Segurança em Redes de Sensores Sem Fio

Jean Fellipe de Almeida Pimentel

JUIZ DE FORA

MAIO, 2012

Segurança em Redes de Sensores Sem Fio

Jean Fellipe de Almeida Pimentel

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Orientador: Alex Borges Vieira

JUIZ DE FORA
MAIO, 2012

Segurança em Redes de Sensores Sem Fio

Jean Fellipe de Almeida Pimentel

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Alex Borges Vieira
Doutor

Ana Paula Couto da Silva
Doutora

Eduardo Pagani Julio
Mestre

JUIZ DE FORA

25 DE MAIO, 2012

Resumo

Este trabalho tem como finalidade apresentar e descrever os desafios de segurança em redes de sensores sem fio. O estudo aborda o funcionamento, características e arquiteturas a serem consideradas na concepção e no estudo de uma rede de sensores. Devido à sua natureza, muitos são os tipos de ataques que uma rede de sensores sem fio pode sofrer e, dado as suas inúmeras áreas de aplicação, é mostrada a importância de uma segurança efetiva, que garanta o correto funcionamento da rede, impedindo ataques ou minimizando os efeitos por esses provocados. Nesse trabalho também são realizadas simulações de um ataque e uma análise dos efeitos provocados, como atraso na entrega das informações e aumento do consumo de energia.

Palavras-chave: redes de sensores sem fio, segurança, ataques, simulação.

Abstract

This work intends to show and describe the security challenges in wireless sensor networks. It addresses operation, characteristics and architectures that must be considered during the design of a wireless sensor network. By their nature, there are many types of attacks that wireless sensor networks can suffer and, due to it's range of applications, this work shows the importance of effective security, ensuring the correct operation of the network, preventing attacks or minimizing the effects caused by these. Also, in this work was made an attack simulation and was analysed the effects, such as delay in delivery of messages and increased energy consumption.

Keywords: wireless sensor networks, security, attacks, simulation.

Agradecimentos

A Deus, pela vida e oportunidades oferecidas.

Aos meus pais, irmãos e familiares pelos ensinamentos, amor, confiança e apoio em todas as horas.

À minha bisavó Ely (*in memorian*) por todo o amor e ensinamento passado por anos, que sempre me lembrarei.

À minha namorada Amanda, pelo carinho, dedicação e constante incentivo.

Aos amigos do curso, com quem aprendi bastante e que tornaram o ambiente melhor e mais agradável durante esses anos.

Aos velhos amigos, ainda que hoje separados pela distância, estão sempre presentes.

Ao professor Alex, pela orientação, disponibilidade e apoio.

Aos professores do departamento, que se dedicaram a ensinar e que contribuíram para a minha formação pessoal e profissional.

"Computer programming is an art, because it applies accumulated knowledge to the world, because it requires skill and ingenuity, and especially because it produces objects of beauty. A programmer who subconsciously views himself as an artist will enjoy what he does and will do it better."

Donald Knuth

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

AODV	*Ad-hoc On-Demand Distance Vector*
CBR	*Constant Bit Rate*
CDMA	*Code Division Multiple Access*
DARPA	*Defense Advanced Research Projects Agency*
DCC	Departamento de Ciência da Computação
DSR	*Dynamic Source Routing*
FDMA	*Frequency Division Multiple Access*
FTP	*File Transfer Protocol*
GTS	*Guaranteed Time Slot*
IEEE	*Institute of Electrical and Electronics Engineers*
INSENS	*INtrusion-tolerant routing protocol for wireless SEnsor NetworkS*
ISI	*Information Sciences Institute*
MAC	*Media Access Control*
NAM	*Network Animator*
NS-2	*Network Simulator 2*
NSF	*National Science Foundation*
OFDM	*Orthogonal Frequency-Division Multiplexing*
RERR	*Route Error*
RREP	*Route Response*
RREQ	*Route Request*
RSSF	Redes de Sensores sem Fio
SPINS	*Security Protocols for Sensor Networks*
TCP	*Transmission Control Protocol*
TDMA	*Time Division Multiple Access*
TTL	*Time to Live*
UDP	*User Datagram Protocol*
UFJF	Universidade Federal de Juiz de Fora

1 Introdução

Segundo Loureiro *et al* (2003), na última década houve um grande avanço tecnológico nas áreas de sensores, circuitos integrados e comunicação sem fio, estimulando o desenvolvimento e uso de sensores inteligentes. Um sensor inteligente é um *chip* que contém sensores com capacidade de processamento e comunicação de dados.

Redes de Sensores Sem Fio (RSSF) são redes de dispositivos móveis (sensores inteligentes), que tem como objetivo o monitoramento de fenômenos através da coleta de dados por seus participantes. (Akyildiz *et al*, 2002) Podem ser vistas como uma especialização de redes *ad-hoc*, isto é, redes não dependentes de uma infraestrutura centralizada, onde cada participante contribui ativamente para o funcionamento da rede.

Pesquisas na área de RSSF vêm aumentando cada vez mais, pois suas aplicações mostram-se viáveis para diversos cenários, como automação de linhas industriais, segurança de ambientes, exames médicos intracorpóreos etc. E dada a ampla gama de aplicações de RSSFs, faz-se necessário estudos à cerca da segurança da rede, garantindo que informações não sejam interceptadas ou alteradas, rotas não sejam desviadas e recursos computacionais não sejam desperdiçados.

Nessa monografia, foi realizada a simulação de um ataque na camada de roteamento e métricas foram escolhidas para analisar o impacto causado pelo ataque. O resultados mostram que poucos nós podem causar um grande impacto, aumentando consideravelmente o consumo de energia, aumentando os atrasos fim a fim e diminuindo a vazão de entrega.

Esse trabalho está organizado em três capítulos. No capítulo 2 serão apresentadas as características, arquiteturas e os diversos fatores que devem ser levados em consideração durante a concepção e planejamento de uma RSSF. A seguir, no capítulo 3, serão apresentados os atributos desejáveis para a garantia da segurança em uma RSSF e os impactos causados por esses, dado as restrições computacionais dos nós. Ainda, serão descritos alguns tipos comuns de ataques nas camadas de enlace, rede e transporte. Finalizando, no capítulo 4 será apresentado um cenário de utilização de RSSF e um ataque será descrito

e simulado, e será efetuada uma análise dos impactos causados na rede por esse ataque. Por fim, será feita uma conclusão sobre a importância do estudo de RSSF, os problemas de segurança e propostas para trabalhos futuros.

2 Redes de Sensores Sem Fio

Segundo Mateus e Loureiro (1998) , o avanço tecnológico na área da computação ocorre rapidamente, promovendo saltos na tecnologia e novas formas de utilização dos computadores. Com isso, as redes de computadores vêm sofrendo constante evolução e impulsionando pesquisas em comunicação sem fio, apropriadas para situações onde não se pode ou não se deseja usar uma instalação com fios como por exemplo, a utilização de redes em áreas remotas ou com grande quantidade de dispositivos.

Essa rápida evolução possibilitou o surgimento da computação móvel, definida por Mateus e Loureiro (1998) como o ambiente onde os dispositivos portáteis tem capacidade de comunicação com a parte fixa da rede e possivelmente com outros dispositivos móveis, permitindo que os usuários tenham acesso a serviços independente de suas localizações, e mais importante, das mudanças de suas localizações.

Redes sem fio podem operar através de dois modos de comunicação: infraestrutura e *ad-hoc*. Em modo infraestrutura, os computadores se conectam diretamente a um ponto de acesso, que é responsável pelo gerenciamento e provimento dos serviços. Essa centralização ao passo que reduz a complexidade da rede, também a limita, pois os computadores necessitam estar dentro do raio de comunicação do ponto de acesso. O modo *ad-hoc*, por outro lado, não exige pontos de acesso especializados, ficando cada computador responsável pelo provimento dos serviços. O modo *ad-hoc* é complexo pois cada computador precisa implementar mecanismos para garantir o correto funcionamento da rede.

Segundo Akyildiz *et al* (2002), com o aperfeiçoamento nos anos recentes de dispositivos sensores e circuitos integrados com capacidade de computação e comunicação, surgiu uma nova vertente: Redes de Sensores Sem Fio (RSSF). RSSF podem ser vistas como uma especialização de redes *ad-hoc* e são formadas por diversos dispositivos autônomos, de pequenas dimensões, cujo objetivo é monitorar um fenômeno de interesse através da coleta de informações individuais de cada nó. Uma RSSF é representada na Figura 2.2 e segundo Tilak *et al* (1998), os papéis principais em uma RSSF são o observador,

o fenômeno e o nó sensor.

O **observador** é o usuário interessado nas informações colhidas e disseminadas na rede de sensores. A informação pode ser obtida em intervalos definidos ou através de consultas do observador. Além disso, podem existir múltiplos observadores em uma mesma rede de sensores.

O **fenômeno** é a atividade de interesse do observador, que será monitorada pelos sensores, e portando o motivo para a existência da rede de sensores. Em uma rede podemos ter ainda, o monitoramento de múltiplos fenômenos. No caso, por exemplo, de uma rede de sensores voltada à segurança de uma área física, podem ser monitoradas as variações de temperatura e som.

O **nó sensor** é um dispositivo de pequenas dimensões (alguns centímetros) e é composto tipicamente de transmissor-receptor, memória, processador, bateria e sensores. Sua função é monitorar características diversas como temperatura, campo magnético, luz etc. e, a partir daí, enviar os dados capturados através da comunicação sem fio. A figura 2.1 exibe o sensor *Waspmote*.

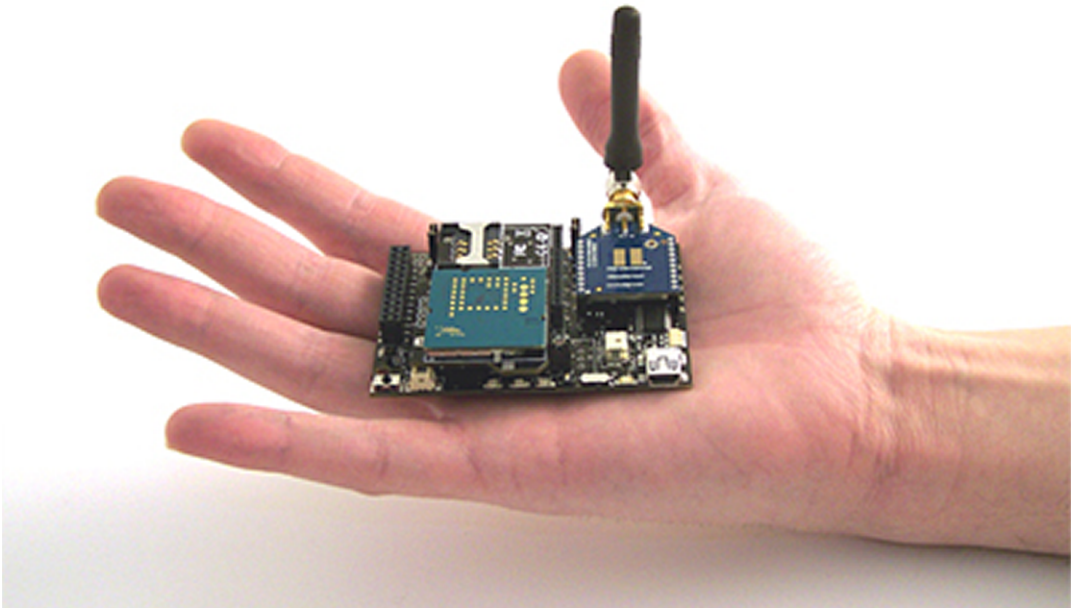

Figura 2.1: Exemplo de nó sensor. (Libelium - Waspmote, 2012)

Em uma rede de sensores pode-se ter de centenas a milhares de nós e é importante, portanto, o baixo custo de cada dispositivo. Dessa forma, os sensores tendem a ser simples e tem como consequência a redução da capacidade de seus componentes. Sendo os nós

sensores limitados fisicamente com pouca capacidade computacional e energética, a rede de sensores sem fio depende então do esforço colaborativo dos nós que a compõe.

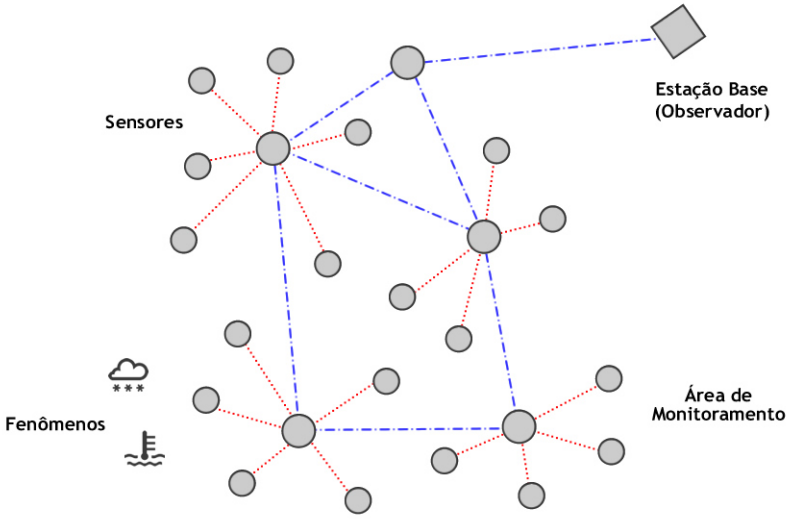

Figura 2.2: Representação de uma RSSF, baseada em (Akyildiz *et al*, 2002)

Segundo Akyildiz *et al* (2002), para se avaliar uma rede de sensores devem ser analisadas as seguintes métricas: tolerância a falhas, escalabilidade, custo de produção e restrições de hardware, topologia e restrições ambientais, meios de transmissão e consumo de energia.

A escalabilidade é crítica, visto que a rede pode conter de centenas à milhares de nós. Dependendo da aplicação, nós podem entrar em modo de espera para o prolongamento da vida útil da rede ou novos nós podem ser incrementados à rede existente e esta deve ser capaz de manter o funcionamento de forma eficiente.

As falhas podem surgir devido às más condições do ambiente, mal funcionamento dos nós sensores ou esgotamento energético da bateria e a rede deve ser capaz de se recuperar e manter o funcionamento. Segundo Pereira *et al* (2006), é desejável que falhas não catastróficas sejam transparentes ao usuário e, uma forma de contornar o problema, é a replicação dos dados monitorados. Porém, duplicação envolve mais processamento e mensagens na rede, consumindo maior energia.

Uma vez que a comunicação dos nós é sem fio, o canal está sempre sujeito à fatores de atenuação do sinal como interferência e obstáculos e devem ser adotadas topologias para

a que informação flua sempre, mesmo com nós comprometidos. As técnicas utilizadas, visando a tolerância à falhas, podem ser custosas, devendo então ter seus usos ponderados de acordo com os requisitos da aplicação.

Segundo Akyildiz *et al* (2002), enquanto redes tradicionais almejam velocidades maiores e baixa perda de pacotes, protocolos de redes de sensores devem focar no uso otimizado da energia disponível. O consumo de energia deve ser prioritário na construção ou escolha de protocolos de roteamento e comunicação, com o intuito de prolongar o tempo de vida da rede.

