o OLSR apresentou um desempenho inferior aos outros protocolos, nos cenários com maior mobilidade (i.e., pausa de 0), um resultado já esperado. Mas, neste caso, isso ocorreu porque o intervalo de tempo em que os dispositivos esgotam suas baterias é superior no OLSR, tendo mais tempo para que a inacessibilidade dos dispositivos interfira sobre a rede, o que provoca erros de transmissão com mais frequência e ocasiona maior atraso para o envio e recebimento dos pacotes.

Vale destacar que, não é possível ter uma boa compreensão do desempenho dos protocolos apenas com base no atraso médio fim a fim. A análise desta métrica deve ser em conjunto com a taxa de entrega de pacotes, pois o protocolo pode até demorar mais para entregar os dados da fonte até o destino, mas enquanto entregar a maior parte deles seu desempenho ainda poderá ser considerado satisfatório. Isso irá depender também do tipo de dados de aplicação que a rede estará trafegando; ou seja, se a aplicação exigir que esses dados sejam entregues o mais rapido possível, com um mínimo de atraso (e.g., uma aplicação de vídeo e áudio), a análise desta métrica será fundamental na avaliação do desempenho dos protocolos.

Pelo que foi observado nos resultados baseados no modelo linear, o protocolo OLSR apesar de ter apresentado baixo atraso médio fim a fim, suas taxas de entrega de pacotes também foram baixas, por isso não obteve bom desempenho. Já o protocolo AODV, que apresentou valores altos de atraso, entregou a maior parte dos pacotes enviados. O DSR apresentou baixos atrasos e altas taxa de entrega. Nos resultados baseados no modelo de Rakhmatov-Vrudhula, o OLSR além de apresentar alto atraso nos cenários com mobilidade máxima, também apresentou baixa fração de entrega de pacotes.

5.2 Relação Entre Métricas

A Relação Métrica x Métrica objetiva analisar o impacto de uma métrica sobre outra para cada protocolo de roteamento. Para as seis métricas utilizadas, teremos um total de 30 rela-

ções possíveis. Uma correlação, ou coeficiente de correlação de Pearson, representa o grau de relacionamento entre duas variáveis aleatórias, ou ainda, entre duas métricas. Esse coeficiente é representado por valores entre -1 e 1, e quanto mais esses valores se aproximam de -1 ou 1 maior é a dependência linear entre as duas variáveis. Uma correlação negativa, significa que as variáveis possuem uma relação inversa; isto é, quando uma aumenta, a outra diminui. O coeficiente de correlação igual a 0, significa que as duas variáveis não dependem linearmente uma da outra; no entanto, uma dependência não linear pode existir.

Tabela 5.5: Matriz de correlação entre métricas para o modelo linear de bateria

Métrica	Protocolo	CE	NS	TVR	FEP	CRN	AM
Consumo de Energia (CE)	AODV	1	$-,640^{*}$	$-,081$	$-,048$	$,025$	$,072$
	DSR	1	$-,584^{*}$	$-,185^{**}$	$-,066$	$-,053$	$,048$
	OLSR	1	^{e}	^{e}	^{e}	^{e}	^{e}
Nós Sobreviventes (NS)	AODV	$-,640^{*}$	1	$,073$	$,052$	$-,026$	$-,043$
	DSR	$-,584^{*}$	1	$,284^{*}$	$,105$	$,209^{*}$	$-,063$
	OLSR	^{e}	1	^{e}	^{e}	^{e}	^{e}
Tempo de Vida da Rede (TVR)	AODV	$-,081$	$,073$	1	$-,902^{*}$	$,923^{*}$	$,055$
	DSR	$-,185^{**}$	$,284^{*}$	1	$-,257^{*}$	$,496^{*}$	$,106$
	OLSR	^{e}	^{e}	1	$-,218^{*}$	$,242^{*}$	$-,163^{**}$
Fração de Entrega de Pacotes (FEP)	AODV	$-,048$	$,052$	$-,902^{*}$	1	$-,973^{*}$	$-,021$
	DSR	$-,066$	$,105$	$-,257^{*}$	1	$-,510^{*}$	$-,094$
	OLSR	^{e}	^{e}	$-,218^{*}$	1	$-,487^{*}$	$-,214^{*}$
Carga de Roteamento Normalizada (CRN)	AODV	$,025$	$-,026$	$,923^{*}$	$-,973^{*}$	1	$,026$
	DSR	$-,053$	$,209^{*}$	$,496^{*}$	$-,510^{*}$	1	$,247^{*}$
	OLSR	^{e}	^{e}	$,242^{*}$	$-,487^{*}$	1	$-,033$
Atraso Médio (AM)	AODV	$,072$	$-,043$	$,055$	$-,021$	$,026$	1
	DSR	$,048$	$-,063$	$,106$	$-,094$	$,247^{*}$	1
	OLSR	^{e}	^{e}	$-,163^{**}$	$-,214^{*}$	$-,033$	1

