MAPA: MECANISMO DE AVALIAÇÃO E PUNIÇÃO DE NÓS EGOÍSTAS EM REDES AD HOC

Reinaldo Bezerra Braga

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

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

Prof. José Neuman de Souza, Dr.

Prof. Luís Henrique Maciel Kosmalski Costa, Dr.

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2008

BRAGA, REINALDO BEZERRA

MAPA: Mecanismo de Avaliação e Punição de nós egoístas em redes Ad hoc[Rio de Janeiro] 2008

XII, 77 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia Elétrica, 2008)

Dissertação - Universidade Federal do Rio de Janeiro, COPPE

1. Redes Ad hoc
2. Segurança
3. Detecção de Intrusão

I. COPPE/UFRJ II. Título (série)

À minha família.

Agradecimentos

Um agradecimento especial à minha família Antônio Braga, Maria Lucielma, Telma Maria, Raimundo Bezerra, Liduína Bezerra e aos demais familiares pelo carinho, atenção, compreensão, confiança e incentivo ao longo de toda a minha vida.

À Carina, que sempre esteve ao meu lado, pelo apoio, paciência, carinho, compreensão e dedicação em todos os momentos. À minha segunda família, que Deus colocou na minha vida, Liduína Teixeira, Mauro Oliveira, Karol Teixeira e Carolina Teixeira, pela confiança, motivação, carinho e conselhos. Além de um agradecimento "tradicionalíssimo" ao meu amigão Cláudio Lenz Cesar.

Ao professor Otto, meu orientador e responsável por uma importante parte da minha formação acadêmica e profissional, por sua amizade, conselhos e orientação. Aos professores do GTA, Luís Henrique, Rezende, Aloysio e Rubi, pela amizade e ensinamentos.

A todos do GTA, em especial aos amigos Danilo, Marcelo Duffles, Natalia, Kleber, Igor, Miguel, Carlos Henrique, Sávio, Natanael, André, Raphael, Marcel, Vinícius e ao pessoal da Iniciação Científica, pelas risadas e pelo incentivo durante o trabalho. Além dos amigos da UFRJ, Ivomar, Luiz Hoffmann, Diana, Jorge e Fehmi, que sempre estiveram à disposição quando precisei.

Agradeço em particular aos professores José Neuman de Souza e Luís Henrique Costa pela participação na banca examinadora.

Aos funcionários do Programa de Engenharia Elétrica da COPPE/UFRJ pela presteza no atendimento na secretaria do Programa. A todos que me incentivaram, contribuindo de forma direta ou indireta, para a minha formação acadêmica e profissional. Ao CNPq, à Capes, ao UOL, à FINEP, à RNP e ao FUNTTEL pelo financiamento da pesquisa.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

MAPA: MECANISMO DE AVALIAÇÃO E PUNIÇÃO DE NÓS EGOÍSTAS EM REDES AD HOC

Reinaldo Bezerra Braga

Março/2008

Orientador: Otto Carlos Muniz Bandeira Duarte

Programa: Engenharia Elétrica

As redes ad hoc confiam na cooperação dos nós para que as funções de roteamento e encaminhamento de pacotes sejam realizadas com sucesso. Entretanto, os nós com comportamento egoísta se beneficiam da característica de cooperação destas redes para não encaminhar pacotes, podendo reduzir o desempenho da rede. Portanto, é fundamental a detecção e uma atitude contra os nós egoístas em redes ad hoc. Entretanto, a precisão nas detecções e punições é dificultada devido aos falso-positivos gerados por conseqüência dos problemas temporários das redes ad hoc, tais como as colisões, a disputa de acesso ao meio e o desvanecimento. Neste trabalho, é apresentado o Mecanismo de Avaliação e Punição de nós egoístas em redes Ad hoc (MAPA). A avaliação e a punição dos nós egoístas são realizadas com base nos resultados das detecções coletadas localmente por cada nó. Através de uma análise matemática e de simulações, é apresentada a eficiência do MAPA no processo de avaliação e punição de nós egoístas, reduzindo o total de falso-positivos e aumentando a taxa de entrega de pacotes da rede.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

MAPA: EVALUATION AND PUNISHMENT MECHANISM TO AVOID SELFISH NODES IN AD HOC NETWORKS

Reinaldo Bezerra Braga

March/2008

Advisor: Otto Carlos Muniz Bandeira Duarte

Department: Electrical Engineering

Ad hoc networks rely on node cooperation to perform routing and data forwarding. Therefore, nodes with selfish behavior can benefit from this cooperation to drop packets, decreasing the network performance. For this reason, one of the main challenges is to detect and to respond to the selfish behavior of nodes. However, the detections and responses accuracy is affected by the false positives caused by temporary problems in ad hoc networks, such as collision, medium access contention, and fading. This work presents the MAPA, a mechanism that performs evaluations and punishments of selfish nodes based on the results of only local detections. The mathematical analysis and simulations show that the MAPA is efficient on evaluations and punishments to the selfish nodes, decreasing the number of false positives and improving the network delivery rate.

Sumário

Lista de Figuras

Lista de Acrônimos

SDI :	*Sistema de Detecção de Intrusão*;
AODV :	*Ad Hoc On-Demand Distance Vector Routing*;
DSR :	*Dynamic Source Routing*;
OLSR :	*Optimized Link State Routing*;
DSDV :	*Destination-Sequenced Distance-Vector*;
RREQ :	*Route Request*;
RREP :	*Route Reply*;
RERR :	*Route Error*;
MPR :	*MultiPoint Relays*;
RIP :	*Routing Information Protocol*;
MAC :	*Medium Access Control*;
WEP :	*Wired Equivalent Privacy*;
IP :	*Internet Protocol*;
PGP :	*Pretty Good Privacy*;
CONFIDANT :	*Cooperation Of Nodes, Fairness In Dynamic Ad-hoc NeTworks*;
GloMoSim :	*Global Mobile Information Systems Simulation Library*;
CORE :	*A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad hoc Networks*;
OCEAN :	*Observation-based Cooperation Enforcement in Ad hoc Networks*;
SDE :	*Sistema de Detecção de nós Egoístas*;
MAPA :	*Mecanismo de Avaliação e Punição de nós egoístas em redes Ad hoc*;
PMF :	*Probability Mass Function*.

Capítulo 1

Introdução

REDES AD HOC são caracterizadas pela ausência de infra-estrutura ou de administração centralizada. Portanto, estas redes confiam na cooperação dos nós para realizarem as funções de roteamento e encaminhamento de dados. A partir desta característica, um nó pode decidir não cooperar no encaminhamento de pacotes com o objetivo de atacar a rede ou de simplesmente economizar os seus recursos computacionais. Estes nós que não cooperam no encaminhamento são chamados de nós egoístas e podem reduzir o desempenho da rede ao descartarem os pacotes que deveriam ser encaminhados. Desta forma, é necessário utilizar mecanismos capazes de evitar que estes nós mal comportados sejam usados na comunicação entre os nós cooperativos das redes ad hoc.

1.1 Motivação

As propostas convencionais contra maus comportamentos utilizam mecanismos criptográficos para identificar e autenticar os nós, além de proteger o conteúdo das mensagens. Entretanto, estes mecanismos sozinhos não garantem que toda estação autenticada se comportará corretamente na rede. Portanto, torna-se necessária a utilização de mecanismos para detectar e punir os nós autenticados que realizam maus comportamentos. Estes mecanismos são conhecidos na literatura como sistemas de detecção de intrusão.

De acordo com Kang *et al.*, dois modelos de detecção de maus comportamentos po-

dem ser utilizados em redes ad hoc [1]. No primeiro modelo, baseado em assinaturas, cada nó mantém uma base de assinaturas dos eventos de maus comportamentos conhecidos. Dessa forma, qualquer evento que possua uma assinatura semelhante a uma assinatura presente na base de assinaturas é classificado como um mau comportamento. Já no segundo modelo, baseado em anomalias, todo nó usa uma base de eventos normais conhecidos e classifica como mau comportamento qualquer evento diferente dos eventos da base.

De acordo com Anantvalee e Wu, os modelos baseados em anomalias não requerem a análise de uma grande quantidade de assinaturas de maus comportamentos, tais como as análises realizadas pelos modelos baseados em assinaturas [2], pois para cada modificação de mau comportamento é necessária a criação de uma nova assinatura. Além disso, os modelos baseados em anomalias não necessitam de atualizações constantes em sua base de eventos e possibilitam a análise de detecções de forma local.

1.2 Objetivo

Em redes ad hoc, os modelos de detecção baseados em assinaturas e anomalias geram falso-positivos nas punições, que ocorrem, por exemplo, quando um nó cooperativo é punido por ter sido classificado como um nó egoísta. Os principais fatores causadores das detecções incorretas são os problemas temporários que ocorrem nas redes ad hoc, tais como o desvanecimento, as colisões e a disputa de acesso ao meio.

Análises de comportamento de canais sem fio mostram que o desvanecimento é um problema comum em redes ad hoc, mesmo quando os cenários são estáticos [3] [4]. Portanto, devido às freqüentes variações dos canais provocadas pelo desvanecimento, um nó que está monitorando o reencaminhamento de pacotes pelos nós vizinhos pode não conseguir realizar esta tarefa corretamente. Nas colisões, os falso-positivos ocorrem quando um nó não percebe que o pacote foi corretamente encaminhado pelo seu vizinho, devido a uma colisão [5]. Já na disputa de acesso ao meio, um nó pode ser detectado enquanto aguarda a liberação do meio compartilhado para encaminhar os pacotes que estão na fila. Portanto, ao considerar os problemas em redes ad hoc, é importante que o mecanismo de

detecção e punição seja eficiente para reduzir a quantidade de falso-positivos nas punições.

Neste trabalho, é apresentado um mecanismo que aumenta a precisão nas respostas aplicadas aos nós egoístas da rede. Esta precisão é obtida através de avaliações realizadas pelo Mecanismo de Avaliação e Punição de nós egoístas em redes Ad hoc (MAPA), que utiliza as informações dos eventos monitorados por um sistema de detecção baseado em anomalias, chamado de Sistema de Detecção de nós Egoístas (SDE). O SDE monitora o encaminhamento de pacotes realizado pelo próximo salto na rota e contabiliza todos os pacotes encaminhados ou não. Após observar uma quantidade determinada de pacotes não encaminhados, o SDE passa estas informações para o MAPA, que é responsável por avaliar todos estes eventos monitorados e por determinar se o nó avaliado será bloqueado temporariamente ou definitivamente da comunicação com o nó que o avaliou.

O objetivo principal da proposta é reduzir o número de bloqueios definitivos enviados aos nós cooperativos, aumentando a oportunidade do nó cooperativo, que foi incorretamente detectado, provar que não é um nó egoísta. Um bloqueio significa que um nó detectado é excluído de qualquer funcionalidade da rede, tanto na criação como no encaminhamento de pacotes. Além disto, o MAPA tem o propósito de punir uma maior quantidade de nós egoístas sem gerar uma elevada quantidade de falso-positivos. Os resultados da simulação mostram que o SDE e o MAPA são mais eficientes do que as outras propostas analisadas, devido à menor razão entre a quantidade de falso-positivos gerada para cada punição corretamente aplicada. A partir dos resultados, é observado que o SDE/MAPA aumenta a taxa de entrega da rede em até 27%, mesmo com a presença de 12,5% dos nós da rede descartando pacotes que deveriam ser encaminhados.

1.3 Organização

Este trabalho está organizado da seguinte forma. No capítulo 2 são apresentadas as características e os procedimentos comuns de funcionamento das redes ad hoc. Além disso, são mostrados os principais tópicos relacionados à segurança em redes ad hoc. No capítulo 3 são apresentadas as características dos Sistemas de Detecção de Intrusão (SDIs),

além dos desafios encontrados para utilizar estes sistemas em redes ad hoc. São apresentados ainda os trabalhos relacionados ao tema de detecção de nós egoístas em redes ad hoc. No Capítulo 4, é apresentada a proposta deste trabalho, através de uma abordagem detalhada, além disso, são descritos os parâmetros assumidos e os resultados da análise matemática do mecanismo proposto. No Capítulo 5 são apresentadas as premissas assumidas para os cenários de rede ad hoc e discutidos os resultados obtidos na simulação. Por fim, no Capítulo 6 são apresentadas a conclusão e os trabalhos futuros.

Capítulo 2

Segurança em Redes Ad Hoc

ATUALMENTE , as redes sem fio passam por um processo de ascensão em sua tecnologia, principalmente nas redes ad hoc. As redes ad hoc surgiram com funcionalidades particulares, capazes de formar uma rede sem infra-estrutura pré-definida, adaptando-se a qualquer ambiente e a diferentes modelos de aplicação. Estas redes podem ser utilizadas em aplicações voltadas para operações militares, interações entre usuários e áreas de desastres. Para funcionarem, os nós assumem as funções de roteadores, repassando pacotes para os seus vizinhos, até que as informações enviadas na rede alcancem o seu destino.

O custo de instalação e a facilidade de configuração são os principais atrativos para a escolha das redes ad hoc. Entretanto, o meio de comunicação sem fio, a ausência de infra-estrutura e o roteamento colaborativo em múltiplos saltos tornam estas redes alvos fáceis para diversos tipos de ataques. Assim, a segurança torna-se um desafio nas redes ad hoc [6].

2.1 Redes Ad Hoc

Redes ad hoc, conforme definido em [7], são coleções de nós que cooperam para formar uma rede sem usar qualquer infra-estrutura, tais como pontos de acessos. Além disso, quando um nó está fora do alcance de transmissão do rádio de outro nó, a comunicação

é realizada através de múltiplos saltos, exigindo a colaboração dos nós intermediários. Assim, a mobilidade dos nós e a limitação da capacidade do meio sem fio, juntos com os efeitos de transmissão, como atenuações, propagação através de múltiplos caminhos e interferências, são combinações que geram os desafios operacionais dos protocolos de roteamento nas redes ad hoc.

Para que as redes ad hoc funcionem de forma correta e eficiente, os protocolos de roteamento devem considerar as características particulares destas redes em suas implementações. Dentre os protocolos de roteamento mais utilizados em redes ad hoc, quatro podem ser citados: o *Ad Hoc On-Demand Distance Vector Routing* (AODV) [8]; o *Dynamic Source Routing* (DSR) [9]; o *Destination-Sequenced Distance-Vector* (DSDV) [10]; e o *Optimized Link State Routing* (OLSR) [11]. O principal interesse em apresentar estes protocolos é entender o funcionamento de cada um, como também analisar as similaridades e as diferenças nos processos de descoberta, atualização e manutenção de rotas.

2.1.1 Roteamento

Os protocolos de roteamento das redes ad hoc são classificados de acordo com a sua característica de funcionamento em reativos e pró-ativos. O AODV e o DSR são exemplos de protocolos de roteamento reativos, pois realizam a descoberta de rota sob demanda, ou seja, quando necessária. Já o DSDV e o OLSR são classificados como protocolos de roteamento pró-ativos, pois as rotas são criadas e mantidas previamente em tabelas, estando disponíveis quando houver a necessidade de uso. A seguir, são apresentadas as principais características de funcionamento destes protocolos.

AODV - É um protocolo reativo para redes ad hoc e baseia-se em vetor de distância. Para realizar descobertas de rotas em redes ad hoc, este protocolo utiliza mensagens de requisição de rotas, conhecidas como *Route Request* (RREQ), e mensagens de resposta às requisições de rota, chamadas de *Route Reply* (RREP). A mensagem RREQ contém o endereço do nó de origem e do nó de destino que se deseja alcançar. Devido à característica de ser um protocolo reativo, esta mensagem é enviada quando um nó necessita transmitir dados para um destino para o qual ele não co-

nhece uma rota. O envio desta mensagem é feito através de difusão para todos os vizinhos que são atingidos pelo raio de comunicação do nó, e é encaminhada na rede em um processo de inundação. Quando um nó conhece uma rota para o destino informado na mensagem de RREQ, ele então responde em *unicast*, através da mensagem de RREP, para a origem. Portanto, o caminho seguido pelo RREQ estabelece a rota até o destino. Para evitar uma descoberta de rota a cada novo pacote, o AODV utiliza um cache, que mantém as rotas descobertas. Ao ser quebrada uma rota existente, é utilizada a mensagem de erro de rota, conhecida como *Route Error* (RERR), que é enviada para a origem, indicando a necessidade de outra rota.

DSR - É um protocolo reativo para redes ad hoc e baseia-se no roteamento pela origem. Este protocolo utiliza as mesmas mensagens usadas pelo AODV no procedimento de descoberta de rotas, ou seja, através de requisições e respostas às requisições de rotas. Entretanto, ao ser descoberta uma rota, o encaminhamento de pacotes é feito através de roteamento pela origem. Isto significa que cada pacote que sai da origem contém a informação de todos os nós intermediários que devem ser utilizados até que o pacote alcance o nó de destino. Na requisição de rotas, o DSR também utiliza as mensagens de RREQ, que são enviadas por difusão na rede, e as mensagens de RREP, que são respondidas em *unicast* a cada requisição de rota. Para formar uma rota, cada nó insere o seu endereço na mensagem de RREQ. Estes endereços são registrados a cada salto percorrido, permitindo que o roteamento pela origem seja realizado. O DSR também utiliza o armazenamento de rotas em cache, porém, de forma mais significativa do que o cache usado pelo AODV, fazendo com que os nós registrem as rotas utilizadas por pacotes previamente encaminhados por eles na rede. Além disso, o cache do DSR é utilizado em algumas otimizações das funções de descoberta e de manutenção de rotas.

OLSR - É um protocolo pró-ativo que cria e atualiza periodicamente as suas tabelas de rotas, independente se estas serão usadas ou não. Devido à característica comum em todo protocolo baseado em estado de enlace, o OLSR também necessita de inundações realizadas periodicamente na rede, informando os estados dos enlaces de cada nó com seus respectivos vizinhos. Como a inundação acarreta uma sobrecarga da rede, o OLSR tenta aperfeiçoar este procedimento de inundação através da seleção

de nós que são responsáveis pelas inundações na rede, chamados de *MultiPoint Relays* (MPR). Para selecionar os MPRs é usada uma técnica que escolhe um grupo de vizinhos de um salto que seja capaz de alcançar uma maior quantidade de vizinhos de dois saltos. Assim, a inundação pode ser realizada de forma mais eficiente.

DSVD - É um protocolo pró-ativo e baseia-se no roteamento por vetor de distância. Suas características de funcionamento fazem com que este protocolo seja bastante semelhante ao *Routing Information Protocol* (RIP) [12]. As principais diferenças estão na convergência mais rápida e nas melhorias utilizadas para evitar o problema de contagem para o infinito.

Para a proposta apresentada neste trabalho poderiam ser utilizados os dois protocolos de roteamento reativos, o AODV e o DSR, pois os dois utilizam o procedimento de manutenção de rotas quando algum problema de desconexão na rede é identificado. No entanto, o mecanismo proposto foi comparado a outro mecanismo de detecção de maus comportamentos em redes ad hoc, que utiliza o DSR como protocolo de roteamento. Por esta razão, o DSR foi escolhido para trabalhar em conjunto com o sistema de detecção de nós egoístas proposto neste trabalho. O uso do mesmo protocolo de roteamento permite que os resultados sejam comparados de forma mais justa nas análises de desempenho das propostas. Portanto, para entender melhor o protocolo de roteamento DSR, as suas principais características de funcionamento são detalhadas a seguir.

2.1.2 Dynamic Source Routing (DSR)

Como citado anteriormente, o DSR utiliza uma técnica de roteamento baseada na origem, ou seja, o remetente informa a seqüência completa dos nós responsáveis pelo encaminhamento dos pacotes até um destinatário. Essa seqüência completa de nós é inserida no cabeçalho de cada pacote enviado, identificando os nós de origem, os nós intermediários e o destinatário da mensagem.

Quando um nó necessita de uma rota para um determinado destinatário, ele faz uma consulta nas informações de rotas contidas em cache ou nas respostas obtidas através do procedimento de descoberta de rotas. Assim, para enviar um pacote para outro nó, o

remetente constrói uma rota e a envia no cabeçalho do pacote, informando o endereço de cada nó da rota, através do qual o pacote será encaminhado ordenadamente até alcançar o destino. O remetente então transmite o pacote para o nó que estiver a um salto de distância, ou seja, para o próximo salto na rota. Cada nó, ao receber o pacote, verifica se é o destinatário da mensagem e, caso não seja, o pacote é encaminhado para o próximo salto informado na rota.

Cada nó móvel pertencente à rede ad hoc armazena em cache as rotas que já foram descobertas anteriormente no procedimento de descoberta de rotas. Esse cache é consultado sempre que um nó deseja enviar um pacote para algum destinatário pertencente à rede. Se a rota não for encontrada em cache, o remetente então inicia o procedimento de descoberta de rotas. Este procedimento é realizado através do envio de uma mensagem de solicitação de rota para a rede, no qual permite a descoberta de novas rotas para um determinado nó de destino. Após essa descoberta, a nova rota é armazenada em cache. É importante citar que cada rota em cache está associada a um temporizador, que ao expirar, resulta em sua remoção.

