Uma Análise do Roteamento em Redes *Ad Hoc*

UFRJ

Projeto submetido para a obtenção do título de

Engenheiro Eletrônico e de Computação

ao Departamento de Eletrônica e de Computação

da Escola Politécnica da UFRJ

por

Bernardo Antunes Maciel Villela

Uma Análise do Roteamento em Redes *Ad Hoc*

Autor:

Bernardo Antunes Maciel Villela

Orientador:

Prof. Otto Carlos Muniz Bandeira Duarte, Dr. Ing.

Examinadores:

Luís Henrique Maciel Kosmalski Costa, Dr.

Marcelo Gonçalves Rubinstein, D.Sc.

Mauros Campello Queiroz, M.Sc.

Departamento de Eletrônica e de Computação - Escola Politécnica - UFRJ

Rio de Janeiro, RJ - Brasil

Dezembro de 2002

A meus pais, Tarciso e Monica,

e a meus irmãos, Daniel, Francisco, Helena e Vicente.

Agradecimentos

A toda a minha família, por todo o amor, sacrifício, orientação e apoio ao longo da minha vida.

A toda a equipe do GTA, em particular ao professor Otto e aos alunos de mestrado Kleber, Eric, Pedro, Aline e Daniel, por toda a orientação, conselhos e amizade.

A todos os meus amigos, em particular os da turma 97/1, que me acompanharam ao longo da minha formação como engenheiro.

Resumo do Projeto Final apresentado ao DEL/UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro Eletrônico e de Computação.

Uma Análise do Roteamento em Redes *Ad Hoc*

Bernardo Antunes Maciel Villela

Dezembro de 2002

Orientador: Otto Carlos Muniz Bandeira Duarte
Departamento: Engenharia Eletrônica e de Computação

Atualmente, o mundo presencia um evidente crescimento dos sistemas de comunicação móvel. Enquanto o paradigma tradicional de comunicação lida com redes fixas, a mobilidade levanta um novo conjunto de questões, técnicas e soluções. Podem-se citar inúmeras aplicações que usufruem da flexibilidade das redes móveis. Mas há também vários tópicos em aberto para serem estudados e resolvidos.

Neste trabalho são analisadas as redes móveis *ad hoc*, isto é, redes sem infra-estrutura, onde os nós móveis podem se comunicar diretamente, se um estiver no raio de alcance do outro, senão o uso de nós intermediários pode ser feito para se alcançar um destino. Um dos problemas mais desafiadores de redes *ad hoc*, focalizados neste trabalho, trata de como se fazer o roteamento dos pacotes. É um problema complexo que os algoritmos tradicionais não resolvem eficientemente ou falham completamente. Por isso novos algoritmos, específicos para redes *ad hoc*, foram propostos, sendo analisado o desempenho dos protocolos DSR (*Dynamic Source Routing*) e AODV (*Ad Hoc On Demand Distance Vector Routing*), que constituem duas das melhores soluções na atualidade nesta área.

Foram feitas simulações no NS (*Network Simulator*), variando-se o padrão de mobilidade e de carga na rede. Por fim, ambos os protocolos foram analisados segundo a taxa de entrega de pacotes, a sobrecarga de roteamento e a otimização do caminho.

Palavras Chave

Redes de computadores

Internet

Redes móveis *ad hoc*

Roteamento

Simulação

Sumário

Lista de Figuras

Capítulo I

Introdução

I.1 Motivação

Atualmente, constata-se um grande crescimento dos sistemas de comunicação móvel. A comunidade científica teve, pois, que buscar técnicas e soluções para os vários questionamentos levantados por este novo paradigma de redes, as redes móveis. Já existem várias aplicações de redes móveis e muitas outras ainda certamente surgirão. Estas vão desde o mundo dos negócios até aplicativos de entretenimento, de veículos, ou mesmo serviços de emergência. Mas há também vários tópicos em aberto, como problemas de interferência, freqüências permitidas, pequena banda passante, bem como problemas relativos à segurança, ao suprimento de energia e ao meio compartilhado [1]. Tais condições abrem muito espaço para a pesquisa e desenvolvimento.

Podem-se dividir as redes móveis em dois grupos: redes com infra-estrutura e redes *ad hoc*. No primeiro grupo, toda comunicação entre nós móveis se faz através de estações de suporte à mobilidade, que podem se comunicar com a parte fixa da rede. Já nas redes *ad hoc* [2], estudadas neste trabalho, não se precisa de nenhuma infra-estrutura: os nós móveis podem se comunicar diretamente, se um estiver no raio de alcance do outro, senão o uso de nós intermediários pode ser feito para se alcançar um destino. A complexidade de redes *ad hoc* é maior porque cada nó precisa implementar mecanismos de acesso ao meio que resolvam problemas característicos do meio sem fio (tais como o problema do terminal escondido ou exposto [1]) e talvez lidar com prioridade para oferecer qualidade de serviço. A grande vantagem de redes *ad hoc* está em oferecer uma alta flexibilidade. Estas aparecem como única solução quando não existe infra-estrutura ou esta for muito cara, não confiável ou demasiado demorada para se construir. Assim, são inúmeras as aplicações para redes *ad hoc*: reuniões não planejadas, espetáculos (que precisam de instalações rápidas), situações de emergência como enchentes, terremotos ou furacões (quando a infra-estrutura pára de funcionar), operações militares, etc.

Um dos problemas mais desafiadores de redes *ad hoc* trata-se de como rotear os pacotes. A complexidade do roteamento aumenta em uma rede *ad hoc* pois todo nó deve agir como um roteador em potencial e a topologia pode mudar dinâmica e constantemente. Além disso há de se considerar problemas de interferência entre nós vizinhos, ocorrência de enlaces assimétricos, uso de banda passante, energia da bateria para a troca de informações de roteamento, etc. Os algoritmos tradicionais, voltados para redes fixas, se mostram claramente ineficientes para resolver este problema por não se preocuparem em solucionar alguns desafios impostos pelo meio sem fio. Por isso novos algoritmos, específicos para redes *ad hoc*, foram propostos, tendo sido estudados e analisados neste trabalho os protocolos *Dynamic Source Routing* (DSR) [3] e *Ad Hoc On Demand Distance Vector Routing* (AODV) [4, 12], que constituem duas das melhores soluções na atualidade, conforme pode ser verificado na literatura científica.

De forma resumida, pode-se adiantar que ambos os protocolos agem sob demanda, devido a limitações de banda e de energia das redes móveis. O DSR caracteriza-se por se basear no roteamento por fonte e uso intensivo de *cache*. Já o AODV se baseia no uso de tabelas mais conservadoras, números de seqüência e mensagens de *hello* (para efetuar a manutenção das rotas).

Tanto o AODV como o DSR foram simulados com a ferramenta de simulação NS [5]. Nele, podem-se variar parâmetros relevantes para uma rede, tais como o tipo de tráfego, tamanho do pacote, o formato e o tamanho da área de simulação, a velocidade dos nós, o tempo de pausa entre cada movimento, o número de nós no cenário, o número de pares fonte-destino e a duração da simulação.

Com a variação destes parâmetros, podemos simular no NS inúmeras configurações de redes. Uma configuração é dada por um cenário de mobilidade, um padrão de comunicação e o algoritmo de roteamento usado e, uma vez simulada, seus resultados são escritos em arquivos. A informação destes últimos é filtrada e analisada, por meio de programas, a fim de se chegar a dados úteis para análise e construção de gráficos.

Assim, para cada algoritmo de roteamento foram tomadas medidas relevantes [6, 7], tais como:

- taxa de entrega de pacotes: razão entre o número de pacotes enviados e recebidos pela aplicação;
- sobrecarga de roteamento:

  em número de pacotes: relaciona o número de pacotes de roteamento enviados durante a simulação;

  em número de *bytes*: relaciona o número de *bytes* de roteamento transmitidos durante o período de simulação, isto é, somam-se o número de *bytes* contidos em pacotes de roteamento com os *bytes* referentes ao cabeçalho de pacotes com dados provenientes da aplicação;
- otimização do caminho: se contabiliza a diferença entre o número de enlaces do caminho ótimo no momento da geração do pacote e o número de enlaces efetivamente percorrido pelo pacote até chegar ao destino.

Tais métricas foram bastante úteis na comparação e análise dos protocolos. Elas ajudaram a verificar o funcionamento dos protocolos, refletindo características específicas de cada um. Assim, o desempenho dos protocolos pôde ser verificado para diferentes padrões de mobilidade e de carga. Além disso, outros fatores importantes, tais como o consumo de energia e de banda na rede receberam devida atenção.

I.1 Motivação

O próximo capítulo desta dissertação discute as características das comunicações sem fio, apresentando suas vantagens e aplicações, bem como dificuldades impostas pelo meio sem fio. O Capítulo III, primeiramente, apresenta as características das redes *ad hoc*, citando algumas aplicações. Em seguida, focaliza-se no roteamento neste tipo de redes, detalhando os desafios a serem vencidos e as categorias de roteamento *ad hoc*. Os Capítulos IV e V explicam o funcionamento dos protocolos DSR e AODV, respectivamente. No Capítulo VI caracteriza-se o simulador de rede utilizado e a sua extensão de mobilidade e redes sem fio. Em seguida, apresenta-se o modelo de simulação usado, explicando-se os parâmetros variados e suas razões de escolha. Há também uma breve explicação dos *scripts* gerados para se fazer as simulações e sua posterior análise. No Capítulo VII, primeiramente, explicam-se as métricas usadas para se avaliar os protocolos. Em seguida, apresentam-se os resultados obtidos acompanhados de sua análise. Por fim, o Capítulo VIII apresenta as principais conclusões deste trabalho, acompanhadas de sugestões para trabalhos futuros.

Capítulo II

Redes Sem Fio

II.1 Introdução

Com o custo cada vez mais baixo e o poder de processamento cada vez maior, os dispositivos sem fio já adquiriram espaço na sociedade. O seu uso vem se popularizando e intensificando. Já fazem parte de nosso cotidiano aparelhos tais como *laptops*, *palmtops*, PDAs (*personal digital assistants*), telefones celulares, *pagers*, sensores ... E novos aparelhos, sistemas e aplicativos certamente ainda estão por vir.

II.2 Aplicações

Primeiramente serão citadas algumas aplicações de redes sem fio [1], como forma de se ilustrar o grande interesse atual por esta área:

- veículos: vários tipos de aplicações podem ser imaginadas para os diferentes meios de transporte. Assim, podem receber informações tais como notícias, músicas, condições da estrada, do tempo, etc. Pode-se obter informação de logística, resultando em economia de dinheiro e tempo. Pode-se usar o sistema GPS (*Global Positioning System*) para localização;

- situações de emergência: uma ambulância que levasse um paciente poderia transmitir informações relevantes para o hospital, de modo a possibilitar toda uma preparação para acolher e tratar este novo paciente. Além disso seria possível consultar especialistas para um diagnóstico mais rápido. Outro exemplo seria o caso de calamidades da Natureza, como por exemplo enchentes, terremotos ou furacões, quando pode-se resultar que apenas os dispositivos sem fio e sem infra-estrutura permaneçam em funcionamento;

- negócios: muitas aplicações podem ser vislumbradas para o mundo dos negócios, que se torna cada vez mais dinâmico e competitivo. Pode-se imaginar assim que uma determinada empresa precise, por exemplo, manter atualizada sua base de dados a todo momento quanto às diversas aplicações que seus diversos agentes espalhados pelo mundo ou por uma região estão realizando;

- substituição de redes cabeadas: redes sem fio se mostram como alternativas interessantes para áreas de difícil acesso, para prédios históricos que não podem ser danificados com cabeamento, ou mesmo eventos, tais como comícios e espetáculos, que precisem de uma rápida instalação;

- entretenimento em geral: pode-se imaginar uma pessoa em um *shopping center*, recebendo informações sobre lojas com descontos, ou ainda um turista que receba detalhes sobre os pontos turísticos pelos quais ele passe. Outra área com muito potencial para aplicações de redes sem fio são os jogos eletrônicos.

II.3 Características das Comunicações Sem Fio

Os engenheiros e pesquisadores já adquiriram muita experiência nas redes tradicionais cabeadas. As diferentes características das comunicações sem fio exigem ainda muito trabalho em busca de um melhor conhecimento e deixam tópicos em aberto, tais como:

- interferência: o cabo coaxial e o par trançado permitem melhor isolamento contra interferência. Já a informação no ar, em redes sem fio, está sujeita a muita interferência, resultando em maiores taxas de erros e, conseqüentemente, maiores taxas de perdas de pacotes;

- regulamentações: a disponibilidade de espectro de freqüência é um recurso escasso, que tem que ser coordenado e regulado. Há que se procurar usá-lo mais eficientemente, através de modulação e multiplexagem;

- banda estreita: a taxa de transmissão de dispositivos sem fio vem aumentando, mas é bem menor do que a encontrada em redes cabeadas, assim faz-se necessário adaptar aplicações que requerem grande banda para ambientes com características diferentes;

- segurança: a interface de rádio é bem mais fácil de ser escutada. Deve-se, pois, usar criptografia, autenticação e outros mecanismos, na tentativa de prover segurança aos dados;

- meio compartilhado: muitos usuários podem disputar o acesso ao meio que é compartilhado, levantando novos problemas e dificultando o provisionamento de qualidade de serviço. Por exemplo, a detecção de colisões torna-se mais difícil no meio sem fio.

Tais particularidades das comunicações sem fio, como será mostrado ao longo deste trabalho, influenciam, de forma decisiva, as características do roteamento em redes sem fio.

Agora que já foram explicadas as características das comunicações sem fio, as quais fornecem as variáveis que delimitam o problema, pode-se prosseguir apresentando as vantagens das redes sem fio e, em seguida, suas desvantagens.

