Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

Roteamento Adaptativo em Redes Ad Hoc sem Fio: Modelagem e Simulação

Autor: Roger Souza de Paula
Orientador: Prof. Dr. Ivanil Sebastião Bonatti

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: **Telecomunicações e Telemática**.

Banca Examinadora

Ivanil Sebastião Bonatti, Dr. ..... DT/FEEC/Unicamp
Carlos Magnus Carlson Filho, Dr. ..... FATEC/São José do Rio Preto
Paulo Cardieri, Dr. ..... DECOM/FEEC/Unicamp
Shusaburo Motoyama, Dr. ..... DT/FEEC/Unicamp

Campinas, SP - Brasil

Fevereiro 2006

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

Paula, Roger Souza de
P281r Roteamento adaptativo em redes ad hoc sem fio: modelagem e simulação / Roger Souza de Paula. – Campinas, SP: [s.n.], 2006.

Orientador: Ivanil Sebastião Bonatti.
Dissertação (Mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Sistemas de comunicação sem fio. 2. Programação (Matemática). 3. Heurística. 4. Simulação (computadores). 5. Desempenho. 6. SDL (Linguagem de Programação de Computador). I. Bonatti, Ivanil Sebastião. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: Adaptive routing on wireless ad hoc networks: modelling and simulation
Palavras-chave em Inglês: Wireless communication systems, Programming (Mathematical), Heuristic, Simulation (computers), Performance, SDL (Computer Programming Language)
Área de concentração: Telecomunicações e Telemática
Titulação: Mestre em Engenharia Elétrica
Banca examinadora: Carlos Magnus Carlson Filho, Paulo Cardieri e Shusaburo Motoyama
Data da defesa: 01/02/2006

Resumo

Uma rede ad hoc é uma coleção de estações móveis sem fio formando dinamicamente uma rede temporária sem necessidade de qualquer infra-estrutura de rede pré-existente ou administração centralizada. Protocolos de roteamento utilizados em redes ad hoc sem fio devem ser capazes de se ajustarem automaticamente a ambientes extremos, como os de alta mobilidade e baixa largura de banda. Assim, avanços recentes em pesquisas sobre redes sem fio concentram-se cada vez mais na adaptação de tais protocolos, diante da inter-relação entre as várias medidas de desempenho, como aquelas relacionadas às alterações na topologia (quebra de enlaces, mobilidade dos nós etc.) e aos parâmetros de qualidade de serviço (vazão, atraso etc.). Esta tese trata da modelagem e simulação de redes ad hoc sem fio, contribuindo de forma significativa nas áreas de modelo matemático, avaliação de desempenho e especificação de protocolos. Na primeira, é proposta uma nova estratégia que seleciona a rota de menor colisão dentre os mínimos caminhos existentes entre duas estações da rede. Na segunda área, sendo a de maior contribuição, uma análise de desempenho detalhada é realizada entre o protocolo de roteamento proposto nesta tese, *Minimal Congestion On-Demand Routing* (MCOR), e o *Dynamic Source Routing* (DSR), um protocolo de referência da literatura. Por fim, é apresentado um modelo de validação do protocolo proposto através da ferramenta de especificação e validação, *Specification and Description Language* (SDL).

Palavras-chave: Sistemas de comunicação sem fio, programação (matemática), heurística, simulação (computadores), desempenho, SDL (linguagem de programação de computador).

Abstract

An ad hoc network is a collection of wireless mobile nodes forming dynamically a temporary network without the use of any preexisting network infrastructure or centralized administration. Routing protocols used in wireless ad hoc networks must be capable to adjust automatically to extreme environments, as high mobility and low bandwidth. Hence, recent avances in wireless research focus more and more on the adaptation of such protocols, due to the interrelationship among various performance measures, like as those related to topological changes (link breakages, node mobility etc.) and quality of service parameters (throughput, delay etc.). This thesis argues the modelling and simulation of wireless ad hoc networks, contributing significantly on areas of mathematical model, performance evaluation and protocols specification. First, is proposed a new strategy that selects the route of minimal collision, among all minimum hop paths between two stations. Second, being the area of major contribution, a detailed performance analysis is executed between the routing protocol proposed in this thesis, Minimal Congestion On-Demand Routing (MCOR), and the Dynamic Source Routing (DSR), a reference protocol in the literature. Lastly, is presented a validation model of the proposed protocol through validation and specification tool, Specification and Description Language (SDL).

Keywords: Wireless communication systems, programming (mathematical), heuristic, simulation (computers), performance, SDL (computer programming language).

Faxina na Alma

Não importa onde você parou...
em que momento da vida você cansou...
Recomeçar é dar uma nova chance a si mesmo...
é renovar as esperanças na vida e o mais importante...
acreditar em você de novo.

Sofreu muito nesse período?
foi aprendizado...
Chorou muito?
foi limpeza da alma...
Ficou com raiva das pessoas?
foi para perdoá-las um dia...
Sentiu-se só por diversas vezes?
é porque fechaste a porta até para os anjos...
Acreditou que tudo estava perdido?
era o início da tua melhora...

Pois é... agora é hora de reiniciar... de pensar na luz...
de encontrar prazer nas coisas simples de novo.
Um corte de cabelo arrojado... diferente?
Um novo curso... ou aquele velho desejo de aprender
pintar... desenhar... dominar
o computador... ou qualquer outra coisa...
Olha quanto desafio...
quanta coisa nova nesse mundão de meu Deus te esperando.

Tá se sentindo sozinho?
besteira...
tem tanta gente que você afastou com o seu "período de isolamento"...
tem tanta gente esperando apenas um sorriso teu para "chegar" perto de você.
Quando nos trancamos na tristeza...
nem nós mesmos nos suportamos...
ficamos horríveis...
o mal humor vai comendo nosso fígado...
até a boca fica amarga.

Recomeçar... hoje é um bom dia para começar novos desafios.
Onde você quer chegar? ir alto... sonhe alto... queira o melhor do melhor...
queira coisas boas para a vida...
pensando assim trazemos prá nós aquilo que desejamos...
se pensamos pequeno... coisas pequenas teremos...
já se desejarmos fortemente o melhor e principalmente lutarmos pelo melhor...
o melhor vai se instalar na nossa vida.

E é hoje o dia da faxina mental...
joga fora tudo que te prende ao passado...
ao mundinho de coisas tristes...
fotos... peças de roupa, papel de bala... ingressos de cinema...
bilhetes de viagens...
e toda aquela tranqueira que guardamos quando nos julgamos apaixonados...
jogue tudo fora...
mas principalmente... esvazie seu coração...
fique pronto para a vida... para um novo amor...

Lembre-se somos apaixonáveis...
somos sempre capazes de amar muitas e muitas vezes...
afinal de contas...
Nós somos o "Amor"...
Porque somos do tamanho daquilo que vemos, e não do tamanho da nossa altura.
Sempre vai existir um ser além de nós, e confia nele agora, que ele guiará os teus passos...

Carlos Drumond de Andrade

Aos meus pais, Nelsina e Valter
e a minha irmã Ellen

Agradecimentos

Primeiramente agradeço a Deus, pela saúde em primeiro lugar, pela sabedoria, pela lucidez e principalmente pelo dom da vida.

A Nossa Senhora de Aparecida, virgem santíssima mãe de Jesus, pelos vários momentos de reflexão, pela infinita perseverança de vencer na vida e por iluminar o meu árduo caminho do conhecimento.

Ao meu orientador, Prof. Ivanil Sebastião Bonatti, pela amizade, presteza, dedicação, paciência e por confiar na minha competência para a execução deste trabalho. Pessoa de caráter conservador e ética profissional inigualável, que contribuiu significativamente para o meu crescimento ético e intelectual.

Ao Prof. Paulo Cardieri, meu futuro orientador, pelos preciosos ensinamentos em redes sem fio. Ao amigo Cristiano Agulhari, que contribuiu em vários pontos na implementação e validação dos algoritmos desta tese.

À toda minha família, sem exceção, pela afetividade, carinho e incentivo. Em especial, dedico este trabalho a minha mãe Nelsina. Esta pessoa maravilhosa, que além de ter me proporcionado a luz da vida, sempre me apoiou nos momentos de alegria e tristeza, de saúde e doença e de sucesso e fracasso em toda minha vida acadêmica. Suas palavras sempre revigoram meu espírito guerreiro. Ao meu pai Valter, que apesar de estar passando por uma fase transitória, sempre torceu por mim. A minha irmã Ellen, pela sua meiguice, companheirismo e amor fraterno. Aos meus avós, também padrinhos, José e Gercina, pessoas pelas quais tenho um carinho e admiração muito grandes. E por fim, ao meu tio Altair, pela sua simplicidade e disponibilidade em ajudar os outros. Este trabalho é dedicado a todos vocês.

Aos grandes amigos que me acompanharam nesta rápida passagem por Campinas, e que me ajudaram de alguma forma ou de outra: Luis Maranesi, Gilmar Carlos, Andriélber Oliveira, Gustavo Modesto, Henrique Marinho, Renato Alves, Leandro Marques e Gustavo Zuliani.

À Cooperativa Brasil pelos vários momentos de descontração e prazer. O forró indiscutivelmente tornou-se uma verdadeira paixão na minha vida.

À Faculdade de Engenharia Elétrica e de Computação (FEEC). Um lugar realmente mágico, onde "o conhecimento é sentido no ar" – foi esta a primeira impressão que tive daqui. Convivi com pessoas portadoras de uma sabedoria ampla e inimaginável.

À CAPES, pelo apoio financeiro.

Sumário

Lista de Figuras

Lista de Tabelas

Glossário

ACK	Acknowledgment
AODV	Ad Hoc On-Demand Distance Vector
ARP	Address Resolution Protocol
CBR	Constant Bit Rate
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear to Send
DARPA	Defense Advanced Research Projects Agency
DCF	Distributed Coordination Function
DIFS	DCF InterFrame Space
DLAR	Dynamic Load-Aware Routing
DOSPR	Delay-Oriented Shortest Path Routing
DSDV	Destination-Sequenced Distance Vector
DSR	Dynamic Source Routing
DSSS	Direct-Sequence Spread-Spectrum
EIFS	Extended InterFrame Space
FHSS	Frequency-Hopping Spread-Spectrum
FTP	File Transfer Protocol
HR-DSSS	High-Rate Direct-Sequence Spread-Spectrum
IETF	Internet Engineering Task Force
IP	Internet Protocol
LAOR	Load-Aware On-Demand Routing
LBAR	Load-Balanced Ad Hoc Routing
LSR	Load-Sensitive Routing

MAC	Medium Access Control
MACA	Multiple Access with Collision Avoidance
MACAW	Multiple Access with Collision Avoidance for Wireless
MANET	Mobile Ad Hoc Network
MCOR	Minimal Congestion On-Demand Routing
MSC	Message Sequence Chart
NAV	Network Allocation Vector
OFDM	Orthogonal Frequency Division Multiplexing
PCF	Point Coordination Function
PHY	Physical Layer
PIFS	PCF InterFrame Space
PRNET	Packet Radio Network
RTS	Request to Send
SIFS	Short InterFrame Space
SDL	Specification and Description Language
TCP	Transmission Control Protocol
TORA	Temporally-Ordered Routing Algorithm
ITU-T	International Telecommunication Union-Telecom Standardization
UDP	User Datagram Protocol
WLAN	Wireless Local Network
ZRP	Zone Routing Protocol

Lista de Símbolos

$\mathcal{V}$	Conjunto das estações móveis na rede ad hoc
$\mathcal{E}$	Conjunto dos enlaces da rede ad hoc
$\mathcal{V}^i$	Conjunto de todas as estações que podem detectar o sinal irradiado pela estação i
$\mathcal{V}_j$	Conjunto de todas as estações cujos sinais podem ser detectados pela estação j
x_{ij}	Demanda do tráfego encaminhado no enlace (i, j)
x_{ijs}	Demanda do tráfego encaminhado no enlace (i, j) que tem como origem o nó s
ρ_{lm}	Matriz de tráfego entre o nó origem l e o nó destino m
x	Vazão do tráfego escoado da rede
y	Vazão do tráfego oferecido da rede
η	Parâmetro de ajuste dos protocolos de acesso aleatório
τ	Probabilidade de uma estação transmitir em um *slot*
n	Número de estações na rede
y_{ij}	Demanda do tráfego oferecido no enlace (i, j)
Ω	Conjunto de todas as soluções do problema de caminho mínimo
γ_j^i	Quantidade de tráfego que pode colidir durante a transmissão da estação i para a estação j
p	Probabilidade média de colisão
m	Número máximo de iterações do algoritmo
s	Nó origem do tráfego na rede
π	Matriz de predecessores da rede
W	Matriz de pesos dos enlaces (i, j)

Trabalhos Publicados pelo Autor

1. R.S. Paula, C.M. Agulhari, I.S. Bonatti e P.L.D. Peres. "Encaminhamento de Colisão Mínima em Redes Ad Hoc sem Fio". *XXII Simpósio Brasileiro de Telecomunicações* (SBrT'05), Campinas, São Paulo, Brasil, pg. 715-720, Setembro 2005.

2. R.S. Paula e I.S. Bonatti. "Roteamento Sob Demanda e de Congestionamento Mínimo para Redes Ad Hoc sem Fio". *24º Simpósio Brasileiro de Redes de Computadores* (SBRC'06), Curitiba, Paraná, Brasil, submetido, Maio 2006.

3. R.S. Paula, I.S. Bonatti e C.M. Agulhari. "Validação de um Protocolo de Roteamento Sob Demanda e de Congestionamento Mínimo para Redes Ad Hoc sem Fio". *24º Simpósio Brasileiro de Redes de Computadores* (SBRC'06), Curitiba, Paraná, Brasil, submetido, Maio 2006.

Capítulo 1

Introdução

Use your mentality
Wake up to reality

— From the song "I've Got You under My Skin",
by Cole Porter

1.1 Redes Ad Hoc Móveis

As raízes das redes ad hoc podem ser traçadas em meados de 1968 através do projeto ALOHANET, dirigido pela Universidade do Hawaii [1]. Tal projeto consistia de uma rede via rádio para conectar os computadores da universidade nas maiores ilhas do Hawaii. A rede era *single-hop*, logo apenas os nós em alcance direto poderiam se comunicar. Com o crescimento do projeto ALOHANET, o DARPA (*Defense Advanced Research Project Agency*) [2] subsidiou o desenvolvimento das chamadas *Packet Radio Networks* (PRNETs) em 1973 [3]. Ao contrário do ALOHANET, nas PRNETs a comunicação era *multi-hop* em uma área geográfica maior, onde os dispositivos eram montados em plataformas móveis, como caminhões. Uma das características primordiais das PRNETs era seu rápido desenvolvimento. Uma vez instalado, o sistema era auto-inicializável e auto-organizável. Assim, os nós eram capazes de descobrir a conectividade via rádio entre os nós vizinhos e organizar as estratégias de roteamento baseado nessa conectividade.

Uma Rede Ad Hoc Móvel (MANET) é uma coleção de estações móveis sem fio formando uma rede temporária sem a necessidade de qualquer administração centralizada ou infra-estrutura fixa [4, 5, 6]. Em tal rede, cada nó móvel opera não somente como um *host*, mas também como um roteador, encaminhando pacotes para outras estações que podem não estar diretamente no alcance de transmissão um dos outros. Cada nó participa em um protocolo ad hoc que permite descobrir caminhos *multi-hop* para qualquer outro nó da rede. Por exemplo, na Figura 1.1, os nós **A** e **C** devem incluir o nó **B** para se comunicarem.

Uma rede ad hoc é uma rede autônoma, possuindo algumas propriedades desejáveis como auto-organização, auto-configuração e auto-adaptação. O termo ad hoc significa "que pode tomar várias formas" e "que pode ser móvel, independente ou conectado", sendo utilizado para um determinado fim. Nós ou dispositivos ad hoc

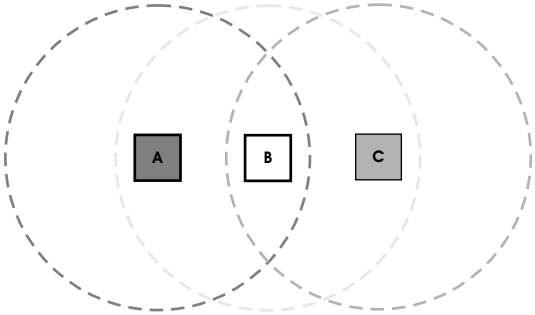


Fig. 1.1: Uma rede ad hoc de três nós, onde os nós **A** e **C** devem descobrir uma rota através de **B** para se comunicarem. Os círculos indicam o alcance de transmissão de cada transceptor via rádio do nó. Os nós **A** e **C** não estão no alcance de transmissão direta um do outro, desde que o círculo de **A** não cobre **C**.

devem ser capazes de detectar a presença de outros dispositivos e executar os procedimentos necessários para permitir a comunicação e compartilhar informação e serviços.

Redes ad hoc móveis oferecem benefícios únicos e versatilidade para certos ambientes e aplicações. Primeiro, elas podem ser criadas em qualquer instante e em qualquer lugar, já que não utilizam qualquer estrutura estacionária, como estações rádio-base. Segundo, tais redes podem ser intrinsecamente resilientes a falhas, não operando em restrições de uma topologia fixa. Alguns campos de aplicação das redes ad hoc incluem as áreas militar, emergência e comercial, envolvendo a troca de informações localmente. A rede ad hoc permite robustez para o tratamento da informação em caso de perda ou movimento dos nós. A Figura 1.2 mostra uma representação conceitual. Em caso de emergência, como terremotos ou enchentes por exemplo, os bombeiros podem enviar informações sobre as condições do ambiente e agilizar a resgate dos feridos quando a infra-estrutura foi destruída ou não estiver disponível. Pessoas podem se comunicar e trocar dados em pequenas áreas, como em um escritório, centro de convenções ou sala de aula, sem a preocupação de encontrarem uma estrutura física. Padrões como o Bluetooth [7] e *frameworks* como a Piconet [8] existem e continuam sendo desenvolvidas atualmente para aplicações comerciais.

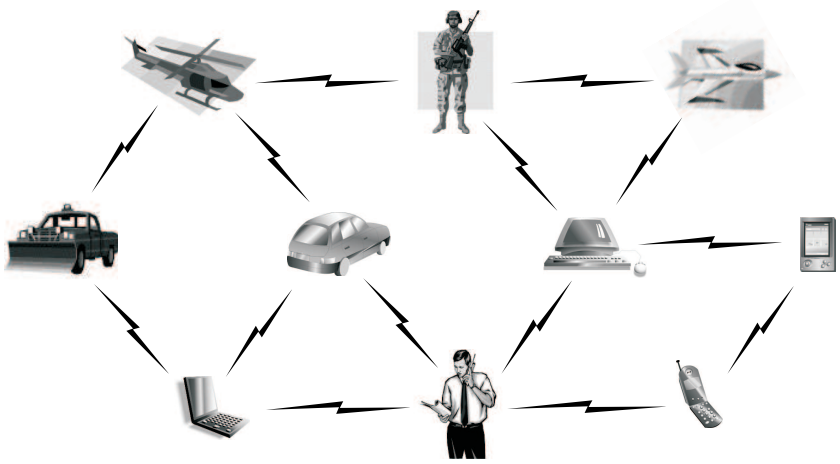


Fig. 1.2: Uma representação conceitual de uma rede ad hoc móvel.

Existe um grupo de trabalho do IETF (*Internet Engineering Task Force*) [9], chamado de MANET (*Mobile Ad Hoc Networks Working Group*) [10], que tem por objetivo principal desenvolver especificações para protocolos de roteamento e tecnologias relacionadas à camada de rede em redes ad hoc. O grupo de trabalho também considera questões relacionadas ao endereçamento, segurança e a interação entre os protocolos das camadas inferior e superior. Várias RFCs e Internet-Drafts envolvendo sugestões e padronizações de protocolos já foram publicadas, sendo encontradas em [10].

A ausência de uma infra-estrutura fixa implica que os nós se comunicam diretamente uns com os outros, ponto-a-ponto. A implementação das redes ad hoc impõe limitações no consumo de energia às estações, e portanto, no alcance de transmissão; esses nós geralmente devem satisfazer limitações de tamanho para portabilidade. Todas as estações movem-se independentemente umas das outras; logo as alterações na topologia resultante devem ser capazes de prover a comunicação entre todos os nós da rede de forma apropriada. Outro problema é a restrição da quantidade de largura de banda disponível que deve atender adequadamente as aplicações. A seguir, são resumidas as principais características de uma MANET [11].

- **Topologia dinâmica**: os nós são livres para se moverem arbitrariamente com diferentes velocidades; logo, a topologia da rede (tipicamente *multi-hop*) pode se alterar de maneira imprevisível.
- **Largura de banda limitada**: enlaces sem fio possuem menor capacidade do que as redes infra-estruturadas. Em adição, a vazão das comunicações sem fio, considerando o efeito de múltiplo acesso, desvanecimento, ruído, condições de interferência etc., é geralmente muito menor do que a taxa máxima de transmissão.
- **Consumo de energia limitado**: alguns ou todos nós da rede podem utilizar baterias, sendo que o critério mais importante para a otimização do sistema pode ser a conservação de energia.
- **Segurança**: os nós móveis são geralmente mais propensos à ameaças de segurança do que as redes cabeadas. Portanto, ataques do tipo *eavesdropping*, *spoofing* e *denial of service* (DoS) [12] devem ser cuidadosamente considerados.

Assim, dadas a natureza dinâmica da topologia e a imprecisão do estado da rede, a garantia de um roteamento efetivo é um dos principais desafios na implementação de redes ad hoc.

1.2 O Problema do Roteamento em Redes Ad Hoc

O problema básico de roteamento é encontrar uma seqüência ordenada de nós intermediários que podem transportar um pacote de dados através da rede de sua origem até seu destino. Em soluções tradicionais para o problema de roteamento, como o *hop-by-hop*, cada nó na rede mantém uma tabela de roteamento: para cada destino conhecido, a tabela lista o próximo nó para o qual o pacote deve ser encaminhado.

A tabela de roteamento em cada nó pode ser vista como uma parte de uma estrutura de dados distribuída que, quando em conjunto, descrevem a topologia da rede. A função do protocolo de roteamento é garantir de que toda a estrutura de dados possua uma visão correta e consistente da topologia atual da rede. Caso

contrário, os pacotes podem entrar em *loop* e/ou serem perdidos. Esse problema torna-se ainda mais difícil com o aumento do número de nós (cuja informação deve ser consistente) e com a variação da topologia atual da rede sem fio.

O protocolo de roteamento deve ser eficiente em ambientes em que os nós são estacionários e a largura de banda não é um fator limitante. Mas ainda, o mesmo protocolo deve funcionar bem quando a largura de banda entre os nós for baixa e os níveis de mobilidade e/ou alteração na topologia forem altos. Já que geralmente é impossível saber *a priori* o ambiente no qual o protocolo será implementado, este deve ser capaz de se adaptar automaticamente.

Muitos protocolos possuem alguns comportamentos periódicos, significando que algumas de suas operações são executadas regularmente em algum intervalo independente de eventos externos do meio. Esses comportamentos tipicamente limitam a habilidade de os protocolos adaptarem-se às oscilações do ambiente. Se o intervalo de periodicidade for muito curto, o protocolo será ineficiente visto que suas ações serão mais freqüentes do que as mudanças da topologia da rede. Já se for muito longo, o protocolo não reagirá suficientemente rápido a tais alterações, e os pacotes serão perdidos.

Protocolos periódicos podem ajustar seu intervalo de periodicidade para tentar sincronizar com a taxa de alteração das condições da rede [13]. Todavia, esta aproximação sofrerá com o *overhead* associado ao mecanismo de ajuste e o atraso entre uma alteração e a seleção de um novo intervalo. No pior caso, quando rajadas (*bursts*) são seguidas de períodos estáveis da topologia da rede, esta adaptação do período de periodicidade poderia resultar em um protocolo que utilizasse um longo intervalo durante os períodos de rajadas e um curto nos períodos estáveis. Este caso pode ser bem comum, como por exemplo, quando um grupo de pessoas entra em uma sala para uma reunião, e permanece por lá um certo período de tempo.

Uma alternativa ao protocolo de roteamento periódico é aquele que opera de modo sob demanda. Protocolos sob demanda são baseados na premissa que se um problema ou estado inconsistente puder ser detectado antes de causar algum dano permanente, então todo trabalho para corrigir o problema ou manter o estado consistente pode ser adiado até que seja realmente necessário. Eles operam utilizando a mesma filosofia "preguiçosa" como os algoritmos otimistas [14, 15].

Outra vantagem dos protocolos sob demanda é o fato de permitir que o *overhead* seja automaticamente escalável de acordo com a reação às mudanças na topologia. Essa escalabilidade diminui dramaticamente o *overhead* do protocolo, eliminando a necessidade de qualquer atividade periódica, tais como anúncios de rota e pacotes de detecção de vizinhos.

Em uma rede com fio, alterações na topologia são raras. Os *hosts* e os demais nós possuem sua própria localização na rede. Este é o comportamento esperado de uma rede com fio. Quebras de enlace ocorrerão somente quando houver uma interrupção física, como a queda de um *host* ou a ruptura de um cabo. Para esse tipo de infra-estrutura, protocolos clássicos de roteamento, como vetor-distância (*distance-vector*) ou estado do enlace (*link state*), funcionam muito bem. Para manter as tabelas de roteamento atualizadas, os roteadores trocam informações periodicamente enviando mensagens de atualização uns aos outros. No caso de uma quebra de enlace, as rotas devem ser recalculadas e propagadas na rede.

Obviamente, esta implementação não é adequada para redes ad hoc. Nestas, mudanças de enlaces são comuns já que os nós estão em constante movimento. Considere, por exemplo, o caso em que dois nós estão se

comunicando enquanto movem-se em sentidos opostos. Enquanto um nó estiver dentro do alcance de transmissão um do outro, a comunicação permanecerá estabelecida. Caso contrário, será suspensa. Quanto maior o número de nós neste cenário, mais enlaces se formarão e/ou desaparecerão e novas rotas para os destinos deverão ser computadas.

Diante das diferenças entre redes com e sem fio, um protocolo de roteamento ad hoc, além de ser escalável e prover rotas sem *loop* (características de um protocolo de roteamento clássico), deve tratar peculiaridades comuns às redes ad hoc móveis como:

- Topologia dinâmica;
- Limitação de largura de banda;
- Segurança;
- QoS fim-a-fim;
- Limitação de consumo de energia;
- Suporte a enlaces assimétricos;
- Interfaces com redes externas.

O roteamento eficiente de pacotes é, portanto, um desafio primordial em redes ad hoc sem fio. Existem muitos protocolos propostos na literatura (alguns deles são brevemente discutidos no próximo capítulo). Trabalhos como [16, 17, 18, 19] comparam o desempenho entre eles.

1.3 Organização da Tese

A tese foi dividida em seis capítulos, nos quais são apresentados conceitos, metodologias, simulações e especificações sobre redes ad hoc sem fio, com ênfase em técnicas de roteamento.

O Capítulo 2 descreve os principais conceitos e os protocolos de roteamento para redes ad hoc avaliados nesta tese. Uma discussão detalhada sobre protocolos que inferem o balanceamento de carga na rede também é realizada.

O Capítulo 3 propõe a formulação de um modelo matemático para minimizar a probabilidade média de colisão em uma rede ad hoc. O encaminhamento do tráfego á avaliado através dos múltiplos caminhos mínimos existentes na rede; redes de pequeno e médio porte são consideradas.

O Capítulo 4 inicialmente apresenta alguns estudos de casos simples com objetivo de validar o simulador. Posteriormente, são discutidos os resultados para o desempenho dos protocolos de roteamento, sendo um de referência na literatura e um outro proposto nesta tese. Para se realizar uma comparação efetiva da consistência e escalabilidade dos protocolos, diversos cenários de simulação são testados.

O Capítulo 5 discute a implementação de um modelo formal para o protocolo proposto através de uma linguagem de especificação e validação de ampla aceitação na indústria de telecomunicações.

O Capítulo 6 apresenta as conclusões, assim como as sugestões para trabalhos futuros.

Capítulo 2

Protocolos de Roteamento Ad Hoc

The art of war teaches us to rely not on the likelihood of the enemy's not coming, but on our own readiness to receive him; not on the chance of his no attacking, but rather on the fact that we have made our position unassailable.

— *The Art of War,* Sun Tzu

2.1 Resumo dos Principais Protocolos

Desde o advento das redes de pacote via rádio DARPA (*Defense Advanced Research Projects Agency*) no início dos anos 70, inúmeros protocolos foram propostos para redes ad hoc móveis. Tais protocolos devem tratar as típicas limitações dessas redes, como baixo consumo de energia, baixa largura de banda e altas taxas de erros. De acordo com a Figura 2.1, os protocolos de roteamento podem ser divididos em: proativos ou dirigidos por tabela e reativos ou sob demanda. As linhas pontilhadas representam uma relação de descendência entre os protocolos. Logo a seguir, alguns deles são brevemente comentados.

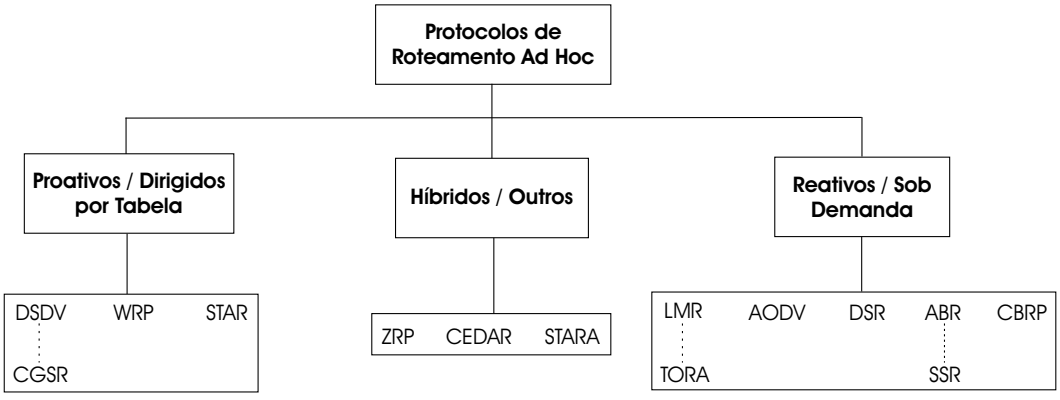


Fig. 2.1: Categorização dos protocolos de roteamento ad hoc.

Os protocolos proativos mantém uma informação de roteamento consistente e atualizada de todos os nós móveis na rede. Cada nó mantém uma ou mais tabelas de roteamento para armazenar a informação de roteamento, e respondem a mudanças na topologia propagando atualizações das rotas a fim de sustentar uma visão consistente da rede. A diferença entre eles consiste no número de tabelas de roteamento relacionadas e no método que as alterações na estrutura na rede são propagadas.

Em contrapartida, os protocolos reativos criam as rotas somente quando desejadas pelo nó origem. Quando um nó requer uma rota a um destino, ele inicia um processo de descoberta de rota na rede. Esse processo é completado quando uma rota for encontrada ou todas as tentativas de descoberta fracassarem. Uma vez a rota descoberta e estabelecida, ela é mantida por algum procedimento de manutenção de rota até o destino tornar-se inacessível ou a rota não for mais necessária.

Existem ainda alguns protocolos que combinam essas duas técnicas, denominados de híbridos. Tais protocolos dividem a rede em zonas (*clusters*), sendo em cada uma aplicado um protocolo proativo; um protocolo reativo então é empregado para realizar o roteamento entre as diferentes zonas.

A. Destination-Sequenced Distance Vector

O *Destination-Sequenced Distance Vector* (DSDV) [20] é um protocolo proativo baseado no algoritmo de roteamento clássico de Bellman-Ford [21]. O roteamento é realizado através do uso de tabelas de roteamento, mantidas em cada nó. A complexidade do protocolo é a geração e manutenção dessas tabelas.

Cada tabela de roteamento contém uma lista dos endereços de todos os nós na rede. Para um pacote atingir cada um desses endereços, a tabela mantém o endereço do próximo *hop*. Em adição, as tabelas mantém a métrica de rota (neste caso o número mínimo de *hops*) e um número de seqüência (*route sequence number*). Periodicamente, ou imediatamente quando alguma alteração na topologia da rede é detectada, cada nó envia um pacote de atualização da tabela de roteamento em *broadcast*. Cada pacote possui o endereço do nó destino, o número de *hops* e o número de seqüência. Os nós vizinhos, ao receberem a atualização, incrementam o número de *hops* e retransmitem o pacote. Esse processo repete-se até todos os nós da rede receberem uma cópia de um pacote de atualização com sua correspondente métrica. Caso um nó receba um pacote de atualização duplicado, a rota com menor número de *hops* é utilizada.