2.1 Aplicações de Redes de Sensores Sem Fio

Com o potencial de observação de fenômenos, redes de sensores sem fio mostram-se como soluções para diversas situações e problemas, podendo ser empregadas em diversas áreas. ZigBee Alliance (2012) apresenta algumas:

- **Controle** - Controlar serviços de forma automatizada. Podem ser utilizadas por uma distribuidora de água, gás ou energia com intuito de monitorar parâmetros do sistema como fluxo, pressão e temperatura. Também podem ser empregadas em linhas de produção, identificando produtos com problemas ou aquecimento das máquinas;

- **Ambiente** - Monitorar ambientes internos como prédios e residências, controlando temperatura e iluminação, e externos como florestas e lagos, vulcões, áreas de tempestade, movimentação tectônica etc;

- **Tráfego** - Monitoramento do fluxo em rodovias ou trânsito urbano, melhorando a movimentação dos veículos. Podem ser utilizadas em um sistema inteligente, capaz de desviar fluxos e alterar a temporização dos semáforos em função do trânsito da região em tempo real;

- **Segurança** - Prover segurança em centros comerciais e estacionamentos através de sensores como câmeras, temperatura e movimentos;

- **Medicina/Biologia** - Monitorar órgãos e sistemas internamente, através de sensores minúsculos injetados no corpo humano ou animal, aumentando o nível de precisão dos exames e gerando diagnósticos melhores, bem como a liberação de medicamentos de acordo com o estado do paciente e nos locais corretos;

- **Militar** - Detectar movimentos inimigos, presença de materiais radioativos, gases venenosos, explosivos. Reconhecer regiões inimigas e proteger áreas governamentais de alta segurança;

2.2 Características

Segundo Silva (2006), as redes de sensores sem fio podem ser homogêneas ou heterogêneas em relação aos tipos, funcionalidades e dimensões dos nós sensores. Em uma RSSF cujo objetivo é a segurança de uma área física, o monitoramento de temperatura e som podem ser feitos pelo mesmo nó sensor ou em nós distintos e podem apresentar características diferentes de acordo com o fenômeno a ser observado, como maior ou menor necessidade de memória, bateria ou processador.

Outro ponto está relacionado ao volume de dados, frequência de coleta e processamento. Se for de responsabilidade do nó realizar um processamento dos dados colhidos, haverá um impacto em seu tempo de vida, uma vez que serão exigidos maior processamento e memória, consequentemente consumindo maior quantidade de energia. Por outro lado, se os dados são colhidos, distribuídos e processados fora da rede, haverá um menor gasto de energia.

O estudo e projeto de uma RSSF envolve a análise de muitos parâmetros. Ruiz (2003) classificou as RSSFs segundo a configuração, o sensoriamento, a comunicação e o processamento que executa.

2.2.1 Classificação Segundo a Configuração

Segundo Ruiz (2003), a configuração está relacionada com os requisitos das aplicações, forma e dimensão do monitoramento, características do ambiente, escolha dos nós e tipos de serviços disponibilizados.

Composição - Homogênea, quando os nós apresentam o mesmo hardware ou heterogêneas, onde alguns nós possuem hardwares especializados. Redes homogêneas flexibilizam as tarefas e aumentam a tolerância a falhas, pois nós podem assumir as responsabilidades de nós perdidos. Em outros casos, a aplicação pode exigir nós mais especializados, que otimizem certos processos.

Organização - Hierárquica, quando os nós são organizados em diferentes níveis e responsabilidades ou plana, onde todos apresentam o mesmo nível e responsabilidade. Hierarquização pode diminuir o volume de informações trocadas pelos nós, melhorando a comunicação, pois os nós pais podem processar e agrupar os dados dos filhos. Por outro lado, uma organização plana é mais tolerante a falhas e o gerenciamento da hierarquia como a eleição dos líderes e controle dos filhos é custosa.

Mobilidade - Estacionária, onde os nós permanecem nos mesmos lugares onde foram depositados ou móvel, onde os nós se movimentam no cenário. Quanto maior a mobilidade, melhor deve ser o gerenciamento de descoberta de rotas para que os nós não percam a conectividade com a rede.

Densidade - Densa, com alta concentração de nós por unidade de área; esparsa, com baixa concentração; ou balanceada. A quantidade de nós está diretamente relacionada à escalabilidade e confiabilidade da rede. Em redes densas, pode ocorrer um revezamento entre os nós, onde alguns podem entrar em modo de espera para economia de energia até que sejam necessários. Redes densas, se não controladas de modo eficiente podem gerar uma sobrecarga de mensagens, causando interferências e falhas. Redes esparsas podem sofrer com problemas de comunicação, principalmente em redes móveis, uma vez que a perda de um nó pode influenciar fortemente a rede.

Distribuição - Regular, quando há uma distribuição uniforme dos nós sobre a área monitorada ou irregular, quando não há. Para redes estáticas e hierarquizadas, a distribuição regular permite a otimização do controle de hierarquia, pois já se sabe quantos filhos um nó pai irá gerenciar. Em redes móveis, a distribuição pode torna-se irregular e comprometer a comunicação, seja com a perda de conectividade com

alguns nós ou impactando a hierarquia dos nós.

2.2.2 Classificação Segundo o Sensoriamento

De acordo com Ruiz (2003), o ponto mais importante em uma RSSF é o sensoriamento, isto é, a coleta dos dados por meio dos sensores e depende dos tipos de fenômenos a serem observados. A frequência do sensoriamento tem um impacto imediato sobre a rede pois está relacionada diretamente com a quantidade de dados trafegados e com o consumo de energia e pode ser classificado de acordo com o seu comportamento.

A coleta pode ser contínua, quando acontece sem interrupção; periódica, quando acontece em intervalos regulares e definidos; reativa, onde é solicitada pelo observador ou devido a ocorrência de um evento; ou em tempo real, onde é coletada a maior quantidade de dados no menor tempo possível. A coleta periódica e a reativa apresentam como vantagem a economia de energia, sendo que a de tempo real é a mais adequada para aplicações que envolvem riscos e tomadas de decisões rapidamente.

2.2.3 Classificação Segundo a Comunicação

Ruiz (2003), já dizia que a comunicação em uma rede de sensores pode ser dividida entre comunicação de aplicação e comunicação de infraestrutura. A comunicação de aplicação está relacionada com a transferência dos dados monitorados dos nós sensores até o observador, e a de infraestrutura, está relacionada com a comunicação necessária para configuração e gerenciamento dos dispositivos, assegurando o funcionamento e otimizando a rede. A comunicação é influenciada diretamente pelo objetivo da RSSF, já que deve satisfazê-lo, e necessita ser bem otimizada para manter a rede eficiente.

A classificação sob a perspectiva da aplicação pode ser programada, onde os nós disseminam os dados em intervalos regulares; sob demanda, onde os nós disseminam os dados em resposta à eventos ou por solicitação do observador; ou contínua, onde os nós disseminação os dados à medida que vão sendo coletados ou processados, sem interrupção. A forma contínua pode levar à um menor tempo de vida da rede devido ao consumo de energia dos nós e o aumento do envio dos dados da aplicação pode degradar a comunicação de infraestrutura da RSSF ao aumentar a utilização do canal.

Ainda de acordo com Ruiz (2003), sob a perspectiva de infraestrutura, podem ser classificadas de acordo com o fluxo dos dados, tipo de conexão, tipo de transmissão e modo de alocação de canais.

Fluxo de Dados - *Broadcast*, onde os dados são enviados para todos os nós vizinhos, que repetem o procedimento, até que os dados alcancem toda a rede; *multicast*, onde os nós formam grupos e os dados são enviadas somente para seus membros; *unicast*, onde nós enviam os dados a somente um nó por vez; *gossiping*, onde enviam os dados para um nó vizinho selecionado aleatoriamente, onde este repete o procedimento; ou *bargaining*, onde há uma negociação prévia e os nós só enviam os dados caso haja o nó destino manifeste interesse. Devido à falta de protocolos eficientes para endereçamento e roteamento, *broadcast* e *multicast* tendem a ser mais utilizados. A vantagem da utilização de *broadcast* é a alta tolerância a falhas, não apresentando problemas em mudanças na topologia dada a disponibilidade dos dados na rede. Em contrapartida, resulta em uma alta sobrecarga de mensagens na rede. O *multicast* apresenta uma sobrecarga menor que a utilização de *broadcast* e em conjunto com técnicas de agregação de dados, o pode otimizar a comunicação.

Tipo de Conexão - Simétrica, onde todos os nós apresentam o mesmo alcance de transmissão, ou assimétrica, onde possuem alcances diferentes. Em redes simétricas, as vantagens são as mesmas apresentadas na Seção 2.2.1, Composição Homogênea, isto é, flexibilização de tarefas e aumento de tolerância a falhas. Nas redes assimétricas, os diferentes alcances podem servir como *gateways* entre nós pais no caso de redes hierarquizadas, ou provendo melhores rotas para a comunicação.

Transmissão - *Simplex*, onde a comunicação flui em apenas um sentido, isto é, um transmissor em uma extremidade e um receptor na outra; *half-duplex*, onde a comunicação flui em ambos os sentidos, mas apenas um sentido por vez; ou *full-duplex*, quando a comunicação é realizada simultaneamente em ambos os sentidos. Redes com comunicação *simplex* são as mais básicas e simples, não havendo meios de se verificar a recepção dos dados. Por outro lado, *half-duplex* e *full-duplex* permitem detecção de erros e pedidos de retransmissões, com este último sendo capaz de transmitir mais

informações por unidade de tempo devido à comunicação simultânea.

Alocação de Canal - Estática, onde a largura de banda é dividida pela quantidade de nós, seja na frequência (*FDMA*), no tempo (*TDMA*), no código (*CDMA*) ou ortogonalmente (*OFDM*); ou dinâmica, onde não existe a divisão e os nós disputam o acesso ao canal. Com alocação estática, pode ocorrer desperdício do canal dado que alguns nós podem não ter o que transmitir e a banda reservada para o nó não será utilizada. A alocação dinâmica reduz esse desperdício, porém técnicas de controle de acesso ao canal fazem-se necessárias, contornando colisões durante as transmissões.

Segundo Pereira *et al* (2006), devemos ter bastante atenção à essas características, dado que a interação entre o modo de envio de dados da camada de aplicação e da camada de infraestrutura causa um impacto significante no desempenho da rede.

2.2.4 Classificação Segundo o Processamento

Segundo Ruiz (2003), dada as complexidades de uma rede de sensores sem fio e as restrições dos nós, a divisão de tarefas pode otimizar e garantir um correto funcionamento, obtendo-se um adequado tempo de vida útil. Ainda, os algoritmos devem ser projetados para permitir que nós individuais ou grupos de nós processem e partilhem os dados de forma eficiente. E, dentre esses processamentos, métodos de correlação de dados podem ser empregados com o objetivo de reduzir os dados trafegados entre os nós, sendo muito utilizados: a fusão de dados, agregação, compressão, filtragem e supressão seletiva.

O processamento dos nós pode ser de infraestrutura, onde o processamento está relacionado com o funcionamento básico da rede como roteamento, controle de acesso ao meio, eleição de líderes, dentre outros; ou localizada, onde os nós executam processamentos locais básicos sobre os dados coletados pelo sensores; ou correlação, onde um nó ou um grupo de nós executam procedimentos de correlação como os citados anteriormente, sobre os dados coletados. Com essas responsabilidades definidas, cada nó pode desempenhar seu papel da melhor maneira, seja mantendo a infraestrutura, gerenciando a captura de dados através dos sensores ou processando as informações obtidas.

2.3 Arquitetura de uma Rede de Sensores Sem Fio

Um nó pode realizar tarefas diferentes: sensoriamento do ambiente, processamento das informações e gerenciamento do tráfego entre os diversos nós. Ao contrário de redes estruturadas, redes de sensores não possuem uma entidade central para gerenciamento da rede e portanto, cada nó deve ser capaz de autoconfigurar-se e estabelecer rotas com os outros nós.

De acordo com Castro (2010), a rede deve ser capaz de se auto gerenciar, balanceando a carga transmitida pela rede e otimizando as comunicações para obter a precisão requerida pelos observadores. E, a pilha de protocolos deve combinar o roteamento consciente de energia disponível, viabilizar a comunicação eficiente com o meio sem fio e promover os esforços de cooperação dos nós sensores.

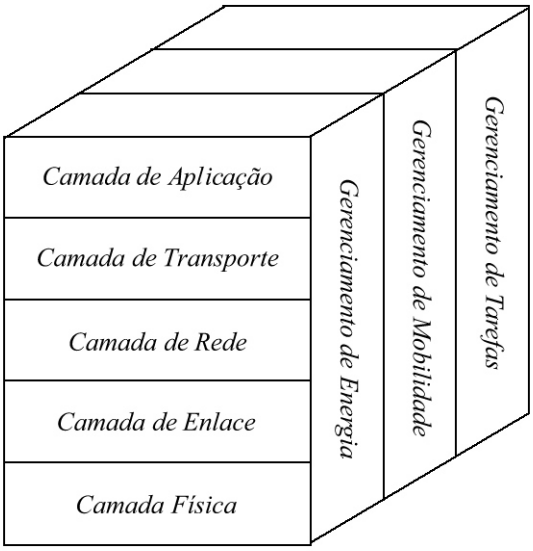

Figura 2.3: Camadas de um rede de sensores. (Akyildiz *et al*, 2002)

Na Figura 2.3 podem ser visualizadas as camadas de uma rede de sensores sem fio, onde cada camada é responsável por:

- **Física** - Transmitir através do canal de comunicação, controlando a velocidade e vazão da transmissão na rede;
- **Enlace** - Gerenciar o fluxo de dados e controlar o acesso ao meio, minimizando as

colisões da transmissão. Pode ainda, detectar e corrigir erros da camada física.

- **Rede** - Controlar o endereçamento e roteamento dos pacotes entre origem e destino, escolhendo o melhor caminho entre os nós. Também, controlar o congestionamento da rede.

- **Transporte** - Manter o fluxo de dados, incluindo ordenar pacotes, corrigir erros e enviar confirmações de recebimento de pacotes.

- **Aplicação** - Fornecer os serviços da rede de sensores aos usuários finais, observadores.

Dadas as responsabilidades, protocolos podem ser pesquisados e desenvolvidos e embora as pesquisas estejam cada vez mais frequentes, redes de sensores sem fios estão em um estágio inicial dada as muitas limitações técnicas ainda existentes. Um dos principais grupos interessados na pesquisa e desenvolvimento de RSSF é a ZigBee Alliance, um grupo formado por empresas de diversos segmentos como AT&T, Cisco, Intel, Samsung, Siemens, dentre outras.

Segundo ZigBee Alliance (2012), o intuito do grupo é o desenvolvimento de um padrão de comunicação confiável para dispositivos de baixo custo e com baixo consumo de energia, prolongando a vida útil da rede. Ainda, o padrão ZigBee é construído com base no padrão IEEE 802.15.4, voltado para as camadas física e de enlace e desenvolvido pelo *Institute of Electrical and Electronics Engineers* (IEEE), sendo a ZigBee Alliance responsável pelas camadas de transporte e rede. A Tabela 2.1 apresenta os responsáveis por cada camada no padrão ZigBee.

Responsabilidade	**Camada**
Usuário	Aplicação
ZigBee Alliance	Transporte
	Rede
IEEE 802.15.4	Enlace
	Física

Tabela 2.1: Responsabilidades das camadas no ZigBee

Segundo o padrão ZigBee, os nós podem ser classificados de acordo com as funções que exercem. Sob essa perspectiva, um dispositivo pode ser um coordenador, responsável

pela inicialização da rede através da seleção do canal de comunicação e gerenciamento do endereçamento dos nós; ou nó final, sendo responsável apenas pelo sensoriamento da área; ou roteador, atuando como ponte entre nós finais e coordenadores. Nós coordenadores geralmente apresentam maior capacidade, podendo ser estes dispositivos utilizados como interface de comunicação com o observador. Nós roteadores, opcionais, permitem um aumento da área monitorada.

