UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Darlan Vivian

Ariadne-BFT: Uma Proposta de Extensão do Protocolo Ariadne Baseada na Descoberta e Seleção de Rotas do Protocolo BFTR

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação

Prof. Dr. Carlos Becker Westphall

Florianópolis, março de 2007

Ariadne-BFT: Uma Proposta de Extensão do Protocolo Ariadne Baseada na Descoberta e Seleção de Rotas do Protocolo BFTR

Darlan Vivian

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

_____

Prof. Dr. Rogério Cid Bastos
Coordenador do Programa de Pós-Graduação em Ciência da Computação

Banca Examinadora

_____

Prof. Dr. Carlos Becker Westphall
Orientador

_____

Prof. Dr. José Marcos Silva Nogueira
UFMG

_____

Prof. Dr. Mario Antônio Ribeiro Dantas
UFSC

_____

Prof. Dr. Roberto Willrich
UFSC

"*Depois de algum tempo, você aprende que heróis são pessoas que foram suficientemente corajosas para fazer o que era necessário fazer, enfrentando as consequências...*"

(William Shakespeare)

AGRADECIMENTOS

Agradeço aos meus pais Anacleto e Loreni pelo carinho, pela força, por torcerem o tempo todo por mim e por acreditarem sempre que eu conseguiria vencer mais este desafio em minha vida. Vocês me ensinaram a viver com honestidade, dignidade e determinação. Agradeço também ao meu irmão Jardel pela confiança e apoio. Esta vitória também é de vocês!

Agradeço ao meu orientador professor Carlos Becker Westphall pela ajuda e pela oportunidade de realizar este trabalho. Agradeço aos professores José Marcos, Mario Dantas e Roberto Willrich pelos valiosos comentários e sugestões.

Agradeço em especial a minha namorada Luciana, que soube ouvir meus desabafos e reclamações, que compreendeu meus momentos de silêncio e ausência, sempre com muito carinho e amor. Agradeço ao Luis, Lourdes, Louise e Liana pela ajuda de sempre e pelos momentos alegres que passamos juntos.

Agradeço fortemente a todos os meus amigos pelas festas e churrascos que fizemos juntos e principalmente pela parceria de sempre! Agradeço em especial ao Eduardo Alchieri e ao Jean Geremia, moradores da República da Corrupção. Agradeço também aos meus grandes amigos Antônio Nandi, Diego Carvalho, Ciro Damo, Diego Pacheco, Ronan Tormena, Anderson Vedoveto, Raphael Schambeck, Fernando Veronese, Diego Mazzuco, Juliano Romani, Alexandre Schulter, Marcelo Schulter, Thiago Marafon, Francisco Hillesheim, Daniel Casarotto, Ivan Salvadori, Jonas Pesente, Thiago Merege Pereira, Bruno Nakayama, Felipe Karasiak, Leonardo Ono, Fernando Ebers, André Bieluczyk, Tito Galvão e Adrian Basso. Agradeço a Susane Schmidt, Natália Scaraveli, Ana Claudia Bieluczyk, Carolina Scheidt, Francielly Lorenzetti, Francine Garghetti, Juliana Albiero, Regiane Vivian.

Agradeço os meus amigos e colegas de trabalho Marcelo Brocardo, Sérgio Roberto de Lima, Carlos Tatara, Leandro Oliveira, Leandro Aguiar, André Cardoso, Fabrício Costa e Tiago de Rolt.

Finalmente, agradeço a Deus pela minha saúde, felicidade, família e meus amigos.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE GRÁFICOS

LISTA DE SIGLAS

AODV	*Ad Hoc On-Demand Distance Vector Routing*
ARAN	*Authenticated Routing for Ad hoc Networks*
AP	*Access Point*
BFTR	*Best-effort Fault-Tolerant Routing*
BSA	*Basic Service Area*
BSS	*Basic Service Set*
AES	*Advanced Encryption Standard*
CBR	*Constant Bit Rate*
CCA	*Clear Channel Assessment*
CFP	*Contention Free Period*
CGSR	*Clusterhead Gateway Switch Routing*
CRC	*Cyclic Redundancy Code*
CSMA/CA	*Carrier Sense Multiple Access Collision Avoidance*
CSMA/CD	*Carrier Sense Multiple Access Collision Detection*
CTS	*Clear To Send*
CW	*Contention Window*
DBPSK	*Differential Binary Phase Shift Keying*
DCF	*Distributed Coordination Function*
DES	*Data Encryption Standard*
DFS	*Dynamic Frequency Selection*
DIFS	*Distributed Interframe Space*
DoS	*Denial of Service*
DQPSK	*Differential Quadrature Phase Shift Keying*
DS	*Distribution System*
DSDV	*Destination-Sequenced Distance-Vector Routing*
DSR	*Dynamic Source Routing*
DSSS	*Direct Sequence Spread Spectrum*
ESN	*Enhanced Security Network*
ESS	*Extended Service Set*
ERP	*Extended Rate*

FCC	*Federal Communications Comission*
FHSS	*Frequency Hopping Spread Spectrum*
FIFO	*First In First Out*
GFSK	*Gaussian Frequency Shift Keying*
GPS	*Global Positioning System*
HAP	*Hardware Access Point*
HCF	*Hybrid Coordination Function*
HMAC	*Hash function for Message Authentication Code*
HR/DSSS	*High-Rate Direct Sequence*
IBSS	*Independent Basic Service Set*
IEEE	*Institution of Electrical and Electronic Engineers*
IETF	*Internet Engineering Task Force*
IP	*Internet Protocol*
IR	*Infrared Rays*
ISM	*Industrial, Scientific and Medical*
ISO	*International Organization for Standardization*
ISO/IEC	*Open Systems Interconnections Basic Reference Model*
LAN	*Local Area Network*
LBT	*Listening Before Talking*
LLC	*Logic Link Control*
MAC	*Medium Access Control*
MAC	*Message Authentication Code*
MAN	*Metropolitan Area Network*
MANET	*Mobile Ad Hoc Network*
MIMO	*Multiple-In, Multiple-Out*
NAV	*Network Allocation Vector*
NS-2	*Network Simulation 2*
OFDM	*Orthogonal Frequency Division Multiplexing*
OSI	*Open Systems Interconnect*
OTcl	*Object Tool Command Language*
PCF	*Point Coordination Function*
PDA	*Personal Digital Assistants*

PHY	*Physical Layer*
PIFS	*Point Coordination Interframe Space*
PLCP	*Physical Layer Convergence Procedure*
PMD	*Physical Medium Dependent*
PPM	*Pulse Position Modulation*
QoS	*Quality of Service*
RAP	*Rushing Attack Prevention*
RDP	*Route Discovery Packet*
REP	*Route Replay*
RFC	*Reques For comments*
RED	*Random Early Detection*
RERROR	*Route Error*
RREP	*Route Reply*
RREQ	*Route Request*
RIP	*Routing Information Protocol*
RF	*Radio Frenquency*
RTS	*Request To Send*
SADSR	*Security-Aware Adaptive Dynamic Source Routing Protocol*
SAODV	*Secure Ad Hoc On-Demand Distance Vector Protocol*
SAP	*Software Access Point*
SAR	*Security-Aware Routing Protocol*
SEAD	*Secure Efficient Ad hoc Distance Vector Protocol*
SFQ	*Stichastic Fair Queue*
SIFS	*Short Interframe Space*
SLA	*Service Level Agreement*
SSID	*Service Set Identifier*
TKIP	*Temporal Key Integrity Protocol*
TORA	*Temporally Ordered Routing Algorithm*
TPC	*Transmit Power Control*
TTL	*Time to Live*
UDP	*User Datagram Protocol*
WLAN	*Wireless Local Area Network*

WLL	*Wireless Local Loop*
WMAN	*Wireless Metropolitan Area Network*
WPAN	*Wireless Personal Area Network*
WRED	*Weighted Random Early Detection*
WRP	*Wireless Routing Protocol*
WWAN	*Wireless Wide Area Network*

RESUMO

As redes sem fio *ad hoc* surgiram para estender a mobilidade e a flexibilidade no ambiente sem fio, onde um conjunto de nós, que podem agir como roteadores, formam a infra-estrutura de roteamento na rede. A maioria dos protocolos de roteamento propostos não utiliza mecanismos de segurança pois assumem que o ambiente é composto apenas por nós confiáveis. Mas devido à mudança dinâmica de topologia e ambiente aberto, uma rede *ad hoc* é extremamente vulnerável à presença de nós maliciosos os quais podem degradar desempenho ou até mesmo impedir o funcionamento da rede. Para evitar ou minimizar os problemas causados pela falta de segurança, os seguintes protocolos de roteamento seguro foram propostos: Ariadne, SAODV, BFTR, SEAD e ARAN. Neste contexto, a presente dissertação tem como objetivo analisar estes protocolos de roteamento seguro utilizados em redes sem fio *ad hoc* e propor melhorias nos mecanismos de segurança projetados para proteger a rede contra ataques interrupção de roteamento e de consumo de recursos. Como resultado desta análise, é proposto o Ariadne-BFT (Ariadne *Best-effort Fault-Tolerant*), uma extensão do protocolo de roteamento seguro Ariadne baseado nos algoritmos de descoberta e seleção de rotas do protocolo BFTR. O Ariadne-BFT incrementa a segurança no roteamento em relação ao protocolo Ariadne original, com a proteção contra o ataque *Blackhole*. Experimentos são executados sob a forma de simulações para análise e comparação de desempenho, através das quais é demonstrado que o protocolo Ariadne-BFT possui um desempenho superior em relação às métricas latência e razão de entrega de pacotes.

Palavras-chave: Redes *Ad Hoc*, Protocolo de Roteamento Seguro, Ariadne, BFTR, Ariadne-BFT, *Blackhole*

ABSTRACT

The ad hoc wireless networks appeared to extend mobility and flexibility in the wireless environment, where a group of mobile nodes cooperate by forwarding packets for each other. These networks could be rapidly deployed without the support of fixed infrastructure. Most existing designs of routing protocols not use security mechanisms assuming a trusted environment where each node in the network is cooperative and well behaved. However, in adversarial environment, misbehaving nodes always exist, and may significantly degrade the routing performance or even impeding the network operation. To avoid or minimize the problems caused by security lack, the following secure routing protocols are proposed: Ariadne, SAODV, BFTR, SEAD and ARAN. In this context, the present work has the objective to analyze these secure routing protocols used in ad hoc networks and to propose improvements in security mechanisms design to protect the networks against routing disruption and resources consumption attacks. As a result of this analysis is proposed Ariadne-BFT (Ariadne Best-effort Fault-Tolerant), an Ariadne routing protocol extension based on the route discovery and route selection algorithms of the BFTR protocol. Ariadne-BFT increases the routing security in comparision to the protocol Ariadne with the protection against the Blackhole attack. The Ariadne-BFT protocol is evaluated through extensive simulations which demonstrated that the Ariadne-BFT greatly improves the performance in latency and packet delivery ratio.

Keywords: *Ad Hoc networks*, *Secure Routing Protocol*, Ariadne, BFTR, Ariadne-BFT, *Blackhole*

1 Introdução

Nos últimos anos a comunicação sem fio tem ganhado destaque entre as tecnologias para transmissão de dados, estando cada vez mais presente em nosso cotidiano. Esta mudança é tão evidente que, hoje, é impossível imaginar um executivo de empresa que não tenha seu dispositivo móvel, seja este qual for, para se manter conectado a qualquer momento, esteja onde ele estiver.

A conectividade sem fio possibilitou uma revolução mundial no mercado de telefonia adicionando mobilidade na transmissão de voz, permitindo que as pessoas se mantenham conectadas umas as outras independentemente de suas localizações. Novas tecnologias pretendem realizar o mesmo com as redes de computadores, dentre as quais tem ganhando destaque às redes WLAN (*Wireless Local Area Network*) baseadas no padrão IEEE 802.11 [GAS05].

O IEEE 802.11, também conhecido como Wi-Fi (*Wireless-Fidelity*), desenvolve uma série de especificações para um conjunto de equipamentos utilizados no estabelecimento de uma WLAN. Os padrões atualmente homologados são [GAS05]: o 802.11b, que tem velocidade de 11 Mbps e opera na freqüência de 2,4 GHz; 802.11a, que opera na freqüência de 5 GHz e atinge velocidade de 54 Mbps; e 802.11g, que também opera em 2,4 GHz, mas tem velocidade de 54 Mbps. A definição do novo padrão 802.11n deverá impulsionar ainda mais a utilização das WLANs, pois com este novo padrão as redes poderão atingir até 540 Mpbs de taxa de transferência - o que o faz 50 vezes mais rápido que o 802.11b e 10 vezes mais rápido que o 802.11a e o 802.11g.

Em uma WLAN os usuários podem transferir arquivos, acessar e utilizar serviços da Internet com mobilidade e sem a necessidade de cabos. Isto tem revolucionado principalmente o mercado corporativo, aumentando o nível de produtividade em relação ao formato antigo, que mantinha a atuação dos profissionais restrita ao ambiente tradicional de escritório.

As WLANs podem ser classificadas em redes com infra-estrutura e redes sem infra-estrutura. Neste trabalho serão consideradas as redes sem infra-estrutura, também conhecidas como redes sem fio *ad hoc*. As redes *ad hoc* são redes formadas por um sistema autônomo de dispositivos móveis, que se comunicam entre sim sem a necessidade de uma infra-estrutura, como por exemplo um AP (*Access Point*). Desta

forma os protocolos de roteamento para redes *ad hoc* devem ser focados na mobilidade e em requisitos de segurança para garantir o bom funcionamento da rede [COR99].

As redes *ad hoc* apresentam algumas vantagens em relação às redes sem fio estruturadas tais como mobilidade, rápida instalação, alta conectividade entre os nós e tolerância a falhas devido à existência de diversas rotas entre os dispositivos móveis. Entretanto, a maioria dos protocolos de roteamento utilizados em redes *ad hoc* não utiliza mecanismos de segurança pois assumem que o ambiente é composto apenas por nós confiáveis [ARG05]. Esta suposição é particularmente perigosa em redes *ad hoc* pois o roteamento depende dos nós intermediários que formam a rota entre a fonte e o destino de um pacote. Devido à mudança dinâmica de topologia e ambiente aberto, uma rede *ad hoc* é extremamente vulnerável a presença de nós maliciosos [HUY04b]. Esses nós podem degradar o desempenho ou até impedir o funcionamento da rede através de vários ataques, os quais podem ser classificados em ataques de interrupção de roteamento e de consumo de recursos [HUY04b].

Enquanto estes ataques são possíveis em redes cabeadas, a natureza das redes sem fio aumenta seus efeitos, dificultando a detecção ou prevenção de ataques [YIS01]. Simulações realizadas em [MAR00] utilizando o protocolo DSR demonstraram que com a presença de 10% a 40% de nós maliciosos na rede, a vazão média da rede diminuiu entre 16% e 32%. Várias pesquisas tem sido realizadas com o objetivo de evitar ou ao menos minimizar os problemas causados pela falta de segurança no roteamento em redes *ad hoc*. Desta forma os seguintes protocolos de roteamento seguro foram propostos [HUY04b]: Ariadne, SAODV, BFTR, SEAD e ARAN.

Neste trabalho será tratada a questão da segurança no roteamento em redes sem fio *ad hoc*. Para isto, foram analisadas as características das redes sem fio *ad hoc*, o funcionamento dos protocolos de roteamento e os mecanismos de segurança utilizados pelos protocolos de roteamento seguro para proteger as redes contra o ataque de nós maliciosos. Como resultado, é proposto o Ariadne-BFT, uma extensão do protocolo de roteamento seguro Ariadne [HUY02a] baseado nos mecanismos de segurança utilizados no protocolo BFTR [XUE04]. O Ariadne-BFT foi implementado utilizando o simulador NS-2, possibilitando, ao final do trabalho, a execução de uma série de simulações para a coleta de métricas utilizadas na análise de desempenho.

1.1 Objetivos

Este trabalho tem como objetivo analisar a questão da segurança no roteamento em redes sem fio *ad hoc* e propor melhorias nos protocolos de roteamento que utilizam mecanismos de segurança para proteger as redes contra o ataque de nós maliciosos. Desta forma, os seguintes objetivos específicos deverão ser atingidos:

1. Compreender as características, funcionamento e variações do padrão IEEE 802.11 e analisar os aspectos da topologia das redes sem fio *ad hoc*;
2. Estudar os principais protocolos de roteamento das redes *ad hoc*, com um enfoque principal nos protocolos de roteamento seguro;
3. Identificar as falhas de segurança no roteamento e analisar os ataques realizados por nós maliciosos;
4. Identificar possíveis melhorias nos protocolos de roteamento seguro e propor modificações nos mecanismos de segurança;
5. Implementar as modificações nos mecanismos de segurança dos protocolos usando o simulador NS-2 (*Network Simulator* 2);
6. Realizar experimentos através de simulações no NS-2 coletando várias métricas para a análise do funcionamento do protocolo com as adaptações propostas e para comparação de desempenho.
7. Contribuir para a evolução dos protocolos de roteamento seguro para redes *ad hoc*, aumentando a proteção da rede sem comprometer o desempenho.

1.2 Justificativa e Motivações

Os principais fatores que motivaram a escolha do tema deste trabalho estão relacionados a segurança em redes sem fio *ad hoc* e principalmente aos protocolos de roteamento. Atualmente a segurança no roteamento é uma das áreas que tem ganhado mais destaque nas pesquisas em redes *ad hoc* [ARG05] [WUB06].

As WLANs tem se sobressaído nos últimos anos entre as tecnologias sem fio para transmissão de dados, proporcionando flexibilidade, conectividade e mobilidade de

usuários em diversos ambientes [GAS05]. Entretanto a mudança dinâmica de topologia, ambiente aberto e a falta de uma infra-estrutura, tornam uma rede sem fio *ad hoc* extremamente vulnerável à presença de nós maliciosos. Portanto é fundamental a utilização de protocolos de roteamento seguro para evitar que esses nós consigam degradar o desempenho ou até impedir o funcionamento da rede através de vários tipos de ataques [HUY04b] [ARG05].

O projeto de protocolos de roteamento seguro enfrenta alguns desafios, devido a alguns fatores tais como [COR99] [ROY99] [HUY04b]: natureza dinâmica da rede (mobilidade dos nós), constante entrada e saída de nós na rede, recursos limitados dos dispositivos (poder de processamento e energia) e efeitos da comunicação sem fio (enfraquecimento do sinal e interferência). Sendo assim, diversos centros de pesquisa mundiais possuem iniciativas na área de roteamento seguro para redes *ad hoc*, com o intuito de definir novos protocolos de roteamento seguro ou detectar limitações e aprimorar os mecanismos dos protocolos já existentes, fortalecendo cada vez mais utilização das redes *ad hoc*.

1.3 Contribuições do Trabalho

Este trabalho possui algumas contribuições na área de protocolos de roteamento seguro para redes *ad hoc*. O protocolo BFTR ainda não possui um estudo comparativo em relação aos demais protocolos de roteamento seguro. Neste trabalho, é realizada uma análise dos mecanismos de segurança do protocolo BFTR bem como dos ataques dos quais este protocolo protege a rede, em relação aos demais protocolos de roteamento seguro. Além disso, são realizadas simulações para a comparação de desempenho do BFTR em relação aos protocolos de roteamento seguro Ariadne e SEAD [VIV06b], os quais são referência em diversos trabalhos na área de roteamento seguro.

A principal contribuição é a proposta e implementação do Ariadne-BFT (Ariadne *Best-effort Fault-Tolerant*), uma extensão do protocolo de roteamento seguro Ariadne baseado nos algoritmos de descoberta e seleção de rotas do protocolo BFTR. O Ariadne-BFT incrementa a segurança no roteamento em relação ao protocolo Ariadne original, com a proteção contra o ataque *Blackhole*. Ao final do trabalho, são realizadas uma série simulações no NS-2 para análise e comparação de desempenho, através das

quais é demonstrado que o protocolo Ariadne-BFT apresenta um desempenho superior em relação às métricas latência e razão de entrega de pacotes.

1.4 Organização do Trabalho

No próximo capítulo deste trabalho, é apresentada uma visão geral sobre o funcionamento das redes sem fio, do padrão IEEE 802.11 e suas variações e apresenta os componentes que formam a arquitetura bem como as topologias das WLANs. Dentre as WLANs serão apresentadas em maiores detalhes as redes sem infra-estrutura, conhecidas como redes sem fio *ad hoc*.

No capítulo 3 são apresentados as características, propriedades e desafios da tarefa de roteamento em uma rede *ad hoc*. São demonstrados os principais protocolos de roteamento dos grupos reativo e pró-ativo, os quais são utilizados como base para os protocolos de roteamento seguro.

No capítulo 4 é abordada a questão da segurança em redes sem fio *ad hoc* relacionada aos protocolos de roteamento seguro. Serão apresentados os possíveis ataques que nós maliciosos podem executar para degradar o funcionamento do roteamento. Na seqüência, serão estudados os protocolos de roteamento seguro Ariadne, BFTR, SEAD, SAODV e ARAN em relação às suas características de funcionamento e dos mecanismos utilizados para garantir segurança no roteamento.

No capítulo 5 é apresentado um comparativo entre os protocolos de roteamento seguro Ariadne, BFTR, SEAD, SAODV e ARAN em relação a defesa oferecida por esses protocolos para proteger a rede contra os ataques apresentados capítulo anterior. São apresentadas as características do simulador NS-2 e as métricas utilizadas para a análise desses protocolos. É demonstrado o ambiente de simulação e os resultados dos experimentos realizados para a comparação de desempenho entre os protocolos reativos Ariadne e BFTR, foco principal desta dissertação

No capítulo 6 é proposto o Ariadne-BFT, uma extensão do protocolo Ariadne, baseada nos algoritmos utilizados no protocolo BFTR para descoberta e seleção das rotas. O objetivo do Ariadne-BFT é garantir uma proteção contra o ataque *Blackhole* e melhorar o desempenho do Ariadne no roteamento de pacotes.

No capítulo seguinte são realizados experimentos através de simulações para demonstrar o funcionamento do protocolo Ariadne-BFT e comparar o seu desempenho em relação ao protocolo Ariadne original. É apresentado o ambiente de simulação utilizado nos experimentos e em seguida são analisados os resultados dos experimentos através de gráficos gerados com base nos dados coletados para cada métrica.

Finalmente, no último capítulo são apresentadas as conclusões do trabalho e a perspectiva sobre trabalhos futuros.

2 Redes Sem Fio

O objetivo deste capítulo é dar uma visão geral sobre o funcionamento das redes sem fio, descrever o padrão IEEE 802.11 e suas variações e apresentar os componentes que formam a arquitetura bem como as topologias das WLANs. Dentre as WLANs serão apresentadas em maiores detalhes as redes sem infra-estrutura, conhecidas como redes *ad hoc*.

2.1 Aspectos Preliminares

Nos últimos anos a comunicação sem fio têm sido cada vez mais utilizada para a troca de informações entre dispositivos como *notebooks*, *handhelds*, *smartphones*, PDAs e telefones celulares. Estes dispositivos são utilizados nos mais variados ambientes tais como campus de universidades, aeroportos, empresas e até mesmo em florestas e campos de batalha. Por permitirem a mobilidade de seus usuários, facilitam a utilização do poder computacional, tornando transparente a disseminação da informação e a cooperação dos dispositivos na realização das mais variadas tarefas [RUB04].

Diferentes padrões e tecnologias de rede sem fio surgiram nos últimos anos para acomodar esta vasta gama de aplicações e coberturas, como por exemplo: Redes Locais sem Fio ou WLAN (*Wireless Local Area Network*), Redes Metropolitanas sem Fio ou WMAN (*Wireless Metropolitan Area Network*), Redes de Longa Distância sem Fio ou WWAN (*Wireless Wide Area Network*) e as Redes Pessoais Sem Fio ou WPAN (*Wireless Personal Area Network*).

Uma WLAN pode ser definida como uma extensão de uma rede local cabeada tradicional, que converte os pacotes de dados em ondas de rádio ou raios infravermelhos e os envia para outros equipamentos da rede ou para um AP (*Acess Point*) que pode servir como uma conexão para uma LAN (*Local Area Network*) [GAS05]. Na Figura 1 é possível visualizar o exemplo de uma WLAN típica.

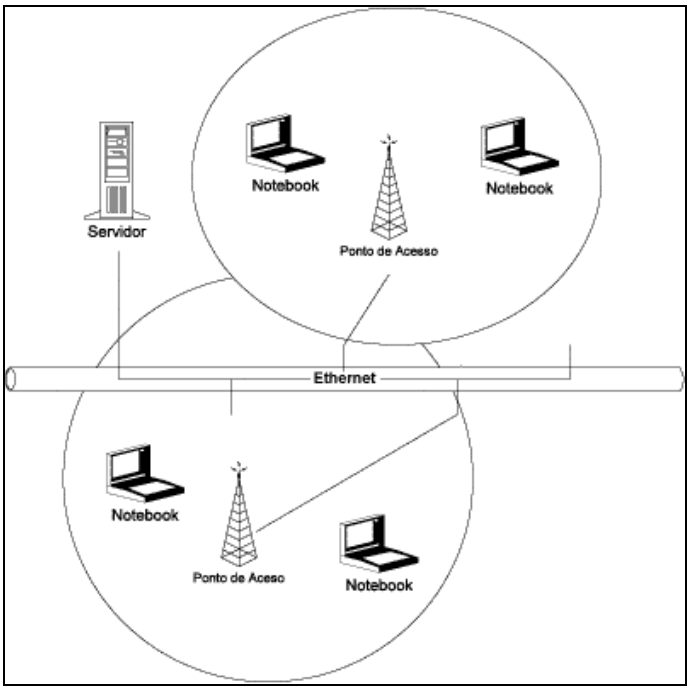


Figura 1 – Exemplo de uma WLAN típica [SIL98]

Neste ambiente típico, o ponto de acesso AP é conectado a uma rede local *Ethernet* convencional. Os pontos de acesso não apenas fornecem a comunicação com a rede convencional, como também intermedeiam o tráfego entre os dispositivos vizinhos, num esquema de micro células com *roaming* semelhante a um sistema de telefonia celular [SIL98]. As WLANs constituem-se como uma alternativa para as LANs, pois fornecem as mesmas funcionalidades, porém de forma flexível, são de fácil instalação e proporcionam uma boa conectividade.

2.2 IEEE 802.11

O comitê de padrões IEEE 802 do IEEE (*Institute of Electrical and Electronics Engineers*) desenvolve uma série de especificações para as tecnologias de LAN (*Local Area Network*). Os padrões mais amplamente usados são os da família *Ethernet* (802.3), *Token ring* (802.5) e WLAN (802.11). Na Figura 2 é possível visualizar a relação entre os vários padrões da família IEEE 802 e as camadas do modelo OSI criado pela ISO (*International Organization for Standardization*) [ISO04].

