Fernando Parente Garcia

EPMOSt: Um Sistema de Monitoramento Passivo Energeticamente Eficiente para Redes de Sensores Sem Fio

Fortaleza – Ceará - Brasil
2014

Fernando Parente Garcia

EPMOSt: Um Sistema de Monitoramento Passivo Energeticamente Eficiente para Redes de Sensores Sem Fio

Tese submetida à coordenação do programa de Mestrado e Doutorado em Ciência da Computação da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Doutor em Ciência da Computação.

Orientadora: Prof[a]. Rossana M. C. Andrade, Dra.

Coorientador: Prof. José Neuman de Souza, Dr.

UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO

Fortaleza - Ceará - Brasil
2014

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca de Ciências e Tecnologia

G199e Garcia, Fernando Parente.
EPMOSt: um sistema de monitoramento passivo energeticamente eficiente para redes de sensores sem fio / Fernando Parente Garcia. – 2014.
115 f. : il. color.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Ciências, Departamento de Computação, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2014.
Área de Concentração: Ciência da Computação.
Orientação: Profa. Dra. Rossana Maria de Castro Andrade.
Coorientação: Prof. Dr. José Neuman de Souza.

1. Rede de Sensores Sem Fio. 2. Eficiência energética. I. Título.

CDD 005

Fernando Parente Garcia

EPMOSt: Um Sistema de Monitoramento Passivo Energeticamente Eficiente para Redes de Sensores Sem Fio.

Tese submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação, da Universidade Federal do Ceará, como requisito para a obtenção do grau de Doutor em Ciência da Computação.

Aprovada em 26 de novembro de 2014

BANCA EXAMINADORA

Profa. Dra. Rossana Maria de Castro Andrade
Orientadora
Universidade Federal do Ceará – UFC

Prof. Dr. José Neuman de Souza
Coorientador
Universidade Federal do Ceará – UFC

Prof. Dr. Miguel Franklin de Castro
Universidade Federal do Ceará – UFC

Prof. Dr. Daniel Gonçalves Gomes
Universidade Federal do Ceará – UFC

Prof. Dr. Ângelo Roncalli Alencar Brayner
Universidade de Fortaleza - UNIFOR

Prof. Dr. Antonio Alfredo Ferreira Loureiro
Universidade Federal de Minas Gerais - UFMG

Fortaleza, 26 denovembro de 2014

Aos meus pais Huygens e Maria, à minha amada esposa Cléce e ao meu querido filho André.

AGRADECIMENTOS

À Deus, por estar sempre ao meu lado, dando-me coragem para enfrentar todos os obstáculos da vida.

Aos meus pais, Huygens e Maria, que com toda dedicação possível ensinaram-me a enfrentar desafios e superar os obstáculos com humildade, honestidade e perseverança.

À minha amada esposa, Cléce, pela compreensão, dedicação e amor no decorrer desses anos.

Ao meu querido filho, André, que com seu amor e carinho tem tornado a minha vida muito mais alegre.

À minha família, por todo o apoio que me foi concedido ao longo de toda a vida.

À professora Rossana, pela dedicada orientação durante todo o curso de doutorado, pela confiança e paciência nos momentos mais delicados, e por compartilhar seu conhecimento e experiência, fundamentais ao bom desenvolvimento desta tese.

Ao professor Neuman, pelo apoio e incentivo, e pelas importantes contribuições durante o desenvolvimento deste trabalho.

Ao professor Antonio Loureiro, pela acolhida durante a semana que passei na UFMG, por sua participação na banca avaliadora da proposta de tese e pelas sugestões que contribuíram significativamente para o enriquecimento deste trabalho.

Aos professores Ângelo Brayner e Miguel Franklin, por suas participações nas bancas avaliadoras de qualificação e de proposta de tese. Suas críticas e sugestões ajudaram bastante a melhorar este trabalho.

Ao professor Danielo Gomes, por aceitar o convite para participar da banca avaliadora desta tese.

Aos alunos de iniciação científica, Gabriel e Gonçalves, que trabalharam nas simulações realizadas neste trabalho.

Aos amigos e funcionários do GREat e do MDCC. Em especial, Darilú, Janaina e Orley.

E por fim, a todas as pessoas que contribuíram direta ou indiretamente na realização deste trabalho, meus sinceros agradecimentos.

“Cada sonho que você deixa para trás, é um pedaço do seu futuro que deixa de existir."

Steve Jobs

RESUMO

Sistemas de monitoramento permitem depurar e analisar o funcionamento de uma Rede de Sensores Sem Fio (RSSF). No monitoramento passivo, uma rede de monitoramento adicional é implantada com o intuito de capturar e analisar os pacotes transmitidos pela rede a ser monitorada, denominada rede alvo. Quando se deseja monitorar uma RSSF em um ambiente real, um sistema de monitoramento passivo energeticamente eficiente é necessário, pois, caso contrário, a rede de monitoramento pode ter um tempo de vida bem menor do que a rede alvo. Nesta tese, inicialmente, os principais sistemas de monitoramento passivo propostos para RSSF foram identificados, analisados e comparados. Durante as pesquisas realizadas na literatura, não foi identificado nenhum sistema de monitoramento passivo que se preocupasse em reduzir o consumo de energia da rede de monitoramento. Sendo assim, esta tese propõe um sistema de monitoramento passivo energeticamente eficiente para RSSF, denominado EPMOSt (**E**nergy-efficient **P**assive **MO**nitoring **S**ys**t**em), que prolonga o tempo de vida da rede de monitoramento. O EPMOSt utiliza dois mecanismos para reduzir o consumo de energia da rede de monitoramento: eleição de *sniffers* (nós da rede de monitoramento) e agregação de cabeçalhos. A eleição de *sniffers* garante que durante a maior parte do tempo apenas um *sniffer* captura os pacotes transmitidos por um determinado nó da rede alvo, reduzindo assim a transmissão de pacotes capturados redundantes através da rede de monitoramento e, consequentemente, reduzindo consideravelmente o consumo de energia desta rede. Com a agregação de cabeçalhos, apenas as informações presentes nos cabeçalhos dos pacotes capturados são enviadas através da rede de monitoramento. Assim, os cabeçalhos de vários pacotes podem ser enviados na mesma mensagem de monitoramento, reduzindo assim o *overhead* de transmissão e, consequentemente, reduzindo também o consumo de energia da rede de monitoramento. Experimentos com sensores reais e com um simulador de RSSF são realizados em vários cenários para avaliar o sistema de monitoramento proposto. Os resultados obtidos mostram a eficiência energética do EPMOSt e a viabilidade de utilizá-lo para monitorar RSSF em ambientes reais.

Palavras-chave: Monitoramento Passivo; Eficiência Energética; Redes de Sensores Sem Fio.

ABSTRACT

Monitoring systems are important for debugging and analyzing Wireless Sensor Networks (WSN). In passive monitoring, a monitoring network needs to be deployed in addition to the network to be monitored, called target network. The monitoring network captures and analyzes packets sent by the target network. An energy-efficient passive monitoring system is necessary when there is a need to monitor a WSN in a real scenario because the lifetime of the monitoring network is extended and, consequently, the target network benefits from the monitoring for a longer time. In this thesis, initially, the main passive monitoring systems proposed for WSN have been identified, analyzed and compared. During the literature review, no passive monitoring system for WSN that aims to reduce the energy consumption of the monitoring network has been identified. Therefore, this thesis proposes an **E**nergy-efficient **P**assive **MO**nitoring **S**yst**e**m for WSN (EPMOSt) that extends the lifetime of the monitoring network. EPMOSt uses two mechanisms to reduce the energy consumption of the monitoring network: sniffer election and aggregation of headers. By using the sniffer election, in general only one sniffer (a node of the monitoring network) captures packets sent by a given node of the target network, thereby reducing the transmission of packets captured by the monitoring network and, thus, considerably reducing the energy consumption of this network. By using aggregation of headers, only the information present in the headers of captured packets is sent through the monitoring network. Thus, the headers of several packets may be sent in the same monitoring message, hence reducing the overhead of transmission and consequently reducing the energy consumption of the monitoring network. Experiments performed with real sensors and with a WSN simulator in various scenarios are conducted to evaluate the proposed monitoring system. The obtained results show the energy efficiency of the EPMOSt and the viability of using it to monitor WSN in real scenarios.

Keywords: Passive Monitoring; Energy-Efficient; Wireless Sensor Networks.

SUMÁRIO

1 INTRODUÇÃO 18
1.1 Contextualização e motivação 18
1.2 Hipótese e questões de pesquisa 20
1.3 Objetivo e principais contribuições 21
1.4 Metodologia 22
1.5 Organização do documento 24
2 RSSF E GERENCIAMENTO DE REDES 25
2.1 Redes de sensores sem fio 25
2.1.1 Componentes 26
2.1.2 Características 27
2.1.3 Classificação 29
2.1.4 Aplicações 31
2.2 Gerenciamento de redes 33
2.2.1 Áreas funcionais 33
2.2.2 Infraestrutura de gerenciamento de redes 36
2.2.3 Protocolo de gerenciamento SNMP 38
2.2.3.1 Base de informações de gerenciamento (MIB) 39
2.2.3.2 Mensagens SNMP 41
2.3 Gerenciamento de RSSF 43
2.3.1 Desafios 43
2.3.2 Requisitos 44
2.3.3 Sistemas de monitoramento de RSSF 45
2.4 Considerações finais 47
3 SISTEMAS DE MONITORAMENTO PASSIVO PARA RSSF 48
3.1 Etapas da revisão bibliográfica 48
3.2 SNTS (Sensor Network Troubleshooting Suite) 52
3.3 SNIF (Sensor Network Inspection Framework) 53
3.4 Pimoto 55
3.5 LiveNet 57
3.6 PMSW (Passive Monitoring System in Wireless Sensor Networks) 58
3.7 Análise comparativa 60

3.8 Considerações finais 62
4 O SISTEMA DE MONITORAMENTO EPMOSt 63
4.1 Descrição do sistema 63
4.2 Especificação do sistema 66
4.2.1 Sniffer 66
4.2.1.1 Mecanismo de eleição 66
4.2.1.2 Processo de monitoramento 69
4.2.2 Monitor Local 71
4.2.3 Servidor - Análise do trace 72
4.2.4 Servidor - Agente SNMP 75
4.2.4.1 MIB do EPMOSt 75
4.2.4.2 Teste do agente SNMP 78
4.2.5 Servidor - On/Off 83
4.3 Considerações finais 84
5 EXPERIMENTOS 86
5.1 Experimentos em um *testbed* 86
5.1.1 Descrição dos experimentos 86
5.1.2 Métricas 88
5.1.3 Resultados e discussão 91
5.2 Experimentos com simulador 94
5.3 Considerações finais 102
6 CONCLUSÕES E TRABALHOS FUTUROS 103
6.1 Análise da hipótese e das questões de pesquisa 103
6.2 Resultados alcançados e principais contribuições 105
6.3 Trabalhos futuros 109
REFERÊNCIAS BIBLIOGRÁFICAS 112

LISTA DE FIGURAS

Figura 1.1 – Monitoramento passivo 19

Figura 1.2 – Metodologia para o desenvolvimento da tese 23

Figura 2.1 – Componentes de uma RSSF 26

Figura 2.2 – Hardware de um nó sensor 27

Figura 2.3 – Nós sensores 27

Figura 2.4 – Infraestrutura de gerenciamento 36

Figura 2.5 – Parte da árvore de nomeação de objetos ASN1 40

Figura 2.6 – Subárvore de nomeação de objetos do módulo System 41

Figura 2.7 – Sistemas de gerenciamento vs. sistemas de monitoramento 46

Figura 3.1 – Rede de monitoramento no SNTS 52

Figura 3.2 – Sistema de monitoramento SNIF 54

Figura 3.3 – Interface de usuário do SNIF 54

Figura 3.4 – Sistema de monitoramento Pimoto 55

Figura 3.5 – Tela do Wireshark exibindo um pacote da rede alvo 56

Figura 3.6 – Funcionamento do Sistema de monitoramento PMSW 58

Figura 4.1 – Visão geral do sistema de monitoramento EPMOSt 64

Figura 4.2 – Diagrama de atividades do EPMOSt 64

Figura 4.3 – Nó sensor MicaZ 66

Figura 4.4 – Eleição quando ainda nenhum *sniffer* está capturando os pacotes do nó alvo 67

Figura 4.5 – Eleição quando SY está capturando os pacotes e RSSI(SY) ≥ RSSI(SX) 67

Figura 4.6 – Eleição quando SY está capturando os pacotes e RSSI(SY) < RSSI(SX) 68

Figura 4.7 – Diagrama de atividades do processo de monitoramento 69

Figura 4.8 – Formato do pacote enviado pelos *sniffers* sem agregação de cabeçalhos 70

Figura 4.9 – Formato do pacote enviado pelos *sniffers* com agregação de cabeçalhos 70

Figura 4.10 – Exibição de pacotes da rede alvo recebidos pelo monitor local 71

Figura 4.11 – Exibição de pacotes capturados inseridos no banco de dados 72

Figura 4.12 – Tela inicial da aplicação análise *trace* 72

Figura 4.13 – *Trace* com pacotes redundantes 73

Figura 4.14 – Informações sobre os nós da rede alvo 74

Figura 4.15 – Informações sobre os *paths* da rede alvo 74

Figura 4.16 – MIB do EPMOSt 75

Figura 4.17 – Tabelas da MIB do EPMOSt 76

Figura 4.18 – Exibição da MIB do EPMOSt 79

Figura 4.19 – Exibição das informações dos nós da rede alvo 79

Figura 4.20 – Exibição das informações dos *paths* da rede alvo 80

Figura 4.21 – Exibição de informações estatísticas sobre a rede de monitoramento 80

Figura 4.22 – Exibição de informações estatísticas sobre os *sniffers* 81

Figura 4.23 – Tela inicial da ferramenta *ManageEngine MIB Browser Free Tool* 82

Figura 4.24 – Exibição da tabela *nodesTable* da MIB do EPMOSt 82

Figura 4.25 – Exibição da tabela *pathTable* da MIB do EPMOSt 83

Figura 4.26 – Mecanismo para ligar/desligar o monitoramento 84

Figura 5.1 – Cenário utilizado nos experimentos 87

Figura 5.2 – Distribuição dos *sniffers* 87

Figura 5.3 – Energia consumida pela rede de monitoramento vs. quantidade de *sniffers* 91

Figura 5.4 – Energia consumida por cada *sniffer* vs. quantidade de *sniffers* 93

Figura 5.5 – Pacotes capturados pela rede de monitoramento vs. quantidade de *sniffers* 93

Figura 5.6 – Interface do COOJA durante a simulação do cenário da Figura 5.2b 95

Figura 5.7 – Energia consumida pela rede de monitoramento vs. quantidade de sniffers para os cenários com 21 nós sensores da rede alvo (Figura 5.2) 96

Figura 5.8 – Energia consumida por cada sniffer vs. quantidade de sniffers para os cenários com 21 nós sensores da rede alvo (Figura 5.2)..... 97

Figura 5.9 – Pacotes capturados pela rede de monitoramento vs. quantidade de sniffers para os cenários com 21 nós sensores da rede alvo (Figura 5.2)..... 98

Figura 5.10 – Interface do COOJA durante a simulação do cenário com 50 nós sensores da rede alvo e 15 sniffers 99

Figura 5.11 – Energia consumida pela rede de monitoramento vs. quantidade de sniffers para os cenários com 50 nós sensores da rede alvo..... 99

Figura 5.12 – Energia consumida por cada sniffer vs. quantidade de sniffers para os cenários com 50 nós sensores da rede alvo..... 100

Figura 5.13 – Pacotes capturados pela rede de monitoramento vs. quantidade de sniffers para os cenários com 50 nós sensores da rede alvo..... 101

LISTA DE TABELAS

Tabela 2.1 – Classificação das RSSF quanto ao sensoriamento 29

Tabela 2.2 – Classificação das RSSF quanto à configuração 30

Tabela 2.3 – Classificação das RSSF quanto ao tipo de processamento 30

Tabela 2.4 – Classificação das RSSF quanto à comunicação 31

Tabela 2.5 – Modelo FCAPS 34

Tabela 3.1 – Artigos identificados na revisão bibliográfica 50

Tabela 3.2 – Sistemas de monitoramento passivo para RSSF 51

Tabela 3.3 – Características dos sistemas de monitoramento 61

Tabela 4.1 – Objetos representados na *nodesTable* 77

Tabela 4.2 - Objetos representados na *pathTable* 77

Tabela 4.3 - Objetos representados na *snifferTable* 77

Tabela 4.4 - Objetos representados na *monitorNetworkTable* 78

Tabela 6.1 – Trabalhos relacionados vs. EPMOSt 106

Tabela 6.2 – Artigos científicos produzidos relacionados diretamente com a tese 108

Tabela 6.3 – Artigos científicos secundários produzidos 109

LISTA DE SIGLAS

ASN1	Abstract Syntax Notation 1
CMIP	Common Management Information Protocol
CTR	Contribuição
DSD	Diagnosis based on Sensing Data
DSN	Deployment Support Network
DSN	Destination Sequence Number
EPMOSt	Energy-efficient Passive MOnitoring System
FCAPS	Fault, Configuration, Accounting, Performance And Security
GPL	General Public License
GREat	Grupo de Redes de Computadores, Engenharia de Software e Sistemas
IETF	Internet Engineering Task Force
IoT	Internet of Things
IP	Internet Protocol
ISO	International Organization for Standardization
ITU	International Telecommunication Union
LoNI	Lossy Nodes Inference
LQI	Link Quality Indicator
MET	Meta
MIB	Management Information Base
nesC	network embedded system C
PC	Personal Computer
PMSW	Passive Monitoring System in Wireless sensor networks
PROP	Propriedade
QP	Questão de pesquisa

RFID	Radio-Frequency IDentification
RSSI	Received Signal Strength Indicator
RSSF	Redes de Sensores sem Fio
SGBD	Sistema de Gerenciamento de Banco de dados
SMI	Structure of Management Information
SNIF	Sensor Network Inspection Framework
SNMP	Simple Network Management Protocol
SNTS	Sensor Network Troubleshooting Suite
SQL	Structured Query Language
StArt	State of the Art through Systematic Review
TCP	Transmission Control Protocol
TMN	Telecommunications Management Network
UML	Unified Modeling Language
WSN	Wireless Sensor Network

1 INTRODUÇÃO

Esta tese propõe um sistema de monitoramento passivo energeticamente eficiente para redes de sensores sem fio, denominado **EPMOSt** (***E**nergy-efficient **P**assive **MO**nitoring **S**yst**e**m*). Na seção 1.1 deste capítulo, a motivação para escolha do tema de pesquisa desta tese é discutida. A hipótese e as questões de pesquisa que nortearam o desenvolvimento desta tese são descritas na seção 1.2. Na seção 1.3, os objetivos, as metas e as principais contribuições desta pesquisa são elencados. Em seguida, na seção 1.4, a metodologia utilizada para a elaboração da tese é detalhada. A seção 1.5 finaliza o capítulo apresentando a estrutura organizacional do restante deste documento.

1.1 Contextualização e motivação

A miniaturização dos componentes eletrônicos e a evolução das tecnologias de comunicação sem fio têm estimulado o desenvolvimento e uso de Redes de Sensores Sem Fio (RSSF) em várias aplicações, tais como monitoramento ambiental, detecção sísmica e monitoramento da saúde estrutural de edificações, entre outras. Em geral, as RSSF são compostas por nós sensores de tamanho reduzido alimentados por baterias e utilizam comunicação sem fio de pequeno alcance [Cheour et al. 2011]. Normalmente estas redes possuem um grande número de nós distribuídos, têm restrições de energia, e devem possuir mecanismos para auto-configuração e adaptação devido a problemas como falhas de comunicação e perda de nós [Loureiro et al. 2003].

O monitoramento de uma RSSF em operação é importante para depurar e analisar o seu funcionamento. Utilizando-se um sistema de monitoramento, várias informações sobre o funcionamento da RSSF podem ser obtidas, tais como descoberta de topologia, morte e reinicialização de nós, nós isolados, laços de roteamento, perda de pacotes e latência da rede, entre outras [Ringwald and Romer 2007].

Em RSSF, o monitoramento da rede pode ser dividido em monitoramento ativo e monitoramento passivo [Xu et al. 2011]. No monitoramento ativo são inseridas linhas de código na aplicação executada pelos nós sensores para obter informações sobre o funcionamento da rede. Neste caso, os pacotes de monitoramento são enviados juntamente com os pacotes de dados da rede alterando o comportamento e funcionamento da rede monitorada e consumindo os recursos desta rede. Por exemplo, no sistema de monitoramento

ativo Simpathy [Ramanathan et al. 2005] cerca de 30% da largura de banda da rede é utilizada pelo tráfego de monitoramento. Além disso, falhas na RSSF podem afetar o sistema de monitoramento. No entanto, o monitoramento ativo permite que o próprio nó sensor possa analisar as informações obtidas com o monitoramento e alterar seus parâmetros de funcionamento de acordo com estas informações. Sistemas de monitoramento ativo para RSSF são propostos em [Girod et al. 2007], [Tolle and Culler 2005], [Suhonen et al. 2008] e [Chen and Shin 2009].

No monitoramento passivo, uma rede de monitoramento adicional é implantada juntamente com a rede que deve ser monitorada (rede alvo), conforme mostrado na Figura 1.1. A rede de monitoramento captura e analisa os pacotes transmitidos pela rede alvo, não consumindo nenhum recurso da rede alvo. Além disso, uma falha na rede alvo não compromete o funcionamento do mecanismo de monitoramento [Hänninen at al. 2011]. Portanto, quando se deseja reduzir a utilização de recursos da rede alvo e/ou isolar as falhas da rede alvo das falhas da rede de monitoramento, é melhor utilizar um sistema de monitoramento passivo.

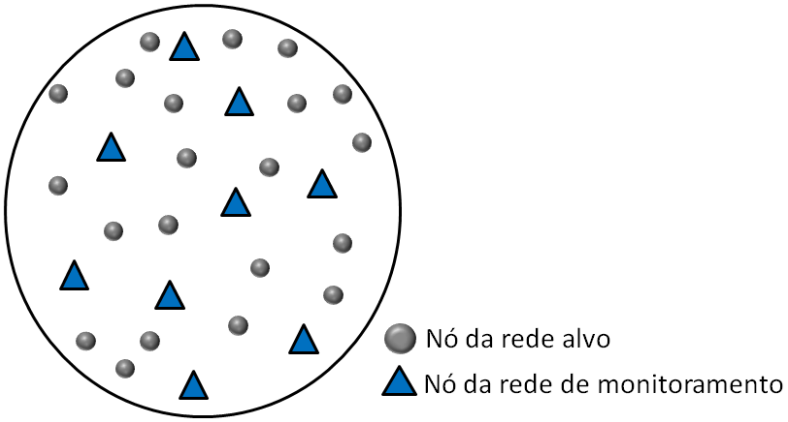


Figura 1.1 – Monitoramento passivo

No entanto, mesmo utilizando monitoramento passivo não é possível garantir que todos os pacotes enviados pela rede alvo sejam capturados. Devido ao alcance limitado do rádio dos nós da rede alvo, os nós da rede de monitoramento somente conseguem capturar os pacotes quando estão localizados na vizinhança do remetente. Além disso, as colisões de pacotes que ocorrem no canal de comunicação sem fio, os sinais de rádio de fraca intensidade e a limitação de recursos contribuem para que os nós da rede de monitoramento não capturem todos os pacotes enviados pelos nós da rede alvo [Xu et al. 2010]. Sendo assim, os sistemas de monitoramento passivo devem se preocupar em reduzir a quantidade de pacotes enviados pela rede alvo que não são capturados.

É importante ressaltar ainda que o tempo de vida de uma RSSF pode ser de até vários anos e nem todos os problemas aparecem durante as primeiras semanas após a implantação da rede [Hänninen et al. 2011]. Sendo assim, um sistema de monitoramento energeticamente eficiente é necessário quando se deseja monitorar uma RSSF em um cenário real durante um longo período de tempo, pois, caso contrário, a rede de monitoramento pode ter um tempo de vida bem menor do que a rede alvo. Como exemplo, em [Liu et al. 2010], os autores descrevem a utilização de uma rede alvo para monitoramento de oceanos e ressaltam a importância de se monitorar esta rede através de um sistema de monitoramento passivo energeticamente eficiente.

Pelas características das RSSF mencionadas anteriormente, é importante minimizar os recursos incluindo também aqueles utilizados pelos sistemas de monitoramento e, por isso, este trabalho foca em sistemas de monitoramento passivo, que não consomem nenhum recurso da rede alvo monitorada.

Portanto, a motivação para o desenvolvimento desta pesquisa de doutorado consiste em propor e desenvolver um sistema de monitoramento passivo energeticamente eficiente para redes de sensores sem fio para que a rede alvo seja beneficiada pelo monitoramento por mais tempo. Além da eficiência energética, o sistema proposto deve disponibilizar as informações obtidas com o monitoramento de modo que facilite a administração da rede alvo e também deve se preocupar em reduzir a quantidade de pacotes enviados pela rede alvo que não são capturados.

1.2 Hipótese e questões de pesquisa

Considerando que o monitoramento de uma RSSF em operação é importante para depurar e analisar o seu funcionamento e que, conforme citado na seção 1.1, em certas aplicações é importante monitorar uma RSSF em um cenário real ("*in situ*") durante um longo período de tempo, esta tese de doutorado busca investigar a seguinte hipótese:

> *É possível propor e desenvolver um sistema de monitoramento passivo energeticamente eficiente para RSSF mantendo a taxa percentual de pacotes capturados da rede alvo nos mesmos patamares das taxas obtidas pelos sistemas de monitoramento passivo existentes na literatura, de modo a tornar possível o monitoramento de RSSF in situ durante um longo período de tempo e a facilitar a administração da rede alvo.*

A partir desta hipótese, três questões de pesquisa (QP) foram levantadas para serem investigadas neste trabalho, são elas:

QP01: *Quais são os principais requisitos que um sistema de monitoramento passivo para RSSF deve possuir?*

QP02: *Quais mecanismos deverão ser utilizados para reduzir o consumo de energia da rede de monitoramento e, consequentemente, prolongar o tempo de vida desta rede sem reduzir a taxa de pacotes capturados da rede alvo?*

QP03: *Como o sistema proposto deverá disponibilizar as informações obtidas com o monitoramento de modo a facilitar a administração da rede alvo?*

1.3 Objetivo e principais contribuições

Levando-se em consideração o que foi apresentado nas seções anteriores, esta tese de doutorado tem como objetivo principal propor um sistema de monitoramento passivo energeticamente eficiente para RSSF. A redução do consumo de energia da rede de monitoramento deve, portanto, tornar possível o monitoramento de RSSF *in situ* durante um longo período de tempo. O sistema proposto deve também disponibilizar as informações obtidas com o monitoramento de modo que facilite a administração da rede alvo.

Para alcançar este objetivo, quatro metas (MET) devem ser atingidas:

MET01: *Realizar revisão bibliográfica sobre sistemas de monitoramento passivo para RSSF (**QP01**);*

MET02: *Propor um sistema de monitoramento passivo energeticamente eficiente para RSSF (**QP02, QP03**);*

MET03: *Implementar o sistema de monitoramento proposto (**QP02**, **QP03**);*

MET04: *Realizar experimentos em ambientes reais e em simulador para avaliar o sistema de monitoramento proposto (**QP02**, **QP03**).*

Com isso, as contribuições (CTR) esperadas desta tese de doutorado são listadas a seguir:

CTR01: *Um sistema de monitoramento (**MET03**, **MET04**).*

Os detalhes dos requisitos, projeto e implementação de todos os módulos do sistema proposto são descritos com o intuito de facilitar a replicação do sistema em outros

ambientes. Além disso, todos os módulos são validados e avaliados através de experimentos em ambientes reais e em simulador.

CTR02: *Um mecanismo de eleição de sniffers (**MET02**).*

O mecanismo elege quais nós da rede alvo terão seus pacotes capturados por quais *sniffers* (nós da rede de monitoramento). O mecanismo de eleição proposto deve garantir que durante a maior parte do tempo apenas um *sniffer* captura os pacotes enviados por um determinado nó da rede alvo, reduzindo assim a transmissão de pacotes capturados redundantes através da rede de monitoramento e, consequentemente, reduzindo o consumo de energia desta rede.

CTR03: *A especificação de uma MIB* (*Management Information Base*) *para sistemas de monitoramento passivo para RSSF (**MET02**).*

Através de uma revisão bibliográfica são identificados os objetos gerenciados que devem compor uma MIB específica para sistemas de monitoramento passivo para RSSF.

CTR04: *O desenvolvimento de um agente SNMP (**MET02**, **MET03**).*

O agente SNMP (*Simple Network Management Protocol*) deve permitir integrar o sistema de monitoramento proposto com ferramentas de gerência de rede que suportem tal protocolo, facilitando a administração da rede alvo.

CTR05: *Uma análise comparativa dos sistemas de monitoramento passivo para RSSF (**MET01**).*

Os principais requisitos que um sistema de monitoramento passivo para RSSF deve possuir são inicialmente identificados. Em seguida, é realizada uma análise comparativa entre os sistemas de monitoramento existentes na literatura levando-se em consideração tais requisitos.

1.4 Metodologia

A metodologia adotada para o desenvolvimento desta tese de doutorado é apresentada na Figura 1.2 e segue as etapas:

I. Realização de revisão bibliográfica com o intuito de identificar os principais sistemas de monitoramento passivo para RSSF existentes na literatura (**MET01**).

II. Análise dos sistemas de monitoramento identificados na etapa I com o intuito de identificar quais são os requisitos mais importantes que um sistema de monitoramento passivo para RSSF deve possuir (**MET01**).

III. Proposta de um sistema de monitoramento passivo energeticamente eficiente para RSSF, e que também contemple os principais requisitos identificados na etapa II. Para realizar esta etapa, são executadas as seguintes atividades:

a) Levantamento dos requisitos do sistema de monitoramento proposto (**MET02**).

b) Especificação do sistema de monitoramento proposto (**MET02**).

c) Implementação do sistema de monitoramento proposto (**MET03**, **MET04**).

d) Realização de experimentos com o sistema de monitoramento proposto com o intuito de comparar os resultados obtidos com os resultados dos sistemas de monitoramento identificados na etapa I. Os experimentos são realizados com sensores reais (*testbed*) e com um simulador de RSSF. (**MET04**).

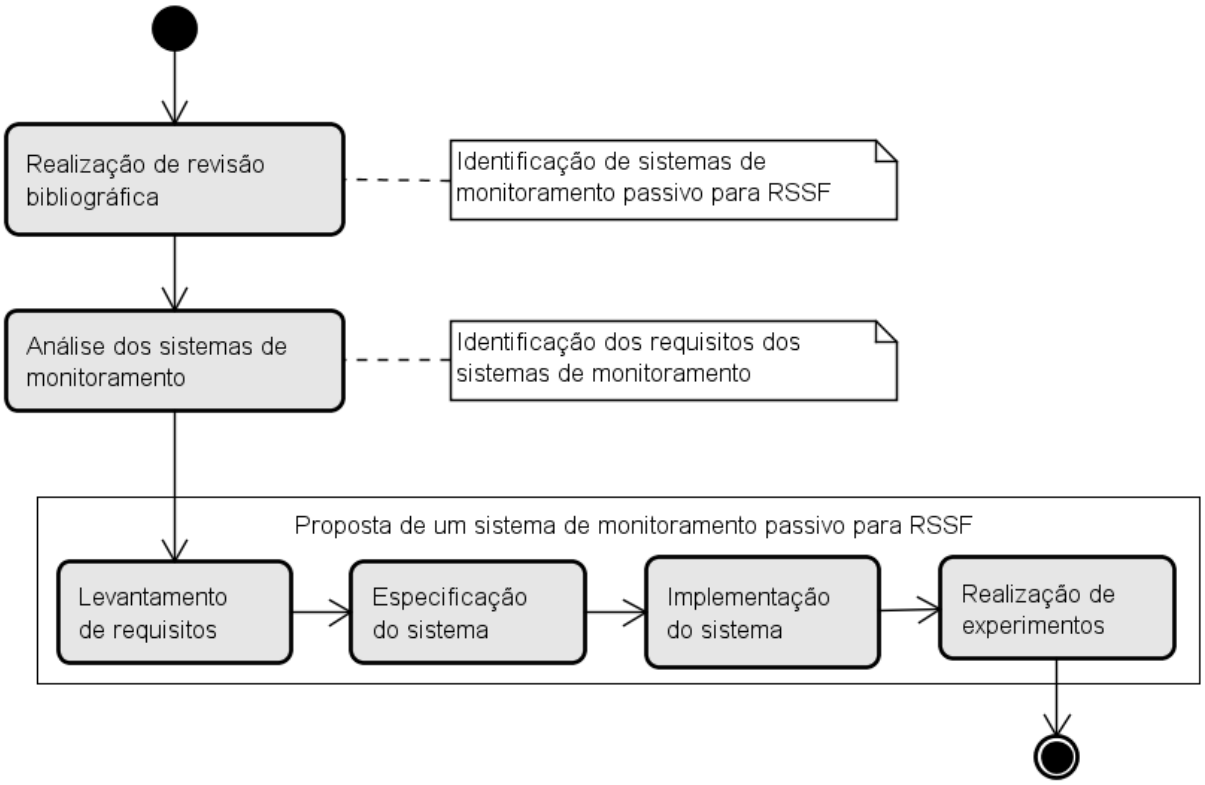


Figura 1.2 – Metodologia para o desenvolvimento da tese

1.5 Organização do documento

Este capítulo apresentou a problemática que motiva esta tese de doutorado, a hipótese assumida, as questões de pesquisa investigadas, os objetivos e as principais contribuições, como também a metodologia utilizada para alcançar os objetivos.

O restante deste documento está organizado da seguinte maneira:

Capítulo 2 – Este capítulo aborda os principais temas que compõem a base teórica desta tese de doutorado. Nele são apresentados os conceitos fundamentais relacionados às redes de sensores sem fio e ao gerenciamento de redes. Este capítulo também discute os principais desafios e requisitos de um sistema de gerenciamento de RSSF, e os conceitos relacionados com sistemas de monitoramento para RSSF.

Capítulo 3 – Este capítulo é dedicado aos trabalhos relacionados a esta tese de doutorado. Nele os principais sistemas de monitoramento passivo propostos na literatura para redes de sensores sem fio são identificados, descritos, analisados e comparados.

Capítulo 4 – Este capítulo descreve o sistema de monitoramento EPMOSt proposto nesta tese de doutorado. Nele são descritos de maneira detalhada a especificação e o funcionamento de cada um dos módulos do sistema.

Capítulo 5 – Este capítulo descreve os experimentos realizados para avaliar o sistema de monitoramento EPMOSt, e apresenta e discute os resultados obtidos.

Capítulo 6 – Este capítulo é dedicado às considerações finais da tese e possíveis direcionamentos para pesquisas futuras.

2 RSSF E GERENCIAMENTO DE REDES

Este capítulo aborda os principais assuntos que representam a base teórica desta tese de doutorado. Na seção 2.1 são apresentados conceitos fundamentais relacionados com as RSSF, seus componentes, suas características e suas aplicações. Na seção 2.2 são abordados os principais conceitos de gerenciamento de redes, incluindo o modelo FCAPS, a infraestrutura necessária para gerenciamento de redes e o protocolo de gerenciamento SNMP (*Simple Network Management Protocol*). Os principais desafios e requisitos de um sistema de gerenciamento de RSSF, e os conceitos relacionados com sistemas de monitoramento para RSSF são discutidos na seção 2.3. Por fim, na seção 2.4 são apresentadas as considerações finais deste capítulo.

2.1 Redes de sensores sem fio

Os avanços nas áreas de microeletrônica, sensoriamento, processamento de sinais, comunicação sem fio e rede propiciaram o surgimento e a evolução das RSSF. Com isso, espera-se que no futuro as RSSF tenham um impacto significativo em nossas vidas. Aplicações propostas para RSSF incluem monitoramento ambiental, predição de desastres naturais, segurança territorial, meios de transporte, produção industrial e aplicações militares [Swami et al. 2007].

Normalmente, as RSSF são compostas por uma grande quantidade de nós sensores distribuídos com a finalidade de monitorar eventos e fenômenos em um ambiente específico [Cheour et al. 2011]. Em geral, estas redes têm severas restrições de energia e devem possuir mecanismos para autoconfiguração e autoadaptação devido a problemas como falhas de comunicação e falhas dos nós. Uma RSSF requer um alto grau de cooperação entre os nós para executar as tarefas definidas para a rede [Loureiro et al. 2003].

As RSSF diferem-se de outras tecnologias de rede por exigirem a adoção de novos paradigmas de projeto. Geralmente, as RSSF são projetadas e implantadas para propósitos específicos. Portanto, o projeto da rede deve levar em consideração as especificidades da aplicação à qual a rede se destina. Assim, a natureza de difusão (*broadcast*) do meio de transmissão deve ser levada em consideração. Para sensores operados por bateria, a conservação da energia é um dos parâmetros de projeto mais importantes, pois a substituição

das baterias pode ser difícil ou até mesmo impossível em muitas aplicações. Portanto, o projeto da rede deve minimizar o consumo de energia para prolongar o tempo de vida da rede.

Além disso, as restrições de energia e largura de banda e a implantação de redes com milhares ou até milhões de nós impõem novos desafios para a alocação de recursos e gerenciamento da rede. Sendo assim, os algoritmos distribuídos tradicionais, como protocolos de comunicação, protocolos de roteamento, protocolos de gerenciamento, mecanismos de segurança, algoritmos de agregação de dados e eleição de líder, devem ser revistos para serem utilizados nestas redes [Swami et al. 2007].