Nas redes ad hoc, quando um nó muda a sua posição ou simplesmente se desliga da rede, uma quebra de rota pode ocorrer. Assim, o DSR executa os processos de correção e atualização de rotas, chamado de procedimento de manutenção de rotas. Portanto, ao ser detectada uma desconexão entre o nó atual e o próximo salto da rota, o mecanismo de manutenção de rota é iniciado. A partir daí, o nó de origem é alertado sobre a quebra de rota. Ao receber o alerta, o nó de origem remove todas as rotas que contêm aquele enlace que foi classificado como inativo. Assim, o nó de origem verifica se existe uma rota secundária armazenada em cache e, se existir, esta rota será usada, caso contrário, é iniciado um novo procedimento de descoberta de rota para o destinatário do pacote. A seguir são mostrados os procedimentos de descoberta e manutenção de rotas, utilizados pelo DSR.

Descoberta de Rotas

O procedimento de descoberta de rotas permite que um nó descubra dinamicamente uma rota para qualquer outro nó em uma rede ad hoc, mesmo que este não esteja no seu raio de alcance. Um nó inicia o procedimento de descoberta de rotas através do

envio, por difusão, de uma mensagem de requisição de rota, conhecida como *Route Request* (RREQ). Esta mensagem contém os endereços do remetente e do destinatário da mensagem e é difundida na rede, em um processo de inundação, até que uma rota para o destinatário da mensagem seja encontrada. Desta forma, quando esta mensagem de descoberta de rota alcança o destinatário ou algum nó que conheça uma rota para o destinatário, é iniciado o procedimento de resposta à requisição de rota, através da mensagem conhecida como *Route Reply* (RREP). O RREP então informa os endereços de todos os nós pertencentes à rota entre o nó de origem e o destinatário.

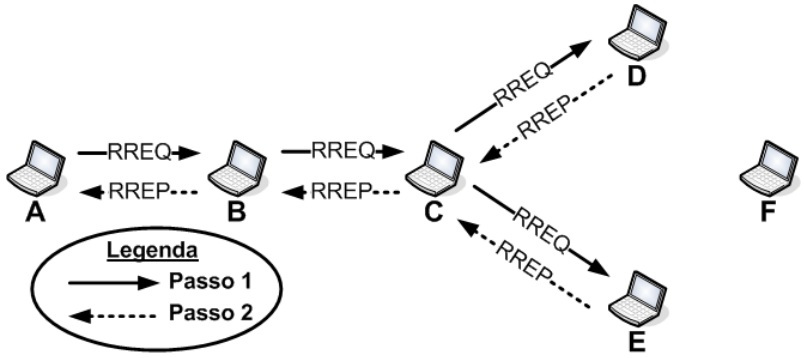


Figura 2.1: Descoberta de rotas no DSR.

De acordo com a Figura 2.1, o nó A deseja encontrar uma rota para o nó D. Logo, o nó A inicia o procedimento de descoberta de rota, enviando por difusão uma mensagem RREQ, contendo os endereços da origem e do destino do pacote. Ao receber o RREQ, o nó B checa se ele é o destinatário da mensagem. Como o nó B não é o destinatário da mensagem, então ele verifica se existe uma rota em cache para o destinatário. Ao observar que não existe uma rota em cache, o nó B adiciona o seu endereço na mensagem e encaminha por difusão o RREQ para todos os seus vizinhos. Portanto, o nó C seguirá o mesmo procedimento do nó B. Quando o nó D receber o RREQ e observar que é o destinatário da mensagem, então ele informará a rota descoberta para o nó A, através da mensagem RREP. Além disso, esta figura mostra a presença de um nó intermediário, representado por E, que também conhece uma rota para o nó D, formada pelos nós E, F e D. Logo, o nó E utiliza a rota em cache conhecida para construir uma nova rota do nó A ao nó D. Após ser construída a rota, o nó E envia uma mensagem RREP para o nó A, contendo uma nova rota para o nó D.

Além dos endereços de origem e de destino, cada RREQ possui um campo de gravação de rota, conhecido como *Route Record*. Este é o campo que as estações intermediárias utilizam para registrar os seus endereços, antes de repassarem a mensagem de requisição de rotas para os seus vizinhos. Cada RREQ também contém um identificador único de mensagem de requisição, conhecido como (*Request ID*), que é definido por um valor numérico, atribuído seqüencialmente. Este identificador é usado em conjunto com o endereço do nó de origem para evitar que a mesma requisição de rota seja processada e encaminhada mais de uma vez. Portanto, a cada requisição de rota recebida, os nós primeiramente verificam seus pares de identificadores (*Initiator Address, Request ID*). Desta forma, ao receber uma mensagem de requisição de rota, cada nó executa os seguintes passos:

- se o par (*Initiator Address, Request ID*) da requisição de rota atual for encontrado na lista de entradas das rotas mais recentes, então o nó descarta o RREQ e não o processa novamente. Desta forma, o nó não encaminha a mesma mensagem RREQ mais de uma vez, até que o tempo de vida do par de identificadores expire;

- se o endereço do nó já está listado no *Route Record*, então o RREQ é descartado e não será processado novamente. Assim, existe uma prevenção contra a ocorrência de *loops*, de modo que se o endereço do nó está presente no *Route Record*, significa que o RREQ já passou por ele uma vez e retornou;

- se o nó atual é o destinatário da requisição ou conhece alguma rota para o destinatário, então ele terá que iniciar o procedimento de resposta de rota para o nó de origem. Para isso, ele verifica se existe alguma rota em cache para o nó de origem que requisitou a descoberta de rota. Caso seja encontrada, esta rota em cache será usada para o envio do RREP, caso contrário, a seqüência de endereços registrados no *Route Record* da mensagem RREQ é invertida e utilizada para enviar a resposta a esta requisição;

- como último passo, se o nó não é o destinatário da mensagem ou desconhece uma rota para o destinatário, então o nó adiciona o seu endereço no campo *Route Record* e repassa a mensagem de requisição de rota por difusão.

Manutenção de Rotas

Os protocolos de roteamento baseados nos modelos convencionais, tais como vetor de distância e estado do enlace, necessitam de algum procedimento de manutenção e/ou atualização de rotas. Nos protocolos baseados em roteamento por estado de enlace são utilizadas atualizações periódicas dos enlaces ativos, que implicam em constantes cálculos de rotas em cada nó. Por outro lado, no DSR não existem pacotes de atualização, pois ele utiliza outro método para determinar se uma rota está válida ou não. Este método é conhecido como procedimento de manutenção de rotas.

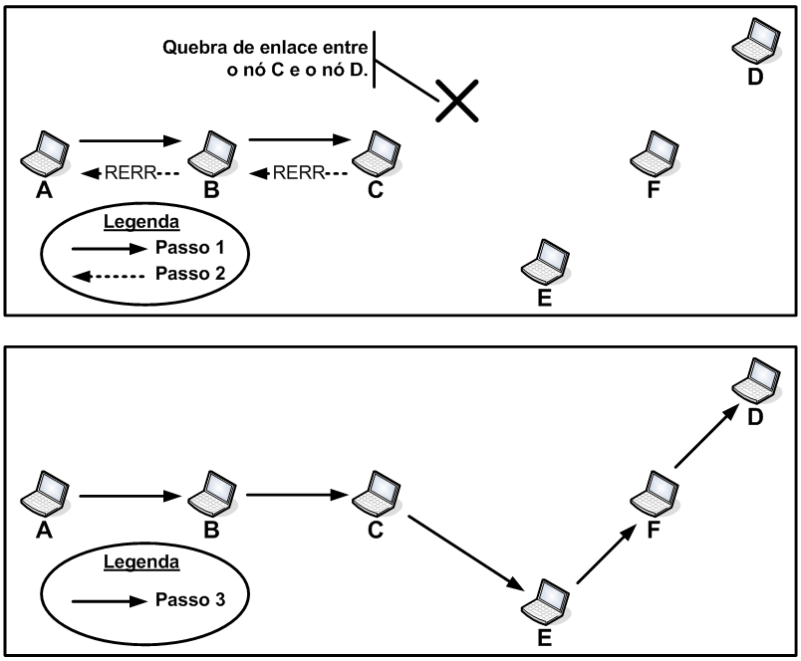


Figura 2.2: Manutenção de rotas no DSR.

Seguindo o exemplo apresentado na Figura 2.1, pode-se considerar que foram descobertas duas rotas, formadas pelos conjuntos de endereços $Rota1 = \{A, B, C, D\}$ e $Rota2 = \{A, B, C, E, F, D\}$. Na Figura 2.2 são apresentados o procedimento de envio dos pacotes após a descoberta de rotas, juntamente com o procedimento de manutenção de rotas. Neste exemplo, o nó A seleciona a $Rota1$ para enviar os seus pacotes para o nó D. Durante o encaminhamento dos pacotes, o nó C observa que o nó D não está mais em seu alcance de transmissão. A partir deste instante, o nó C executa o passo dois, ou seja, o nó C inicia o procedimento de manutenção de rotas, relatando a quebra do enlace entre ele e o nó D, através de uma mensagem de erro de rota. Esta mensagem, chamada

de *Route Error* (RERR), contém os endereços dos nós C e D, e tem como destino o nó que originou os pacotes, o nó A. Ao receber o RERR, o nó A exclui a $Rota1$ de seu cache e verifica se existe outra rota para o nó D. Ao encontrar a $Rota2$ em seu cache, o nó A inicia o passo três, enviando os seus pacotes para o nó D através desta rota, finalizando assim o processo de manutenção e atualização de rotas.

Para detectar a quebra de enlace, o DSR necessita de informações passadas pelo protocolo IEEE 802.11 da camada MAC. Portanto, ao receber um pacote da camada de rede, o MAC envia o quadro para o próximo salto e aguarda a confirmação de recebimento do próximo salto na rota. Caso o protocolo da camada MAC não receba nenhuma confirmação de recebimento, então o MAC envia um sinal para a camada de rede. Ao receber esse sinal, o DSR inicia o procedimento de manutenção de rotas e classifica o enlace como inativo. Assim, o nó que detecta a quebra de enlace remove as rotas que contêm aquele enlace e envia a mensagem de erro de rota (RERR) para o endereço de origem da rota. Além disso, as estações intermediárias que recebem a mensagem de RERR também removem de seu cache todas as rotas que possuem aquele enlace que foi classificado como inativo.

Pode-se concluir que o reconhecimento de recebimento dos pacotes enviados por um nó é dependente da camada de enlace. Entretanto, para realizar o reconhecimento sem utilizar a camada de enlace, os nós poderiam ficar em modo promíscuo, observando o encaminhamento dos pacotes realizados por seus vizinhos. Este método é conhecido como reconhecimento passivo, já que não há qualquer interação entre a troca de mensagens pelos nós envolvidos. Portanto, é importante citar que ambos os nós envolvidos em um reconhecimento passivo devem ter o mesmo alcance de transmissão. Existem outros métodos de reconhecimento, tal como o de inserção de um bit quando se deseja obter reconhecimento do pacote, além do método de reconhecimento realizado nas camadas de transporte e de aplicação.

Otimizações

No DSR são apresentados alguns procedimentos que otimizam o *overhead* de mensagens de controle na rede, aperfeiçoando também o modelo de manutenção de rotas. As principais otimizações são divididas em três, que são apresentadas com mais detalhes a

seguir [9].

Uso Completo do Cache de Rotas - Como primeiro exemplo de otimização no uso completo de *cache* pode ser citado o mecanismo de adição de rota, no qual cada estação registra em *cache* as rotas de todos os pacotes encaminhados. Com esta otimização é possível conhecer novas rotas sem a necessidade de executar os procedimentos de descoberta e manutenção de rotas, reduzindo o tráfego das mensagens de controle e, conseqüentemente, diminuindo o *overhead* de mensagens de controle na rede. O segundo exemplo de otimização é a utilização do *cache* para responder às requisições de rota, evitando que os pacotes fiquem circulando na rede em busca do destinatário da mensagem. Entretanto, com esta otimização, aumentam as chances de duas estações responderem à requisição de rota ao mesmo tempo para a origem. Outro problema relacionado a este segundo exemplo é o anúncio desnecessário de rotas, não considerando outras métricas que melhoram o desempenho da rede. Portanto, para tentar resolver este problema, foi proposta uma solução que obriga que cada estação aguarde um tempo aleatório antes de responder a um *Route Request*. Como último exemplo de otimização por uso de *cache*, é proposto um mecanismo que limita o número de saltos a partir da origem. Dessa forma, no momento do envio de cada pacote de requisição, durante as descobertas de rota, são usados os seguintes procedimentos: limitar o número de saltos com valor igual a um, ou seja, fazer requisições de rotas somente aos seus vizinhos, caso eles tenham a rota em *cache* para o destinatário; definir um valor máximo de saltos de forma incremental, até que a mensagem possa atingir todos os nós da rede, se dentro de um pequeno intervalo de tempo a requisição não for respondida.

Redução do Tamanho das Rotas - A idéia principal deste método de otimização é reduzir o número de saltos das rotas utilizadas na rede. Por exemplo, quando um nó intermediário recebe um pacote endereçado para outro nó, ele pode checar se o pacote pode ser roteado por ele mesmo, pois ele possui uma rota menor do que a que está sendo utilizada. Para isso acontecer, este nó deve enviar um *Route Reply* para o remetente do pacote, anunciando a melhor rota descoberta. Com esta otimização, os nós da rede que alteram as suas posições e ficam com oportunidades de contato

direto de transmissão para o nó de destino, podem informar esta mudança para o nó remetente, evitando que o pacote seja encaminhado por uma rota com um maior número de saltos.

Tratamento de Erros Melhorado - Utiliza uma técnica que otimiza o procedimento de manutenção de rotas, principalmente quando existem constantes quebras de enlaces nas redes ad hoc. Se um nó intermediário detecta um enlace quebrado e possui uma rota alternativa em *cache*, ele pode tentar salvar o pacote recebido enviando-o pela rota conhecida em *cache* [13]. Além disso, ele poderá informar a nova rota para o nó que originou os pacotes.

2.2 Segurança em Redes Ad Hoc

Atualmente, as pesquisas realizadas na área de redes ad hoc têm apresentado novas soluções nas funcionalidades de auto-cofiguração e auto-manutenção destas redes. Para isso, estas soluções assumem que as redes ad hoc são formadas por ambientes confiáveis e cooperativos, analisando somente os aspectos relacionados aos problemas de canal de acesso e roteamento em múltiplos saltos. Entretanto, a segurança deve ser tratada como um fator principal para o correto funcionamento destas redes e, além disso, oferece um ambiente tolerante a possíveis ataques [14]. Como todo ambiente colaborativo, as redes ad hoc também só podem ser consideradas seguras quando oferecerem um ambiente que satisfaça os principais requisitos de segurança, que são a disponibilidade, a confidencialidade, a autenticidade, a integridade, o não-repúdio e a privacidade.

Com base na necessidade de criação de um ambiente seguro nas redes sem fio, a pesquisa na área de segurança em redes ad hoc pode ser considerada como um tópico em aberto, principalmente por possuir um conjunto de desafios particulares, que dificultam a criação de um modelo de segurança ideal. Estes desafios podem ser representados por características particulares, tais como a arquitetura de rede aberta, o meio sem fio compartilhado, a limitação de recursos computacionais, e a topologia de rede altamente dinâmica. Conseqüentemente, as soluções existentes de segurança em redes cabeadas não podem ser aplicadas diretamente nas redes ad hoc. De acordo com [15], os componentes que fazem

parte de um modelo de segurança podem ser apresentados como mostrados na Figura 2.3.

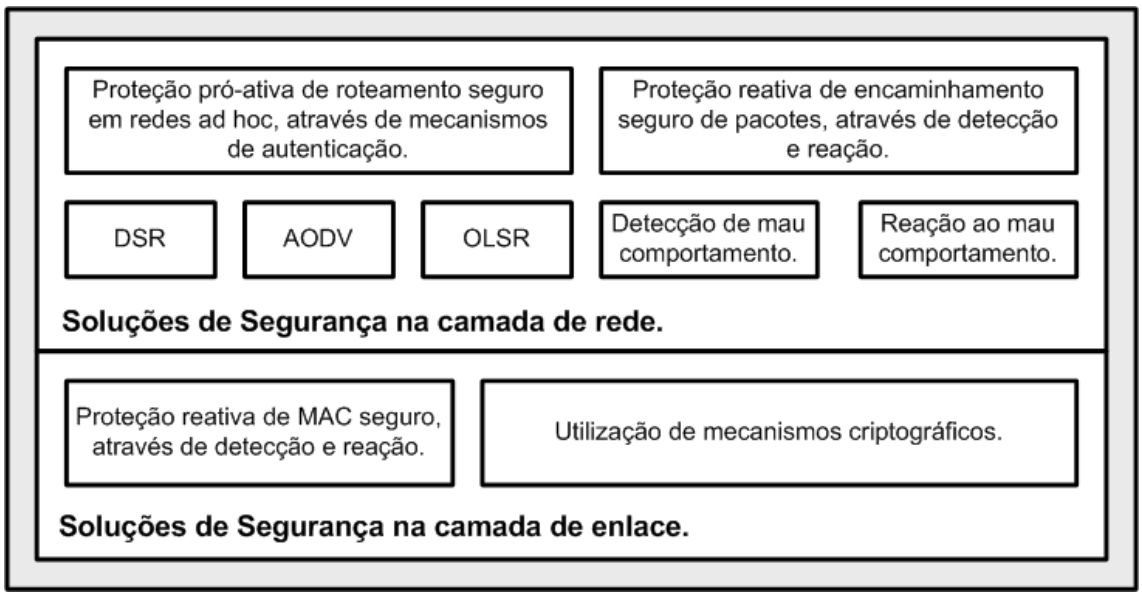


Figura 2.3: Componentes de um modelo de segurança em redes ad hoc.

De acordo com a Figura 2.3, pode-se observar a existência de mecanismos que objetivam a prevenção de ataques na camada de rede, representados pelo módulo de proteção pró-ativa. Os mecanismos pró-ativos trabalham em conjunto com os mecanismos reativos, que são responsáveis pela detecção e punição de ataques que conseguiram burlar os mecanismos de prevenção. Tanto os mecanismos pró-ativos como os mecanismos reativos são implementados em conjunto com os protocolos de roteamento da camada de rede. Já na camada de enlace são apresentados um modelo de detecção e reação a ataques e os mecanismos criptográficos.

Como citado anteriormente, é assumido que as redes ad hoc de múltiplos saltos são formadas somente por nós colaborativos. Entretanto, essa premissa não pode ser considerada verdadeira em redes ad hoc, pois a cooperação é uma característica assumida e não forçada. Portanto, nós maliciosos podem facilmente corromper as operações da rede, por violarem as especificações assumidas pelos protocolos. Seguindo as características apresentadas dos desafios de segurança em redes ad hoc, é possível listar alguns ataques que podem ocorrer em redes ad hoc.

2.2.1 Ataques

Ao apresentar o modelo de segurança em redes ad hoc na Figura 2.3, é possível observar que os ataques podem ter como objetivo a degradação dos protocolos usados tanto na camada de rede quanto na camada de enlace. No entanto, este trabalho é voltado somente para o estudo de soluções de segurança na camada de rede. Assim, são apresentadas somente as características referentes às vulnerabilidades encontradas na camada de rede, que, ao serem alvos de um ataque, causam uma redução no desempenho das rede ad hoc. Portanto, os ataques aos protocolos de roteamento em redes ad hoc, geralmente, caminham para uma destas duas categorias:

Ataques por quebra de rotas - Os atacantes tentam fazer com que os pacotes de dados legítimos sejam roteados de forma incorreta. Como exemplo, pode ser citado o ataque de buraco negro (*Blackhole*) [16], no qual um nó malicioso atrai as rotas para si e realiza o descarte de pacotes;

Ataques por consumo de recursos - Os atacantes injetam pacotes dentro da rede, na tentativa de consumir os recursos disponíveis, tais como largura de banda e, até mesmo, a memória ou o processamento dos nós da rede.

Além desta divisão, os ataques nas redes ad hoc podem ser realizados de duas formas, que se classificam em ataques passivos e ataques ativos. Os ataques passivos não afetam as operações da rede, sendo caracterizados pela espionagem, não alterando o conteúdo dos dados. Por outro lado, os ataques ativos são aqueles em que o atacante cria, altera, descarta ou inviabiliza o uso dos dados em trânsito [17].

Ataques Passivos

Os ataques passivos são aqueles que os atacantes não participam ativamente na degradação do desempenho da rede. No entanto, os atacantes são classificados como espiões do tráfego de dados na rede. Este ataque não é considerado tão prejudicial porque o atacante não modifica as informações ou descarta qualquer pacote. O atacante somente se aproveita do meio inseguro para roubar informações.

Para se proteger de ataques de espionagem, existem soluções implementadas nas camadas superiores, sendo capazes de proteger a rede dos ataques de espionagem, sem quebrar o sigilo da informação. Entretanto, ao utilizar soluções em camadas mais altas, a topologia da rede pode ser exposta para o atacante, pois ataques de espionagem podem ser realizados nas informações de roteamento. Portanto, o sigilo através de mecanismos de criptografia se torna uma necessidade nos protocolos de roteamento [18].

Ataques Ativos

Os ataques ativos são aqueles que os atacantes participam ativamente do comprometimento das operações e serviços da rede. Neste ataque, o atacante pode descartar, modificar e replicar pacotes, como também fabricar mensagens e se passar por outros nós da rede. A seguir, são descritos alguns destes ataques com base em seu modo de execução:

Descarte de pacotes

Neste ataque, um atacante pode descartar todos ou apenas alguns dos pacotes recebidos por ele e, dessa forma, comprometer o correto funcionamento da rede, ver Figura 2.4. Estes ataques podem ser classificados de acordo com a sua forma de execução. Logo, ao serem descartados todos os pacotes, o ataque é chamado de buraco negro. Já para o descarte seletivo, o ataque é chamado de buraco cinza.

