UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

SISTEMA DE DETECÇÃO DE INTRUSÃO POR ANOMALIA NO COMPORTAMENTO PARA REDES AD-HOC

FÁBIO MIZIARA SILVEIRA

ORIENTADOR: RICARDO STACIARINI PUTTINI

DISSERTAÇÃO DE MESTRADO

PUBLICAÇÃO: 293/07

BRASÍLIA / DF: 2/2007

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

SISTEMA DE DETECÇÃO DE INTRUSÃO POR ANOMALIA NO COMPORTAMENTO PARA REDES AD-HOC

FÁBIO MIZIARA SILVEIRA

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE.

APROVADA POR:

RICARDO STACIARINI PUTTINI, Doutor, ENE-UnB
(ORIENTADOR)

RAFAEL TIMÓTEO DE SOUSA JR., Doutor, ENE-UnB
(EXAMINADOR INTERNO)

JACIR LUIZ BORDIM, Doutor, CIC-UnB
(EXAMINADOR EXTERNO)

PAULO HENRIQUE PORTELA DE CARVALHO, Doutor, ENE-UnB
(SUPLENTE)

DATA: BRASÍLIA/DF, 28 DE FEVEREIRO DE 2007.

FICHA CATALOGRÁFICA

SILVEIRA, FÁBIO MIZIARA. Sistema de Detecção de Intrusão por Anomalia no Comportamento para Redes Ad-Hoc [Distrito Federal] 2007, (89)p., 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2007).

Dissertação de Mestrado – Universidade de Brasília, Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

1. MANET 2. IDS
3. Comportamento

I. ENE/FT/UnB. II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

SILVEIRA, FÁBIO MIZIARA(2007). Sistema de Detecção de Intrusão por Anomalia no Comportamento para Redes Ad-Hoc. Dissertação de Mestrado, Publicação 293/2007, Departamento de Engenharia Elétrica, Universidade de Brasília , Brasília , DF, (89)p.

CESSÃO DE DIREITOS

NOME DO AUTOR: Fábio Miziara Silveira
TÍTULO DA DISSERTAÇÃO: Sistema de Detecção de Intrusão por Anomalia no Comportamento para Redes Ad-Hoc.
GRAU/ANO: Mestre/2007.

Fábio Miziara Silveira
SQSW 305 Bloco B Apto 102
CEP 70673-422 – Cruzeiro – DF - Brasil

À minha família, e a todos que sempre me apoiaram.

AGRADECIMENTOS

Ao orientador, amigo e professor Ricardo Staciarini Puttini, que me deu esta oportunidade e me mostrou o caminho certo para a conclusão deste trabalho.

Aos Prof. Rafael Timóteo de Sousa Júnior e Jacir Luiz Bordim, pelas conversas enriquecedoras, colaboração e amizade.

Aos meus pais, Joaquim e Claudia, e à minha noiva, Mariane, pelo imenso apoio e incentivo dado durante todo o tempo em que estive envolvido neste trabalho e pela total confiança depositada para que eu pudesse chegar até o fim de mais essa etapa.

A todos os meus amigos e amigas que me incentivaram durante todos esses anos juntos na caminhada para o sucesso.

A todos, os meus sinceros agradecimentos.

Fábio Miziara Silveira

RESUMO

A segurança em Manet possui muitas questões em aberto. Devido a suas características, este tipo da rede necessita a proteção preventiva e corretiva. Neste trabalho, nós focamos na proteção corretiva propondo um modelo de IDS por anomalia para Manet. O projeto e o desenvolvimento do IDS consideram 3 estágios principais: construção do comportamento normal, detecção da anomalia e atualização do modelo considerado normal. Um modelo paramétrico de mistura é usado para modelar o comportamento a partir dos dados de referência. O algoritmo de detecção é baseado em critérios de classificação Bayesiana. As variáveis da MIB são usadas fornecer a informação necessária ao IDS. Experiências com ataques de DoS e Scanner que validam o modelo também são apresentadas.

ABSTRACT

Manet security has a lot of open issues. Due to its characteristics, this kind of network needs preventive and corrective protection. In this work, we focus on corrective protection proposing an anomaly IDS model for Manet. The design and development of the IDS are considered in our 3 main stages: normal behavior construction, anomaly detection and model update. A parametrical mixture model is used for behavior modeling from reference data. The associated Bayesian classification leads to the detection algorithm. MIB variables are used to provide IDS needed information. Experiments of DoS and scanner attacks validating the model are presented as well.

ÍNDICE

Capítulo **Página**

ÍNDICE DE TABELAS

ÍNDICE DE FIGURAS

SIGLAS

AODV - *Ad Hoc On-Demand Distance Vector*

ARP - *Address Resolution Protocol*

CBR - *Constant Bit Rate*

DDoS - *Distributed Denial of Service*

DNS - *Domain Name System*

DoS - *Denial of Service*

EM - *Estimação-Maximização*

FTP - *File Transfer Protocol*

GMM - *Gaussian Mixture Model*

ICMP - *Internet Control Message Protocol*

IDS - *Intrusion Detection System*

IDWG - *Intrusion Detection Working Group*

IETF - *Internet Engineering Task Force*

IP - *Internet Protocol*

MAC - *Medium Access Control*

MANET - *Mobile Ad-Hoc Network*

MIB - *Management Information Base*

ML - *Máxima Verossimilhança*

NS - *Network Simulator*

SNMP - *Simple Network Management Protocol*

TCL - *Tool Command Language*

TCP - *Transmission Control Protocol*

UDP - *User Datagram Protocol*

1. INTRODUÇÃO

As redes móveis ad hoc, também chamadas de MANET, têm apresentado um grande crescimento nos últimos anos, devido à sua maior facilidade de instalação, quando comparadas às redes tradicionais, à possibilidade de movimentação dos dispositivos durante o uso da rede, permitindo assim maior praticidade e flexibilidade, e à tolerância à falhas, já que a permanente adaptação e reconfiguração das rotas permitem que perdas de conectividade entre os nodos possam ser facilmente resolvidas desde que uma nova rota possa ser estabelecida. Por isso, estas redes possuem aplicações em ambientes militares, comerciais e educacionais [1].

Neste tipo de rede, por não existir uma entidade central, pela característica dinâmica da topologia, pela necessidade de economia de energia e pelo fato de a comunicação ocorrer por meio de ondas de rádio, os serviços fornecidos requerem algoritmos distribuídos e adaptativos visando economia de energia, baixo tempo de convergência e robustez. Assim, os nodos dependem uns dos outros para prover qualquer serviço que dependa da rede (e.g. roteamento) [1].

Um fato importante a ser observado é que prover um suporte à segurança em redes MANET é um desafio devido às características citadas acima que acarretam, como conseqüências, em vulnerabilidade a ataques, em necessidade de disposição dos serviços em qualquer hora e em qualquer lugar e em necessidade de soluções escaláveis para redes sem fio de larga escala.

A segurança em redes móveis Ad Hoc (Manet) é um tópico ativo nas pesquisas recentes. A maior parte dos trabalhos atuais em segurança de Manet foca em algum tipo de proteção preventiva [2,3]. Entretanto, as entidades de rede em uma Manet consistem de hardware e software, geralmente sem boa proteção física, e por isso a ocorrência de mau funcionamento de entidades comprometidas não pode ser negligenciada. Conseqüentemente, a segurança deve ser projetada de uma maneira que o serviço de rede se mantenha robusto mesmo na presença de nós mal comportados. No geral, comprometer uma entidade da rede pode levar a revelar toda a informação confidencial ao invasor, o que faz com que a maioria dos mecanismos preventivos de segurança

falhe. Os sistemas de detecção de Intrusão (IDS) são comuns em cenários onde um mecanismo corretivo de segurança é requerido para lidar com as limitações dos mecanismos somente preventivos de segurança.

No que diz respeito aos projetos de IDS, duas abordagens básicas podem ser consideradas: detecção de intrusão por uso incorreto e por anomalia no comportamento. Na detecção por uso incorreto, uma assinatura do ataque deve ser fornecida explicitamente, conduzindo a uma identificação positiva de uma ocorrência do ataque. Se a fonte do ataque (e.g. nó comprometido) puder também ser identificada como parte do processo da detecção, uma ação corretiva simples (resposta) consiste em excluir o nó do atacante da rede. Um caso de um sistema de segurança baseado em proteção preventiva e corretiva pela combinação do emprego de forte autenticação e IDS por uso incorreto pode encontrado em [47]. A detecção por anomalia é completamente diferente. O comportamento atual do sistema monitorado (e.g. rede) é comparado repetidamente com algum comportamento de referência, que é indicado previamente (comportamento normal). Neste caso, como a existência dos ataques não é realizada explicitamente, a fonte do problema não pode ser precisamente identificada. Assim, as ações corretivas (resposta) devem concentrar na mitigação do efeito do ataque.

Neste trabalho, é apresentado o projeto de um IDS para MANET que segue a abordagem de detecção por anomalia no comportamento. Dada a ausência de centralização, a mobilidade dos nodos e os enlaces sem fio das Manets, algumas (senão todas) as tarefas requeridas no processo de detecção de intrusão devem ser executadas de forma distribuída e cooperativa [58].

Nós estamos especialmente interessados em detectar o comportamento anômalo do tráfego da rede devido a ataques por inundação de pacotes (e.g. DoS) e ataques de scanner em redes móveis ad hoc. Este projeto é baseado em um modelo estatístico do comportamento de referência usando modelos de mistura [44] a fim lidar com um tráfego observado composto pela mistura dos perfis diferentes de tráfego devido às aplicações diferentes de rede. O algoritmo de detecção é baseado em critérios de classificação Bayesiana.

Além disso, são realizados experimentos para a validação do modelo apresentado. Para a realização dos experimentos aplica-se os modelos de geração de trafego e modelo de mobilidade para redes móveis Ad Hoc. Por fim, é apresentado um modelo de simulação para a geração dos dados necessários para a avaliação do desempenho do IDS apresentado.

Este trabalho se divide em cinco capítulos, sendo que o segundo trata do estado da arte dos Sistemas de Detecção de Intrusão, apresentando as principais abordagens no assunto. No Capítulo 3 é apresentado um Sistema de Detecção de Intrusão por anomalia, mostrando um modelo de mistura que é utilizada para modelar o comportamento e também o algoritmo de detecção. O Capítulo 4 mostra os experimentos realizados para validar o modelo e seus resultados. Por fim, o Capítulo 5 conclui esta dissertação com as considerações finais e trabalhos futuros.

2. TRABALHOS RELACIONADOS

A maior parte das pesquisas realizadas hoje em segurança de Manet está focada em mecanismos de proteção preventiva dos protocolos básicos (e.g. roteamento), por meio de um mecanismo de autenticação [60,61,62]. De uma maneira geral, essas soluções com mecanismos de proteção somente preventivos não são tolerantes à existência de entidades comprometidas na rede. Essas soluções de segurança podem ser reforçadas por mecanismos de segurança corretivos, tais como sistemas de detecção de intrusão (IDS).

Os sistemas de detecção de intrusão são projetados para detectar ataques contra sistemas de informação em geral. Realmente, é muito difícil, senão impossível, fornecer sistemas de informação possivelmente seguros e mantê-los seguros durante toda sua utilização. Desta forma, os sistemas de detecção de intrusão têm o objetivo de monitorar a utilização destes sistemas com a intenção de detectar o aparecimento de estados inseguros.

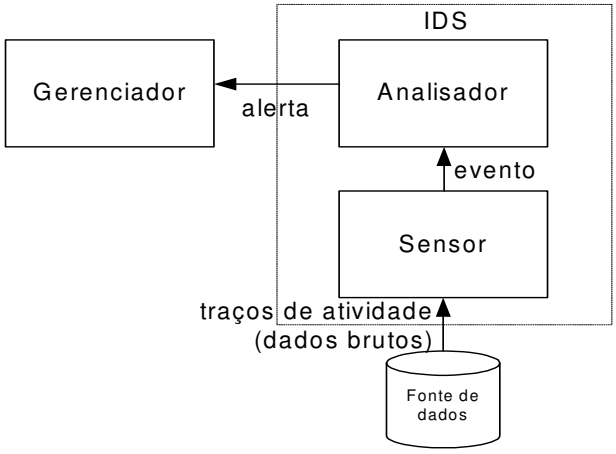

Figura 2-1 – Framework de detecção de intrusão do IDWG
(Fonte: R. Puttini [1] p. 37)

As pesquisas acerca de detecção de intrusão atualmente, possuem seu interesse dividido em três processos básicos: coleta de dados, projeto do algoritmo de detecção (análise) e gerenciamento de alertas. O grupo de trabalho do IETF (IDWG), sobre detecção de intrusão, define os componentes que executam essas tarefas [56], conforme

mostrado na Figura 2-1. O módulo sensor é responsável por coletar os dados brutos acerca da operação do sistema que esta sendo monitorado (e.g. traços de auditoria, pacotes de rede). Os dados coletados são pré-processados e resultando em eventos que são enviados ao módulo analisador, onde os eventos gerados são avaliados em termos de um mecanismo de detecção de intrusões. Se esses eventos forem sensitivos, o módulo analisador gera alertas, que são repassados ao módulo gerenciador. Este módulo, por fim, além de realizar a correlação e classificação dos alertas, com objetivo de refinar a análise prévia, provê as informações necessárias para a resposta aos ataques detectados.

Para a classificação dos sistemas de detecção de intrusão atuais H. Debar et al. [19] propõem uma taxionomia, mostrada na Figura 2-2. Os critérios de classificação propostos são Método de detecção, Comportamento em caso de detecção, Fonte de dados, Paradigma de detecção e Freqüência de uso.

O método de detecção descreve as características do módulo analisador. Quando o IDS usa informações acerca do comportamento normal do sistema monitorado, buscando detectar variações deste estado normal, o método de detecção é dito com base em comportamento. Se o IDS utiliza informações acerca dos ataques que podem ser detectados (assinaturas de ataques), diz-se tratar de um método de detecção por uso incorreto.

O comportamento em caso de detecção descreve a resposta do IDS aos ataques detectados. Quando o sistema reage ativamente a um ataque executando ações corretivas (fechando brechas) ou pró-ativas (registrando possíveis atacantes, fechando serviços), o sistema é classificado com ativo. Se o sistema meramente gera e envia alertas (incluindo pager, etc.), ele é dito passivo.

A fonte de dados discrimina os IDS com base no tipo de informação de entrada que eles analisam. Essa informação pode ser trilha de auditoria (e.g. log de sistema) em um computador, pacotes de rede, log de aplicativos ou mesmo alertas gerados por outros sistemas de detecção de intrusão.

O paradigma de detecção descreve o mecanismo de detecção usado pelo IDS. Estes sistemas podem avaliar estados (seguro/inseguro) ou transições (de seguro para inseguro).

Por fim, a Freqüência de uso descreve a utilização do IDS. Certos IDS são usados na monitoração contínua e em tempo real do sistema alvo, enquanto outros são executados periodicamente.

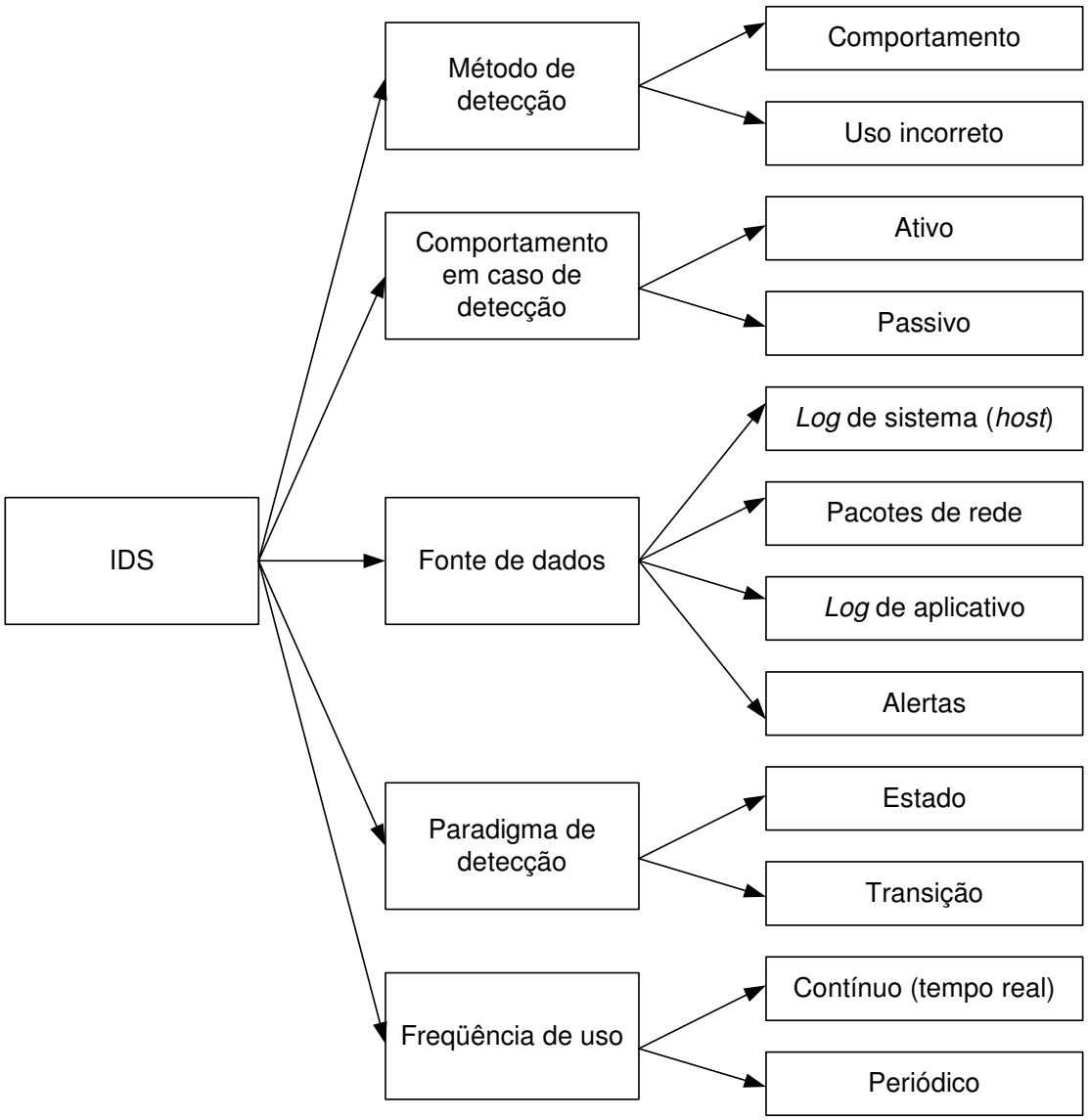

Figura 2-2 – Taxionomia dos Sistemas de Detecção de Intrusão

(Fonte: R. Puttini [1] p. 39)

Os IDS no contexto Manet possuem requisitos específicos que não são compatíveis com abordagens tradicionais para detecção de intrusão. Assim como os demais serviços de segurança, o IDS deve ser distribuído, auto-organizado e, se possível, operar localizadamente. Como o projeto de IDS para Manet é um assunto bastante recente, apresentamos a seguir um apanhado das principais iniciativas de concepção e de projeto de sistemas de detecção de intrusão que atendam a esses requisitos, mesmo que esses sistemas não tenham sido especificamente projetados para nosso ambiente alvo.