2.1.1 Componentes

A Figura 2.1 mostra os componentes básicos de uma RSSF. Uma RSSF não exige a definição de uma infraestrutura, assim, o posicionamento dos nós na rede pode ser aleatório, pois os próprios nós são capazes de se comunicar e se organizar. Os nós sensores são capazes de coletar determinado tipo de dado e enviar os dados coletados por ele, ou recebidos de outro nó, para um dos nós vizinhos. Esta comunicação sem fio entre os nós é realizada até que o nó chamado de ***sink*** (sorvedouro) receba os dados. O nó sorvedouro tipicamente possui um hardware mais robusto (pode ser um microcomputador, por exemplo) e é capaz de enviar os dados coletados através da Internet, de uma conexão por satélite ou de outra infraestrutura de rede. Esta topologia baseada em múltiplos saltos é chamada de *multihop*. A topologia *multihop* permite a redução do consumo de energia na transmissão, pois evita que todos os nós da rede precisem transmitir dados diretamente ao nó sorvedouro. Em vez disso, os nós só precisam realizar transmissões aos seus vizinhos e podem utilizar rádios de baixa potência [Loureiro et al. 2003].

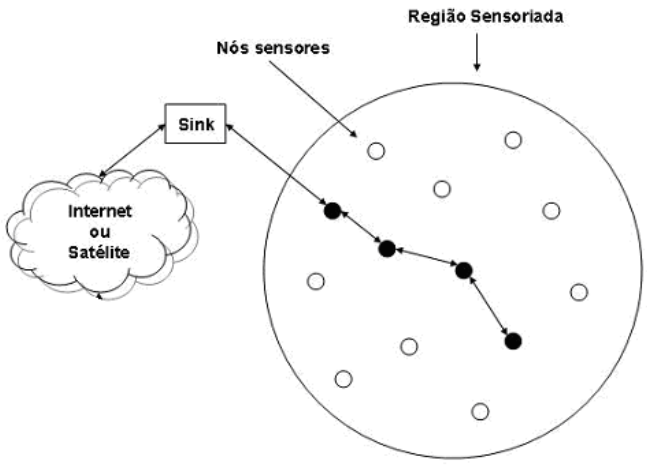


Figura 2.1 – Componentes de uma RSSF

Cada nó sensor é um elemento computacional com suprimento de energia, processador, memória, interface de comunicação sem fio e unidade de sensoriamento, conforme mostrado na Figura 2.2.

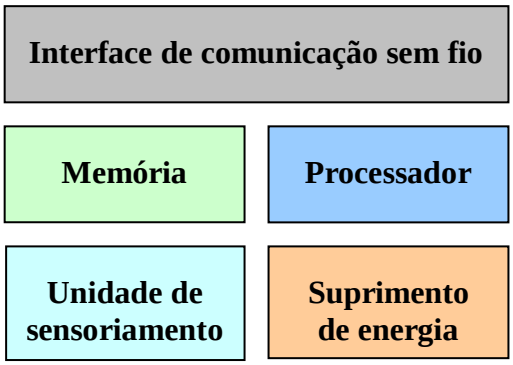


Figura 2.2 – Hardware de um nó sensor (Adaptado de [Loureiro et al. 2003])

A maioria das aplicações de RSSF utiliza pequenos sensores (vide Figura 2.3), o que consequentemente reduz o tamanho e a capacidade de seus componentes. Portanto, geralmente, as RSSF utilizam comunicação sem fio de pequeno alcance e possuem severas restrições de consumo de energia, capacidade de processamento, capacidade de memória e largura de banda.

(a) **(b)**

Figura 2.3 – Nós sensores. (a) Sensor MicaZ [Crossbow 2013]; (b) Sensor *smart dust* [UAT Robotics 2014].

2.1.2 Características

As RSSF apresentam características particulares conforme as áreas em que são aplicadas. Isto faz com que questões específicas tenham que ser resolvidas. Algumas destas características e questões são discutidas em [Loureiro et al. 2003] e [Zhang and Li 2009], e são relacionadas a seguir:

- **Endereçamento dos nós:** dependendo da aplicação, cada sensor pode ser endereçado unicamente ou não. Por exemplo, sensores colocados no corpo humano devem ser endereçados unicamente caso se deseje saber exatamente o local de onde o dado está

sendo coletado. Por outro lado, sensores monitorando o ambiente em uma dada região externa possivelmente não precisam ser identificados individualmente já que o ponto importante é saber o valor de uma determinada variável nessa região.

- **Agregação dos dados:** indica a capacidade de uma RSSF de agregar ou sumarizar dados coletados pelos sensores. Caso a rede tenha essa funcionalidade, é possível reduzir o número de mensagens que precisam ser transmitidas por ela. Neste caso, os dados coletados são agregados e enviados ao nó sorvedouro.
- **Mobilidade dos nós:** indica se os sensores podem se mover ou não em relação ao sistema em que estão coletando dados. Por exemplo, sensores colocados numa floresta para coletar dados de umidade e temperatura são estáticos, enquanto sensores colocados na superfície de um oceano para medir o nível de poluição da água são móveis.
- **Quantidade de nós:** a escalabilidade da rede é uma questão importante, pois redes contendo até milhares de sensores são previstas para aplicações ambientais como monitoramento de oceanos e florestas.
- **Limitação de energia:** em várias aplicações, os sensores poderão ser colocados em áreas remotas que não permitem acesso para sua manutenção. Neste caso, o tempo de vida de um sensor depende basicamente da capacidade da sua bateria. Para aumentar a longevidade dos sensores, os algoritmos e protocolos devem se preocupar com o consumo de energia.
- **Auto-organização da rede:** sensores podem ficar inativos por causa de sua destruição física ou falta de energia, como também podem ficar incomunicáveis devido a problemas no canal de comunicação sem fio ou por terem sido desligados por decisão de um algoritmo de gerenciamento da rede com o intuito de economizar energia. A situação contrária também pode acontecer: sensores inativos se tornarem ativos ou novos sensores passarem a fazer parte da rede. Nestas situações, é fundamental que existam mecanismos de auto-organização para que a rede continue a executar as suas funções de maneira satisfatória.
- **Tarefas colaborativas:** geralmente uma RSSF executa alguma tarefa colaborativa onde é importante detectar e estimar eventos de interesse e não apenas prover

mecanismos de comunicação. Devido às restrições das RSSF, geralmente os dados são agregados para melhorar o desempenho no processo de detecção de eventos.

- **Capacidade de responder a consultas:** uma consulta sobre uma informação coletada em uma região sensoriada pode ser encaminhada para um nó individual ou um grupo de nós. Neste caso, os nós da RSSF devem ter a capacidade de responder a estas consultas.
- **Restrições dos dados coletados:** indica se os dados coletados pelos sensores têm algum tipo de restrição como um intervalo de tempo máximo para disseminação de seus valores para uma dada entidade.

O conhecimento dessas características básicas da rede de sensores sem fio a ser gerenciada é fundamental para o desenvolvimento de soluções de gerência, pois as mesmas devem contemplar as características e limitações específicas destas redes.

2.1.3 Classificação

A classificação de uma RSSF depende de seu objetivo e da sua área de aplicação. A aplicação influencia diretamente nas funções exercidas pelos nós da rede, assim como na arquitetura desses nós (e.g., processador, memória, tipos de sensores, fonte de energia, tipo de rádio), na quantidade e distribuição dos nós que compõem a rede, na escolha dos protocolos da pilha de comunicação, no tipo de dado que será tratado, no tipo de serviço que será provido pela rede e, consequentemente, no tempo de vida da rede [Ruiz et al. 2004].

Tabela 2.1 – Classificação das RSSF quanto ao sensoriamento

Sensoriamento	
Coleta Periódica	Os nós sensores coletam dados sobre o(s) fenômeno(s) em intervalos regulares. Um exemplo são as aplicações que monitoram o canto dos pássaros, onde os sensores fazem a coleta durante o dia e permanecem desligados durante a noite, no caso de pássaros de canto diurno.
Coleta Contínua	Os nós sensores coletam os dados continuamente. Um exemplo são as aplicações de detecção de atividades sísmicas que coletam dados continuamente para a formação de base de dados para pesquisas.
Coleta Reativa	Os nós sensores coletam dados quando ocorrem eventos de interesse ou quando solicitados pelo observador. Um exemplo são as aplicações que detectam a presença de objetos na área monitorada.
Coleta Tempo real	Os nós sensores coletam a maior quantidade de dados possível no menor intervalo de tempo. Um exemplo são aplicações que envolvem risco para vidas humanas tais como aplicações em escombros ou áreas de desastres.

De acordo com [Ruiz et al. 2004], as RSSF podem ser classificadas segundo o sensoriamento (vide Tabela 2.1), a configuração (vide Tabela 2.2), o tipo de processamento (vide Tabela 2.3), e também segundo o tipo de comunicação que executa (vide Tabela 2.4).

Tabela 2.2 – Classificação das RSSF quanto à configuração

Configuração		
Composição	Homogênea	Rede composta por nós que apresentam a mesma capacidade de hardware. Eventualmente os nós podem executar softwares diferentes.
	Heterogênea	Rede composta por nós com diferentes capacidades de hardware.
Organização	Hierárquica	Rede em que os nós estão organizados em grupos (*clusters*). Cada grupo terá um líder (*cluster-head*) que poderá ser eleito pelos nós comuns. Os grupos podem organizar hierarquias entre si.
	Plana	Rede em que os nós não estão organizados em grupos.
Mobilidade	Estacionária	Todos os nós sensores permanecem no local onde foram depositados durante todo o tempo de vida da rede.
	Móvel	Rede em que os nós sensores podem ser deslocados do local onde inicialmente foram depositados.
Densidade	Balanceada	Rede que apresenta uma concentração e distribuição de nós por unidade de área considerada ideal segundo a função objetivo da rede.
	Densa	Rede que apresenta uma alta concentração de nós por unidade de área.
	Esparsa	Rede que apresenta uma baixa concentração de nós por unidade de área.
Distribuição	Irregular	Rede que apresenta uma distribuição não uniforme dos nós na área monitorada.
	Regular	Rede que apresenta uma distribuição uniforme dos nós na área monitorada.

Tabela 2.3 – Classificação das RSSF quanto ao tipo de processamento

Tipo de processamento	
Infraestrutura	Os nós sensores executam procedimentos relacionados à infraestrutura da rede como, por exemplo, algoritmos de controle de acesso ao meio, roteamento, eleição de líderes, descoberta de localização e criptografia.
Localizado	Os nós sensores executam além dos procedimentos de infraestrutura, algum tipo de processamento local básico como, por exemplo, interpretação dos dados coletados pelos sensores.
Correlação	Os nós estão envolvidos em procedimentos de correlação de dados como, por exemplo, fusão, supressão seletiva, contagem, compressão e agregação.

Tabela 2.4 – Classificação das RSSF quanto à comunicação

Comunicação		
Disseminação	Programada	Os nós disseminam dados em intervalos regulares.
	Contínua	Os nós disseminam os dados continuamente.
	Sob demanda	Os nós disseminam os dados em resposta à consulta do observador e à ocorrência de eventos.
Tipo de Conexão	Simétrica	Todas as conexões existentes entre os nós sensores, com exceção do nó sorvedouro têm o mesmo alcance.
	Assimétrica	As conexões entre os nós comuns têm alcances diferentes.
Transmissão	*Simplex*	Os nós sensores possuem transceptor que permite apenas a transmissão da informação.
	Half-Duplex	Os nós sensores possuem transceptor que permite transmitir ou receber em um determinado instante.
	Full-Duplex	Os nós sensores possuem transceptor que permite transmitir e receber dados ao mesmo tempo.
Alocação de canal	Estática	Neste tipo de rede, se existir **n** nós, a largura de banda é dividida em **n** partes iguais.
	Dinâmica	Neste tipo de rede não existe atribuição fixa de largura de banda. Os nós disputam o canal para comunicação dos dados.
Fluxo de Informação	Inundação (*Flooding*)	Neste tipo de rede, os nós sensores fazem difusão de suas informações para seus vizinhos que por sua vez repetem esse processo até alcançar o ponto de acesso. Esta abordagem promove um alto *overhead*, mas é imune às alterações dinâmicas de topologia.
	Multicast	Neste tipo de rede os nós formam grupos e usam o *multicast* para comunicação entre os membros do grupo.
	Unicast	Neste tipo de rede, os nós sensores podem se comunicar diretamente com o ponto de acesso usando protocolos de roteamento *multihop* (múltiplos saltos).
	Gossiping	Neste tipo de rede, os nós sensores selecionam os nós para os quais enviam os dados.
	Bargaining	Neste tipo de rede, os nós enviam os dados somente se o nó destino manifestar interesse.

2.1.4 Aplicações

As RSSF são eficientes para o monitoramento, rastreamento e controle de fenômenos onde a utilização de uma rede com cabos é difícil ou tem alto custo. As RSSF podem ser utilizadas em diversas áreas de aplicação, tais como monitoramento e preservação de ambientes, processos de produção industrial, automação de transportes, segurança, medicina, agricultura e logística [Karl and Willig 2005].

A seguir são descritas algumas aplicações de RSSF de forma mais detalhada, com o intuito de mostrar o enorme potencial de utilização destas redes [Loureiro et al. 2003] [Cheour et al. 2011]:

- **Monitoramento ambiental:** as RSSF podem ser utilizadas para detectar riscos naturais, tais como terremotos e inundações. Podem ser utilizadas também para monitorar os movimentos de um tornado ou detectar focos de incêndio em uma floresta.
- **Monitoramento de habitat:** nós sensores podem ser espalhados em uma região para monitorar o comportamento do seu ecossistema ou de uma espécie em particular.
- **Monitoramento estrutural de edificações:** sensores podem ser dispostos em pontos estratégicos de uma edificação para monitorar a saúde estrutural da edificação e detectar possíveis falhas estruturais.
- **Agricultura:** as RSSF têm potencial para serem amplamente utilizadas na agricultura de precisão, particularmente nas áreas de manejo da irrigação, fertilização planejada e controle de pragas.
- **Medicina:** bio sensores podem ser implantados no corpo humano para monitorar parâmetros fisiológicos do paciente, tais como batimento cardíaco ou pressão arterial. Os dados coletados pelos sensores podem então ser enviados para um centro médico caso seja detectada alguma anomalia.
- **Tráfego de veículos:** sensores podem ser utilizados para monitorar o tráfego de veículos em rodovias e malhas viárias urbanas.
- **Militar:** sensores podem detectar movimentos inimigos, explosões e a presença de materiais perigosos como gases venenosos ou radiações.

Além destas aplicações, as RSSF podem contribuir consideravelmente com o desenvolvimento da computação ubíqua. A computação ubíqua tem como objetivo tornar a interação pessoa-máquina "invisível", ou seja, integrar a computação com as ações e comportamentos naturais das pessoas [Weiser 1993]. Por exemplo, nós sensores poderiam coletar os dados de interesse do ambiente (temperatura, umidade, luminosidade, presença de pessoas, etc.) onde estão inseridos e transmitir estes dados para processadores embutidos em equipamentos domésticos (ar condicionado, aparelhos de TV, *smartphones*, *tablets*, etc.) de

modo que uma aplicação possa executar operações baseadas nos dados capturados [Borges Neto and Andrade 2010] [Ribeiro and Castro 2011].

As RSSF também podem ser utilizadas na infraestrutura da Internet das Coisas (IoT – *Internet of Things*) coletando e disponibilizando dados de sensoriamento. Segundo a *International Telecommunications Union* [ITU 2012], a IoT é uma infraestrutura global para a Sociedade da Informação, que permite serviços avançados através da interligação (física e virtual) de "coisas" baseada nas tecnologias de informação e comunicação. Diversas "coisas" podem fazer parte da IoT, tais como computadores, *smartphones*, *tablets*, sensores, *smart cards*, etiquetas RFID (*Radio-Frequency IDentification*), entre outras. As aplicações da IoT variam desde um simples controle de estoque através do uso de etiquetas RFID até sistemas de monitoramento de grandes aglomerados urbanos (cidades inteligentes) [Elkhodr et al. 2013].

2.2 Gerenciamento de redes

À medida que as redes usadas em uma organização e as aplicações distribuídas que elas aceitam crescem em tamanho e complexidade, o gerenciamento destas redes torna-se cada vez mais complexo e crítico, tornando necessária a utilização de ferramentas automatizadas de gerenciamento de redes (e.g., Nagios [Nagios 2013], Net-SNMP [Net-SNMP 2013], etc.). A necessidade destas ferramentas é cada vez maior, bem como a dificuldade em fornecê-las, se a rede incluir equipamentos de diversos fabricantes. Além disso, a crescente descentralização dos serviços de rede torna cada vez mais difícil gerenciar uma rede de maneira coerente e coordenada [Stallings 2005].

O gerenciamento de redes inclui o fornecimento, a integração e a coordenação de elementos de hardware, de software e humanos, para monitorar, testar, consultar, configurar, analisar, avaliar e controlar os recursos da rede, e de seus elementos, para satisfazer às exigências operacionais, de desempenho e de qualidade de serviço em tempo real a um custo razoável [Kurose e Ross 2013].

2.2.1 Áreas funcionais

A ISO (*International Organization for Standardization*) criou um modelo de gerenciamento de redes que é útil para situar os cenários de gerenciamento de forma mais

estruturada. Neste modelo foram definidas cinco áreas funcionais de gerenciamento de redes: gerenciamento de falhas, de desempenho, de configuração, de contabilização e de segurança [Kurose e Ross 2013].

A partir do conceito de áreas funcionais definidas pela ISO, a ITU (*International Telecommunication Union*) criou o modelo FCAPS, formado a partir das iniciais de cada área de gerenciamento, conforme mostrado na Tabela 2.5 [Kurose e Ross 2013].

O objetivo do gerenciamento de falhas é registrar, detectar e reagir às condições de falha da rede. Uma falha é qualquer "condição" que faça com que a rede opere incorretamente ou produza resultados indesejáveis. Praticamente qualquer problema na rede é uma falha, seja esta falha no hardware ou no software. São alguns exemplos de falhas: rompimentos de enlaces, erros em softwares de rede e paradas em equipamentos de rede [Comer 2006].

Tabela 2.5 – Modelo FCAPS

Áreas de gerenciamento	
F	*Fault detection and correction* (detecção e correção de falhas)
C	*Configuration and operation* (configuração e operação)
A	*Accounting and billing* (contabilização e bilhetagem)
P	*Performance assessment and optimization* (desempenho e otimização)
S	*Security assurance and protection* (segurança e proteção)

Quando ocorre uma falha, é importante tomar as seguintes medidas, o mais rapidamente possível [Stallings 2005]:

- Determinar exatamente onde a falha ocorreu;
- Isolar, quando possível, o restante da rede da falha, de modo que a rede possa continuar operando sem interferência da falha;
- Reconfigurar ou modificar a rede de modo que minimize o impacto de operar sem os componentes falhos; e
- Reparar ou substituir os componentes falhos para restaurar a rede ao seu estado inicial.

O gerenciamento de configuração tem como objetivo permitir que o gerente de rede saiba quais dispositivos fazem parte da rede administrada e quais são suas configurações de hardware e software. O gerenciamento de configuração se preocupa em inicializar uma rede e desligar toda a rede ou apenas uma parte dela. Preocupa-se também em manter, acrescentar e

atualizar as relações entre componentes e o estado dos próprios componentes durante a operação da rede [Kurose e Ross 2013]. A reconfiguração de uma rede frequentemente é desejada em resposta a uma avaliação de desempenho ou em apoio à atualização de rede, recuperação de falha ou verificações de segurança [Stallings 2005].

O gerenciamento de contabilização e bilhetagem permite que o gerente da rede especifique, registre e controle o acesso de usuários e dispositivos aos recursos da rede. Quotas de utilização, cobrança por utilização e alocação privilegiada de recursos fazem parte do gerenciamento de contabilização [Kurose e Ross 2013]. O gerente de rede poderá desejar monitorar o uso de recursos da rede por diversas razões [Stallings 2005]:

- Um usuário ou dispositivo pode estar abusando dos seus privilégios de acesso e sobrecarregando a rede;
- Os usuários ou dispositivos podem estar usando de forma ineficaz a rede, e o gerente de rede pode ajudar na alteração de procedimentos para melhorar o desempenho; e
- O gerente de rede poderá planejar o crescimento da rede, se as atividades dos usuários ou dispositivos forem conhecidas em detalhes suficientes.

O gerenciamento de desempenho e otimização tem como objetivos quantificar, medir, informar, analisar e controlar o desempenho de diferentes componentes da rede [Kurose e Ross 2013]. No gerenciamento de desempenho, três aspectos fundamentais devem ser considerados pelo gerente da rede: o que será medido, como deverá ser medido e o que fazer com os resultados das medições [Comer 2006]. O gerente da rede deve determinar quais entidades são relevantes e devem ser medidas, tais como: enlaces individuais, elementos de rede, serviços de rede e aplicações.

Alguns exemplos de medições de desempenho que podem ser realizadas são: medir o tráfego sobre um enlace e calcular a taxa de utilização deste enlace; medir a quantidade de pacotes transmitidos, recebidos e descartados em um determinado componente de rede (roteador, switch, etc.); e medir o tráfego correspondente a um determinado serviço ou aplicação da rede como, por exemplo, o tráfego correspondente a um servidor web. Os resultados das medições de desempenho podem ser utilizados para identificar possíveis gargalos antes que estes causem problemas aos sistemas finais. Por exemplo, as medições do tráfego de um enlace podem mostrar que a taxa de utilização do enlace está crescendo e alguma ação deverá ser realizada para evitar que este enlace venha futuramente a congestionar. Neste caso, o gerente da rede poderia antecipadamente ampliar a capacidade do

enlace, ou até mesmo alterar as tabelas de roteamento para encaminhar parte do tráfego sobre este enlace por outras rotas, otimizando assim os recursos da rede.

O objetivo do gerenciamento de segurança é controlar o acesso aos recursos da rede de acordo com alguma política definida [Kurose e Ross 2013]. Algumas funções do gerenciamento de segurança são: geração e distribuição de chaves de criptografia; gerenciamento de senhas e outras informações de autenticação; monitoramento e controle de acesso aos recursos da rede; coleta, armazenamento e avaliação de registros de *log* para auditoria; gerenciamento de firewalls para monitorar e controlar o acesso à rede [Stallings 2005].

2.2.2 Infraestrutura de gerenciamento de redes

Normalmente, um sistema de gerenciamento de redes possui cinco componentes principais: entidade gerenciadora, dispositivo gerenciado, agente de gerenciamento, protocolo de gerenciamento e base de informações de gerenciamento. A Figura 2.4 mostra a infraestrutura de gerenciamento de redes e seus componentes.

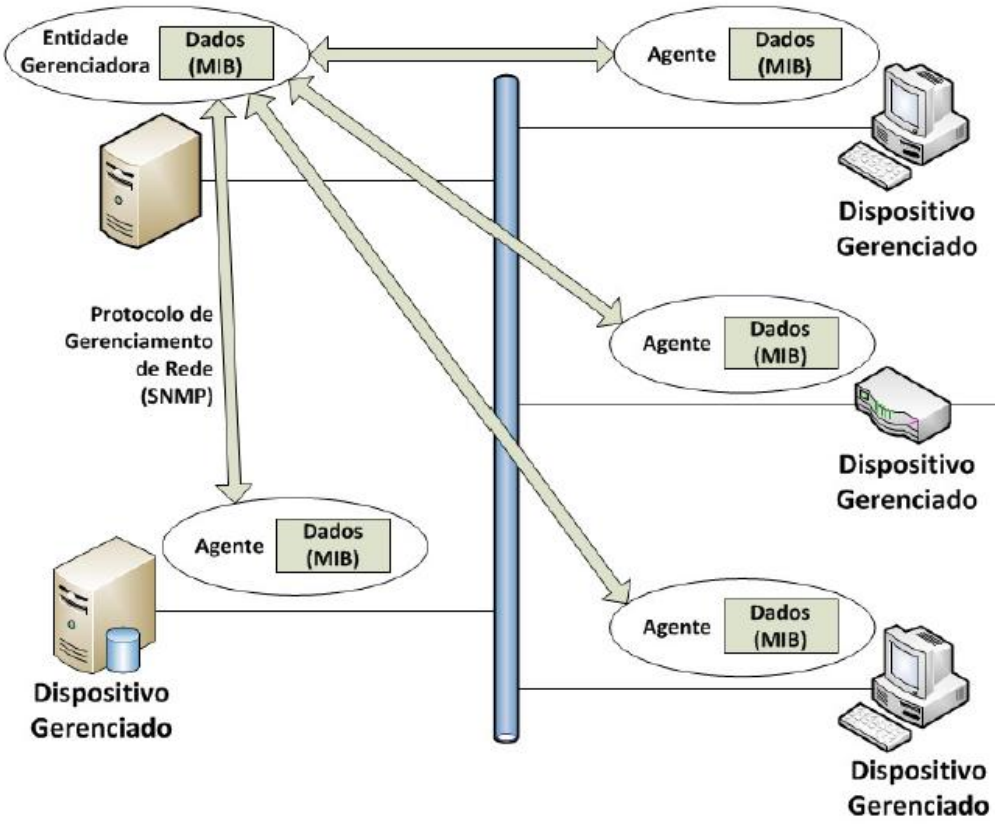


Figura 2.4 – Infraestrutura de gerenciamento (adaptado de [Kurose e Ross 2013])

A entidade gerenciadora é uma aplicação que é executada em uma estação de gerência e tem como funções coletar, processar, analisar e apresentar as informações de gerenciamento da rede. É na estação gerenciadora que o administrador humano (gerente da rede) visualiza as informações obtidas a partir do gerenciamento e interage com os dispositivos da rede.

Um dispositivo gerenciado é um equipamento de rede (incluindo seus softwares) que reside em uma rede gerenciada. Um dispositivo gerenciado pode ser um roteador, um switch, uma impressora, um computador ou qualquer outro equipamento de rede. Um dispositivo gerenciado pode conter diversos objetos gerenciados. Um objeto gerenciado é, basicamente, uma variável de dados que representa um item do dispositivo gerenciado. Por exemplo, um roteador possui diversos objetos gerenciados: endereço IP, quantidade de datagramas IP descartados, tabela de rotas, entre outros.

A coleção de objetos gerenciados é denominada de base informações de gerenciamento (*Management Information Base* - MIB). Portanto, em cada dispositivo gerenciado existe uma MIB que armazena dados sobre os objetos gerenciados daquele dispositivo.

O dispositivo gerenciado possui também um agente de gerenciamento, que é um processo (software) que se comunica com a entidade gerenciadora e que executa ações locais nos dispositivos gerenciados sob o comando e controle da entidade gerenciadora. O agente é responsável por atualizar as informações da MIB de acordo com o estado do dispositivo. Por exemplo, em um roteador, o agente pode coletar informações sobre a quantidade de datagramas IP descartados e armazená-las na MIB do roteador. O agente também pode alterar a configuração de um parâmetro do dispositivo para atender um comando da entidade gerenciadora. Por exemplo, a entidade gerenciadora pode enviar um comando para o agente que executa em um roteador pedindo para este agente alterar a tabela de rotas.

Para viabilizar a comunicação entre a entidade gerenciadora e os processos agentes que executam nos dispositivos gerenciados, é necessária a utilização de um protocolo de gerenciamento. Assim, a entidade gerenciadora utiliza o protocolo de gerenciamento para interrogar o estado dos dispositivos gerenciados (enviando mensagens solicitando valores dos objetos armazenados na MIB) e alterar o estado dos dispositivos (enviando mensagens solicitando a alteração de valores dos objetos armazenados na MIB). Alguns protocolos de gerenciamento permitem também que o agente informe à entidade gerenciadora a ocorrência de eventos excepcionais, como falhas de componentes ou violações de parâmetros de

desempenho. O agente de um roteador pode, por exemplo, enviar uma mensagem para a entidade gerenciadora informando o rompimento de um enlace de comunicação.

A infraestrutura de gerenciamento mostrada na Figura 2.4 é genérica e se aplica, em geral, a vários padrões de gerenciamento de redes que vêm sendo propostos há anos. Os padrões de gerenciamento mais difundidos são o CMIP (*Common Management Information Protocol*) [ITU-T 1992] definido pela ISO (*International Organization for Standardization*), o TMN (*Telecommunications Management Network*) [ITU-T 1995] definido pelo ITU (*International Telecommunication Union*) e o SNMP (*Simple Network Management Protocol*) [RFC 1157 1990] definido pela IETF (*Internet Engineering Task Force*). Atualmente, o SNMP é o protocolo de gerenciamento mais usado e disseminado [Kurose e Ross 2013]. O SNMP será abordado de forma detalhada na próxima seção.

2.2.3 Protocolo de gerenciamento SNMP

O SNMP (*Simple Network Management Protocol*) é um protocolo de gerenciamento de redes amplamente usado em redes baseadas na arquitetura TCP/IP. A primeira versão do SNMP foi proposta pela IETF em 1989. Desde então o SNMP evoluiu do SNMPv1 para o SNMPv2 e chegou à sua versão mais recente, o SNMPv3, lançada em 1999 e atualizada em 2002, onde o aspecto de segurança foi bastante melhorado em relação ao SNMPv2 [Stallings 2005].

A definição da estrutura de gerenciamento do SNMP inclui cinco itens [Comer 2006]:

- **Formato das mensagens:** o protocolo SNMP especifica os formatos das mensagens que serão utilizadas para a comunicação entre a entidade gerenciadora e os dispositivos gerenciados.
- **Codificação das mensagens:** o SNMP especifica como as mensagens são codificadas em binário para que possam ser enviadas através da rede.
- **Especificação dos objetos gerenciados:** os objetos gerenciados devem ser especificados em uma MIB de acordo com as características do dispositivo gerenciado. Portanto, a MIB definida para um determinado dispositivo deve conter os objetos relacionados com aquele dispositivo. Além disso, a especificação da MIB é totalmente independente do formato e da codificação das mensagens.
- **Agente:** o agente é uma entidade implementada em hardware e/ou software que é executado no dispositivo gerenciado e usa o protocolo SNMP para comunicar-se com a

entidade gerenciadora. Em geral, o fabricante do equipamento é responsável por desenvolver e disponibilizar o agente SNMP que executa no dispositivo.

- **Entidade gerenciadora:** é o software que executa na estação de gerência e provê a interface entre o gerente da rede e o protocolo SNMP. Este software atende as solicitações do gerente da rede e envia mensagens SNMP para os dispositivos gerenciados, e recebe respostas dos dispositivos e mostra os resultados para o gerente. Existem algumas ferramentas de gerência gratuitas que suportam o protocolo SNMP, tais como *Nagios* [Nagios 2013], *Net-SNMP* [Net-SNMP 2013], *SNMP MIB Browser Android Tool* [MIB Browser 2013] e *ManageEngine MIB Browser Free Tool* [ManageEngine 2014], e que podem desempenhar a função de entidade gerenciadora em uma rede.

2.2.3.1 Base de informações de gerenciamento (MIB)

A MIB (*Management Information Base*) é uma estrutura de dados que armazena objetos gerenciados cujos valores, coletivamente, refletem o estado atual da rede. Esses valores podem ser consultados e/ou alterados por uma entidade gerenciadora através do envio de mensagens SNMP ao agente que está executando no dispositivo gerenciado [Kurose e Ross 2013].

Os objetos gerenciados são definidos de acordo com a SMI (*Structure of Management Information*). A SMI é uma linguagem de definição de dados e objetos especificada pela IETF para assegurar que a sintaxe e a semântica dos dados de gerenciamento sejam bem definidas e não apresente ambiguidades. Portanto, no SNMP, os objetos gerenciados devem ser definidos de acordo com a linguagem SMI e devem ser agrupados em **módulos MIB** [Kurose e Ross 2013].

Existem módulos MIB padronizados pela IETF para roteadores, switches, computadores e diversos outros equipamentos de rede. Além disso, existem módulos MIB privados que são definidos pelos próprios fabricantes dos equipamentos. Para permitir a utilização e integração de vários módulos em uma mesma rede, a IETF adotou uma estrutura padronização de identificação (nomeação) de objetos publicada pela ISO denominada ASN1 (*Abstract Syntax Notation 1*).

Na estrutura de nomeação ASN1, os objetos são nomeados hierarquicamente, conforme pode ser observado na Figura 2.5. Observe que cada nó da árvore possui um nome e

um número. Desta forma, qualquer nó da árvore pode ser identificado pela sequência de nomes ou números que especificam o trajeto da raiz até o nó. A árvore ASN1 foi definida pela ISO para nomear objetos em várias áreas de interesse. Em nosso caso, estamos interessados em nomear objetos de rede, que estão sob o nó *Internet* (1.3.6.1 ou *iso.org.dod.internet*). Sob o nó *Internet* existem sete outros nós, dentre os quais se destacam os nós *management*, *private* e *experimental*. Sob o nó *management* e MIB-2 encontram-se a definição dos módulos MIB padronizados pela IETF: *system*, *interface*, *ip*, *tcp*, entre outros. Sob o nó *private* encontram-se a definição de objetos de mais de quatro mil empresas registradas na IETF. Sob o nó *experimental* poderão ser nomeados objetos que estão em fase de desenvolvimento e testes.

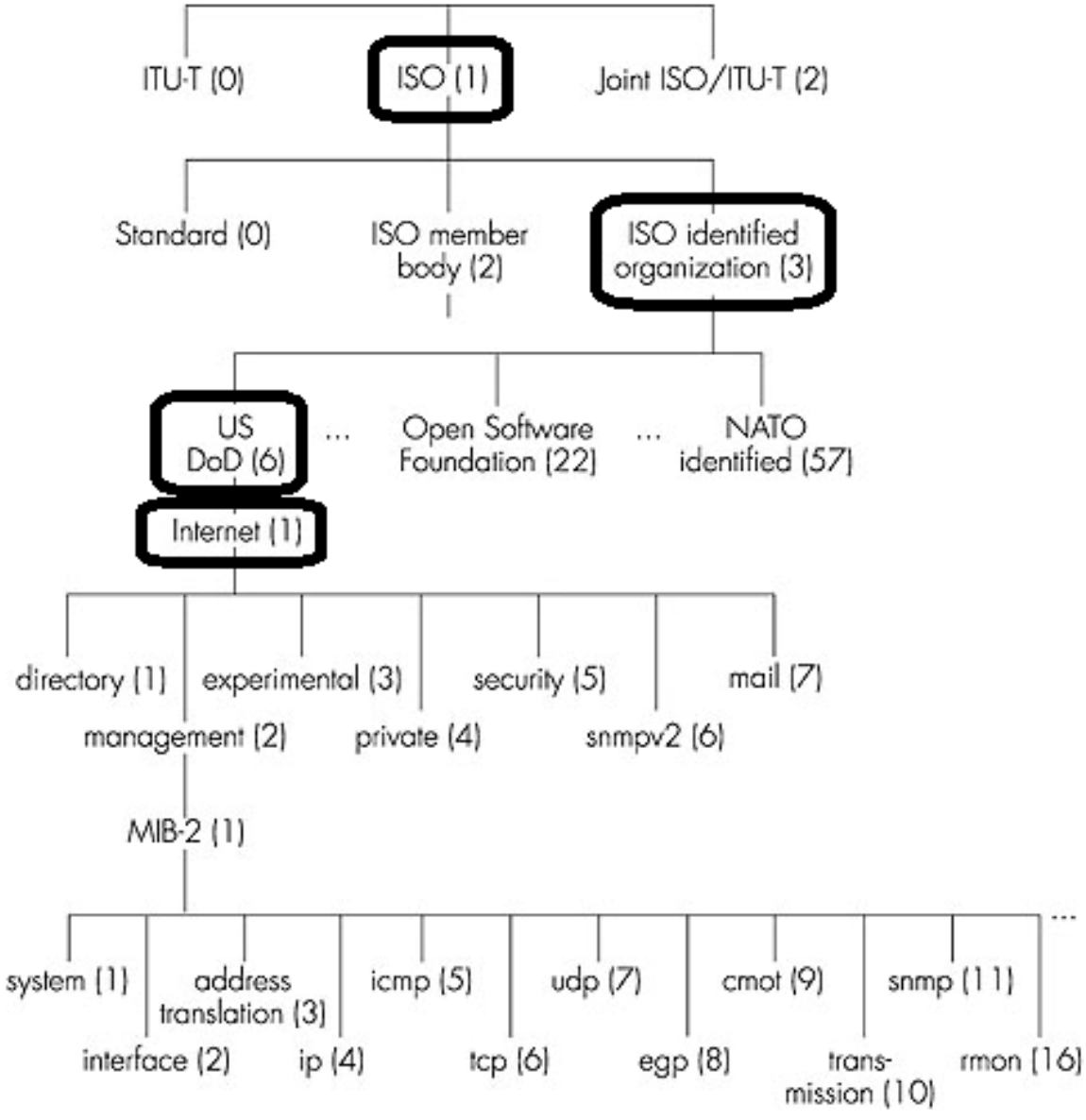


Figura 2.5 – Parte da árvore de nomeação de objetos ASN1 (Adaptado de [Kurose e Ross 2013])

A Figura 2.5 mostra apenas uma parte da árvore ASN1. No entanto, cada módulo MIB padronizado possui uma subárvore com a identificação de seus objetos. A Figura 2.6 mostra os objetos que estão sob o módulo *system*. Desta forma, o objeto *sysDescr*, que armazena a descrição do dispositivo gerenciado, pode ser identificado como 1.3.6.1.2.1.1.1 ou *iso.org.dod.internet.mgmt.mib2.system.sysdescr*.

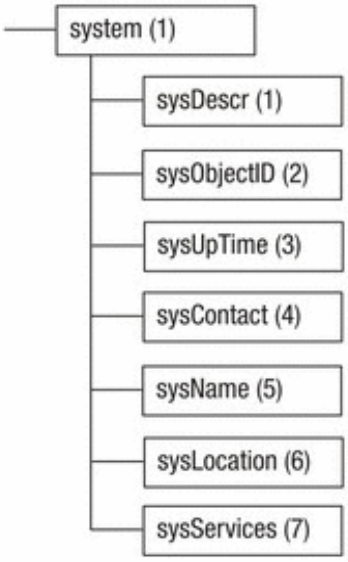


Figura 2.6 – Subárvore de nomeação de objetos do módulo *System*

2.2.3.2 Mensagens SNMP

O protocolo SNMP provê a comunicação entre a entidade gerenciadora e os agentes. Esta comunicação pode ser realizada de dois modos: modo **comando-resposta** e modo **evento**.