No ataque de buraco negro, o atacante participa do procedimento de descoberta de rotas, atraindo as rotas da rede para si. Após realizar esta atração de rotas, ele inicia o procedimento de descarte em todos os pacotes de dados recebidos [16]. Alguns autores, tais como [17] e [19] afirmam que todos os tipos de pacotes são descartados no ataque de buraco negro, inclusive os pacotes de controle. No entanto, isto se torna inviável porque se o nó não encaminha os pacotes de controle, ele nunca será usado nas rotas da rede e, portanto, não poderá descartar os pacotes de dados em nenhum momento.

No ataque de buraco cinza, o atacante seleciona, de forma aleatória ou determinística, os pacotes que serão descartados. Este é considerado um ataque difícil de ser detectado, pois quanto mais aleatório for o descarte, mais difícil será a detecção. O ataque de buraco cinza utiliza a mesma técnica do buraco negro para atrair as rotas. Neste ataque, o nó pode encaminhar os pacotes de controle sem alterar qualquer informação de rota. Além disto,

os pacotes de controle também podem ser escolhidos no sorteio que definirá os pacotes que serão descartados [20].

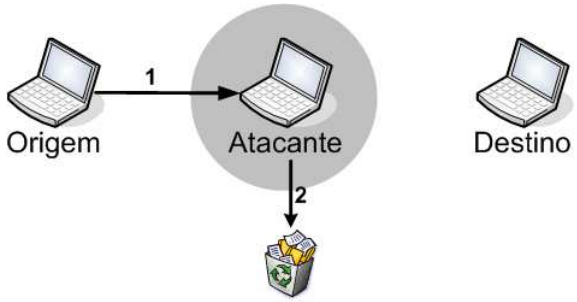


Figura 2.4: Descarte de pacotes.

Modificação das informações dos pacotes

Para funcionarem corretamente, alguns protocolos de roteamento assumem que as estações alteram o conteúdo dos pacotes que trafegam na rede, principalmente nos campos de controle do protocolo. Através dessas mudanças nas informações contidas nos pacotes, um nó atacante é capaz de enviar pacotes por rotas maiores, alterar campos importantes nos pacotes de controle dos protocolos de roteamento e, até mesmo, passar informações incorretas para os nós da rede. Este último exemplo é conhecido como ataque bizantino, e tenta enganar os protocolos de roteamento, modificando as informações de roteamento [21] [22].

O nome atribuído ao ataque bizantino é uma referência ao problema dos generais bizantinos [23]. Este ataque ocorre quando um ou mais nós da rede ad hoc tentam confundir o protocolo de roteamento, através da alteração das informações contidas nos campos de controle. Dentre as formas usadas para provocar este ataque, podem ser citadas: a alteração das informações de roteamento das mensagens; o envio de falsos pacotes de roteamento; a escolha de piores caminhos; e a formação de loops na rede.

O ataque bizantino é difícil de ser combatido, pois para os nós da rede, o funcionamento do protocolo de roteamento está sendo realizado normalmente, entretanto, os pacotes estão transportando informações alteradas. Para entender melhor o problema dos generais bizantinos, é apresentado um exemplo na Figura 2.5, que mostra um general passando informações diferentes para dois soldados.

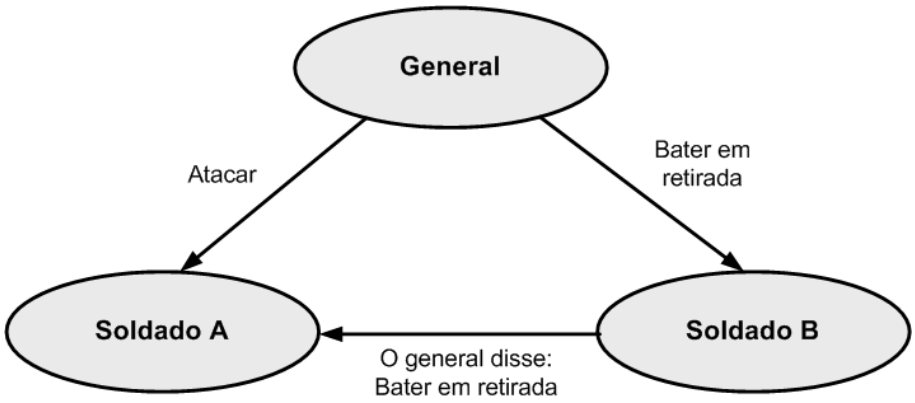


Figura 2.5: Problema dos generais bizantinos.

Disfarce

O principal objetivo do ataque de disfarce [24] é fazer com que os outros nós da rede sejam incapazes de identificar um nó atacante. O disfarce pode ser realizado através da criação de falsas identidades ou da clonagem de identidades de outros nós da rede. Dessa forma, os nós atacantes podem executá-lo nas camadas que utilizam algum tipo de identificação, ou seja, tanto na camada de enlace como na camada de rede. Portanto, o nó atacante é capaz de trocar o endereço MAC ou o endereço IP dele, para se passar por um nó pertencente à rede.

O ataque mais comum de disfarce é conhecido como Sybil [25]. Neste ataque, o nó atacante é capaz de trocar diversas vezes a identidade dele, criando-as ou clonando-as de outros nós da rede. Geralmente, estes ataques ocorrem na camada de rede e podem ser combatidos através do uso de mecanismos fortes de autenticação.

Criação de mensagens

Apesar de ter sido citado no ataque por modificação de pacotes, o ataque bizantino também pode ser citado neste tipo de ataque, pois falsas mensagens podem ser criadas na rede e envidas para os outros nós. Além disso, este ataque pode ser usado para inundar uma rede, causando um desperdício do uso de memória dos outros nós, como também o consumo desnecessário de processamento.

O estouro da tabela de roteamento [26] pode ser citado como um ataque de criação de mensagens. Neste ataque, o nó malicioso tenta sobrecarregar o protocolo de roteamento

através do envio de descobertas de rotas para nós inexistentes. O ataque por estouro da tabela de roteamento é prejudicial, principalmente, aos protocolos de roteamento pró-ativos, pois estes armazenam todas as rotas anunciadas pelos nós vizinhos. Como soluções existentes para combater este ataque, podem ser usadas técnicas para limitar o número máximo de rotas nas tabelas de roteamento, além disso, obrigar que os anúncios de rotas só possam ser recebidos quando forem originados de nós autenticados.

Replicação de pacotes

De acordo com a Figura 2.6, este ataque tem como objetivo principal aumentar o desperdício de recursos da rede ad hoc, além de ocupar o meio de transmissão. Para isso, o nó envia réplicas de pacotes de roteamento antigos ou atuais para a rede.

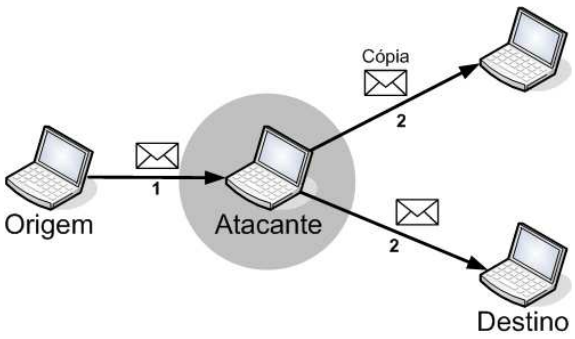


Figura 2.6: Replicação de pacotes.

Além destes ataques citados, existem diversos ataques em redes ad hoc que estão sendo estudados. Dentre estes ataques, podem ser colocados como exemplos o túnel de minhoca (*Wormhole*) [27][1] e o ataque da pressa[2].

2.2.2 Desafios e Soluções de Segurança em Redes Ad hoc

Ao tentar criar uma infra-estrutura de segurança eficiente para redes ad hoc, é importante considerar que estas redes apresentam desafios particulares em suas funcionalidades,

[1]Este ataque é ralizado através da criação de túneis. Desta forma, os nós maliciosos podem informar um caminho de apenas um salto até o destino, ao invés de informar a existência de vários saltos que realmente existem.

[2]Neste ataque, um nó malicioso tem como objetivo enviar respostas às requisições de rota de forma mais rápida do que o normal.

principalmente por necessitarem da colaboração dos nós no roteamento e no encaminhamento de pacotes. Além disso, um canal sem fio pode ser acessado tanto por nós maliciosos quanto por nós cooperativos. Como resultado, torna-se difícil definir uma linha de defesa precisa para estes ambientes, pois o limite que separa a segurança de uma rede interna da segurança de uma rede externa ad hoc ainda permanece como um desafio [15].

Como mostrado na Figura 2.3, a segurança em redes ad hoc pode ser dividida em dois grupos: segurança reativa e segurança pró-ativa. Os grupos pró-ativos tentam solucionar os problemas de segurança na primeira linha de defesa, ou seja, tipicamente através de mecanismos criptográficos. Por outro lado, o grupo reativo tenta detectar as anormalidades na rede para, posteriormente, iniciar alguma resposta que seja capaz de combatê-las. Cada grupo de segurança citado tem sua importância ao ser usado nas redes ad hoc, pois todos tentam oferecer soluções de segurança aos diferentes problemas encontrados. Por exemplo, os protocolos de roteamento seguro adotam características pró-ativas de segurança na troca de mensagens realizada entre os nós, enquanto que a soluções reativas são usadas nos problemas de segurança que ocorrem no encaminhamento de pacotes na rede.

Devido à ausência de uma linha de defesa clara, as soluções de segurança em redes ad hoc devem ser construídas a partir da união dos grupos reativos e pró-ativos. Assim, é possível concluir os três passos principais para oferecer um ambiente seguro em redes: a prevenção, a detecção e a reação. A prevenção utiliza técnicas que dificultam o acesso de nós atacantes nas redes ad hoc. Entretanto, é mostrado que a segurança não está completamente livre de estações maliciosas e que o sistema não está totalmente seguro. Assim, quando um nó malicioso é capaz de passar pela primeira linha de defesa, os mecanismos de detecção e de reação são utilizados para detectá-lo e puni-lo. Portanto, os mecanismos reativos também são indispensáveis para as soluções de segurança em redes ad hoc.

Neste trabalho são abordados os sistemas de detecção e reação[3] aos maus comportamentos ocorridos em redes ad hoc.

[3]No texto os termos resposta e punição são usados como sinônimos de reação

Capítulo 3

Sistemas de Detecção e Resposta à Intrusão em Redes Ad Hoc

SISTEMA de detecção de intrusão (SDI) é a principal solução utilizada no conjunto de mecanismos reativos de segurança em redes. O primeiro requisito para se usar um SDI é definir os comportamentos que indicam a presença de um intruso na rede. Para isto, devem ser padronizados os conjuntos de comportamentos que implicam na classificação de ações normais ou anormais na rede. Além deste requisito, existe a necessidade de utilização de mecanismos de autenticação de nós na rede, que são oferecidos pelos sistemas que fazem parte do conjunto pró-ativo de segurança em redes.

3.1 Sistemas de Detecção e Resposta à Intrusão

Um sistema de detecção de intrusão pode ser definido como um alerta de segurança gerado a cada detecção de intrusos realizada em uma rede. O SDI utiliza mecanismos de defesa que detectam os maus comportamentos realizados e tentam combater estes maus comportamentos através de punições, aumentando a segurança da rede. Para realizar as detecções, o SDI monitora continuamente todos os eventos realizados pelas estações da rede e envia alertas quando detecta os eventos considerados suspeitos. Para evitar os maus comportamentos detectados, os SDIs podem utilizar apenas mensagens de alerta

como também realizar bloqueios de conexão em uma rede ad hoc. Em outras palavras, detecção de intrusão é um processo que detecta e pune as estações que realizam maus comportamentos em uma rede. De acordo com Heady *et al* [28]., o mau comportamento pode ser definido como qualquer ação que comprometa a integridade, a confidencialidade ou a disponibilidade dos recursos.

Como citado no Capítulo 2, existem sistemas responsáveis pela primeira linha de defesa em uma infra-estrutura de segurança. Estes sistemas são utilizados de forma pró-ativa contra as ações que comprometem o funcionamento de uma rede. Nas redes cabeadas, os sistemas de *firewalls* [29] podem ser citados como um destes sistemas pertencentes à primeira linha de defesa. Os *firewalls* são utilizados para restringir e liberar acessos a partir de endereços IP, portas ou serviços de uma rede, seguindo uma política de permissões previamente estabelecida.

Ao obter o acesso em uma rede, seja através de configurações pré-definidas ou através de ataques, uma estação maliciosa pode executar ações que comprometam o correto funcionamento da rede. A partir desta característica, torna-se necessária a utilização de uma segunda linha de defesa, como, por exemplo, um Sistema de Detecção de Intrusão (SDI), para combater estas estações maliciosas [30]. Ao considerar esta característica de segunda linha de defesa, o SDI não pode ser confundido com outros mecanismos que fazem parte do conjunto reativo de segurança, tais como:

- anti-vírus, pois estes são desenvolvidos para detectar programas maliciosos, tais como vírus, *trojans* e *worms*;
- sistemas de registros, pois são sistemas que objetivam somente o registro das atividades executadas em um sistema operacional;
- ferramentas que checam vulnerabilidades, pois são responsáveis por localizar falhas de segurança em sistemas operacionais e em redes.

Dentre os elementos básicos que formam um SDI, a coleta de dados, originada pelo monitoramento de eventos, é considerada o fator principal para atingir um funcionamento mais eficiente, pois estes dados que caracterizam o comportamento das estações. Os dados

coletados podem ser obtidos através de entradas de teclados, registros de comandos ou registros de eventos realizados na rede. Desta forma, estes dados podem ser utilizados no momento que estão sendo monitorados, podendo também ser armazenados ou removidos a qualquer instante da base utilizada pelo SDI. Para aumentar ainda mais a eficiência de um SDI é importante que seja obtida uma grande quantidade de dados monitorados, pois quanto maior for o volume de dados, mais precisa será a detecção. Seguindo estas características apresentadas, um SDI pode analisar estes eventos monitorados seguindo dois modos de detecção: a detecção baseada em assinaturas e a detecção baseada em anomalias [30].

Detecção baseada em anomalias - Neste modo de detecção, o SDI utiliza uma base de informações contendo todos os padrões de comportamentos normais conhecidos, ver Figura 3.1. Portanto, qualquer comportamento executado por uma estação que se desvia desse padrão é classificado como uma intrusão. A vantagem deste modo de detecção é a fácil adaptação às modificações dos ataques, não permitindo que novos ataques possam comprometer a rede. Por outro lado, o principal problema está relacionado à elevada quantidade de falso-positivos gerada nas detecções.

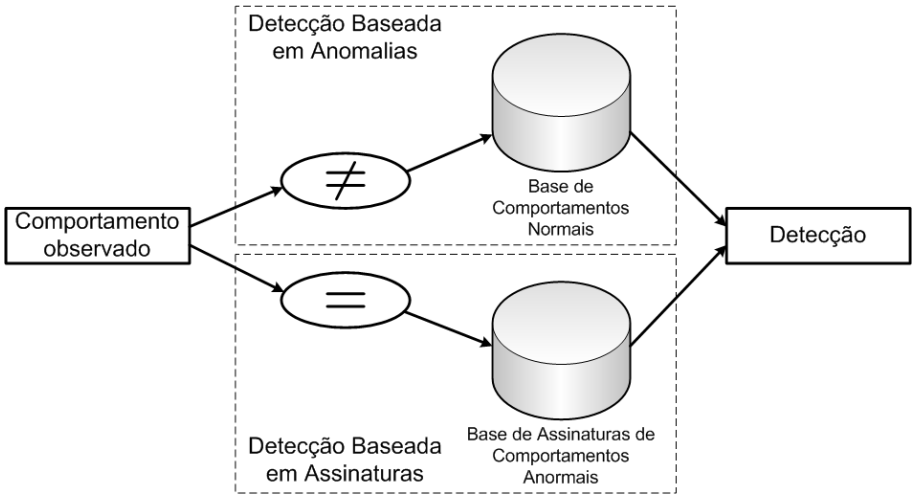


Figura 3.1: Modelos de detecção de intrusão.

Detecção baseada em assinaturas - Também chamado de modo de detecção baseado em mau uso [30], este modo mantém uma base de assinaturas contendo todos os tipos de maus comportamentos conhecidos. Assim, qualquer evento igual a uma

dessas assinaturas é classificado como uma intrusão, ver Figura 3.1. A vantagem deste modo de detecção é a maior precisão na detecção de comportamentos maliciosos, pois o mau comportamento detectado é conhecido pelo SDI. Entretanto, este modo possui uma falha grave, pois permite que um atacante obtenha sucesso ao se desviar um pouco dos comportamentos que determinam a assinatura registrada na base do SDI. Para tentar minimizar este problema, é necessário utilizar mecanismos que executem constantes atualizações nas assinaturas dos comportamentos maliciosos, implicando em um aumento na base de assinaturas e um maior consumo de recursos.

Além destes dois modos de detecção, existe um terceiro modo, que utiliza tanto a detecção baseada em anomalias como a detecção baseada em assinaturas. Primeiramente, a detecção é realizada pelo modo baseado em anomalias, detectando os comportamentos que se desviam do padrão de normalidade da rede. Para tentar reduzir a quantidade de falso-positivos, a detecção baseada em assinaturas, então, verifica se este comportamento detectado faz parte de sua base de assinaturas. Caso seja comprovada a existência deste evento na base de assinaturas, então o nó conclui que a detecção está correta e inicia o procedimento de resposta à intrusão [31].

Este terceiro modo de detecção tem uma maior probabilidade de detectar corretamente um nó malicioso na rede. Entretanto, todos os problemas de ambos os modos de detecção citados anteriormente são herdados, assim como novos problemas são criados. Como exemplos destes novos problemas, podem ser citados: a necessidade do dobro de consumo de processamento; a atualização constante da base de assinaturas do modelo baseado em assinaturas; e a manutenção de duas bases de assinaturas no SDI, uma de comportamentos normais e outra de comportamentos anormais.

3.2 Resposta à Intrusão

As respostas às detecções realizadas pelos sistemas de detecção podem ser aplicadas de diferentes formas. Assim, alguns sistemas utilizam somente as notificações geradas pelas detecções e enviam mensagens de alerta para os outros nós da rede, relatando a

existência de nós suspeitos [32] [33]. Existem também os sistemas que utilizam respostas manuais às intrusões. Estes sistemas são auxiliados pela informação sobre o tipo de mau comportamento detectado, que ajudam na definição da melhor resposta que será aplicada pelo administrador da rede [34] [35]. Os sistemas manuais de respostas às intrusões têm o mesmo problema dos sistemas de alertas, que é o atraso na aplicação de punição. Portanto, existem soluções que tentam minimizar esse problema, utilizando respostas automáticas [36] [37] [38].

3.3 Sistemas de Detecção e Resposta à Intrusão em Redes Ad Hoc

Para criar um sistema de detecção e resposta à intrusão em redes ad hoc é necessário considerar o aspecto principal que qualquer SDI deve tentar alcançar, a eficiência. A eficiência é alcançada quando o SDI utiliza somente a quantidade necessária de informações para obter detecções precisas sobre as intrusões. Portanto, um sistema não pode ser somente eficaz, ele tem que ser eficiente, pois ser eficaz significa somente a detecção do mau comportamento, sem se importar com a quantidade de recursos que estão sendo gastos e com a quantidade de informações que estão sendo geradas desnecessariamente.

Em redes ad hoc, a eficiência deve ser colocada em primeiro lugar, pois, geralmente, os recursos destas redes são limitados. Desta forma, um SDI em redes ad hoc é eficiente quando atinge a precisão na detecção de maus comportamentos nas redes, economizando, ao máximo, os recursos computacionais dos nós. Para que esta eficiência possa ser alcançada, alguns requisitos desafiadores devem ser considerados:

- o SDI não pode trazer novos problemas para as redes ad hoc, ou seja, o SDI deve tentar ser robusto ao ponto de não enfraquecer os outros sistemas utilizados pelos nós de uma rede ad hoc;
- o SDI tem que ser transparente para os usuários e para os outros serviços da rede, mantendo-se sempre ativo para detectar os maus comportamentos;

- o SDI deve utilizar uma capacidade mínima de recursos computacionais para detectar e punir as intrusões. Portanto, torna-se inviável o uso de SDIs que sobrecarregam os recursos e que utilizam algoritmos muito complexos;

- o sistema tem que ser capaz de se manter ativo em casos de desastres nas redes ad hoc. Dessa forma, o SDI deve ser tolerante a falhas, permitindo uma recuperação dos sistemas corrompidos e possibilitando o retorno do estado anterior ao desastre;

- além de detectar e punir os nós que realizam maus comportamentos em redes ad hoc, o SDI tem que fortalecer o seu sistema para não permitir que ele seja facilmente danificado ou tenha as suas funções interrompidas por nós maliciosos. Além disso, é importante que o SDI seja capaz de perceber se ele está sendo alvo de algum tipo de ataque;

- o SDI deve ter um sistema próprio de respostas às intrusões detectadas, preferencialmente, sem intervenções humanas;

- o SDI deve ser capaz de reduzir a quantidade de falso-positivos e falso-negativos gerados pelas detecções, pois estas implicam em uma maior precisão nas punições;

- o SDI deve oferecer suporte para operar em conjunto com outros SDIs na detecção e na resposta à intrusão. Atualmente, existem esforços para especificar as operações dos SDIs, tais como a *Internet Engineering Task Force* (IETF) *Intrusion Detection Work Group* (IDWG) [39].