Com o aumento do número de nós na rede, o tamanho das tabelas de roteamento e a largura de banda necessária para transmitir os pacotes de atualização também aumentam. Este *overhead* de roteamento é a principal desvantagem do DSDV.

B. Temporally-Ordered Routing Algorithm

O *Temporally-Ordered Routing Algorithm* (TORA) [22] é um protocolo de roteamento distribuído baseado no algoritmo de enlace reverso [23]. É um protocolo sob demanda designado para prover múltiplas rotas a um destino, estabelecê-las rapidamente e minimizar o *overhead* de comunicação localizando as mudanças na topologia quando possível. A otimalidade de rota (caminho mínimo) é deixada em segundo plano, sendo que rotas mais longas geralmente são utilizadas para evitar o *overhead* de descoberta de novas rotas. Não é necessário (e nem desejável) manter rotas entre cada par origem-destino em todo instante. Em termos de

requerimento de memória, cada nó mantém uma estrutura de nível do nó bem como o estado de todos enlaces por conexão suportados pela rede.

C. Dynamic Source Routing

O *Dynamic Source Routing* (DSR) é designado para permitir que os nós descubram uma rota na origem com múltiplos *hops* a qualquer destino na rede ad hoc. Cada pacote a ser encaminhado possui em seu cabeçalho uma lista completa e ordenada dos nós que precisa percorrer. A maior vantagem do roteamento na origem é que os nós intermediários não precisam manter nenhuma informação de roteamento. O protocolo DSR é discutido com maiores detalhes na próxima seção.

D. Ad Hoc On-Demand Distance Vector

O protocolo *Ad Hoc On-Demand Distance Vector* (AODV) [24] é essencialmente uma combinação do DSR e do DSDV. Ele utiliza o mecanismo básico sob demanda da descoberta de rota e manutenção de rota do DSR, e o roteamento *hop-by-hop*, números de seqüência e pacotes periódicos de atualização do DSDV. O maior benefício do AODV sobre o DSR é que a rota na origem não precisa ser incluída em cada pacote. Isto resulta em uma redução do *overhead* do protocolo. Infelizmente, o AODV requer atualizações periódicas as quais consomem mais largura de banda do que a banda economizada pela não inclusão da rota na origem nos pacotes.

E. Zone Routing Protocol

O *Zone Routing Protocol* (ZRP) [25] é considerado como um protocolo ad hoc híbrido, já que combina elementos proativos e reativos. Uma zona de roteamento é similar a um *cluster* com exceção de que cada nó age como um *clusterhead* (líder do *cluster*). As zonas podem se sobrepor. Cada nó especifica um raio da zona em termos de número de *hops*. O tamanho da zona escolhido pode, portanto, afetar o desempenho da comunicação ad hoc.

No ZRP, uma zona compreende um conjunto de poucos nós com um, dois ou mais *hops* a partir do *clusterhead*. Dentro dessa zona, um protocolo de roteamento proativo é usado. Isto implica que as atualizações são executadas por nós pertencentes à zona. Cada nó, então, possui uma rota a todos os demais nós de sua zona. Se um nó residir fora da zona de origem, um método de roteamento sob demanda é utilizado.

2.2 Roteamento Dinâmico na Origem

Este seção fornece uma breve revisão da operação do protocolo DSR, suficiente apenas para o leitor compreender sua análise nos capítulos seguintes. A versão 10 da Internet-Draft que define o DSR [26] provê uma descrição detalhada da versão atual do DSR utilizada nos experimentos realizados nesta tese.

2.2.1 Roteamento na Origem

O protocolo *Dynamic Source Routing* (DSR) [26, 27, 28, 29] é baseado no roteamento na origem explícito, onde o emissor de cada pacote determina uma lista ordenada de nós através da qual o mesmo deve percorrer até o destino. A grande vantagem do esquema de rota na origem é que os nós intermediários não necessitam manter informação atualizada ao encaminhar os pacotes, já que o nó origem toma todas decisões de roteamento. Esta técnica, em conjunto com o fato de o protocolo agir sob demanda, elimina a necessidade de qualquer anúncio periódico de rota ou detecção de pacotes vizinhos.

O protocolo DSR consiste de dois mecanismos básicos (mostrados na Figura 2.2) para o realizar roteamento em uma rede ad hoc sem fio:

- **Descoberta de Rota**, no qual o nó **S** desejando enviar um pacote ao nó destino **D**, obtém uma rota na origem até **D**. Este processo é utilizado somente quando **S** tenta enviar um pacote a **D** e não possui uma rota para tal.

- **Manutenção de Rota**, no qual o nó **S**, enquanto estiver utilizando uma certa rota para **D**, detecta uma alteração na topologia da rede de modo que a rota não pode ser mais utilizada. O nó **S** então verifica se possui outra rota para **D**, ou pode realizar uma Descoberta de Rota para encontrar uma novo caminho. A Manutenção de Rota somente é realizada quando **S** estiver, de fato, enviando pacotes para **D**.

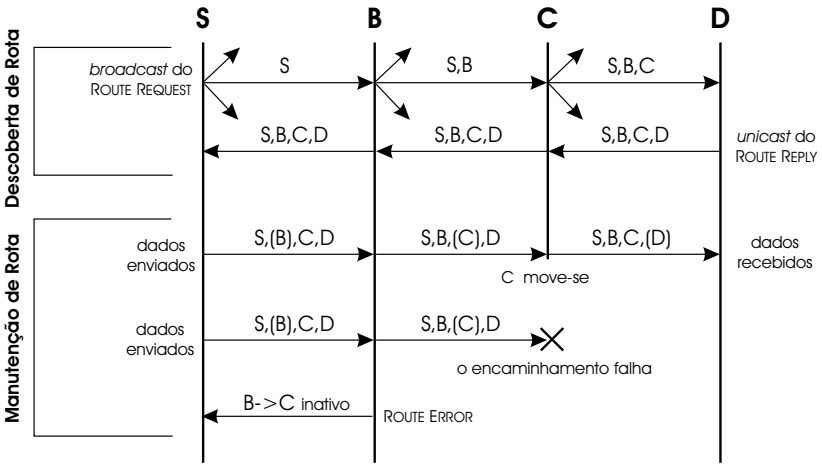


Fig. 2.2: Operação básica do protocolo DSR mostrando na seqüência: a construção da rota na origem durante a propagação do ROUTE REQUEST, o retorno do ROUTE REPLY, o envio de dados usando a rota obtida e o envio de um ROUTE ERROR causado por uma falha na rede. O próximo *hop* é indicado pelo endereço entre parênteses.

Embora o DSR utilize rotas na origem, avanços recentes permitem uma extensão compatível ao protocolo, conhecido como *flow state* (estado de fluxo) [19, 26], que permite o roteamento da maioria dos pacotes sem uma rota na origem explícita em seu cabeçalho. Esta extensão reduz o *overhead* do protocolo e ainda preserva as propriedades fundamentais da operação do DSR.

2.2.2 Descoberta de Rota

Quando um nó **S** origina um novo pacote para um nó **D**, o nó origem inclui no cabeçalho do pacote a rota na origem informando a seqüência de *hops* que o pacote deve seguir. Geralmente, **S** obtém uma rota adequada (e previamente conhecida) em seu ***Route Cache*** (*cache* de rota). Caso nenhuma rota seja encontrada em seu *cache*, o nó inicia o processo de Descoberta de Rota para encontrar dinamicamente uma nova rota ao destino **D**. Neste caso, **S** é chamado de "iniciador" e **D** de "alvo" da Descoberta de Rota.

A Figura 2.3 ilustra um exemplo de Descoberta de Rota na qual o nó **A** tenta descobrir uma rota para o nó **E**. Para iniciar a Descoberta de Rota, **A** transmite uma mensagem de ROUTE REQUEST (pedido de rota) como um pacote de *broadcast* local, o qual é recebido por todos os nós dentro de seu raio de transmissão. Cada mensagem de ROUTE REQUEST contém os endereços do iniciador e do alvo da Descoberta de Rota além de um identificador único (*request id*), determinado pelo iniciador do pedido. Cada ROUTE REQUEST também contém uma lista dos endereços dos nós intermediários pelos quais esta cópia em particular da mensagem de ROUTE REQUEST foi encaminhada. Esta lista contendo a rota é inicializada como uma lista vazia pelo iniciador da Descoberta de Rota.

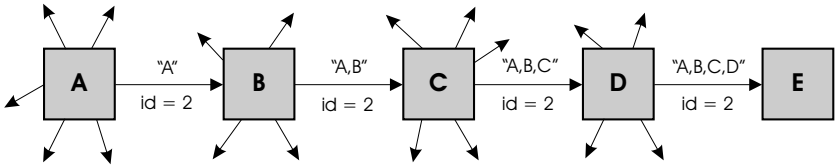


Fig. 2.3: Exemplo de Descoberta de Rota na qual o nó **A** é o iniciador e o nó **E** o alvo.

Quando um nó recebe um ROUTE REQUEST, três situações podem ocorrer. Primeiro, sendo ele o alvo da Descoberta de Rota, o nó retorna uma mensagem de ROUTE REPLY ao iniciador fornecendo uma cópia da rota acumulada do ROUTE REQUEST; ao receber este ROUTE REPLY, o iniciador armazena esta rota em seu *Route Cache* para envio de pacotes subseqüentes a este mesmo destino. Segundo, se o nó que recebeu este ROUTE REQUEST processou recentemente outra mensagem de ROUTE REQUEST deste iniciador com a mesma tripla <iniciador, alvo, identificador>, ou se seu próprio endereço estiver presente na lista de rota, ele descarta o pedido. Caso contrário, o nó anexa seu próprio endereço na lista de rota do ROUTE REQUEST e o propaga através de um *broadcast* local (com o mesmo identificador).

Ao retornar um ROUTE REPLY ao iniciador da Descoberta de Rota, tal como na Figura 2.3, o nó **E** verificará seu *Route Cache* para encontrar uma rota de volta para **A**. Se positivo, **E** usará esta rota para transmitir o pacote contendo o ROUTE REPLY. Caso contrário, **E** pode realizar sua própria Descoberta de Rota para o nó alvo **A**, mas para evitar uma recursão infinita de Descobertas de Rota, o nó deve fazer um *piggyback* deste ROUTE REPLY, que é o procedimento pelo qual o ROUTE REPLY é encapsulado dentro da mensagem de ROUTE REQUEST. É possível fazer um *piggyback* de outros pacotes pequenos também, como o pacote SYN do TCP [30] utilizando este mesmo procedimento.

O nó **E** pode reverter a seqüência de *hops* presente no ROUTE REQUEST e enviar o ROUTE REPLY com tal rota. Para protocolos MAC tais como o IEEE 802.11 que requer uma troca bidirecional de quadros como parte de sua operação [31], a rota obtida deve ser revertida para: 1) Testar se a rota descoberta é de fato

bidirecional antes de o iniciador da Descoberta de Rota começar a utilizá-la; 2) Evitar o *overhead* de uma segunda Descoberta de Rota.

Ao iniciar uma Descoberta de Rota, o nó emissor armazena uma cópia do pacote original (que disparou o processo de descoberta) em um *buffer* local denominado *Send Buffer*. O *Send Buffer* contém uma cópia de cada pacote que ainda não pôde ser transmitido pela inexistência de uma rota até o destino especificado. Cada pacote no *Send Buffer* é logicamente associado ao tempo em que foi inserido no *buffer*, sendo descartado após um certo período de tempo. Se necessário, para evitar o estouro do *Send Buffer*, uma fila FIFO ou outra estratégia de substituição pode ser utilizada para liberar os pacotes antes de sua expiração.

Enquanto um pacote residir no *Send Buffer*, o nó origem pode realizar ocasionalmente uma nova Descoberta de Rota para o mesmo endereço de destino do pacote. Entretanto, o nó deve limitar o número de tentativas de Descobertas de Rota, pois o nó destino pode não ser alcançável. Em particular, devido ao movimento dos nós e do alcance de transmissão limitado, a rede sem fio pode estar particionada, não existindo, portanto, nenhuma seqüência de *hops* até o destino.

Se uma nova Descoberta de Rota for realizada para cada pacote enviado por um nó em uma partição da rede, então um número grande de ROUTE REQUESTs improdutíveis serão propagados apenas em um subconjunto da rede ad hoc. Para reduzir este *overhead*, o nó pode utilizar um algoritmo de *backoff* exponencial, duplicando o intervalo de tempo entre duas Descobertas de Rota sucessivas iniciadas para o mesmo destino. Este mecanismo é similar ao utilizado na Internet a fim de limitar o número de pedidos ARP enviados para um único endereço IP destino [32].

2.2.3 Manutenção de Rota

Cada nó deve receber a confirmação de recepção correta pelo próximo *hop* ao longo da rota e, se necessário, o pacote é retransmitido (em um número máximo de vezes) até a confirmação de recebimento. Por exemplo, na situação ilustrada na Figura 2.4, o nó **A** originou um pacote para **E** utilizando uma rota na origem através de **B**, **C** e **D**. Neste caso, o nó **A** é responsável pelo recebimento do pacote em **B**, o nó **B** é responsável pelo recebimento em **C**, e assim sucessivamente. Esta confirmação de recebimento pode, em muitos casos, ser provida sem nenhum custo ao DSR como parte padrão do protocolo MAC em uso (como a transmissão de um pacote *acknowledgment* definido no protocolo IEEE 802.11 [31]) ou por *acknowledgment* passivo [33] (por exemplo, **B** confirma o recebimento em **C** detectando a transmissão do pacote de **C** para **D**). Se nenhum desses mecanismos estiver disponível, o nó transmissor pode ajustar um *bit* no cabeçalho do pacote para solicitar que um procedimento interno específico do DSR retorne a confirmação do próximo *hop*; este procedimento de confirmação normalmente será transmitido diretamente ao nó transmissor, mas, se o enlace entre os nós for unidirecional, o *acknowledgment* pode percorrer um caminho diferente.

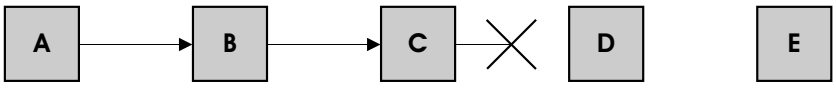


Fig. 2.4: Exemplo de Manutenção de Rota. Nela o nó **C** é incapaz de encaminhar o pacote de **A** para **E** através do seu próximo *hop* **D**.

Se o pacote for retransmitido por um nó em um número máximo de vezes e nenhuma confirmação for recebida, o nó gera uma mensagem de ROUTE ERROR ao iniciador do pacote, identificando o enlace pelo qual o pacote não pôde ser encaminhado. Por exemplo, na Figura 2.4, se **C** for incapaz de entregar ao próximo *hop* **D**, então **C** retorna um ROUTE ERROR para **A**, expressando que o enlace entre **C** e **D** está inativo. O nó **A** remove esse enlace de seu *cache*, e qualquer retransmissão do pacote original fica a cargo de protocolos da camada superior tais como o TCP. Para enviar uma retransmissão ou outros pacotes para o mesmo destino **E**, o nó **A** verifica seu *Route Cache* para obter outra rota; caso encontre, o nó envia o pacote através da nova rota. Caso contrário, **A** pode realizar uma nova Descoberta de Rota para o mesmo alvo.

2.2.4 Otimizações do Protocolo

Nesta seção são apresentadas algumas características adicionais do protocolo DSR para os procedimentos de Descoberta de Rota, Manutenção de Rota e *cache*.

A. Descoberta de Rota

ROUTE REQUESTs não Propagáveis. Quando for realizar uma Descoberta de Rota, o nó primeiramente envia um ROUTE REQUEST com o limite de propagação máxima (*hop limit*) de um *hop*, proibindo seus vizinhos de transmiti-lo. Este mecanismo provê uma maneira acessível para determinar se o alvo é vizinho do iniciador ou se seu vizinho possui uma rota em *cache* para o destino. Se nenhum ROUTE REPLY for recebido em um determinado período de tempo, então o nó envia um novo ROUTE REQUEST com o limite de propagação ajustado ao seu valor máximo permitido.

Respostas a ROUTE REQUESTs através do *Cache*. Se um nó receber um ROUTE REQUEST para um destino **D** para o qual já possua uma rota, o nó gera um ROUTE REPLY baseado na informação presente em seu *cache* ao invés de retransmitir o ROUTE REQUEST. Esta característica reduz a latência das mensagens de ROUTE REPLY e previne que ROUTE REQUESTs propaguem em toda a rede.

ROUTE REPLYs Gratuitos. Quando um nó, operando em modo de recepção promíscuo[1], detecta um pacote no qual ele não é o próximo *hop*, o nó em questão faz uma verificação na lista de endereços da porção ainda não processada da rota contida no ROUTE REQUEST a fim de encontrar seu próprio endereço. Se o endereço do nó estiver listado nessa parte da rota, então os nós intermediários listados entre o nó em questão e o nó que propagou o ROUTE REQUEST não são mais necessários. O nó transmite então um ROUTE REPLY "gratuito" ao iniciador inserindo a rota mais curta sem esses *hops* adicionais. Ao receber o ROUTE REPLY, o nó origem adiciona tal rota em seu *cache*. Como o *Route Cache* pode armazenar mais de uma rota para o mesmo destino, a rota mais curta não precisa necessariamente sobrescrever a mais longa. Caso a mais curta não funcione, o emissor pode utilizar a rota mais longa.

Prevenção de Tempestades de ROUTE REPLYs. A habilidade de os nós responderem a um ROUTE REQUEST baseados em sua informação de *Route Cache* pode resultar em possíveis "tempestades" (*storms*) de ROUTE REPLYs em alguns casos. Se um nó transmitir um ROUTE REQUEST aos seus vizinhos e cada um

[1] Modo de operação da interface de rede em que o *hardware* entrega cada pacote recebido ao *software* de rede sem nenhuma filtragem baseada no endereço de destino a nível de camada de enlace.

deles possuir uma rota em seu *Route Cache*, então cada vizinho enviará um ROUTE REPLY. Tais respostas simultâneas podem causar congestionamento local e aumentar o número de colisões na rede.

Para reduzir estes efeitos, se um nó colocar sua interface de rede em modo promíscuo, ele poderá adiar seu ROUTE REPLY por um curto período de tempo de acordo com o comprimento da rota contida na resposta. Caso o nó, dentro desse período de espera, detecte que o iniciador da Descoberta de Rota esteja utilizando uma rota de menor comprimento do que a sua prevista, ele pode inferir que o iniciador já recebeu um ROUTE REPLY contendo uma rota mais curta, e cancelar sua mensagem de ROUTE REPLY "atrasada".

B. Manutenção de Rota

Salvaging. Quando um nó intermediário, ao encaminhar um pacote, descobre através da Manutenção de Rota que o próximo *hop* da rota está inalcançável, ele examina seu *Route Cache* para encontrar uma outra rota para o mesmo destino. Se a rota existir, o nó recupera o pacote (caracterizando o *salvaging*), substituindo a rota original no cabeçalho do pacote pela rota obtida e o retransmite. Se a rota não existir em seu *cache*, o nó descarta o pacote (e não envia um ROUTE REQUEST) e retorna um ROUTE ERROR à origem do pacote.

O nó intermediário não realiza uma Descoberta de Rota para recuperar o enlace danificado já que é bem provável que o nó origem possua uma rota alternativa, evitando, portanto, um *overhead* desnecessário. Um contador é mantido no cabeçalho do pacote indicando o número de vezes que o mesmo foi recuperado.

ROUTE ERRORs Gratuitos. Quando um nó origem **S** recebe um ROUTE ERROR de um pacote que originou, **S** propaga esse ROUTE ERROR a todos seus vizinhos via *piggyback* no próximo ROUTE REQUEST que enviar. Desta maneira, informações desatualizadas nos *caches* dos nós vizinhos de **S** não gerarão ROUTE REPLYs contendo o enlace inválido.

C. Cache

Snooping. Quando um nó encaminha um pacote, ele verifica a porção ainda não processada da rota na origem. Caso o nó esteja listado, ele adiciona tal rota em seu *Route Cache* a partir de si mesmo até o destino especificado.

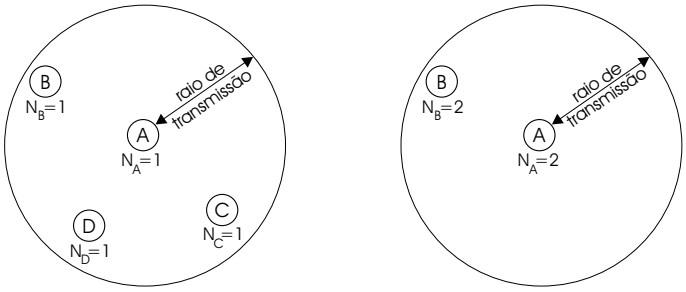
Tapping. Os nós operam suas interfaces de rede em modo promíscuo, desabilitando a filtragem de endereços da interface e possibilitando o protocolo de rede a receber todos os pacotes que a interface detectar. Estes pacotes são examinados para a obtenção de rotas úteis ou mensagens de ROUTE ERROR e posteriormente são descartados. Esta otimização permite que um nó adicione informações úteis ao *Route Cache*, sem nenhum uso adicional da largura de banda da rede.

2.3 Protocolos baseados em Controle de Congestionamento

Nesta seção são brevemente discutidos alguns esquemas de roteamento baseados em controle de congestionamento propostos na literatura. Tais esquemas utilizam a informação de carga na rede como métrica de seleção do roteamento em MANETs.

Em [34] Lee e Gerla propuseram o protocolo *Dynamic Load-Aware Routing* (DLAR). A métrica de seleção de rota é o número de pacotes enfileirados em cada nó intermediário que o ROUTE REQUEST percorreu. Três variações foram feitas a partir do número obtido de pacotes: a soma, a média e o uso de limitante inferior. Em [35], Hassanein e Zhou propuseram o protocolo *Load-Balanced Ad Hoc Routing* (LBAR). Nele, a métrica de seleção ou carga na rede é definido como o número total de rotas atravessando o nó e seus vizinhos. Nesses dois protocolos, o nó destino determina a melhor rota.

Em [36], foi apresentado o protocolo *Load-Sensitive Routing* (LSR). A métrica é dada pela soma dos pacotes presentes na interface de rede do nó móvel e seus vizinhos. Ao contrário dos dois primeiros, no LSR o nó origem seleciona a rota. Embora a técnica empregada pelo LSR seja mais precisa do que a discutida pelo DLAR e LBAR, o efeito do acesso das contenções na camada MAC (por exemplo, o IEEE 802.11) não é considerado. Portanto, o LSR pode estimar o mesmo atraso de acesso de contenção para diferentes cenários em que essa métrica for utilizada. Considere a Figura 2.5 em que o nó **A** possui números diferentes de vizinhos. O número de pacotes enfileirados na interface do móvel k é dado por N_k. De acordo com método implementado pelo LSR, o tráfego na rede do nó **A** nas Figuras 2.5(a) e (b) é igual a 4. Entretanto, o período de contenção do nó **A** na Figura 2.5(a) é potencialmente maior do que na Figura 2.5(b). Em consideração a esse atraso de contenção, Sheu e Chen sugeriram o protocolo *Delay-Oriented Shortest Path Routing* (DOSPR) [37] e analisaram o atraso de acesso ao meio de um nó móvel em uma rede sem fio utilizando o IEEE 802.11.



(a) Nó A com três vizinhos (b) Nó A com um vizinho

Fig. 2.5: Atrasos diferentes de contenção para a mesma métrica de tráfego.

Em [38], foi proposto um algoritmo de controle de congestionamento baseado em dois parâmetros: a) *limite de fila*, que é tamanho máximo do *buffer* da interface de rede do nó; b) *limite de encaminhamento*, como sendo o número máximo de *bytes* enviados pelo nó. Caso o número médio de pacotes enfileirados no *buffer* exceda o limite, então um pacote de notificação de congestionamento é enviado a todos seus vizinhos. Todos os nós também monitoram a quantidade de dados enviados ao seu vizinho. Se esse volume médio de dados ultrapassar um certo nível, o nó armazena o endereço do seu vizinho em uma tabela. Caso um nó **A**, que estiver encaminhando ao seu vizinho **B** um pacote com destino ao nó **C**, receber uma notificação de congestionamento e verificar que seu limite de encaminhamento para **B** foi excedido, então **A** tenta obter uma outra rota para **C**.

Em [39], foi apresentado uma variante do protocolo AODV (*Ad Hoc On-Demand Distance Vector*), denominado *Load-Aware On-Demand Routing* (LAOR). Nele, o caminho entre um par origem-destino é obtido a partir do atraso total estimado da rota e do número de *hops*.

Em [40], é apresentado um novo algoritmo de roteamento distribuído para realizar controle de congestionamento dinâmico em redes de acesso sem fio. O algoritmo constrói uma árvore com tráfego balanceado, que simplifica o roteamento e evita o encaminhamento de estados por destino e fluxo para reserva. O método consegue melhor utilização da rede diminuindo as taxas de bloqueio de largura de banda do que outras técnicas anteriormente empregadas.

2.4 Roteamento Sob Demanda e de Congestionamento Mínimo

O congestionamento ocorre em uma rede ou uma porção da mesma quando a quantidade total de tráfego enviado excede a capacidade disponível. Seus principais efeitos são o atraso excessivo na entrega dos pacotes ou a perda dos mesmos pela falta de espaço nos *buffers*. Uma variedade de condições podem contribuir para o congestionamento: arquitetura da rede sem fio, especificações dos dispositivos (como tamanho do *buffer*, taxa de processamento etc.), tamanho dos pacotes e o protocolo de transporte em uso.

A maioria dos protocolos de roteamento para redes ad hoc utiliza o menor número de *hops* para encaminhar um pacote. Entretanto, esse procedimento pode sobrecarregar alguns nós, principalmente os nós centrais, levando a um aumento do número de colisões em certas áreas da rede, e conseqüentemente degradando o desempenho da mesma. Logo, um caminho de mínimo *hop* às vezes pode causar atrasos maiores dos pacotes do que outros caminhos alternativos, já que alguns nós da rede podem estar com muito tráfego. Tais nós podem também sofrer com o alto consumo da potência das baterias. Os protocolos baseados no número mínimo de *hops* não distribuem a carga de tráfego de maneira balanceada na rede, podendo levar a uma alta taxa de perda de pacotes além do esgotamento rápido da potência das baterias dos nós móveis.

A Figura 2.6 mostra um exemplo de uma rede que utiliza o caminho mínimo como seleção de rota. Pelo nó em destaque, passa um total de 9 rotas diferentes (combinação entre os três pares origem-destino), havendo, portanto, um grande número de colisões, caso vários pacotes de diferentes rotas cheguem aproximadamente no mesmo instante. Rotas curtas geralmente compreendem enlaces com raios de transmissão longos. Com o aumento do alcance de transmissão, a qualidade do canal sem fio via rádio diminui, levando a mais perdas e retransmissões dos pacotes [41]. Enlaces de longo alcance inibem os outros nós de se comunicarem caso eles estejam dentro do alcance de interferência do par transmissor/receptor.

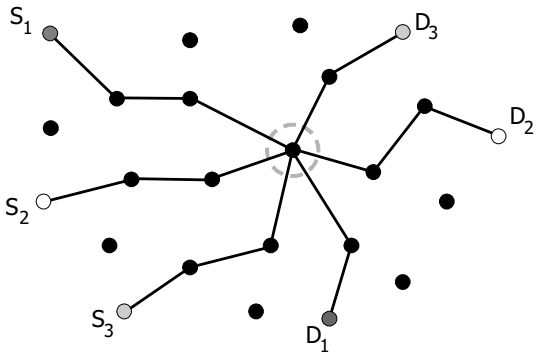


Fig. 2.6: Exemplo de uma rede com um nó central congestionado.

A partir destas considerações, um dos desafios da tese foi adaptar o DSR a fim de tratar o congestionamento. Nesta seção é discutido o protocolo de roteamento proposto para redes ad hoc sem fio, denominado MCOR (*Minimal Congestion On-Demand Routing*). O protocolo MCOR é uma extensão do DSR, que foi discutido na Seção 2.2. As principais diferenças entre o MCOR e o DSR básico são:

1) O MCOR utiliza como métrica de seleção a menor soma do número de pacotes enfileirados no *buffer* de cada nó intermediário percorrido pelo ROUTE REQUEST;

2) Cada nó propaga um ROUTE REQUEST ou realiza um *salvaging* somente se o meio não estiver congestionado;

3) O MCOR não permite que qualquer nó intermediário retorne um ROUTE REPLY ao iniciador, a fim de evitar rotas com informação desatualizada.

2.4.1 Descoberta e Manutenção de Rota

O MCOR, tal como o DSR, constrói rotas sob demanda. Quando um nó origem não possui uma rota a um nó destino, o processo de descoberta é iniciado. O nó origem envia um pacote de ROUTE REQUEST em *broadcast*. O ROUTE REQUEST possui os endereços de origem e destino, um identificador único e a informação de carga da rota, dada pelo número total de pacotes da fila da interface de transmissão de cada nó intermediário pelo qual a mensagem atravessou. Cada nó intermediário, ao receber um ROUTE REQUEST, além de adicionar seu próprio endereço na lista de rota (previsto no DSR básico), atualiza a carga da rota, adicionando o tamanho da fila de sua interface de transmissão. Assume-se que os enlaces são bidirecionais e que cada nó móvel utiliza o protocolo IEEE 802.11 DCF (*Distributed Coordination Function*) e que o mesmo seja capaz de monitorar a fila da interface da camada MAC. Uma descrição do protocolo IEEE 802.11 é apresentada no Apêndice A.

Quando o pacote de ROUTE REQUEST atinge o destino, um pacote de ROUTE REPLY é transmitido pelo caminho inverso do ROUTE REQUEST. O MCOR permite que o alvo da Descoberta de Rota responda a ROUTE REQUESTs duplicados, de modo a prover tolerância a enlaces inválidos e mais escolhas de rotas ao nó origem. Ao contrário do DLAR (*Dynamic Load-Aware Routing*), o nó destino não aguarda um certo período de tempo para responder os ROUTE REQUESTs, de modo que o nó origem possa obter rapidamente a rota e enviar os pacotes presentes no *Send Buffer*. Ao contrário também do DLAR, o nó origem é responsável pela escolha da rota. O MCOR proíbe qualquer nó intermediário gerar um ROUTE REPLY ao iniciador, mesmo quando este possui uma rota ao destino, com o intuito de se obter rotas com a informação mais atualizada possível.

O ROUTE REPLY possui também a carga (soma do número de pacotes nos *buffers*) do ROUTE REQUEST, que indica indiretamente se o meio está ou não congestionado. O MCOR adiciona o número de pacotes enfileirados em cada nó intermediário e seleciona a rota com a menor soma. Se houver empate, a rota com menor número de *hops* é selecionada. Quando houver múltiplas rotas com a menor soma e o menor número de *hops*, a rota que foi armazenada primeiro é utilizada. Assim sendo, considere a Figura 2.7, na qual o número de pacotes existentes no *buffer* da estação k é dado por Q_k. A rota i possui soma de 30 (i.e., 10 + 2 + 4 + 14), a rota j de

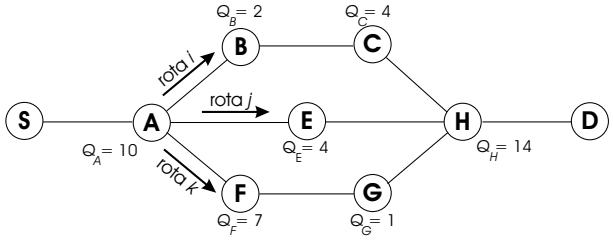


Fig. 2.7: Aplicação do algoritmo de roteamento do MCOR em uma rede ad hoc.

28 (i.e., 10 + 4 + 14) e a rota k de 32 (i.e., 10 + 7 + 1 + 14). Portanto, a rota j é utilizada para transmitir os pacotes do nó origem **S** ao nó destino **D**.

Quando o iniciador recebe o ROUTE REPLY, ele envia o pacote pela rota obtida. Caso ele receba outro ROUTE REPLY contendo uma rota com uma carga menor, os pacotes são enviados pela nova rota. Todas as rotas, mesmo não utilizadas, continuam sendo armazenadas como acontece no DSR. Quando um caminho estabelecido torna-se inativo, um pacote de ROUTE ERROR é enviado à origem.

2.4.2 Modificações na Descoberta de Rota e Salvaging

Com o objetivo de melhorar o desempenho e diminuir o *overhead* do MCOR, duas outras adaptações foram feitas nos processos de Descoberta de Rota e *salvaging* do DSR básico.