Sob a perspectiva de capacidade computacional, os dispositivos podem ser classificados como FFD (*Full Function Device*), onde o nó é capaz de atuar em várias funções: coordenador, roteador ou nó sensor; ou RFD (*Reduced Function Device*), onde o nó apresenta funções reduzidas, atuando apenas como nó final. Nós RFDs são dispositivos mais simples, de menor custo de produção, com menor consumo de energia e comunicam-se apenas com dispositivos FFD.

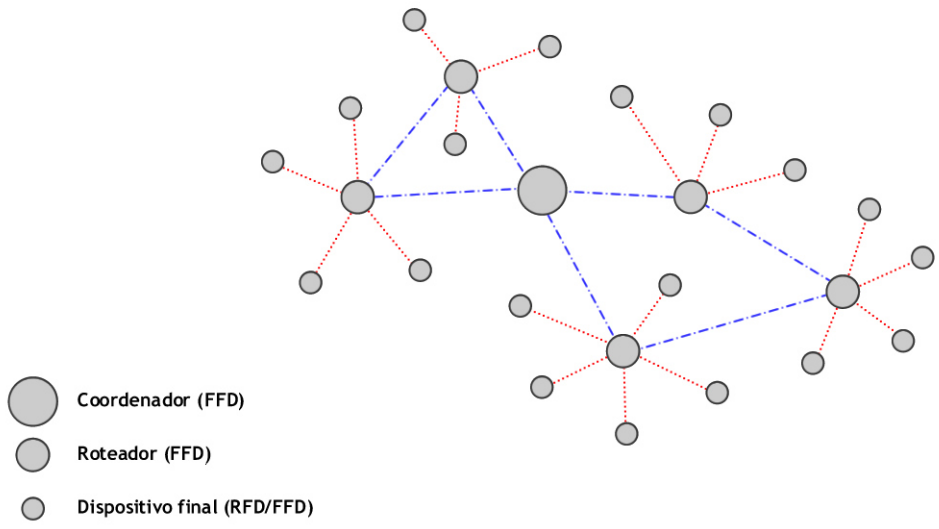

Figura 2.4: Funções e tipos de dispositivos em uma rede de sensores sem fio

O padrão define topologias para a rede de sensores: estrela, árvore e ponto a ponto. Na topologia estrela, a comunicação é centralizada, isto é, a comunicação entre nós finais, se necessária, é realizada somente através do nó coordenador. A topologia árvore é formada por *clusters* onde cada um destes possui um nó principal responsável pela comunicação externa. Um nó pertencente a um *cluster* nunca se comunica com o exterior, sendo toda a informação repassada através do nó principal, aproximando-se as-

sim da topologia estrela. De modo oposto, a topologia ponto a ponto é descentralizada, permitindo que nós comuniquem-se diretamente entre si e reduzindo a carga no nó coordenador. A Figura 2.4 exemplifica uma topologia estrela e sua classificação segundo o padrão ZigBee.

Sendo o intuito da Zigbee Alliance o desenvolvimento de um padrão de comunicação com baixo consumo de energia, o padrão Zigbee suporta dois modos de operação: *beaconing* e *non-beaconing*. *Beacons* são *frames* especiais que contém informações sobre a rede e no modo *beaconing*, são enviados periodicamente por determinados nós, sinalizando suas presenças na RSSF. Desta forma, os nós precisam estar ativos somente enquanto um *beacon* é transmitido, permitindo que possam entrar em modo de espera no intervalo entre esses e assim, reduzindo o consumo de energia. Com a não utilização de *beacons*, os dispositivos devem ser mantidos em modo ativo durante todo o tempo, aumentando o consumo e reduzindo o tempo de vida da rede.

Cabe ressaltar a importância do estudo e da implementação de mecanismos de segurança em uma rede de sensores sem fio. Do contrário, as informações, a rede e até possivelmente o fenômeno observado podem ficar comprometidos.

3 Segurança

Segundo Margi *et al* (2009), diferentes aplicações de redes de sensores sem fio possuem diferentes requisitos de segurança. Por exemplo, em um monitoramento climático de uma floresta, os pesquisadores que utilizarão os dados esperam que estes reflitam a realidade do que foi sensoreado na floresta, sendo importante então a integridade dos dados coletados e transmitidos.

Ainda segundo Margi *et al* (2009), embora os dados monitorados em uma floresta não sejam sigilosos, a disposição de máquinas e dispositivos em uma planta fabril poderia levar a prejuízos ou vantagens à concorrência. Assim, além da integridade, o sigilo desses dados nessa aplicação é de extrema importância.

Em Gerheim (2010), para que uma rede cabeada ou sem fio possa fornecer dados com segurança, técnicas devem ser utilizadas nas informações trafegadas de modo que requisitos sejam atendidos. Esses requisitos surgem da necessidade de garantir que a informação recebida seja exatamente igual à informação enviada e que os nós de origem e destino sejam legítimos. Segundo Kurose e Ross (2005), Pereira *et al* (2006) e Anjum e Mouchtaris (2007), os requisitos que devem ser levado em consideração no estudo e projeto de uma rede segura, são:

Confidencialidade - Tal requisito diz que somente o nó remetente e o nó destinatário devem entender a informação transmitida. Ainda que outros dispositivos, inclusive de outras redes, façam a captura dos pacotes trafegados, não devem ser capazes de compreender os dados ali contidos, impedindo assim o roubo de informação. É necessário o uso de criptografia para satisfazer esse requisito, ficando as chaves criptográficas em poder dos nós. Se cada nó tiver sua própria chave, ou ainda, quanto maior a quantidade de chaves que um nó puder utilizar, maior confidencialidade será garantida.

Integridade - A mensagem enviada pode percorrer múltiplos nós antes de alcançar o seu destino, podendo sofrer interferências, seja através do meio de transmissão ou

através de um ataque onde o nó malicioso pode injetar, alterar ou remover dados da mensagem original, mesmo estando criptografada. A integridade então garante que os dados não foram alterados durante o trânsito e esse requisito geralmente é obtido através de funções *hash*, funções que geram assinaturas para cada entrada de dados.

Autenticidade - A autenticidade garante a identidade dos nós envolvidos. Assegura que as mensagens provenientes de um determinado nó originaram realmente dele. Tal requisito faz-se necessário para garantir o correto funcionamento da rede, evitando que nós maliciosos se passem por outros nós, disseminando falsas informações. A autenticidade pode ser alcançada com o uso de chaves públicas e privadas onde o nó emissor utiliza sua chave privada e transmite a mensagem. Esta somente poderá ser decifrada com a chave pública do nó emissor, garantindo assim que a origem é autêntica.

Dados recentes - Segundo Pereira *et al* (2006), é importante garantir que os dados que trafegam sejam recentes, ou seja, que não são dados antigos que foram reinjetados na rede e esse mesmo conceito pode ser aplicado às chaves criptográficas em uso. Esses dados, mesmo que autênticos, não devem ser considerados como válidos e dependendo da aplicação da rede, podem causar grandes danos.

Disponibilidade - De acordo com Anjum e Mouchtaris (2007), a rede deve estar sempre disponível para as partes autorizadas. Assim, devem ser implementados mecanismos para impedir ataques de negação de serviço (como a injeção de pacotes inúteis para provocar uma sobrecarga na rede, consumindo recursos computacionais dos nós e reduzindo o tempo de vida) que podem ocorrer em qualquer camada da rede.

Segundo Margi *et al* (2009), levando-se em conta as características das redes de sensores sem fio, cujos nós apresentam poucos recursos de processamento, memória e energia limitados, os mecanismos de segurança empregados devem ser escaláveis (em termos de energia e atraso). Para a garantia da confidencialidade é necessário o uso de criptografia, e para a autenticidade são adicionadas assinaturas às mensagens. A criptografia e geração da assinatura envolvem processamentos no nó de origem e em todos

os nós onde a informação necessite ser lida ou verificada. Esses métodos tornam maiores as informações, que custarão mais energia para transmissão e roteamento, assim como os mecanismos para distribuição de chaves também envolverão maior tráfego de mensagens na rede.

Outro fator importante é o uso do modo de espera, onde os nós reduzem suas funções para a economia de energia. O comportamento do nó ao entrar ou deixar o modo de espera pode acarretar falhas na sincronização e troca de chaves. Nesse caso, o nó pode ficar impedido de trocar informações na rede.

De acordo com Hu e Sharma (2005), os desafios para implementações de mecanismos de segurança em uma RSSF incluem:

- **Minimizar o consumo de recursos vs. maximizar a segurança** - Embora os recursos sejam a pouca memória, capacidade de processamento e energia limitada, a energia talvez seja a maior restrição em uma RSSF. Com o processamento extra requerido pelas funções de segurança, o consumo energético influencia diretamente o nível de segurança obtido no nó sensor;

- **Topologia suscetível a ataques** - Ao contrário de redes estruturadas, com barreiras físicas e mecanismos como *firewalls* e dado a natureza da comunicação sem fio, ataques em uma RSSF podem surgir de todas as direções. Um intruso pode ainda danificar fisicamente os sensores, impedindo-os de realizar suas funções, pode capturar alguns dispositivos para a extração e análise posterior de informações sensíveis como chaves criptográficas, algoritmos em utilização, padrões de criptografias ou autenticações, e pode ainda substituir ou acrescentar nós maliciosos que efetuarão diversos tipos de ataques à rede;

- **Características da comunicação em RSSFs** - Com a comunicação dos nós apresentando alcance limitado, baixa taxa de transmissão e em alguns casos, unidirecional, os mecanismos existentes para redes cabeadas podem ser impraticáveis, exigindo soluções mais adequadas.

3.1 Tipos de Ataques

Segundo Anjum e Mouchtaris (2007), os ataques à redes de sensores podem ser classificados em duas categorias: ataques passivos ou ataques ativos. Um ataque é considerado passivo quando o nó atacante não interfere no funcionamento da rede, isto é, apenas monitora e realiza a captura dos dados que trafegam na rede. Caso contrário, quando os nós atacantes tem como objetivo prejudicar a rede, seja interrompendo a comunicação, falsificando dados ou sobrecarregando a rede, o ataque é considerado ativo.

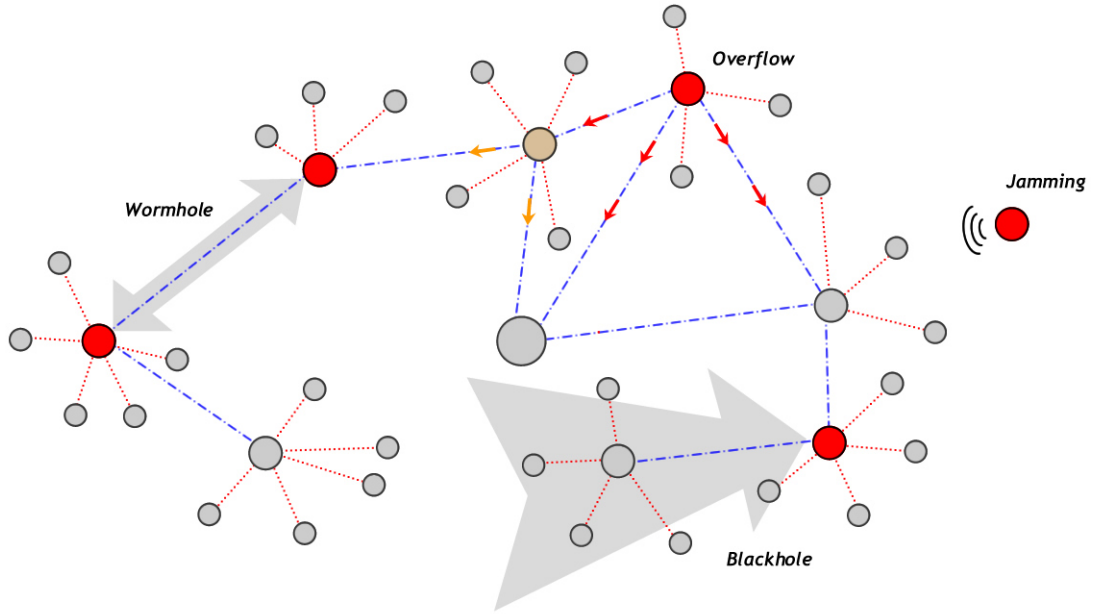

Figura 3.1: Alguns tipos de ataques, baseada em (Hu e Sharma, 2005)

Alguns tipos de ataques são específicos a determinadas camadas, sendo mais incisivos os ataques sobre as camadas mais baixas. Diferentes tipos de ataques podem ainda ser combinados, tornando mais difícil contornar o problema e exigindo melhores estratégias para defesa. A Figura 3.1 exibe alguns tipos comuns de ataque.

3.1.1 Ataques à Camada Física

Wu et al (2007) descrevem dois tipos de ataques relacionados à camada física. O primeiro tipo de ataque é a escuta não autorizada (*eavesdropping*), um tipo de ataque passivo. Devido ao fato das transmissões serem por rádio, um nó malicioso pode facilmente entrar na área de cobertura da rede de sensores e interceptar os dados trafegados. Para impedir

a interpretação das informações capturadas, deve-se fazer uso de criptografia, mas tal técnica não impede de fato a captura.

O segundo tipo de ataque é a interferência (*jamming*), classificado como ativo. Nesse ataque o nó malicioso emite sinais de rádio com o objetivo de sobrecarregar e interromper a comunicação da rede de sensores. Para minimizar o ataque, pode ser realizado um aumento da potência de transmissão dos nós objetivando a diminuição do ruído causado pelo nó malicioso. Porém, tal técnica diminui consideravelmente o tempo de vida da rede, já que uma grande parcela do consumo de energia corresponde à transmissão.

Mpitziopoulos *et al* (2009) descrevem outras técnicas como espalhamento de espectro com saltos de frequência para contornar o problema. Com uso de tal técnica, os canais utilizados para transmissão dos dados se alteram várias vezes por segundo de modo pseudoaleatório. A segurança da transmissão será elevada, uma vez que o nó malicioso, sem conhecimento da sequência de frequência utilizadas, necessitará de um esforço maior para análise dos dados transmitidos. Porém, cabe ressaltar que esses tipos de rádios são mais complexos e apresentam um maior consumo de energia.

3.1.2 Ataques à Camada de Enlace

Ataques à camada de enlace podem prejudicar a recepção correta dos dados. Esses ataques podem ser realizados através da indução de colisões e de transmissões de pacotes danificados, podendo resultar na retransmissão de mensagens continuamente por parte dos nós atacados.

Segundo Hu e Sharma (2005), a camada de enlace também está sujeita ao ataque de interferência. Nessa camada o ataque consiste na transmissão de cabeçalhos de *frames* da sub-camada *MAC* (*Media Access Control*), o que leva outros nós a identificarem a transmissão e considerarem o canal como ocupado, aguardando por alguns momentos até que seja efetuada uma nova tentativa de acesso ao meio. O ataque ainda busca identificar a topologia da rede através do monitoramento do tamanho, sequência das mensagens e de seu tempo de resposta.

