Detecção de Intrusos Descentralizada em Redes de Sensores Sem Fio

Ana Paula R. da Silva[1], **Marcelo H. T. Martins**[1], **Bruno P. S. Rocha**[1]
Antonio A. F. Loureiro[1], **Linnyer B. Ruiz**[1], **Hao C. Wong**[1]

[1]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais
Caixa Postal 702 – 30123-970 Belo Horizonte, MG, Brasil

{anapaula,marcelo,bpontes,loureiro,linnyer,hcwong}@dcc.ufmg.br

Abstract. *Wireless sensor networks (WSNs) have applications that may attract adversaries' attention, such as natural resources mapping and enemy monitoring in a battlefield. In these cases, an intrusion detection system (IDS) becomes necessary. Besides preventing damages caused by intruders, an IDS can acquire information related to attack techniques which may assist in their prevention. In this work, we present a decentralized IDS that fits WSNs demands and restrictions. Our proposal is based on inference about network behavior obtained from perceived events on the WSN. Simulation results show its accuracy and precision in detecting different types of attack and how energy consumption is affected by its introduction into the network.*

Resumo. *Redes de Sensores Sem Fio (RSSF) possuem aplicações que podem atrair o interesse de adversários, como o mapeamento de recursos naturais e a monitoração do inimigo no campo de batalha. Nesses cenários, faz-se necessária a utilização de um sistema de detecção de intrusos (IDS) que, além de evitar que intrusos causem danos à rede, pode adquirir informações sobre técnicas de ataque, auxiliando na sua prevenção. Neste trabalho propomos um IDS descentralizado que atende às demandas e restrições das RSSFs. O IDS proposto baseia-se na inferência de comportamento da rede, obtido a partir de eventos percebidos neste ambiente. Resultados de simulação apresentam uma avaliação da eficácia e precisão do IDS na detecção de diversos tipos de ataques e uma avaliação do consumo de energia na rede.*

1. Introdução

Redes de Sensores sem Fio (RSSFs) constituem um novo paradigma de monitoração ambiental com muitas aplicações em potencial. Formadas tipicamente por centenas ou até milhares de nós de tamanho reduzido, esses dispositivos utilizam comunicação *ad hoc* para transmissão de dados e possuem recursos limitados em termos de reserva de energia, largura de banda, capacidade de processamento e armazenamento. RSSFs são projetadas para atuarem em ambientes muitas vezes inóspitos e podem estar envolvidas em aplicações críticas, tais como mapeamento de recursos naturais [Mainwaring et al. 2002] e monitoração de movimentação inimiga em um campo de batalha [Arora et al. 2004]. Nessas aplicações, as RSSFs podem se tornar alvo de interesse dos adversários.

Devido à natureza não-confiável da comunicação sem fio, ao fato de serem implantadas em ambientes abertos e desprotegidos, serem constituídas por dispositivos de tamanho reduzido e baixo custo, as RSSFs estão sujeitas a vários tipos de ataque [Karlof and Wagner 2003, Wood and Stankovic 2002]. Mecanismos preventivos podem ser aplicados para protegê-las de ataques, como mostrado em [Karlof et al. 2004,

Perrig et al. 2002]. No entanto, não existem garantias de que os métodos preventivos serão capazes de deter os intrusos. Sendo assim, faz-se necessário um mecanismo capaz de identificar esses ataques e seus responsáveis. Além de evitar que um intruso cause danos à rede, um sistema de detecção de intrusos (*Intrusion Detection System* – IDS) pode adquirir informações sobre as técnicas de ataques, auxiliando no desenvolvimento de sistemas de prevenção.

Neste cenário, vários desafios devem ser considerados. Primeiramente, as RSSFs são direcionadas à aplicações, isto é, projetadas com características específicas para as aplicações a que se destinam. A variedade de configurações de rede dificulta a modelagem do comportamento "usual" ou "esperado" do sistema. Além disso, métodos desenvolvidos para redes tradicionais não são diretamente aplicáveis, pois a disponibilidade de recursos é ordens de grandeza maior que nas RSSFs.

Por serem dispositivos de baixo custo, os nós sensores não dispõem de muita memória disponível, o que dificulta o armazenamento de informações sobre detecções. Além disso, o acesso aos dados de detecção pode ser dificultado caso a RSSF esteja instalada em um ambiente inóspito ou algum nó seja descartado ou avariado durante seu funcionamento. O *software* dos nós sensores deve ser projetado de modo a economizar a maior quantidade de energia possível, para que o tempo de vida da rede possa ser prolongado. Finalmente, as falhas não são exceção em uma RSSF e devem ser uma das preocupações ao se projetar IDSs para esse tipo de rede. Além das restrições acima, ainda é importante que o intruso seja detectado em tempo real, minimizando os prejuízos causados à aplicação.

O objetivo deste trabalho é estudar as principais questões que envolvem a detecção de intrusos em RSSFs e propor um IDS que atenda às demandas e restrições desse tipo de rede. As principais contribuições deste trabalho são: proposta de um modelo de IDS descentralizado adequado às limitações e peculiaridades das RSSFs; avaliação da eficácia e precisão do IDS proposto na detecção de oito tipos ataques por meio de simulação; avaliação dos custos de utilização deste IDS em termos de consumo de energia; e o desenvolvimento de um simulador simplificado capaz de simular as principais características das RSSF e do IDS proposto. O IDS proposto baseia-se na inferência de comportamento da rede, obtido a partir da análise de eventos detectados pelo monitor, nó em que o IDS se encontra instalado.

Este artigo está organizado como a seguir. A seção 2. descreve brevemente os trabalhos relacionados. A seção 3. descreve os ataques considerados e as regras de detecção propostas para o IDS aqui descrito. A seção 4. discute o algoritmo proposto utilizado no IDS. A seção 5. apresenta algumas questões de projeto associadas ao IDS e cenário de simulação. A seção 6. apresenta e discute vários resultados de simulação obtidos na modelagem do IDS proposto. Finalmente a seção 7. apresenta a conclusão deste trabalho.

2. Trabalhos Relacionados

A detecção de intrusos é um dos tópicos de discussão mais importantes na área de segurança de redes. Várias soluções já foram propostas para redes tradicionais [Ilgun et al. 1995, Porras and Neumann 1997, Paxson 1999, Huang et al. 1999, Kumar and Spafford 1995], mas as restrições de recursos das RSSFs tornam a aplicação direta dessas soluções inviável.

Redes *ad hoc* são similares às RSSFs, visto que também possuem restrições em termos de recursos, apesar de não serem tão severas quanto as das RSSFs. Algumas

soluções já foram apresentadas na área de detecção de intrusos em redes *ad hoc* e podem servir como base para trabalhos semelhantes em RSSFs.