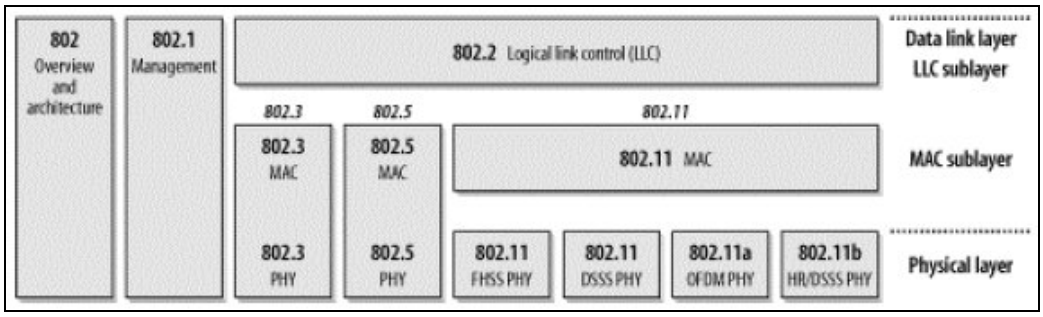


Figura 2 - Família IEEE 802 e sua relação com o modelo OSI [GAS05]

As especificações do IEEE 802 são focadas nas duas últimas camadas do modelo OSI porque ambas englobam a camada física PHY (*Physical Layer*) e a camada de acesso ao meio MAC (*Medium Access Control*), as quais estão presentes em todas as redes da família IEEE 802. A camada MAC representa um conjunto de regras que determinam como acessar o meio e enviar dados, sendo que os detalhes de transmissão e recepção são de responsabilidade da camada PHY [GAS05]. O grupo de trabalho IEEE 802.11 é o responsável pela especificação das camadas PHY e MAC para as redes locais sem fio WLANs.

A conclusão do padrão IEEE 802.11 para WLANs foi essencial para o desenvolvimento das tecnologias para redes sem fio. Com esta padronização foi possível maximizar a interoperabilidade entre diferentes tipos de WLANs como também introduzir otimizações no desempenho. Antes da adoção do padrão IEEE 802.11, a comercialização de equipamentos para redes sem fio era baseado em tecnologias proprietárias.

2.2.1 Camada PHY 802.11

Devido ao fato de que muitos dispositivos utilizam as faixas ISM (*Industrial, Scientific and Medical*) em uma determinada área, é necessário que exista uma regulamentação de utilização destas faixas para impedir que vários sinais causem interferência uns nos outros. Sendo assim, foi desenvolvida uma tecnologia que permite que a largura da banda seja compartilhada entre diversos equipamentos conhecida como espalhamento de espectro (*Spread Spectrum*).

O espalhamento de espectro utiliza funções matemáticas para difundir a força do sinal em diversas faixas de freqüências. Estas faixas são adotadas por vários países, inclusive o Brasil, denominadas internacionalmente como bandas ISM. O grupo de trabalho IEEE 802.11 utiliza as faixas de 2,4 GHz e 5,8 GHz (Figura 3) para a definição da camada PHY.

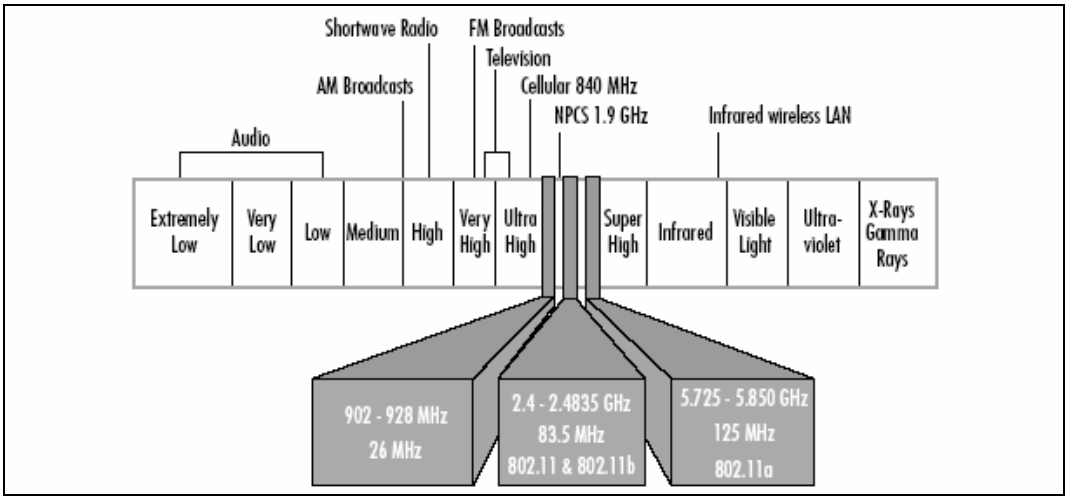


Figura 3 - Freqüências das bandas ISM [OUE02]

Inicialmente o padrão IEEE 802.11 padronizou três opções de espelhamento de espectro para a camada física [GAS05]: FHSS (*Frequency Hopping Spread Spectrum*), DSSS (*Direct Sequence Spread Spectrum*) e a IR (*Infrared Rays*). Em seguida três técnicas adicionais baseadas na tecnologia de ondas de rádio foram desenvolvidas: HR/DSSS (*High-Rate Direct Sequence*), OFDM (*Orthogonal Frequency Division Multiplexing*) e a ERP (*Extended Rate*). A escolha de determinada técnica dependerá de vários fatores relacionados com a aplicação dos usuários e o ambiente onde a WLAN irá operar.

Como apresentado na Figura 4, a camada física é dividida em duas sub camadas [GAS05]: a PLCP (*Physical Layer Convergence Procedure*) e a PMD (*Physical Medium Dependent*). A PLCP faz a ligação entre os frames da camada MAC e as transmissões por ondas de rádio através das técnicas de espelhamento de espectro. A sub camada PMD é responsável pela transmissão de todos os bits recebidos pela PLCP através de uma antena.

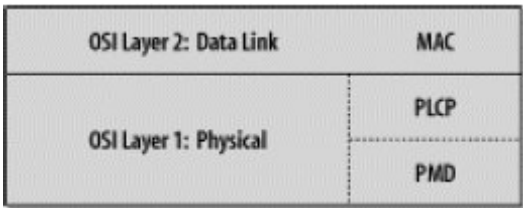


Figura 4 - Arquitetura lógica da camada física 802.11 [GAS05]

2.2.2 Camada MAC 802.11

A camada MAC (*Medium Access Control*) deve aparecer para a camada LLC (*Logic Link Control*) e superiores como qualquer outra rede 802.x, como por exemplo, uma rede *Ethernet* 802.3 [GAS05]. O mecanismo de acesso ao meio para 802.11 é o CSMA/CA (*Carrier Sense Multiple Access Collision Avoidance*). O CSMA/CA é semelhante ao CSMA/CD (*Carrier Sense Multiple Access Collision Detection*) usado no padrão 802.3 (*Ethernet*), mas com algumas diferenças essenciais.

Diferente do padrão 802.3 que envia um sinal até que uma colisão seja detectada, o CSMA/CA somente inicia uma transmissão quando nenhuma outra unidade esteja transmitindo. Embora o método de acesso CSMA/CD seja muito utilizado nas redes IEEE 802.3, não é adequado às redes 802.11, pois a colisão resultante da existência de nós escondidos em uma WLAN se torna ainda mais complexa pois geralmente é utilizada uma ligação *half-duplex*.

O controle de acesso ao meio é feito através de funções de coordenação. A camada MAC do 802.11 define as seguintes funções de acesso ao meio: DCF (*Distributed Coordination Function*), PCF (*Point Coordination Function*) e HCF (*Hybrid Coordination Function*). A arquitetura da camada MAC pode ser visualizada na Figura 5 [GAS05].

O DCF é a função básica de acesso ao meio usando o mecanismo CMSA/CA. Como em uma *ethernet*, primeiramente o meio é analisado antes de iniciar uma transmissão [CRO97]. Para evitar colisões, os *frames* são enviados em intervalos variados. Em algumas situações o DCF pode utilizar as técnicas de CTS (*Clear To Send*) e RTS (*Request To Send*) para reduzir a quantidade de colisões.

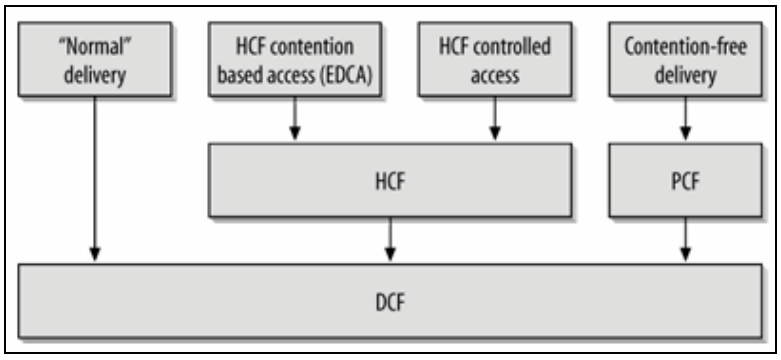


Figura 5 - Funções de coordenação da camada MAC 802.11 [GAS05]

Outro tipo de acesso da camada MAC do 802.11 é o PCF. Apesar da implementação do DCF ser obrigatório, esse não é o caso do PCF. No modo PCF, um único ponto controla o acesso ao meio, através de consulta a cada estação, proporcionando a oportunidade de transmitir sem contenção [CRO97]. O coordenador de ponto situa-se no AP e desta forma esta função é restrita as redes sem fio com infra-estrutura.

Algumas aplicações necessitam de um serviço com uma qualidade superior ao proporcionado pela utilização de um acesso simples mas não tão rigoroso quanto o PCF. A função HCF permite aos nós utilizarem múltiplas filas de serviço e realizar um balanceamento dos acessos ao meio priorizando aplicações que necessitam de um acesso com melhor qualidade [GAS05].

2.2.3 Variações do Padrão IEEE 802.11

Com os avanços das técnicas de processamento de sinais e a necessidade de melhorias no padrão 802.11, principalmente no que diz respeito à velocidade de transmissão, qualidade de serviço e segurança, vários grupos de trabalho surgiram na IEEE para aprimorar este padrão. Desta forma, variações deste padrão foram especificados, apresentando características diferentes porém com a mesma arquitetura, estimulando novas pesquisas e a padronização de produtos para as WLANs. A seguir serão apresentadas as principais variações do padrão IEEE 802.11:

1. **Padrão IEEE 802.11b**

Em setembro de 1999, o IEEE ratificou uma revisão do padrão 802.11, chamado 802.11 HR/DSSS ou 802.11b, que provê taxas mais altas de transmissão de dados, mantendo o protocolo 802.11. A arquitetura básica, características, e serviços do 802.11b são definidos pelo padrão 802.11 original sendo que a revisão da especificação afeta somente a camada física PHY, adicionando taxas mais altas de transferência de dados e conectividade mais robusta. [IEE99b].

A contribuição fundamental do 802.11b foi à adição para o padrão de WLANs da unificação da camada física que suporta duas novas velocidades, 5.5 Mbps e 11 Mbps. Para realizar isto, o DSSS teve que ser selecionado como técnica exclusiva da camada física para o padrão porque a freqüência não pode suportar velocidades mais altas sem violar a regulação atual do FCC (*Federal Communications Commission*). Desta forma, as redes baseadas no 802.11b vão operar a 1 e 2 Mbps usando DSSS, mas não irão operar a 1 e 2 Mbps usando FHSS.

802.11b WLANs apresentam troca dinâmica de taxa de transmissão, permitindo que a taxa de dados seja automaticamente ajustada para compensar interferências. Numa rede ideal, os usuários transmitem dados à taxa de 11 Mbps. Porém, quando os dispositivos são movidos além de uma certa distância ou se uma interferência significativa está presente, os dispositivos 802.11b transmitirão a velocidades mais baixas, como 5.5, 2, e 1 Mbps [OUE02].

2. **Padrão IEEE 802.11a**

O padrão IEEE 802.11a é uma das extensões de camada físicas do padrão 802.11 original. Abandonando completamente o espectro de expansão, o 802.11a usa uma técnica de codificação OFDM. O equipamento 802.11 opera a 5 GHz e suporta taxas de transmissão de até 54 Mbps [IEE99c].

Quando o IEEE concluiu os padrões para redes de comunicação sem fio 802.11a e 802.11b em 1999, sua meta era criar uma tecnologia padrão. Este padrão deveria contornar múltiplos tipos de codificação física, freqüências e aplicações do mesmo modo que o padrão *Ethernet* 802.3 foi desenvolvido com sucesso para 10 Mbps, 100 Mbps e a 1 Gbps sobre fibra ótica e vários tipos de cabos de cobre [OUE02].

A desvantagem de usar a camada MAC do 802.11b é que o padrão 802.11a herda as mesmas ineficiências presentes na implementação do 802.11b. A camada MAC do 802.11b é apenas aproximadamente 70% eficiente. Atualmente o *throughput* máximo das implementações, baseadas no 802.11b, estão entre 5.5 e 6 Mbps. De forma idêntica, o padrão 802.11a que possui limite de 54 Mbps, apresenta *throughput* máximo entre 30 a 35 Mbps considerando o *overhead* adicional causado pela camada física. Mas ao contrário do padrão 802.11b, o 802.11a não têm que transmitir seus cabeçalhos a 1 Mbps. Assim, este padrão ganha aproximadamente 5% de eficiência se comparado ao 802.11b [OUE02].

3. **Padrão IEEE 802.11g**

Neste padrão a camada física é uma extensão do IEEE 802.11b com uma taxa de transmissão de 54 Mbps usando a codificação ERP. Entretanto, para manter compatibilidade com os padrões anteriores, o padrão 802.11g utiliza algumas variações do ERP tais como: ERP-DSSS, ERP-OFDM e DSSS-OFDM. Este padrão permite a utilização mista da rede, ou seja, equipamentos que utilizam o padrão 802.11b podem compartilhar a mesma rede com os equipamentos que utilizam o 802.11g. Mas devido à existência de mecanismos de proteção para garantir esta compatibilidade, o *throughput* da rede pode diminuir em até 50% [GAS05].

4. **Padrão IEEE 802.11d**

O padrão IEEE 802.11d [IEE01] foi desenvolvido para áreas fora dos chamados cinco grandes domínios reguladores (EUA, Canadá, Europa, Japão e Austrália). O 802.11d tem um frame estendido que inclui campos com informações dos países, parâmetros de freqüência e tabelas com parâmetros.

5. **Padrão IEEE 802.11e**

Inicialmente o objetivo do padrão 802.11e [IEE05] era prover segurança e qualidade de serviço para a camada MAC. Posteriormente, somente às questões relacionada com QoS e diferenciação de serviços entre diferentes classes de tráfego para as estações de uma WLAN foram abordadas por este padrão, as quais são necessárias para suporte de voz, vídeo e dados.

6. **Padrão IEEE 802.11f**

O IEEE 802.11f [IEE03] está definindo as recomendações práticas para os serviços dos APs, as primitivas, o conjunto de funções e os protocolos que deverão ser compartilhados pelos múltiplos fornecedores para operarem em rede.

7. **Padrão IEEE 802.11i**

O padrão 802.11i [IEE04] tem o objetivo de melhorar as funções de segurança do protocolo da camada MAC, que agora é conhecido como ESN (*Enhanced Security Network*). Para isso os seguintes protocolos são avaliados: WEP (*Wired Equivalent Protocol*), TKIP (*Temporal Key Integrity Protocol*), AES (*Advanced Encryption Standard*), IEEE 802.1x para autenticação e criptografia.

O grupo de trabalho 802.11i está trabalhando na integração do AES dentro da camada MAC. O AES segue o padrão do DES (*Data Encryption Standard*). Como o DES o AES usa criptografia por blocos. Diferente do DES, o AES pode exceder as chaves de 1024 bits, reduzindo as possibilidades de ataques [GAS05].

8. **Padrão IEEE 802.11n**

O padrão IEEE 802.11n ainda deve passar por algumas revisões até sua aprovação final. Este novo padrão usará uma tecnologia chamada de MIMO (*Multiple-In, Multiple-Out*), que permitirá aos chips usar várias antenas, cada uma capaz de manipular mais de um sinal de dados ao mesmo tempo. Isso deve melhorar seu alcance e aumentar o volume de dados trafegados simultaneamente, suficiente para transmissão de vídeo sem interrupções. Além disso, os equipamentos do novo padrão devem ser compatíveis com os anteriores 802.11a, 802.11b e 802.11g, em suas respectivas velocidades. O 802.11n pode atingir, potencialmente, 540 Mpbs de taxa de transferência, sendo 50 vezes mais rápido que o 802.11b e 10 vezes mais rápido que o 802.11a e o 802.11g [GAS05].

2.3 Arquitetura 802.11

Para que uma WLAN ofereça suporte à mobilidade de estações de uma forma transparente para as camadas superiores, é necessário que exista uma interação entre

alguns componentes da rede sem fio. Estes componentes compreendem o BSS (*Basic Service Set),* IBSS (*Independent Basic Service Set*)*,* as estações, o meio sem fio, o AP (*Access Poin*), o DS (*Distribution System*) e o ESS (*Extended Service Set*) [GAS05].

As topologias BSS (Figura 6) consistem em pelo menos um AP conectado à infra-estrutura de uma rede cabeada e a um conjunto de estações sem fios que estão sob o controle direto de uma mesma função de coordenação. Esta função é quem determina quando cada nó pode enviar e receber dados utilizando o meio de transmissão sem fio. A área ocupada pelos membros de um BSS é denominada de BSA (*Basic Service Area*) [OUE02]. O AP age como o servidor lógico para uma única célula de LAN sem fio ou canal. A comunicação entre duas estações finais acontece a partir de uma estação para o AP e do AP para a outra estação.

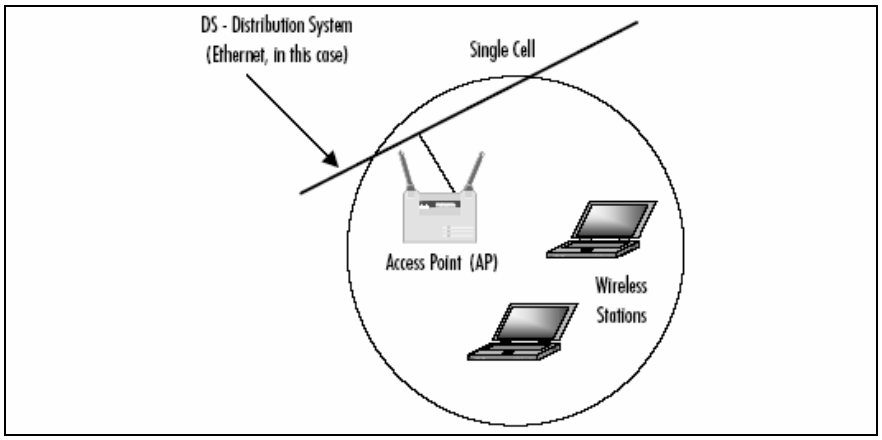


Figura 6 - Conjunto básico de serviços (BSS) [OUE02]

As IBSS (Figura 7) possuem uma configuração bastante semelhante a uma rede ponto-a-ponto de uma casa ou escritório na qual não existe a necessidade de que algum nó funcione como servidor [GAS05]. As topologias IBSS incluem várias estações sem fio que se comunicam diretamente com outras estações, sem a intervenção de um AP ou qualquer conexão com uma rede cabeada.

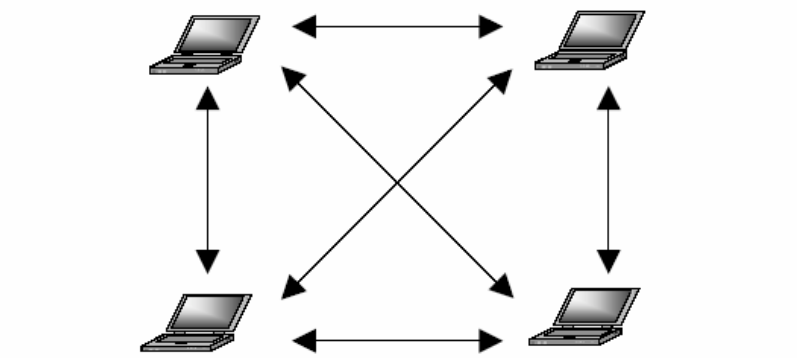


Figura 7 - Conjunto básico de serviços independente (IBSS) [OUE02]

As topologias ESS (Figura 8) consistem em um conjunto de BSS (cada um com seu AP), geralmente referenciadas como células. Estas células normalmente estão conectadas através de um DS. O ESS é visto pela camada de protocolos superior (IP) como uma simples rede 802, do mesmo modo que uma rede *Ethernet* 802.3 usando *bridge* é vista como uma simples rede 802 pelas camadas de protocolo superiores [OUE02]. O DS é necessário se, por exemplo, bancos de dados, aplicações e serviços de impressão de uma rede são acessíveis somente através de uma rede cabeada.

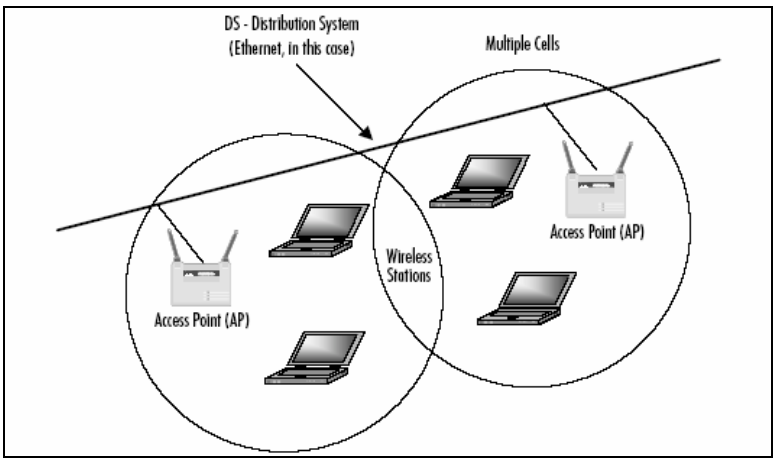


Figura 8 - Conjunto de serviços estendidos (ESS) [OUE02]

De acordo com os componentes definidos na arquitetura do padrão IEEE 802.11, as WLANs podem ser configuradas de dois modos distintos:

1. **Redes com infra-estrutura**

As redes com infra-estrutura são formadas pelas estações e pelos APs que são os responsáveis por boa parcela da funcionalidade da rede (Figura 6). Quando uma estação dentro de um BSS deseja se comunicar com outra estação, a comunicação obrigatoriamente é interceptada pelo AP, para posteriormente ser enviada à estação de destino.

Uma estação pode entrar em *roaming*, isto é, passar de uma BSS para outra através de um software e hardware que mantém uma conexão de rede fixa através do monitoramento da força do sinal proveniente do AP e procurando sempre um sinal de melhor qualidade. Normalmente isto é completamente transparente ao usuário o qual não percebe que um ponto de acesso diferente está sendo usado. Algumas configurações de ponto de acesso requerem autenticação de segurança ao trocar de ponto de acesso, normalmente na forma de requisição de uma senha [GAS05].

2. **Redes sem infra-estrutura**

Também conhecidas como redes *Ad Hoc* ou MANET (*Mobile Ad Hoc Network*) são redes formadas somente por estações móveis dentro de uma área restrita, que se comunicam sem a necessidade de um ponto de acesso AP. Na Figura 7 pode ser visualizado uma redes *ad hoc*, onde o BSS é denominado de IBSS. Uma rede *ad hoc* é, portanto, um sistema autônomo de nós móveis.

O sistema pode operar em isolamento, ou pode apresentar *gateways* para se conectar com uma rede fixa. Os nós são equipados com transmissores sem fio que utilizam antenas que podem ser unidirecionais (broadcast), altamente direcional (ponto-a-ponto), possivelmente dirigível, ou a combinação destas possibilidades [IEE99c].

2.4 Redes Sem Fio Ad Hoc

Com os recentes avanços de desempenho dos computadores e das tecnologias de comunicação sem fio, é natural o crescimento de aplicações avançadas em redes móveis. O objetivo das redes a*d hoc* é suportar operações robustas e eficientes pela adição de funcionalidades de roteamento entre os nós móveis que formam a rede. Tais

redes são previstas para serem dinâmicas, aleatórias e possuírem uma alta freqüência de variações.

Em uma rede sem fio *ad hoc*, os nós que formam esta rede cooperam entre si na transmissão de pacotes devido à limitação de transmissão individual de cada nó [BRO98]. A rota de um nó remetente para um nó destino pode requerer vários nós intermediários criando uma rota *multihop* (múltiplos saltos) do remetente até o destinatário.

Redes *ad hoc* não requerem nenhuma administração centralizada ou uma infra-estrutura de rede fixa como estações base ou pontos de acesso e pode ser rapidamente configurada de acordo com as necessidades. Desta forma, este tipo de rede de computadores pode ser usada em cenários onde não exista infra-estrutura, ou onde a infra-estrutura existente não satisfaça as exigências da aplicação por razões como custo viabilidade física. Exemplos de aplicações para redes *ad hoc* variam desde operações militares até a interação entre participantes presentes em uma reunião ou estudantes durante uma conferência.

A topologia das redes *ad hoc* pode mudar com o tempo devido à movimentação dos nós ou ajuste dos parâmetros de transmissão e recepção. Caso os nós não estejam na mesma área de cobertura do sinal, a rota entre eles pode ser formada por vários *hops* (saltos) através de um ou mais nós na rede, caracterizando esta topologia como sendo *multihop.*

A meta das redes móveis *ad hoc* é estender mobilidade no ambiente sem fio, onde um conjunto de nós, que pode ser tanto roteadores como *hosts*, formam a infra-estrutura de roteamento na rede [COR99]. Para isto existe um grupo de trabalho do IETF (*Internet Engineering Task Force*) chamado *Mobile Ad Hoc Network Working Group* que trabalha na discussão e definição de protocolos de roteamento para redes móveis *Ad Hoc*. A RFC 2501 [COR99] enumera algumas das características das redes sem fio *ad hoc* tais como:

1. **Topologia dinâmica**

Os nós são livres para se mover arbitrariamente; assim, a topologia da rede (que é tipicamente *multihop*) pode mudar fortuitamente e rapidamente em intervalos de

tempos imprevisíveis, podendo apresentar *links* tanto bidirecionais quanto unidirecionais.

2. **Limitação da Largura de Banda**

Links sem fio continuarão tendo uma capacidade significativamente mais baixa se comparados aos *links* cabeados. Além disso, o *throughput* alcançado em uma comunicação sem fio é consideravelmente prejudicado devido aos efeitos causados por acessos múltiplos, desvanecimento do sinal, ruído e condições de interferência.

3. **Limitação de energia**

Alguns ou todos os dispositivos (nós) que compõem uma MANET precisam confiar em baterias ou em outros meios limitados de energia. Sendo assim, um dos critérios mais importantes em um protocolo é a otimização do consumo de energia durante o roteamento.

4. **Limitações na segurança**

Redes sem fio móveis são geralmente mais propensas a ameaças de segurança física do que as redes com fio. Assim, o aumento da possibilidade de ocorrência de ataques de *eavesdropping* (escutar clandestinamente), *spoofing* e DoS (*Denial of Service*) devem ser considerados cuidadosamente.

2.5 Sumário

Neste capítulo foram apresentadas as características das WLANs e principalmente das redes do tipo *ad hoc* possibilitando uma melhor compreensão da camada física (PHY), da camada de acesso ao meio (MAC) e das variações do padrão IEEE 802.11. Este conhecimento facilitará o estudo dos protocolos de roteamento e das questões de segurança que envolve as redes sem fio *ad hoc*.