Um segundo tipo de ataque à camada de enlace é a falsificação de pacotes de controle de acesso ao meio (*MAC spoofing*), onde um nó malicioso modifica seu endereço

de interface de rede na tentativa de se passar por um nó autorizado. Assim o nó é capaz de capturar *frames* destinados ao nó original, sendo também capaz de transmitir informações inválidas. Esse ataque leva a uma duplicação de endereços na rede, podendo causar um aumento de tráfego devido à retransmissões de dados. O ataque pode ser evitado utilizando-se mecanismos que garantam a autenticidade dos dados e cabeçalhos das mensagens transmitidas.

Um terceiro tipo de ataque é o de repetição (*replay attack*), cujo funcionamento pode se baseado em um ataque de homem do meio (*man-in-the-middle*). No ataque homem do meio, realizado através do ataque de falsificação citado anteriormente, um nó malicioso pode se passar por A para um nó B, e se passar por B para o nó A, interceptando a comunicação e capturando informações sensíveis. O nó A, por exemplo, pode enviar uma senha para o nó B, requisitando acesso a algum recurso. O nó malicioso repassa a comunicação e mantém armazenado tal dado, que posteriormente utilizará, ganhando o acesso ao recurso. Nesse caso, somente a criptografia dos dados não evita o ataque, já que o nó malicioso pode armazenar e fazer uso da mesma forma, sem que seja necessário o conhecimento da informação real. Pode ser feito uso de *nonces*, onde B antes de fornecer acesso ao recurso, solicita que uma informação distinta, *nonces*, seja criptografada pelo suposto nó A. O nó malicioso agora é incapaz de prosseguir o ataque por não possuir as chaves do nó A legítimo.

3.1.3 Ataques à Camada de Rede

As RSSFs devem focar no provimento de uma maior segurança na camada de rede pois os ataques à esta podem influenciar o desempenho, disponibilidade, confiabilidade e roteamento do tráfego. Segundo Campista e Duarte (2003), "esta é a mais afetada e a que causa maiores danos. Isso se deve a sua característica de transmissão ser por múltiplos saltos, o que obriga que os dados passem por nós intermediários até atingir o seu destino."

Um desses ataques é o buraco negro (*blackhole*), onde o nó malicioso forja mensagens de roteamento, convencendo os nós autênticos que ele tem o menor caminho para um dado destino. Como geralmente em uma RSSF o nó de destino tende a ser único, o nó malicioso pode concentrar todo o tráfego através dele. A partir daí, poderá descartar

os pacotes recebidos, impedindo a comunicação ou iniciar um ataque de homem do meio.

Em um segundo tipo de ataque semelhante ao buraco negro, o buraco cinza (*greyhole*) difere no fato de não descartar todos os pacotes, dificultando sua identificação uma vez que o nó pode entender que apenas alguns serviços ou outros nós estão indisponíveis.

Um terceiro tipo de ataque é o buraco de minhoca (*wormhole*), um túnel criado por dois ou mais nós maliciosos. Cada nó malicioso, também através de mensagens de roteamento forjadas, convence os nós vizinhos que a melhor rota para qualquer caminho é através do túnel criado. A partir desse ataque, todo o tráfego é direcionado através dos túneis criados, permitindo que os nós maliciosos capturem uma quantidade ainda maior de dados trafegados. Causam ainda um atraso na entrega dos pacotes, bem como podem isolar regiões da rede ao se negarem a repassar os pacotes recebidos.

Um quarto tipo de ataque que pode ser efetuado é o *loop*, quando nós maliciosos injetam rotas circulares na rede, fazendo com que a informação fique em roteamento contínuo até que os pacotes sejam descartados devido ao seu *TTL* (*Time to Live*).

Um quinto tipo de ataque é o de múltiplas identidades (*sybil*). Algumas redes podem aplicar mecanismos de redundância de rotas, garantindo que se um nó apresentar defeito, seja de seu conhecimento uma rota alternativa. Aproveitando-se dessa característica da rede, o nó malicioso apresenta-se com múltiplas identidades, fazendo com que os nós legítimos o adicionem múltiplas vezes em suas tabelas de roteamento.

Finalizando, um sexto tipo de ataque é o de inundação (*flood*), que é caracterizado como o envio de muitas falsas mensagens de roteamento. Dado que dispositivos de uma RSSF apresentam memória limitada e que portando o tamanho da tabela de roteamento dos nós também é, o nó acaba sobrescrevendo sua tabela original ao receber novas rotas, comprometendo o funcionamento da rede.

3.1.4 Ataques à Camada de Transporte

Segundo Hu e Sharma (2005), não há real necessidade de medidas de segurança na camada de transporte. Com as camadas de enlace e rede seguras, resta para a camada de transporte o trabalho habitual como controlar o fluxo, reordenar pacotes, recuperação de

erros. Aplicar medidas de segurança também nessa camada pode ser redundante e levar à um consumo ainda maior de energia, reduzindo o tempo de vida da rede.

Porém, a camada de transporte pode sofrer alguns tipos de ataques. Um desses é o ataque de inundação de pacotes *SYN* (*SYN flood*) em casos de protocolos de transportes como o TCP ou similares. Nesses protocolos há o procedimento de aperto de mão em três etapas (*three-way handshake*) para o estabelecimento de uma conexão entre eles. O nó origem envia uma mensagem *SYN*, e ao receber, o nó destino responde com uma mensagem *SYN-ACK*. O nó origem confirma então através de uma mensagem *ACK* para estabelecer a conexão. O ataque consiste no não envio da mensagem *ACK* pelo nó malicioso, fazendo com que o nó destino fique aguardando por algum tempo pela resposta. Com o envio de muitas mensagens *SYN*, o nó destino pode ter seus recursos esgotados, e assim ficando impossibilitado de responder aos nós legítimos.

Ainda, há o ataque de dessincronização em protocolos como o TCP, que garantem entrega ordenada utilizando números de sequência para controle. Nesse ataque, o nó malicioso se passando por outro, envia mensagens de controle com números de sequência falsos, fazendo com que o nó origem retransmita pacotes, consumindo energia do nó.

3.2 Segurança no Roteamento

Baseando-se nos objetivos de segurança, desafios e potenciais ataques em RSSF vistos anteriormente, e segundo Hu e Sharma (2005), uma das questões-chave para a segurança das redes é um bom mecanismo de segurança no nível de roteamento. E o roteamento de dados em RSSF possui alguns desafios característicos.

A topologia é um fator complicador devido a uma grande quantidade de nós "(tipicamente considera-se implantações com centenas a milhares de nós)". Ainda, há a possibilidade de mudança de topologia devido a nós sem energia, adição de novos nós, uso de modo de espera, ou simplesmente com a mobilidade dos sensores. (Margi *et al*, 2009)

Muitos protocolos de roteamento de RSSF são bastante simples e por isso tornam-se mais suscetíveis a ataques (Karlof e Wagner, 2002), haja vista os ataques à camada de rede descritos anteriormente, que baseiam-se na falsificação, supressão ou inundação da rede com pacotes de roteamento. De acordo com Karlof e Wagner (2002), a maioria dos

ataques externos podem ser evitados com a utilização de criptografia na camada de enlace e autenticação com uso de chaves compartilhadas. Porém, tais mecanismos são completamente ineficazes na presença de ataques internos, através de nós comprometidos. Nesse caso, mecanismos mais sofisticados são necessários para fornecer uma razoável proteção.

Segundo Campista e Duarte (2003), os algoritmos de melhor desempenho são aqueles que apresentam uma boa proteção ao mesmo tempo que consomem o mínimo de energia possível. As primeiras soluções incluíam mecanismos de segurança em protocolos de roteamento para redes *ad-hoc*, como o protocolo *AODV* (*Ad-hoc On-Demand Distance Vector*) e o protocolo *DSR* (*Dynamic Source Routing*). Visando um melhor desempenho, começaram a surgir soluções com protocolos que foram concebidos já considerando a segurança, onde em Campista e Duarte (2003) são destacados alguns, como *INSENS* (*INtrusion-tolerant routing protocol for wireless SEnsor NetworkS*), *Ariadne* e o de maior aceitação, *SPINS* (*Security Protocols for Sensor Networks*).

É colocado ainda em Hu e Sharma (2005), duas outras questões-chaves para a segurança. A primeira é o gerenciamento de chaves criptográficas, que é difícil devido a topologia de RSSFs, conectividade intermitente e às limitações dos nós sensores. A outra é a prevenção de ataques de negação de serviço, quem tem como objetivo diminuir ou eliminar a capacidade de funcionamento da rede.

Estudos na área mostram-se necessários pois o provimento de segurança em RSSFs é complexo devido aos recursos limitados dos nós sensores, como o uso de criptografia que resulta em um maior consumo de energia. Portanto, é necessário estabelecer, diante da aplicação desejada, quais requisitos de segurança são relevantes e que devem ser levados em consideração na fase de concepção da rede. Ainda, devem ser analisados os impactos que esses mecanismos poderão causar, como por exemplo, um atraso na comunicação.

4 Simulação de um Ataque *RREQ Flood*

Apesar dos recentes avanços na área de redes de sensores, ainda há muito a ser pesquisado e desenvolvido. Segundo Silva Filho *et al* (2009), a dificuldade de desenvolvimento ocorre devido ao alto custo de implantação de uma RSSF para estudos, sendo o uso de simuladores a alternativa mais viável e difundida.

Os simuladores são ferramentas que possibilitam os estudos de diversos cenários, avaliação de técnicas e testes com novos protocolos. Permitem ainda simulações com hardwares ainda não existentes no mercado e de cenários extremamente complexos, difíceis de serem estudados em um ambiente real, tudo isso com a obtenção de resultados que podem ser bem próximos do reais.

Dentre os simuladores disponíveis atualmente no mercado para simulação de redes de sensores, alguns pontos foram avaliados para a escolha do simulador a ser utilizado. Foram levados em consideração a disponibilidade do código-fonte, existência de documentação, linguagem utilizada, modelos de consumo de energia e desenvolvimento constante. Assim, com base os dados apresentados na Tabela 4.1, observou-se que o *NS-2* (*Network Simulator 2*) e o *OMNet++* são os mais indicados. Dentre esses, o *NS-2* foi escolhido para esse trabalho por ser bastante difundido e conhecido dentro do ambiente acadêmico, contando com ferramentas que auxiliam todo o processo de experimentação.

	NS-2	**OMNet++**	**TOSSIM**	**QualNet**
Código-fonte	Sim	Sim	Sim	Não
Linguagem	C++/Tcl	C++	C++/Python	C++
Documentação	+++	++	++	+
Modelos de Energia	Sim	Sim	Sim	Sim
Última Versão	2011	2011	2007	2010

Tabela 4.1: Simuladores Disponíveis

O *NS-2* [1], desenvolvido em C++ e Tcl, é resultado de um projeto financiado pela DARPA (*Defense Advanced Research Projects Agency*) e NSF (*National Science Foundation*) que tinha como objetivo a construção de um simulador de redes que possibilitasse o

[1] http://nsnam.isi.edu/nsnam/

estudo de protocolos existentes e auxiliasse na criação de novos, sendo desenvolvido inicialmente pela universidade de Berkeley, ISI (*Information Sciences Institute*) e Xerox. É capaz de simular variados tipos de redes, com ou sem fio, diversos protocolos e simulações de tráfego como FTP, Web, Telnet etc.

O simulador é baseado em módulos e componentes e a simulação é realizada através de scripts Tcl, que através de comandos, geram o cenário desejado. Tem como saída *traces*, onde estão contidos todos os eventos ocorridos durante a simulação e um arquivo *nam*, com informações que podem ser consumidas pelo programa Nam (*Network Animator*), desenvolvido também pelo grupo e responsável pela visualização dos eventos ocorridos.

4.1 Definição do Ataque Simulado

Um dos protocolos de roteamento utilizados em RSSFs é o *AODV* (*Ad-hoc On-Demand Distance Vector*), que é um protocolo adaptativo proposto para cenários de alta mobilidade, com baixo processamento e consumo de banda. A tabela de rotas é simples, armazenando apenas o próximo salto para o nó em questão e o protocolo é reativo, isto é, as descobertas de rotas são realizadas quando é necessária a comunicação com um nó destino cuja rota é desconhecida.

O controle de rotas é realizado através de pacotes especiais: *RREQ* (*Route Request*), para pedidos de rotas; *RREP* (*Route Response*) para a resposta aos pedidos de rotas; e *RERR* (*Route Error*) para indicação de rotas indisponíveis. A Figura 4.1 ilustra o funcionamento básico do protocolo.

1. Um nó A deseja se comunicar com o nó C. Verifica sua tabela e descobre que não conhece rotas para o nó desejado. Envia um pacote *RREQ* via *broadcast*, pedindo por uma rota para C.

2. O nó B receba o pedido, verifica sua tabela e descobre que conhece uma rota. Envia em seguida um pacote *RREP* para A, com as informações.

3. O nó A recebe o *RREP* e atualiza sua tabela de rotas.

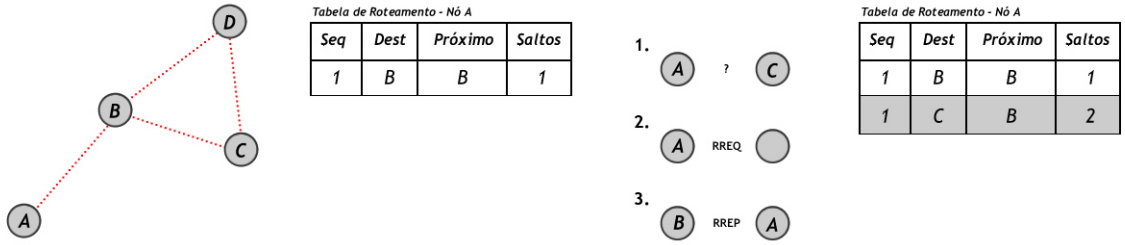

Figura 4.1: Descoberta de Rotas no AODV

Dado a memória limitada dos nós sensores, a comunicação de controle de rotas é constante e necessária. Como informado na subseção 3.1.3, a camada de rede é a mais vulnerável a ataques e causa grandes danos. Por ser simples e de fácil utilização em um rede de sensores real, o ataque escolhido para a simulação foi o *RREQ Flooding*. Este é um ataque de negação de serviço que visa a inundação da rede através de uma grande quantidade de requisições de rotas. Nós maliciosos podem gerar de centenas à milhares de pacotes *RREQ*, saturando a rede e impedindo que requisições genuínas sejam processadas e respondidas, impossibilitando o roteamento.

4.2 Cenário Simulado

A mineração é um importante setor da economia brasileira. De acordo com os dados publicados pelo IBRAM (2012) (Instituto Brasileiro de Mineração), o setor foi responsável por 10% do PIB (Produto Interno Bruto) em 2010, com um faturamento de US$ 157 bilhões e o PNM 2030 (Plano Nacional de Mineração 2030) prevê investimentos de US$ 350 bilhões em pesquisas para expansão ou descobertas de jazidas ou aberturas de novas minas e unidades de transformação mineral para os próximos 20 anos. Ainda, de acordo com o IBRAM (2012), no período de 2011 a 2015, o setor privado investirá US$ 68,5 bilhões na atividade mineral.