Em uma primeira análise, é possível dizer que muito se tem feito e estudado recentemente sobre sistemas de detecção de intrusão distribuídos. A Tabela 2-1 apresenta as principais propostas de sistemas de detecção de intrusão distribuídos, que já atingiram maturidade suficiente em seu desenvolvimento para permitir a validação de princípios e direções consideradas em cada uma das propostas.

Pode-se verificar que todas as propostas apresentadas na Tabela 2-1, exceto CSM, EMERALD e SPARTA, são hierarquicamente organizadas em torno de um nodo central. Este nodo central é o cerne do IDS e usa informações coletadas de forma distribuída para detectar intrusões. Nestas arquiteturas, a distribuição está restrita ao processo de coleta de dados. Devido a esse fato, no contexto das manets, essas arquiteturas se mostram inadequadas.

Uma arquitetura completamente distribuída é encontrada no CMS. Neste sistema, um IDS local é instalado em cada nodo cooperando para uma identificação colaborativa do originador de conexões de rede. Já a arquitetura do EMERALD foi especialmente projetada para acomodar necessidades de escalabilidade em redes grandes. Este IDS é feito de nodos genéricos comunicantes, denominados monitores, que são instalados em cada sistema. Entre as arquiteturas na Tabela 2-1 apenas SPARTA foi especificamente projetado para ambientes de rede sem fio. Entretanto, esse sistema não considera ataques contra a camada de rede sento projetado para detectar ataques contra aplicações distribuídas.

Tabela 2-1 – Principais propostas de IDS distribuídos

IDS	Fonte de Dados	Método de Detecção	Pré-processamento distribuído	Detecção centralizada	Análise em tempo-real	Tipo de Resposta
AAFID [50]	Sistema	Uso incorreto	Sim	Sim	Sim	Passivo
DIDS [52]	Sistema/Rede	Híbrido	Sim	Sim	Sim	Passivo
Grids [51]	Sistema/Rede	Híbrido	Sim	Sim	Não	Passivo
CSM [55]	Sistema	Anomalia	Sim	Não	Sim	Ativa
JiNao [22]	MIB/Rede	Híbrido	Sim	Sim	Sim	Passivo
EMERALD [43]	Sistema/Rede	Híbrido	Sim	Não	Sim	Ativo
IDA [10]	Sistema	Uso incorreto	Agentes móveis	Sim	Sim	Passivo
SPARTA [32]	Sistema/Rede	Uso incorreto	Agentes móveis	Não	Sim	Passivo

Não se pode dizer que o projeto de IDS para Manet é uma questão completamente nova, este assunto já foi tratado recentemente [9,64,36,41,54,58]. Y Zhang e W. Lee [58] introduzem os requisitos básicos para este tipo especial de IDS. Em outros trabalhos, R Puttini et al. introduzem os conceitos preliminares de arquitetura [9,42,45]. S. Gwalani et al. propõem em [24], um IDS para Manet que é essencialmente projetado para reforçar a segurança do protocolo de roteamento. Entretanto, este IDS possui uma arquitetura centralizada, não atendendo a um dos requisitos das Manets.

Um IDS para detecção de ataques contra o protocolo de roteamento formado por vários sensores, que monitoram promiscuamente os enlaces de rede, é apresentado por V. Mittal e G. Vigna [41]. Noções de colaboração aparecem neste IDS, porém o mecanismo de detecção supõe que informações globais de topologia estão disponíveis.

Em Manet, o mais apropriado seria usar informações de topologia localizada, pois a topologia é dinâmica e as informações de topologia globais podem não estar completamente atualizadas.

Projetos de IDS para Manet com base em uma estratégia de detecção por anomalia são apresentados por Y. Huang et al. [64] e C.-Y. Tseng et al. [54]. O inconveniente desses trabalhos reside na ausência de cooperação entre os nodos, sendo que cada nodo age isoladamente na detecção de ataques.

Por fim, uma estratégia de detecção e resposta à intrusão para lidar com nodos não cooperantes em redes ad hoc é apresentado por S. Marti et al. em [36]. Entretanto, não há nenhuma noção de serviços de segurança colaborativa nesta abordagem. No trabalho em [57] é apresentada uma solução de segurança com base em uma versão modificada do AODV, que utiliza um mecanismo de detecção de intrusão combinado com um sistema de fichas de filiação (tokens) que são usadas para garantir o acesso dos nodos aos serviços de roteamento. Entretanto, a esta solução não se incorpora nenhuma proteção preventiva (autenticação). Ao invés, apenas um mecanismo simples de verificação de vizinhança é usado. Infelizmente, como mencionado anteriormente, este mecanismo é baseado em uma hipótese errônea de que endereços MAC não podem ser personificados. Adicionalmente, o mecanismo de detecção de intrusão está restrito apenas à inundação de mensagens RREP, não generalizando para lidar com todos os ataques descritos em termos de fabricação, modificação e personificação de outras mensagens do protocolo de roteamento.

3. SISTEMA DE DETECÇÃO DE INTRUSÃO

O IDS possui uma função fundamental e talvez indispensável nos modelos de segurança para redes móveis Ad-Hoc. A função do IDS nas Manets é prover uma monitoração pró-ativa do estado da segurança. Como resultado desta monitoração, o IDS permite identificar violações da política de segurança (ataques). Além disso, por meio de uma resposta também pró-ativa à detecção de ataques, o IDS interage com outros serviços de segurança (e.g. autenticação, certificação, controle de acesso) para eliminar as causas de um ataque ou mitigar os seus efeitos. Para eliminar as causas de um ataque ou mitigar os efeitos, por exemplo, pode-se revogar o certificado de nodos mal comportados ou reconfigurar a filtragem de pacotes para descartar os pacotes que violam a política de segurança da rede.

Neste capítulo é apresentada uma proposta de um sistema de detecção de intrusão projetado para Manet. Para isso, define-se uma nova arquitetura para IDS, levando-se em conta os requisitos específicos do ambiente Manet. Desta forma, o IDS apresentado possui características de distribuição, auto-organização e localização. A concepção do IDS considera uma completa distribuição do processo de detecção de intrusão, ao invés de limitar a distribuição às tarefas de coleta de dados, que é a abordagem mais comum na maioria dos projetos de IDS.

O procedimento de detecção de intrusão possui basicamente dois passos. A coleção e a análise de dados de auditoria fornecidos pela rede, pelo sistema ou pelas aplicações. Quando as intrusões são detectadas, estas devem ser reportadas à gerência de segurança. Da mesma maneira, a detecção pode disparar uma resposta automática, com objetivo de extinguir as causas ou efeitos da intrusão. Devido a ausência de centralização, a mobilidade dos nodos e os enlaces sem fio das Manets, algumas (senão todas) as tarefas requeridas no processo de detecção de intrusão descritas acima devem ser executadas de forma distribuída e cooperativa [58].

No projeto deste trabalho, um IDS local é colocado em cada nodo da Manet. Os IDSs comunicam-se entre si através de um mecanismo que leva em conta as restrições de contexto Manet, i.e. banda passante limitada ou conectividade pobre. Se um IDS

falha em cooperar durante algum período de tempo (e.g. o nodo moveu-se, falhou ou está comprometido), o serviço de detecção de intrusão não pode ser degradado. As redundâncias inerentes às Manets servem de compensação para os nodos que não estão cooperando no processo de detecção, sendo possível que mais de um nodo possa acompanhar e detectar a evolução de um mesmo ataque.

Devido aos requisitos de uso da banda passante e de escalabilidade a colaboração deve ser mantida entre um número restrito de nodos. A colaboração entre os nodos é mantida apenas entre os nodos da vizinhança. A razão de se executar algumas das tarefas de detecção apenas na vizinhança local é duplamente justificada pela natureza das Manets. Um nodo destas redes deve sempre coletar e manter informações sobre seus vizinhos, qualquer que seja o protocolo de roteamento que esteja em uso. Além disso, qualquer informação indo para ou vindo de um determinado nodo deve ser roteada através de um de seus vizinhos, quando o seu encaminhamento pode ser promiscuamente monitorado devido à natureza difusora dos enlaces sem fio [59]. Assim, vizinhos de nodo que está sendo atacando são as fontes primárias de informação sobre o estado do nodo que sofre os efeitos de um ataque, naturalmente elegíveis. Os vizinhos são também elegíveis como pares para a colaboração com objetivo de descobrir novas informações que não estão disponíveis localmente, acerca de uma intrusão em curso.

3.1. DETECÇÃO DE INTRUSÃO POR COMPORTAMENTO

Partindo do principio de que uma intrusão pode ser detectada pela observação de um desvio do comportamento normal ou esperado de um sistema ou usuário, as técnicas de detecção de intrusão baseadas em comportamento podem ser projetadas. O comportamento normal ou válido é extraído de informações prévias de referência sobre o sistema monitorado. Então, o IDS compara o modelo do comportamento de referência com a atividade corrente e gera alarmes cada vez que uma divergência em relação ao modelo original é observada. Isto significa que todo comportamento observado que não pode ser ajustado ao modelo de referência admitido previamente passa ser considerado como anomalia e possivelmente representam a ocorrência de um ataque.

Muitos sistemas baseados em modelagem do comportamento tem sido propostos e testados e, ainda que eles tenham diferentes características e arquiteturas, sua concepção e desenvolvimento podem ser, de maneira geral, descritos em termos de três fases: Construção de um modelo de comportamento normal ou válido; Detecção e Atualização do modelo de comportamento assumido [19].

O Estagio de Construção de um modelo de comportamento normal ou válido consiste na modelagem do comportamento de referência do sistema. Neste estágio, a maioria das hipóteses acerca das fontes de informação que serão usadas e de como esses dados podem ser processados para construir um modelo que descreva, de forma consistente e desejavelmente completa a operação do sistema. De uma maneira sistemática, mas possivelmente particular, este estágio pode ser dividido em Identificação de qual tipo de informação de auditoria deve ser usada para descrever o comportamento normal do sistema, Construção do modelo de comportamento e Obtenção de informação prévia de referencia (treinamento).

Primeiramente temos a identificação de qual tipo de informação de auditoria deve ser usada para descrever o comportamento normal do sistema. Em geral, o mesmo tipo de informação usado neste estágio de modelagem deve ser usado (como entrada) no estágio de detecção. É importante notar ainda que um pré-processamento das informações coletadas é comumente requerido.

Em seguida temos a construção do modelo de comportamento. Vários tipos de modelos podem ser usados para descrever o comportamento normal de um sistema. Muitos sistemas foram desenvolvidos usando-se modelagem estatística [29], redes neurais ou algoritmos genéticos, além de diversas outras técnicas. Estes modelos têm, em geral, uma arquitetura prévia com diversos parâmetros que devem ser ajustados automaticamente usando-se algum tipo de algoritmo de aprendizagem ou otimização. Em alguns casos, a arquitetura e o algoritmo de aprendizado estão fortemente ligados, sendo este realizado através de um processo iterativo e adaptativo para o ajuste progressivo dos parâmetros do modelo. Nesse tipo de modelo existe ainda a possibilidade de se atualizar os parâmetros previamente ajustados para acompanhar as mudanças que ocorrem no comportamento normal do sistema. Outras abordagens,

baseadas em sistemas especialistas, são igualmente possíveis, mas a atualização automática do modelo devida a mudanças do comportamento pode ser mais difícil.

Por fim, temos a obtenção de informação prévia de referencia (treinamento). Mesmo depois de se definir o tipo de informação a ser usado e a arquitetura do modelo, obter-se um bom conjunto de informações iniciais de referência não é tarefa evidente. De fato, as exigências que esses dados iniciais não contenham qualquer tipo de utilização anômala e que esse conjunto de dados seja representativo de todo o comportamento normal do sistema são condições usualmente difíceis de satisfazer.

A fase de Detecção consiste na realização de inferências acerca do estado de operação do sistema, comparando-se informações adquiridas do uso corrente do sistema com o modelo de comportamento ajustado no estágio anterior. Esses novos dados relativos à utilização posterior do sistema são apresentados ao IDS. A concepção do algoritmo de detecção pode variar em função do tipo de informação usado ou da arquitetura do sistema, mas também deve considerar outros critérios como desempenho e robustez, caso trate-se de um sistema que realize a detecção em tempo real. Independentemente do tipo de modelo utilizado, o algoritmo deve permitir uma clara definição para o desvio a ser avaliado. O desvio pode ser definido binariamente, isto é, para toda informação nova apresentada ao IDS este vai discriminá-la em normal ou anômala. Alternativamente, o desvio pode assumir a forma de um teste de significância, i.e. um comportamento observado pode ser avaliado em válido/inválido com uma probabilidade dada. Em muitos IDS que operam com técnicas de detecção por anomalia, requer-se que os alertas gerados nesta fase sejam pós-processados com objetivo de se eliminar falsos positivos. Isto ocorre, essencialmente, porque em detecção de intrusão por comportamento, ao contrário da detecção de intrusão por uso incorreto, não se tem o reconhecimento positivo de um ataque, mas apenas a indicação de uma atividade não observada durante a modelagem ocorreu.

Por fim, temos a atualização do modelo de comportamento assumido. Na medida em que o comportamento de utilização de um sistema muda, o modelo de comportamento deve ser atualizado para se evitar a indicação (alertas) errônea de anomalias pelo IDS. Esta atualização pode ser realizada continuamente, mas atualizações periódicas podem ser toleradas, mesmo em sistema que operem em tempo real. Usualmente, as atualizações do modelo de comportamento ocorrem

gradativamente e em longo termo, evitando-se a ocorrência de adaptações distorcidas devido a uma utilização errônea por um curto período de tempo. Assim, se uma grande mudança do comportamento de utilização do sistema estiver para ser realizada, é necessário um reinício do sistema, construindo-se novamente o modelo de comportamento do sistema a partir de novas informações de referência que reflitam o novo comportamento, ou mesmo alterando-se aspectos da arquitetura do modelo de comportamento do sistema para adaptá-lo à nova realidade. É importante salientar que essa atualização gradativa do sistema dá a oportunidade a um adversário de progressivamente induzir um comportamento errôneo ao sistema que será aprendido com um comportamento aceitável pelo mecanismo de atualização do modelo de comportamento válido. Esta é uma das desvantagens mais marcantes da abordagem de detecção de intrusão por comportamento.

Neste trabalho, deseja-se projetar um IDS por comportamento. Utiliza-se uma abordagem de modelagem estatística para a construção do modelo de comportamento. Nesse tipo de abordagem, é usualmente necessário mapear-se os eventos de auditoria disponíveis para coleta e análise em variáveis aleatórias, isto é, em domínios numéricos, ainda que alguns eventos de auditoria já sejam observáveis nesta forma. Em um primeiro exercício, pretende-se observar valores numéricos que reflitam as condições de tráfego e uso da banda passante por determinados tipos de aplicações e protocolos em uma Manet. Nesse sentido, vale ressaltar que a modelagem estatística é usualmente complexa, uma vez que aplicações e protocolos diferentes possuem regras estatísticas bem distintas. Assim, decide-se construir um modelo de mistura de distribuições e ajustá-lo ao conjunto de dados candidato a caracterizar o tráfego normal de uma Manet.

Obviamente, a caracterização do que seria um perfil de tráfego normal para uma Manet ainda é um problema aberto e pouco consenso foi construído a este respeito. Assim, nossa modelagem deve ser ajustada a um perfil de tráfego considerado normal para uma Manet específica, cujas condições de operação estejam claramente definidas. Isto faz com que o trabalho ora apresentado tenha um caráter preliminar. Não obstante, e a exemplo de trabalhos similares nesta mesma ceara [14], os resultados obtidos preliminarmente permitem discriminar com sucesso ataques que se caracterizam por mudanças significativas nos padrões de tráfego, tais como ataques de DoS e de scanner

de rede. Desse modo, fica sendo este o objetivo principal do IDS baseado em comportamento descrito nesta seção.

3.1.1. Modelos de Mistura de Distribuições para Caracterização Estatística do Comportamento

Um modelo de mistura de distribuições é usualmente utilizado para modelar a função densidade de probabilidade (f.d.p.) de uma variável aleatória y d-dimensional (cujas realizações são extraídas do domínio de informações de auditoria) e pode ser formalmente definido como se segue:

Seja $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n]^T$ um vetor de realização observável de y. Um modelo de mistura de distribuições para esses dados é definido expressando-se a f.d.p. dos dados como a combinação linear de funções nucleares básicas conforme mostrado na Eq. 3-1:

$$p(\mathbf{y}_i) = \sum_{k=1}^{K} w_k g_k(\mathbf{y}_i, \boldsymbol{\theta}_k) \qquad \text{Eq. 3-1}$$

onde: g representa cada função nuclear, wk são os fatores de ponderação de cada função nuclear e θk são os parâmetros das funções nucleares. O vetor $\boldsymbol{\Psi} = [w_1, w_2, ..., w_k, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_k]$ representa todos os parâmetros desconhecidos do modelo de mistura, os quais se deseja ajustar para enquadrar $\boldsymbol{\Psi}$ à Y. O número K é a ordem do sistema, geralmente fixo.

Este modelo de mistura finito vem sendo usado para modelar distribuições de diversos fenômenos supostamente aleatórios [39]. Um algoritmo iterativo para otimização, por um critério de máxima verossimilhança (ML) foi apresentado em [20] e é denominado algoritmo de estimação-maximização (EM).

Como o conjunto de dados de referência deve conter informações sobre diferentes comportamentos válidos, é normalmente útil que tais dados sejam clusterizados. O uso de modelos de mistura em custerização automática de dados é

imediato, pela adoção de um modelo de mistura parametrizado [17,48]. Este modelo é definido assumindo-se que cada função nuclear individualmente representa a f.d.p. de cada cluster no conjunto de dados. Assim, um modelo de mistura de ordem K é diretamente aplicável em situações onde $\overline{Y}$ pode ser identificado como originário de uma mistura populacional de K grupos. Nestes casos, os coeficientes wk equivalem à probabilidade de cada cluster p(k). Do mesmo modo, as probabilidades posteriores de cada realização p(yi | k) podem ser obtidas, dado os valores de cada função nuclear em $\overline{y}$. Uma vez que p(yi) pode ser diretamente estimada de Eq. 3-1, pelo teorema de Bayes, pode-se obter uma estimativa para as probabilidades posteriores na forma p(k| yi)=p(yi |k)p(k)/p(yi). O conhecimento prévio da ordem do modelo pode não estar disponível e é conveniente que este possa ser inferido automaticamente. Neste trabalho, desenvolve-se um algoritmo para determinação automática de K, baseado em uma otimização por um critério de maximização de entropia. Este algoritmo, adaptado de [48] para modelos paramétricos, é descrito na seção seguinte.