3 Roteamento em Redes *Ad Hoc*

Neste capítulo serão apresentados as características, propriedades e desafios da tarefa de roteamento em uma rede *ad hoc*. Praticamente todos protocolos definidos para as redes *ad hoc* seguem dois padrões distintos, identificados como protocolos reativos e pró-ativos. Desta forma serão demonstrados os principais protocolos de roteamento dos grupos reativo e pró-ativo, os quais são utilizados como base para alguns protocolos de roteamento seguro.

3.1 Aspectos Preliminares

A tarefa de roteamento em uma rede *ad hoc* é mais complexa do que em redes cabeadas, pois este depende de muitos fatores incluindo topologia, seleção de roteadores e iniciação da requisição de rota. Suas características específicas são utilizadas como heurísticas pelos algoritmos de roteamento para identificar eficientemente o caminho pelo qual o pacote deve ser enviado.

Um dos principais desafios no projeto de um protocolo para redes *ad hoc* está no fato de que um nó precisa conhecer a localização de seus nós vizinhos para determinar uma rota para os pacotes [JUB87]. Além disso, como o número de nós da rede pode ser grande, encontrar a rota para os destinatários requer uma grande e freqüente troca de informações de controle entre os nós. Assim, a quantidade de tráfego de atualizações pode ser bastante alta, e é ainda maior quando os nós com alta mobilidade estiverem presentes [AGR04].

Cada protocolo de roteamento *ad hoc* possui vantagens e desvantagens, de acordo com determinadas situações, portanto não é possível determinar qual é o melhor protocolo mas sim qual o mais adequado para uma aplicação em específico [ROY99] [BOU04]. No entanto a RFC 2501 [COR99] especifica uma série de propriedades que um protocolo deve possuir, tais como:

1. **Operação distribuída**

Para evitar a centralização que leva à vulnerabilidade. Esta propriedade é essencial para o roteamento em uma rede Ad Hoc.

2. **Livre de *loops***

Para que os pacotes não fiquem trafegando durante um período de tempo relativamente grande na rede, pode ser usada como solução uma variável do tipo TTL (*Time To Live*), mas uma abordagem melhor estruturada é mais indicada, por exemplo à utilização de número de seqüência.

3. **Operação sob demanda**

Ao em vez de assumir uma distribuição uniforme de tráfego dentro da rede (e manter as rotas sempre atualizadas), possibilitar que o protocolo de roteamento consiga adaptar o padrão de tráfego baseado na demanda. Se implementado corretamente, os recursos de energia e a largura da banda da rede serão utilizados mais eficientemente sem desperdiçar tempo atualizando rotas desnecessárias.

4. **Operação pró-ativa**

Em certos contextos, a latência adicional causada pela operação baseada na demanda pode ser inaceitável. A operação pró-ativa é desejável nos casos onde os recursos de energia e a largura da banda permitirem.

5. **Segurança**

Sem alguma forma de segurança na camada de enlace ou rede, um protocolo de roteamento *ad hoc* fica vulnerável a muitos tipos de ataques. Mecanismos de proteção para evitar a degradação e ou alteração no funcionamento do protocolo são desejáveis.

6. **Período de inatividade**

Como resultado da conservação de energia, ou de outra necessidade, os nós podem parar de transmitir e\ou receber por períodos arbitrários de tempo. Um protocolo deve ser capaz de gerenciar tais períodos sem trazer conseqüências adversas.

7. **Suporte a enlaces unidirecionais**

Ligações bidirecionais são tipicamente assumidas no projeto de protocolos de roteamento, no entanto muitos deles são incapazes de funcionar corretamente sobre ligações unidirecionais.

3.2 Protocolos de Roteamento Reativo

Nos protocolos reativos (*on-demand*), quando um nó de origem deseja enviar um pacote para um nó destino, este é armazenado em um buffer enquanto é executado o processo de descoberta de rotas em toda rede. Este processo é finalizado quando uma rota completa ou uma combinação de rotas até o destino é encontrada, possibilitando a transmissão do pacote. Após o estabelecimento da rota, os protocolos reativos mantêm esta rota até o momento em que o nó destino não seja mais alcançável ou a rota não seja mais utilizada.

Os seguintes protocolos de roteamento pertencem a este grupo: DSR (*Dynamic Source Routing*), o AODV (*Ad Hoc On-Demand Distance Vector Routing*) e o TORA (*Temporally Ordered Routing Algorithm*). A seguir será apresentado o funcionamento destes protocolos dando maior ênfase ao DSR e AODV por serem à base de implementação de alguns protocolos de roteamento seguro.

3.2.1 DSR (*Dynamic Source Routing*)

O protocolo DSR [JOH96] [JOH04] é um protocolo de roteamento reativo e sua característica principal é a utilização de roteamento por fonte (*source routing*), onde o nó de origem que está enviando um pacote sabe toda a rota salto a salto até o destino. Assim, o DSR permite que a estação de origem determine o caminho que será utilizado pelo pacote na rede para chegar até o seu destino. Esse caminho é listado no cabeçalho do pacote de dados e é chamado de *source route*. O nó origem então transmite o pacote para o primeiro nó identificado na rota (executa o primeiro salto). Quando um nó receber o pacote, e não for o destino final do pacote, simplesmente transmite o pacote ao próximo nó identificado na rota do pacote (executa o próximo salto). Quando o pacote alcança seu destino final é entregue a camada de transporte naquele nó.

Cada nó móvel que forma a rede *ad hoc* mantém uma tabela (*cache*) de rotas na qual armazena as rotas descobertas. Quando um pacote é enviado a outro nó, o remetente primeiramente confere sua tabela para verificar se já não possui uma rota para o nó destino. Se existe uma rota, o remetente irá utilizá-la para transmitir o pacote. Caso nenhuma rota seja encontrada, o remetente tenta descobrir uma nova rota usando o

processo de descoberta de rota. Enquanto espera pela descoberta da rota o nó continua seu processo normalmente, e armazena o pacote a ser enviado num *buffer* para transmiti-lo quando a rota for descoberta, ou descartar o pacote, deixando a tarefa de retransmissão para as camadas superiores do protocolo, caso necessário. A cada entrada na *cache* de rotas é associado um período de vencimento depois do qual a entrada é removida da *cache*.

Enquanto um nó estiver usando qualquer rota, a operação correta e contínua desta rota é monitorada por este nó através do mecanismo de manutenção da rota. Quando a rota apresentar algum problema, o mecanismo de descoberta de rotas pode ser usado novamente para descobrir uma nova e correta rota para o destino.

Como descrito acima, o protocolo DSR é composto de dois mecanismos que trabalham juntos para permitir a descoberta e manutenção de rotas [JOH96] [JOH04]:

1. **Descoberta de rota**

Mecanismo através do qual um nó A, que deseja enviar um pacote para o nó B, obtém uma rota para B. Este mecanismo é usado apenas quando o nó que deseja enviar um pacote não possui uma rota para o nó destino.

Um nó que deseja descobrir uma rota executa um *broadcast* de um pacote RREQ (*Route Request*) que será recebido por todos os nós dentro do alcance da transmissão. O pacote RREQ identifica o nó destino da rota que esta sendo solicitada. Se a descoberta de rota tem êxito o nó origem (nó que iniciou a descoberta da rota) recebe um pacote RREP (*Route Reply*) que lista uma seqüência de nós (saltos) da rede através dos quais o nó destino é alcançado. Para descobrir pedidos de requisição de rota duplicados, cada nó mantém uma lista com os pares (*initiator address, request id*) que foram recentemente recebidos. *Initiator address* é o endereço do nó que solicitou a descoberta da rota e *request id* é um identificador único da requisição.

Quando um nó recebe um pacote de pedido de rota, age de acordo com os seguintes passos [JOH96]:

- Se o par (*initiator address, request id*) para esta requisição é encontrado na lista das requisições recebidas recentemente, então este pedido é descartado.

- Se o endereço local está contido na lista de nós por onde o pacote deve passar para chegar ao destino, então este pedido é descartado.

- Se o endereço do destinatário é igual ao endereço local, então envia uma cópia da seqüência de nós por onde a mensagem passou para chegar a este nó (destino) para o nó que fez o pedido da rota. Caso este nó não seja o destino, mas conhece a rota para se chegar ao destino, esta rota é concatenada com a seqüência de nós por onde a mensagem já passou. Em seguida, um pacote RREP é enviado ao nó que fez o pedido da rota.

- Senão adiciona o endereço local na lista de nós por onde a mensagem passou e executa um novo *broadcast* com o pacote RREQ.

A resposta (RREP) enviada ao nó que fez o pedido da rota é gerada quando a requisição chega no nó destino ou em um nó que possui em sua *cache* uma rota para o nó destino ainda válida. Para retornar esta resposta é preciso ter uma rota para o nó de origem (nó que fez o pedido da rota). Caso a *cache* possua uma rota para o nó origem ela é usada, caso contrário, se *links* simétricos são suportados a resposta seguirá o caminho inverso da rota descoberta. Se *links* simétricos não são suportados, o pacote é armazenado em um buffer enquanto uma nova descoberta de rota é executada. Este mecanismo de descoberta de rotas e propagação da resposta é exemplificado nas Figura 9 e Figura 10, respectivamente.

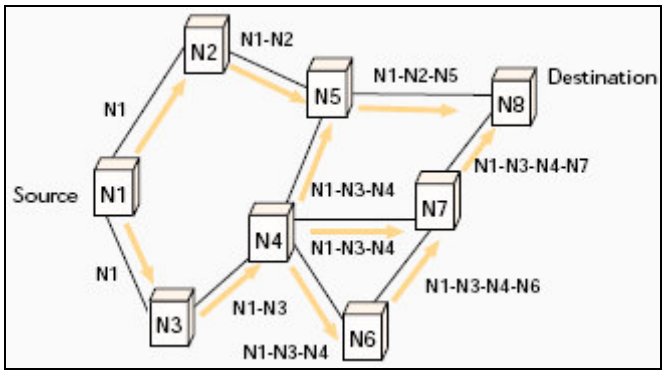


Figura 9 – Protocolo DSR – Exemplo de descobrimento de rota [ROY99]

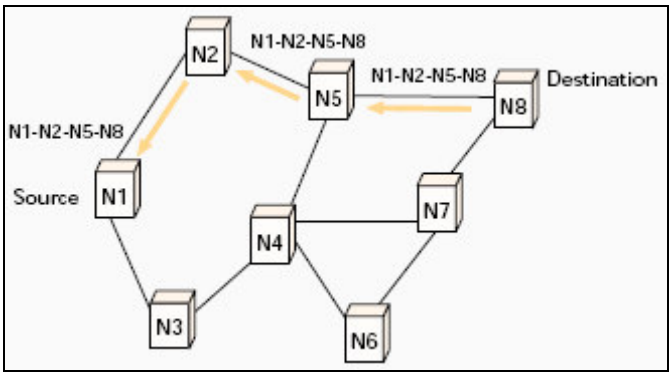


Figura 10 – Protocolo DSR – Exemplo de propagação de um pacote RREP [ROY99]

2. **Manutenção de rota**

É o mecanismo através do qual um nó é capaz de detectar, enquanto esta utilizando uma rota, se a topologia da rede sofreu mudanças e a rota não pode mais ser usada por algum motivo, como a quebra de algum *link*. Isto é feito através do uso de pacotes de erros de rota (RERROR) ou de pacotes ACK (*Acknowledgments*). Quando todas as rotas na *cache* forem descartadas, este mecanismo dispara uma nova descoberta de rotas. Para controlar o *overhead* gerado pela repetição no descobrimento de rota, os pacotes RREQ são enviados em intervalos variados.

Como este protocolo utiliza roteamento dinâmico, se adapta rapidamente quando a movimentação dos nós é freqüente e requer um pequeno ou nenhum *overhead* durante períodos nos quais os nós se movem com uma menor freqüência.

3.2.2 AODV (*Ad Hoc On-Demand Distance Vector Routing*)

O protocolo AODV [PER99] [PER03b] é essencialmente uma combinação dos protocolos DSR e DSDV pois utiliza basicamente os mecanismos de descoberta e manutenção de rotas do DSR juntamente com o roteamento por saltos e números de seqüência utilizados pelo DSDV.

Quando um nó A necessita uma rota para o nó B, é iniciado um *broadcast* de pacotes RREQ pela rede, incluindo o último número de seqüência conhecido para o nó B. Os pacotes RREQ são enviados pelos nós intermediários até atingir o nó B ou algum nó que tenha uma rota para o nó B como exemplificado na Figura 11 . Assim que o nó B receber o pacote RREQ imediatamente é gerado um pacote RREP contendo o número

de saltos necessários para alcançar o nó B e o número de seqüência mais atual para esta rota. Todos os nós intermediários que participam transmitindo os pacotes de RREP para o nó A contribuem para a criação de uma rota inversa para o nó A.

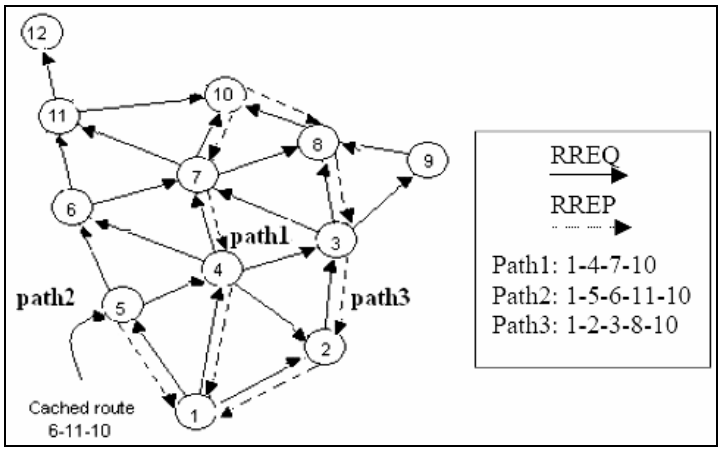


Figura 11 – Protocolo AODV - Descoberta de rotas [PER03b]

Para realizar a manutenção das rotas, o protocolo AODV exige que cada nó transmita em intervalos regulares uma mensagem de *hello*. Caso algum nó apresente uma quantidade *n* de falhas ao receber a mensagem de um nó vizinho, o *link* é considerado inválido. Na especificação do protocolo AODV é sugerido que seja utilizada a camada física para detectar *links* quebrados entre nós vizinhos [PER99]. Quando um *link* inválido é detectado, um pacote RERROR é transmitido para toda rede e um novo processo de descoberta de rota é iniciado para manter as tabelas de roteamento atualizadas.

3.3 Protocolos de Roteamento Pró-Ativos

Nos protocolos pró-ativos (*table-driven*), os nós que formam a rede, mantém as rotas de todos os possíveis destinatários em uma tabela de forma que quando um nó desejar enviar um pacote, basta consultar a tabela e utilizar a rota imediatamente. Nestes protocolos cada nó deve manter uma tabela com as informações de roteamento e disparar pela rede atualizações periódicas de rotas para garantir que os dados das tabelas estejam sincronizados em relação à estrutura atual da rede [ROY99].

Fazem parte deste grupo os protocolos DSDV (*Destination-Sequenced Distance-Vector Routing*), WRP (*Wireless Routing Protocol*), CGSR (*Clusterhead Gateway Switch Routing*), OLSR (*Optimized Link State Routing*) entre outros. A seguir serão apresentados os protocolos DSDV e WRP, com maiores detalhes em relação ao funcionamento do DSDV pois é utilizado como base pelo protocolo de roteamento seguro SEAD.

3.3.1 DSDV (*Destination Sequenced Distance Vector Routing*)

Um protocolo de roteamento por vetor de distância encontra os caminhos mais curtos entre os nós de uma rede através da implementação distribuída do algoritmo clássico de *Bellman-Ford*. Os protocolos de vetor de distância são fáceis de implementar e são eficientes na utilização da memória e da capacidade de processamento da CPU de cada nó [HUY02a]. O RIP (*Routing Information Protocol*) é um exemplo popular de protocolo de vetor de distância, o qual é extensamente usado em redes IP de tamanho moderado. O roteamento por vetor de distância pode ser usado para o roteamento em uma rede *ad hoc* na qual cada nó age como um roteador.

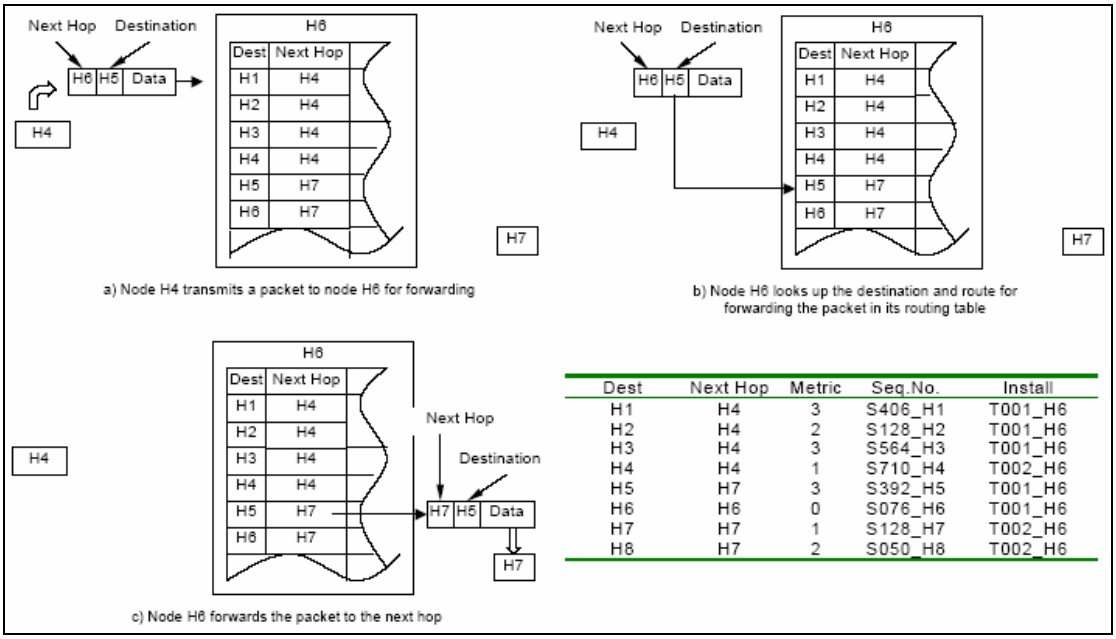
O protocolo DSDV é uma adaptação do RIP para o roteamento em redes *ad hoc*. O DSDV possui o atributo *sequence number*, para cada entrada na tabela de roteamento do protocolo RIP convencional. Através da utilização deste novo atributo, os nós móveis podem identificar que a informação de uma rota está desatualizada evitando a formação de *loops* no roteamento.

Cada nó em uma rede *Ad Hoc* mantém uma tabela que lista todos os possíveis destinos dentro da rede. Cada entrada nesta tabela contém o endereço (identidade) de um possível destino, a menor distância conhecida (normalmente em número de *hops*) para aquele destino e o endereço do nó vizinho que será o primeiro salto neste caminho mais curto para aquele destino. À distância para o destino é conhecida como *metric* nas entradas da tabela. Quando um nó estiver fazendo o roteamento de um pacote para algum destino, o nó transmite o pacote para o roteador (nó) vizinho indicado, e cada roteador usa sua própria tabela para realizar o próximo *hop* do pacote em busca do seu destino [PER94].

Periodicamente ou no instante em que a topologia da rede sofrer alguma mudança, cada nó móvel transmite informações de roteamento fazendo *broadcasting* ou *multicasting* de um pacote de atualização da tabela de roteamento para os nós vizinhos. Este pacote de atualização inicia com o atributo *metric* valendo um. Isto indica que cada vizinho receptor é um *hop* longe do outro nó. Depois de receber o pacote de atualização, os nós vizinhos atualizam as suas tabelas de roteamento incrementando de uma unidade o atributo *metric* e retransmitem este pacote para aos seus nós vizinhos correspondentes. O processo será repetido até todos os nós dentro da rede *Ad Hoc* receberem uma cópia do pacote de atualização [PER94].

Se os pacotes de atualização possuírem o mesmo *sequence number* para o mesmo nó, será usado aquele com o menor valor para *metric* e a outra rota será descartada ou armazenada como uma rota menos preferível.

A Figura 12 mostra um exemplo de uma rede *Ad Hoc* antes e depois do movimento dos nós. O nó H4 quer enviar um pacote ao nó H5. O nó H4 confere sua tabela de roteamento e identifica que o próximo *hop* é o nó H6. Então H4 envia o pacote para H6 como mostrado na Figura 12a. O nó H6 procura o próximo *hop* para alcançar o nó de destino H5 em sua tabela de roteamento quando recebe o pacote (Figura 12b).



Dest	Next Hop	Metric	Seq.No.	Install
H1	H4	3	S406_H1	T001_H6
H2	H4	2	S128_H2	T001_H6
H3	H4	3	S564_H3	T001_H6
H4	H4	1	S710_H4	T002_H6
H5	H7	3	S392_H5	T001_H6
H6	H6	0	S076_H6	T001_H6
H7	H7	1	S128_H7	T002_H6
H8	H7	2	S050_H8	T002_H6

Figura 12 – Protocolo DSDV - Exemplo de roteamento [HEG03]

O nó H6 então retransmite o pacote para o próximo *hop* H7 como demonstrado na tabela de roteamento da Figura 12c. O procedimento de roteamento é repetido ao longo do caminho até que o pacote chegue ao nó destino H5. É possível visualizar também na Figura 12 a tabela de roteamento do nó H6 antes do movimento dos nós, sendo que o campo *Install* da tabela ajuda a determinar quando uma rota antiga deverá ser apagada [HEG03].

Existem dois tipos de pacotes de atualização: o *full dump*, que considera toda informação de roteamento disponível e o *incremental*, que só considera as informações de roteamento que sofreram mudança desde o último *full dump*. Na Figura 13 é mostrado um exemplo de como um nó manipula um pacote de atualização do tipo *incremental*.

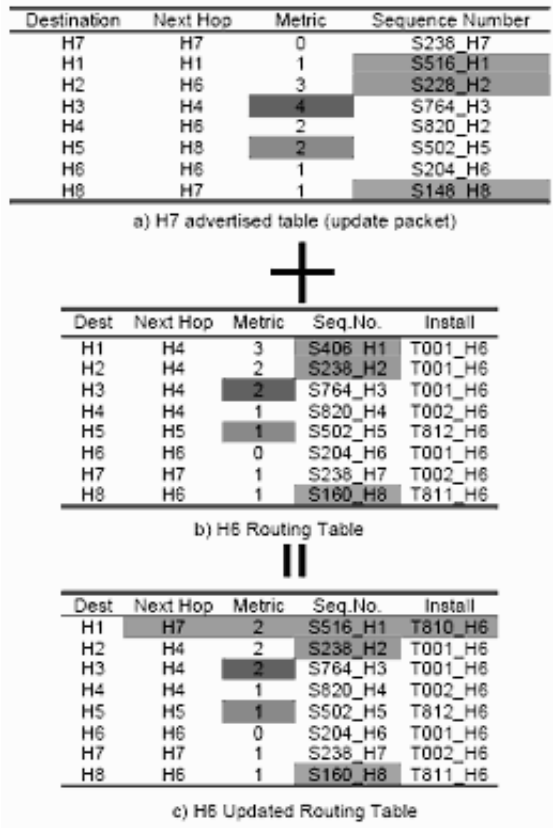


Figura 13 – Protocolo DSDV - Exemplo de atualização de informações de rota [HEG03]

Os pacotes de atualização full dump dificilmente são transmitidos quando um pequeno movimento de nós móveis está acontecendo. Atualizações do tipo incremental são transmitidas entre atualizações full dump quando ocorrem mudanças parciais da

tabela de roteamento como o recebimento de um novo sequence number ou mudanças significantes de rota (como mostrado na Figura 13a) [HEG03].

3.3.2 WRP (*Wireless Routing Protocol*)

O protocolo pró-ativo WRP proposto em [MUR96] tem como principal característica manter as informações de roteamento em todos os nós que compõem a rede *ad hoc*. Desta forma, cada nó possui quatro tabelas, que devem estar sempre atualizadas: tabela de distância, roteamento, custo do enlace e lista de retransmissão de mensagens. Esta última tabela armazena quais mensagens de atualização precisam ser retransmitidas e quais nós vizinhos precisam confirmar o recebimento da mensagem.

No WRP, quando um nó detecta mudanças nas rotas, é enviada uma mensagem de atualização para os nós vizinhos. Esta mensagem contém uma lista de informações de roteamento e uma lista de quais nós devem confirmar a atualização. Desta forma os nós vizinhos atualizam suas tabelas de roteamento e verificam a existência de novas rotas para outros nós. Caso alguma nova rota seja identificada, as informações necessárias são enviadas para o nó que iniciou o processo de atualização.

A detecção de novos nós na rede é feita através da troca de mensagens de reconhecimento e outras mensagens de roteamento [MUR96]. Sendo assim, os nós que não estão participando ativamente da rede devem enviar uma mensagem de *hello* periodicamente, caso contrário o nó será descartado das tabelas de roteamento. Desta forma, quando um novo nó deseja participar da rede, basta enviar uma mensagem de *hello*.

3.4 Sumário

Nesse capítulo foram apresentados as características e o funcionamento dos protocolos de roteamento reativo DSR e AODV e dos protocolos pró-ativos DSDV e WRP. Determinar precisamente qual destes protocolos é o mais adequado para todos os cenários não é uma tarefa fácil [ROY99]. Cada um destes protocolos apresenta vantagens e desvantagens sendo que a escolha de determinado protocolo dependerá

também do tipo de aplicação ao qual se destinam. Simulações realizadas em [BRO98] [BOU04] para comparação de desempenho reforçam esta idéia.

Entretanto, é possível afirmar que os protocolos reativos apresentam vantagens em relação aos protocolos pró-ativos, pois apresentaram um desempenho superior em diversos trabalhos [BRO98] [MAL99] [BOU04] [VIV06a]. Os protocolos reativos têm demonstrado maior eficiência em relação às mudanças na conectividade entre os nós, diminuindo o *overhead* de roteamento quando os nós apresentam uma mobilidade baixa.

Estes protocolos não consideram a existência de nós maliciosos na rede, os quais podem comprometer o roteamento dos pacotes. No próximo capítulo, serão apresentados protocolos de roteamento seguro baseados nos protocolos DSR, AODV e DSDV.

4 Segurança em Redes *Ad Hoc*

Neste capítulo será abordada a questão da segurança em redes sem fio *ad hoc* relacionada aos protocolos de roteamento seguro. Serão apresentados os possíveis ataques que nós maliciosos podem executar para degradar o funcionamento do roteamento. Na seqüência, serão estudados os protocolos de roteamento seguro Ariadne, BFTR, SEAD, SAODV e ARAN em relação às suas características de funcionamento e dos mecanismos utilizados para garantir segurança no roteamento.

4.1 Aspectos Preliminares

Como discutido nos capítulos anteriores, devido às suas características, uma rede sem fio *ad hoc* é extremamente vulnerável ao ataque de nós maliciosos. Enquanto ataques são possíveis em redes cabeadas, a natureza das redes sem fio aumenta seus efeitos, dificultando a detecção ou prevenção de ataques [YIS01]. Simulações realizadas em [MAR00] utilizando o protocolo DSR demonstraram que com a presença de 10% a 40% de nós maliciosos na rede, a vazão média da rede diminuiu entre 16% e 32%. Além disso, nas simulações realizadas em [XUE04] com o protocolo DSR, verificou-se uma redução de 30% na razão de entrega de pacotes com a existência de 20% de nós maliciosos. Portanto, é fundamental a utilização de protocolos de roteamento seguro para proteger a rede contra vários tipos de ataques, os quais podem degradar o desempenho ou até impedir o funcionamento da rede.