Segundo PORMIN - MME (2012), os métodos de lavra mais utilizados no Brasil são de 2 tipos, a céu aberto e subterrânea e segundo Germani (2012), dentre o tipo de lavra subterrânea, um dos métodos utilizados é o de câmara e pilares, sendo empregado em algumas minas como:

- **Mina Urucum** - Extração de manganês em Corumbá - MS, da Companhia Vale do Rio Doce;
- **Mina Morro Agudo** - Extração de zinco e chumbo em Paracatu - MG, da Companhia Mineira de Metais;
- **Mina Taquari-Vassouras** - Extração de potássio em Rosário do Catete - SE, da Companhia Vale do Rio Doce;

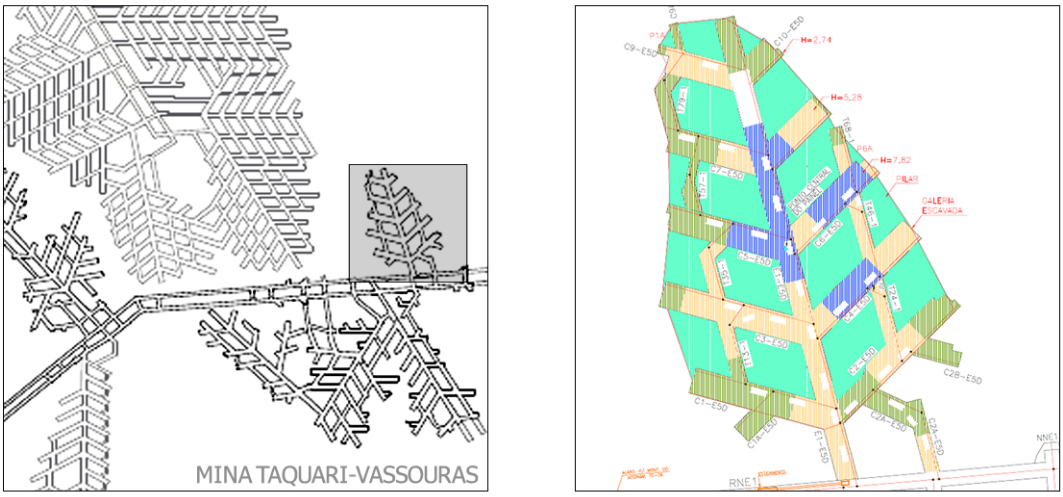

Figura 4.2: Planta da mina subterrânea Taquari-Vassouras. Destaque em um dos painéis de extração. (Fontes e Pinto, 2004)

A Figura 4.2 mostra uma parte da planta da mina Taquari-Vassouras e destaca um dos painéis de extração de minério, aproximando a visualização das câmaras e pilares.

Para fins de simulação, foi gerado um cenário simplificado, Figura 4.3, obedecendo as médias das medidas do painel destacado anteriormente. As câmaras apresentam uma largura de 15 metros e os pilares apresentam 30 metros de aresta.

Para o cenário base, foram distribuídos 52 nós sensores, distantes 8,5 metros um do outro e alinhados ao centro de cada câmara, isto é, distantes 7,5 metros de cada pilar. Essas distâncias têm como objetivo reduzir a interferência causada pelas reflexões através das paredes, diminuindo a zona de detecção do sensor através do parâmetro de limiar de recepção no *NS-2*.

Em uma aplicação real, não podemos garantir uma distribuição perfeita dos nós. Afim de aproximar a simulação à realidade, foram gerados 50 novos cenários em cima

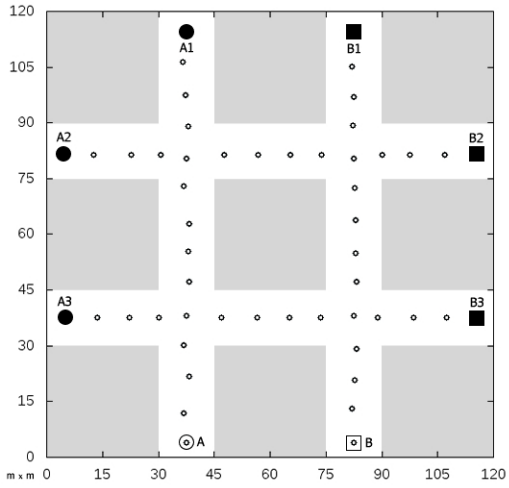

Figura 4.3: Generalização do painel de extração.

do cenário base, onde a posição dos nós foi variada aleatoriamente em um intervalo de 1 metro, fazendo com que os nós acabassem distantes entre 6,5 e 10,5 metros.

Cinco situações foram simuladas: sem ataques, 3 nós atacantes (5,7%), 6 nós atacantes (11,5%), 9 nós atacantes (17,3%) e 12 nós atacantes (23,0%). Para cada uma, foram utilizados 10 dos cenários gerados anteriormente, onde cada cenário foi simulado duas vezes com sementes do NS-2 distintas. A escolha de quais seriam os nós atacantes também foi aleatória para cada simulação. Assim, para cada situação foram obtidos 20 resultados distintos.

Os cenários foram simulados com a utilização dos protocolos TCP, que pode se assemelhar à um monitoramento ambiental, onde é necessário a obtenção de todos os eventos que ocorreram para análises estatísticas e UDP, que pode se assemelhar à uma aplicação de controle industrial cujos valores devem ser obtidos em tempo real para tomadas de decisões. Nas simulações com TCP, a fonte de tráfego utilizada foi o modelo FTP e nas com UDP, uma fonte CBR. Em ambos os protocolos haviam 6 fontes de tráfego, visíveis na Figura 4.3, com origem nos nós A1, A2 e A3 e destino ao nó sorvedouro A e com origem nos nós B1, B2 e B3 e destino ao nó sorvedouro B. Esses nós foram escolhidos por estarem localizados nas extremidades das câmaras, fazendo com que os dados trafegassem através da maior quantidade de nós possíveis.

4.3 Definição das Métricas

De posse dos cenários criados, algumas métricas foram escolhidas para análise e estudo dos resultados. Definidas, scripts foram executados posteriormente sobre os arquivos de *trace* gerados nas simulações para a extração dos dados.

- **AODV** - Quantidade de pacotes do *AODV*. Foram observadas a quantidade de pacotes enviados e de pacotes recebidos, explicitando a quantidade de pacotes *RREQ* em cada uma das medidas;

- **Vazão de Entrega** - Relação entre pacotes de dados enviados e recebidos. Definida pela equação:

$$taxa\ entrega = \frac{pacotes\ de\ dados\ recebidos}{pacotes\ de\ dados\ enviados}$$

- **Dados Enviados** - Quantidade de pacotes de dados enviados.

- **Média de Atraso Fim a Fim** - Para cada pacote recebido, calculou-se o intervalo de tempo entre a saída do nó origem e chegada ao nó destino. Após, calculou-se a média de atraso da rede.

$$atraso_{pacote} = hora\ de\ sa\acute{i}da_{destino} - hora\ de\ chegada_{origem}$$

$$m\acute{e}dia\ de\ atraso = \frac{\sum_i^n atraso_{pacote_i}}{n}, onde\ n\ =\ total\ de\ pacotes\ recebidos.$$

- **Média de Consumo** - Com base em Castro (2010), foi definida uma energia inicial para os nós sensores e após a simulação, a energia de cada nó foi medida e seu consumo calculado através das equações:

$$consumo_{n\acute{o}} = energia\ inicial - energia\ final$$

$$consumo_{rssf} = \frac{\sum_i^{52} consumo_{n\acute{o}_i}}{52}, onde\ 52\ =\ total\ de\ n\acute{o}s.$$

4.4 Avaliação dos Resultados

Com os dados extraídos dos arquivos de *trace* de cada simulação, foram calculadas as médias e desvios padrões. As Tabela 4.2 e Tabela 4.3 apresentam os resultados das simulações TCP e UDP, respectivamente, com os desvios padrões entre parênteses.

Simulações TCP					
Quant. de Nós Atacantes	**0%**	**5,7%**	**11,5%**	**17,3%**	**23,0%**
AODV Enviados	4257 *(1112)*	5150 *(1068)*	5972 *(1260)*	7326 *(1048)*	7939 *(1837)*
RREQ Enviados	3864 *(1041)*	4827 *(950)*	5394 *(1141)*	6807 *(1109)*	7255 *(1863)*
AODV Recebidos	8988 *(2542)*	10823 *(2378)*	12445 *(2754)*	15226 *(2295)*	16307 *(4051)*
RREQ Recebidos	8255 *(2392)*	10171 *(2344)*	11337 *(2511)*	14182 *(2385)*	14980 *(4111)*
Vazão de Entrega (*%*)	72,232 *(4,52)*	71,101 *(3,35)*	69,481 *(3,45)*	67,901 *(3,59)*	63,557 *(1,91)*
Dados Enviados	501 *(189)*	486 *(186)*	472 *(169)*	424 *(100)*	356 *(115)*
Atraso Fim a Fim (*ms*)	1307,586 *(165,61)*	1458,765 *(266,91)*	1649,645 *(340,30)*	1747,950 *(342,51)*	2258,779 *(458,43)*
Média de Consumo (*J*)	0,271 *(0,017)*	0,281 *(0,010)*	0,292 *(0,026)*	0,296 *(0,021)*	0,316 *(0,014)*

Tabela 4.2: Resultados das Simulações FTP/TCP

Simulações UDP					
Quant. de Nós Atacantes	**0%**	**5,7%**	**11,5%**	**17,3%**	**23,0%**
AODV Enviados	1301 *(1094)*	1928 *(930)*	2175 *(1099)*	2611 *(922)*	3417 *(920)*
RREQ Enviados	1242 *(1103)*	1807 *(950)*	2106 *(1111)*	2429 *(903)*	3164 *(976)*
AODV Recebidos	2711 *(2418)*	4110 *(2070)*	4640 *(2226)*	5689 *(2403)*	7224 *(2836)*
RREQ Recebidos	2543 *(2441)*	3805 *(2126)*	4433 *(2267)*	5009 *(2307)*	6535 *(2689)*
Vazão de Entrega (*%*)	69,422 *(16,23)*	64,986 *(20,50)*	45,624 *(13,15)*	38,777 *(18,87)*	33,847 *(17,46)*
Dados Enviados	2138 *(0)*	2138 *(0)*	2138 *(0)*	2138 *(0)*	2138 *(0)*
Atraso Fim a Fim (*ms*)	318,273 *(63,66)*	330,741 *(51,73)*	358,123 *(96,34)*	438,873 *(90,06)*	541,499 *(118,16)*
Média de Consumo (*J*)	0,270 *(0,012)*	0,283 *(0,013)*	0,293 *(0,018)*	0,308 *(0,021)*	0,315 *(0,022)*

Tabela 4.3: Resultados das Simulações CBR/UDP

4.4.1 AODV

As Figuras 4.4 e 4.5 exibem as variações da quantidade de pacotes AODV durante os cenários. Pode-se perceber que o aumento de nós maliciosos reflete na quantidade de pacotes *RREQ* enviados, confirmando o funcionamento do ataque. Nota-se também que a variação de pacotes recebidos tem uma taxa de crescimento maior que a variação de pacotes enviados. Isso se deve à característica *broadcast* do protocolo AODV, onde o pacote enviado por um nó é recebido por vários.

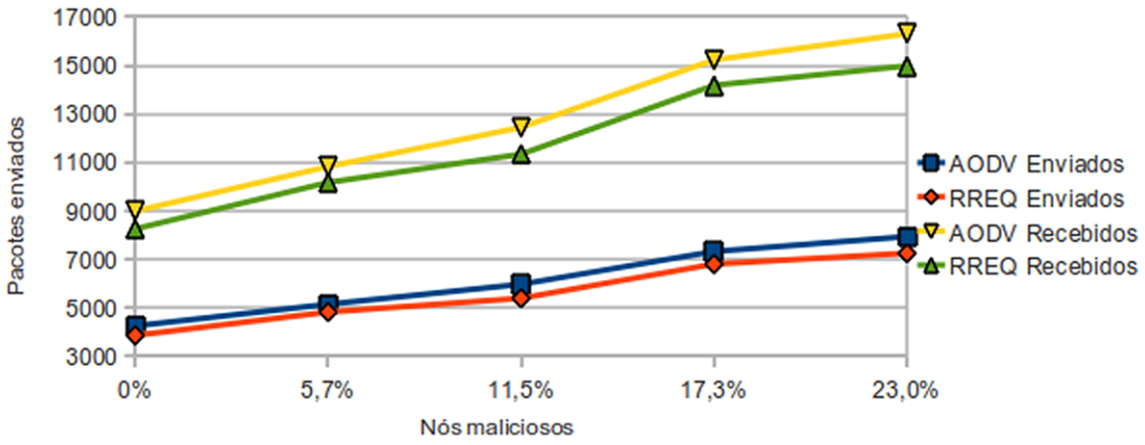

Figura 4.4: Variação de pacotes AODV nos cenários TCP

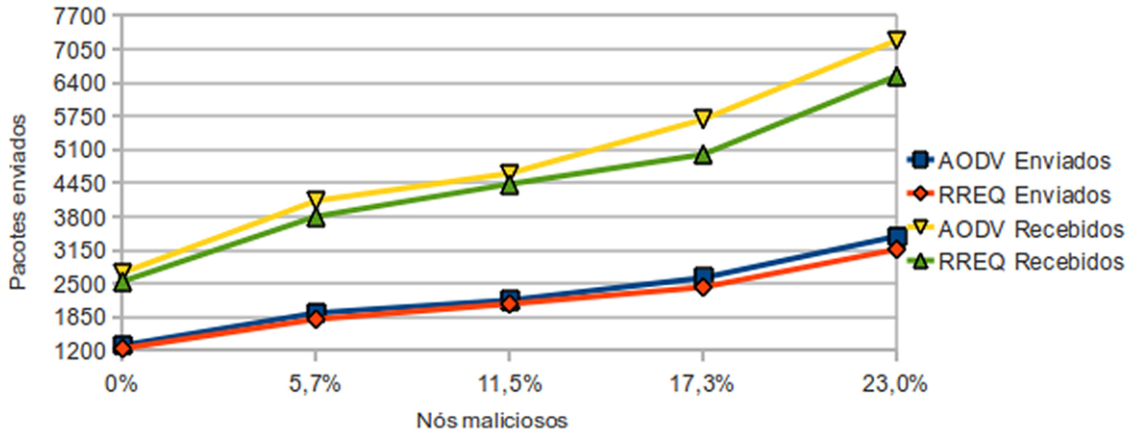

Figura 4.5: Variação de pacotes AODV nos cenários UDP

4.4.2 Vazão de Entrega

Com os ataques, muitos pacotes de dados acabam sendo perdidos durante as transmissões devido à sobrecarga e a Figura 4.6 e a Figura 4.7 apresentam os impactos na vazão de entrega. Nota-se que nos cenários TCP a queda foi menos acentuada devido ao controle de erros do protocolo. Dado que o protocolo UDP não apresenta esse controle, era esperada e pode ser visualizada uma queda considerável da vazão de entrega nesses cenários.