II.4 Vantagens das Redes Sem Fio

Algumas vantagens que se devem destacar são:

- flexibilidade: desprezando efeitos como os de interferência ou obstáculos, nós podem se comunicar dentro de sua área de cobertura, sem restrições de cabos ou de se manter estáticos, podendo ser colocados em qualquer lugar;
- mobilidade: permite a mobilidade dos equipamentos. Na própria mobilidade está intrínseco um grande grau de flexibilidade;
- fácil planejamento: dentro da área de cobertura, dois nós podem se comunicar desde que sigam o mesmo padrão, não sendo necessário um planejamento prévio da disposição de cabos e tomadas;
- redução do tempo de instalação: o fácil planejamento reduz o tempo de instalação, o que é muito interessante para redes temporárias como feiras, exposições etc;
- discreção: pequenos dispositivos em rede podem ser escondidos em bolsos ou em prédios históricos, permitindo, de forma bem discreta, a intercomunicação;
- áreas de difícil cabeamento: é conveniente para a instalação em áreas de difícil ou perigoso acesso, além de áreas onde o cabeamento seria prejudicial como prédios históricos;
- robustez: redes sem fio, principalmente redes *ad hoc*, são capazes de sobreviver a desastres como terremotos ou furacões.

II.5 Dificuldades e Desvantagens das Redes Sem Fio

Agora apresentam-se algumas desvantagens e dificuldades encontradas nas redes sem fio, que devem ser contornadas ou minimizadas:

- baixa qualidade de serviço (QoS): devido à pequena banda passante e altas taxas de erro, as redes sem fio oferecem menor qualidade que as redes cabeadas;
- consumo de energia: o suprimento de energia consiste em um desafio. Afinal, quanto maior o desempenho e quanto mais funcionalidades apresentam os dispositivos, mais

rápido se consomem as baterias. Além disso, a própria transmissão de dados no meio sem fio se mostra bastante dispendiosa em termos de energia. Este problema vem sendo, pois, bastante estudado;

- custo: os dispositivos de redes sem fio atualmente custam mais caro que os das redes cabeadas, mas, em contrapartida, se economiza em cabeamento e conectores;
- interoperabilidade do sistema: atualmente o que se vê é um processo de padronização lento, o que levou ao surgimento de muitas soluções proprietárias, que só funcionam com nós dos mesmos fabricantes;
- segurança: oferecem baixa privacidade, já que o meio é compartilhado, precisando de mecanismos que ofereçam segurança;
- riscos à saúde: um tópico importante de pesquisa trata da influência da comunicação sem fio no corpo humano. Assim, por exemplo, tanto um aparelho de microondas quanto WLANs (*Wireless Local Area Networks*) operam na freqüência de 2,4 GHz, mas a potência da radiação é diferente, sendo de 1 W para WLANs e de 800 W dentro do forno de microondas. Tratam-se, pois, de casos distintos, sendo ainda necessário mais estudos para se entender até que ponto o ser humano é afetado pela exposição por um longo período de tempo a uma baixa potência de radiação;
- restrições legais: os dispositivos devem estar de acordo com as regulamentações nacionais de espectro de freqüências e de potência máxima dos transmissores.

Muitas outras aplicações ainda poderiam ser mencionadas, mas é interessante notar que, ao longo do tempo, os dispositivos de comunicação sem fio vêm se tornando mais potentes, menores e menos pesados, além de conter interfaces mais funcionais para os usuários.

Capítulo III

Redes *Ad Hoc*

III.1 Introdução

Basicamente, podem-se dividir as redes móveis sem fio em redes infra-estruturadas e redes *ad hoc*.

Redes infra-estruturadas como o próprio nome diz são redes que, para seu funcionamento, fazem uso de uma infra-estrutura. Assim uma unidade móvel, ainda que se encontre próxima a uma outra unidade móvel, não se comunica diretamente com esta outra, mas tem de efetuar a comunicação através de um ou mais pontos de acesso, tal como ocorre por exemplo em redes celulares. Estes pontos de acesso, que nada mais são do que estações de suporte à mobilidade, e todo o equipamento que os interconecta caracterizam a parte infra-estruturada da rede. A ligação entre os pontos de acesso é tipicamente feita através de redes cabeadas, mas não necessariamente: um exemplo seria as redes celulares por satélite, onde os satélites constituem os pontos de acesso.

As redes infra-estruturadas possuem maior simplicidade, já que o ponto de acesso concentra a maior parte da funcionalidade da rede, deixando os clientes sem fio bastante simples. O ponto de acesso é responsável por exemplo pelo controle de acesso ao meio e por achar uma rota até o destino, além de poder efetuar outras funções como por exemplo oferecer qualidade de serviço.

Já em redes *ad hoc* não há *a priori* nenhuma infra-estrutura. Se dois nós móveis se encontram dentro do raio de alcance de transmissão, eles podem se comunicar diretamente, senão eles podem ainda se comunicar se houver outros nós móveis dispostos a cooperar na comunicação, que viabilizem uma rota, permitindo à fonte alcançar o destino.

A complexidade de redes *ad hoc* é maior, já que cada nó precisa se preocupar com o controle de acesso ao meio, com o roteamento dos pacotes entre os nós da rede, e, se necessário, com a diferença de prioridades entre tráfegos para oferecer qualidade de serviço. Assim, cada nó torna-se mais complexo precisando implementar mecanismos que exerçam estas funcionalidades. Mas, conforme será explicado neste capítulo, as qualidades de redes *ad hoc*, tais como a grande flexibilidade e robustez apresentadas, fazem destas redes uma opção muito interessante em determinadas situações.

Como exemplo de WLAN (*Wireless Lan*) tipicamente *ad hoc*, temos o *Bluetooth*. Já o IEEE 802.11 e o HIPERLAN 1 (*High Performance Local Area Network*) são tipicamente para redes infra-estruturadas, mas fornecem suporte adicional para redes *ad hoc*.

III.2 Aplicações

Redes *ad hoc* constituem verdadeiros sistemas autônomos, capazes de se auto-organizar na forma de um grafo arbitrário e possivelmente dinâmico. Tais características rapidamente interessaram ao meio científico, militar e empresarial, que logo vislumbraram aplicações para este sistema altamente robusto e flexível:

- negócios e trabalho em grupo: em alguns ambientes de negócios, o trabalho se dá predominantemente fora dos escritórios, logo uma rede *ad hoc* pode ser interessante. Além disso, é sempre plausível que, de vez em quando, ocorram reuniões fora do prédio da empresa, afastadas de uma infra-estrutura previamente instalada, e, possivelmente combinadas de última hora, então soluções *ad hoc* se mostram bem adaptadas;

- aplicações militares: por exemplo, pode ser importante militar e estrategicamente possuir redes em ambientes hostis no caso de uma guerra, onde uma infra-esrutrutura, caso existisse, seria claramente não confiável e, para desenvolver uma nova, tomaria bastante tempo e seria difícil de se manter. Redes *ad hoc* altamente flexíveis, dinâmicas e rápidas de se construir representariam uma solução interessante;

- situações de emergência: uma infra-estrutura (tanto cabeada quanto baseada em pontos de acesso) pode não resistir a calamidades da Natureza, como por exemplo enchentes, terremotos ou furacões. Assim, uma rede de dispositivos sem fio e sem infra-estrutura, que se implemente de forma rápida e confiável, constitui a solução mais robusta para que uma equipe de resgate possa agir;

- veículos: pode-se formar uma rede de veículos, trocando informação de forma a se evitar acidentes, por exemplo. E, em caso de acidente, pode-se acionar a Polícia e uma ambulância;

- entretenimento em geral: jogos eletrônicos que façam uso da flexibilidade de redes *ad hoc*, sendo jogados em dispositivos móveis em um ambiente qualquer, possivelmente fazendo uso de informações de localização geográfica de cada participante como conhecimento relevante para se alcançar o objetivo do jogo.

Outras aplicações poderiam ser ainda citadas e outras mais ainda hão de ser imaginadas.

III.3 Roteamento em Redes *Ad Hoc*

O roteamento em redes *ad hoc* possui claramente características particulares, de forma que os algoritmos de roteamento tradicionais de redes cabeadas não se aplicam eficientemente aos requisitos mais complexos das redes *ad hoc*. Mesmo os algoritmos de roteamento de redes celulares também não se aplicam, pois nestas as estações bases são responsáveis pelo roteamento. Se uma unidade móvel destino se encontra em uma determinada célula, então se roteia o pacote até a estação base desta célula que por *broadcast* consegue entregar pacotes a qualquer dispositivo móvel presente em sua célula. Já numa rede *ad hoc*, se um nó não estiver ao alcance direto de outro para se fazer a comunicação entre os dois terá que se fazer uso de informações de roteamento para se achar o melhor caminho, em um ambiente no qual cada nó pode agir como roteador: não há estações base que concentram a inteligência do sistema. Assim, cada nó, por meio de algoritmos distribuídos, troca informações e decide se é possível alcançar um determinado destino e, se houver mais de um caminho, determina o melhor.

III.3.1 Dificuldades e Desafios do Roteamento em Redes *Ad Hoc*

O uso de roteamento é certamente necessário: afinal por razões de potência de transmissão, sem contar o tempo de vida da bateria, trata-se de um caso bastante particular uma rede em que todos nós conseguem se alcançar diretamente, o que dispensaria o roteamento.

A seguir, apresentam-se alguns desafios que dificultam o roteamento em redes *ad hoc* e, conseqüentemente, influenciam o projeto de novos algoritmos de roteamento específicos para estas redes:

- topologia dinâmica: em redes *ad hoc*, nós podem se mover arbitrariamente e com diferentes velocidades, sendo difícil, em muitos casos, prever um padrão de movimento. As tabelas de roteamento têm que se esforçar para, de alguma maneira, refletir, rápido o suficiente, tais mudanças para poder retratar aceitavelmente a topologia atual da rede. Além disso, a simples presença de obstáculos (por exemplo, paredes ou ônibus em uma rua) pode afetar momentaneamente a topologia;

- escassez de banda e de energia: uma maneira de se poder acompanhar uma topologia que mude rapidamente seria trocar mensagens com maior freqüência, mas não se pode esquecer que em redes sem fio, a banda disponível é escassa, logo se no meio veicularem muitas mensagens de roteamento, haverá pouca banda para que as aplicações possam usar. Além disso, para se transmitir mensagens de roteamento se gasta energia, outro recurso escasso, logo pode se tornar bastante dispendioso um grande número de mensagens de roteamento. Além disso, o nó intermediário tem que aceitar gastar seus recursos para fazer o intermédio da comunicação entre fonte e destino;

- enlaces assimétricos: ao contrário do que ocorre em redes cabeadas, o fato de uma estação A receber pacotes de uma estação B não significa necessariamente que B consegue igualmente receber pacotes de A. O enlace reverso pode ser mais fraco, mais forte, de igual intensidade ou mesmo não existir. Isto complica o roteamento de tal maneira que mesmo em redes *ad hoc* muitos protocolos evitam este problema, trabalhando apenas, ou preferencialmente, com rotas cujos enlaces sejam simétricos;

- interferência e ruído: no ar uma interferência pode afetar gravemente comunicações. Assim, duas transmissões muito próximas podem destruir-se mutuamente ou uma mais forte destruir a mais fraca. Mesmo as condições ambientais podem mudar e, assim, enlaces ficarem mais fracos ou deixarem de existir, trocando a topologia.

Precisa-se então conceber algoritmos de roteamento que usem eficientemente a banda e a energia dos nós, chegando a rotas válidas, de forma rápida e estável. Para obter melhores resultados, um protocolo de roteamento pode fazer uso de informações, tais como conectividade e interferência, provenientes das camadas inferiores.

III.3.2 Categorias de Protocolos de Roteamento de Redes *Ad Hoc*

Os protocolos de roteamento de redes *ad hoc* podem ser categorizados primariamente como pró-ativos ou reativos (sob demanda). Nos pró-ativos, de forma semelhante aos protocolos para redes fixas, cada nó a todo momento possui informação em sua tabela referente a todos os possíveis destinos. Assim, quando um pacote deve ser enviado a um destino, a rota já é conhecida e pode ser usada imediatamente. Já os protocolos reativos agem sob demanda, isto é, uma rota para um determinado destino é descoberta apenas quando for necessário enviar-lhe um pacote.

Supondo-se uma troca de mensagens de roteamento com freqüência suficientemente elevada, de forma a se garantir o conhecimento de rotas válidas em cada nó, e uma rede com banda suficientemente elevada para que não haja congestionamento, os protocolos pró-ativos, como é fácil perceber, requereriam o atraso mínimo para se enviar um pacote, já que possuem previamente a rota. Isto pode ser importante para uma determinada aplicação. Mas esta aplicação tem que acatar os prejuízos que são consideráveis em termo de consumo de energia e de banda, dois recursos escassos. Assim, um tráfego proporcionalmente elevado de roteamento faz diminuir a vazão efetiva, além de poder acarretar congestionamento e descarte de pacotes. Em última análise, nesta situação não ideal (como a suposta no início do parágrafo), o congestionamento poderia vir a degradar o funcionamento correto do roteamento, seu tempo de resposta e a entrega de pacotes ao destino. Por tudo isso, a alternativa de protocolos reativos vem se mostrando uma solução mais eficiente. Nesta se leva em conta que determinados nós ou mesmo considerável parte da rede podem passar bastante tempo sem se comunicar, logo é mais eficiente gastar banda e energia apenas para descobrir rotas necessárias. Ora, é inútil descobrir uma rota que não será usada em pouco tempo, pois dependendo do dinamismo da rede, esta rota possui grandes chances de já ter sido modificada no momento de seu uso realmente. Mas esta abordagem reativa acarreta maior latência, pois caso não se conheça previamente a rota para um determinado destino, um procedimento de descoberta de rota terá de ser efetuado. Outra possibilidade que pode se mostrar interessante é usar uma abordagem híbrida, que tenta fazer um uso conveniente dos benefícios das duas abordagens.