A segurança começa pela garantia de algumas propriedades como confidencialidade, integridade, autenticação, não repúdio e disponibilidade [ZHO99]. Além destas propriedades, em [YIS01] é recomendado que durante o projeto de um protocolo de roteamento seguro também sejam consideradas as propriedades de tempo e ordenação. Para alcançar estas propriedades, algumas técnicas podem ser utilizadas (Tabela 1). Entretanto, é necessário um rigoroso balanceamento entre desempenho e segurança, pois como a tarefa de roteamento é essencial na rede, os mecanismos de segurança apresentados na Tabela 1 não podem limitar o funcionamento desta operação.

Tabela 1 – Propriedades e técnicas para garantir segurança no roteamento [YIS01]

Propriedade	Técnica
Tempo	*Timestamp*
Ordenação	Número de seqüência
Autenticação	Senha, certificado digital
Autorização	Credencial
Integridade	*Hash*, assinatura digital
Confidencialidade	Criptografia
Não-repúdio	Encadeamento de assinaturas digitais

Os ataques realizados em uma rede *ad hoc* podem ser classificados de acordo com a pilha de protocolos de rede. Na Tabela 2 (baseada em [WUB06]), são listados exemplos de ataques em relação a cada camada da pilha de protocolos. Al guns ataques podem ocorrer em múltiplas camadas. Fazem parte do escopo deste trabalho apenas os ataques realizados na camada de rede.

Tabela 2 – Exemplos de ataques por camada da pilha de protocolos de rede

Camada	Ataques
Camada de aplicação	Repudiação, Corrupção de dados
Camada de transporte	*Session hijacking*, *Flooding*
Camada de rede	*Wormhole*, *Blackhole*, *Rushing Attack*, *Blackmail*, *Replaying*, *Routing Table Poisoning, Routing Table Overflow*
Camada de enlace	Análise de tráfego, Monitoração, Interrupção MAC 802.11, WEP *weakness*
Camada física	*Jamming*, *Eavesdropping*, Interceptação
Ataques em múltiplas camadas	DoS, *Impersonation*, *Replaying*, *man-in-the-middle*

4.2 Ataques

Neste trabalho, os ataques executados por nós maliciosos na camada de rede serão classificados de acordo com suas características, em dois conjuntos de ataques, como definido em [HUY04b]:

1. **Ataques de interrupção de roteamento**

Basicamente estes ataques têm o objetivo de evitar ou prejudicar o roteamento de pacotes legítimos na rede através do descarte de pacotes e alteração nas informações de roteamento. Como exemplos de ataques, podemos citar o *Wormhole*, *Blackhole*, *Blackmail, Rushing Attack.*

2. **Ataques de consumo de recursos**

Um nó malicioso pode injetar pacotes de dados falsos na rede com o objetivo de consumir recursos como: largura de banda, poder de processamento e memória. De forma semelhante, pacotes de roteamento falsos também podem ser injetados consumindo ainda mais recursos da rede. Exemplos deste tipo de ataque são: *Replaying, Routing Table Poisoning e Routing Table Overflow.*

Para uma aplicação, estes dois tipos de ataques podem ser classificados como ataques de DoS (*Denial of Service*) [HUY04b]. Abaixo serão apresentados vários ataques com características de quebra de rotas e de consumo de recursos:

1. ***Wormhole***

Este é um ataque de interrupção de roteamento, no qual é criado um túnel na rede usando um par de nós A e B ligados através de uma conexão privada na rede [HUY03a] [HUY06]. Todos os pacotes que são recebidos por A são transmitidos através do túnel para o nó B que retransmite por *broadcast* os pacotes para o resto da rede. De forma semelhante, o nó B envia todos os pacotes recebidos para o nó A. Em túneis com distâncias mais longas do que o intervalo normal de transmissão sem fio de um único salto, o invasor pode fazer com que o pacote enviado pelo túnel chegue mais cedo do que os outros pacotes transmitidos por uma rota de *multi-hops* normal.

Como descrito em [HUY03a], este ataque é particularmente perigoso quando utilizado contra protocolos de roteamento reativo como o DSR. Uma poderosa aplicação do ataque *Wormhole* pode ser conseguida quando cada pacote RREQ é enviado através do túnel diretamente para o nó destino. Quando os nós vizinhos deste nó recebem o RREQ, eles seguem o processamento normal do protocolo retransmitindo por *broadcast* o RREQ e, então, descartam, sem processar todos os outros pacotes RREQ recebidos e

originados a partir desse mesmo descobrimento de rota. Dessa maneira, esse ataque evita que seja descoberta qualquer outra rota que não seja através do *Wormhole*. Através da Figura 14 apresentada em [ARG05] pode ser visualizado um exemplo deste ataque em uma rede composta por 6 nós e um túnel entre os nós A e B.

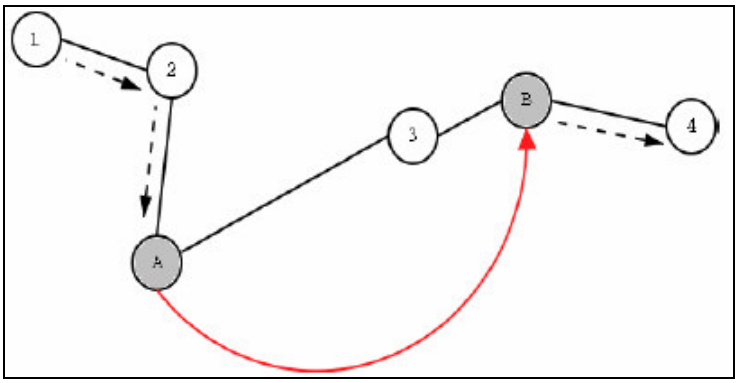


Figura 14 - Ataque *Wormhole* em uma rede *ad hoc* [ARG05]

Um invasor pode explorar o *Wormhole* para, por exemplo, descartar todos os pacotes de dados ao invés de propagá-los. Pode iniciar um ataque permanente de DoS no qual nenhuma outra rota para o destino será descoberta enquanto o invasor mantiver o *Wormhole* para pacotes RREQ e desta forma poderá descartar seletivamente ou modificar determinados pacotes de dados.

2. ***Rushing Attack***

O *Rushing Attack* também é um ataque do tipo interrupção de roteamento, o qual é direcionado contra protocolos de roteamento reativos que utilizam supressão de duplicatas em cada nó [HUY02a]. Um invasor espalha pacotes de RREQ rapidamente por toda a rede, suprimindo qualquer pedido de rota legítimo, posterior ao espalhado, uma vez que os nós excluem os pedidos duplicados devido à ordem de supressão de duplicatas.

Em termos gerais, se um nó malicioso consegue reenviar pacotes RREQ com uma eficiência maior que os nós legítimos, isto aumentará a probabilidade de que as rotas descobertas incluam o nó atacante. Então dois nós maliciosos poderiam utilizar o *Wormhole* para a execução mais eficiente do *Rushing Attack* [HUY03b] através da criação de um túnel de comunicação (Figura 14) entre eles no qual a velocidade de

transmissão fosse maior do que a transmissão entre nós legítimos. Sendo assim, este túnel faria parte da maioria das rotas descobertas.

3. ***Blackhole***

Outro exemplo de ataque de interrupção de roteamento é o *Blackhole* [HUY02a]. Neste ataque um nó malicioso cria um "buraco negro" no roteamento, isto é, pacotes que forem enviados por uma rota que contenha este nó, serão excluídos. Dependendo da posição deste nó, o ataque pode causar um efeito totalmente destrutivo na rede, particionando a rede ou até impedindo seu funcionamento.

Uma premissa de funcionamento deste ataque é que o nó malicioso se torne atrativo durante a escolha de rotas. Para tanto, vários métodos podem ser utilizados. Um nó malicioso pode inicialmente se passar por um nó confiável e apresentar um bom comportamento para, em seguida, iniciar o ataque. Outra forma seria o nó malicioso enviar pacotes de roteamento falsos, criando rotas menores com o objetivo de atrair o tráfego da rede para si e em seguida descartar os pacotes.

Um caso especial deste ataque é *Grayhole*. Nesta variação, o invasor exclui seletivamente alguns pacotes, encaminhando, por exemplo, apenas os pacotes de roteamento enquanto exclui os pacotes de dados.

4. ***Blackmail***

Este ataque de interrupção de roteamento é executado contra protocolos de roteamento no qual cada nó mantém uma lista negra com os nós que são considerados maliciosos e que deverão ser ignorados durante o roteamento [HUY02a]. São utilizados mecanismos para identificar nós maliciosos e propagar mensagens de alerta para toda a rede. Sendo assim, um nó malicioso pode enviar mensagens falsas para tentar isolar um nó confiável de toda a rede. Este ataque pode ser minimizado utilizando mecanismos para garantir o não-repúdio das mensagens de alerta [ARG05].

5. ***Routing Table Poisoning***

Os protocolos de roteamento utilizam tabelas para manter informações sobre as rotas na rede. Neste ataque de consumo de recursos, um nó malicioso pode fabricar pacotes de roteamento falsos ou modificar pacotes legítimos recebidos de outros nós e

em seguida enviá-los para toda rede, modificando as tabelas de roteamento dos nós [WUB06]. Por exemplo, um atacante pode enviar mensagens de atualização de rotas que não correspondem realmente a modificações na topologia da rede. Desta forma, este ataque pode resultar na utilização de rotas menos eficientes, na criação de *loops* no roteamento, em gargalos e até isolando determinadas partes da rede [ARG05].

6. ***Routing Table Overflow***

Neste ataque os nós maliciosos enviam para toda a rede pacotes falsos ou corrompidos com objetivo de consumir recursos dos nós e prejudicar o estabelecimento de novas rotas [ARG05]. Os protocolos de roteamento pró-ativos são mais vulneráveis a este tipo de ataque devido à atualização das rotas acontecer antes mesmo de serem necessárias [WUB06].

7. ***Replaying***

O ataque *Replaying* é considerado um ataque de consumo de recurso, pois o nó malicioso injeta na rede pacotes desnecessários com o único objetivo de consumir recursos como largura de banda da rede e poder de processamento dos nós. No *Replaying* um nó malicioso injeta na rede pacotes de roteamento (por exemplo, RREQ e RREP) capturados anteriormente, fazendo com os nós da rede atualizam as tabelas de roteamento com rotas incorretas [ARG05].

4.3 Segurança no Roteamento

A maioria dos protocolos de roteamento utilizados em redes *ad hoc* não utiliza mecanismos de segurança pois assumem que o ambiente é composto apenas por nós confiáveis [ARG05]. Esta suposição é particularmente perigosa em redes *ad hoc* pois o roteamento depende dos nós intermediários que formam a rota entre a fonte e o destino de um pacote. Devido à mudança dinâmica de topologia e ambiente aberto, uma rede *ad hoc* é extremamente vulnerável a presença de nós maliciosos e conseqüentemente a um conjunto de ataques [HUY04b]. Não há garantia de que o caminho de comunicação (escolhido no roteamento) é livre de nós maliciosos, os quais não seguiram

corretamente o protocolo com o objetivo de capturar informações, alterar dados e degradar o desempenho da rede [MAR00].

O projeto de protocolos de roteamento seguro para redes *ad hoc* enfrenta alguns desafios, pois devem operar de forma eficaz considerando alguns fatores tais como [COR99] [ROY99] [HUY04b]: natureza dinâmica da rede (mobilidade dos nós), recursos limitados dos nós (memória, poder de processamento e energia) e efeitos da comunicação sem fio (enfraquecimento do sinal e interferência). Entretanto como apresentado no item 3.1, a RFC 2501 [COR99] sugere que os protocolos utilizem mecanismos de segurança para evitar a degradação de desempenho e ou alteração no funcionamento do protocolo.

Focalizando a segurança da rede e considerando as características e as limitações da rede e dos nós que a compõem, foram propostos alguns protocolos de roteamento seguro para as redes *ad hoc* dentre eles: Ariadne, BFTR (Best-*effort Fault-Tolerant Routing*), SEAD (*Secure Efficient Ad hoc Distance vector*), SAODV (*Secure Ad Hoc On-Demand Distance Vector*), ARAN (*Authenticated Routing for Ad Hoc Networks*), SAR (*Security-Aware Routing Protocol*) e SADSR (*Security-Aware Adaptive Dynamic Source Routing*).

Cada protocolo implementa seu mecanismo de segurança utilizando diferentes técnicas (ver Tabela 1) tais como criptografia simétrica e cadeias de *hash* [HUY03c]. Apesar da criptografia assimétrica ter um custo computacional superior se comparado com a simétrica, a assinatura digital também têm sido utilizada para garantir segurança no roteamento, como demonstrado nos protocolos ARAN [SAN02] e SAODV [ZAP02]. A seguir serão apresentados os protocolos de roteamento seguro Ariadne, BFTR, SEAD, SAODV e ARAN.

4.3.1 Ariadne

O protocolo *Ariadne* [HUY02a] é baseado no DSR e utiliza criptografia simétrica para verificar a autenticidade e integridade das mensagens de roteamento trocadas entre os nós. Este protocolo garante que o nó alvo de um processo de descoberta de rota possa autenticar o nó de origem. Além disso, permite que o nó de origem possa autenticar cada nó intermediário que compõem a rota descoberta. Garante

também que nenhum nó intermediário possa alterar as informações de roteamento, como por exemplo, alterar a lista de nós do pacote RREP.

As mensagens de roteamento podem ser autenticadas de três formas [HUY02a]: troca de chaves secretas entre cada par de nós, troca de chaves secretas entre os nós e autenticação *broadcast*, ou assinaturas digitais. A implementação do Ariadne neste trabalho utiliza o mecanismo TESLA [PER05], um esquema eficiente de autenticação *broadcast* que requer uma sincronização fraca de tempo entre os nós. Os parâmetros de *timeout* do DSR foram alterados pelo Ariadne para considerar a latência imposta pelo TESLA durante a divulgação das chaves para autenticação (ver Tabela 6 do item 6.3).

O TESLA baseia-se na utilização de códigos de autenticação de mensagens MAC (*Message Authentication Code*) e em cadeias de *hash*. O MAC é utilizado para garantir a integridade e autenticidade das informações trocadas por duas entidades através de um canal de comunicação inseguro. O TESLA determina que o nó que iniciar a descoberta de rota deve criar uma cadeia de *hash* cujos valores serão utilizados como chaves para o cálculo do MAC (utilizando por exemplo o HMAC[1]) para autenticação das mensagens. O nó de origem deve divulgar o último valor da cadeia de *hash* gerada, sendo que a cadeia deve ser utilizada no sentido inverso da geração para autenticação das mensagens [PER05].

Durante o envio de um pacote, o TESLA exige que o nó de origem calcule o tempo médio de transmissão fim-a-fim do pacote até o destino, divulgando a chave utilizada depois de decorrido esse tempo. Para verificar se a chave recebida está correta, o nó de destino deve aplicar a função *hash* sobre a chave um número adequado de vezes e comparar o resultado com o último elemento da cadeia de *hash*, que foi divulgado pelo nó de origem. Se houver atraso no recebimento do pacote ou a chave for divulgada antes que o nó de destino receba o pacote, o pacote deve ser descartado.

Os mecanismos de descoberta de rotas e manutenção de rotas sofreram algumas alterações em relação aos usados pelo DSR. Desta forma, novos campos foram adicionados às mensagens no intuito de atingir os objetivos de segurança especificados por este protocolo. A seguir seguem as modificações realizadas nestes mecanismos:

[1] Tipo de *Message Authentication Code* que utiliza função de *hash* e uma chave secreta.

1. **Descoberta de rota**

Na Figura 15 é apresentado um exemplo de descobrimento de rota no *Ariadne*. O nó de origem S deseja descobrir a rota para atingir o nó D. Neste exemplo devemos assumir que os nós de origem e destino já realizaram o compartilhamento das chaves secretas K_{SD} e K_{DS}, respectivamente [HUY02a]. No cabeçalho do pacote RREQ, além dos campos utilizados pelo DSR (endereço do nó de origem, do nó de destino, ID da requisição e lista de nós), o protocolo Ariadne adicionou o campo *ti* (*time interval*), utilizado pelo TESLA, um campo para a cadeia de *hash* e outra para a lista MAC.

Primeiramente, o nó S inicia a cadeia de *hash* calculando o MAC dos campos do pacote RREQ, com sua chave secreta K_{SD}, para garantir a autenticidade das informações. Em seguida o nó S realiza o *broadcast* do pacote RREQ para seus nós vizinhos.

Quando um nó intermediário, por exemplo nó A da Figura 15, recebe a requisição, é criada a nova cadeia de *hash* h_1 executando a função de *hash* sobre o seu próprio endereço e o valor da cadeia de *hash* anterior ($h_1 = H[A, h_0]$). Também é calculado o próximo elemento M_A da lista MAC usando sua própria chave TELSA K_{Ati} para um específico intervalo de tempo *ti*. Antes de retransmitir o pacote RREQ, o nó A atualiza o valor da cadeia de *hash* com o novo valor h_1 e adiciona o M_A na lista MAC.

S:	$h_0 = \text{MAC}_{K_{SD}}(\texttt{REQUEST}, S, D, id, ti)$
$S \rightarrow *$:	$\texttt{REQUEST}, S, D, id, ti, h_0, (), ()\rangle$
A:	$h_1 = H[A, h_0]$
	$M_A = \text{MAC}_{K_{A_{ti}}}\ \texttt{REQUEST}, S, D, id, ti, h_1, (A), ())$
$A \rightarrow *$:	$\texttt{REQUEST}, S, D, id, ti, \mathbf{h_1}, (\mathbf{A}), \mathbf{M_A})$
B:	$h_2 = H[B, h_1]$
	$M_B = \text{MAC}_{K_{B_{ti}}}(\texttt{REQUEST}, S, D, id, ti, h_2, (A, B), (M_A))$
$B \rightarrow *$:	$\texttt{REQUEST}, S, D, id, ti, \mathbf{h_2}, (A, \mathbf{B}), (M_A, \mathbf{M_B})\rangle$
C:	$h_3 = H[C, h_2]$
	$M_C = \text{MAC}_{K_{C_{ti}}}(\texttt{REQUEST}, S, D, id, ti, h_3, (A, B, C), (M_A, M_B))$
$C \rightarrow *$:	$\texttt{REQUEST}, S, D, id, ti, \mathbf{h_3}, (A, B, \mathbf{C}), (M_A, M_B, \mathbf{M_C})$
D:	$M_D = \text{MAC}_{K_{DS}}(\texttt{REPLY}, D, S, ti, (A, B, C), (M_A, M_B, M_C))$
$D \rightarrow C$:	$\texttt{REPLY}, D, S, ti, (A, B, C), (M_A, M_B, M_C), \mathbf{M_D}, ()$
$C \rightarrow B$:	$\texttt{REPLY}, D, S, ti, (A, B, C), (M_A, M_B, M_C), M_D, (\mathbf{K_{C_{ti}}})$
$B \rightarrow A$:	$\texttt{REPLY}, D, S, ti, (A, B, C), (M_A, M_B, M_C), M_D, (K_{C_{ti}}, \mathbf{K_{B_{ti}}})$
$A \rightarrow S$:	$\texttt{REPLY}, D, S, ti, (A, B, C), (M_A, M_B, M_C), M_D, (K_{C_{ti}}, K_{B_{ti}}, \mathbf{K_{A_{ti}}})$

Figura 15 – Protocolo Ariadne - Descobrimento de rota [HUY02a]

Quando o nó D recebe o pacote RREQ, é verificada a validade da requisição conferindo se as chaves TESLA dos nós intermediários ainda são válidas (isto é, se ainda não foram divulgadas) para o intervalo de tempo *ti*. É verificado também se a cadeia de *hash* está correta calculando novamente toda a cadeia e comparando o resultado. Caso o pacote RREQ seja válido, é gerado um pacote RREP para ser enviado para o nó S.

O pacote RREP é composto pelos campos: endereço do nó de origem e destino, intervalo de tempo (*ti*), lista de nós, lista MAC, MAC do destino (M_D) e lista de chaves TESLA. O campo M_D é calculado usando a chave K_{DS} compartilhada com o nó S. Cada nó intermediário no caminho de volta para o nó S adiciona sua chave TESLA (por exemplo, a chave K_{Ati} utilizada para calcular M_A). Utilizando a lista de chaves TESLA e o campo M_D o nó S verifica a autenticidade da rota, evitando a utilização de rotas modificadas.

2. **Manutenção de rota**

Para evitar que os nós maliciosos enviem pacotes RERROR falsos, o protocolo *Ariadne* também exige que estes pacotes sejam autenticados, como apresentado no mecanismo de descobrimento de rota. Desta forma, sempre que algum problema acontecer no roteamento, o pacote RERROR deverá ser autenticado em todos os saltos até atingir o nó de origem [HUY02a].

4.3.2 BFTR (*Best-effort Fault-Tolerant Routing*)

O protocolo de roteamento seguro BFTR é baseado no DSR e seu objetivo é manter o desempenho no roteamento, com uma alta taxa de entrega de pacotes, mesmo com a presença de nós maliciosos. O projeto deste protocolo tem como base a redundância intrínseca as redes sem fio *ad hoc*, isto é, a existência de várias rotas entre os nós. Simulações realizadas em [XUE04] demonstram que mesmo com uma densidade elevada de nós maliciosos, a probabilidade de existência de rotas confiáveis se mantém alta, permitindo que o desempenho no roteamento possa ser mantido em níveis aceitáveis.

A idéia básica do protocolo BFTR é a seguinte: enquanto um nó malicioso pode apresentar vários comportamentos, o comportamento de um nó confiável é geralmente o mesmo: os pacotes são transmitidos corretamente com um desempenho aceitável [XUE04], isto é, com uma alta taxa de entrega de pacotes. Conseqüentemente, uma rota livre de nós maliciosos apresenta o mesmo comportamento. As rotas que desviarem deste padrão são descartadas pelo BFTR. Desta forma, o BFTR evita vários ataques de DoS [HUY04b] (por exemplo, o *Blackhole*) através da análise das respostas do nó destino.

Este protocolo não exige dos nós intermediários um suporte de segurança. Em vez disso, o nó de origem somente analisa o desempenho fim-a-fim para julgar se o pacote foi recebido com sucesso pelo nó destino. Entretanto, este algoritmo supõe que exista uma relação de confiança já estabelecida entre os nós de origem e destino através do compartilhamento de chaves secretas, para a verificação de autenticidade dos pacotes de roteamento.

BFTR implementa três mecanismos: descoberta de rota, escolha da rota e manutenção da rota, sendo que a descoberta e a manutenção são bastante semelhantes ao do protocolo DSR. A seguir uma descrição do funcionamento destes mecanismos no BFTR:

1. **Descoberta de rota**

Da mesma forma que o protocolo DSR, o BFTR realiza um *broadcast* de pacotes RREQ na rede durante o processo de descobrimento de rota. Mas ao contrário do DSR que identifica apenas um sub-conjunto de rotas devido à supressão de pacotes RREQ duplicados, o BFTR descobre todas as possíveis rotas entre os nós. Para isto o BFTR não utiliza supressão de RREQ duplicados e exige que sejam enviadas múltiplas respostas RREP. Para cada RREQ recebido, o nó deve enviar um pacote RREP pelo mesmo caminho utilizado pelo RREQ recebido, para que sejam descobertas várias rotas entre o nó de origem e de destino. Os pacotes RREP devem ser assinados com uma chave secreta compartilhada, para permitir a verificação de autenticidade no nó de origem e evitar ataques de consumo de recursos como *Replaying* e *Routing Table Poisoning*.

2. **Seleção de rota**

Tendo um conjunto de rotas obtidas pelo mecanismo de descoberta de rota, o nó de origem determina localmente qual a melhor rota utilizando o algoritmo BFTR da Figura 16 [XUE04]. Inicialmente, o algoritmo realiza uma ordenação das rotas, na qual as rotas menores são movidas para o topo da pilha. Após a ordenação, é selecionada a rota mais curta que será utilizada para enviar os pacotes. Esta rota será utilizada enquanto o teste de rota (Figura 16) obtiver o comportamento esperado. Caso o teste indique que o caminho deve ser descartado, a próxima rota mais curta é selecionada. Quando todas as rotas forem descartadas, uma nova descoberta de rota é iniciada.

```
route() {
        while (true) {
                Ω  = route discovery(); /* envia pacotes RREQ e gera um conjunto de rotas */
                Ω' = sort(Ω); /* ordena as rotas de acordo com o tamanho */
                while (Ω' ≠ ∅) {
                        π = dequeue(Ω'); /* seleciona a rota mais curta */
                        test(π, α, p0, n0); /* analisa o comportamento da rota */
                }
        }
}
```

Figura 16 – Protocolo BFTR - Algoritmo de escolha da melhor rota [XUE04]

Durante o teste de rota (Figura 17) são utilizadas algumas heurísticas para determinar quando uma rota deve ser rejeitada. Desta forma, são considerados a razão de entrega de pacotes desejada, quantidade de pacotes enviados cujo comportamento será analisado e a probabilidade de erro (isto é, a probabilidade que uma rota adequada possa ser rejeitada erroneamente).

```
/* π é a rota sendo testada ; α probabilidade de erro;
p0 razão de entrega de pacotes esperada para uma rota confiável
n0 número de pacotes que serão analisados */
test (π, α, p0, n0) {
        n = 0; /* inicializa contagem do número de pacotes enviados */
        while (true) {
                envia pacotes pela rota π;
                atualiza n e calcula w considerando min(n, n0) pacotes;
                if (w < f (α, p0, min(n, n0)))
                        rejeita π;
        }
}
```

Figura 17 – Protocolo BFTR - Algoritmo de teste e seleção de rota [XUE04]

No algoritmo de teste de rota, o nó de origem necessita saber o número de pacotes *n0* que foram enviados pela rota que será analisada para calcular *w*. Para que isto seja possível, o nó de destino envia pacotes ACK para o nó de origem. Os pacotes ACK são assinados, utilizando o compartilhamento de chaves, garantindo o cálculo correto de *w*.

Um pacote é considerado enviado com sucesso quando é recebido com um *delay* menor ou igual a valor *d*. Então, para cada pacote enviado o nó de origem inicia um contador. Se no momento que o pacote ACK for recebido o valor do contador for maior que 2 * *d*, o pacote é considerado perdido. O valor de *d* é estimado durante a fase de descobrimento de rota.

3. **Manutenção da rota**

Este mecanismo não apresenta modificações em relação ao DSR. Sempre que for detectada alguma falha em uma rota, ela será descartada e a próxima rota presente na tabela de roteamento será utilizada. O protocolo BFTR não faz distinção quanto a problemas nas rotas devido a mobilidade dos nós ou existência de nós maliciosos. Em ambos os casos a rota será considera imprópria e será descartada. Caso todas as rotas forem descartadas, este mecanismo inicia uma nova descoberta e seleção de rotas. Da mesma forma que o DSR, os pacotes de RREQ são enviados em intervalos de tempo para controlar o *overhead* que poderia ser causado por vários descobrimentos de rotas repetidos.