Em [Hu et al. 2003], ataques de *Wormhole* são detectados em redes *ad hoc* por meio da avaliação do tempo gasto na transmissão de um pacote de um ponto a outro da rede, e da autenticação dos nós. São apresentados dois protocolos utilizados na camada de enlace: *Slot Authenticate MAC* e *TIK*. Ambos necessitam de sincronismo confiável entre os nós. Devido à complexidade em se manter nós sincronizados em uma RSSF, essa premissa não foi utilizada neste trabalhos. Em [Jr. et al. 2004], um método de detecção de ataques de *Wormhole* e *HELLO flood* em RSSFs também é proposto. Através da identificação da potência do sinal recebido em comparação com a potência do sinal observado na rede, é possível determinar se um ataque está ocorrendo ou não. Neste trabalho, utilizou-se uma estratégia simples baseada na topologia da rede, na qual é suficiente que o nó monitor tenha conhecimento apenas dos identificadores dós nós que estão dentro do seu raio de alcance para detecção do ataque de *Wormhole*. A estratégia proposta em [Jr. et al. 2004] também pode ser aproveitada na aplicação de uma das regras do sistema proposto, caso os nós da rede possuam meios para medir a potência do sinal recebido (RSSI). Nosso trabalho propõe uma solução mais abrangente, capaz de detectar vários tipos de ataques.

Em [Marti et al. 2000], é apresentada a aplicação de *watchdogs* em redes *ad hoc*, como técnica para detecção de nós mal comportados, e o método denominado *pathrater*, que auxilia na escolha das rotas a serem consideradas na transmissão de mensagens, evitando esses nós. Neste trabalho, utilizou-se uma idéia semelhante a da aplicação de *watchdogs*. Como poderá ser visto em seções posteriores, o nó monitor vigia seus vizinhos e observa quais ações cada um deles toma em relação às mensagens recebidas provenientes de outros nós em direção ao seu destino final. Se o vizinho do nó monitor alterar, atrasar, replicar ou não retransmitir a mensagem que deveria ser retransmitida, uma falha é contabilizada. Além desta técnica, outras são utilizadas para detectar demais tipos de ataque.

Em [Deng et al. 2003], é proposto um protocolo de roteamento tolerante a falhas que tem como objetivo manter a rede funcionando mesmo na presença de nós intrusos, utilizando rotas redundantes. Muitos dos ataques encontrados na literatura não podem ser tolerados, o que motiva o desenvolvimento de um IDS que seja adequado às RSSFs.

Em [da Silva et al. 2004], é apresentado um estudo geral e uma série de possibilidades de construção de um IDS para RSSFs. Algumas idéias propostas foram utilizadas como base para a modelagem deste trabalho.

3. Ataques Considerados e Regras Utilizadas

Os seguintes ataques sobre RSSF são considerados neste trabalho:

- **Interferência (*Jamming*):** neste ataque, o intruso possui um transceptor potente configurado para utilizar a mesma freqüência dos nós sensores, podendo ocupar o canal de comunicação com ruído e impedir que os nós sensores recebam qualquer tipo de mensagem.
- **Alteração de dados:** neste ataque, o intruso captura uma mensagem e a retrasmite de forma alterada.
- **Negligência de Dados, *Blackhole* e *Selective Forwarding*:** nestes ataques, o intruso ignora mensagens que deveria enviar (Negligência de Dados) ou retransmitir. No caso do ataque de *Blackhole*, o intruso omite todas as mensagens que deveria retransmi-

tir, enquanto no *Selective Forwarding* ele deixa de retransmitir algumas mensagens de forma aleatória ou seguindo algum critério.

- **Repetição e Atraso:** nestes ataques, o intruso, respectivamente, repete ou atrasa as mensagens que deveria restransmitir.
- ***Wormhole*:** neste ataque, o intruso, com um transceptor mais potente que os dos demais nós da rede, transmite mensagens a pontos distantes para, por exemplo, confundir os nós sensores em relação à sua vizinhança.

Foram considerados apenas ataques em nível de dados/rede, enquanto a inclusão de tratamento de ataques em nível de aplicação será feito em trabalhos futuros.

Foi utilizada a metodologia para construção de um IDS apropriado para uma RSSF específica proposta em [da Silva 2005]. A seguir, são apresentadas as definições das regras utilizadas no IDS proposto:

Regra de intervalo: Considerando um intervalo de disseminação de dados pré-definido, uma falha é contabilizada se o período entre a transmissão de duas mensagens consecutivas for menor do que um limite estabelecido. O ataque provavelmente detectado por esta regra é o Negligência de Dados.

Regra de retransmissão: Um nó deve sempre retransmitir a mensagem que receber, caso ele não seja seu destino final. Caso o monitor perceba que seu nó vizinho deixou de retransmitir tal mensagem, uma tentativa de ataque é contabilizada. Os ataques que podem ser detectados por esta regra são o *Selective Forwarding* e o *Blackhole*.

Regra de integridade: Os dados (*payload*) contidos na mensagem recebida por um nó sensor não devem ser alterados durante a sua retransmissão (neste modelo não foram consideradas aplicações que utilizam fusão de dados). Esta regra detecta tentativas de ataque de Alteração de Dados.

Regra de atraso: A retransmissão de uma mensagem por um nó deve ser feita dentro de um período pré-definido. Caso a mensagem seja atrasada, este comportamento é identificado como uma tentativa de ataque de Atraso de mensagens.

Regra de repetição: Não foi considerado neste modelo nenhum tipo de tratamento de erros, como a retransmissão de mensagens em caso de falha na transmissão. Sendo assim, um nó deverá transmitir uma mensagem apenas uma vez. Esta regra detecta tentativas de ataque de Repetição de mensagens.

Regra de alcance de rádio: Mensagens que são recebidas ou coletadas por um nó devem ser originadas em nós que estejam dentro de seu raio de alcance. Esta regra é capaz de detectar tentativas de ataque de *Wormhole*.

Regra de interferência: O número de colisões que ocorrem durante a transmissão de mensagens deve ser menor ou igual a um número de colisões esperadas, característico do meio de comunicação. O ataque de interferência (*jamming*) pode ser detectado por esta regra.

No caso das regras de retransmissão, integridade, atraso, repetição e de intervalo, o monitor é capaz de não somente detectar a tentativa de intrusão, como também descobrir o identificador e a localização do invasor.

4. Algoritmo Proposto

A figura 1 mostra a arquitetura de um nó comum onde foi instalado o IDS, assumindo assim o papel de monitor. Esse mesmo nó ainda executa as funções de nó comum (sensoriamento do ambiente, envio de mensagens com os dados sensoriados e retransmissão de mensagens recebidas). O IDS instalado neste nó possui três módulos de *software*, cada um responsável por uma das fases descritas a seguir.

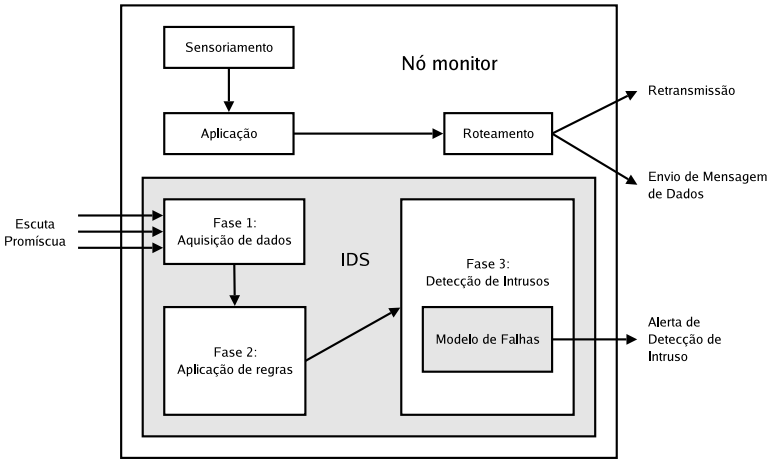

Figura 1. Arquitetura do nó monitor

Fase 1: Aquisição de Dados

Nesta fase, as mensagens são coletadas em escuta promíscua pelo nó monitor e os dados de interesse são filtrados antes que possam ser armazenados para posterior análise. São armazenados apenas os campos de mensagem necessários para a aplicação das regras de detecção, diminuindo o espaço ocupado em memória e o tempo de processamento no nó monitor. Mensagens sobre as quais não é possível aplicar nenhuma regra não são armazenadas com o intuito de ocupar o mínimo de memória (recurso escasso em RSSF) e diminuir o processamento tanto quanto possível para fins de economia de energia.