No processo de Descoberta de Rota do MCOR, um nó envia um ROUTE REQUEST que se propaga em toda a rede a fim de encontrar uma rota até o alvo. Contudo, essa inundação de mensagens de ROUTE REQUEST durante a Descoberta de Rota pode ter dois efeitos negativos: 1) Em áreas relativamente congestionadas, a retransmissão dos pacotes de ROUTE REQUEST pode sobrecarregar ainda mais o meio sem fio; 2) A resposta a ROUTE REQUESTs que atravessaram áreas congestionadas só pode resultar em rotas descobertas na mesma área; logo o envio dos pacotes através dessas rotas pode levar a uma degradação do desempenho da rede. Foi aplicado então o seguinte procedimento: caso um nó intermediário detecte que seu meio sem fio esteja congestionado, ele não processa ou não retransmite o ROUTE REQUEST, descartando-o. Todavia, esta mudança possui duas desvantagens: 1) O nó iniciador pode não ser capaz de descobrir uma rota ao destino mesmo que ela exista, caso ela esteja em áreas com um alto índice de tráfego; 2) As rotas descobertas podem ser mais longas que o número mínimo de *hops* existente entre um par origem-destino, já que os ROUTE REQUESTs evitarão áreas congestionadas; isto pode ocasionar um leve aumento do *overhead* do protocolo em outras áreas da rede.

Como discutido na Seção 2.2.4, o *salvaging* é o mecanismo pelo qual um nó intermediário recupera um pacote quando o enlace relativo ao próximo *hop* torna-se inválido. O nó intermediário então checa seu *Route Cache* para encontrar uma rota até o destino, aumentando a probabilidade de sucesso de entrega do pacote. Entretanto, a rota encontrada pode ser inválida, já que as estações podem ter se movido depois de seu armazenamento. Existe também o problema de este novo caminho não possuir a informação atualizada da carga da rota. Portanto, explorou-se também o efeito de um nó intermediário não recuperar o pacote, e descartá-lo, caso o meio sem fio esteja congestionado. Isso porque, em uma área congestionada, o processo de *salvaging*

aumentará o *overhead* de roteamento. Três problemas em potencial são mencionados para esta melhoria: 1) Este procedimento pode impedir que alguns pacotes sejam entregues em áreas congestionadas; 2) Alguns nós podem não detectar rotas através da sondagem dos pacotes (*snooping*), diminuindo, portanto, a qualidade de seu *Route Cache*; 3) Quando um nó tenta transmitir um pacote ao seu próximo *hop* através de um enlace inativo, um pacote RTS é repetido várias vezes (quando utilizando um protocolo MAC como o IEEE 802.11) para tentar completar a transmissão com sucesso. Entretanto, isso pode ocasionar um aumento da latência dos pacotes dos nós vizinhos (realizam o *virtual channel sensing*) além de aumentar o número de colisões na área em questão.

2.5 Resumo do Capítulo

Neste capítulo foi mostrado um resumo dos principais protocolos ad hoc existentes. Foi apresentada também uma visão geral da operação do protocolo DSR. A versão básica do DSR utiliza roteamento na origem explícito em que cada pacote transporta em seu cabeçalho uma lista completa e ordenada dos nós pelos quais o pacote percorrerá. Isto permite ao emissor selecionar e controlar as rotas utilizadas em seus próprios pacotes, o suporte a múltiplas rotas para qualquer destino (por exemplo, para balanceamento de carga) e garante que as rotas usadas não possuam *loops*. Além disso, outros nós que estejam encaminhando ou detectando qualquer um desses pacotes podem facilmente armazenar esta informação de roteamento para uso futuro.

No DSR, os processos de Descoberta de Rota e Manutenção de Rota operam inteiramente sob demanda. Ao contrário de outros protocolos, o DSR não requer nenhum tipo de pacote periódico de qualquer camada da rede, como por exemplo, anúncios de roteamento, estado do enlace ou detecção de pacotes vizinhos. Diante desse comportamento do protocolo, o número de pacotes de *overhead* será praticamente nulo quando todos os nós na rede sem fio estiverem aproximadamente estacionários com relação uns aos outros, em que todas as rotas necessárias para comunicação já foram descobertas.

A contribuição do capítulo consistiu na apresentação do protocolo de roteamento MCOR. Tal protocolo é uma variante do DSR, que faz o controle de congestionamento na rede ad hoc sem fio. Ele possui um comportamento inteiramente sob demanda, sendo que sua métrica para escolha das rotas é a menor soma dos pacotes enfileirados no *buffer* de cada nó intermediário.

Capítulo 3

Modelo Analítico

The Devil said to Daniel Webster: "Set me a task I can't carry out, and I'll give you anything in the world you ask for."

Daniel Webster: "Fair enough. Prove that for n greater than 2, the equation $a^n + b^n = c^n$ has no non-trivial solution in the integers."

They agreed on a three-day period for the labor, and the Devil disappeared.

At the end of three days, the Devil presented himself, haggard, jumpy, biting his lip. Daniel Webster said to him, "Well, how did you do at my task? Did you prove the theorem?"

"Eh? No... no, I haven't proved it."

"Then I can have whatever I ask for? Money? The Presidency?"

"What? Oh, that—of course. But listen! If we could just prove the following two lemmas..."

— *The Mathematical Magpie,* Clifton Fadiman

3.1 Considerações Iniciais

A ocorrência de colisões pode degradar o desempenho das redes ad hoc *multi-hop*. Para evitar colisões, muitos protocolos foram propostos na literatura tais como MACAW (*Multiple Access with Collision Avoidance for Wireless*) [42] e CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) [31, 43]. No protocolo CSMA/CA com RTS/CTS (*Request to Send*/*Clear to Send*) a transmissão de pacotes e seu *acknowledgment* são precedidos por pacotes RTS e CTS entre um par de estações que deseja se comunicar. Os demais nós que detectarem pacotes RTS ou CTS adiarão seu acesso ao canal para evitar colisões.

Vários artigos buscam diminuir o congestionamento na rede propondo alterações na forma com que o protocolo DSR define o caminho entre as estações, resultando em caminhos alternativos não mínimos que aumentam a quantidade de tráfego em circulação na rede [34, 36, 44]. A avaliação dessas propostas é em geral feita por simulação de eventos discretos.

Na proposta desta tese, o pacote de resposta contém, além do identificador das estações intermediárias,

um quantificador do estado de congestionamento do nó (tempo de espera médio ou número médio de pacotes armazenados) que permitem selecionar, entre duas rotas de mesmo número de enlaces, aquela de menor congestionamento. O encaminhamento é resultado da solução de um problema de programação matemática de dois níveis, sendo o inferior o problema clássico de otimização de caminho mínimo e o superior o de avaliação global da colisão na rede.

Os algoritmos de encaminhamento são processos concorrentes, executados de maneira distribuída em cada uma das estações da rede. A análise de desempenho dos algoritmos de encaminhamento é tipicamente feita através de simuladores de eventos discretos, dentre os quais destaca-se o ns-2 [45]. O uso desses simuladores exige muito tempo e a execução exaustiva de vários casos para análise comparativa dos algoritmos. Alternativamente, neste capítulo, os protocolos de encaminhamento são avaliados através de um modelo analítico estático representando um limitante superior para o desempenho da rede. O modelo analítico é obtido considerando que os nós da rede são estáticos e que o tráfego entre as estações é estatisticamente estacionário. Essas e outras hipóteses simplificadoras foram necessárias para a obtenção de um modelo analítico do desempenho da rede medido através do número médio de colisões. A mobilidade das estações pode degradar o desempenho da rede devido à eventual perda de enlaces e do encaminhamento por rotas não mínimas. O objetivo principal desse capítulo é apresentar um método relativamente simples e preliminar de avaliação dos algoritmos de encaminhamento. Estudos de casos (bastante trabalhosos) através de simulações de eventos discretos deverão ser realizados para as estratégias de encaminhamento que se mostrarem promissoras.

3.2 Modelo de Colisão

Considere que uma rede ad hoc sem fio pode ser representada por um grafo descrito pelos conjuntos $\mathcal{V}$ e $\mathcal{E}$. O conjunto $\mathcal{V}$ contém as estações móveis (supostas fixas nesse modelo), e o conjunto $\mathcal{E}$ contém os enlaces (arestas) da rede. A aresta (i, j) pertence ao conjunto $\mathcal{E}$ se o sinal eletromagnético irradiado pela estação i puder ser detectado pela estação j, ou seja, o nível de energia do pacote recebido é superior ao limiar de detecção da estação j. Devido aos mecanismos de controle de potência de cada estação e das condições assimétricas de propagação entre duas estações o grafo da rede não é necessariamente bidirecional, ou seja, a existência do enlace (i, j) não implica na existência do enlace (j, i). Assim, o conjunto $\mathcal{V}_j$ representa todas as estações cujos sinais podem ser detectados pela estação j enquanto que o conjunto $\mathcal{V}^i$ é composto por todas estações que podem detectar o sinal irradiado pela estação i. O conjunto $\mathcal{V}_j \backslash \{i\}$ é definido como $\mathcal{V}_j \backslash \{i\} = \mathcal{V}_j - \{i\}$.

Para avaliação do desempenho do algoritmo de encaminhamento proposto, consideram-se aplicações de tráfego em tempo real com vazão média constante entre os nós da rede. Os pacotes gerados pelas aplicações são enviados à rede e não são perdidos, isto é, os *buffers* das estações são supostos de tamanho infinito e as colisões devido ao mecanismo de múltiplo acesso dão origem a retransmissões. Além disso, o nível de energia dos pacotes e os códigos corretores de erro são tais que a probabilidade de perda de pacote por erro é considerada desprezível. Portanto, a vazão medida nos enlaces é composta pelo tráfego originado nas estações e pelas retransmissões devido às colisões.

A matriz de tráfego ρ, normalizada pela capacidade nominal de transmissão das estações, descreve a vazão de demanda entre as estações. A vazão x_{ij} é a demanda de tráfego encaminhado, ou seja, a demanda de tráfego

no enlace (i, j) resultante do algoritmo de encaminhamento aplicado à matriz ρ. O tráfego no enlace (i, j) é composto das parcelas x_{ijs} que representam a quantidade de tráfego encaminhada pelo enlace (i, j) que tem como origem o nó s e é dada pela equação

$$x_{ij} = \sum_{s} x_{ijs} \quad \forall (i,j) \in \mathcal{E} \tag{3.1}$$

Como não há perda de pacotes na rede, em cada nó intermediário a vazão do tráfego é conservada, conforme descrito na equação

$$\sum_{v \in \mathcal{V}^k} x_{kvs} - \sum_{u \in \mathcal{V}_k} x_{uks} = \begin{cases} \sum\limits_{t \in \mathcal{V} \setminus \{s\}} \rho_{st}, & k = s \\ -\rho_{sk}, & k \in \mathcal{V} \setminus \{s\} \end{cases} \tag{3.2}$$

O lado direito da equação representa ou o tráfego que parte do nó-origem $k = s$, ou o tráfego que termina em um nó-destino ($k \neq s$). No modelo, considere que todo nó da rede é destino de tráfego.

Se não houvesse retransmissão dos pacotes que sofrem colisão (ou perda por erro) o tráfego oferecido ao enlace (i, j) seria dado pela vazão x_{ij} e o tráfego escoado seria diminuído da parcela perdida. Nas redes sem fio, devido à variação das condições de propagação do meio de transmissão, essa estratégia resultaria em um desempenho inaceitável para as conexões. Portanto, os protocolos de acesso ao meio prevêem a retransmissão dos pacotes, aumentando a probabilidade de que um pacote seja recebido com sucesso e, conseqüentemente, aumentando o tráfego oferecido aos enlaces.

A vazão y do tráfego oferecido (tráfego escoado acrescido das possíveis retransmissões) ao meio de transmissão, cuja vazão do tráfego escoado é x, é modelada [46] por

$$x = ye^{-\eta y} \tag{3.3}$$

com η igual a 2 no caso do protocolo *Aloha* e igual a 1 no *Slotted Aloha*.

Nesta tese, a Equação (3.3) foi adaptada para o protocolo CSMA/CA através de uma aproximação baseada em um modelo proposto por Bianchi [47] no qual o tráfego oferecido ao meio é $y = n\tau$, sendo τ a probabilidade de uma estação transmitir em um *slot* e n o número de estações. A probabilidade de um pacote ser transmitido com sucesso, isto é, de não haver colisão é dada por

$$\frac{x}{y} = \lim_{n \to \infty} \frac{n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n} = \frac{ye^{-y}}{1-e^{-y}} \cong e^{-\eta y} \tag{3.4}$$

Apesar de o objetivo de Bianchi no artigo [47] ser a obtenção da vazão de saturação (máxima vazão) da rede, a Equação (3.4) estabelece a relação entre o tráfego oferecido e o tráfego total no meio em função do tráfego oferecido e a equação tem validade para $y < 1$.

A vazão do tráfego escoado em função da vazão do tráfego oferecido (tráfego escoado mais retransmissões) é mostrada na Figura 3.1 para os protocolos de acesso aleatório: *Aloha*, *Slotted Aloha* e CSMA/CA. O valor do parâmetro η para o protocolo CSMA/CA é igual a 0.53. Esse valor foi obtido pelo ajuste da curva exponencial com a curva obtida do artigo do Bianchi [47].

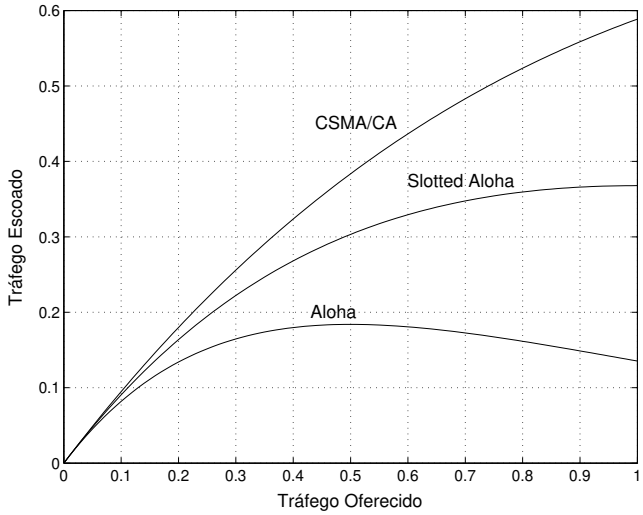


Fig. 3.1: Vazão do tráfego escoado em função do tráfego oferecido ao meio para os protocolos de acesso aleatório: *Aloha*, *Slotted Aloha* e CSMA/CA.

Em uma rede ad hoc, quando a estação i transmite um pacote para a estação j, existe a possibilidade de que haja colisão com pacotes transmitidos por outras estações cujos sinais podem alcançar a estação j. A probabilidade de colisão depende da intensidade de tráfego emitido por essas estações com tráfego total dado por

$$\gamma_j^i = \sum_{u \in \mathcal{V}_j \setminus \{i\}} \sum_{v \in \mathcal{V}^u} y_{uv} \tag{3.5}$$

A primeira somatória indica as estações cujo sinal pode atingir a estação j com exceção do sinal transmitido pela própria estação i. A segunda somatória indica que todo o tráfego emitido por uma estação que alcance j deve ser considerado, e não apenas os tráfegos cujo destino é j.

Portanto, a relação entre as vazões dos tráfegos escoado e oferecido a cada um dos enlaces da rede ad hoc é dada por

$$x_{ij} = y_{ij} e^{-\eta \gamma_j^i} \tag{3.6}$$

3.2.1 Programação Matemática

A determinação do encaminhamento que minimiza a probabilidade média de colisões na rede pode ser obtida pela resolução do problema de programação matemática que minimiza o tráfego oferecido (e, portanto, o número de retransmissões) descrito por

$$\min_{x_{ijs} \in \Omega} \sum_{(i,j) \in \mathcal{E}} y_{ij} \tag{3.7}$$

sendo Ω dado por

$$\Omega = \arg \min_{x_{ijs} \geq 0} \sum_{s \in \mathcal{V}} \sum_{(i,j) \in \mathcal{E}} x_{ijs} \tag{3.8}$$

com x_{ijs} satisfazendo a Equação (3.2).

As vazões y_{ij} são definidas pelo sistema de equações

$$y_{ij} \ \exp(-\eta \sum_{u \in \mathcal{V}_j \setminus \{i\}} \sum_{v \in \mathcal{V}^u} y_{uv}) = \sum_{s \in \mathcal{V}} x_{ijs}, \quad \forall (i,j) \in \mathcal{E} \tag{3.9}$$

O conjunto Ω contém todos os possíveis encaminhamentos de mínimo *hop* da rede em questão, ou seja, a Equação (3.8) é de fato um problema de caminho mínimo na rede. Note que apenas as rotas de caminho mínimo são consideradas na busca dos caminhos de mínima colisão. Essa restrição poderia ser aliviada, mas o uso de caminhos não mínimos aumentaria o tráfego total na rede.

A probabilidade média de colisão na rede é definida por

$$p = 1 - \frac{\sum_{s \in \mathcal{V}} \sum_{(i,j) \in \mathcal{E}} x_{ijs}}{\sum_{(i,j) \in \mathcal{E}} y_{ij}} \tag{3.10}$$

Dessa forma, o mínimo da função descrita pela Equação (3.7) corresponde à minimização da probabilidade média de colisão na rede, pois o numerador da fração da Equação (3.10) é constante no conjunto Ω.

3.3 Heurística de Solução

O problema de otimização proposto na Seção 3.1 é de difícil solução pelos métodos convencionais de programação matemática, pois o conjunto Ω não é necessariamente conexo e a função descrita na Equação (3.7) é não-linear, tendo em vista as restrições da Equação (3.9). Entretanto, vários métodos de otimização baseados em buscas aleatórias têm sido desenvolvidos na literatura e são apropriados à resolução desse problema de otimização. O GRASP (*Greedy Randomized Adaptive Procedure*) [48] é uma heurística na qual cada iteração consiste da construção de um candidato factível e de uma busca local.

O Algoritmo 1 é o procedimento geral da heurística de solução que busca o encaminhamento de mínima colisão por um número finito de iterações. Não é possível garantir a determinação da solução de mínima colisão, mas uma série de experimentos comprovou a eficácia do algoritmo para encontrar boas soluções. Em cada iteração do procedimento principal são repetidos os algoritmos: RandomDijkstra (Algoritmo 2) que determina aleatoriamente uma das possíveis matrizes de caminhos mínimos; TráfegoEncaminhado (Algoritmo 3) que calcula o tráfego escoado em cada um dos enlaces da rede; e Relaxação (Algoritmo 4) que obtém a probabilidade média de colisão na rede.

O algoritmo RandomDijkstra é uma variante do clássico algoritmo Dijkstra de caminhos mínimos [21], na qual a inserção de um novo nó na árvore de caminhos mínimos é obtida sorteando-se o índice dos nós de mesma distância.

Algoritmo 1 $[p_{min}] = \text{ColisãoMínima}(\mathcal{V}, \mathcal{E}, \rho, m)$

{ $\mathcal{V}$ é o conjunto de nós da rede; $\mathcal{E}$ é o conjunto de enlaces; ρ é a matriz de demanda; m é o número máximo de iterações.}

$n \leftarrow \#(\mathcal{V})$
for all $s \in \mathcal{V}$ **do**
  $\pi \leftarrow \text{RandomDijkstra}(\mathcal{V}, \mathcal{E}, s, W, \pi)$
end for
$\pi_{min} \leftarrow \pi$
$X \leftarrow \text{TráfegoEncaminhado}(\pi, \rho, n)$
$p_{min} \leftarrow \text{Relaxação}(\mathcal{V}, \mathcal{E}, X, \epsilon)$
for $c = 1$ to m **do**
  {A busca local altera uma linha da matriz π}
  for all $s \in \mathcal{V}$ **do**
    $\pi \leftarrow \text{RandomDijkstra}(\mathcal{V}, \mathcal{E}, s, W, \pi_{min})$
    $X \leftarrow \text{TráfegoEncaminhado}(\pi, \rho, n)$
    $p \leftarrow \text{Relaxacao}(\mathcal{V}, \mathcal{E}, X, \epsilon)$
    if $p < p_{min}$ **then**
      $p_{min} \leftarrow p$
      $\pi_{min} \leftarrow \pi$
    end if
  end for
  {A busca global altera a matriz π}
  for all $s \in \mathcal{V}$ **do**
    $\pi \leftarrow \text{RandomDijkstra}(\mathcal{V}, \mathcal{E}, s, W, \pi)$
  end for
end for

Algoritmo 2 $[\pi] = \text{RandomDijkstra}(\mathcal{V}, \mathcal{E}, s, W, \pi)$

{ π é a matriz de encaminhamento que é atualizada em sua linha s; W é a matriz de pesos dos enlaces que são supostos unitários.}

for all $v \in \mathcal{V}$ **do**
  $d_v \leftarrow +\infty$
  $\pi_{sv} \leftarrow -1$
end for
$d_s \leftarrow 0$
$\pi_{ss} \leftarrow s$
$\mathcal{Q} \leftarrow \mathcal{V}$
while $\mathcal{Q} \neq \{\}$ **do**
  $u \leftarrow \text{Random}\big(\arg\min_{v \in \mathcal{Q}} d_v\big)$ {Sorteia entre os nós de mínima distância}
  $\mathcal{Q} \leftarrow \mathcal{Q} \backslash \{u\}$
  for all $v \in \mathcal{V}^u \cap \mathcal{Q}$ **do**
    if $d_v > d_u + w_{uv}$ **then**
      $d_v \leftarrow d_u + w_{uv}$
      $\pi_{sv} \leftarrow u$
    end if
  end for
end while

O algoritmo TráfegoEncaminhado computa o tráfego em cada enlace percorrendo em sentido reverso a árvore de caminhos mínimos determinada pelo RandomDijkstra.

Algoritmo 3 $[X] = \text{TráfegoEncaminhado}(\pi, \rho, n)$

{ X é a matriz de tráfego encaminhado; π é a matriz de encaminhamento (predecessores); ρ é a matriz de demanda; n é o número de nós da rede.}

$X \leftarrow 0$
for $s = 1$ to n **do**
  for $j = 1$ to n **do**
    $v \leftarrow j$
    $u \leftarrow \pi_{sv}$
    while $v \neq s$ **do**
      $x_{uv} \leftarrow x_{uv} + \rho_{sj}$
      $v \leftarrow u$
      $u \leftarrow \pi_{sv}$
    end while
  end for
end for

O algoritmo Relaxação é o clássico algoritmo de iteração de ponto fixo [49]. Para os encaminhamentos nos quais a relaxação diverge, o tráfego oferecido é fixado no valor 1, indicando encaminhamentos não factíveis.

Algoritmo 4 $[p] = \text{Relaxação}(\mathcal{V}, \mathcal{E}, X, \epsilon)$

$Y \leftarrow X$
$Z \leftarrow 0$
while $\max_{(u,v)\in\mathcal{E}} |Y - Z| > \epsilon$ **do**
  $Z \leftarrow Y$
  for all $(i, j) \in \mathcal{E}$ **do**
    $\gamma_j^i = \sum_{u\in\mathcal{V}_j\setminus\{i\}} \sum_{v\in\mathcal{V}^u} z_{uv}$
    $y_{ij} \leftarrow x_{ij} e^{\eta\gamma_j^i}$
    if $y_{ij} > 1$ **then**
      $y_{ij} \leftarrow 1$
    end if
  end for
end while

$$p \leftarrow 1 - \frac{\sum_{s\in\mathcal{V}} \sum_{(i,j)\in\mathcal{E}} x_{ijs}}{\sum_{(i,j)\in\mathcal{E}} y_{ij}}$$

3.4 Exemplos Numéricos

Nesta seção são discutidos alguns estudos de casos, ilustrando a aplicação da heurística de solução através da análise exaustiva e simulação numérica.

3.4.1 Análise Exaustiva

Uma rede de seis nós, um nó central e cinco eqüidistantes, é mostrada na Figura 3.2. Só há demanda entre os nós não adjacentes, isto é, nós que não são interligados por enlaces diretos. Os arcos com setas indicam o encaminhamento de menor colisão média obtido pela análise exaustiva de todas as possibilidades. Observe que não há tráfego encaminhado pelo nó central, pois todo pacote enviado pelo nó central para um determinado nó da periferia poderia colidir com pacotes enviados por quaisquer dos demais nós da rede. Os pacotes são enviados do nó 1 para o nó 3 pelo nó 2, e para o nó 4 pelo nó 5. Esses pacotes não colidem entre si. Encaminhamentos similares ocorrem para os demais nós da rede.

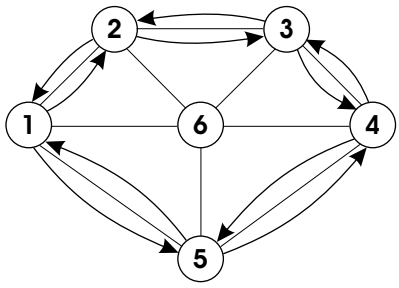


Fig. 3.2: Rede de seis nós na qual não há demanda entre nós adjacentes. Os arcos com setas indicam o encaminhamento de mínima colisão média.

Nem sempre a solução uniforme (encaminhamento similar para todos os nós da rede) é a de menor colisão média para as redes uniformes, como poderia ser inferido do exemplo da Figura 3.2. Uma rede de quatro nós, também sem demanda de tráfego adjacente, é mostrada na Figura 3.3, na qual o encaminhamento de mínima colisão foi obtido por análise exaustiva. Observe que o encaminhamento obtido concentra tráfego nos estações 2 e 3.

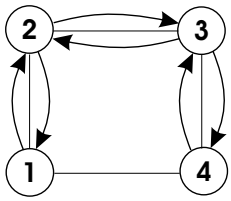


Fig. 3.3: Rede de quatro nós na qual não há demanda entre nós adjacentes. Os arcos com setas indicam o encaminhamento de mínima colisão média.

3.4.2 Simulação Numérica

Uma rede ad hoc de dez nós é mostrada na Figura 3.4, na qual não há tráfego entre nós adjacentes e a demanda de tráfego é uniforme entre nós não adjacentes.

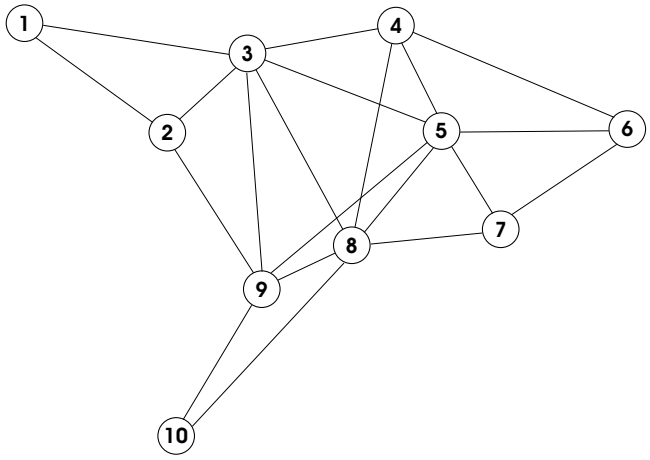


Fig. 3.4: Rede de dez nós na qual não há demanda entre nós adjacentes.

O total de tráfego oferecido à rede em função do número de iterações do algoritmo GRASP aplicado à rede da Figura 3.4 é mostrado na Figura 3.5. Os valores mostrados no gráfico entre dois asteriscos consecutivos são resultados das dez (número de nós da rede) buscas locais na qual apenas o tráfego originado de um dos nós é re-encaminhado. Note que todos os pontos da curva referem-se a encaminhamentos de caminho mínimo. O algoritmo proposto permite determinar entre os possíveis caminhos mínimos aquele que produz o menor tráfego oferecido na rede, e portanto, o de menor colisão média.

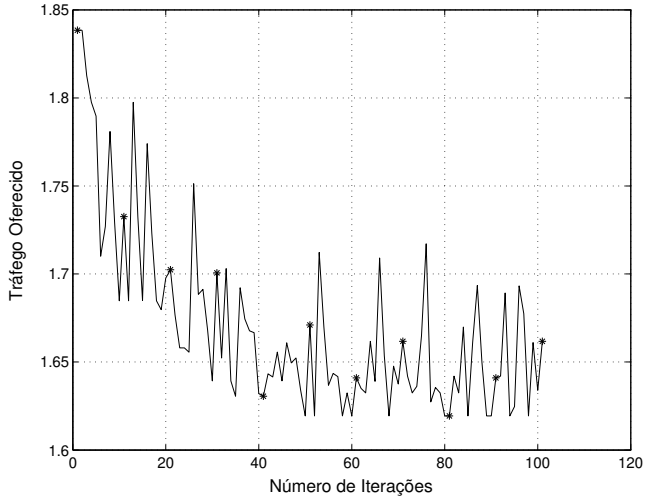


Fig. 3.5: Total do tráfego oferecido à rede da Figura 3.4 em função do número de iterações. Os valores entre dois asteriscos consecutivos são resultantes das dez buscas locais.

3.5 Resumo do Capítulo

Um modelo analítico para minimizar o número médio de colisões em uma rede ad hoc móvel foi proposto nesse capítulo. A probabilidade média de colisões foi obtida por um modelo similar ao clássico usado na avaliação do protocolo de múltiplo acesso *Aloha*. O encaminhamento é obtido pela resolução de um problema de programação matemática de dois níveis. O nível inferior é o problema clássico de caminho mínimo, e o nível superior é uma avaliação da probabilidade média de colisão. O problema global é complexo e sua solução foi obtida usando-se uma heurística baseada na técnica GRASP. Os resultados obtidos são encorajadores e a etapa seguinte é a simulação da estratégia de busca de caminhos de mínima colisão no ns-2 [45]. No ambiente ns-2 espera-se avaliar o efeito da mobilidade das estações principalmente no tempo de transferência dos pacotes, e não apenas no número de colisões.

Capítulo 4

Modelo de Simulação

Analysis and observation, theory and experience must never disdain or exclude each other; on the contrary, they support each other.

— *On War,* Carl Von Clausewitz

4.1 Avaliação de Protocolos Ad Hoc através do Simulador ns-2

A simulação possibilita uma aproximação dos experimentos através da realização de testes que podem ser executados e re-executados modificando uma variável da simulação e mantendo as outras constantes. É possível avaliar o comportamento das redes sem fio com um número de nós maior do que o existente no equipamento físico, ou de redes com características ainda não fabricadas.

A grande desvantagem da simulação é o risco da implementação de simplificações no modelo. Como não é possível reproduzir a realidade em um modelo de computador, alguns componentes da simulação devem ser estatísticos, aproximados ou mesmo ignorados. O modelo de simulação pode produzir resultados incorretos.

O ns-2 [45] é um simulador de eventos discretos desenvolvido pela Universidade da Califórnia na Berkeley e pelo projeto VINT [50], através do suporte do DARPA (*Defense Advanced Research Projects Agency*). No início do projeto, em meados de 1995, os estudos se concentravam no estudo do TCP e outros protocolos de redes com fio. O simulador não provia o suporte necessário para a simulação de redes sem fio *multi-hop* ou protocolos MAC.

No final de 1995, a Universidade Carnegie Mellon, através do Projeto Monarch, implementou algumas extensões ao ns-2 para redes sem fio, incluindo novos elementos nas camadas física, de enlace e de rede dentro do ambiente de simulação. A partir delas, é possível construir um ambiente detalhado de simulação para redes ad hoc. Mais detalhes sobre as extensões do ns-2, documentos de ajuda ou outras informações estão disponíveis no *web site* do Projeto Monarch [51]. A versão mais atual do simulador ns-2 é a 2.29, lançada em 22 de outubro de 2005.

4.1.1 Modelo de Propagação

A atenuação das ondas de rádio entre as antenas no terreno foi modelada considerando-se a atenuação do sinal em $1/r^2$ para pequenas distâncias e $1/r^4$ para as demais distâncias. O ponto de transição é chamado de ***distância de referência***, tipicamente de 100 metros para ambientes *outdoor* utilizando antenas de baixo ganho de 1.5 m e operando em uma faixa de 1-2 GHz [52]. Portanto, o modelo de propagação do sinal do simulador combina o modelo de propagação do espaço livre (*free space model*) e o modelo do terreno plano (*two-ray ground reflection model*). Quando o transmissor estiver dentro da distância de referência do receptor, o modelo do espaço livre é utilizado, com a atenuação do sinal de $1/r^2$. Além dessa distância, o simulador utiliza o modelo do terreno plano no qual o sinal decresce de $1/r^4$.

4.1.2 Modelo do Móvel

O modelo para cada nó móvel possui uma posição e uma velocidade movendo-se em uma topografia do tipo *flat*. A posição de cada nó é calculada em função do tempo, sendo utilizada pelo modelo de propagação de rádio para calcular o atraso de propagação entre duas estações e o nível de potência do sinal recebido em cada nó móvel.