Ao observar todos esses requisitos para a criação de um SDI eficiente, pode-se concluir que as vulnerabilidades e as características particulares das redes ad hoc são fatores que dificultam a implementação. Devido à característica da ausência de uma infraestrutura de auditoria central, os SDIs possuem restrições para realizarem o monitoramento de comportamentos. Como conseqüência, os SDIs em redes ad hoc são limitados a monitorar somente o tráfego de entrada e saída em cada nó.

Outra característica importante é que cada nó da rede ad hoc consegue observar apenas certa parcela dos nós da rede, determinada pelo alcance de seu rádio. Assim, os algoritmos passam a requerer a implementação de algum mecanismo capaz de trocar informações

sobre as detecções realizadas, que implica em uma maior necessidade de utilização de modelos de detecção distribuídos. Entretanto, caso seja usado um algoritmo de detecção que trabalhe de forma distribuída, é importante que seja adicionado algum mecanismo de confiança na rede, pois os nós da rede precisam confiar nas informações de detecções passadas por terceiros.

Uma dificuldade de implementação dos algoritmos de detecção em redes ad hoc está relacionada à elevada quantidade de falso-positivos gerada nas detecções. Portanto, os SDIs devem ser cautelosos nas detecções de comportamentos suspeitos, pois existe uma dificuldade em distinguir um ataque de um problema temporário da rede ad hoc. Dentre esses problemas temporários, podem ser citados: as colisões ambíguas [5], o desvanecimento e os atrasos no encaminhamento dos pacotes.

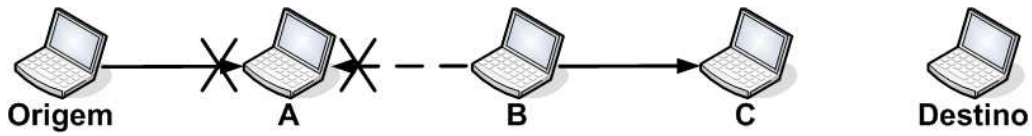


Figura 3.2: Colisão ambígua.

De acordo com a Figura 3.2, a colisão ambígua ocorre quando o nó A não percebe que o pacote foi encaminhado corretamente pelo nó B. Seguindo o exemplo, considere o caso de uma rede de múltiplos saltos e que o nó A envia um pacote para B no mesmo instante em que B encaminha o pacote para C. Uma colisão ocorre em A, pois este não pode detectar que o pacote foi encaminhado corretamente para C. Portanto, devido a este problema, o nó A conclui que o nó B está se comportando de forma anormal na rede e o classifica como um nó malicioso, gerando um falso-positivo.

Os falso-positivos relacionados aos problemas de desvanecimento ocorrem quando as variações do canal provocam perdas das informações que confirmam um encaminhamento de pacotes. Seguindo o exemplo da Figura 3.2, o nó A pode não perceber que o pacote foi encaminhado pelo nó B se um problema de desvanecimento ocorrer entre eles. Desta forma, a variação do canal faz com que o nó A classifique o nó B como um nó malicioso. Trabalhos de análises de comportamento dos canais sem fio [4] [3] mostram que, mesmo para redes estáticas, as variações dos canais provocam elevadas taxas de perdas. Isso se deve ao fato do desvanecimento ocorrer com uma freqüência elevada, dificultando a

implementação dos mecanismos comuns de roteamento [40].

O último fator que favorece a ocorrência de falso-positivos está relacionado ao atraso no encaminhamento dos pacotes, que ocorre devido aos mecanismos de controle de acesso ao meio das redes ad hoc. Em outras palavras, uma estação que deve encaminhar um pacote pode ser classificada como maliciosa por esperar um longo tempo a liberação do meio sem fio. Analisando o exemplo da Figura 3.2, a estação B pode ser classificada como maliciosa pelo nó A por aguardar a liberação do meio sem fio, que está sendo usado por outro nó, como por exemplo, o nó C.

As respostas aos nós detectados nas redes ad hoc podem ser aplicadas de diferentes formas, podendo ou não considerar um nível de punição para cada tipo de comportamento anormal detectado. Dentre as formas de respostas aplicadas, podem ser citadas como exemplos: a reinicialização do canal de comunicação entre os nós, forçando a redistribuição de chaves; e a exclusão de nós detectados nas rotas utilizadas pelos nós cooperativos da rede, evitando a utilização de nós maliciosos. Além destas definições sobre a forma de aplicação de respostas em redes ad hoc, é importante que toda solução de detecção de intrusão exemplifique os maus comportamentos que serão tratados pelo SDI.

É importante citar o trabalho pioneiro na área de detecção de intrusão em redes ad hoc, apresentado pelos autores Zhang e Lee [41]. Este trabalho apresenta um sistema de detecção de intrusão cooperativo e distribuído, que utiliza a cooperação de cada nó da rede para realizar as funções de detecção e resposta aos nós maliciosos. O sistema utiliza agentes que são executados em cada nó da rede. Estes agentes coletam as informações referentes às detecções realizadas localmente. Assim, ao ser relatada alguma anomalia na rede, os mecanismos cooperativos de detecção e de resposta são iniciados.

Seguindo o modelo apresentado na Figura 3.3, cada nó realiza de forma independente os seus procedimentos de detecção de intrusão, através da utilização de agentes do SDI. Estes agentes monitoram as atividades locais, tais como as atividades do sistema, do usuário, e de comunicação da rede. Além do monitoramento, os agentes também realizam a detecção de intrusão com base nos registros de atividades locais, e iniciam os procedimentos de respostas aos nós detectados. Caso a detecção de intrusão, realizada localmente, não seja tão evidente, então o SDI inicia o procedimento de detecção global, através da

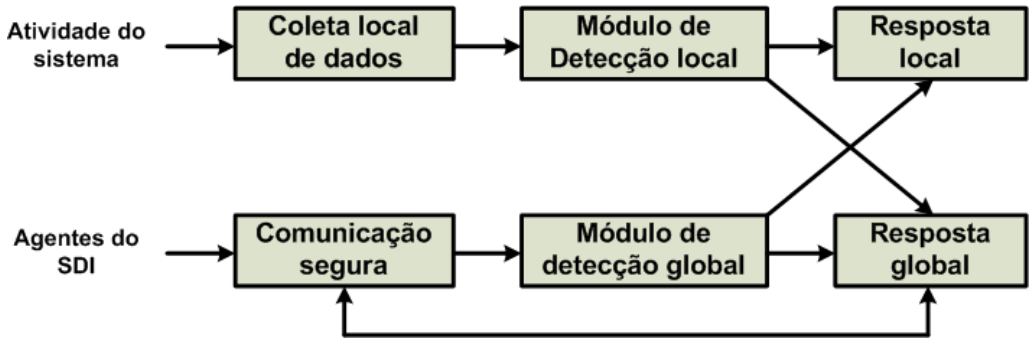


Figura 3.3: Um modelo de detecção de intrusão em redes ad hoc.

colaboração dos agentes dos nós vizinhos.

O módulo de coleta de dados é responsável por reunir os registros das atividades locais e os registros das atividades executadas pelo módulo de detecção local para detectar um comportamento malicioso. A partir destes registros, os procedimentos de detecção global podem ser realizados. O próximo passo após a detecção é a resposta aplicada às atividades detectadas. Portanto, se uma detecção for realizada, o sistema de detecção de intrusão inicia o procedimento de resposta, que pode ser executado de forma local ou global. A resposta local é feita através de mensagens de alerta enviadas para o usuário local. Já a resposta global é feita através de uma ação cooperativa entre os agentes do SDI para eleger a melhor resposta a ser usada. Por fim, o módulo de comunicação segura oferece um nível de confidencialidade e integridade nas mensagens trocadas pelos agentes do SDI.

3.4 Trabalhos Relacionados

O trabalho de Zhang e Lee [41] trouxe importantes contribuições para a área de segurança em redes ad hoc, principalmente por apresentar um modelo distribuído e cooperativo de detecção e de resposta à intrusão, que utiliza técnicas de detecção baseadas no modelo de anomalias, além de respostas baseadas em graus de incerteza. Portanto, após este trabalho, diversos pesquisadores apresentaram soluções com o objetivo de criar um modelo eficiente para a detecção e a resposta aos comportamentos maliciosos em redes ad hoc. Alguns destes trabalhos são apresentados a seguir.

3.4.1 Watchdog e Pathrater

Marti *et al* [5] foram os primeiros a implementar uma solução de detecção e resposta à intrusão em redes ad hoc, testando a eficiência desta solução em ambientes simulados de redes ad hoc. O sistema de detecção é chamado de Watchdog, e é um dos sistemas mais conhecidos na detecção de nós maliciosos em redes ad hoc. Este sistema se baseia no monitoramento de eventos realizados pelos nós vizinhos. O Watchdog foi implementado usando o protocolo de roteamento *Dynamic Source Routing* (DSR) [9] e tem como principal objetivo observar se o pacote enviado para um nó intermediário da rota é encaminhado corretamente para o próximo salto. Para o monitoramento funcionar de forma correta, o Watchdog necessita que seja habilitado o modo promíscuo na camada de rede, ou seja, o Watchdog precisa observar todos os pacotes que trafegam em seu raio de alcance.

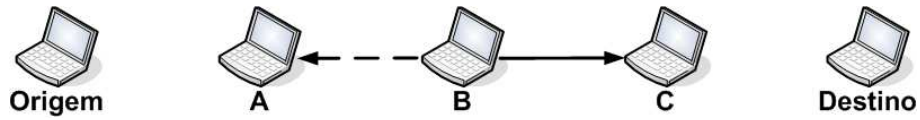


Figura 3.4: Funcionamento do Watchdog.

A partir do exemplo da Figura 3.4, o funcionamento do Watchdog pode ser apresentado. É assumido que existe um caminho entre os nós Origem e Destino, passando pelos nós intermediários A, B e C. O nó de origem, então, envia o pacote para o nó A, que ao recebê-lo, repassa-o para o nó B. Após encaminhar o pacote, o nó A inicia o monitoramento do nó B. Este monitoramento tem como objetivo observar se o nó B encaminha corretamente o pacote para o próximo salto da rede, ou seja, para o nó C. Caso o pacote não seja encaminhado, o nó A classifica esse comportamento do nó B como anormal.

O Watchdog utiliza um *buffer*, que armazena os pacotes por um determinado período de tempo. Ao receber cada pacote observado pelo modo promíscuo da camada de rede, o Watchdog o compara com os pacotes contidos neste *buffer*. Caso o pacote seja igual a algum pacote no buffer, então, ele é removido do *buffer*. Caso o pacote permaneça no *buffer* até expirar o tempo, um contador de eventos anormais detectados relativo ao nó que não o encaminhou é incrementado. Quando esse contador atinge o valor máximo, definido pelo limiar do Watchdog, o nó detectado é classificado como malicioso. A partir desse ponto é iniciado o procedimento de resposta, através do mecanismo chamado de

Pathrater.

O Pathrater tem como objetivo evitar que os nós maliciosos sejam utilizados nas rotas da rede. Para isso, este mecanismo utiliza uma métrica que é definida a partir das detecções locais realizadas por cada nó da rede. Para calcular a métrica, todos os nós utilizam uma variável associada para cada vizinho, que inicia com um valor de 0.5 para cada nó. Portanto, para representar a detecção de um nó malicioso, é assumido que esta variável decrementa em 0.1 no caso do nó ser detectado como malicioso, e incrementa em 0.01 a cada intervalo de 200ms sem o nó ser detectado. O valor máximo que esta variável pode alcançar é 0.8. Ao usar esta variável como métrica de roteamento para a seleção de rotas na rede, é possível a formação de rotas somente com nós cooperativos, evitando os nós maliciosos.

A desvantagem do Watchdog está relacionada com a quantidade de falso-positivos gerada devido aos problemas das redes ad hoc, citados na Seção 3.3. Estes falso-positivos são prejudiciais para o Pathrater, pois os nós maliciosos deixam de ser evitados nas rotas, causando uma degradação no desempenho da rede. Como os nós maliciosos não são bloqueados da rede, eles podem ser usados em rotas futuras e continuar descartando os pacotes, reduzindo a taxa de entrega da rede. Além destes problemas citados, os autores citam outras fraquezas que podem prejudicar o sistema de detecção do Watchdog, tais como a colisão no receptor, o conluio, a potência de transmissão limitada e o descarte seletivo de pacotes.

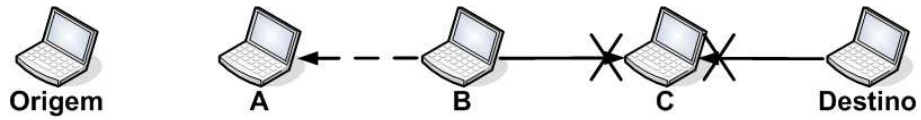


Figura 3.5: Colisão no receptor.

As colisões no receptor podem ocorrer de acordo com o exemplo apresentado na Figura 3.5. Portanto, quando A percebe que o nó B encaminhou o pacote, ele classificará o comportamento como normal. Entretanto, o pacote não é recebido por C, pois uma colisão ocorreu no receptor da mensagem. Desta forma, pode-se dizer que as colisões no receptor podem não assegurar que o pacote realmente foi recebido pelo nó C. Além disso, caso o nó B seja um nó malicioso, ele não precisará encaminhar a mensagem novamente,

pois o nó A já classificou o comportamento dele como normal.

O conluio é um ataque classificado como um ataque forte, pois é difícil de ser evitado e detectado. De acordo com a Figura 3.6, o nó malicioso pode ser detectado e ter bloqueada a sua comunicação com os outros nós da rede. Entretanto, mesmo que sejam aplicadas punições fortes para o nó malicioso, ele poderá continuar utilizando a rede através do nó que está em conluio com ele, ou seja, através do nó comparsa. Este ataque é difícil de ser detectado porque o nó comparsa realiza corretamente todas as funções cooperativas da rede ad hoc, possibilitando que o nó malicioso continue atrapalhando o funcionamento da rede.

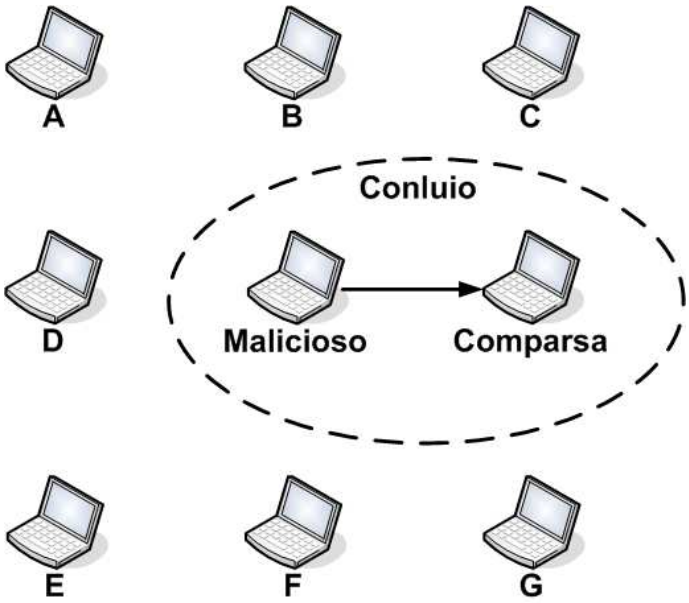


Figura 3.6: Conluio de nós.

O problema de limitação na potência de transmissão de um nó pode fazer com que um encaminhamento não seja percebido pelo nó que está monitorando. Em outras palavras, um nó pode enviar o pacote para o próximo salto usando uma determinada potência de transmissão e o próximo salto encaminhar esse pacote em uma potência diferente. Como as potências de transmissão utilizadas pelos nós são diferentes, o nó que está monitorando poderá não escutar o encaminhamento realizado e classificar o próximo salto como um nó malicioso.

O descarte seletivo dificulta o funcionamento dos mecanismos de detecção de intrusão, por executar os descartes de forma dinâmica. Para um ataque de descarte seletivo ser

executado, o nó atacante define o procedimento de escolha dos pacotes que serão descartados, podendo seguir alguma lógica ou simplesmente sorteando. Quanto mais aleatório for este sorteio, mais difícil será a detecção. Por outro lado, este ataque não prejudica a rede como o ataque de descarte total de pacotes, pois alguns pacotes ainda são encaminhados pelos nós maliciosos que realizam o descarte seletivo.

De acordo com Anantvalee e Wu, o Watchdog e o Pathrater são eficazes na escolha de rotas que evitam os nós maliciosos nas redes ad hoc. Entretanto, esses mecanismos permitem que os nós egoístas continuem encaminhando seus pacotes sem receber nenhum tipo de punição [2]. Desta forma, os nós egoístas são detectados, mas não são excluídos, podendo afetar negativamente o desempenho da rede.

3.4.2 CONFIDANT

Com base no modelo de detecção do Watchdog, Buchegger e Boudec criaram o *Cooperation Of Nodes, Fairness In Dynamic Ad-hoc NeTworks* (CONFIDANT) [42]. Este mecanismo continua seguindo a idéia de que cada nó da rede monitora seus nós vizinhos e utiliza o protocolo DSR. Objetivando a criação de um ambiente de detecção e resposta à intrusão, os autores dividiram o sistema em quatro componentes: o monitor; o gerenciador de confiança; o sistema de reputação; e o gerenciador de caminhos. A Figura 3.7 apresenta estas quatro componentes do CONFIDANT. Desta forma, ao usar as informações de reputação de todos os nós pertencentes às rotas encontradas, o nó é capaz de determinar a rota mais segura.

O Monitor (*Neighborhood Watch*)

Este componente trabalha da mesma forma do Watchdog na detecção de nós egoístas, identificando os nós que se desviam do padrão de comportamentos normais da rede. O monitoramento é realizado somente do nó atual para o próximo nó da rota. O ataque escolhido para ser detectado pelo *Neighborhood Watch* foi o não encaminhamento de pacotes, executados por nós egoístas. Para simular um ambiente de redes ad hoc, os autores utilizaram uma ferramenta conhecida por GloMoSim [43]. Após ser detectado o desvio do comportamento normal, o mecanismo de reputação é iniciado.

O Gerenciador de Confiança

Com base na idéia de descentralização do gerenciamento de confiança, apresentado em [44], os autores definiram o modelo de funcionamento deste componente. O gerenciador de confiança trabalha com entradas e saídas de mensagens de alarme. As mensagens de alarme são enviadas para prevenir os nós cooperativos dos nós maliciosos. Os alarmes de saída são gerados pelos nós que observaram ou receberam anúncios de comportamentos maliciosos. Cada nó que recebe essas mensagens de ALARME é chamado de nó amigo, e seu endereço é administrado em uma lista de nós amigos. Este componente de confiança é considerado o maior desafio para os autores, pois é necessário definir uma forma dinâmica de eleição de amigos, exigindo um modelo de confiança nas informações passadas pelos nós.

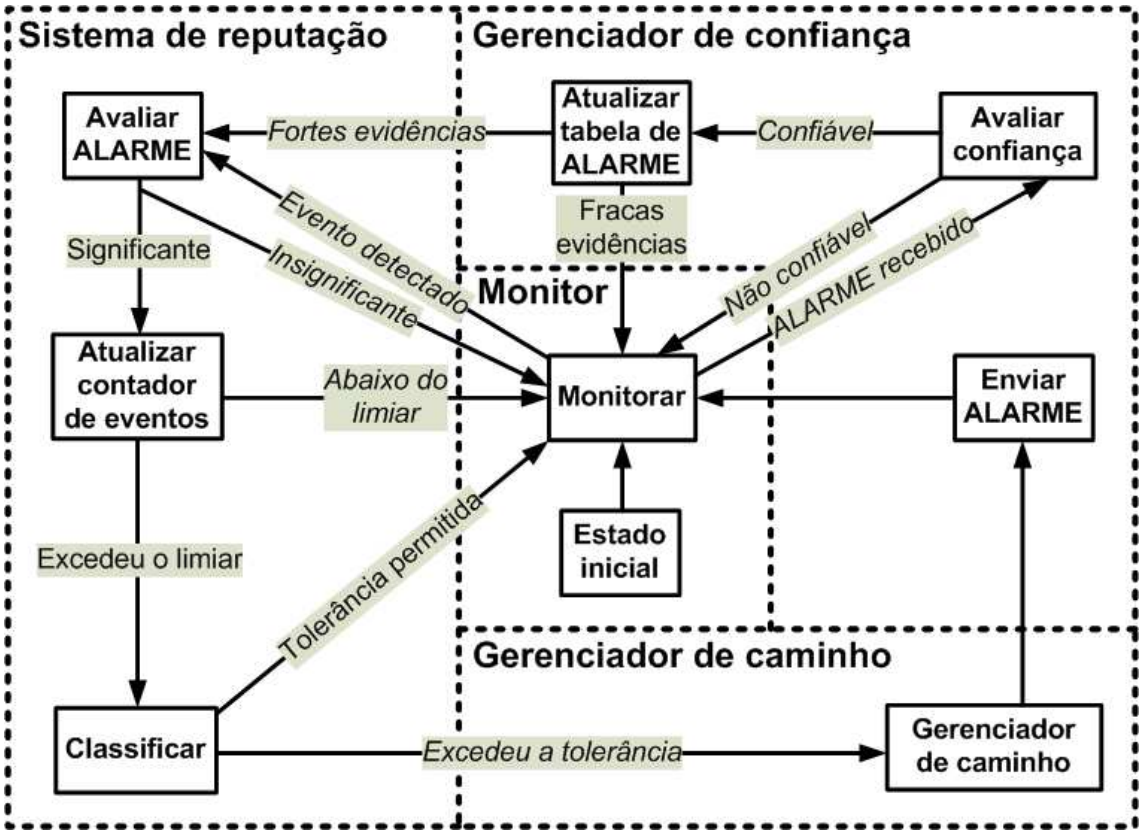


Figura 3.7: Componentes do CONFIDANT.