A título de exemplo, são citados alguns protocolos segundo cada abordagem [8, 9]:

- protocolos pró-ativos: DSDV (*Destination-Sequenced Distance Vector*), OLSR (*Optimized Link State Routing*) e WRP (*Wireless Routing Protocol*);

- protocolos sob demanda: DSR (*Dynamic Source Routing*), AODV (*Ad Hoc On-demand Distance Vector*) e TORA (*Temporally Ordered Routing Algorithm*);
- híbridos: ZRP (*Zone Routing Protocol*), FSR (*Fisheye State Routing*) e LANMAR (*LandMark Routing*).

Vale ainda mencionar que existem protocolos que se aproveitam de informações de localização para se descobrir as rotas. Alguns exemplos destes protocolos são: GPSR (*Greedy Perimeter Stateless Routing*), LAR (*Location-Aided Routing*), DREAM (*Distance Routing Effect Algorithm for Mobility*) e RDMAR (*Relative Distance Micro-discovery Ad hoc Routing*).

Há ainda outros protocolos que usam outras métricas, como por exemplo o PAR (*Power Aware Routing*), que usa a potência como métrica para se achar o melhor caminho.

Nos próximos capítulos serão descritos mais detalhadamente os dois protocolos escolhidos para este projeto: o DSR e o AODV. Esta escolha se deve a vários motivos: conforme foi pesquisado na literatura científica, estes são ainda hoje dois dos melhores protocolos dentre a enorme gama de algoritmos existentes. Eles já passaram por uma série de alterações, buscando-se o aperfeiçoamento e estabilidade, corrigindo-se erros e chegando-se a melhores valores e parâmetros. E continuam sendo trabalhados e melhorados até hoje: as suas últimas especificações são bem recentes. Além disso, estão entre os mais citados na literatura científica e muitos de seus conceitos servem como base para outros algoritmos criados depois. E, por fim, estão implementados no simulador de redes NS [5].

Capítulo IV

DSR - *Dynamic Source Routing*

IV.1 Introdução

O Departamento de Ciência da Computação da *Rice University*, por meio do projeto MONARCH (*MObile Networking ARCHitectures*) [10] vem contribuindo bastante na área de redes móveis e sem fio. Este grupo está trabalhando no projeto de protocolos e interfaces, implementação, avaliação de desempenho e validação em áreas que vão desde a camada de enlace até a camada de apresentação, no padrão OSI (*Open System Interconnection*). Pretende-se permitir a comunicação de dispositivos móveis entre si e com outros fixos, de forma transparente e adaptativa, fazendo um uso eficiente dos recursos da rede. Duas de suas grandes contribuições na área de redes *ad hoc*, que devem ser ressaltadas, são o desenvolvimento do protocolo DSR e das extensões no NS que permitiram a simulação de redes *ad hoc* neste simulador.

Para se compreender melhor o DSR [3], deve-se ter bem claro em mente seus objetivos de projeto: ele busca uma solução ao mesmo tempo simples e eficiente para redes *ad hoc*, que, fazendo uso de uma sobrecarga de roteamento muito baixa, permita que rotas sejam descobertas com reações rápidas a possíveis mudanças na topologia da rede. Assim, além de utilizar o roteamento sob demanda, o DSR usa o roteamento por fonte, isto é, a fonte coloca, em cada pacote que envia, o caminho completo e ordenado que o pacote deve percorrer até se chegar ao destino. Esta aposta se justifica: o roteamento por fonte dispensa a necessidade de informações atualizadas de roteamento nos nós intermediários através dos quais o pacote passa em seu caminho até o destino e também constitui uma maneira trivial de se evitar *loops*.

Quando um nó fonte deseja enviar pacotes a um nó destino, antes de mais nada, uma rota, caso não seja previamente conhecida, precisa ser descoberta e, depois, enquanto se usa uma dada rota para o envio de pacotes se verifica se esta rota permanece válida, caso contrário o nó fonte deve usar uma outra rota já conhecida ou começar uma nova descoberta de rotas. Assim, caracterizam-se os dois mecanismos básicos do DSR: a descoberta de rotas e a manutenção de rotas. No DSR estes dois mecanismos agem completamente sob demanda, objetivando redução da sobrecarga de roteamento: não há necessidade de que qualquer tipo de pacote de qualquer camada seja trocado periodicamente na rede. Como conseqüência, o DSR não usa a distribuição periódica de informações para a atualização de tabelas de roteamento, como fazem os protocolos de roteamento pró-ativos, nem implementa qualquer mecanismo de mensagens periódicas para verificar-se a validade de um enlace, como faz por exemplo a implementação do AODV com o uso de mensagens de *hello*, conforme será visto no Capítulo V. Portanto, no DSR a sobrecarga de roteamento pode mesmo chegar a zero, quando nenhum nó deseja transmitir pacotes ou quando os nós permanecem aproximadamente parados e as rotas já foram descobertas e se encontram estáveis. Se os nós começam a se mover mais rápido, a sobrecarga de roteamento aumenta unicamente o suficiente para responder corretamente à demanda das rotas atualmente em uso.

Vale ressaltar ainda que o DSR, pela maneira como foi implementado o seu mecanismo de descoberta e manutenção de rotas, suporta o uso de enlaces unidirecionais e rotas assimétricas, funcionalidade que o AODV não oferece (seu projeto se baseia em enlaces simétricos). Entretanto, muitos dos principais protocolos de acesso ao meio para redes sem fio, tais como IEEE 802.11, MACA e MACAW, requerem o uso de enlaces bidirecionais, para a troca (opcional para o IEEE 802.11) de pacotes RTS (*Ready To Send*) e CTS (*Clear To Send*) para resolver o problema do terminal escondido e, no caso do IEEE 802.11, este requer ainda ACKs da camada enlace. Quando usado em cima de protocolos de acesso ao meio como estes, o DSR pode aproveitar esta informação e usá-las para otimizações, tais como se existe uma rota da fonte para o destino, pode-se concluir que também existe uma do destino para a fonte.

IV.2 Detalhes do Funcionamento do DSR

IV.2.1 Descoberta de Rota

Quando um nó fonte **A** quer enviar um pacote a um nó destino **D**, ele procura a melhor rota em um *cache* de rotas, denominado "*Route Cache*", que armazena rotas anteriormente aprendidas seja pelo processo de descoberta de rotas anteriores ou seja por escuta em modo promíscuo. Este *cache* é continuamente atualizado e se uma entrada não é usada durante um certo tempo *RouteCacheTimeout*, ela é apagada. No caso de escuta em modo promíscuo, se desabilita a filtragem de pacotes e se faz com que o protocolo de rede receba todos os pacotes que chegam a sua interface de rede. Destes pacotes, podem-se aprender informações potencialmente úteis, sem aumentar a sobrecarga de roteamento, o que é muito interessante devido à escassez de banda. A quantidade de informação aprendida depende logicamente de se os enlaces são unidirecionais ou bidirecionais.

Se a fonte previamente já havia aprendido o caminho para o destino, então para encaminhar-lhe seus pacotes, basta colocar todo os nós do caminho de **A** até **D** no cabeçalho dos seus pacotes (roteamento por fonte). Mas se a fonte ainda não conhece nenhuma rota que chegue até o destino, então faz-se necessário efetuar o procedimento de descoberta de rota, que se especifica a seguir, com a ajuda da Figura IV.1.

Para a descoberta de rota, a fonte inunda a rede com uma requisição de rota para dinamicamente achar uma rota para o destino. Assim, inicialmente a fonte envia uma mensagem chamada *Route Request* a todos os seus vizinhos (**B**, **C** e **E** na figura) por *broadcast*, que, caso não sejam o destino ou não tenham em seu *Route Cache* uma rota

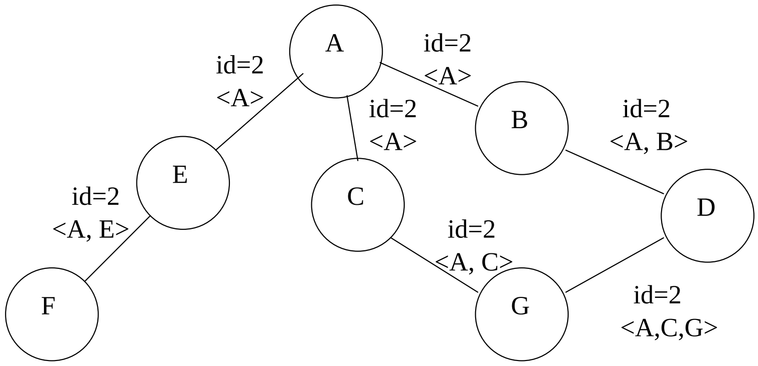


Figura IV.1: Inundação de mensagens de pedidos de rota (RREQs).

para o destino, repassarão a todos os seus vizinhos sucessivamente, implementando assim a inundação da rede. Cada *Route Request* carrega um identificador para o nó fonte, outro para o destino e outro para a mensagem. Além disso, carrega uma lista de todos os nós pelos quais se passou até se chegar ao nó atual. Na figura, evidencia-se esta lista do caminho parcial e o identificador da mensagem, que no caso vale dois. Mostra-se todo o caminho que o *Route Request* seguiria na rede, isto é, ele inundaria a rede passando por todos os nós alcançáveis desde **A**.

Quando um nó que não seja o destino recebe um *Route Request*, ele verifica pelo identificador de fonte, de destino e de mensagem se ele não transmitiu recentemente esta requisição e só repassará aos vizinhos caso seja uma nova requisição (e caso não conheça uma rota para o destino). Isto é um mecanismo para se evitar sobrecarregar a rede com um número excessivo de mensagens de roteamento. Ora, imagine que o nó **B** acabou de inundar seus vizinhos com a requisição de rota que vem de **A**. Logo em seguida, ele recebe de um vizinho a mesma requisição de rota (sabe-se que é a mesma por causa desses três identificadores). Ora, se ele voltasse a inundar a rede com esta nova requisição ele estaria consumindo a banda para achar caminhos piores, o que é indesejável. Caso um nó repasse a requisição aos vizinhos, ele acrescenta a si próprio à lista do caminho parcial percorrido pelo *Route Request*. E assim, a requisição vai se espalhando pela rede.

Quando a requisição finalmente chega ao destino **D**, este confere em seu *Route Cache* para ver se conhece um caminho para a fonte para responder-lhe com o caminho por meio de um *Route Reply*. Caso conheça previamente, é só enviar a resposta. Caso contrário, o destino começa um novo *Route Request* para descobrir um caminho até a fonte. Para evitar uma possível recursividade infinita, o *Route Reply* vai contido no *Route Request* sob a forma de *piggyback*. Este procedimento ocorre no caso geral em que existem enlaces unidirecionais. Desta forma, o DSR consegue se adaptar bem ao roteamento assimétrico, o que não se consegue com o AODV. Mas conforme explicado anteriormente, quando o DSR se encontra em cima de protocolos de acesso ao meio como o IEEE 802.11, o MACA e o MACAW, que requerem enlaces bidirecionais, o roteamento torna-se bem mais simples e o destino pode enviar a resposta à fonte usando o caminho reverso do *Route Request*.

Alternativamente, quando um nó intermediário recebe um *Route Request* e percebe que conhece um caminho até o destino, se nesse caminho não houver nenhum nó "duplicado", um nó que já constava do caminho parcial no *Route Request*, então este nó não precisa repassar a requisição para seus vizinhos, lhe basta concatenar este caminho parcial com o seu caminho até o destino e enviar um *Route Reply* para a fonte como explicado acima.

O DSR permite ao mecanismo de descoberta de rotas usar um limite de saltos para limitar o alcance da inundação de *Route Requests*. Este limite é implementado com o campo TTL (*Time To Live*) do IP. Assim, a cada nó que passa o *Route Request*, o valor do TTL é decrementado de uma unidade. Se este valor chegar a zero antes de chegar ao destino o pacote é descartado. Usando este limite, podem-se implementar alguns algoritmos diferentes para a inundação. Um exemplo seria primeiro enviar um *non-propagating Route Request* com TTL valendo um. Assim, se verifica inicialmente se o destino é um vizinho ou se um vizinho sabe uma rota para o destino (neste caso o *Route Cache* do vizinho é quase uma extensão do *cache* da fonte). Se com este mecanismo inicial se conseguir uma rota para o destino, ter-se-á uma grande economia de banda ao se evitar uma inundação de mensagens. Caso não se obtenha a rota, faz-se a inundação da rede. Uma outra possibilidade seria implementar um mecanismo de "anel expansivo". Nele, primeiramente tenta-se um *Route Request* com TTL igual a um, se não conseguir uma rota, tenta-se novamente com TTL igual a dois e assim vai dobrando o TTL sucessivamente até se conseguir uma rota. Mas com isto, a latência média para se descobrir uma rota aumenta.

Um pacote que desencadeou o mecanismo de descoberta de rotas é guardado em um *buffer* chamado de *Send Buffer*, enquanto espera por uma rota até o destino. Enquanto ali espera, este pacote deve, de tempos em tempos, desencadear um novo processo de descoberta de rotas. Assim, pode-se implementar um mecanismo de *exponential back-off* para o tempo entre sucessivas tentativas de descobertas de rota. Depois de um certo *timeout*, o pacote é descartado do *Send Buffer*, uma vez que pode ocorrer que o nó destino se encontre em uma partição da rede inalcançável a partir do nó origem.

Um outro mecanismo previsto é o de *"gratuitous" Route Reply*. Se um nó **B** por exemplo, ao fazer a escuta em modo promíscuo, escuta um pacote sendo encaminhado da fonte até o destino, de tal maneira que **B** faz parte do caminho pelo qual ainda falta ao pacote passar, mas **B** não é o próximo nó do caminho, fica evidente a existência de um possível atalho. **B** deve então enviar à fonte um *"gratuitous" Route Reply* com o caminho da fonte até o nó de quem **B** escutou concatenado com o resto do caminho de **B** até o destino.

