Sistemas de Monitoramento Passivo para RSSF – Soluções Existentes e uma Nova Proposta Energeticamente Eficiente

Fernando P. Garcia[1,2,3], José Neuman de Souza[1,2,a], Rossana M. C. Andrade[1,2,b]

Universidade Federal do Ceará (UFC)
[1]Mestrado e Doutorado em Ciência da Computação (MDCC)
[2]Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat)
[3]Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)
fernandoparente@ifce.edu.br, neuman@ufc.br, rossana@ufc.br

Resumo. *Sistemas de monitoramento são importantes para depurar e analisar o funcionamento de uma rede de sensores sem fio (RSSF) em operação. No monitoramento passivo, uma rede de monitoramento adicional é implantada juntamente com a rede que deve ser monitorada (chamada de rede alvo). Esta rede de monitoramento captura e analisa os pacotes transmitidos pela rede alvo. Quando se deseja monitorar continuamente uma RSSF em um cenário 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. Neste artigo, inicialmente, nós identificamos, analisamos e comparamos os principais sistemas de monitoramento passivo propostos para RSSF. Durante as nossas pesquisas, não identificamos nenhum sistema de monitoramento passivo que se preocupasse em reduzir o consumo de energia da rede de monitoramento. Sendo assim, este artigo propõe um sistema de monitoramento passivo energeticamente eficiente para RSSF. Experimentos foram realizados na plataforma MicaZ com alguns módulos implementados do sistema e os resultados já demonstram a eficiência energética do sistema de monitoramento proposto.*

Abstract. *Monitoring systems are important for debugging and analyzing wireless sensor networks (WSN). In passive monitoring, a monitor network needs to be deployed in addition to the target network. This monitor network captures and analyzes packets transmitted by the target network. An energy-efficient passive monitoring system is necessary when we have to monitor a WSN in a real scenario over long periods; otherwise the lifetime of the network monitor can be shorter than the lifetime of the target network. In this paper, initially, we identify, analyze and compare the main passive monitoring systems proposed for WSN. During our research, we did not identify any passive monitoring system for WSN that aims to reduce the energy consumption of the monitor network. Therefore, we propose an energy-efficient passive monitoring system for WSN. Experiments were performed in the MicaZ platform with some modules and their results already show the energy efficiency of the proposed monitoring system.*

[a] Bolsista de produtividade D-1 do CNPq
[b] Bolsista de produtividade DT-2 do CNPq

1. Introdução

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, entre outras. Em geral, as RSSF são compostas por nós sensores de tamanho reduzido operados por baterias e que utilizam comunicação sem fio de pequeno alcance. Além disso, estas redes possuem severas restrições de consumo de energia, capacidade de processamento, capacidade de memória e largura de banda [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, *loops* 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. 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. No monitoramento passivo, uma rede de monitoramento adicional é implantada juntamente com a rede que deve ser monitorada (**rede alvo)**. A rede de monitoramento captura e analisa os pacotes transmitidos pela rede alvo, não consumindo nenhum recurso da rede alvo. Portanto, quando é necessário reduzir a utilização de recursos da rede alvo, é melhor utilizar um sistema de monitoramento passivo. 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.

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 importante quando se deseja monitorar continuamente uma RSSF em um cenário real ("*in situ*"), pois caso contrário a rede de monitoramento pode ter um tempo de vida bem menor do que a rede alvo. 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.

Diante deste contexto, inicialmente, nós identificamos, analisamos e comparamos os principais sistemas de monitoramento passivo propostos para RSSF. Durante as pesquisas realizadas, não foi identificado nenhum sistema de monitoramento passivo para RSSF que se preocupasse em reduzir o consumo de energia da rede de monitoramento. Portanto, este trabalho propõe um sistema de monitoramento passivo energeticamente eficiente para RSSF, cujo principal objetivo é reduzir o consumo de energia da rede de monitoramento, e consequentemente prolongar o seu tempo de vida.

O restante deste artigo está organizado da seguinte forma: Na seção 2, os principais sistemas de monitoramento passivo propostos na literatura para RSSF são analisados e comparados. A seção 3 apresenta e discute o sistema de monitoramento proposto neste trabalho. Na seção 4 os detalhes de implementação de alguns módulos do sistema proposto são abordados. A seção 5 descreve os experimentos realizados com esses módulos, e apresenta e discute os resultados obtidos. As conclusões e trabalhos futuros são apresentados na seção 6.

2. Monitoramento passivo em redes de sensores sem fio

Após uma revisão bibliográfica em artigos publicados nas bibliotecas digitais *IEEE Xplore Digital Library* (http://ieeexplore.ieee.org/xplore) e *ACM Digital Library* (http://dl.acm.org) com a data de publicação a partir do ano de 2007 foram selecionados cinco sistemas de monitoramento passivo propostos especificamente para RSSF: 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].