Os ALARMES de entrada são os anúncios de detecções de maus comportamentos gerados pelos nós amigos. Ao receber um ALARME, cada nó verifica a confiança que ele tem no nó que gerou aquele anúncio. Portanto, antes de iniciar qualquer reação, o nó

primeiramente filtra o ALARME de entrada de acordo com o nível de confiança do nó que o reportou. O mecanismo de confiança é similar à solução usada em um sistema de validação e certificação para redes ad hoc, conhecido como *Pretty Good Privacy* (PGP) [45]. O princípio de confiança usado pelo PGP baseia-se na relação entre uma chave pública criptografada e um nome de usuário. O gerenciador usa a mesma idéia para considerar um anúncio como sendo confiável.

Para executar as suas funcionalidades, o gerenciador de confiança utiliza três módulos: a tabela de alarme, que armazena as informações sobre os ALARMES recebidos; a tabela de gerenciamento de confiança, que determina o nível de confiança dos nós, para classificar a confiança do ALARME recebido; e, por fim, a lista de amigos, que contém os registros de todos os nós que enviam freqüentemente os ALARMES.

O Sistema de Reputação

Os autores tiveram a idéia de usar um sistema de reputação com base nos ambientes de leilão online. Nestes ambientes, o sistema de reputação é utilizado para determinar uma confiança nas informações dos usuários, através da observação dos comentários inseridos por outros usuários sobre os produtos vendidos ou comprados por eles. Em [46], Resnick *et al* apresentam com mais detalhes os procedimentos seguidos por estes sistemas de reputação.

Para evitar uma centralização de informações sobre os nós maliciosos em um nó específico, cada nó mantém a sua lista negra, que contém as identificações dos nós que foram classificados como suspeitos. Através destas listas, um nó pode enviar um alerta sobre os nós que devem ser evitados nas mensagens de requisição de rotas. Desta forma, os nós podem verificar em sua lista negra, os nós que devem ser evitados, antes de encaminhar o pacote. Entretanto, os autores citam o problema de distinção entre um nó que é suspeito de executar um mau comportamento e um nó que realmente é malicioso, ou seja, o falso-positivo também é um problema citado e tratado através deste sistema de reputação. Outro problema comentado pelos autores é o estouro das listas negras, que pode ser resolvido através de temporizadores para cada identificador registrado.

O sistema de reputação só considera um comportamento malicioso, quando um alerta

de mau comportamento ultrapassar um limiar de regras toleradas. Além disso, as classificações são feitas de forma que os nós consideram um peso maior nas suas detecções do que nas detecções passadas por terceiros. Quando a reputação é determinada e ultrapassa um valor de tolerância definido, o sistema de reputação faz uma requisição ao gerenciador de caminhos.

O Gerenciador de Caminhos

Este componente do CONFIDANT realiza quatro funcionalides: refazer os caminhos existentes de acordo com a métrica de segurança, como por exemplo, a partir da reputação dos nós da rota; remover os nós maliciosos dos caminhos conhecidos; tratar as rotas que necessitam de um nó classificado como malicioso, podendo até ignorá-las; assumir a responsabilidade de agir ao recebimento de mensagens de requisição de rotas recebidas, que contêm os identificadores dos nós suspeitos de comportamentos maliciosos.

Analisando os componentes e as funcionalidades do CONFIDANT, é possível apresentar uma desvantagem relacionada aos falso-positivos nas detecções, pois os autores citam os falso-positivos como um problema e apresentam um sistema de reputação para resolvê-lo. No entanto, o sistema de reputação apresenta uma falha grave, que está relacionada ao controle que um nó malicioso pode ter nas decisões referentes às punições realizadas na rede, já que as punições são realizadas com base nas informações de detecção realizadas por terceiros. Desta forma, um nó malicioso pode enviar falsas acusações na rede e provocar punições aos nós cooperativos. Neste contexto, existe a necessidade de implementação de um mecanismo dinâmico de confiança para analisar a reputação dos nós que enviam alertas de detecção. Por exemplo, um nó malicioso é capaz de ganhar uma boa reputação encaminhando pacotes corretamente por um determinado tempo. Após conseguir essa boa reputação, o nó pode iniciar uma inundação de falsas acusações na rede, incentivando a ocorrência de falso-positivos. Para resolver este problema de confiança, estes mesmos autores apresentam soluções usando estatísticas Bayesianas [47] [48].

3.4.3 CORE

Michiardi e Molva [49] também apresentam uma técnica de prevenção contra nós que realizam ataques de egoísmo, chamada de *Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad hoc Networks* (CORE). Como citado na Seção 2.2.1, o ataque de egoísmo é realizado por um nó que resolve não cooperar nas funcionalidades colaborativas da rede ad hoc, seja para atacar a rede ou, simplesmente, para economizar os seus recursos. O principal objetivo da técnica é forçar a cooperação do nó, utilizando funcionalidades baseadas em um sistema de monitoramento e em um sistema de reputação. O sistema de reputação utiliza métodos, chamados de diretos e indiretos, que são detalhados a seguir.

De acordo com os autores, os comportamentos egoístas devem ser tratados com cuidado, pois em alguns trabalhos [50] [51], este comportamento pode não ser considerado como malicioso, já que o nó não tem a intenção de prejudicar a rede. Desta forma, os autores utilizam um valor neutro de reputação inicial, ou seja, igual a zero. Além disso, as funcionalidades da rede são divididas em diferentes categorias, tais como a descoberta de rotas ou o encaminhamento de pacotes do protocolo de roteamento DSR. Cada uma destas funcionalidades exercem diferentes níveis de influência no desempenho da rede. Por exemplo, o encaminhamento de pacotes exerce uma maior influência no desempenho da rede do que a descoberta de rotas. Portanto, esta influência da funcionalidade é usada como um peso para calcular a reputação de um nó.

O sistema de reputação do CORE é semelhante ao sistema do CONFIDANT, no qual cada nó pode enviar e receber alertas sobre outros nós da rede. Entretanto, a principal diferença do CORE está na forma de tratamento das informações recebidas por observações realizadas por terceiros. Neste contexto, o CORE recebe dos outros nós somente os alertas que se referem aos comportamentos normais observados, enquanto que os comportamentos anormais são detectados somente através do monitoramento local. Em outras palavras, o CORE tenta se prevenir das falsas acusações enviadas por nós maliciosos, que podem ser consideradas o ponto fraco do CONFIDANT.

Os autores dividem o CORE em dois componentes: o primeiro utiliza o Watchdog

para observar o comportamento dos nós; e o segundo classifica e verifica a reputação de cada nó. O Watchdog trabalha da mesma forma como foi apresentado na Seção 3.4.1. Já o sistema de reputação utiliza diversas tabelas, uma para cada funcionalidade da rede e outra para os valores de reputação de cada nó.

Considerando os problemas observados na proposta de Marti *et al*, o CORE também considera a existência de falso-positivos nas detecções realizadas pelo Watchdog. Para tentar evitar as punições de nós devido às detecções erradas, os autores apresentam uma equação que avalia os maus comportamentos que são detectados esporadicamente na rede.

O principal problema desta proposta é a ausência de resultados no artigo. Torna-se difícil avaliar o desempenho desta proposta somente através da estruturação do modelo apresentado. Portanto, ao considerar o objetivo principal desta proposta, ou seja, o tratamento de eventos normais, é possível concluir que os resultados não serão tão diferentes dos resultados apresentados pelo CONFIDANT. Isto foi concluído pela seguinte razão: ao tratar os anúncios de comportamentos normais ou anormais na rede, enviados por outros nós, pode-se entender que os nós maliciosos ainda poderão influenciar na detecção de nós egoístas, pois eles podem aumentar a reputação de outros nós maliciosos. Entretanto, ao considerar a detecção local de comportamentos maliciosos, pode-se dizer que este é o principal fator positivo que poderão diferenciar os resultados do CORE com os resultados do CONFIDANT.

3.4.4 OCEAN

O *Observation-based Cooperation Enforcement in Ad hoc Networks* (OCEAN) [52] também foi proposto como uma extensão do protocolo DSR. Com base nas outras propostas apresentadas, o OCEAN também utiliza um sistema de monitoramento e um sistema de reputação para detectar e punir os nós egoístas e mal intencionados em redes ad hoc. Entretanto, este sistema confia somente nas informações de detecção coletadas localmente, com o objetivo de evitar o problema de falsos alertas enviados na rede. Portanto, o OCEAN retorna na idéia de realizar a detecção somente através de monitoramentos locais de eventos.

Para definir os comportamentos não cooperativos nas redes ad hoc, estes autores usam uma classificação bastante interessante. Se o nó encaminha as mensagens de roteamento e descarta as mensagens de dados, estes são considerados mal intencionados. Por outro lado, se os nós não encaminham nenhum tipo de mensagens, estes são considerados egoístas. O OCEAN é visto como um sistema de incentivo em redes ad hoc, que força o nó a cooperar para poder utilizar a rede. Inicialmente, todos os nós começam com a sua reputação igual a zero, e esta reputação pode ser incrementada ou decrementada de acordo com o comportamento destes nós na rede.

O funcionamento do OCEAN é realizado através de um sistema de créditos. Todos os nós da rede utilizam uma variável que determina o crédito de cada vizinho. Assim, cada nó observa o envio de pacotes de todos os seus nós vizinhos, independente se estes estão se comunicando diretamente ou não. Ao observar os encaminhamentos de pacotes, o sistema de reputação utiliza um sistema de crédito que incrementa a variável ao observar cada encaminhamento de pacotes e decrementa esta variável ao observar o não encaminhamento de pacotes. Se crédito do vizinho atingir um valor menor do que o limite mínimo permitido, o nó monitor descartará qualquer pacote gerado pelo nó com baixo crédito.

Um problema que pode ser citado neste mecanismo é a não apresentação da forma que o OCEAN trata os bloqueios em série, sem ter que utilizar uma mensagem informando que o nó terceiro está bloqueado. Os bloqueios em série ocorrem quando um nó bloqueia outro nó que não encaminhou uma mensagem para o próximo salto, por saber que esse próximo salto é um nó malicioso. Além disto, os autores desconsideram a existência dos falso-positivos na rede, citando apenas o problema de não conseguir detectar as colisões no receptor [5].

3.5 Comparação entre os Sistemas de Detecção de Intrusão em Redes Ad hoc

Ao analisar os trabalhos apresentados, que objetivam a criação de um sistema de detecção e resposta à intrusão em redes ad hoc, observa-se algumas características importantes em suas implementações. Estas características são apresentadas de forma comparativa na Tabela 3.1.

Tabela 3.1: Comparação entre os Sistemas de Detecção de Intrusão em Redes Ad hoc.

Sistema de detecção	**Características**	**Metodologia**	**Falso-positivos**
Watchdog e Pathrater.	Sistema de detecção baseado em um limiar que determina a quantidade de maus comportamentos tolerados, até que o nó seja punido. Punição realizada através de uma métrica que exerce influência na formação de rotas do protocolo de roteamento, tentando evitar nós classificados como maliciosos.	Detecção baseada em anomalias e realizada localmente.	O mecanismo tolera uma quantidade determinada de maus comportamentos detectados para aplicar uma resposta
CONFIDANT.	Sistema de detecção que exige a implementação de um mecanismo de confiança, pois considera as informações de detecções realizadas por terceiros. O sistema de punições baseia-se em níveis de evidências para formar caminhos mais seguros na rede, evitando nós maliciosos.	Detecção baseada em anomalias e realizada de forma distribuída.	Utiliza um sistema de reputação para reduzir a quantidade de falso-positivos.
CORE.	Sistema de detecção semelhante ao CONFIDANT, porém, considera somente os anúncios de bons comportamentos, enviados por terceiros. Punição realizada através do bloqueio de comunicação entre o nó monitor e o nó detectado.	Detecção baseada em anomalias e realizada de forma distribuída.	Usa um modelo que aplica punições de acordo com a freqüência de detecções de maus comportamentos em uma janela de tempo.
OCEAN.	Sistema de detecção que utiliza o Watchdog. Mecanismo de reputação que incentiva a cooperação do nó nas funcionalidades da rede. Aplica a punição de acordo com o tipo de comportamento não cooperativo detectado.	Detecção baseada em anomalias e realizada localmente.	Executa punições de acordo com o mau comportamento detectado.

Seguindo as características apresentadas nessa tabela, é possível observar que todos os mecanismos utilizam o modelo baseado em anomalias, justificando que o modelo baseado em assinaturas necessita de um uso maior de memória para armazenar todas as assinaturas

de padrões de ataques e de variações destes ataques.

Ainda ao analisar a tabela, nota-se que existe uma equivalência nas formas que as detecções são realizadas, duas baseadas em sistemas de detecções locais e duas baseadas em sistemas de detecções distribuídas. Portanto, as principais diferenças destes sistemas apresentados estão no modo que as punições são aplicadas, alguns considerando somente os maus comportamentos detectados e outros considerando ambos os comportamentos, sejam estes normais ou maliciosos.

Todas as propostas citam o problema do falso-positivo na rede e apresentam soluções que tentam reduzir a influência deste problema no mecanismo de punição. Com base nesta necessidade de tornar o falso-positivo menos prejudicial ao sistema de detecção de intrusão, este trabalho apresenta uma solução que tem como objetivo aumentar a precisão na detecção de comportamentos egoístas nas redes ad hoc. A proposta apresentada é formada pelo Sistema de Detecção de nós Egoístas (SDE), que detecta os nós egoístas na rede ad hoc, e pelo Mecanismo de Avaliação e Punição de nós egoístas em redes Ad hoc (MAPA), que aplica as punições aos nós detectados.

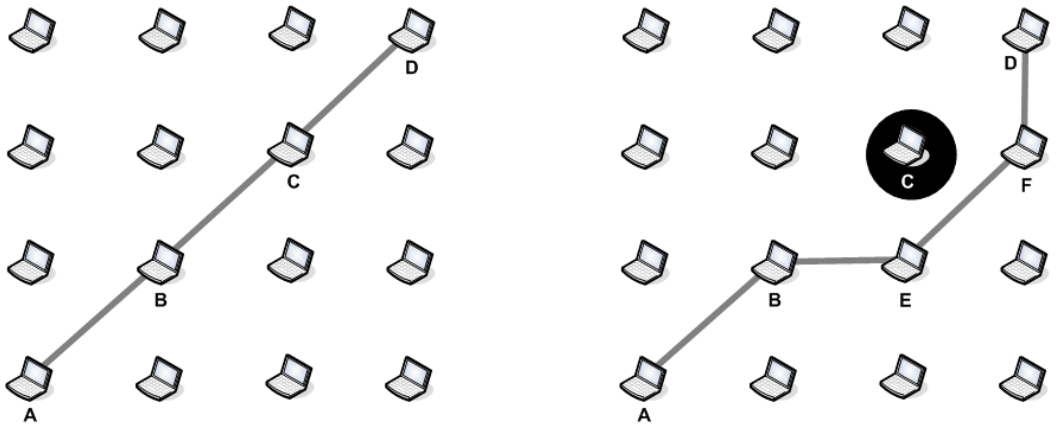
Capítulo 4

O Mecanismo Proposto

ESTE trabalho apresenta a proposta de um mecanismo de avaliação e punição que calcula a probabilidade de um nó detectado ser realmente um nó egoísta. Com base nessa probabilidade, o Mecanismo de Avaliação e Punição de nós egoístas em redes Ad hoc (MAPA) determina se o nó será bloqueado temporariamente ou definitivamente. Enquanto o nó estiver bloqueado, toda rota armazenada ou recém descoberta contendo o endereço do nó egoísta é descartada.

Ao executar uma punição, o sistema executa um novo procedimento de descoberta de rota, que é iniciado a partir do nó que detectou o nó egoísta. Após ser descoberta uma nova rota, o SDI solicita o envio de uma mensagem de erro de rota para o nó fonte, relatando o problema ocorrido. Estas mensagens de erro de rota são utilizadas pelo protocolo de roteamento *Dynamic Source Routing* (DSR) no procedimento de manutenção de rotas [9], descrito na Seção 2.1.2. Para evitar que os nós egoístas continuem utilizando a rede como no Pathrater, os pacotes criados pelos nós egoístas não são encaminhados pelos nós vizinhos que o bloquearam.

A Figura 4.1(a) apresenta os procedimentos básicos realizados pelo SDE e pelo MAPA em uma rede ad hoc, após um nó malicioso ser detectado em uma rota utilizada. Portanto, de acordo com a figura, assume-se a existência de uma rota conhecida, ligando o nó fonte A ao nó de destino D, passando pelos nós intermediários B e C. Ao receber os pacotes do nó A, o nó B os encaminha para o nó C, que é o próximo salto na rota. Em certo



(a) Antes da detecção e da punição. (b) Após a detecção e a punição.

Figura 4.1: Exemplo do funcionamento do SDE e do MAPA em uma rede ad hoc.

instante, o nó B percebe que o nó C não está encaminhando os pacotes e, então, inicia o procedimento de avaliação. Após a avaliação, o nó C é bloqueado pelo nó B. Neste momento, o nó B pode utilizar uma segunda rota armazenada em *cache* ou iniciar um procedimento de descoberta de rota a partir dele mesmo. Ao encontrar a nova rota, o nó B envia uma mensagem de erro de rota para o nó fonte, inserindo a nova rota descoberta nesta mensagem, evitando o nó C. Como ilustrado na Figura 4.1(b), o nó A passará a utilizar a nova rota para enviar os pacotes para D. Este procedimento de descoberta de rota a partir de um nó intermediário é uma otimização utilizada pelo DSR para evitar que os pacotes presentes em um nó intermediário sejam descartados durante o procedimento de manutenção de rotas.

Os mecanismos de detecção, avaliação e punição foram implementados em um simulador. Esta implementação foi realizada em uma camada criada acima da camada de rede, ver Figura 4.2. As detecções são realizadas pelo módulo de monitoramento do Sistema de Detecção de nós Egoístas (SDE), que se baseia no modelo de detecção do Watchdog [5]. Além deste módulo, o SDE utiliza o módulo de recuperação, que executa os procedimentos de registro de nós egoístas e inicia a recuperação de rotas comprometidas. As avaliações e as punições são executadas pelo MAPA e ocorrem após uma ou mais detecções de eventos egoístas serem detectados pelo SDE.

4.1 Sistema de Detecção de nós Egoístas (SDE)

O SDE é responsável por observar os eventos de envio, recebimento e encaminhamento realizados pelo protocolo de roteamento *Dynamic Source Routing* (DSR). Para realizar a detecção, o SDE utiliza o módulo de monitoramento, que analisa estes eventos observados em um modelo de detecção baseado em anomalias. O modelo baseado em anomalias foi escolhido por reduzir a possibilidade de um nó malicioso realizar um ataque e não ser detectado, pois ele é obrigado a executar um evento normal para obter sucesso. Quando a punição é aplicada ao nó detectado como egoísta, o SDE também é responsável por iniciar o procedimento de recuperação da rota, através do módulo de recuperação, ilustrado na Figura 4.2. Para entender o funcionamento dos módulos contidos em cada mecanismo, são descritas a seguir as principais características de cada um.

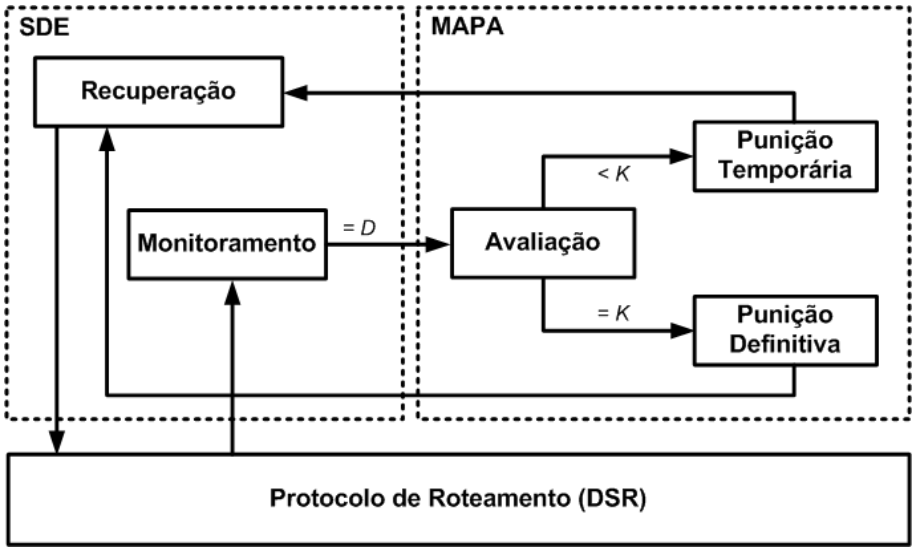


Figura 4.2: Arquitetura do SDE e do MAPA.