Os dados de interesse extraídos são armazenados em vetores, funcionando como *buffers*. Essas informações permanecem armazenadas até o momento em que o espaço destinado a elas se esgota, momento em que a próxima fase de detecção é acionada.

Fase 2: Aplicação de Regras

Nesta fase, as mensagens armazenadas pelo nó monitor são avaliadas de acordo com as regras instaladas no IDS. Se na aplicação de qualquer uma das regras for detectada uma falha (i.e., uma mensagem, ou uma seqüência de mensagens, não está de acordo com os parâmetros estipulados para uma determinada regra), uma falha é contabilizada e a mensagem é descartada sendo que nenhuma outra regra é aplicada sobre ela. Adotou-se esta estratégia em consideração às restrições computacionais dos nós pertencentes à RSSF. Uma vez que a mensagem não está de acordo com uma das regras aplicadas sobre ela, tem-se um indicativo de comportamento anômalo na rede. Além disso, nossa estratégia permite que os monitores economizem energia de processamento, além de testar as mensagens mais rapidamente, diminuindo a latência na detecção. Nota-se então um

compromisso entre precisão de detecção, custo em processamento e tempo de execução. A seqüência de aplicação das regras foi escolhida de forma que as regras mais simples fossem testadas primeiro. Caso a mensagem resulte em uma falha no teste mais simples, os testes mais complexos não serão executados. Esta estratégia foi escolhida para fins de economia de processamento e conseqüente economia de energia.

Fase 3: Detecção de Intrusos

Ao final do processamento dos vetores, as falhas contabilizadas são comparadas com um modelo de falhas naturais da RSSF específica. Um intruso é detectado se o número de falhas observadas for maior que o número esperado de falhas naturais da rede.

Modelo de Falhas: O número esperado de falhas naturais em cada configuração de rede é definido dinamicamente pelo nó monitor. Para cada nó pertecente à sua vizinhança, o monitor mantém um histórico de falhas que é atualizado periodicamente após a fase de aplicação de regras.

Como a expectância do número de falhas leva tempo para se estabilizar, um grande número de falsos positivos seria detectado pelos nós monitores no início do ciclo de vida da rede. Para que isso não ocorra, uma parcela inicial do tempo de simulação, denominada *etapa de aprendizagem*, foi reservada para que essa média se estabilize. Durante essa etapa, foi assumido que nenhum ataque é realizado e assim o nó monitor considera todos os comportamentos anômalos detectados como falhas naturais da rede.

5. Simulador

Alguns simuladores foram propostos para avaliar soluções em RSSFs [Park et al. 2000, Shnayder et al. 2004, Fall and Varadhan 2001] e são utilizados na obtenção de resultados de estudos em áreas relacionadas. Em alguns casos, no entanto, esses simuladores não são adequados, como ocorreu na solução de detecção de intrusos em RSSFs. Como se trata de uma idéia nova, os simuladores avaliados não previam as funcionalidades necessárias para esta nova abordagem, além de apresentarem alta complexidade, baixa performance e serem de difícil manipulação e agregação de novas funcionalidades.

Visto que, até onde foi pesquisado, nenhum dos simuladores estudados atendiam a todos os requisitos necessários para a aplicação proposta, decidiu-se implementar uma solução própria [Martins et al. 2005]. Dessa maneira, tem-se controle total sobre o comportamento da aplicação, além de possibilitar a criação de uma ferramenta robusta e de menor complexidade, voltada especificamente para o problema de detecção de intrusos em RSSFs.

Foi implementado um modelo de eventos discretos, no qual os objetos de análise (estação base, nós comuns, monitores e intrusos) mantêm seus estados durante a simulação até a ocorrência de algum evento como, por exemplo, a recepção ou o envio de uma mensagem, a ocorrência de um sensoriamento ou a ativação de um ataque. Os eventos de sensoriamento de rede são gerados aleatoriamente e os nós não são sincronizados, na tentativa de aproximar o simulador do que seria o comportamento de uma rede real. Os seguintes módulos foram implementados: rede, mensagem, nó sensor, nó monitor, nó intruso, gerador de eventos e ataques, IDS e coletor de estatísticas. O módulo de rede é responsável pela troca de mensagens entre os demais módulos de tal forma que possa simular o funcionamento de uma RSSF real.

5.1. Cenário

Foram modelados quatro tipos de nós: nó comum, monitor, intruso e estação base. O *nó comum* possui as funções de sensor e roteador, i.e., sensoria o ambiente, enviando os dados coletados para a estação base, e retransmite mensagens de dados originadas em nós vizinhos. O *monitor*, além das funções de nó comum, é responsável pela monitoração de seus vizinhos em busca de indícios de ataques, de acordo com as fases descritas na seção 4.. O *intruso* oscila entre o comportamento de um nó comum e o comportamento de um invasor. As ações características de um comportamento intrusivo dependem do ataque considerado. No modelo proposto, a *estação base* é o destino final de todas as mensagens de dados.

Foram considerados apenas ataques sobre mensagens de dados, como descritos na seção 3.. Três tipos de falhas naturais na rede foram propostas, como descritas a seguir:

1. **Alteração de dados**: ocorre quando os dados da mensagem (*payload*) são modificados acidentalmente para um valor diferente do original. O nó monitor pode confundir esta falha natural com o ataque de Alteração de dados.
2. **Perda de mensagens**: ocorre quando a mensagem é acidentalmente perdida durante sua transmissão. Neste caso, o nó que originou a mensagem não percebe a perda. O nó monitor pode confundir esta falha natural com os ataques de Negligência de Dados, *Blackhole* e *Selective Forwarding*.
3. **Colisão de mensagens**: ocorre quando a mensagem é acidentalmente perdida durante a transmissão. Neste caso, o nó onde a mensagem foi originada detecta a perda devido a uma colisão. O nó monitor poderá confundir esta falha natural com o ataque de *Jamming*.

Neste modelo, não foram consideradas falhas naturais de atraso, retição e falhas naturais que podem ser confundidas com o ataque de *Wormhole*.

A mensagem de dados contém os seguintes campos: destino imediato (*next hop*), tipo de mensagem, fonte imediata (*previous hop*), origem, destino final, número de seqüência e dados de sensoriamento. Esses campos são suficientes para a aplicação de todas as regras descritas. Foi simulada uma rede plana e fixa [Ruiz et al. 2003], com distribuição aleatória dos nós. Os nós são unicamente identificados e possuem um alcance de rádio fixo.