Para dados multivariados, o caso especial de funções nucleares gaussianas multivariadas forma um modelo conhecido como modelo de mistura de gaussianas (GMM). Este modelo, em particular, pode ser facilmente ajustado iterativamente pelo algoritmo EM, pois existem formas fechadas para a computação realizada em cada iteração. Além disso, o algoritmo possui boas propriedades de convergência, dado uma correta estimativa de K. Assim, neste trabalho, considera-se o caso de um GMM. Portanto, a descrição do algoritmo EM apresentada aqui está particularizada para este caso. Em [20,39] pode-se encontrar os detalhes de uma descrição mais genérica do algoritmo EM. Esta abordagem pode parecer um pouco restritiva, mas alguns pontos precisam ser destacados acerca do GMM. Em análise de clusterização, a aplicação de GMM parametrizados é largamente adotada, pois os clusters assumem formato elíptico. Entretanto, para um conjunto de dados contendo um grupo ou grupos de observações derivados de um número de populações normais maior que a ordem do sistema ou observações que não têm característica normal, modelos mais gerais precisam ser usados. Modelos paramétricos mais genéricos, usando as distribuições uniforme, gaussiana, gaussiana com deslocamento e escalonamento, além de distribuições t, podem ser facilmente derivados, pois o algoritmo EM já foi definido para esses casos

[39,17,8]. Outro tipo importante de função nuclear para a qual seria interessante derivar-se o algoritmo EM são as distribuições de Pareto, largamente usadas para modelagem de tráfego intermitente, em rajada. Outra alternativa possível consiste na adoção de modelos de mistura semi-paramétricos [48], onde um modelo de mistura de ordem superior é ajustado aos dados e diferentes misturas das funções nucleares ajustadas pelo algoritmo EM são otimizadas para descrever a f.d.p. de cada cluster (i.e. a f.d.p. de cada cluster é formada por diferentes modelos de mistura de ordem alta, permitindo que a ordem do modelo seja mais alta e, portanto, mais genérica que o número de clusters nos dados.

Para o caso de GMM, gk em Eq. 3-1 é substituído por $\phi(\mathbf{y}_i, \boldsymbol{\mu}_k, \mathbf{R}_k)$, que denota uma f.d.p. normal multivariada com média μk e matriz de covariância Rk. A Eq. 3-1 pode ser reescrita como Eq. 3-2:

$$p(\mathbf{y}_i) = \sum_{k=1}^{K} w_k \phi(\mathbf{y}_i, \boldsymbol{\mu}_k, \mathbf{R}_k) \qquad \text{Eq. 3-2}$$

onde: $\boldsymbol{\Psi} = [w_1, w_2, ..., w_K, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_K, \mathbf{R}_1, \mathbf{R}_2, ..., \mathbf{R}_K]$

3.1.1.1. Algoritmo EM

A estimação por máxima verossimilhança consiste de encontrar-se uma estimativa $\boldsymbol{\Psi}*$ para $\boldsymbol{\Psi}$ que maximize a verossimilhança de y para um conjunto de observações $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n]^T$. Assumindo-se que $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n$ sejam realizações independentes do vetor característico Y, a função de verossimilhança logarítmica como função de $\boldsymbol{\Psi}$ é dada por Eq. 3-3:

$$\log L(\boldsymbol{\Psi}) = \sum_{j=1}^{n} \log\left(\sum_{k=1}^{K} w_k \phi(\mathbf{y}_j, \boldsymbol{\mu}_k, \mathbf{R}_k) \right) \qquad \text{Eq. 3-3}$$

Uma estimativa de $\boldsymbol{\Psi}$ com máxima verossimilhança é dada pelas raízes da Eq. 3-3, que corresponde a um máximo local de Eq. 3-4:

$$\frac{\partial \log L(\boldsymbol{\Psi})}{\partial \boldsymbol{\Psi}} = 0 \qquad \text{Eq. 3-4}$$

Dado que é difícil otimizar $\boldsymbol{\Psi}$ diretamente, são introduzidas variáveis ocultas (não observadas) zjk, onde zjk é definido como 1 ou 0 se yj é proveniente ou não do k-ésimo componente do modelo de mistura (j =1 ,..., n; k =1 ,..., K). O vetor de dados completo (não observado) X é formado por $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]^T$, onde $\mathbf{z}_j = [z_{j1}, z_{j2}, ..., z_{jK}]^T$ são os vetores de variáveis ocultas para uma realização yj com $\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_n$ sendo realizações independentes de uma distribuição multinomial consistindo de um experimento em K categorias, com respectivas probabilidades w1,...,wK. As realizações $\mathbf{x}_1 = (\mathbf{y}_1^T, \mathbf{z}_1^T)^T, ..., \mathbf{x}_K = (\mathbf{y}_K^T, \mathbf{z}_K^T)^T$ são consideradas independentes e identicamente distribuídas.

Para esta especificação, a função de verossimilhança logarítmica para o vetor complete X é dada por:

$$\log L_c(\boldsymbol{\Psi}) = \sum_{k=0}^{K}\sum_{j=1}^{n} z_{jk} \log\{w_k \phi(\mathbf{y}_j, \boldsymbol{\mu}_k, \mathbf{R}_k)\} \qquad \text{Eq. 3-5}$$

O algoritmo EM [20] é efetivo quando maximizar a verossimilhança do vetor de dados completes (X) é mais simples que maximizar a verossimilhança dos dados incompletos (Eq. 3-3). O algoritmo EM é executado iterativamente e consiste de dois passos em cada iteração: passo E (estimação) e passo M (maximização). Considerando $\boldsymbol{\Psi}^i$ como uma estimativa de $\boldsymbol{\Psi}$ na i-ésima iteração, o passo E requer o cálculo de Eq. 3-6:

$$Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^i) = E_{\boldsymbol{\Psi}^i}(\log L_c(\boldsymbol{\Psi}) \mid \mathbf{Y}) \qquad \text{Eq. 3-6}$$

onde: $Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^i)$ é o valor esperado condicional de $\log L_c(\boldsymbol{\Psi})$, dado os dados observados Y e o ajuste atual $\boldsymbol{\Psi}^i$ para $\boldsymbol{\Psi}$.

Uma vez que $\log L_c(\boldsymbol{\Psi})$ é uma função linear das variáveis ocultas zjk, o passo E é executado simplesmente substituindo-se zjk por seu valor esperado condicional, dado $\mathbf{y}_j$, usando-se $\boldsymbol{\Psi}^i$ para $\boldsymbol{\Psi}$. Isto é, zjk é substituído em Eq. 3-6 por Eq. 3-7:

$$\tau_k(\mathbf{y}_j; \boldsymbol{\Psi}^i) = E_{\boldsymbol{\Psi}^i}(z_{jk} \mid \mathbf{y}_j) = \frac{w_k^i \phi(\mathbf{y}_j, \boldsymbol{\mu}_k^i, \mathbf{R}_k^i)}{\sum_{k'=1}^{K} w_{k'}^i \phi(\mathbf{y}_j, \boldsymbol{\mu}_{k'}^i, \mathbf{R}_{k'}^i)} \qquad \text{Eq. 3-7}$$

Pode-se reconhecer $\tau_k(\mathbf{y}_j;\boldsymbol{\Psi}^i)$ na Eq. 3-7 como a estimativa corrente da probabilidade posterior da j-ésima realização ($\mathbf{y}_j$) ter vindo do k-ésimo grupo, i.e. p(k | $\mathbf{y}_j$). A Eq. 3-7 pode ser, então, reescrita como:

$$p(k \mid \mathbf{y}_i) = \frac{w_k^i \phi(\mathbf{y}_i, \boldsymbol{\mu}_k^i, \mathbf{R}_k^i)}{\sum_{k'=1}^{K} w_{k'}^i \phi(\mathbf{y}_i, \boldsymbol{\mu}_{k'}^i, \mathbf{R}_{k'}^i)} = \frac{p(k)\, p(\mathbf{y}_i \mid k)}{p(\mathbf{y}_i)} \qquad \text{Eq. 3-8}$$

Esta equação é una expressão do teorema de Bayes, reconhecendo-se que a estimativa da probabilidade a priori de cada cluster (p(k)) é dada pela estimativa corrente do fator de ponderação w_k^i.

Substituindo-se Eq. 3-8 em Eq. 3-7, obtêm-se a expressão para o passo E:

$$Q(\boldsymbol{\Psi};\boldsymbol{\Psi}^i) = \left(\sum_{j=1}^{n} \log \sum_{k=1}^{K} w_k^i \phi(\mathbf{y}_j, \boldsymbol{\mu}_k^i, \mathbf{R}_k^i) \right) / n = \left(\sum_{j=1}^{n} \log(p(\mathbf{y}_j)) \right) / n \qquad \text{Eq. 3-9}$$

No passo M na (i +1)-ésima iteração, o objetivo é escolher $\boldsymbol{\Psi}^{i+1}$ que maximize $Q(\boldsymbol{\Psi};\boldsymbol{\Psi}^i)$. Assim, o ajuste atual para as proporções de mistura (w_k^{i+1}), os componentes de média ($\boldsymbol{\mu}_k^{i+1}$) e as matrizes de covariância ($\mathbf{R}_k^{i+1}$) são dadas explicitamente por Eq. 3-10:

$$w_k^{i+1} = \sum_{i=1}^{n} p(k \mid \mathbf{y}_i) / n \qquad \text{Eq. 3-10}$$

$$\boldsymbol{\mu}_k^{i+1} = \sum_{i=1}^{n} p(k \mid \mathbf{y}_i)\mathbf{y}_i \,/ \sum_{i=1}^{n} p(k \mid \mathbf{y}_i)$$

$$\mathbf{R}_k^{i+1} = \sum_{i=1}^{n} p(k \mid \mathbf{y}_i)(\mathbf{y}_i - \boldsymbol{\mu}_k^{i+1})(\mathbf{y}_i - \boldsymbol{\mu}_k^{i+1})^T \,/ \sum_{i=1}^{n} p(k \mid \mathbf{y}_i)$$

Uma característica interessante do algoritmo EM é que a verossimilhança da mistura L ($\boldsymbol{\Psi}$) não pode nunca decrescer após uma seqüência EM. Assim, $L(\boldsymbol{\Psi}^{i+1}) \geq L(\boldsymbol{\Psi}^{i})$, implicando na convergência de L ($\boldsymbol{\Psi}$) para um certo valor L*, se a seqüência de valores para a verossimilhança for limitada. Os passos E e M são alternados repetidamente até a verossimilhança ou as estimativas para os parâmetros mudem de um valor arbitrariamente pequeno, indicando a convergência do algoritmo.

O algoritmo EM pode ser sintetizado da seguinte forma:

Algoritmo 2 – EM

1: Inicia-se $\boldsymbol{\Psi}^0$ com valores aleatórios para $w_1^0, w_2^0, ..., w_K^0, \boldsymbol{\mu}_1^0, \boldsymbol{\mu}_2^0, ..., \boldsymbol{\mu}_K^0, \mathbf{R}_1^0, \mathbf{R}_2^0, ..., \mathbf{R}_K^0$

2: Para i = 0, calcula-se Li conforme a Eq. 3-9.

3: Para i =i+1, calcula-se $\boldsymbol{\Psi}^{i+1}$ (Eq. 3-10) e Li+1 (Eq. 3-9)

4: Se Li+1 - Li > δ (constante de convergência), repete-se 3

5: Atualiza-se os valores reais dos parâmetros $\boldsymbol{\Psi}^* = \boldsymbol{\Psi}^i$.

3.1.1.2. Principais problemas na aplicação do algoritmo EM e soluções propostas

O primeiro problema na aplicação do algoritmo EM, conforme descrito na última seção, está relacionado com o fato da função de verossimilhança ter, em geral,

múltiplos máximos locais. Assim, diferentes iniciações podem levar a diferentes modelos ajustados, correspondentes a máximos distintos da função. Isto é especialmente crítico no caso de iniciações aleatórias, uma vez que qualquer máximo local pode ser atingido resultando em ajustes sub-ótimos ou mesmo inadequados. Para tratar deste problema, diversos procedimentos de iniciação são propostos [39,65,17]. Entre eles, uma solução imediata consiste em se fazer um número fixo (Cmax) de iniciações aleatórias para cada aplicação do algoritmo EM, e utilizar aquela que resulta em um máximo valor para o passo E, após a convergência. A utilização de uma pré-clusterização [39,65] também pode ser usada (o que provê valores estimados para as probabilidades, a priori, de cada cluster).

Em adição, no caso de componentes das matrizes de covariância não terem restrições (i.e. considera-se que esta matriz é uma matriz cheia e não uma matriz diagonal, por exemplo), a função de verossimilhança não é limitada, dado que cada ponto de dados acarreta em uma singularidade no vértice do espaço de parâmetros [39]. Além disso, cuidados especiais devem ser tomados para os casos onde um componente (cluster) ajustado possui uma variância generalizada (i.e. o determinante da matriz de covariâncias) muito pequena (não nula), o que acarreta em valores relativamente grandes para o máximo local. Este componente corresponde a um cluster contendo poucos pontos, que estão relativamente próximos uns dos outros. Existe, portanto, uma necessidade de se monitorar o tamanho relativo das proporções das misturas ajustadas, os componentes das variâncias generalizadas com objetivo de eliminar estes máximos locais espúrios. Existe também uma necessidade de se monitorar as distâncias euclidianas entre as médias de componentes ajustados, com objetivo de verificar se os clusters implicados representam uma separação real entre as médias ou se trata-se de um ou mais clusters que caíram quase em um sub-espaço do vetor característico original [39].

Dadas as adaptações e cuidados mencionados nos parágrafos anteriores, uma versão modificada do algoritmo EM é sumarizada a seguir:

Algoritmo 3 – EM Modificado

1: Inicia-se um contador c = 0

2: c = c+1

3: Inicia-se $\boldsymbol{\Psi}^0$ com valores aleatórios para $w_1^0, w_2^0, ..., w_K^0, \boldsymbol{\mu}_1^0, \boldsymbol{\mu}_2^0, ..., \boldsymbol{\mu}_K^0, \mathbf{R}_1^0, \mathbf{R}_2^0, ..., \mathbf{R}_K^0$

4: Para i = 0, calcula-se Li conforme a Eq. 3-9.

5: Para i =i+1, calcula-se $\boldsymbol{\Psi}^{i+1}$ (Eq. 3-10) e Li+1 (Eq. 3-9)

6: Se Li+1 - Li > δ (constante de convergência), repete-se 5

7: Se o determinante de qualquer uma das matrizes de covariâncias < ε (uma constante pequena), repete-se 2

8: Se (c = 1) ou (Li > Lopt) então faz-se Lopt = Li e $\boldsymbol{\Psi}$ opt = $\boldsymbol{\Psi}^i$

9: Se c <= Cmax, repete-se 2

10: Atualiza-se os valores reais dos parâmetros $\boldsymbol{\Psi}$ * = $\boldsymbol{\Psi}$ opt.

3.1.1.3. Estimação automática da ordem ótima do modelo

Para os propósitos do algoritmo EM, a ordem do modelo (K) deve ser assumida a priori. Considerando, no entanto, que, em muitos casos, o número de partições não é conhecido a priori, é útil que se tenha um mecanismo para se descobrir o número de partições mais provável, para um dado modelo. O objetivo aqui consiste em se construir uma estimativa para K que implique em uma "partição ideal", isto é, p(k | yi) é próximo da unidade para um valor de k e próximo de zero, para todos os outros valores, para cada realização yi. Como descrito em [48], esta partição ideal deve ser obtida pela minimização da entropia de Shannon dado os dados observados (Y), que deve ser avaliada para cada observação como Eq. 3-11:

$$H_K = -\sum_{k=1}^{K} p(k \mid \mathbf{y}_i) \log(p(k \mid \mathbf{y}_i)) \qquad \text{Eq. 3-11}$$

O valor esperado desta entropia é avaliado tirando-se a média de HK sobre todos os dados observados Eq. 3-12:

$$E^*(H_K) = -\sum_{i=1}^{n} \sum_{k=1}^{K} p(k \mid \mathbf{y}_i) \log(p(k \mid \mathbf{y}_i)) / n \qquad \text{Eq. 3-12}$$

onde: E* denota o estimador da esperança.

Então, procede-se ao ajuste de Kmax modelos com ordens diferentes (K = 1, 2, ..., Kmax) e avalia-se a entropia esperada (Eq. 3-12) para cada um deles. O modelo que resultar em uma medida mínima é considerado como o modelo ótimo.

O algoritmo EM com estimação automática de ordem ótima é sumarizado a seguir:

Algoritmo 4 – EM com Estimação de Ordem Ótima

1: Inicia-se K = 0, Hopt = 0, Kopt = 1

2: K = K+1

3: Ajusta-se o modelo de ordem K aos dados Y, usando-se o algoritmo EM modificado (algoritmo 3).

4: Estima-se a esperança de H_K (Eq. 3-12)

5: Se (K = 1) ou (H_K < Hopt) então faz-se Hopt = H_K ; Kopt = K; e $\boldsymbol{\Psi}$ opt = $\boldsymbol{\Psi}$ *

6: Se K < Kmax (uma constante fixa), repete-se 2

7: Atualiza-se a ordem real do modelo com o valor ótimo: K = Kopt

8: Atualiza-se os valores reais dos parâmetros $\boldsymbol{\Psi}^* = \boldsymbol{\Psi}_{opt}$.

3.1.1.4. Algoritmo de detecção

Durante a fase de detecção, o modelo de comportamento já está computado e disponível para a realização de inferências sobre nodos dados apresentados ao sistema. O objetivo consiste em definir alguma penalidade λ (e.g. $0 \leq \lambda \leq 1$), indicando o grau de normalidade de uma realização de certamente anômalo ($\lambda = 0$) a certamente normal ($\lambda = 1$).

Muitas abordagens diferentes para definir este critério para o modelo estatístico de comportamento representado pela Eq. 3-1 são possíveis. Neste trabalho, define-se um procedimento de detecção formado por duas etapas: uma classificação (Bayesiana) e uma inferência acerca da pertinência em um determinado cluster.

A classificação é direta para modelos de mistura parametrizados e consiste da avaliação das probabilidades posteriores de cada cluster condicionadas ao novo dado y', isto é, p(k | y') (Eq. 3-8) para k = (1, 2,..., K).