IV.2.2 Manutenção de Rota

Quando pacotes estão sendo enviados da fonte até o destino, cada nó ao longo deste caminho é responsável por verificar se o pacote foi recebido pelo próximo nó. Assim, no caminho **A**, **B**, **D**, na Figura IV.1, **A** é responsável pelo recebimento de **B**, assim como **B** é responsável pelo recebimento de **D**. Um nó deve retransmitir o pacote até um número máximo de tentativas até que se obtenha a confirmação de recebimento. Em muitos casos, este mecanismo de confirmação é facilmente implementado. Por exemplo, se o DSR estiver em cima do IEEE 802.11, podem-se aproveitar os ACKs da camada de enlace, definidos por este protocolo. Pode-se imaginar um ACK passivo também, no qual **A** sabe que **B** recebeu pois escutou a transmissão de **B** para o próximo nó. Se estes mecanismos não estiverem disponíveis, serão necessários ACKs em cada enlace implementados especificamente pelo DSR, o que consumiria mais banda.

Se um nó não recebe a confirmação do próximo nó após um número máximo de tentativas, ele interpreta que o enlace caiu. Se este nó for o nó fonte, ele confere em seu *Route Cache* se conhece um outro caminho, senão começa uma nova descoberta de rota para o destino. Além disso, retira este caminho inválido de seu *Route Cache*, bem como outras rotas que usem este enlace. Se o nó que constatar a queda de um enlace for um nó intermediário, então ele envia uma mensagem *Route Error* para a fonte, através de uma rota contida em seu *cache* ou proveniente de uma nova descoberta de rota (caso não possua rota até a fonte no *cache* e não se possa garantir que os enlaces são todos bidirecionais, de forma a permitir-se o uso da rota reversa deste nó até a fonte). Nesta mensagem de *Route Error* identifica-se o enlace que não é mais válido, permitindo à fonte remover este enlace de seu *cache* e truncar neste ponto todas as rotas contendo este enlace. Se a fonte não conhecer um caminho alternativo, ela terá que novamente começar uma descoberta de rota. Além disso, este nó intermediário que enviou o *Route Error* deve consultar o seu *Route Cache* e ver se conhece uma outra rota até o destino, pois caso conheça ele deve efetuar um *packet salvaging*, isto é, ele deve enviar esta mensagem até o destino através da sua rota, em vez de simplesmente descartá-la. Além disso, ele deve incrementar um contador referente ao número de vezes em que um pacote foi salvo, para evitar *loops*.

Vale mencionar também que após o recebimento de uma mensagem de *Route Error*, caso se faça necessário começar uma nova descoberta de rotas, este novo *Route Request* leva consigo o *Route Error*. Assim, os outros nós podem atualizar seu *Route Cache* retirando as entradas que usam o enlace que caiu e se evita que a fonte receba como resposta novamente um caminho que use este enlace.

Capítulo V

AODV - *Ad hoc On-demand Distance Vector Routing*

V.1 Introdução

O AODV foi projetado para uso em redes *ad hoc*, pretendendo-se adaptar rápida e dinamicamente às variações nas condições de enlace da rede, achando rotas de forma a evitar-se uma sobrecarga de banda e mesmo de memória e processamento nos nós na atividade de roteamento. A equipe de autores mantém um bom sítio [11] na Internet com informações sobre o AODV, sobre sua especificação atual e versões antigas, apresentações anteriores do protocolo e artigos relacionados. Ele ainda apresenta páginas com implementações do protocolo e com códigos para simuladores, além de possibilitar ao interessado aderir à lista de *e-mails* sobre o AODV.

O AODV, assim como o DSR, implementa um sistema de aquisição de rotas sob demanda, com o mesmo intuito de se economizar banda e bateria. Um nó não precisa conhecer a rota até um outro nó até que se faça necessária a comunicação entre eles. Neste caso, se um nó não conhece rota válida para o destino (isto é, apenas quando necessário), faz-se o procedimento de descoberta de rota. Para a manutenção de rota, o AODV oferece duas possibilidades para se detectar a queda de um enlace. Primeiramente, de forma independente da camada inferior, o AODV pode implementar um mecanismo de detecção de vizinhança, no qual se usa o envio periódico de mensagens de *broadcast* local, chamadas *hello*, para se confirmar a conectividade local entre vizinhos. O não recebimento de uma mensagem de *hello* de um antigo vizinho durante um certo período indica a queda de um enlace. De forma alternativa, com o intuito de se economizar banda, podem-se usar informações provenientes da camada de acesso ao meio (por exemplo IEEE 802.11), para se detectar a queda de um enlace. A manutenção de rota é implementada de forma a que, caso se perceba a queda de um enlace, esta informação seja disseminada a todos os nós que sabidamente usem este enlace em alguma rota ativa, isto é, os nós que efetivamente precisem desta informação. Define-se rota ativa como uma entrada "recente o suficiente" na tabela de roteamento com uma métrica finita. Somente rotas ativas podem ser usadas para o envio de pacotes. Na tabela de roteamento, existem também entradas com rotas não ativas, isto é, rotas inválidas, que serão apagadas após um certo *timeout*.

Ao contrário do DSR, o AODV não usa o roteamento por fonte, o que obriga que todos os nós intermediários dinamicamente estabeleçam entradas em tabelas de roteamento locais para cada destino. Mas em compensação, evita-se a sobrecarga na rede em função de, no caso do roteamento por fonte, ter-se que levar em cada pacote todo o caminho da fonte até o destino. Esta economia tende a ser tão maior quanto maior for o diâmetro da rede. Além disso, o roteamento por fonte e a possível existência de múltiplas rotas para cada destino no DSR acarretam uma maior sobrecarga de memória por parte deste protocolo em relação ao AODV.

Ainda como novidade em relação ao DSR, cada nó no AODV mantém um número de seqüência monotonicamente crescente que é usado para se evitar *loops* e como forma de se verificar se uma informação de rota é recente ou não. Quando ocorre a queda de um enlace, as rotas que o utilizam são invalidadas através de operações que envolvem o número de seqüência e a métrica da rota.

O AODV se baseia, assim como o DSR, nos mecanismos de manutenção e descoberta de rota, através da troca de *Route Replies*, *Route Requests* e *Route Errors* (estes últimos na implementação são considerados um caso especial de *Route Reply*). Assim, são construídas tabelas de roteamento nos nós, gerenciadas pelo protocolo, com os seguintes campos principais, os quais serão melhor explicados em seguida:

- endereço IP do destino;
- número de seqüência do destino;
- interface;
- contador de saltos;
- último contador de saltos;
- próximo salto;
- predecessores;
- tempo de vida.

V.2 Detalhes do Funcionamento do AODV

V.2.1 Descoberta de Rota

O processo de descoberta de rotas se dá também através da inundação de *Route Requests*. Cada nó possui dois contadores: um número de seqüência e um identificador de *broadcast*. Se um nó fonte precisa se comunicar com um nó destino para o qual ele não possui entrada em sua tabela de roteamento, então ele incrementa o seu número de seqüência e inunda a rede com uma requisição de rota, que leva os seguintes campos: endereço da fonte e do destino, número de seqüência da fonte e último número de seqüência do destino conhecido pela fonte, identificador de *broadcast* e contador de saltos (vale zero inicialmente).

Cada nó que recebe um *Route Request* (RREQ) pode:

- responder com um *Route Reply* (RREP) caso seja o destino ou possua em sua tabela uma entrada ativa para o destino com número de seqüência maior ou igual ao último conhecido pela fonte que é especificado no RREQ. Assim, a fonte consegue especificar quão recente tem de ser uma informação de forma a se poder aceitar que um nó intermediário responda, certo de que sua resposta contém uma informação nova;

- caso não seja uma nova requisição, isto é, caso ele já tenha feito prosseguir previamente a inundação desta mensagem, descarta-se o pacote redundante. Uma nova requisição pode ser identificada unicamente através do endereço da fonte e identificador de *broadcast* (este identificador é incrementado a cada novo RREQ enviado). Caso seja um novo RREQ, o nó incrementa o contador de saltos e inunda seus vizinhos com esta mensagem.

Enquanto um RREQ vai sendo inundado pela rede vão se guardando caminhos reversos de todos nós alcançáveis pela requisição até a fonte, de forma que se o destino for alcançável, onde quer que ele esteja, a rede saberá como fazer retornar a resposta até a fonte. Para implementar isto, todo nó intermediário que inunda a rede com um RREQ guarda automaticamente o endereço do vizinho de quem recebeu o primeiro RREQ em uma entrada relativa ao endereço da fonte. Nesta entrada, guarda-se também o número de seqüência da fonte e o tempo de expiração para uma entrada de rota reversa. Este tempo deve ser suficiente para que a requisição inunde a rede e uma resposta possa retornar à fonte através da rota reversa, logo seu valor depende do diâmetro da rede.

Quando, na circunstância explicada anteriormente, um nó, possivelmente o próprio destino, responde a uma requisição de rota, ele envia pelo caminho reverso um RREP contendo os seguintes campos: endereço da fonte e do destino, número de seqüência do destino, contador de saltos e tempo de vida para a rota (tempo em que se considera a rota ativa: *active_route_timeout*). Quando o destino responde à requisição, imediatamente antes de enviar o RREP, ele atualiza o seu número de seqüência para o máximo entre seu valor atual e o seu valor segundo o que constava no RREQ. Ao longo do caminho reverso, em cada nó percorrido pela resposta, guarda-se, em uma entrada referente ao destino, o próximo salto para alcançar o destino, que é o vizinho de quem se recebeu o RREP, bem como o número de seqüência do destino. Além disso, o tempo de vida para as entradas na tabela de roteamento tanto para fonte quanto para o destino será dado pelo tempo de uma rota ativa (*active_route_timeout*), na qual vai ser efetuada uma comunicação. A cada vez que chega um pacote usando uma determinada rota, o tempo de vida da entrada para o destino e para o próximo nó é aumentado de forma a ficar válido ainda por mais um período relativo a *active_route_timeout*. Vale ressaltar que nos nós não percorridos pelo RREP, a entrada para a fonte terminará por ser apagada (caso não se mantenha por causa de uma outra comunicação) após expirar o tempo em que se considera válida uma rota reversa.

Um nó intermediário transmite o primeiro RREP que recebe. Depois disso, para diminuir o número de respostas propagando, ele só propaga outra caso possua número de seqüência do destino maior ou o mesmo número mas com métrica menor. Assim, consegue-se fazer chegar à fonte informação atualizada e referente ao caminho mais curto. No momento em que se chega a primeira resposta, os dados já podem ser transmitidos. Caso posteriormente, uma melhor rota seja conhecida, o protocolo a utilizará de forma transparente para a aplicação.

Para se economizar banda e energia, o AODV também prevê uma inundação controlada, usando o mecanismo de "anel expansivo" de forma ligeiramente diferente com relação ao DSR. Primeiro, tenta-se com um RREQ com um valor mínimo de TTL (*TTL_start*). Se não chegar resposta durante um certo período, tenta-se de novo incrementando o TTL, isto é, o alcance da inundação de um valor, de um valor *TTL_increment*. Se ainda não se obtiver resposta, tentam-se sucessivos acréscimos ao valor de TTL, até se chegar a um limite, a partir do qual, decide-se por inundar a rede por completo. Neste último caso, podem-se tentar até um número máximo de tentativas, depois do qual se conclui que não há rota e se descartam os pacotes. Quando se recebe um RREP, o contador de saltos do RREP é armazenado como último contador de saltos na tabela de roteamento. Uma entrada de roteamento permanece ativa durante um certo período e, depois disso, ela ainda é guardada durante um outro período antes de apagá-la. Assim, durante este tempo, guarda-se informação sobre o estado das rotas da rede, de forma que, se for necessária a descoberta de nova rota para um determinado destino, pode-se usar então o campo último contador de saltos para este destino, incrementado de um valor de acréscimo (*TTL_increment*), como o valor de TTL para a primeira tentativa de inundação de RREQs.

O AODV também prevê o uso de *gratuitous* RREPs. Caso uma requisição de rota seja respondida apenas por nós intermediários, então o nó destino terminará por não conhecer uma rota até a fonte, obrigando-o a fazer o procedimento de descoberta de rota para a fonte, caso seja necessária uma comunicação bidirecional. Para evitar esta inundação, pode-se enviar o RREQ, indicando por meio de um *flag* que se deseja usar o recurso de *gratuitous* RREP. Neste caso, um nó intermediário, além de responder à fonte normalmente com um RREP, envia um *gratuitous* RREP para o destino, que permite estabelecer a rota reversa do destino até a fonte.