Cada móvel possui uma ou mais interfaces de rede sem fio, conectadas a um modelo de canal físico único. Quando a interface de rede transmite um pacote, este é passado ao objeto do canal físico apropriado. Esse objeto computa então o atraso de propagação a partir do emissor para cada uma das interfaces do canal, escalonando o evento de recepção do pacote. Esse evento notifica o modelo da interface de recepção indicando que o primeiro *bit* de um pacote chegou. Neste instante, o nível de potência de cada pacote recebido é comparado com dois valores diferentes: o alcance de interferência (*carrier sense threshold*) e o alcance (ou raio) de transmissão (*received threshold*)[1]. Se o nível de potência estiver abaixo do alcance de interferência, o pacote é descartado como ruído. Se o nível de potência estiver acima do alcance de interferência, mas abaixo do alcance de transmissão, o pacote é marcado com erro antes de ser passado à camada MAC. Caso contrário, o pacote é simplesmente entregue à camada MAC.

Uma vez o pacote recebido no modelo da camada MAC, é realizado um teste para verificar se o estado de recepção está desocupado. Caso o receptor não esteja, duas situações podem ocorrer. Se o nível de potência do pacote já recebido for 10 dB[2] maior ou igual do que o nível de potência recebida do novo pacote, este é descartado e a interface de recepção continua normalmente sua operação de recepção corrente. Caso contrário, uma colisão ocorre e os dois pacotes são descartados.

Se a camada MAC estiver desocupada durante a chegada do pacote na interface de rede, ela computa o tempo de transmissão do pacote, escalonando um evento de recepção do pacote. Quando esse evento ocorre, a camada MAC verifica se o pacote não contém erros, filtra o endereço de destino e o transfere para a camada superior da pilha de protocolos.

[1] No ns-2 seus valores padrões são ajustados para uma potência correspondente a 550 m e 250 m, respectivamente.
[2] Este valor corresponde ao limiar de captura (*capture threshold*) no ns-2.

4.1.3 Controle de Acesso ao Meio

A camada de enlace do ns-2 inclui um modelo completo do padrão IEEE 802.11 [31] do protocolo MAC (*Medium Access Control*) sob o modo de operação DCF (*Distributed Coordination Function*) para implementar a contenção das estações no meio sem fio. O DCF é similar ao MACA [53] e MACAW [42], sendo designado para utilizar os mecanismos de *physical channel sense* e *virtual channel sense* com o objetivo de reduzir a probabilidade de colisões devido ao problema dos terminais escondidos. O protocolo IEEE 802.11 é descrito no Apêndice A.

4.1.4 Resolução de Endereço

Visto que os protocolos de roteamento operam na camada de rede usando endereçamento IP, o modelo de simulação do ARP [54] está presente na simulação para converter endereços IP para endereços MAC. A natureza *broadcast* dos pacotes de ARP REQUEST e sua interação com protocolos sob demanda lhe fazem um importante detalhe da simulação.

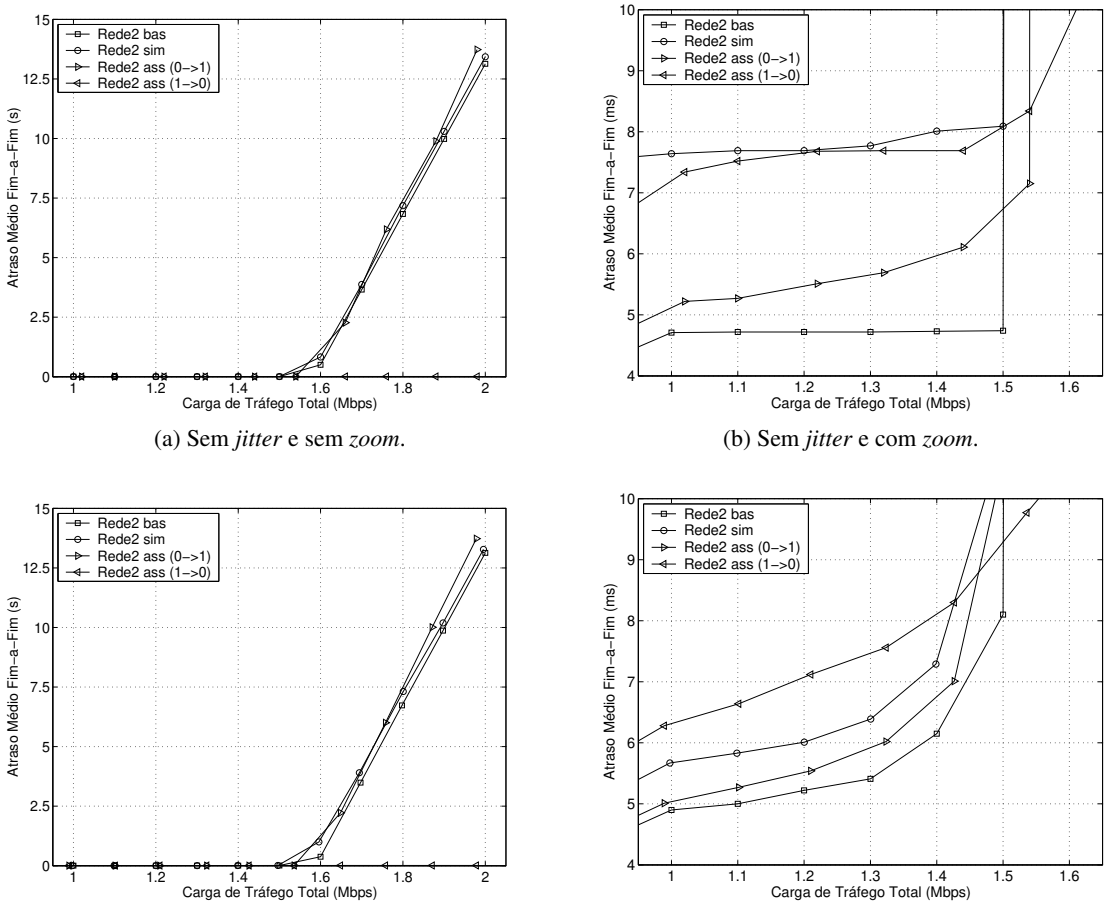
4.2 Simulações Elementares

Para garantir a implementação dos protocolos, em particular do DSR, e a corretude das medidas de desempenho, foram realizados alguns testes em redes pequenas para se compreender o comportamento do sistema. Foram utilizados um raio de transmissão de 250 m e um alcance de interferência de 550 m em uma rede sem fio de 2 Mbps. Utilizou-se o protocolo IEEE 802.11 DCF com o método básico, isto é, não foi utilizado o envio do RTS/CTS. O *buffer* de cada estação possui tamanho infinito. O tamanho dos pacotes é 1000 *bytes* e a taxa de envio é variável. As fontes foram iniciadas em 0 segundos. Para cada caso, foram utilizados dois tempos de simulação, 100 e 200 segundos, ambos com um tráfego constante CBR de 100 kpbs. Para o primeiro tempo de simulação, o objetivo era obter a intensidade máxima de tráfego para a qual a rede entregava 100% dos pacotes, pois no final da simulação alguns pacotes eram perdidos. Já no segundo, foi avaliado o atraso médio fim-a-fim dos pacotes em função deste tráfego máximo, agora com nenhum pacote perdido. Foi avaliado também a inclusão de *jitter*, ou seja, uma variação aleatória do instante de transmissão dos pacotes.

No primeiro teste, dois nós, 0 e 1, enviam tráfego um ao outro. Os nós estão a 200 m. Nesse teste, três situações foram simuladas: 1) Na primeira, somente o nó 0 envia tráfego ao nó 1 – denominado tráfego básico; 2) Na segunda ambos nós enviam tráfego simétrico, isto é, quantidades de tráfego iguais; 3) Na última, ambos nós enviam tráfego assimétrico, com o nó 0 enviando um quantidade de tráfego 10 vezes maior que o enviado pelo nó 1. Diante destas situações, dois cenários gerais foram realizados; sem e com *jitter*. Para os cenários sem *jitter*, as conexões chegam no mesmo instante (dependendo da intensidade do tráfego aplicada), já que o tráfego é constante. Com isso o atraso dos pacotes aumenta (no caso de duas ou mais estações estarem enviando no mesmo instante), pois o protocolo MAC faz o controle do acesso ao meio baseado em contenção das fontes.

Os resultados são ilustrados na Figura 4.1. São mostradas duas curvas do atraso médio fim-a-fim em função da carga total na rede para o caso sem/com *jitter*, sendo a segunda curva uma ampliação da primeira até o ponto de saturação da rede. Até 1.5 Mbps a fração de pacotes entregues foi 100%. No cenário sem *jitter*,

Figura 4.1(b), verifica-se que o tráfego simétrico obteve o pior resultado, como esperado. Isto porque os dois nós enviam intensidades de tráfegos similares e nos mesmos instantes, sendo que ambos disputam a posse do meio sem fio. O tráfego assimétrico de 0 para 1 resulta em um atraso um pouco pior do que o do caso básico, visto que o primeiro tráfego é um pouco influenciado pelo baixo tráfego de 1 para 0.



(a) Sem *jitter* e sem *zoom*.

(b) Sem *jitter* e com *zoom*.

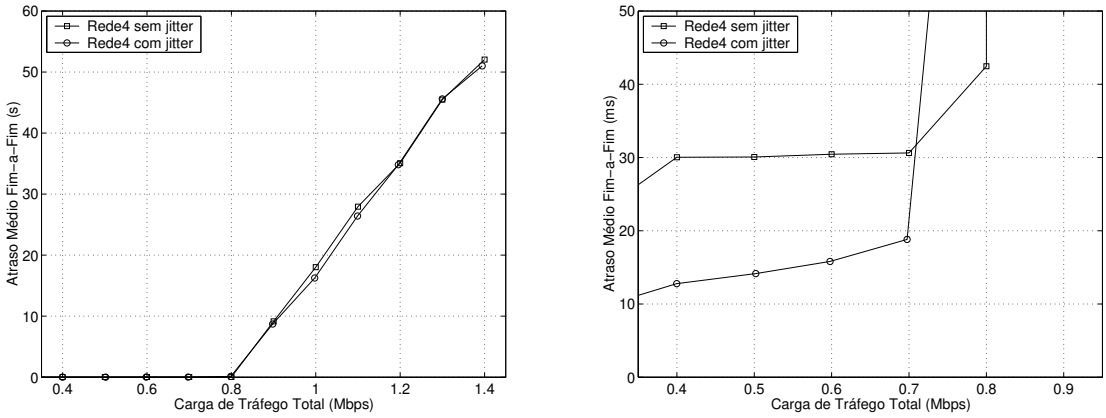
(c) Com *jitter* e sem *zoom*.

(d) Com *jitter* e com *zoom*.

Fig. 4.1: Resultados para a rede de dois nós.

A Figura 4.1(d) mostra o cenário com *jitter*. Os pacotes chegam aleatoriamente e, portanto, sofrem menos ação do processo de contenção do protocolo MAC. A diferença marcante é que o atraso do tráfego assimétrico de 1 para 0 foi o pior. Como o tráfego dos nós origens 0 e 1 são enviados de modo aleatório, considerando o caso do tráfego simétrico, as estações não precisam aguardar tanto tempo para transmitirem, quando comparado ao cenário sem *jitter*.

O segundo teste ilustra a rede de quatro nós apresentada na Seção 3.4.1, com intuito de validar o modelo analítico. Nela, há tráfego somente entre os nós adjacentes; isto gera um total de 4 conexões. Os resultados são esboçados na Figura 4.2. A fração de pacotes entregues foi 100% até 0.8 Mbps. Mais uma vez, observando a Figura 4.2(b), o caso com *jitter* obteve um atraso médio fim-a-fim menor, devido à baixa atuação do protocolo da camada MAC quando comparado ao caso sem *jitter*.



(a) Sem *zoom*. (b) Com *zoom*.

Fig. 4.2: Resultados para a rede de quatro nós.

Para os dois testes realizados foi feito um estudo detalhado dos atrasos dos pacotes. O atraso médio fim-a-fim foi dividido em três intervalos de tempo: tempo de espera no *buffer*, tempo de transferência e atraso do ACK; estes são mostrados na Tabela 4.1. Em particular, o tempo de transferência é medido na camada física, incluindo o atraso devido às todas possíveis colisões que o pacote venha a sofrer. Para uma análise da discrepância entre os resultados, são apresentados os valores de intensidade de tráfego para os casos em que a rede atinge sua saturação máxima de tráfego na primeira parte da tabela correspondente à rede estudada e o valor sucessor da saturação obtida na segunda. Mais uma vez, em média, os atrasos dos pacotes para o caso em que as conexões utilizam *jitter* são menores.

Rede de 2 Nós						
Teste	**Intensidade de Tráfego (Mbps)**	***Jitter***	**Atraso Médio Fim-a-Fim (ms)**	**Tempo de Espera no *Buffer* (ms)**	**Tempo de Transferência (ms)**	**Atraso do ACK (ms)**
Rede2 bas	1.5	não	4.72	0.30	4.39	0.30
Rede2 sim			8.09	3.67	4.39	0.30
Rede2 assim (0->1)	1.44	não	7.15	2.73	4.40	0.30
Rede2 assim (1->0)			8.34	3.97	4.34	0.30
Rede2 bas	1.5	sim	8.10	3.68	4.39	0.30
Rede2 sim			10.86	6.49	4.34	0.35
Rede2 assim (0->1)	1.44	sim	12.32	7.90	4.40	0.30
Rede2 assim (1->0)			9.77	5.41	4.34	0.30
Rede2 bas	1.6	não	502.79	498.38	4.39	0.30
Rede2 sim			829.31	824.90	4.39	0.30
Rede2 assim (0->1)	1.55	não	2270.22	2265.85	4.39	0.30
Rede2 assim (1->0)			11.17	6.80	4.34	0.30
Rede2 bas	1.6	sim	379.36	374.99	4.34	0.30
Rede2 sim			997.884	993.42	4.39	0.30
Rede2 assim (0->1)	1.55	sim	2219.49	2215.12	4.34	0.30
Rede2 assim (1->0)			11.24	6.87	4.34	0.30
Rede de 4 Nós						
Rede4	0.7	não	30.03	10.76	19.24	0.60
Rede4		sim	18.81	4.93	13.86	0.60
Rede4	0.8	não	42.45	15.66	26.77	0.60
Rede4		sim	130.65	48.23	82.39	0.60

Tab. 4.1: Detalhes do atraso dos pacotes para cada teste realizado.

4.3 Descrição dos Experimentos

O principal objetivo dos experimentos é avaliar a habilidade de os protocolos de roteamento reagirem às mudanças na topologia durante a entrega dos pacotes ao seus destinos. Para isso, a metodologia básica é aplicar uma variedade de tráfegos na rede a ser simulada. Cada pacote originado testa se o protocolo de roteamento pode obter uma rota para seu destino a qualquer momento.

O desempenho do protocolo proposto foi avaliado através do simulador de rede ns-2, versão 2.28 [45]. Para criação do ambiente de simulação, dois tipos de arquivos são usados, um descrevendo o padrão de movimento dos nós e o outro, o padrão de tráfego da rede. As simulações modelaram uma rede de estações móveis formando uma rede ad hoc sem fio, movendo-se em uma área retangular (*site*) de 1500 m x 300 m durante 300 segundos de tempo simulado. Foi escolhida uma área retangular para permitir o uso de rotas longas quando comparadas com as de uma área quadrangular com mesma densidade de nós. As características físicas do rádio para cada interface de rede de um nó móvel, tais como, ganho da antena, potência transmitida e sensibilidade do receptor, foram ajustadas aproximadamente ao *radio spectrum spread sequence direct* (DSSS) de uma Lucent WaveLAN [55]. A interface WaveLAN opera como um meio de rádio compartilhado com uma taxa de *bits* nominal de 2 Mbps e um alcance de transmissão nominal de 250 m. Esta versão do simulador provê uma especificação das camadas física e MAC incluindo a modelagem de *backoff*, contenção, colisões, captura e propagação. O IEEE 802.11 DCF é usado como protocolo da camada MAC, utilizando o CSMA/CA. O modelo de propagação combina ambos modelos do espaço livre e terreno plano. O tamanho do *buffer* de cada estação é igual a 50 pacotes. Os demais parâmetros de configuração foram mantidos em seus valores padrões. A Tabela 4.2 resume os principais parâmetros utilizados nas simulações.

Parâmetros Gerais da Simulação	
Simulador	ns-2.28
Número de simulações	10
Área do site	1500 m x 300 m
Tempo de simulação	300 segundos
Camada de transporte	UDP/TCP
Tipo da conexão	CBR/FTP
Camada MAC	IEEE 802.11 DCF
Taxa dos pacotes (CBR)	4 pacotes/segundo
Tipo do emissor–TCP	Agent/TCP
Tipo do receptor–TCP	Agent/TCPSink
Tamanho da janela (TCP)	32
Tamanho dos pacotes	512 bytes
Tamanho do buffer	50 pacotes
Largura de banda	2 Mbps
Alcance de transmissão	250 m
Alcance de interferência	550 m
Limiar de captura	10 dB
Potência de transmissão	24.5 dBm
Freqüência da portadora	914 MHz
Modelo de propagação	Terreno plano
Tipo da antena	Omni antenna

Tab. 4.2: Parâmetros gerais de configuração da simulação.

Nas simulações, comparou-se a versão básica do DSR e a extensão proposta, MCOR, que inclui a informação de congestionamento. Os valores da quantidade de pacotes presentes na fila de transmissão para habilitar as otimizações da propagação do ROUTE REQUEST e do *salvaging* descritas na Seção 2.4.2 para o MCOR foram iguais a 15 e 10, respectivamente. Se o número de pacotes presentes no *buffer* de um nó intermediário for maior que 15, o ROUTE REQUEST não é propagado; a mesma idéia é válida para o processo de *salvaging*. Estes valores foram escolhidos por experimentação e ajustados através de testes preliminares da simulação.

Para os testes realizados, foi utilizado um Pentium 4 de 2.8 GHz com 512 MB de memória rodando versão Fedora 3 do Linux. Cada rodada da simulação, equivalente a 300 segundos simulados (5 minutos), requereu de 5 a 80 minutos de tempo computacional, dependendo do número de fontes e da mobilidade dos nós.

4.3.1 Modelos de Movimento e Comunicação

Os nós na simulação movem-se de acordo com o modelo *Random Waypoint* [19]. Esse modelo de movimento é caracterizado por um valor de ***tempo de pausa*** (*pause time*) que afeta o comportamento de movimento de cada nó móvel e, portanto, a mudança da topologia. Os cenários são gerados ajustando um certo valor de tempo de pausa, e seguindo o seguinte algoritmo:

> Cada nó inicia a simulação em uma posição aleatória dentro da área simulada. Depois que a simulação começa, o nó mantém-se estacionário por um período de *tempo de pausa* segundos. Ele então seleciona aleatoriamente uma nova posição na área, movendo-se com uma velocidade uniformemente distribuída entre uma velocidade mínima (quando não especificada é igual a 0) e uma máxima. Atingindo o destino, o nó aguarda novamente um período de *tempo de pausa* segundos, e seleciona um outro destino. Esse processo é repetido até o final da simulação.

O desempenho dos dois protocolos foi comparado considerando-se o uso de fontes CBR (*Constant Bit Rate*) e FTP (*File Transfer Protocol*), utilizando os protocolos de transporte UDP e TCP, respectivamente. Cada conexão foi escolhida aleatoriamente entre um par origem-destino. O tamanho dos pacotes foi fixado em 512 *bytes*. No caso da fonte CBR, a taxa de envio é de 4 pacotes por segundo. Cada conexão é ponto-a-ponto, iniciando-se a partir de uma distribuição uniforme entre 0 e 180 segundos, permanecendo ativa até o final da simulação.

Desde que o TCP possui um controle de congestionamento baseado no AIMD (*Additive Increase Multiplicative Decrease*), as estatísticas em cada nó possui um significado menor do aquelas obtidas pelo UDP. Sendo assim, considerou-se a dinâmica dos pacotes fim-a-fim, discutindo a vazão (*throughput*) média no destino e o atraso médio fim-a-fim de cada conexão.

Os dois protocolos, DSR e MCOR, foram submetidos às mesmas condições de ambiente e tráfego. Cada rodada do simulador aceita como entrada dois arquivos que descrevem cada cenário: o primeiro descreve o padrão de movimento dos nós, informando o instante exato do movimento de cada nó no decorrer da simulação; e o segundo caracteriza o padrão de tráfego da rede, indicando os pares origem-destino e o início de cada fonte na rede. Foi gerado um total de 350 arquivos de cenário variando o padrão de movimento e tráfego na rede.

4.3.2 Métricas

As seguintes métricas foram utilizadas para a avaliação dos dois protocolos de roteamento:

- **Fração de pacotes entregues**: relação entre o número de pacotes recebidos com sucesso em seus respectivos destinos e o número de pacotes originados pela camada de aplicação.
- **Atraso médio fim-a-fim**: tempo transcorrido entre a origem do pacote na aplicação fonte até seu recebimento na aplicação destino.
- **Overhead de roteamento**: relação entre o número de pacotes de controle de roteamento e o número de pacotes de dados entregues no destino. Cada *hop* de um pacote de controle de roteamento é contado como transmissão de um novo pacote de controle.
- **Número médio de hops**: número médio de *hops* utilizados pelos pacotes durante sua transmissão.

A fração de pacotes entregues descreve a taxa de perda resultante dos protocolos de transporte, tendo efeito na vazão máxima que a rede pode suportar. Esta métrica caracteriza ambas qualidade e corretude do protocolo de roteamento.

O *overhead* de roteamento relaciona-se com a escalabilidade do protocolo, sua atuação em ambientes congestionados ou de baixa carga além de sua eficiência em termos de consumo da energia das baterias dos nós.

O atraso médio fim-a-fim mostra a eficiência do protocolo de roteamento, principalmente em ambientes com alto tráfego. Ele inclui todos os atrasos possíveis causados por armazenamento durante a Descoberta de Rota, enfileiramento na interface de rede, contenção e retransmissão na camada MAC e tempos de transferência e propagação.

O número médio de *hops* indica a habilidade de o protocolo utilizar eficientemente os recursos da rede, selecionando o caminho mais curto da origem até destino (embora o caminho mais curto nem sempre é o mais desejado, já que este pode estar sobrecarregado).

Os dados estatísticos das métricas utilizadas nos experimentos foram coletados desde o início da simulação, ou seja, a partir de 0 segundos. Em alguns trabalhos, como [36], as métricas são avaliadas somente depois dos 30 segundos iniciais de tempo simulado, após os nós finalizarem o processo de inicialização.

4.3.3 Cenários Estudados

Para se avaliar os dois protocolos, foram feitas algumas combinações a partir de quatro parâmetros ajustáveis nas simulações de uma rede ad hoc móvel: número de nós, tempo de pausa, velocidade máxima do nó móvel e número de conexões utilizadas. Foram gerados quatro cenários variando-se um parâmetro, e mantendo os outros três fixos. Estes são descritos a seguir:

- **Cenário 1**: o estudo foi feito a partir da mobilidade dos *hosts*. Foram utilizados padrões de movimento com sete velocidades máximas permitidas dos nós - 1, 5, 10, 15, 20, 25 e 30 m/s, 50 móveis, tempo de pausa de 10 segundos e 30 conexões.

- **Cenário 2**: foi feita uma apreciação da carga de tráfego na rede. Utilizaram-se padrões de tráfego com um conjunto de seis conexões diferentes - 10, 20, 30, 40, 50 e 60 conexões, 50 móveis, tempo de pausa de 0 segundos, e velocidade máxima permitida de 1 e 10 m/s.

- **Cenário 3**: testou-se a densidade da rede sem fio. Foram utilizados padrões de movimento com sete números de móveis diferentes - 20, 30, 40, 50, 60, 70 e 80 móveis, tempo de pausa de 0 segundos, velocidade máxima dos nós de 1 e 20 m/s e 20 conexões.

- **Cenário 4**: por último, foi realizada uma análise sob o tráfego TCP. Parâmetros utilizados: 50 móveis, tempo de pausa de 0 segundos, velocidade máxima dos nós de 1 e 20 m/s e 30 conexões.

Na Tabela 4.3 são mencionados os parâmetros utilizados em cada cenário, sendo que nos três primeiros são aplicadas conexões CBR e no último, conexões FTP.

Parâmetros dos Cenários				
	Número de Móveis	**Tempo de Pausa (s)**	**Velocidade Máxima (m/s)**	**Número de Conexões**
Cenário 1	50	10	1, 5, 10, 15 20, 25, 30	30
Cenário 2	50	0	1, 10	10, 20, 30, 40 50, 60
Cenário 3	20, 30, 40, 50, 60, 70, 80	0	1, 20	20
Cenário 4	50	0	1, 20	30

Tab. 4.3: Parâmetros dos cenários estudados.

Visto que o desempenho dos protocolos é bastante sensível ao padrão de movimento, foram gerados 10 arquivos de padrões de movimento (10 sementes) para cada parâmetro variável dos cenários estudados. Nos cenários 1, 2, 3 e 4 foram utilizados 70, 120, 140 e 20 arquivos respectivamente, resultando em 350 arquivos de cenários. Ambos os protocolos, DSR e MCOR, foram estudados sistematicamente com os mesmos 350 cenários.

4.3.4 Metodologia da Simulação

A Figura 4.3 apresenta a metodologia básica dos experimentos das simulações. Cada rodada do simulador aceita um arquivo de cenário, composto pelos arquivos de padrão de movimento e tráfego. O arquivo de saída (*trace file*) criado por cada simulação é armazenado em disco e analisado por uma variedade de *scripts awk*, que computa todas as métricas e informações adicionais da simulação. A partir destes dados, os gráficos são traçados no matlab. Todos os arquivos utilizados na simulação, além da implementação do MCOR em C++ podem ser consultados em [56].

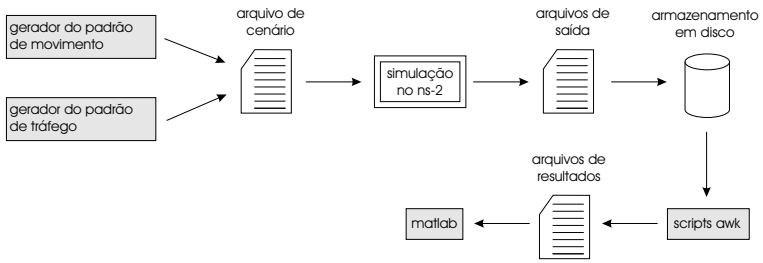


Fig. 4.3: Resumo da metodologia de simulação.

4.4 Resultados

Nesta seção são apresentados os resultados para cada um dos cenários. Uma avaliação sobre as métricas empregadas nos cenários é discutida na Seção 4.4.5. Os resultados de cada simulação foram obtidos dentro de um intervalo de confiança de 95%, calculado pelo método de *Bootstrap* [57]. Cada ponto no gráfico representa uma média de 10 rodadas para a mesma configuração dos cenários estudados, ou seja, utilizaram-se 10 sementes diferentes.

4.4.1 Cenário 1

Neste cenário, estudou-se o impacto da mobilidade nas métricas de desempenho. Note que também pode-se alterar o modelo de mobilidade da rede variando o tempo de pausa e fixando a velocidade máxima dos nós. O número de *hosts* móveis foi fixado em 50, o número de conexões em 30 e o tempo de pausa em 10 segundos. As velocidades máximas permitidas para cada nó são: 1, 5, 10, 15, 20, 25 e 30 m/s.

A fração de pacotes entregues para ambos protocolos foi menor que 77%, como mostrado na Figura 4.4(a). Quando a mobilidade é baixa (velocidade máxima de 1 m/s), o DSR e o MCOR entregam aproximadamente a mesma quantidade de pacotes. À medida que a mobilidade aumenta, as taxas entregues de ambos protocolos caem progressivamente, mas com uma diminuição mais acentuada do DSR.

O atraso médio fim-a-fim do MCOR foi bem inferior do que o DSR, como mostrado na Figura 4.4(b). Para o caso da velocidade máxima permitida de 30 m/s, a diferença entre os protocolos chegou a 2 s.

A carga de roteamento dos dois protocolos aumenta com o nível de mobilidade da rede sem fio, conforme exibido na Figura 4.4(c). O MCOR apresentou um desempenho bem superior ao DSR, diminuindo o *overhead* em 42% para mobilidade de 30 m/s.

Como apresentado na Figura 4.4(d), o número médio de *hops* manteve-se comparativamente estável com o aumento da mobilidade, com resultados um pouco melhores do MCOR.

Para validação dos resultados das simulações do cenário 1, a Tabela 4.4 apresenta os intervalos de confiança determinados pela técnica *Bootstrap* aplicada na diferença entre as métricas obtidas pelo DSR e o MCOR em cada uma das dez amostras da simulação para três velocidades máximas. Os resultados mostram a robustez e a consistência do MCOR em todas as métricas avaliadas.

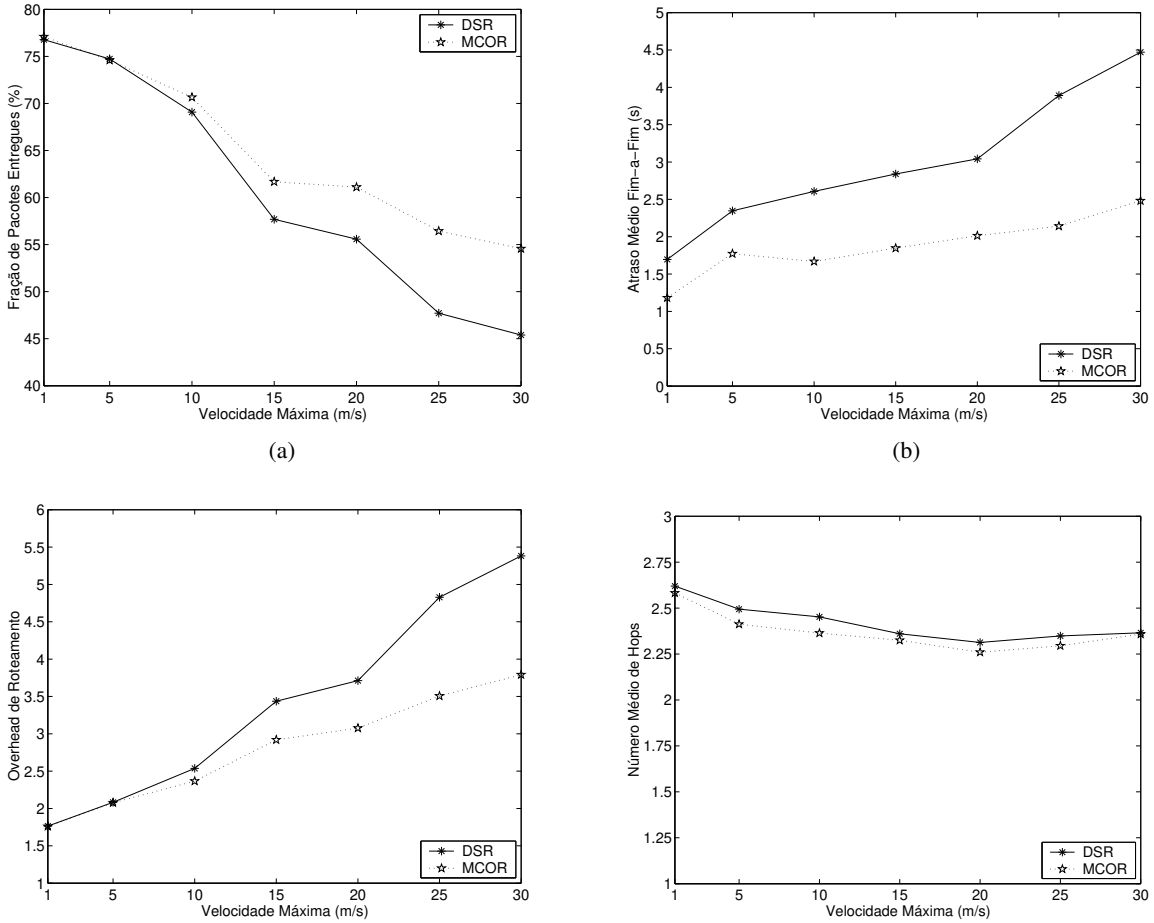


Fig. 4.4: Resultados para o cenário 1.

Métrica	Velocidade Máxima (m/s)	Limite Inferior	Média	Limite Superior
Fração de pacotes entregues (%)	10	– 2.87	– 1.57	– 0.40
	20	– 6.51	– 5.47	– 4.46
	30	– 10.58	– 9.18	– 7.83
Atraso médio fim-a-fim (s)	10	0.68	0.94	1.16
	20	1.00	1.28	1.57
	30	1.72	1.99	2.29
Overhead de roteamento	10	0.11	0.17	0.25
	20	0.47	0.56	0.61
	30	1.37	1.59	1.88
Número médio de hops	10	0.05	0.08	0.12
	20	0.05	0.07	0.09
	30	– 0.01	0.03	0.06

Tab. 4.4: Resultados do intervalo de confiança de 95% para o cenário 1.