4.1.1 Módulo de Monitoramento

O SDE utiliza o módulo de monitoramento para realizar a detecção de nós egoístas em rede ad hoc. Para isso, este módulo armazena um identificador para cada pacote enviado ao próximo salto da rota. Este identificador possui um temporizador que é removido somente se o pacote encaminhado pelo nó monitorado for recebido pelo nó que está monitorando. O temporizador do SDE atribui o mesmo valor do temporizador utilizado pelo

DSR para aguardar a resposta de uma descoberta de rota. Se o temporizador do identificador expirar, a detecção de evento egoísta é realizada e o identificador é removido. Assim, o SDE é capaz de monitorar o encaminhamento de pacotes executado pelo próximo salto na rota, quando este não for o destinatário do pacote.

Diferente do OCEAN, no SDE, o nó não observa a comunicação realizada entre dois vizinhos adjacentes, somente quando ele faz parte da rota. Após a detecção e o bloqueio serem executados, caso não seja encontrada uma rota secundária que evite o nó bloqueado, o nó destino é classificado como inalcançável. Entretanto, o nó fonte pode optar pelo envio de pacotes por rotas com a presença de nós egoístas, pois existe a possibilidade do nó egoísta estar executando um descarte seletivo. Para isso, o nó fonte teria que utilizar um marcador para indicar o encaminhamento forçado através de nós bloqueados.

Como citado anteriormente, os falso-positivos nas punições ocorrem devido às detecções incorretas realizadas pelo SDE. Estas detecções incorretas são provocadas por problemas temporários que ocorrem nas redes ad hoc. Para tentar reduzir a influência que as detecções incorretas exercem nas punições aplicadas aos nós da rede, o SDE utiliza um limiar de tolerância D. Este limiar determina o número de vezes que os eventos egoístas devem ser detectados. Quando este limiar é atingido, o total de eventos normais e egoístas contabilizados é passado para o MAPA, que realiza a avaliação. O parâmetro que contabiliza todos os eventos observados até a ocorrência do *D-ésimo* evento é representado por e. Dessa forma, se $e = 100$ e $D = 5$, então se pode afirmar que a quinta detecção ocorreu no centésimo evento monitorado. Logo, sabe-se que foram observados 95 eventos normais e 5 eventos egoístas.

A Figura 4.3 apresenta com detalhes os mecanismos presentes no módulo de monitoramento. Primeiramente, o mecanismo de análise registra todos os pacotes que devem ser encaminhados pelo próximo salto da rota. Este mecanismo é responsável por inserir o temporizador em cada pacote registrado. Dessa forma, ao expirar o temporizador, o mecanismo de análise contabiliza o registro como um evento anormal observado. Por outro lado, se o registro for removido antes do temporizador expirar, o mecanismo de análise contabiliza o registro como evento normal observado. Após serem observados D eventos anormais, o mecanismo de análise passa os valores desses contadores ao módulo de

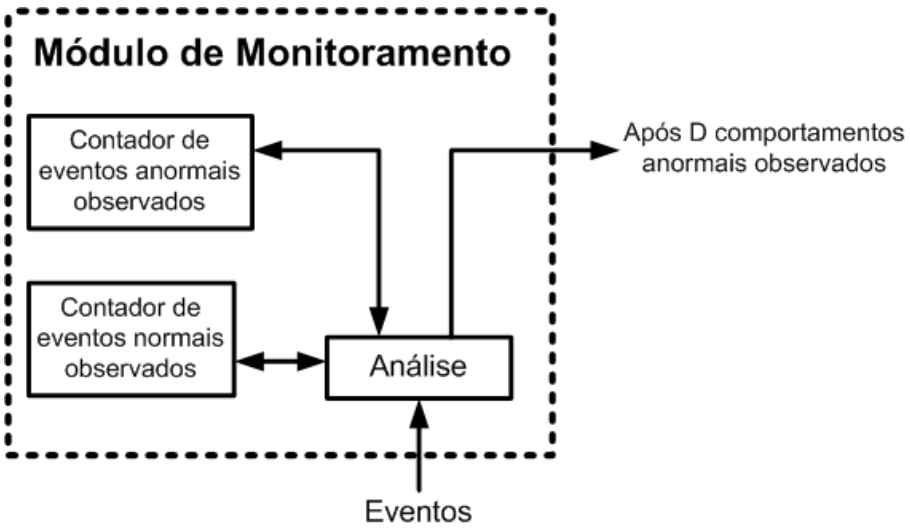


Figura 4.3: Módulo de monitoramento.

avaliação do MAPA.

4.1.2 Módulo de Recuperação

Após a punição ser aplicada, o nó monitor é responsável por tentar descobrir uma nova rota, verificando se existe uma rota secundária armazenada em cache ou iniciando um novo procedimento de descoberta de rotas. Portanto, ao descobrir a nova rota, o módulo de recuperação é responsável por requisitar ao DSR o envio de uma mensagem de erro de rota para o nó de origem, utilizando uma marcação que identifica o motivo do erro de rota. Esta marcação é inserida em um campo disponível no cabeçalho da mensagem de erro de rota.

Apesar de usarem a mesma forma de detecção, o SDE e o Watchdog diferem em alguns aspectos. O SDE monitora e considera todos os eventos de encaminhamento do nó monitorado, pois o MAPA utiliza as informações sobre os pacotes que foram encaminhados ou não. Por outro lado, o Watchdog não considera os pacotes corretamente encaminhados, pois ele necessita somente das informações sobre os eventos de não encaminhamento. Além disso, após o nó ser bloqueado, o SDE é responsável por requisitar uma nova rota para o protocolo de roteamento e por solicitar a inserção da nova rota na mensagem de erro que será enviada ao nó fonte.

Uma vez que o SDE trabalha observando as mensagens de controle do roteamento, é possível que um nó atacante utilize essas mensagens para tentar enganar os nós da rede. Este ataque é conhecido na literatura como ataque bizantino, e foi apresentado na Seção 2.2.1. Portanto, o SDE foi implementado para ser capaz de trabalhar em conjunto com as soluções de defesa contra o ataque bizantino, tais como as soluções apresentadas em [21] [22]. Estas soluções são disponibilizadas por mecanismos pró-ativos de segurança, que utilizam mecanismos de autenticação e autoridades certificadoras para aumentarem o grau de confiança nas informações passadas pelos usuários da rede.

4.2 Mecanismo de Avaliação e Punição de nós egoístas em redes Ad hoc (MAPA)

O MAPA é responsável por aplicar com precisão as punições aos nós detectados, com base nas informações coletadas pelo SDE. Para realizar as punições, o MAPA utiliza estas informações e inicia a avaliação no instante em que a *D-ésima* detecção de evento egoísta é observada pelo SDE. A avaliação então retorna um resultado que indica a probabilidade do nó ser egoísta. Esta probabilidade é representada por p e é obtida através de

$$p = \frac{D}{e}. \tag{4.1}$$

A cada avaliação realizada, o MAPA determina se o nó receberá um bloqueio temporário ou um bloqueio definitivo. Nos bloqueios temporários o nó é bloqueado, por um determinado intervalo de tempo, para executar qualquer funcionalidade da rede com o nó que o bloqueou, ou seja, qualquer pacote originado pelo nó bloqueado não será encaminhado. Já nos bloqueios definitivos, este bloqueio é permanente. Para identificar os nós bloqueados, é considerada a existência de um mecanismo seguro de identificação e autenticação dos nós, como os mecanismos apresentados em [53] e [54].

Para um nó receber uma punição máxima, ou seja, ser bloqueado definitivamente da rede, ele deve persistir em realizar eventos egoístas. O parâmetro que determina a quantidade de bloqueios temporários aplicados até o bloqueio definitivo é representado por k. Por exemplo, se $k = 2$, então o nó detectado é temporariamente bloqueado na primeira

avaliação do MAPA e será definitivamente bloqueado quando a segunda avaliação for requisitada pelo SDE.

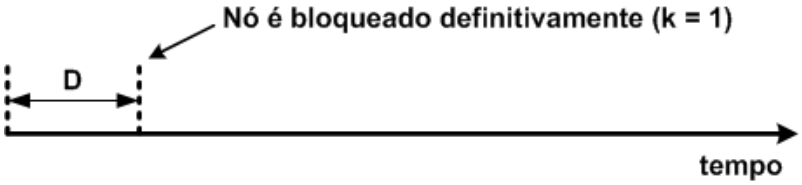
Outro parâmetro utilizado pelo MAPA é o limiar L, que determina o tempo de duração dos bloqueios temporários. O valor de L é atribuído de acordo com a tolerância desejada. Este limiar é comparado ao valor de p (Equação 4.1), calculado a cada avaliação realizada. A partir desta comparação, o tempo que o nó permanecerá bloqueado em cada bloqueio temporário é definido da seguinte forma:

- se $p < L$, o nó é bloqueado durante um período de $t_{b(curto)}$ unidades de tempo;
- se $p \geq L$, o nó é bloqueado durante um período de $t_{b(longo)}$ unidades de tempo.

4.3 Bloqueios Temporários e Bloqueios Definitivos

Para determinar a tolerância do número de eventos detectados até o nó ser definitivamente bloqueado, o SDE e o MAPA utilizam os parâmetros D e k. De acordo com a Figura 4.4(a), se $k = 1$, então o nó é definitivamente bloqueado no instante em que o *D-ésimo* evento egoísta é detectado, sem receber um bloqueio temporário anteriormente. Portanto, o nó detectado como egoísta tende a usar os recursos da rede por um menor intervalo de tempo. Na Figura 4.4(b), quando o valor de $k = 2$, o nó é temporariamente bloqueado uma vez, ou seja, $k-1$ vezes antes de ser bloqueado definitivamente. É importante citar que o intervalo de tempo mostrado através do parâmetro D representa o tempo que o SDE gasta para detectar D eventos anormais.

Além dos parâmetros citados anteriormente, o SDE e o MAPA utilizam o parâmetro I, que determina a quantidade de eventos normais que devem ser executados pelo nó para que o último bloqueio temporário seja desconsiderado. Quando o SDE observa que I eventos normais foram executados após um bloqueio temporário ter sido realizado, ele solicita que o MAPA desconsidere o último incremento realizado no contador de bloqueios temporários. Por exemplo, se $k = 3$ e o nó realizou I eventos normais após ter recebido o segundo bloqueio temporário, então este segundo bloqueio temporário será desconsiderado pelo MAPA. No entanto, se este mesmo nó executar mais I eventos normais, ele não



(a) k = 1.

(b) k = 2.

Figura 4.4: Intervalo de tempo da primeira detecção até o bloqueio definitivo.

terá o primeiro bloqueio temporário desconsiderado, pois o parâmetro I é determinado a partir das informações obtidas na última avaliação.

4.4 Análise Matemática

Nesta seção são descritos os parâmetros utilizados na análise matemática, assim como as conclusões de acordo com os resultados obtidos. A análise matemática foi realizada na ferramenta Maple, versão 11 [55].

Ao entender o funcionamento do SDE e do MAPA, é possível notar que o SDI é capaz de alterar o seu nível de tolerância, apenas ajustando os parâmetros utilizados em cada módulo. Assim, a proposta apresentada pode ser classificada de acordo com a sua forma de detecção, avaliação e punição. A primeira pode ser chamada de Intolerante, pois bloqueia uma estação após a ocorrência de um único evento malicioso detectado, assumindo os valores de $D = 1$ e de $K = 1$.

A segunda forma de execução engloba diferentes níveis de tolerância, ou seja, existe a possibilidade de ajustar os valores de D e/ou de k, assumindo valores maiores do que

um. Esta segunda técnica determina a tolerância através das atribuições em conjunto da quantidade de eventos maliciosos observados e da quantidade de bloqueios temporários que são aplicados. Portanto, se o nó pretende bloquear mais e observar menos, ele deve ajustar os valores de k para serem maiores do que um e de D para serem mais próximos a um. Por outro lado, se a técnica pretende aplicar menos bloqueios e observar mais eventos, o ideal é atribuir valores mais próximos a um para k e valores maiores do que um para D.

Para analisar matematicamente a probabilidade de cada nó detectado ser um nó egoísta é usada uma função de probabilidade de massa, baseada em uma distribuição binomial negativa [56]. A distribuição binomial negativa tem sido bastante utilizada na análise de desempenho de sistemas de segurança tolerantes [57] [58] [59]. Portanto, devido à sua característica de calcular a probabilidade de ocorrerem determinados eventos em um total de eventos aleatórios observados, esta distribuição foi escolhida.

Ao utilizar o parâmetro D na função de probabilidade de massa (PMF) de uma distribuição binomial negativa, é possível representar a equação por

$$p_X(e) = \binom{e-1}{D-1} p_{nó}^{D} (1 - p_{nó})^{e-D}; \tag{4.2}$$

ou seja, $p_{nó}$ representa a probabilidade do nó executar D eventos egoístas em e eventos observados. Nota-se na proposta que o parâmetro k é independente do parâmetro D, portanto, ao ser adicionado o parâmetro k na equação, obtém-se

$$p_X(e) = \binom{e-1}{kD-1} p_{nó}^{kD} (1 - p_{nó})^{e-kD}. \tag{4.3}$$

A distribuição binomial negativa é uma variação da distribuição geométrica, que calcula a probabilidade de um evento ocorrer em um total de eventos observados. Portanto, a distribuição geométrica pode representar o funcionamento do mecanismo Intolerante, pois assume que os valores dos parâmetros D e k são iguais a um. A partir desta observação, é possível reduzir a Equação 4.3 a uma distribuição geométrica, como demonstrado

na Equação 4.4.

$$\begin{aligned} p_X(e) &= \binom{e-1}{kD-1} p_{nó}^{kD} (1-p_{nó})^{e-kD} \\ p_X(e) &= \binom{e-1}{1-1} p_{nó}^{1} (1-p_{nó})^{e-1} \\ p_X(e) &= p_{nó}(1-p_{nó})^{e-1}. \end{aligned} \tag{4.4}$$

Ao considerar a probabilidade do nó executar $k \cdot D$ eventos egoístas, a média do número de eventos realizados até os $k \cdot D$ eventos egoístas ocorrerem pode ser obtida por

$$e = \frac{kD}{p_{nó}}. \tag{4.5}$$

Portanto, a probabilidade de punição calculada a cada D eventos egoístas detectados, dado um total de e eventos observados, é representada pela equação

$$P_p(e) = \sum_{1}^{e} p_X(e). \tag{4.6}$$

Por fim, a probabilidade de bloqueio definitivo ao nó que recebeu $(k-1)$ bloqueios temporários pode ser representada pela seguinte equação:

$$P_b(e) = \sum_{1}^{e} \binom{e-1}{kD-1} p_{nó}^{kD} (1-p_{nó})^{e-kD}. \tag{4.7}$$

A partir da Equação 4.7, é possível analisar a probabilidade de punição de todas as formas descritas anteriormente. Para observar um possível caso de falso-positivo na punição, foi atribuído o valor de 0.05 para $p_{nó}$, pois este valor pode representar a detecção de um comportamento anormal para cada vinte comportamentos normais realizados, ou seja, uma pequena probabilidade do nó ser egoísta. Na Figura 4.5, ao assumir o valor de k igual a 1 é possível analisar a influência do parâmetro D no bloqueio definitivo. Assim, quanto maior for o valor de D, maior será a quantidade de eventos observados, antes da punição ser aplicada.

A Figura 4.6 mostra que se o valor de $D > 1$ e os valores de k forem aumentados, então o total de eventos observados será ainda maior. Com esses resultados é possível analisar a influência dos parâmetros D e k na tolerância do nó em executar os bloqueios

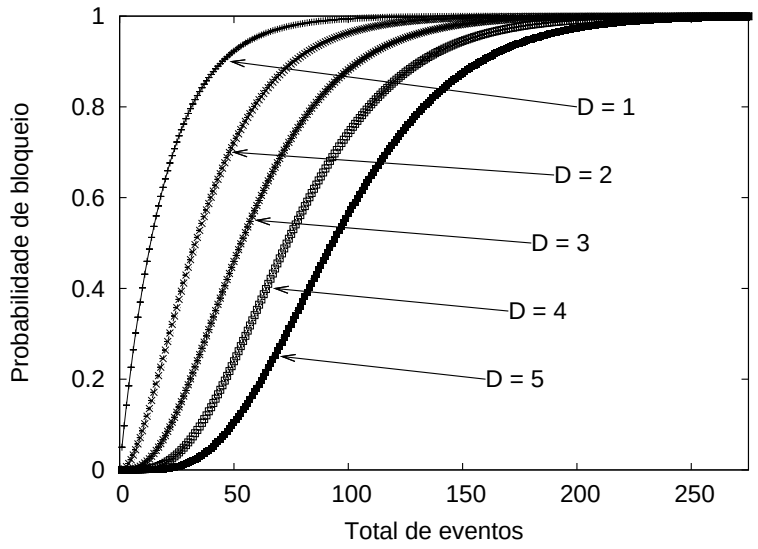


Figura 4.5: Probabilidade de bloqueio definitivo com $k = 1$ e D variando.

definitivos. Entretanto, não pode ser analisada a influência que o tempo exerce na punição dos nós e na eficiência da proposta.

É importante destacar que as detecções, as avaliações e as punições realizadas pelos módulos são sem-memória, ou seja, após o cálculo de uma probabilidade, o número de eventos é zerado para a realização do próximo cálculo. Assim, cada cálculo de uma probabilidade é independente de cálculos anteriores. Valendo também para a atribuição dos valores do parâmetro I, que assume diferentes valores a cada avaliação realizada.

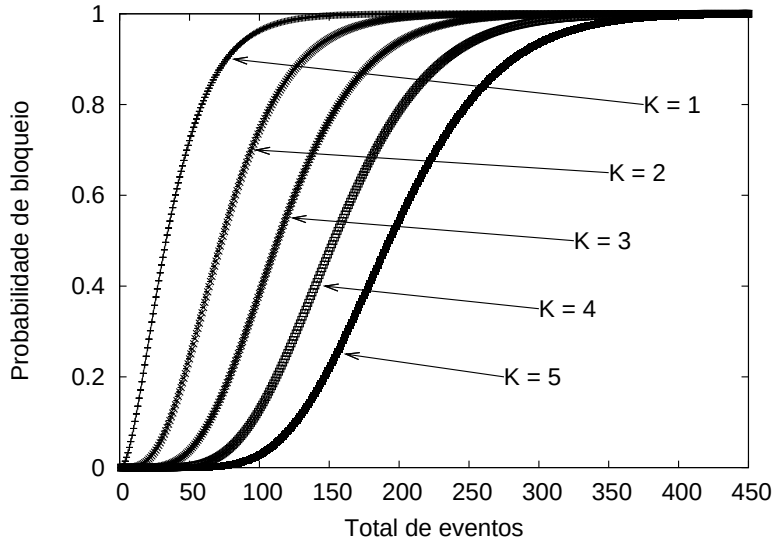


Figura 4.6: Probabilidade de bloqueio definitivo com $D = 2$ e k variando.

Para validar os resultados obtidos nas análises matemática, o SDE e o MAPA foram implementados em um ambiente de simulação. Através da simulação é possível determinar a eficiência destes mecanismos, incluindo todos os parâmetros utilizados pela proposta. O capítulo a seguir descreve todos os detalhes do ambiente simulado, como também os resultados obtidos.

Capítulo 5

Resultados de Simulação

NESTE capítulo são mostrados os parâmetros utilizados e as premissas estabelecidas para a configuração da rede nos cenários simulados. Em seguida, são mostrados os resultados obtidos através destas simulações. As simulações foram realizadas no *Berkeley Network Simulator* versão 2.31 [60].

5.1 Parâmetros e Premissas

Para a simulação foram criados cenários nos quais a quantidade de nós geradores de tráfego foi variada. Desta forma, foi possível avaliar os mecanismos de detecção e punição em diferentes condições de carga da rede. Além disso, foi assumido que ataques de conluio não ocorrem na rede. No entanto, o SDE pode trabalhar em conjunto com um mecanismo de prevenção específico para este tipo de mau comportamento [61]. Os nós da rede foram configurados para não utilizarem informações de detecções realizadas por terceiros nas avaliações do MAPA. Portanto, todas as avaliações são realizadas somente através das informações de detecção observadas localmente.

Foram utilizados 81 nós nos cenários simulados, nos quais 10 desses nós eram escolhidos aleatoriamente para serem nós egoístas. Os nós foram dispostos em forma de grade de 9 linhas por 9 colunas. A dimensão desta grade foi de 180 metros para cada lado e o alcance do rádio de cada nó foi definido em 26 metros. O protocolo de roteamento utili-

zado foi o *Dynamic Source Routing* (DSR) e o protocolo MAC foi o IEEE 802.11g com a taxa de 54 Mbps. O tráfego de dados utilizado foi o *Constant Bit Rate* (CBR), variando entre 200 kbps e 250 kbps. Foram executadas 50 rodadas, com o tempo de simulação igual a 250 segundos por rodada. A cada rodada os nós assumiam posições aleatórias na rede. Os resultados mostrados nos gráficos foram obtidos com um intervalo de confiança igual a 95%.