V.2.2 Manutenção de Rota

Se um nó ativo se move, mudando assim a topologia da rede, de forma que um dado enlace que estava sendo usado não existe mais, então um RREP especial é enviado a todos os nós afetados. Para saber a quem é necessário informar a queda do enlace, o nó que detecta a queda verifica quais destinos eram atingidos por rotas que usavam este enlace e informa a todos os seus vizinhos que, sabidamente, o usem como próximo passo para se chegar a estes destinos. Isto é implementado usando o campo predecessores na tabela de roteamento de cada nó que guarda para cada destino quais vizinhos usam este nó para se chegar ao destino. Esta informação de predecessores é atualizada dinamicamente, de forma que se um nó não recebe nenhum pacote de um de seus predecessores durante um período de *active_timeout*, este predecessor é apagado. Se verificar-se necessário informar a queda a apenas um nó, o protocolo o faz através do envio de um pacote *unicast* diretamente ao nó interessado, mas se for preciso avisar a mais de um nó então informa-se preferencialmente por *broadcast*. Para tal fim, usa-se um RREP especial que lista cada um desses destinos (anteriormente alcançáveis através do enlace que caiu) com um número de seqüência para cada um deles uma unidade maior que o último conhecido (ficando este último ultrapassado). Esta mensagem deve ser propagada por toda a rede, de forma a atingir todos os nós que usavam em alguma rota este enlace que caiu. Assim, cada nó por onde passa esta mensagem deve decidir localmente se deve continuar repassando a mensagem (possivelmente alterando adequadamente a mensagem), além de atualizar a tabela de roteamento, registrando que os destinos especificados se encontram inalcançáveis (métrica infinita) e atualizando seus números de seqüência. Para se detectar a queda de um enlace, podem ser usadas mensagens periódicas de *hello* ou, de forma alternativa, informações provenientes da camada de acesso ao meio. Será explicado, primeiramente, o uso de mensagens de *hello*. Os nós podem conhecer seus vizinhos de duas maneiras: quando um nó escuta um *broadcast* de outro nó, ele atualiza sua informação de conectividade local colocando este vizinho; outra maneira seria caso um nó não tenha mandado pacotes para todos seus vizinhos no período de um intervalo *hello-interval*, então ele deve explicitamente se mostrar vivo e acessível como vizinho, mandando por *broadcast* aos vizinhos uma mensagem de *hello*. Esta mensagem é um tipo especial de RREP que contém o endereço do nó e seu número de seqüência, que não é trocado por ocasião da transmissão de mensagens de *hello*. O seu valor de TTL é um, logo apenas os vizinhos recebem e atualizam sua informação de conectividade local. O recebimento de um *hello* de um novo vizinho ou o não recebimento do mesmo de um antigo vizinho durante alguns períodos (*allowed_hello_loss*) do intervalo entre sucessivos envios de *hello* caracterizam uma mudança na topologia. Neste segundo caso, se deduz que um enlace caiu e caso este enlace fosse usado atualmente em rotas ativas, esta informação é propagada como explicado anteriormente. Já no caso de a camada de acesso ao meio oferecer um *feedback* útil quanto à detecção de quedas de enlace, pode-se usá-lo como forma de se economizar banda. No caso do IEEE 802.11 por exemplo, podem-se usar os ACKs da camada de enlace: assim, se um nó envia um pacote *unicast* para um outro nó anteriormente alcançável, mas não recebe em seguida a confirmação do recebimento através de um ACK, pode-se concluir que o enlace caiu, devendo-se proceder ao mecanismo de disseminação da informação de queda do enlace.

Pode-se usar o *hello* para descobrir enlaces bidirecionais (conforme necessário pelo mecanismo de envio de RREPs do AODV): basta listar em cada *hello* todos os nós de quem se escutou um *hello*. Assim, nós podem verificar se conseguem se escutar mutuamente, o que implica enlaces bidirecionais.

Cada fonte, ao final da propagação pela rede da queda de um enlace, decide se refaz ou não o processo de descoberta de rota. Caso ela tenha informação ainda a transmitir certamente terá que aprender uma nova rota, mas caso não usasse aquela rota há muito tempo e estivesse já perto de expirá-la pode-se evitar uma inundação da rede.

Capítulo VI

Simulações

VI.1 Network Simulator - NS

Para se fazer as simulações necessárias para o teste, análise e comparação dos protocolos de roteamento, foi usado o *Network Simulator* (NS) [5], que é capaz de executar simulações suficientemente completas, retratando o que ocorre nas camadas de uma rede. Além disso, é gratuito, de código aberto e encontra-se disponível na Internet.

O NS é um simulador orientado a eventos discretos desenvolvido pela UCB (*University of California at Berkeley*) e pelo projeto VINT (*Virtual InterNetwork Testbed*), que vem evoluindo desde 1989 até o presente momento, sempre tentando acompanhar a evolução das redes de computadores. Programadores espalhados por todo o mundo ajudam a contribuir com o desenvolvimento do NS, acrescentando novas funcionalidades e consertando erros. O NS foi escrito nas linguagens C++ e OTcl (*Object Tool command language*). Seu núcleo foi implementado em C++. As simulações são configuradas por *scripts* OTcl, nos quais se especificam os nós, os enlaces, os protocolos utilizados, se modelam a geração de tráfego, as perdas de pacotes associadas a falhas nos nós e nos enlaces, pode-se gerar congestionamento, etc.

A estrutura do NS é composta por agentes, um ponto de entrada no nó, um classificador de endereços e um classificador de portas. Os agentes implementam protocolos, gerando e consumindo pacotes. Como exemplos, podem-se citar agentes TCP (*Transmission Control Protocol*), UDP (*User Datagram Protocol*) e os agentes nulos (que só consomem pacotes). Um pacote gerado por um agente é primeiro passado ao ponto de entrada do nó ao qual o agente está associado. Este ponto de entrada recebe os pacotes dos agentes e também os pacotes destinados a este nó. Depois do ponto de entrada, o pacote passa pelo classificador de endereços, o qual decide se deve ser enviado para outro nó por um enlace de saída ou se o pacote deve ser entregue a outro agente associado ao mesmo nó. Neste último caso, o pacote é repassado ao classificador de porta que entrega o pacote ao devido agente.

VI.1.1 Extensão de Mobilidade e Redes Sem Fio no NS

Diante da falta de suporte para os aspectos físicos e protocolos MAC para simular redes sem fio com múltiplos saltos, o grupo Monarch [10] acrescentou ao NS funcionalidades, tais como:

- mobilidade do nó. Cada nó móvel é uma entidade responsável por guardar sua posição e velocidade como função do tempo;
- interface de rede com propriedades como potência de transmissão, ganho da antena e sensibilidade do receptor. Cada nó pode ter uma ou mais interfaces. Cada uma destas é conectada a um canal. Assume-se que os canais são ortogonais, de forma que a transmissão e recepção de um canal não interfira com a de outro;
- camada física incluindo o modelo de rádio de propagação, calculando-se atraso de propagação e atenuação no meio. Um pacote enviado em um canal é escutado por todas as estações que escutam este canal. Estas verificam, segundo o modelo de atenuação, se o sinal é efetivamente recebido ou não como dado útil;
- o protocolo MAC do IEEE 802.11 usando DCF (*Distributed Coordination Function*). Assim, antes de transmitir um pacote *unicast*, precisa-se reservar o meio através de pacotes *Request To Send* (RTS) e *Clear To Send* (CTS). E, para cada pacote

unicast, o receptor envia uma confirmação para o emissor. Já pacotes *broadcast* não são precedidos pela troca de RTS e CTS e nem são em seguida confirmados, mas são enviados somente quando se escuta o meio como livre;

- os protocolos de roteamento *ad hoc* DSR, AODV, DSDV e TORA.

Assim como nas redes tradicionais, os nós usam o protocolo ARP (*Address Resolution Protocol*) para mapear endereços IP em endereços MAC. Assim, quando um nó fonte deseja enviar um pacote a outro nó cujo endereço IP se conhece, mas não o MAC, envia-se por *broadcast* um pacote de *ARP Request*.

Foi implementada cada interface de rede com uma fila de capacidade configurável. Acima do seu limite, descartam-se os pacotes. Separadamente cada protocolo de roteamento sob demanda, como por exemplo o DSR e o AODV, são capazes de armazenar em outra fila um certo limite de pacotes esperando pela descoberta de rota.

Visando a difusão da informação de roteamento de forma rápida, implementou-se a interface de rede fornecendo prioridade aos pacotes de roteamento: assim estes pacotes são colocados no começo da fila de transmissão, enquanto que os outros (ARP e dados) são postos no final. A implementação do grupo Monarch também se beneficia do fato de que a camada de enlace utilizada é IEEE 802.11 usando DCF: logo pode-se concluir da ausência de confirmação de um pacote que um enlace caiu.

Em particular com relação à implementação do DSR, todos os nós operam em modo promíscuo para poder aprender informações potencialmente úteis sem sobrecarga de roteamento. A inundação controlada de requisição de rota implementada foi aquela em que se questiona primeiro os vizinhos e só depois inunda-se a rede. E, já que o IEEE 802.11 requer a troca de RTS/CTS/dado/ACK na entrega de pacotes *unicast* (de forma a requerer enlaces bidirecionais), então a resposta a uma requisição de rota usa sempre o caminho oposto ao percorrido pelo RREQ. Assim, a fonte só aprende caminhos bidirecionais.

Com relação ao AODV, foi implementada primeiramente uma versão usando mensagens periódicas de *hello* para se verificar a queda de um enlace, de forma independente da camada de enlace. Mas, posteriormente, para efeitos de comparação e aproveitando-se a implementação do 802.11 na camada de enlace, o grupo Monarch implementou uma versão usando somente o *feedback* da camada de enlace, de forma a se dispensar o uso de mensagens de *hello*. Esta segunda abordagem que usa um conhecimento da camada inferior também é permitida por sua especificação. Desta forma, não se sobrecarrega a rede com mensagens periódicas. Ao contrário da versão com uso de *hello*, ela é totalmente sob demanda e, assim, a queda de um enlace só pode ser descoberta no momento em que for necessário enviar um pacote através deste enlace. Após simular as duas versões, o grupo Monarch verificou que esta última abordagem possuía melhor desempenho [6]. Além disso, os próprios autores do AODV, em recente artigo [7], usaram esta abordagem em sua simulação no NS do AODV, logo foi também esta a escolhida para este trabalho.

VI.2 Modelo de Simulação

Objetivando testar e analisar como os algoritmos de roteamento DSR e AODV, previamente detalhados, reagem quando exigidos realmente, foram usadas as funcionalidades do NS para simular algumas situações relevantes. Nele, podem-se variar parâmetros importantes para uma rede, tais como o tipo de tráfego, tamanho do pacote, o formato e o tamanho da área de simulação, a velocidade dos nós, o tempo de pausa entre cada movimento, o número de nós no cenário, o número de conexões e a duração da simulação. O trabalho visou uma compreensão mais geral do comportamento do roteamento em redes *ad hoc*, em detrimento de uma segunda abordagem possível que analisaria um caso específico, tomado de uma situação real, mas que poderia resultar em favorecimento de um determinado protocolo que se saísse melhor diante de uma configuração tão particular, o que inviabilizaria que se chegasse a conclusões mais gerais.

Certamente uma análise pertinente dos protocolos depende de uma boa escolha dos parâmetros que influenciam diretamente o padrão de mobilidade e tráfego das redes. Em seguida, apresentam-se à esquerda os parâmetros a ser variados e à direita os valores escolhidos para uso na simulação.

- Tipo de tráfego:

CBR ou TCP	CBR com taxa de 4 pacotes/s
Tamanho do pacote	512 *bytes*

- Área da simulação:

Formato (retang. ou quad.)	Retangular
Área	1200m x 500m

- Grau de mobilidade:

Velocidade dos nós	1 e 20 m/s
Tempo de pausa	0s, 50s, 100s, 300s e 600s

- Outros:

Número de nós no cenário	60 nós
Número de pares fonte-destino	10 ou 30
Tempo de simulação	600s

Tal escolha se justifica, conforme apresentado a seguir. Primeiramente, na simulação a rede deve ser implementada em uma região e deve se constituir de um certo número de nós. Para a área de simulação, usou-se uma área retangular de 1200m x 500m. Ora, tendo-se usado um raio de alcance de cada nó móvel de 250m, esta área já delimita uma primeira informação relevante: o diâmetro máximo do grafo referente à topologia da rede. Assim, para um grafo conexo no pior caso um nó fonte estará a 6 enlaces de distância de seu destino. Para se apreciar a importância de tal informação, basta verificar por exemplo que a complexidade em número de passos para se descobrir uma rota tanto no DSR quanto no AODV é O(d), sendo d o diâmetro da rede. Também depois de uma falha para se descobrir uma nova rota, tem-se complexidade O(d) para os dois (porém ressalte-se que no melhor caso, no DSR, uma rota alternativa já constaria no *cache* e poderia ser usada diretamente, tendo complexidade de melhor caso de valor zero). Além disso, deu-se preferência a uma área retangular em vez de quadrada justamente para se obter, para uma mesma área, caminhos maiores em média. Para o número de nós, escolheu-se 60, o que confere uma densidade razoável à rede. A densidade é outra informação relevante, uma vez que em uma rede *ad hoc* para poder haver comunicação entre todos os nós, a densidade não pode ser muito pequena, pois assim, se a área for grande quando comparada com a área de alcance de cada nó individualmente, não será possível estabelecer caminhos para viabilizar o alcance a todos os nós. Porém a densidade não pode ser excessivamente elevada também, pois o meio é compartilhado e isto acarretaria congestionamento, caso muitos nós desejassem enviar pacotes. Logo a densidade é um parâmetro que deve ser levado em conta se for necessário manter uma rede o mais conexa possível durante um tempo considerável.

Ora, precisava-se testar uma das características mais imediatas de redes *ad hoc*: sua mobilidade. Assim, usou-se um modelo chamado "*random waypoint*" que se caracteriza por uma certa velocidade e tempo de pausa. Assim, no começo se escolhe para cada nó um ponto de início aleatoriamente dentro da área de simulação. Cada nó, primeiramente permanece aí parado por um período dado pelo tempo de pausa. Terminado este período, ele escolhe uma nova posição aleatoriamente e se dirige até lá com uma velocidade distribuída uniformemente entre zero e uma velocidade máxima. Chegando lá, o nó volta a ficar parado por um período de pausa, repetindo o processo até o fim da simulação. Procurou-se escolher valores que tentassem imitar a realidade. Assim, por exemplo, usou-se duas possíveis velocidades máximas: 1 m/s, que pretende representar pessoas andando, e 20 m/s, que representa um carro se movendo. Para os valores do tempo de pausa, escolheram-se 0, 50, 100, 300 e 600 segundos. Ora, um tempo de pausa de 0s refere-se a um padrão de mobilidade constante, o que é exatamente oposto do tempo de pausa de 600s, que representa uma rede com nós que, apesar de ter liberdade de movimento, ficam estáticos durante toda a simulação, uma vez que 600s foi também o tempo total de simulação. Assim, com duas velocidades e cinco tempos de pausa possíveis têm-se dez padrões de mobilidades distintos, permitindo-se analisar como os protocolos reagem aos diferentes dinamismos de mudança de topologia.