A inferência acerca da pertinência a um cluster específico é um pouco mais complexa. Todas as funções nucleares de distribuição usadas em nosso modelo são contínuas por natureza. Assim, considerar-se a probabilidade posterior do novo dado, condicionada à probabilidade do cluster p (y' | k) pela simples avaliação da f.d.p. no novo ponto não tem significado prático. Uma abordagem mais realista consiste em avaliar a probabilidade do novo dado estar contido em algum intervalo de pertinência (Π_k), definido como uma função da nova observação y' e dos parâmetros da distribuição do cluster (e.g. μk e $\mathbf{R}$ k), o que pode ser formalmente expresso como Eq. 3-13:

$$p(\mathbf{y}' \in \Pi_k \mid k) = \int_{\Pi_k} g_k(\mathbf{y}, \boldsymbol{\theta}_k) d\Pi_k \qquad \text{Eq. 3-13}$$

De fato, a probabilidade definida na Eq. 3-13 se parece com uma função de distribuição acumulativa (f.d.a), se Π_k for definido conforme mostrado na Eq. 3-14 abaixo [30]:

$$\Pi_k = \left\{ \mathbf{y} \in \Re^d \mid \frac{\left\| (\mathbf{y} - \boldsymbol{\mu}_k) \right\|^2}{\left\| \mathbf{R}_k \right\|} \geq \gamma^2 \right\} \qquad \text{Eq. 3-14}$$

onde: $\| \ \|^2$ e $\| \ \|$ denotam algum tipo de operadores para calculo da norma e γ é uma constante que depende de y'.

Finalmente, a função de penalidade para a detecção pode ser definida, conforme a Eq. 3-15:

$$\lambda(\mathbf{y}') = \sum_{k=1}^{K} p(k \mid \mathbf{y}') p(\mathbf{y}' \in \Pi_k \mid k) \qquad \text{Eq. 3-15}$$

3.1.1.5. Algoritmo de detecção para operação em tempo-real com GMM

O procedimento para a construção do modelo de comportamento de referência é usualmente executado off-line. Restrições acerca da complexidade computacional não são severas neste estágio. Entretanto, é usualmente desejável que os estágios de detecção e atualização do modelo de comportamento possam ser executados continuamente. Assim, os algoritmos para detecção e atualização do modelo devem ser projetados para operação em tempo-real. Nesta seção, mostra-se como o processo de detecção pode ser computado, em tempo real.

A Eq. 3-13 não pode ser sempre avaliada analiticamente. Uma solução geral seria avaliar esta equação integral numericamente, mas isto pode ser proibitivo, pois tal

avaliação numérica é computacionalmente intensiva mesmo nos casos unidimensional ou bidimensional, fazendo a execução em tempo-real difícil ou mesmo impossível [23]. De fato, a Eq. 3-13 pode ser difícil para funções nucleares arbitrárias gk, um algoritmo computacionalmente eficiente para avaliação desta integral por ser estabelecido no caso especial de distribuições Gaussianas. Assim, quando utiliza-se um GMM, a avaliação da Eq. 3-13 pode ser feita escolhendo-se convenientemente os elementos indefinidos desta equação, isto é, o operador de norma e γ. Define-se, portanto, Π_k como o espaço complementar (côncavo) da elipsóide de isodensidade (em ℜd), cuja fronteira contém y' e tem como centro de gravidade μk. Isso significa que Π_k é limitado internamente por uma superfície elipsoidal d-dimensional, formada por todos os pontos que possuem o mesmo valor de densidade que y' (i.e. $\phi(\mathbf{y},\boldsymbol{\mu}_k,\mathbf{R}_k)=\phi(\mathbf{y}',\boldsymbol{\mu}_k,\mathbf{R}_k)$). Assim, reescrevendo-se a Eq. 3-14, tem-se a Eq. 3-16:

$$\Pi_k=\left\{\mathbf{y}\in\Re^d \mid \sum_{\alpha\beta}(y_\alpha-\mu_\alpha)\left[R_k^{-1}\right]_{\alpha\beta}(y_\beta-\mu_\beta)\geq\gamma^2\right\} \qquad \text{Eq. 3-16}$$

onde: $\mathbf{y}=(y_1,y_2,...,y_d)^T$; $\boldsymbol{\mu}=(\mu_1,\mu_2,...,\mu_d)^T$; $\left[R_k^{-1}\right]_{\alpha\beta}$ é o elemento da α-ésima linha e da β-ésima coluna da matriz de covariância inversa, e γ é dada pela Eq. 3-17:

$$\gamma^2=\sum_{\alpha\beta}(y'_\alpha-\mu_\alpha)\left[R_k^{-1}\right]_{\alpha\beta}(y'_\beta-\mu_\beta) \qquad \text{Eq. 3-17}$$

Essa estratégia pode ser ilustrada para os espaços uni e bidimensionais, conforme mostrados nas Figura 3-1 e Figura 3-2, respectivamente. Esta última foi desenhada para uma distribuição Gaussiana bivariada, com matriz de covariância diagonal (não correlacionada).

Este procedimento pode ser usado inclusive no cada de distribuições Gaussianas multivariadas, com matriz de covariância sem restrições, dado que é sempre possível encontrar uma transformação linear que mapeie uma distribuição Gaussiana

multivariada qualquer em uma distribuição Gaussiana descorrelacionada (matriz de covariância diagonal) com o mesmo valor de γ.

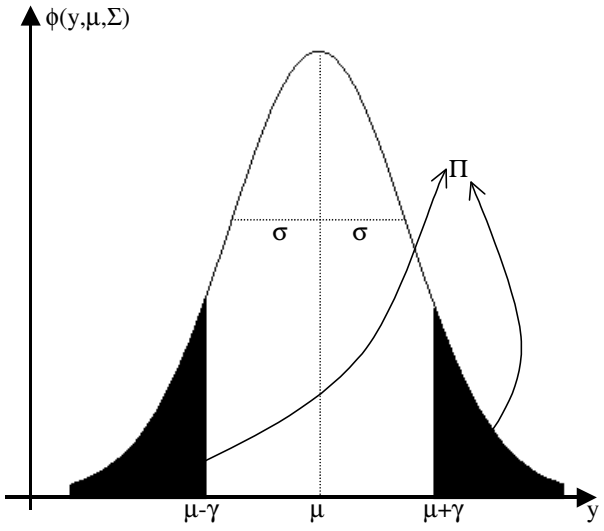

Figura 3-1 - Π para um cluster com distribuição Gaussiana unidimensional
(Fonte: R. Puttini [1] p. 127)

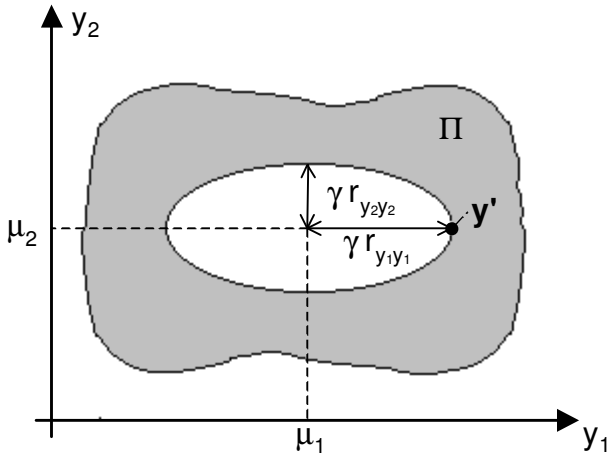

Figura 3-2 - Π para um cluster com distribuição Gaussiana bivariada e matriz de covariância diagonal
(Fonte: R. Puttini [1] p. 127)

Como os dados observados podem pertencer a um espaço multidimensional ($\Re d$), uma distância generalizada γ' é definida na Eq. 3-18. Isto possibilita a normalização das probabilidades expressas na Eq. 3-13 para dados pertencentes a espaços dimensionais diferentes, o que permite a redução da computação para o espaço unidimensional, o que pode ser realizado por um procedimento simples de lookup em tabela.

$$\gamma' = \gamma / \sqrt{d} \qquad \text{Eq. 3-18}$$

3.1.1.6. Atualização recursiva dos parâmetros ajustados do modelo

Como o comportamento na utilização dos sistemas de informação muda constantemente, o modelo de comportamento de referência deve também ser atualizado para evitar falsos positivos. A atualização deve ser considerada como uma adaptação do modelo original com objetivo de acomodar variações suaves no comportamento do sistema, dado que o modelo pode tornar-se inválido ou incompleto no caso de mudanças expressivas.

Na abordagem proposta neste trabalho, considera-se a possibilidade de se atualizar o modelo de comportamento através da atualização recursiva e contínua dos parâmetros do modelo. Assim, a atualização é realizada nas probabilidades dos clusters (wk) e nos parâmetros das distribuições nucleares. Utiliza-se estimadores usuais para a contínua estimação destas estatísticas do modelo [53]. É importante salientar que tanto a verossimilhança logarítmica quanto a entropia podem ser igualmente estimadas e comparadas com valores prévios (e.g. valores obtidos após a fase de treinamento), dado que esses valores podem fornecer uma idéia de "quão bom" está o novo modelo, quando comparado com o modelo de referência. Os estimadores para a atualização recursiva e contínua das probabilidades, a priori, de clusters e dos momentos de primeira e segunda ordem da distribuição são mostrados na Eq. 3-19.

$$w^{i+1} = w^i(1-\eta_1) + \eta_1 p(z_k \mid \mathbf{y}_i) \qquad \text{Eq. 3-19}$$

$$\boldsymbol{\mu}_k^{i+1} = \boldsymbol{\mu}_k^i(1-\eta_2 p(z_k)^{i+1}) + \eta_2 p(z_k)^{i+1}\mathbf{y}_i$$

$$\mathbf{R}_k^{i+1} = \mathbf{R}_k^i(1-\eta_3 p(z_k)^{i+1}) + \eta_3 p(z_k)^{i+1}(\mathbf{y}_i - \boldsymbol{\mu}_k^{i+1})(\mathbf{y}_i - \boldsymbol{\mu}_k^{i+1})^T$$

Estes estimadores são aplicáveis apenas nos casos onde as mudanças do comportamento acontecem a longo termo e o sistema, aplicação e/ou usuários mantêm-se estáveis. As constantes η1, η2 e η3 devem ser cuidadosamente escolhidas para evitar instabilidades ((1/n pode ser uma primeira escolha para η1 e valores ainda menores devem ser usados para η2 e η3, pois mudanças nos momentos da distribuição têm mais energia que mudanças na probabilidades a priori de clusters).

3.1.2. Caracterização de Tráfego Normal em uma Manet e Construção do Modelo de Comportamento Normal

Deseja-se construir um modelo de comportamento para caracterizar as condições normais de tráfego em uma Manet. De uma maneira geral, não há um consenso sobre qual seria um perfil de tráfego que possa ser considerado típico em Manet. De fato, com exceção de alguns protocolos de controle e sinalização da rede (e.g. roteamento) que estão presentes em quase todas as Manets, é provável que cada rede desse tipo tenha um perfil de tráfego que seja dependente da aplicação para a qual a rede foi projetada. Assim sendo, a caracterização do que seria um tráfego normal para uma Manet deve ser realizada caso a caso, ajustando-se o modelo de comportamento normal a uma situação específica, referente a uma aplicação definida da rede.

Outro aspecto que merece destaque consiste no fato de ser difícil se obter amostras reais do tráfego de uma Manet em operação, que sejam comprovadamente livres de traços de possíveis intrusões. Assim, uma alternativa que tem sido muito adotada em trabalhos similares acerca de Manet consiste na realização de simulações.

Uma grande vantagem desse tipo de abordagem consiste em se simular diversos fatores tais como mobilidade e utilização da rede, de forma repetível e controlada. Entretanto, sempre cabe questionar a validade de simulações quando se pretende na realidade a modelagem de ambientes reais, que possuem muitos fatores que não tenham sido considerados de maneira adequada na simulação. Pela dificuldade de se montar uma Manet real, utiliza-se neste trabalho uma caracterização do perfil de tráfego normal derivado de uma simulação. Cabe ressaltar que o processo de treinamento do modelo de comportamento e o processo de detecção de intrusão são exatamente os mesmos casos os dados reais acerca do tráfego na rede estejam disponíveis. Assim, pretende-se validar um processo de detecção de intrusão por comportamento, usando-se dados simulados e, em trabalhos posteriores, aplicar-se este processo a situações mais reais, onde os dados reais de tráfego para treinamento e detecção de intrusão estejam disponíveis.

Por se tratar de uma simulação, três aspectos são definidos para caracterização do tráfego gerado na Manet: tráfego de controle, as aplicações que são executadas nos nodos e o modelo de mobilidade dos nodos:

- Tráfego de controle: consiste basicamente do tráfego gerado pelo protocolo de roteamento (UDP), além do tráfego ARP. Não se considera o tráfego gerado pelo protocolo de autoconfiguração, pois a rede simulada tem um número fixo de nodos, não sendo consideradas novas adesões ou partidas da rede. Também não considera-se tráfego de DNS, pois esta é uma questão ainda aberta em Manet.
- Aplicações executadas nos nodos: para se ter um cenário suficientemente representativo, considera-se a utilização de quatro tipos de tráfego gerado por diferentes aplicações em todos os nodos da rede simulada. São elas: sessão remota simples (e.g. telnet), transferência de dados em rajadas (e.g. FTP), transferência contínua de dados com taxa de bits constante (CBR) (e.g. videoconferência ou áudio-conferência) e aplicação simples de pergunta-resposta assíncrona (e.g. ping). Para cada um desses tipos de tráfego são definidas ainda algumas condições para distribuição do tráfego em toda a rede. Esses parâmetros são ajustados para se ter uma ocupação média dos enlaces sem fio em torno de 20% da sua capacidade.

- Sessão remota simples (telnet)
    - utiliza o TCP;
    - o tráfego gerado é bidirecional;
    - intervalo entre mensagens: processo de Poisson;
    - múltiplas sessões entre origens/destinos diferentes, sendo os nodos de origem e destino (uniformemente distribuído), o tempo de início (processo de Poisson) e a duração da seção (normalmente distribuída) aleatoriamente definidos.
- Transferência de dados em rajada (FTP)
    - utiliza o TCP;
    - tamanho do "arquivo" aleatório (normalmente distribuído);
    - múltiplas transferências entre origens/destinos diferentes, sendo os nodos de origem e destino (uniformemente distribuído) e o tempo de início (processo de Poisson).
- Transferência de dados cbr (videoconferência)
    - utiliza o UDP;
    - taxa cbr fixa de 128kbps;
    - múltiplas transferências entre origens/destinos diferentes, sendo os nodos de origem e destino (uniformemente distribuído), o tempo de início (processo de Poisson) e a duração da seção (normalmente distribuída) aleatoriamente definidos.
- Aplicação simples de pergunta-resposta assíncrona (ping)
    - utiliza o ICMP;
    - sempre envia 4 requisições, espaçadas no tempo de 1 segundo;
    - sempre envia-se resposta;
    - múltiplas transferências entre origens/destinos diferentes, sendo os nodos de origem e destino (uniformemente distribuído) e o tempo de início (processo de Poisson) aleatoriamente definidos.

- Modelo de mobilidade: Atualmente existem duas maneiras de se representar padrões de movimento de usuários de uma rede móvel [5]. Uma forma é através da captura de informações do comportamento real de movimentação do nodo, ou seja, com o uso de registros (traces). A outra maneira é através de modelos de mobilidade,onde tenta-se representar o comportamento de movimentação dos nós sem o uso desses registros de movimentação. A captura dos registros de movimentação possibilita uma observação do comportamento real de movimentação dos nodos, principalmente quando se tem um grande número de nós durante um longo período de observação. Entretanto, em ambientes muito dinâmicos, como em redes ad hoc, capturar esses registros não é uma tarefa fácil. Com isso é necessária a utilização dos modelos de mobilidade. Considera-se aqui a utilização de 4 modelos de mobilidade, Random Waypoint, Gauss-Markov, Manhattan Grid e Reference Point Group Mobility. Utiliza-se, para efeitos de simulação, uma Manet com 10 nodos em uma área de 300m x 300m e um alcance de transmissão de 100m, resultando em uma vizinhança média de 4,6 nodos.
    - Modelo Random Waypoint - Neste modelo, um nó móvel é posicionado, a princípio, em um local escolhido aleatoriamente, segundo uma distribuição uniforme. Este nó móvel permanece no mesmo local por um certo período de tempo (tempo de pausa). Ao fim deste período, o nó escolhe um destino, também de forma aleatória e segundo uma distribuição uniforme, na área de simulação. A velocidade do nó móvel também encontra-se uniformemente distribuída entre: [V_{min}, V_{max}]. O nó móvel então percorre o caminho até o novo destino com a velocidade escolhida. Uma vez que o destino é alcançado, o nó móvel pára por um período de tempo (tempo de pausa) específico antes de reiniciar o processo.

    - Modelo Gauss-Markov - Neste modelo [4], a alteração da posição do nó é governada por uma cadeia de Markov discreta. Cada estado da cadeia representa um valor de deslocamento. Sendo que o estado central representa pausa no deslocamento e os estados da direita e da

esquerda representam deslocamentos com orientação positiva e negativa, respectivamente. No Modelo Gauss-Markov, é atribuído um conjunto de valores de incremento na posição atual, que variará no intervalo [1, n], ou seja, o incremento corresponde à velocidade que o nó usa para mudar da posição atual para a próxima posição. A velocidade é dada por uma progressão geométrica definida no intervalo [1, n]. A cadeia de Markov que rege o comportamento de movimentação neste modelo possui probabilidade m de mudança para os estados à direita e m para os estados à esquerda, consequentemente a probabilidade de permanência no estado é (1 – 2m). A implementação realizada para este modelo parte de uma probabilidade 'm' (probabilidade de incremento da posição do nó para esquerda e para direita) e distribui ('m' para a esquerda e 'm' para a direita) essa probabilidade segundo uma progressão geométrica de ordem 1/2. A cada movimentação do nó, é gerado um valor aleatório que, obedecendo às probabilidades da cadeia de Markov, vai determinar o novo estado ao qual o nó vai pertencer. Se este estiver no segundo estado à direita do zero, por exemplo, o nó vai deslocar-se de dois em dois passos no sentido positivo. Deste modo, o estado da cadeia informa o número de passos que devem ser dados pelo nó, além do sentido de movimentação, e, com este valor, calcula-se a velocidade que ele atinge.

- Modelo Manhattan Grid - Neste modelo, a área geográfica apresenta-se como um conglomerado de blocos de prédios divididos por ruas ou avenidas, com forma tipicamente retangular. Um nó móvel é posicionado, a princípio, em um local escolhido aleatoriamente, em algum ponto de uma das avenidas. Este nó móvel movimenta-se somente pelas avenidas estabelecidas.
- Modelo Reference Point Group Mobility - Esse modelo foi desenvolvido por Hong et al. em [6]. Nele cada grupo tem um centro de referência lógico. O movimento do centro define o comportamento de todo o grupo incluindo localização, velocidade,

direção, aceleração etc. Desse modo, a trajetória do grupo é composta por um caminho até o centro. Os nodos são distribuídos uniformemente dentro da área geográfica do grupo. Para cada nodo, é atribuído um ponto de referência, aos quais, segue o movimento do grupo. Um nodo é colocado aleatoriamente na vizinhança do ponto de referência. O esquema de referência num ponto admite o comportamento de movimentação aleatória independente para cada nodo, além do movimento do grupo.

Um modelo de mistura de gaussianas pode ser ajustado para as condições de tráfego geradas de acordo com as premissas definidas acima, bastando para isso definir as variáveis que são monitoras. Exemplos de variáveis que podem ser monitoradas são: definidas acima, considerando-se as seguintes variáveis monitoradas: taxa de conexões/sessões entrantes, duração de uma sessão, número de pacotes recebidos com erros, etc. Pode-se observar que essas variáveis são correlacionadas de maneira característica para as aplicações consideradas.