O protocolo BFTR é do tipo melhor esforço e não garante a melhor performance no roteamento. A probabilidade da entrega de pacotes falhar aumenta maior a quantidade de nós maliciosos presentes na rede. Tal situação também poderá ocorrer quando a rede estiver bastante sobrecarregada. Do ponto de vista do protocolo, estes dois casos não são distinguíveis, mas ambos indicam que a rede não está propicia a atingir um bom desempenho no roteamento; nenhum protocolo irá trabalhar eficientemente nestas condições [XUE04].

4.3.3 SEAD (*Secure Efficient Ad Hoc Distance Vector Routing*)

O SEAD é um protocolo pró-ativo de roteamento seguro para redes *Ad Hoc* que utiliza o roteamento por vetor de distância baseado no protocolo DSDV. O projeto do SEAD [HUY02b] utiliza a função de *hash* na implementação de cada uma de suas funcionalidades para criar um protocolo eficiente e prático e que seja robusto contra múltiplos tipos de ataques, os quais podem gerar estados incorretos nas tabelas de roteamento de qualquer nó da rede. A implementação do protocolo SEAD foi inspirada no DSDV-SQ, uma versão otimizada do protocolo DSDV como demonstrado nas comparações realizadas em [JOH99].

O objetivo deste protocolo é proteger a rede contra ataques que modificam informações de roteamento transmitidas por durante a fase de atualização das tabelas de roteamento pois o roteamento pode ser interrompido caso sejam modificados os campos *sequence number* ou *metric* da mensagem de atualização da tabela de roteamento.

Para isto, o protocolo SEAD autentica os campos *sequence number* e o *metric* usando elementos da cadeia de *hash* sem utilizar a criptografia assimétrica devido as limitações das redes *ad hoc* discutidas no capítulo 3. Entretanto, o protocolo SEAD assume que exista um mecanismo de autenticação e distribuição de chaves para que os nós possam autenticar os elementos da cadeia de *hash* [HUY02b].

Cada nó usa um elemento específico de sua cadeia de *hash* em cada atualização de roteamento iniciada no próprio nó (*metric* 0). Baseado neste elemento inicial, a cadeia de *hash* permite que um nó intermediário realize a autenticação dos campos *metric* e *sequence number* das mensagens de atualização recebidas, desde que este nó já possua seu elemento autenticado da mesma cadeia de *hash*. A utilização da função de *hash* em cada entrada da tabela de roteamento previne que um nó malicioso anuncie uma rota para algum destino reivindicando um *sequence number* maior que o do próprio destino. Igualmente, um nó não pode anunciar uma rota melhor que aquelas anunciadas, pois o *metric* existente em uma rota não pode ser diminuído devido à natureza da cadeia de *hash* [HUY02b].

Quando um nó recebe uma atualização de roteamento, este confere a autenticidade da informação de cada entrada na atualização usando o endereço do destinatário, o *sequence number* e o *metric* da entrada recebida juntamente com o último

valor de *hash* recebido da cadeia de *hash* do destinatário. Em seguida o nó calcula o *hash* dos elementos recebidos quantas vezes for o correto e compara com o valor autenticado anteriormente. Dependendo da comparação, a rota pode ser aceita ou descartada caso o valor de *hash* calculado e o valor de *hash* autêntico forem diferentes.

A fonte de cada mensagem de atualização de roteamento no SEAD também deve ser autenticada, caso contrário, pode permitir a criação de *loops* no roteamento. Duas maneiras são propostas para realizar a autenticação: o primeiro é baseado no mecanismo de autenticação por broadcast TESLA, utilizado pelo protocolo Ariadne [HUY02a]. A segunda alternativa que não requer sincronização de tempo na rede como o TESLA, é utilizar o compartilhamento de chaves entre os nós da rede e utilizar estas chaves em conjunto com o MAC (*Message Authentication Code*).

No protocolo SEAD todos os nós que compõem a rede devem ter uma cadeia de *hash* válida. Os elementos desta cadeia são utilizados sucessivamente para autenticar cada mensagem de roteamento, desde que um elemento inicial autenticado tenha sido utilizado. Esta cadeia tem um tamanho máximo e deve ser gerada novamente quando todos os seus elementos já tenham sido utilizados.

4.3.4 SAODV (*Secure Ad Hoc On-Demand Distance Vector Routing*)

O protocolo SAODV [ZAP02] [ZAP06] é uma extensão segura do protocolo AODV para proteger a rede contra o ataque de nós maliciosos. Para isto utiliza assinatura digital para autenticar os campos imutáveis das mensagens de roteamento e cadeias de *hash* para garantir a autenticidade do campo *metric* durante uma descoberta de rotas, pois este é o único campo das mensagens de roteamento que sofre alterações no protocolo AODV [ZAP02]. Como o protocolo SAODV utiliza criptografia assimétrica para assinatura digital, é necessária a existência de um gerenciamento de certificados digitais para que os nós possam adquirir suas identidades digitais e verificar a validade dos certificados dos outros nós.

Para a transmissão das informações adicionais utilizadas pelos mecanismos de segurança, o SAODV incorporou novos campos nas mensagens de roteamento do AODV como pode ser visualizado na Figura 18 [ZAP02]. O campo *Hash Function*

representa o tipo de algoritmo de *hash*. *Max Hop Count* define o número máximo de nós por onde a mensagem pode passar. O campo *Hash* representa um número randômico e o campo *Top Hash* guarda o resultado da aplicação da função de *hash* sobre o número randômico tantas vezes quanto for o *Max Hop Count*.

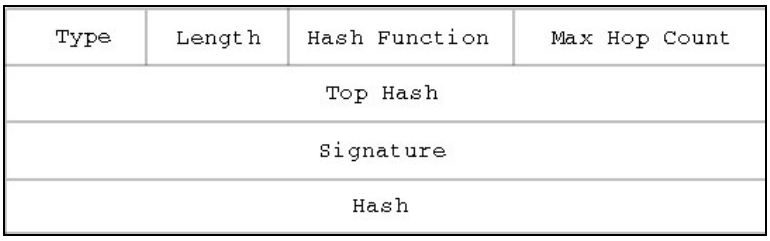


Figura 18 – Protocolo SAODV - Extensões incorporadas nas mensagens do AODV [ZAP02]

Para enviar um pacote RREQ ou RREP o nó deve: definir o campo *Max Hop Count* com o valor do TTL, gerar o número randômico e aplicar a função de *hash* tantas vezes quanto for o *Max Hop Count* e armazenar o valor no campo *Top Hash*. Em seguida deve assinar digitalmente todos os campos menos o número randômico e o *Hop Count* do cabeçalho do AODV. Quando um nó intermediário recebe este pacote, esse verifica a integridade dos campos assinados digitalmente e os compara com o resultado da aplicação da função de *hash* n vezes sobre o número randômico, onde n = (*Max Hop Count – Hop Count*), com o valor armazenado no campo *Top Hash*. Após esta verificação, é aplicada a função de *hash* no campo *Hash* para contabilizar mais um salto no roteamento e então o pacote é enviado para o próximo nó intermediário com o campo *Hash* atualizado.

Os pacotes RERROR também tem os campos assinados digitalmente exceto o campo *sequence number* de destino Figura 19. Como o destinatário do pacote (nó D) não autenticou o *sequence number* de destino, devido a falha na transmissão, o nó A não deve atualizar sua tabela de roteamento baseado no pacote ERROR enviado pelo nó C. Desta forma os pacotes de RERROR são utilizados no SAODV apenas para identificar quando uma rota deve ser completamente removida da tabela de roteamento.

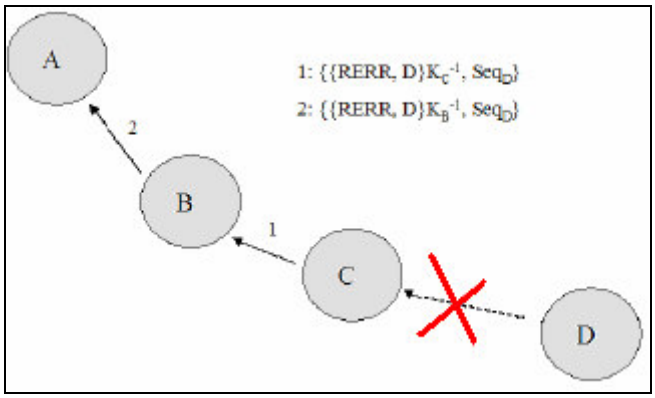


Figura 19 – Protocolo SAODV - Manutenção de rota [ZAP02]

4.3.5 ARAN (*Authenticated Routing for Ad Hoc Networks*)

O ARAN proposto em [SAN02], é um protocolo reativo que utiliza como base o protocolo AODV e faz uso da criptografia assimétrica para adicionar segurança no roteamento em redes *ad hoc*. Neste protocolo cada nó possui um certificado digital emitido por uma AC (Autoridade Certificadora) confiável, o qual é utilizado para a garantir a autenticidade e não-repúdio na troca de mensagens de roteamento entre os nós que compõem a rede.

O roteamento no protocolo ARAN inicialmente consiste na emissão de certificados por uma AC para cada nó que deseja se conectar a rede. O protocolo considera que cada nó conheça o certificado da AC em questão. Em seguida, é executado o processo de descoberta de rotas.

Na de descoberta de rotas é realizada a autenticação fim-a-fim entre nós, como exemplificado na Figura 20 [ARG05]. Os nós realizam o *broadcast* de pacotes RDP (*Route Discovery Packet*) assinados, que incluem o certificado do nó de origem, *nonce*, *timestamping* e o endereço do nó de destino. Cada nó intermediário deve validar a assinatura do pacote RDP, atualizar sua tabela de roteamento, remover o certificado e a assinatura do nó anterior (menos do nó de origem), assinar o pacote RDP com seu certificado e enviá-lo para seus nós vizinhos. A assinatura do pacote RDP evita que nós maliciosos injetem pacotes para alterar as rotas ou formar *loops*. Quando o nó destino (nó C) recebe o RDP, envia diretamente (*unicast*) para o nó de origem (nó A) através do caminho inverso, um pacote REP (*Route Replay*) assinado, contendo seu certificado e o

mesmo *nounce* presente no RDP. Desta forma o nó de origem pode validar o REP comparando o *nounce* e verificando a assinatura associada.

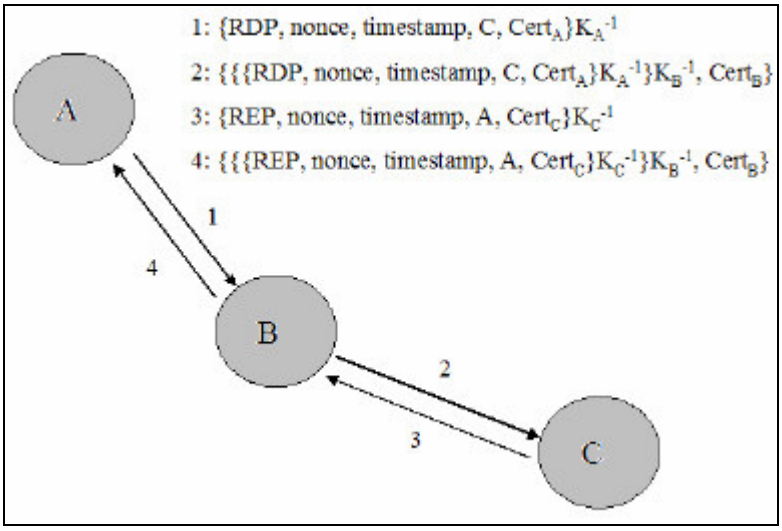


Figura 20 – Protocolo ARAN - Descoberta de rota [ARG05]

O protocolo ARAN não permite que os nós intermediários que possuam uma rota para o nó destino enviem um pacote REP para o nó que originou a descoberta de rota. Isto evita o surgimento de *Routing Loops* no roteamento mas implica em um aumento da latência na rede [SAN02].

O processo de manutenção de rotas é realizado através do *broadcast* de mensagens de erro (ERR) assinadas pelos nós que geraram a mensagem. A assinatura das mensagens ERR garante o não-repúdio, evitando que nós maliciosos injetem na rede falsos avisos de *links* quebrados. As mensagens ERR também incluem o *nounce* e *timestamping* para evitar ataques *Replaying*.

O protocolo ARAN utiliza certificados com tempo de validade limitado. Desta forma, o servidor de certificação realiza o broadcast de mensagens de revogação de certificados. Todos o nós retransmitem as mensagens e atualizam as rotas para evitar que nós com o certificados revogados sejam utilizados no roteamento. Este processo não é seguro pois os nós maliciosos podem simplesmente descartar estas mensagens.

5 Análise dos Protocolos de Roteamento Seguro

Neste capítulo é apresentado um comparativo entre os protocolos de roteamento seguro Ariadne, BFTR, SEAD, SAODV e ARAN em relação a defesa oferecida por esses protocolos para proteger a rede contra os ataques apresentados no item 4.2. São apresentadas as características do simulador NS-2 e as métricas utilizadas para a análise desses protocolos. Finalmente, é demonstrado o ambiente de simulação e os resultados dos experimentos realizados para a comparação de desempenho entre os protocolos reativos Ariadne e BFTR, foco principal desta dissertação.

5.1 Protocolos de Roteamento Seguro versus Ataques

Na Tabela 3 baseada em [ARG05] é apresentado uma comparação dos protocolos de roteamento discutidos no item 4.3 em relação a defesa contra aos ataques apresentados no item 4.2. O ataque *Blackmail* é aplicado em protocolos de roteamento, como o *Watchdog* and *Pathrater* [MAR00], que utilizam um IDS (*Intrusion Detection System*) para detectar os nós maliciosos. Portanto este ataque não pode ser usado contra os protocolos analisados neste trabalho.

Tabela 3 - Defesa dos protocolos de roteamento seguro contra os ataques

Ataques	**Protocolos de Roteamento Seguro**				
	Ariadne	**BFTR**	**SAODV**	**SEAD**	**ARAN**
Wormhole	não	não	não	não	não
Rushing Attack	não	não	não	NA	não
Blackhole	não	sim	não	não	não
Blackmail	-	-	-	-	-
Replaying	sim	sim	sim	sim	sim
Routing Table Poisoning	sim	sim[2]	sim	sim	sim
Routing Table Overflow	sim	sim	não	sim	não

[2] O protocolo BFTR não exige autenticação das mensagens de roteamento entre os nós intermediários, apenas entre o nó de origem e destino, diminuindo a proteção contra este ataque.

Para prevenir a rede contra o ataque *Blackhole*, uma solução seria a utilização de um mecanismo para explorar a redundância de rotas existentes em uma rede *ad hoc*. Desta forma, apenas o protocolo BFTR apresenta defesa contra o ataque *Blackhole* devido a seus mecanismos de descoberta e seleção de rotas. Este protocolo analisa constantemente o desempenho das rotas através da métrica razão de entrega de pacotes. Desta forma, as rotas compostas por nós maliciosos, executando o ataque *Blackhole*, terão um comportamento anormal e serão automaticamente evitadas pelo BFTR [XUE04]. A redundância de rotas também é utilizada no mecanismo 2ACK, proposto recentemente em [LIU06], para proteger o roteamento contra o *Blackhole*.

Como apresentado, nenhum dos protocolos estudados apresenta mecanismos de defesa contra o ataque *Wormhole*. Mas em [HUY03a] [HUY06] é proposto um mecanismo chamado *Packet Leashes* para proteger a rede contra o *Wormhole*. Para isto, são adicionadas informações extras nos pacotes enviados, as quais são utilizadas no nó de destino para verificar se o pacote foi enviado por uma rota cujo tamanho seja irreal. Este mecanismo pode ser temporal ou geográfico cujos requisitos operacionais são: sincronização extremamente precisa do tempo ou sincronização menos rígida do tempo e localização geográfica através de GPS (*Global Positioning System*). Já em [LAZ05] [POO07] é apresentado um *framework* baseado na localização geográfica dos nós, o qual possibilita a detecção deste ataque e previne a rede contra este ataque.

Assim como para o *Wormhole*, nenhum dos protocolos reativos apresenta defesa contra o *Rushing Attack*. Para evitar este ataque, foi proposto o mecanismo RAP (*Rushing Attack Prevention*) [HUY03b] para garantir segurança no processo de descobrimento de rotas. O RAP é composto por três partes: detecção segura de nós vizinhos, delegação segura de rotas e seleção randômica de pacotes RREQ.

Nos protocolos reativos, um nó B considera um nó A vizinho quando recebe uma mensagem via *broadcast* do nó A. A detecção segura de nós vizinhos permite ao nó B verificar se o nó A está dentro do limite máximo de transmissão. Quando o nó A envia o RREQ para o nó B, a mensagem de delegação de rota é assinada, permitindo ao nó B retransmitir o RREQ. Quando o nó B verifica que o nó A está dentro do alcance de transmissão, o nó B assina uma mensagem de aceite de delegação. A supressão de duplicatas é substituída pela seleção randômica de RREQ, minimizando as chances de que os nós maliciosos dominem todas as rotas retornadas.

Todos os protocolos de roteamento seguro analisados neste trabalho possuem mecanismos para proteger a rede contra os ataques de consumo de recursos *Replaying* e *Routing Table Poisoning*. Como apresentado no item 4.3, estes protocolos utilizam determinadas técnicas (Tabela 1) para verificar a autenticidade e integridade dos pacotes de roteamento transmitidos entre os nós durante o processo de descobrimento e atualização de rotas. Desta forma, estes protocolos impedem a utilização de pacotes de roteamento desatualizados, modificados ou falsos e conseqüentemente a utilização de rotas incorretas.

Os protocolos SAODV e ARAN são vulneráveis ao ataque de consumo de recursos *Routing Table Overflow* pois utilizam criptografia assimétrica para autenticação entre os nós [HUY04b] [ARG05]. Isto acontece pelo fato de que os nós maliciosos podem inundar a rede com pacotes de roteamento falsos ou com assinaturas falsas. Desta forma, como os nós legítimos não conseguem verificar as assinaturas com uma velocidade suficiente, a maioria dos pacotes recebidos é descartada, degradando o funcionamento da rede.

5.2 Simulador e Script de Análise

O NS-2 (*Network Simulator* 2) é um *software* de simulação de redes que permite a simulação de protocolos para LANs e WANs. Foi desenvolvido na Universidade de *Berkeley* utilizando as linguagens de programação C++ e OTcl (*Object Tool Command Language*), sendo um *software* orientado a objeto. O NS-2 permite a simulação de protocolos de rede como o TCP e UDP, comportamento de tráfego em FTP, Telnet, Web e CBR (utilizado neste trabalho) e utilização de vários mecanismos de gerenciamento de filas como FIFO, RED e CBQ. O NS-2 implementa o padrão 802.11 e a pilha de protocolos inclui protocolos de roteamento como o DSR, DSDV e AODV.

Em [HAG05] é apresentado um estudo no qual é comparado os resultados gerados pelos simuladores de rede NS-2 e GloMoSim com os resultados obtidos em uma rede *ad hoc* real composta por *laptops* equipados com placas no padrão 802.11b. O simulador NS-2 apresentou bons resultados, principalmente com a utilização de taxas de transmissão superiores a 11Kbps. Foi verificado também que a simulação de topologias dinâmicas num ambiente real é uma tarefa complexa, sendo que sem os simuladores

seria quase impossível a realização de diversos trabalhos na área de redes sem fio *ad hoc*.

Uma visão simplificada do NS-2 é apresentada na Figura 21. Para a execução de uma simulação, é necessário informar para o simulador um *script* de configuração escrito na linguagem OTcl, no qual estão definidas as propriedades do ambiente e a localização dos arquivos de tráfego e movimentação dos nós (no caso de redes sem fio *ad hoc*).

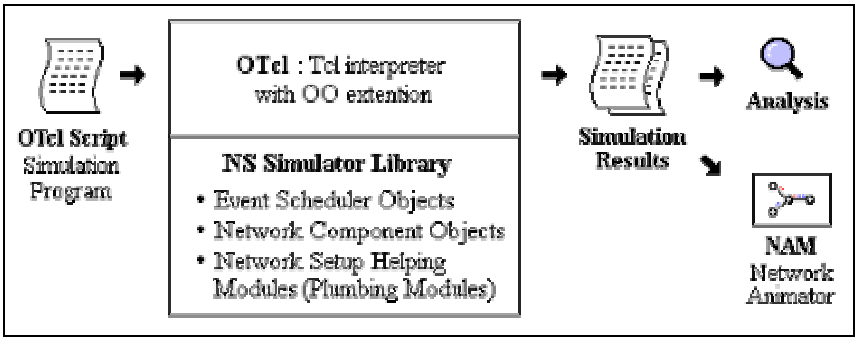


Figura 21 - Visão simplificada do simulador NS-2 [CHU02]

Quando uma simulação é realizada, o NS-2 produz um ou mais arquivos de saída (*trace*) que contém as informações sobre a simulação. O nível das informações gravadas nos arquivos de *trace* depende do *script* de configuração. Estes dados podem ser utilizados para análise de simulações ou como entrada para uma ferramenta de simulação gráfica.

Nesta dissertação foi utilizada a versão 2.29 do simulador NS-2 para realização das simulações. Para analisar os arquivos de *trace* gerados pelo NS-2 a cada nova simulação, foi desenvolvido um *script* (ver Anexo A) na linguagem PERL baseado nos *scripts* utilizados em [CUR03]. Este *script* identifica os diferentes tipos de eventos registrados pelo NS-2 e calcula as métricas apresentadas no item 5.3.

5.3 Métricas para Análise dos Protocolos

Para que seja possível comparar o desempenho entre protocolos e analisar o funcionamento dos mesmos, são necessárias algumas métricas para facilitar este processo. A RFC 2501 [COR99] recomenda que sejam utilizadas a métrica latência e a

métrica vazão para verificar a eficácia da política de roteamento, coletadas sob uma perspectiva externa de outras políticas que utilizam o roteamento. Recomenda também que seja utilizada a métrica razão de entrega de pacotes, possibilitando uma avaliação da eficiência do roteamento sob uma perspectiva interna. Neste trabalho, as seguintes métricas foram utilizadas:

1. **Latência Média Fim-a-Fim**

Calculada como sendo o tempo médio que os pacotes de dados do tipo CBR levam para ir de um ponto a outro da rede. A latência é calculada considerando o tempo em que o pacote chega ao canal de comunicação através do nó origem e a observação do recebimento do último *bit* do pacote no nó destino. Inclui por exemplo, o tempo de aquisição de rota, codificação e decodificação na interface de rede e processamento nos nós intermediários [SAN02]. Quanto menor for a latência, menor será o tempo de resposta da rede possibilitando uma melhor interatividade na comunicação entre dois nós.

2. **Razão de Entrega de Pacotes de Dados**

Razão entre o total de pacotes de dados recebidos pelos nós de destino e o total de pacotes enviados pelos nós de origem. Esta métrica é freqüentemente utilizada na análise de protocolos de roteamento, pois demonstra se o protocolo está atingindo realmente seu objetivo principal que é garantir que os pacotes cheguem ao seu destino [BRO98].

3. ***Overhead* de Roteamento (pacotes)**

Quantidade total de pacotes de roteamento transmitidos pela rede. Por exemplo, um pacote RREQ retransmitido por três nós será contado três vezes nesta métrica [HUY02a]. Esta é uma importante métrica para comparação entre protocolos, pois avalia a escalabilidade do protocolo, o seu funcionamento em uma rede congestionada e sua eficiência em termos de consumo de energia dos nós. Protocolos que enviam uma grande quantidade de pacotes de roteamento podem aumentar a probabilidade de colisão entre pacotes e aumentar a latência de pacotes de dados [BRO98]. Neste trabalho, os

pacotes de roteamento são do tipo RREQ, RREP e RERROR os quais são utilizados pelos protocolos reativos.

4. ***Overhead* de Roteamento (bytes)**

Semelhante a métrica anterior, representa a quantidade em bytes do total de dados de roteamento transmitidos pela rede durante o descobrimento e manutenção das rotas [HUY02a]. Demonstra o impacto no *overhead* causado pela adição de novas informações no cabeçalho dos pacotes de roteamento.

5. **Descarte de Pacotes**

É o índice que mede a quantidade de pacotes descartados durante a transmissão de pacotes por todos os nós que compõem a rede. Quanto menor a perda de pacotes, maior a eficiência da rede. Um pacote pode ser descartado por várias razões, como por exemplo quando a fila na interface de rede dos nós atingir seu limite, quando não existir uma rota para o destino, quando a rede estiver muito sobrecarregada, etc.

6. **Vazão**

Também denominada de *throughput*, é utilizada para analisar a transmissão de dados na rede e determina a quantidade de dados movida entre cada um dos nós que compõem a rede durante o tempo total da simulação. Na maioria das redes esta métrica sofre variações no decorrer no tempo e nesta dissertação é medida em Kbps (kilobits por segundo).

5.4 Comparação de desempenho entre Ariadne e BFTR

Dentre os protocolos de roteamento seguro estudados, apenas os autores dos protocolos Ariadne, BFTR e SEAD disponibilizaram o código implementado no simulador NS-2 para a realização deste trabalho. O código fonte dos protocolos Ariadne e SEAD são originários do projeto *Monarch* [MON04a], desenvolvido na CMU (*Carnegie Mellon University*). Já o código do BFTR foi obtido diretamente com os autores [XUE04], membros do *Monet Research Group* [MON04b] da UIUC (*University of Illinois at Urbana-Champaign*).

Como parte inicial desta dissertação, foram realizadas simulações para comparar o desempenho entre os protocolos Ariadne, BFTR e SEAD, cujos resultados são demonstrados em [VIV06b]. No item 5.4.2, será apresentada apenas a comparação de desempenho entre os protocolos Ariadne e BFTR em relação às métricas latência e razão de entrega de pacotes.

5.4.1 Ambiente de Simulação

Para realizar os experimentos com os protocolos *Ariadne* e BFTR foi utilizado o simulador NS-2 (*Network Simulator 2*), o qual tem sido utilizado extensivamente para a análise do desempenho dos protocolos de roteamento de redes *Ad Hoc*. A Tabela 4 apresenta os parâmetros utilizados nas simulações onde podemos observar quais foram os parâmetros que sofreram variações e quais permaneceram fixos ao longo das simulações.

Nas simulações foi utilizado o modelo de propagação *Two Ray Ground Reflection* que implementa a influência de fenômenos físicos como força do sinal, atraso na propagação e interferência no comportamento de uma rede sem fio. O protocolo de acesso ao meio utilizado é o IEEE 802.11 DCF (*Distributed Coordination Function*). A configuração da interface sem fio dos nós segue a especificação do dispositivo *Lucent 914MHz WaveLAN.*

Tabela 4 - Parâmetros utilizados nas simulações com os protocolos Ariadne e BFTR

Tipo de canal: *Wireless Channel*	**Tipo de tráfego:** Dados – CBR
Modelo de propagação: *Two Ray Ground*	**Largura de banda:** 2 Mbps
Tipo de interface: *Wireless Phy*	**Protocolos:** Ariadne, BFTR
Camada PHY: 802.11b	**Número de estações móveis:** 50
Tipo de camada de ligação: LL	**Tamanho dos pacotes:** 512 bytes
Modelo de antena: Omni Antenna	**Tipo de Fila:** FIFO
Dimensões do ambiente: 1500 X 300m	**Número máximo de pacotes na fila:** 50
Modelo de mobilidade: *Random Trip Model*	**Tempo de pausa:** 0s, 30s, 60s, 120s, 300s, 600s
Protocolo de conexão: UDP	**Tempo de simulação:** 600s

Os nodos deslocam-se respeitando os limites da rede de acordo com o modelo de mobilidade individual *Random Trip Model* [BOU05], no qual o percurso de cada nodo é caracterizado por períodos de movimentação e de pausa. O *Random Trip Model* é um modelo de mobilidade que generaliza o modelo *Random Waypoint* [JOH96] para cenários realísticos de simulação. Durante a simulação de funcionamento de uma rede, existe um tempo inicial para que a rede simulada atinga um estado próximo a de uma rede real. Entretanto, o *Random Trip Model* permite que a simulação de uma rede sem fio inicie em um estado próximo do ideal, melhorando a qualidade dos resultados dos experimentos. Nos padrões utilizados determinou-se que a velocidade de movimentação de um nó tem um valor definido aleatoriamente entre 0 m/s e 20 m/s. Os arquivos que definem a movimentação dos nodos no simulador NS-2 foram gerados utilizando os scripts implementados em [BOU05].