Ora, tanto o DSR quanto o AODV são protocolos sob demanda, logo todo o mecanismo de descoberta de rotas só ocorre depois que uma rota se faz necessária para se efetuar uma comunicação. Além disso, a razão de ser de uma rede de comunicações e mesmo a razão de existirem protocolos de roteamento é exatamente viabilizar comunicações. Assim, como especificação do tráfego na rede, preferiu-se usar um tráfego a taxa constante (CBR: *Constant Bit Rate*), em vez de usar TCP. O TCP faz controle de congestionamento, logo ele procura adaptar o seu próprio tráfego às características da rede naquele instante, buscando uma vazão elevada, mas ao mesmo tempo evitando o congestionamento. Esta característica do TCP prejudica a perfeita comparação entre os protocolos por meio da simulação. Ora, pretende-se pôr os protocolos à prova com igualdade de condições, isto é, modelo de mobilidade e tráfego exatamente iguais, mas, com o TCP e seus mecanismos de controle de congestionamento, o momento em que os pacotes seriam enviados pela fonte para os dois protocolos deixariam de ser necessariamente iguais (o mesmo vale para a posição dos nós). Além disso, o TCP tentaria evitar o congestionamento que poderia ser um dos efeitos secundários de um dado protocolo de roteamento e, como tal, o TCP estaria mascarando uma informação relevante. Além de oferecer esta igualdade de condições aos protocolos, um tráfego CBR até o fim de sua transmissão está constantemente requisitando serviço à rede, o que é interessante para avaliar o desempenho dos protocolos de roteamento. Escolheu-se usar 10 ou 30 pares fonte-destino com tráfego CBR com taxa de 4 pacotes por segundo. Mudar a taxa seria aproximadamente equivalente a mudar na mesma proporção o número de pares fonte-destino [6]. Preferiu-se usar um número de pares fonte-destino distintos, em vez de taxas distintas, pois assim força-se que os protocolos de roteamentos atuem mais, já que, desta forma, haverá mais fontes demandando rotas até um destino. Um número de fontes maior força também a capacidade da rede, no sentido que haverá mais pacotes trafegando, podendo chegar a uma condição de congestionamento, condição na qual uma possível sobrecarga de roteamento será mais fortemente sentida. Para o tamanho do pacote, escolheu-se usar 512 *bytes* que não é um valor excessivamente grande, o que causaria problema de congestionamento para ambos os protocolos, seguido por grande descarte de pacotes, devido à falta de diversidade espacial no meio compartilhado [6].

VI.3 Geração e Filtragem da Informação

O objetivo foi, pois, simular ambos os protocolos em igualdade de condições, isto é, com o padrão de mobilidade e de tráfego exatamente iguais. Só assim uma análise e comparação direta poderiam ser feitas. A seguir, segue uma breve explicação de como foi feito para alcançar este intuito. Todo o processo foi feito de forma a ser bastante genérico e automatizado, de forma a se facilitar futuramente a execução de simulações semelhantes variando novos parâmetros ou comparando outros protocolos de roteamento.

Para gerar os padrões de mobilidade, fez-se um *script* que usava a funcionalidade da extensão do grupo Monarch ao NS que permite gerar um cenário de mobilidade. Este *script* recebe informações de um arquivo de configuração dos parâmetros do cenário, que define o número de nós, o tempo de pausa, a velocidade máxima dos nós, o tempo de simulação e a dimensão horizontal e vertical da área de simulação. No caso desta simulação, estes valores foram apresentados nas tabelas da seção anterior. Especifica-se também como entrada para este *script* quantos cenários com as mesmas características se deseja gerar. Então foram feitos 10 cenários diferentes para cada configuração de parâmetros (por exemplo, 60 nós, 50s de pausa, velocidade de 1m/s, 600s de simulação e área de 1200m x 500m), já que se fosse usado apenas um cenário, como este é gerado aleatoriamente, por acaso poder-se-ia privilegiar um determinado protocolo. Repetindo a simulação para uma série de cenários distintos (com os mesmos parâmetros), e tirando a média no final, obtém-se um resultado bem mais confiável. Ao final deste processo, foram criados e armazenados 100 cenários distintos (2 velocidades possíveis x 5 tempos de pausa x 10), que, em linguagem OTcl, detalham onde cada nó começa a simulação e todo o seu padrão de movimento e de pausa que se segue desde então.

De forma equivalente, para gerar os padrões de conexão fez-se outro *script*. Este recebe informações de um arquivo sobre as possíveis configurações de tráfego. Nesta simulação foram usados, conforme explicado anteriormente, 10 ou 30 pares fonte-destino, com tráfego CBR de taxa 4 pacotes por segundo em uma rede com 60 nós. Assim, este *script* gera dois arquivos (um para cada conjunto de parâmetros) que detalham, em linguagem OTcl, a criação de agentes UDP (*User Datagram Protocol*), que são associados a nós fontes, os quais enviarão em um determinado instante uma série de pacotes a intervalo de tempo constante (CBR) a um determinado nó destino.

Foi feito então um *script* OTcl que recebe como entrada: o número de nós no cenário, as dimensões horizontal e vertical, um determinado arquivo de padrão de mobilidade e outro de tráfego (explicados anteriormente), o tempo de simulação, o algoritmo de roteamento que será testado e o arquivo onde será apresentada a saída. Este *script* primeiramente prepara toda a inicialização para a simulação de uma rede *ad hoc*. Assim, por exemplo configura-se o canal (sem fio), o modelo de propagação, a interface de rede, a camada MAC (802.11), a política e o tamanho da fila, a antena (omnidirecional), a área de simulação (1200m x 500m), o número de nós, o protocolo de roteamento utilizado (DSR ou AODV) e o instante quando deve terminar a simulação. Cria-se ainda uma série de outros objetos necessários para se efetuar a simulação. Em seguida, carrega-se um par de arquivos de padrão de mobilidade e de tráfego, que vão exatamente pôr à prova o protocolo de roteamento. Então faz-se executar a simulação até o seu fim. Um arquivo de saída detalha tudo o que ocorreu na rede. Nele, há uma linha para cada evento ocorrido em cada camada durante a simulação, isto é, transmissão do nó inicial, recepção, descarte e envio por nó intermediário. Assim, por exemplo a transmissão de um pacote UDP resulta em linhas para a transmissão na camada de transporte (nível de agente), de rede e de enlace no nó fonte e de recepção em cada uma dessas camadas no nó destino, além do que, se houver nó intermediário, haverá também linhas referentes à recepção e envio de pacotes nas camadas de rede e de enlace deste nó. Para diminuir a quantidade de informação nos arquivos de saída que, mesmo compactada, se mostrou muito grande, resolveu-se trabalhar apenas com os eventos da camada de transporte e de rede. Em outras palavras, foi desabilitada a marcação de eventos da camada enlace, a qual detalharia a troca de pacotes ARP e de controle de acesso ao meio. Esta informação, apesar de interessante, é de caráter secundário quando se está focalizando os protocolos de roteamento. Além disso, ela retrataria o funcionamento do 802.11, que não é necessariamente o protocolo escolhido para a camada de enlace em uma rede qualquer.

Foi feito ainda um *script* para fazer simular cada par possível de padrão de mobilidade e de tráfego para cada protocolo de roteamento, resultando em 400 arquivos de saída (100 cenários de mobilidade x 2 padrões de tráfego x 2 protocolos de roteamento). Cada linha destes segue um padrão para viabilizar o processamento futuro desta informação. Alguns campos destes arquivos são: tipo de evento (transmissão por nó inicial, recepção, descarte ou envio por nó intermediário de pacotes), instante do evento, identificador do nó, suas coordenadas no espaço, nível do evento: AGT (transporte), RTR (rede), MAC (enlace), razões para um descarte, endereço e número da porta da fonte e do destino, tamanho do pacote, valor do TTL, identificador do próximo nó para chegar ao destino, número de seqüência do pacote, além do número de vezes que um pacote foi enviado por um nó intermediário e o número ótimo possível.

Estes arquivos de saída contêm toda a informação que um observador da rede (o próprio simulador) com acesso a todas as camadas de todos os nós pôde obter. Porém, desta forma, acaba-se por reunir tanta informação que, para a análise por parte de um ser humano (e não um computador), se faz imperativa a filtragem dos dados. Com esta finalidade, foram feitos *scripts* em *tcsh*, que, em cada linha do arquivo de saída, procurava as colunas importantes e processava esta informação de forma a se obter a informação relevante, além de se compreender melhor os eventos que estão acontecendo e verificar o perfeito funcionamento do simulador.

Capítulo VII

Análise dos Protocolos

VII.1 Métrica

Como métrica para analisar o desempenho dos protocolos de roteamento DSR e AODV, escolheu-se utilizar:

- taxa de entrega de pacotes: razão entre o número de pacotes enviados pela aplicação na fonte e o número de pacotes recebidos pela aplicação no destino;
- sobrecarga de roteamento:

  em número de pacotes: quantidade de pacotes de roteamento transmitidos (conta-se em cada enlace no qual se envia);

  em número de *bytes*: quantidade total de *bytes* enviados referentes à camada de rede, isto é, somam-se a quantidade de *bytes* em cada pacote de roteamento e a quantidade de *bytes* de cabeçalho da camada de rede nos pacotes que carregam os dados enviados pela aplicação na fonte;
- otimização do caminho: a diferença entre o número de enlaces percorridos por um nó para chegar ao destino e o número de enlaces no caminho ótimo no momento em que se gerou o pacote.

VII.1 Métrica

Primeiramente, a taxa de entrega de pacotes é fundamental em toda rede, está relacionada com a taxa de perda de pacotes e a máxima vazão que se pode obter. Quando se enviam pacotes, naturalmente deseja-se que eles atinjam o seu destino. Pode-se até admitir, em determinadas aplicações (como tráfego de voz, por exemplo), alguma perda na rede, mas, com certeza, quanto maior a taxa de entrega de pacotes, melhor será. Além disso, ainda que se tolere alguma perda, não se pode aceitar que o mau funcionamento do protocolo de roteamento seja o responsável maior por esta perda. Uma taxa de entrega alta é um indício do funcionamento correto e completo do protocolo, isto é, ele consegue fazer o seu papel de achar uma rota até o destino e o faz sem sobrecarregar excessivamente a rede, de forma que os pacotes da aplicação conseguem percorrer todo o caminho sem serem descartados.

A sobrecarga de roteamento é altamente relevante. Além de influenciar diretamente a escalabilidade do protocolo, em uma rede com maior propensão ao congestionamento, devido à baixa banda, o tráfego referente ao roteamento deve ser o menor possível quando comparado com o tráfego de dados provenientes da aplicação. Em uma rede congestionada, uma sobrecarga de roteamento leva ao descarte de pacotes em geral, o que prejudica não só a vazão, mas também à descoberta e atualização de rotas. Além disso, para se enviar pacotes de roteamento, se está gastando energia nos nós, que é outro recurso escasso em redes sem fio. Ainda pode-se dizer que um maior número de pacotes de roteamento, aumenta a probabilidade de colisão, o que levaria a maiores atrasos, um pacote ficaria mais tempo em média na fila até conseguir ser recebido pelo próximo nó. Além de se contabilizar a sobrecarga em número de pacotes, decidiu-se fazê-lo também através do número de *bytes* como maneira de avaliar as conseqüências do roteamento por fonte do DSR.

A terceira métrica é a otimização do caminho que reflete diretamente o funcionamento correto de protocolos de roteamento que se propõem a descobrir o menor caminho de um nó até outro, como é o caso do DSR e do AODV. O simulador, com seu conhecimento total da rede, guarda, ao longo da simulação, todas as menores distâncias entre cada par de nós. Estas são tomadas baseadas unicamente no raio de alcance e na distância física entre os nós, logo não leva em conta efeitos de interferência e ruído no meio sem fio. Ora, como os nós têm liberdade de movimento, a qualquer instante um nó pode sair do raio de alcance de outro ou, de forma inversa, pode entrar no seu alcance, viabilizando um caminho entre os dois e também entre todos os nós alcançáveis pelos dois. Pode também ocorrer que, devido ao movimento, entre dois nós que já podiam se comunicar, de repente uma outra rota se faça possível e seja de menor custo, isto é, possua menos saltos. Tudo isto tem que ser observado ao longo de toda a simulação. Além disso, o simulador registra para cada pacote no arquivo de saída quantas vezes ele foi propagado. Estas funcionalidades nos permitem verificar a diferença entre o caminho ótimo e o caminho tomado pelo pacote até o destino. Esta é uma métrica importante, já que caminhos maiores que o ótimo consomem mais recursos escassos como energia e banda da rede como um todo, além do que a latência torna-se claramente maior. Um caminho alternativo um pouco maior pode-se mostrar interessante para fazer o balanceamento de carga em redes congestionadas, mas nenhum dos dois protocolos se aventura premeditadamente neste quesito.

VII.2 Análise dos Resultados

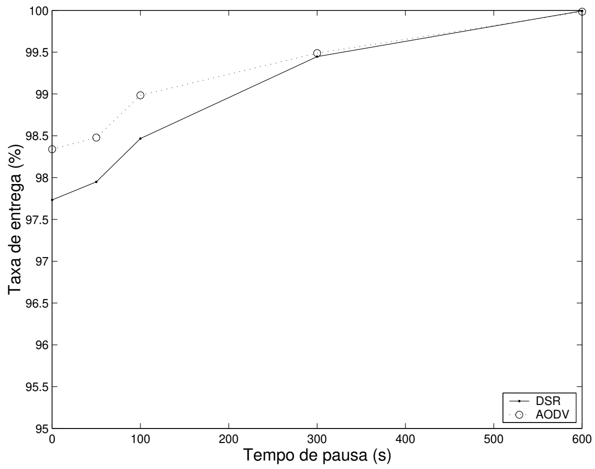
Foram feitos *scripts* para gerar os gráficos apresentados a seguir de cada uma das métricas. Os gráficos apresentados referem-se à velocidade de 20m/s, que se mostrou mais ilustrativa. Tal velocidade implica um maior dinamismo na troca de rotas, sendo, pois, interessante analisar como os protocolos reagem quando forçados desta maneira.