As mensagens são transmitidas em *multihop* seguindo a árvore de roteamento gerada a partir do algoritmo distribuído de Propagação de Informação (*Propagation of Information* – PI [Segall 1983]). Como não cogitou-se a inclusão de fusão de dados nos nós sensores, a mensagem recebida por um nó comum deve ser retransmitida sem nenhuma alteração em seu *payload*. No modelo de RSSF considerado, não existe nenhum tipo de tratamento de falhas de transmissão, tais como confirmação de recebimento (*ACK*) e retransmissão de mensagens. Um nó deverá receber mensagens somente de nós que estejam dentro de seu raio de comunicação. Foi proposto um intervalo máximo em que um nó deve retransmitir as mensagens que passam por ele. Esse intervalo foi definido empiricamente por meio de observações sobre a variação de produção de mensagens de dados no simulador e conseqüente carga da rede.

Neste modelo, falhas naturais de rede seguem um modelo probabilístico, que é aplicado cada vez que uma mensagem está prestes a ser enviada por um nó. Uma falha natural podem ser erroneamente detectada como um ataque, resultando em um falso positivo.

Foi simulada uma rede contendo 100 nós distribuídos aleatoriamente, na qual o intervalo de disseminação de dados dos nós sensores era de 40 iterações do simulador.

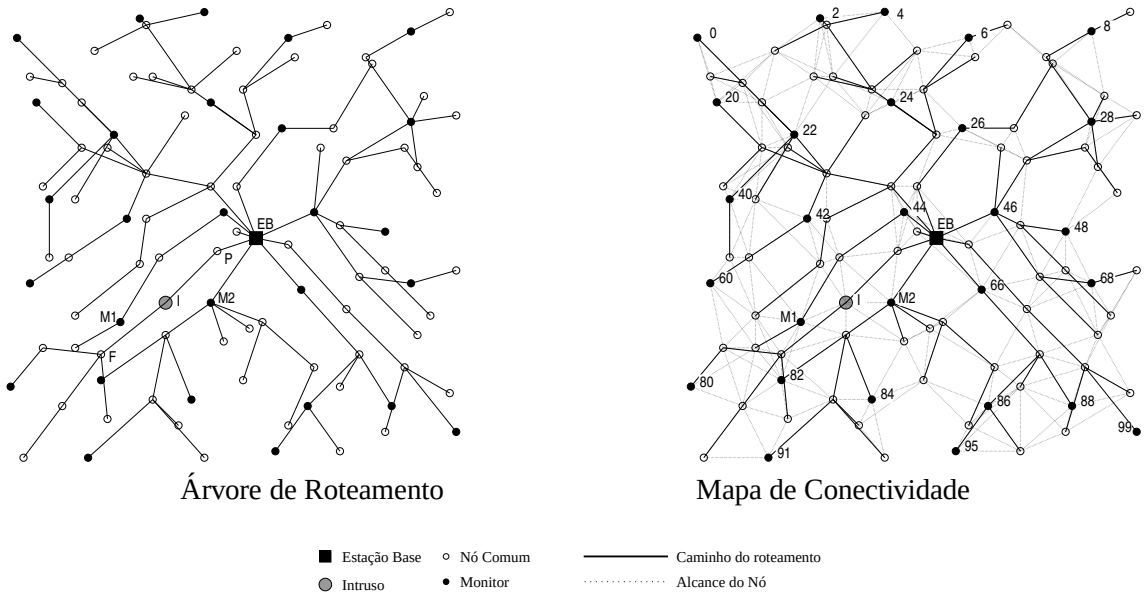

Figura 2. (a) Árvore de roteamento, (b) Mapa de conectividade

A cada iteração, um nó pode receber, processar e enviar uma mensagem. A figura 2 ilustra duas representações da rede utilizada nos experimentos. Na figura 2(a) encontra-se a árvore de roteamento, enquanto a figura 2(b) apresenta o mapa de conectividade da mesma topologia. Uma aresta conectando dois vértices do mapa indica que os nós estão dentro dos seus respectivos raios de comunicação. A aresta pontilhada representa o alcance entre os nós, enquanto a aresta contínua representa, além do alcance, o caminho de roteamento estabelecido na topologia.

Os nós monitores foram distribuídos de forma a cobrir os demais nós da rede como mostra a figura 2. Nesta figura encontram-se os monitores M_1 e M_2, que são os vizinhos ao nó intruso I, e conseqüentemente, os únicos que podem observar seu comportamento diretamente. Apesar dos nós serem cobertos por um ou mais monitores, a visão de cada um destes não é necessariamente a mesma. Considera-se na figura 2 P como sendo o pai do nó intruso I na árvore de roteamento e F como sendo o filho do intruso I na árvore de roteamento. O nó monitor M_1 é capaz de ouvir mensagens vindas do nó F, mas não é capaz de ouvir mensagens transmitidas por P, enquanto o monitor M_2 consegue ouvir as mensagens transmitidas por P, mas não as transmitidas por F. Dependendo do ataque simulado, um dos monitores conseguirá detectar um comportamento anômalo do nó observado e o outro não. Quando as análises feitas pelos nós monitores chegam à estação base, pode-se considerar todo o conjunto de informações e avaliar de forma mais ampla o comportamento da rede. Exemplos dessa avaliação mais ampla são mostrados nas análises dos resultados referentes aos ataques de *Jamming* e Repetição.

6. Experimentos e Resultados

Nesta seção são apresentados os resultados das simulações envolvendo o IDS proposto.

6.1. Considerações Iniciais

O objetivo dos experimentos abaixo foi avaliar o desempenho da solução de IDS proposta. Em particular, a eficácia na detecção dos ataques considerados e o número de falsos positivos obtidos pelos nós monitores. Entende-se por eficácia de detecção a porcentagem de ataques corretamente detectados pelo IDS.

Sob o ponto de vista do monitor, o processo de monitoração é dividido em etapas. A primeira etapa se inicia quando o vetor de avaliação está vazio e começa a ser preenchido com mensagens coletadas em escuta promíscua. Esta etapa termina quando o vetor está totalmente preenchido e o processamento das mensagens armazenadas pode ser disparado. O tamanho do vetor define a duração da etapa de escuta promíscua e, assim, a quantidade de mensagens que poderão ser relacionadas entre si em busca de indícios de intrusos na rede. A princípio, observa-se um compromisso entre o custo de armazenamento e eficácia na detecção. Quanto menor o vetor e, conseqüentemente, o custo de armazenamento, menor será a duração desta etapa e maiores serão as perdas de seqüência de mensagens, implicando em menor eficácia na detecção. No entanto, essa premissa não foi válida para o experimento com o ataque de Atraso de mensagens, o que será justificado durante sua análise. A fim de avaliar esse compromisso, variou-se o tamanho do vetor entre 30, 60, 100, 200 e 400 mensagens, para cada um dos ataques. Esses valores foram escolhidos com base nas restrições de memória do nó sensor Mica2 [Hill et al. 2000].