4.4.2 Cenário 2

Estes experimentos permitem analisar a variação da carga de tráfego na rede. A Figura 4.5 exibe os resultados. O número de estações móveis é 50, a velocidade máxima é 10 m/s (média de 5 m/s) e não há pausa. O

número de conexões varia entre 10 e 60.

A fração de pacotes entregues do DSR, Figura 4.5(a), varia de 98.51% a 51.22%, enquanto que do MCOR varia de 98.64% a 53.78%; ambos quando o número de conexões aumenta de 10 para 40. Para cargas superiores à 40 conexões (655 kbps), ambas taxas do DSR e MCOR diminuem gradualmente, pois a rede está sobrecarregada.

Como delineado na Figura 4.5(b), o MCOR possui atrasos médios inferiores aos do DSR.

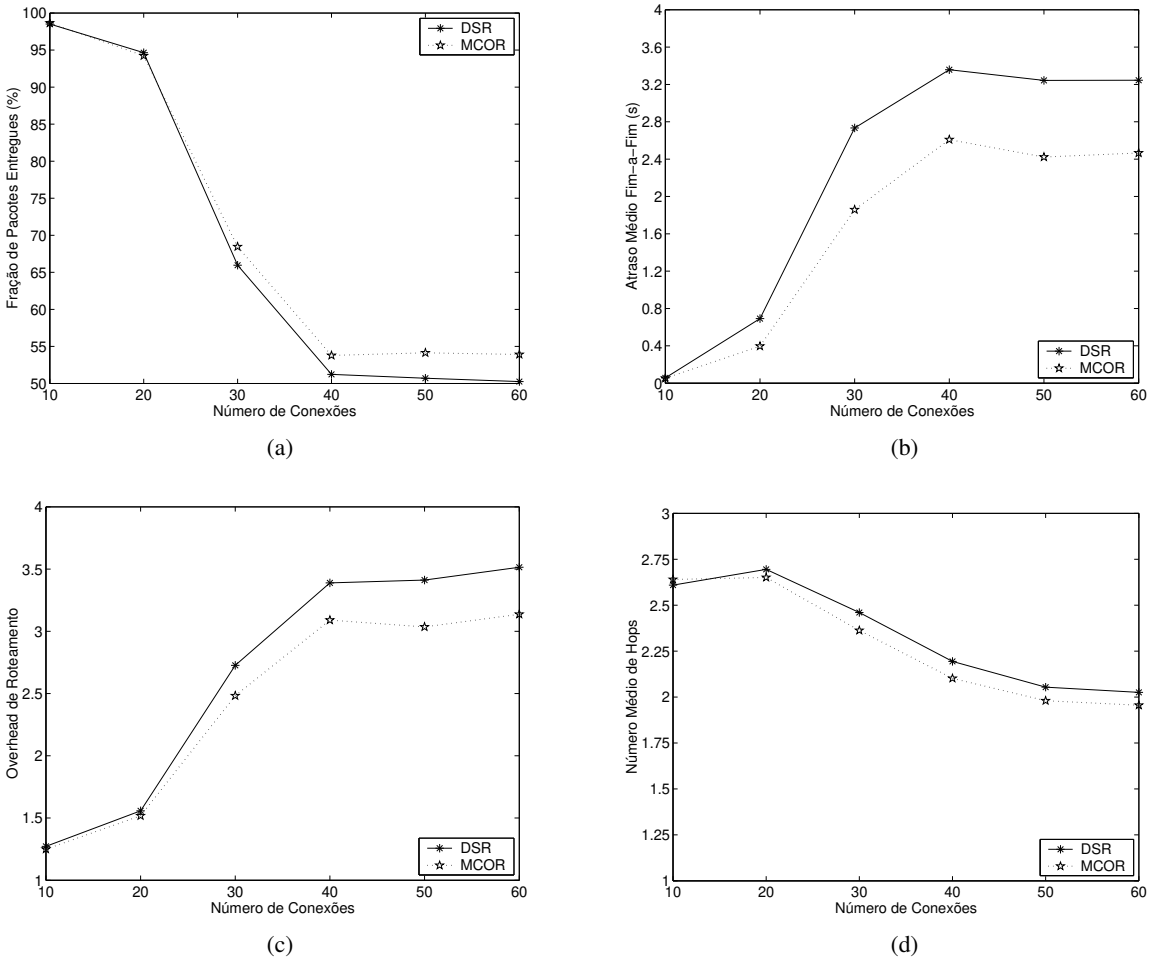


Fig. 4.5: Resultados para o cenário 2, com os móveis deslocando-se com velocidade máxima de 10 m/s.

O *overhead* de roteamento do MCOR também é inferior ao DSR, conforme mostrado na Figura 4.5(c).

Neste cenário, o número médio de *hops* sofreu uma diminuição considerável comparado aos outros cenários, pois manteve-se o número de nós constante e aumentou-se o número de conexões, e portanto, a chance de um certo nó enviar tráfego a um nó vizinho mais próximo.

Os resultados para o cenário 2 com baixa velocidade dos nós – velocidade máxima de 1 m/s, são exibidos na Figura 4.6. Os maiores ganhos do MCOR em relação ao DSR foram no atraso médio fim-a-fim, Figura 4.6(b).

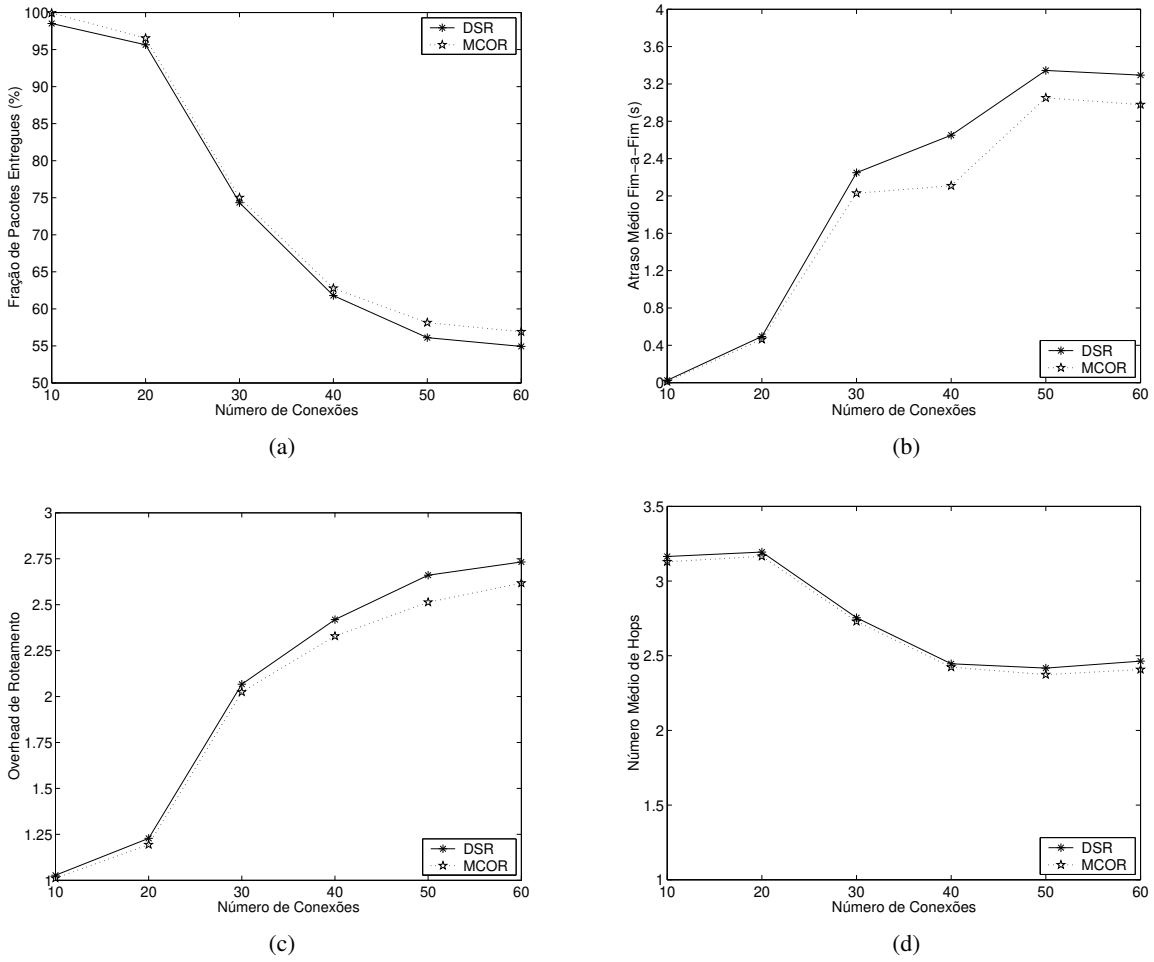


Fig. 4.6: Resultados para o cenário 2, com os móveis deslocando-se com velocidade máxima de 1 m/s.

4.4.3 Cenário 3

O terceiro conjunto de experimentos investiga o efeito da densidade da rede sem fio. Todos os móveis movem-se aleatoriamente em uma velocidade máxima de 20 m/s (média de 10 m/s) continuamente no espaço 1500 m x 300 m (tempo de pausa de 0 segundos). O número de móveis varia entre 20 e 80. O número de conexões é 20. A Figura 4.7 mostra os resultados para o cenário 3.

A taxa de entrega dos pacotes do DSR diminui mais rapidamente do que o MCOR com o aumento do número dos *hosts* móveis. Para uma rede densa de 70 móveis, o MCOR entregou 77.29% dos pacotes oferecidos, enquanto a versão básica do DSR entregou apenas 72.10%, tendo um ganho, portanto, de aproximadamente 7%. Entretanto, para uma rede esparsa, com 30 móveis, o MCOR não demonstrou grande diferença, entregando 90.13% dos pacotes enquanto o DSR 89.47%.

A Figura 4.7(b) mostra que o atraso médio do MCOR é bem inferior ao obtido pelo DSR. Para o número máximo de estações utilizadas, o atraso médio para os pacotes entregues com sucesso para o MCOR foi de 1.91 s, enquanto para o DSR, 3.64 s.

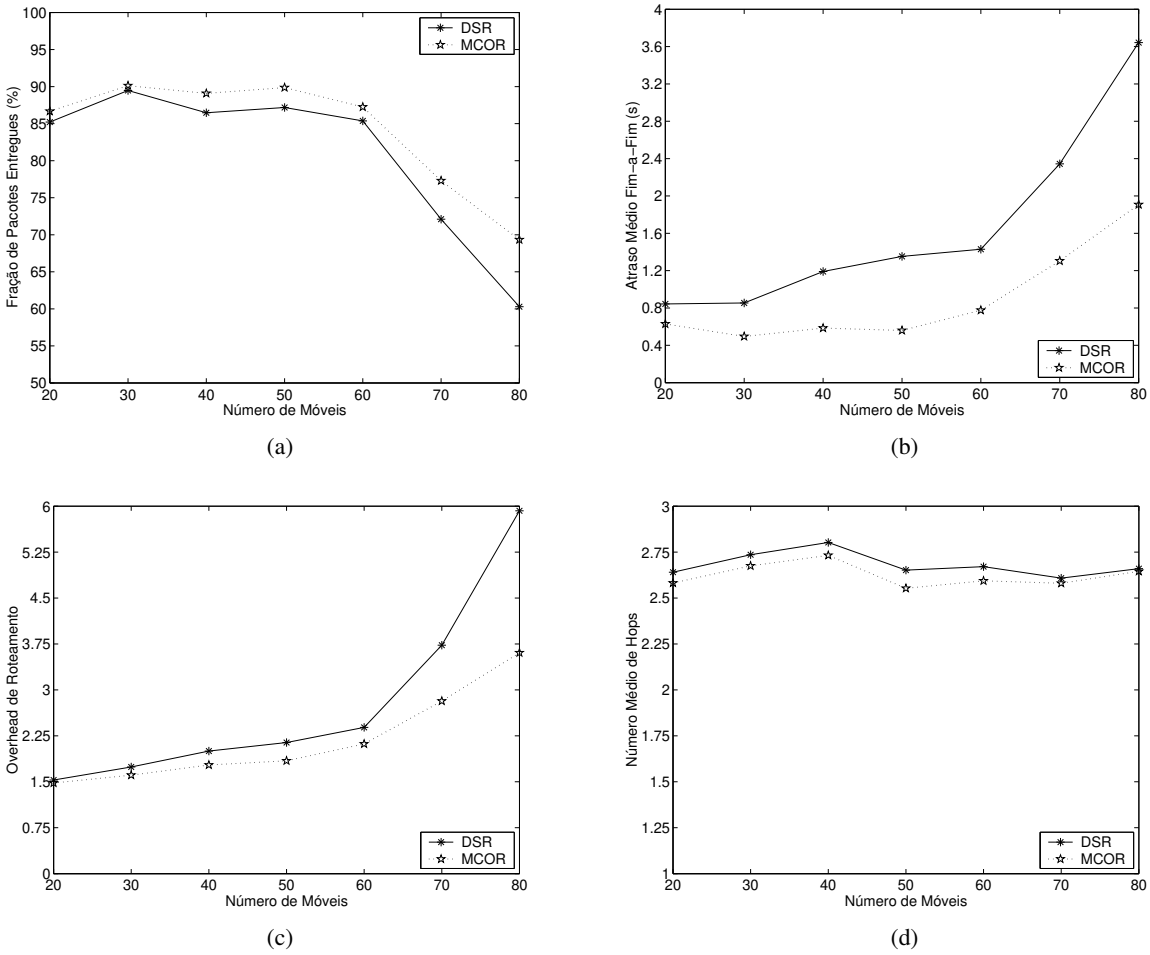


Fig. 4.7: Resultados para o cenário 3, com os móveis deslocando-se com velocidade máxima de 20 m/s.

Com relação ao *overhead* de roteamento, apresentado na Figura 4.7(c), o MCOR não demonstrou ganhos consideráveis em relação ao DSR até o cenário de 60 móveis. A partir desse valor, o DSR introduziu um *overhead* de pacotes bem superior do que o MCOR, sendo que para o caso de 80 estações, esse *overhead* foi praticamente o dobro.

Na Figura 4.7(d) é apresentado o número médio de *hops* utilizados pelos dois protocolos. O MCOR foi um pouco melhor que o DSR neste quesito.

A Figura 4.8 mostra os resultados para o cenário 3, considerando uma velocidade máxima de 1 m/s das estações. Observando a Figuras 4.8(a)-(d), percebe-se uma grande flutuação dos resultados. Como já mencionado, os cenários são muito sensíveis aos padrões de tráfego e movimento dos móveis; mas deve-se observar a correspondência entre os resultados: por exemplo, para uma alta taxa de fração de entrega dos pacotes tem-se um atraso médio fim-a-fim baixo, e vice-versa. Mais uma vez, o MCOR não ofereceu grandes melhorias para cenários de baixa velocidade.

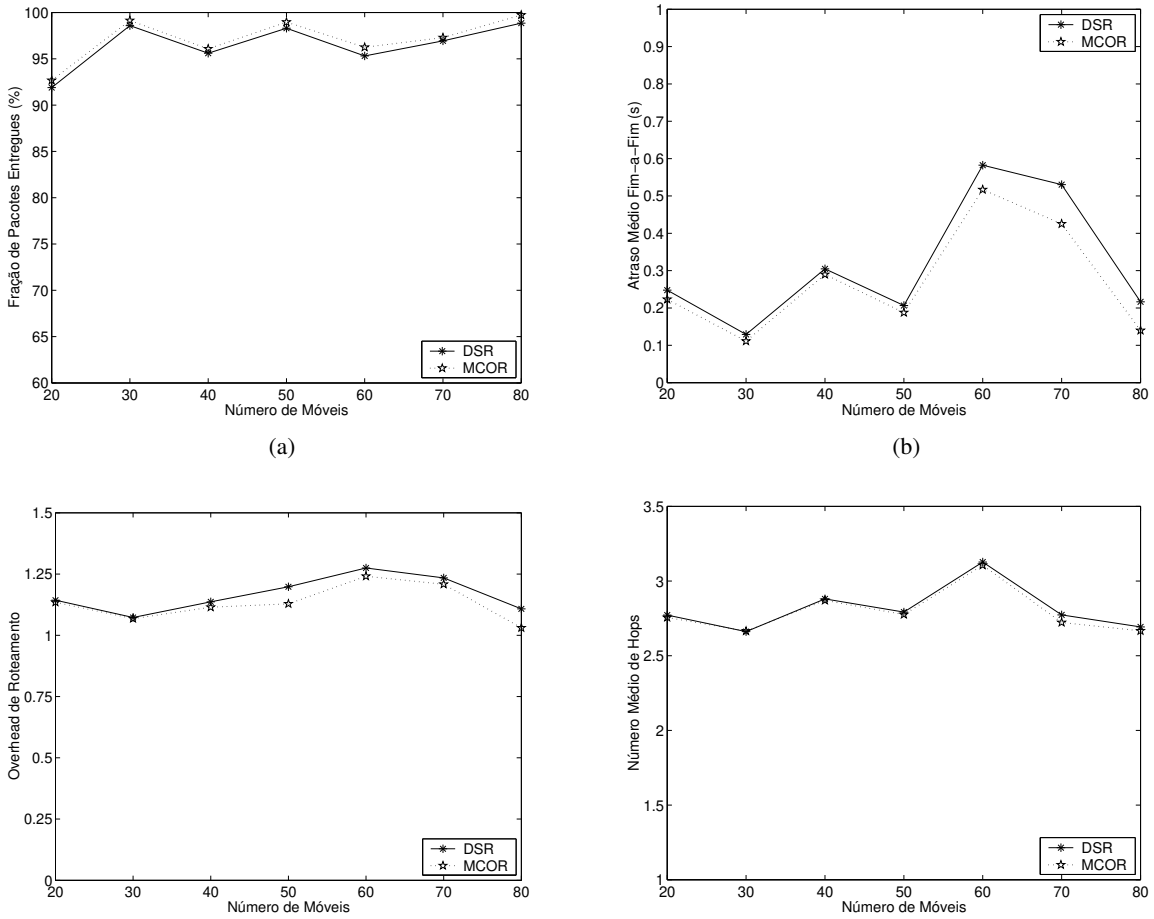


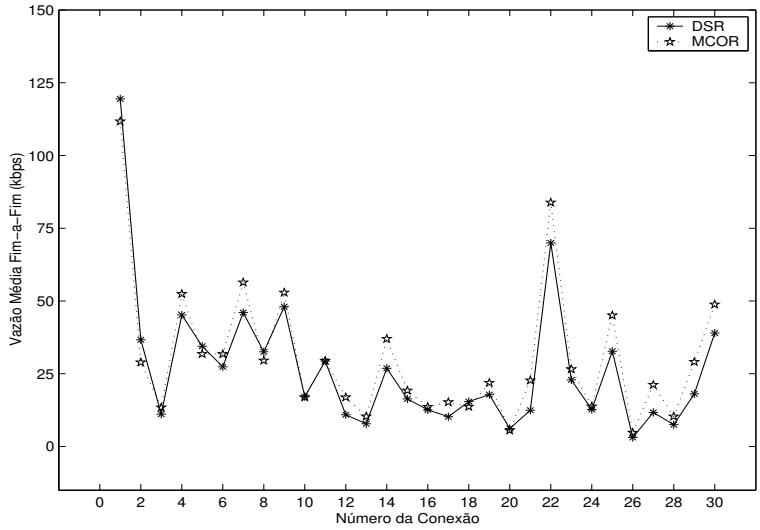
Fig. 4.8: Resultados para o cenário 3, com os móveis deslocando-se com velocidade máxima de 1 m/s.

4.4.4 Cenário 4

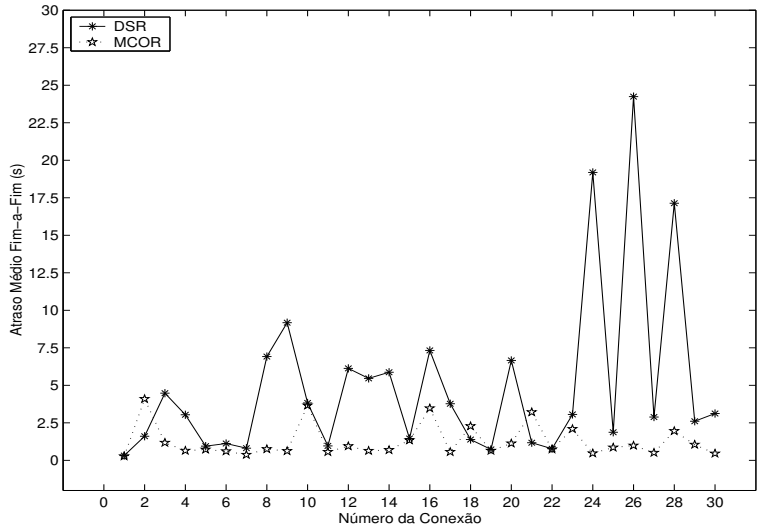
Este último conjunto de experimentos considera a avaliação do tráfego TCP em função da velocidade dos móveis. O número de estações móveis é 50, o tempo de pausa é de 0 segundos e o número de conexões é 30. Duas velocidades máximas dos móveis são consideradas: 20 e 1 m/s. Para os resultados, são considerados somente a vazão média medida no nó destino e o atraso médio fim-a-fim das 30 conexões.

Os resultados para a velocidade máxima de 20 m/s são apresentados na Figura 4.9. Com relação à vazão, Figura 4.9(a), o MCOR oferece melhorias substanciais. Para a conexão 22, por exemplo, o MCOR obteve uma vazão de 83.82 kbps, enquanto o DSR 69.97 kbps. Entretanto, houve casos, como por exemplo a conexão 1, em que o DSR conseguiu uma vazão de 119.45 kbps e o MCOR 111.72 kbps.

Para o atraso médio fim-a-fim das conexões, o MCOR apresenta excelentes resultados, conforme demonstrado na Figura 4.9(b). Na figura, poucos foram os pontos para os quais o DSR foi melhor que o MCOR. Em contrapartida, houve casos extremos, como por exemplo da conexão 26, em que o atraso médio obtido pelo MCOR foi de 0.99 s enquanto pelo DSR foi de 24.24 s.



(a)



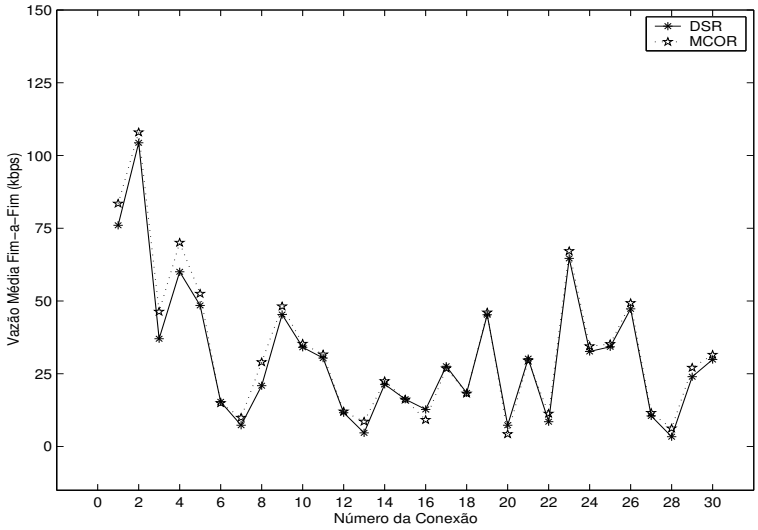
(b)

Fig. 4.9: Resultados para o cenário 4, com os móveis deslocando-se com velocidade máxima de 20 m/s.

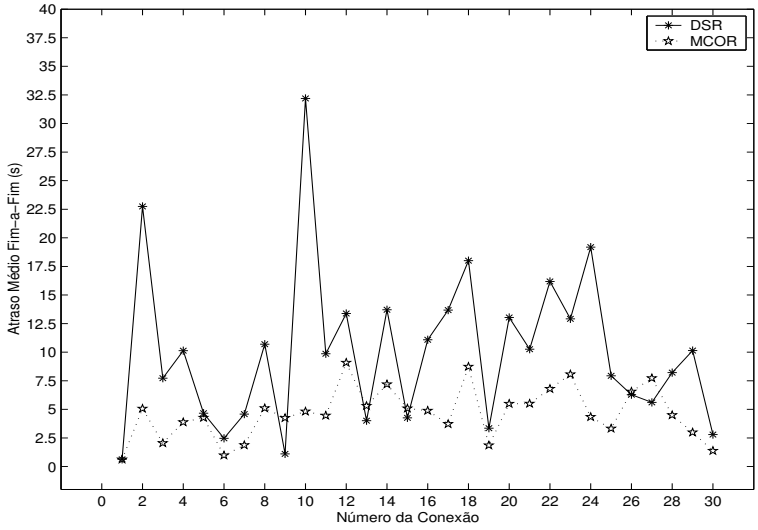
Consideram-se, na Figura 4.10, os resultados para as estações movimentando-se em uma velocidade máxima de 1 m/s. A vazão média do MCOR demonstrou poucas melhorias em relação ao DSR, como mostrado na Figura 4.10(a). Entretanto, para o atraso médio fim-a-fim – Figura 4.10(b), o MCOR obteve bons resultados, como no caso da conexão 10, na qual o MCOR produziu um atraso de 4.81 s enquanto o DSR de 32.19 s. Analisando as Figuras 4.9(b) e 4.10(b), percebe-se que para uma velocidade máxima alta de 20 m/s, o MCOR foi mais estável do que para o caso de baixa velocidade dos nós de 1 m/s.

Para ambos os casos, a informação de carga do MCOR, que seleciona as rotas menos congestionadas, possui impacto marcante na atuação do protocolo TCP. Ela permite que a partida lenta[3] (*slow start*) do TCP

[3] Durante o controle de congestionamento implementado pelo TCP, duas fases são executadas: a fase de partida lenta, onde o



(a)



(b)

Fig. 4.10: Resultados para o cenário 4, com os móveis deslocando-se com velocidade máxima de 1 m/s.

seja mais rápida quando utilizado o DSR, ocasionando, portanto, um aumento do número de pacotes entregues no destino e uma diminuição do atraso médio.

A Tabela 4.5 mostra os resultados da média e dos limites inferior e superior do intervalo de confiança para algumas conexões obtidas pelo *Bootstrap*. Observe que em média, a vazão do DSR é menor que a do MCOR em 1.6 kbps e o atraso médio é maior em 3.69 s.

crescimento da janela de congestionamento é exponencial e a fase de controle de congestionamento, na qual o crescimento é linear.

Métrica	Número da Conexão	Limite Inferior	Média	Limite Superior
Vazão média fim-a-fim (kbps)	4	– 19.58	– 7.22	3.37
	14	– 0.41	3.11	6.57
	20	– 0.19	0.68	1.62
	26	– 3.64	– 1.64	0.34
	28	– 10.70	– 2.80	2.75
	1–30	– 3.29	– 1.60	– 0.27
Atraso médio fim-a-fim (s)	4	– 0.47	2.39	7.58
	14	0.01	4.62	13.31
	20	– 0.04	5.53	12.75
	26	1.34	23.25	52.47
	28	1.06	15.18	37.44
	1–30	1.96	3.69	5.48

Tab. 4.5: Resultados do intervalo de confiança de 95% para o cenário 4 com velocidade máxima de 20 m/s.

4.4.5 Comentários Gerais

Nesta seção são discutidos os resultados para cada métrica avaliada nos cenários estudados, especialmente os três primeiros.

A. Fração de Pacotes Entregues

Em todos os cenários, o MCOR obteve desempenho superior ao DSR sob a ótica da taxa de entrega dos pacotes. Três razões explicam esse fenômeno. Primeiro, o MCOR utiliza uma métrica que escolhe a rota com menos tráfego, reduzindo a perda de pacotes por saturação do *buffer* nos nós congestionados. Em contraste, o DSR básico não considera nenhuma informação de carga na rede na seleção da rota, escolhendo sempre o menor caminho disponível no momento o qual pode incluir nós congestionados. Segundo, o DSR usa o mecanismo de *cache* para retornar um ROUTE REPLY ao iniciador; isto pode resultar em informação desatualizada e caminhos errados para os pacotes. Esse fato é agravado principalmente no primeiro cenário no qual há grande mobilidade dos nós, e portanto, muitas quebras de enlaces. Assim, muitos pacotes são perdidos no processo de Manutenção de Rota, ocasionando um desempenho ruim do DSR. Terceiro, a otimização da propagação de ROUTE REQUESTs do MCOR a partir da medição do congestionamento evita possíveis áreas congestionadas no processo de descoberta, disseminando a carga de tráfego de forma mais balanceada na rede.

Para os cenários 2 e 3, nos quais as cargas de tráfego e/ou número de estações é baixo, a melhoria da fração de pacotes entregues pelo MCOR é relativamente pequena, pois existe pouca chance de se formarem áreas congestionadas.

B. Atraso Médio Fim-a-Fim

Dentre todas as métricas avaliadas, o atraso médio fim-a-fim foi o que apresentou as maiores discrepâncias entre os dois protocolos, pois o MCOR seleciona a rota com menos tráfego, reduzindo o atraso de enfileiramento nas filas das interfaces de transmissão dos nós. O atraso médio aumentará caso o caminho inclua nós

com alto índice de tráfego. Esta situação pode ocorrer no DSR no qual os nó centrais da rede estão provavelmente sendo sobrecarregados (uso do caminho mínimo), mas é menos provável acontecer no MCOR, desde que o protocolo possui mecanismos para ajustar o tráfego dinamicamente a ocupar rotas menos congestionadas.

Outro fator que possivelmente influenciou o desempenho superior do MCOR foi a execução do *salvaging* baseado na informação de congestionamento. Caso um determinado nó esteja bastante congestionado é preferível descartar o pacote do que reenviá-lo por uma nova rota. Com esta ação, o MCOR além de desconsiderar o tempo de transferência do pacote que estava sendo encaminhado (possivelmente sob uma rota congestionada), diminui o tempo de espera para primeiramente conseguir transmitir o pacote na camada de enlace (já que o pacote pode sofrer novas tentativas de retransmissão através da nova rota) e posteriormente o tempo extra de transferência do pacote gasto para percorrer a nova rota utilizada.

Para os casos nos quais o número de fontes é baixo e o número de móveis é pequeno (cenários 2 e 3, respectivamente), os resultados do DSR e MCOR têm pouca diferença, pois para redes com baixo tráfego e pouco densas, o índice de congestionamento tem pouca importância.

C. Overhead de Roteamento

Nos três cenários, o MCOR implementou melhorias significativas no *overhead* dos pacotes de controle de roteamento. No MCOR, o fato de os nós intermediários não enviarem ROUTE REPLYs contento rotas previamente armazenadas em seu *Route Cache* foi compensado pelo fato de os mesmos descartarem os pacotes de ROUTE REQUEST a fim de evitar nós congestionados. Em contraste no DSR, apesar do uso do *cache* para responder aos pedidos de rota, muitos ROUTE REQUESTs são propagados, inundando a rede sem fio.

Outro fator desvantajoso do DSR é que estes ROUTE REPLYs podem conter informação desatualizada dos enlaces na rota, ocasionando a transmissão de pacotes de ROUTE ERROR; e caso o nó origem não possua uma rota alternativa para o pacote, uma nova Descoberta de Rota deve ser realizada, e portanto, mais ROUTE REQUESTs são enviados para obter uma nova rota.

D. Número Médio de Hops

Os resultados para o número médio de *hops* dos cenários estudados podem parecer estranhos, pois o DSR utiliza o menor caminho disponível para entregar um pacote. Entretanto, se existe uma rota (mínima) entre o nó origem **A** e o nó destino **E** com 2 *hops*, no caso desconhecida por **A**[4], e este nó possui em seu *Route Cache* uma rota com 4 *hops* para **E**, então esta será utilizada.

No caso do MCOR, a rota é estabelecida a partir de um ROUTE REPLY enviado diretamente pelo nó alvo, podendo ser mais curta do que a fornecida pelo DSR, já que a primeira é baseada na informação mais recente das localizações das estações. O DSR, em contrapartida, utiliza rotas presentes nos *caches* dos nós intermediários, as quais não exploram a topologia corrente da rede. Isso explica o melhor desempenho do MCOR em relação ao DSR.

[4] Neste caso o nó iniciador **A** do pedido de rota ainda não recebeu o ROUTE REPLY correspondente a esta rota de 2 *hops* até o nó alvo **E**.