No modo comando-resposta, a entidade gerenciadora envia uma mensagem de requisição a um agente, que então realiza alguma ação e envia uma mensagem de resposta à entidade gerenciadora. A mensagem de requisição pode ser usada para consultar ou modificar valores de objetos da MIB do dispositivo gerenciado.

Em geral, as ferramentas de gerenciamento, tais como Nagios [Nagios 2013] e Net-SNMP [Net-SNMP 2013], enviam periodicamente mensagens de consulta com o intuito de obter informações sobre os dispositivos e mostrar estas informações em uma interface para o gerente da rede. Este mecanismo de consulta periódica é denominado de *polling*. Vale ressaltar que existe um compromisso entre o intervalo de *polling* e a tráfego gerado pelo sistema de gerência. Ao utilizar um intervalo de *polling* pequeno, o gerente da rede obtém

informações mais recentes sobre o estado dos dispositivos, porém aumenta-se o tráfego gerado pelo protocolo SNMP. Em contrapartida, aumentando-se o intervalo de *polling*, o tráfego gerado pelo protocolo SNMP diminui, porém as alterações nos estados dos dispositivos são visualizadas pelo gerente de rede com um retardo maior.

No modo evento, denominado também de modo *trap*, o agente envia uma mensagem para a entidade gerenciadora sem que esta tenha solicitado. As mensagens *trap* são utilizadas para notificar uma entidade gerenciadora na ocorrência de um evento excepcional que resulte em alterações nos valores dos objetos da MIB de um dispositivo gerenciado. Por exemplo, uma mensagem de *trap* pode ser enviada pelo agente de um roteador quando ocorrer a queda de um enlace que esteja conectado a este roteador.

É comum que as ferramentas de gerenciamento utilizem os dois modos de operação simultaneamente. Neste caso, a entidade gerenciadora utiliza *polling* para interrogar periodicamente o estado dos dispositivos e os agentes são configurados para enviar mensagens de *trap* para a entidade gerenciadora na ocorrência de eventos excepcionais.

O SNMPv2 define sete tipos de mensagens, que são descritas a seguir [Schmidt 2001]:

- ***GetRequest***: mensagem enviada da entidade gerenciadora a um agente para requisitar o valor de um ou mais objetos MIB.
- ***GetNextRequest***: mensagem enviada da entidade gerenciadora a um agente para requisitar o valor do próximo objeto da MIB. Deve ser utilizado após um *GetRequest* e permite que uma entidade gerenciadora descubra a estrutura de uma MIB de um dispositivo enviando mensagens *GetNextRequest* consecutivamente.
- ***GetBulkRequest***: mensagem enviada da entidade gerenciadora a um agente para requisitar grandes blocos de dados. Um mensagem *GetBulkRequest* pode ser utilizada, por exemplo, para solicitar a tabela de rotas de um roteador.
- ***SetRequest***: mensagem enviada da entidade gerenciadora a um agente para alterar o valor de um ou mais objetos MIB.
- ***InformRequest***: mensagem enviada de uma entidade gerenciadora para outra entidade gerenciadora informando os valores dos objetos MIB gerenciados pela primeira. Em uma rede grande, pode ser necessário utilizar mais de uma entidade gerenciadora. Neste caso, a mensagem *InformRequest* permite que o gerente da rede possa visualizar o estado de todos os dispositivos gerenciados a partir de uma única entidade gerenciadora.

- ***Response***: mensagem enviada do agente a uma entidade gerenciadora em resposta a uma das cinco mensagens descritas anteriormente.
- ***Trap*****:** mensagem enviada do agente a uma entidade gerenciadora informando a ocorrência de um evento excepcional.

É importante salientar que os identificadores dos objetos MIB, abordados na seção 2.2.3.1, cujos valores estão sendo requisitados ou alterados são especificados no corpo das mensagens. Os formatos detalhados de cada uma das sete mensagens definidas no protocolo SNMP podem ser encontrados em [Schmidt 2001], [Stallings 2005] e [Kurose e Ross 2013].

2.3 Gerenciamento de RSSF

As técnicas de gerenciamento tradicionais não se aplicam diretamente às RSSF, pois estas técnicas têm como principais objetivos minimizar o tempo de resposta e prover informações abrangentes sobre a rede, enquanto que nas RSSF as técnicas de gerenciamento têm como um dos principais objetivos reduzir o consumo de energia da rede através da redução do tráfego entre os nós [Zhang and Li, 2009].

Em vários cenários, as RSSF devem operar continuamente durante um longo período de tempo. Nestes cenários, o gerenciamento de redes é importante para manter a rede funcionando corretamente. Apesar de existirem diversos protocolos e sistemas de gerência para RSSF, ainda não existe uma solução de gerenciamento padronizada para estas redes [Sohraby et al. 2007].

2.3.1 Desafios

As características e restrições das RSSF impõem alguns desafios específicos para o gerenciamento destas redes [Sohraby et al. 2007]:

i. Em geral, as aplicações utilizadas em RSSF são adaptativas e devem minimizar a utilização dos recursos da rede. Portanto, o sistema de gerência deve ser capaz de obter informações sobre os recursos dos nós e da rede. Por exemplo, deve-ser conhecer o nível de bateria de um nó sensor antes de alterar seu modo de operação de ativo para inativo ou vice-versa. As redes tradicionais não possuem este requisito.

ii. Muitas aplicações de RSSF precisam garantir que toda a área de monitoramento esteja sendo monitorada pela rede. Nestas aplicações, o gerenciamento da topologia pode ser

utilizado para descobrir áreas de monitoramento que não estejam cobertas pela rede. Geralmente, existem três abordagens que podem ser utilizadas pelo gerenciamento de topologia para aumentar a área de cobertura da rede: aumentar a potência das interfaces de rádio dos nós sensores, aumentar a quantidade de nós sensores na área que está sendo monitorada, e redistribuir os nós na área de monitoramento.

iii. Nas RSSF, os nós geralmente são organizados de forma *ad hoc*. Os parâmetros desta rede *ad hoc* podem então ser obtidos e alterados pelo sistema de gerência de rede.

iv. A colaboração e a cooperação entre os nós são necessárias para otimizar o desempenho do sistema. A gerência de rede é uma ferramenta eficiente para prover a plataforma necessária para esta finalidade.

2.3.2 Requisitos

Um sistema de gerenciamento para RSSF pode coletar diversas informações sobre o funcionamento da rede, tais como nível de bateria, consumo de energia, cobertura dos nós, topologia, largura de banda e latência da rede. A partir da análise das informações coletadas, o sistema de gerência pode também atuar sobre a rede alterando a frequência de operação e a largura de banda da rede, ligando e desligando nós, alterando a topologia da rede e realizando outras ações com o intuito de recuperar eventuais falhas ou melhorar o desempenho da rede.

Um sistema de gerência eficiente para RSSF deve ser autônomo. O gerenciamento de configuração deve permitir a auto-organização e autoconfiguração dos nós sensores. Devido às RSSF sofrerem pouca ou quase nenhuma intervenção humana após a sua implantação, é desejável que o gerenciamento de falhas possua autodiagnóstico e autocura. Outro desafio importante é que a falha em um nó ou em poucos nós não impacte a operação da rede inteira [Zhang and Li 2009].

Desta forma, um sistema de gerência para RSSF deve atender os requisitos específicos de tais redes. Os principais requisitos são relacionados a seguir [Sohraby et. al 2007] [Yu et. al 2006]:

i. As soluções de gerência devem ser eficientes no que se refere ao consumo de energia. Além disso, estas soluções devem usar a menor largura de banda possível, pois a comunicação tem um custo energético muito alto em RSSF.

ii. As soluções de gerência devem ser simples e práticas, pois as RSSF são sistemas distribuídos com recursos limitados.

iii. As soluções de gerência devem ser escaláveis, pois as RSSF podem ser constituídas de dezenas a milhares dc nós.

iv. Uma MIB para RSSF deve conter um modelo de informação geral para os nós sensores, para as características da rede e para as aplicações, e as soluções de gerência devem disponibilizar as informações através de uma agente SNMP.

v. As soluções de gerência devem permitir a integração com a arquitetura TCP/IP, pois em muitos cenários as informações monitoradas pela RSSF são disponibilizadas em um computador conectado a uma rede local ou à Internet.

vi. As soluções de gerência devem ser aplicáveis a diferentes plataformas de sensores.

vii. As soluções de gerência devem ser robustas e tolerantes a falhas devendo se adaptar às mudanças de topologia da rede, pois é comum um ou mais nós da rede morrerem, serem ligados ou desligados ou perderem a comunicação com outros nós, como também a inserção de novos nós na rede.

viii. As soluções de gerência para RSSF devem prover uma interface para as aplicações, pois as aplicações podem melhorar seu desempenho a partir da obtenção de informações de gerenciamento.

ix. As soluções de gerência devem ser implementadas como um *middleware*, para facilitar a integração da solução de gerência com diferentes plataformas de sensores.

Estes requisitos impossibilitam a utilização de protocolos e sistemas de gerência tradicionais (e.g., protocolo SNMP) em RSSF, e reforçam a necessidade de sistemas de gerenciamento específicos para RSSF.

2.3.3 Sistemas de monitoramento de RSSF

Um sistema de gerenciamento completo deve abranger as cinco áreas funcionais definidas pelo modelo FCAPS, conforme mostrado na Figura 2.7. Por outro lado, os sistemas de monitoramento abrangem apenas as áreas de detecção de falhas e medição de desempenho.

O monitoramento de uma RSSF é importante para depurar e analisar o seu funcionamento. Utilizando-se um sistema de monitoramento, várias informações sobre o

funcionamento da RSSF podem ser obtidas, tais como descoberta de topologia, morte e reinicialização de nós, nós isolados (sem comunicação com outros nós), laços de roteamento, perda de pacotes, latência da rede e detecção de tráfego anormal [Ringwald and Romer 2007].

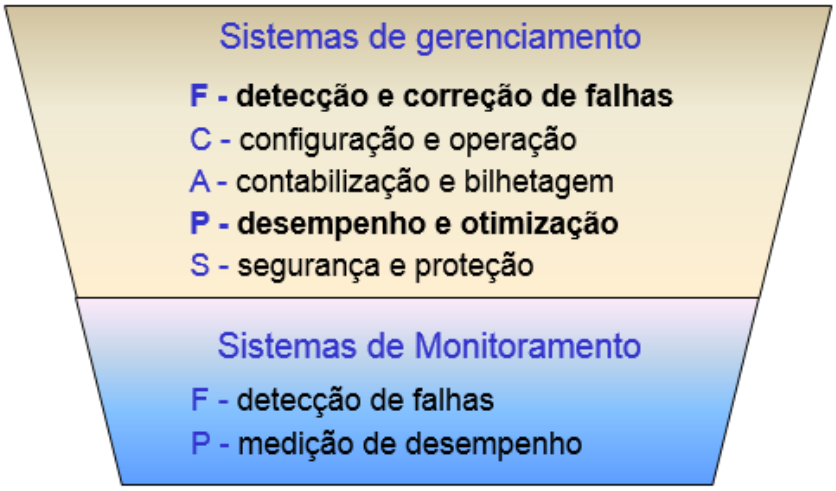


Figura 2.7 – Sistemas de gerenciamento vs. sistemas de monitoramento

Em RSSF, o monitoramento da rede pode ser dividido em monitoramento ativo e monitoramento passivo.

No monitoramento ativo são inseridas linhas de código na aplicação que executa nos nós sensores para obter informações sobre o funcionamento da rede. Neste caso, os pacotes de monitoramento são enviados juntamente com os pacotes de dados da rede, alterando o comportamento e funcionamento da rede monitorada, e consumindo os recursos desta rede. Por exemplo, no sistema de monitoramento ativo Simpathy [Ramanathan et al. 2005] cerca de 30% da largura de banda da rede é utilizada pelo tráfego gerado pelo monitoramento. Além disso, falhas na RSSF podem afetar o sistema de monitoramento e impedir a entrega de informações de monitoramento quando estas são mais necessárias. No entanto, o monitoramento ativo permite que o próprio nó sensor possa analisar as informações obtidas com o monitoramento e alterar seus parâmetros de funcionamento de acordo com estas informações. Sistemas de monitoramento ativo para RSSF são propostos em [Girod et al. 2007], [Tolle and Culler 2005], [Suhonen et al. 2008] e [Chen and Shin 2009].

No monitoramento passivo, uma rede de monitoramento adicional é implantada juntamente com a rede que deve ser monitorada (denominada de rede alvo). A rede de monitoramento captura e analisa os pacotes transmitidos pela rede alvo, não consumindo nenhum recurso da rede alvo [Xu et al. 2011]. Além disso, uma falha na rede alvo não compromete o funcionamento do sistema de monitoramento [Hänninen at al. 2011].

Entretanto, a implantação de uma rede adicional com o propósito de monitorar a rede alvo incrementa o custo total da RSSF.

Quando se deseja reduzir o custo de implantação da RSSF, é melhor utilizar um sistema de monitoramento ativo, pois não é necessário implantar uma rede de monitoramento adicional. Por outro lado, quando se deseja reduzir a utilização de recursos da rede alvo e/ou isolar as falhas da rede alvo das falhas da rede de monitoramento, é melhor utilizar um sistema de monitoramento passivo.

No entanto, utilizando-se monitoramento passivo não é possível garantir que todos os pacotes enviados pela rede alvo sejam capturados. Devido ao alcance limitado do rádio dos nós da rede alvo, os nós da rede de monitoramento somente conseguem capturar os pacotes quando estão localizados na vizinhança do remetente. Além disso, as colisões de pacotes que ocorrem no canal de comunicação sem fio, os sinais de rádio de fraca intensidade e a limitação de recursos contribuem para que os nós da rede de monitoramento não capturem todos os pacotes enviados pelos nós da rede alvo [Xu et al. 2010]. Sendo assim, os sistemas de monitoramento passivo devem se preocupar em reduzir a quantidade de pacotes enviados pela rede alvo que não são capturados. Portanto, a utilização de mecanismos para reduzir o consumo de energia da rede de monitoramento, com o intuito de prolongar o tempo de vida desta rede, sem aumentar a quantidade de pacotes enviados pela rede alvo que não são capturados é uma tarefa desafiante no desenvolvimento de sistemas de monitoramento passivo para RSSF.

2.4 Considerações finais

Neste capítulo, os principais assuntos que representam a base teórica desta tese de doutorado foram discutidos. Foram apresentados conceitos fundamentais relacionados às RSSF e ao gerenciamento de redes. Além disso, foram apresentados os principais desafios e requisitos de um sistema de gerenciamento de RSSF, e os conceitos relacionados com sistemas de monitoramento para RSSF. O conhecimento dos conceitos apresentados neste capítulo foi necessário para a concepção e o desenvolvimento do sistema de monitoramento passivo proposto no capítulo 4. O próximo capítulo aborda os principais sistemas de monitoramento passivo para RSSF propostos na literatura.

3 SISTEMAS DE MONITORAMENTO PASSIVO PARA RSSF

Neste capítulo, os principais sistemas de monitoramento passivo propostos na literatura para redes de sensores sem fio são descritos, analisados e comparados. A seção 3.1 descreve as etapas da revisão bibliográfica utilizada para identificar os sistemas de monitoramento passivo propostos especificamente para RSSF que são descritos neste capítulo. Nas seções 3.2 a 3.6, os sistemas identificados na revisão bibliográfica são descritos de forma detalhada. Na seção 3.7 é realizada uma análise comparativa entre os sistemas de monitoramento descritos. Por fim, na seção 3.8 são apresentadas as considerações finais deste capítulo.

3.1 Etapas da revisão bibliográfica

Com o intuito de identificar os trabalhos relacionados com esta tese de doutorado foi realizada uma revisão bibliográfica baseada no guia para revisão sistemática proposto em [Kitchenham 2004]. O objetivo desta revisão bibliográfica foi identificar os artigos que propõem sistemas de monitoramento passivo para RSSF cujo funcionamento é independente do funcionamento da rede a ser monitorada (rede alvo). Mais especificamente, para isso estes sistemas devem contemplar duas propriedades (**PROP**):

PROP01: Utilizar uma rede de monitoramento independente constituída por um conjunto de *sniffers* que capturam os pacotes enviados pelos nós da rede alvo de forma promíscua.

PROP02: Não realizar nenhuma instrumentação (modificação do código) na aplicação que é executada pelos nós da rede alvo.

As etapas envolvidas na revisão bibliográfica são descritas a seguir:

i. *Questões de pesquisa:* para atingir o objetivo da revisão bibliográfica descrito anteriormente, foram definidas as seguintes questões de pesquisa:
   a) *Questão principal*: Quais são os sistemas de monitoramento passivo propostos especificamente para RSSF?
   b) *Questão secundária 1*: Quais destes sistemas contemplam a propriedade **PROP01**?
   c) *Questão secundária 2*: Quais destes sistemas contemplam a propriedade **PROP02**?

ii. *Definição da string de busca:* com base nas palavras-chave das questões de pesquisa, foi elaborada a seguinte string de busca:

> *(("sensor network" or "sensor networks") and "wireless") and ("passive inspection" or "passive monitoring" or "passive diagnosis" or "passive sniffer" or "passive sniffers")*

iii. *Definição das fontes de pesquisa:* foram realizadas buscas nas seguintes bases: Scopus (http://www.scopus.com), IEEE Xplore Digital Library (http://ieeexplore.ieee.org/xplore) e ACM Digital Library (http://dl.acm.org).

iv. *Definição dos critérios de inclusão e exclusão:* (a) O artigo deve apresentar um sistema de monitoramento passivo proposto especificamente para RSSF; (b) O artigo deve estar disponível na web; e (c) O artigo deve estar escrito em português ou inglês.

v. *Seleção dos artigos:* Para realizar esta etapa foram executadas as seguintes atividades:
   a) A string de busca (etapa *ii*) foi executada em cada uma das fontes de pesquisa (etapa *iii*). Os resultados da busca em cada uma das bases foram exportados para o formato ***BibTex*** e posteriormente importados para a ferramenta StArt[1] [Start 2014].
   b) Para selecionar um conjunto inicial de artigos, os títulos e resumos de todos os artigos resultantes das buscas foram lidos e confrontados com os critérios definidos na etapa *iv*.
   c) Os artigos selecionados no item **b** foram lidos por completo e novamente confrontados com os critérios definidos na etapa *iv*. Os artigos aceitos foram documentados e encaminhados para extração dos dados, enquanto que os artigos rejeitados foram documentados com uma justificativa de exclusão.

vi. *Extração dos dados:* Os artigos aceitos foram novamente lidos por completo para se extrair as informações relevantes com as questões de pesquisa definidas na etapa *i*.

Esta revisão bibliográfica foi conduzida nos meses de Março e Abril de 2014 e a seleção dos artigos seguiu o procedimento definido na etapa *v*. Conforme pode ser observado na Tabela 3.1, após a execução da string de busca (etapa *v.a*) em cada uma das fontes de pesquisa foram identificados **123** artigos, sendo **25** da Scopus, **17** do IEEE e **81** da ACM.

[1] Ferramenta desenvolvida pelo Laboratório de Pesquisa em Engenharia de Software (LaPES) da Universidade Federal de São Carlos (UFSCar) com o intuito de auxiliar a condução do processo de revisão sistemática.

Após os títulos e resumos destes **123** artigos serem lidos e confrontados com os critérios de inclusão e exclusão (etapa *v.b),* **16** artigos foram inicialmente aceitos, **88** artigos foram rejeitados e **19** artigos foram identificados como duplicados.

Tabela 3.1 – Artigos identificados na revisão bibliográfica

Fonte de pesquisa	Artigos			
	Identificados	Inicialmente aceitos	Rejeitados	Duplicados
ACM	81	5	72	4
IEEE	17	2	5	10
Scopua	25	9	11	5
Total	**123**	**16**	**88**	**19**

Em seguida, os **16** artigos inicialmente aceitos foram lidos por completo e novamente confrontados com os critérios de inclusão e exclusão (etapa *v.c*). Nesta etapa, três artigos foram identificados como sendo uma versão anterior de artigos publicados posteriormente pelos mesmos autores e, portanto, foram descartados. Um artigo ([Garcia et al. 2012]) foi rejeitado porque apresenta as ideias iniciais desta tese de doutorado. Os **12** artigos restantes foram aceitos e encaminhados para extração de dados (etapa *vi*), quais sejam: SNTS [Khan et al. 2007], SNIF [Ringwald and Romer 2007], Pimoto [Awad et al. 2008], LiveNet [Chen et al. 2008], [Romer and Ringwald 2008], [Liu et al. 2010], LoNI [Yang et al. 2011], [Hanninen et al. 2011], PMSW [Xu et al. 2011], DSD [Jiangwu et al. 2012], [Kunzel at al. 2012] e Minerva [Sommer and Kusy 2013].

Durante a etapa de extração de dados (etapa *vi*) foram identificadas quais propriedades (**PROP01** e/ou **PROP02**) são contempladas por cada um dos sistemas de monitoramento propostos nos **12** artigos selecionados na revisão bibliográfica, conforme pode ser observado na Tabela 3.2.

O sistema de monitoramento proposto por [Hanninen et al. 2011] não contempla a propriedade **PROP01** porque utiliza apenas um sniffer, que deve ser manualmente implantando próximos aos nós da rede alvo que se deseja monitorar. O Minerva [Sommer and Kusy 2013] não contempla a propriedade **PROP01** porque cada nó da rede alvo é conectado fisicamente, através do seu pino de debug, a um nó de monitoramento. No DSD [Jiangwu et al. 2012], as falhas da rede alvo são detectadas analisando-se apenas os dados recebidos pelo nó sorvedouro. Portanto, o DSD também não contempla a propriedade **PROP01**, pois não

utiliza nenhuma rede de monitoramento. O sistema de monitoramento proposto por [Kunzel et al. 2012] foi projetado especificamente para monitorar RSSF que utilizam o padrão WirelessHART [HART 2011]. WirelessHART é um protocolo de comunicação sem fio projetado para operar em ambientes industriais. Portanto, este sistema não contempla a propriedade **PROP01**, pois o funcionamento da rede de monitoramento depende do protocolo de comunicação utilizado pela rede alvo.

Tabela 3.2 – Sistemas de monitoramento passivo para RSSF

Artigo	Propriedade	
	PROP01	PROP02
SNTS [Khan et al. 2007]	✓	✓
SNIF [Ringwald and Romer 2007]	✓	✓
Pimoto [Awad et al. 2008]	✓	✓
LiveNet [Chen et al. 2008]	✓	✓
[Romer and Ringwald 2008]	✓	
[Liu et al. 2010]	✓	
LoNI [Yang et al. 2011]	✓	
[Hanninen et al. 2011]		✓
PMSW [Xu et al. 2011]	✓	✓
DSD [Jiangwu et al. 2012]		✓
[Kunzel at al. 2012]		✓
Minerva [Sommer and Kusy 2013]		✓

O LoNI [Yang et al. 2011] e o sistema proposto por [Liu et al. 2010] não contemplam a propriedade **PROP02** porque a aplicação que é executada pelos nós da rede alvo é modificada para que o próprio nó inclua seu endereço (*id*) em todos os pacotes que são encaminhados através dele. O sistema de monitoramento proposto por [Romer and Ringwald 2008] também não contempla a propriedade **PROP02**, pois a aplicação que é executada pelos nós da rede alvo é modificada para enviar informações sobre o status dos nós.

Conforme mostrado na Tabela 3.2, apenas os sistemas de monitoramento passivo SNTS [Khan et al. 2007], SNIF [Ringwald and Romer 2007], Pimoto [Awad et al. 2008], LiveNet [Chen et al. 2008] e PMSW [Xu et al. 2011] contemplam simultaneamente as

propriedades **PROP01** e **PROP02** e, portanto, operam independentemente do funcionamento da rede alvo. Estes sistemas são descritos de forma detalhada nas seções 3.2 a 3.6.

3.2 SNTS (Sensor Network Troubleshooting Suite)

O SNTS (*Sensor Network Troubleshooting Suite*) é um sistema de monitoramento passivo para RSSF proposto por [Khan et al. 2007]. No SNTS, nós *sniffers*, denominados pelos autores de *debug nodes*, "ouvem" passivamente o canal de comunicação e coletam os pacotes enviados pelos nós da rede alvo.

A Figura 3.1 mostra os nós da rede alvo (App Mote) e os *sniffers* (Debug Mote). Ao capturar um pacote, o *sniffer* inclui um registro em sua memória não volátil (memória *flash*, por exemplo) com o conteúdo do pacote, e uma marca de tempo (*timestamp*) baseada no seu próprio relógio. Após o período de captura dos dados, os *sniffers* são manualmente recolhidos e os registros dos pacotes capturados são transferidos para um computador, onde serão analisados posteriormente (*offline*).

Figura 3.1 – Rede de monitoramento no SNTS [Khan et al. 2007]

Após os registros serem armazenados no computador, faz-se necessário ajustar o *timestamp* de cada registro, pois os *sniffers* não são sincronizados. Isto é feito da seguinte forma: antes de implantar a rede de monitoramento, cada *sniffer* tem o seu relógio ajustado com o relógio de um nó base (nó sensor conectado ao computador e não utilizado no monitoramento). Ao final da captura dos dados, o relógio de cada *sniffer* é comparado com o

relógio do nó base e os *timestamps* de seus registros são ajustados de acordo com a diferença entre estes relógios.

Após o ajuste do relógio é realizada a junção dos pacotes capturados, onde os registros são ordenados pelo tempo e os registros redundantes são removidos. Dois ou mais registros são considerados redundantes quando a diferença entre seus *timestamps* é inferior a um valor ξ determinado heuristicamente. Vale ressaltar que podem ocorrer registros redundantes quando dois ou mais *sniffers* capturam pacotes de um mesmo nó da rede alvo.

Para analisar os dados após a junção dos pacotes, os autores desenvolveram uma ferramenta, utilizando técnicas de aprendizagem de máquina, para verificar o comportamento e detectar possíveis falhas da rede alvo. Antes de executar a ferramenta é necessário configurar em arquivos texto o formato dos pacotes capturados e regras que definam o comportamento esperado da rede alvo. Por exemplo, se a rede alvo estiver executando uma aplicação em tempo real, pode-se configurar o retardo máximo (diferença entre os *timestamps* dos nós de destino e origem) suportado pela aplicação. Neste caso, a ferramenta mostra as mensagens que por ventura excedem este retardo máximo.

O SNTS ajuda o desenvolvedor de aplicações para RSSF a encontrar e resolver falhas em tempo de desenvolvimento. No entanto, torna-se inviável utilizar o SNTS para monitorar RSSF implantadas em cenários reais ("*in situ*") em que seja impraticável recolher os *sniffers*, como por exemplo, as aplicações militares ou aplicações para monitoramento ambiental (e.g., florestas, oceanos, etc.) descritas na seção 2.1.4.

3.3 SNIF (Sensor Network Inspection Framework)

Em [Ringwald and Romer 2007], os autores propõem um *framework* de inspeção passiva denominado SNIF (*Sensor Network Inspection Framework*). No SNIF, uma rede de monitoramento sem fio, denominada pelos autores de *Deployment Support Network* (DSN), é implantada juntamente com a rede alvo. Cada nó DSN possui duas interfaces de rádio, sendo uma usada para capturar passivamente os pacotes enviados pelos nós da rede alvo e outra usada para enviar os pacotes capturados para o nó sorvedouro (e.g., um computador) através da rede DSN, conforme mostrado na Figura 3.2.

Os pacotes capturados pelos nós DSN são marcados com um *timestamp* e encaminhadas até o sorvedouro onde são ordenados pelo *timestamp* e os pacotes redundantes

são removidos. Em seguida, os pacotes são decodificados de acordo com a descrição dos seus campos definida em um arquivo parametrizável. Após a decodificação, os pacotes são analisados utilizando uma árvore de decisão para inferir o status dos nós da rede alvo e encontrar possíveis falhas nesta rede. Portanto, os pacotes capturados são analisados de forma *online*.

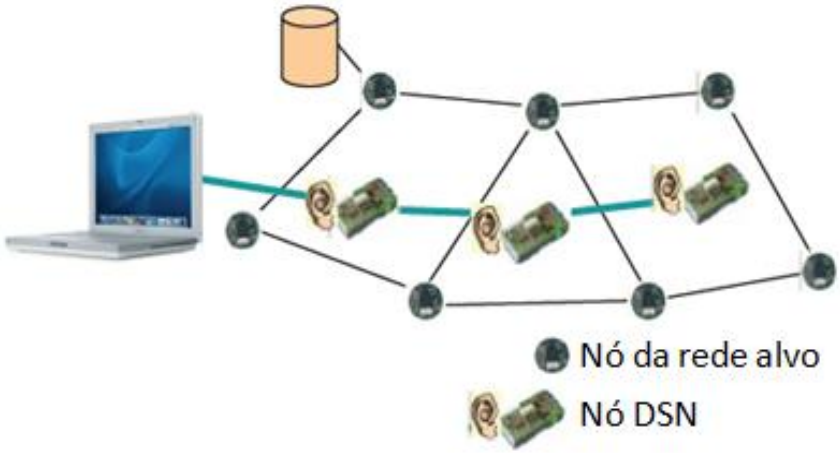


Figura 3.2 - Sistema de monitoramento SNIF (Adaptado de [Ringwald and Romer 2007])

Finalmente, as informações obtidas a partir do monitoramento da rede alvo são mostradas em uma interface gráfica desenvolvida pelos autores. A Figura 3.3 mostra a interface de usuário do SNIF, onde as cores indicam o estado dos nós da rede alvo (verde: ok, amarelo: problema não grave, vermelho: problema grave). Ao selecionar um nó, a interface exibe suas métricas.

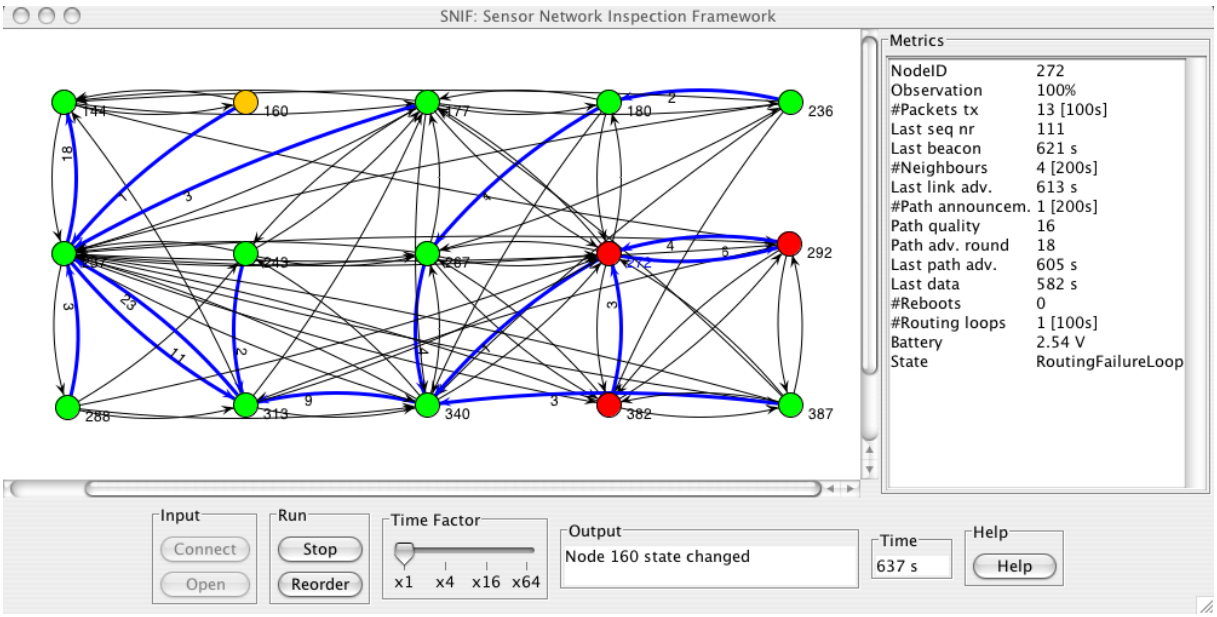


Figura 3.3 – Interface de usuário do SNIF [Ringwald and Romer 2007]

O SNIF pode ser utilizado pelo administrador da rede para detectar eventuais falhas nos nós da rede alvo, como também para disponibilizar informações sobre os *links* e sobre os *paths* da rede alvo. Entretanto, o SNIF não possui nenhum mecanismo de sincronização dos relógios dos nós DSN, podendo então ocasionar erros na ordenação dos pacotes capturados e na remoção de pacotes redundantes, pois pacotes capturados no mesmo instante por diferentes nós DSN podem ter *timestamps* distintos.

3.4 Pimoto

O Pimoto é um sistema de monitoramento passivo distribuído para RSSF proposto por [Awad et al. 2008]. A Figura 3.4 mostra a estrutura hierárquica do Pimoto.

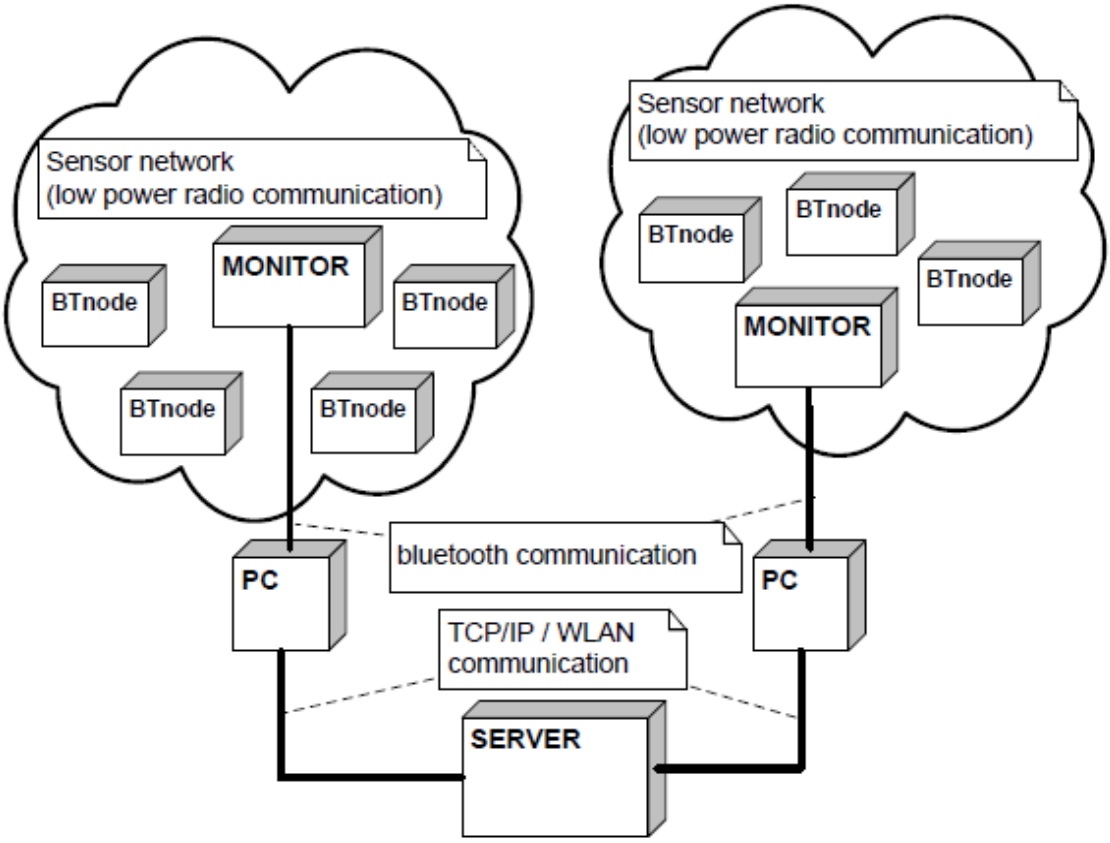


Figura 3.4 - Sistema de monitoramento Pimoto [Awad et al. 2008]

No Pimoto, a rede alvo é subdividida em "ilhas de monitoramento". Em cada ilha de monitoramento é implantado um *sniffer*, denominado pelos autores de monitor, que é responsável por capturar em modo promíscuo os pacotes enviados pelos nós da sua ilha e enviar os pacotes capturados diretamente para um gateway (computador) através de um rádio *Bluetooth*. Desta forma, o Pimoto não captura pacotes redundantes, pois os pacotes enviados

por um determinado nó da rede alvo são capturados apenas pelo *sniffer* da sua ilha. O *sniffer* funciona como um tipo de *cluster-head* em uma rede hierárquica (vide seção 2.1.3).

O mesmo gateway pode receber os pacotes capturados de vários *sniffers*. O gateway inclui em cada pacote capturado o *timestamp* e o endereço do *sniffer* e, em seguida, envia os pacotes capturados para um servidor central utilizando uma rede TCP/IP sem fio ou cabeada. O servidor decodifica e mostra os pacotes capturados na ferramenta Wireshark[2] [Wireshark 2013] utilizando um *plugin* desenvolvido pelos autores, conforme mostrado na Figura 3.5.