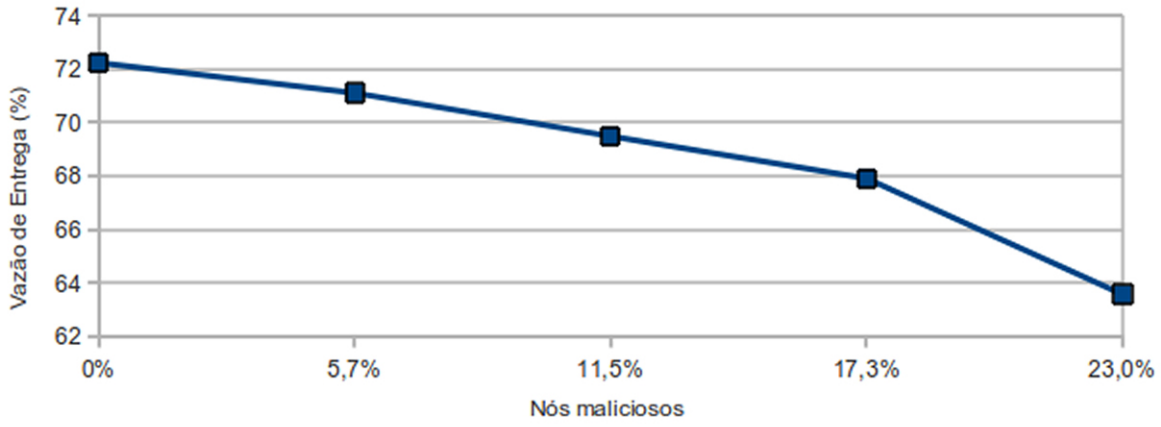

Figura 4.6: Variação da vazão de entrega nos cenários TCP

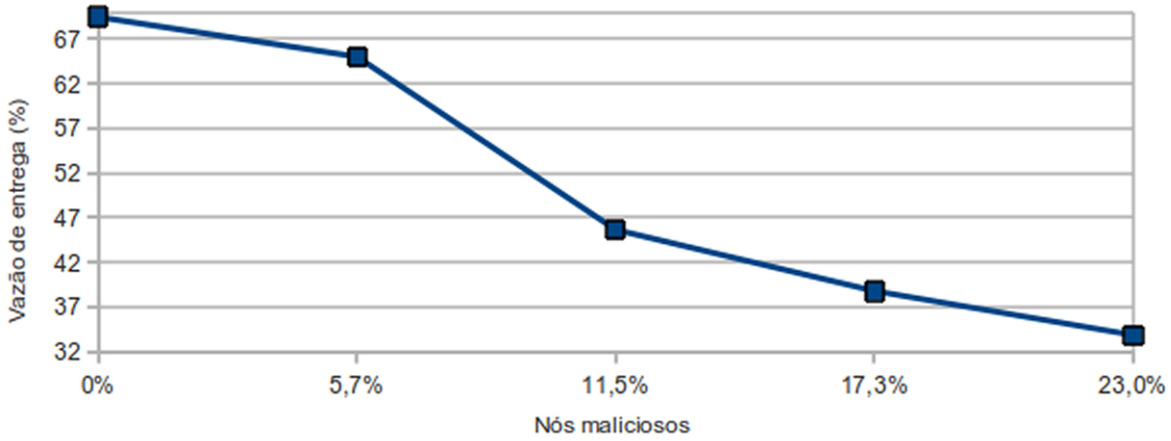

Figura 4.7: Variação da vazão de entrega nos cenários UDP

4.4.3 Dados Enviados

As quantidades de pacotes enviados no intervalo de tempo simulado são exibidas na Figura 4.8 para TCP e na Figura 4.9 para UDP. Nos cenários TCP, nota-se que com os ataques dos nós maliciosos, a quantidade de dados na rede diminuiu. Impactada pela sobrecarga da rede causada pelos ataques, a redução é devido à natureza do protocolo TCP, que é orientado à conexão e envolve retransmissões e esperas por respostas. Por outro lado, nos cenários UDP visualiza-se um envio constante, uma vez que o UDP, ao contrário do TCP, não é orientado à conexão e não apresenta nenhum controle de erros, enviando fluxos de dados continuamente sem garantias de entrega.

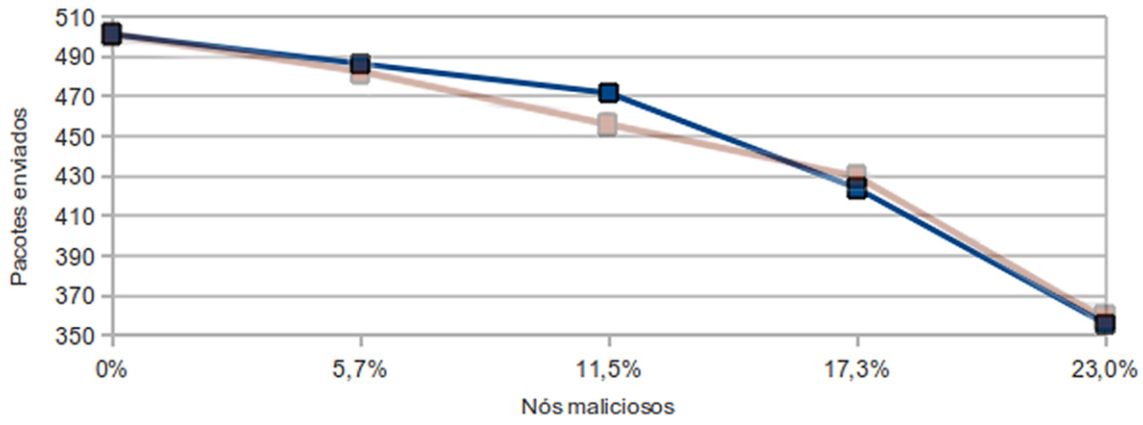

Figura 4.8: Variação de pacotes de dados nos cenários TCP, em contraste com o gráfico da variação da vazão de entrega

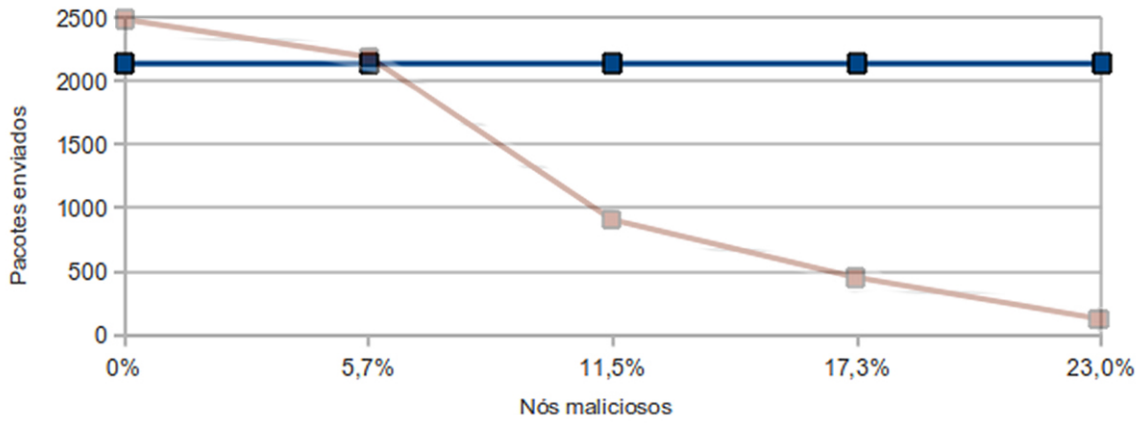

Figura 4.9: Variação de pacotes de dados nos cenários UDP, em contraste com o gráfico da variação da vazão de entrega

4.4.4 Atraso Fim a Fim

Visualiza-se na Figura 4.10 e Figura 4.11, os atrasos fim a fim dos pacotes transmitidos. Com a sobrecarga causada pelos ataques, pacotes são perdidos e reenviados nos cenários TCP, impactando no intervalo de tempo entre o envio dos pacotes e a chegada ao destino. Ainda, o custo e a espera por acesso ao meio aumentam em ambos os cenários.

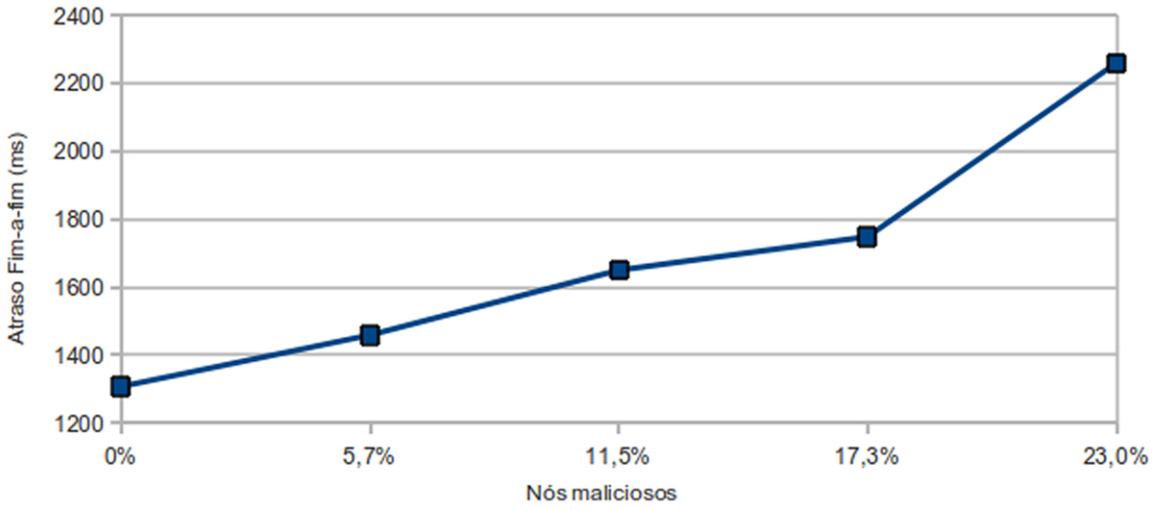

Figura 4.10: Variação do atraso fim a fim nos cenários TCP

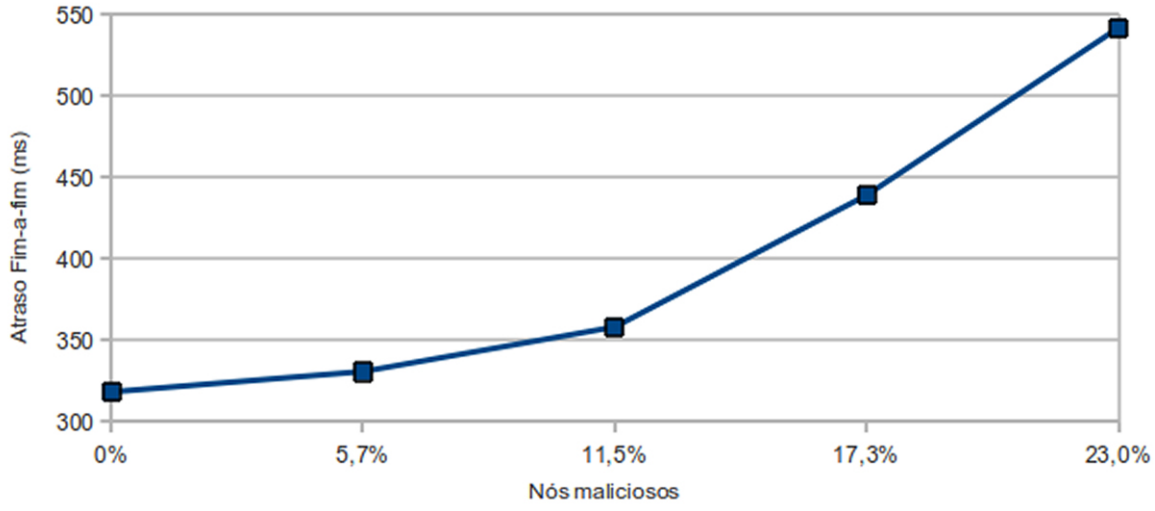

Figura 4.11: Variação do atraso fim a fim nos cenários UDP

4.4.5 Consumo de Energia

A energia dos nós sensores é limitada e é importante um baixo consumo para que o tempo de vida seja prolongado. A Figura 4.12 bem como a Figura 4.13 exibem as variações do consumo de energia da rede e nota-se que as taxas de crescimento nos cenários TCP e UDP são parecidas. O aumento do consumo nos cenários UDP e TCP está relacionado com o aumento dos ataques à rede, exigindo maior processamento dos nós. Nos cenários TCP, a quantidade de pacotes de dados enviados diminui, porém, as retransmissões de pacotes aumentam devido à sobrecarga.

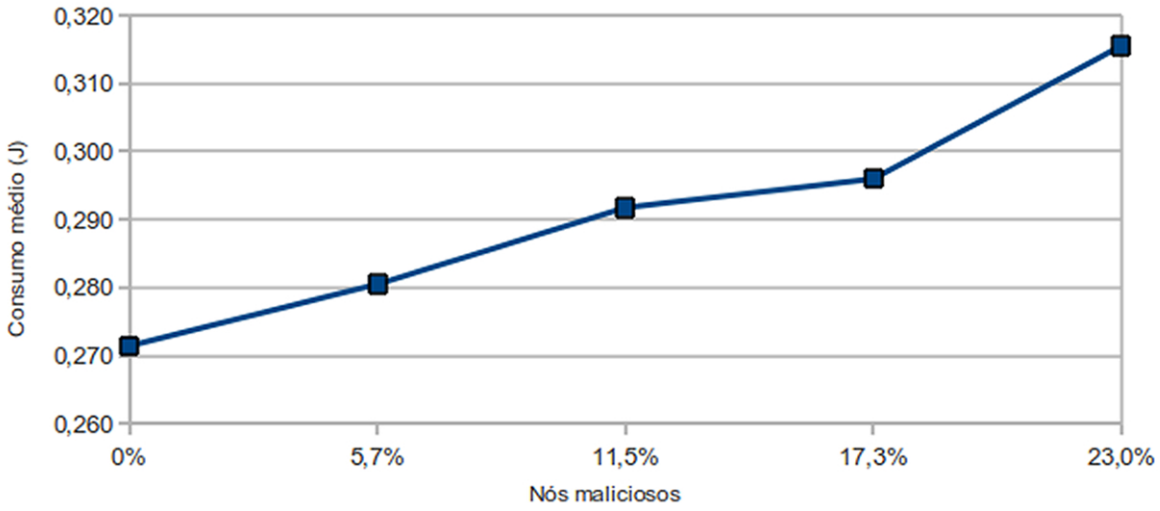

Figura 4.12: Variação do consumo de energia nos cenários TCP

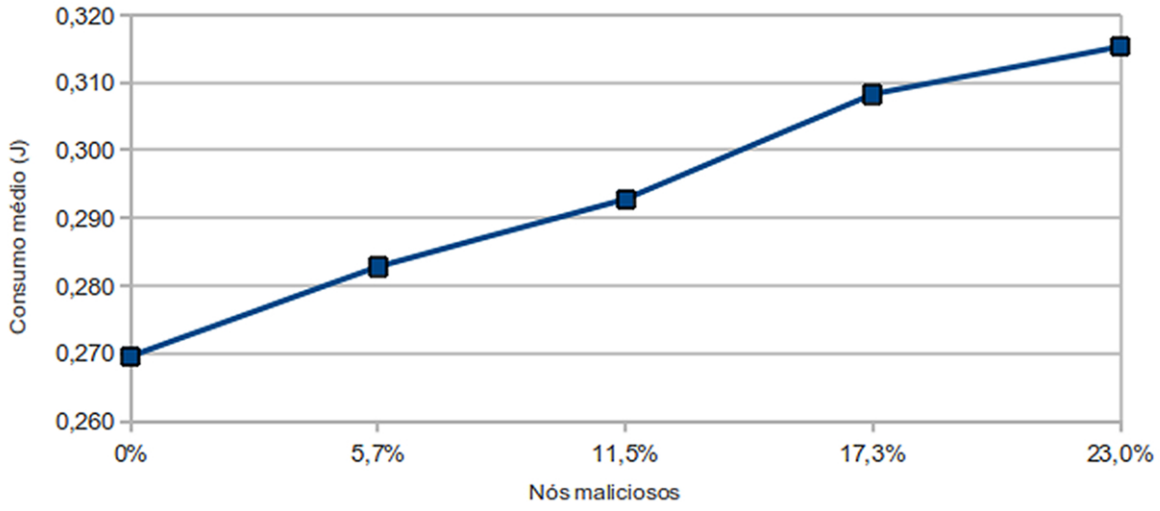

Figura 4.13: Variação do consumo de energia nos cenários UDP

4.4.6 Resumo dos Resultados

Dado o ataque *RREQ Flooding*, mostrou-se que cenários TCP e UDP sofrem efeitos diferentes, relacionados com suas características. Nos cenários TCP, a vazão de entrega diminui levemente, podendo entregar ainda 62% no pior dos cenários. Porém, houve redução da quantidade de dados enviados e um aumento no atraso fim a fim. Dependendo do objetivo da RSSF, esse atraso pode impactar fortemente no seu propósito. Nos cenários UDP, embora o envio de dados não tenha sido impactado, a vazão de entrega foi extremamente comprometida, podendo também causar um forte impacto no objetivo da RSSF. Ainda, o aumento do consumo de energia em ambos os cenários leva à diminuição do tempo de vida da rede podendo comprometer também o objetivo.