Todas as simulações tiveram duração de 10000 iterações, executadas 33 vezes para cada configuração. Considerou-se apenas um intruso aplicando um único tipo de ataque em cada simulação. A "etapa de aprendizagem" da rede durou 1000 iterações (10% do tempo de vida da rede), na qual nenhum ataque é aplicado sobre a rede, permitindo ao nó monitor avaliar a quantidade de falhas naturais que ocorreram nesse período. Logo após, o nó intruso inicia o seu ciclo de ataque em que promove ataques contra a rede durante aproximadamente 22% do seu tempo total de atividade. Esse valor foi escolhido empiricamente, de maneira a simplificar a obtenção dos resultados. Um estudo sobre sua variação e como esse valor afeta os demais parâmetros de observação foi deixado para trabalhos futuros. O IDS possui uma tolerância de 10% de falhas, o que significa que a razão entre as falhas observadas a cada etapa de processamento de mensagens pode ser até 10% maior que o número esperado de falhas naturais, sem que um indício de ataque seja gerado. Variou-se a probabilidade de ocorrência de falhas naturais em 1%, 10% e 20%.

A seguir são apresentados os resultados relativos à eficácia na detecção e número de falsos positivos. O número de falsos positivos é mostrado em números absolutos, obtidos pelos monitores M_1 e M_2 (vizinhos do intruso I) ao final da simulação.

6.2. Eficácia na Detecção

Um aspecto comum observado em quase todos os ataques foi o fato de um tamanho menor de vetor de mensagens no monitor resultar em um número maior de falsos positivos detectados. Isso ocorreu porque para análises feitas com vetores menores o número esperado de falhas era impreciso, já que um menor número de mensagens foi capturado entre as etapas de processamento. Essa imprecisão levou o IDS a indicar falsos positivos quando o número de falhas naturais esteve um pouco acima da taxa esperada para a rede considerada. Variações do tamanho do vetor também impactaram no nível de detecção, assim como na variação da probabilidade de falhas naturais. A utilização de vetores menores faz com que a seqüência de mensagens coletadas sejam cada vez menores, afetando a detecção. Levando-se em conta esses parâmetros, são discutidos a seguir os resultados obtidos para cada tipo de ataque simulado.

6.2.1. Repetição, Atraso e *Wormhole*

A eficácia na detecção de intrusos e o número de falsos positivos para os ataques de Repetição, Atraso e *Wormhole* são mostrados nas figuras 3 a 8. Como os ataques não

são confundidos com nenhum tipo de falha natural da rede pelo nós monitores, percebe-se que a detecção de intrusos e o número de falsos positivos não são influenciados pela variação de taxa de ocorrência de falhas naturais na rede. A detecção do ataque de Atraso é diretamente proporcional ao tamanho do vetor, já que com um tamanho de vetores menor o tempo gasto para preenchê-lo pode ser menor que o momento em que o próximo ataque será realizado, fazendo com que o IDS não o detecte. Isso explica o baixo desempenho obtido nos experimentos com vetores de 30 e 60 mensagens. Nos experimentos com ataques de Repetição, a taxa de detecção ficou acima de 90% para todos os casos. Nos ataques de *Wormhole*, 100% das tentativas de intrusão foram detectadas em todos os cenários, visto que o nó monitor sempre verificava o canal de comunicação antes de enviar uma mensagem.

Como mostrado a seguir, o número de falsos positivos para estes ataques foram menores que os obtidos nos demais experimentos, já que neste caso os ataques não são confundidos com falhas naturais da rede. Percebe-se também que no ataque de *Wormhole*, utilizando vetores de pelo menos 100 mensagens, foram obtidos menos de 70 falsos positivos reportados pelos IDS em 10000 iterações de simulação, o que significa aproximadamente um falso positivo a cada 140 iterações. A presença de falsos positivos nesse tipo de ataque é devido à consideração errônea, por parte de um nó monitor, de alguma falha vinda de um nó que não é intruso.

No ataque de Repetição, o monitor M_2 reportou falsos positivos ao detectar como intruso o nó P, pai do intruso I. Isso se deve ao fato de que no modelo proposto de rede não foi considerado nenhum tipo de supressão de mensagens repetidas. Dessa maneira, o nó P simplesmente retransmite as mensagens repetidas que recebe de I.

No ataque de Atraso de mensagens, obteve-se um maior número de falsos negativos porque o monitor captura mensagens consideradas atrasadas quando seu vetor de monitoração está quase cheio. Sendo assim, as mensagens supostamente atrasadas só são recebidas pelo monitor no início da próxima fase de aquisição de dados.

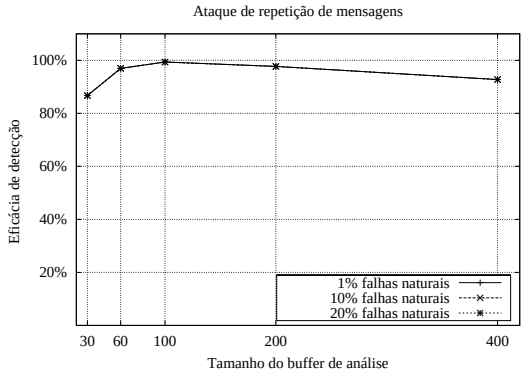

Figura 3. Eficácia na detecção do ataque de Repetição de mensagem

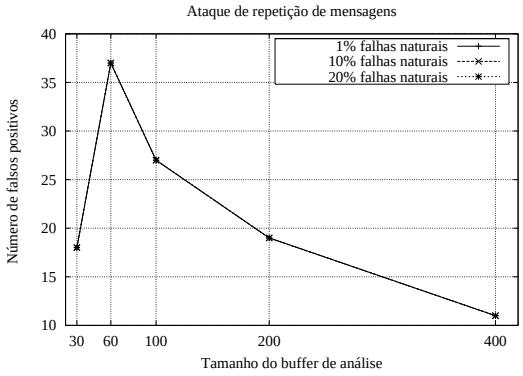

Figura 4. Falsos positivos no ataque de Repetição de mensagem

6.2.2. Alteração de Dados

A eficácia na detecção e o número de falsos positivos para o ataque de Alteração de dados são mostrados nas figuras 9 e 10. É importante lembrar que o ataque de Alteração de dados pode ser confundido com a falha natural de Alteração de dados pelo nó monitor.