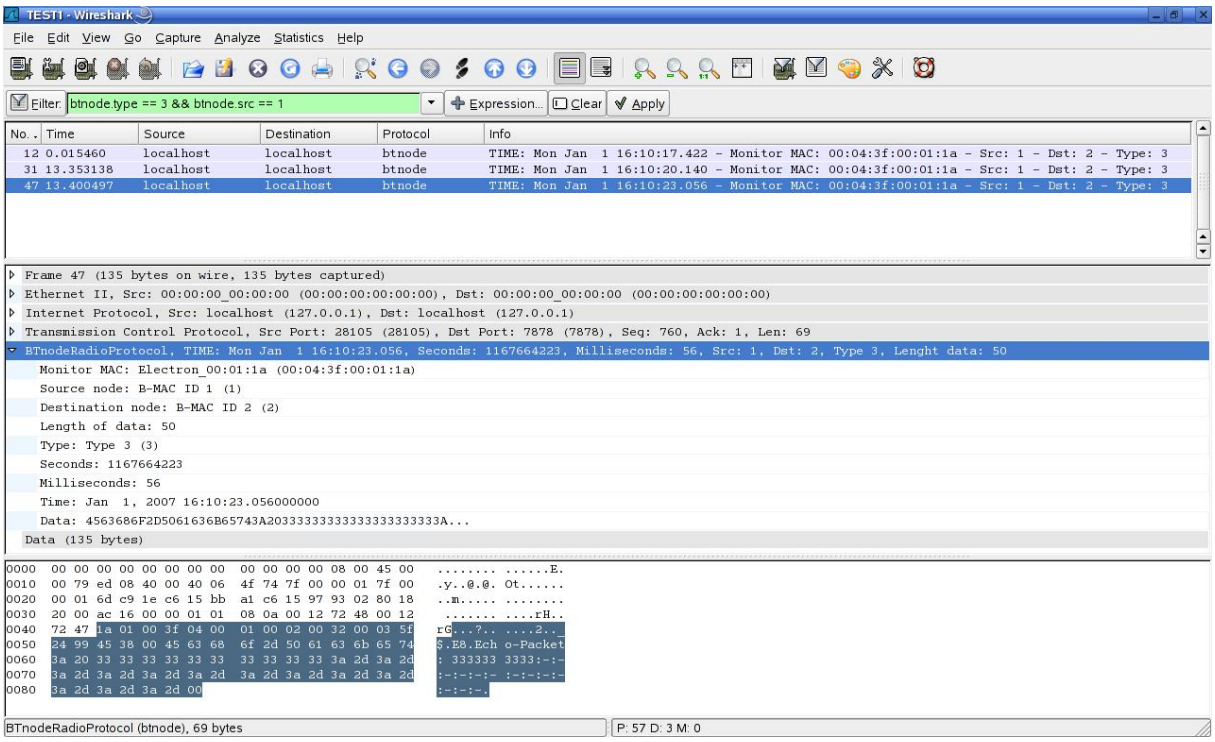


Figura 3.5 – Tela do Wireshark exibindo um pacote da rede alvo [Awad et al. 2008]

No Pimoto, diferentemente dos outros sistemas de monitoramento abordados neste capítulo, os *sniffers* não capturam nem transmitem pacotes redundantes, contribuindo assim para a redução do consumo de energia da rede de monitoramento. Por outro lado, o Pimoto não possui nenhum mecanismo para analisar e inferir o comportamento da rede alvo. Os pacotes capturados podem ser apenas visualizados no Wireshark e, portanto toda a análise dos pacotes deve ser realizada pelo administrador da rede. Além disso, a utilização do Pimoto pode ser inviável para RSSF com muitos nós distribuídos em uma área geográfica grande, pois é necessária uma infraestrutura composta por vários gateways interligados ao servidor.

[2] Wireshark é uma ferramenta utilizada para análise de tráfego de rede muito utilizada pelos gerentes de rede, como também pela comunidade acadêmica.

3.5 LiveNet

Em [Chen et al, 2008], os autores propõem o LiveNet, um conjunto de ferramentas e técnicas para registrar o comportamento de uma RSSF implantada em um cenário real. O LiveNet é constituído por três componentes principais: uma infraestrutura de monitoramento passiva composta por nós *sniffers* que coletam e armazenam os pacotes de dados enviados pelos nós da rede alvo; um processo de *merging* que agrupa os pacotes coletados em um único *trace*; e um conjunto de algoritmos para analisar o *trace*. Enquanto a captura dos pacotes é realizada em tempo real, o *merging* e a análise do *trace* são realizados de forma *offline*.

No LiveNet, os pacotes capturados pelos *sniffers* podem ser armazenados em uma memória *flash* ou enviados para um computador através da porta serial para futuras análises. Além disso, os *sniffers* são sincronizados apenas durante a implantação da rede de monitoramento, podendo então ocasionar erros no processo de *merging* e, consequentemente, na fase de análise dos dados.

A análise dos dados utiliza o *trace* gerado pelo processo de *merging* para reconstruir o comportamento da rede alvo. As informações que podem ser obtidas são: topologia da rede, caminhos de roteamento, largura de banda utilizada, tráfego gerado por cada nó da rede, descoberta de "*hotspots*" (nós que enviam e/ou recebem muito mais pacotes do que os outros nós) e causas de eventuais perdas de pacotes.

O LiveNet tem como principal objetivo descobrir a topologia e obter informações sobre o comportamento da rede alvo, enquanto que os sistemas de monitoramento passivo discutidos nas seções anteriores (SNTS, SNIF e Pimoto) preocupam-se principalmente em detectar as falhas na RSSF que está sendo monitorada. No entanto, torna-se inviável utilizar o LiveNet para monitorar RSSF implantadas em cenários em que seja impraticável recolher os dados armazenados nas memórias *flash* dos *sniffers* ou enviar os dados coletados pelos *sniffers* através de uma rede cabeada, como por exemplo as aplicações militares ou aplicações para monitoramento ambiental descritas na seção 2.1.4.

3.6 PMSW (Passive Monitoring System in Wireless Sensor Networks)

O PMSW **(***a Passive Monitoring System in Wireless sensor networks*), proposto por [Xu et al. 2011], tem como principal objetivo analisar os pacotes capturados em uma RSSF através de monitoramento passivo.

A Figura 3.6 mostra o funcionamento do sistema de monitoramento PWSW. No PMSW, nós *sniffers* (*sniffing node)* são implantados na área de monitoramento juntamente com os nós sensores (*sensor node*) da rede alvo. Cada *sniffer* captura em modo promíscuo os pacotes de dados e ACK (pacote de confirmação de recebimento) dos nós da rede alvo que estão na sua área de cobertura e envia os pacotes capturados para o seu *gateway* (*monitoring node*) através da rede de monitoramento utilizando roteamento em múltiplos saltos. Um *sniffer* conecta-se a apenas um *gateway* através de um canal de rádio com frequência diferente da frequência de operação dos nós da rede alvo. Em alguns cenários pode ser necessário implantar *gateways* em diferentes partes da rede, pois o alcance do rádio dos *sniffers* é limitado.

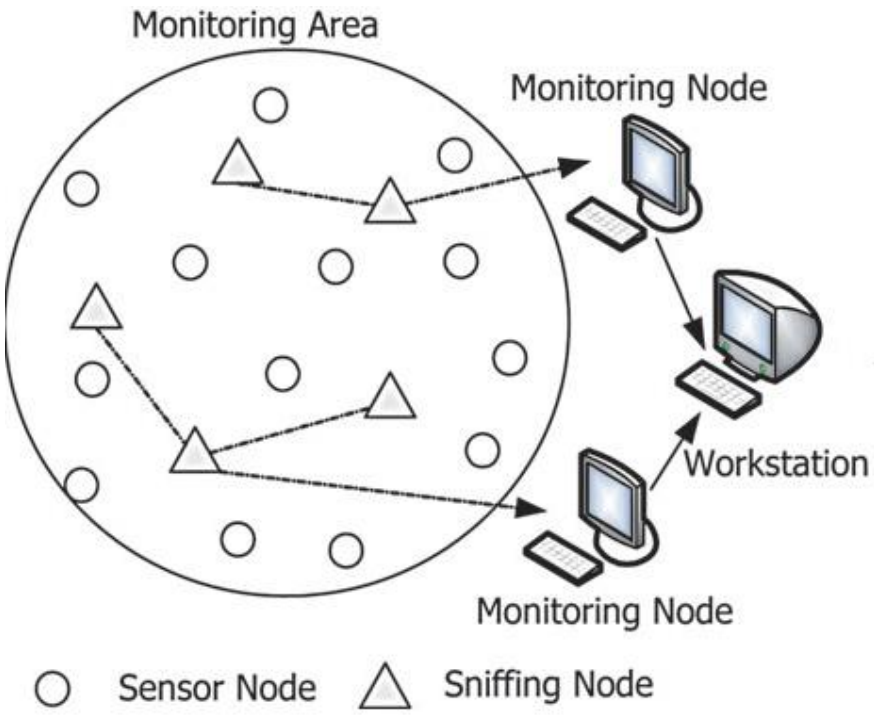


Figura 3.6 – Funcionamento do Sistema de monitoramento PMSW [Xu et al. 2011]

Ao receber os pacotes capturados por seus *sniffers*, o *gateway* cria um arquivo de *trace* local. Cada registro deste *trace* contém as informações de um pacote e um *timestamp* baseado

no relógio do *gateway*. Em seguida, cada *gateway* envia o *trace* gerado para um servidor (Workstation*)* através de uma rede TCP/IP.

O servidor recebe os *traces* gerados por todos os *gateways*, e faz a junção dos *traces* recebidos, gerando assim um único *trace* global. Durante o algoritmo de junção, faz-se necessário ajustar o *timestamp* de cada registro, pois os *gateways* não são sincronizados. Isto é feito da seguinte forma: um dos *gateways* é selecionado como o nó de referência de tempo. Em seguida, o relógio de cada *gateway* é comparado com o relógio do nó de referência e os *timestamps* de seus registros são ajustados de acordo com a diferença entre estes relógios. Após o ajuste dos *timestamps*, os registros são ordenados pelo tempo e os registros redundantes são removidos. Vale ressaltar que podem ocorrer registros redundantes quando dois ou mais *sniffers* capturam pacotes de um mesmo nó sensor da rede alvo.

Após a junção dos *traces*, o servidor executa um algoritmo de inferência, utilizando uma máquina de estados finitos, que tem como objetivos inferir pacotes não capturados pelos *sniffers* e verificar se os pacotes de dados transmitidos foram recebidos pelos respectivos destinos.

Em seguida, é realizada a análise do *trace* com o intuito de avaliar o desempenho e detectar eventuais falhas da rede alvo. No PMSW, as seguintes informações podem ser obtidas a partir desta análise: morte de nós, reinicialização de nós, nós isolados (nós que não possuem outros nós na sua área de cobertura), laços de roteamento, enlaces com alta perda de pacotes, enlaces com alta latência, topologia da rede, entre outras. Para a visualização destas informações, os autores desenvolveram uma ferramenta web.

Diferentemente dos demais trabalhos analisados, o PMSW implementa um algoritmo para inferir pacotes não capturados pela rede de monitoramento, gerando assim um *trace* com mais pacotes e, consequentemente, obtendo informações mais precisas sobre o funcionamento da rede alvo. No entanto, são capturados apenas pacotes de dados e de confirmação (ACK), enquanto que os demais pacotes de controle, tais como pacotes de roteamento e de eleição de cluster, não são capturados nem analisados. Além disso, o ajuste do relógio dos *gateways* não é suficiente para garantir a remoção de todos os pacotes redundantes devido à latência da rede de monitoramento, pois dois *sniffers* podem capturar o mesmo pacote e enviar para o *gateway* utilizando rotas distintas, fazendo com que estes pacotes sejam recebidos em momentos distintos.

3.7 Análise comparativa

Nesta seção é realizada uma análise comparativa entre os sistemas de monitoramento passivo descritos nesta tese de doutorado levando-se em consideração as características que foram obtidas a partir da revisão bibliográfica e que são relacionadas a seguir:

- **Eficiência energética -** verifica se o sistema se preocupa em minimizar o consumo de energia dos nós da rede de monitoramento;
- **Modo de análise -** informa se a análise dos pacotes capturados é realizada de modo *online* ou *offline*;
- **Pacotes capturados -** relaciona quais tipos de pacotes (dados e/ou controle) são capturados pela rede de monitoramento;
- **Análise de eventos -** descreve quais informações de monitoramento são obtidas ao se analisar os pacotes capturados da rede alvo;
- **Pacotes redundantes -** verifica se o sistema de monitoramento captura pacotes redundantes. Pacotes redundantes são capturados quando dois ou mais *sniffers* capturam pacotes de um mesmo nó da rede alvo;
- **Ferramenta de visualização –** descreve o tipo da ferramenta utilizada pelo sistema para a visualização das informações obtidas a partir do monitoramento da rede alvo.

As características *modo de análise* e *análise de eventos* foram definidas em [Xu et al. 2011]. As características *eficiência energética*, *pacotes capturados*, *pacotes redundantes* e *ferramenta de visualização* foram identificadas a partir do estudo dos sistemas de monitoramento passivo abordados neste capítulo. A captura destas características é uma contribuição adicional desta tese de doutorado, pois elas podem ser utilizadas para comparar sistemas de monitoramento passivo propostos para RSSF.

A Tabela 3.3 mostra um quadro comparativo dos sistemas de monitoramento descritos neste capítulo. Observando-se os trabalhos relacionados descritos neste capítulo e a Tabela 3.3 pode-se constatar que:

i. Nenhum dos sistemas é energeticamente eficiente, pois não se preocupam em minimizar o consumo de energia dos nós da rede de monitoramento. A eficiência energética é importante quando se deseja monitorar redes de sensores implantadas em áreas remotas em que seja impraticável a substituição das baterias dos *sniffers* (vide seção 2.1.2).

Tabela 3.3 – Características dos sistemas de monitoramento

Característica	Trabalhos relacionados				
	SNTS	**SNIF**	**Pimoto**	**LiveNet**	**PMSW**
Eficiência Energética	Não	Não	Não	Não	Não
Modo de Análise	Offline	Online	Online	Offline	Online
Pacotes Capturados	Dados + Controle	Dados + Controle	Dados + Controle	Dados	Dados + Controle (ACK)
Análise de Eventos	Análise de falhas	Análise de falhas	Não possui	Não possui	Análise de falhas e desempenho
Pacotes Redundantes	Sim	Sim	Não	Sim	Sim
Ferramenta de Visualização	Desenvolvida pelos autores	Desenvolvida pelos autores	Plugin do Wireshark	Desenvolvida pelos autores	Desenvolvida pelos autores

ii. O SNIF, o Pimoto e o PMSW analisam os pacotes capturados de forma *online*, enquanto que no SNTS e no LiveNet os pacotes capturados são armazenados em memórias não voláteis, que devem ser recolhidas manualmente, para posteriormente serem analisados (*offline*). A análise *online* permite que o administrador da rede possa detectar e reagir às condições de falha da rede alvo de forma mais rápida.

iii. O SNTS, o SNIF e o Pimoto capturam pacotes de dados e de controle (roteamento, eleição de líderes, ACK, etc.) enviados pela rede alvo. O Livenet captura apenas os pacotes de dados, e o PMSW captura pacotes de dados e pacotes de confirmação de recebimento (ACK).

iv. O SNTS, o SNIF e o PMSW analisam os dados capturados com o intuito de detectar eventos de falha ou desempenho da rede alvo enquanto que o Pimoto e o Livenet apenas mostram os *traces* dos pacotes capturados e, portanto, toda a análise dos pacotes deve ser realizada pelo administrador da rede.

v. Apenas o Pimoto não captura pacotes redundantes, pois a rede alvo é subdividida em ilhas de monitoramento e cada ilha possui apenas um *sniffer*, que é responsável por capturar os pacotes enviados pelos nós da sua ilha. No Pimoto, a localização dos *sniffers* é determinada manualmente pelo administrador durante a implantação da rede de modo que todos os nós da ilha de monitoramento estejam na área de cobertura do seu *sniffer*. A

captura de pacotes redundantes aumenta o consumo de energia da rede de monitoramento, pois os *sniffers* transmitem pacotes redundantes através desta rede.

vi. Apenas o Pimoto exibe as informações de monitoramento em uma ferramenta de gerência utilizada pela comunidade de gerência de rede (i.e., Wireshark).

Conforme mencionado no capítulo 1, esta tese propõe um sistema de monitoramento passivo energeticamente eficiente para RSSF, que prolonga o tempo de vida da rede de monitoramento e, consequentemente, beneficia a rede alvo pelo monitoramento por mais tempo. Além disso, as informações obtidas com o monitoramento são disponibilizadas através de um agente SNMP a fim de integrar o sistema proposto com ferramentas de gerência SNMP e facilitar a administração da rede alvo.

3.8 Considerações finais

Neste capítulo os sistemas de monitoramento passivo para redes de sensores sem fio SNTS [Khan et al. 2007], SNIF [Ringwald and Romer 2007], Pimoto [Awad et al. 2008], LiveNet [Chen et al, 2008] e PMSW [Xu et al. 2011] foram descritos, analisados e comparados. O próximo capítulo aborda o sistema de monitoramento EPMOSt (***E****nergy-efficient* ***P****assive* ***MO****nitoring* ***S****yst****e****m*) proposto nesta tese de doutorado.

4 O SISTEMA DE MONITORAMENTO EPMOSt

Este capítulo descreve o sistema de monitoramento EPMOSt (***E**nergy-efficient **P**assive **MO**nitoring **S**yst**e**m*) proposto nesta tese de doutorado. A seção 4.1 apresenta o sistema EPMOSt. Na seção 4.2, os detalhes de especificação do sistema proposto são abordados. Por fim, na seção 4.3 são apresentadas as considerações finais deste capítulo.

4.1 Descrição do sistema

Conforme mencionado na seção 1.1, um sistema de monitoramento passivo energeticamente eficiente é necessário caso se deseje monitorar continuamente uma RSSF em um cenário real, pois caso contrário a rede de monitoramento pode ter um tempo de vida bem menor do que a rede alvo devido à má utilização da energia dos *sniffers*. Então, nesta tese de doutorado é proposto um sistema de monitoramento passivo para RSSF, denominado EPMOSt, que tem como principal objetivo reduzir o consumo de energia da rede de monitoramento. Além disso, o sistema proposto disponibiliza as informações obtidas com o monitoramento através de um agente SNMP. O agente SNMP permite integrar o sistema proposto com qualquer ferramenta de gerência que suporte o protocolo SNMP, tais como Nagios [Nagios 2013], Net-SNMP [Net-SNMP 2013], *SNMP MIB Browser Android Tool* [MIB Browser 2013] e *ManageEngine MIB Browser Free Tool* [ManageEngine 2014].

A Figura 4.1 mostra a visão geral do sistema de monitoramento EPMOSt, onde uma rede de monitoramento aparece implantada juntamente com a rede alvo. Um nó da rede de monitoramento, denominado de *sniffer*, captura em modo promíscuo os pacotes enviados por um ou mais nós da rede alvo, insere uma marca de tempo (*timestamp*) em cada pacote capturado, agrega os cabeçalhos (*headers*) de vários pacotes em uma mensagem de monitoramento e envia esta mensagem, através da rede de monitoramento, para o monitor local. O monitor local recebe as mensagens de monitoramento de vários *sniffers* e insere as informações dos pacotes capturados em um arquivo de *trace* (banco de dados) localizado no servidor. O servidor, por sua vez, executa uma aplicação que analisa o *trace* gerado por um ou mais monitores locais para extrair diversas informações sobre a rede alvo (tempo em que o nó está ativo, perda de pacotes, morte e reinicialização de nós, quantidade de pacotes enviados e recebidos por cada nó, entre outras). Estas informações são disponibilizadas para o

administrador da rede e são também armazenadas em uma MIB para serem acessadas por um agente SNMP.

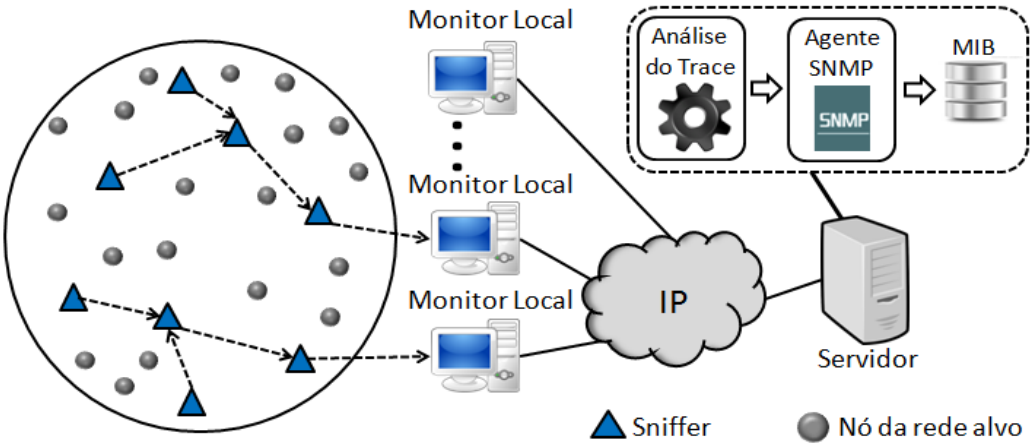


Figura 4.1 – Visão geral do sistema de monitoramento EPMOSt

A Figura 4.2 mostra o diagrama de atividades UML (*Unified Modeling Language*) do EPMOSt, no qual os pacotes de um determinado nó da rede alvo são capturados por apenas um *sniffer* com o intuito de reduzir a transmissão de pacotes redundantes e, consequentemente, reduzir o consumo de energia da rede de monitoramento.

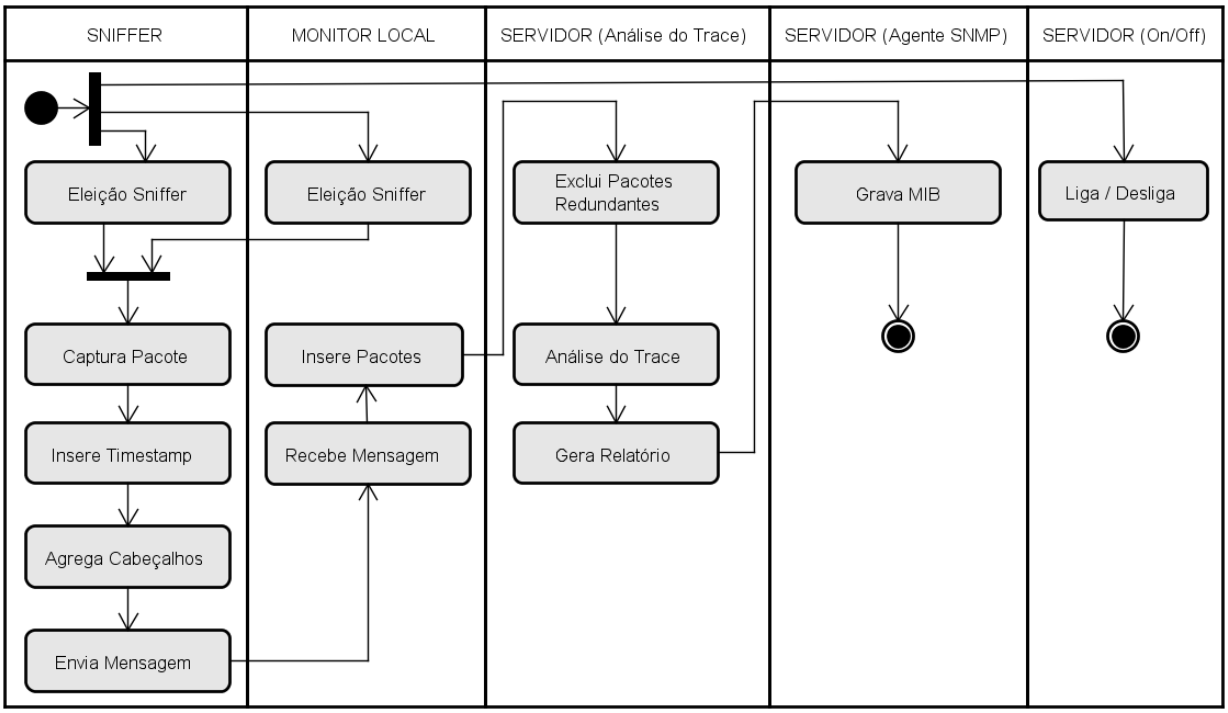


Figura 4.2 – Diagrama de atividades do EPMOSt

No EPMOSt, inicialmente é executado um mecanismo (**Eleição Sniffer**) para eleger quais nós da rede alvo terão seus pacotes capturados por quais *sniffers*. Este mecanismo de eleição é realizado pelos *sniffers* e pelo monitor local levando-se em consideração o RSSI

(*Received Signal Strength Indicator*), que indica o nível de potência do sinal recebido. Este mecanismo de eleição é explicado de maneira detalhada na seção 4.2.1.1.

Ao **capturar um pacote** da rede alvo, o *sniffer* **insere um *timestamp*** neste pacote. Após capturar alguns pacotes, o *sniffer* pode utilizar um mecanismo para **agregar os cabeçalhos** destes pacotes em uma mensagem de monitoramento para **enviar** para o monitor local. A agregação dos cabeçalhos tem como objetivo reduzir a quantidade de dados enviados pela rede de monitoramento e, consequentemente, reduzir o consumo de energia desta rede. Neste caso, apenas as informações presentes nos cabeçalhos dos pacotes enviados pela rede alvo serão monitoradas. Entretanto, caso se deseje monitorar também os dados enviados pela rede alvo, basta não utilizar este módulo de agregação. O mecanismo de agregação de cabeçalhos é explicado de maneira detalhada na seção 4.2.1.2.

O monitor local **recebe as mensagens** de monitoramento enviadas pelos *sniffers* e **insere os pacotes** capturados em um arquivo de *trace* (banco de dados) localizado no servidor. A comunicação entre o monitor local e o servidor é realizada através de uma rede IP. Em alguns cenários poderá ser necessário implantar monitores locais em diferentes partes da rede devido ao alcance limitado do rádio dos *sniffers*. Em um cenário mais restrito pode-se utilizar apenas um computador para desempenhar as funções de monitor local e de servidor.

O servidor tem como principal função extrair informações sobre a rede alvo a partir do *trace* gerado pelos monitores locais. Para tanto, inicialmente, o servidor **exclui os pacotes redundantes** que foram inseridos no *trace*. Vale ressaltar que podem existir pacotes redundantes no *trace* quando dois ou mais *sniffers* capturam pacotes de um mesmo nó sensor da rede alvo. O algoritmo utilizado para exclusão dos pacotes redundantes é explicado de maneira detalhada na seção 4.2.3.

Em seguida, o servidor executa a **análise do trace** para obter informações sobre a rede alvo. Estas informações são utilizadas para **gerar um relatório** que será exibido para o administrador da rede e são também **gravadas em uma MIB** por um **agente SNMP**. Desta forma, qualquer ferramenta de gerência que utilize o protocolo SNMP pode se comunicar com o agente SNMP e exibir as informações obtidas a partir do monitoramento da rede alvo. A seção 4.2.3 descreve as informações obtidas a partir da análise do trace, e a seção 4.2.4 descreve a MIB especificada para o EPMOSt e o agente SNMP desenvolvido neste trabalho.

O EPMOSt também possui um mecanismo para **ligar e desligar** (on/off) o monitoramento com o intuito de reduzir o consumo de energia dos *sniffers*. Para tanto, o

usuário pode programar no servidor os períodos de tempo em que a rede alvo será monitorada. O servidor então envia um comando para cada monitor local para ligar ou desligar o monitoramento. O monitor local, por sua vez, envia este comando para os *sniffers*. Este mecanismo será explicado de maneira detalhada na seção 4.2.5.

4.2 Especificação do sistema

Esta seção descreve os detalhes de especificação de cada um dos módulos do sistema de monitoramento EPMOSt apresentados na seção 4.1.

4.2.1 Sniffer

Neste trabalho, a rede de monitoramento utiliza como *sniffers* nós da plataforma MicaZ (vide Figura 4.3), desenvolvida pela *Crossbow Technology*. Esta plataforma foi escolhida por ser muito utilizada pela comunidade acadêmica e por ser utilizada em outros trabalhos de RSSF ([Rocha et al. 2012], [Cavalcante et al. 2012]) do grupo de pesquisa GREat (Grupo de Redes de Computadores, Engenharia de Software e Sistemas), ao qual este trabalho está vinculado. Além disso, o laboratório do GREat já dispõe de kits de desenvolvimento da plataforma MicaZ, os quais possibilitam a realização de experimentos utilizando *testbed*.

Figura 4.3 – Nó sensor MicaZ

A aplicação de monitoramento embarcada nos *sniffers* foi desenvolvida utilizando a linguagem de programação nesC (*network embedded system* C) [Gay et al. 2003] e executa sobre o sistema operacional TinyOS [Hill et al. 2000].

4.2.1.1 Mecanismo de eleição

Após a implantação da rede de monitoramento, os *sniffers* e o monitor local iniciam o mecanismo **Eleição Sniffer** para eleger quais nós da rede alvo terão seus pacotes capturados por quais *sniffers*. Este mecanismo é executado quando um *sniffer* captura pela primeira vez

um pacote de um determinado nó da rede alvo e leva em consideração o nível de potência do sinal recebido (i.e., RSSI). É importante ressaltar que o nó da rede alvo envia seu endereço (**id**) em cada pacote transmitido.

Quando um *sniffer* $\mathbf{S_X}$ captura pela primeira vez um pacote de um nó **A** da rede alvo, ele envia uma mensagem de inclusão de um novo nó para o monitor local informando o endereço deste nó (**A**) e o RSSI correspondente. Caso nenhum outro *sniffer* esteja capturando pacotes do nó **A**, o monitor local envia uma mensagem para $\mathbf{S_X}$ iniciar a captura dos pacotes enviados por **A**. O Sniffer $\mathbf{S_X}$ envia então uma mensagem de confirmação (ACK) para o monitor local e inicia a captura dos pacotes enviados pelo nó **A**. A Figura 4.4 mostra o diagrama de sequências UML para este caso.

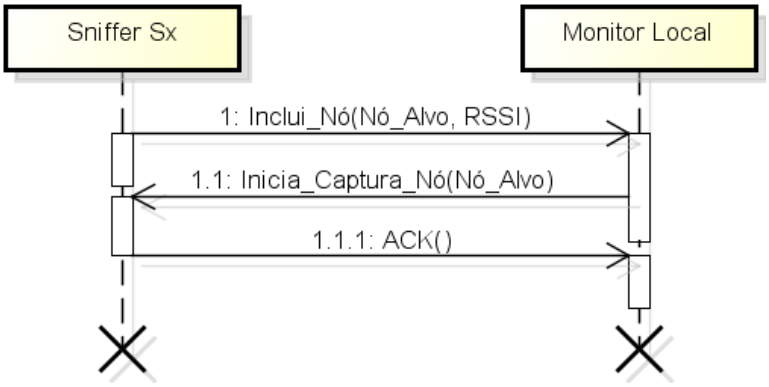


Figura 4.4 – Eleição quando ainda nenhum *sniffer* está capturando os pacotes do nó alvo

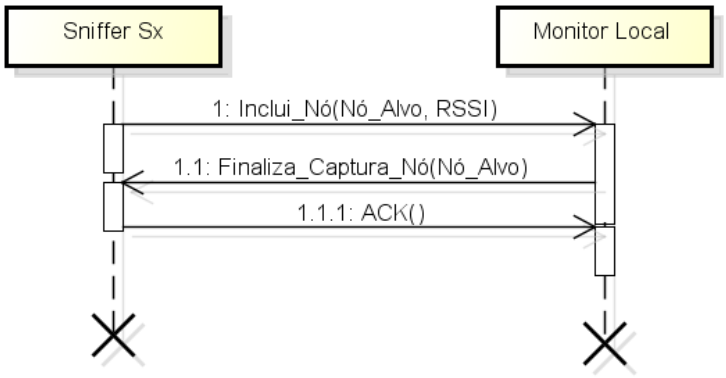


Figura 4.5 – Eleição quando S_Y está capturando os pacotes do nó alvo e RSSI(S_Y) ≥ RSSI(S_X)

No entanto, se já houver outro *sniffer* $\mathbf{S_Y}$ capturando pacotes do nó **A**, o monitor local analisa qual dos dois *sniffers* está recebendo os pacotes de **A** com maior RSSI, pois, em geral, quanto maior o valor do RSSI, melhor é a qualidade do sinal. Caso $\mathbf{S_Y}$ esteja recebendo o sinal de **A** com RSSI maior ou igual do que $\mathbf{S_X}$, o monitor local envia uma mensagem para $\mathbf{S_X}$ informando que ele não deve capturar os pacotes de **A**, conforme pode ser observado na

Figura 4.5. Porém, se $\mathbf{S_Y}$ estiver recebendo o sinal de **A** com RSSI menor do que $\mathbf{S_X}$, o monitor local envia uma mensagem para $\mathbf{S_X}$ capturar os pacotes de **A** e envia uma mensagem para $\mathbf{S_Y}$ parar de capturar os pacotes de **A**, conforme mostrado na Figura 4.6.

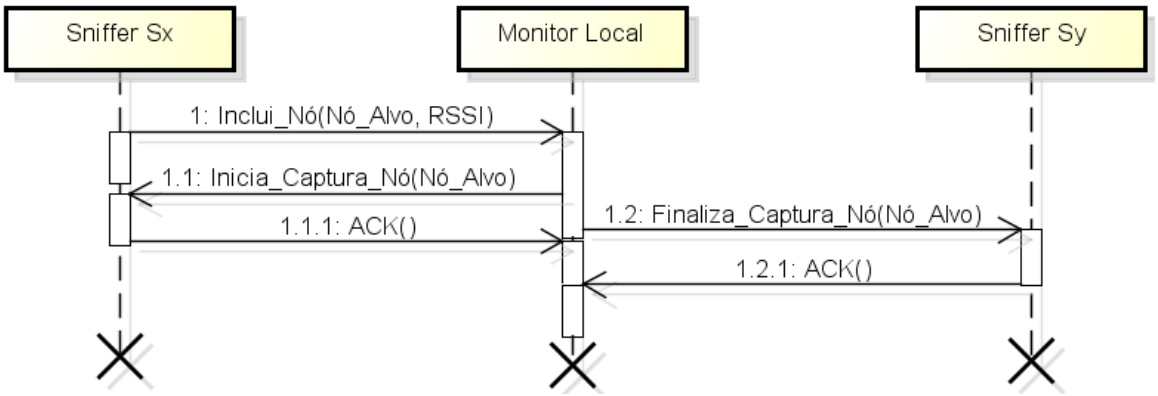


Figura 4.6 – Eleição quando S_Y está capturando os pacotes do nó alvo e RSSI(S_Y) < RSSI(S_X)

Em todas as situações apresentadas nos diagramas de sequência das Figuras 4.4, 4.5 e 4.6, após enviar uma mensagem para um *sniffer*, o monitor local aguarda uma mensagem de confirmação (ACK) deste *sniffer*. Caso a mensagem de confirmação não tenha sido recebida após um intervalo de tempo (*timeout*) pré-definido, o monitor local retransmite a mensagem. Se após três tentativas de retransmissão, o monitor local não receber nenhuma mensagem de confirmação, o *sniffer* é considerado inoperante e o mecanismo de eleição é reinicializado para que os pacotes dos nós da rede alvo cobertos por este *sniffer* possam ser capturados por outro(s) *sniffer*(*s*).

É importante ressaltar que este mecanismo de eleição não trata a eventual morte de *sniffers* após o término da eleição. Portanto, se um *sniffer* apresentar alguma falha de funcionamento ou esgotar a sua energia após a eleição, os nós da rede alvo monitorados por este *sniffer* não terão mais seus pacotes capturados até que o mecanismo de eleição seja reinicializado pelo gerente da rede.

É importante destacar que o RSSI mede a intensidade do sinal na antena do receptor durante a recepção do pacote. Dois aspectos devem ser levados em consideração quando o valor do RSSI é analisado: (i) o RSSI é medido apenas nos pacotes que foram recebidos corretamente, ou seja, o RSSI não é medido nos pacotes que sofreram interferências [Borges et al. 2011]; (ii) o RSSI representa a média da intensidade do sinal durante a recepção do preâmbulo e do cabeçalho da camada física, e não de todos os bytes que compõem o pacote [IEEE 2007]. Apesar destas considerações, o RSSI é uma métrica muito utilizada para avaliar a qualidade do sinal recebido [Vlavianos et al. 2008].

O mecanismo de eleição proposto garante que durante a maior parte do tempo apenas um *sniffer* captura os pacotes enviados por um determinado nó da rede alvo, reduzindo assim a transmissão de pacotes capturados redundantes através da rede de monitoramento. É possível capturar pacotes redundantes apenas quando ocorrer a situação apresentada no diagrama da Figura 4.6, devido ao intervalo de tempo entre as mensagens 1.1 e 1.2. Conforme mostrado nos resultados obtidos a partir dos experimentos (capítulo 5), o mecanismo de eleição proposto reduz consideravelmente o consumo de energia da rede de monitoramento.

4.2.1.2 Processo de monitoramento

Após a execução do mecanismo de eleição explicado na seção 4.2.1.1, os *sniffers* iniciam o processo de monitoramento, onde cada *sniffer* captura em modo promíscuo os pacotes enviados pelos nós da rede alvo que ele monitora, e que foram selecionados pelo mecanismo de eleição. A Figura 4.7 mostra o diagrama de atividades UML do processo de monitoramento.

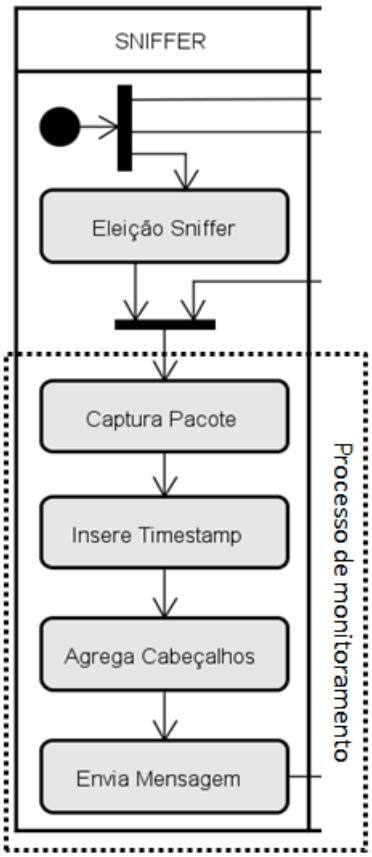


Figura 4.7 – Diagrama de atividades do processo de monitoramento

Conforme mencionado na seção 4.1, a agregação de cabeçalhos é opcional. Por exemplo, a agregação de cabeçalhos não deve ser utilizada quando a rede alvo executa uma

aplicação que captura dados em tempo real (aplicação com restrições de tempo), pois os *sniffers* precisariam aguardar que alguns pacotes fossem capturados para só então enviar as informações presentes nos cabeçalhos destes pacotes para o monitor local.