4.5 Resumo do Capítulo

Neste capítulo foram explorados alguns mecanismos que consideram informação de congestionamento em redes ad hoc *multi-hop*. Enquanto alguns trabalhos [58, 59] concentram-se no roteamento baseado na topologia da rede, garantindo alguns parâmetros de serviço aos fluxos, nesta tese foi desenvolvido um novo protocolo que se beneficia da observação de áreas congestionadas da rede sem fio para atender o tráfego, tanto no caso de melhor esforço, como o UDP, tanto no caso de entrega confiável através do TCP, de forma mais eficiente. Uma dada estação é capaz de detectar o grau de ocupação do meio físico em torno de sua área, monitorando o tamanho da fila de sua interface de transmissão. Esta medição reflete não somente no comportamento do nó em questão, mas também nos nós que dividem o mesmo meio físico, e conseqüentemente, nas rotas utilizadas.

Realizou-se uma análise comparativa do DSR básico e sua versão estendida, o MCOR. Para uma avaliação do desempenho dos dois protocolos, quatro cenários foram utilizados: um variando a mobilidade dos *hosts* móveis; o segundo, a carga de tráfego entre os nós; o terceiro, a densidade da rede e o último, o uso de conexões TCP. Comparativamente com o DSR, os resultados mostram que através da inferência da informação de carga da rede, o MCOR conseguiu melhorias substanciais em termos de entrega de pacotes, latência, *overhead* e escalabilidade principalmente nos cenários de alta de baixa mobilidade dos nós. Embora essas melhorias tenham sido feitas tendo como base os procedimentos do DSR, técnicas similares são possíveis também em outros protocolos de roteamento ad hoc. No DSDV, por exemplo, um nó poderia aumentar o tempo entre dois anúncios consecutivos durante os períodos nos quais o meio físico estiver particularmente ocupado; e um *host* móvel utilizando o AODV poderia não realizar um reparo local caso o mesmo detectasse que seu meio físico estivesse congestionado.

Capítulo 5

Modelo de Validação

Despite the refusal of VADM Poindexter and LtCol North to appear, the Board's access to other sources of information filled much of this gap. The FBI provided documents taken from the files of the National Security Advisor and relevant NSC staff members, including messages for the PROF system between VADM Poindexter an LtCol North. The PROF messages were conversations by computer, written at the time events occurred and presumed by the writers to be protected from disclosure. In this sense, they provide a first-hand, contemporaneous account of events.

— The Tower Commision Report,
to President Reagan on the Iran-Contra Affair, 1987

5.1 A Linguagem de Descrição e Especificação

A ferramenta SDL (*Specification and Description Language*) [60, 61] é uma linguagem de programação de alto nível orientada a objetos padronizada pela ITU-T (*International Telecommunication Union*) e com recomendação Z.100 [62]. É utilizada principalmente para a descrição de sistemas complexos de comunicação, tempo real e dirigidos a eventos. Em seu nível de maior abstração, o sistema é descrito em um contexto geral, com todas as interações com o ambiente. Nos demais níveis, são inseridos os componentes do sistema, aumentando assim, o grau de detalhamento do modelo.

Um sistema SDL consiste na interconexão de vários módulos, denominados ***blocos***. Um bloco pode ser recursivamente dividido em outros blocos, formando uma hierarquia. Os ***canais*** definem os caminhos de comunicação entre os blocos e o ambiente. O comportamento de cada bloco baseia-se em ***processos***, os quais são descritos por máquinas de estados finitos extendidas (*extended finite state machines*) executadas concorrentemente. Os processos se comunicam através da troca de ***sinais*** assíncronos, que podem transportar vários parâmetros. Cada processo contém uma fila do tipo FIFO (*First-In Firt-Out*), na qual os sinais são armazenados. Quando um processo recebe um sinal dessa fila, ocorre a transição entre ***estados***.

O SDT – Telelogic Tau™ SDL Suite [63] versão 4.4.0 é o aplicativo usado para a modelagem em SDL do protocolo proposto na tese. Esta ferramenta possui, entre outras, as seguintes funcionalidades:

- SDL Organizer, que permite a modelagem do sistema a ser validado;
- Simulator UI, que permite a simulação do modelo;
- Validator UI, que permite validar o modelo, criando cenários de simulação e apontando falhas, como por exemplo, *deadlocks*.

Para descrição dos principais elementos da linguagem SDL, é apresentada a seguir a modelagem e validação de um protocolo de comunicação conhecido como protocolo *simplex stop-and-wait* [64].

5.2 O Protocolo Simplex Stop-and-Wait

O protocolo *simplex stop-and-wait* é um protocolo de camada de enlace que provê a comunicação entre duas estações: a estação transmissora, após o envio de um quadro (*frame*), deve esperar um sinal de confirmação (*acknowledgment*) antes de enviar o próximo quadro. A estação receptora, assim que receber um quadro da estação remota, deverá enviar o sinal de confirmação à estação transmissora e entregar o quadro recebido para a camada de rede.

Um número de seqüência é inserido no cabeçalho de cada quadro para evitar problemas com quadros duplicados. No caso mais simples, esse número possui apenas 1 *bit* e o número de seqüência dos quadros transmitidos alterna entre 0 e 1, motivo pelo qual o protocolo é também chamado de protocolo de ***bit alternado***. Caso o número de seqüência do quadro recebido seja o número esperado, o quadro é entregue para a camada de rede; caso contrário, este é descartado. A mensagem de confirmação também possui um número de seqüência.

5.2.1 Modelagem através do SDL

O nível do sistema para a modelagem do protocolo *simplex stop-and-wait* é mostrado na Figura 5.1 contendo o bloco Local, representando a camada de enlace da estação local, e o bloco Remoto, representando a camada de enlace da estação remota. Cada bloco pode conter outros blocos ou processos. Ambos os blocos são do tipo Term e contêm a mesma descrição e qualquer alteração no tipo Term é refletida nos blocos Local e Remoto. A interface entre os blocos de mesmo tipo e o ambiente é representada por meio de portas (*gates*) – G1, G2, G3 e G4. O canal denominado R1, transporta o sinal Pacote, originado no ambiente (camada de rede de cada estação) e destinado ao bloco Local. Os sinais e o tipo de informação que transportam são definidos na estrutura chamada Text.

O tipo Term é mostrado na Figura 5.2 que contém o processo Transmissao e o processo Recepcao. A notação $(a, b) = (1, 1)$ junto ao nome dos processos indica que são criadas a instâncias para o processo, podendo ser criadas dinamicamente até o máximo de b instâncias. O processo Transmissao recebe os sinais Pacote e Ack e envia o sinal Quadro; e o processo Recepcao, recebe o sinal Quadro e envia os sinais Dados e Ack.

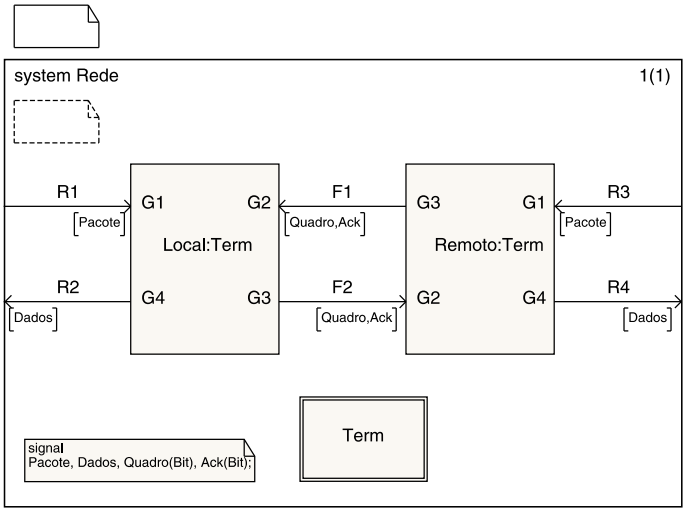


Fig. 5.1: Nível do sistema para a modelagem do protocolo *simplex stop-and-wait*.

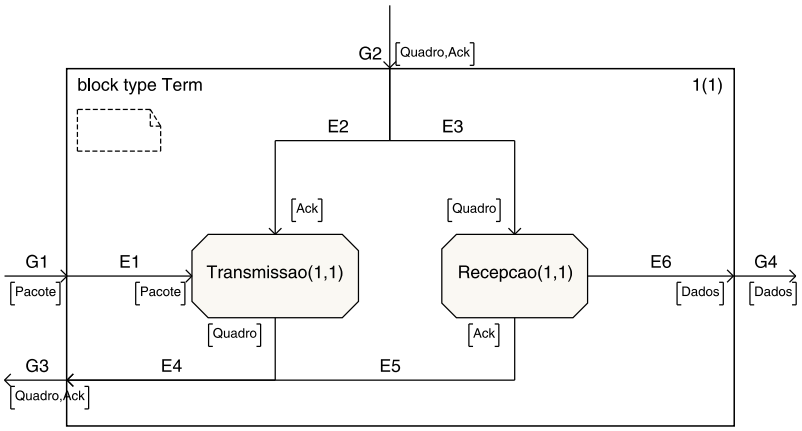


Fig. 5.2: Bloco do tipo Term.

A máquina de estados finitos do processo Transmissao é mostrada na Figura 5.3. A instância do processo Transmissao inicia-se em seu estado inicial, executa a inicialização das variáveis *seq*, *ackrem* e *aresp*, e estaciona no estado *Idle*. A recepção de um sinal Pacote (símbolo *Input*) ativa a transição entre o estado *Idle* e o estado *EsperaAck*, transmitindo o sinal Quadro(seq) (símbolo *Output*). O atributo *seq* contém o número de seqüência do quadro. A recepção de um sinal Ack ativa a transição do estado *EsperaAck* para o estado *Idle* se *ackrem* contiver o valor esperado, senão o processo retorna ao estado *EsperaAck*. Comentários podem ser usados e não afetam a execução do processo. O símbolo na forma de um paralelogramo contendo o nome do sinal Pacote é do tipo *Save* que representa uma fila para esses sinais.

A Figura 5.4 mostra um exemplo em que dois sinais do tipo Pacote chegam ao processo Transmissao sendo que ambos precisam ser transmitidos à estação remota. A Figura 5.4(a) representa a fila do processo Transmissao logo após a chegada dos dois sinais. Como a máquina de estados está no estado *Idle*, um dos sinais do tipo Pacote (o primeiro a chegar) é consumido, sendo efetuada a transição e resultando na fila mostrada na

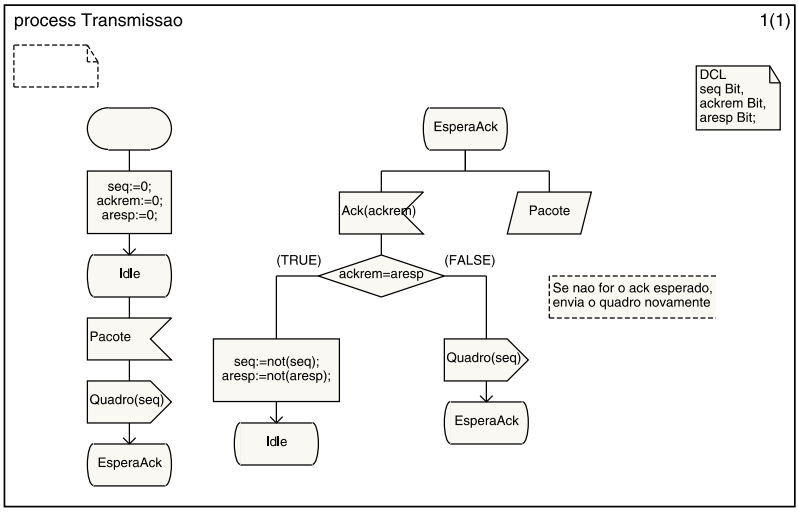


Fig. 5.3: Máquina de estados do processo Transmissao.

Figura 5.4(b). O estado *EsperaAck* é alcançado. Supondo que, pouco antes desse estado ser alcançado, o sinal Ack chega ao processo, tem-se então a fila mostrada na Figura 5.4(c). A transição do estado *EsperaAck* é realizada com o consumo do sinal Ack; porém, o sinal que está no topo da fila é o sinal Pacote. Este último é armazenado através do procedimento *Save*. Isto impede o ***consumo implícito de sinal***, isto é, que uma transição seja realizada sem o consumo de um sinal, o que caracterizaria uma falha no protocolo. Com o uso do *Save*, o sinal Ack será consumido e a transição será realizada, resultando na fila representada pela Figura 5.4(d). Supondo que não haja problemas com a comparação do valor *ackrem*, o estado *Idle* é alcançado, e o sinal Pacote, ainda no topo da fila, é consumido conforme o previsto.

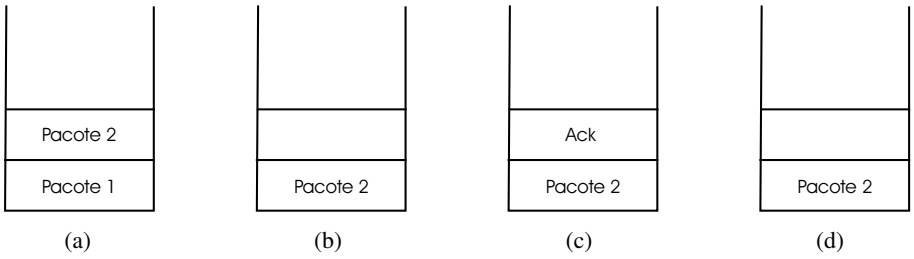


Fig. 5.4: Estados da fila de sinais do exemplo. (a) Após a recepção de dois sinais Pacote; (b) Após o consumo do sinal que estava no topo da fila; (c) Fila após o recebimento do sinal Ack; (d) Fila após o consumo do sinal Ack e antes do consumo do sinal Pacote.

A máquina de estados finitos do processo Recepcao é mostrada na Figura 5.5. A variável *sresp* (número de seqüência remoto esperado) é inicializada, sendo alcançado o estado *Idle*. Quando o sinal Quadro chega ao processo, o sinal é consumido e a transição é efetuada. Se o número de seqüência não for o esperado, o Quadro é descartado e o processo retorna ao estado *Idle*. Caso contrário, os Dados recebidos são enviados à camada de rede, o sinal Ack é enviado, o valor de *sresp* é alterado e o processo retorna ao estado *Idle*.

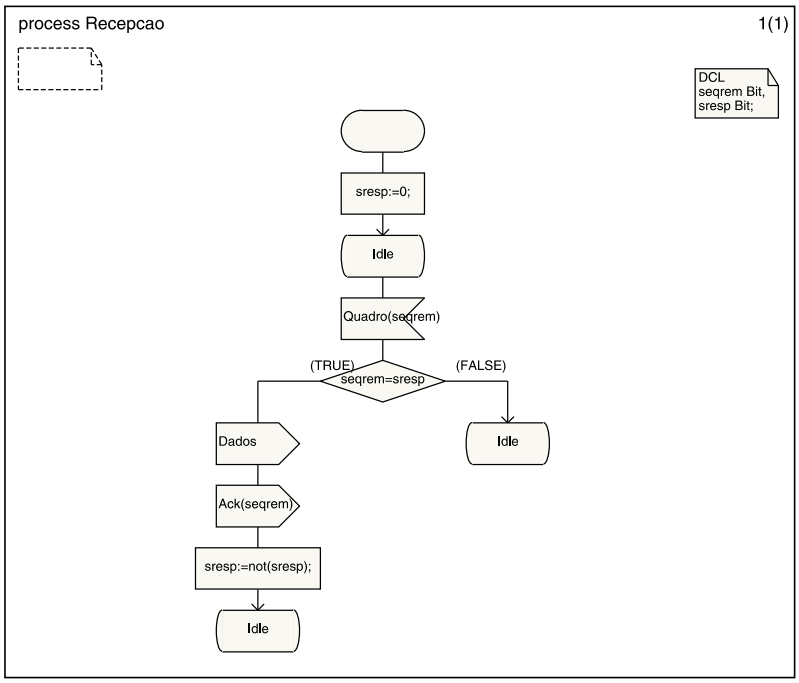


Fig. 5.5: Máquina de estados do processo Recepcao.

5.2.2 Ferramenta de Simulação e Validação

O aplicativo Simulator UI é utilizado para que sejam simulados casos específicos do modelo. O usuário (ambiente do sistema) gera e recebe os sinais tratados pelo ambiente. A visualização dos sinais pode ser feita através da observação da máquina de estados, sendo explicitado o comportamento do sistema símbolo a símbolo; ou via Diagramas de Seqüências (*Message Sequence Charts* – MSC) que mostram os sinais trocados entre os processos.

O Diagrama de Seqüências gerado pela modelagem do protocolo *simplex stop-and-wait*, quando a estação local envia um pacote à estação remota, é mostrado na Figura 5.6. O processo env_0 é o ambiente Environment, controlado pelo usuário. O usuário envia o sinal Pacote ao processo Transmissao do bloco Local. Este envia um Quadro (número de seqüência 0) para o processo Recepcao do bloco Remoto. Ao receber o sinal, o bloco Remoto gera o sinal Dados, enviado a sua camada de rede (ambiente) e o sinal Ack, enviado ao processo Transmissao do bloco Local. Em seguida, ambos os processos voltam ao estado *Idle*, indicando o fim da comunicação com sucesso.

O aplicativo de validação (Validator UI) é usado para detectar virtualmente possíveis inconsistências no modelo que não foram detectadas na simulação, como por exemplo *deadlocks* e sinais consumidos de forma implícita.

O Validator UI possui cinco algoritmos: *Bit-State*, *Random Walk*, *Tree Walk*, *Exhaustive* e *Tree Search*. Os algoritmos diferenciam-se na forma de percorrer os símbolos, possuindo dois parâmetros em comum: *Time-out*, que indica quantos minutos de validação serão executados, e *Depth*, que indica a profundidade máxima da árvore de símbolos a ser percorrida. Os relatórios contêm as amostras da simulação e os possíveis erros

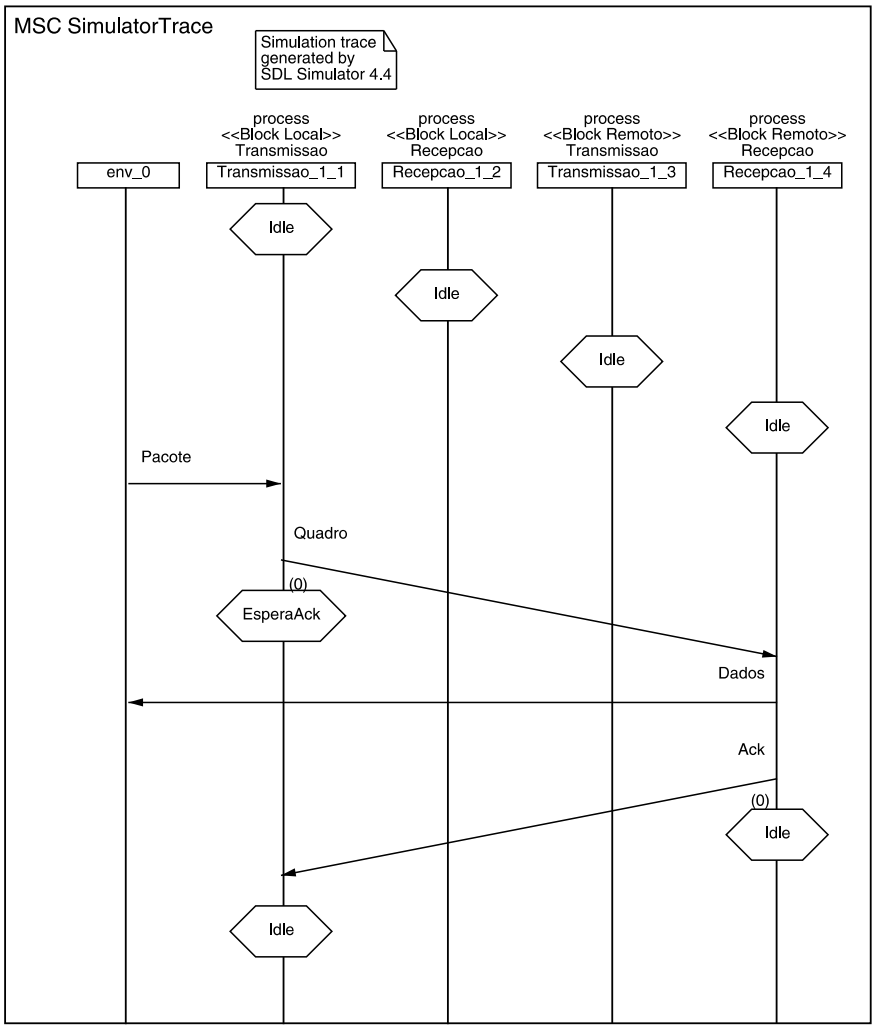


Fig. 5.6: Diagrama de Seqüências da comunicação entre duas estações no modelo *simplex stop-and-wait*.

encontrados no modelo.

O SDL é uma linguagem bastante útil na simulação e validação de sistemas e protocolos, de forma relativamente rápida e concisa, antes de sua implementação que permite:

- Prover um conjunto de conceitos bem definidos para a construção do modelo;
- Permitir uma especificação clara, concisa e sem ambigüidades;
- Habilitar uma análise da especificação para verificar a completeza e corretude;
- Fornecer uma visão clara de todo o sistema complexo a ser modelado;
- Possibilitar o uso de ferramentas computacionais para criar, manter, analisar e simular as especificações;
- Suportar a geração de aplicações sem a necessidade da fase tradicional de código.

5.3 Especificação do Modelo

Nesta seção é apresentado o modelo de validação do protocolo de roteamento MCOR através da linguagem SDL. Para tanto, uma especificação simplificada do protocolo CSMA/CA também foi necessária para uma modelagem do acesso de contenção ao meio. Utilizaram-se os parâmetros de configuração do DSSS, apresentados no Apêndice A. Uma visão completa do sistema é esboçada na Figura 5.7. Os blocos ou processos não comentados estão disponíveis para consulta em [65].

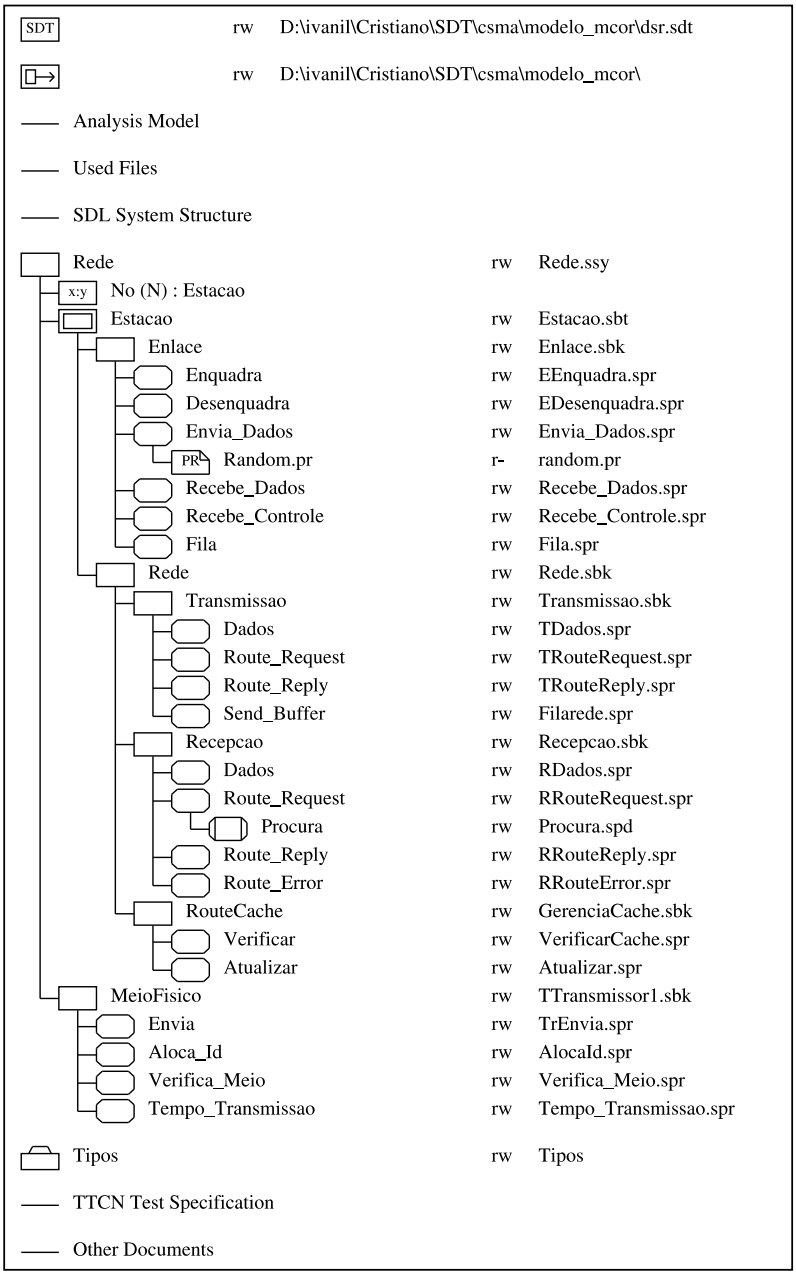


Fig. 5.7: Visão geral do sistema através do Organizer.

5.3.1 Protocolo de Roteamento

O nível do sistema para a modelagem do MCOR é exibido na Figura 5.8. O sistema contém dois blocos principais: o bloco No, que representa cada estação da rede, com N instâncias do tipo Estacao; e o bloco MeioFisico, que descreve o meio de transmissão dos pacotes. Estes são descritos a seguir. O ambiente representa a camada de transporte, atuando no envio e recebimento de segmentos. Também através do ambiente é possível determinar, através do sinal Posicao, a posição de cada estação na rede.

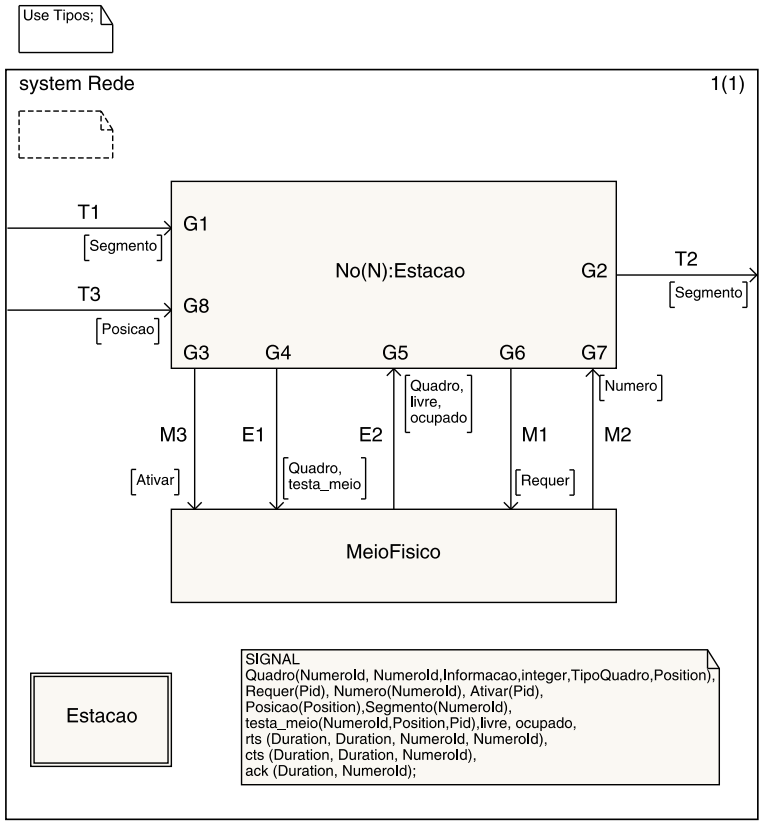


Fig. 5.8: Nível do sistema para a modelagem do protocolo MCOR.

A. Bloco No

O bloco No, mostrado na Figura 5.9, contém os blocos Rede e Enlace, que implementam, respectivamente, as camadas de rede e enlace de cada estação na rede sem fio. No modelo, a camada de rede utiliza o protocolo MCOR, enquanto a camada de enlace utiliza uma versão simplificada do protocolo CSMA/CA para controle de acesso ao meio.

O bloco Rede, Figura 5.10, subdivide-se em três blocos – Transmissao, que implementa o processo de transmissão dos pacotes; Recepcao, responsável pela recepção dos pacotes de dados na estação destino e Route-Cache, que gerencia o *Route Cache* de cada estação móvel.

As Figuras 5.11 e 5.12 expõem os blocos Transmissao e Recepcao, respectivamente. O bloco Transmissao

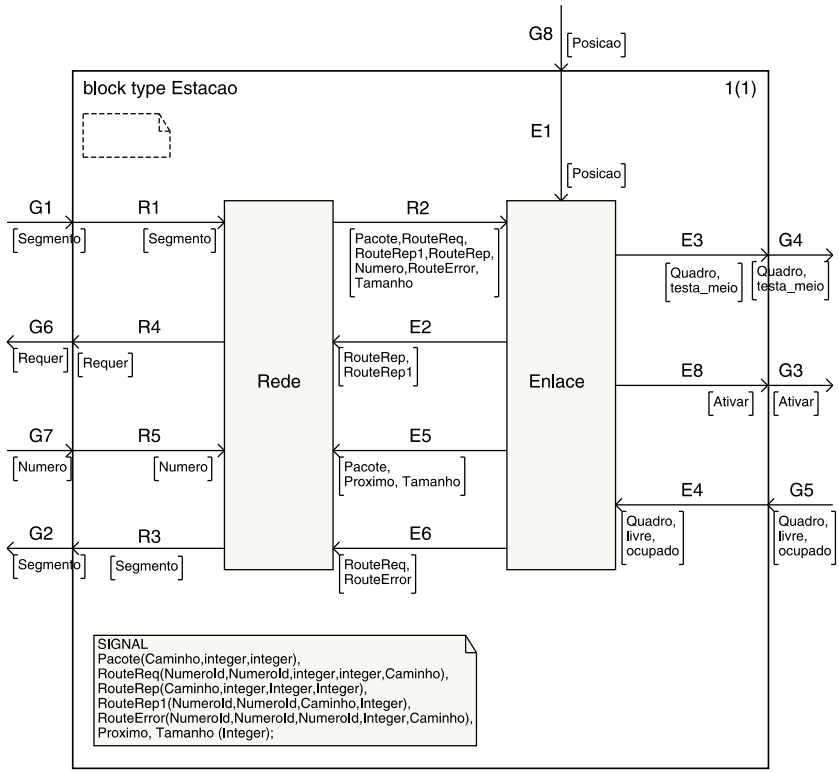


Fig. 5.9: Estrutura interna do bloco No.

possui três processos designados ao envio do pacote de dados, e às operações de ROUTE REQUEST e ROUTE REPLY[1]. Além disso, existe o processo Send_Buffer que implementa a fila do *Send Buffer*, contendo os pacotes ainda não transmitidos devido a inexistência de uma rota para o destino. O bloco Recepcao possui os mesmos três processos mencionados para o bloco Transmissao, mais o processo que trata o recebimento do ROUTE ERROR.

Antes de realizar a Descoberta de Rota, o nó origem consulta seu *cache* para averiguar se existe alguma rota até o nó destino. A estação deve também incluir a rota contida em um ROUTE REPLY recebido ou retirar o enlace inativo caso receba um ROUTE ERROR. Tais atividades são desempenhadas pelo bloco RouteCache, mostrado na Figura 5.13. Este bloco é composto por dois processos, Verificar e Atualizar, os quais executam as funções mencionadas, respectivamente. O bloco Enlace implementa o protocolo de acesso ao meio, sendo discutido na próxima seção.

B. Bloco MeioFisico

A Figura 5.14 exibe o bloco MeioFisico, que atua como uma entidade intermediária para os processos de envio e recebimento de pacotes entre as estações. Este bloco contém as informações sobre as estações que estão transmitindo em um determinado momento. O processo Envia inicializa os parâmetros de cada quadro recebido, como origem, destino, tamanho etc. O processo Tempo_Transmissao computa o intervalo de tempo

[1] O envio do ROUTE ERROR é tratado no processo Envia_Dados.