Há algumas pesquisas que apresentam possíveis soluções à esse tipo de ataque, que devem ser analisadas e implementadas de acordo com o cenário. Uma dessas possíveis soluções é ao estabelecimento de um limite máximo de *RREQ* por intervalo de tempo. Esta poderia ter sido aplicada mais facilmente no cenário simulado, uma vez que os nós não são móveis. Por outro lado, em cenários de alta mobilidade, os nós podem atingir o limite facilmente e perderem o contato até que possam enviar novas requisições, prejudicando o objetivo da rede.

Apesar da mobilidade influenciar na quantidade de *RREQ*, este não é o único fator. Dado que a memória dos sensores é limitada e que há uma grande quantidade de nós na rede, as informações de roteamento sobre os nós vizinhos podem não caber na memória de um nó sensor, exigindo *RREQ*s constantemente para obtenção de rotas que foram sobrescritas.

A solução anterior pode ser aperfeiçoada aplicando-se limites individuais para cada nó, de acordo com a média dos nós na rede e detectando nós com comportamento fora do padrão. Esses aperfeiçoamentos estão diretamente relacionados com o cenário da aplicação e com o objetivo da rede e devem ser bem analisados, dado que podem causar efeitos contrários, como por exemplo, provocar um atraso ainda maior nos pacotes em cenários com necessidade de coleta e processamento em tempo real.

5 Conclusões

Foram apresentados nesse trabalho os conceitos de redes de sensores sem fio e seus grandes desafios, destacando-se a limitada quantidade de energia. Devido às limitações e características peculiares, protocolos gerais de redes estruturadas ou redes *ad-hoc* devem ser otimizados ou até mesmo devem ter suas abordagens repensadas, seja através de algoritmos melhores ou abrindo mão de certas características para obtenção de outras. Ainda que os dispositivos físicos apresentem evolução, energia ainda será um problema fundamental.

Não há métricas consideradas corretas para a análise da qualidade de uma RSSF. Métricas como consumo energético, vazão de entrega, atraso fim a fim, devem ser analisadas caso a caso, de acordo com o real propósito da rede.

Uma RSSF apresenta uma alta complexidade e mostrou-se que o estudo e planejamento são essenciais para um correto funcionamento. Deve-se fazer uso sempre que possível de ferramentas de simulação, ainda que não sejam completamente fiéis ao ambiente real, para averiguação do funcionamento esperado. Embora existam diversos simuladores disponíveis, há diversos protocolos que ainda não foram implementados e contribuições são desejáveis.

Há diversos ataques à qual uma RSSF pode sofrer, e devido aos recursos limitados, um cenário com segurança ideal ainda é utópico. Como mecanismos de segurança requerem um maior esforço computacional, devem ser estudados e projetados de modo a não comprometer o funcionamento da rede. Algumas otimizações em processos de criptografia podem reduzir significantemente o esforço adicional.

Com a simulação de um ataque *RREQ Flood*, foi possível visualizar o impacto que poucos nós podem causar à uma rede. As simulações mostram que a rede pode ter seu tempo de vida reduzido significativamente, e que em um ataque pode ocorrer até 66% de perdas de pacotes. Ainda, o atraso fim a fim pode sofrer um aumento de até 72%, podendo impactar no objetivo da rede de sensores sem fio.

Como sugestão para trabalhos futuros, é proposto a implementação de algumas soluções contra o ataque *RREQ Flood* e análise dos efeitos destas, como consumo de

energia, atraso fim a fim e vazão de entrega. A partir desses efeitos será possível decidir sobre quais soluções são melhores para determinado cenário.

Ainda, é proposto a simulação de cenários que utilizem criptografia para garantir integridade e autenticidade, premissas de segurança apresentadas no início do capítulo 3. Tais premissas são importantes para evitar alguns tipos de ataques e a análise das métricas com e sem as premissas podem indicar o impacto dessa segurança.

Referências Bibliográficas

Akyildiz, I. F.; Su, W.; Sankarasubramaniam, Y. ; Cayirci, E. Wireless sensor networks: a survey. **Computer Networks**, v.38, p. 393–422, 2002.

Anjum, F.; Mouchtaris, P. **Security for Wireless Ad Hoc Networks**. John Wiley & Sons, 2006.

Campista, M. E. M.; Duarte, O. C. M. B. Segurança em redes de sensores. **Universidade Federal do Rio de Janeiro**, 2003.

Castro, B. P. **Redes de Sensores Sem Fio (RSSF)**. Departamento de Ciência da Computação - Universidade Federal de Juiz de Fora, 2010. Monografia.

Fontes, S. L.; Pinto, C. L. L. **Planejamento da Lavra dos Pilares do Painel E5D da Mina subterrânea de Potássio de Taquarí-Vassouras - CVRD.** In: III Congresso Brasileiro de Lavra a Céu Aberto e Lavra Subterrânea, Belo Horizonte - MG, 2004.

Darcy José Germani. **A MINERAÇÃO NO BRASIL**, Maio 2002.

Gerheim, W. K. M. **Estudo comparativo dos protocolos de roteamento seguro de redes em malha sem fio**. Departamento de Ciência da Computação - Universidade Federal de Juiz de Fora, 2010. Monografia.

Hu, F.; Sharma, N. K. Security considerations in ad hoc sensor networks. **Ad Hoc Networks**, v.3, n.1, p. 69–89, 2005.

Instituto Brasileiro de Mineração. Retrospectiva 2011. **Indústria da Mineração**, Janeiro 2012.

Karlof, C.; Wagner, D. **Secure routing in wireless sensor networks: Attacks and countermeasures**. In: In First IEEE International Workshop on Sensor Network Protocols and Applications, p. 113–127, 2002.

Kurose, J. F.; Ross, K. W. **Redes de Computadores e a Internet: Uma abordagem top-down**. Addison Wesley, 2005.

Libelium. **Waspmote**. `http://www.libelium.com/products/waspmote`. Acessado em Maio/2012.

Loureiro, A. A. F.; Nogueira, J. M. S.; Ruiz, L. B.; de Freitas Mini, R. A.; Nakamura, E. F. ; Figueiredo, C. M. S. **Redes de Sensores Sem Fio**. In: XXI Simpósio Brasileiro de Redes de Computadores, Natal - CE, 2003.

Margi, C. B.; Jr., M. A. S.; de B. Carvalho, T. C. M. ; Barreto, P. S. L. M. **Minicursos: SBSEG 2009 / IX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais**, p. 149–194. Sociedade Brasileira de Computação, Porto Alegre - RS, 2009.

Mateus, G. R.; Loureiro, A. A. F. **Introdução a computação móvel**. DCC/IM, COPPE/UFRJ, 1998.

Mpitziopoulos, A.; Gavalas, D.; Konstantopoulos, C. ; Pantziou, G. A survey on jamming attacks and countermeasures in wsns. **Communications Surveys Tutorials, IEEE**, v.11, n.4, p. 42–56, 2009.

Portal de Apoio ao Pequeno Produtor Mineral - Ministério de Minas e Energia. **Métodos de Lavra**, Fevereiro 2008.

Pereira, M. R.; de Amorim, C. L. ; de Castro, M. C. S. Tutorial sobre Rede de Sensores. **Cadernos do IME**, v.14, p. 39–53, 2003.

Ruiz, L. B. **MANNA: A Management Architecture for Wireless Sensor Networks**. 2003. Tese de Doutorado - Departamento de Ciência da Computação - Universidade Federal de Minas Gerais.

Silva, R. C. **Redes de Sensores Sem Fio**. Departamento de Ciência da Computação - Universidade Estadual de Montes Claros, 2006. Monografia.

Silva Filho, P. R. S.; Pinheiro, R. G.; Medeiros, A. C. ; Frery, A. C. **Análise da Confiabilidade dos Simuladores Network Simulator e OMNeT++**. In: Anais da IX Escola Regional de Computação Bahia Alagoas Sergipe (IX ERBASE), Ilhéus - BA, 2009.

Tilak, S.; Abu-Ghazaleh, N. B. ; Heinzelman, W. A Taxonomy of Wireless Micro-Sensor Network Models. **ACM MOBILE COMPUTING AND COMMUNICATIONS REVIEW**, v.6, p. 28–36, 2002.

Wu, B.; Chen, J.; Wu, J. ; Cardei, M. **A survey of attacks and countermeasures in mobile ad hoc networks**. In: Wireless Network Security, Signals and Communication Technology, chapter 5, p. 103–135. Springer US, 2007.

ZigBee Alliance. **Standards**. `http://www.zigbee.org/About/AboutTechnology/Standards.aspx`. Acessado em Janeiro/2012.

ZigBee Alliance. **The Alliance**. `http://www.zigbee.org/About/AboutAlliance/TheAlliance.aspx`. Acessado em Janeiro/2012.

A Apêndice

A.1 Alteração do protocolo AODV

Para a simulação do ataque, foram necessárias alterações no código do protocolo *AODV*.

Diff - Arquivo **aodv.h**

```
*** aodv-orig.h
--- aodv.h
***************
*** 274,279 ****
--- 274,280 ----

    double  PerHopTime(aodv_rt_entry *rt);

+   bool    malicious;

    nsaddr_t    index;    // IP Address of this node
    u_int32_t   seqno;    // Sequence Number
```

Diff - Arquivo **aodv.cc**

```
*** aodv-orig.cc
--- aodv.cc
***************
*** 84,89 ****
--- 84,94 ----
        return TCL_OK;
      }

+     if(strcmp(argv[1], "hacker") == 0) {
+       malicious = true;
+       return TCL_OK;
+     }
+
      if(strncasecmp(argv[1], "start", 2) == 0) {
        btimer.handle((Event*) 0);

***************
*** 145,150 ****
--- 150,156 ----
    index = id;
    seqno = 2;
    bid = 1;
+   malicious = false;
```

```
    LIST_INIT(&nbhead);
    LIST_INIT(&bihead);
***************
*** 160,165 ****
--- 166,175 ----
  void
  BroadcastTimer::handle(Event*) {
    agent->id_purge();
+   if (agent->malicious == true) {
+     agent->sendRequest(0);
+     printf("send request... \n");
+   }
    Scheduler::instance().schedule(this, &intr, BCAST_ID_SAVE);
  }
```

A.2 Scripts de Simulação

Arquivo de simulação - **cenario-exemplo.tcl**

```
# ======================================================================
# Configurações
# ======================================================================
$defaultRNG seed 0

set val(chan)   Channel/WirelessChannel    ;# channel type
set val(prop)   Propagation/TwoRayGround   ;# radio-propag. model
set val(netif)  Phy/WirelessPhy/802_15_4   ;# network interface
set val(mac)    Mac/802_15_4               ;# network protocol
set val(ifq)    Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)     LL                         ;# link layer type
set val(ant)    Antenna/OmniAntenna        ;# antenna model
set val(ifqlen) 50                         ;# max packet in ifq
set val(nn)     52                         ;# number of nodes
set val(rp)     AODV                       ;# routing protocol
set val(x)      120
set val(y)      120

set val(nam)     cenario-exemplo.nam
set val(traffic) ftp

set appTime     10                         ;# in seconds
set stopTime    300                        ;# in seconds

# ======================================================================
# Inicialização do Simulador
# ======================================================================
set ns_        [new Simulator]
set tracefd    [open ./cenario-exemplo.tr w]
$ns_ trace-all $tracefd
set namtrace   [open ./cenario-exemplo.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
```

```
33 $ns_ puts-nam-traceall {# nam4wpan #}     ;# special handling
34
35 Mac/802_15_4 wpanCmd verbose on
36 Mac/802_15_4 wpanNam namStatus on
37 #Mac/802_15_4 wpanNam ColFlashClr gold
38
39 # For model 'TwoRayGround'
40 set dist(5m)  7.69113e-06
41 set dist(9m)  2.37381e-06
42 set dist(10m) 1.92278e-06
43 set dist(11m) 1.58908e-06
44 set dist(12m) 1.33527e-06
45 set dist(13m) 1.13774e-06
46 set dist(14m) 9.81011e-07
47 set dist(15m) 8.54570e-07
48 set dist(16m) 7.51087e-07
49 set dist(20m) 4.80696e-07
50 set dist(25m) 3.07645e-07
51 set dist(30m) 2.13643e-07
52 set dist(35m) 1.56962e-07
53 set dist(40m) 1.20174e-07
54 Phy/WirelessPhy set CSThresh_ $dist(15m)
55 Phy/WirelessPhy set RXThresh_ $dist(15m)
56
57 set topo       [new Topography]
58 $topo load_flatgrid $val(x) $val(y)
59
60 set god_       [create-god $val(nn)]
61 set chan_1_    [new $val(chan)]
62
63 # configure node
64 $ns_ node-config -adhocRouting $val(rp) \
65         -llType $val(ll) \
66         -macType $val(mac) \
67         -ifqType $val(ifq) \
68         -ifqLen $val(ifqlen) \
69         -antType $val(ant) \
70         -propType $val(prop) \
71         -phyType $val(netif) \
72         -topoInstance $topo \
73         -agentTrace ON \
74         -routerTrace ON \
75         -macTrace OFF \
76         -movementTrace OFF \
77         -channel $chan_1_ \
78         -energyModel "EnergyModel" \
79         -initialEnergy 100 \
80         -rxPower 30e-3 \
81         -txPower 81e-3 \
82         -idlePower 3e-6 \
83         -sleepPower 45e-6
84
85 for {set i 0} {$i < $val(nn) } {incr i} {
```

```
86      set node_($i) [$ns_ node]
87      $node_($i) random-motion 0
88  }
89
90  # ==================================================================
91  # Inicialização dos nós
92  # ==================================================================
93
94  source cenario-exemplo.scn
95
96  $ns_ at 0.0 "$node_(0) NodeLabel PAN Coor"
97  $ns_ at 0.0 "$node_(0) sscs startPANCoord 1"
98
99  $ns_ at 0.0 "$node_(13) NodeLabel To 0"
100 $ns_ at 0.0 "$node_(40) NodeLabel To 0"
101 $ns_ at 0.0 "$node_(28) NodeLabel To 0"
102 $ns_ at 0.0 "$node_(27) NodeLabel To 14"
103 $ns_ at 0.0 "$node_(51) NodeLabel To 14"
104 $ns_ at 0.0 "$node_(39) NodeLabel To 14"
105
106 set t 1.0
107 for {set i 1} {$i < $val(nn) } {incr i} {
108     $ns_ at 0.5 "$node_($i) sscs startDevice 1 1 1";
109     set t [expr $t + 0.05]
110 }
111
112 for {set i 0} {$i < $val(nn)} {incr i} {
113     $ns_ initial_node_pos $node_($i) 2
114 }
115
116 # ==================================================================
117 # Inicialização do tráfego TCP
118 # ==================================================================
119 proc ftptraffic { src dst starttime stoptime } {
120    global ns_ node_
121    set tcp($src) [new Agent/TCP]
122    eval \$tcp($src) set packetSize_ 10
123    set sink($dst) [new Agent/TCPSink]
124    eval $ns_ attach-agent \$node_($src) \$tcp($src)
125    eval $ns_ attach-agent \$node_($dst) \$sink($dst)
126    eval $ns_ connect \$tcp($src) \$sink($dst)
127    set ftp($src) [new Application/FTP]
128    eval \$ftp($src) attach-agent \$tcp($src)
129    $ns_ at $starttime "$ftp($src) start"
130    $ns_ at $stoptime "$ftp($src) stop"
131 }
132
133 proc cbrtraffic { src dst interval starttime stoptime } {
134    global ns_ node_
135    set udp($src) [new Agent/UDP]
136    eval $ns_ attach-agent \$node_($src) \$udp($src)
137    set null($dst) [new Agent/Null]
138    eval $ns_ attach-agent \$node_($dst) \$null($dst)
```

```
139     eval $ns_ connect \$udp($src) \$null($dst)
140     set cbr($src) [new Application/Traffic/CBR]
141     eval \$cbr($src) set packetSize_ 10
142     eval \$cbr($src) set interval_ $interval
143     eval \$cbr($src) set random_ 0
144     eval \$cbr($src) attach-agent \$udp($src)
145     $ns_ at $starttime "$cbr($src) start"
146     $ns_ at $stoptime "$cbr($src) stop"
147 }
148
149 puts [format "Acknowledgement for data: %s" [Mac/802_15_4 wpanCmd
        ack4data]]
150 $ns_ at $appTime "Mac/802_15_4 wpanNam PlaybackRate 0.17ms"
151 $ns_ at [expr $appTime + 0.5] "Mac/802_15_4 wpanNam PlaybackRate
       1.5ms"
152
153 puts "\nTraffic: ftp"
154 ftptraffic 13 0 $appTime [expr $stopTime - 0.5]
155 ftptraffic 40 0 $appTime [expr $stopTime - 0.5]
156 ftptraffic 28 0 $appTime [expr $stopTime - 0.5]
157 ftptraffic 27 14 $appTime [expr $stopTime - 0.5]
158 ftptraffic 51 14 $appTime [expr $stopTime - 0.5]
159 ftptraffic 39 14 $appTime [expr $stopTime - 0.5]
160
161 # puts "\nTraffic: udp"
162 # cbrtraffic 13 0 1.5 $appTime [expr $stopTime - 0.5]
163 # cbrtraffic 40 0 1.5 [expr $appTime + 1] [expr $stopTime - 0.5]
164 # cbrtraffic 28 0 1.5 $appTime [expr $stopTime - 0.5]
165 # cbrtraffic 27 14 1.5 [expr $appTime + 1] [expr $stopTime - 0.5]
166 # cbrtraffic 51 14 1.5 $appTime [expr $stopTime - 0.5]
167 # cbrtraffic 39 14 1.5 [expr $appTime + 1] [expr $stopTime - 0.5]
168
169 # start random generation
170 $ns_ at 50 "[$node_(11) set ragent_] attacker rreqflood"
171 $ns_ at 50 "[$node_(23) set ragent_] attacker rreqflood"
172 $ns_ at 50 "[$node_(32) set ragent_] attacker rreqflood"
173 # end random generation
174
175
176 $ns_ at $appTime "puts \"\nTransmitting data ...\n\""
177
178 for {set i 0} {$i < $val(nn) } {incr i} {
179     $ns_ at $stopTime "$node_($i) reset";
180 }
181
182 $ns_ at $stopTime "stop"
183 $ns_ at $stopTime "$ns_ halt"
184
185 proc stop {} {
186     global ns_ tracefd appTime val env
187     $ns_ flush-trace
188     close $tracefd
189 }
```

```
190
191 puts "\nStarting Simulation..."
192 $ns_ run
```