Quando a agregação de cabeçalhos não é utilizada, o *sniffer* gera, para cada pacote capturado, uma mensagem de monitoramento contendo o seu endereço (*sniffer address*), uma marca de tempo (*timestamp*) e todos os bytes do pacote capturado. Esta mensagem de monitoramento é enviada para o monitor local através da rede de monitoramento utilizando roteamento em múltiplos saltos. O formato do pacote enviado pelo *sniffer* é mostrado na Figura 4.8. O cabeçalho (*header*) é inserido pelo protocolo da camada de enlace da plataforma MicaZ e tem tamanho fixo de cinco bytes, e a área de dados (*payload*) transporta a mensagem de monitoramento. O tamanho máximo do *payload* na plataforma MicaZ é de 29 bytes e, portanto, caso a quantidade de bytes da mensagem de monitoramento seja maior do que este valor, é necessário que o *sniffer* envie mais de um pacote para o monitor local.

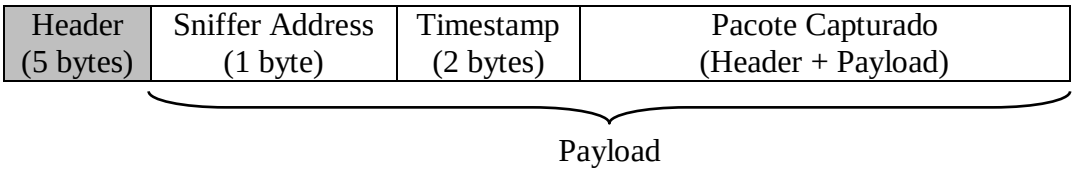


Figura 4.8 – Formato do pacote enviado pelos *sniffers* sem agregação de cabeçalhos

Quando a agregação de cabeçalhos é utilizada, apenas as informações presentes nos cabeçalhos (*headers*) dos pacotes capturados são enviadas pelo *sniffer* para o monitor local. Neste caso, os cabeçalhos de vários pacotes podem ser enviados na mesma mensagem de monitoramento, reduzindo assim o *overhead* de transmissão e, consequentemente, reduzindo o consumo de energia dos *sniffers*. O formato do pacote enviado pelo *sniffer* é mostrado na Figura 4.9. A mensagem de monitoramento transporta o *timestamp* e o *header* de **N** pacotes. O tamanho máximo de **N** depende do tamanho do cabeçalho dos pacotes enviados pela rede alvo, pois neste caso a mensagem de monitoramento não pode ser maior do que o tamanho máximo do *payload* na plataforma MicaZ (29 bytes).

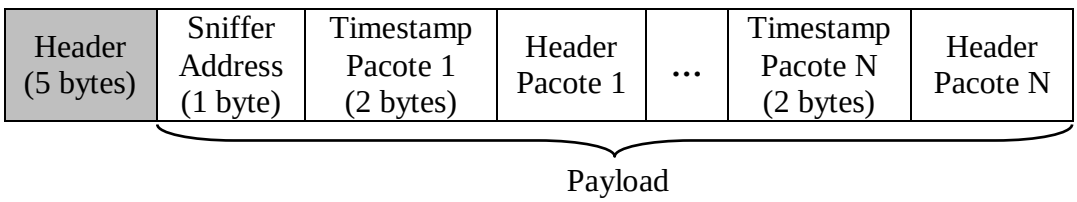


Figura 4.9 – Formato do pacote enviado pelos *sniffers* com agregação de cabeçalhos

4.2.2 Monitor Local

Conforme explicado na seção 4.2.1.1, o monitor local executa o mecanismo de eleição juntamente com os *sniffers*. Além disso, o monitor local recebe as mensagens de monitoramento enviadas pelos *sniffers* contendo os pacotes capturados da rede alvo e insere os pacotes capturados em um arquivo de *trace* (banco de dados) localizado no servidor. Para tanto, a aplicação do monitor local estabelece uma conexão remota com o SGBD (Sistema de Gerenciamento de Banco de dados) instalado no Servidor através de uma rede IP. O servidor de banco de dados "MySQL Server 5.5" [MySQL 2013] foi utilizado neste trabalho por ser um SGBD bastante difundido e ser um software livre com licença GPL (*General Public License*).

A Figura 4.10 mostra o conteúdo, no formato hexadecimal, de alguns pacotes da rede alvo que foram recebidos pelo monitor local e inseridos no banco de dados do servidor. Ao armazenar os pacotes capturados em um banco de dados, pode-se facilmente obter informações sobre a rede alvo utilizando cláusulas SQL (*Structured Query Language*) [SQL 2014]. Para exemplificar, a Figura 4.11 exibe alguns pacotes enviados pelo nó da rede alvo cujo endereço (*source_addr*) é 10, quando a agregação de cabeçalhos não é utilizada. As colunas *data_packet* e *all_packet* contém, respectivamente, os bytes da área de dados (*payload*) e os bytes de todo o pacote no formato hexadecimal.

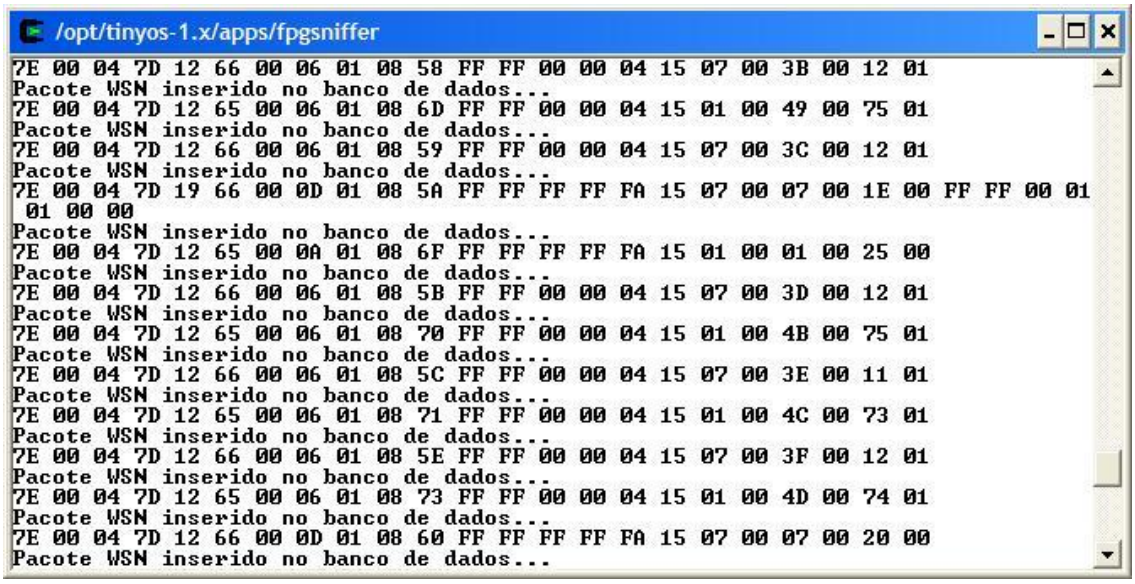
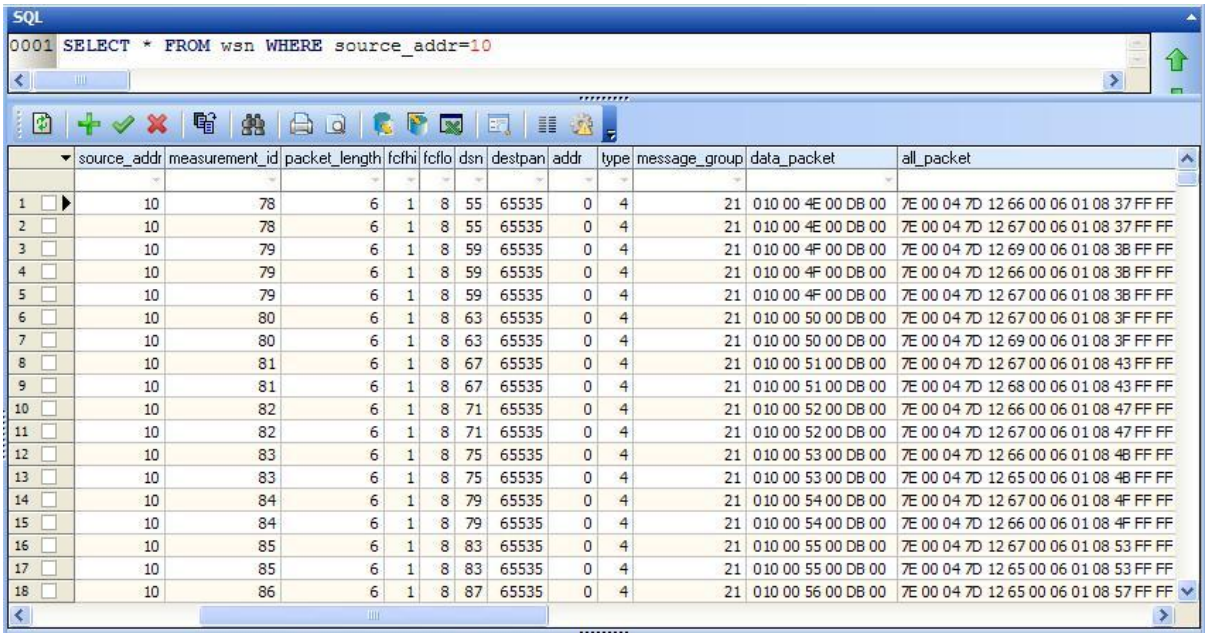


Figura 4.10 – Exibição de pacotes da rede alvo recebidos pelo monitor local

A aplicação do monitor local foi implementada utilizando a linguagem de programação Java, por ser uma tecnologia multiplataforma e possibilitar que o mesmo código execute em diferentes sistemas operacionais (Linux, Windows, etc.).



	source_addr	measurement_id	packet_length	fcfhi	fcflo	dsn	destpan	addr	type	message_group	data_packet	all_packet
1	10	78	6	1	8	55	65535	0	4	21	010 00 4E 00 DB 00	7E 00 04 7D 12 66 00 06 01 08 37 FF FF
2	10	78	6	1	8	55	65535	0	4	21	010 00 4E 00 DB 00	7E 00 04 7D 12 67 00 06 01 08 37 FF FF
3	10	79	6	1	8	59	65535	0	4	21	010 00 4F 00 DB 00	7E 00 04 7D 12 69 00 06 01 08 3B FF FF
4	10	79	6	1	8	59	65535	0	4	21	010 00 4F 00 DB 00	7E 00 04 7D 12 66 00 06 01 08 3B FF FF
5	10	79	6	1	8	59	65535	0	4	21	010 00 4F 00 DB 00	7E 00 04 7D 12 67 00 06 01 08 3B FF FF
6	10	80	6	1	8	63	65535	0	4	21	010 00 50 00 DB 00	7E 00 04 7D 12 67 00 06 01 08 3F FF FF
7	10	80	6	1	8	63	65535	0	4	21	010 00 50 00 DB 00	7E 00 04 7D 12 69 00 06 01 08 3F FF FF
8	10	81	6	1	8	67	65535	0	4	21	010 00 51 00 DB 00	7E 00 04 7D 12 67 00 06 01 08 43 FF FF
9	10	81	6	1	8	67	65535	0	4	21	010 00 51 00 DB 00	7E 00 04 7D 12 68 00 06 01 08 43 FF FF
10	10	82	6	1	8	71	65535	0	4	21	010 00 52 00 DB 00	7E 00 04 7D 12 66 00 06 01 08 47 FF FF
11	10	82	6	1	8	71	65535	0	4	21	010 00 52 00 DB 00	7E 00 04 7D 12 67 00 06 01 08 47 FF FF
12	10	83	6	1	8	75	65535	0	4	21	010 00 53 00 DB 00	7E 00 04 7D 12 66 00 06 01 08 4B FF FF
13	10	83	6	1	8	75	65535	0	4	21	010 00 53 00 DB 00	7E 00 04 7D 12 65 00 06 01 08 4B FF FF
14	10	84	6	1	8	79	65535	0	4	21	010 00 54 00 DB 00	7E 00 04 7D 12 67 00 06 01 08 4F FF FF
15	10	84	6	1	8	79	65535	0	4	21	010 00 54 00 DB 00	7E 00 04 7D 12 66 00 06 01 08 4F FF FF
16	10	85	6	1	8	83	65535	0	4	21	010 00 55 00 DB 00	7E 00 04 7D 12 67 00 06 01 08 53 FF FF
17	10	85	6	1	8	83	65535	0	4	21	010 00 55 00 DB 00	7E 00 04 7D 12 65 00 06 01 08 53 FF FF
18	10	86	6	1	8	87	65535	0	4	21	010 00 56 00 DB 00	7E 00 04 7D 12 65 00 06 01 08 57 FF FF

Figura 4.11 – Exibição de pacotes capturados inseridos no banco de dados

4.2.3 Servidor – Análise do trace

A aplicação *análise trace* executa no servidor (vide Figura 4.1) e foi implementada utilizando também a linguagem de programação Java. A Figura 4.12 mostra a tela inicial desta aplicação. A principal função da aplicação *análise trace* é extrair informações sobre a rede alvo a partir do *trace* gerado pelos monitores locais. Para tanto, inicialmente, os pacotes redundantes que existem no *trace* são excluídos.

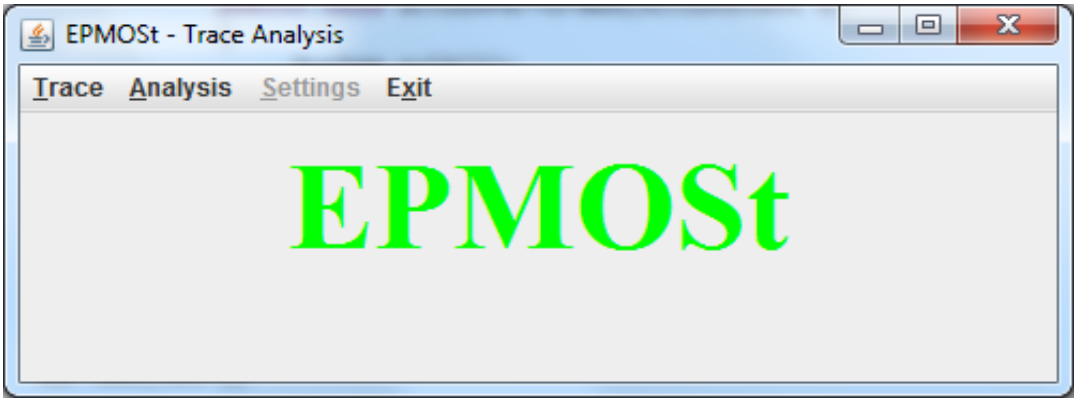
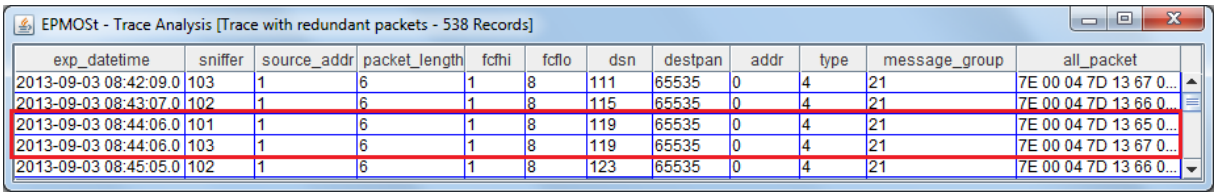


Figura 4.12 – Tela inicial da aplicação *análise trace*

Na plataforma MicaZ, os pacotes redundantes podem ser detectados analisando-se o campo DSN (*Destination Sequence Number*) presente no *header* (cabeçalho) dos pacotes enviados pelos nós sensores. O DSN possui tamanho de oito bits e é incrementado pelo nó de origem a cada pacote enviado. Quando o DSN atinge o valor de 255, o DSN do próximo

pacote enviado pelo nó sensor terá valor zero [Crossbow 2013]. Portanto, se dois ou mais pacotes possuem o mesmo endereço de origem, o mesmo DSN e a diferença entre seus *timestamps* é menor do que Δt, significa que se trata do mesmo pacote. O valor definido para Δt deve ser menor do que o intervalo de tempo decorrido entre dois pacotes enviados por um determinado nó sensor com mesmo DSN. Desta forma, este trabalho utiliza Δt igual a 10 segundos, pois nos experimentos realizados um determinado nó sensor não envia mais do que 255 pacotes neste intervalo de tempo e, portanto, também não envia dois pacotes com mesmo DSN neste intervalo de tempo.

A Figura 4.13 mostra um trecho de um *trace* com os pacotes enviados pelo nó da rede alvo cujo endereço (*source_addr*) é 1. Pode-se observar na Figura 4.13 a existência de dois pacotes enviados pelo nó de endereço 1 com valor de DSN igual a 119 e com o mesmo *timestamp* (*exp_datetime*). Neste caso, um destes pacotes é excluído do *trace*.


EPMOSt - Trace Analysis [Trace with redundant packets - 538 Records]

exp_datetime	sniffer	source_addr	packet_length	fcfhi	fcflo	dsn	destpan	addr	type	message_group	all_packet
2013-09-03 08:42:09.0	103	1	6	1	8	111	65535	0	4	21	7E 00 04 7D 13 67 0...
2013-09-03 08:43:07.0	102	1	6	1	8	115	65535	0	4	21	7E 00 04 7D 13 66 0...
2013-09-03 08:44:06.0	101	1	6	1	8	119	65535	0	4	21	7E 00 04 7D 13 65 0...
2013-09-03 08:44:06.0	103	1	6	1	8	119	65535	0	4	21	7E 00 04 7D 13 67 0...
2013-09-03 08:45:05.0	102	1	6	1	8	123	65535	0	4	21	7E 00 04 7D 13 66 0...

Figura 4.13 – *Trace* com pacotes redundantes

Após a exclusão dos pacotes redundantes, o *trace* é analisado para se extrair diversas informações sobre os nós e sobre os *paths* da rede alvo. A tela da aplicação *análise trace* mostrada na Figura 4.14 disponibiliza as informações sobre os nós de uma rede alvo utilizada para validar a aplicação, quais sejam: *nodeId* (endereço do nó), *timeAwake* (tempo em que o nó está ativo), *lastSeq* (DSN do último pacote enviado pelo nó), *lastTimestamp* (marca de tempo do último pacote enviado pelo nó), *sendPacketNode* (quantidade de pacotes enviados pelo nó), *recvPacketNode* (quantidade de pacotes recebidos pelo nó), *sendDataPacket* (quantidade de pacotes de dados enviados pelo nó), *recvDataPacket* (quantidade de pacotes de dados recebidos pelo nó), *sendBytesNode* (quantidade de bytes enviados pelo nó) e *recvBytesNode* (quantidade de bytes recebidos pelo nó).

Neste caso, conforme pode ser observado na Figura 4.14, apenas o nó com endereço (*nodeId*) **0** recebe pacotes de dados, pois trata-se do nó sorvedouro (*sink*). Os nós com endereços de **1** a **21** enviam pacotes de dados com as informações coletadas dos sensores para o nó sorvedouro. O endereço **65535** corresponde ao endereço de *broadcast* da rede, para onde são enviados todos os pacotes do protocolo de roteamento utilizado pela rede alvo. O cenário,

relativo aos experimentos, que foi utilizado para obter estas informações é descrito de forma detalhada na seção 5.1.1.

EPMOSt - Trace Analysis [Target Nodes - 23 Records]

nodeId	timeAwake	lastSeq	lastTimestamp	sendPacketNode	recvPacketNode	sendDataPacket	recvDataPacket	sendBytesNode	recvBytesNode
0	1778	114	2013-10-07 14:08:45.0	92	694	0	681	3496	11811
1	1779	149	2013-10-07 14:08:50.0	116		33		3660	
2	1778	150	2013-10-07 14:08:44.0	112		31		3554	
3	1778	153	2013-10-07 14:08:36.0	116		33		3704	
4	1776	155	2013-10-07 14:08:34.0	112		32		3552	
5	1775	148	2013-10-07 14:08:33.0	119		32		3752	
6	1777	150	2013-10-07 14:08:41.0	114		29		3606	
7	1778	151	2013-10-07 14:08:41.0	116		35		3660	
8	1778	155	2013-10-07 14:08:39.0	116		33		3638	
9	1778	147	2013-10-07 14:08:37.0	117		33		3708	
10	1777	150	2013-10-07 14:08:38.0	116		30		3704	
11	1777	150	2013-10-07 14:08:41.0	118		34		3725	
12	1777	156	2013-10-07 14:08:45.0	115		34		3622	
13	1778	147	2013-10-07 14:08:40.0	118		33		3692	
14	1777	150	2013-10-07 14:08:35.0	117		33		3654	
15	1778	152	2013-10-07 14:08:38.0	118		33		3780	
16	1778	155	2013-10-07 14:08:49.0	111		32		3470	
17	1778	148	2013-10-07 14:08:43.0	116		32		3660	
18	1795	152	2013-10-07 14:08:52.0	116		32		3660	
19	1778	153	2013-10-07 14:08:49.0	117		30		3720	
20	1794	154	2013-10-07 14:08:52.0	118		34		3626	
21	1794	153	2013-10-07 14:08:52.0	119		33		3806	
65535					1835		0		68638

Figura 4.14 – Informações sobre os nós da rede alvo

A tela da aplicação *análise trace* mostrada na Figura 4.15 disponibiliza as informações sobre alguns *paths* da rede alvo, quais sejam: *pathId* (*path* no formato origem→destino), *srcNode* (endereço do nó de origem), *dstNode* (endereço do nó de destino), *sendPacketPath* (quantidade de pacotes enviados no *path*), *sendBytesPath* (quantidade de bytes enviados no *path*), *timeBeginning* (marca de tempo do primeiro pacote enviado) e *timeEnding* (marca de tempo do último pacote enviado).

EPMOSt - Trace Analysis [Paths of the Target Network - 43 Records]

pathId	srcNode	dstNode	sendPacketPath	sendBytesPath	TimeBeginning	TimeEnding
0 -> 65535	0	65535	92	3496	2013-10-07 13:39:07.0	2013-10-07 14:08:45.0
1 -> 0	1	0	33	572	2013-10-07 13:39:11.0	2013-10-07 14:08:29.0
1 -> 65535	1	65535	83	3088	2013-10-07 13:39:31.0	2013-10-07 14:08:50.0
2 -> 0	2	0	32	624	2013-10-07 13:39:45.0	2013-10-07 14:08:05.0
2 -> 65535	2	65535	80	2930	2013-10-07 13:39:06.0	2013-10-07 14:08:44.0
3 -> 0	3	0	33	572	2013-10-07 13:38:58.0	2013-10-07 14:08:16.0
3 -> 65535	3	65535	83	3132	2013-10-07 13:39:18.0	2013-10-07 14:08:36.0
4 -> 0	4	0	32	534	2013-10-07 13:39:16.0	2013-10-07 14:08:34.0
4 -> 65535	4	65535	80	3018	2013-10-07 13:38:58.0	2013-10-07 14:08:16.0
5 -> 0	5	0	34	566	2013-10-07 13:39:15.0	2013-10-07 14:08:33.0
5 -> 65535	5	65535	85	3186	2013-10-07 13:38:58.0	2013-10-07 14:08:33.0

Figura 4.15 – Informações sobre os *paths* da rede alvo

O gerente da rede pode utilizar as informações sobre os nós e sobre os *paths* para analisar o comportamento da rede alvo. Por exemplo, os valores de *sendPacketNode*, *sendDataPacket* e de *sendBytesNode* podem ser utilizados para determinar o tráfego gerado

pela aplicação que é executada em cada nó sensor. O valor de *lastTimestamp* pode indicar que um determinado nó parou de enviar dados, o que pode significar uma possível morte deste nó.

4.2.4 Servidor – Agente SNMP

Conforme mencionado na seção 4.1, as informações obtidas a partir da análise do *trace* são armazenadas em uma MIB por um agente SNMP. O agente SNMP foi desenvolvido utilizando o *framework* "WebNMS SNMP Agent Toolkit Java Edition" [WebNMS 2013]. Este *framework* foi utilizado porque possibilita o desenvolvimento rápido de agentes SNMP baseados em Java. A subseção 4.2.4.1 descreve a MIB proposta para o sistema de monitoramento EPMOSt e a subseção 4.2.4.2 mostra os testes realizados para validar o funcionamento do agente SNMP desenvolvido neste trabalho.

4.2.4.1 MIB do EPMOSt

Um agente SNMP lê e armazena as informações de gerenciamento em uma MIB [Kurose e Ross 2013]. A MIB é uma estrutura de dados que armazena objetos gerenciados cujos valores, coletivamente, refletem o estado atual dos dispositivos que estão sendo gerenciados. Esses valores podem ser consultados e/ou alterados por uma ferramenta de gerência através do envio de mensagens SNMP ao agente. Na MIB os objetos são nomeados hierarquicamente, de modo que qualquer nó da árvore pode ser identificado pela sequência de nomes (ou números) que especificam o trajeto da raiz até o nó (vide seção 2.2.3.1).

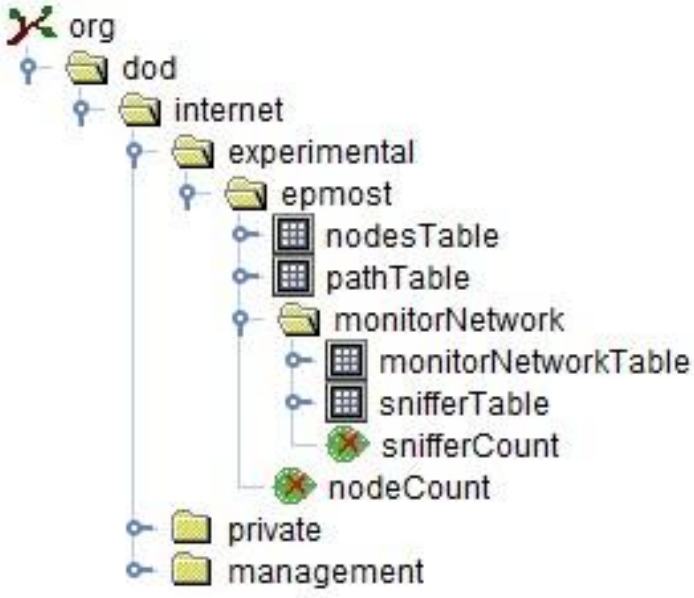


Figura 4.16 – MIB do EPMOSt

A MIB mais utilizada é definida pela RFC 1213 [McCloghrie and Rose 1991]. Conforme mostrado na Figura 4.16, sob o nó *Internet* desta MIB destacam-se as subárvores *management*, *private* e *experimental*. Sob o nó *management* encontram-se as definições dos módulos MIB padronizados pela IETF (*Internet Engineering Task Force*). Sob o nó *private* encontram-se as definições de objetos de empresas registradas na IETF. Sob o nó *experimental* poderão ser nomeados objetos que estão em fase de desenvolvimento e testes. Portanto, a MIB do EPMOSt foi definida sob o nó *experimental*.

A MIB do EPMOSt possui quatro tabelas: *nodesTable*, que armazena informações sobre cada um dos nós da rede alvo; *pathTable*, que armazena informações sobre cada um dos *paths* da rede alvo; *monitorNetworkTable* armazena informações estatísticas sobre a rede de monitoramento; e *snifferTable* para armazenamento de informações sobre cada um dos *sniffers*. Esta MIB possui também dois objetos escalares: *nodeCount*, que representa a quantidade de nós da rede alvo; e *snifferCount*, que representa a quantidade de *sniffers*. A Figura 4.17 mostra as quatro tabelas definidas na MIB do EPMOSt.

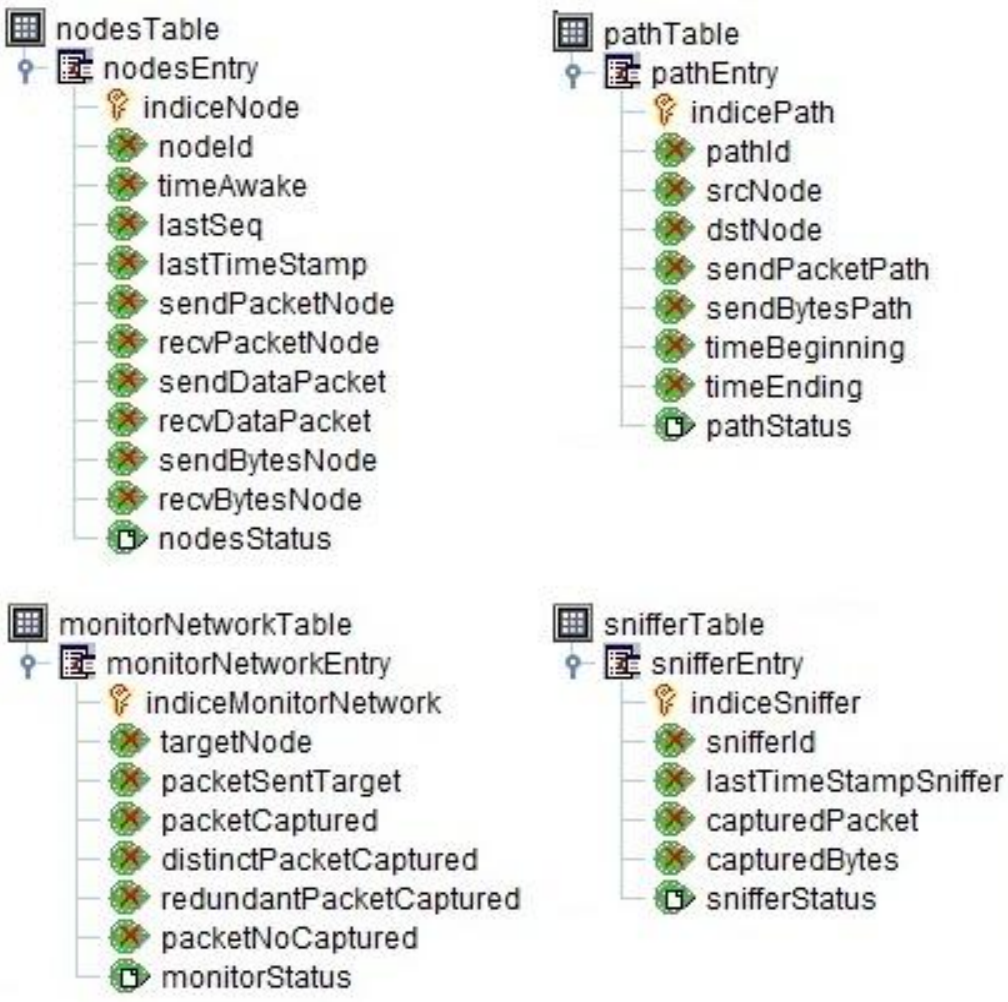


Figura 4.17 – Tabelas da MIB do EPMOSt

Os objetos gerenciados definidos em cada uma das tabelas da MIB do EPMOSt são descritos nas Tabelas 4.1 a 4.4. Os objetos com ***** ao lado do seu nome foram criados a partir do estudo dos sistemas de monitoramento passivo para RSSF abordados no capítulo 3 (SNTS, SNIF, Pimoto, LiveNet e PMSW). A criação destes objetos é uma contribuição adicional deste trabalho. Os demais objetos descritos nas Tabelas 4.1 a 4.4 foram capturados de trabalhos que descrevem MIBs para RSSF: [Jacquot et al. 2009], [Zhang and Li 2009], [Xu et al. 2011] e [Ye et al. 2011].

Tabela 4.1 – Objetos representados na *nodesTable*

Nome do objeto	Descrição do objeto
nodeId	Endereço do nó da rede alvo
timeAwake	Tempo (segundos) em que o nó está ativo
lastSeq	DSN do último pacote enviado pelo nó
lastTimestamp	Marca de tempo do último pacote enviado pelo nó
sendPacketNode	Quantidade de pacotes enviados pelo nó
recvPacketNode	Quantidade de pacotes recebidos pelo nó
sendDataPacket *	Quantidade de pacotes de dados enviados pelo nó
recvDataPacket *	Quantidade de pacotes de dados recebidos pelo nó
sendBytesNode	Quantidade de bytes enviados pelo nó
recvBytesNode	Quantidade de bytes recebidos pelo nó

Tabela 4.2 - Objetos representados na *pathTable*

Nome do objeto	Descrição do objeto
pathId	*Path* no formato origem→destino
srcNode	Endereço do nó de origem do *path*
dstNode	Endereço do nó de destino do *path*
sendPacketPath	Quantidade de pacotes enviados no *path*
sendBytesPath	Quantidade de bytes enviados no *path*
timeBeginning	Marca de tempo do primeiro pacote enviado no *path*
timeEnding	Marca de tempo do último pacote enviado no *path*

Tabela 4.3 - Objetos representados na *snifferTable*

Nome do objeto	Descrição do objeto
snifferId	Endereço do sniffer
lastTimestampSniffer	Marca de tempo do último pacote capturado pelo sniffer
capturedPacket	Quantidade de pacotes capturados pelo sniffer
capturedBytes	Quantidade de bytes capturados pelo sniffer

Tabela 4.4 - Objetos representados na *monitorNetworkTable*

Nome do objeto	Descrição do objeto
targetNode	Endereço do nó da rede alvo
packetSentTarget	Quantidade de pacotes enviados pelo nó da rede alvo
packetCaptured	Quantidade de pacotes do nó da rede alvo capturados
distinctPacketCaptured *	Quant. de pacotes distintos do nó da rede alvo capturados
redundantPacketCaptured *	Quantidade de pacotes do nó da rede alvo redundantes
packetNotCaptured *	Quantidade de pacotes do nó da rede alvo não capturados

4.2.4.2 Teste do Agente SNMP

O agente SNMP é responsável por acessar as informações de monitoramento armazenadas na MIB e repassá-las para a ferramenta de gerência através de mensagens do protocolo SNMP. Desta forma, qualquer ferramenta de gerência que utilize o protocolo SNMP pode se comunicar com o agente e exibir as informações obtidas a partir do monitoramento da rede alvo.

Para testar o agente desenvolvido neste trabalho foram utilizadas as ferramentas de gerência "*SNMP MIB Browser Android Tool*" [MIB Browser 2013] e "*ManageEngine MIB Browser Free Tool"* [ManageEngine 2014]. Estas duas ferramentas foram escolhidas por serem disponibilizadas para o usuário de forma gratuita. A principal funcionalidade destas ferramentas é prover a comunicação com os agentes, através do envio de mensagens do protocolo SNMP, para consultar e/ou alterar os objetos de uma MIB.

A ferramenta "*SNMP MIB Browser Android Tool*" [MIB Browser 2013] foi desenvolvida para executar em dispositivos móveis com sistema operacional *Android*. Desta forma, o gerente da rede pode obter informações sobre o comportamento da RSSF monitorada a partir do seu *smartphone* ou *tablet*. Esta ferramenta foi então instalada em um *smartphone* com sistema operacional *Android*, e foram realizadas consultas em todos os objetos da MIB do EPMOSt. Todas as consultas foram realizadas com sucesso. As Figuras 4.18 a 4.22 exemplificam o funcionamento desta ferramenta de gerência. A Figura 4.18 mostra a estrutura da MIB do EPMOSt.

Ao clicar em *nodesTable* na tela mostrada na Figura 4.18, o usuário visualiza os nós da rede alvo (Figura 4.19a). Ao clicar sobre um determinado nó na tela mostrada na Figura 4.19a, o usuário visualiza as informações referentes a este nó (Figura 4.19b).