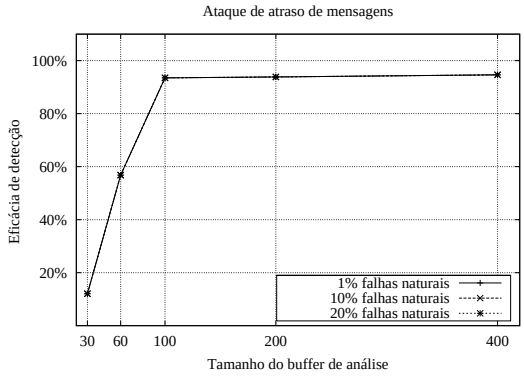

Figura 5. Eficácia na detecção do ataque de Atraso de mensagem

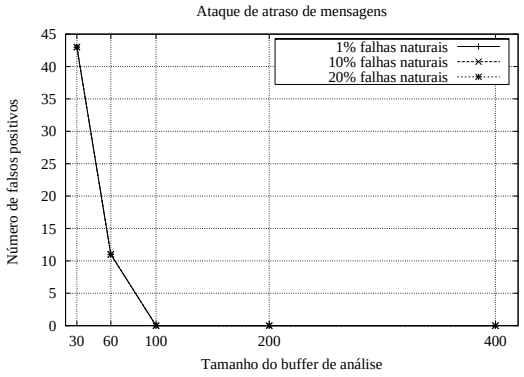

Figura 6. Falsos positivos no ataque de Atraso de mensagem

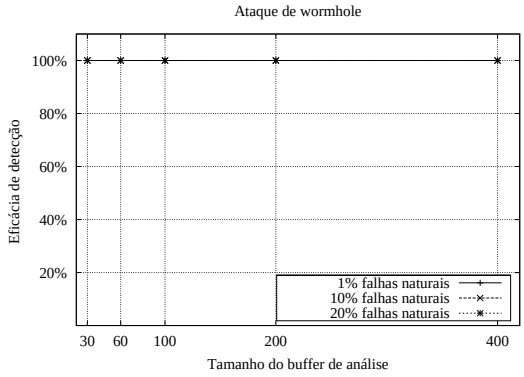

Figura 7. Eficácia na detecção do ataque de *Wormhole*

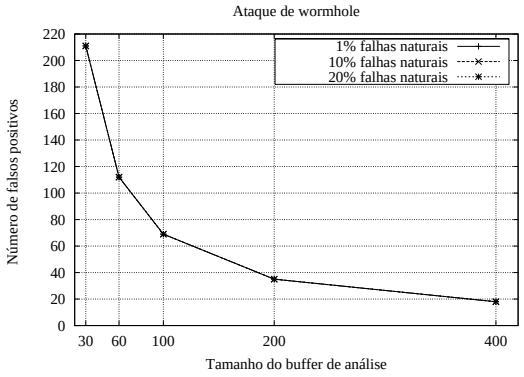

Figura 8. Falsos positivos no ataque de *Wormhole*

Neste tipo de ataque, monitores com vetores menores apresentam taxas de detecção mais altas. A estimação da média de falhas naturais para esse tipo de ataque é menos sensível que as médias relacionadas a ataques de perdas de mensagens, onde cada falha também diminui o número total de mensagens coletadas (e, conseqüentemente, usadas para calcular esta média). Por isso, este ataque apresenta maior eficácia quando monitores com vetores menores são utilizados. Por outro lado, vetores menores levam o IDS a reportar mais falsos positivos, visto a maior imprecisão nesse mesmo cálculo.

6.2.3. *Blackhole*, *Selective Forwarding* e Negligência de Dados

A eficácia de detecção e número de falsos positivos na análise dos ataques de *Blackhole*, *Selective forwarding* e Negligência são apresentados nas figuras 11 a 16. Como observado na seção 3., os três ataques podem ser confundidos com falhas naturais de perda de mensagem. Assim como no caso do ataque de Alteração de dados, o monitor M_1 detectou as tentativas de ataque do nó intruso I, enquanto o monitor M_2 não observou nenhuma anomalia na rede.

Observa-se que em cenários com vetores acima de 60 mensagens, são obtidas taxas de detecção acima de 80%, o que demontra a eficácia do sistema proposto.

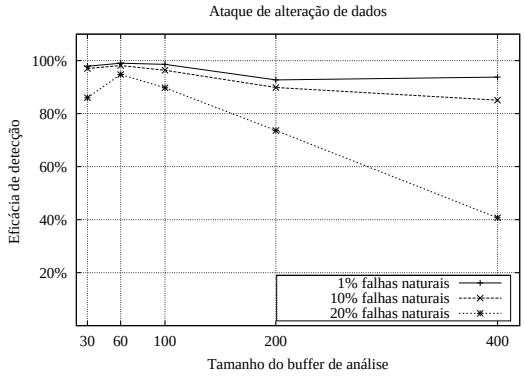

Figura 9. Eficácia na detecção do ataque de Alteração de dados

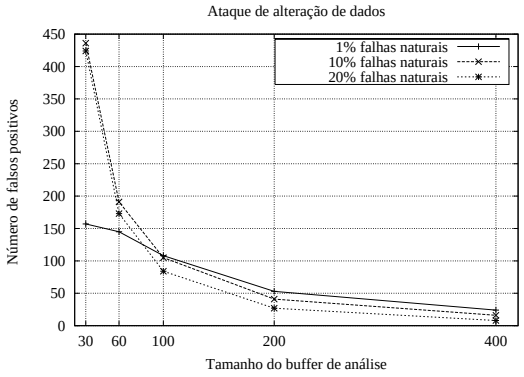

Figura 10. Falsos positivos no ataque de Alteração de dados

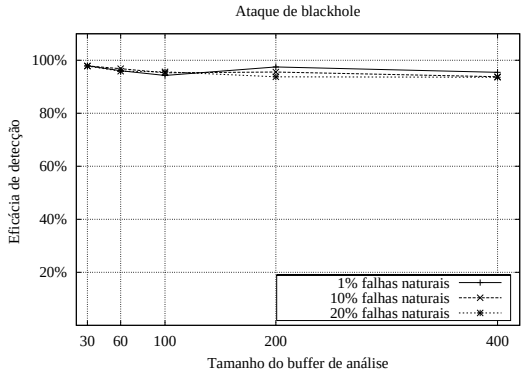

Figura 11. Eficácia na detecção do ataque de *Blackhole*

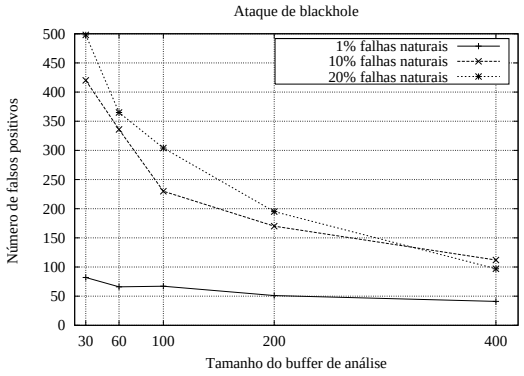

Figura 12. Falsos positivos no ataque de *Blackhole*

6.2.4. Jamming

As figuras 17 e 18 apresentam, respectivamente, os resultados da análise de eficácia na detecção e o número de falsos positivos para o ataque de *Jamming*. Devido à semelhança com a falha natural de colisão de mensagens, o ataque de *Jamming* pode ser confundido com esta pelo IDS, resultando em um falso positivo. Apesar disso, resultados demonstram que o IDS proposto apresentou taxas de detecção sempre acima de 90%. Observa-se também que o número de falsos positivos diminui à medida que o tamanho do vetor aumenta.

Ao mesmo tempo que um ataque de *Jamming* foi detectado, demais monitores da rede acusaram nós comuns de estarem promovendo ataques de *Blackhole*, *Selective forwarding* e Negligência de dados. Isso ocorreu devido à indisponibilidade do canal de comunicação, o que impede que esses nós enviem suas próprias mensagens ou retransmitam mensagens recebidas.

Apesar do ataque ser detectado corretamente por um nó monitor, a identificação do intruso não pode ser feita de maneira direta. Pode-se contornar esse problema por meio da observação dos nós inocentes que são acusados de ataques de *Blackhole*, *Selective forwarding* e Negligência de dados. Sabendo quem são eles, identificam-se os vizinhos do nó intruso e a área onde ele se encontra.