* Correlação para 99% de confiança

** Correlação para 95% de confiança

^{e} Não foi possível calcular, pois uma das métricas é constante

A Tabela 5.5, referente ao modelo linear, e a Tabela 5.6, referente ao modelo de Rakhmatov-Vrudhula, mostram a matriz de correlação entre todos os pares de métricas avaliadas para os três protocolos de roteamento. Diante disso, adotou-se as seguintes diretrizes [34] para interpretar os valores apresentados nas tabelas: valor entre 0 e $(-)0,29$ significa uma correlação fraca; entre $(-)0,30$ e $(-)0,49$, correlação média; de $(-)0,5$ à $(-)1$, correlação forte. Nota-se que, alguns valores não foram possíves de serem calculados para o

protocolo OLSR, pois nestes casos as métricas consumo de energia (CE) e nós sobreviventes (NS) não sofreram variação nos experimentos.

Com base no modelo linear, identificou-se alguns pares de métricas que apresentaram forte correlação com alto nível de confiança em pelo menos um dos protocolos de roteamento: <consumo de energia (CE), nós sobreviventes (NS)>; <tempo de vida da rede (TVR), fração de entrega de pacotes (FEP)>; <tempo de vida da rede (TVR), carga de roteamento normalizada (CRN)>; <fração de entrega de pacotes (FEP), carga de roteamento normalizada (CRN)>.

Tabela 5.6: Matriz de correlação entre métricas para o modelo de Rakhmatov-Vrudhula

Métrica	Protocolo	CE	NS	TVR	FEP	CRN	AM
Consumo de Energia (CE)	AODV	1	$-1,000^{*}$	$-,531^{*}$	$-,248^{*}$	$,009$	$,068$
	DSR	1	$-,975^{*}$	$,436^{*}$	$,071$	$-,085$	$,191^{**}$
	OLSR	1	^{e}	^{e}	^{e}	^{e}	^{e}
Nós Sobreviventes (NS)	AODV	$-1,000^{*}$	1	$,531^{*}$	$,248^{*}$	$-,009$	$-,068$
	DSR	$-,975^{*}$	1	$-,226^{*}$	$-,084$	$,052$	$-,198^{**}$
	OLSR	^{e}	1	^{e}	^{e}	^{e}	^{e}
Tempo de Vida da Rede (TVR)	AODV	$-,531^{*}$	$,531^{*}$	1	$,301^{*}$	$,363^{*}$	$,152$
	DSR	$,436^{*}$	$-,226^{*}$	1	$-,024$	$-,120$	$,059$
	OLSR	^{e}	^{e}	1	$-,472^{*}$	$,575^{*}$	$,213^{*}$
Fração de Entrega de Pacotes (FEP)	AODV	$-,248^{*}$	$,248^{*}$	$,301^{*}$	1	$-,600^{*}$	$-,417^{*}$
	DSR	$,071$	$-,084$	$-,024$	1	$-,327^{*}$	$-,156^{**}$
	OLSR	^{e}	^{e}	$-,472^{*}$	1	$-,369^{*}$	$-,523^{*}$
Carga de Roteamento Normalizada (CRN)	AODV	$,009$	$-,009$	$,363^{*}$	$-,600^{*}$	1	$,577^{*}$
	DSR	$-,085$	$,052$	$-,120$	$-,327^{*}$	1	$,538^{*}$
	OLSR	^{e}	^{e}	$,575^{*}$	$-,369^{*}$	1	$,468^{*}$
Atraso Médio (AM)	AODV	$,068$	$-,068$	$,152$	$-,417^{*}$	$,577^{*}$	1
	DSR	$,191^{**}$	$-,198^{**}$	$,059$	$-,156^{**}$	$,538^{*}$	1
	OLSR	^{e}	^{e}	$,213^{*}$	$-,523^{*}$	$,468^{*}$	1

* Correlação para 99% de confiança

** Correlação para 95% de confiança

^{e} Não foi possível calcular, pois uma das métricas é constante

Baseado no modelo realista, os pares de métricas que apresentaram forte correlação com alto nível de confiança foram: <consumo de energia (CE), nós sobreviventes (NS)>; <consumo de energia (CE), tempo de vida da rede (TVR)>; <fração de entrega de pacotes (FEP), carga de roteamento normalizada (CRN)>; <fração de entrega de pacotes (FEP), atraso médio fim a fim (AM)>; <carga de roteamento normalizada (CRN), atraso médio fim a fim

(AM)>.

Alguns dos relacionamentos entre essas métricas já foram até mesmo explicadas anteriormente, na análise da seção anterior; e esses coeficientes de correlações ajudam a comprovar isto. Um exemplo óbvio é o par <consumo de energia (CE), nós sobreviventes (NS)>, pois estão diretamente relacionados e quando o cosumo de energia é maior, a tendência é sobreviverem menos dispositivos. Isso é comprovado mais claramente no modelo de Rakhmatov-Vrudhula. Outro exemplo, que foi identificado nos dois modelos de bateria é o par <fração de entrega de pacotes (FEP), carga de roteamento normalizada (CRN)>. Estas métricas estão diretamente ligadas, pois as duas dependem da quantidade de pacotes recebidos. Observou-se, no AODV, uma maior correlação entre estas métricas, constatando-se que sua taxa de entrega de pacotes aumentou com a redução da carga de roteamento.