Para velocidade de 1m/s, a topologia muda mais lentamente, as rotas são mais estáveis e o trabalho dos protocolos de roteamento torna-se mais fácil. Nesta configuração, ambos os protocolos se mostraram como boas opções, conseguindo taxas de entrega muito boas: para 10 pares fonte-destino ambos os protocolos conseguiram entregar quase 100% dos pacotes, independente do tempo de pausa. De resto, a sobrecarga de roteamento segue as características que serão explicadas na análise extensiva que se fará em seguida para os resultados obtidos.

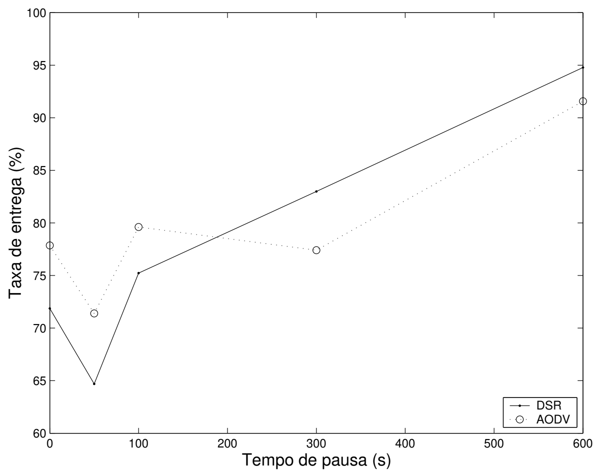
Para os gráficos de taxa de entrega de pacotes e de sobrecarga de roteamento, o eixo horizontal refere-se aos tempos de pausa em segundos. O eixo vertical depende do objetivo de cada gráfico, mas deve-se reparar que a sua faixa de valores varia, de forma a se obter curvas mais ilustrativas.

VII.2.1 Taxa de Entrega de Pacotes

Nas Figuras VII.1(a) e VII.1(b) são apresentados gráficos para a taxa de entrega de pacotes para 10 e 30 pares fonte-destino, respectivamente. Em seguida, faz-se a análise pertinente.



(a) 10 pares fonte-destino



(b) 30 pares fonte-destino

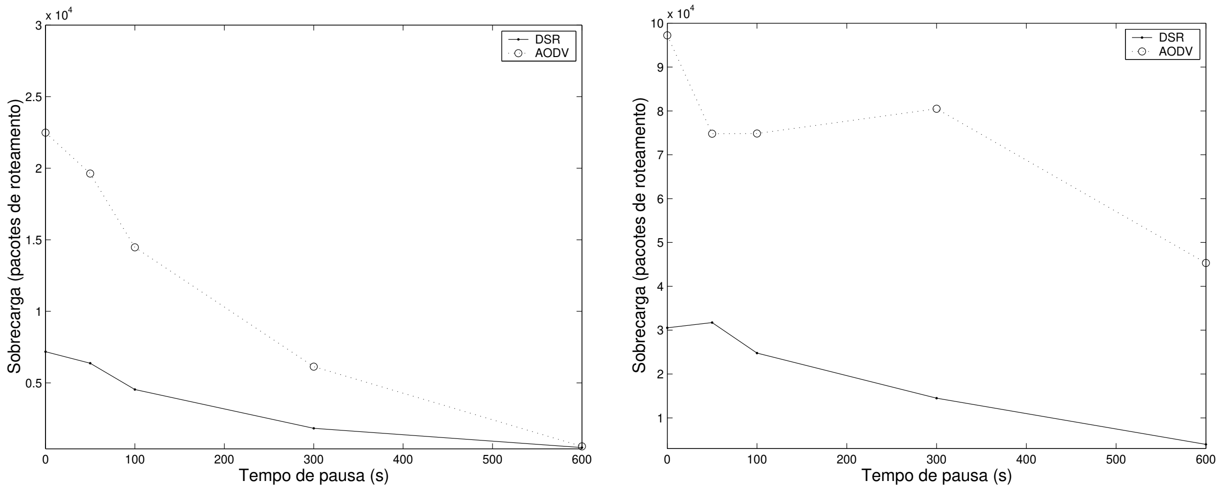
Figura VII.1: Taxa de entrega para nós com velocidade máxima de 20 m/s.

Quanto à taxa de entrega de pacotes, constata-se para ambos os protocolos uma sensível diferença de desempenho para diferentes cargas na rede. Para 10 pares fonte-destino, os dois protocolos possuem desempenho muito bom, que se mantém acima de 97% para todos os valores de pausa. Eles possuem comportamento quase idêntico. Já para uma maior carga, 30 pares fonte-destino, as curvas já se mostraram mais diferentes: verificou-se que para alta mobilidade ambos os protocolos têm dificuldade, perdem um número relevante de pacotes, mas, ao se diminuir a mobilidade (aumentando os tempos de pausa), eles melhoram o seu desempenho. Vale salientar neste gráfico que ocorre uma inversão importante: o AODV consegue entregar um maior percentual de pacotes para alta mobilidade, enquanto que para baixa mobilidade o DSR possui melhor desempenho.

Podem-se apreciar alguns fatos. A tendência geral das curvas é subir, isto é, aumentar a taxa de entrega com o aumento do tempo de pausa. Isto é perfeitamente esperado, pois quanto maior o tempo de pausa, durante mais tempo os nós ficam parados, dando tempo para que rotas se estabilizem e possam ser usadas. Já com maior mobilidade, as rotas precisam ser alteradas com mais freqüência, tornando bem mais árduo o trabalho dos protocolos de roteamento. Em particular, quedas de enlace ocorrem com mais freqüência e certamente perdas de pacotes também. Além disso, o desempenho para 30 pares fonte-destino é pior do que para 10 pares fonte-destino. Isto também era esperado, pois a capacidade da rede está sendo mais requisitada, gerando congestionamento e perda de pacotes. Mas interessante é notar como os protocolos reagem a esta realidade que não é incomum, devido à banda reduzida do meio sem fio. Percebe-se que o AODV reage melhor à alta mobilidade e o DSR, à baixa mobilidade. Ora, o DSR se baseia em roteamento por fonte e pode possuir em seu *cache* múltiplas rotas para um destino. Já o AODV se baseia em tabelas de roteamento mais tradicionais e possui no máximo uma rota para cada destino em cada nó. Assim, quanto maior for a mobilidade, mais rápido as rotas alternativas presentes no *cache* vão se desatualizar, resultando mais freqüentemente em perda de tempo e de pacotes confiar-se em uma rota no *cache*. Por exemplo, diante de uma queda de enlace que estava sendo usado em algum caminho, valeria mais à pena descobrir uma nova rota como faz o AODV do que pegar uma do *cache*. Além disso, o DSR em sua abordagem, presume que os nós não se movam tão rápido que o roteamento por fonte se torne inviável, isto é, se a mobilidade for tanta que, antes de se percorrer todo o caminho especificado até o destino, este caminho já não exista mais, isto inviabilizaria o roteamento por fonte caso fosse um evento comum. Mas para baixa mobilidade, as rotas ficam mais estáveis e o uso de *cache* e a escuta em modo promíscuo se mostram uma opção interessante, aprendendo-se caminhos sem se sobrecarregar uma rede congestionada, de forma que o DSR se mostra superior nesta configuração. Assim descobrem-se rotas ainda atuais, evitando-se a inundação da rede por meio de um mecanismo de descoberta de rotas como faz o AODV.

VII.2.2 Sobrecarga de Roteamento

Nas Figuras VII.2(a) e VII.2(b) são apresentados gráficos para a sobrecarga de roteamento em número de pacotes para 10 e 30 pares fonte-destino, respectivamente.



(a) 10 pares fonte-destino

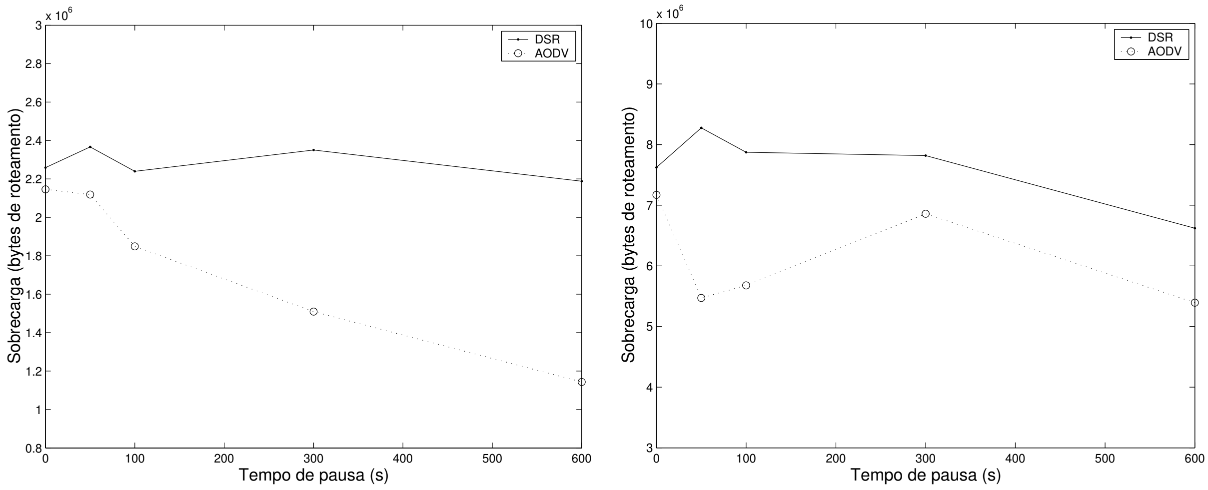
(b) 30 pares fonte-destino

Figura VII.2: Sobrecarga de roteamento (em número de pacotes) para nós com velocidade máxima de 20 m/s.

Estes gráficos são bastante ilustrativos de algumas características de projeto de cada um dos protocolos. Primeiramente, para a sobrecarga de roteamento em número de pacotes percebe-se que ambos os protocolos trocam consideravelmente mais pacotes de roteamento para um maior número de pares fonte-destino. Isto é uma conseqüência direta do roteamento sob demanda que ambos implementam. Logo se ninguém quisesse se comunicar, não haveria nenhum pacote de roteamento (lembre-se que não estão sendo usadas mensagens periódicas de *hello* para o AODV). Mas com o aumento da demanda, aumenta a troca de RREQs e RREPs para a descoberta de rotas. Além disso, caso as rotas se alterem, aumenta também a troca de mensagens de queda de enlace, que também é sob demanda, conforme implementado. Outro efeito esperado é que o número de pacotes de roteamento para ambos os protocolos aumenta com o aumento da mobilidade, isto é, menores tempos de pausa. Ora, quanto menor o tempo de pausa, a uma velocidade de 20m/s, mais freqüentemente rotas se alterarão e haverá quedas de enlaces, logo com mais freqüência se dará o procedimento de descoberta de rotas (troca de RREQs e RREPs) e de manutenção de rotas (acarretando troca de RERRs). Assim, quanto menor a mobilidade, as rotas podem se estabilizar e o número de pacotes de roteamento cai sensivelmente.

Além disso, percebe-se que o DSR consegue fazer seu trabalho com bem menor número de pacotes de roteamento que o AODV. Isto é conseqüência direta do uso intensivo de *cache* com múltiplas rotas possíveis e escuta em modo promíscuo por parte do DSR, enquanto que no AODV se mantém em cada nó para cada destino no máximo uma entrada na tabela de roteamento. Assim, a queda de um enlace no AODV implica nova inundação da rede para se descobrir uma rota, isto é, O(N) mensagens, onde N é o número de nós na rede, para inundar a rede com RREQ. Já o DSR, diante do evento de queda de enlace, tem complexidade de melhor caso zero, isto é, no caso de haver uma rota alternativa em seu *cache* que foi aprendida previamente. Trata-se de um procedimento muito bem comportado e preocupado com a sobrecarga na rede e a banda disponível para as aplicações. Claro que no pior caso, o nó não possuirá uma outra rota em seu *cache* e, assim como o AODV, terá complexidade O(N) para inundar a rede com RREQs. Assim, a quantidade de pacotes de roteamento do tipo RREQ é bem maior no AODV e é a principal responsável pela grande diferença de número de pacotes de roteamento. Como o DSR se aproveita do conhecimento de múltiplas rotas, ele permite que o nó destino responda a uma descoberta de rotas com várias rotas, o que faz aumentar o número de RREPs. Mas este aumento é bem menos sensível, uma vez que RREPs são *unicast* do nó destino até a fonte, ao contrário de RREQs que inundam a rede. Além disso, RREPs ocorrem como resposta a um procedimento de descoberta de rota e o DSR o usa bem menos, já que utiliza o *cache*. Vale ainda lembrar quando se discute a sobrecarga de roteamento que os pacotes de roteamento possuem prioridade em relação aos outros pacotes nas filas de saída de cada nó, conforme implementado no NS como forma de se acompanhar melhor o dinamismo da rede.

Nos gráficos de sobrecarga de roteamento, o DSR apresentou um desempenho consideravelmente melhor. Mas, para uma análise mais profunda e justa, deve-se lembrar que o aprendizado em modo promíscuo e uso de *cache* do DSR estão intrinsecamente ligados ao seu roteamento por fonte e este, claramente, acarreta informações de roteamento maiores, pois devem conter toda a rota do pacote. Assim, desejou-se contabilizar a sobrecarga de roteamento em número de *bytes*, que é apresentada nas Figuras VII.3(a) e VII.3(b) para 10 e 30 pares de fonte-destino, respectivamente.



(a) 10 pares fonte-destino

(b) 30 pares fonte-destino

Figura VII.3: Sobrecarga de roteamento (em número de *bytes*) para nós com velocidade máxima de 20 m/s.