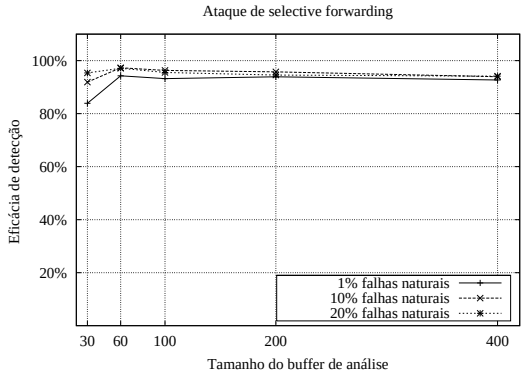

Figura 13. Eficácia na detecção do ataque de *Selective forwarding*

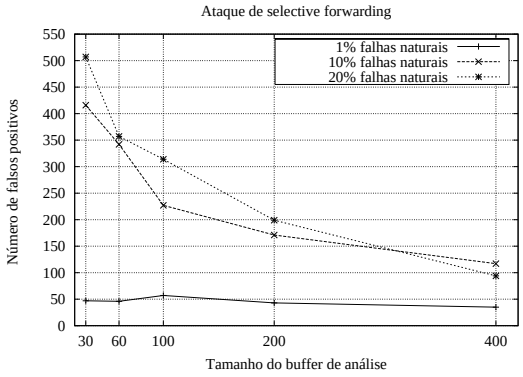

Figura 14. Falsos positivos no ataque de *Selective forwarding*

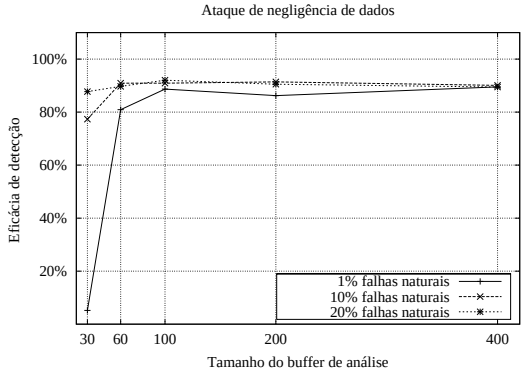

Figura 15. Eficácia na detecção do ataque de Negligência de dados

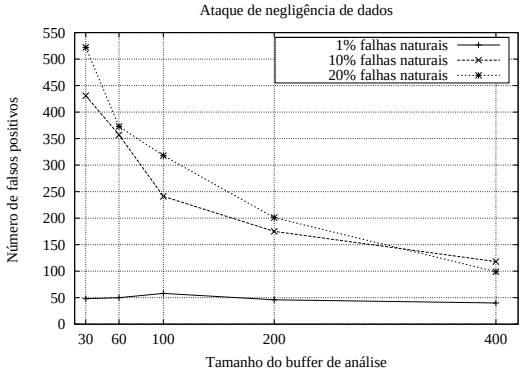

Figura 16. Falsos positivos no ataque de Negligência de dados

6.3. Consumo de Energia

Nos experimentos realizados, considerou-se o consumo de energia nas seguintes situações: recepção, transmissão e verificação de mensagens feitas por cada nó da rede. Entende-se por recepção o processamento completo da mensagem que chega ao nó, tal como faz o nó comum, quando a mensagem é endereçada a ele, e o monitor, com todas as mensagens que o alcançam. Entende-se por verificação a leitura e verificação do cabeçalho da mensagem, que é desprezada caso não seja endereçada ao nó que a recebe. Fazendo isso, pode-se reduzir o consumo de energia do nó, aumentando seu tempo de vida na rede.

Foram utilizadas mensagens de 36 bytes – mesmo tamanho utilizado em aplicações do TinyOS [TinyOS], sendo 2 bytes correspondentes ao endereço de destino imediato da mensagem (campo considerado na verificação), e taxa de transmissão de dados de 0.26 $\mu s/bit$, como definida em [Shnayder et al. 2004]. Considerando a corrente despendida no nó durante a recepção (7,0 mA) e transmissão de mensagens (21,5 mA, na maior potência), valores também extraídos de [Shnayder et al. 2004], pode-se calcular o consumo de energia em cada uma das situações:

- $Q_{Transmissao}$ = 3 × 21.5 mA × 0.26 ×$10^{-6} s/bit$× 288 $bits$ = 0.48375 $mJ/messagem$;
- $Q_{Recepcao}$ = 3 × 7.0 mA × 0.26 ×$10^{-6} s/bit$× 288 $bits$ = 0.1575 $mJ/messagem$;
- $Q_{Verificacao}$ = 3 × 7.0 mA × (0.26 ×$10^{-6} s/bit$× 16 $bits$ = 0.00875 $mJ/messagem$;

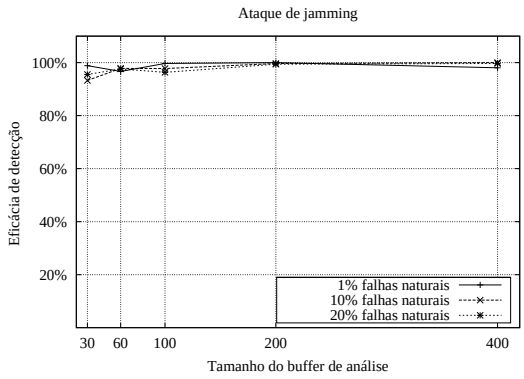

Figura 17. Eficácia na detecção do ataque de *Jamming*

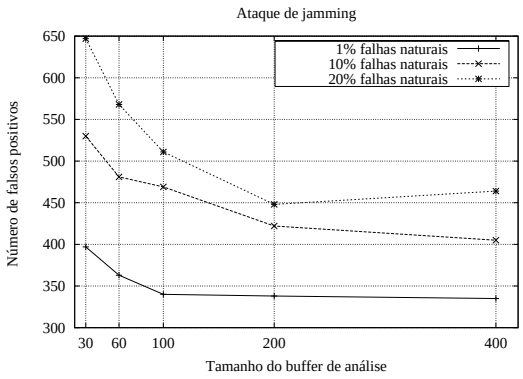

Figura 18. Falsos positivos no ataque de *Jamming*

onde Energia Dissipada (Q) = Diferença de potencial $\times$ Corrente $\times$ Tempo e Tempo = Taxa de transmissão $\times$ Tamanho da mensagem.

Com o objetivo de medir o consumo de energia no modelo de rede proposto, dois cenários foram simulados. No primeiro cenário, nós monitores não foram distribuídos na rede, enquanto no segundo os módulos IDSs foram instalados em alguns nós da rede. Em ambos os cenários, a topologia e posição dos nós foram as mesmas (figura 2). Como mostrado nas figuras 19 e 20, os nós monitores consomem mais energia do que os nós comuns correspondentes no primeiro cenário. Durante o funcionamento do IDS, um nó monitor escuta e recebe completamente todas as mensagens de seus vizinhos, incluindo aquelas não endereçadas a ele, o que explica o maior consumo de energia. Os nós monitores que mais consumiram energia foram aqueles mais próximos da estação base.