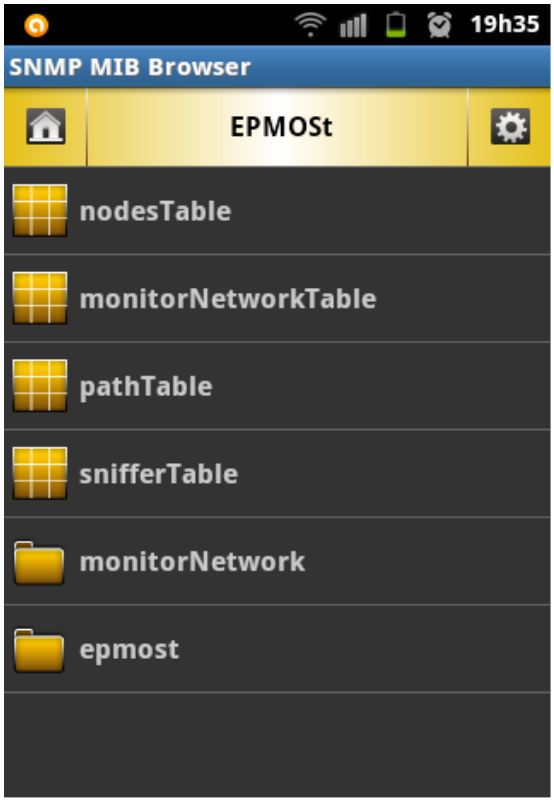
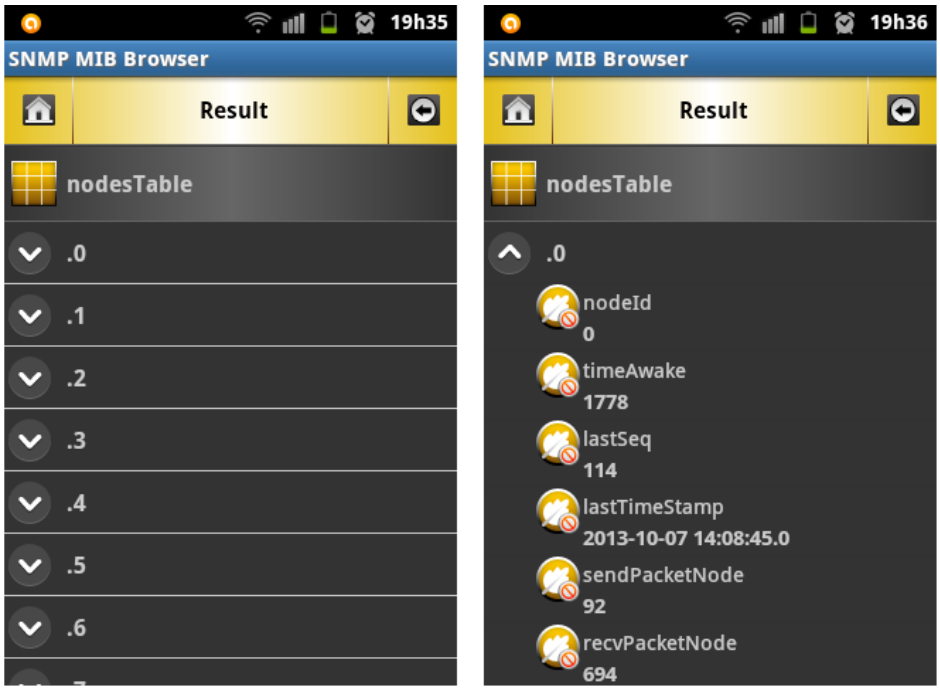


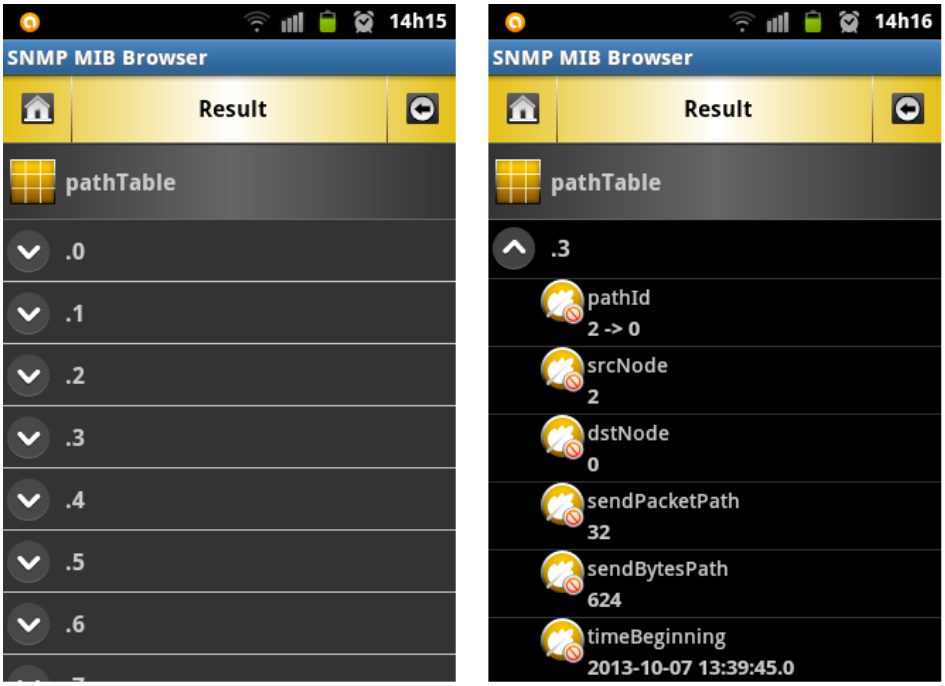
Figura 4.18 – Exibição da MIB do EPMOSt



(a) **(b)**

Figura 4.19 – Exibição das informações dos nós da rede alvo

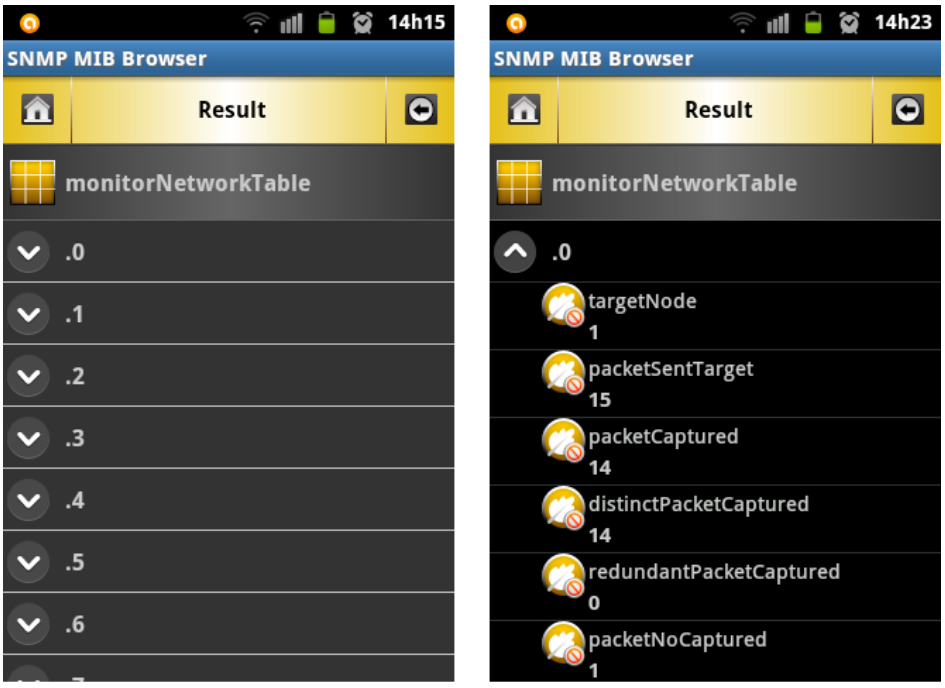
Ao clicar em *pathTable* na tela mostrada na Figura 4.18, o usuário visualiza os *paths* da rede alvo (Figura 4.20a). Ao clicar sobre um determinado *path* na tela mostrada na Figura 4.20a, o usuário visualiza as informações referentes a este *path* (Figura 4.20b).



(a) **(b)**

Figura 4.20 – Exibição das informações dos p*aths* da rede alvo

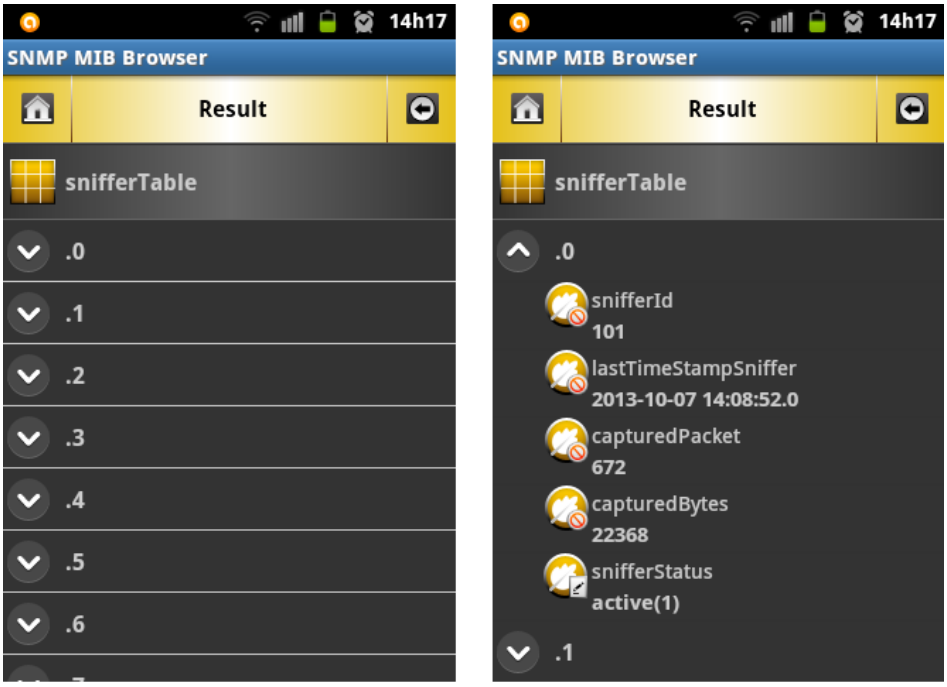
Ao clicar em *monitorNetworkTable* na tela mostrada na Figura 4.18, o usuário visualiza os nós da rede alvo (Figura 4.21a). Ao clicar sobre um determinado nó na tela mostrada na Figura 4.21a, o usuário visualiza as informações estatísticas sobre o monitoramento deste nó (Figura 4.21b).



(a) **(b)**

Figura 4.21 – Exibição de informações estatísticas sobre a rede de monitoramento

Ao clicar em *snifferTable* na tela mostrada na Figura 4.18, o usuário visualiza os *sniffers* (Figura 4.22a). Ao clicar sobre um determinado *sniffer* na tela mostrada na Figura 4.22a, o usuário visualiza informações estatísticas referentes a este *sniffer* (Figura 4.22b).



(a) **(b)**

Figura 4.22 – Exibição de informações estatísticas sobre os *sniffers*

A ferramenta "*ManageEngine MIB Browser Free Tool*" [ManageEngine 2014] também foi utilizada para testar o agente SNMP. Este software foi desenvolvido para executar em computadores com sistema operacional *Windows*. Desta forma, o gerente da rede também pode obter informações sobre o comportamento da RSSF monitorada a partir do seu computador. Esta ferramenta foi então instalada em um computador com sistema operacional *Windows 7* e foram realizadas consultas em todos os objetos da MIB do EPMOSt. Todas as consultas foram realizadas com sucesso. As Figuras 4.23 a 4.25 exemplificam o funcionamento desta ferramenta de gerência. A Figura 4.23 mostra a tela inicial da ferramenta, onde aparece a estrutura da MIB do EPMOSt. As Figuras 4.24 e 4.25 mostram as tabelas *nodesTable* (nós da rede alvo) e *pathTable* (*paths* da rede alvo), respectivamente.

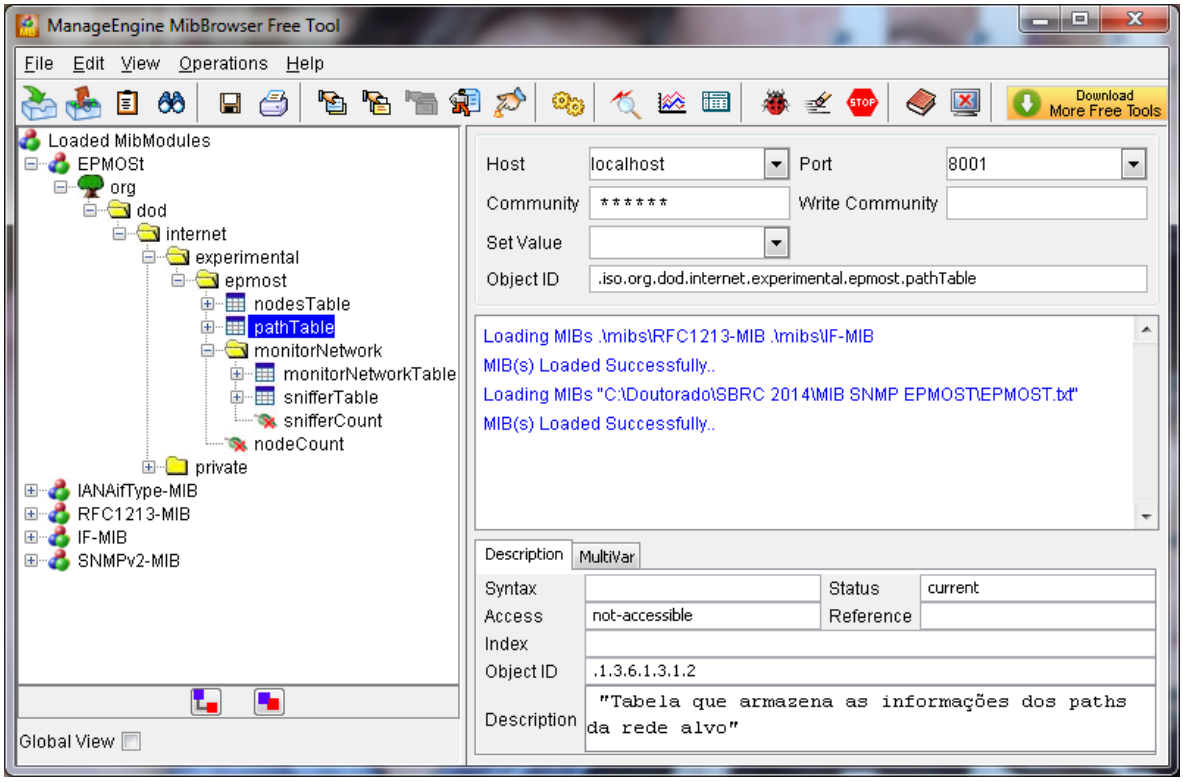


Figura 4.23 – Tela inicial da ferramenta *ManageEngine MIB Browser Free Tool*

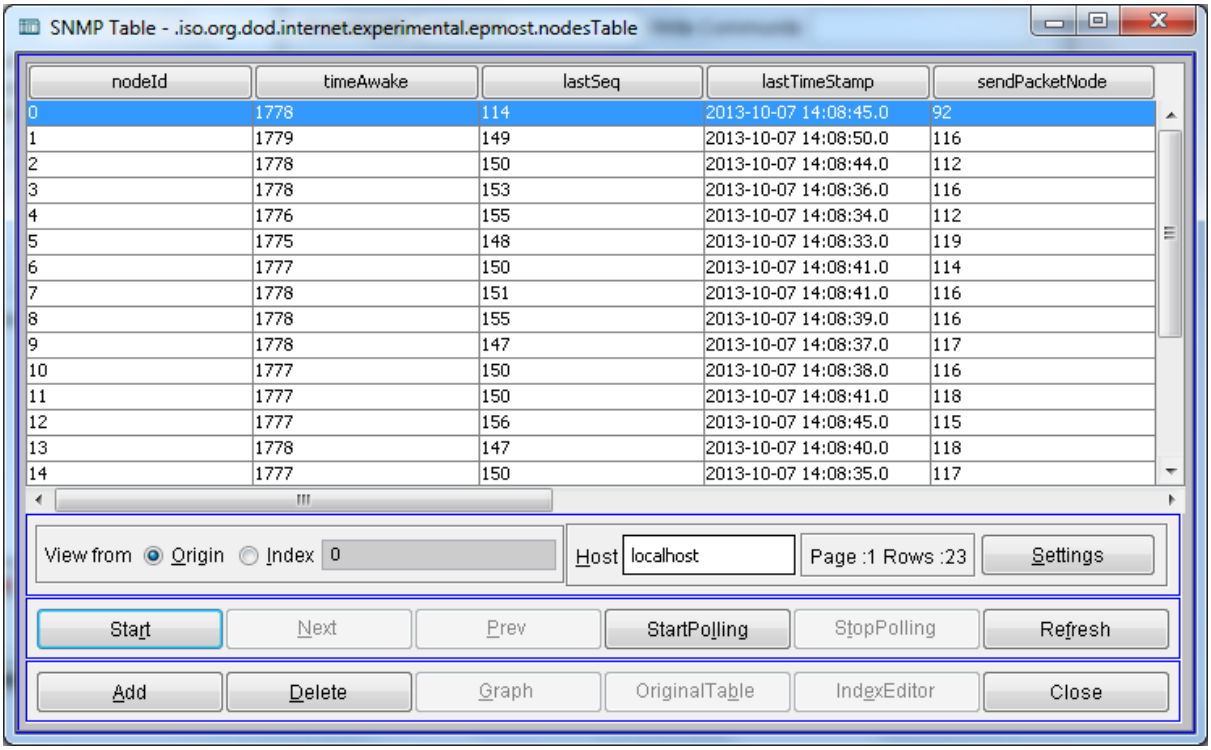


Figura 4.24 – Exibição da tabela *nodesTable* da MIB do EPMOSt

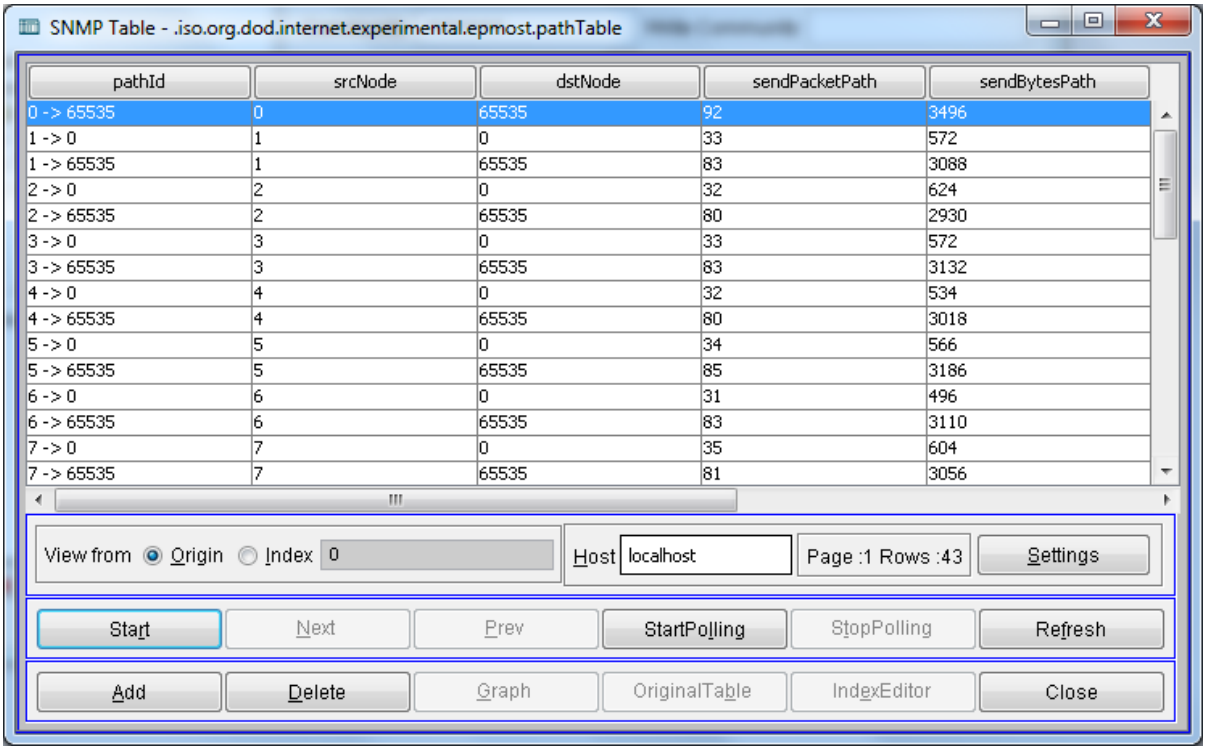


Figura 4.25 – Exibição da tabela path*Table* da MIB do EPMOSt

Portanto, os testes realizados com as ferramentas "*MIB Browser Android Tool"* [MIB Browser 2013] e *"ManageEngine MIB Browser Free Tool*" [ManageEngine 2014] mostram que o agente SNMP desenvolvido neste trabalho permite integrar o sistema de monitoramento EPMOSt com ferramentas de gerência que suportem o protocolo SNMP, facilitando a administração da RSSF monitorada.

4.2.5 Servidor - On/Off

Conforme descrito na seção 4.1, o EPMOSt possui um mecanismo para ligar e desligar (**on/off**) o monitoramento com o intuito de reduzir o consumo de energia dos *sniffers*. Assim, a partir do servidor, o usuário pode ativar (**on**) ou desativar (**off**) o monitoramento. O usuário pode também agendar os períodos de tempo em que a rede alvo será monitorada e neste caso, o servidor envia automaticamente o comando **on/off** de acordo com o agendamento realizado.

A Figura 4.26 mostra o diagrama de sequências UML deste mecanismo. O servidor envia o comando **on/off** para o monitor local, que o envia para a estação base. Em seguida, a estação base envia o comando em difusão (*broadcast*) para os *sniffers*. Quando um *sniffer* recebe o comando, ele envia uma mensagem de confirmação (ACK) para a estação base. Se a

estação base não receber a confirmação de todos os *sniffers*, ela retransmite o comando diretamente para os *sniffers* que não confirmaram o recebimento.

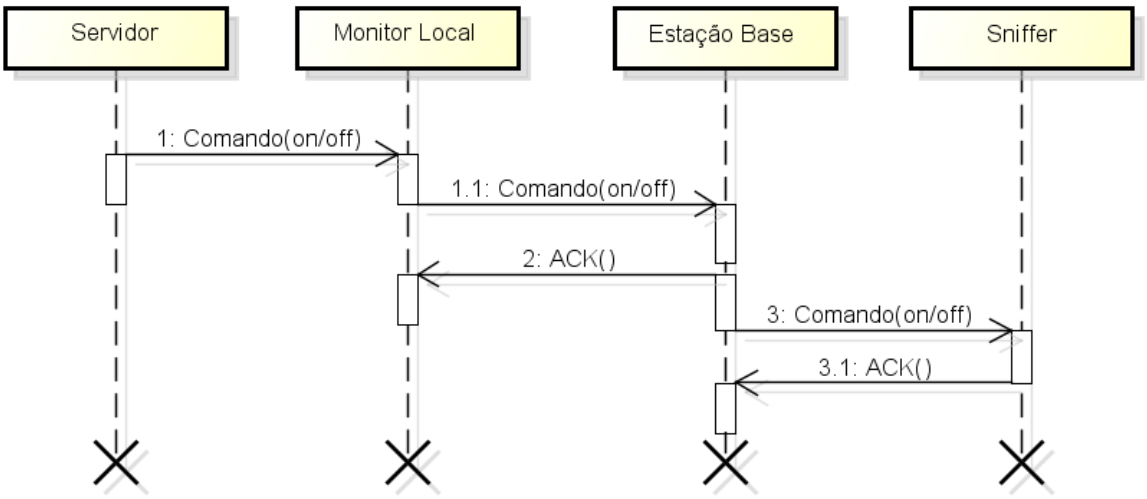


Figura 4.26 – Mecanismo para ligar/desligar o monitoramento

Ao receber um comando **on**, o *sniffer* inicializa o mecanismo de eleição descrito na seção 4.2.1.1. Ao receber um comando **off**, o *sniffer* comuta para o modo inativo (*sleep*), reduzindo assim o consumo de energia em até 95% [Jurdak et al. 2008]. Então, a cada 10 *ms* (intervalo de tempo definido em [Jurdak et al. 2008]), o *sniffer* comuta para o modo de recepção para verificar se algum pacote (comando) foi enviado para sua interface de rádio. Se receber um comando **on**, o *sniffer* permanece no estado de recepção e reinicia o processo de captura dos pacotes, caso contrário retorna ao modo inativo.

Em uma versão futura do EPMOSt, o monitoramento poderá ser automaticamente ativado ou desativado de acordo com as informações inferidas sobre o ciclo de trabalho da rede alvo. Por exemplo, ao analisar o histórico dos pacotes capturados, o servidor poderá perceber que a rede alvo envia pacotes apenas em determinados períodos do dia. Neste caso, o servidor poderá automaticamente ativar o monitoramento um pouco antes do início de cada período e desativar um pouco depois do final de cada período, reduzindo assim o consumo de energia da rede de monitoramento.

4.3 Considerações finais

Este capítulo apresentou o sistema de monitoramento EPMOSt proposto nesta tese de doutorado, e descreveu o funcionamento de cada um dos seus módulos de maneira detalhada. Também foi demonstrado que o EPMOSt pode ser integrado, através do seu agente SNMP,

com ferramentas de gerência que suportem o protocolo SNMP, facilitando a administração da RSSF monitorada. O próximo capítulo descreve os experimentos realizados para avaliar o EPMOSt, e apresenta e discute os resultados obtidos.

5 EXPERIMENTOS

Este capítulo descreve e analisa os experimentos realizados para avaliar o sistema de monitoramento EPMOSt. A seção 5.1 descreve os experimentos realizados em um *testbed*, bem como apresenta e discute os resultados obtidos. A seção 5.2 descreve os experimentos realizados através de simulações, e também apresenta e discute os resultados obtidos. Por fim, na seção 5.3 são apresentadas as considerações finais deste capítulo.

5.1 Experimentos em um *testbed*

Esta seção descreve os experimentos realizados em um *testbed* utilizando nós sensores da plataforma MicaZ e as métricas utilizadas para avaliar o sistema de monitoramento EPMOSt, bem como apresenta e discute os resultados obtidos.

5.1.1 Descrição dos experimentos

Os experimentos foram realizados utilizando nós MicaZ com sistema operacional TinyOS [Hill et al. 2000]. A plataforma MicaZ possui como principais características: microprocessador ATMEGA128L, 4KB de memória RAM, 128KB de memória ROM e transceptor de rádio frequência CC2420. O rádio CC2420 utiliza o padrão IEEE 802.15.4 e opera na faixa de frequência não licenciada de 2,4 GHz com taxa de transmissão efetiva de 250 Kbps [MPR-MIB 2013].

A Figura 5.1 ilustra um exemplo do cenário utilizado para a realização dos experimentos. A rede alvo é composta por **22** nós, sendo **21** nós sensores e um nó sorvedouro. A distância entre cada nó da rede alvo e seus vizinhos é de 0,3 m. Os nós sensores executam uma aplicação que a cada minuto mede a temperatura do ambiente e envia a informação para o nó sorvedouro utilizando roteamento em múltiplos saltos. A área de dados (*payload*) dos pacotes enviados pelo nó sensor contém a temperatura medida e um contador que é incrementado a cada medição de temperatura.

A rede de monitoramento é composta por **N** *sniffers* e uma estação base. Os *sniffers* capturam os pacotes enviados pelos nós da rede alvo e enviam para a estação base utilizando roteamento em múltiplos saltos. A estação base envia os pacotes recebidos dos *sniffers*, através de um cabo USB, para um computador. Neste cenário, o computador desempenha as

funções do monitor local (geração do *trace*) e do servidor mostrados na Figura 4.1. Foram realizados experimentos com 3, 5, 7, 9 e 11 *sniffers* distribuídos na área monitorada conforme mostrado na Figura 5.2. É importante ressaltar que a estação base tem como principal função intermediar o envio de pacotes entre os *sniffers* e o monitor local, e não captura nenhum pacote enviado pelos nós da rede alvo.

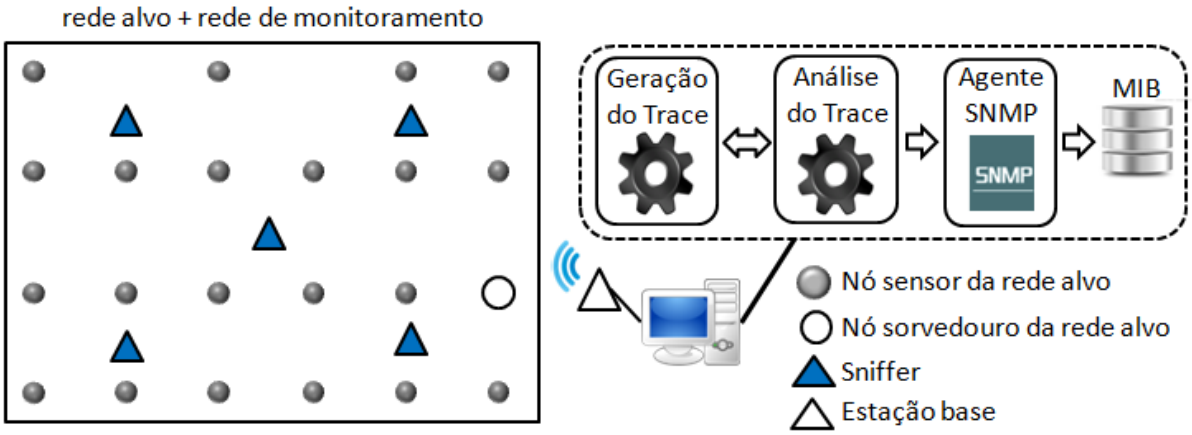


Figura 5.1 – Cenário utilizado nos experimentos

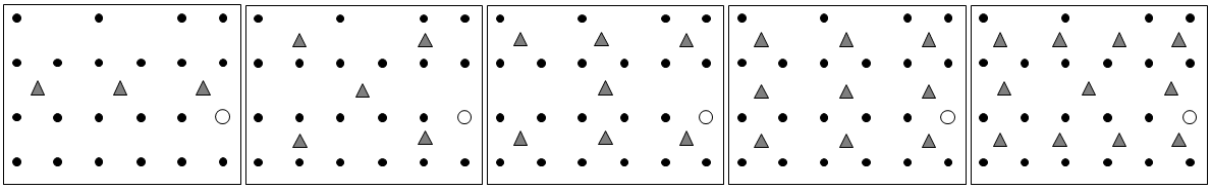


Figura 5.2 - Distribuição dos sniffers. (a) 3 sniffers; (b) 5 sniffers; (c) 7 sniffers; (d) 9 sniffers; (e) 11 sniffers.

Neste trabalho são realizados experimentos utilizando *testbed* em cenários com no máximo 34 nós (21 nós sensores da rede alvo, um nó sorvedouro da rede alvo, 11 *sniffers* e uma estação base) devido a quantidade de nós da plataforma MicaZ disponíveis no laboratório do GREat.

Para cada um dos cenários mostrados na Figura 5.2, três tipos de experimentos foram realizados: "*com eleição sem agregação*", "*com eleição com agregação*" e "*sem eleição*". No experimento "*com eleição sem agregação*", os *sniffers* executam a aplicação descrita na seção 4.2.1, onde é implementado o mecanismo de eleição proposto na seção 4.2.1.1, mas não utilizam o mecanismo de agregação de cabeçalhos proposto na seção 4.2.1.2. No experimento "*com eleição com agregação*", os sniffers executam os mecanismos de eleição e de agregação de cabeçalhos.

No experimento "*sem eleição*", os *sniffers* não executam nenhum mecanismo de eleição e capturam todos os pacotes enviados pelos nós da rede alvo que estão na área de cobertura dos seus rádios. Em seguida, os *sniffers* enviam todos os bytes dos pacotes capturados para o monitor local, que então armazena no banco de dados. Neste caso, vários *sniffers* podem capturar um único pacote enviado por um determinado nó da rede alvo. Vale ressaltar que esta é a estratégia utilizada por todos os sistemas de monitoramento descritos no capítulo 3 (SNTS, SNIF, Pimoto, LiveNet e PMSW).

5.1.2 Métricas

Para a avaliação dos experimentos foram definidas as seguintes métricas: energia consumida pela rede de monitoramento na transmissão dos pacotes capturados (Et), energia média consumida por cada *sniffer* na transmissão dos pacotes capturados ($EtSniffer$) e porcentagem de pacotes distintos capturados pela rede de monitoramento ($\%Pcapturados$). Estas métricas foram definidas a partir da análise dos sistemas de monitoramento passivo descritos no capítulo 3.

No EPMOSt, assim como em todos os sistemas de monitoramento analisados no capítulo 3 (SNTS, SNIF, Pimoto, Livenet e PMSW), os *sniffers* "escutam" todos os pacotes que trafegam nas suas interfaces de rádio. Portanto, a energia consumida na recepção de pacotes pelo sistema proposto é similar à energia consumida pelos sistemas descritos nos trabalhos relacionados, e por isso não foi utilizada como métrica de avaliação. No entanto, se o ciclo de trabalho (*duty cycle*) da rede alvo for conhecido, o monitoramento pode ser ativado um pouco antes do início do ciclo de trabalho e desativado um pouco depois do final do ciclo de trabalho da rede alvo, conforme descrito na seção 4.2.5. Desta forma, quando o monitoramento estiver desativado, os *sniffers* comutam para o modo inativo (*sleep*), contribuindo para reduzir ainda mais o consumo de energia da rede de monitoramento.

É importante ressaltar que a energia consumida pelos *sniffers* durante o processo de eleição (vide seção 4.2.1.1) não foi avaliada neste trabalho. Entretanto, este consumo de energia deve ser pequeno quando comparado ao consumo de energia dos *sniffers* para a transmissão dos pacotes capturados, pois o mecanismo de eleição é executado uma única vez, quando a rede de monitoramento é ativada.

A quantidade total de pacotes enviados pelos nós sensores da rede alvo ($PenvAlvo$) é obtida através da Equação 1, onde $contMediçãoInicial_i$ e $contMediçãoFinal_i$ são,

respectivamente, o número da primeira e da última medição de temperatura realizada pelo nó *i*, e *k* representa a quantidade de nós sensores da rede alvo. Nos cenários utilizados nestes experimentos e mostrados na Figura 5.2, o valor de *k* é 21.

$$PenvAlvo = \sum_{i=1}^{k} (contMediçãoFinal_i - contMediçãoInicial_i + 1) \quad (1)$$

A quantidade de pacotes distintos do nó *i* capturados pela rede de monitoramento ($Pcapturados_i$) é determinada verificando-se quais pacotes do nó *i* existem no intervalo [$contMediçãoInicial_i$, $contMediçãoFinal_i$]. Logo, a quantidade total de pacotes distintos capturados ($Pcapturados$) é obtida através da Equação 2.

$$Pcapturados = \sum_{i=1}^{k} Pcapturados_i \quad (2)$$

A porcentagem de pacotes distintos capturados pela rede de monitoramento ($\%Pcapturados$) é determinada pela Equação 3.

$$\%Pcapturados = 100 * Pcapturados/PenvAlvo \quad (3)$$

Conforme explicado na seção 4.2.3, a quantidade de pacotes redundantes do nó *i* ($Predundantes_i$) é determinada analisando-se o campo DSN (*Destination Sequence Number*) e o *timestamp* dos pacotes enviados pelo nó *i*. Logo, a quantidade total de pacotes redundantes capturados pela rede de monitoramento ($Predundantes$) é obtida através da Equação 4.

$$Predundantes = \sum_{i=1}^{k} Predundantes_i \quad (4)$$

Para calcular a energia consumida pela rede de monitoramento na transmissão dos pacotes foi utilizado o modelo de energia para sensores MicaZ definido em [Jurdak et al. 2008] e utilizado em [Rocha et al. 2012], [Garcia et al. 2013] e [Garcia et al. 2014]. Neste modelo, a energia consumida na transmissão (Et) é determinada pela Equação 5, onde $Psent$ é a quantidade de pacotes enviados, $Plength$ é o tamanho do pacote em bytes, TB é o tempo gasto na transmissão de um byte, It é o valor da corrente elétrica no modo de transmissão e V é a tensão elétrica da bateria.

$$Et = Psent\ x\ Plength\ x\ TB\ x\ It\ x\ V \quad (5)$$

Os valores utilizados para TB, It e V foram 32 µS, 17.4 mA e 3 Volts, respectivamente. Estes valores foram obtidos no documento de especificação da plataforma MicaZ (*datasheet*), que também são iguais aos valores apresentados em [Rocha et al. 2012] e [Jurdak et al. 2008]. Substituindo-se estes valores na Equação 5, obtém-se a Equação 6.

$$Et = 1{,}6704\ x\ 10^{-6}\ x\ Psent\ x\ Plength \quad (6)$$

É importante destacar que, ao utilizar nós sensores reais, a energia consumida na transmissão pode ser ligeiramente diferente da energia calculada pela Equação 5, pois os valores de It e V podem sofrer pequenas variações durante o tempo de vida da bateria.

Nos tipos de experimentos "*sem eleição*" e "*com eleição sem agregação*", a quantidade de pacotes enviados pelos *sniffers* é determinada pela Equação 7. Nestes dois tipos de experimentos, cada pacote enviado pelos *sniffers* tem tamanho ($Plength$) de **23** bytes, sendo cinco bytes de *header* (cabeçalho) e **18** bytes da mensagem de monitoramento (vide Figura 4.8). Então, a energia consumida pela rede de monitoramento na transmissão dos pacotes capturados da rede alvo é determinada pela Equação 8.

$$Psent = Pcapturados + Predundantes \quad (7)$$

$$Et = 38{,}42\ x\ 10^{-6}\ x\ (Pcapturados + Predundantes) \quad (8)$$

Quando a agregação de cabeçalhos é utilizada, conforme explicado na seção 4.2.1.2, os cabeçalhos de vários pacotes da rede alvo são enviados na mesma mensagem de monitoramento com o intuito de reduzir o consumo de energia dos *sniffers*. Em nossos experimentos, a mensagem de monitoramento transporta o endereço do *sniffer* (01 byte), o *timestamp* (02 bytes) e o cabeçalho (11 bytes) de dois pacotes da rede alvo. Portanto, no experimento "*com eleição com agregação*", cada pacote enviado pelos *sniffers* tem tamanho ($Plength$) de **32** bytes, sendo cinco bytes de cabeçalho e **27** bytes da mensagem de monitoramento (vide Figura 4.9). Neste tipo de experimento, a quantidade de pacotes enviados pelos *sniffers* é determinada pela Equação 9. Substituindo-se $Psent$ determinado pela Equação 9 e $Plength$ igual a 32 na Equação 6 obtém-se a Equação 10, que representa a energia consumida pela rede de monitoramento na transmissão dos pacotes capturados quando a agregação de cabeçalhos é utilizada.

$$Psent = (Pcapturados + Predundantes)/2 \quad (9)$$

$$Et = 53{,}45\ x\ 10^{-6}\ x\ (Pcapturados + Predundantes)\ /\ 2 \quad (10)$$

A energia média consumida por cada *sniffer* na transmissão dos pacotes capturados ($EtSniffer$) é determinada pela Equação 11, onde N é a quantidade de *sniffers* e Et pode ser calculada pelas Equações 8 ou 10 de acordo com tipo do experimento realizado.

$$EtSniffer = Et/N \tag{11}$$

5.1.3 Resultados e discussão

Conforme mencionado anteriormente, para cada um dos cenários mostrados na Figura 5.2, três tipos de experimentos foram realizados: "*com eleição sem agregação*", "*com eleição com agregação*" e "*sem eleição*". Para cada cenário e para cada tipo de experimento foram realizados 10 experimentos com duração de 15 minutos. Todos os experimentos foram realizados no mesmo ambiente e sob as mesmas condições. Além disso, em cada cenário, os nós da rede alvo e da rede de monitoramento foram mantidos na mesma localização durante a realização dos 10 experimentos. Os resultados mostrados nos gráficos das Figuras 5.3 a 5.5 referem-se aos valores médios dos 10 experimentos realizados com intervalo de confiança de 95%. As margens de erro dos experimentos são perceptíveis apenas no gráfico da Figura 5.5, pois nos gráficos das Figuras 5.3 e 5.4 seus valores são muito pequenos e não podem ser visualizados.