3.1.3. Detecção de Ataques de DoS e Scanner de Portas

Diversos tipos de ataques de DoS vem sendo criados nos últimos tempos. De um modo geral, para realização desses ataques, um adversário deve ter comprometido um determinado número de alvos, nos quais são instaladas as ferramentas de geração do ataque. O ataque, propriamente dito, ocorre em uma segunda fase, quando as ferramentas instaladas em todos os alvos disponíveis geram um tráfego excessivo contra um novo alvo particular, inundando-o de pacotes de tráfego espúrio e provocando a indisponibilidade do alvo por sobrecarga na rede e em sua capacidade de processamento. Este trabalho interessa-se em detectar apenas a segunda fase do ataque, isto é, quando diversos nodos estão gerando tráfego espúrio para um mesmo nodo alvo que se deseja tornar indisponível. Um ataque é considerado DDoS, quando este é necessariamente originado de mais de um nodo comprometido.

Um apanhado das principais ferramentas e respectivos ataques de DoS conhecidos pode ser visualizado na Tabela 3-1.

Tabela 3-1 – Caracterização do Tráfego Gerado por Ataques de DoS

Ataque de DoS	Tipo de Tráfego Gerado
Smurf	inundação de pacotes ICMP echo-reply
Trinoo	inundação de datagramas UDP em portas aleatórias
TFN e TFN2K	inundação de pacotes ICMP, UDM e TCP syn (flag syn setado); pacotes errôneos; smurf
TFN2K (ping flood)	inundação pacotes ICMP e smurf
TFN2K Targa 3	Pacotes IP inválidos
stacheldraht v.2.666	inundação de pacotes ICMP, UDP, TCP syn (flag syn setado), TCP null (nenhum flag setado), TCP ack (flag ack setado) e smurf
Shaft	inundação de pacotes ICMP, UDP, TCP syn (flag syn setado)
mstream	inundação de pacotes TCP ack (flag ack setado)

No caso de ataques de scanner de portas, tem-se, de maneira similar, uma geração excessiva de tráfego ilegítimo contra o alvo que está sendo escaneado, com a diferença que este tráfego não é necessariamente originado em múltiplos nodos, como no caso do DDoS. Entretanto, é possível que se tenha um ataque de scanner de portas distribuído, onde as informações acerca do nodo escaneado são coletadas a partir de

nodos distintos. A caracterização do tráfego gerado por ataques de scanner de porta é mostrada na Tabela 3-2.

Tabela 3-2 – Caracterização do Tráfego Gerado por Scanner de Portas

Ataque de Scanner de Portas	Tipo de Tráfego Gerado
Scanner de portas TCP	pedidos sucessivos de abertura de conexões TCP, em portas diferentes
Scanner de portas UDP	datagramas UDP sucessivos, em portas diferentes.

Para que se possa fazer a detecção destes ataques usando-se o modelo de detecção por comportamento proposto nas seções anteriores, são criados modelos de comportamento (e detecção) separados para o tráfego TCP, UDP, ICMP e IP. Conforme mostrado na Tabela 3-3, para cada modelo monitora-se um conjunto de variáveis pertinentes. No que diz respeito às variáveis consideradas para monitoração, utiliza-se variáveis padronizadas da MIB II [14], facilitando a coleta dessas informações, pois tais informações podem estar facilmente disponíveis nos nodos da Manet com a instalação de agentes SNMP padronizados, além de se utilizar um módulo sensor similar ao definido para a detecção de intrusão por uso incorreto.

A Tabela 3-3 mostra ainda quais ataques se pretende detectar utilizando-se um modelo de comportamento normal do sistema do tipo GMM e tendo como dados de referência o tráfego simulado, gerado conforme as premissas da seção anterior. Utiliza-se o mecanismo de detecção por comportamento proposto, com objetivo de discriminar a ocorrência do ataque em relação ao tráfego normal da rede.

Tabela 3-3 – Modelos de Comportamento e Variáveis Monitoradas

Modelo de Comportamento	Variáveis a serem monitoradas	Ataques possivelmente detectados
TCP	- número/taxa de conexões ou entrantes - duração de uma conexão - tcpInErrs - tcpNoPorts	- TFN e TFN2K - stacheldraht - shaft - mstream - scanner de TCP
UDP	- udpInDatagrams - udpInErrs - udpNoPorts	- trinoo - TFN e TFN2K - stacheldraht - shaft - scanner de UDP
ICMP	- icmpInEchos - icmpOutEchos - icmpInErrs*	- smurf - TFN (ping flood) - stacheldraht - shaft
IP	- ipReasmFails	- TFN2K (Targa 3)

Finalmente, pode-se utilizar uma monitoração colaborativa, onde um sensor é definido para escutar promiscuamente o meio de comunicação sem fio e sintetizar informações sobre o tráfego de vizinhos. Entretanto, deve-se destacar que essa abordagem não é muito efetiva no caso de DoS, pois o nodo que monitora na vizinhança do nodo alvo tornar-se-ia igualmente indisponível. Para o caso de ataques de scanner, este método pode ser efetivo, permitindo que os vizinhos detectem ataques contra um nodo alvo.

3.1.4. Caracterização do Modelo de Tráfego dos Ataques

Deseja-se construir um modelo de comportamento para caracterizar as condições de tráfego dos ataques de DoS, DDoS e Scanner em uma Manet. De uma maneira geral, não há um consenso sobre qual seria um perfil de tráfego que possa ser considerado típico de um ataque em Manet. Da mesma forma que na caracterização do Modelo de tráfego normal um aspecto que merece destaque consiste no fato de ser difícil se obter amostras reais do tráfego de uma Manet em operação, que possuam comprovadamente os traços de possíveis intrusões que queremos simular. Assim, a alternativa que será adotada consiste na realização de simulações.

Os mesmos aspectos definidos para a caracterização do tráfego normal na seção 3.1.2, são utilizados aqui também. A caracterização do tráfego com ataque será dado pela caracterização do trêfego normal adicionado do tráfego adicional gerado pelo ataque. Os ataques implementados serão descritos abaixo:

- DoS: Para a geração do ataque de DoS, simula-se a geração de um tráfego UDP CBR (8Mbps) em um nodo de origem escolhido aleatoriamente, em direção a um único nodo de destino.

- DDoS: Para a geração do ataque de DoS, simula-se a geração de um tráfego UDP CBR (4Mbps) em dois nodos de origem escolhido aleatoriamente, em direção a um único nodo de destino.

- Scanner: Para a geração de um ataque de scanner, escolhe-se um par origem-destino aleatoriamente e faz com que esta origem envie pedidos de conexão TCP ao destino, em uma taxa de 10 pedidos por

segundo. No destino, faz-se um dreno que, a cada 30 pedidos de conexões uma é aceita (i.e. indicando um "match" com uma porta que esteja respondendo).

3.1.5. Resposta a Intrusões

No caso de ataques de DoS, a origem dos ataques não pode ser claramente identificada nos pacotes de tráfego espúrio que são gerados, pois estes contém informações errôneas, na maioria do casos. A alternativa de defesa que se apresenta para esse tipo de ataque consiste em se evitar o encaminhamento do tráfego espúrio, o que exige uma colaboração de todas as entidades da rede que encaminham o tráfego desde seus pontos de origem (i.e. os nodos que tenham sido vítimas de comprometimento e possuam as ferramentas do ataque instaladas em seus sistema) até o alvo final (o nodo que sofre o ataque de DoS). No caso de Manet, essa colaboração já existe, por princípio, devido à presença de um IDS local em cada nodo. Pretende-se, como continuação deste trabalho, se investigar a possibilidade de se correlacionar alertas gerados por diversos nodos que estão no caminho dos pacotes de tráfego espúrio, com objetivo de identificar-se o caminho destes pacotes, que poderiam ser filtrados de forma automática, enquanto o ataque durar.

Este método de resposta à intrusão, que não foi desenvolvido no contexto deste trabalho por limitações no tempo da pesquisa, deve ser ainda mais eficaz no caso de ataques de scanner de portas, pois, nestes casos, a origem dos pacotes é verdadeira e pode inclusive ser identificada. Isto possibilitaria um outro tipo de resposta – baseada na revogação do certificado da origem do ataque. A identificação da origem do ataque não aconteceria propriamente pela detecção de intrusão por comportamento proposta, mas sim pelo mecanismo de correlação de alertas que permitiria identificar o caminho dos pacotes espúrios desde sua origem ao seu destino (nodo alvo).

4. EXPERIMENTAÇÃO E RESULTADOS

Para validação do modelo de IDS por comportamento proposto, um modelo de simulação em ambientes de distribuição e topologia mais genéricas é também proposto. Nesta seção será mostrada a implementação deste modelo bem como os resultados e avaliações das simulações realizadas.

4.1. AMBIENTE DE SIMULAÇÃO

A Figura 4-1 ilustra o modelo de geração e processamento de dados de simulação para verificação da aplicabilidade das técnicas de detecção de intrusão por comportamento, apresentadas na seção 3.1, à detecção de ataques contra Manets.

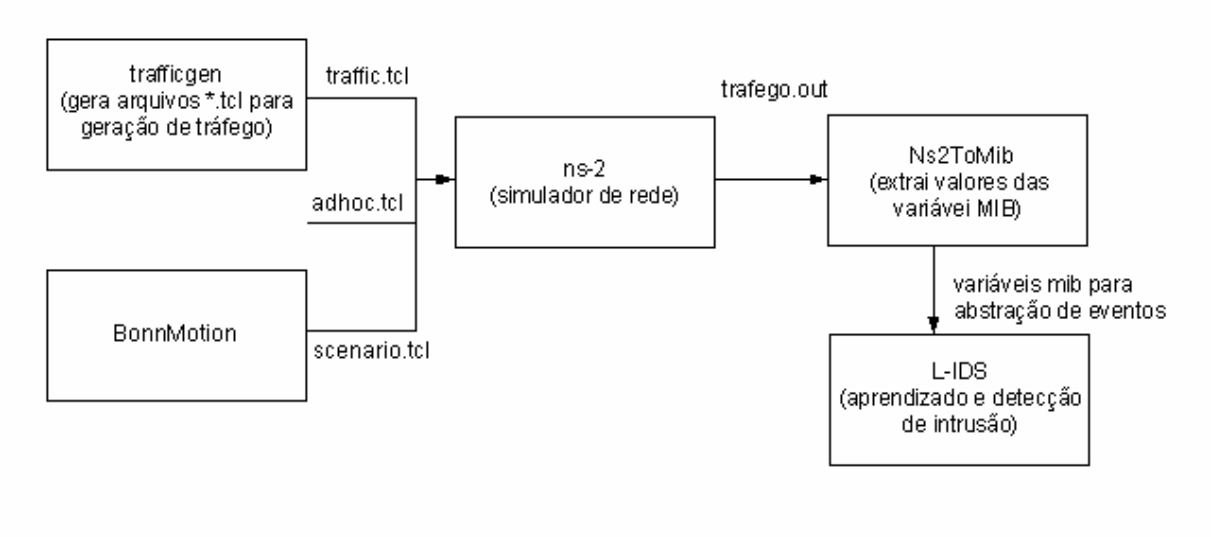

Figura 4-1 – Processo de geração da simulação

4.1.1. Trafficgen

O Trafficgen é um software desenvolvido durante o trabalho de mestrado para geração scripts de trafego baseados em modelos de geração de trafego. Na nossa simulação ele é responsável pela geração do trafego considerado normal de acordo com a descrição feita na seção 3.1.2 para realização do treinamento e pela geração do trafego gerado pelos ataques utilizados para a validação da detecção. Como resultado são gerados scripts no formato TCL (traffic.tcl) que serão interpretados pelo NS-2 juntamente com o script de cenário para gerar a simulação. Por meio de um arquivo de

configuração, podemos escolher quais são exatamente as aplicações que serão utilizadas na simulação e configurar todas as características do trafego gerado pelas aplicações escolhidas. Alem disso é possível também, configurar as características dos ataques que serão realizados na simulação.

4.1.2. BonnMotion

O Bonnmotion é um software desenvolvido para criar cenários de mobilidade baseados em modelos de mobilidade para redes AdHoc. Ele foi desenvolvido pelo grupo de Sistemas de Comunicação no Instituto de Ciência da Computação IV na Universidade de Bonn, Alemanha, onde serve como ferramenta de investigação de características de redes Ad Hoc. Na nossa simulação ele é responsável pela geração dos scripts no formato TCL (scenario.tcl) com os cenários de mobilidade descritos na seção 3.1.2. Os scripts gerados aqui serão interpretados pelo NS-2, juntamente com os scripts de trafego para gerar a simulação. Por meio de um arquivo de configuração, podemos escolher qual o modelo de mobilidade que será utilizada e configurar todas as características do modelo escolhido.

4.1.3. adhoc.tcl

O adhoc.tcl é um script responsável por ajustar as características básicas da simulação, como o número de nodos, o alcance da transmissão, o protocolo de roteamento, o tempo de simulação, entre outros. Além disso, o script é responsável por informar quais são os scripts de cenário e de trafego que serão utilizados da simulação. Assim, o NS-2 deve encontrar nesse script todas as informações necessárias para a realização da simulação.

4.1.4. NS-2

O Network Simulator – 2 (NS-2) é um dos softwares de simulação de redes mais populares. Este software foi desenvolvido na Universidade de Berkeley e hoje é o software mais utilizado no meio cientifico para a simulação de redes. Na nossa simulação o NS é responsável por interpretar os scripts TCL gerados pelo TrafficGen e

pelo BonnMotion e gerar como resultado um arquivo de trace contendo todos os pacotes gerados, encaminhados e recebidos em todos os nodos da rede (trafego.out).

4.1.5. Ns2ToMib

O Ns2ToMib é um software desenvolvido durante o trabalho de mestrado para transformar o arquivo de trace (trafego.out) em arquivos com as informações das variáveis da MIB que serão monitoradas. No entanto, as variáveis MIB devem ser mantidas e monitoradas em cada nodo. Desse modo, esse arquivo é decomposto em vários outros arquivos, um por nodo da rede. Em cada arquivo são colocados apenas os pacotes gerados, recebidos ou encaminhados por este nodo. Assim, este arquivo corresponde a um *dump* de pacotes capturados por um analisador de rede, com a interface de captura em modo não promíscuo. Em seguida, cada um desses arquivos é processado produzindo como saída (arquivos *.mib) uma lista de amostras para os valores das variáveis MIB, amostradas em um intervalo de tempo que pode ser definido por passagem de parâmetros na chamada do comando.

4.1.6. IDS

Finalmente, um módulo do IDS coletor de dados foi desenvolvido a partir de [1], com as modificações para permitir injetar essas informações no IDS. Esse coletor de dados, desenhado para processamento de dados off-line, executa consultas periódicas que retornam, para os instantes de tempo quando a consulta é executada, os valores assumidos pelas variáveis MIB que se encontram armazenados no arquivo *.mib. Dessa forma o IDS gera arquivos de saída para que se possa analisar o desempenho do modelo implementado.

4.2. SIMULAÇÃO E RESULTADOS

Dois modelos de tráfego são analisados mais de perto: TCP e UDP. O uso desses modelos em separado faz com que exista uma discriminação implícita entre todo tráfego UDP e TCP gerados. Assim, o modelo de comportamento usando UDP servirá para modelar apenas a aplicação de videoconferência e o protocolo de roteamento. Já no caso do TCP, modela-se o tráfego gerado pelas aplicações Telnet e FTP.

Para a detecção de intrusão, considera-se normal um evento para o qual o valor de λ (Eq. 3-15) é maior ou igual a media subtraída do desvio padrão de todos os valores de λ para todas as realizações observadas durante a fase de treinamento. Considera-se intrusão com baixa probabilidade um evento para o qual o valor de λ é maior ou igual ao menor valor de λ dentre todos as realizações observadas durante a fase de treinamento. Por fim, considera-se alta probabilidade de intrusão um evento para o qual o valor de λ é menor que o menor valor de λ dentre todos as realizações observadas durante a fase de treinamento.

4.2.1. Modelo UDP

No caso do modelo UDP, apenas as variáveis udpInDatagrams (datagramas UDP que entram em um nodo) e ipForwDatagrams (datagramas IP encaminhados pelo nodo) serão utilizadas. Como essas variáveis são monotonicamente crescentes, defini-se como resultado da abstração de eventos a geração de um evento de aprendizado (realização), cujo valor (udpIn; ipForw) é obtido subtraindo-se do valor atual da consulta periódica corrente o seu valor precedente (consulta periódica antecedente).

Para fazer o treinamento e o ajuste do modelo, todos os eventos de treinamento gerados em todos os nodos são consumidos em um mesmo L-IDS, procedendo-se ao ajuste do GMM aos dados de referência (eventos). Para se evitar singularidades (i.e. formação de um cluster com média zero e variância pequena), os eventos onde udpInDatagrams e ipForwDatagrams forem igual a zero são descartados como normais tanto no processo de aprendizagem quanto no processo de detecção de intrusão.

O resultado dessa etapa são os parâmetros do modelo de misturas, que são colocados em uma mensagem e distribuídos para todos os L-IDS da rede. Nesse instante (recebimento da mensagem) o processo de detecção de intrusão pode começar.

Para uma melhor avaliação do modelo serão realizadas simulações em cenários diferenciados, porem algumas variáveis da simulação serão fixadas. As variáveis fixadas são:

- Quantidade de nodos: 10;
- Vizinhança média: 4,6 nodos;
- Dimensões do cenário: 300 metros x 300 metros;

- Tempo de simulação: 1000 segundos;

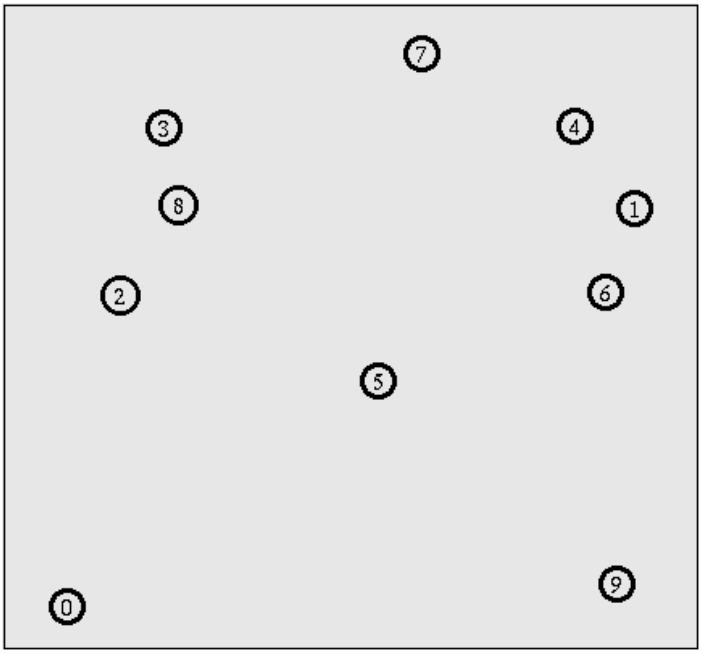

Figura 4-2 – Cenário sem mobilidade

4.2.1.1. Cenário sem movimentação dos nodos

Cenários completamente estáticos são pouco freqüentes nas Manets, já que um grande motivador para a utilização de redes móveis Ad Hoc está exatamente na mobilidade. No entanto cenários com pouca mobilidade podem ser encontrados com alguma freqüência. A escolha desse cenário como o primeiro a ser analisado se deu devido a facilidade de se analisar um cenário estático.