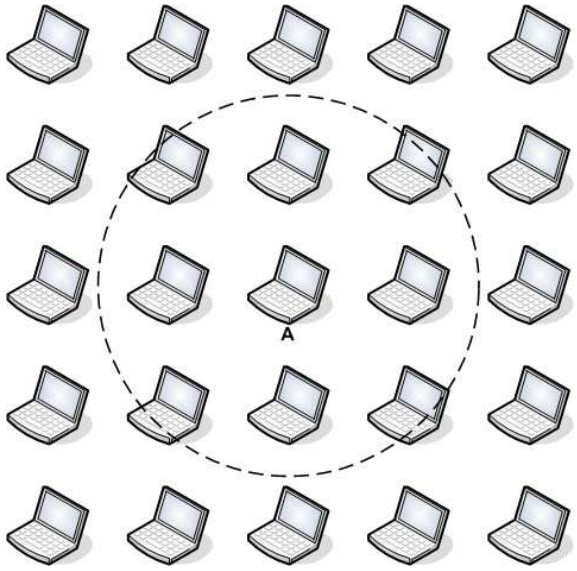


Figura 5.1: Representação gráfica da rede ad hoc simulada.

A Figura 5.1 apresenta uma rede ad hoc de acordo com a disposição dos nós, além de mostrar o alcance máximo de cada rádio. Todos os nós possuem a mesma área de alcance do nó que se encontra em destaque na figura, o nó A. Portanto, os nós centrais são capazes de atingir até 8 nós ao realizar alguma transmissão na rede, como também monitorar esta mesma quantidade de nós. É importante lembrar que os nós que geram tráfego na rede são sorteados a cada 50 segundos de simulação, dentro da mesma rodada de 250 segundos, ou seja, são analisadas cinco diferentes posições de geração de tráfego em uma mesma rodada.

Dois tipos de descartes de pacotes foram simulados nos cenários analisados, seguindo os modelos de descarte executados por dois ataques comuns em redes ad hoc, conhecidos na literatura como buraco negro [16] e buraco cinza [20], e descritos na Seção 2.2.1. Portanto, no ataque do tipo buraco negro o nó egoísta descarta todos os pacotes que chegam até ele e que deveriam ser encaminhados. Já no ataque do tipo buraco cinza os nós

descartam os pacotes aleatoriamente selecionados que deveriam ser encaminhados.

Para ataque do tipo buraco cinza é sorteado um valor entre 0 e 20, que determina o número de pacotes que devem ser encaminhados até o descarte ser realizado. Por exemplo, caso seja sorteado um valor igual a 12, o nó atacante encaminhará 12 pacotes normalmente e descartará o próximo pacote que ele receber. Após realizar o descarte, o nó atacante repetirá este procedimento de sorteio. Como falado anteriormente, este ataque tem como objetivo enganar os mecanismos de detecção de nós egoístas.

Para o limiar de tolerância L foi atribuído um valor igual a 0.2. Com esse valor pode-se dizer que o MAPA é mais tolerante, pois para ser executado um bloqueio temporário de maior período de tempo, a probabilidade do nó ser egoísta tem que ser maior do que 0.2. Os bloqueios temporários foram configurados para serem de 2 segundos quando $L > p$ e 4 segundos quando $L \leq p$. Para registrar os endereços dos nós bloqueados é utilizada uma lista. O tamanho dessa lista depende da capacidade computacional de cada nó. Caso a lista fique cheia, os endereços registrados há mais tempo são removidos para que os novos endereços dos nós detectados sejam registrados.

Para determinar o valor de I foi considerada a condição de que o SDE não conseguiria diferenciar os dois ataques de egoísmo na rede, já que o ataque do tipo buraco negro é considerado o caso extremo do ataque do tipo buraco cinza. Isso foi considerado porque o SDE poderia simplesmente usar um valor de $I = 1$ para não bloquear definitivamente os nós cooperativos em uma rede sob ataque de buraco negro, ou seja, como os nós egoístas descartam todos os pacotes no ataque de buraco negro, o SDE só precisaria observar um encaminhamento correto para desconsiderar o último bloqueio temporário enviado ao nó cooperativo. No entanto, o ataque de buraco cinza não permite esta facilidade na atribuição do valor de I, pois os descartes são realizados de forma aleatória. Neste contexto, o valor de I foi escolhido de forma a tornar a proposta adaptável às diferentes formas de comportamentos egoístas na rede. Portanto, o valor de I é determinado a partir dos valores dos parâmetros L, p, e e D de cada avaliação, da seguinte forma:

- quando $L > p$ na avaliação atual, então I assume o valor de e;
- quando $L \leq p$ na avaliação atual, então I assume o valor do resultado de $D \cdot e$.

Para analisar o impacto do SDE/MAPA, foram implementados o Watchdog e o Pathrater, que são mecanismos de referência na área de detecção de maus comportamentos em redes ad hoc. Os parâmetros do Watchdog e do Pathrater foram definidos de acordo com os valores apresentados na Seção 3.4.1. Além destes mecanismos, foi implementado um mecanismo que bloqueia definitivamente os nós a cada detecção realizada, chamado de mecanismo Intolerante.

Na análise dos resultados foi observado que ambos os mecanismos de detecção, SDE e Watchdog, geram aproximadamente a mesma quantidade de detecções incorretas ao utilizarem o mesmo valor no limiar de detecção. Portanto, conclui-se que o principal fator causador do falso-positivo na punição está na relação entre a detecção realizada e a punição máxima aplicada ao nó. Por exemplo, no SDE/MAPA, está na relação entre a detecção de D eventos egoístas no SDE e o bloqueio definitivo em k incidências de avaliações no MAPA. Já no Watchdog/Pathrater está na relação entre a quantidade de detecções de eventos egoístas determinada pelo limiar do Watchdog e a redução de reputação no Pathrater.

Para comparar os resultados de falso-positivos e de detecções corretas, a quantidade de detecções e punições de eventos egoístas foi definida de forma equivalente para as propostas SDE/MAPA e Watchdog/Pathrater. Além disso, esta quantidade de detecções foi escolhida com o objetivo de analisar o quanto a tolerância em observar um determinado número de eventos egoístas pode afetar negativamente o desempenho da rede. Portanto, o valor do limiar de detecção utilizado na proposta do Watchdog/Pathrater foi igual a 10, ou seja, o Pathrater reduzirá a reputação do nó após 10 detecções realizadas pelo Watchdog. Para assumir com equivalência a quantidade de detecções utilizada pelo Watchdog/Pathrater, o SDE/MAPA pode ser analisado de quatro diferentes formas, variando os valores de k e D, ou seja: $k = 1$ e $D = 10$, não analisando a influência dos bloqueios temporários utilizados pelo MAPA; $D = 1$ e $k = 10$, aplicando um bloqueio temporário a cada detecção realizada pelo SDE; $k = 5$ e $D = 2$, executando uma punição temporária a cada duas detecções realizadas pelo SDE; e $k = 2$ e $D = 5$, realizando um bloqueio temporário a cada 5 eventos egoístas detectados.

Como se pretende analisar a influência de todos os parâmetros da proposta apresen-

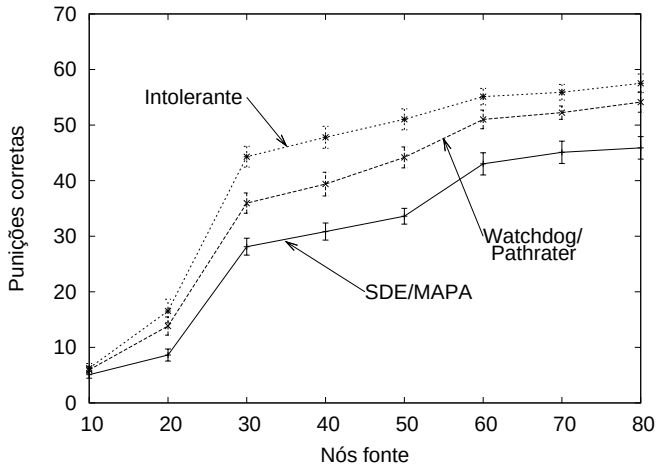
tada, foi escolhida a opção que atribui a quantidade mínima de bloqueios temporários, ou seja, $D = 5$ para o SDE e $k = 2$ para o MAPA. Desta forma, foi possível analisar a influência de um bloqueio temporário e um bloqueio definitivo, totalizando 10 eventos detectados para o nó ser evitado definitivamente, equivalente a quantidade de detecções utilizada pelo Watchdog/Pathrater.

5.2 Resultados

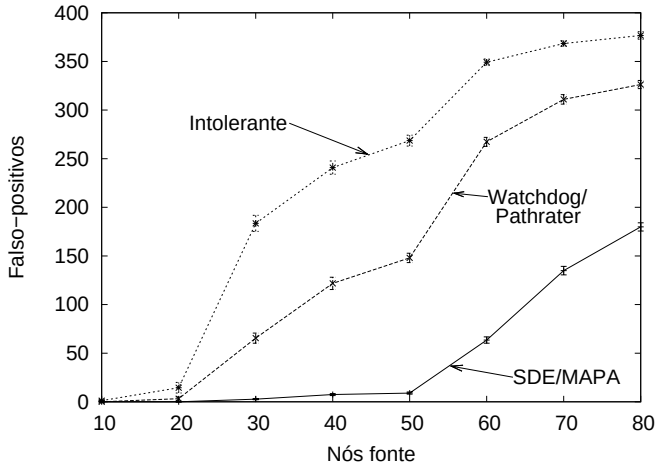
As Figuras 5.2 e 5.3 apresentam a quantidade de nós egoístas que recebem as punições corretamente e a taxa de falso-positivos gerada por cada proposta. Estes resultados foram divididos em cenários onde os nós egoístas executavam o descarte total de pacotes (buraco negro) ou o descarte aleatório de pacotes (buraco cinza).

De acordo com os resultados apresentados nas Figuras 5.2 e 5.3, a proposta que mais detectou nós egoístas na rede foi a proposta Intolerante. Entretanto, esta proposta foi considerada a mais ineficiente, devido à elevada quantidade de falso-positivos gerada. Ao observar a Figura 5.4, pode-se concluir que este mecanismo de detecção e punição não é recomendado em uma rede ad hoc, pois para conseguir punir uma maior quantidade de nós egoístas este mecanismo realiza diversas punições incorretas aos nós cooperativos e, conseqüentemente, reduz o desempenho da rede.

Apesar de ter obtido uma maior quantidade de punições corretas, o Pathrater gerou uma maior quantidade de falso-positivos quando comparado com o MAPA. De acordo com os valores definidos na análise do Pathrater, apresentados na Seção 3.4.1, cada nó da rede inicia com a reputação neutra, ou seja, igual a 0.5 e pode atingir um valor máximo de 0.8. A partir destes valores estabelecidos, o Pathrater reduz a reputação do nó em 0.1 ao detectar uma quantidade de eventos egoístas determinada pelo limiar do Watchdog, ou seja, reduz a cada 10 eventos egoístas detectados nas simulações realizadas. Para tentar minimizar a influência que os falso-positivos exercem no mecanismo de escolha de rotas, o Pathrater assume que se um nó não for detectado pelo Watchdog durante um período de 200ms de simulação, então a sua reputação será incrementada em 0.01. O problema está na possibilidade do nó egoísta ter a sua reputação aumentada enquanto está sendo



(a) Punições corretas.



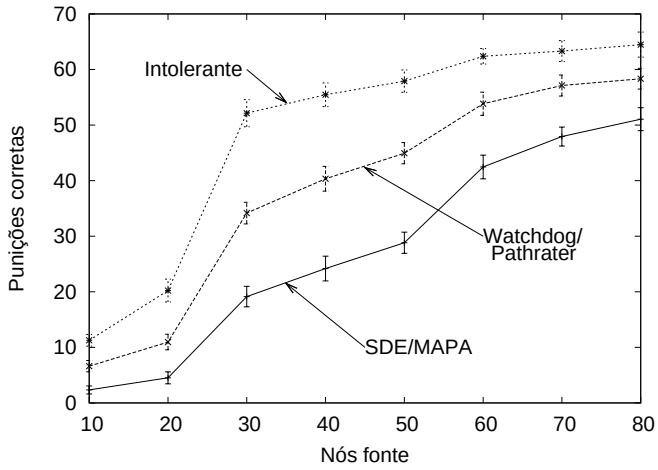
(b) Falso-positivos.

Figura 5.2: Rede ad hoc sob ataque de buraco negro.

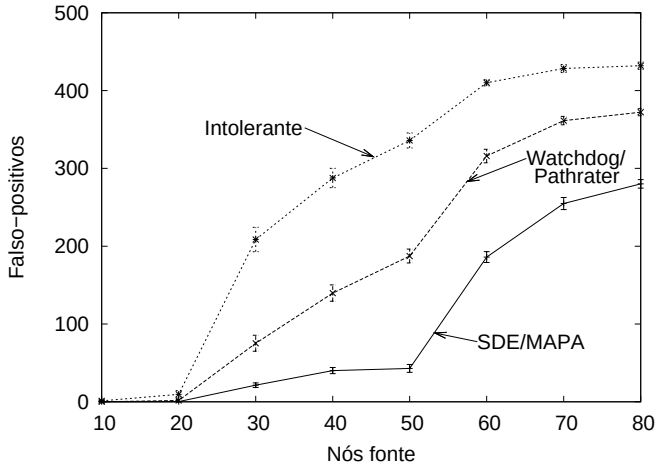
evitado, podendo voltar a ser usado novamente, e prejudicar o desempenho da rede.

Ao analisar os resultados apresentados, é possível notar que o MAPA é mais eficiente na punição máxima aplicada ao nó monitorado em ambos os cenários, pois ele é capaz de bloquear corretamente uma quantidade significativa de nós egoístas, gerando uma menor taxa de falso-positivos, como apresentado na Figura 5.4. Esta eficiência na detecção é conseqüência da avaliação que o MAPA realiza, considerando todos os eventos executados pelo nó monitorado, tanto os eventos normais como os eventos egoístas.

Além da avaliação, outro fator que favorece a redução dos falso-positivos e o aumento das punições corretas é o parâmetro I. O parâmetro I faz com que os bloqueios definitivos



(a) Punições corretas.

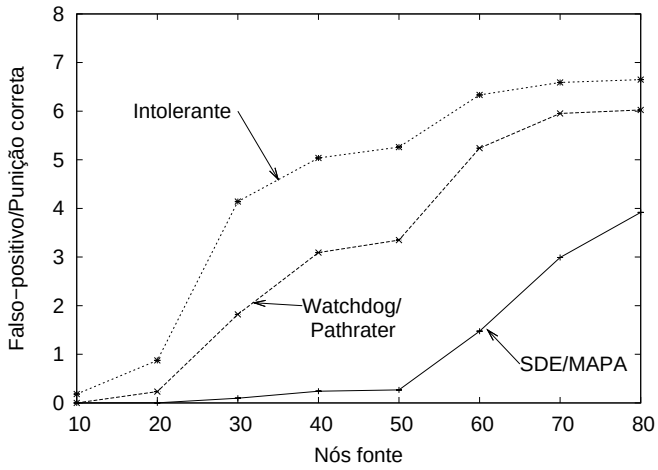


(b) Falso-positivos.

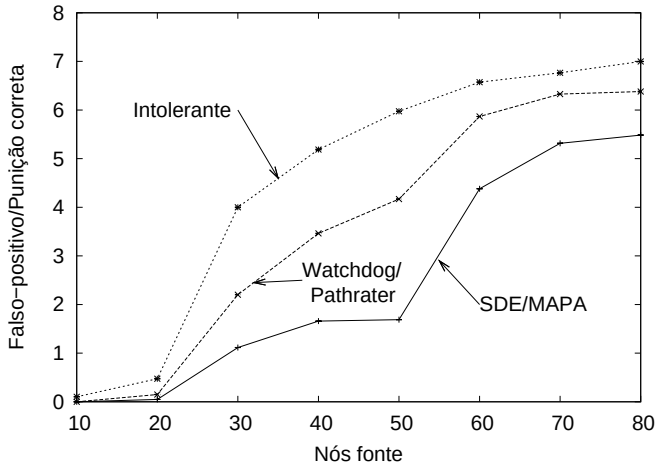
Figura 5.3: Rede ad hoc sob ataque de buraco cinza.

sejam mais precisos, ou seja, se um nó foi bloqueado temporariamente devido a algum problema temporário da rede ad hoc, então ele poderá provar que é um nó cooperativo ao realizar I encaminhamentos de pacotes. Além disso, os nós egoístas são detectados com uma maior facilidade, pois para afetarem significativamente o desempenho da rede ao tentarem economizar os seus recursos ou atacar a rede, os nós egoístas terão que descartar os pacotes com uma maior freqüência. Assim, quanto mais freqüentes forem os descartes realizados, mais facilmente o nó egoísta será detectado.

O bloqueio temporário é um fator que influencia na menor quantidade de punições corretas realizadas pelo MAPA, pois a cada bloqueio temporário executado, os nós egoístas bloqueados são removidos da tabela de rotas dos nós que os detectaram. Portanto, a



(a) Rede ad hoc sob ataque de buraco negro.



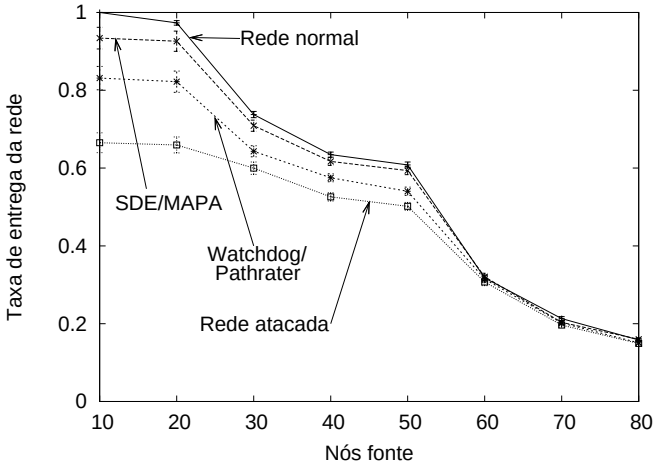
(b) Rede ad hoc sob ataque de buraco cinza.

Figura 5.4: Quantidade de falso-positivos gerada para cada punição correta.

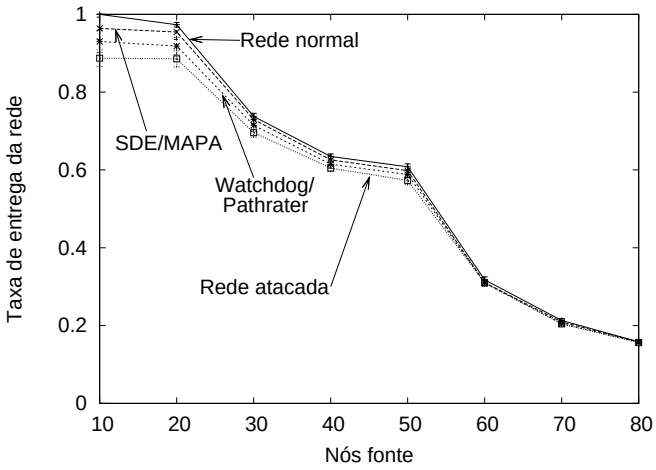
queda na quantidade de detecções corretas ocorre porque os nós cooperativos começam a usar as rotas que evitam estes nós bloqueados, fazendo com que os nós egoístas sejam requisitados com uma menor freqüência. Além disso, como o DSR armazena as rotas em *cache*, o nó egoísta poderá não ser usado por um tempo maior do que o período determinado pelo bloqueio temporário, já que as novas rotas descobertas poderão ser usadas por um longo período de tempo.

Para analisar a taxa de entrega da rede foi simulado um cenário para servir de referência como o melhor caso, sem a presença de nós egoístas, observando somente a influência da saturação da rede na taxa de entrega. Para o pior caso foi simulado um cenário contendo 10 nós egoístas, que representam 12,5% do total de nós, e não foi usado nenhum

mecanismo de detecção e punição na rede. De acordo com Buchegger *et al*, quando mais de 10% dos nós da rede são maliciosos, o desempenho da rede começa a ser degradado significativamente [50]. Ao observar os resultados apresentados, nota-se uma queda nos valores a partir do instante em que 20 nós estão gerando tráfego na rede. Analisando a Figura 5.5, este instante pode representar o início da saturação da rede.



(a) Rede ad hoc sob ataque de buraco negro.



(b) Rede ad hoc sob ataque de buraco cinza.

Figura 5.5: Taxa de entrega da rede.

A partir dos resultados obtidos na Figura 5.5, foi observado que tanto o SDE/MAPA quanto o Watchdog/Pathrater alcançaram melhores resultados quando comparados com uma rede que não utiliza um mecanismo de detecção e punição de nós egoístas, mesmo quando a rede está próxima da saturação. No entanto, o SDE/MAPA foi a proposta que mais se aproximou da curva ideal da taxa de entrega da rede. Isto é conseqüência da

utilização de bloqueios temporários enviados aos nós detectados antes do bloqueio definitivo ser executado, fazendo com que o nó seja punido ao descartar a metade dos pacotes que o Pathrater necessitaria para punir um nó egoísta. Portanto, mesmo com a baixa saturação na rede, o Pathrater obteve um pior desempenho, devido ao dobro de pacotes que precisariam ser descartados para que o nó egoísta tivesse a sua reputação reduzida. Além disso, os falso-positivos no Pathrater causam a redução na reputação dos nós cooperativos, fazendo com que a reputação de um nó cooperativo possa tornar-se menor ou igual à reputação de um nó egoísta. Como a reputação é uma métrica para a seleção de rotas, os nós egoístas continuam sendo usados nas rotas escolhidas pelo Pathrater. As Figuras 5.5(a) e 5.5(b) mostram a influência destes problemas na taxa de entrega da rede.