O interessante agora é notar que, quando o parâmetro de sobrecarga de roteamento se altera levemente e passa a contabilizar o número de *bytes*, a situação se inverte, ou seja, o AODV passa a apresentar desempenho um pouco melhor. Ora, no DSR cada pacote, seja de roteamento ou seja proveniente da aplicação, precisa carregar consigo todo o seu caminho, logo este roteamento por fonte acarreta uma sobrecarga de 4 *bytes* por cada nó no caminho, enquanto que o AODV requer apenas no pacote o indispensável, isto é, o endereço de destino e da fonte, contabilizando 8 *bytes*. Esta sobrecarga do DSR é tão maior quanto maior for o número médio de nós em cada caminho usado na rede, prejudicando claramente a escalabilidade do protocolo. Por isso, no *draft* do DSR [3] assume-se que o diâmetro da rede é pequeno.

Resumindo, o DSR se saiu consideravelmente melhor que o AODV quanto à sobrecarga em número de pacotes e o AODV se saiu melhor quanto à sobrecarga em número de *bytes*. Termina-se por concluir que o DSR se mostrou melhor quanto à sobrecarga de roteamento em geral, pois é mais custoso para a rede enviar maior número de pacotes do que de *bytes*. Assim, o custo no AODV de se mandar muitos pequenos pacotes é maior que o de se fazer o roteamento por fonte pelo DSR, pois se consome mais tanto em potência quanto em utilização da rede, para se disputar o meio várias vezes para transmitir pacotes do que para se enviar alguns *bytes* a mais em um pacote já existente. Uma análise mais profunda levaria em conta também a sobrecarga em termos de pacotes da camada de enlace, mas isto varia de acordo com os protocolos de camada de enlace escolhidos, o que foge do escopo do protocolo de roteamento.

VII.2.3 Otimização do Caminho

Por fim, desejou-se analisar a diferença entre o caminho percorrido pelo pacote e o caminho ótimo, como forma de medir quão próximo de seu objetivo de roteamento pelo caminho mais curto os protocolos conseguiam chegar. Esta métrica influencia, conforme observado no início do capítulo, na economia de energia e de banda da rede. Assim, o uso do menor caminho faz um melhor uso dos recursos da rede e diminui a latência.

O procedimento para se calcular esta diferença se baseou em uma facilidade do simulador que permite registrar o caminho ótimo no momento da geração do pacote e o número de enlaces percorridos por cada pacote até chegar ao seu destino. O grupo Monarch, autor desta funcionalidade, já a utilizara anteriormente [6]. O resultado é apresentado na Figura VII.4. Nela, o eixo Y representa o percentual de pacotes recebidos e o eixo X relaciona o número de enlaces percorridos a mais que o caminho ótimo. Assim, por exemplo, o percentual de pacotes recebidos através de zero enlaces a mais que o ótimo representa o percentual de pacotes que chegou ao destino através do melhor caminho.

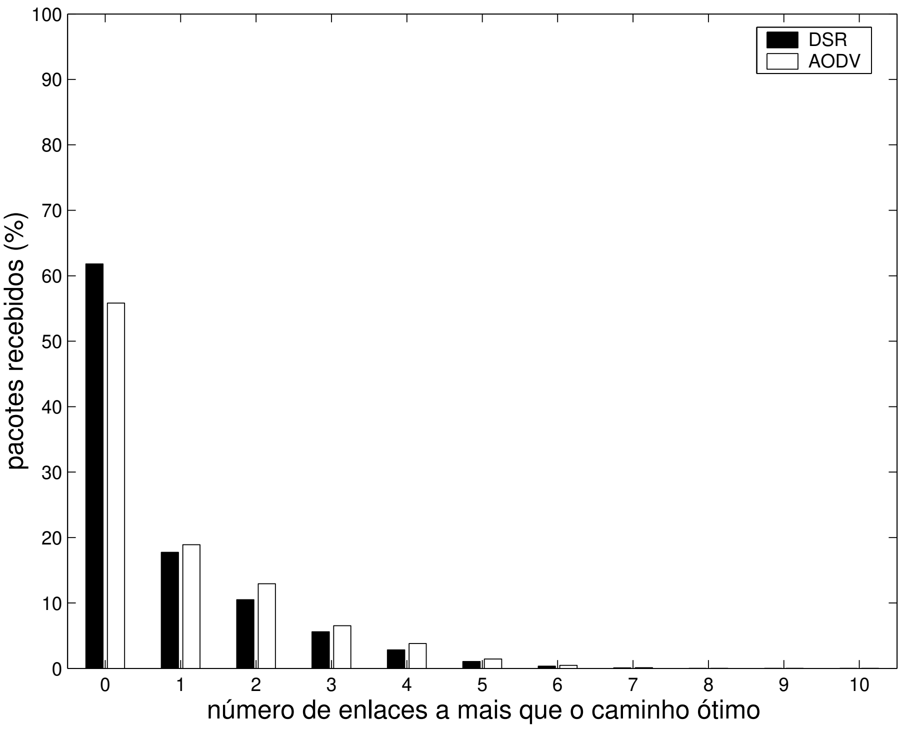


Figura VII.4: Comparação com o caminho ótimo.

Verificou-se que, devido ao dinamismo na mudança de topologia em redes *ad hoc*, o caminho ótimo no momento da geração de um pacote não é necessariamente o mesmo do momento em que ele é precisamente enviado pela rede. Ora, quando se considera o período de espera na fila e o tempo para se efetuar, caso necessário, um procedimento de descoberta de rotas (que se for mal sucedido será ainda repetido até um certo limite de vezes), percebe-se que pode ocorrer que o melhor caminho mude durante este intervalo, ficando menor ou mesmo maior. Pode ocorrer inclusive que um pacote para o qual a distância era infinita no momento de sua geração passe a ter rota até o destino antes de ser descartado. Todos estes tipos de eventos foram verificados. Uma vez que, quando um pacote chega através de um caminho menor que o ótimo no momento de sua geração, ele provavelmente utilizou efetivamente o melhor caminho, neste caso, adotou-se a política de também contabilizar estes pacotes no gráfico como tendo percorrido o caminho ótimo.

Pode-se verificar pela Figura VII.4 que ambos os protocolos conseguem entregar a maioria de seus pacotes através de caminhos ótimos, sendo que o DSR apresentou um melhor desempenho quanto à otimalidade do caminho. Além disso, para ambos os protocolos, mais de 85% dos pacotes foram recebidos através de caminhos com no máximo 2 enlaces a mais que o ótimo, o que pode ser considerado bem próximo ao melhor caminho e caracteriza o bom funcionamento dos dois protocolos.

Capítulo VIII

Conclusão

Este trabalho se focalizou na problemática de redes móveis sem fio *ad hoc*. Com o aumento da capacidade de processamento dos dispositivos móveis, o crescimento da popularidade de redes móveis em geral tornou-se evidente. Conforme explicado, tal interesse se justifica pela flexibilidade, mobilidade, facilidade de planejamento, robustez e redução do tempo de planejamento que as redes móveis possibilitam. Como uma subdivisão de redes móveis, as redes *ad hoc* são procuradas por oferecer o máximo de flexibilidade, acompanhadas de um alto grau de robustez. Assim, redes *ad hoc* são a melhor opção quando não há infra-estrutura ou esta verifica-se muito cara ou muito demorada para se construir. Além disso, redes *ad hoc* são altamente atraentes para situações de emergência, desastre natural ou operações militares.

Mostrou-se que um dos maiores desafios destas redes é o roteamento. Afinal, permite-se que a topologia mude de forma arbitrária e freqüente. Além disso, devido ao meio sem fio, faz-se imperativo economizar banda e preocupar-se com segurança na comunicação. Outro recurso escasso é a energia nos nós e deve ser igualmente economizado. Outros problemas, tais como a existência de enlaces assimétricos e e interferência e ruído, vêm tornar mais complexo ainda o roteamento.

Esta dissertação analisou os protocolos de roteamento DSR e AODV, duas das melhores opções atualmente para o roteamento em redes *ad hoc*. As suas características principais foram detalhadamente explicadas. Ambos os protocolos investem no roteamento sob demanda como forma de se economizar banda e energia nos nós. Além disso, os dois se baseiam em dois procedimentos: descoberta e manutenção de rotas. Em suma, o DSR baseia-se no roteamento por fonte, escuta em modo promíscuo e uso intensivo de *cache*. Já o AODV se caracteriza por usar tabelas de roteamento mais tradicionais e números de seqüência para evitar *loops* e identificar quão recentes são as mensagens. Estas diferenças influenciaram diretamente os resultados obtidos.

Foi usado o simulador de redes NS para se efetuar simulações que permitissem avaliar o desempenho dos protocolos. Este permite a variação de parâmetros relevantes para uma rede *ad hoc*. Alguns valores interessantes foram fixados e fez-se variar a mobilidade e a carga na rede para ver como os protocolos reagiam.

Para cada protocolo de roteamento, foram avaliadas a taxa de entrega de pacotes, a sobrecarga de roteamento e a otimização do caminho, métricas escolhidas para se verificar o funcionamento correto e eficiente dos protocolos, além de seu uso dos recursos da rede.

Para velocidade de 1m/s, ambos os protocolos apresentaram bom desempenho, conseguindo altas taxas de entrega de pacotes, sendo comprovado o seu bom funcionamento. Já para velocidade de 20m/s, a mudança de topologia e a queda de enlaces ocorrem de forma mais dinâmica e freqüente. Foram observados, pois, alguns fatos interessantes. Conforme esperado, houve um decréscimo na taxa de entrega de pacotes com o aumento da mobilidade, o mesmo ocorrendo com o aumento da carga. Para uma menor carga na rede, obteve-se para ambos os protocolos uma taxa de entrega muito boa e quase equivalente. Já para uma maior carga, a taxa de entregas piorou sensivelmente e ocorreu um fenômeno interessante: o AODV se mostrou melhor para alta mobilidade e o DSR o superou para baixa mobilidade. Isto, conforme explicado, se deveu diretamente as diferentes abordagens de roteamento: o DSR usa o aprendizado em modo promíscuo e consulta rotas alternativas no *cache*, enquanto que o AODV usa tabelas de roteamento com no máximo uma entrada por destino.

Para a sobrecarga de roteamento, verificou-se, conforme esperado, o aumento do envio de pacotes de roteamento com o aumento de fontes querendo transmitir, conseqüência direta do roteamento sob demanda, que ambos implementam. O mesmo ocorreu com o aumento da mobilidade, que faz a topologia da rede mudar mais freqüentemente. Foi interessante notar que o DSR com seu uso *cache*, escuta em modo promíscuo e roteamento por fonte enviou consideravelmente menos pacotes de roteamento que o AODV. No entanto, este roteamento por fonte implica uma maior quantidade de *bytes* de roteamento por parte do DSR que do AODV. Mas, por fim, justificou-se que é mais custoso para a rede enviar vários pequenos pacotes como faz o AODV do que enviar alguns *bytes* a mais em pacotes já existentes. Assim, o DSR se mostrou superior e mais bem comportado quanto à sobrecarga da rede.

Como última métrica, foi analisada a otimização do caminho. Foi analisado o mérito desta métrica e a dificuldade em implementá-la. Verificou-se que ambos os protocolos conseguem enviar a maior parte dos pacotes através de caminhos próximos do ótimo, tentando assim fazer melhor uso dos recursos da rede.

Como trabalhos futuros, existem vários nesta área tão desafiadora que é a de redes *ad hoc*. Assim, pode-se continuar este trabalho de simulação, fazendo variar outros parâmetros tais como a densidade (caracterizada pela área de simulação e o número de nós) ou o tamanho dos pacotes. De forma alternativa, podem-se simular cenários específicos, além de se inserir no modelo obstáculos e interferência. Ainda na área de protocolos de roteamento, podem-se simular outros protocolos mais recentes. Pode-se ainda, com os conhecimentos adquiridos, propor melhorias ou novos algoritmos de roteamento em redes *ad hoc*, seja utilizando a mesma métrica de menor número de saltos ou outras tais como potência nos nós. Pode-se ainda fazer uso de informações de localização para se investir no roteamento geográfico.

Mas pode-se ainda investir em outras áreas afins, tais como a qualidade de serviço, *multicast* ou segurança em redes *ad hoc*. Outra área de estudo são algoritmos que tentam descobrir um padrão de movimento dos nós, o que seria de grande utilidade para o roteamento *ad hoc*.

Referências Bibliográficas

[1] J. Schiller, *Mobile Comunications*. Addison-Wesley Wesley Pub Co., 1^a ed., 1997.

[2] S. Corson e J. Macker, “Routing protocol performance issues and evaluation considerations”, *rfc2501.txt*, janeiro 1999.

[3] D. A. Maltz, D. B. Johnson, Y. C. Hu. e J. G. Jetcheva, “The dynamic source routing protocol for mobile ad hoc networks”, *Internet Draft, draft-ietf-manet-dsr-06.txt*, fevereiro 2002.

[4] C. E. Perkins, E. M. Belding-Royer e S. R. Das, “Ad hoc on-demand distance vector routing”, *Internet Draft, draft-ietf-manet-aodv-10.txt*, junho 2002.

[5] *Network Simulator*. http://www.isi.edu/nsnam/ns/.

[6] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu. e J. Jetcheva, “A performance comparison of multi-hop wireless ad hoc network routing protocols”, *In Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, outubro 1998.

[7] S. R. Das, C. E. Perkins e E. M. Royer, “Performance comparison of two on-demand routing protocols for ad hoc networks”, *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pp. 3–12, março 2000.

[8] C. M. Cordeiro e D. P. Agrawal, *Minicursos - cap.3: Mobile Ad Hoc Networking*. SBRC, 2002.

[9] E. M. Royer e C.-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks", *IEEE Personal Communications*, pp. 46–55, abril 1999.

[10] Department of Computer Science - Rice University, http://www.monarch.cs.rice.edu, *Monarch Project - Mobile Networking Architectures*.

[11] *AODV Site*. http://www.moment.cs.ucsb.edu/AODV/aodv.html.

[12] C. E. Perkins e E. M. Royer, "Ad-hoc on-demand distance vector routing", *MILCOM'97 panel on Ad Hoc Networks*, nov 1997.