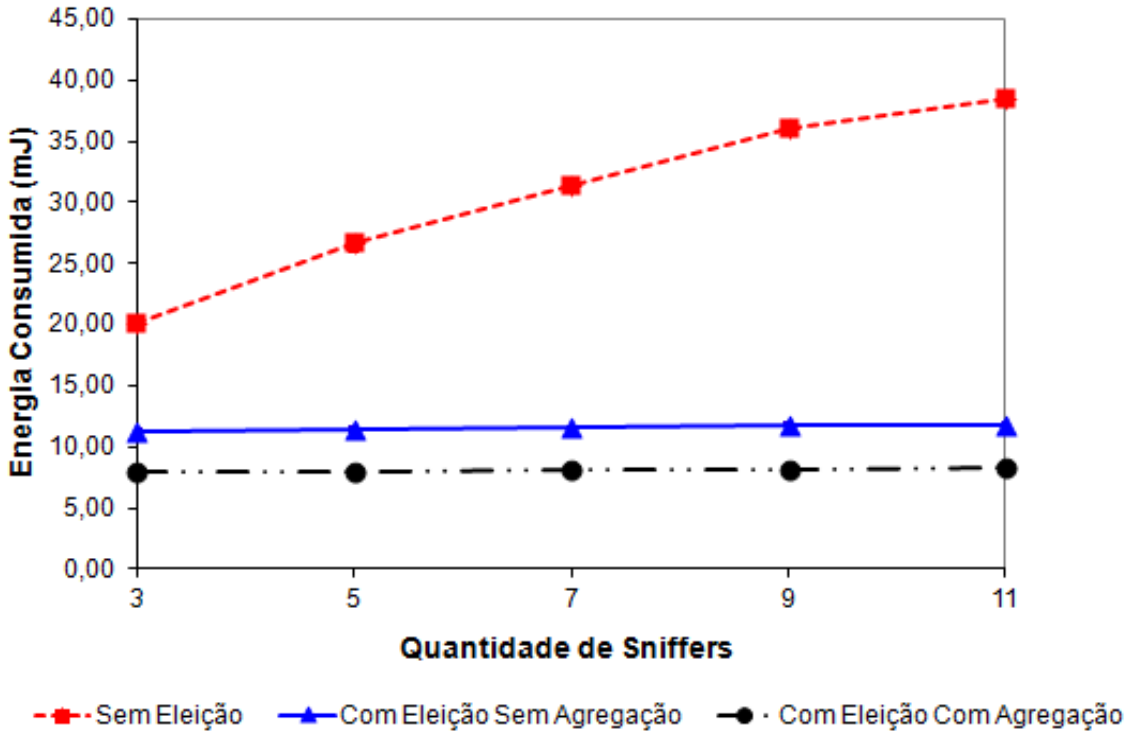


Figura 5.3 – Energia consumida pela rede de monitoramento vs. quantidade de *sniffers*

A Figura 5.3 mostra a energia consumida pela rede de monitoramento na transmissão dos pacotes capturados (Et) em função da quantidade de *sniffers*. Pode-se observar que quando o mecanismo de eleição não é utilizado, a energia consumida pela rede de monitoramento aumenta quando a quantidade de *sniffers* aumenta. Isto acontece porque os pacotes enviados por um determinado nó da rede alvo são capturados por uma quantidade maior de *sniffers*, aumentando assim a quantidade de pacotes redundantes capturados e, consequentemente, aumentando o consumo de energia da rede de monitoramento na transmissão destes pacotes. Nos dois tipos de experimentos em que o mecanismo de eleição é utilizado, o consumo de energia da rede de monitoramento permanece quase constante, pois quando a quantidade de *sniffers* aumenta, cada *sniffer* captura pacotes de uma quantidade menor de nós da rede alvo, mas a quantidade total de pacotes enviados pela rede de monitoramento quase não sofre alterações.

Pode-se perceber ainda na Figura 5.3 que, para 11 *sniffers*, a energia consumida pela rede de monitoramento é de 38,4 mJ quando o mecanismo de eleição e a agregação de cabeçalhos não são utilizados. Ao utilizar o mecanismo de eleição sem agregação de cabeçalhos, o consumo de energia é de 11,8 mJ, que corresponde a uma redução de 69,3%. Esta redução no consumo de energia deve-se ao fato de que a utilização do mecanismo de eleição reduz significativamente a quantidade de pacotes redundantes transmitidos pelos *sniffers*. Quando é utilizado o mecanismo de eleição com agregação de cabeçalhos, o consumo de energia da rede de monitoramento é de apenas 8,27 mJ, que corresponde a uma redução de 78,5%. Estes resultados comprovam que o sistema de monitoramento EPMOSt reduz consideravelmente a energia consumida pela rede de monitoramento na transmissão dos pacotes capturados.

A Figura 5.4 mostra a energia média consumida por cada *sniffer* na transmissão dos pacotes capturados ($EtSniffer$) em função da quantidade de *sniffers*. Pode-se observar que, para os três tipos de experimentos, quando a quantidade de *sniffers* aumenta, ocorre uma redução da energia consumida por cada *sniffer*. Isto acontece porque os *sniffers* formam uma RSSF independente com topologia baseada em múltiplos saltos. Então, quando a quantidade de *sniffers* aumenta, os *sniffers* ficam mais próximos uns dos outros e a potência de transmissão das suas interfaces de rádio é reduzida, reduzindo assim a área de cobertura de cada *sniffer* e, consequentemente, cada *sniffer* captura pacotes de uma quantidade menor de nós da rede alvo. Pode-se observar também que a utilização do mecanismo de eleição reduz

consideravelmente o consumo de energia dos *sniffers* e, consequentemente, prolonga o tempo de vida da rede de monitoramento.

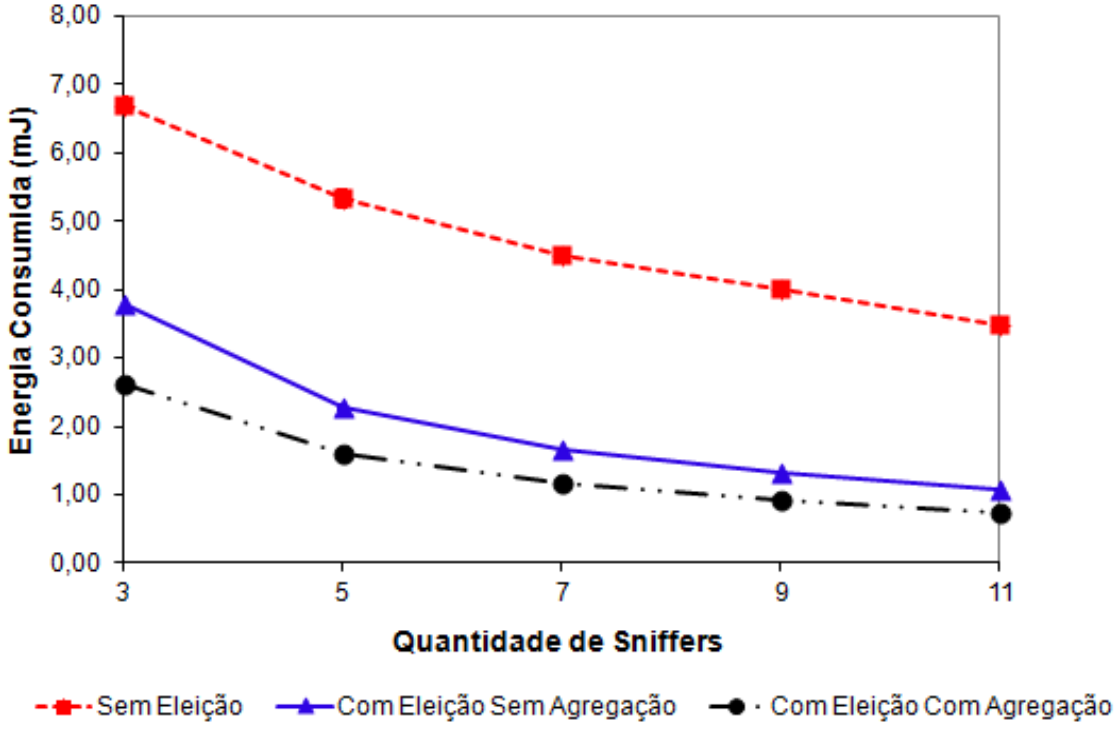


Figura 5.4 – Energia consumida por cada *sniffer* vs. quantidade de *sniffers*

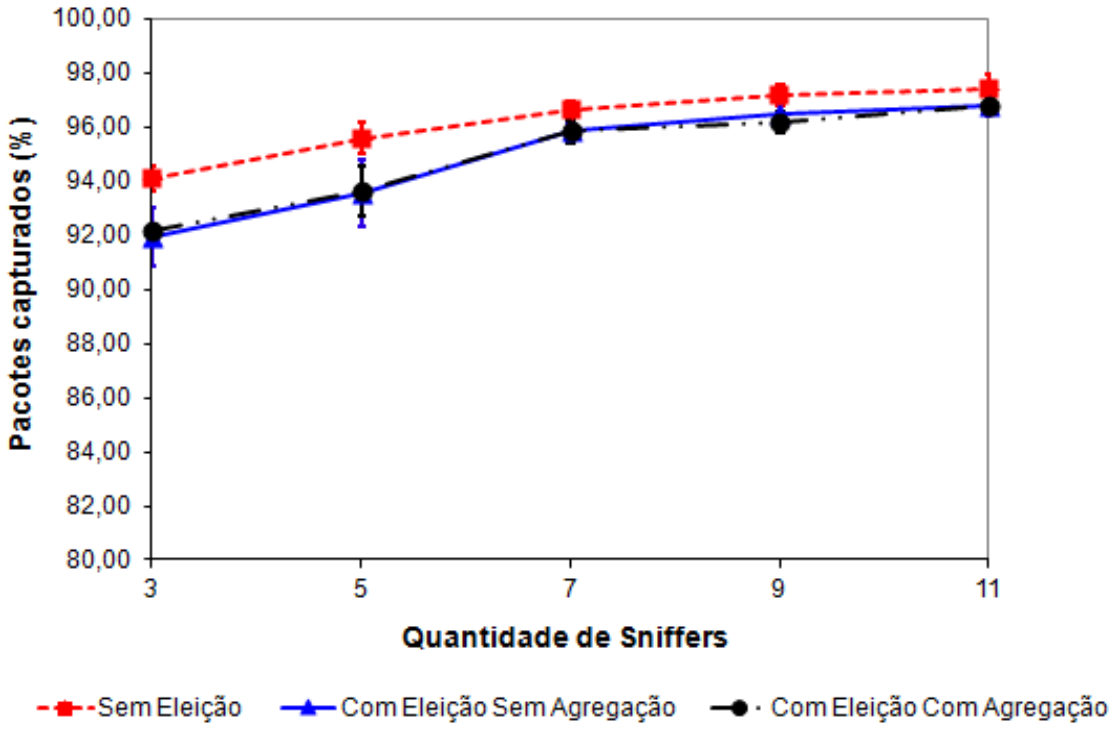


Figura 5.5 – Pacotes capturados pela rede de monitoramento vs. quantidade de *sniffers*

A Figura 5.5 mostra a porcentagem de pacotes distintos capturados pela rede de monitoramento ($\%Pcapturados$) em função da quantidade de *sniffers*. Pode-se observar que, para os três tipos de experimentos, quando a quantidade de *sniffers* aumenta, a porcentagem de pacotes capturados também aumenta. Isto acontece porque os *sniffers* ficam mais próximos dos nós da rede alvo e, portanto, recebem os sinais de rádio com maior nível de potência (RSSI). Pode-se perceber ainda que quando não é utilizado o mecanismo de eleição, a porcentagem de pacotes capturados é um pouco maior do que nos dois tipos de experimentos que utilizam o mecanismo de eleição, pois o mesmo pacote pode ser capturado por mais de um *sniffer*, aumentando assim a probabilidade de capturá-lo. No entanto, esta diferença entre os pacotes capturados reduz com o aumento da quantidade de *sniffers* e é de apenas 0,62% com 11 *sniffers*.

Os resultados apresentados nesta seção demonstram que o EPMOSt, quando comparado com os sistemas de monitoramento descritos no capítulo 3 (que não utilizam nenhum mecanismo de eleição de *sniffers* e de agregação de cabeçalhos), reduz consideravelmente a energia consumida pela rede de monitoramento na transmissão dos pacotes capturados e mantém a porcentagem de pacotes capturados próxima aos valores obtidos sem a utilização do mecanismo de eleição. Estes resultados comprovam a viabilidade de se utilizar o EPMOSt quando se deseja monitorar uma RSSF durante um longo período de tempo.

5.2 Experimentos com simulador

Os experimentos apresentados na seção 5.1 utilizaram sensores reais da plataforma MicaZ em cenários com no máximo 34 nós (21 nós sensores da rede alvo, um nó sorvedouro da rede alvo, 11 *sniffers* e uma estação base) devido a quantidade de sensores disponíveis no laboratório do GREat. Com o intuito de avaliar o comportamento do EPMOSt em cenários com uma quantidade maior de nós e também de comparar com os resultados obtidos utilizando o *testbed*, optou-se por realizar simulações.

O simulador COOJA [Osterlind et al. 2006] foi escolhido por ser um simulador de RSSF que combina a simulação de baixo nível do hardware do nó sensor e a simulação de alto nível do comportamento do nó em uma única simulação. Além disso, O COOJA é flexível e extensível, de modo que todas as camadas do nó sensor podem ser alteradas ou substituídas

utilizando diferentes modelos de plataforma dos nós sensores (entre as quais, MicaZ), de sistema operacional e de rádio transmissão [Osterlind et al. 2006].

Para a realização das simulações, o COOJA foi instalado em uma máquina virtual *Ubuntu Server 12.04.4 LTS* com quatro núcleos de processamento e 28 GB de memória RAM hospedada na plataforma de computação em nuvem *Microsoft Azure* [Azure 2014].

Inicialmente foram realizadas simulações com os mesmos cenários utilizados no *testbed* (vide Figura 5.2) com o intuito de comparar os resultados obtidos nas simulações com os resultados obtidos a partir dos experimentos realizados com sensores reais. Para exemplificar a utilização do COOJA, a Figura 5.6 mostra a interface do COOJA durante a simulação do cenário mostrado na Figura 5.2b. Neste cenário, os nós sensores da rede alvo têm identificadores de 01 a 21, o nó sorvedouro da rede alvo tem identificador 25 e os *sniffers* têm identificadores 22, 23, 24, 26 e 27.

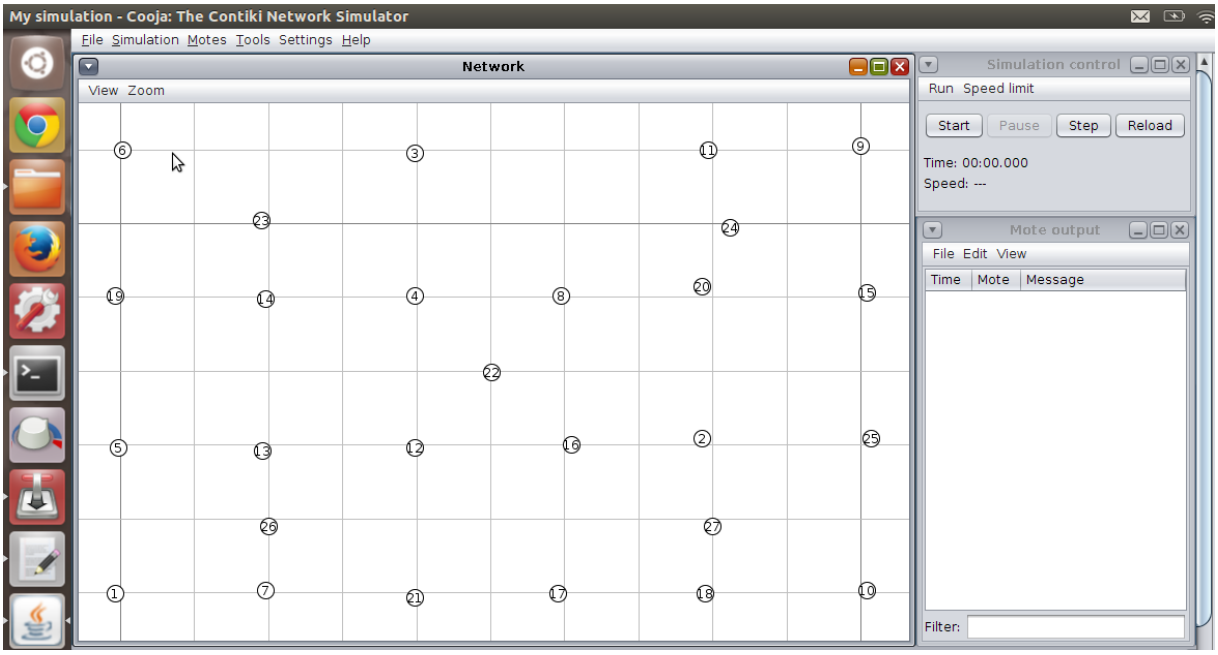


Figura 5.6 – Interface do COOJA durante a simulação do cenário da Figura 5.2b

Para cada um dos cenários mostrados na Figura 5.2, dois tipos de experimentos foram realizados: "*com eleição sem agregação*" e "*sem eleição*". Para cada cenário e para cada tipo de experimento foram realizadas 10 simulações com duração de 15 minutos. Os resultados mostrados nos gráficos das Figuras 5.7 a 5.9 referem-se aos valores médios das 10 simulações realizadas com intervalo de confiança de 95%.

A Figura 5.7 mostra a energia consumida pela rede de monitoramento na transmissão dos pacotes capturados (Et) em função da quantidade de *sniffers*. Pode-se observar que as curvas dos tipos de experimentos "*sem eleição*" e "*com eleição sem agregação*" representadas na Figura 5.7 apresentam comportamentos similares aos comportamentos das curvas representadas na Figura 5.3, as quais foram obtidas a partir de experimentos utilizando o *testbed*. Quando o mecanismo de eleição não é utilizado, a energia consumida pela rede de monitoramento aumenta quando a quantidade de *sniffers* aumenta. Por outro lado, quando o mecanismo de eleição é utilizado, o consumo de energia da rede de monitoramento permanece quase constante.

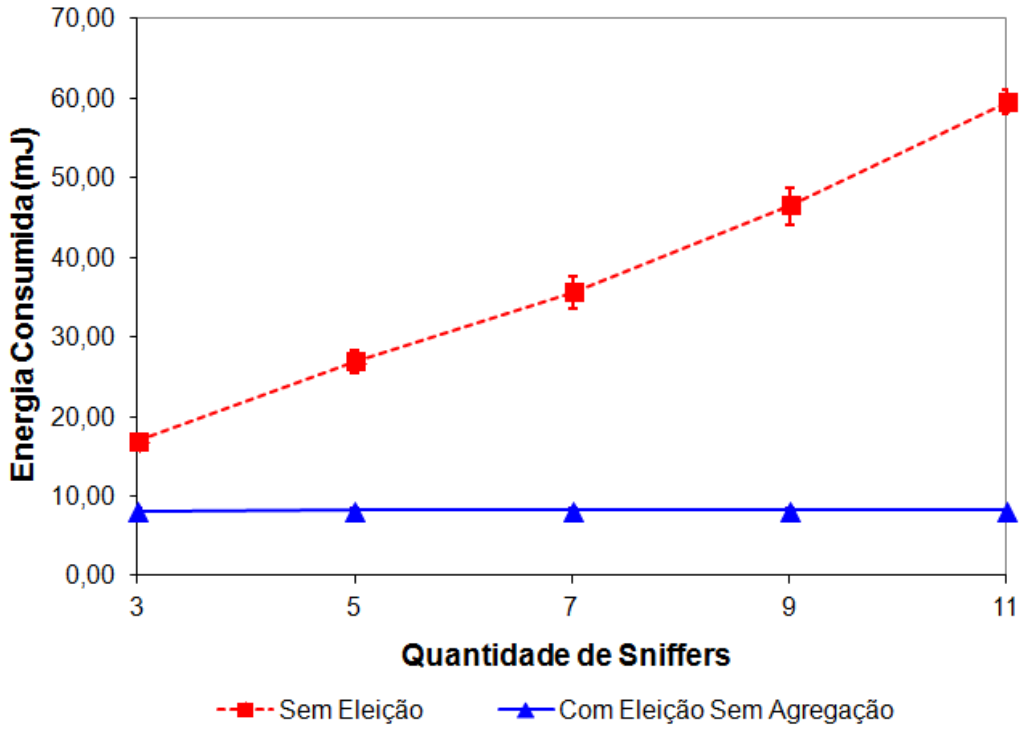


Figura 5.7 – Energia consumida pela rede de monitoramento vs. quantidade de *sniffers* para os cenários com 21 nós sensores da rede alvo (Figura 5.2)

Pode-se perceber ainda na Figura 5.7 que, para 11 *sniffers*, a energia consumida pela rede de monitoramento é de 59,5 mJ quando o mecanismo de eleição não é utilizado. Ao utilizar o mecanismo de eleição sem agregação de cabeçalhos, o consumo de energia é de 8,2 mJ, que corresponde a uma redução de 86,2%. Nos experimentos utilizando o *testbed*, a redução do consumo de energia é de 69,3% quando a rede de monitoramento possui 11 *sniffers*. Esta diferença entre os valores obtidos na simulação e no *testbed* deve-se ao fato de que, quando o mecanismo de eleição não é utilizado, a energia consumida pela rede de monitoramento para a transmissão dos pacotes capturados na simulação é maior do que a

energia consumida no *testbed*, pois a quantidade de pacotes redundantes capturados pelos *sniffers* é maior na simulação.

A Figura 5.8 mostra a energia média consumida por cada *sniffer* na transmissão dos pacotes capturados ($EtSniffer$) em função da quantidade de *sniffers*. Pode-se observar que a curva do tipo experimento "*com eleição sem agregação*" representada na Figura 5.8 apresenta comportamento similar ao comportamento da curva representada na Figura 5.4, que foi obtida a partir de experimentos utilizando o *testbed*. Quando o mecanismo de eleição é utilizado, a energia consumida por cada *sniffer* reduz quando a quantidade de *sniffers* aumenta.

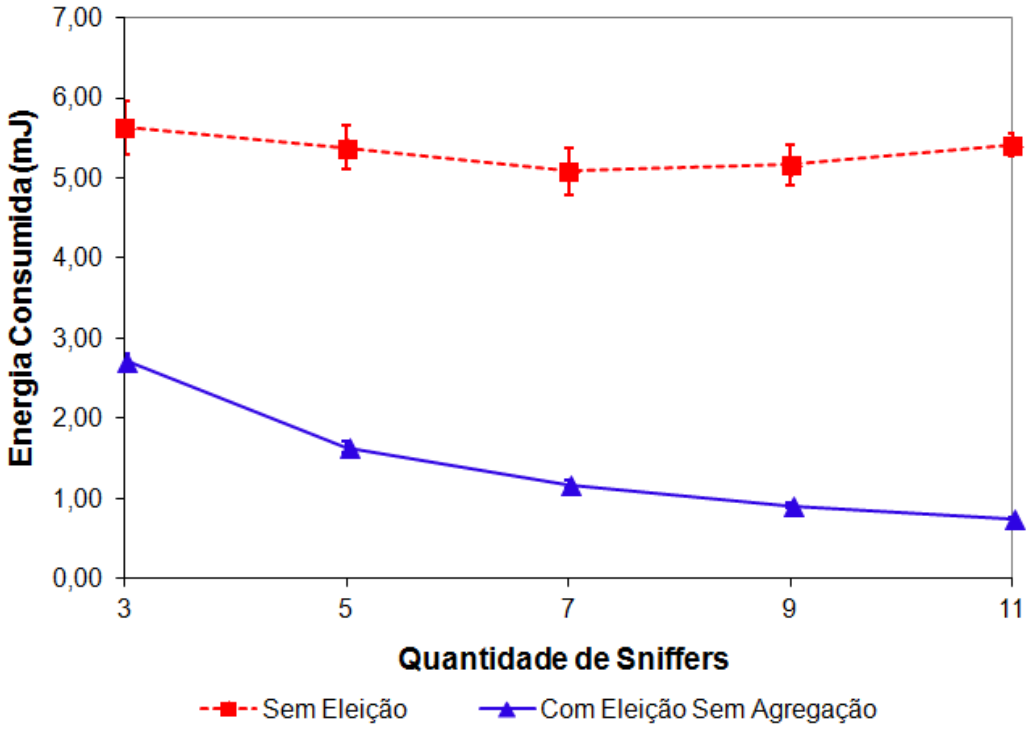


Figura 5.8 – Energia consumida por cada *sniffer* vs. quantidade de *sniffers* para os cenários com 21 nós sensores da rede alvo (Figura 5.2)

Pode-se perceber ainda na Figura 5.8 que, diferentemente dos experimentos realizados utilizando o *testbed*, no tipo de experimento "*sem eleição*" a energia consumida por cada *sniffer* permanece quase constante (varia de 5,1 mJ a 5,6 mJ) quando a quantidade de *sniffers* aumenta. Isto acontece porque no simulador, diferentemente do *testbed*, a potência de transmissão das interfaces de rádio dos *sniffers* não é reduzida quando os *sniffers* ficam mais próximos uns dos outros.

A Figura 5.9 mostra a porcentagem de pacotes distintos capturados pela rede de monitoramento ($\%Pcapturados$) em função da quantidade de *sniffers*. Pode-se observar que,

para os tipos de experimentos "*sem eleição*" e "*com eleição sem agregação*", a porcentagem de pacotes capturados permanece acima de 98,7% para qualquer quantidade de *sniffers* e atinge o valor de 99,5% quando a rede de monitoramento possui 11 *sniffers*. Por outro lado, a porcentagem de pacotes capturados variou de 91,9% a 97.5% nos experimentos que utilizaram o *testbed* (vide Figura 5.5). Acredita-se que esta diferença pode ser atribuída à presença de interferências e/ou ruídos durante a transmissão dos pacotes pelos nós da plataforma MicaZ utilizados no *testbed*, pois as interfaces de rádio destes nós operam na faixa de frequência não licenciada de 2,4 GHz, que também é utilizada por diversos outros dispositivos (e.g., redes de computadores sem fio e dispositivos com interfaces *Bluetooth*).

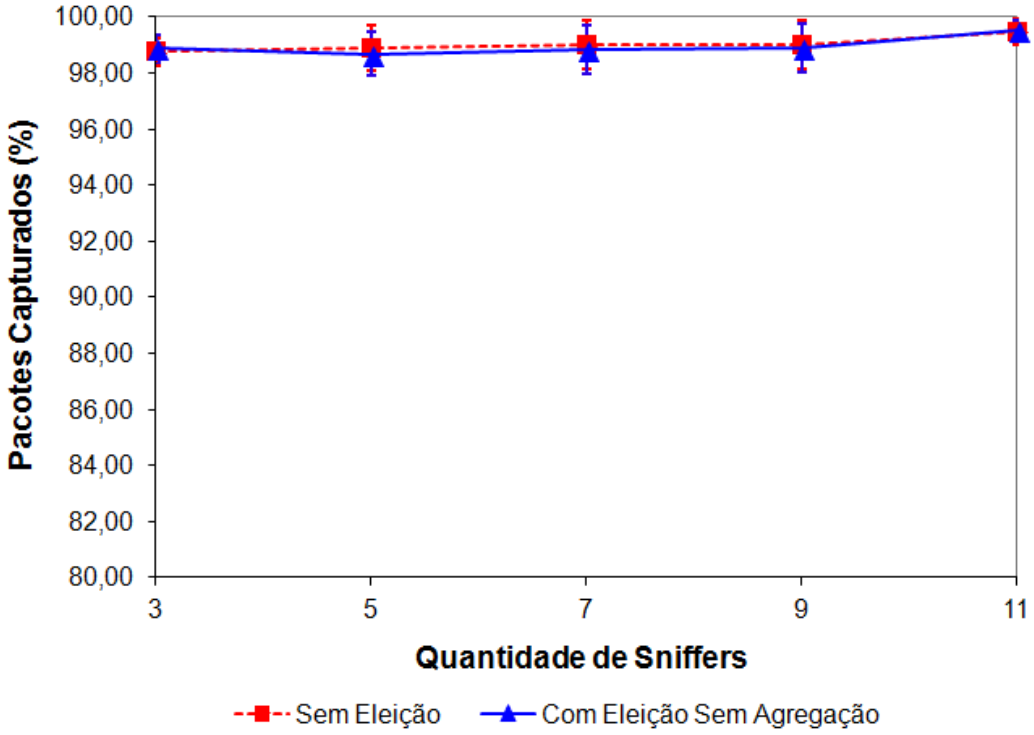


Figura 5.9 – Pacotes capturados pela rede de monitoramento vs. quantidade de *sniffers* para os cenários com 21 nós sensores da rede alvo (Figura 5.2)

Os experimentos apresentados até aqui foram realizados em cenários com no máximo 34 nós. Entretanto, foram também realizadas simulações para avaliar o comportamento do EPMOSt em cenários com uma quantidade maior de nós. Nestes cenários a rede alvo é composta por **52** nós, sendo **50** nós sensores e **02** nós sorvedouros, onde cada nó sorvedouro recebe os dados enviados por **25** nós sensores. A rede de monitoramento é composta por **N** *sniffers* e uma estação base. Foram realizadas simulações em quatro cenários, com 10, 15, 20 e 25 *sniffers* distribuídos na área monitorada. Para exemplificar, a Figura 5.10 mostra a interface do COOJA durante a simulação do cenário com 15 *sniffers*. Neste cenário, os nós

sensores da rede alvo têm identificadores de 01 a 50, os nós sorvedouros da rede alvo têm identificadores 51 e 52, e os *sniffers* têm identificadores 53 a 67.

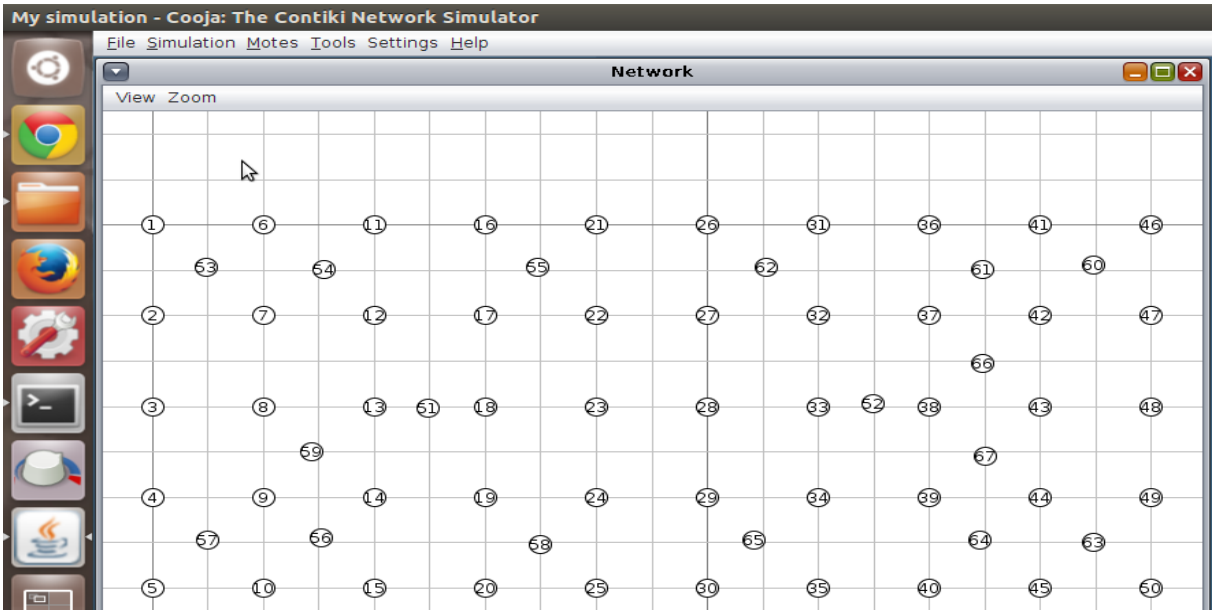


Figura 5.10 – Interface do COOJA durante a simulação do cenário com 50 nós sensores da rede alvo e 15 sniffers

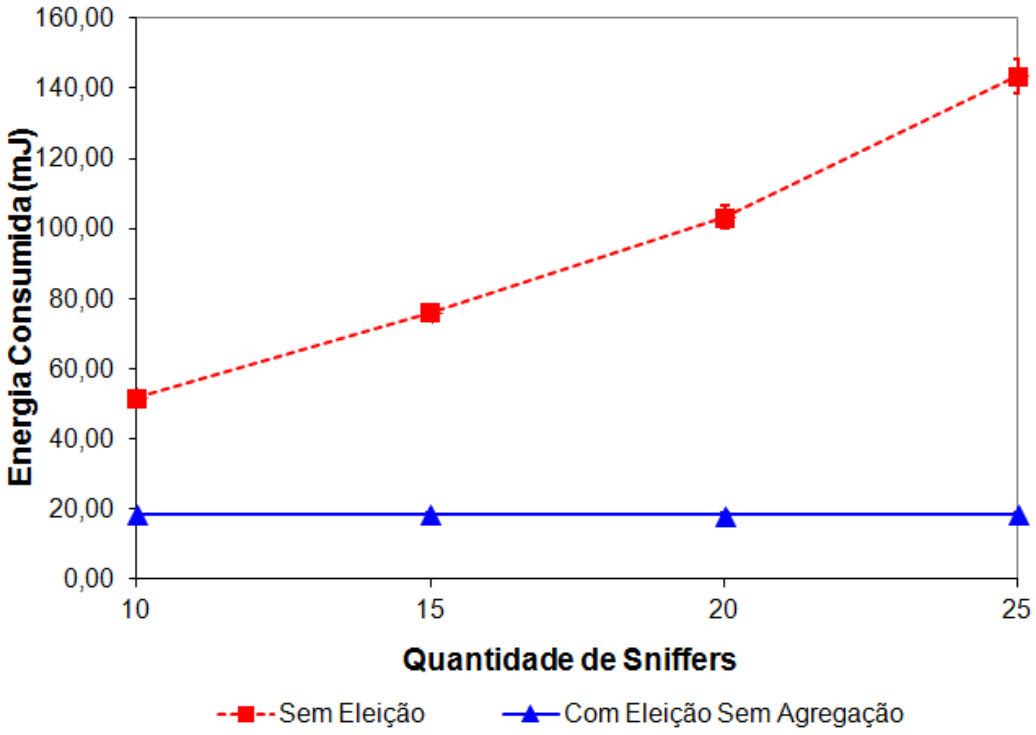


Figura 5.11 – Energia consumida pela rede de monitoramento vs. quantidade de *sniffers* para os cenários com 50 nós sensores da rede alvo

Para cada um dos quatro cenários, dois tipos de experimentos foram realizados: "*com eleição sem agregação*" e "*sem eleição*". Para cada cenário e para cada tipo de experimento foram realizadas 10 simulações com duração de 15 minutos. Os resultados mostrados nos gráficos das Figuras 5.11 a 5.13 referem-se aos valores médios das 10 simulações realizadas com intervalo de confiança de 95%.

A Figura 5.11 mostra a energia consumida pela rede de monitoramento na transmissão dos pacotes capturados (Et) em função da quantidade de *sniffers* para os cenários com 50 nós sensores da rede alvo. Pode-se observar que as curvas representadas na Figura 5.11 apresentam comportamentos similares aos comportamentos das curvas representadas na Figura 5.7, as quais foram obtidas a partir de simulações em cenários com 21 nós sensores da rede alvo. Nos cenários com 50 nós sensores da rede alvo, a redução do consumo de energia da rede de monitoramento varia de 64,4% (10 *sniffers*) a 87,1% (25 *sniffers*) quando o mecanismo de eleição é utilizado.

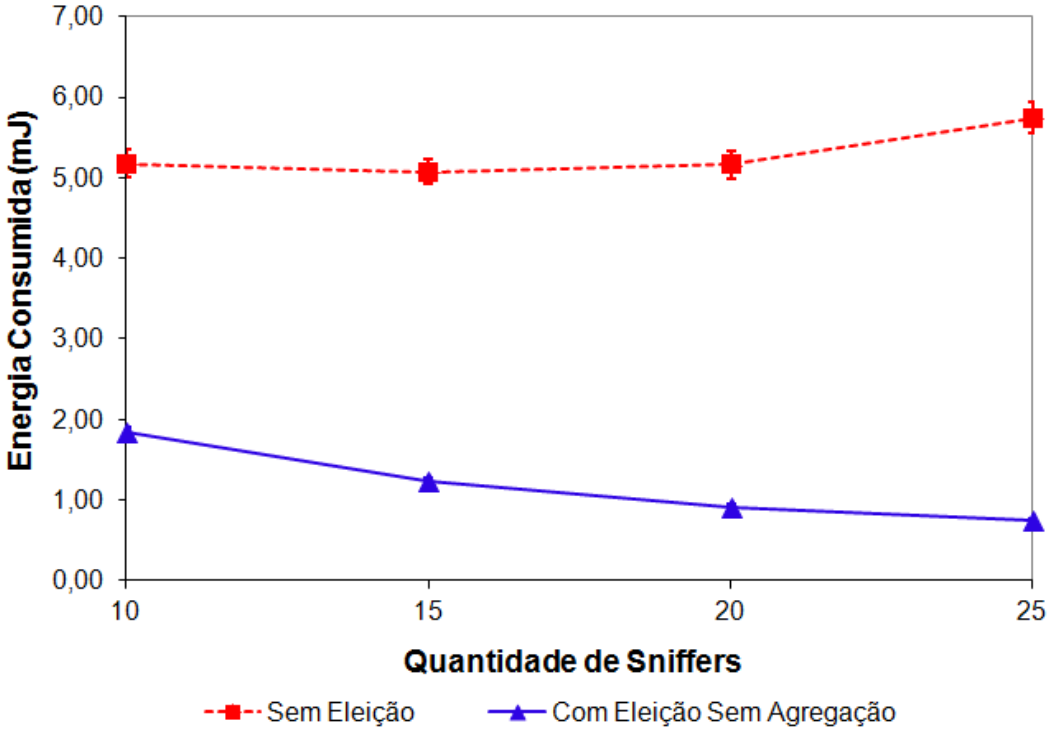


Figura 5.12 – Energia consumida por cada *sniffer* vs. quantidade de *sniffers* para os cenários com 50 nós sensores da rede alvo

A Figura 5.12 mostra a energia média consumida por cada *sniffer* na transmissão dos pacotes capturados ($EtSniffer$) em função da quantidade de *sniffers* para os cenários com 50 nós sensores da rede alvo. Pode-se observar que as curvas representadas na Figura 5.12

apresentam comportamentos similares aos comportamentos das curvas representadas na Figura 5.8, as quais foram obtidas a partir de simulações em cenários com 21 nós sensores da rede alvo. Pode-se perceber ainda na Figura 5.12 que, quando o mecanismo de eleição não é utilizado, a energia consumida por cada *sniffer* permanece acima de 5,0 mJ para qualquer quantidade de *sniffers*. Ao utilizar o mecanismo de eleição, a energia consumida por cada *sniffer* reduz substancialmente e varia de 1,84 mJ (10 *sniffers*) a 0,74 mJ (25 *sniffers*).