Este modelo de simulação exemplifica um ambiente *wireless multihop* com dimensões de 1500 X 300m, onde existem 50 nodos que se comunicam com o auxilio de algum nodo intermediário, quando necessário. A dimensão da rede será retangular, pois esta característica implica em um aumento na média de saltos durante o roteamento em relação a uma área quadrangular como demonstrado em [BRO98] [HUY02a], gerando um ambiente mais dinâmico. Não foram utilizados nodos maliciosos na rede.

O tempo de pausa será de 0s, 30s, 60s, 120s, 300s, 600s para avaliar o comportamento do protocolo em relação a diferentes padrões de mobilidade. O tempo de pausa de 0s representa a movimentação constante dos nodos (mobilidade alta) e o tempo de pausa de 600s indica que os nodos ficam fixos no lugar (mobilidade baixa/nula).

Os pacotes têm tamanho de 512 bytes, simulando apenas tráfego do tipo CBR (*Constant Bit Rate*), que são enviados por um canal utilizando a largura de banda de 2 Mbps. Nestes experimentos, foi utilizada a configuração padrão do simulador NS-2, que utiliza o padrão FHSS 802.11b para a camada física, cujos parâmetros (Tabela 5) estão definidos no primeiro *draft* do padrão 802.11b [IEE99a].

O tempo de cada simulação foi de 600 segundos. Para cada tempo de pausa, a execução das simulações foi replicada vinte vezes para garantir maior confiabilidade estatística aos resultados. Para permitir a comparação entre os protocolos, foram

utilizados os mesmos cenários de movimentação e comunicação para cada réplica das simulações.

Tabela 5 – Parâmetros de configuração do padrão FHSS 802.11b

CWMin_ 15	dataRate_ 2.0e6
CWMax_ 1023	basicRate_ 1.0e6
SlotTime_ 50us	ShortRetryLimit_ 7
SIFS_ 28us	LongRetryLimit_ 4
PreambleLength_ 96	RTSThreshold_ 0
PLCPHeaderLength_ 32	CCAtime_ 27us
PLCPDataRate_ 1.0e6	

.

Os dados são enviados através de uma conexão UDP (*User Datagram Protocol*) com uma taxa de transmissão de aproximadamente 16 Kbps. O tempo de duração e ativação de cada conexão foi estabelecido de maneira aleatória, assim cada uma tem a mesma probabilidade de ocorrer a qualquer momento. Os arquivos de tráfego que definem a troca de dados entre os nodos foram gerados pelo *script* cbrgen.tcl desenvolvido pela CMU e disponibilizado juntamente com a distribuição do NS-2 [FAL07].

Foi utilizada uma fila do tipo FIFO, na qual os pacotes vão sendo armazenados na ordem em que eles chegam, e assim que possível são enviados nesta mesma ordem. Se chegarem pacotes depois da fila ter atingido o seu limite, estes serão descartados. A decisão sobre a prioridade de atendimento é determinada pela ordem de chegada. O comprimento da fila foi fixado em no máximo 50 pacotes.

5.4.2 Resultados e Análises

No Gráfico 1 e no Gráfico 2 é apresentado uma comparação de desempenho entre os protocolos Ariadne e BFTR, em relação as métricas latência média fim-a-fim e razão de entrega de pacotes de dados. Uma comparação mais detalhada incluindo métricas como pacotes descartados e vazão é apresentada em [VIV06b].

O bom desempenho do protocolo BFTR para a métrica latência em relação ao protocolo Ariadne foi gerado pelo algoritmo de análise e descobrimento da melhor rota

entre os nós. Assim, com a utilização do protocolo BFTR os nós utilizaram rotas mais curtas e conseqüentemente o envio e recebimento de pacotes teve uma diminuição na latência.

Os protocolos Ariadne e BFTR apresentaram variações na latência com o aumento do tempo de pausa na movimentação dos nós. Como pode ser observado no Gráfico 1, os protocolos *Ariadne* e BFTR apresentaram comportamentos distintos durante toda a simulação apesar de ambos utilizarem em sua implementação o protocolo DSR.

Com o aumento do tempo de pausa dos nós, a latência do protocolo Ariadne diminuiu ao contrário do protocolo BFTR cuja latência aumentou. Este comportamento do BFTR pode ser resultado da diminuição na mobilidade da rede. Pois dessa forma houve uma redução na quantidade de rotas alternativas que poderiam ser analisadas pelo algoritmo de descobrimento da melhor rota do protocolo BFTR, as quais poderiam ser utilizadas pelos nós durante o envio e recebimento de dados.

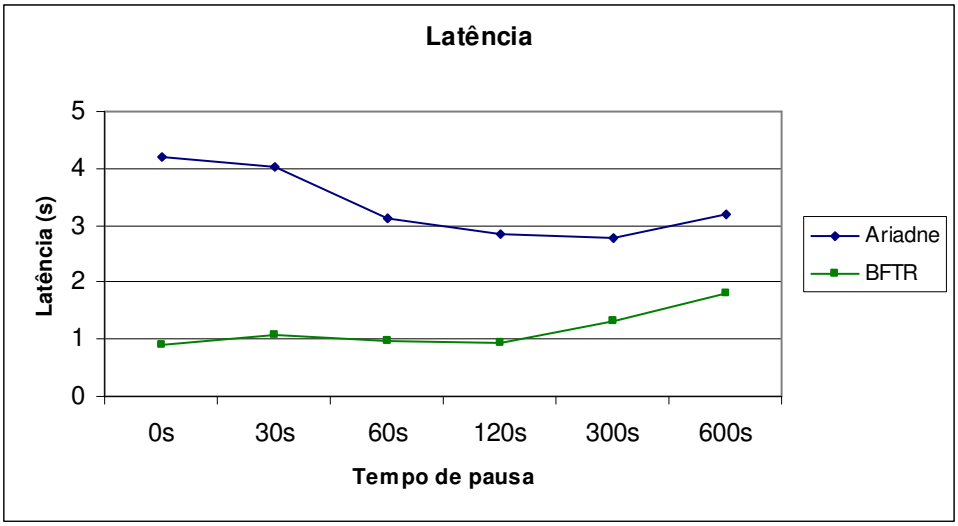


Gráfico 1 – Protocolos Ariadne e BFTR - Latência x Tempo de Pausa

Ambos os protocolos apresentaram uma alteração na tendência da métrica latência quando os nós apresentaram uma mobilidade nula, isto é, com tempo de pausa de 600s. Isto pode ser explicado pelo fato de que a ausência total de mobilidade prejudicou o processo de descoberta de novas rotas. Nesta situação, é praticamente utilizado o mesmo conjunto de rotas durante toda a simulação. Então, caso as rotas com

um desempenho baixo não apresentem uma interrupção no roteamento, serão utilizadas durante toda a simulação, podendo resultar em um aumento da latência média da rede.

A métrica razão de entrega de pacotes de dados é freqüentemente utilizada na análise de protocolos de roteamento, pois demonstra se o protocolo está atingindo realmente seu objetivo principal que é garantir que os pacotes chegam ao seu destino. De modo geral, o protocolo BFTR também apresentou um melhor desempenho em relação a esta métrica. Isto pode ser explicado pelo fato do algoritmo de seleção de rotas do BFTR utilizar esta métrica como parâmetro para selecionar as melhoras rotas. Este mecanismo dá prioridade para as rotas mais curtas e que possuam uma razão de entrega de pacotes elevada, garantindo um bom desempenho no roteamento.

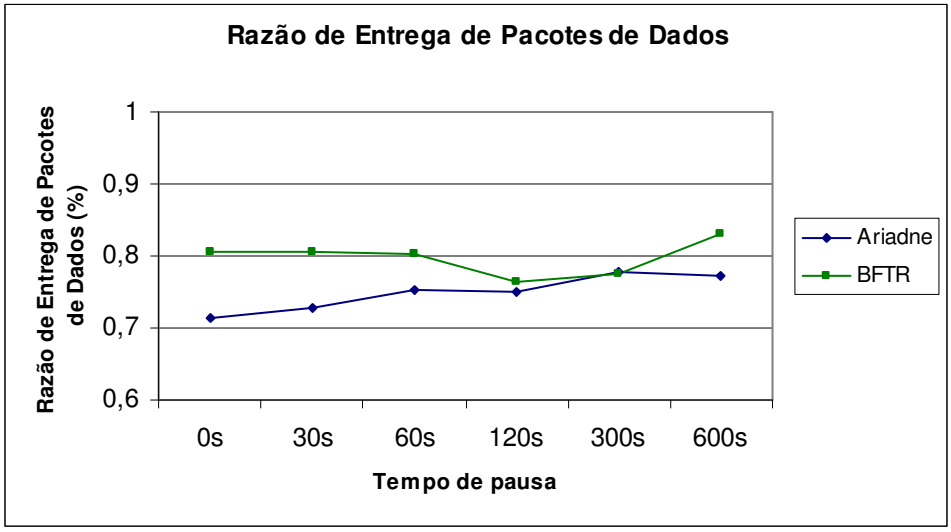


Gráfico 2 - Protocolos Ariadne e BFTR - Razão de Entrega de Pacotes de Dados x Tempo de Pausa

5.5 Sumário

O projeto de um protocolo de roteamento seguro tem como meta proteger a rede *ad hoc* contra ataques em específico ou a um conjunto de ataques. Este protocolo deve apresentar um balanceamento correto entre desempenho e segurança, pois como a tarefa de roteamento é essencial na rede, os mecanismos de segurança não podem limitar esta operação.

Como discutido no capítulo 3, os protocolos de roteamento reativo tem apresentado um desempenho superior em relação aos protocolos pró-ativos. Esta

tendência também foi detectada entre aos protocolos de roteamento seguro Ariadne, BFTR e SEAD, nas simulações realizadas em [VIV06b], principalmente em relação à métrica razão de entrega de pacotes. Nas simulações realizadas em [YOV04], o protocolo Ariadne também apresentou um desempenho superior nesta métrica, quando comparado com o SEAD, em um ambiente composto por clusters e com taxa de transmissão de 19,2 Kbps.

Todos os protocolos de roteamento seguro apresentados são vulneráveis aos ataques *Wormhole* e o *Rushing Attacks*. Para solucionar este problema, foram propostos os mecanismos *Packet Leashes* [HUY03A] e RAP [HUY03b].

Como apresentado neste capítulo, os protocolos Ariadne e SAODV utilizam algumas técnicas (Tabela 1) para garantir segurança no processo de descoberta de rotas, cujo princípio de funcionamento é baseado no DSR. Desta forma, os mecanismos de descoberta e seleção de rotas implementados no BFTR poderiam ser adaptados nestes protocolos como uma alternativa para melhorar o desempenho no roteamento e garantir uma proteção contra o ataque *Blackhole*.

Na comparação de desempenho entre os protocolos Ariadne e BFTR, o protocolo BFTR teve uma latência inferior em relação ao protocolo Ariadne devido ao algoritmo de descoberta e análise da melhor rota. Em relação à métrica razão de entrega de pacotes o BFTR também alcançou um comportamento pouco superior ao obtido pelo Ariadne.

A possibilidade de melhoria no processo de descoberta de rotas do protocolo Ariadne baseado nos algoritmos do BFTR e os resultados obtidos na comparação de desempenho entre o Ariadne e o BFTR (ver item 5.4), são as motivações para o próximo capítulo.

6 Ariadne-BFT

Neste capítulo será proposto o Ariadne-BFT (Ariadne *Best-effort Fault-Tolerant*), uma extensão do protocolo Ariadne, baseada nos algoritmos utilizados no protocolo BFTR para descoberta e seleção das rotas. O objetivo desta extensão é melhorar o desempenho do Ariadne no roteamento de pacotes e garantir uma proteção contra o ataque *Blackhole*.

Os protocolos Ariadne e BFTR foram implementados utilizando como base o protocolo reativo DSR e desta forma ambos herdaram uma estrutura semelhante de funcionamento. Apesar disso, estes protocolos utilizam conceitos de segurança distintos.

Para garantir segurança no processo de roteamento, o Ariadne utiliza criptografia simétrica para verificar a autenticidade e integridade das mensagens de roteamento enviadas entre os nós. Já no protocolo BFTR são implementados algoritmos que utilizam a redundância de rotas entre os nós para manter o desempenho no roteamento mesmo com a presença de nós maliciosos.

6.1 Considerações sobre os Protocolos Ariadne e BFTR

Como apresentado em 4.3.1, o protocolo Ariadne utiliza criptografia simétrica para garantir autenticidade e integridade no processo de descoberta e manutenção de rota. Desta forma impede que algum nó intermediário possa remover um nó presente na lista de nós do pedido de rota ou nas mensagens de resposta de rota. O Ariadne analisado utiliza o mecanismo de autenticação TESLA para executar a autenticação *broadcast* entre os nós da rede.

Apesar de o protocolo Ariadne realizar a autenticação na troca de mensagens entre os nós através do TESLA, o processo de roteamento continua vulnerável ao ataque *Blackhole* (ver Tabela 3). Como apresentado no item 5, uma solução para este problema seria a utilização de um mecanismo para aproveitar a redundância de rotas existentes em uma rede *ad hoc*.

Na Figura 22 são apresentadas duas redes compostas por nós maliciosos, nas quais o nó de origem S deseja enviar um pacote para o nó de destino D. Na rede (a),

todas as rotas entre o nó S e D passam por um nó malicioso. Neste cenário, nenhum protocolo possui uma solução para este problema [XUE04]. Já a rede (b) apresenta uma rota confiável, através da qual os pacotes poderiam ser enviados corretamente.

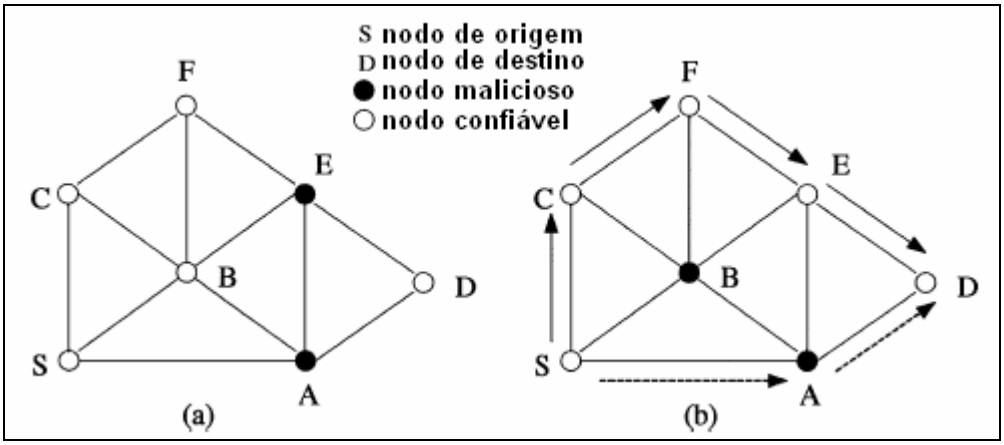


Figura 22 - Rede *ad hoc* composta por nós maliciosos [XUE04]

O protocolo Ariadne, da mesma forma que o DSR, não explora a redundância de rotas, pois:

1. Não identifica todas as rotas existentes entre os nós no processo de descoberta de rota, devido ao descarte de pacotes RREQ duplicados.
2. Não considera o comportamento da rota (isto é, o desempenho na entrega de pacotes) durante a escolha de rota. Por exemplo, na rede (b) da Figura 22 o protocolo Ariadne poderia escolher a rota (S → A → D) ao invés da rota confiável (S → C → F → E → D), mesmo que esta tenha sido descoberta.

Em contrapartida, no protocolo BFTR os mecanismos de descoberta e seleção de rotas utilizam algoritmos que consideram a redundância de rotas para manter o desempenho no roteamento. Tendo um conjunto de rotas obtidas pelo mecanismo de descoberta de rota, o nó de origem determina localmente qual a melhor rota a ser utilizada (ver item 4.3.2), priorizando as rotas que apresentam uma alta razão de entrega de pacotes. Desta forma, o BFTR garante uma proteção contra o ataque *Blackhole*, pois o comportamento das rotas é analisado constantemente.

Foi demonstrado pelos autores do BFTR [XUE04], que mesmo quando uma rede possui uma densidade alta de nós maliciosos, a probabilidade de existência de rotas

confiáveis se mantém elevada. Sendo assim, o BFTR utiliza o desempenho fim-a-fim da transmissão de pacotes para analisar e selecionar as rotas que apresentam a melhor razão de entrega de pacotes e desta forma manter o desempenho no roteamento da rede, mesmo com a presença de nós maliciosos.

O protocolo BFTR não exige autenticação na troca de mensagens entre os nós intermediários de uma determinada rota, apenas entre o nó de origem e o nó de destino [XUE04]. Esta característica limita a proteção no roteamento contra o ataque *Routing Table Poisoning*. Isto permite que os nós intermediários possam alterar ou remover nós da lista de nós do pacote RREQ [HUY02a] durante uma descoberta de rota.

Por outro lado, no protocolo Ariadne, toda a troca de mensagens entre os nós são autenticadas utilizando o TESLA (ver item 4.3.1). Cada nó que iniciar uma descoberta de rota deve criar uma cadeia de *hash*, cujos valores serão utilizados como chaves para o cálculo do MAC das informações do pacote RREQ, garantindo que nenhum salto tenha sido omitido ou alterado durante o caminho até o nó alvo do processo de descoberta de rota.

6.2 Proposta de melhoria no protocolo Ariadne

Como afirmado pelos autores do protocolo BFTR [XUE04] e analisado neste trabalho, o protocolo Ariadne apresenta uma proteção mais robusta em relação ao BFTR, contra ataques no roteamento de uma rede *ad hoc*, pois garante a autenticidade e integridade na troca de mensagens entre os nós. Desta forma, optou-se por modificar o processo de descoberta e seleção de rotas do protocolo Ariadne com base nos algoritmos utilizados no BFTR e criar uma extensão deste protocolo, batizada de Ariadne-BFT.

Como demonstrado nas simulações do item 5.4, o protocolo BFTR apresentou os melhores valores nas métricas latência e razão de entrega de pacotes em relação ao Ariadne. Este resultado pode ser explicado pelo fato do algoritmo de seleção de rotas do BFTR dar prioridade para as rotas mais curtas e que possuam uma razão de entrega de pacotes elevada, proporcionando um bom desempenho no roteamento.

Portanto, o objetivo do Ariadne-BFT é melhorar o desempenho do Ariadne com base na redundância de rotas - característica intrínseca das redes *ad hoc* - utilizando os algoritmos de descoberta e seleção de rota (Figura 16 e Figura 17) implementados no BFTR. Desta forma, o Ariadne-BFT também garante uma proteção contra o ataque *Blackhole*, pois o comportamento das rotas é analisado constantemente.

6.3 Características do Ariadne-BFT

A implementação do Ariadne-BFT foi realizada utilizando como base o código do protocolo Ariadne [HUY02a] desenvolvido como parte do projeto *Monarch* [MON04a] e o código gerado por [XUE04] na definição do protocolo BFTR. Ambos foram implementados no simulador NS-2.

O Ariadne-BFT manteve inalterado todo o processo de autenticação e verificação de integridade realizada durante a troca de mensagens entre os nós. Da mesma forma, foram mantidos os parâmetros de configuração do DSR (Tabela 6), sobre o qual foi desenvolvido o Ariadne e do TESLA (Tabela 7), ambos definidos em [HUY02a]. Estes parâmetros foram mantidos para permitir uma verificação mais precisa do impacto causado no desempenho com as modificações realizadas no Ariadne-BFT em relação ao Ariadne original.

Tabela 6 – Protocolo Ariadne-BFT - Parâmetros do DSR

Parâmetro	Valor
Timeout inicial do pacote RREQ	2 segundos
Timeout máximo do pacote RREQ	40 segundos
Tamanho da *cache*	32 rotas
Política de reposição da *cache*	FIFO

Tabela 7 – Protocolo Ariadne-BFT - Parâmetros do TESLA

Parâmetro	Valor
TESLA *ti(time interval)*	1 segundo
Tempo pessimista de propagação fim-a-fim (τ)	0,2 segundos
Tempo máximo de erro na sincronização (Δ)	0,1 segundos
Tamanho do *hash*	80 *bits*

Na Tabela 8 são apresentados os parâmetros utilizados pelos algoritmos do BFTR durante a análise das rotas (ver Figura 17). O parâmetro que representa a razão de entrega de pacotes esperada ($p0$), definido em 0,95 pelos autores do BFTR [XUE04], neste trabalho foi alterado para 0,90.

Tabela 8 – Protocolo Ariadne-BFT - Parâmetros BFTR

Parâmetro	Valor
Número de pacotes que serão analisados ($n0$)	20
Razão de entrega de pacotes esperada ($p0$)	0,90
Probabilidade de erro (α)	0,01

Os valores definidos no BFTR foram definidos com base numa série de simulações, através das quais foi verificado que estes seriam os valores mais adequados para a análise das rotas [XUE04]. Entretanto, devido ao *overhead* imposto pela autenticação e verificação dos pacotes no Ariadne, a razão de entrega de pacotes tende a ser naturalmente menor que atingida no protocolo BFTR. Desta forma optou-se por diminuir o valor de $p0$ no Ariadne-BFT, para permitir que rotas com um desempenho menor também sejam consideradas rotas confiáveis. Na finalização do protocolo Ariadne-BFT, foram realizadas várias simulações de teste, nas quais foi observada uma significativa melhora no desempenho com esta alteração. O mesmo não aconteceu com o protocolo BFTR.

Ao contrário do Ariadne que identifica apenas um sub-conjunto de rotas devido a supressão de pacotes RREQ duplicados, o Ariadne-BFT identifica todas as possíveis rotas entre os nós, as quais são utilizadas no algoritmo de seleção de rotas (Figura 16). Para isto, a descoberta de rota foi modificada, para não utilizar supressão de RREQ duplicados. Assim, para cada RREQ recebido, o nó destino deve enviar um pacote RREP pelo mesmo caminho utilizado pelo RREQ recebido, possibilitando a existência de múltiplas rotas entre os nós.

As modificações propostas no Ariadne-BFT são do tipo melhor esforço e não garantem a melhor performance no roteamento. Da mesma forma que no BFTR, a probabilidade da entrega de pacotes falhar é diretamente proporcional ao aumento de nós maliciosos na rede. Tal situação também poderá ocorrer quando a rede estiver muito sobrecarregada. Do ponto de vista do protocolo, estes dois casos não são distinguíveis,

mas ambos indicam que a rede não está propicia a atingir um bom desempenho no roteamento; nenhum protocolo irá trabalhar com eficiência nestas condições [XUE04].

6.4 Trabalhos Relacionados

Várias pesquisas tem sido realizadas na área de segurança no roteamento em redes *ad hoc*. Em [HUY02a] é apresentado o protocolo de roteamento seguro Ariadne, baseado no DSR e que utiliza criptografia simétrica para verificar a autenticidade e integridade das mensagens de roteamento enviadas entre os nós. Ariadne utiliza o mecanismo TESLA [PER05], um esquema eficiente de autenticação *broadcast* que requer uma sincronização fraca de tempo entre os nós. Este protocolo protege a rede contra os ataques de consumo de recursos (como por exemplo, *Replaying* e *Routing Table Poisoning*) mas não apresenta proteção contra o ataque *Blackhole*. O protocolo Ariadne-BFT incrementa a segurança no roteamento em relação ao protocolo Ariadne original, pois utiliza mecanismos contra o ataque *Blackhole* baseado nos algoritmos de descoberta e seleção de rotas do protocolo BFTR.

Em [XUE04] é apresentado o protocolo reativo BFTR, o qual implementa algoritmos para manter o desempenho no roteamento mesmo com a presença de nós maliciosos. O BFTR utiliza a redundância de rotas, identificando um conjunto de rotas entre os nós, as quais são constantemente analisadas considerando a razão de entrega de pacotes e a latência. As rotas que desviarem de um padrão pré-definido são descartadas. Desta forma, o BFTR protege a rede contra o ataque *Blackhole*. Entretanto o BFTR supõe apenas a autenticação na troca de mensagens entre o nó de origem e o nó de destino. Esta característica limita a proteção contra o ataque *Routing Table Poisoning* [HUY02a], permitindo que nós intermediários possam alterar ou remover algum nó da lista de nós do pacote RREQ durante uma descoberta de rota.

Recentemente foi proposto o mecanismo 2ACK [LIU06], que também foi desenvolvido sobre o protocolo DSR e utiliza a redundância de rotas (característica intrínseca das redes *ad hoc*) para proteger a rede contra nós maliciosos executando o ataque *Blackhole*. Ao contrário do BFTR que analisa o comportamento da rota, neste esquema são analisados os *links* entre os nós. A idéia principal do 2ACK é o envio de

dois pacotes ACK na direção oposta da rota. Para minimizar o overhead no roteamento, apenas uma fração dos pacotes recebidos exige o envio de pacotes ACK. Na comparação de desempenho com o BFTR, o 2ACK apresentou uma melhora significativa na métrica razão de entrega de pacotes, mas apresentou um *overhead* superior no roteamento.

A utilização da redundância de rotas para proteger a rede contra ataques de interrupção no roteamento já havia sido discutida em [ZHO99], no qual é proposta a utilização múltiplas rotas para a transmissão de dados. Entretanto, esta técnica resulta em um elevado *overhead* de roteamento, devido à transmissão redundante de dados pela rede.

Em [HUY02b] é proposto um protocolo de roteamento pró-ativo baseado no DSDV chamado SEAD. O objetivo deste protocolo é proteger a rede contra ataques que modificam informações de roteamento transmitidas por *broadcast* durante a fase de atualização das tabelas de roteamento. Para isto, autentica os campos *sequence number* e o *metric* usando elementos da cadeia de *hash*. O SEAD supõe a existência de algum esquema para autenticação *broadcast* como o TESLA [PER05]. Assim como Ariadne, este protocolo não apresenta proteção contra o ataque *Blackhole*.

Em [ZAP02] [ZAP06] é descrito o protocolo SAODV, uma extensão segura do protocolo AODV para proteger a rede contra o ataque de nós maliciosos. Para isto utiliza assinatura digital para autenticar os campos imutáveis das mensagens de roteamento e cadeias de *hash* para garantir a autenticidade do campo *metric* durante uma descoberta de rotas. A criptografia assimétrica também é utilizada no ARAN [SAN02], um protocolo reativo que utiliza como base o protocolo AODV. Cada nó possui um certificado digital emitido por uma entidade confiável, o qual é utilizado para a garantir a autenticidade e não-repúdio na troca de mensagens. Entretanto, ambos os protocolos são vulneráveis ao ataque de consumo de recursos *Routing Table Overflow* pois a criptografia assimétrica consome muitos recursos dos dispositivos móveis [HUY04b] [ARG05].