4.2.1.1.1. Treinamento

Com o intuito de encontrar a periodicidade ideal de realização das consultas das variáveis em questão, foram realizadas simulações variando a periodicidade da amostragem. Os valores utilizados para a periodicidade nas simulações foram, 0,1; 0,3; 0,6; 1; 3; 6; 10; 30; 60 e 100 segundos. Desta forma foram realizados os treinamentos

para cada simulação afim de encontrar os respectivos ajustes do GMM. Figuras abaixo ilustram o resultado do treinamento para as simulações com periodicidade de consultas de 0,3; 3 e 30 segundos. Nesses gráficos são mostrados a formação dos clusters e os pontos que representam cada realização.

No que diz respeito ao ajuste do GMM aos dados resultantes das simulações para o modelo UDP, observa-se a formação de dois *clusters* bem definidos para as simulações com periodicidade de amostragem em 1; 3 e 6 segundos. Certamente, em cada treinamento, o cluster com menor media de udpInDatagrams representa modelagem do trafego da aplicação de pergunta e resposta (PING) e o cluster com maior media representa a modelagem do trafego da aplicação de videoconferência (fonte CBR a 128kbps). Obviamente, existe uma contribuição do tráfego do protocolo AODV no valor da média e do desvio desses *clusters*.

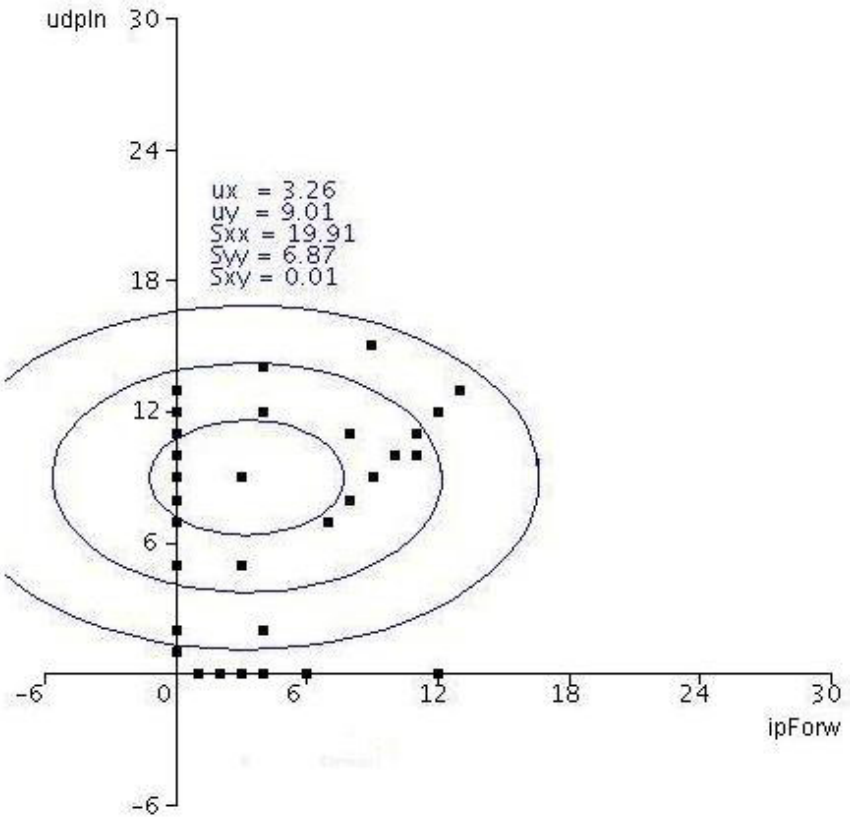

Figura 4-3 - Modelagem com periodicidade de consulta de 0,3 segundos.

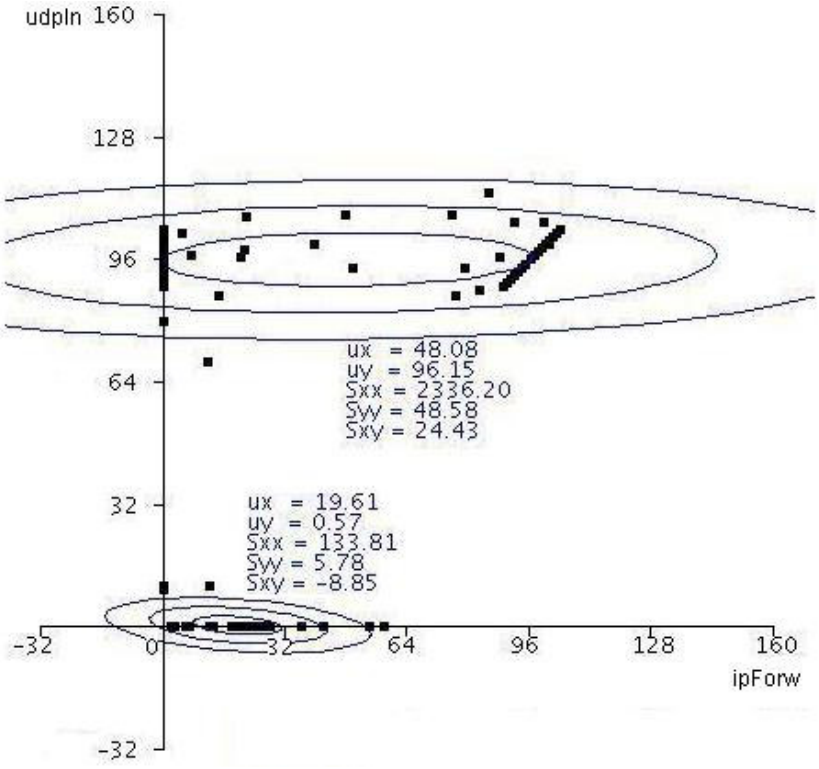

Figura 4-4 - Modelagem com periodicidade de consulta de 3 segundos.

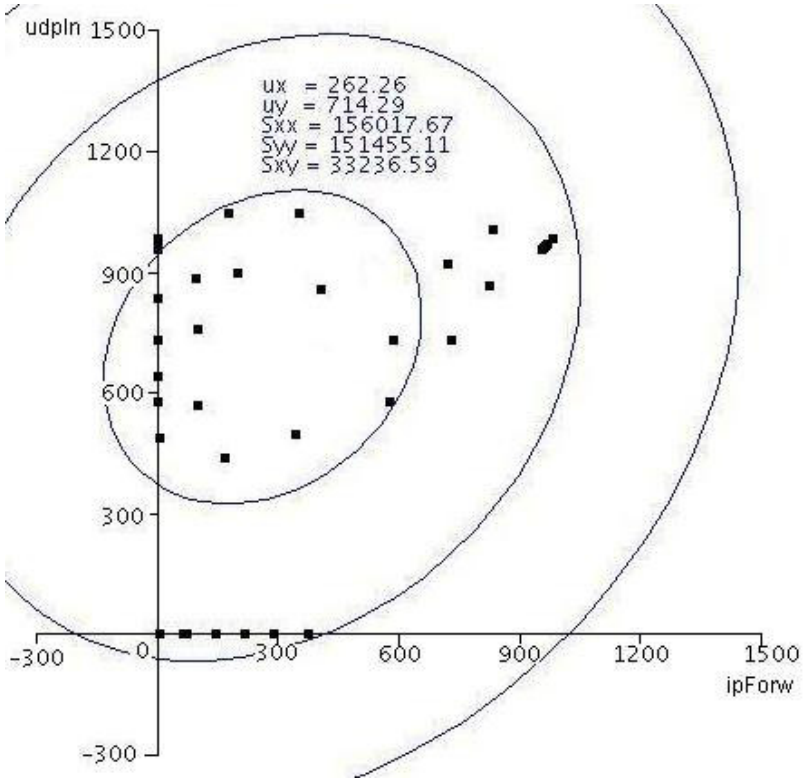

Figura 4-5 - Modelagem com periodicidade de consulta de 30 segundos.

4.2.1.1.2. Detecção

Após a realização dos treinamentos foram realizadas simulações dos dois ataques descritos abaixo e em seguida foram realizadas simulações das analises do trafego gerado com a intenção de verificar a qualidade da detecção da intrusão para cada um dos treinamentos mostrados anteriormente.

4.2.1.1.2.1. Ataque de DoS

Para a geração do ataque de DoS, escolhemos como origem o nodo 0 e como destino o nodo 7 e em seguida realizamos o procedimento descrito na seção 3.1.4. O ataque é ilustrado na Figura 4-6.

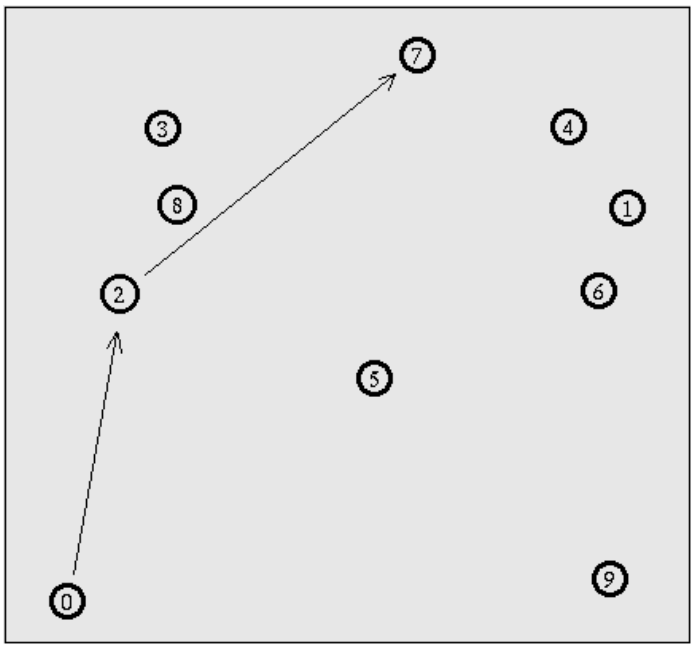

Figura 4-6 – Ataque de DoS(Cenário sem Mobilidade)

Aplicando-se o modelo detecção, são detectadas situações anômalas em todos os nodos que encaminham o tráfego desde a origem até o destino. Na Tabela 4-1 são mostrados os resultados obtidos para cada intervalo de polling em que foi simulada a utilização do IDS para detecção de intrusão com alta probabilidade. O intervalo de polling representa a freqüência em que o IDS consulta as variáveis que estão sendo monitoradas. Na tabela temos os resultados das taxas de falso positivo, falso positivo durante o ataque e falso negativo. Esses resultados consistem em resultados

consolidados de todos os nodos que participam da simulação. Falso positivo representa a detecção de ataque por qualquer nodo em um momento em que não está sendo realizado o ataque. Falso positivo durante o ataque representa a detecção de ataque em nodos que não estão no caminho do ataque durante a realização do ataque. Falso negativo representa a não detecção de ataque em nodos que estão no caminho do ataque durante a realização do ataque.

Tabela 4-1 – Taxas de falso positivo e falso negativo para Ataque de DoS para detecção com alta probabilidade.

Intervalo de Polling	Falso positivo	Falso positivo durante o ataque	Falso negativo
0,1 segundo	2,33%	1,23%	7,33%
0,3 segundo	1,00%	1,47%	7,33%
0,6 segundo	0,67%	0,73%	5,52%
1 segundo	0,17%	1,25%	1,13%
3 segundos	0,00%	0,00%	0,03%
6 segundos	0,50%	0,83%	0,00%
10 segundos	1,25%	1,08%	0,00%
30 segundos	1,75%	1,10%	0,00%
60 segundos	2,47%	2,08%	0,00%
100 segundos	2,67%	2,25%	0,00%

Podemos então verificar que obtivemos as menores taxas de falso positivo, falso positivo durante o ataque e falso negativo com amostragens de 3 em 3 segundos. Esse resultado, bastante satisfatório do ponto de vista de detecção de DoS só é possível graças a análise conjunta de duas variáveis udpInDatagrams e ipForwDatagrams.

Na Tabela 4-2 são mostrados os resultados obtidos para cada intervalo de polling em que foi simulada a utilização do IDS para detecção de intrusão com baixa probabilidade. Na tabela temos os resultados das taxas de falso positivo, falso positivo durante o ataque e falso negativo da mesma forma em que na Tabela 4-1.

Tabela 4-2 – Taxas de falso positivo e falso negativo para Ataque de DoS para detecção com baixa probabilidade.

Intervalo de Polling	Falso positivo	Falso positivo durante o ataque	Falso negativo
0,1 segundo	7,24%	7,83%	0,00%
0,3 segundo	7,36%	7,67%	0,00%
0,6 segundo	7,19%	7,52%	0,00%
1 segundo	5,17%	5,67%	0,00%
3 segundos	4,00%	4,25%	0,00%
6 segundos	6,00%	6,25%	0,00%
10 segundos	7,84%	7,58%	0,00%
30 segundos	7,33%	7,22%	0,00%
60 segundos	7,27%	7,58%	0,00%
100 segundos	8,45%	8,73%	0,00%

Da mesma forma que na detecção com alta probabilidade, podemos verificar que obtivemos as menores taxas de falso positivo, falso positivo durante o ataque e falso negativo com amostragens de 3 em 3 segundos. Analisando os dados da Tabela 4-1 podemos observar que mesmo no melhor resultado com a periodicidade de 3 segundos, temos a taxa de falso positivo maior que 0, mostrando que em alguns poucos casos um ataque não será detectado. Porem analisando a Tabela 4-2 podemos observar que mesmo no melhor resultado com a periodicidade de 3 segundos, tem a taxa de falso negativo maior que 0, mostrando que em alguns casos um ataque detectado não é ataque. Esses resultados, então, justificam a utilização de três estados de detecção como mostrado no inicio da seção 4.2.

4.2.1.1.2.2. Ataque de DDoS

Para a geração do ataque de DDoS, escolhemos como origem nos nodos 0 e 9 e com destino o nodo 7 e por fim, realizamos o procedimento descrito na seção 3.1.4. O ataque é ilustrado na Figura 4-7.

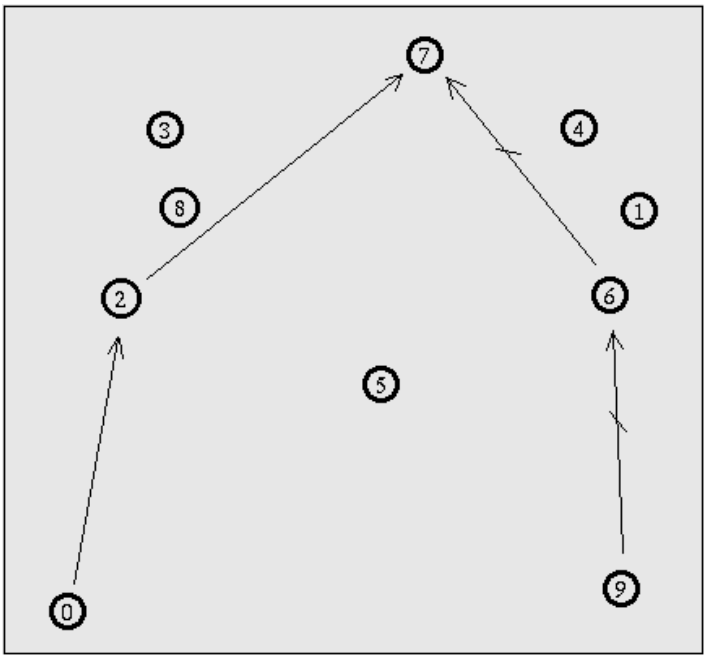

Figura 4-7 – Ataque de DDoS(Cenário sem Mobilidade)

Aplicando-se o modelo detecção, são detectadas situações anômalas em todos os nodos que encaminham o tráfego desde a origem até o destino. Na Tabela 4-3 são mostrados os resultados obtidos para cada intervalo de polling em que foi simulada a utilização do IDS para detecção de intrusão com alta probabilidade. O intervalo de polling representa a freqüência em que o IDS consulta as variáveis que estão sendo monitoradas. Na tabela temos os resultados das taxas de falso positivo, falso positivo durante o ataque e falso negativo. Esses resultados consistem em resultados consolidados de todos os nodos que participam da simulação. Falso positivo representa a detecção de ataque por qualquer nodo em um momento em que não está sendo

realizado o ataque. Falso positivo durante o ataque representa a detecção de ataque em nodos que não estão no caminho do ataque durante a realização do ataque. Falso negativo representa a não detecção de ataque em nodos que estão no caminho do ataque durante a realização do ataque.

Tabela 4-3 – Taxas de falso positivo e falso negativo para Ataque de DDoS para detecção com alta probabilidade.

Intervalo de polling	Falso positivo	Falso positivo durante o ataque	Falso negativo
0,1 segundo	4,37%	4,23%	10,33%
0,3 segundo	3,40%	3,67%	10,57%
0,6 segundo	2,67%	2,73%	7,52%
1 segundo	0,77%	1,18%	1,98%
3 segundos	0,00%	0,00%	0,17%
6 segundos	0,58%	0,87%	0,06%
10 segundos	2,25%	2,08%	0,01%
30 segundos	2,75%	2,10%	0,00%
60 segundos	4,47%	4,08%	0,00%
100 segundos	3,67%	5,25%	0,00%

Podemos então verificar que obtivemos as menores taxas de falso positivo, falso positivo durante o ataque e falso negativo com amostragens de 3 em 3 segundos. Esse resultado, bastante satisfatório do ponto de vista de detecção de DDoS só é possível graças a análise conjunta de duas variáveis udpInDatagrams e ipForwDatagrams.

Na Tabela 4-4 são mostrados os resultados obtidos para cada intervalo de polling em que foi simulada a utilização do IDS para detecção de intrusão com baixa probabilidade. Na tabela temos os resultados das taxas de falso positivo, falso positivo durante o ataque e falso negativo da mesma forma em que na Tabela 4-3.

Tabela 4-4 – Taxas de falso positivo e falso negativo para Ataque de DoS para detecção com baixa probabilidade.