A Figura 5.13 mostra a porcentagem de pacotes distintos capturados pela rede de monitoramento ($\%Pcapturados$) em função da quantidade de *sniffers* para os cenários com 50 nós sensores da rede alvo. Pode-se observar que, para os dois tipos de experimentos, a porcentagem de pacotes capturados permanece acima de 93,4% para qualquer quantidade de *sniffers*. Pode-se perceber ainda que, quando não é utilizado o mecanismo de eleição, a porcentagem de pacotes capturados é ligeiramente maior do que nos cenários que utilizam o mecanismo de eleição, pois o mesmo pacote pode ser capturado por mais de um *sniffer*, aumentando assim a probabilidade de capturá-lo. No entanto, esta diferença entre os pacotes capturados reduz com o aumento da quantidade de *sniffers* e é de apenas 0,41% com 25 *sniffers*.

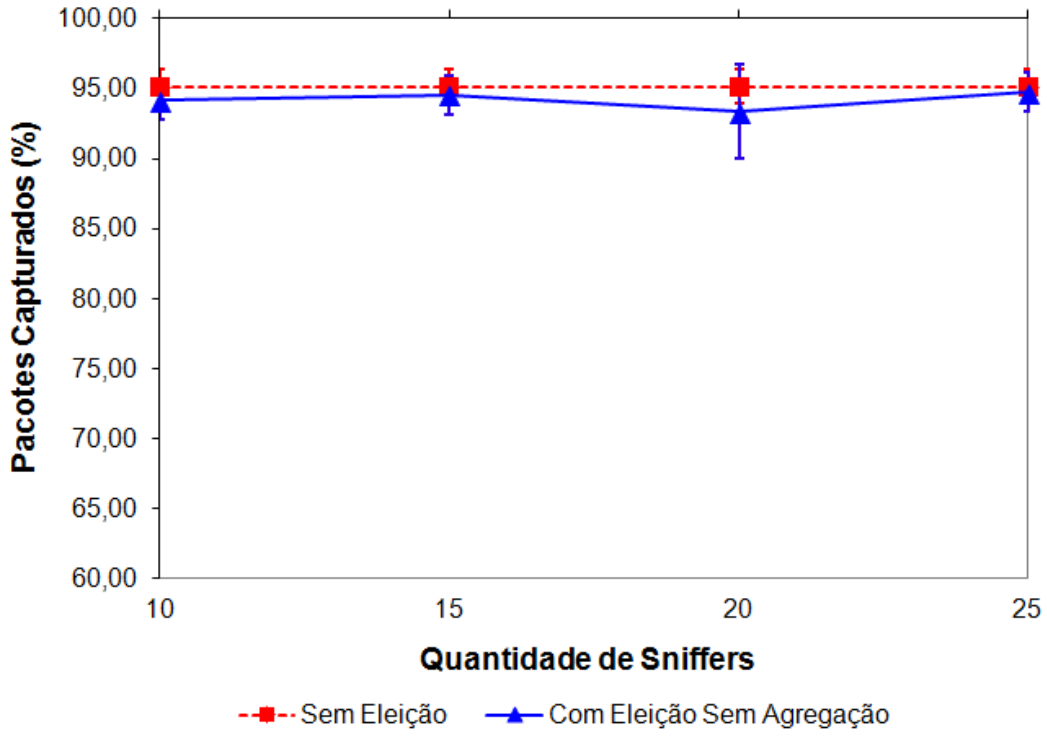


Figura 5.13 – Pacotes capturados pela rede de monitoramento vs. quantidade de *sniffers* para os cenários com 50 nós sensores da rede alvo

Para avaliar o comportamento do sistema de monitoramento EPMOSt foram realizadas simulações em duas redes alvo: uma rede com 21 nós sensores e 01 nó sorvedouro, e uma rede com 50 nós sensores e 02 nós sorvedouros. Os resultados obtidos comprovam que o sistema de monitoramento EPMOSt apresentou comportamento similar em ambas as redes. Além disso, os resultados apresentados demonstram também que nestas duas redes alvo o mecanismo de eleição de *sniffers* utilizado pelo EPMOSt reduz consideravelmente o consumo de energia da rede de monitoramento e mantém a porcentagem de pacotes capturados próxima aos valores obtidos sem a utilização do mecanismo de eleição.

As simulações realizadas na rede alvo com 21 nós sensores e 01 nó sorvedouro utilizaram os mesmos cenários dos experimentos realizados no *testbed*, e os resultados obtidos nas simulações e no *testbed* foram descritos, analisados e comparados.

5.3 Considerações finais

Neste capítulo os experimentos realizados para avaliar o sistema de monitoramento EPMOSt foram descritos, bem como os resultados alcançados foram apresentados e discutidos. Foram realizados experimentos utilizando nós sensores da plataforma MicaZ e o simulador COOJA, e os resultados obtidos comprovaram a eficiência energética do EPMOSt e a viabilidade de utilizá-lo para monitorar RSSF em cenários reais, pois a redução do consumo de energia da rede de monitoramento contribui para prolongar o tempo de vida desta rede. No próximo capítulo são apresentadas as conclusões finais sobre esta tese de doutorado e os possíveis direcionamentos para trabalhos futuros.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este capítulo é dedicado às considerações finais desta tese de doutorado. A hipótese e as questões de pesquisa são analisadas na seção 6.1. A seção 6.2 apresenta as principais contribuições e os resultados alcançados com esta tese, incluindo as publicações alcançadas no decorrer do curso de doutorado deste candidato. Por fim, a seção 6.3 é dedicada aos possíveis trabalhos futuros.

6.1 Análise da hipótese e das questões de pesquisa

A hipótese e as questões de pesquisa que nortearam o desenvolvimento desta tese de doutorado são analisadas nesta seção.

As questões de pesquisa (QP) são discutidas a seguir:

QP01: *Quais são os principais requisitos que um sistema de monitoramento passivo para RSSF deve possuir?*

A partir da revisão bibliográfica realizada, foram elencados os três requisitos mais relevantes:

- ✓ **Eficiência energética:** Um sistema de monitoramento passivo energeticamente eficiente é necessário quando se deseja monitorar uma RSSF em um cenário real, pois assim é possível monitorar a rede por mais tempo.
- ✓ **Análise online:** A análise *online* das informações obtidas com o monitoramento permite ao gerente da rede ter conhecimento das condições de operação da rede de forma mais rápida e, consequentemente, agir mais rapidamente para corrigir eventuais falhas.
- ✓ **Integração com outras ferramentas de gerência de rede:** A disponibilização das informações obtidas com o monitoramento através de ferramentas de gerência que já são frequentemente utilizadas pelos gerentes de rede facilitam o gerenciamento e a administração da rede monitorada.

QP02: *Quais mecanismos deverão ser utilizados para reduzir o consumo de energia da rede de monitoramento e, consequentemente, prolongar o tempo de vida desta rede sem reduzir a taxa de pacotes capturados da rede alvo?*

Os três mecanismos utilizados pelo EPMOSt para reduzir o consumo de energia da rede de monitoramento são elencados:

- ✓ **Eleição de *sniffers*:** O mecanismo de eleição garante que durante a maior parte do tempo apenas um *sniffer* captura os pacotes enviados por um determinado nó da rede alvo, reduzindo assim a transmissão de pacotes capturados redundantes através da rede de monitoramento e, consequentemente, reduzindo consideravelmente o consumo de energia desta rede.
- ✓ **Agregação de cabeçalhos:** Quando a agregação de cabeçalhos é utilizada, apenas as informações presentes nos cabeçalhos dos pacotes capturados são enviadas através da rede de monitoramento. Desta forma, os cabeçalhos de vários pacotes podem ser enviados na mesma mensagem de monitoramento, reduzindo assim o *overhead* de transmissão e, consequentemente, reduzindo o consumo de energia da rede de monitoramento.
- ✓ **Ligar/desligar o monitoramento:** O EPMOSt possui um mecanismo para ligar e desligar o monitoramento com o intuito de reduzir o consumo de energia da rede monitoramento. Desta forma, o usuário do sistema pode ativar ou desativar o monitoramento remotamente (a partir do servidor), como também pode agendar remotamente os períodos de tempo em que a rede alvo deverá ser monitorada.

Os resultados dos experimentos realizados em um *testbed* e com o simulador demonstram que a utilização destes mecanismos reduz consideravelmente o consumo de energia da rede de monitoramento.

QP03: *Como o sistema proposto deverá disponibilizar as informações obtidas com o monitoramento de modo a facilitar a administração da rede alvo?*

As ferramentas de gerência baseadas no protocolo SNMP são bastante conhecidas e utilizadas pelos administradores de rede. Além disso, diversas destas ferramentas podem ser instaladas de forma gratuita em computadores, *smartphones* ou *tablets*. Por estes motivos, o EPMOSt disponibiliza as informações obtidas com o monitoramento através de um agente SNMP. O agente SNMP permite integrar o EPMOSt com ferramentas de gerência que suportem tal protocolo, facilitando a administração da rede alvo.

Após as questões de pesquisa serem respondidas, a validade da hipótese de pesquisa investigada nesta tese de doutorado é analisada a partir dos resultados obtidos através dos experimentos realizados.

Hipótese: É possível propor e desenvolver um sistema de monitoramento passivo energeticamente eficiente para RSSF mantendo a taxa percentual de pacotes capturados da rede alvo nos mesmos patamares das taxas obtidas pelos sistemas de monitoramento passivo existentes na literatura, tornando possível o monitoramento de redes de sensores sem fio *in situ* durante um longo período de tempo.

Resultado: Validada. Os resultados dos experimentos apresentados no capítulo 5 demonstram que o EPMOSt, quando comparado com os sistemas de monitoramento propostos nos trabalhos relacionados (que não utilizam nenhum mecanismo de eleição de *sniffers* e de agregação de cabeçalhos), reduz consideravelmente o consumo de energia da rede de monitoramento em todos os cenários avaliados. Esta redução no consumo de energia alcança até 87,1% no cenário com 50 nós da rede alvo e 25 *sniffers*. Apesar do EPMOSt alcançar uma taxa percentual de pacotes capturados da rede alvo ligeiramente menor do que os trabalhos relacionados, esta taxa aumenta com o aumento da quantidade de *sniffers* e apresenta uma redução de apenas 0,62% nos experimentos realizados utilizando o *testbed* quando a rede de monitoramento possui 11 *sniffers*. Nos experimentos realizados utilizando o simulador, o mecanismo de eleição de *sniffers* utilizado pelo EPMOSt provoca uma redução no percentual de pacotes capturados inferior a 1,76% em todos os cenários avaliados. Portanto, estes resultados comprovam a viabilidade de utilizar o EPMOSt para monitorar redes de sensores sem fio *in situ* durante um longo período de tempo.

6.2 Resultados alcançados e principais contribuições

Conforme mencionado anteriormente, esta tese de doutorado propõe o EPMOSt, um sistema de monitoramento passivo para RSSF que reduz o consumo de energia da rede de monitoramento e que disponibiliza as informações obtidas com o monitoramento através de um agente SNMP.

Os principais resultados alcançados com esta tese de doutorado foram:

- O sistema de monitoramento EPMOSt, cuja principal vantagem é reduzir o consumo de energia da rede de monitoramento e, consequentemente, beneficiar a rede alvo pelo monitoramento por mais tempo. Essa redução no consumo de energia é alcançada,

principalmente, pela utilização dos mecanismos de eleição de *sniffers* e de agregação de cabeçalhos propostos no capítulo 4.

- A especificação de uma MIB *(Management Information Base*) para sistemas de monitoramento passivo para RSSF; e
- O desenvolvimento de um agente SNMP, que permite integrar o sistema de monitoramento EPMOSt com ferramentas de gerência de rede que suportem o protocolo SNMP, facilitando a administração da rede alvo.

Na seção 3.7 é apresentado um quadro comparativo (Tabela 3.3) dos sistemas de monitoramento passivo analisados nesta tese de doutorado (SNTS, SNIF, Pimoto, LiveNet e PMSW). A Tabela 6.1 mostra os mesmos dados exibidos na Tabela 3.3, porém são acrescentados os dados referentes ao EPMOSt, realçando as principais contribuições desta tese para o monitoramento passivo de redes de sensores sem fio.

Tabela 6.1 – Trabalhos relacionados vs. EPMOSt

Característica	**Trabalhos relacionados**					**EPMOSt**
	SNTS	**SNIF**	**Pimoto**	**LiveNet**	**PMSW**	
Eficiência Energética	Não	Não	Não	Não	Não	Sim
Modo de Análise	Offline	Online	Online	Offline	Online	Online
Pacotes Capturados	Dados + Controle	Dados + Controle	Dados + Controle	Dados	Dados + Controle (ACK)	Dados + Controle
Análise de Eventos	Análise de falhas	Análise de falhas	Não possui	Não possui	Análise de falhas e desempenho	Análise de falhas e desempenho
Pacotes Redundantes	Sim	Sim	Não	Sim	Sim	Sim, mas são raros
Ferramenta de Visualização	Desenvolvida pelos autores	Desenvolvida pelos autores	Plugin do Wireshark	Desenvolvida pelos autores	Desenvolvida pelos autores	Agente SNMP

Observando-se a Tabela 6.1 pode-se constatar que:

i. Apenas o EPMOSt é energeticamente eficiente. Os resultados dos experimentos realizados demonstram que o EPMOSt reduz consideravelmente o consumo de energia da rede de monitoramento em todos os cenários avaliados.

ii. O SNIF, o Pimoto, o PMSW e o EPMOSt analisam os pacotes capturados de forma *online*, enquanto que no SNTS e no LiveNet os pacotes capturados são armazenados em memórias não voláteis para posteriormente serem analisados (*offline*).

iii. O SNTS, o SNIF, o Pimoto e o EPMOSt capturam pacotes de dados e de controle (roteamento, eleição de líderes, ACK, etc.) enviados pela rede alvo. O Livenet captura apenas os pacotes de dados, e o PMSW captura pacotes de dados e ACK.

iv. O SNTS, o SNIF, o PMSW e o EPMOSt analisam os dados capturados com o intuito de detectar eventos de falha ou desempenho da rede alvo enquanto que o Pimoto e o Livenet apenas mostram os *traces* dos pacotes capturados.

v. Apenas o Pimoto não captura pacotes redundantes, pois a rede alvo é subdividida em ilhas de monitoramento e cada ilha possui apenas um *sniffer*, que é responsável por capturar os pacotes enviados pelos nós da sua ilha. Conforme explicado na seção 4.2.1.1, o mecanismo de eleição de *sniffers* utilizado pelo EPMOSt garante que durante a maior parte do tempo apenas um *sniffer* captura os pacotes enviados por um determinado nó da rede alvo.

vi. Apenas o EPMOSt disponibiliza as informações obtidas com o monitoramento através de um agente SNMP. Desta forma, ferramentas de gerência que utilizem o protocolo SNMP podem se comunicar com o agente e exibir as informações obtidas a partir do monitoramento da rede alvo.

Os resultados obtidos no decorrer do desenvolvimento desta tese foram publicados na forma de artigos científicos. A Tabela 6.2 apresenta, em ordem cronológica, as publicações produzidas que são relacionadas diretamente com esta tese de doutorado.

O primeiro artigo apresenta as ideias iniciais desta tese de doutorado e foi apresentado em um simpósio de doutorado. Esse artigo propõe a arquitetura de um sistema de monitoramento passivo energeticamente eficiente para RSSF, mas não implementa nenhum dos módulos da arquitetura proposta.

O segundo artigo descreve a versão inicial do sistema de monitoramento EPMOSt proposto nesta tese de doutorado, onde alguns módulos do sistema foram especificados, implementados e validados. Este artigo apresenta também alguns experimentos realizados na plataforma MicaZ e os resultados obtidos já demonstravam a eficiência energética do EPMOSt.

Tabela 6.2 – Artigos científicos produzidos relacionados diretamente com a tese

No.	Referência	Participação	Tipo	Qualis
1	**GARCIA, F. P.**, SOUZA, J. N., ANDRADE, R. M. C. (2012) "An energy-efficient passive monitoring system for wireless sensor networks" In: IEEE Sustainable Internet and ICT for Sustainability (SustainIT). IEEE: Pisa, 2012, pp. 1-3.	Autor principal	Simpósio de doutorado	-
2	**GARCIA, F. P.**, SOUZA, J. N., ANDRADE, R. M. C. (2013) "Sistemas de Monitoramento Passivo para RSSF – Soluções Existentes e uma Nova Proposta Energeticamente Eficiente" In: XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), pp. 179-192.	Autor principal	Conferência	B2
3	**GARCIA, F. P.**, ANDRADE, R. M. C., OLIVEIRA, C. T., SOUZA, J. N. (2014) "Evoluindo um Sistema de Monitoramento Passivo Energeticamente Eficiente para Redes de Sensores Sem Fio" In: XIX Workshop de Gerência e Operação de Redes e Serviços (WGRS), pp. 13-16.	Autor principal	Workshop	B5
4	**GARCIA, F. P.**, ANDRADE, R. M. C., OLIVEIRA, C. T., SOUZA, J. N. (2014) "EPMOSt: An Energy-Efficient Passive Monitoring System for Wireless Sensor Networks" In: Sensors Journal, vol. 14, pp. 10804-10828.	Autor principal	Periódico	A2

O terceiro artigo descreve uma versão evoluída do EPMOSt, onde é adicionado um agente SNMP a fim de integrar este sistema de monitoramento com ferramentas de gerência que suportem tal protocolo, facilitando a administração da rede alvo.

O quarto artigo descreve e avalia todos os módulos do EPMOSt (vide Figura 4.2). Além disso, foram realizados novos experimentos, utilizando *testbed*, em diversos cenários e os resultados obtidos comprovam a viabilidade de se utilizar o EPMOSt para monitorar RSSF em cenários reais.

Os resultados dos experimentos realizados utilizando o simulador COOJA ainda serão publicados em um novo artigo.

As publicações apresentadas na Tabela 6.3 foram elaboradas juntamente com um aluno de mestrado e não fazem parte das contribuições específicas desta tese de doutorado. Entretanto, foram úteis para entender o processo de desenvolvimento de aplicações para RSSF utilizando nós sensores da plataforma MicaZ.

Tabela 6.3 – Artigos científicos secundários produzidos

No.	Referência	Participação	Tipo	Qualis
1	CAVALCANTE, T. M., **GARCIA, F. P.**, GOMES, D. G., E ANDRADE, R. M. C. (2011) "Avaliação de Desempenho dos Algoritmos Criptográficos Skipjack e RC5 para Redes de Sensores Sem Fio" In: Simpósio Brasileiro de Computação Ubíqua e Pervasiva (SBCUP), pp.1103-1112.	Coautor	Conferência	B5
2	CAVALCANTE, M. T., **GARCIA, F. P.**, ANDRADE, R. M. C. (2012) "Avaliação de Desempenho de Mecanismos de Segurança para Redes de Sensores Sem Fio" In: XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), pp. 277-290.	Coautor	Conferência	B2

6.3 Trabalhos futuros

As contribuições apresentadas nesta tese de doutorado trazem perspectivas interessantes para futuras pesquisas, as quais são identificadas a seguir:

- **Alteração do mecanismo de eleição de *sniffers*:** conforme descrito na seção 4.2.1.1, o mecanismo de eleição de *sniffers* especificado no EPMOSt leva em consideração apenas o nível de potência do sinal recebido (i.e., RSSI). Um dos trabalhos futuros consiste em alterar este mecanismo de eleição para considerar também a qualidade do enlace (LQI - *Link Quality Indicator*), a energia residual da bateria dos *sniffers* e a quantidade de nós monitorados por cada *sniffer*. A utilização conjunta dos valores de RSSI e LQI permite estimar de forma mais precisa a qualidade dos sinais de rádio recebidos pelos *sniffers*, enquanto que a utilização do nível de energia da bateria dos *sniffers* e da quantidade de nós monitorados por cada *sniffer* tem o objetivo de balancear o consumo de energia dos *sniffers* e evitar que alguns *sniffers* tenham sua energia esgotada bem antes de outros.
- **Avaliação dos tempos de vida da rede de monitoramento e da rede alvo:** os resultados dos experimentos realizados apresentados no capítulo 5 comprovam que o EPMOSt reduz consideravelmente o consumo de energia da rede de monitoramento e, consequentemente, prolonga o tempo de vida desta rede. Assim, a rede alvo é beneficiada pelo monitoramento por mais tempo. Um possível direcionamento de pesquisa consiste em realizar novos experimentos com o intuito de mensurar e avaliar

os tempos de vida da rede de monitoramento e da rede alvo. Desta forma, seria possível determinar por quanto tempo a rede de monitoramento consegue monitorar a rede alvo em determinados cenários, facilitando assim a tomada de decisões durante a implantação de novas RSSF em ambientes reais.

- **Avaliação do EPMOSt em outras plataformas de sensores:** em todos os experimentos apresentados nesta tese de doutorado, a rede alvo é composta por nós sensores da plataforma MicaZ. Novos experimentos poderão ser realizados para avaliar a utilização do EPMOSt no monitoramento de redes alvo constituídas por nós de outras plataformas de RSSF, tais como BTnodes [BTnodes 2014], Sun SPOT [Sun SPOT 2014] e Arduino [Arduino 2014], com o intuito de mensurar a eficiência energética do sistema de monitoramento proposto também nestas plataformas.

- **Avaliação da influência do roteamento no consumo de energia dos *sniffers*:** os experimentos realizados mediram a energia consumida pelos *sniffers* para a transmissão dos pacotes capturados, porém os *sniffers* também consomem energia durante a transmissão e recepção de pacotes de roteamento. Um dos trabalhos futuros seria investigar a influência do roteamento no consumo de energia dos *sniffers*.

- **Ativação/desativação automática do monitoramento:** conforme descrito na seção 4.2.5, o EPMOSt possui um mecanismo, que executa no servidor, para ativar e desativar o monitoramento com o intuito de reduzir o consumo de energia dos *sniffers*. Outro trabalho futuro seria ativar e desativar o monitoramento automaticamente de acordo com as informações inferidas sobre o ciclo de trabalho da rede alvo. Por exemplo, ao analisar o histórico dos pacotes capturados, o servidor poderia perceber que a rede alvo envia pacotes apenas em determinados períodos do dia. Neste caso, o servidor poderia automaticamente ativar o monitoramento um pouco antes do início de cada período e desativar um pouco depois do final de cada período.

- **Avaliação do consumo de energia de cada camada da rede de monitoramento:** novos experimentos poderão ser realizados com o intuito de avaliar separadamente o consumo de energia das camadas de enlace, de rede e de aplicação dos *sniffers*. A partir dos resultados obtidos através destes experimentos, poderão ser propostas novas estratégias para reduzir o consumo de energia da rede de monitoramento.

- **Definição da quantidade ideal de *sniffers*:** os experimentos apresentados nesta tese de doutorado mediram a energia consumida pelos *sniffers* para a transmissão dos

pacotes capturados e a porcentagem de pacotes distintos capturados pelos *sniffers*, porém não foi realizada nenhuma análise para determinar a quantidade ideal de *sniffers* necessários para monitorar a rede alvo. Portanto, novos experimentos poderão ser realizados com o intuito de determinar a relação entre a quantidade de nós da rede alvo e a quantidade ideal de nós da rede de monitoramento.

- **Avaliação da escalabilidade do EPMOSt:** Os experimentos apresentados nesta tese de doutorado foram realizados em cenários com no máximo 52 nós da rede alvo e 25 *sniffers*. Portanto, o simulador COOJA poderá ser utilizado para simular o funcionamento do mecanismo de eleição de *sniffers* em uma rede com uma maior densidade, com o intuito de avaliar a escalabilidade do EPMOSt.

REFERÊNCIAS BIBLIOGRÁFICAS

Arduino (2014) “Arduino”, http://www.arduino.cc, Agosto.

Awad, A., Nebel, R., German, R. and Dressler, F. (2008) “On the need for passive monitoring in sensor networks” In: Proceedings of the 2008 11th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools. Washington: IEEE Computer Society, pp. 693-699.

Azure (2014) “Microsoft Azure”, https://azure.microsoft.com/pt-br/, Agosto.

Borges, V. C. M., Curado, M. and Monteiro, E. (2011) “Cross-layer routing metrics for Mesh networks: current status and research directions” In: Computer Communications, Vol. 34, pp. 681–703.

Borges Neto, J. B. and Andrade, R. M. C. (2010) “Wireless sensor networks advances for ubiquitous computing”. In: Mendes Neto, F. M., Ribeiro Neto, P. F. (Org.). Designing solutions-based ubiquitous and pervasive computing: new issues and trends. USA: Hershey-PA: IGI Global, pp. 175-189.

BTnodes (2014) “BTnodes - A Distributed Environment for Prototyping Ad Hoc Networks”, http://www.btnode.ethz.ch, Agosto.

Cavalcante, M. T., Garcia, F. P., Andrade, R. M. C. (2012) “Avaliação de Desempenho de Mecanismos de Segurança para Redes de Sensores Sem Fio” In: XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), pp. 277-290.

Chen, B. R., Peterson, G., Mainland, G. and Welsh, M. (2008) “LiveNet: using passive monitoring to reconstruct sensor network dynamics” In: Distributed Computing in Sensor Systems, pp. 79-98.

Chen, Z. and Shin, K. G. (2009) “Post-deployment performance debugging in wireless sensor networks,” In: 30th IEEE Real-Time Systems Symposium (RTSS ’09), pp. 313-322.

Cheour, R., Lahmar, K. and Abid, M. (2011) “Evolution of wireless sensor networks and necessity of power management technique” In: IEEE Faible Tension Faible Consommation (FTFC), pp. 75-78.

Comer, Douglas E. (2006) “Automated network management systems: currente and future capabilities” Upper Saddle River: Pearson Prentice Hall, 2006.

Crossbow (2013) “Crossbow”, http://bullseye.xbow.com:81/index.aspx, Julho.

Elkhodr, M., Shahrestani, S. and Cheung, H. (2013) “The Internet of Things: Vision & Challenges” In: TENCON Spring Conference, pp. 218-222.

Garcia, F. P., Souza, J. N., Andrade, R. M. C. (2012) “An energy-efficient passive monitoring system for wireless sensor networks” In: IEEE Sustainable Internet and ICT for Sustainability (SustainIT). IEEE: Pisa, 2012, pp. 1-3.

Garcia, F. P., Souza, J. N., Andrade, R. M. C. (2013) “Sistemas de monitoramento passivo para RSSF – soluções existentes e uma nova proposta energeticamente eficiente” In: XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), pp. 179–192.

Garcia, F. P., Andrade, R. M. C., Oliveira, C. T., Souza, J. N. (2014) “EPMOSt: An Energy-Efficient Passive Monitoring System for Wireless Sensor Networks” In: Sensors Journal, vol. 14, pp. 10804-10828.

Gay, D., Levis, P., Behren, R. V., Welsh, M., Brewer, E. and Culler, D. (2003) “The NesC Language: A Holistic Approach to Networked Embedded System”. In: ACM SIGPLAN 2003 conference on Programming language design and implementation, pp. 1-11.

Girod, L., Ramanathan, N., Elson, J., Stathopoulos, T., Lukac, M. and Estrin, D. (2007) “Emstar: a software environment for developing and deploying heterogeneous sensor–actuator networks” In: ACM Transactions on Sensor Networks, Vol. 3, No.3, Article 13.

Hanninen, M., Suhonen, J., Hamalainen, T. D. And Hannikainen, M. (2011) “Practical monitoring and analysis tool for WSN testing” In: IEEE Conference on Design and Architectures for Signal and Image Processing (DASIP), pp. 1-8.

HART (2011) “HART Communication Foundation - WirelessHART Technology”, http://www.hartcomm.org/protocol/wihart/wireless_technology.html. Maio de 2014.

Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D. and Pister, K. (2000) “System Architecture Directions for Networked Sensors” In: Proceedings of ACM ASPLOS IX International Conference On Architectural Support For Programming Languages And Operating Systems, pp. 93-104.

IEEE (2007) “IEEE Standard for Information Technology – Telecommunications and Information Exchange Between Systems – Local and Metropolitan Area Networks – Specific Requirements – Part 11: Wireless LAN Medium Access Control and Physical Layer Specifications”, IEEE Std.

ITU (2012) "Global Standards for the Internet of Things". International Telecommunications Union.

ITU-T (1992) “Rec. X.721, Information Technology-Open Systems Interconnection-Structure Of Management Information: Definition Of Management Information”. International Telecommunications Union.

ITU-T (1995) "CCITT Rec. M.3010: Principles for a Telecommunications Management Network”. International Telecommunications Union.

Jacquot, A., Chanet, J., Mean Hou, K., Diao, X. and Li, Jian-Jin. (2009) “LiveNCM : A new wireless management tool” In: 9th IEEE AFRICON, pp. 1–6.

Jiangwu, N., Huadong, M. and Lufeng, M. (2012) “Passive Diagnosis for WSNs Using Data Traces” In: 8th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS), pp. 273–280.

Jurdak, R., Ruzzelli, A. G. and O'Hare, G. (2008) “Adaptive radio modes in sensor networks: How deep to sleep?” In: IEEE Communications Society Conference on Ad Hoc and Sensor Networks, pp. 386-394.

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

Khan, M. M. H., Luo, L., Huang, C. and Abdelzaher, T. (2007) “SNTS: sensor network troubleshooting suite” In: Proceedings of the 3rd IEEE International Conference on Distributed Computing in Sensor Systems. Berlin: Springer, pp. 142-157.

Kitchenham, B. (2004) “Procedures for Performing Systematic Reviews” In: Joint Technical Report Software Engineering Group, Department of Computer Science Keele University, United King and Empirical Software Engineering, National ICT Australia.

Kunzel, G., Winter, J. M., Muller, I., Pereira, C. E. and Netto, J. C. (2012) “Passive Monitoring Software Tool for Evaluation of Deployed WirelessHART Networks” In: Brazilian Symposium on Computing System Engineering (SBESC). Natal, Brazil: pp. 7-12.

Kurose, J. F. and Ross, K. W. (2013) “Redes de computadores e a Internet: uma abordagem top-down”. 6 ed. São Paulo: Pearson Education do Brasil.

Liu, Y., Liu, K. and Li, M. (2010) “Passive Diagnosis for Wireless Sensor Networks” In: IEEE ACM Transactions on Networking, Vol. 18, No. 4, pp. 1132-1144.

Loureiro, A. A. F., Nogueira, J. M. S., Ruiz, L. B., Mini, R. A. F., Nakamura, E. F. e Figueiredo, C. M. S. (2003) “Redes de Sensores Sem Fio” In: Simpósio Brasileiro de Redes de Computadores.

ManageEngine (2014) “ManageEngine MIB Browser Free Tool”, http://www.manageengine.com/products/mibbrowser-free-tool. Janeiro.

McCloghrie, K., Rose, M. (1991) “Management Information Base for Network Management of TCP/IP-based internets: MIB-II. RFC 1213”, pp. 1–70.

MIB Browser (2013) “SNMP MIB Browser Android Tool”, http://www.manageengine.com/free-snmp-mibbrowser-android. Outubro.

MPR-MIB (2013) “MPR-MIB Users Manual”, http://www-db.ics.uci.edu/pages/research/quasar/MPR-MIB%20Series%20User%20Manual%207430-0021-06_A.pdf: CrossBow Technology Inc., Maio.

MySQL (2013) “MySQL”, http://mysql.org. Novembro.

Nagios (2013) “Nagios“, http://www.nagios.org: Nagios Enterprises, Fevereiro.

Net-SNMP (2013) “Net-SNMP”, http://www.net-snmp.org. Fevereiro.

Osterlind, F., Dunkels, A., Eriksson, J., Finne, N. and Voigt, T. (2006) “Cross-Level Sensor Network Simulation with COOJA” In: 31st IEEE Conference on Local Computer Networks, pp. 641-648.

Ramanathan, N., Chang, K., Kapur, R., Girod, L., Kohler, E. and Estrin, D. (2005) “Sympathy for the Sensor Network Debugger” In: ACM Conference on Embedded Networked Sensor Systems (SenSys), pp. 255-266.

RFC 1157 (1990) “Simple Network Management Protocol – SNMP” In: Internet Engineering Task Force (IETF).

Rocha, A. R., Pirmez, L., Delicato, F. C., Lemos, E., Santos, I., Gomes, D. G., Souza, J. N. (2012) “WSNs clustering based on semantic neighborhood relationships“ In: Elsevier Computer Networks, vol. 56, pp. 1627-1645.

Ribeiro, L. B., Castro, M. F. (2011) “BiO4SeL: Um Algoritmo Autonômico Biologicamente Inspirado para Roteamento em Redes de Sensores Sem Fio” In: XXXVIII Seminário Integrado de Software e Hardware (SEMISH), pp. 1207-1220.

Ringwald, M. and Romer, K. (2007) “Deployment of sensor networks: problems and passive Inspection” In: Proceedings of the 5th IEEE Workshop on Intelligent Solutions in Embedded Systems. New York: IEEE, pp. 179-192.

Romer, K. and Ringwald, M. (2008) “Increasing the visibility of sensor networks with passive distributed assertions” In: Proceedings of the Workshop on Real-World Wireless Sensor Networks (REALWSN '08). New York: ACM, pp. 36-40.

Ruiz, L. B., Correia, L. H. A., Vieira, L. F. M., Macedo, D. F., Nakamura, E. F., Figueiredo, C. M. S., Vieira, M. A. M., Mechelane, E. H. , Camara, D., Loureiro, A. A. F., Nogueira, J. M. S. and Silva Jr, D. C. (2004) “Arquiteturas para Redes de Sensores Sem Fio“ In: Minicurso no Simpósio Brasileiro de Redes de Computadores (SBRC).

Schmidt, Kevin J. (2001) “SNMP Essencial”. 1 ed. Rio de Janeiro: Campus.

Sohraby, K., Minoli, D. and Znati, T. (2007) “Wireless sensor networks Technology, protocols, and applications”. New Jersey: John Wiley & Sons.

Sommer, P. and Kusy, B. (2013) “Minerva: distributed tracing and debugging in wireless sensor networks” In: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys '13). New York: ACM.

SQL (2014) “SQL Clauses”, http://www.sqlclauses.com. Outubro.

Stallings, William (2005) “Redes e sistemas de comunicação de dados”. 4 ed. Rio de Janeiro: Campus.

Start (2014) “StArt = State of the Art through Systematic Review”, http://lapes.dc.ufscar.br/tools/start_tool, Abril.

Suhonen, J., Kohvakka, M., Hännikäinen, M., and Hämäläinen, T. D. (2008) “Embedded software architecture for diagnosing network and node failures in wireless sensor networks,” In: Embedded Computer Systems: Architectures, Modeling, and Simulation, vol. 5114 of Lecture Notes in Computer Science, pp. 258–267.

Sun SPOT (2014) “Sun SPOT World”, http://www.sunspotworld.com/, Agosto.

Swami, A., Zhao, Q., Hong, Y. and Tong, L. (2007) “Wireless sensor networks, signal processing and communications perspectives” New Jersey: John Wiley & Sons.

Tolle, G. and Culler, D. (2005) “Design of an application-cooperative management system for wireless sensor networks,” In: Proceeedings of the 2nd European Workshop on Wireless Sensor Networks, pp. 121-132.

UAT Robotics (2014) “UAT Robotics and Embedded Systems“, http://uatrobotics.blogspot.com.br. Dezembro.

Vlavianos, A., Law, L. K., Broustis, I., Krishnamurthy, S. V. and Faloutsos, M. (2008) “Assessing Link Quality in IEEE 802.11 Wireless Networks: Which is the Right Metric?” In: IEEE Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 1-6.

WebNMS (2013) “WebNMS SNMP Agent Toolkit Java Edition”, http://www.webnms.com/javaagent/. Maio de 2014.

Weiser, M. (1993) “Some computer science issues in ubiquitous computing” In: Communications of the ACM 36(7), pp. 75-84.

Wireshark (2013) “Wireshark”, http://www.wireshark.org, Fevereiro.

Xu, X., Tong, C. and Wan, J. (2010) “Improve the Completeness of Passive Monitoring Trace in Wireless Sensor Network” In: IEEE Asia-Pacific Services Computing Conference (APSCC), pp. 560-566.

Xu, X., Wan, J., Zhang, W., Tong, C. and Wu C. (2011) “PMSW: a passive monitoring system in wireless sensor networks” In: International Journal of Network Management, vol. 21, pp. 300-325.

Yang, Y., Xu, Y. and Li, X. (2011) “Lossy nodes inference based on end-to-end passive monitoring inwireless sensor networks” In: High Technology Letters, vol. 17, pp. 388–394.

Ye, J., Zhao, Z., Li, H. and Chen, H. (2011) “Hierachical heterogeneous wireless sensor network management system” In: IEEE Conference on Wireless Communications and Signal Processing (WCSP), pp. 1–5.

Yu, M., Mokhtar, H. and Merabti, M. (2006) “A Survey of Network Management Architecture in Wireless Sensor Network” http://www.cms.livjm.ac.uk/pgnet2006/Programme/Papers/2006-093.pdf, Março de 2012.

Zhang, B. and Li, G. (2009) “Survey of Network Management Protocols in Wireless Sensor Network” In: IEEE International Conference on E-Business and Information System Security (EBISS), pp. 1-5.