Arquivo de posicionamento dos nós - **cenario-exemplo.scn**

```
1  $node_(0) set X_ 47.5
2  $node_(0) set Y_ 14.0
3  $node_(1) set X_ 47.1
4  $node_(1) set Y_ 22.1
5  $node_(2) set X_ 47.9
6  $node_(2) set Y_ 31.3
7  $node_(3) set X_ 47.3
8  $node_(3) set Y_ 39.1
9  $node_(4) set X_ 47.5
10 $node_(4) set Y_ 48.0
11 $node_(5) set X_ 47.4
12 $node_(5) set Y_ 56.3
13 $node_(6) set X_ 47.8
14 $node_(6) set Y_ 64.1
15 $node_(7) set X_ 48.0
16 $node_(7) set Y_ 72.6
17 $node_(8) set X_ 47.8
18 $node_(8) set Y_ 82.3
19 $node_(9) set X_ 47.5
20 $node_(9) set Y_ 90.5
21 $node_(10) set X_ 47.3
22 $node_(10) set Y_ 99.0
23 $node_(11) set X_ 47.1
24 $node_(11) set Y_ 106.6
25 $node_(12) set X_ 47.3
26 $node_(12) set Y_ 115.9
27 $node_(13) set X_ 47.0
28 $node_(13) set Y_ 124.6
29 $node_(14) set X_ 92.5
30 $node_(14) set Y_ 14.0
31 $node_(15) set X_ 92.0
32 $node_(15) set Y_ 21.8
33 $node_(16) set X_ 93.5
34 $node_(16) set Y_ 30.1
35 $node_(17) set X_ 92.3
36 $node_(17) set Y_ 39.8
37 $node_(18) set X_ 92.5
38 $node_(18) set Y_ 48.0
39 $node_(19) set X_ 93.1
40 $node_(19) set Y_ 55.9
41 $node_(20) set X_ 91.6
42 $node_(20) set Y_ 64.6
43 $node_(21) set X_ 92.4
44 $node_(21) set Y_ 72.7
45 $node_(22) set X_ 91.9
46 $node_(22) set Y_ 82.3
47 $node_(23) set X_ 92.5
```

```
48 $node_(23) set X_ 90.5
49 $node_(24) set X_ 93.4
50 $node_(24) set Y_ 99.9
51 $node_(25) set X_ 92.3
52 $node_(25) set Y_ 107.5
53 $node_(26) set X_ 92.4
54 $node_(26) set Y_ 116.6
55 $node_(27) set X_ 92.4
56 $node_(27) set Y_ 125.0
57 $node_(28) set X_ 14.9
58 $node_(28) set Y_ 47.5
59 $node_(29) set X_ 23.9
60 $node_(29) set Y_ 47.5
61 $node_(30) set X_ 31.5
62 $node_(30) set Y_ 47.5
63 $node_(31) set X_ 40.8
64 $node_(31) set Y_ 47.5
65 $node_(32) set X_ 56.8
66 $node_(32) set Y_ 47.5
67 $node_(33) set X_ 66.2
68 $node_(33) set Y_ 47.5
69 $node_(34) set X_ 74.2
70 $node_(34) set Y_ 47.5
71 $node_(35) set X_ 84.0
72 $node_(35) set Y_ 47.5
73 $node_(36) set X_ 101.0
74 $node_(36) set Y_ 47.5
75 $node_(37) set X_ 107.7
76 $node_(37) set Y_ 47.5
77 $node_(38) set X_ 116.4
78 $node_(38) set Y_ 47.5
79 $node_(39) set X_ 125.3
80 $node_(39) set Y_ 47.5
81 $node_(40) set X_ 15.5
82 $node_(40) set Y_ 91.5
83 $node_(41) set X_ 23.2
84 $node_(41) set Y_ 91.5
85 $node_(42) set X_ 32.3
86 $node_(42) set Y_ 91.5
87 $node_(43) set X_ 41.2
88 $node_(43) set Y_ 91.5
89 $node_(44) set X_ 58.5
90 $node_(44) set Y_ 91.5
91 $node_(45) set X_ 65.6
92 $node_(45) set Y_ 91.5
93 $node_(46) set X_ 75.3
94 $node_(46) set Y_ 91.5
95 $node_(47) set X_ 82.3
96 $node_(47) set Y_ 91.5
97 $node_(48) set X_ 100.6
98 $node_(48) set Y_ 91.5
99 $node_(49) set X_ 107.7
100 $node_(49) set Y_ 91.5
```

```
101 $node_(50) set X_ 117.6
102 $node_(50) set Y_ 91.5
103 $node_(51) set X_ 126.3
104 $node_(51) set Y_ 91.5
```

A.3 Script de Análise dos Resultados

Arquivo de análise do resultado das simulações - **analize.pl**

```
1  #!/usr/bin/perl
2  $tracefile=$ARGV[0];
3
4  $ofile="result.txt";
5  $ofile_csv="result.csv";
6  open OUT, ">>$ofile" or die "$0 cannot open output file $ofile: $
       !";
7  open OUT_CSV, ">>$ofile_csv" or die "$0 cannot open output file
       $ofile: $!";
8
9  open (DR,STDIN);
10 $gclock=0;
11
12 #Data Packet Information
13 $dataSent = 0;
14 $dataRecv = 0;
15 $routerDrop = 0;
16
17 #AODV Packet Information
18 $aodvSent = 0;
19 $aodvRecv = 0;
20 $aodvDrop = 0;
21
22 $aodvSentRequest = 0;
23 $aodvRecvRequest = 0;
24 $aodvDropRequest = 0;
25
26 $aodvSentReply = 0;
27 $aodvRecvReply = 0;
28 $aodvDropReply = 0;
29
30 while(<>){
31     chomp;
32     if (/^s/){
33         if (/^s.*AODV/) {
34             $aodvSent++;
35             if (/^s.*REQUEST/) {
36                 $aodvSendRequest++;
37             }
38             elsif (/^s.*REPLY/) {
39                 $aodvSendReply++;
40             }
```

```
41          }
42          elsif (/^s.*AGT/) {
43              $dataSent++;
44          }
45
46      } elsif (/^r/){
47          if (/^r.*AODV/) {
48              $aodvRecv++;
49              if (/^r.*REQUEST/) {
50                  $aodvRecvRequest++;
51              }
52              elsif (/^r.*REPLY/) {
53                  $aodvRecvReply++;
54              }
55
56          }
57          elsif (/^r.*AGT/) {
58              $dataRecv++;
59          }
60
61      } elsif (/^D/) {
62          if (/^D.*AODV/) {
63              $aodvDrop++;
64              if (/^D.*REQUEST/) {
65                  $aodvDropRequest++;
66              }
67              elsif (/^D.*REPLY/) {
68                  $aodvDropReply++;
69              }
70
71          }
72          if (/^D.*RTR/) {
73              $routerDrop++;
74          }
75      }
76
77  }
78
79  close DR;
80
81  $delivery_ratio = 100*$dataRecv/$dataSent;
82
83  print OUT "AODV Sent                          : $aodvSent\n";
84  print OUT "   |_ Request      : $aodvSendRequest\n";
85  print OUT "   |_ Reply        : $aodvSendReply\n";
86  print OUT "AODV Recv                          : $aodvRecv\n";
87  print OUT "   |_ Request      : $aodvRecvRequest\n";
88  print OUT "   |_ Reply        : $aodvRecvReply\n";
89  print OUT "AODV Drop                          : $aodvDrop\n";
90  print OUT "   |_ Request      : $aodvDropRequest\n";
91  print OUT "   |_ Reply        : $aodvDropReply\n";
92  print OUT "Data Sent                          : $dataSent\n";
93  print OUT "Data Recv                          : $dataRecv\n";
```

```
94  print OUT "Router Drop                        : $routerDrop\n";
95  print OUT "Delivery Ratio (%)                 : $delivery_ratio\n";
96
97  print OUT_CSV "$aodvSent,";
98  print OUT_CSV "$aodvSendRequest,";
99  print OUT_CSV "$aodvSendReply,";
100 print OUT_CSV "$aodvRecv,";
101 print OUT_CSV "$aodvRecvRequest,";
102 print OUT_CSV "$aodvRecvReply,";
103 print OUT_CSV "$aodvDrop,";
104 print OUT_CSV "$aodvDropRequest,";
105 print OUT_CSV "$aodvDropReply,";
106 print OUT_CSV "$dataSent,";
107 print OUT_CSV "$dataRecv,";
108 print OUT_CSV "$routerDrop,";
109 print OUT_CSV "$delivery_ratio,";
110
111 close OUT;
112 close OUT_CSV;
```

Arquivo de análise do resultado das simulações - **analize.php**

```
1  <?php
2  set_time_limit(15);
3
4  $handle = fopen($argv[1], 'r');
5  if($handle) {
6
7      $start_time = array();
8      $end_time = array();
9      $delay = array();
10     $energy = array();
11
12     $max_time = 0;
13
14     while(!feof($handle)) {
15         $buffer = explode(' ', fgets($handle, 4096));
16         if($buffer[0] == 's' && $buffer[3] == 'AGT') $start_time[
               $buffer[6]] = $buffer[1];
17
18         if($buffer[0] == 'r' && $buffer[7] == 'tcp') $end_time[
               $buffer[6]] = $buffer[1];
19         if($buffer[0] == 'D' && $buffer[7] == 'tcp') $end_time[
               $buffer[6]] = -1;
20
21         /**
22          * UDP
23          * if($buffer[0] == 'r' && $buffer[7] == 'cbr') $end_time
                [$buffer[6]] = $buffer[1];
24          * if($buffer[0] == 'D' && $buffer[7] == 'cbr') $end_time
                [$buffer[6]] = -1;
25         */
26
```

```
27          if($buffer[0] == 'N') $energy[$buffer[4]] = $buffer[6];
28
29          if(in_array($buffer[0], array('s','r','D')))
30              $max_time = max($max_time, $buffer[1]);
31          elseif($buffer[0] == 'N')
32              $max_time = max($max_time, $buffer[2]);
33      }
34      fclose($handle);
35
36      foreach($start_time as $key => $start)
37          if(isset($end_time[$key]) && $end_time[$key] > 0)
38              $delay[$key] = $end_time[$key] - $start;
39
40      $avg = array_sum($delay)/count($delay);
41      file_put_contents("result.txt", 'Average End-to-End Delay (ms
           )  : '.($avg * 1000)."\n", FILE_APPEND);
42      file_put_contents("result.csv", ($avg * 1000).",",
           FILE_APPEND);
43
44      $nodes_energy = '';
45      $nodes_energy_csv = '';
46      for($i = 0; $i < 52; $i++) {
47          $energy[$i] = 100 - $energy[$i];
48          $nodes_energy .= '  |_ Node '.str_pad($i, 2, " ",
               STR_PAD_LEFT).'     : '.number_format($energy[$i], 6)."
               \n";
49          $nodes_energy_csv .= number_format($energy[$i], 6).",";
50      }
51
52      $avg = array_sum($energy)/count($energy);
53      file_put_contents("result.txt", 'Average Energy Consumption (
           J) : '.number_format($avg, 6)."\n", FILE_APPEND);
54      file_put_contents("result.csv", number_format($avg, 6).",",
           FILE_APPEND);
55      file_put_contents("result.txt", $nodes_energy."\n\n",
           FILE_APPEND);
56      file_put_contents("result.csv", $nodes_energy_csv.'|'.
           $max_time."\n", FILE_APPEND);
57  }
```