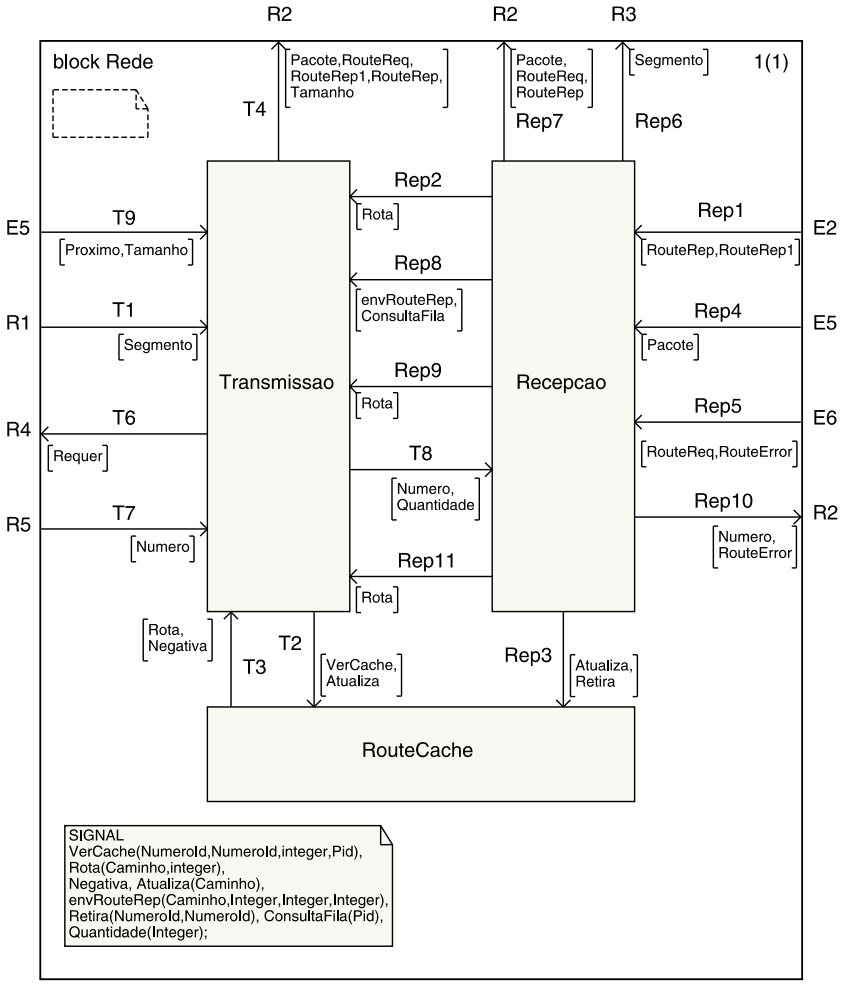


Fig. 5.10: Estrutura interna do bloco Rede.

necessário para que o quadro seja transmitido. Finalmente, o processo Verifica_Meio informa o estado do canal (livre ou ocupado) à estação requisitante.

No processo de inicialização do modelo da rede sem fio, um número identificador único é alocado a cada estação. Esse identificador é determinado pelo processo Aloca_Id. A estação transmite o sinal Requer a este processo, que retorna um número de identificação único (sinal Numero), utilizado pela estação durante todo seu período ativo.

5.3.2 Protocolo de Controle de Acesso ao Meio

Sob ponto de vista do Organizer (Figura 5.7), o protocolo CSMA/CA encontra-se implementado no bloco Enlace, exibido na Figura 5.15, mais especificamente nos processos Envia_Dados e Recebe_Dados. Dois processos auxiliares, Verifica_Meio e Recebe_Controle, são utilizados.

Os dados enviados pelo bloco Rede chegam ao processo Enquadra, que encapsula o pacote e o envia para a fila da interface de transmissão da estação, implementada pelo processo Fila. Caso o tamanho da fila atinja um

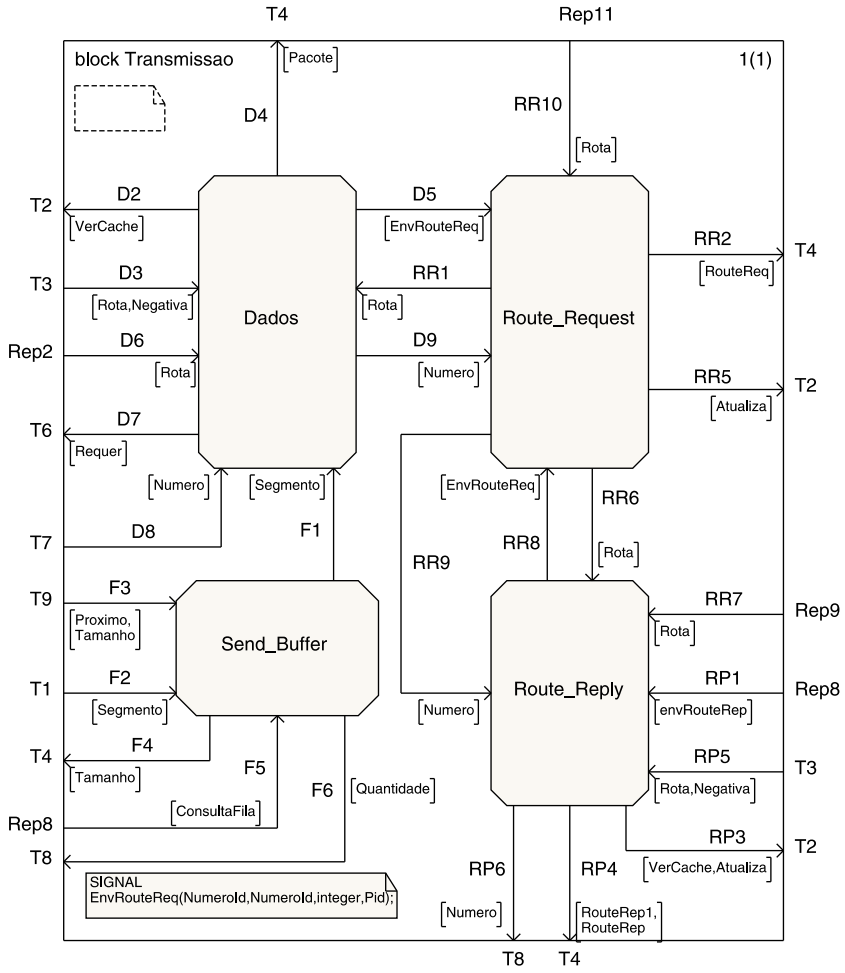


Fig. 5.11: Estrutura interna do bloco Transmissao.

valor máximo, o nó intermediário descarta o ROUTE REQUEST recebido. Observe que o *buffer* contém tanto pacotes de controle de roteamento, como pacote de dados.

Os processos Envia_Dados e Recebe_Dados são responsáveis pela especificação do protocolo CSMA/CA propriamente dito, através do *handshaking* de quatro vias. Como discutido no Apêndice A, este mecanismo reduz a probabilidade de ocorrer uma colisão através do envio dos pacotes de controle RTS/CTS e ajuste do NAV das estações não participantes da transmissão. O processo Recebe_Controle implementa um filtro, que separa os pacotes de dados e controle, sendo repassados aos demais processos do bloco sistematicamente. O processo Desenquadra desencapsula o pacote e o envia para a camada de rede, representado pelo bloco Rede (Figura 5.10).

5.3.3 Comentários

Nesta seção, são discutidas algumas observações e comentários gerais nas especificações dos protocolos MCOR e CSMA/CA.

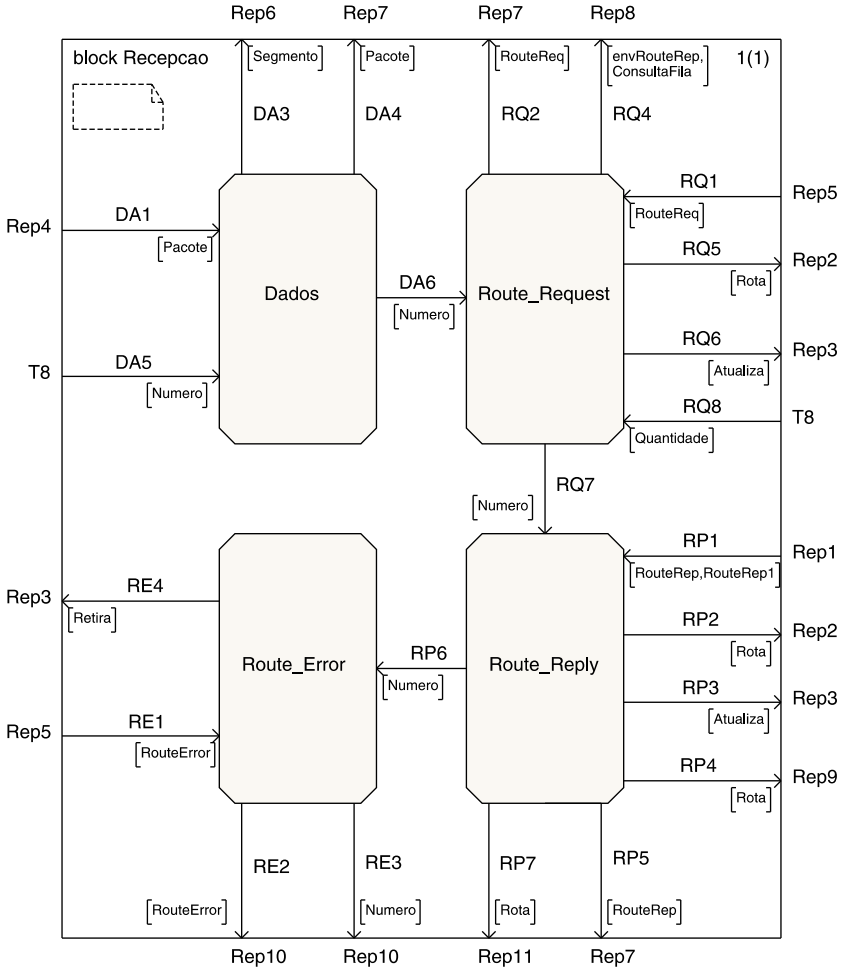


Fig. 5.12: Estrutura interna do bloco Recepcao.

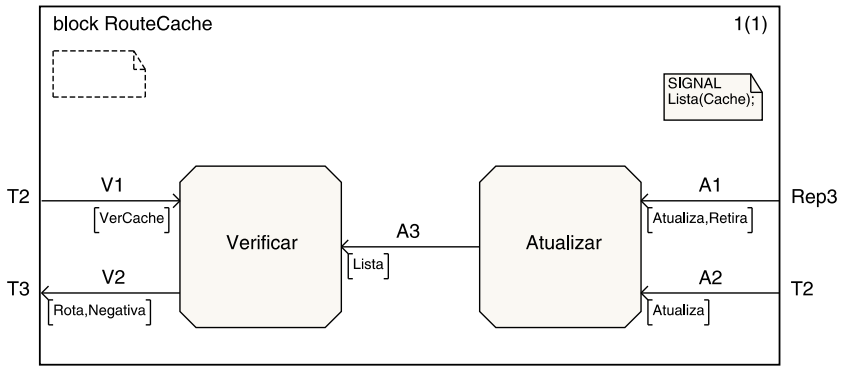


Fig. 5.13: Estrutura interna do bloco RouteCache.

Para descrever o comportamento característico de uma rede ad hoc sem fio, vários processos foram especificados cujas métricas são mostradas na Tabela 5.1.

Para o protocolo de roteamento MCOR, a especificação SDL modelou seus mecanismos e procedimentos

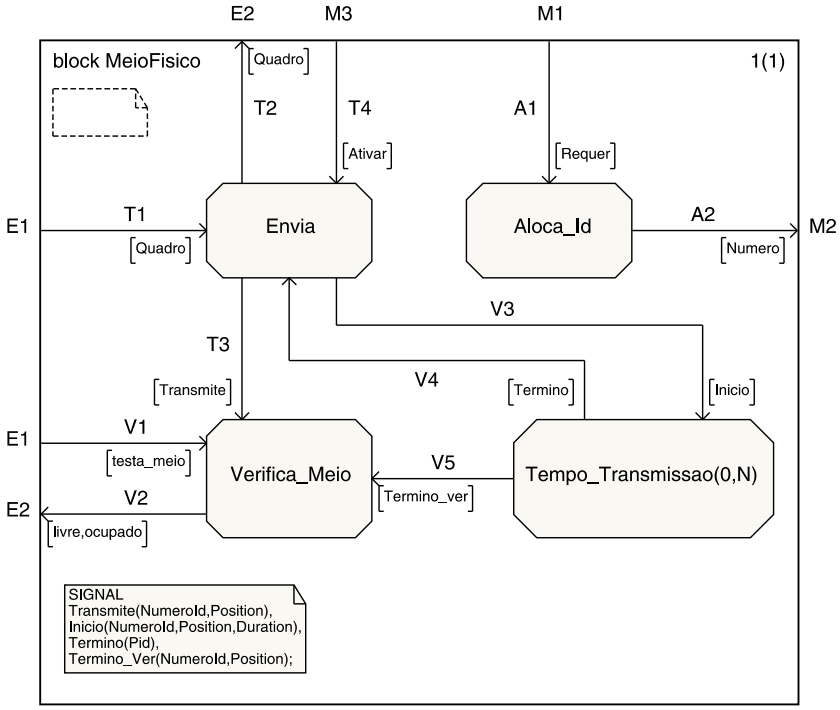


Fig. 5.14: Estrutura interna do bloco MeioFisico.

Dados Estatísticos	
Linhas	1896
Blocos	7
Processos	20
Estados	53
Sinais	34
Temporizadores	7

Tab. 5.1: Alguns dados estatísticos da especificação em SDL do protocolo MCOR.

peculiares, discutidos no Capítulo 2. Os processos básicos de Descoberta e Manutenção de Rota, e estruturas de dados, como *Send Buffer* e *Route Cache*, foram discutidos. Algumas melhorias (comentadas na Seção 2.2.4), como *salvaging*, prevenção de tempestades de ROUTE REPLYs e estado de fluxo, não foram implementadas no modelo.

A descrição SDL da camada de acesso ao meio implementou uma série de simplificações no protocolo CSMA/CA, já que o foco principal da modelagem foi o protocolo de roteamento. Na especificação, por exemplo, admite-se que no canal não há a ocorrência de colisões. Assim, algumas modificações seriam necessárias no processo Envia_Dados a fim de tratar a possibilidade de duas ou mais estações escolherem o mesmo *slot* de tempo para a transmissão. Além disso, a modelagem considerou o alcance de interferência igual ao raio de transmissão. No protocolo IEEE 802.11 em especial, o valor do primeiro é cerca de duas vezes maior que o último.

A validação através da descrição SDL apresenta muitas vantagens, como a eliminação de erros e ambigüi-

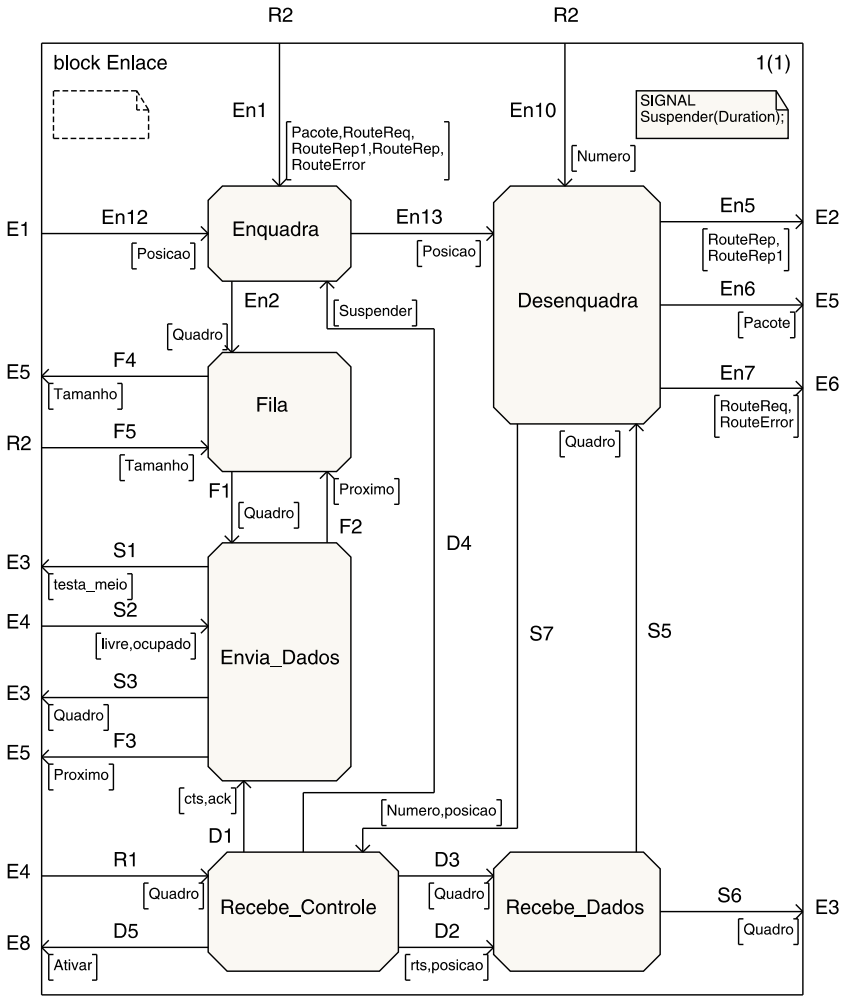


Fig. 5.15: Estrutura interna do bloco Enlace.

dades no projeto, além da descrição de conceitos consistentes e bem definidos do protocolo. Este modelo formal também é bem adaptado às redes ad hoc sem fio, já que os nós podem ser adicionados e eliminados dinamicamente, propiciando, desta forma, um ambiente para geração de testes para a provisão de requerimentos da rede.

5.4 Simulação do Modelo

Uma das vantagens mais significantes do SDL é a possibilidade do sistema ser simulado facilmente nas fases de projeto. Conseqüentemente, erros e ambigüidades podem ser detectados e corrigidos antes da fase de implementação, aumentando a qualidade do modelo e diminuindo o tempo e custo do projeto consideravelmente.

Durante a simulação do sistema, o modelo formal do SDL pode ser convertido automaticamente em código fonte C, que pode ser utilizado em uma aplicação executável. A funcionalidade do modelo pode ser verificada

através da análise estática, bem como através da exploração exaustiva. O simulador testa todos os caminhos de execução do sistema SDL nos quais um conjunto de regras são checados e violações como *deadlocks*, *loops* e estouro de pilha são reportadas. A qualquer momento a simulação pode ser interrompida, sendo que estados, transições, sinais e variáveis podem ser examinados.

Para a simulação do protocolo MCOR, três testes de cenários foram realizados: o envio de um ROUTE REQUEST, o envio de um pacote de dados e o recebimento de um ROUTE ERROR. A rede ad hoc sem fio simulada compreende um conjunto de quatro estações (numeradas de 1 a 4) em configuração de cadeia, na qual cada estação está distante 200 m das outras. O raio de transmissão é de 250 m. Nas figuras a seguir, alguns processos foram suprimidos para facilitar a visualização das mensagens trocadas entre os principais processos do modelo. Os resultados com todos os detalhes podem ser consultados em [65].

A Figura 5.16 exibe o início do processo de Descoberta de Rota, em que o nó 1 (iniciador) tenta descobrir uma rota até o nó 3 (alvo). Ao receber o pacotes de dados, o nó 1 verifica em seu *Route Cache* se existe uma rota até o nó 3. A resposta negativa é enviada pelo processo Verificar. O nó 1 então envia um ROUTE REQUEST a todos seus vizinhos com limite de propagação de um *hop*; o pedido é transferido ao processo Enquadra do bloco Enlace.

Ao receber o ROUTE REPLY contendo a rota com destino ao nó 3, o nó 1 atualiza seu *Route Cache* através do processo Atualizar. A partir deste momento, o nó origem 1 é capaz de enviar o pacote de dados até o nó destino 3. O pacote, que estava armazenado no *Send Buffer*, é então enviado pelo processo Dados do bloco Transmissao. Posteriormente, o mesmo é encapsulado em um quadro pelo processo Enquadra. O procedimento de envio do pacote é mostrado na Figura 5.17.

A Figura 5.18 ilustra o envio de um ROUTE ERROR. Um sinal Posicao, alterando a posição da estação 3, foi enviado pelo ambiente, mantendo-a fora do alcance das demais estações da rede. O nó 2, ao detectar a mudança na topologia, envia uma mensagem de ROUTE ERROR para o nó 1 contendo o enlace inativo (2, 3). As estações que receberem o ROUTE ERROR, retiram o enlace pertencente à rota de seu *cache*. Os círculos representam processos excluídos da figura.

5.5 Resumo do Capítulo

A linguagem SDL foi desenvolvida para o projeto de sistemas de telecomunicações, incluindo sistemas de comunicação de dados e tempo real. Através dela, é possível descrever a inter-relação entre o sistema e o ambiente através do refinamento da estrutura interna do primeiro, contribuindo para confirmar a consistência e a corretude do modelo. Além disso, o SDL permite avaliar soluções alternativas do sistema através dos procedimentos de validação e simulação, visto que a obtenção de tais soluções é usualmente infactível através das linguagens de programação mais comuns devido ao alto custo e tempo envolvidos.

O objetivo deste capítulo foi a especificação dos protocolos de roteamento e controle de acesso ao meio, MCOR e CSMA/CA respectivamente, através da linguagem SDL. No protocolo de roteamento MCOR, algumas características adicionais do protocolo não foram descritas. Entretanto, a modelagem do protocolo foi validada, visto que seus mecanismos básicos foram seguidos rigorosamente pelos vários processos especificados. No protocolo CSMA/CA, dois processos principais controlavam o acesso ao canal baseado em contenção

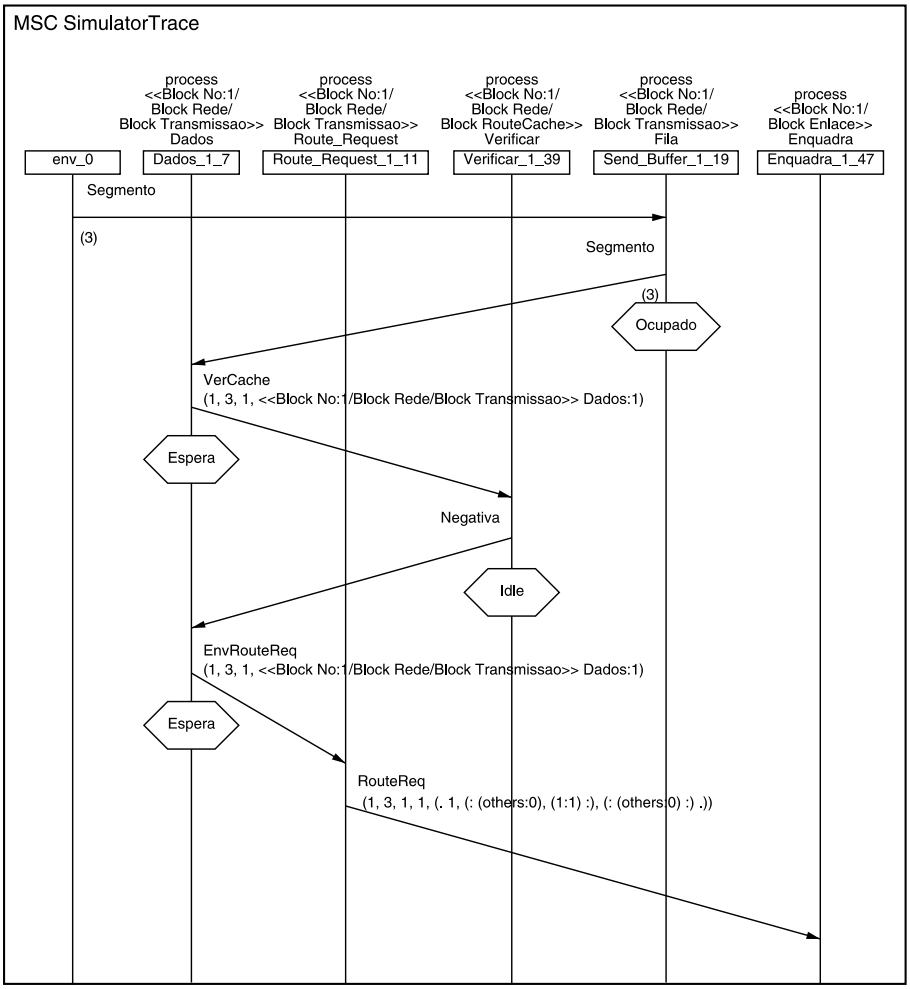


Fig. 5.16: Diagrama de Seqüências do envio de um ROUTE REQUEST.

– um representando a estação origem e o outro, a estação destino. Outros dois processos auxiliares determinavam se o meio estava livre ou não e identificavam cada tipo de pacote (controle ou dados). A principal simplificação foi com respeito à inexistência de colisões.

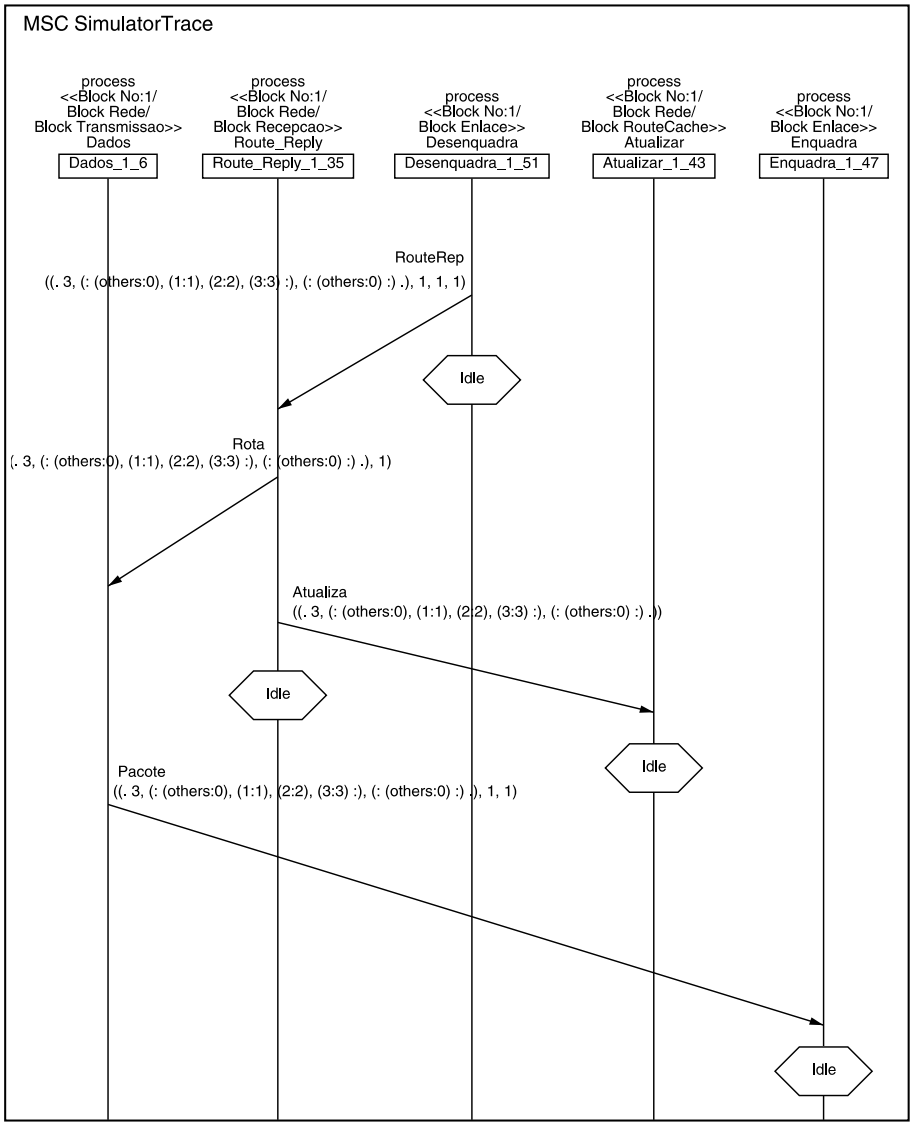


Fig. 5.17: Diagrama de Seqüências do envio de um pacote de dados.

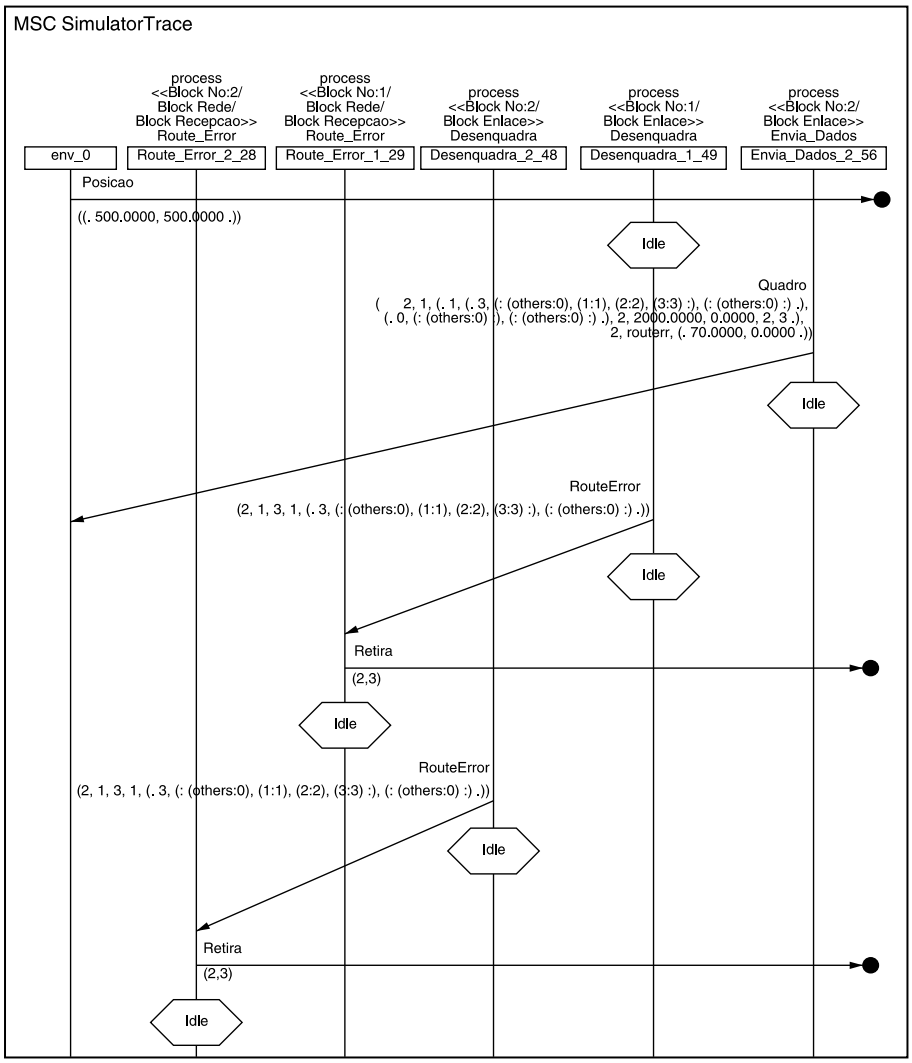


Fig. 5.18: Diagrama de Seqüências do recebimento de um ROUTE ERROR.

Capítulo 6

Conclusões

They agreed that Graham should set the test for Charles Mabledene. It was neither more nor less than that Dragon should get Stern's code. If he had the 'in' at Utting which he claimed to have this should be possible, only loyalty to Moscow Centre would prevent it. If he got the key to the code he would prove his loyalty to London Central beyond a doubt.

— *Talking to Strange Men,* Ruth Rendell

6.1 Contribuições da Tese

Devido às suas características intrínsecas, como fácil instalação e flexibilidade, o projeto de redes ad hoc é uma maneira rápida e conveniente de interconectar dispositivos sem fio utilizando poucos pontos de acesso WLAN e gradativamente menos potência de transmissão. Campos de pesquisa em redes sem fio prevêem que, futuramente, os produtos que aplicam conceitos ad hoc caminhem em direção a permitir que dispositivos próximos compartilhem dinamicamente a informação em alcances pequenos de transmissão e baixa potência consumida, dando origem às chamadas PANs (*Personal Area Networks*). Tais redes conectam dispositivos móveis transportados pelos usuários a outros dispositivos estacionários, tipicamente em distâncias até 10 metros.

Dentre as diversas áreas de pesquisa de redes ad hoc, esta tese, em mais detalhes, fez três principais contribuições nas áreas de modelo matemático, avaliação de desempenho e especificação de protocolos.

Na área de modelo matemático, foi discutida uma forma de a rede escolher, dentre os múltiplos caminhos mínimos existentes, aquele que resultaria no de menor probabilidade de colisão. Esse método torna-se mais importante para redes com tráfegos esparsos, em que algoritmo possuirá mais opções de rota para o encaminhamento dos pacotes.

Na área de avaliação de desempenho, foi demonstrada a importância da inclusão de mecanismos para o tratamento do congestionamento, que podem ser implementados nos protocolos de roteamento ad hoc, em especial o DSR. A maior contribuição desta tese foi a extensão do DSR para o balanceamento de carga da

rede, o MCOR. Foi examinado com detalhes o desempenho do DSR e MCOR sob o ponto de vista de vários cenários, com o intuito de demonstrar a consistência e a escalabilidade do MCOR em ambientes diversos.