Intervalo de Polling	Falso positivo	Falso positivo durante o ataque	Falso negativo
0,1 segundo	8,10%	8,03%	0,00%
0,3 segundo	7,66%	7,63%	0,00%
0,6 segundo	7,17%	7,32%	0,00%
1 segundo	5,47%	5,62%	0,00%
3 segundos	4,06%	4,65%	0,00%
6 segundos	6,30%	6,27%	0,00%
10 segundos	7,86%	7,88%	0,00%
30 segundos	7,53%	7,62%	0,00%
60 segundos	7,27%	7,58%	0,00%
100 segundos	8,45%	8,73%	0,00%

Da mesma forma que na detecção com alta probabilidade, podemos verificar que obtivemos as menores taxas de falso positivo, falso positivo durante o ataque e falso negativo com amostragens de 3 em 3 segundos. Analisando os dados da Tabela 4-3 podemos observar que mesmo no melhor resultado com a periodicidade de 3 segundos, tem a taxa de falso positivo maior que 0, mostrando que em alguns poucos casos um ataque não será detectado. Porem analisando a Tabela 4-4 podemos observar que mesmo no melhor resultado com a periodicidade de 3 segundos, tem a taxa de falso negativo maior que 0, mostrando que em alguns casos um ataque detectado não é ataque. Esses resultados, então, justificam a utilização de três estados de detecção como mostrado no inicio da seção 4.2.

Outra análise em relação a este ataque de DDoS faz-se ainda necessária: obviamente o nodo que recebe todo o tráfego gerado (de todos os seus vizinhos) vai tornar-se rapidamente indisponível (o próprio ns-2 acusa a geração de vários erros de encaminhamento e descarte de pacotes na vizinhança do nodo de destino). Entretanto, os nodos que estão distantes, apesar de estarem gerando/encaminhando uma quantidade expressiva de dados, não estão necessariamente quebrados com o ataque. Como o

sistema de detecção de intrusão identifica anomalias em todos os nodos do caminho, sugere-se que, caso haja uma interação entre esses nodos intermediários, pode-se bloquear o encaminhamento dos pacotes vindos dessa origem. Esse encaminhamento deve ser bloqueado com base nos endereços de enlace e não nos endereços de destino do datagrama IP, pois esses são facilmente falsificados e, nos ataques de DDoS mais avançados, são constantemente alterados (a cada pacote).

4.2.1.2. Cenário com movimentação dos nodos

Como foi apresentado, cenários com pouca ou nenhuma mobilidade são pouco freqüentes em redes móveis Ad Hoc. Por isso, para a validação do modelo, faz-se necessário a realização de simulações com modelos de mobilidade diferenciados. Para isso foram utilizados os modelos: Random Waypoint, Gauss-Markov, Manhattan Grid e Reference Point Group Mobility. Estes modelos de mobilidade são descritos com maiores detalhes na seção 3.1.2. Para cada um dos modelos de mobilidade, foram realizadas simulações de ataques de DoS e DDoS. Assim como no cenário sem movimentação, aplicou-se o modelo detecção, e foram detectadas situações anômalas em todos os nodos que encaminham o tráfego desde a origem até o destino. Foi possível verificar que as menores taxas de falso positivo, falso positivo durante o ataque e falso negativo foram obtidas com amostragens de 3 em 3 segundos. Assim como no cenário sem movimentação, foram obtidas taxas de erro na detecção dos ataques sempre menores que 0,1% em todos os modelos de mobilidade. Esses resultados validam de forma satisfatória o modelo do IDS, para os cenários dos 4 modelos de mobilidade adotados.

4.2.2. Modelo TCP

No caso do modelo UDP, apenas as variáveis tcpPassiveOpens (número de conexões abertas passivamente no nodo) e tcpInSegs (número de segmentos, inclusive com erro e para abertura de conexão, recebidos) serão utilizadas. Da mesmo modo que no caso do modolo UDP, como essas variáveis são monotonicamente crescentes, defini-se como resultado da abstração de eventos a geração de um evento de aprendizado (realização), cujo valor (tcpPO ; tcpIN) é obtido subtraindo-se do valor atual da consulta periódica corrente o seu valor precedente (consulta periódica antecedente).

Também repetindo o procedimento para o caso UDP, para fazer o treinamento e o ajuste do modelo, todos os eventos de treinamento gerados em todos os nodos são consumidos em um mesmo IDS, procedendo-se ao ajuste do GMM aos dados de referência (eventos). Para se evitar singularidades (i.e. formação de um cluster com média zero e variância pequena), os eventos onde tcpPassiveOpens e tcpInSegs forem igual a zero são descartados como normais tanto no processo de aprendizagem quanto no processo de detecção de intrusão.

O resultado dessa etapa são os parâmetros do modelo de misturas, que são colocados em uma mensagem e distribuídos para todos os IDS da rede. Nesse instante (recebimento da mensagem) o processo de detecção de intrusão pode começar.

Para uma melhor avaliação do modelo serão realizadas simulações em cenários diferenciados, porem algumas variáveis da simulação foram fixadas. As variáveis fixadas são:

- Quantidade de nodos: 10;
- Vizinhança média: 4,6 nodos;
- Dimensões do cenário: 300 metros x 300 metros;
- Tempo de simulação: 1000 segundos;

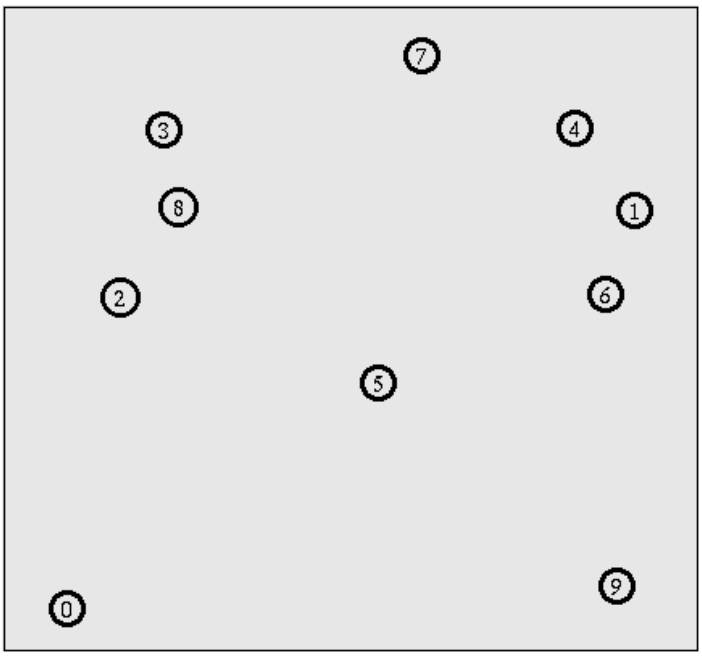

Figura 4-8 – Cenário sem mobilidade

4.2.2.1. Cenário sem movimentação dos nodos

Como foi apresentado na seção 4.2.1.1, cenários completamente estáticos são pouco freqüentes nas Manets, já que um grande motivador para a utilização de redes móveis Ad Hoc está exatamente na mobilidade. No entanto cenários com pouca mobilidade podem ser encontrados com alguma freqüência. A escolha desse cenário como o primeiro a ser analisado se deu devido a facilidade de se analisar um cenário estático.

4.2.2.1.1. Treinamento

Com o intuito de encontrar a periodicidade ideal de realização das consultas das variáveis em questão, foram realizadas simulações variando a periodicidade da amostragem. Os valores utilizados para a periodicidade nas simulações foram, 0,1; 0,3; 0,6; 1; 3; 6; 10; 30; 60 e 100 segundos. Desta forma foram realizados os treinamentos para cada simulação a fim de encontrar os respectivos ajustes do GMM. Os gráficos mostrados nas Figuras abaixo ilustram o resultado do treinamento para as simulações com periodicidade de consultas de 0,3; 3 e 30 segundos. Nesses gráficos são mostrados a formação dos clusters e os pontos que representam cada realização.

No que diz respeito ao ajuste do GMM aos dados resultantes das simulações para o modelo TCP, observa-se a formação de dois *clusters* bem definidos para as simulações com periodicidade de amostragem em 1; 3 e 6 segundos. Certamente, em cada treinamento, os clusters estão modelando respectivamente o trafego das aplicações Telnet e FTP.

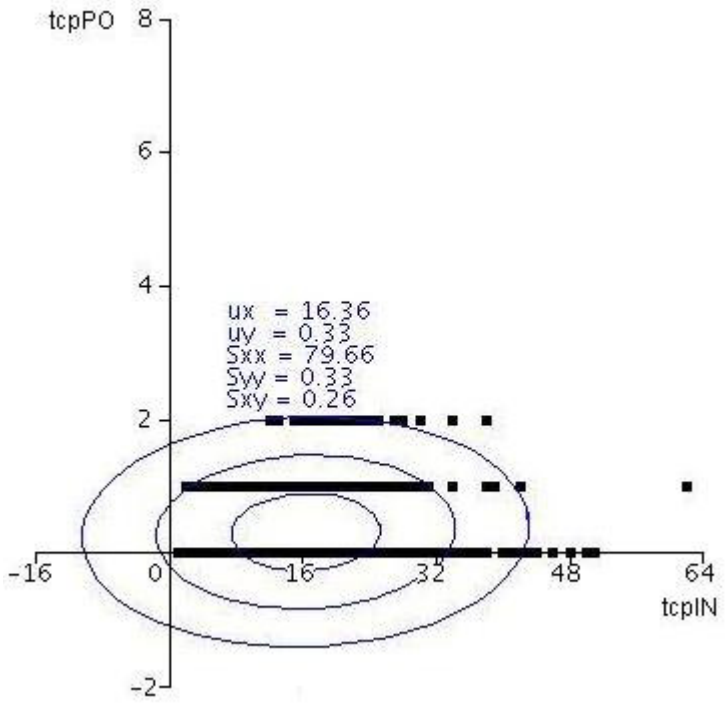

Figura 4-9 – Modelagem com periodicidade de consulta de 0,3 segundos.

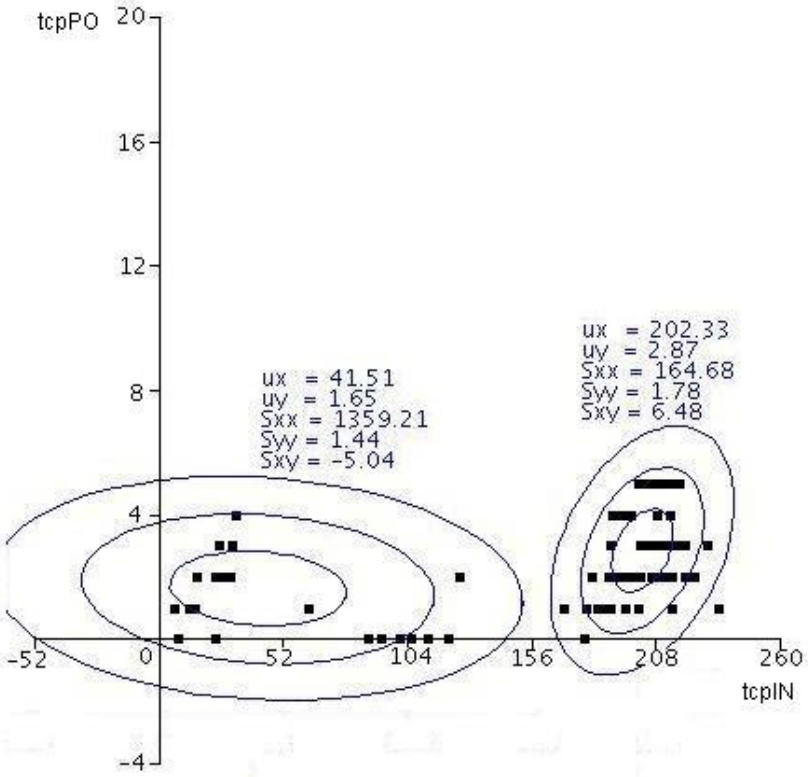

Figura 4-10 - Modelagem com periodicidade de consulta de 3 segundos.

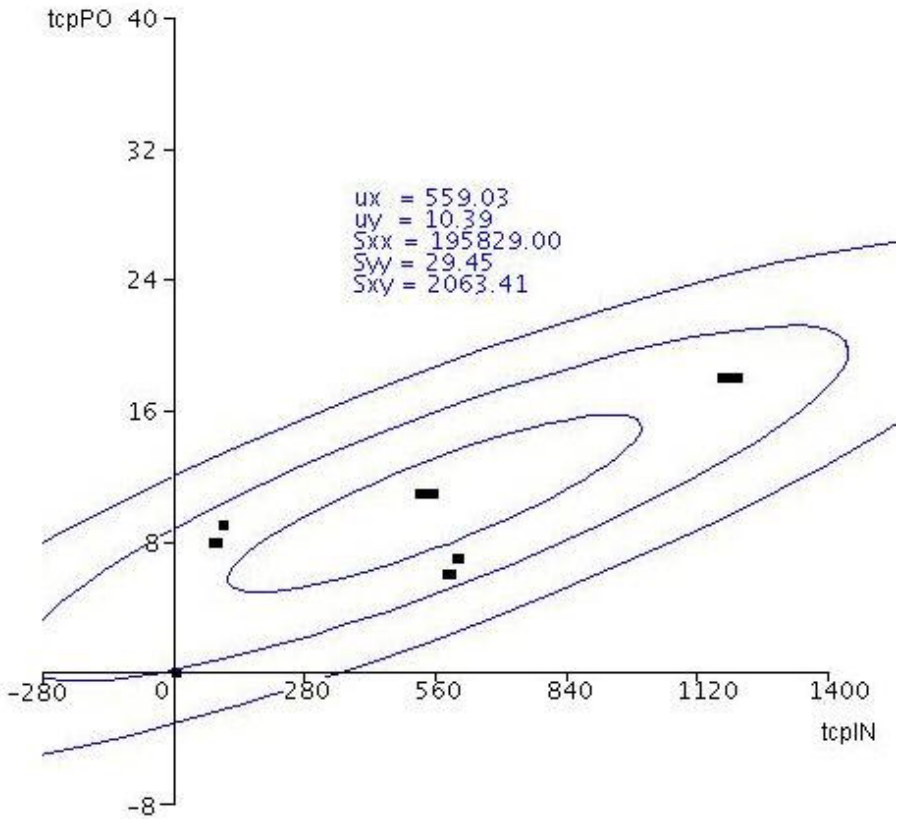

Figura 4-11 - Modelagem com periodicidade de consulta de 30 segundos.

4.2.2.1.2. Detecção

Após a realização dos treinamentos foram realizadas simulações do ataque descrito abaixo e em seguida foram realizadas simulações das analises do trafego gerado com a intenção de verificar a qualidade da detecção da intrusão para cada um dos treinamentos mostrados anteriormente.

4.2.2.1.2.1. Ataque de Scanner

Para a geração de um ataque de scanner, escolhe-se o nodo0 como origem e o nodo7 como destino e realiza-se o procedimento descrito na seção 3.1.4. O ataque é ilustrado pela Figura 4-12.

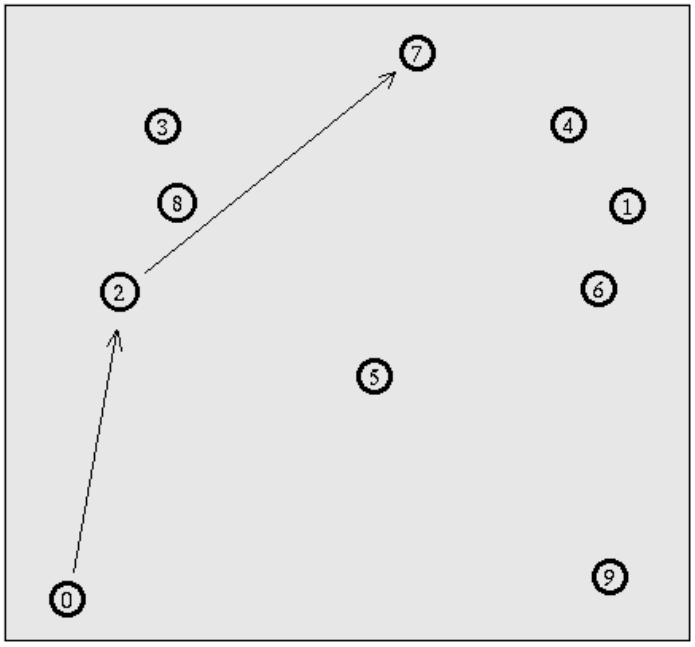

Figura 4-12 – Ataque de Scanner (Cenário sem Mobilidade)

Aplicando-se o modelo detecção, são detectadas situações anômalas em todos os nodos que encaminham o tráfego desde a origem até o destino. Na Tabela 4-5 são mostrados os resultados obtidos para cada intervalo de polling em que foi simulada a utilização do IDS para detecção de intrusão com alta probabilidade. O intervalo de polling representa a freqüência em que o IDS consulta as variáveis que estão sendo monitoradas. Na tabela temos os resultados das taxas de falso positivo, falso positivo durante o ataque e falso negativo. Esses resultados consistem em resultados consolidados de todos os nodos que participam da simulação. Falso positivo representa a detecção de ataque por qualquer nodo em um momento em que não está sendo realizado o ataque. Falso positivo durante o ataque representa a detecção de ataque em nodos que não estão no caminho do ataque durante a realização do ataque. Falso negativo representa a não detecção de ataque em nodos que estão no caminho do ataque durante a realização do ataque.

Tabela 4-5 – Taxas de falso positivo e falso negativo para Ataque de Scanner para detecção com alta probabilidade.

Intervalo de polling	Falso positivo	Falso positivo durante o ataque	Falso negativo
0,1 segundo	0,33%	1,25%	8,33%
0,3 segundo	0,00%	1,67%	8,33%
0,6 segundo	0,17%	0,83%	4,52%
1 segundo	0,17%	1,25%	0,89%
3 segundos	0,00%	0,00%	0,05%
6 segundos	0,50%	0,83%	0,01%
10 segundos	1,50%	2,08%	0,00%
30 segundos	1,75%	1,25%	0,00%
60 segundos	2,47%	2,08%	0,00%
100 segundos	2,67%	1,25%	0,00%

Podemos então verificar que obtivemos as menores taxas de falso positivo, falso positivo durante o ataque e falso negativo com amostragens de 3 em 3 segundos. Esse resultado, bastante satisfatório do ponto de vista de detecção de Scanner só é possível graças a análise conjunta de duas variáveis tcpPassiveOpens e tcpInSegs.

Na Tabela 4-6 são mostrados os resultados obtidos para cada intervalo de polling em que foi simulada a utilização do IDS para detecção de intrusão com baixa probabilidade. Na tabela temos os resultados das taxas de falso positivo, falso positivo durante o ataque e falso negativo da mesma forma em que na Tabela 4-5.

Tabela 4-6 – Taxas de falso positivo e falso negativo para Ataque de Scanner para detecção com baixa probabilidade.