Capítulo 6

Conclusões

REDES ad hoc necessitam da cooperação dos nós para que as funções de roteamento e encaminhamento de dados sejam realizadas corretamente. Entretanto, a premissa de cooperação não pode ser considerada verdadeira, pois a cooperação é uma característica assumida e não forçada. Dessa forma, as operações destas redes podem ser facilmente corrompidas por nós maliciosos, que violam as especificações dos protocolos. Para evitar que estes nós possam prejudicar o funcionamento das redes ad hoc, existem mecanismos de segurança classificados em dois grupos: pró-ativo e reativo. Os mecanismos pró-ativos tentam solucionar os problemas de segurança na primeira linha de defesa, ou seja, tipicamente através de mecanismos criptográficos. Por outro lado, os mecanismos reativos objetivam a detecção das anormalidades na rede para, posteriormente, iniciar alguma resposta que seja capaz de combatê-las.

O objetivo deste trabalho é propor um mecanismo reativo eficiente, através de um sistema de detecção de nós maliciosos. Dentre todos os requisitos para a criação de um mecanismo reativo de segurança em redes ad hoc, citados no Capítulo 3, uma das principais dificuldades está relacionada à elevada quantidade de falso-positivos gerada nas detecções. Os falso-positivos ocorrem quando um nó cooperativo é classificado como um nó malicioso. Portanto, os SDIs devem ser cautelosos nas detecções de comportamentos suspeitos, pois existe uma dificuldade em distinguir um ataque de um problema temporário da rede ad hoc.

Neste trabalho foi apresentado o MAPA, um mecanismo eficiente para evitar nós egoístas em redes ad hoc. Além do MAPA, foi implementado um mecanismo de detecção de nós egoístas, chamado de SDE. Estes dois mecanismos são usados em conjunto para detectar, avaliar e punir os nós egoístas em redes ad hoc. O principal objetivo do MAPA é reduzir a quantidade de falso-positivos nas punições aplicadas aos nós detectados pelo SDE, calculando a probabilidade de um nó detectado ser realmente um nó egoísta.

A partir dos resultados obtidos na análise matemática, foi possível observar que o SDE e o MAPA se mostraram mais tolerantes e precisos nas punições aplicadas aos nós egoístas, evitando bloqueios definitivos nas detecções erradas e observando por mais tempo o comportamento das estações. Nota-se ainda que, ao utilizar o SDE/MAPA, é adicionada uma funcionalidade que força a participação das estações egoístas, oferecendo outras oportunidades para que estas se comportem corretamente na rede. Além disso, o SDE/MAPA obedece todos os requisitos para a criação de um sistema de detecção de intrusão em redes ad hoc, apresentados no Capítulo 3. Portanto, pode-se concluir sobre a proposta: não adiciona novos problema à rede; realiza punições de forma transparente para o usuário; utiliza uma menor quantidade de recursos computacionais, pois o algoritmo implementado é simples e eficiente; é capaz de continuar ativo após desastres, pois a detecção e a punição são realizadas localmente; não permite que outros usuários afetem o seu funcionamento; reduz a quantidade de falso-negativos, por utilizar o modelo de detecção baseado em anomalias; e reduz a quantidade de falso-positivos, pois avalia todas as detecções antes de executar uma punição.

A eficiência da proposta foi comprovada através dos resultados obtidos nas simulações, pois se pôde concluir que ao utilizar o SDE/MAPA, nas condições de rede simuladas, a taxa de entrega da rede é aumentada em até 27% quando comparado com uma rede que não utiliza um mecanismo de detecção e punição de nós egoístas. Além disso, o SDE/MAPA também aumentou a taxa de entrega da rede em até 12% em relação à taxa de entrega do Watchdog/Pathrater. Portanto, nas condições de rede simuladas, a proposta que se mostrou mais eficiente na detecção e punição de nós egoístas em redes ad hoc foi o SDE/MAPA. Os principais fatores que influenciaram na eficiência do MAPA foram: os menores valores obtidos na razão entre a quantidade de falso-positivos gerada para cada punição aplicada corretamente (Figura 5.4); a menor influência que os falso-positivos

exercem sobre o SDE/MAPA; e o uso do parâmetro I, que observa os comportamentos normais realizados, possibilitando que um nó cooperativo avaliado e punido prove que não é um nó egoísta. Além destes fatores, o MAPA não permite que os nós punidos se beneficiem dos nós da rede como acontece no Pathrater [5], pois o nó que é bloqueado temporariamente ou definitivamente é proibido de realizar qualquer comunicação com os nós que o bloquearam.

As principais contribuições deste trabalho foram: a análise matemática da quantidade de eventos que devem ser observados para que uma punição seja aplicada, baseando-se na probabilidade do nó ser malicioso; a criação de um mecanismo preciso de punição de nós egoístas em redes ad hoc; a possibilidade de ajuste da tolerância de cada nó em relação ao número de eventos maliciosos que devem ser observados; a utilização de um parâmetro capaz de oferecer novas oportunidades aos nós cooperativos que foram classificados como maliciosos; e a possibilidade de utilizar a proposta com outros protocolos de roteamento, pois o único requisito é conhecer o próximo salto da rota utilizada. Além destas contribuições, foi importante analisar o grau de tolerância que o mecanismo de detecção é capaz de assumir, sem provocar tantos prejuízos no desempenho da rede como um todo.

Foi observado que a proposta apresentada melhorou o desempenho da rede ao detectar e isolar os nós egoístas. Entretanto, existem alguns tópicos que podem ser pesquisados como trabalhos futuros, com o objetivo de melhorar algumas fraquezas encontradas nos mecanismos propostos. Portanto, estas principais fraquezas são listadas a seguir:

- ao utilizar um valor fixo de D nas detecções, um nó malicioso pode descobrir o valor de D, tentar ser detectado no máximo $D - 1$ vezes, e continuar utilizando a rede, evitando a D-ésima detecção. Uma possível solução para esse problema seria determinar o valor de D com base na saturação da rede ou na mobilidade dos nós. Entretanto, ao usar somente esta forma de atribuição do valor de D, um nó malicioso continua sendo capaz de descobri-lo, pois ele pode aprender a lógica utilizada para determinar o valor de D. Portanto, além de definir um valor de D para cada nó vizinho através desta observação, seria importante utilizar um número aleatório sorteado para ser adicionado ou reduzido ao valor de D. Esta solução também pode ser utilizada para o valor de k usado pelo MAPA, pois ao conhecer

os valores de D e de k, o nó malicioso pode tentar receber somente as punições temporárias, evitando receber um bloqueio definitivo;

- como I é determinado pela quantidade de eventos realizados pelo nó malicioso detectado na última avaliação, então o nó malicioso seria capaz de saber quantos comportamentos, egoístas e normais, ele executou até ter sido bloqueado temporariamente. Desta forma, o nó malicioso poderia fazer com que as punições temporárias fossem desconsideradas, realizando no máximo ($e \cdot D$) eventos normais após cada bloqueio temporário. Como uma possível solução, poderia ser proposto um mecanismo semelhante ao descrito anteriormente para D e k, ou seja, adicionar ou subtrair um número sorteado aleatoriamente ao valor de I.

- como o total de eventos observados (e) pode assumir valores elevados, seria interessante utilizar uma janela de eventos, que determinaria uma quantidade de eventos normais necessária para que o nó não receba uma punição ao ser detectado. O problema em utilizar esta janela é a possibilidade de ocorrência de falso-negativos, pois um nó malicioso poderia se comportar corretamente por um determinado período de tempo, iniciar um ataque, e não ser detectado pelo SDE.

Como trabalhos futuros, seria interessante observar o desempenho do SDE/MAPA na detecção e na punição de novos tipos de ataques em redes ad hoc. Além disso, poderiam ser feitas análises de desempenho do SDE/MAPA em cenários reais, onde uma grande quantidade de falso-positivos pode ser gerada nas detecções e nas respostas aplicadas aos nós da rede.

Referências Bibliográficas

[1] DAE-KI KANG; FULLER, D.; HONAVAR, V. Learning classifiers for misuse and anomaly detection using a bag of system calls representation. *Em Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop, 2005. IAW '05.* (Junho de 2005), 118–125.

[2] ANANTVALEE, T., E WU, J. A survey on intrusion detection in mobile ad hoc networks. *Wireless/Mobile Network Security - Springer*, 7 (2006), 159–180.

[3] MACLEOD, H.; LOADMAN, C. C. Z. Experimental studies of the 2.4-GHz ISM wireless indoor channel. *Em Proceedings of the 3rd Annual Communication Networks and Services Research Conference, 2005.* (Maio de 2005), 63–68.

[4] MOSS, J., FITTON, M., STREET, A., BROWN, K., CONSTANTINOU, C., E EDWARDS, D. Spatio-temporal variability analysis of the wideband microcellular environment. *Em 48th IEEE Vehicular Technology Conference. VTC 98. vol.1* (Maio de 1998), 293–297.

[5] MARTI, S., GIULI, T. J., LAI, K., E BAKER, M. Mitigating routing misbehavior in mobile ad hoc networks. Em *Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCom)* (New York, NY, USA, 2000), pág. 255–265.

[6] FERNANDES, N. C., MOREIRA, M. D. D., VELLOSO, P. B., COSTA, L. H. M. K., E DUARTE, O. C. M. B. Ataques e mecanismos de segurança em redes ad hoc. Em *Minicursos do Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais - SBSeg'2006* (Santos, SP, Brazil, Agosto de 2006), Sociedade Brasileira de Computação–SBC, pág. 49–102.

[7] HU, Y.-C., E PERRIG, A. A survey of secure wireless ad hoc routing. *IEEE Security and Privacy Magazine vol.2*, 3 (Maio/Junho de 2004), 28–39.

[8] PERKINS, C.E.; ROYER, E. Ad-hoc on-demand distance vector routing. *Em Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications.* (Fevereiro de 1999), 90–100.

[9] JOHNSON, D. B., E MALTZ, D. A. Dynamic source routing in ad hoc wireless networks. Em *Mobile Computing* (1996), vol. 353, Mobile Computing (ed. T. Imielinski and H. Korth), Kluwer Academic Publishers, pág. 153–181.

[10] PERKINS, C. E., E BHAGWAT, P. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. Em *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications* (New York, NY, USA, 1994), ACM, pág. 234–244.

[11] JACQUET, P., MUHLETHALER, P., CLAUSEN, T., LAOUITI, A., QAYYUM, A., E VIENNOT, L. Optimized link state routing protocol for ad hoc networks. *Em Proceedings of IEEE International Multi Topic Conference, IEEE INMIC. Technology for the 21st Century.* (Dezembro de 2001), 62–68.

[12] MALKIN, G. RFC 2453 - routing information protocol version 2. Relatório técnico, IETF Request for Comments, Novembro de 1998.

[13] JOHNSON, D. B., MALTZ, D. A., E BROCH, J. *DSR: the dynamic source routing protocol for multihop wireless ad hoc networks*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, Dezembro de 2000, pág. 139–172.

[14] PATWARDHAN, A., PARKER, J., JOSHI, A., IORGA, M., E KARYGIANNIS, T. Secure routing and intrusion detection in ad hoc networks. Em *PERCOM '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications* (Washington, DC, USA, 2005), IEEE Computer Society, pág. 191–199.

[15] YANG, H., LUO, H., YE, F., LU, S., E ZHANG, L. Security in mobile ad hoc networks: challenges and solutions. *IEEE Wireless Communications [see also IEEE Personal Communications]* (Fevereiro de 2004).

[16] AL-SHURMAN, M., YOO, S.-M., E PARK, S. Black hole attack in mobile ad hoc networks. Em *ACM-SE 42: Proceedings of the 42nd annual Southeast regional conference* (New York, NY, USA, 2004), ACM Press, pág. 96–97.

[17] GUPTE, S., E SINGHAL, M. Secure routing in mobile wireless ad hoc networks. *Elsevier Ad Hoc Networks Journal vol.1*, 1 (Julho de 2003), 151–174.

[18] KARLOF, C.; WAGNER, D. Secure routing in wireless sensor networks: attacks and countermeasures. *Em Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications.* (Maio de 2003), 113–127.

[19] ISLAM, M.M.; POSE, R. K. C. An intrusion detection system for suburban ad-hoc networks. *Em IEEE TENCON 2005, Region 10* (Novembro de 2005), 1–6.

[20] HU, Y.-C., PERRIG, A., E JOHNSON, D. B. Ariadne: a secure on-demand routing protocol for ad hoc networks. *Wireless Network vol.11. Kluwer Academic Publishers*, 1-2 (2005), 21–38.

[21] MING YU; KULKARNI, S.; LAU, P. A new secure routing protocol to defend byzantine attacks for ad hoc networks. *Em 7th IEEE Malaysia International Conference on Communication Networks. vol.2* (Novembro de 2005), 1126–1131.

[22] MARANO, S., MATTA, V., E TONG, L. Distributed detection in the presence of byzantine attack in large wireless sensor networks. Em *Military Communications Conference, MILCOM 2006* (Outubro de 2006), vol. 1, pág. 1–4.

[23] LAMPORT, L., SHOSTAK, R., E PEASE, M. The byzantine generals problem. *Em ACM Transactions on Programming Languages and Systems (TOPLAS) vol.4*, 3 (1982), 382–401.

[24] SANZGIRI, K., DAHILL, B., LEVINE, B. N., SHIELDS, C., E BELDING-ROYER, E. M. A secure routing protocol for ad hoc networks. Em *ICNP '02: Proceedings*

of the 10th IEEE International Conference on Network Protocols (Washington, DC, USA, 2002), IEEE Computer Society, pág. 78–89.

[25] PIRO, C., SHIELDS, C., E LEVINE, B. N. Detecting the sybil attack in mobile ad hoc networks. *Securecomm and Workshops* (Agosto/Setembro de 2006), 1–11.

[26] JUN, L., ZHE, L., DAN, L., E YE, L. A security enhanced aodv routing protocol based on the credence mechanism. *Em Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing. vol.2* (23-26 de Setembro de 2005), 719–722.

[27] HU, Y.-C., PERRIG, A., E JOHNSON, D. B. Rushing attacks and defense in wireless ad hoc network routing protocols. Em *WiSe '03: Proceedings of the 2nd ACM workshop on Wireless security* (New York, NY, USA, 2003), ACM, pág. 30–40.

[28] HEADY, R., LUGER, G., MACCABE, A., E SERVILLA, M. The architecture of a network level intrusion detection system. Relatório Técnico CS90-20, University of New Mexico, Department of Computer Science, Agosto de 1990.

[29] AL-SHAER, E. Managing firewall and network-edge security policies. *Em IEEE/IFIP Network Operations and Management Symposium, 2004. NOMS 2004. vol.1* (Abril de 2004), 926.

[30] MISHRA, A., NADKARNI, K., E PATCHA, A. Intrusion detection in wireless ad hoc networks. Em *IEEE Wireless Communications* (Fevereiro de 2004), vol. 11, IEEE Computer Society, pág. 48–60.

[31] HAN, H., LU, X.-L., REN, L.-Y., E CHEN, B. Taichi: An open intrusion automatic response system based on plugin. *Em International Conference on Machine Learning and Cybernetics.* (Agosto de 2006), 66–77.

[32] HEBERLEIN, L. T., MUKHERJEE, B., E LEVITT, K. N. Internetwork security monitor: An intrusion-detection system for large-scale networks. Em *15th National Computer Security Conference* (Outubro de 1992), pág. 262–271.

[33] ZAMBONI, D. M. SAINT: A security analysis integration tool. Em *Proceedings Systems Administration, Networking and Security Conference* (1996), Washington DC.

[34] SAMFAT, D.; MOLVA, R. IDAMN: an intrusion detection architecture for mobile networks. *IEEE Journal on Selected Areas in Communications vol.15*, 7 (Setembro de 1997), 1373–1380.

[35] FRINCKE, D. A., MCCONNELL, J., TOBIN, D., MARCON, J., E POLLA, D. A framework for cooperative intrusion detection. Em *Proceedings of 21st National Information Systems Security Conference* (Outubro de 1998).

[36] ASAKA, M., OKAZAWA, S., E TAGUCHI, A. The implementation of IDA: An intrusion detection agent system. Em *Proceedings of the 11th FIRST Conference* (Junho de 1999).

[37] NETWORK ICE. Blackice user's manual, version 1.0, 2000. Disponível em http://www.networkice.com/Support/Docs/BlackICEProUG.pdf.

[38] ANZEN COMPUTING. Anzen flight jacket for nfr, 2000. Disponível em http://www.anzen.com/afj/afj_overview.html.

[39] MICHAEL ERLINGER. Intrusion detection exchange format (IDWG), 2008. Disponível em http://www.ietf.org/html.charters/OLD/idwg-charter.html.

[40] CUNHA, D. O., DUARTE, O. C. M. B., PUJOLLE, G. An enhanced routing metric for fading wireless channels. Em *IEEE Wireless Communications and Networking Conference - IEEE WCNC* (Las Vegas, USA, Março/Abril de 2008), IEEE Computer Society.

[41] ZHANG, Y., E LEE, W. Intrusion detection in wireless ad-hoc networks. Em *Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCom '00)* (New York, NY, USA, 2000), ACM Press, pág. 275–283.

[42] BUCHEGGER, S., E BOUDEC, J.-Y. L. Performance analysis of the CONFIDANT protocol. Em *MobiHoc '02: Proceedings of the 3rd ACM international symposium*

on Mobile ad hoc networking & computing (New York, NY, USA, 2002), ACM, pág. 226–236.

[43] ZENG, X., BAGRODIA, R., E GERLA, M. GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks. Em *Workshop on Parallel and Distributed Simulation* (1998), pág. 154–161.

[44] BLAZE, M., FEIGENBAUM, J., E LACY, J. Decentralized trust management. Em *SP '96: Proceedings of the 1996 IEEE Symposium on Security and Privacy* (Washington, DC, USA, 1996), IEEE Computer Society, pág. 164.

[45] MIT PRESS. *The Official PGP User's Guide*. Cambrigde, 1995.

[46] RESNICK, P., KUWABARA, K., ZECKHAUSER, R., E FRIEDMAN, E. Reputation systems. *Communications of the ACM vol.43*, 12 (2000), 45–48.

[47] BUCHEGGER, S., E BOUDEC, J.-Y. L. The effect of rumor spreading in reputation systems for mobile ad-hoc networks. Em *WiOpt'2003: Modeling and Optimization in Mobile, Ad hoc and Wireless Networks* (Março de 2003).

[48] BUCHEGGER, S., E BOUDEC, J.-Y. L. Coping with false accusations in misbehavior reputation systems for mobile ad hoc networks. Em *EPFL Technical Report* (2003), no. IC/2003/31.

[49] MICHIARDI, P., E MOLVA, R. CORE: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. Em *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security* (Deventer, The Netherlands, The Netherlands, 2002), Kluwer, B.V., pág. 107–121.

[50] BUCHEGGER, S., E BOUDEC, J.-Y. L. Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. Em *Proceedings of the Tenth Euromicro Workshop on Parallel, Distributed and Network-based Processing* (Canary Islands, Spain, Janeiro de 2002), IEEE Computer Society, pág. 403–410.

[51] ZHONG, S., LI, L. E., LIU, Y. G., E YANG, Y. R. On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks: an in-

tegrated approach using game theoretic and cryptographic techniques. *Wireless Network vol.13*, 6 (2007), 799–816.

[52] BANSAL, S., E BAKER, M. Observation-based Cooperation Enforcement in Ad Hoc Networks. *Em ArXiv Computer Science e-prints* (Julho de 2003).

[53] CHIANG, T.-C., E HUANG, Y.-M. Group keys and the multicast security in ad hoc networks. *Em International Conference on Parallel Processing Workshops (ICPPW)* (2003), 385.

[54] BOUASSIDA, M. S., CHRISMENT, I., E FESTOR, O. Efficient group key management protocol in manets using the multipoint relaying technique. Em *Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06)* (Washington, DC, USA, 2006), IEEE Computer Society, pág. 64.

[55] CHAR, B. W., GEDDES, K., LEONG, B., MONAGAN, M., E WATT, S. *Maple V Language Reference Manual*. Springer-Verlag, New York, 1991.

[56] KNUTH, D. E. *The art of computer programming, volume 1 (3rd ed.): fundamental algorithms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.

[57] CHENG, W., OOI, W. T., MONDET, S., GRIGORAS, R., E MORIN, G. An analytical model for progressive mesh streaming. Em *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia* (New York, NY, USA, 2007), ACM, pág. 737–746.

[58] NAGAPPAN, N., E BALL, T. Use of relative code churn measures to predict system defect density. Em *ICSE '05: Proceedings of the 27th international conference on Software engineering* (New York, NY, USA, 2005), ACM, pág. 284–292.

[59] QIMING CHEN; CHUANWEN JIANG; WENZHENG QIU; MCCALLEY, J. Probability models for estimating the probabilities of cascading outages in high-voltage

transmission network. *Em IEEE Transactions on Power Systems vol.21*, 3 (Agosto de 2006), 1423–1431.

[60] FALL, K., E VARADHAN, K. *The ns Manual*. UC Berkeley, LBL, USC/ISI, and Xerox, 2006.

[61] YOUNIS, M., GHUMMAN, K., E ELTOWEISSY, M. Key management in wireless ad hoc networks: collusion analysis and prevention. *Em 24th IEEE International on Performance, Computing, and Communications Conference. IPCCC 2005.* (2005), 199–203.