Na área de especificação dos protocolos, utilizou-se uma linguagem de validação e simulação baseada em conceitos de orientação a objetos para a prototipação e verificação do funcionamento do MCOR. Características da linguagem, como reusabilidade e formalismo, aumentaram a qualidade e a clareza da modelagem. O procedimento de simulação não detectou erros, como perda de sinais e *deadlocks*, mostrando, portanto, a consistência do modelo descrito.

6.2 Trabalhos Futuros

Nesta seção são mencionadas algumas possibilidades de avanços e refinamentos ao longo das três contribuições da tese.

6.2.1 Modelo Matemático

O modelo matemático proposto no Capítulo 3 não faz referência a nenhum elemento da camada de acesso ao meio. No protocolo IEEE 802.11 DCF por exemplo, existem os tempos de espera dos intervalos de espaçamento, como o SIFS e o DIFS, além dos tempos de atraso de propagação do meio, transmissão do pacote de dados e dos pacotes de controle – RTS, CTS e ACK (como descrito no Apêndice A). Em [47, 66], Bianchi propôs um estudo na vazão de saturação máxima do protocolo IEEE 802.11 dividindo o problema em duas partes: 1) Obter a probabilidade de uma única estação transmitir em um *slot* de tempo aleatório através de um modelo de cadeia de Markov; 2) Expressar a vazão dos métodos básico e com RTS/CTS a partir do valor resultante da probabilidade. Em ambos métodos, o autor computou o tempo médio que o meio está ocupado diante uma transmissão com e sem (isto é, que sofreu colisão) sucesso, considerando os tempos de transferência de todas as entidades durante a transmissão do pacote. Já em [67], os autores realizaram um refinamento no modelo de Bianchi, calculando o atraso médio e a probabilidade de perda dos pacotes, levando em conta o número de tentativas para transmitir um determinado pacote. Uma extensão do modelo apresentado pode adicionar tais considerações a fim de torná-lo mais completo e preciso.

6.2.2 Avaliação de Desempenho

O trabalho mais promissor da tese seria aperfeiçoar o mecanismo de escolha de rotas do MCOR. Na Seção 2.3 foram discutidos alguns protocolos que consideram algumas informações para o balanceamento de carga no roteamento, como o número de fluxos passando em um nó, estimação do atraso médio total, número de rotas que passam em seus vizinhos etc. Trabalhos como [68, 69] fazem uma estimação dos recursos disponíveis (no caso a largura de banda) de cada nó, além de propor alguns mecanismos mais avançados de QoS para protocolos de roteamento reativos. A inclusão de tais métodos com o propósito de aumentar a eficiência e acurácia do MCOR é uma extensão natural do trabalho realizado nesta tese.

6.2.3 Especificação de Protocolos

A especificação do protocolo MCOR através do SDL, apresentado no Capítulo 5, não considera algumas das características adicionais do protocolo, tais como o *salvaging* e a extensão do estado de fluxo. Logo, um dos trabalhos futuros seria incluir tais características na especificação, com o intuito de torná-la compatível com a versão do protocolo discutida no Capítulo 2. A inclusão do tratamento da ocorrência de colisões entre os pacotes na modelagem do protocolo de controle de acesso ao meio, CSMA/CA, também o tornaria mais completo.

Referências Bibliográficas

[1] J. M. McQuillan and D. C. Walden. The ARPA network design decisions. In *Computer Networks*, volume 1, pages 243–289, August 1977.

[2] Defense Advanced Reserach Project Agency (DARPA). Available at http://www.darpa.mil.

[3] R. E. Kahn, S. A. Gronemeyer, J. Burchfiel, and R. Kunzelman. Advances in packet radio technology. *Proceedings of the IEEE*, 66(11):1468–1496, November 1978.

[4] D. B. Johnson and D. A. Maltz. Protocols for adaptive wireless and mobile networking. In *IEEE Personal Communications*, pages 34–42, February 1996.

[5] M. S. Corson, J. P. Maker, and J. H. Cernicione. Internet-based mobile ad hoc networking. In *IEEE Internet Computing*, pages 63–70, August 1999.

[6] Z. J. Haas, M. Gerla, D. B. Johnson, C. E. Perkins, M. B. Pursley, M. Steenstrup, and C.-K. Toh. Guest editorial. In *IEEE Journal on Selected Areas Communications*, number 17, pages 1329–1332, 1999.

[7] C. Bisdikian. An overview of the bluetooth wireless technology. In *IEEE Communications Magazine*, pages 86–94, December 2001.

[8] F. Bennett, D. Clarke, J. B. Evans, A. Hopper, A. Jones, and D. Leask. Piconet: Embedded mobile networking. In *IEEE Personal Communications*, pages 8–15, October 1997.

[9] The Internet Engineering Task Force (IETF). Available at http://www.ietf.org.

[10] Internet Engineering Task Force (IETF) Mobile Ad Hoc Networks (MANET) Working Group Charter. Available at http://www.ietf.org/html.charters/manet-charter.html.

[11] S. Corson. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. RFC 2501, January 1999.

[12] S. Northcutt and J. Novak. *Network Intrusion Detection – An Analyst's Handbook*. New Riders, 3th edition, 2002.

[13] R. V. Boppana and S. P. Konduru. An adaptive distance vector routing algorithm for mobile, ad hoc networks. In *Proceedings of the IEEE INFOCOM*, pages 1753–1762, Anchorage, Alaska, April 2001.

[14] S. B. Davidson. Optimism and consistency in partitioned distributed database systems. *ACM Transactions on Database Systems*, 9(3):456–481, September 1984.

[15] S. B. Davidson. Consistency in partitioned networks. *ACM Computing Surveys*, 17(3):341–370, September 1985.

[16] A. Boukerche. A performance comparison of routing protocols for ad hoc networks. In *IEEE 56th Vehicular Technology Conference - VTC 2002-Fall*, pages 1940–1946, April 2001.

[17] H. Jiang and J. J. Garcia-Luna-Aceves. Performance comparison of three routing protocols for ad hoc networks. In *Tenth International Conference on Computer Communications and Networks*, pages 547–554, October 2001.

[18] G. Mazzini F. Bertocchi, P. Bergamo and M. Zorzi. Performance comparison of routing protocols for ad hoc networks. In *IEEE Global Telecommunications Conference - GLOBECOM '03*, volume 2, pages 1033–1037, December 2003.

[19] J. Broch, D. A. Maltz, D. B. Johnson, Yih-Chun Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking - MOBICOM '98*, pages 85–97, October 1998.

[20] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of SIGCOM 94*, pages 234–244, Aug. 1994.

[21] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Networks Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., New Jersey, 1993.

[22] V. Park and S. Corson. *Temporally-Ordered Routing Algorithm (TORA) version 1: Functional Specification*. IETF Internet-Draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-tora-spec-04.txt, July 2001. Work in progress.

[23] V. Park and S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of IEEE INFOCOM'97*, March 1996.

[24] C. E. Perkins, E. M. Belding-Royer, and S. R. Das. *Ad Hoc On-Demand Distance Vector (AODV) Routing*. IETF Internet-Draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-11.txt, June 2002. Work in progress.

[25] Z. J. Haas, M. R. Pearlman, and P. Samar. *The Zone Routing Protocol (ZRP) for Ad Hoc Networks*. IETF Internet-Draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-zone-zrp-02.txt, July 2002. Work in progress.

[26] D. B. Johnson, D. A. Maltz, and Y.-C. Hu. *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)*. IETF Internet-Draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt, July 2004. Work in progress.

[27] D. B. Johnson, D. A. Maltz, and J. Brosh. DSR - The dynamic source routing protocol for multihop wireless ad hoc networks. In Charles E. Perkins, editor, *Ad Hoc Networking*, chapter 5, pages 139–172. Addison-Wesley, 2001.

[28] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In Tomasz Imielinski and Hank Korth, editors, *Mobile Computing*, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.

[29] D. B. Johnson. Routing in ad hoc networks of mobile hosts. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, pages 158–163, December 1994.

[30] J. B. Postel; editor. Transmission Control Protocol. RFC 793, September 1981.

[31] IEEE Computer Society LAN MAN Standards Committee. *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,* IEEE Std 802.11-1999. The Institute of Electrical and Electronics Engineers, New York, 1999.

[32] R. T. Braden; editor. Requirements for Internet Hosts - Communication Layers. RFC 1122, October 1989.

[33] J. Jubin and J. D. Tornow. The darpa packet radio network protocols. In *Proceedings of IEEE 75th International Conference on Distributed Computing Systems*, volume 75, pages 21–32, January 1987.

[34] S.-J. Lee and M. Gerla. Dynamic load-aware routing in ad hoc networks. In *IEEE International Conference on ICommunications - ICC'01*, volume 10, pages 3206–3210, June 2001.

[35] H. Hassanein and A. Zhou. Routing with load balancing in wireless ad hoc networks. In *ACM MSWiM, Rome, Italy*, pages 89–96, July 2001.

[36] K. Wu and J. Harms. Load-sensitive routing for mobile ad hoc networks. In *Tenth International Conference on Computer Communications and Networks*, pages 540–546, October 2001.

[37] S-T. Sheu and J. Chen. A novel delay-oriented shortest path routing protocol for mobile ad hoc networks. In *IEEE ICC'01*, pages 1930–1934, Helsink, Finland, June 2001.

[38] B. J. Hogan, M. Barry, and S. McGrath. Congestion avoidance in source routed ad hoc networks. Technical report, University of Limerky, Ireland, 2004.

[39] J.-H. Song, V. Wong, and V. C. M. Leung. Load-aware on-demand routing (LAOR) protocol for mobile ad hoc networks. In *The 57th IEEE Semiannual Vehicular Technology Conference - VTC'03*, volume 3, pages 1753–1757, April 2003.

[40] P. H. Hsiao, A. Huang, H. T. Kung, and D. Vlah. Load-balancing routing for wireless access networks. In *Proceedings of the IEEE INFOCOM*, pages 965–986, Anchorage, March 2001.

[41] B. Awerbuch, D. Holmer, and H. Rubens. High throughput route selection in multi-rate ad hoc wireless networks. Technical report, Johns Hopkins University, Baltimore, Maryland, 2004.

[42] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: A media access protocol for wireless LANs. In *Proceedings of ACM SIGCOMM'94 Conference on Commnucations Architectures, Protocols and Applications*, pages 212–225, October 1994.

[43] M. S. Gast. *802.11 Wireless Networks - The Definitive Guide*. O'Reilly, April 2002.

[44] Y.-C. Hu and D. B. Johnson. Exploiting congestion information in network and higher layer protocols in multihop wireless Ad Hoc networks. In *Proceedings of 24th International Conference on Distributed Computing Systems*, pages 301–310, March 2004.

[45] The network simulator - ns-2. Available at http://www.isi.edu/nsnam/ns.

[46] L. Kleinrock. *Queueing Systems Volume II: Computer Applications*. John Wiley & Sons, Inc., New York, 1976.

[47] G. Bianchi. IEEE 802.11-saturation throughput analysis. *IEEE Communications Letters*, 2(12):318–320, December 1998.

[48] T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedure. *Journal of Global Optimization*, (6):109–133, 1995.

[49] E. Kreyszig. *Advanced Engineering Mathematics*. John Wiley & Sons, 6th edition, 1988.

[50] K. Fall and K. Varadhan; editors. *ns Notes and Documentation*, October 2005. The VINT Project, UC Berkeley, LBNL, USC/ISI and Xerox PARC, October 2005. Available at http://www.isi.edu/nsnam/vint/index.html.

[51] The CMU Monarch Project. Computer Science Department, Carnegie Mellon University. Available at http://www.monarch.cs.cmu.edu.

[52] T. S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, New Jersey, 1996.

[53] P. Karn. MACA —a new channel access method for packet radio. In *Proceedings of the 9th Computer Networking Conference*, pages 134–140, September 1990.

[54] D. C. Plummer. An Ethernet Address Resolution Protocol: Or Converting Network Protocol Address to 48 bit. RFC 826, November 1982.

[55] B. Tuch. Development of WaveLAN, an ISM Band Wireless LAN. *AT&T Technical Journal*, 72(4):27–33, 1993.

[56] R. S. Paula and I. S. Bonatti. Implementação e simulação do protocolo MCOR no simulador ns-2. Technical report, Departamento de Telemática, Faculdade de Engenharia Elétrica e de Computação, September 2005. Available at http://www.dt.fee.unicamp.br/~ivanil.

[57] A. M. Zoubir and B. Boashash. The bootstrap and its application in signal processing. *IEEE Signal Processing Magazine*, 15(1):56–76, January 1998.

[58] T.-W. Chen. Efficient routing and quality of service support for ad hoc wireless networks. Master's thesis, University of California, Los Angeles, 1998.

[59] S. Chakrabarti and A. Mishra. Quality of service in mobile ad hoc networks. In Mohammad Ilyas, editor, *The Handbook of Ad Hoc Wireless Networks*, chapter 3, pages 1–29. CRC Press, Boca Raton, Florida, 2003.

[60] ITU-T Recommendation Z.100 Specification and Description Language (SDL), August Geneva, 2002.

[61] SDL Forum Society. Available at http://www.sdl-forum.org.

[62] ITU-T Recommendation Z.100, 2000.

[63] Telelogic TAU SDL Suite. Available at http://www.telelogic.com.

[64] A. S. Tanenbaum. *Computer Networks*. Prentice Hall, New Jersey, 4th edition, 2003.

[65] R. S. Paula, C. M. Agulhari, and I. S. Bonatti. Especificação e validação do protocolo MCOR através da linguagem SDL. Technical report, Departamento de Telemática, Faculdade de Engenharia Elétrica e de Computação, November 2005. Available at http://www.dt.fee.unicamp.br/~ivanil.

[66] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, March 2000.

[67] V. Vitsas, P. Chatzimisios, A. C. Boucouvalas, P. Raptis, K. Paparrizos, and D. Kleftouris. Enhancing performance of the IEEE 802.11 distributed coordination function via packet bursting. In *IEEE Communications Society*, pages 245–252, October 2004.

[68] C. H. P. Augusto, C. R. Cerveira, and J. F. Resende. Controle de admissão adaptativo para redes ad hoc IEEE 802.11. In *XXII Simpósio Brasileiro de Telecomunicações - SBrT'05*, pages 721–725, September 2005.

[69] S. Lohier and S.-M. Senouci. Quality of service for ad hoc networks. *Lecture Notes in Computer Science*, pages 27–41, May 2003.

[70] IEEE Computer Society LAN MAN Standards Committee. *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,* IEEE Std 802.11-1997. The Institute of Electrical and Electronics Engineers, New York, 1997.

Apêndice A

O Protocolo IEEE 802.11

What is the concept of defense: the parrying of a blow. What is its characteristic feature: awaiting the blow.

— *On War,* Carl Von Clausewitz

A.1 A Pilha de Protocolos do IEEE 802.11

Em 1997, o IEEE adotou o primeiro padrão para WLANs (*Wireless Local Networks*), denominado IEEE 802.11, à taxa de dados de 2 Mbps. Desde então, grupos de especificação (designados por letras) têm implementado extensões ao padrão IEEE 802.11.

O padrão IEEE 802.11 especifica uma camada de Controle de Acesso ao Meio (MAC) e uma camada Física (PHY) para redes locais sem fio. O uso de ondas de rádio na camada física requer relativamente uma PHY complexa. O 802.11 divide a PHY em dois componentes genéricos: o Procedimento de Convergência da Camada Física (PLCP), para mapear os quadros MAC no meio, e o sistema Dependente do Meio Físico (PMD), para transmitir os quadros. Acima da camada MAC existe a camada comum de Controle de Enlace Lógico (LLC), introduzida pelo 802.2 e utilizada por quaisquer camadas inferiores de LANs (*Local Networks*).

A camada MAC provê controle de acesso baseado em contenção sob uma variedade de camadas físicas. Três camadas físicas foram padronizadas na revisão incial do 802.11: luz infra-vermelho (IR), *Frequency-Hopping Spread-Spectrum* (FHSS) e *Direct-Sequence Spread-Spectrum* (DSSS) [70]. Ambas camadas de rádio operam na banda 2.4 GHz ISM (sem licenciamento) com taxa de dados a 1 ou 2 Mbps. Em 1999, outras duas camadas físicas baseadas em tecnologia via rádio foram desenvolvidas para atingirem larguras de bandas maiores [31]: *Orthogonal Frequency Division Multiplexing* (OFDM) designado para o IEEE 802.11a e *High-Rate Direct-Sequence Spread-Spectrum* (HR-DSSS) para o IEEE 802.11b. Elas operam em até 54 Mbps na banda 5 GHz ISM e até 11 Mbps na banda 2.4 GHz ISM, respectivamente. Em 2001, um segundo tipo de modulação OFDM foi introduzido, operando na banda 2.4 GHz ISM e com uma taxa máxima de dados de 54 Mbps. Uma visão geral do 802.11 é mostrada na Figura A.1.

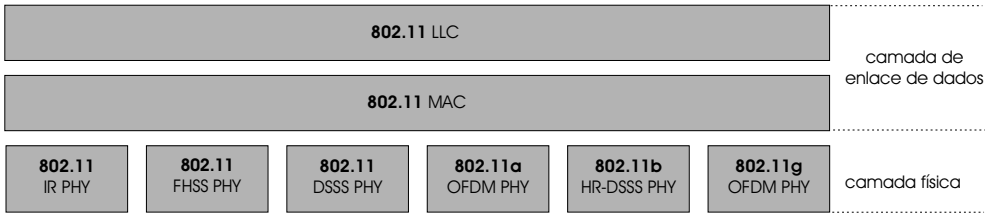


Fig. A.1: Parte da pilha de protocolos do 802.11.

O protocolo IEEE 802.11 suporta dois modos de operação para o controle de acesso, o *Distributed Coordination Function* (DCF) e o *Point Coordination Function* (PCF). O DCF é o mecanismo básico para o protocolo CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*). Como o Ethernet, ele primeiro checa se o canal está livre antes de começar a transmitir. Para evitar colisões, estações utilizam um *backoff* aleatório antes de cada quadro, com o primeiro transmissor apropriando-se do canal. Já o PCF fornece serviços livres de contenção. Estações especiais, chamadas de coordenadoras de ponto, garantem um meio sem contenção através de um sistema de *polling*. Essas estações residem em pontos de acesso, logo o PCF é restrito a redes infra-estruturadas.

A.2 Função de Coordenação Distribuída

No método de acesso básico DCF através do CSMA/CA, a estação que deseja transmitir primeiramente sente o meio; se ele estiver ocupado (isto é, se outras estações estão transmitindo), ela aguarda, caso contrário a estação transmite imediatamente.

O protocolo CSMA/CA não trata a capacidade de as estações detectarem uma colisão ouvindo sua própria transmissão. Logo, *acknowledgments* positivos são empregados para certificar a recepção com sucesso de cada pacote. O *acknowledgment* não é enviado se o pacote estiver corrompido ou for descartado por colisões. Um algoritmo CRC (*Cyclic Redundancy Check*) é utilizado para examinar a integridade dos dados. Colisões entre as estações ocorrem quando duas ou mais delas começam a transmitir no mesmo instante. Caso um *acknowledgment* não seja recebido, presumem-se que os dados foram perdidos e uma retransmissão é realizada.

A.2.1 Métodos de Operação

Dois métodos de operação são suportados no CSMA/CA: sem e com RTS/CTS. No primeiro, conhecido como *handshaking* de duas vias, a estação que deseja transmitir ouve o canal. Se estiver livre, ela começa a transmitir. A estação receptora recebendo o quadro corretamente, envia um *acknowledgment* (ACK). Se o canal estiver ocupado, a estação emissora espera até o meio ficar livre e então começa a transmitir. Se uma colisão ocorrer, ambas estações emissoras esperam um tempo aleatório segundo o algoritmo de *backoff* exponencial e tentam depois. O procedimento é descrito na Figura A.2.

O segundo método é conhecido como *handshaking* de quatro vias, mostrado na Figura A.3. Neste, ambos *physical channel sensing* (primeiro método) e *virtual channel sensing* são utilizados. O *virtual channel sensing* é provido pelo NAV (*Network Allocation Vector*), que é um temporizador que indica o período que o

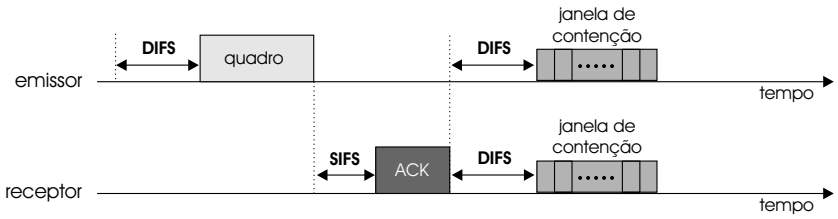


Fig. A.2: Uso apenas do *physical channel sensing* no CSMA/CA.

meio ficará reservado. As estações origem-destino ajustam seu NAV para o tempo de uso esperado do meio, incluindo quaisquer quadros necessários para completar a operação. As outras estações também ajustam seu NAV, significando que quando ele chegar a zero, o meio estará livre. Este mecanismo reduz a probabilidade de colisões diante dos cenários dos problemas de terminal escondido (*hidden station problem*) e terminal exposto (*exposed station problem*), já que qualquer estação que receber o RTS/CTS ficará silenciada durante o período de seu NAV.

Uma estação **A** que deseja transmitir algum quadro, primeiramente envia um pequeno pacote de controle chamado RTS (*Request to Send*), que inclui a fonte, o destino e o NAV, bloqueando o acesso ao meio enquanto o RTS está sendo transmitido. Todas as estações que ouvem o RTS atrasam seu acesso até o fim do NAV. A estação receptora **B** responde (se o meio estiver livre) com um pacote CTS (*Clear to Send*) que inclui a também um NAV (menor que o do RTS). A estação **A** começa a transmitir o quadro e inicia o temporizador do ACK. **B** recebendo o quadro corretamente, responde com um ACK, finalizando o processo. Caso **A** não receba o ACK dentro do período do temporizador, a estação **A** admite que houve um problema na transação e todo o processo é executado novamente. Uma vez completa a troca de RTS/CTS, a estação **A** pode transmitir seu quadro já que o canal fica reservado para sua transação.

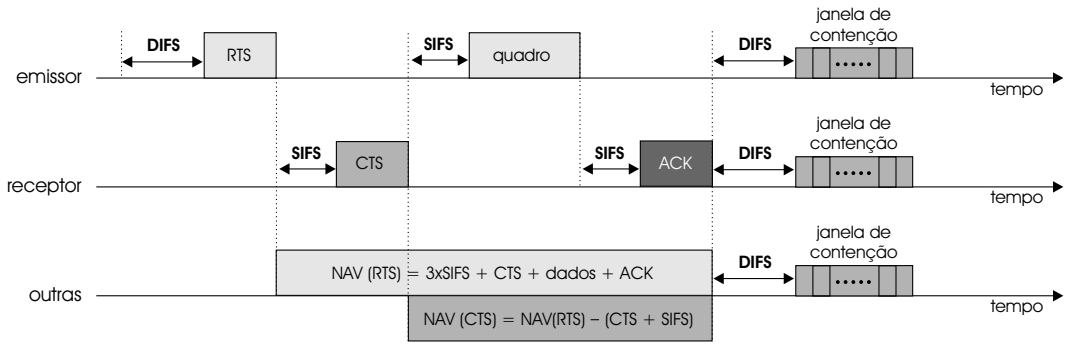


Fig. A.3: Uso do NAV para o *virtual channel sensing* no CSMA/CA.

Ambos os NAVs ajustados nos respectivos cabeçalhos dos pacotes de RTS e CTS impedem que outras estações (que os detectem) acessem o meio durante a transmissão do quadro. Depois de completada a seqüência, o canal pode ser utilizado por qualquer outra estação depois do período do DIFS.

O mecanismo de RTS/CTS é bastante efetivo em termos do desempenho do sistema, especialmente quando os pacotes são grandes e a probabilidade de colisão é alta. Colisões ainda podem ocorrer quando duas ou mais

estações transmitem no mesmo *slot*. O procedimento é controlado pelo ajuste do limiar de RTS (RTS *threshold*) presente no driver da placa 802.11. A troca do RTS/CTS é executada para quadros maiores do que o limiar. Quadros menores são enviados sem o envio prévio dos pacotes RTS/CTS.

A.2.2 Espaçamento entre os Quadros

Como o Ethernet tradicional, o espaçamento entre os quadros requer um acesso coordenado no meio de transmissão. O 802.11 utiliza quatro espaçamentos diferentes entre os quadros transmitidos, cada um com um propósito específico. Estes são descritos na Figura A.4.

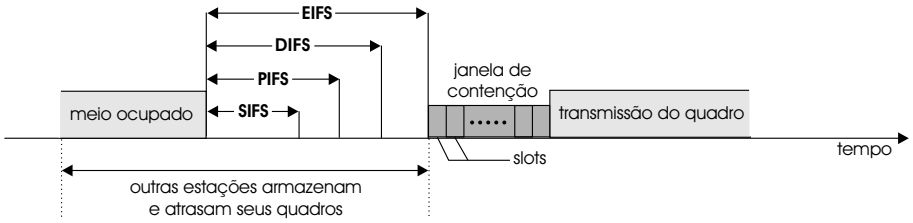


Fig. A.4: Espaçamento entre os quadros no IEEE 802.11.

Para evitar a ocorrência de colisões no CSMA/CA, as estações atrasam sua transmissão até o meio se tornar livre. Variando os espaçamentos entre os quadros, criam-se diferentes níveis de prioridade para os diversos tipos de tráfego. A lógica por detrás disto é simples: tráfegos de alta prioridade não precisam aguardar um período muito longo depois que o meio se tornar livre. Portanto, se um tráfego de nível alto estiver esperando, ele captura o canal antes mesmo que quadros de baixa prioridade tentem. Para cooperar com a interoperabilidade entre as diferentes taxas, o espaçamento entre os quadros possui um valor fixo de tempo, independente da taxa de transmissão empregada (este é apenas um dos muitos problemas causados pela existência de camadas físicas utilizando os mesmos recursos de rádio com técnicas de modulação diferentes). Camadas físicas diferentes, entretanto, podem especificar diferentes espaçamentos de tempo entre os quadros. Cada espaçamento corroborado na Figura A.4 é descrito a seguir.

- **Short InterFrame Space (SIFS)**: o SIFS é utilizado para transmissões de prioridade máxima, como quadros RTS/CTS e *acknowledgments* positivos. Logo que essas transmissões de prioridade alta se iniciam, o meio torna-se ocupado, e os quadros transmitidos depois do SIFS terão prioridade sobre aqueles transmitidos apenas em intervalos mais longos.

- **PCF InterFrame Space (PIFS)**: o PIFS (algumas vezes erroneamente chamado de *priority interframe space*) é usado pelo PCF durante a operação livre de contenção. Estações com dados a serem transmitidos no período livre de contenção podem transmitir depois do período de PIFS.

- **DCF InterFrame Space (DIFS)**: o DIFS é o tempo mínimo livre do meio para serviços baseados em contenção. Estações podem tem um acesso intermediário ao meio caso este fique livre por um período maior que o DIFS.

- **Extended InterFrame Space (EIFS)**: o EIFS é usado somente quando uma estação recebeu uma mensagem de erro durante a transmissão de um quadro.

A.2.3 Backoff Exponencial

Depois que a transmissão de um quadro foi completada e o DIFS transcorrido, estações podem tentar transmitir dados baseados em contenção. Um período denominado janela de contenção ou janela de *backoff* segue o DIFS. Esta janela é dividida em *slots*. O tempo de *slot* é o período necessário para uma estação detectar a transmissão de um quadro de outra estação. O comprimento do *slot* é dependente do meio, assim camadas físicas de alta velocidade utilizam *slot* menores. As estações selecionam um intervalo de *slot* aleatório iniciando um contador (*backoff timer*) para escalonar sua próxima tentativa de transmissão. O contador é decrementado quando o canal torna-se livre, interrompido quando uma transmissão é detectada no canal, e reativado quando o canal ficar livre novamente por um período maior que o DIFS. A estação transmite quando o contador chega a zero.

Particularmente, o DCF adota a técnica de *backoff* exponencial binária. O contador é escolhido uniformemente no intervalo [0, $W_i - 1$], sendo W o tamanho corrente da janela de contenção (CW), i o estágio do *backoff*, com $i \in \{0, 1, 2, ...m\}$ e m o número máximo de tentativas curtas. Na primeira transmissão $W_0 = CW_{min}$, sendo CW_{min} o comprimento mínimo da janela de contenção. A cada retransmissão, W_i é duplicado até que o valor máximo da janela de contenção seja atingido, $CW_{max} = W_i 2^m$. A Figura A.5 ilustra o crescimento da janela de contenção em função do número de transmissões para o DSSS, onde $CW_{min} = 32$ e $CW_{max} = 1024$. O comprimento da janela de contenção depende da camada física, mas o princípio é idêntico.

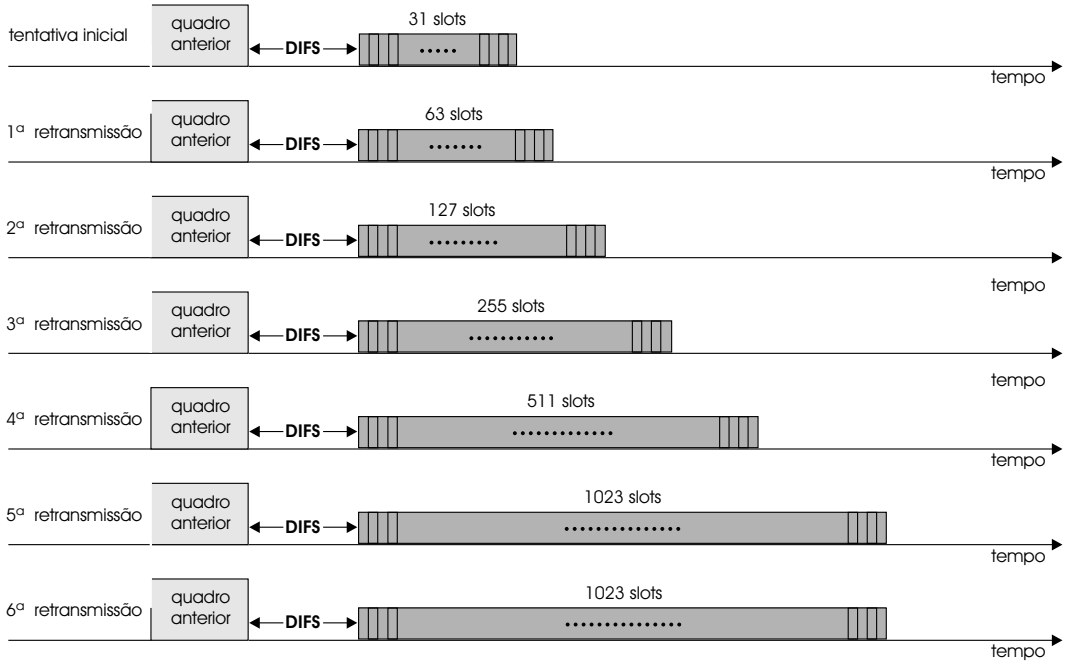


Fig. A.5: Comprimento da janela de contenção no DSSS.

Quando a janela de contenção atinge seu comprimento máximo, ela permanece constante até ser reinicializada. Permitindo janelas de contenção longas quando várias estações estão competindo para acessar o meio, mantêm-se os algoritmos MAC estáveis diante altas cargas de tráfego. A janela de contenção é reinicializada ao seu valor mínimo quando os quadros são transmitidos com sucesso, ou quando o número de tentativas é atingido, e o pacote descartado.

Na Tabela A.1 são apresentados os parâmetros utilizados nas simulações desta tese para o IEEE 802.11, utilizando a camada física DSSS [31].

Parâmetros de configuração do IEEE 802.11	
Tempo do slot	20 μs
Taxa de bits do canal	2 Mbps
Taxa de bits dos pacotes de controle	1 Mbps
Limiar de RTS	0 bytes
Cabeçalho MAC	272 bits
Cabeçalho PHY	192 bits
SIFS	10 μs
PIFS	30 μs
DIFS	50 μs
EIFS	2.492 ms
Comprimento do RTS	160 bits + cabeçalho PHY
Comprimento do CTS	112 bits + cabeçalho PHY
Comprimento do ACK	112 bits + cabeçalho PHY
Atraso de propagação	1 μs
CW_{min}	32
CW_{max}	1024
Número de tentativas curtas	7
Número de tentativas longas	4

Tab. A.1: Parâmetros padrões do IEEE 802.11 utilizando o DSSS.