Intervalo de polling	Falso positivo	Falso positivo durante o ataque	Falso negativo
0,1 segundo	8,27%	8,85%	0,00%
0,3 segundo	8,37%	8,67%	0,00%
0,6 segundo	8,17%	8,57%	0,00%
1 segundo	6,17%	6,67%	0,00%
3 segundos	5,00%	5,25%	0,00%
6 segundos	6,00%	6,25%	0,00%
10 segundos	7,83%	7,58%	0,00%
30 segundos	7,33%	7,25%	0,00%
60 segundos	7,00%	7,58%	0,00%
100 segundos	8,00%	8,75%	0,00%

Da mesma forma que na detecção com alta probabilidade, podemos verificar que obtivemos as menores taxas de falso positivo, falso positivo durante o ataque e falso negativo com amostragens de 3 em 3 segundos. Analisando os dados da Tabela 4-5 podemos observar que mesmo no melhor resultado com a periodicidade de 3 segundos, tem a taxa de falso positivo maior que 0, mostrando que em alguns poucos casos um ataque não será detectado. Porem analisando a Tabela 4-6 podemos observar que mesmo no melhor resultado com a periodicidade de 3 segundos, tem a taxa de falso negativo maior que 0, mostrando que em alguns casos um ataque detectado não é ataque. Esses resultados, então, justificam a utilização de três estados de detecção como mostrado no inicio da seção 4.2.

Fazendo uma analise similar ao ataque de DDoS, como o sistema de detecção de intrusão identifica anomalias em todos os nodos do caminho, sugere-se que, caso haja uma interação entre esses nodos intermediários, pode-se bloquear o encaminhamento dos pacotes vindos dessa origem. Esse encaminhamento deve ser bloqueado com base nos endereços de enlace e não nos endereços de destino do datagrama IP, pois esses são

facilmente falsificados e, nos ataques de DDoS mais avançados, são constantemente alterados (a cada pacote).

4.2.2.2. Cenário com movimentação dos nodos

Do mesmo modo que para o modelo UDP, para a validação do modelo, faz-se necessário a realização de simulações com modelos de mobilidade diferenciados. Para isso foram utilizados os modelos: Random Waypoint, Gauss-Markov, Manhattan Grid e Reference Point Group Mobility. Estes modelos de mobilidade são descritos com maiores detalhes na seção 3.1.2. Para cada um dos modelos de mobilidade, foram realizadas simulações de ataques de Scanner. Assim como no cenário sem movimentação, aplicou-se o modelo detecção, e foram detectadas situações anômalas em todos os nodos que encaminham o tráfego desde a origem até o destino. Foi possível verificar que as menores taxas de falso positivo, falso positivo durante o ataque e falso negativo foram obtidas com amostragens de 3 em 3 segundos. Assim como no cenário sem movimentação, foram obtidas taxas de erro na detecção dos ataques sempre menores que 0,1% em todos os modelos de mobilidade. Esses resultados validam de forma satisfatória o modelo do IDS, para os cenários dos 4 modelos de mobilidade adotados.

5. CONCLUSÃO

Neste trabalho, foi apresentado um novo modelo para modelagem estatística do comportamento da rede, usando um modelo paramétrico de misturas de gaussianas, com detecção de anomalias por uso de critérios de classificação Bayesianos (cálculo de probabilidades a posteori). Esse modelo tem por objetivo permitir a modelagem simultânea de diferentes tipos de eventos (e.g. aplicações) que se reflitam em cima de um mesmo conjunto de variáveis disponíveis para monitoração.

Para validação do modelo de IDS por anomalia apresentado, é proposto um modelo de simulação em ambientes de distribuição e topologia mais genéricas. Trata-se de um modelo de geração e processamento de dados de simulação para verificação da aplicabilidade das técnicas de detecção de intrusão por anomalia à detecção de ataques contra Manets. Para proporcionar uma maior credibilidade à validação, foram estudados os principais modelos de mobilidade e de geração de trafego para redes móveis AdHoc.

Como resultado das experiências realizadas, foi possível encontrar a periodicidade mais adequada para a amostragem das variáveis escolhidas. Foi possível verificar que as taxas de falso positivo e falso negativo, são bastante satisfatórias para um sistema de detecção de intrusão. Portanto, os resultados experimentais indicam que esse tipo de modelo pode ser adequado, com uma escolha cuidadosa das variáveis a serem modeladas e monitoradas.

Foi ainda possível ainda observar na fase de treinamento que existe uma dependência com a escala de tempo para o intervalo de polling. Em outras palavras, o resultado do treinamento pode variar dependendo da escala de tempo utilizada. Esse fenômeno ocorre devido ao fato do modelo de trafego normal utilizado ser um modelo simplificado. Existe uma tendência em se usar modelos de trafego auto-similares, onde essa dependência não deve existir. Fica como sugestão para trabalhos futuros, os testes com trafego auto-similar, e um estudo para verificar se a utilização de trafego auto-similar se aplica para redes móveis Ad Hoc.

Outra análise que pode ser realizada com os resultados obtidos é que nos ataques de DoS ou DDoS, o nodo que recebe todo o tráfego gerado (de todos ou de um dos seus vizinhos) vai tornar-se possivelmente indisponível impossibilitando até mesmo a sua

participação da detecção da intrusão. Uma possibilidade para resolver esse problema, seria a utilização de informações da camada 2, para que nodos que não estão no caminho do ataque, mas que estão na sua vizinhança, possam também detectar o ataque. Fica também como sugestão de trabalhos futuros, a utilização de variáveis da MIB que representem as informações da camada 2.

Com o objetivo de obter uma maior precisão na descoberta do caminho do ataque, foram realizados testes preliminares com a correlação de dados de nodos diferentes. As primeiras analises com relação a essa abordagem, não nos trouxeram bons resultados, mas fica também com uma possibilidade para trabalhos futuros uma reflexão mais apurada desta solução e a realização de outros testes que possibilitem verificar se existe algum ganho na qualidade da informação do caminho do ataque, se for utilizado esse tipo de correlação.

Como ultimas sugestões para trabalhos futuros, é proposto também uma maior validação com dados de referência provenientes de redes reais. Alem disso, muitas melhorias nas pré-condições da concepção do modelo podem ser realizadas, como a utilização de outros tipos de função kernel, a utilização de um modelo semi-paramétrico de mistura e a adoção de modelos estocásticos (e.g. processo de Markov).

Por fim, os resultados deste trabalho nos proporcionaram a publicação e a apresentação de trabalhos em dois importantes congressos internacionais. São eles I2TS'2005 [2] e PWC'2006 [3].

REFERÊNCIAS BIBLIOGRÁFICAS

[1] R. Puttini, *Um Modelo de Segurança Para Redes Móveis Ad Hoc.* Tese de Doutorado, Publicação ENE 004/04, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 2004.

[2] F. Miziara, M. Hanashiro, M. Puttini, R. Sousa, R. Puttini, *An Anomaly Intrusion Detection System for Manet*, I2TS, 2005.

[3] R. Puttini, M. Hanashiro, F. Miziara, R. Sousa, L. Javier, C. Barenco, *On the Anomaly Intrusion-Detection in Mobile Ad-Hoc Network Environments*, PWC, 2005.

[4] Campos, C., Otero, D., and de Moraes, L. (2003). *Realistic individual mobility markovian models for mobile ad hoc networks*. IEEE Wireless Communications and Networking Conference, WCNC, pages 1980–1985.

[5] V. Davies, *Evaluating Mobility Models within an Ad hoc Network*, Master'sthesis, Colorado School of Mines Los Angeles, USA, 2000.

[6] G. P. X. Hong, M. Gerla and C. Chiang, *A GroupMobility Model for Ad hocWireless Networks*, in Proc. of the ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems (MSWiM), aug 1999.

[7] P. Papadimitratos and Z. Haas, *Secure Routing for Mobile Ad hoc Networks*, CNDS, 2002.

[8] S. Akaho, *Mixture Model for Image Understanding and the EM Algorithm*, ETL Technical Report TR-95-13, 1995.

[9] P. Albers, O. Camp, J. Percher, B. Jouga, L. Mé, and R. Puttini - *Security in Ad hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches*. WIS 2002, Ciudad Real Spain, April 2002.

[10] M. Asaka. *The Implementation of IDA: An Intrusion Detection Agent System.* Proceedings of the 11th Annual FIRST Conference on Computer Security Incident Handling and Response, Brisbane, Australia, June, 1999.

[11] J. Balasubraniyan, J. Fernandez, D. Isacoff, E. Spafford, D. Zamboni. *AAFID - Autonomous Agents For Intrusion Detection,* Technical report 98/05, COAST Laboratory Purdue University, June 1998.

[12] M. C. Bernardes, E. Moreira. *Implementation of an intrusion detection system based on mobile agents.* In Proceedings of 2000 International Symposium on Software Engineering for Parallel and Distributed Systems, pp. 158-164, 2000.

[13] L. Buttyan and J. P. Hubaux. *Stimulating cooperation in self-organizing mobile ad hoc networks.* ACM Journal for Mobile Networks (MONET), special issue on Mobile Ad Hoc Networks, 2002.

[14] J. Cabrera, L. Lewis, R. Prasanth, X. Qin, W. Lee, and R. Mehra. *Proactive detection of distributed denial of service attacks using MIB traffic variables – a feasibility study*. Proceedings of the 7th IFIP/IEEE International Symposium on Integrated Network Management, Seattle, WA, USA, may 2001.

[15] S. Capkuny, J. Hubaux, and L. Buttyan. *Mobility Helps Security in Ad Hoc Networks.* In Proceedings of MobiHoc'03.

[16] P.C. Chan, V. K. Wei. *Preemptive distributed intrusion detection using mobile agents.* In Proceedings of the Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2002), pp. 103-108, 2002.

[17] P. Cheeseman and J. Stutz, *Bayesian classification (AutoClass): theory and results. Advances in Knowledge Discovery and Data Mining*, U. M. Fayyad, G. Piatetsky-Shapiro, R. Smyth and R. Uthurusamy (Eds.), Menlo Park, California: The AAAI Press, pp. 61-83, 1996.
[18] D. Curry, H. Debar, and Merrill Lynch. *Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML).* IETF Internet draft. June 2002.
[19] H. Debar, M. Dacier and A. Wespi. *A revised taxonomy for intrusion-detection systems*, IBM Research Report, Zurich, 1999.
[20] A. P. Dempster, N. M. Laird and D. B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm* (with discussion). Journal of the Royal Statistical Society B 39 ,1-38, 1977.
[21] S. Fenet, S, Hassas. *A Distributed Intrusion Detection and response System Based on Mobil Autonomous Agents Using Social Insects Communication Paradigms.* In First International Workshop on Security of Mobile Multiagent Systems (SEMAS2001), Montreal, Canada, May, 2001.
[22] Y.F. Fou, F. Gong, C. Sargor, X. Wu, S. F. Wu, H. C. Chang, F. Wang *JINAO. Design and Implementation of a Scalable Intrusion Detection System for the OSPF Routing Protocol*, Advanced Networking Research, MCNC Computer Science Dept, NC State University, February, 1999.
[23] A. Genz, *Numerical Computation of Multivariate Normal Probabilities*, J. Comp. Graph Statistics vol.1, pp. 141-149 (1992)
[24] S. Gwalani, E. Royer, G. Vigna, R. Kemmerer. *AODVSTAT: Intrusion Detection in AODV*. (work in progress)
[25] G. Helmer, J. Wong, V. Honavar, L. Miller, Y. Wang. *Lightweight Agents For Intrusion Detection.* To be published in The Journal of Systems and Software.
[26] S. Hofmeyr, S. Forrest. *Architecture of an Artificial Immune System. Evolutionary Computation 7(1)*, Morgan-Kaufmann, San Francisco, CA, pp. 1289-1296 (2000).
[27] K. Ilgun, R. A. Kemmerer, and P. A. Porras. *State Transition Analysis: A Rule-Based Intrusion Detection Approach.* IEEE Transactions on Software Engineering, pp. 181-199, March 1995.
[28] W. Jansen. *Intrusion Detection with Mobile Agents*. To be published in Computer Communications Journal, Special Issue on Intrusion Detection.
[29] A. S. Javits and A. Valdets, *The SRI IDES Statistical Anomaly Detector*, Proc. of IEEE Symposium of Research on Security and Privacy, pp. 316-326, may 1991.
[30] R. A. Johnson, D. A. Wichern, D. W. Wichern, *Applied Multivariate Statistical Analysis – 4th Edition*, Prentice-Hall, 1998.
[31] I.T. Jolliffe , *Principal Component Analysis*, Springer Series in Statistics, May 1986.
[32] Christopher Krügel, Thomas Toth, *Flexible, Mobile Agent Based Intrusion Detection for Dynamic Networks.* Proceedings of European Wireless (EW2002), Italy, February 2002.
[33] Carl E. Landwehr, Alan R. Bull, John P. McDermott, and William S. Choi *A taxonomy of computer program security flaws*. ACM Computing Surveys, 26(3):211-254, September 1994.
[34] D. Curry, H. Debar, and Merrill Lynch. *Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML).* IETF Internet draft. June 2002.

[35] W. Lee; S. J. Stolfo; and K. W. Mok. *A data mining framework for building intrusion detection models.* Proceedings of the 1999 IEEE Symposium on Security and Privacy, 1999.
[36] S. Marti, T. J. Giuli, K. Lai, and M. Baker. *Mitigating routing misbehaviour in mobile ad hoc networks.* In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking, Boston, MA, August 2000.
[37] S. Martino. *A mobile agent approach to intrusion detection*, technical report, Joint Research Centre Institute for Systems, Informatics and Safety, Italy, June 1999.
[38] K. McCloghrie; and A. Bierman. *Entity MIB (Version 2).* IETF Request for Comment 2737, December 1999.
[39] G. J. McLachlan, D. Peel, K. E. Basford and P. Adams, *The EMMIX Software for the Fitting of Mixtures of Normal and t –Components*, Journal of Statistical Software, v. 04, 1999.
[40] P. Mell, D. Marks, M. McLarnon. *A Denial-of-Service Resistant Intrusion Detection Architecture.* Computer Networks, Special Issue on Intrusion Detection, Elsevier Science BV, November 2000.
[41] V. Mittal and G. Vigna. *Sensor-based intrusion detection for intra-domain distance-vector routing.* In R. Sandhu, editor, Proceedings of the ACM Conference on Computer and Communication Security (CCS'02), Washington, DC, November 2002. ACM Press.
[42] JM. Percher; R. Puttini; L. Mé; O. Camp; B. Jouga; P. Albers - *Un système de détection d'intrusion distribué pour réseaux ad hoc*, TSI, France, 2004.
[43] Phillip A. Porras, Peter G. Neumann. *EMERAL - Event Monitoring Enabling Responses to Anomalous Live Disturbances*, Conceptual Overview, December, 1996.
[44] R. Puttini, Z. Marrakchi and L. Mé. *Bayesian Classification Model for Real-Time Intrusion Detection*, 22th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering (MAXENT'2002). August 2002.
[45] R. Puttini; J.M Percher; L. Me; O. Camp; R. de Sousa - *A Modular Architecture for a Distributed IDS for Mobile Ad Hoc Networks.* International Conference on Computer Science and Applications in Lecture Notes on Computer Science 2669:91-113, Springer, 2003.
[46] R. Puttini; J.M. Percher; L. Me; R. de Sousa. *A Fully Distributed IDS for Manet*. In Proceedings of 9th IEEE International Symposium on Computers Communications, 2004.
[47] R. Puttini; L. Me; R. de Sousa - *Preventive and Corrective Protection for Mobile Ad Hoc Network Routing Protocols*. In Proceedings of 1st International Conference on Wireless On-demand Network Systems in Lecture Notes on Computer Science, Springer, 2004.
[48] S. J. Roberts, R. Everson and I. Rezek, *Maximum Certainty Data Partitioning*, Pattern Recognition, 33:5, pp. 833-839, 1999.
[49] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and S. Zhou. *Specication-based anomaly detection: a new approach for detecting network intrusions*. In ACM Computer and Communication Security Conference (CCS), 2002.
[50] Eugene H. Spafford and Diego Zamboni. *Intrusion detection using autonomous agents. Computer Networks*, 34(4):547-570, October 2000.
[51] S. Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, D. Zerkle. *GrIDS- A Graph Based Intrusion Detection System*

for Large Networks, Computer Security Laboratory,Department of Computer Science,University of California, Davis, 1996.
[52] Steven R. Snapp, , James Brentano, Gihan V. Dias, Terrance L. Goan, L. Todd Heberlein, Che-Lin Ho, Karl N. Levitt, Biswanath Mukkherjee, Stephen E. Smaha, Tim Grance, Daniel M. Teal, and Doug Mansur. *DIDS-Distributed Intrusion Detection System*, Computer Security Laboratory, Department of Computer Science, University of California, Davis, June 1992.
[53] D. M. Titternington, *Recursive Parameter Estimation using Incomplete Data*, J. R. Statist. Soc. B, n.o 46, pp. 257-267.
[54] C.-Y. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt. *A specification-based intrusion detection system for AODV*. In ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03), October 2003.
[55] Gregory B White, Eric A. Fish and Udo Pooch. *CSM - Cooperating Security Managers: a peer based intrusion detection system*, IEEE Networks, pages 20-23, January/February 1996.
[56] M. Wood and M. Erlinger. *Intrusion Detection Message Exchange Requirements*, IETF Internet Draft, October 22, 2002. http://www.ietf.org/internet-drafts/draft-ietf-idwg-requirements-10.txt
[57] H. Yang, X. Meng and S. Lu, *Self-Organized Network Layer Security in Mobile Ad Hoc Networks*, in the Proceedings of ACM Workshop on Wireless Security – 2002 (WiSe'2002), in conjunction with the ACM MOBICOM2002, September, 2002.
[58] Y. Zhang and W. Lee. *Intrusion detection in wireless ad hoc networks*. Proceedings of 6th ACM Annual International Conference on Mobile Computing and Networking (MOBICOM 2000), ACM Press, New York, pp. 275-283, 2000.
[59] L. Zhou and Z. J. Haas. *Securing ad hoc networks*. IEEE Network Magazine, 13(6):24-30, November/December 1999.
[60] B. Dahill, K. Sanzgiri, B. N. Levine, C. Shields and E. Royer, *A secure routing protocol for ad hoc networks*. In the Proceedings of the 2002 IEEE International Conference on Network Protocols (INCP 2002), Nov. 2002.
[61] M. Guerrero and N. Asokan, *Securing Ad Hoc Routing Protocols*, in the Proceedings of 2002 ACM Workshop on Wireless Security (WiSe'2002), in conjunction with the ACM MOBICOM2002, September, 2002.
[62] P. Papadimitratos and Z. J. Haas. *Secure routing for mobile ad hoc networks.* SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), Jan 2002.
[63] J. Barrus, N. Rowe. *A Distributed Autonomous-Agent Network-Intrusion Detection and Response System.* In the Proceedings of the 1998 Command and Control Research and Technology Symposium, Monterey, CA, June-July 1998.
[64] Y. Huang, W. Fan, W. Lee, and P. Yu. *Cross-feature analysis for detecting ad-hoc routing anomalies.* In The 23rd International Conference on Distributed Computing Systems, May 2003.
[65] C. Fraley and A. E. Raftery. *MCLUST: Software for Model-Based Cluster and Discriminant Analysis*, Technical Report No.342, Department of Statistics, University of Washington, 1998.