O RAP (*Rushing Attack Prevention*) apresentado em [HUY03b] foi projetado para proteger os protocolos reativos (DSR, AODV, *Ariadne*, BFTR, ARAN, SAODV) contra o *Rushing Attack*. Quando integrado com protocolos de roteamento seguro, o RAP não impõe nenhum custo adicional a não ser que o protocolo seguro não consiga

encontrar rotas válidas. Quando o RAP está ativo, a descoberta de rota irá gerar um *overhead* adicional, mas as rotas confiáveis serão encontradas e os pacotes poderão ser transmitidos com segurança ao contrário dos outros protocolos. Entretanto, estes protocolos reativos continuam vulneráveis ao ataque *Wormhole*. Mas em [HUY03a] [HUY06] é proposto o mecanismo *Packet Leashes* para proteger a rede *ad hoc* contra o *Wormhole*. Para isto são adicionadas informações extras nos pacotes enviados, as quais são utilizadas no nó de destino para verificar se o pacote enviado utilizou uma rota cujo tamanho esteja de um padrão determinado.

Em [YOV04] é realizada uma comparação de desempenho entre os protocolos de roteamento seguro Ariadne e SEAD para o suporte na transmissão de dados multimídia em tempo real em redes *ad hoc* composta por clusters. São realizadas simulações para a coleta de métricas, considerando várias condições de tráfego e topologias de rede. Entretanto, é analisado apenas o impacto causado pela adição dos mecanismos de segurança no desempenho da rede, sem considerar o nível de segurança oferecido por esses protocolos.

6.5 Sumário

Neste capítulo, foi proposto o Ariadne-BFT, uma extensão do protocolo Ariadne baseada nos algoritmos utilizados no protocolo BFTR para descoberta e seleção das rotas. Esta extensão apresenta os mesmos princípios de funcionamento do Ariadne, entretanto, implementa mecanismos que utilizam a redundância de rotas da rede *ad hoc* para manter um bom desempenho no roteamento mesmo com a presença de nós maliciosos.

Da mesma forma que o protocolo Ariadne, o Ariadne-BFT protege a rede contra os ataques de consumo de recursos *Replaying, Routing Table Poisoning e Routing Table Overflow*. Além disso, esta extensão possui a vantagem de proteger a rede contra o ataque de interrupção no roteamento *Blackhole* como demonstrado na Tabela 9. Entretanto, assim como os protocolos Ariadne e BFTR, o Ariadne-BFT também deverá utilizar os mecanismos *Packet Leashes* [HUY03A] e RAP [HUY03b] para proteger a rede contra os ataques *Wormhole* e o *Rushing Attacks*.

Tabela 9 - Comparação entre o Ariadne-BFT e os protocolos Ariadne e BFTR

Ataques	Protocolos de Roteamento Seguro		
	Ariadne-BFT	**Ariadne**	**BFTR**
Wormhole	não	não	não
Rushing Attack	não	não	não
Blackhole	sim	não	sim
Replaying	sim	sim	sim
Routing Table Poisoning	sim	sim	sim[3]
Routing Table Overflow	sim	sim	sim

Nó próximo capítulo serão apresentadas às simulações realizadas utilizando o protocolo Ariadne-BFT para validar o seu funcionamento e comparar o seu desempenho em relação ao protocolo Ariadne original e ao BFTR.

[3] O protocolo BFTR não exige autenticação das mensagens de roteamento entre os nós intermediários, apenas entre o nó de origem e destino, diminuindo a proteção contra o ataque *Routing Table Poisoning*.

7 Análise de Desempenho do Ariadne-BFT

O objetivo deste capítulo é a realização de simulações para demonstrar o funcionamento do protocolo Ariadne-BFT e comparar o seu desempenho em relação ao protocolo Ariadne e ao BFTR. É apresentado o ambiente de simulação utilizado nos experimentos e em seguida são analisados os resultados dos experimentos através de gráficos gerados com base nos dados coletados para cada métrica.

7.1 Ambiente de Simulação

Da mesma forma que nos experimentos apresentados no item 5.4, foi utilizado o simulador NS-2 para analisar os protocolos *Ariadne,* BFTR e Ariadne-BFT. A Tabela 10 apresenta os parâmetros utilizados nas simulações onde podemos observar quais foram os parâmetros que sofreram variações e quais permaneceram fixos ao longo das simulações. Também foi utilizado o modelo de propagação *Two Ray Ground Reflection,* o protocolo de acesso ao meio IEEE 802.11 DCF (*Distributed Coordination Function*) e a configuração da interface sem fio do dispositivo *Lucent 914MHz WaveLAN*.

Tabela 10 - Parâmetros utilizados nas simulações

Tipo de canal: *Wireless Channel*	**Tipo de tráfego:** Dados – CBR
Modelo de propagação: *Two Ray Ground*	**Largura de banda:** 11 Mbps
Tipo de interface: *Wireless Phy*	**Protocolos:** Ariadne, BFTR, Ariadne-BFT
Camada PHY: 802.11b	**Número de estações móveis:** 50
Tipo de camada de ligação: LL	**Tamanho dos pacotes:** 512 bytes
Modelo de antena: Omni Antenna	**Tipo de Fila:** FIFO
Dimensões do ambiente: 1500 X 300m	**Número máximo de pacotes na fila:** 50
Modelo de mobilidade: *Random Trip Model*	**Tempo de pausa:** 0s, 30s, 60s, 120s, 300s, 600s
Protocolo de conexão: UDP	**Tempo de simulação:** 600s

Os nós deslocam-se respeitando os limites da rede de acordo com o modelo de mobilidade individual *Random Trip Model* [BOU05], no qual o percurso de cada nó é caracterizado por períodos de movimentação e de pausa. O *Random Trip Model* é um

modelo de mobilidade que generaliza o modelo *Random Waypoint* [JOH96] para cenários realísticos de simulação. Durante a simulação de funcionamento de uma rede, existe um tempo inicial para que a rede simulada atinga um estado próximo a de uma rede real. Entretanto, o *Random Trip Model* permite que a simulação de uma rede sem fio inicie em um estado próximo do ideal, melhorando a qualidade dos resultados dos experimentos. Nos padrões utilizados determinou-se que a velocidade de movimentação tem um valor escolhido aleatoriamente entre 0 m/s e 20 m/s. Os arquivos que definem a movimentação dos nós foram gerados utilizando *scripts* implementados em [BOU05].

Este modelo de simulação exemplifica um ambiente *wireless multihop* com dimensões de 1500 X 300m, onde existem 50 nós que se comunicam com o auxilio de algum nó intermediário, quando necessário. A dimensão da rede será retangular, pois esta característica implica em um aumento na média de saltos durante o roteamento em relação a uma área quadrangular como demonstrado em [BRO98] [HUY02a], gerando um ambiente mais dinâmico. Nestes experimentos, não foram utilizados nós maliciosos na rede.

Os pacotes têm tamanho de 512 bytes, simulando apenas tráfego do tipo CBR (*Constant Bit Rate*), que são enviados por um canal utilizando a largura de banda máxima do padrão IEEE 802.11b de 11 Mbps. O Simulador NS-2 é distribuído utilizando o padrão FHSS 802.11 [IEE99a] para a camada física, como foi apresentado e utilizado nos experimentos realizados no item 5.4.1. Entretanto, nestes experimentos para análise do protocolo Ariadne-BFT, a configuração do NS-2 foi alterada para utilizar o padrão HR/DSSS 802.11b (Tabela 11) cujos parâmetros estão definidos em [IEE99b].

Tabela 11 – Parâmetros de configuração do padrão HR/DSSS 802.11b

CWMin_ 31	dataRate_ 11.0e6
CWMax_ 1023	basicRate_ 1.0e6
SlotTime_ 20us	ShortRetryLimit_ 7
SIFS_ 10us	LongRetryLimit_ 4
PreambleLength_ 144	RTSThreshold_ 0
PLCPHeaderLength_ 48	CCAtime_ 15us
PLCPDataRate_ 1.0e6	RxTxTurnaroundTime_ 5us

O tempo de pausa será de 0s, 30s, 60s, 120s, 300s, 600s para avaliar o comportamento do protocolo em relação a diferentes padrões de mobilidade. O tempo de cada simulação foi de 600 segundos. Para cada tempo de pausa, a execução das simulações foi replicada trinta vezes. Para permitir a comparação entre os protocolos, foram utilizando os mesmos cenários de movimentação e comunicação para cada réplica das simulações.

Assim como nos experimentos apresentados no item 5.4, os dados são enviados através de uma conexão UDP (*User Datagram Protocol*) com uma taxa de transmissão de aproximadamente 16 Kbps. Os arquivos de tráfego que definem a troca de dados entre os nós também foram gerados pelo *script* cbrgen.tcl desenvolvido pela CMU e disponibilizado juntamente com a distribuição do NS-2 [FAL07].

7.2 Resultados e Análises

Neste item serão apresentadas as simulações realizadas com os protocolos Ariadne-BFT, Ariadne e BFTR, cujo objetivo é avaliar o funcionamento do Ariadne-BFT e comparar o seu desempenho principalmente em relação ao protocolo Ariadne original. De modo geral, os protocolos apresentaram um comportamento semelhante com a variação da mobilidade dos nós na rede, fato esperado pois todos utilizam como base o protocolo reativo DSR.

7.2.1 Latência Média Fim-a-Fim

A latência representa o tempo que um pacote leva para ir de um ponto a outro da rede. Desta forma, quanto menor a latência, melhor o tempo de resposta da rede. No Gráfico 3 são apresentados os valores da métrica latência média dos protocolos simulados.

O protocolo Ariadne-BFT apresentou uma diminuição significativa na latência média em relação ao protocolo Ariadne. Isto é resultado das modificações realizadas no processo de descoberta de rotas com base nos algoritmos utilizados pelo protocolo

BFTR. Ao contrário do Ariadne, o protocolo Ariadne-BFT identifica todas as possíveis rotas entre os nós, as quais são utilizadas no algoritmo de seleção de rotas (Figura 16).

Durante a execução deste algoritmo, um pacote é considerado enviado com sucesso quando é recebido com uma latência menor ou igual a *d* (ver item 4.3.2), cujo valor é estimado durante a fase de descobrimento de rota. Desta forma, as rotas que apresentam uma latência superior em relação às outras rotas serão evitadas, melhorando o desempenho no roteamento.

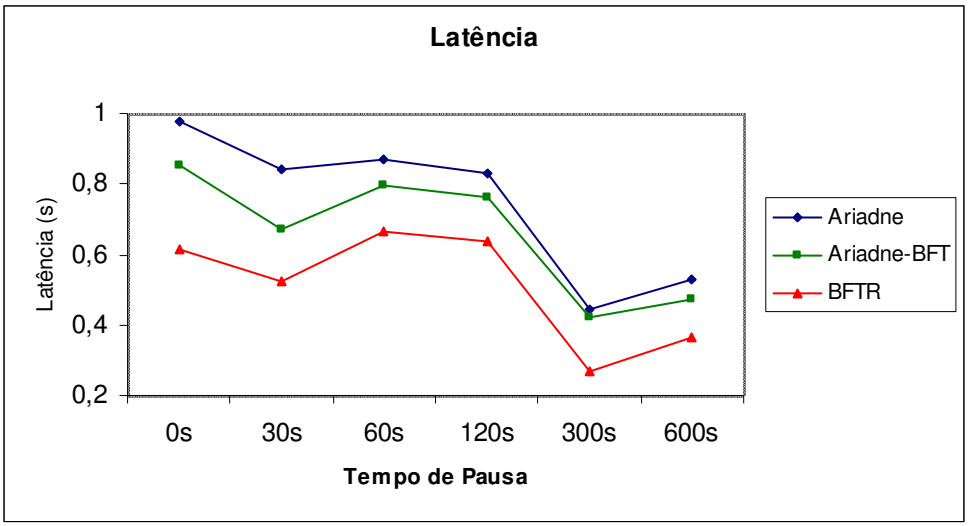


Gráfico 3 – Protocolos Ariadne, Ariadne-BFT e BFTR - Latência x Tempo de Pausa

Apesar das modificações propostas no Ariadne-BFT, este protocolo apresentou um desempenho inferior quando comparado com o protocolo BFTR. Assim como acontece com o protocolo Ariadne, esta diferença no desempenho é devido ao *overhead* gerado pelo processo de autenticação e verificação de todos os pacotes transmitidos entre os nós que compõem a rede.

Todos os protocolos apresentaram uma alteração na tendência da métrica latência quando os nós apresentaram uma mobilidade nula, isto é, com tempo de pausa de 600s a rede apresentou um aumento na latência. Este comportamento também foi verificado nos experimentos apresentados no item 5.4 em relação aos protocolos Ariadne e BFTR. Isto pode ser explicado pelo fato de que a ausência total de mobilidade prejudicou o processo de descoberta de novas rotas, definido pelo protocolo DSR. Nesta situação, é praticamente utilizado o mesmo conjunto de rotas durante toda a simulação

pois a localização dos nós se mantem a mesma. Então, caso as rotas com um desempenho baixo não apresentem uma interrupção no roteamento, serão utilizadas durante toda a simulação, podendo resultar em um aumento da latência média da rede.

De modo geral, para todos os tempos de pausa, os protocolos Ariadne e BFTR apresentaram valores menores para a latência do que os obtidos nos experimentos realizados no item 5.4.2. Esta diferança deve se ao fato de que nesta comparação de desempenho foi utilizada a camada física definida pelo padrão HR/DSSS 802.11b (Tabela 11) e não o padrão FHSS 802.11b (Tabela 5).

7.2.2 Razão de Entrega de Pacotes de Dados

A métrica razão de entrega de pacotes de dados é uma das principais métricas utilizadas na verificação de desempenho de um protocolo, pois demonstra com qual eficiência os pacotes estão sendo recebidos pelos nós de destino [COR99]. Como pode ser visualizado no Gráfico 4, todos os protocolos apresentaram uma melhora na razão de de entrega de pacotes de dados com o aumento no tempo de pausa dos nós. Isto é justificado pois a diminuição na mobilidade dos nós resultou em uma quantidade menor de alterações das rotas e conseqüentemente gerou uma redução na execução do procedimento de descoberta de novas rotas.

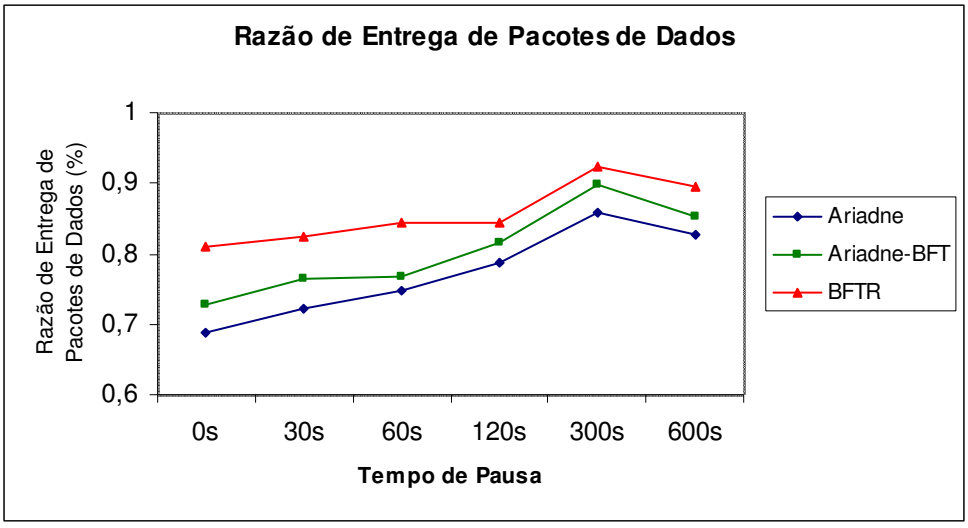


Gráfico 4 - Protocolos Ariadne, Ariadne-BFT e BFTR - Razão de Entrega de Pacotes de Dados x Tempo de Pausa

Com o tempo de pausa de 600s (mobilidade nula) houve uma queda na razão de entrega de pacotes. Para o mesmo tempo de pausa, a rede apresentou um aumento na latência média (ver item 7.2.1), o que resultou em uma diminuição na taxa de entrega de pacotes de dados.

O protocolo Ariadne-BFT atingiu um desempenho superior ao Ariadne na entrega de pacotes. Da mesma forma que na métrica latência, o desempenho do protocolo Ariadne-BFT em relação ao Ariadne pode ser justificado pelas modificações realizadas no processo de descoberta de rotas e da utilização do algoritmo de teste e seleção de rotas (Figura 17). Assim como o BFTR, o Ariadne-BFT também explora a redundância de rotas na rede. Dado um conjunto de rotas entre um nó de origem e de destino, serão utilizadas preferencialmente as rotas mais curtas e que possuam uma razão de entrega de pacotes elevada, melhorando o desempenho na entrega de pacotes.

7.2.3 Overhead de Roteamento (pacotes)

No Gráfico 5 é apresentado o comportamento dos protocolos em relação a métrica *overhead* de roteamento (pacotes). Esta métrica representa a quantidade de pacotes de roteamento transmitidos na rede durante a descoberta e manutenção de rotas. Como os protocolos analisados são do tipo reativo, nesta métrica são considerados apenas os pacotes RREQ, RREP e RERROR. São considerados também os pacotes ACK enviados pelos nós de destino nos protocolos Ariadne-BFT e BFTR.

O protocolo Ariadne-BFT apresentou um *overhead* maior comparado com o protocolo Ariadne pois utilizou uma quantidade maior de pacotes de roteamento. Isto ocorre porque durante a descoberta de rotas, o protocolo Ariadne-BFT não utiliza a supressão de pacotes RREQ duplicados, pois o objetivo é identificar todas as possíveis rotas entre os nós. Além disso, sempre que um pacote for recebido com sucesso, o nó de destino retorna um pacote ACK utilizado para calcular a latência da rota. As rotas são constantemente analisadas e as rotas consideradas não confiáveis são descartadas. Desta forma, processo de descoberta de rotas é sempre reiniciado pelo Ariadne-BFT quando todas as rotas entre o nó de origem e destino forem descartadas, gerando pacotes de roteamento adicionais.

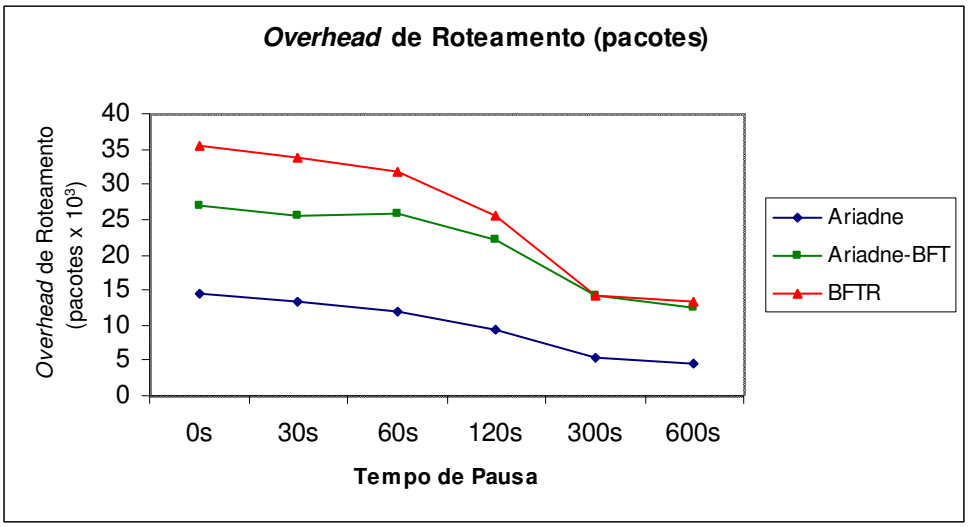


Gráfico 5 - Protocolos Ariadne, Ariadne-BFT e BFTR – *Overhead* de Roteamento x Tempo de Pausa

O Ariadne-BFT utilizou uma quantidade menor de pacotes de roteamento em relação ao protocolo BFTR. Isto pode ser justificado pela diminuição do valor do parâmetro ($p0$) no algoritmo de análise de rotas do protocolo Ariadne-BFT (ver item 6.3). Como o Ariadne-BFT utilizou um valor menor para a razão de entrega de pacotes esperada ($p0$), isto diminuiu a execução do processo de descoberta de rotas e conseqüentemente reduziu a quantidade de pacotes de roteamento.

Quando a rede apresentou uma baixa mobilidade (tempos de pausa de 120s, 300s e 600s), todos os protocolos apresentaram uma diminuição no envio de pacotes de roteamento. Este comportamento ocorre pois estes protocolos são do tipo reativo. Desta forma, quando os nós apresentam uma baixa movimentação, as rotas são alteradas com menos freqüência, diminuindo a execução do procedimento de descoberta de rotas.

7.2.4 Overhead de Roteamento (*bytes*)

Esta métrica permite visualizar o *overhead* gerado pela adição de novas informações nos pacotes de roteamento. No Gráfico 6 pode ser visualizado o comportamento dos protocolos em relação a esta métrica. Da mesma forma que na métrica *Overhead* de Roteamento (pacotes), o protocolo Ariadne-BFT também

apresentou um desempenho inferior ao Ariadne devido às modificações no processo de descoberta de rotas.

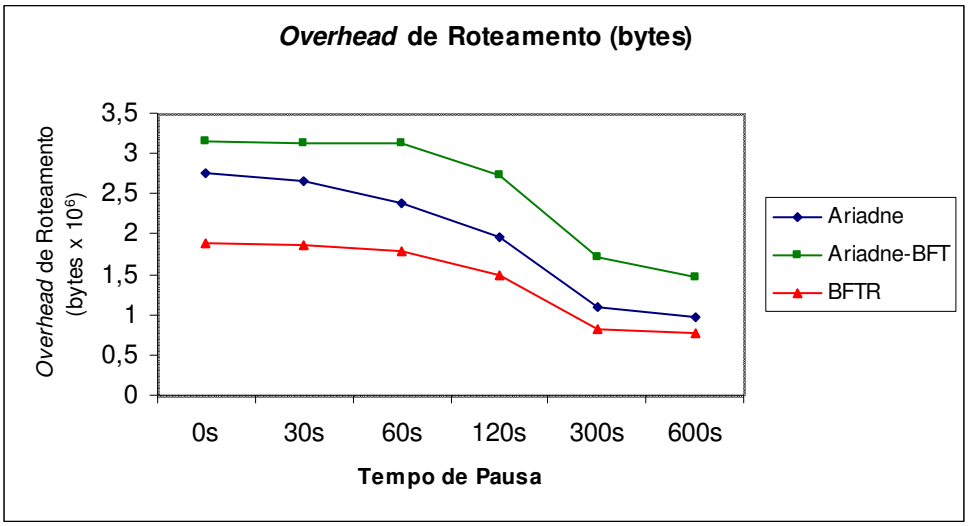


Gráfico 6 - Protocolos Ariadne, Ariadne-BFT e BFTR – *Overhead* de Roteamento x Tempo de Pausa

Já o protocolo BFTR, apesar de apresentar os maiores valores para a métrica *Overhead* de Roteamento (pacotes), obteve os melhores resultados nesta métrica. Esta diferença no desempenho é devido as informações adicionais utilizadas pelos protocolos Ariadne-BFT e Ariadne nos pacotes de roteamento (ver item 4.3.1), necessárias para o processo de autenticação e verificação de integridade dos pacotes de roteamento (RREQ, RREP e RERROR).

Quando a rede apresentou uma baixa mobilidade (tempos de pausa de 120s, 300s e 600s), todos os protocolos apresentaram uma diminuição no envio de pacotes de roteamento e conseqüentemente uma redução nos valores para essa métrica. Da mesma forma que no *overhead* de roteamento (pacotes), este comportamento ocorre pois os protocolos Ariadne-BFT, Ariadne e BFTR são do tipo reativo.

7.2.5 Pacotes Descartados

No Gráfico 7 é possível visualizar a quantidade total de pacotes descartados pelos protocolos durante a simulação. Os pacotes podem ser descartados devido a vários

fatores tais como sobrecarga tráfego de dados entre os nós, tamanho inadequado da fila nas interfaces e colisão de pacotes.

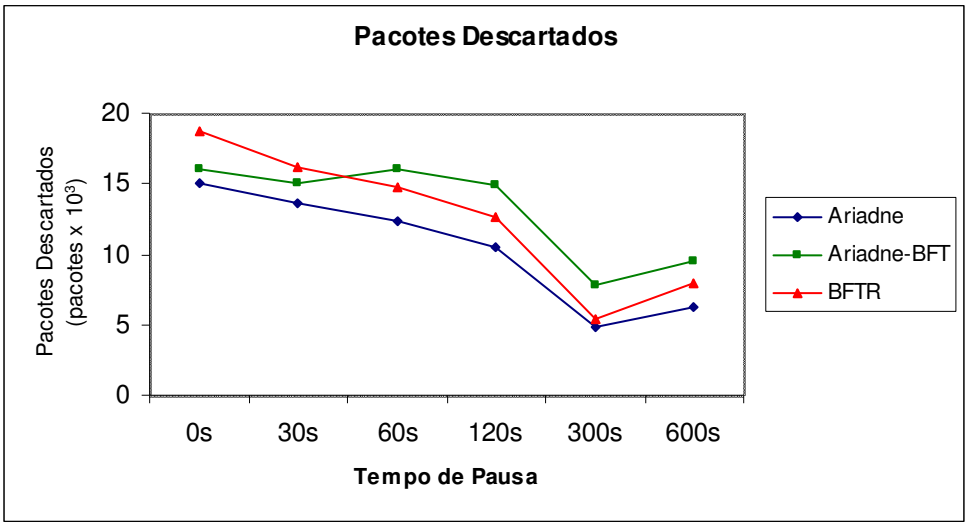


Gráfico 7 - Protocolos Ariadne, Ariadne-BFT e BFTR - Pacotes Descartados x Tempo de Pausa

O protocolo Ariadne-BFT descartou em média 27,01% mais pacotes que o protocolo Ariadne e 4,90% mais pacotes que o protocolo BFTR. O desempenho do Ariadne-BFT pode ser explicado considerando os valores obtidos por este protocolo na métrica *Overhead* de Roteamento (pacotes), pois como o protocolo utilizou uma quantidade maior de pacotes de roteamento durante o processo de descoberta de rotas, isto pode ter aumentado a colisão de pacotes na rede e conseqüentemente o aumento no descarte.

7.2.6 Vazão

Esta métrica é uma medida de transmissão de dados que determina a quantidade de dados enviada entre os nós da rede em um determinado período de tempo. No Gráfico 8 é apresentado a vazão alcançada por cada um dos protocolos durante as simulações. Neste trabalho foi utilizada a largura de banda de 11MB, valor máximo do padrão 802.11b, e tráfego de dados do tipo CBR.