Capítulo 6

Conclusões e Trabalhos Futuros

No presente trabalho, avaliou-se o desempenho de três protocolos de roteamento *unicast* para redes *ad hoc*: OLSR, AODV e DSR. Esta avaliação diferencia-se de outras, pois baseia-se em um modelo de bateria mais realístico, que considera o efeito de relaxação, permitindo representar a recuperação da capacidade da bateria dos dispositivos da rede. Assim, a utilização do modelo de Rakhmatov-Vrudhula possibilitou uma análise mais acurada do consumo de energia dos protocolos, o qual influencia diretamente em métricas como tempo de vida da rede, taxa de entrega de pacotes, carga de roteamento normalizada e atraso fim a fim.

Vale salientar que, dependendo das condições impostas e dos parâmetros utilizados nas configurações dos cenários (e.g., tempo de simulação, capacidade da bateria, padrão de tráfego, padrão de mobilidade, etc), os desempenhos dos protocolos podem variar de avaliação para avaliação, como referido na literatura.

A comparação entre os modelos convencional e realista resultou em diferenças significativas. Quanto ao consumo de energia, por exemplo, o modelo linear apresentou valores superiores, implicando em morte prematura dos dispositivos e menor tempo de utilização da rede, o que representou um ganho de aproximadamente 630% no tempo de vida, quando adotado o modelo realista. No que diz respeito à fração de entrega de pacotes, os dois modelos de bateria tiveram desempenhos semelhantes para os protocolos reativos, com altas taxa de entrega na maioria dos cenários. Os modelos linear e realista também apresetaram baixa carga de roteamento normalizada para os protocolos AODV e DSR, na maioria dos cenários, mas com um leve aumento nos cenários com maior mobilidade e 5 nós fontes. Em contrapartida, o envio de pacotes de dados e de roteamento no modelo realista foi bem superior.

A métrica atraso médio fim a fim foi a que apresentou maior diferença no desempenho dos protocolos, comparando-se os dois modelos de bateria.

Com base em uma análise geral apenas sobre o modelo linear, constatou-se que sob a configuração de parâmetros utilizada neste trabalho, o OLSR obteve rendimento inferior aos demais protocolos, com exceção da métrica atraso fim a fim. O OLSR, por ser um protocolo proativo e por manter sua tabela de rotas sempre atualizada, inundou mais a rede com pacotes de roteamento (i.e., pacotes de controle *HELLO* e *TC*). Essa caracteristica implicou no mau desempenho deste protocolo nos cenários utilizados, pois há um consumo mais elevado de energia em função dessas transmissões, provocando morte mais rápida dos dispositivos e sua inacessibilidade na entrega de pacotes de dados.

Ainda com referência ao modelo linear, os protocolos AODV e DSR, apresentaram desempenhos relativamente próximos para as métricas consumo de energia, tempo de vida da rede e taxa de entrega de pacotes, sem diferença significativa entre estes, em grande parte dos cenários com pouca mobilidade. A maior diferença foi observada nos cenários com 5 nós fontes e tempo de pausa de 0 segundos, em que o AODV apresentou maior tempo de vida e o DSR, maior taxa de entrega de pacotes. O melhor desempenho do protocolo DSR na métrica carga de roteamento, deve-se à não utilização de pacotes *HELLO*. O OLSR apresentou menor atraso fim a fim, enquanto que o AODV, maior atraso e instabilidade nos resultados.

Em geral, para o modelo de Rakhmatov-Vrudhula, comprovou-se também um desempenho inferior do OLSR em relação aos protocolos reativos, com exceção na métrica tempo de vida da rede, em que apresentou superioridade nos cenários com maior tráfego de dados (i.e., 15 e 20 fontes). O protocolo AODV apresentou valores baixos de consumo de energia, e, diferentemente do OLSR, sofreu menos com a redução da densidade dos nós. Quanto às métricas fração de entrega de pacotes e carga de roteamento normalizada, a variação dos parâmetros tempo de pausa e número de nós fontes causaram pouco impacto no desempenho dos protocolos reativos, que apresentaram altas taxas de entrega (i.e., superiores a 90%) e baixa carga de roteamento. Por outro lado, o impacto sofrido pelo OLSR foi maior na variação do número de fontes. O DSR apresentou valores significativamente maiores para taxa de entrega e menores para a carga de roteamento. Por fim, enquanto que no modelo linear, o OLSR obteve os menores valores de atrasos médio fim a fim, no modelo realista seu desempenho inverteu-se, principalmente nos cenários com maior mobilidade. Apesar de

menor atraso no modelo linear, a taxa de entrega de pacotes do OLSR permaneceu baixa comparado aos protocolos reativos.

Em resumo, a Tabela 6.1 mostra os protocolos que obtiveram melhor desempenho em cada uma das métricas analisadas, para os dois modelos de bateria, linear e Rakhmatov-Vrudhula.

Tabela 6.1: Protocolos que obtiveram melhor desempenho em cada métrica.

Métrica	**Modelo Linear**	**Modelo de Rakhmatov-Vrudhula**
Consumo de Energia	AODV	AODV
Tempo de Vida da Rede	AODV	AODV/OLSR
Quantidade de Nós Sobreviventes	AODV	AODV
Fração de Entrega de Pacotes	DSR/AODV	DSR/AODV
Carga de Roteamento Normalizada	DSR/AODV	DSR
Atraso Médio Fim a Fim	OLSR	OLSR/AODV/DSR

6.1 Trabalhos Futuros

A partir da análise dos resultados deste trabalho foi possível listar algumas sugestões para trabalhos futuros:

- Análise baseada em um modelo realístico de bateria, contemplando um número maior de protocolos de roteamento, incluindo os protocolos *multicast*. Além disso, outras métricas (e.g., vazão da rede) poderiam ser inseridas na análise.

- Análise baseada também em modelos de mobilidade mais realísticos, podendo até mesmo incluir terrenos com obtáculos. Essa análise poderá obter resultados mais precisos, visto que o modelo *Random Waypoint* utilizado neste trabalho não apresenta um comportamento real para um cenário de redes *ad hoc*. Além disso, a análise dos protocolos poderia contemplar um padrão de tráfego mais otimizado e que se aproximasse mais da realidade.

- Testar a estabilidade dos protocolos de roteamento quando forem submetidos ao aumento do número de dispositivos na rede. Neste trabalho foram usados apenas 50 nós, e este número poderia ser aumentado, por exemplo, para 100 e 200 nós.

Bibliografia

[1] Mehran Abolhasan, Tadeusz A. Wysocki, and Eryk Dutkiewicz. A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks*, 2(1):1–22, 2004.

[2] Alberto Bisello, Alessandra Giovanardi, and Gianluca Mazzini. Realistic energy evaluation in ad hoc networks. In *VTC Fall*, pages 31–35. IEEE, 2007.

[3] Azzedine Boukerche. Performance evaluation of routing protocols for ad hoc wireless networks. *Mob. Netw. Appl.*, 9(4):333–342, 2004.

[4] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta G. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *MOBICOM*, pages 85–97, 1998.

[5] Elmano Ramalho Cavalcanti. Avaliação de modelos de mobilidade em redes ad hoc sem fio. Master's thesis, Universidade Federal de Campina Grande (UFCG), Junho 2009.

[6] Ching-Chuan Chiang, Hsiao kuang Wu, Winston Liu, and Mario Gerla. *Routing In Clustered Multihop, Mobile Wireless Networks With Fading Channel*, August 1997.

[7] T. Clausen and P. Jacquet. Optimized link state routing protocol (olsr). rfc 3626. Technical report, IETF MANET Working Group, October 2003.

[8] IBM Corporation. *IBM SPSS Statistics 18*, http://www.spss.com/. 2010.

[9] S. Corson and J. Macker. Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations. rfc 2501. Technical report, IETF MANET Working Group, January 1999.

[10] Samir R. Das, Robert Castaneda, and Jiangtao Yan. Simulation-based performance evaluation of routing protocols for mobile ad hoc networks. *Mob. Netw. Appl.*, 5(3):179–189, 2000.

[11] S. Datta and B. Eksiri. Sensor network routing algorithms for realistic battery models. *in Proceedings of the Workshop on Information Fusion and Dissemination in Wireless Sensor Networks*, 2005.

[12] Isabel Dietrich and Falko Dressler. On the lifetime of wireless sensor networks. *ACM Transactions on Sensor Networks*, 5(1), February 2009.

[13] Laura Marie Feeney and Martin Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. pages 1548–1557, 2001.

[14] Mario Gerla. Emerging applications, design challenges and future opportunities. In *Ad Hoc Networks: Technologies and Protocols*, pages 1–22. Springer, 2004.

[15] Z.J. Haas. A new routing protocol for the reconfigurable wireless network. *In Proc. of 1997 IEEE 6th Intern. Conf. on Universal Personal Communications Record: Bridging the Way to the 21st Century (ICUPC'97)*, pages 562–566, 1997.

[16] M. Handy and D. Timmermann. Simulation of mobile wireless networks with accurate modeling of non-linear battery effects. *Int'l. Conf. Appl. Simulation Model.*, pages 532–537, 2003.

[17] J. Heidemann, N. Bulusu, J. Elson, C. Intanagonwiwat, K. Lan, Y. Xu, W. Ye, D. Estrin, and R. Govindan. Effects of detail in wireless network simulation, 2001.

[18] Kenneth Holter. *Comparing AODV and OLSR*, folk.uio.no/kenneho/studies/essay/essay.html. 2005.

[19] Zhan Huawei and Zhou Yun. *Comparison and Analysis AODV and OLSR Routing Protocols in Ad Hoc Network*, 2008.

[20] Aleksandr Huhtonen. Comparing AODV and OLSR routing protocols, May 2004.

[21] ISI/USC. *The Network Simulator (ns-2)*, http://www.isi.edu/nsnam/ns. 2009.

[22] Teerawat Issariyakul and Ekram Hossain. *Introduction to Network Simulator NS2*. Springer, 2009.

[23] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*, pages 62–68, 2001.

[24] Raj Jain. The art of computer systems performance analysis: Techniques for experimental design, measurement, simulation and modeling (book review). *SIGMETRICS Performance Evaluation Review*, 19(2):5–11, 1991.

[25] S. Jayashree, B. S. Manoj, and C. Siva Ram Murthy. On using battery state for medium access control in ad hoc wireless networks. In Zygmunt J. Haas, Samir R. Das, and Ravi Jain, editors, *MOBICOM*, pages 360–373. ACM, 2004.

[26] David B. Johnson, David A. Maltz, and Yih C. Hu. The dynamic source routing protocol for mobile ad hoc networks (dsr). Technical report, IETF MANET Working Group, February 2007.

[27] M.R. Jongerden and B.R.H.M. Haverkort. Which battery model to use? In *24th UK Performance Engineering Workshop*, number DTR08-9 in Technical Report Series of the Department of Computing, Imperial College London, pages 76–88. Imperial College London, July 2008.

[28] S. Kellner, M.Pink, D. Meier, and E.-O. Blass. Towards a Realistic Energy Model for Wireless Sensor Networks. In *Proceedings of IEEE Fifth Annual Conference on Wireless On demand Network Systems and Services*, pages 97–100, Garmisch-Partenkirchen, Germany, January 2008. ISBN 9781424419586.

[29] Kanishka Lahiri, Sujit Dey, Debashis Panigrahi, and Anand Raghunathan. Battery-driven system design: A new frontier in low power design. In *ASP-DAC '02: Proceedings of the 2002 Asia and South Pacific Design Automation Conference*, pages 261+, Washington, DC, USA, 2002. IEEE Computer Society.

[30] G. Malkin. RIP version 2 carrying additional information. RFC 1388, IETF, January 1993.

[31] Cintia B. Margi and Katia Obraczka. Instrumenting network simulators for evaluating energy consumption in power-aware ad-hoc network protocols. In *MASCOTS '04: Proceedings of the The IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems*, pages 337–346, Washington, DC, USA, 2004. IEEE Computer Society.

[32] J. Moy. RFC 1247: OSPF version 2. RFC 1247, IETF, July 1991.

[33] Shree Murthy and J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mobile Networks and Applications*, 1(2):183–197, June 1996.

[34] Julie F. Pallant. *SPSS survival manual: a step by step guide to data analysis using SPSS*, 2 ed. edition, 2005.

[35] UCLA Parallel Computing Laboratory. *The GloMoSim Simulator*, http://pcl.cs.ucla.edu/projects/glomosim/. 1998.

[36] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 1405–1413 vol.3, 1997.

[37] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing. rfc 3561. Technical report, IETF MANET Working Group, July 2003.

[38] C. E. Perkins, E. M. Royer, S. R. Das, and M. K. Marina. Performance comparison of two on-demand routing protocols for ad hoc networks. *Personal Communications, IEEE*, 8(1):16–28, August 2002.

[39] Charles Perkins and Elizabeth Royer. Ad-hoc on-demand distance vector routing. In *In Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, 1997.

[40] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *SIGCOMM Comput. Commun. Rev.*, 24(4):234–244, October 1994.

[41] Michael Pidd. *Computer simulation in management science*. John Wiley & Sons, Inc., New York, NY, USA, 1984.

[42] Daler Rakhmatov and Sarma Vrudhula. Energy management for battery-powered embedded systems. *ACM Trans. Embed. Comput. Syst.*, 2(3):277–324, 2003.

[43] Daler N. Rakhmatov and Sarma B. K. Vrudhula. An analytical high-level battery model for use in energy management of portable electronic systems. In *ICCAD '01: Proceedings of the 2001 IEEE/ACM international conference on Computer-aided design*, pages 488–493, Piscataway, NJ, USA, 2001. IEEE Press.

[44] Arnold D. Robbins. *GAWK: Effective AWK Programming: A User's Guide for GNU Awk*. 3 edition.

[45] Francisco J. Ros. *UM-OLSR software*, http://masimum.inf.um.es. 2004.

[46] E. M. Royer and Chai-Keong Toh. A review of current routing protocols for ad hoc mobile wireless networks. *Personal Communications, IEEE*, 6(2):46–55, August 2002.

[47] S. Sarkar, T. Basavaraju, and C. Puttamadappa. *Ad Hoc Mobile Wireless Networks: Principles, Protocols and Applications*. Auerbach, 1 edition, October 2007.

[48] Paulo Sérgio Sausen. *Gerenciamento Integrado de Energia e Controle de Topologia em Redes de Sensores Sem Fio*. PhD thesis, Universidade Federal de Campina Grande (UFCG), Julho 2008.

[49] SCT. *The Qualnet Simulator*, http://www.scalable-networks.com/. 2003.

[50] M. A. Spohn, P. S. Sausen, J. R. de Brito Sousa, A. Perkusich, and A. M. N. Lima. Simulation of wireless sensor networks based on a realistic battery model. In *Mobile Telephones: Networks, Applications, and Performance*, pages 211–229. Nova Science Publishers, 2008.

Apêndice A

Script Tcl

Código Fonte A.1: Script Tcl

```
1  # ======================================================================
2  # Define options
3  # ======================================================================
4  set val(chan)       Channel/WirelessChannel ;# channel type
5  set val(prop)       Propagation/TwoRayGround;# radio-propagation model
6  set val(netif)      Phy/WirelessPhy         ;# network interface type
7  set val(mac)        Mac/802_11              ;# MAC type
8  set val(ifq)        Queue/DropTail/PriQueue ;# interface queue type
9  set val(ll)         LL                      ;# link layer type
10 set val(ant)        Antenna/OmniAntenna     ;# antenna model
11 set val(ifqlen)     50                      ;# max packet in ifq
12 set opt(nn)         ""                      ;# number of mobile nodes
13 set opt(sc)         ""                      ;# scenery
14 set opt(cp)         ""                      ;# traffic pattern
15 set opt(out)        ""                      ;# trace out
16 set val(rp)         AODV                    ;# routing protocol (DSR and
       OLSR)
17 set opt(x)          ""                      ;# X dimension of the
       topography
18 set opt(y)          ""                      ;# Y dimension of the
       topography
19 set opt(stop)       ""                      ;# simulation time
20 set opt(energy)     ""                      ;# battery capacity (in Joules
       )
```

```
21 set opt(txp)        ""                          ;# power TX
22 set opt(rxp)        ""                          ;# power RX
23 set opt(idp)        ""                          ;# power IDLE
24
25 # ==========================================================
26
27 proc usage {} {
28     global argv0
29
30     puts "\nusage: $argv0 \[-nn nn\] \[-x x\] \[-y y\] \[-stop stop\] \[
          -energy energy\] \[-txp txp\] \[-rxp rxp\] \[-idp idp\] \[-cp cp\]
           \[-sc sc\] \[-out out\]\n"
31 }
32
33 proc getopt {argc argv} {
34   global opt
35   lappend optlist nn x y stop energy txp rxp idp cp sc out
36
37   for {set i 0} {$i < $argc} {incr i} {
38     set arg [lindex $argv $i]
39     if {[string range $arg 0 0] != "-"} continue
40
41     set name [string range $arg 1 end]
42     set opt($name) [lindex $argv [expr $i+1]]
43   }
44 }
45
46 # ==========================================================
47 # Main Program
48 # ==========================================================
49
50 getopt $argc $argv
51 usage
52
53 #
54 # Initialize Global Variables
55 #
```

```
56
57 set ns_   [new Simulator]
58 set chan  [new $val(chan)]
59 set chan1   [new $val(chan)]
60 set prop  [new $val(prop)]
61 set tracefd      [open $opt(out) w]
62 #$ns_ use-newtrace
63 $ns_ trace-all $tracefd
64
65 Mac/802_11 set dataRate_ 11.0Mb
66
67 # Set up the antennas to be centered in the node and 1.5 meters above it
68 Antenna/OmniAntenna set X_ 0
69 Antenna/OmniAntenna set Y_ 0
70 Antenna/OmniAntenna set Z_ 1.5
71 Antenna/OmniAntenna set Gt_ 1.0
72 Antenna/OmniAntenna set Gr_ 1.0
73
74 # Initialize the SharedMedia interface with parameters to make
75 # it work like the 2.4GHz Lucent WaveLAN DSSS radio interface
76 Phy/WirelessPhy set CPThresh_ 10.0  ;#Limiar de Captura
77 Phy/WirelessPhy set CSThresh_ 1.559e-11 ;#Limiar de Carrie Sense
78 Phy/WirelessPhy set RXThresh_ 3.652e-10 ;#Limiar de Potencia de Recepcao
79 Phy/WirelessPhy set Rb_ 11e6    ;#Largura de Banda
80 Phy/WirelessPhy set Pt_ 0.2818    ;#Potencia de Transmissao (250m)
81 Phy/WirelessPhy set freq_ 2.4e9   ;#Frequencia
82 Phy/WirelessPhy set L_ 1.0    ;#Fator de Perda
83
84 # Set up topography object
85 set topo       [new Topography]
86 $topo load_flatgrid $opt(x) $opt(y)
87 $prop topography $topo
88
89 # Create God
90 set god_ [create-god $opt(nn)]
91
92 #
```

```
93 #  Create the specified number of mobilenodes [$opt(nn)] and "attach"
       them
94 #  to the channel.
95
96 # Configure node
97
98          $ns_ node-config -adhocRouting $val(rp) \
99         -llType $val(ll) \
100        -macType $val(mac) \
101        -ifqType $val(ifq) \
102        -ifqLen $val(ifqlen) \
103        -antType $val(ant) \
104        -propType $val(prop) \
105        -phyType $val(netif) \
106        -channel $chan1 \
107        -topoInstance $topo \
108        -agentTrace ON \
109        -routerTrace ON \
110        -macTrace ON \
111        -movementTrace OFF \
112        -energyModel EnergyModel \
113        -initialEnergy $opt(energy) \
114        -txPower $opt(txp) \
115        -rxPower $opt(rxp) \
116        -idlePower $opt(idp)
117
118   for {set i 0} {$i < $opt(nn) } {incr i} {
119     set node_($i) [$ns_ node]
120     $node_($i) random-motion 0  ;# disable random motion
121   }
122
123 # Modelo de cenario
124 source $opt(sc)
125
126 # Modelo de trafego
127 source $opt(cp)
128
```

```
129 #
130 # Tell nodes when the simulation ends
131 #
132 for {set i 0} {$i < $opt(nn) } {incr i} {
133     $ns_ at $opt(stop).001 "$node_($i) reset";
134 }
135 $ns_ at $opt(stop) "stop"
136 $ns_ at $opt(stop).001 "puts \"NS EXITING...\" ; $ns_ halt"
137 proc stop {} {
138     global ns_ tracefd
139     $ns_ flush-trace
140     close $tracefd
141 }
142
143 puts "Starting Simulation..."
144 $ns_ run
```

Apêndice B

Script Awk

Código Fonte B.1: Script Awk

```
1  #! /usr/bin/awk -f
2  #
3  # Generate reports of trace wireless (old trace):
4  #
5  #
   ------------------------------------------------------------------------

6  function sum (array) {
7      s = 0;
8      for (i in array) {
9    s += array[i];
10     }
11     return s;
12 }
13 #
   ------------------------------------------------------------------------

14 function average (array) {
15     s = 0;
16     items = 0;
17     for (i in array) {
18   s += array[i];
19   if (array[i] != 0.0) items++;
20     }
```

```
21      if (s == 0 || items == 0)
22    return 0;
23      else
24    return s / items;
25  }
26  #
      ------------------------------------------------------------------------

27  function max( array ) {
28      begin=1;
29      for (i in array) {
30    if (begin || array[i] > largest){
31        largest = array[i];
32        begin = 0;
33    }
34      }
35      return largest;
36  }
37  #
      ------------------------------------------------------------------------

38  function min(array) {
39      begin=1;
40      for (i in array){
41    if (begin || array[i] < smallest) {
42        smallest = array[i];
43        begin = 0;
44    }
45      }
46      return smallest;
47  }
48
49  #
      #=========================================================================

50
51  BEGIN {
```

```
  energy_initial = 40.0;
  nn = 50; # Number of nodes
  chave = 0;
  for (i = 0 ; i < nn ; i++){
    temp[i] = energy_initial;
  }

  # Initialize variables
  sends=0;
  recvs=0;
  routing_packets=0.0;
  droppedBytes=0;
  droppedPackets=0;
  highest_packet_id=0;
  soma=0;
  recvnum=0;
}

# ==========================================================================

{
  event = $1;

  if (event == "s" || event == "r" || event == "D" || event == "f") {

    # Example of old format (energy model)
    # 1 2          3   4    5   6 7   8   9   10 11 12  13      14
          15 16    17 18    19 20    21 22     23      24  25   26 27
      28 29 30
    # r 2.112017031 _5_ AGT --- 7 cbr 532 [13a 5 4 800] [energy 0.010000
        ei 0.000 es 0.000 et 0.000 er 0.000] ------- [1:0 5:0 30 5] [1] 2
        2
    event = $1;
    time = $2;
    node = $3;
```

```
83      type = $4;
84      packet_id = $6;
85      pkt = $7;
86      energy = $14;
87      e_idle = $16;
88      e_sleep = $18;
89      e_tx = $20;
90      e_rx = $22;
91
92      # Delete trash
93      sub(/^_*/, "", node);
94      sub(/_*$/, "", node);
95      sub(/^\[/, "", energy);
96      sub(/^\]/, "", e_rec);
97
98      if (type == "MAC" && (pkt == "ACK" || pkt == "CTS" || pkt == "RTS")){
99        mac_energy[node] = mac_energy[node] + (temp[node] - energy);
100       temp[node] = energy;
101     }
102     if (type == "RTR" && (pkt == "AODV" || pkt == "DSR" || pkt == "OLSR"
          || pkt == "ZRP")) {
103       rtr_energy[node] = rtr_energy[node] + (temp[node] - energy);
104       temp[node] = energy;
105     }else{
106     if (pkt == "cbr") {
107       cbr_energy[node] = cbr_energy[node] + (temp[node] - energy);
108       temp[node] = energy;
109     }
110
111     idle_energy[node] = e_idle;
112     tx_energy[node] = e_tx;
113     rx_energy[node] = e_rx;
114
115
116     # Calculate packet delivery fraction - PDF
117
```

```
118     if (( event == "s") &&  ( pkt == "cbr" ) && ( type =="AGT" )) {
            sends++; }
119
120     if (( event == "r") &&  ( pkt == "cbr" ) && ( type =="AGT" ))   {
            recvs++; }
121
122     # Calculate delay
123
124     if ( start_time[packet_id] == 0 )  start_time[packet_id] = time;
125     if (( event == "r") &&  ( pkt == "cbr" ) && ( type =="AGT" )) {
            end_time[packet_id] = time;  }
126           else {  end_time[packet_id] = -1;  }
127
128     # Calculate total overhead
129
130     if ((event == "s" || event == "f") && type == "RTR" && (pkt == "AODV"
             || pkt == "DSR" || pkt == "OLSR" || pkt == "ZRP"))
            routing_packets++;
131
132     # Dropped packets
133
134     if (( event == "D" ) && ( pkt == "cbr" )  && ( time > 0 )){
135          droppedBytes=droppedBytes+$8;
136          droppedPackets=droppedPackets+1;
137     }
138
139     # Find the number of packets in the simulation
140     if (packet_id > highest_packet_id)
141       highest_packet_id = packet_id;
142
143   }
144
145   # Energy (Example: N -t 2.556954 -n 0 -e 0.009899)
146   if (event == "N" ) {
147     N_node_energy[$5] = energy_initial - $7;
148     if ($7 == 0) {
149       if (chave == 0) {
```

```
150           lifetime = $3;
151           firstnode = $5;
152           chave = 1;
153         }
154         dead++;
155         perc = dead/nn;
156         if (perc == 0.2) lifetime20 = $3;
157         if (perc == 0.5) lifetime50 = $3;
158       }
159     }
160
161 }
162
163 ##================================================================
164
165 END {
166     avg_usedenergy1 = average(N_node_energy); # Energy Used
167     alive = nn - dead; # Number of nodes alive
168     avg_rtrenergy = average(rtr_energy); # Energy Routing
169     avg_cbrenergy = average(cbr_energy); # Energy CBR
170     avg_macenergy = average(mac_energy); # Energy MAC
171     avg_idleenergy = average(idle_energy); # Energy IDLE
172     avg_txenergy = average(tx_energy); # Energy TX
173     avg_rxenergy = average(rx_energy); # Energy RX
174
175     total = avg_idleenergy + avg_txenergy + avg_rxenergy;
176     idleenergy = (avg_idleenergy/total)*100;
177     txenergy = (avg_txenergy/total)*100;
178     rxenergy = (avg_rxenergy/total)*100;
179
180     total2 = avg_rtrenergy + avg_cbrenergy + avg_macenergy;
181     rtrenergy = (avg_rtrenergy/total2)*100;
182     cbrenergy = (avg_cbrenergy/total2)*100;
183     macenergy = (avg_macenergy/total2)*100;
184
185
186     for ( i in end_time ){
```

```
187       start = start_time[i];
188       end = end_time[i];
189       packet_duration = end - start;
190       if ( packet_duration > 0 ){
191         soma += packet_duration;
192         recvnum++;
193       }
194     }
195
196     delay=soma/recvnum; #delay end-to-end
197     NRL = routing_packets/recvs;  #normalized routing load
198     PDF = (recvs/sends)*100;  #packet delivery ratio[fraction]
199
200     printf("\n");
201     printf("Used(J): %8.6f\n",avg_usedenergy1);
202     printf("\n");
203     printf("Alive: %d\n",alive);
204     printf("Dead: %d\n",dead);
205     if (lifetime != 0.0) printf("LifetimeFirstDead(s): %8.6f\n",lifetime);
          else printf("\n");
206     if (lifetime20 != 0.0) printf("Lifetime20%%Dead(s): %8.6f\n",lifetime20
          ); else printf("\n");
207     if (lifetime50 != 0.0) printf("Lifetime50%%Dead(s): %8.6f\n",lifetime50
          ); else printf("\n");
208     printf("\n");
209     printf("RTR(%%): %3.2f\n",rtrenergy);
210     printf("CBR(%%): %3.2f\n",cbrenergy);
211     printf("MAC(%%): %3.2f\n",macenergy);
212     printf("\n");
213     printf("Idle(%%): %3.2f\n",idleenergy);
214     printf("Tx(%%): %3.2f\n",txenergy);
215     printf("Rx(%%): %3.2f\n",rxenergy);
216     printf("\n");
217     printf("Send(pkts): %.2f\n",sends);
218     printf("Recv(pkts): %.2f\n",recvs);
219     printf("Routing(pkts): %.2f\n",routing_packets++);
220     printf("PDF(%%): %.2f\n",PDF);
```

```
221     printf("NRL(pkts): %.2f\n",NRL);
222     printf("Delay(ms): %.2f\n",delay*1000);
223     printf("DroppedData(packets): %d\n",droppedPackets);
224     printf("DroppedData(bytes): %d\n",droppedBytes);
225 }
```