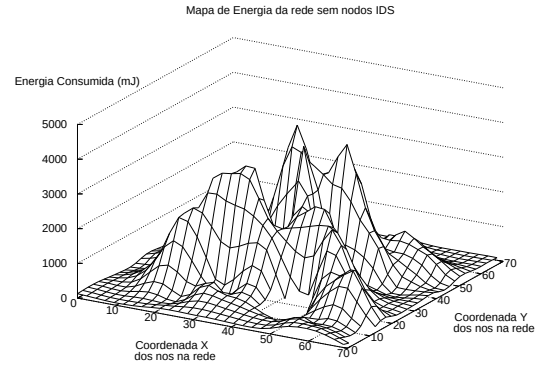

Figura 19. Consumo de energia da rede sem IDS

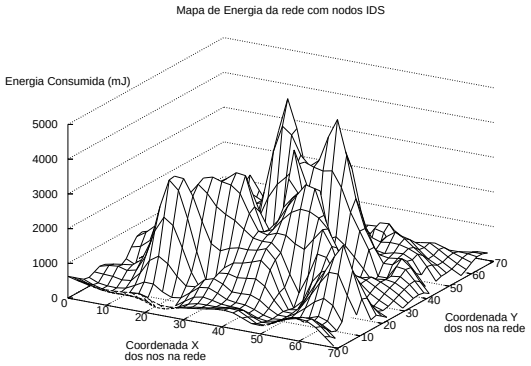

Figura 20. Consumo de energia da rede com IDS

Não foi considerado o consumo de energia no processamento de mensagens pelo IDS e nós comuns, o que poderia nos dar uma melhor aproximação de um cenário real. Isso será tratado em trabalhos futuros.

7. Conclusão

O objetivo deste trabalho foi apresentar um estudo sobre a detecção de intrusos em RSSFs e propor um IDS que atendesse às suas demandas e restrições. Desenvolvemos um IDS

baseado em especificação [Ko et al. 1997, Balepin et al. 2003], já que as características das RSSFs podem variar de acordo com o objetivo da aplicação a qual se destinam.

A detecção é descentralizada, pois os IDSs são distribuídos na rede ao serem instalados em nós monitores. A coleta de informações e seu processamento são feitos de forma distribuída. Sistemas de detecção de intrusos distribuídos são mais escaláveis e robustos. Como consideram vários pontos de vista da rede, é mais difícil do intruso não ser detectado. Resultados de simulação demonstram que o sistema apresentado obteve níveis de detecção acima de 90% em aproximadamente 83% dos cenários testados, mesmo considerando a presença de falhas naturais na rede.

Referências

Arora, A., Dutta, P., Bapat, S., Kulathumani, V., Zhang, H., Naik, V., Mittal, V., Cao, H., Demirbas, M., Gouda, M. G., ri Choi, Y., Herman, T., Kulkarni, S. S., Arumugam, U., Nesterenko, M., Vora, A., and Miyashita, M. (2004). A line in the sand: a wireless sensor network for target detection, classification, and tracking. *Computer Networks*, 46(5):605–634.

Balepin, I., Maltsev, S., Rowe, J., and Levitt, K. (2003). Using specification-based intrusion detection for automated response. In *Proceeding of the 6th International Symposium*, Pittsburgh, PA. RAID 2003, Recent Advances in Intrusion Detection.

da Silva, A. P. R. (2005). Detecção de intrusos descentralizada em redes de sensores sem fio. *Dissertação de Mestrado - DCC - UFMG - Brasil*.

da Silva, A. P. R., Teixeira, F. A., Wong, H. C., and Nogueira, J. M. S. (2004). Aspectos de detecção de intrusos em redes de sensores sem fio (short paper). In *22º Simpósio Brasileiro de Redes de Computadores*, pages 575 – 578, Gramado, RS.

Deng, J., Han, R., and Mishra, S. (2003). A performance evaluation of intrusion-tolerant routing in wireless sensor networks. In *Proceedings of IEEE 2nd International Workshop on Information Processing in Sensor Networks (IPSN '03)*, pages 349–364, Palo Alto, California. IEEE.

Fall, K. and Varadhan, K. (2001). The *ns* manual. http://www.isi.edu/nsnam/ns/doc/.

Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D. E., and Pister, K. S. J. (2000). System architecture directions for networked sensors. In *Architectural Support for Programming Languages and Operating Systems*, pages 93–104.

Hu, Y.-C., Perrig, A., and Johnson, D. B. (2003). Packet leashes: A defense against wormhole attacks in wireless networks. In *Proceedings of IEEE Infocomm 2003*.

Huang, M.-Y., Jasper, R. J., and Wicks, T. M. (1999). A large scale distributed intrusion detection framework based on attack strategy analysis. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(23–24):2465–2475.

Ilgun, K., Kemmerer, R. A., and Porras, P. (1995). State transition analysis: A rule-based intrusion detection approach. *IEEE Transactions on Software Engineering*, 21(3):181–199.

Jr., W. R. P., Figueiredo, T. H. P., Wong, H. C., and Loureiro, A. A. F. (2004). Malicious node detection in wireless sensor networks. In *18th International Parallel and Distributed Processing Symposium*, Santa Fe, NM.

Karlof, C., Sastry, N., and Wagner, D. (2004). Tinysec: a link layer security architecture for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys '04)*, pages 162–175.

Karlof, C. and Wagner, D. (2003). Secure routing in wireless sensor networks: Attacks and countermeasures. In *Proceedings of First IEEE International Workshop on Sensor Network Protocols and Applications*. IEEE.

Ko, C., Ruschitzka, M., and Levitt, K. (1997). Execution monitoring of security-critical programs in distributed systems: a specification-based approach. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, pages 175–187. IEEE Computer Society.

Kumar, S. and Spafford, E. H. (1995). A software architecture to support misuse intrusion detection. In *Proceedings of the 18th National Information Security Conference*, pages 194–204.

Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., and Anderson, J. (2002). Wireless sensor networks for habitat monitoring. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97, New York, NY, USA. ACM Press.

Marti, S., Giuli, T. J., Lai, K., and Baker, M. (2000). Mitigating routing misbehavior in mobile ad hoc networks. In *Mobile Computing and Networking*, pages 255–265.

Martins, M. H. T., da Silva, A. P. R., Loureiro, A. A. F., and Ruiz, L. B. (2005). Simulador para um sistema de detecção de intrusos em redes de sensores sem fio. Relatório Técnico RT.DCC.008/2005, DCC - UFMG.

Park, S., Savvides, A., and Srivastava, M. B. (2000). Sensorsim: a simulation framework for sensor networks. In *Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems (MSWIM '00)*, pages 104–111.

Paxson, V. (1999). Bro: a system for detecting network intruders in real-time. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(23–24):2435–2463.

Perrig, A., Szewczyk, R., Tygar, J. D., Wen, V., and Culler, D. E. (2002). Spins: security protocols for sensor networks. *Wireless Network Journal (WINE)*, 8(5):521–534.

Porras, P. A. and Neumann, P. G. (1997). Emerald: Event monitoring enabling responses to anomalous live disturbances. In *Proceedings of 20th NIST-NCSC National Information Systems Security Conference*, pages 353–365.

Ruiz, L. B., Nogueira, J. M. S., and Loureiro, A. A. F. (2003). MANNA: a management architecture for wireless sensor networks. *IEEE Communications Magazine*, 41(2):116–125. ISSN 0163-6804.

Segall, A. (1983). Distributed network protocols. *IEEE Transactions on Information Theory*, 29:23–35.

Shnayder, V., Hempstead, M., rong Chen, B., Allen, G. W., and Welsh, M. (2004). Simulating the power consumption of large-scale sensor network applications. In *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys '04)*, pages 188–200.

TinyOS. A component-based os for the networked sensor regime. http://www.tinyos.net.

Wood, A. D. and Stankovic, J. A. (2002). Denial of service in sensor networks. *IEEE Computer*, 35(10):54–62.