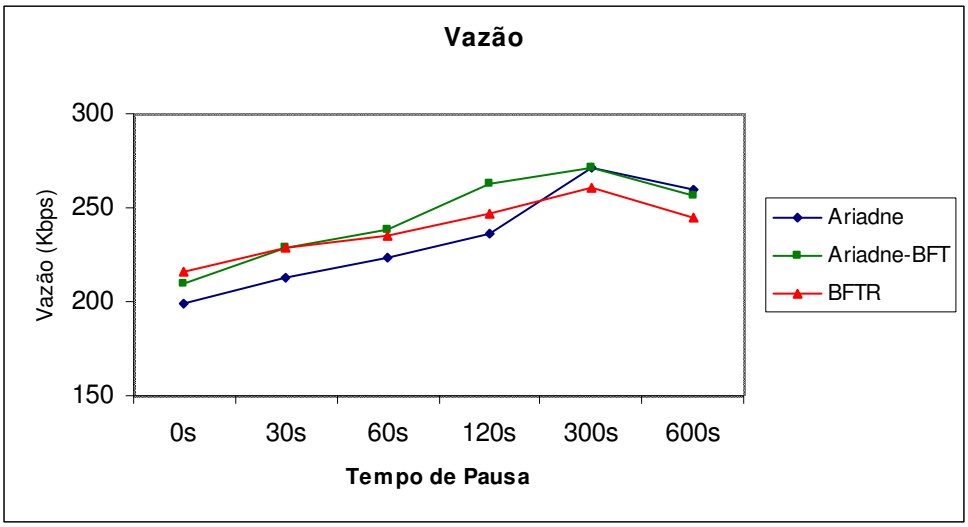


Gráfico 8 - Protocolos Ariadne, Ariadne-BFT e BFTR - Vazão x Tempo de Pausa

O protocolo Ariadne-BFT apresentou uma vazão superior ao protocolo Ariadne até o tempo de pausa de 120s. Em seguida, com a diminuição da mobilidade, a vazão de ambos foi semelhante. Como o protocolo Ariadne-BFT apresentou uma latência menor comparado com o protocolo Ariadne (ver Gráfico 3), o Ariadne-BFT conseguiu enviar um número maior de pacotes de dados e conseqüentemente a vazão foi maior.

8 Conclusões e Trabalhos Futuros

O projeto de um protocolo de roteamento seguro tem como meta proteger a rede *ad hoc* contra ataques em específico ou a um conjunto de ataques. Estes protocolos devem apresentar um balanceamento correto entre desempenho e segurança, pois como a tarefa de roteamento é essencial na rede, os mecanismos de segurança não podem limitar esta operação.

Inicialmente foi apresentada neste trabalho uma visão geral sobre as características das WLANs e do padrão IEEE 802.11, com foco principal nas redes *ad hoc*. Em seguida, foram apresentados os principais protocolos de roteamento dos grupos reativo e pró-ativo, pois são utilizados como base para alguns protocolos de roteamento seguro. Foi verificado que os protocolos reativos tem apresentado uma superioridade em relação aos protocolos pró-ativos, pois apresentaram um desempenho superior em diversos trabalhos [BRO98] [MAL99] [BOU04] [VIV06a]. Os protocolos reativos têm demonstrado maior eficiência em relação às mudanças na topologia da rede, diminuindo o *overhead* de roteamento quando os nós apresentam uma mobilidade baixa.

Na seqüência, é apresentada uma série de ataques executados por nós maliciosos com o objetivo de degradar o roteamento em uma rede *ad hoc*. São analisados os protocolos de roteamento seguro Ariadne, BFTR, SEAD, SAODV e ARAN em relação as suas características de funcionamento e dos mecanismos utilizados para proteger a rede contra estes ataques. Através de simulações, é realizada uma comparação de desempenho entre os protocolos reativos Ariadne e BFTR. Como resultado deste estudo, é proposto o Ariadne-BFT, uma extensão do protocolo Ariadne, baseada nos algoritmos utilizados pelo protocolo BFTR para descoberta, análise e seleção das rotas.

Da mesma forma que o protocolo Ariadne, o Ariadne-BFT protege a rede contra os ataques de consumo de recursos *Replaying, Routing Table Poisoning e Routing Table Overflow*. Entretanto, esta extensão possui a vantagem de proteger a rede contra o ataque de interrupção no roteamento *Blackhole*. É demonstrado através de extensas simulações que o Ariadne-BFT apresenta um desempenho um pouco melhor que o protocolo Ariadne em relação à métrica latência e razão de entrega de pacotes.

O Ariadne-BFT apresenta os mesmos princípios utilizados no protocolo Ariadne para autenticação e verificação de integridade na troca de mensagens entre os nós.

Além disso, esta extensão implementa mecanismos que utilizam a redundância de rotas, característica intrínseca das redes *ad hoc*, para manter um bom desempenho no roteamento mesmo com a presença de nós maliciosos. Mas apesar dos benefícios do Ariadne-BFT em relação ao Ariadne, os resultados das simulações demonstraram que esta extensão apresenta um aumento no *overhead* de roteamento comparado ao Ariadne. Isto ocorre devido aos pacotes de roteamento adicionais utilizados pelo Ariadne-BFT durante o processo de descoberta e manutenção das rotas.

Como continuação deste trabalho sugere-se a análise de desempenho do protocolo de roteamento seguro Ariadne-BFT em uma rede *ad hoc* com a presença de nós maliciosos. O esperado é que o desempenho deste protocolo tenha uma degradação menor se comparado com a versão original do protocolo Ariadne [HUY02a].

Baseado nos resultados obtidos com o protocolo Ariadne-BFT, outra contribuição seria a adaptação dos mecanismos de descoberta e seleção de rotas do protocolo BFTR e sua utilização no protocolo SAODV. Como este protocolo também é baseado no DSR, ganhos semelhantes na latência e entrega de pacotes obtidos com o protocolo Ariadne-BFT poderiam ser alcançados também com o SAODV, além da proteção contra o ataque *Blackhole*.

Em [LIU06] foi recentemente proposto o mecanismo 2ACK. Baseado no funcionamento do protocolo reativo DSR, também utiliza a redundância de rotas para proteger a rede contra nós maliciosos executando o ataque *Blackhole*. Desta forma, poderia ser implementada uma nova extensão do protocolo Ariadne baseada no mecanismo 2ACK. Na seqüência, poderiam ser realizadas simulações para comparar o desempenho entre esta nova extensão e o Ariadne-BFT proposto no presente trabalho.

REFERÊNCIAS

[AGR04] AGRAWAL, Dharma P.; CORDEIRO, Carlos M., **Mobile Ad hoc Networking**, capítulo 3, OBR Research Center for Distributed and Mobile Computing, ECECS University of Cincinnati, Cincinnati, 2004.

[ARG05] ARGYROUDIS, P.G. e O'MAHONY, D., **Secure routing for mobile ad hoc networks**, Communications Surveys & Tutorials, IEEE, vol. 7, pp. 2-21, 2005.

[AWE02] AWERBUCH, B., HOLMER, D., NITA-ROTARU, C. and RUBENS, H., **An On-demand Secure Routing Protocol Resilient to Byzantine Failures**, ACM Workshop on Wireless Security, pp. 21-30, 2002.

[BOU04] BOUKERCHE, A., **Performance evaluation of routing protocols for ad hoc wireless networks**, Mobile Networks and Applications, Kluwer Academic Publishers, 2004, vol. 9, pp. 333-342, 2004.

[BOU05] BOUDEC, J.-Y. Le and VOJNOVIC, M. **Perfect Simulation and Stationarity of a Class of Mobility Models**, IEEE INFOCOM, (Infocom Best Paper Award), 2005.

[BRO98] BROCH, Josh.; MALTZ, David. A.; JOHNSON, David B.; YIH-CHUN, Hu, JETCHEVA, Jorgeta. G., **A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols**, in proceedings of MOBICOM 1998.

[CHU02] CHUNG, J. e CLAYPOOL, M., **NS by Example**, WPI Worcester Polytechnic Institue - Computer Science Department. Disponível em: < http://nile.wpi.edu/NS/>. Acesso em: 10 setembro de 2004.

[COR96] CORSON, M.S.; MACKER, J.; BATSELL, S. **Architectural Considerations for Mobile Mesh Networking** In Proceedings of the IEEE MILCOM, 1996.

[COR99] CORSON, S. e MACKER, J., **Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations**, IETF Network Working Group, RFC 2501, 1999.

[CRO97] CROW, B., WIDJAJA, I., KIM, J., e SAKAI, P., **IEEE 802.11 Wireless Local Area Networks**, IEEE Communications Magazine, vol. 35, pp.116-126, 1997.

[CUR03] CURRAN, E., **SWARM: Cooperative Reinforcement Learning for Routing in Ad-hoc Networks**. Dissertação submetida à University of Dublin, Trinity College, como parte dos requisitos para a obtenção do grau de Mestre em Ciências da Computação. University of Dublin, 2003. Disponível em: <http://www.maths.tcd.ie/~currane/swarm>. Acessado em 28 outubro de 2006.

[FAL07] FALL, K. and VARADHAN, K., **The NS-2 Manual (formerly ns Notes and Documentation)**, The VINT Project (A collaboratoin between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC), 2007.

[GAS05] GAST, Matthew, **802.11 Wireless Networks The Definitive Guide**, O'Reilly, ISBN 0-596-10052-3, 656 pages, April 2005.

[HAG05] HAG, F. and KUNZ, T., **Simulation vs. Emulation: Evaluating Mobile Ad Hoc Network Routing Protocols**, IWWAN 2005 International Workshop on Wireless Ad-Hoc Networks, 2005.

[HEG03] HE,Guoyou, **Destination-Sequenced Distance Vector (DSDV) Protocol**, Networking Laboratory, Helsinki University of Technology. Disponível em: < http://citeseer.ist.psu.edu/531710.html >. Acesso em: 20 outubro 2004.

[HUY02a] HU, Yih-Chun; PERRIG, Adrian; JOHNSON, David B., **ARIADNE: A secure On-Demand Routing Protocol for Ad Hoc Networks**, in proceedings of MOBICOM 2002.

[HUY02b] HU, Yih-Chun; JOHNSON, David B.; PERRIG, Adrian, **SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks**, in the 4th IEEE Workshop on Mobile Computing Systems and Applications, 2002.

[HUY03a] HU, Yih-Chun, PERRIG, A., and JOHNSON, D.B., **Packet leashes: a defense against wormhole attacks in wireless networks**, In Proceedings of IEEE Infocom, Abril 2003.

[HUY03b] HU, Y-C., PERRIG, A., and JOHNSON, D.B., **Rushing attacks and defense in wireless ad hoc network routing protocols**, ACM Press, p. 30-40, 2003.

[HUY03c] HU, Y-C., PERRIG, A. and JOHNSON D. B., **Efficient Security Mechanisms for Routing Protocols**, In Proceedings of Network and Distributed Systems Security, 2003.

[HUY04a] HU, Yih-Chun, PERRIG, A., and JOHNSON, D.B., **Securing quality-of-service route discovery in on-demand routing for ad hoc networks**, ACM Press, p. 106-117, 2004.

[HUY04b] HU, Yih-Chun and PERRIG, Adrian, **A Survey of Secure Wireless Ad Hoc Routing**, IEEE Security and Privacy, IEEE Educational Activities Department, Vol. 2, p. 28-39, 2004.

[HUY06] HU, Yih-Chun e PERRIG, A.J., **Wormhole attacks in wireless networks**, IEEE Journal on Selected Areas in Communications, vol. 24, no. 2, pp. 370-380, 2006.

[IEE99a] IEEE, **Part 11: Wireless LAN Medium Access Control (MAC) and Physical (PHY) Specifications**. IEEE Standard 802.11, 1999.

[IEE99b] IEEE, **Part 11: Wireless LAN Medium Access Control (MAC) and Physical (PHY) Specifications - Higher-Speed Physical Layer Extension in the 2.4 GHz Band.** IEEE Standard 802.11b, 1999.

[IEE99c] IEEE, **Part 11: Wireless LAN Medium Access Control (MAC) and Physical (PHY) Specifications - High-speed Physical Layer in the 5 GHz Band,** IEEE Standard 802.11a, 1999.

[IEE01] IEEE, **Part 11: Wireless LAN Medium Access Control (MAC) and Physical (PHY) Specifications Amendment 3: Specification for operation in additional regulatory domains.** IEEE Standard 802.11d, 2001.

[IEE03] IEEE, **IEEE Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation.** IEEE Standard 802.11f, 2003.

[IEE04] IEEE, **Part 11: Wireless LAN Medium Access Control (MAC) and Physical (PHY) Specifications - Amendment 6: Medium Access Control (MAC) Security Enhancements.** IEEE Standard 802.11i, 2004.

[IEE05] IEEE, **Part 11: Wireless LAN Medium Access Control (MAC) and Physical (PHY) Specifications - Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements.** IEEE Standard 802.11e, 2005.

[ISO04] **INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**. Disponível em:<http://www.iso.org/>. Acesso em: 22 outubro de 2005.

[JOH01] JOHNSON, David B.; MALTZ, D. A.; BROCH, J. **DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks**. In C. Perkins, editor, Ad Hoc Networking, chapter 5, pages 139--172. Addison-Wesley, 2001.

[JOH96] JOHNSON, David B.; MALTZ, D. A., **Dynamic Source Routing in Ad Hoc Wireless Networks (DSR)**. In Mobile Computing, edited by Tomasz Imielinski and Hank Korth, chapter 5, pp. 153–181. Kluwer Academic Publishers, 1996.

[JOH04] JOHNSON, D. B., MALTZ, D. A., HU, Yih-Chun, **The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)**, IETF MANET Working Group, INTERNET-DRAFT, <draft-ietf-manet-dsr-10.txt>, 2004.

[JOH99] JOHANSSON, Per; LARSSON, Tony; HEDMAN, Nicklas; MIELCZAREK, Bartosz; DEGERMARK, Mikael, **Scenario-based Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks**, in proceedings of MOBICOM 1999.

[JUB87] JUBIN, J.; TRUONG, T. **Distributed Algorithm for Efficient and Interference-free Broadcasting in Radio Networks** in Proceedings of INFOCOM, January 1987, 21-32.

[LAZ05] LAZOS, L., POOVENDRAN, R., MEADOWS, C., SYVERSON, P and CHANG, L. W., **Preventing Wormhole Attacks on Wireless Ad Hoc Networks: Graph Theoretic Approach**, IEEE Wireless Communications and Networking Conference, vol. 2, pp. 1193 - 1199, ISBN 0-7803-8966-2, 2005.

[LIU06] LIU, K., DENG, J., VARSHNEY, P. K., e BALAKRISHNAN, K., **An Acknowledgment-based Approach for the Detection of Routing Misbehavior in MANETs**, IEEE Transactions on Mobile Computing, in press, 2006.

[LUO04] LUO, H., KONG, J., ZERFOS, P., LU, S. and ZHANG, L., **URSA: ubiquitous and robust access control for mobile ad hoc networks**, IEEE/ACM Transactions on Networking, IEEE Press, Vol. 12, pp. 1049-1063, 2004.

[MAL99] MALTZ, D. A., BROCH, J., JETCHEVA, J., and JOHNSON, D. B., **The Effects of On-Demand Behavior in Routing Protocols for Multi-Hop Wireless Ad Hoc Networks**, IEEE Journal on Selected Areas in Communications, vol. 17, issue 8, pp. 1439 - 1453, August 1999.

[MAR00] MARTI, S., GIULI, T., LAI, K., and BAKER, M. **Mitigating routing misbehavior in mobile ad hoc networks**, In Proceedings of the Sixth annual ACM/IEEE International Conference on Mobile Computing and Networking, pp. 255-265, 2000.

[MON04a] MONARCH PROJECT, **Wireless and Mobility Extensions to NS2**. Disponível em: <http://www.monarch.cs.cmu.edu/cmu-ns.html>. Acesso em 7 Outubro de 2005.

[MON04b] MONET RESEARCH GROUP, **Multimedia Operating System and Networking Group**. Disponível em: < http://cairo.cs.uiuc.edu/>. Acesso em Novembro de 2005.

[MUR96] MURTHY, S., and Garcia-Luna-Aceves, J. J., **An Efficient Routing Protocol for Wireless Networks (WRP)**, ACM Mobile Networks and Applications, Special Issue on Routing in Mobile Communication Networks, vol. 1, pp. 183–97, 1996.

[PER94] PERKINS, Charles E.; BHAGWAT, Pravin, **Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers**. In Proceedings of the ACM SIGCOMM, pages 234--244, August 1994.

[PER99] PERKINS, C. E. and Royer, E. M., **Ad hoc On-Demand Distance Vector Routing (AODV)**, Proc. 2nd IEEE Workshop Mobile Computing Systems and Applications, New Orleans, LA, pp. 90-100, 1999.

[PER03a] PEREIRA, M. C. **Análise de Desempenho em Redes Wireless Ad Hoc e Estabelecimento de um Acordo de Nível de Serviço Pró-Ativo**. 2003. 145f. Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciências da Computação. Universidade Federal de Santa Catarina – UFSC, 2003.

[PER03b] PERKINS, C. E., ROYER, E., DAS, S., **Ad hoc On-Demand Distance Vector (AODV) Routing**, IETF Network Working Group, RFC 3561, 2003.

[PER05] PERRIG, A., SONG, D., CANETTI, R., TYGAR, J. D., BRISCOE, B., **Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction**, Network Working Group, RFC 4082, 2005.

[POO07] POOVENDRAN, R. e LAZOS, L. **A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks,** Wireless Networks, Kluwer Academic Publishers, vol. 13, pp. 27-59, 2007.

[OUE02] OUELLET, E; PADJEN, R.; PFUND, A. **Building a CISCO Wireless Lan**, Syngress Publishing, 2002. ISBN 1-928994-58-X.

[ROY99] ROYER, E. M., TOH, C-k. **A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks**, IEEE Personal Communications Magazine, vol. 6, n. 6, pp. 46-65, April 1999.

[RUB04] RUBINSTEIN, Marcelo G.; REZENDE, José F. **Qualidade de Serviço em Redes 802.11**, Disponível em:<http://www.gta.ufrj.br/ftp/gta/TechReports/RuRe02.pdf>. Acesso em: 8 abril 2004.

[SAN02] SANZGIRI, K., DAHILL, B., LEVINE, B.N., SHIELDS, C. and ROYER, E.M., **A Secure Routing Protocol for Ad hoc Networks (ARAN)**, Proceedings of 10th IEEE Int'l. Conf. Network Protocols (ICNP'02), IEEE Press, pp. 78-87, 2002.

[SIL98] SILVA, Adailton J. S., **As Tecnologias de Redes Wireless**, Rede Nacional de Ensino e Pesquisa (RNP), Vol 2, N. 5, ISSN 1518-5974, 1998.

[VIV06a] VIVIAN, D., ALCHIERI, E. A. P., WESTPHALL, C. B., **Evaluation of metric of QoS in Ad Hoc networks with the use of Security Routing Protocols**, In: 10th IEEE/IFIP Network Operations and Management Symposium (NOMS), pp. 1-14, ISBN:1-4244-0142-9, 2006.

[VIV06b] VIVIAN, D. e WESTPHALL, C. B. **Comparação de Desempenho de Protocolos de Roteamento Seguro para Redes Sem Fio Ad Hoc**, Anais do XXXIII Seminário Integrado de Software e Hardware – SEMISH, 2006.

[WUB06] WU, B., CHEN, J., WU, J., CARDEI, M., **A Survey on Attacks and Countermeasures in Mobile Ad Hoc Networks**, Wireless/Mobile Network Security, chapter 12, 38 pages, Springer, 2006.

[XUE04] XUE, Y. and NAHRSTEDT, K. **Providing Fault-Tolerant Ad hoc Routing Service in Adversarial Environments**, Wireless Personal Communications, Kluwer Academic Publishers, Vol. 29, p. 367-388, 2004.

[YIS01] YI, Seung, NALDURG, Prasad and KRAVETS, Robin, **Security-aware ad hoc routing for wireless networks (SAR)**, MobiHoc01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing, pp. 299-302, New York, NY, USA, ACM Press, 2001.

[YOV04] YOVANOF, Gregory S.; ERIKCI, Kerem. **Performance Evaluation of Security-Aware Routing Protocols for Clustered Mobile Ad Hoc Networks**, International Workshop on Wireless Ad-Hoc Networks, pp. 286-290, 2004.

[ZAP02] ZAPATA, M.G. e ASOKAN, N., **Secure Ad hoc On-Demand Distance Vector Routing (SAODV)**, ACM Mobile Computing and Communications Review, vol. 3, no. 6, pp.106-107, July 2002.

[ZAP06] ZAPATA, M. G., **Secure Ad hoc On-Demand Distance Vector (SAODV) Routing**, Mobile Ad Hoc Networking Working Group, INTERNET-DRAFT, <draft-guerrero-manet-saodv-06.txt>, 2006.

[ZHO99] ZHOU, L., HAAS, Z. J., **Securing Ad Hoc Networks**, IEEE Network Magazine, vol. 13, no. 6, pp. 24-30, 1999.

ANEXO A – Script para Análise dos Arquivos de Trace do Simulador NS-2

```
#!/usr/bin/perl

# Federal University of Santa Catarina
# Darlan Vivian - darlan@inf.ufsc.br

$tick = 0.0001;
$lastTick=0-$tick;
$lastPacketIdGenerated=0;

#counting by source, how much sent/received
$appPacketsSent = 0;
$appPacketsReceived = 0;
$appLatencyTotal = 0;
$appBytesDelivered = 0;

#The interval we are currently in. Began at $lastTick, $lastTickPacketId
$thisPacketsSent = 0;
$thisPacketsReceived = 0;
$thisLatencyTotal =0;
$thisBytesDelivered = 0;

#The previous interval, that we may still be receiving packets from.
$previousPacketsSent = 0;
$previousPacketsReceived = 0;
$previousLatencyTotal = 0;
$previousBytesDelivered = 0;

#counting by source, how much sent/received
$appPacketsSentOverhead = 0;
$appPacketsReceivedOverhead = 0;
$appLatencyTotalOverhead = 0;
$appBytesDeliveredOverhead = 0;

#The interval we are currently in. Began at $lastTick, $lastTickPacketId
$thisPacketsSentOverhead = 0;
$thisPacketsReceivedOverhead = 0;
$thisLatencyTotalOverhead =0;
$thisBytesDeliveredOverhead = 0;

#The previous interval, that we may still be receiving packets from.
$previousPacketsSentOverhead = 0;
$previousPacketsReceivedOverhead = 0;
$previousLatencyTotalOverhead = 0;
$previousBytesDeliveredOverhead = 0;

#All packets dropped
$appAllPacketsDropped = 0;

while (<>) {
   if(/-t (\S+)/)
   {
          $time=$1;

          if($time > ($lastTick+$tick) )
```

```
    {
       if ($lastTick>0)
       {
          $appPacketsSent += $previousPacketsSent;
               $appPacketsReceived += $previousPacketsReceived;
               $appLatencyTotal += $previousLatencyTotal;
               $appBytesDelivered += $previousBytesDelivered;

        $appPacketsSentOverhead += $previousPacketsSentOverhead;
               $appPacketsReceivedOverhead += $previousPacketsReceivedOverhead;
               $appLatencyTotalOverhead += $previousLatencyTotalOverhead;
               $appBytesDeliveredOverhead += $previousBytesDeliveredOverhead;

       }
       $lastTick = $lastTick + $tick;
       $lastTickPacketId = $lastPacketIdGenerated;

       $previousPacketsSent = $thisPacketsSent;
       $thisPacketsSent = 0;
       $previousPacketsReceived = $thisPacketsReceived;
       $thisPacketsReceived = 0;
       $previousLatencyTotal = $thisLatencyTotal;
       $thisLatencyTotal = 0;
       $previousBytesDelivered = $thisBytesDelivered;
       $thisBytesDelivered = 0;

    $previousPacketsSentOverhead = $thisPacketsSentOverhead;
       $thisPacketsSentOverhead = 0;
       $previousPacketsReceivedOverhead = $thisPacketsReceivedOverhead;
       $thisPacketsReceivedOverhead = 0;
       $previousLatencyTotalOverhead = $thisLatencyTotalOverhead;
       $thisLatencyTotalOverhead = 0;
       $previousBytesDeliveredOverhead = $thisBytesDeliveredOverhead;
       $thisBytesDeliveredOverhead = 0;
    }
# Agent trace (AGT) packets. Only data packets (CBR)
    if (/^.*-Nl AGT.*-It cbr/)
    {
       # Packet sent/received by the app stack
       if (/^r.*-Is (\S+) -Id (\S+).*-Il (\S+).*-Ii (\S+)/)
       {
              $id = $4;
              $bytes = $3;
              if ($recvTime[$id] ==-1)
              {
                 $recvTime[$id] = $time;
                 if ($id <= $lastTickPacketId)
                 {
                        $previousPacketsReceived++;
                        $previousBytesDelivered+=$bytes;
                        $previousLatencyTotal+=$recvTime[$id]-$sendTime[$id];
                 }
                 else
                 {
                        $thisPacketsReceived++;
                        $thisBytesDelivered+=$bytes;
                        $thisLatencyTotal+=$recvTime[$id]-$sendTime[$id];
                 }
```

```
                }
            }

        if (/^s.*-Is (\S+) -Id (\S+).*-Il (\S+).*-Ii (\S+)/)
            {
                $thisPacketsSent++;

                $lastPacketIdGenerated=$4;
                $sendTime[$4] = $time;
                $recvTime[$4] = -1;
            }
          }


      # Route trace (RTR) packets. Only routing packets (DSR)
      if (/^.*-Nl RTR.*-It DSR/)
          {

        # Packet sent/received by the app stack
            if (/^r.*-Is (\S+) -Id (\S+).*-Il (\S+).*-Ii (\S+)/)
            {
                $id = $4;
                $bytes = $3;
                if ($recvTimeOverhead[$id] ==-1)
                {
                   $recvTimeOverhead[$id] = $time;
                   if ($id <= $lastTickPacketId)
                   {
                        $previousPacketsReceivedOverhead++;
                        $previousBytesDeliveredOverhead+=$bytes;
                        $previousLatencyTotalOverhead+=$recvTimeOverhead[$id]-
$sendTimeOverhead[$id];
                   }
                   else
                   {
                        $thisPacketsReceivedOverhead++;
                        $thisBytesDeliveredOverhead+=$bytes;
                        $thisLatencyTotalOverhead+=$recvTimeOverhead[$id]-
$sendTimeOverhead[$id];
                   }
                }
            }

        if (/^s.*-Is (\S+) -Id (\S+).*-Il (\S+).*-Ii (\S+)/)
            {
                $thisPacketsSentOverhead++;

                $lastPacketIdGenerated=$4;
                $sendTimeOverhead[$4] = $time;
                $recvTimeOverhead[$4] = -1;
            }
      }


 # All packets dropped
      if (/^d.*/)
      {
        $appAllPacketsDropped++;
```

```
        }
    }
}

printf "%d, %d, %d, %f, %f, ",
                        $appPacketsSent, $appPacketsReceived, $appBytesDelivered,
                        ($appPacketsReceived/$appPacketsSent),
                        ($appLatencyTotal/$appPacketsReceived);

printf "%d, %d, %d, %f, %f, %d\n",
                        $appPacketsSentOverhead, $appPacketsReceivedOverhead,
$appBytesDeliveredOverhead,
                        ($appPacketsReceivedOverhead/$appPacketsSentOverhead),
                        ($appLatencyTotalOverhead/$appPacketsReceivedOverhead), $appAllPacketsDropped;
```

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo