Detecção de Intrusão em Redes de Sensores Sem Fio utilizando uma abordagem colaborativa e *cross-layer*

Marcus Vinícius de Sousa Lemos[1], **Liliam Barroso Leal**[1], **Raimir Holanda Filho**[1]

[1]Mestrado em Informática Aplicada – Universidade de Fortaleza (Unifor)
Fortaleza – CE – Brazil

{marvin,liliam}@edu.unifor.br, raimir@unifor.br

Abstract. *This paper proposes a new collaborative and decentralized approach for intrusion detection systems on Wireless Sensor Networks. Special nodes, named monitors, will be responsible for monitoring the behavior of neighbor nodes and the malicious activities evidences discovered will be shared and correlated with the purpose of increasing the accuracy in intruders detection. In addition, through a cross-layer approach, the proposed IDS increases the survivability of the network since the detected malicious nodes are isolated in a way that they can not cause more damage. Experiments conducted by simulation have shown that our approach is efficient in terms of security as well as have proved to be feasible from the energy consumption viewpoint.*

Resumo. *Este artigo propõe uma nova abordagem colaborativa e descentralizada para sistemas de detecção de intrusão (IDS) em Redes de Sensores Sem Fio (RSSF). As evidências de atividades maliciosas, descobertas pelos nós monitores, serão compartilhadas e correlacionadas com o propósito de aumentar a precisão na detecção de intrusos. Além disso, através de uma abordagem cross-layer, o IDS proposto influencia o protocolo de roteamento de forma que os nós maliciosos detectados sejam isolados e não possam mais causar prejuízos à rede. Experimentos realizados através de simulação demonstraram que nossa abordagem, além de eficiente do ponto de vista da segurança, apresentou-se viável sob a ótica do consumo de energia.*

1. Introdução

Tipicamente formada por centenas de pequenos dispositivos operados por baterias, as Redes de Sensores Sem Fio (RSSF) utilizam comunicação sem fio de baixo alcance, além de possuírem severas restrições de vários outros recursos como, por exemplo, energia, largura de banda, capacidade de processamento e armazenamento. Muitas vezes esses mesmos nós são espalhados em uma região de difícil acesso, tornando complicado, ou mesmo impossível, a reposição de um nó danificado ou de uma bateria esgotada.

Devido a sua capacidade de sensoriamento, as RSSF têm aplicabilidade em diversas áreas, tais como monitoramento ambiental, sistemas de vigilância e saúde. Podemos perceber, ainda, que muitas das aplicações das RSSF são de missão crítica, tornando-se alvo para possíveis adversários interessados em prejudicar o nível de sensoriamento ou até mesmo esgotar os recursos da rede (como energia), tornando-a inativa. Esse fato ainda é agravado pela própria natureza da deposição da rede. Muitas vezes, os nós sensores

são depositados em regiões remotas ou hostis, tornando-os desprotegidos e suscetíveis a ataques físicos [Alzaid et al. 2008]. Dessa forma, é imperativo que as redes sejam depositadas com algum esquema de segurança. Contudo, o consumo adicional de energia causado pela execução das funções de segurança deve ser recompensado pela economia obtida ao evitar ataques.

O mecanismo de prevenção é a primeira linha de defesa em uma rede, garantindo alguns princípios de segurança como, por exemplo, confidencialidade, autenticação e integridade. Entretanto, a prevenção, principalmente nas RSSF, não é suficiente para garantir a segurança da rede. Como muitas vezes os nós sensores são depositados em áreas abertas e desprotegidas, é possível que um atacante tenha acesso físico a um nó sensor e consiga acessar seus dados armazenados (chaves criptográficas, por exemplo). Assim, percebe-se a importância de um Sistema de Detecção de Intrusão (IDS) capaz de detectar possíveis nós maliciosos que consigam burlar os esquemas de prevenção da rede. Além de prevenir que um atacante possa causar maiores estragos na rede, o sistema de detecção de intrusão pode ser usado para coletar informações relacionadas às técnicas de ataques, ajudando, dessa forma, no desenvolvimento de sistemas de prevenção [Silva et al. 2005].

Em um IDS, os nós responsáveis pela função de monitoramento são chamados monitores (ou agentes) IDS. E, dentro de uma rede sem fio, comportam-se como *watchdogs* [Marti et al. 2000], capturando e analisando os pacotes encaminhados pelos seus vizinhos.

Neste trabalho, propomos um IDS capaz de mitigar atividades maliciosas dentro da rede. As principais características do IDS são:

- As inferências realizadas por cada monitor serão correlacionadas de forma a aumentar a precisão na detecção dos ataques;
- Através de uma abordagem *cross-layer* [Zhou et al. 2005a], a rede poderá reagir de forma a eliminar os nós maliciosos detectados.

Em [Silva et al. 2005] foi mostrado que há possibilidade de ocorrências de falsos positivos devido a uma falta de correlação entre as inferências realizadas por cada monitor. Com o intuito de reduzir, ou mesmo eliminar, essa quantidade significativa de falsos positivos, propomos um Sistema Colaborativo e Descentralizado de Detecção de Intrusão. Nesse sistema, os monitores serão organizados colaborativamente em uma arquitetura baseada em Tabelas *Hash* Distribuídas (*Distributed Hash Tables* - DHT) onde as inferências locais de cada monitor serão correlacionadas por monitores especiais, denominados supervisores, com o objetivo de verificar se há alguma relação entre as atividades suspeitas detectadas.

Após o processo de correlação, os intrusos detectados deverão ser eliminados de forma que não possam causar mais prejuízos à rede. Para isso, escolhemos utilizar uma abordagem *cross-layer* [Zhou et al. 2005a], de forma que o IDS proposto possa influenciar a escolha da rota utilizada pelo protocolo de roteamento baseado nas ações maliciosas detectadas. Entretanto, nossa solução não depende do protocolo de roteamento. É necessário apenas que tal protocolo forneça duas funcionalidades: (1) Roteamento *multipath* e (2) uma interface para que a aplicação possa especificar a rota a ser utilizada.

O restante do artigo é definido a seguir. A Seção 2 descreve os trabalhos relacionados. A Seção 3 apresenta o IDS Colaborativo e Descentralizado proposto enquanto

a Seção 4 ilustra, com mais detalhes, a forma de colaboração entre os monitores. Na Seção 5 descrevemos a simulação realizada para avaliar o sistema proposto e, na Seção 6, os resultados são apresentados. Por fim, na Seção 7 temos as conclusões com alguns direcionamentos futuros.

2. Trabalhos Relacionados

Em [Roman et al. 2006] foi proposto uma técnica conhecida como *Spontaneous Watchdogs*. Essa técnica é eficiente em redes com alta quantidade de nós sensores depositados na região. Para cada pacote circulando na rede, há um conjunto de nós que são capazes de receber esse pacote, além do pacote retransmitido pelo seu próximo *hop*. Conseqüentemente, todos os nós têm a chance de ativar seus agentes globais de forma a monitorar estes pacotes.

Em [Júnior et al. 2004] foi proposto um mecanismo baseado em potência de sinal e informações geográficas para detectar nós maliciosos que estejam realizando ataques de *Hello Flooding* e *Wormholes*. A detecção é realizada através da comparação da potência do sinal recebido com o seu valor esperado. O valor esperado é calculado utilizando-se as informações geográficas e a configuração do transceptor. Além disso, foi proposto um protocolo para disseminar informações sobre os nós maliciosos. A grande desvantagem dessa proposta é justamente limitar-se a apenas dois tipos de ataques (*Hello flooding* e *Wormholes*).

Em [Li et al. 2008] foi proposto um IDS distribuído baseado em grupos. A rede é dividida em vários grupos, onde cada grupo é composto por nós que estão próximos uns aos outros e compartilham a mesma capacidade de sensoriamento. Cada grupo será escalonado a executar o algoritmo de IDS. Durante a execução de um determinado grupo, cada sensor irá monitorar o comportamento dos nós do mesmo grupo. Caso o sensor perceba algum nó com comportamento anormal, encaminhará um aviso para todos os nós acerca desse possível nó malicioso. Se a quantidade de avisos acerca desse possível nó malicioso atingir um limite, a rede então conclui que o nó realmente está fazendo alguma atividade ilícita.

Em [Silva et al. 2005] foi proposto um sistema de detecção de Intrusão descentralizado. A partir das características especificas da RSSF alvo, fornecida pelo projetista da rede, pode-se selecionar regras capazes de detectar possíveis ataques relacionados a essas características. Essas regras serão aplicadas pelos monitores espalhados pela rede aos pacotes transmitidos pelos nós vizinhos. Outra principal característica relacionada a esse IDS é a quantidade de ataques que pode detectar: Buraco Negro (*Blackhole*), Retransmissão Seletiva (*Selective Forwarding*), Repetição, Atraso, Alteração de Dados, Interferência, Canalização, Negligência e Exaustão. Contudo, não há cooperação entre os monitores, podendo gerar falso-positivos e falso-negativos.

Nosso trabalho é diferente dos citados acima pois utilizamos uma abordagem colaborativa para diminuir a taxa de falso-positivos e falso-negativos. Nós especiais, denominados *Supervisores*, serão responsáveis por correlacionar as evidências geradas pelos monitores. Além disso, através da abordagem *cross-layer*, a resistência da rede aumentará uma vez que não haverá mais comunicação entre os nós comuns e os nós maliciosos detectados.

3. O IDS Colaborativo e Descentralizado Proposto

O monitoramento da rede será realizado, de modo distribuído, por nós denominados monitores. Cada monitor, localizado em algum ponto da rede, será responsável por monitorar uma sub-parte da rede, mais precisamente os nós vizinhos a ele. Assim, a partir do tráfego dos nós vizinhos, o monitor poderá inferir quais os nós estão comportando-se fora do padrão. Essa inferência é possível através de um conjunto de regras que especificam qual o comportamento esperado dos nós que compõe a rede. Para isso, utilizamos as regras definidas em [Silva et al. 2005]. Abaixo, listamos esse conjunto de regras com uma breve descrição:

1. **Regra Intervalo**: uma falha é detectada se o atraso entre a chegada de duas mensagens consecutivas é menor ou maior que o permitido.
2. **Regra Retransmissão**: uma falha é detectada se um nó não reencaminhou uma mensagem quando deveria.
3. **Regra Integridade**: os dados devem permanecer inalterados na retransmissão.
4. **Regra Atraso**: as mensagens devem ser retransmitidas dentro de um prazo mínimo de tempo.
5. **Regra Repetição**: uma falha é detectada se uma mensagem é transmitida mais de uma vez ou um limite máximo de vezes.
6. **Regra Alcance de Rádio**: uma falha é detectada caso o monitor receba uma mensagem de um nó que não tem alcance de rádio suficiente.
7. **Regra Destinos Válidos**: verifica se os destinos são válidos
8. **Regra Origens Válidas**: verifica se as origens são válidas
9. **Regra de Interferência**: verifica se o monitor está conseguindo enviar suas próprias mensagens.

Como cada monitor possui apenas uma visão parcial da rede (apenas seus nós vizinhos), uma quantidade significativa de falso-positivos ou falso-negativos pode ser gerada [Silva et al. 2005]. Assim, propomos um modelo colaborativo onde as inferências de cada monitor serão correlacionadas com o objetivo de eliminar a ocorrência desses falsos positivos e falsos negativos. A Figura 1 ilustra uma situação onde ocorreu um falso positivo devido a falta de uma colaboração entre os monitores. Supondo que na rede não exista nenhum tratamento de supressão de mensagens repetidas, o nó N será acusado de ataque do tipo Repetição por parte do monitor M1, apesar de não estar executando o ataque. Como está na rota do verdadeiro atacante, nó I, o nó N simplesmente repassa as mensagens repetidas que recebe. O monitor M2 é vizinho de I e está detectando-o como malicioso. Caso M2 pudesse, de alguma maneira, informar aos outros monitores que o nó I está realizando o ataque, seria fácil inferir que N não é malicioso, pois o mesmo está na rota de I.

Além disso, através de uma abordagem *cross-layer*, o IDS proposto influencia o protocolo de roteamento dos nós na escolha das rotas a ser utilizada. Baseado nas informações relacionadas às atividades maliciosas detectadas, os nós da rede podem simplesmente descartar qualquer pacote originado de um nó malicioso bem como não utilizar esse mesmo nó como rota.

No nosso sistema, consideraremos os seguintes tipos de nós:

- **Básico**: nó que possui a capacidade de sensoriamento. Capta informações do meio em que se encontra e encaminha até a estação base.

- **Monitor**: responsável pela monitoração de seus vizinhos através da escuta em modo promíscuo. Armazena informações de interesse, processando esses dados conforme as regras especificadas.
- **Supervisor**: monitor especial que, além da monitoração dos seus vizinhos, é capaz de correlacionar evidências descobertas por outros monitores. Cada supervisor é responsável por correlacionar um sub-conjunto específico de regras.
- **Intruso**: nó que irá realizar os ataques dentro da rede. Dentro deste trabalho, consideramos apenas ataques do tipo Retransmissão.

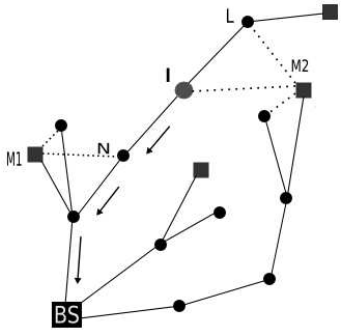


Figura 1. Falso Positivo: Ataque de Repetição

Suponha, por exemplo, uma rede onde exista um supervisor denominado S1 responsável pela regra de Repetição. Caso dois outros monitores, M2 e M3, descubram que algum nó vizinho está violando a regra de repetição, deverão encaminhar essas informações para S1. Assim, S1 poderá verificar se existe alguma correlação entre essas atividades suspeitas, podendo, conseqüentemente, inferir com mais precisão a origem do ataque. A comunicação entre os monitores será realizada utilizando um esquema de criptografia (SNEP [Perrig et al. 2002], por exemplo) de forma que nós comuns ou atacantes não possam ter acesso ao conteúdo da mensagem.

O processo de colaboração é divido em cinco fases. A cada pacote interceptado, o monitor armazena-o em *buffer* de tamanho fixo (Fase 1) e quando esse *buffer* estiver cheio as regras serão aplicadas (Fase 2). Nesta fase, cada pacote será analisado levando-se em conta as regras selecionadas.

Caso alguma regra seja violada numa freqüência maior que a esperada devido às falhas naturais da rede, um indício de comportamento anormal é gerado (Fase 3). Os indícios detectados por cada monitor serão correlacionados pelos supervisores (Fase 4) com o objetivo de verificar se há alguma relação entre essas atividades de forma que um nó não seja falsamente acusado.

Após a correlação dos indícios, a fase 5 é iniciada. Os supervisores informam aos monitores a relação dos nós que foram confirmados como maliciosos e os monitores devem repassar essas informações aos seus nós vizinhos. Dessa forma, através de uma interface definida pelo protocolo de roteamento, a aplicação executada em cada nó vizinho poderá especificar que nenhum pacote proveniente do nó malicioso deverá ser processado, bem como nenhuma rota que utilize tal nó deverá ser criada ou usada.

4. Colaboração entre os monitores

Conforme visto na Seção 3, nossa proposta baseia-se na colaboração entre os monitores de forma que as evidências descobertas por cada um possam ser correlacionadas. Assim, baseado no trabalho definido em [Zhou et al. 2005b], todo os nós monitores serão organizados através de uma arquitetura baseada em Tabelas *Hash* Distribuídas, onde cada monitor poderá assumir a função de supervisor responsável por correlacionar eventos relacionados a uma ou mais regras. Apresentamos, na subseção a seguir, com mais detalhes, o sistema de colaboração.

4.1. O processo de Colaboração

Cada rede monitorada por um IDS será formada por um conjunto de monitores $M = \{m_i | i = 1, 2, \ldots, p\}$, onde cada monitor é responsável por vigiar seus vizinhos. Definimos R como sendo o conjunto de regras selecionadas para o sistema de detecção, $R = \{r_j | j = 1, 2, \ldots, q\}$ que todos os monitores, $m_i \in M$, deverão aplicar a seus nós vizinhos. Definimos R'_i como sendo um subconjunto de R associado a cada m_i. Se $R'_i \neq \emptyset$ então o monitor m_i assumirá a função de supervisor, onde deverá correlacionar eventos relacionados à cada regra $r_j \in R'_i$ reportada pelos outros monitores. No nosso sistema, cada monitor será capaz de desempenhar a função de supervisor, mas apenas se tiver alguma regra associada a ele ($R'_i \neq \emptyset$).

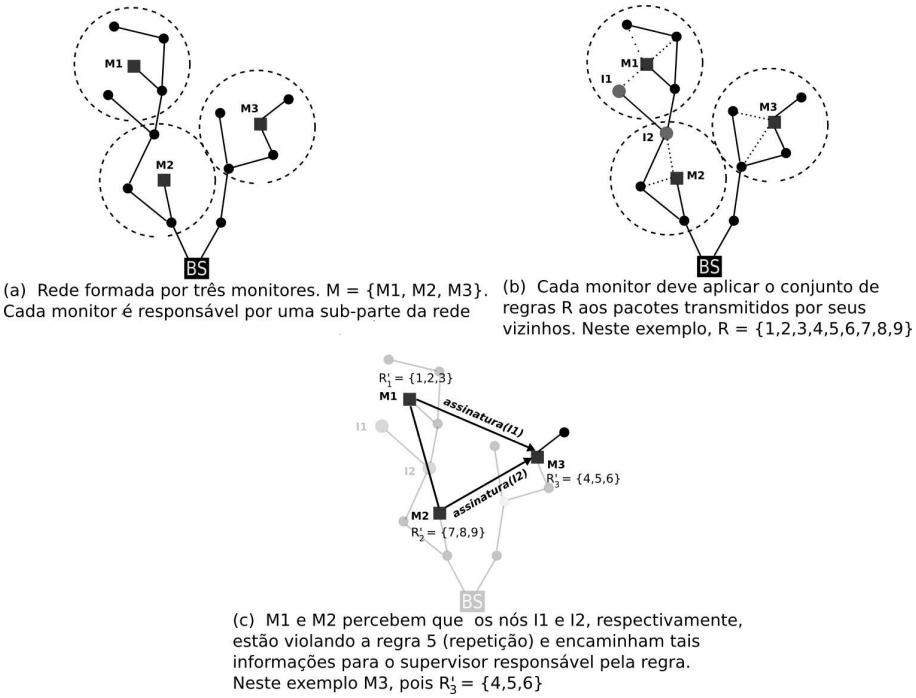
O compartilhamento das informações entre os monitores e supervisores será realizado através de um mecanismo *publish/subscribe* devido às suas vantagens em relação à nossa proposta. Este mecanismo possui as seguintes propriedades [Karl and Willig 2005]:

- **Desacoplados no Espaço**: os monitores não precisam estar cientes dos supervisores, de fato eles podem esquecer completamente de seus identificadores.
- **Desacoplados no Tempo**: publicação e notificação de dados podem acontecer em diferentes períodos de tempo.
- **Desacoplados no Fluxo**: interações entre os nós podem acontecer assincronamente sem bloqueio.

Quando a fase de detecção de intrusão (Fase 3) termina, cada monitor $m_i \in M$ deverá fazer uma assinatura, sobre cada nó suspeito detectado, ao supervisor responsável pela regra violada. Durante a assinatura, o monitor deve enviar algumas informaçoes sobre o nó suspeito. Assim, o supervisor será capaz de correlacionar as mensagens recebidas (Figuras 2a, 2b e 2c).

O Algoritmo 1 resume o processo de assinatura. Inicialmente, todos os supervisores receberão as assinaturas encaminhadas pelos monitores. Em seguida, cada supervisor deverá verificar se é o responsável pela regra violada que está especificada na assinatura. Caso o teste seja afirmativo, a mensagem de assinatura é armazenada em um *buffer* interno. Quando o *buffer* estiver cheio, o supervisor irá processar as regras de correlação, publicando, em seguida, a lista de nós que foram confirmados como maliciosos para os monitores da rede.

Devido ao problema de espaço de armazenamento, definimos um conjunto mínimo de informações que devem ser trocadas entre os monitores, tais como: número da mensagem, regra violada, origem imediata, destino imediato, origem da mensagem e relógio.



(a) Rede formada por três monitores. M = {M1, M2, M3}. Cada monitor é responsável por uma sub-parte da rede

(b) Cada monitor deve aplicar o conjunto de regras R aos pacotes transmitidos por seus vizinhos. Neste exemplo, R = {1,2,3,4,5,6,7,8,9}

(c) M1 e M2 percebem que os nós I1 e I2, respectivamente, estão violando a regra 5 (repetição) e encaminham tais informações para o supervisor responsável pela regra. Neste exemplo M3, pois $R'_3 = \{4,5,6\}$

Figura 2. Processo de Colaboração

Algoritmo 1 - Processamento das assinaturas
enquanto mensagem recebida, **faça**
se *mensagem* == assinatura(R'_i), **então**
buffer ← mensagem
se *buffer* cheio, **então**
executar processo de correlação
publicar nós maliciosos
fim-se
fim-se
fim-enquanto

4.2. Comunicação através de DHT

Vimos que cada monitor deverá enviar informações sobre os nós detectados ao supervisor responsável pela regra violada. Baseado no trabalho de [Zhou et al. 2005b] adotamos um modelo de comunicação baseado em Tabelas Hash Distribuidas (DHT – *Distributed Hash Tables*) para permitir que o monitor identifique os supervisores que são responsáveis por cada regra. Utilizamos o protocolo Chord [Stoica et al. 2001] como base para o nosso mecanismo de DHT, uma vez que tal protocolo provê um mecanismo para mapear chaves a nós com *hashing* consistente [Karger et al. 1997].

Durante o estabelecimento da rede, será associado um ID aleatório (de n bits) para cada monitor e para cada regra selecionada para o IDS. Os monitores deverão armazenar os IDs das regras em alguma estrutura de dados interna. Todos os monitores $m_i \in M$ serão ordenados em um círculo lógico baseado nos IDs de cada monitor, seguindo o sentido horário. Quando um monitor desejar enviar uma informação para o supervisor responsável pela regra de repetição, com chave a_i, será necessário apenas encaminhar a

mensagem para o primeiro monitor m_i cujo identificador m_i é igual ou maior que o valor da chave do ataque (a_i). É importante destacar que esse procedimento permite que a função de supervisor seja atribuída dinamicamente aos monitores. Dessa forma, garantimos a escalabilidade do IDS e a resistência a falhas, pois, caso algum supervisor pare de funcionar, outro monitor automaticamente assumirá as regras que estavam atribuídas ao supervisor que falhou. O tamanho de n deve ser suficiente para suprir todos os monitores e regras da rede.

5. Análise de Performance

É importante destacar que o trabalho proposto funciona como uma arquitetura geral para a detecção de nós maliciosos em uma RSSF. A lista de regras especificada na Seção 3 é apenas um exemplo que utilizamos. O projetista da rede é livre para definir outro conjunto de regras de acordo com os requisitos da rede. Além das regras que os monitores devem usar para detectar potenciais nós maliciosos, os supervisores devem aplicar outro conjunto de regras para correlacionar as mensagens enviadas pelos monitores. Entretanto, neste artigo não tratamos o problema das regras de correlação. Então, para podermos avaliar nossa arquitetura, definimos um conjunto de regras simples como regras de correlação. Em trabalhos futuros, investigaremos regras mais sofisticadas.

Nesta seção, apresentamos comentários sobre as simulações utilizadas de forma a avaliar a arquitetura proposta.

5.1. Análise de Desempenho

Nos experimentos desenvolvidos, utilizamos o simulador Sinalgo [Sinalgo 2007]. Sinalgo é um *framework*, escrito em Java, para validação de algoritmos de rede. Diferentemente de outras ferramentas, como NS2 [NS2 2008] que permite a simulação de outras camadas da pilha de protocolos, Sinalgo foca-se na verificação de algoritmos e abstrai-se das camadas mais baixas.

5.1.1. Características da Rede

Consideramos uma rede plana e fixa cujo nós foram distribuídos de forma aleatória. Cada nó possui uma identificação única e um alcance de rádio fixo. Não existe nenhum tratamento de mensagens repetidas, o que permite ataques do tipo Repetição por parte dos nós maliciosos. Neste trabalho, consideramos apenas ataques do tipo repetição, deixando para trabalhos futuros a análise da detecção de outros tipos de ataques. A rede é composta pelos seguintes tipos de nós: básico, monitor, supervisor, intruso e estação base.

Foi considerado apenas ataques sobre mensagens de dados. Deixamos para trabalhos futuros a análise de ataques a outros tipos de mensagens, como mensagens de configuração e estabelecimento de rotas.

5.1.2. Algoritmo de Roteamento

Para a comunicação dos nós com a estação-base, criamos uma variação do protocolo *Destination-Sequenced Distance Vector* (DSDV) [Perkins and Bhagwat 1994] onde todos

os nós comuns irão transmitir apenas para a estação-base. O sistema inicia sem nenhum conhecimento sobre a identidade ou a topologia dos sensores que estão presentes. Cada nó conhece apenas sua própria identidade. Inicialmente, a estação-base é a origem das mensagens de atualização de roteamento e periodicamente realiza um *broadcast* de sua identidade. Dispositivos que estão no alcance direto da estação-base, ao receberem essa mensagem, atualizam suas tabelas de rotas. Esses nós então fazem um *broadcast* de uma nova mensagem de atualização de rotas para qualquer dispositivo que esteja ao alcance (Figura 3a), informando que há um caminho até a estação-base através deles (nós que enviaram o *broadcast*).

Para implementar nosso esquema de roteamento *multi-path* (múltiplas rotas) os dispositivos irão armazenar as três primeiras mensagens de atualização de rotas que receber (Figura 3b). Por padrão, os nós transmitirão apenas para o nó de quem ele recebeu a primeira mensagem, uma vez que tal nó corresponde ao caminho mais curto. Após o estabelecimento das rotas, cada nó irá, periodicamente, transmitir informações para a estação-base. Essas informações estão relacionadas a eventos gerados aleatoriamente em pontos diferentes no cenário simulado. A transmissão de um nó sensor é direcionada, por padrão, ao primeiro nó do qual ele recebeu a última atualização de rotas. O receptor irá repetir o mesmo processo até que o pacote alcance a estação-base. No sistema, cada nó conhece apenas a identidade do próximo salto (*hop*) que levará o pacote até o destino final.

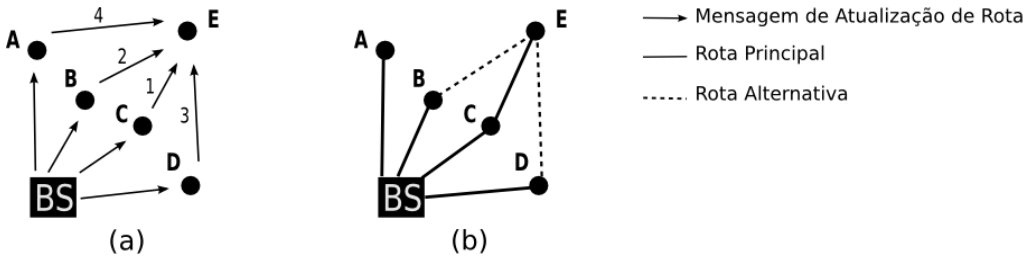


Figura 3. Protocolo de Roteamento Simulado

A comunicação entre os monitores será realizada através de *broadcast*, uma vez que o protocolo de roteamento implementado não permite a comunicação fim-a-fim entre dois nós. É importante destacar que o algoritmo utilizado, apesar de não ser o mais eficiente, demonstrou-se satisfatório, uma vez que, neste trabalho, buscamos apenas analisar o comportamento da nossa aplicação sob um protocolo de roteamento *multi-path*. Deixamos para trabalhos futuros a análise da nossa solução utilizando outros protocolos de roteamento.

5.2. Experimentos realizados

Para medir a eficácia do nosso sistema colaborativo, iremos analisar duas métricas: (1) energia consumida e (2) falsos positivos gerados. Para isso, definimos quatro cenários.

1. *Rede sem ataques e sem monitores*: através deste cenário, iremos analisar o consumo normal de energia da rede, ou seja, sem ataques sendo realizados e sem nós desempenhando função de monitor.

2. *Rede sem ataques e com monitores*: com este cenário, pretendemos analisar o consumo extra gerado pela adição de monitores na rede.
3. *Rede com ataques e sem monitores*: aqui, o objetivo é verificar o consumo gerado pela ação de nós maliciosos em uma rede sem a solução proposta.
4. *Redes com ataques e com monitores*: neste último cenário, iremos analisar o comportamento da nossa solução na detecção dos intrusos e a energia economizada devido ao isolamento dos nós intrusos.

Todas as simulações foram realizadas em um período de 5000 rounds. Foram gerados eventos aleatórios, consistindo de um valor que o nó básico, ao perceber o dado evento, deve transmitir para a estação-base.

Nos cenários 2 e 4, os monitores foram distribuídos uniformemente de modo que todos os nós pudessem ser coberto por pelo menos um monitor. Para determinar se a rede está completamente coberta, nós utilizamos uma versão modificada do trabalho descrito em [Huang and Tseng 2003]. Após a deposição no ambiente, todos os nós devem verificar se pelo menos um de seus vizinhos é um monitor, através de troca de mensagens definido pela aplicação. Caso este teste falhe para algum nó, pode-se concluir quais áreas necessitam de mais monitores. Em seguida, alguma ação pode ser realizada, como, por exemplo, a deposição de mais monitores nessa região.

Contudo, como foi explicado em [Huang and Tseng 2003], o problema tratado aqui é formulado como um problema de decisão, o qual pode responder apenas a uma questão de sim/não (Como, por exemplo, se todos os nós estão cobertos por um monitor). Consequentemente, não podemos determinar o número mínimo de monitores para cobrir todos os nós em uma determinada região. Esta é uma questão que será tratada em trabalhos futuros.

6. Resultados

Nesta seção, analisamos os resultados obtidos nos experimentos realizados. Conforme definido na seção anterior, consideramos duas métricas: a energia consumida e o número de falsos positivos gerados.

6.1. Energia

Para simplificar o processo de análise da energia, consideramos apenas 100 nós em cada cenário. Nós cenários 3 e 4, definimos que apenas 10% dos nós eram maliciosos.

Nos experimentos realizados, foram consideradas as seguintes situações de consumo de energia: transmissão de mensagem, recebimento de mensagem e escuta de mensagem. Enquanto os dois primeiros são atividades normais de um nó, a última atividade refere-se ao processo de verificar o cabeçalho da mensagem, seguido do descarte do pacote caso não seja endereçado ao nó que recebeu ou não interessa no caso de uma retransmissão até a estação-base. Fazendo isso, energia pode ser salva e a vida útil da rede aumentada. Durante os experimentos, assumimos mensagens de 36 bytes (tamanho usado em várias aplicações do TinyOS [TinyOS 2008]).

6.1.1. Modelo de Energia Utilizado

Utilizamos um modelo de energia baseado nos dados definidos em [Schmidt et al. 2007] e [Silva et al. 2005]. Consideramos a taxa de transmissão do nó de $0,26\mu s/bit$, sendo a corrente elétrica que flui pelo nó ao receber um pacote de $7,0mA$ e ao transmitir de $21,5mA$. Assim, definiu-se o seguinte modelo [Silva et al. 2005]:

- $Q_{Transmissao} = 3 * 21,5mA * (0,26 * 10^{-6}s/bit * 288bits) = 0,48375mJ/messagem$
- $Q_{Recepcao} = 3 * 7,0mA * (0,26 * 10^{-6}s/bit * 288bits) = 0,1575mJ/messagem$
- $Q_{ouvir} = 3 * 7,0mA * (0,26 * 10^{-6}s/bit * 16bits) = 0,00875mJ/messagem$

onde Energia Dissipada (Q) = Voltagem x Corrente Elétrica X Tempo, sendo Tempo = Taxa de Transmissão X Tamanho da Mensagem. Neste trabalho, não tratamos o consumo de energia relacionado ao processamento da mensagem, deixando o mesmo para trabalhos futuros.

6.1.2. Consumo de Energia

A Figura 4 mostra o consumo de energia acumulada por *round*, para os quatro cenários definidos em 5.2. Observando o consumo do cenário 3 (rede com ataques e sem monitores), percebe-se o enorme prejuízo causado pela repetição de pacotes na rede por parte dos nós maliciosos. O consumo extra gerado pela ação dos monitores, conforme ilustrado no cenário 2 (normal com monitores) é recompensando pela economia obtida ao detectar e eliminar os nós maliciosos (cenário 4 - ataques com monitores).

A Figura 5 ilustra o consumo de energia gasto em cada *round*. No intervalo compreendido entre os *rounds* 500 e 1300, período onde ocorreu a colaboração entre os monitores, percebe-se um elevado consumo de energia. Como o protocolo de roteamento implementado não permite comunicação fim-a-fim entre os nós, os monitores utilizam-se de *broadcast* para a troca de informações. Isso explica esse alto consumo observado. Contudo, é importante destacar que após a colaboração e o isolamento dos nós maliciosos, o consumo ficou bastante próximo ao consumo observado no cenário 2 (rede sem ataques

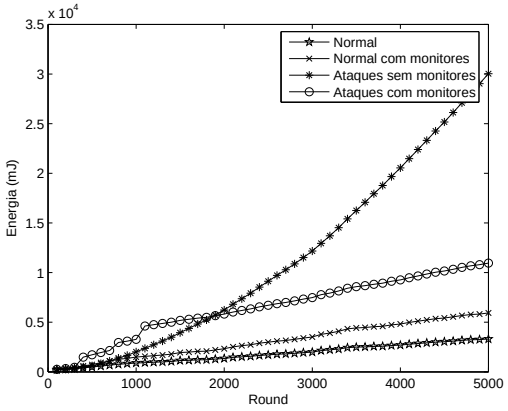


Figura 4. Energia total consumida pelos nós

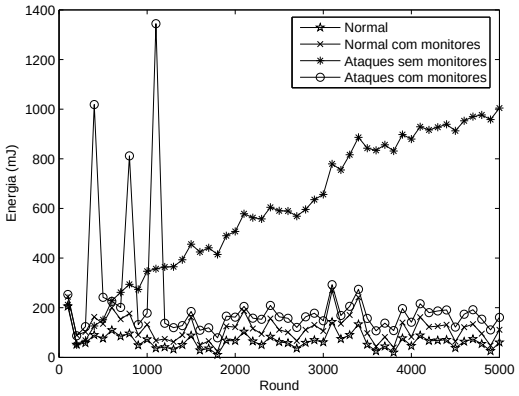


Figura 5. Energia acumulada por nós em cada round

e com monitores). Em trabalhos futuros, pretende-se investigar o consumo da energia utilizando um protocolo de roteamento mais eficiente.

Analisando os cenários 1 (rede normal) e 2 (rede normal com monitores), percebe-se claramente o consumo extra gerado pelos nós monitores devido à escuta em modo promíscuo. Ficou constatado que a execução ininterrupta da função de monitoramento diminui a vida útil dos nós monitores. Assim, em trabalhos futuros, pretendemos investigar meios que possam minimizar esse consumo extra, como o revezamento da função de monitor entre os nós da rede ou a sincronização dos monitores de modo que alguns deles desativem a escuta promíscua enquanto outros permaneçam com a função ativa.

6.2. Falsos Positivos Gerados

A redução da quantidade de falsos positivos é a principal contribuição advinda da abordagem colaborativa proposta neste trabalho. Assim, utilizamos essa métrica como forma de medir a eficácia da nossa solução.

Foram realizadas simulações no cenário 4 com 100, 300, 500, 1000 e 2000 nós. Em cada situação, nós avaliamos a eficácia da detecção com diferentes números de nós maliciosos (5%, 10% e 20% do número total de nós). Os resultados foram expressos na Tabela 1. O número de acertos representa a porcentagem dos números corretamente detectados e FP é a porcentagem de falsos-positivos. Conforme definido na Seção 5.2, os nós estavam realizando o ataque de repetição.

Como podemos perceber, até 500 nós o sistema pode detectar corretamente todos os nós sem gerar falso-positivos e com 1000 e 2000 nós, foi gerado uma pequena quantidade de falso-positivos. Isto deve-se à limitação das regras de correlação. Conforme especificado anteriormente, utilizamos regras de correlação simples apenas para poder avaliar a arquitetura proposta. Entretanto, os resultados apresentados na Tabela 1 deixam claro que nossa proposta é viável e, em trabalhos futuros, investigaremos regras mais sofisticadas.

7. Conclusão

Este trabalho propôs um Sistema de Detecção Colaborativo e Descentralizado para as RSSF onde as atividades maliciosas detectadas por cada monitor são correlacionadas

Tabela 1. Falsos positivos gerado pelo ataque de repetição

Num. de Nós	Nós Maliciosos(%)					
	5		10		20	
	Acertos	FP	Acertos	FP	Acertos	FP
100	100	0	100	0	100	0
300	100	0	100	0	100	0
500	100	0	100	0	100	0
1000	100	0	100	2	100	3
2000	100	2	100	4	100	6

por monitores especiais, chamados supervisores, com o propósito de identificar os verdadeiros nós maliciosos. A troca de informações entre monitores e supervisores é realizada através do protocolo Chord. Chord implementa um mecanismo de *Distributed Hash Table* que garante a escalabilidade do nosso sistema, uma vez que a função de supervisor é distribuída dinamicamente entre os monitores. Além disso, através de uma abordagem *cross-layer*, o IDS influencia o protocolo de roteamento de forma que os nós maliciosos sejam isolados e não possam mais causar prejuízos à rede.

Através de simulações, demonstrou-se que o IDS proposto, além de eficiente do ponto de vista da segurança, é viável sob a ótica do consumo de energia. Contudo, alguns pontos não foram tratados e ganharão mais atenção em trabalhos futuros. Dentre esses pontos, podemos citar: (1) o comportamento dos monitores na detecção de outros tipos de ataques, (2) como a falha de um supervisor afetaria a colaboração e como os monitores restantes poderiam reagir, (3) investigar meios que possam evitar a execução ininterrupta da função de monitoramento e assim economizar energia dos monitores e (4) uma análise do desempenho da nossa solução utilizando um protocolo de roteamento mais eficiente que o utilizado neste trabalho.

Agradecimentos

Os autores agradecem ao Tribunal de Contas do Estado do Piauí (TCE/PI) e à Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico (FUNCAP).

Referências

Alzaid, H., Foo, E., and Nieto, J. G. (2008). Secure data aggregation in wireless sensor network: a survey. In *AISC '08: Proceedings of the sixth Australasian conference on Information security*, pages 93–105, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

Huang, C.-F. and Tseng, Y.-C. (2003). The coverage problem in a wireless sensor network. In *WSNA '03: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pages 115–121, New York, NY, USA. ACM.

Júnior, W. R. P., de Paula Figueiredo, T. H., Wong, H. C., and Loureiro, A. A. F. (2004). Malicious node detection in wireless sensor networks. *Parallel and Distributed Processing Symposium, International*, 1:24b.

Karger, D., Lehman, E., Leighton, T., Levine, M., Lewin, D., and Panigrahy, R. (1997). Consistent hashing and random trees: Distributed caching protocols for relieving hot

spots on the world wide web. In *In ACM Symposium on Theory of Computing*, pages 654–663.

Karl, H. and Willig, A. (2005). *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons.

Li, G., He, J., and Fu, Y. (2008). A distributed intrusion detection scheme for wireless sensor networks. In *ICDCSW '08: Proceedings of the 2008 The 28th International Conference on Distributed Computing Systems Workshops*, pages 309–314, Washington, DC, USA. IEEE Computer Society.

Marti, S., Giuli, T. J., Lai, K., and Baker, M. (2000). Mitigating routing misbehavior in mobile ad hoc networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 255–265, New York, NY, USA. ACM.

NS2 (2008). http://nsnam.isi.edu/nsnam/index.php/main_page.

Perkins, C. E. and Bhagwat, P. (1994). Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *SIGCOMM Comput. Commun. Rev.*, 24(4):234–244.

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

Roman, R., Zhou, J., and Lopez, J. (2006). Applying intrusion detection systems to wireless sensor networks. *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE*, 1.

Schmidt, D., Krämer, M., Kuhn, T., and Wehn, N. (2007). Energy modelling in sensor networks. *Advances in Radio Science*, 5:347–351.

Silva, A. P. R. D., Martins, M. H. T., Rocha, B. P. S., Loureiro, A. A. F., Ruiz, L. B., and Wong, H. C. (2005). Decentralized intrusion detection in wireless sensor networks. In *Q2SWinet '05: Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*, pages 16–23, New York, NY, USA. ACM.

Sinalgo (2007). http://dcg.ethz.ch/projects/sinalgo/.

Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., and Balakrishnan, H. (2001). Chord: A scalable peer-to-peer lookup service for internet applications. pages 149–160.

TinyOS (2008). http://www.tinyos.net.

Zhou, B., Marshall, A., Wu, J., Lee, T.-H., and Liu, J. (2005a). A cross-layer route discovery framework for mobile ad hoc networks. *EURASIP J. Wirel. Commun. Netw.*, 5(5):645–660.

Zhou, C. V., S.Karunasekera, and Leckie, C. (2005b). A peer-to-peer collaborative intrusion detection system. In *Networks, 2005. Jointly held with the 2005 IEEE 7th Malaysia International Conference on Communication., 2005 13th IEEE International Conference on*. IEEE.