2.1. SNTS (*Sensor Network Troubleshooting Suite*)

No SNTS [Khan et al. 2007], nós *sniffers* ouvem passivamente o canal de comunicação e coletam os pacotes enviados pelos nós da rede alvo. 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 *clock*. Após o período de captura dos dados, os *sniffers* são manualmente recolhidos e os registros dos pacotes capturados são transferidos para um computador (PC). Após os registros serem armazenados no PC, 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 *clock* ajustado com o *clock* de um nó base. Ao final da captura dos dados, o *clock* de cada *sniffer* é comparado com o *clock* do nó base e os *timestamps* de seus registros são ajustados de acordo com a diferença entre estes *clocks*. Após o ajuste de *clock*, os registros duplicados são removidos. Para analisar os dados, os autores desenvolveram uma ferramenta utilizando técnicas de aprendizagem de máquina. Antes de executar a ferramenta é necessário configurar regras que definam o comportamento esperado da rede alvo.

O propósito do SNTS é ajudar 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 em que seja impraticável recolher os *sniffers*, como por exemplo, aplicações militares ou aplicações para monitoramento ambiental (e.g., florestas, oceanos, etc.).

2.2. SNIF (*Sensor Network Inspection Framework*)

Em [Ringwald and Romer 2007], os autores propõem um framework de inspeção passiva denominado SNIF. No SNIF, uma rede de monitoramento sem fio, denominada pelos autores de *deployment support network* (DSN), é implantada juntamente com a rede alvo. Os pacotes capturados pelos nós DSN são marcados com um *timestamp* e encaminhados até um computador onde são ordenados pelo *timestamp* e os pacotes

duplicados 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. Finalmente, as informações obtidas são mostradas em uma interface gráfica desenvolvida pelos autores.

O SNIF não possui nenhum mecanismo de sincronização dos *clocks* dos nós DSN, podendo então ocasionar erros na ordenação dos pacotes capturados e na remoção de pacotes duplicados. Além disso, a falta de sincronização pode comprometer a análise e a precisão das informações de monitoramento obtidas.

2.3. Pimoto

No Pimoto [Awad et al. 2008], a rede alvo é subdividida em "ilhas de monitoramento". Em cada ilha de monitoramento é implantado um *sniffer*, que é responsável por capturar os pacotes enviados pelos nós da sua ilha e enviar estes pacotes diretamente para um gateway (computador) através de um rádio Bluetooth. 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. O servidor analisa e mostra os pacotes capturados na ferramenta Wireshark [Wireshark 2012] utilizando um *plugin* desenvolvido pelos autores.

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 usuário. Além disso, a utilização do Pimoto pode ser inviável para RSSF com muitos nós distribuídos em uma área geográfica grande, pois neste caso é necessária uma infraestrutura composta por vários gateways interligados ao servidor.

2.4. LiveNet

Em [Chen et al, 2008] os autores propõem o LiveNet, um conjunto de ferramentas e técnicas para registrar o comportamento de uma RSSF. O LiveNet é constituído por três componentes principais: uma infraestrutura de monitoramento passiva composta por nós *sniffers* que coletam e armazenam os pacotes 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. Desta forma, torna-se inviável utilizar o LiveNet para monitorar RSSF implantadas em cenários em que seja impraticável recolher as memórias *flash* ou utilizar uma rede cabeada para enviar os dados capturados, como por exemplo aplicações militares ou aplicações para monitoramento ambiental. 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.

2.5. PMSW (*Passive Monitoring System in Wireless Sensor Networks*)

No PMSW [Xu et al. 2011], nós *sniffers* são implantados na área de monitoramento juntamente com os nós sensores da rede alvo. Cada *sniffer* captura 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*. Em alguns cenários pode ser necessário implantar *gateways* em diferentes partes da rede, pois o alcance do rádio dos *sniffers* é limitado. 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 *clock* do *gateway*. Em seguida, cada *gateway* envia o *trace* gerado para um servidor através de uma rede TCP/IP.

O servidor recebe os *traces* gerados por todos os *gateways*, e faz o *merging* dos *traces* recebidos, gerando assim um único *trace* global. Após o *merging*, o servidor executa um algoritmo de inferência para inferir pacotes não capturados pelos *sniffers*. Em seguida, é realizada a análise do *trace* com o intuito de avaliar o desempenho e detectar eventuais falhas da rede alvo. Para a visualização das informações obtidas a partir do monitoramento, 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 pacotes de controle, tais como pacotes de roteamento e de eleição de *cluster*, não são capturados nem analisados. Além disso, o ajuste de *clock* dos *gateways* não é suficiente para garantir a remoção de todos os pacotes duplicados 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 pelo *gateway* em momentos distintos.

2.6. Análise comparativa

Nesta seção é realizada uma análise comparativa entre os sistemas de monitoramento passivo abordados neste artigo levando-se em consideração os seguintes aspectos:

- **Inferência -** capacidade do sistema em recuperar pacotes não capturados pela rede de monitoramento a partir dos pacotes capturados;
- **Modo de análise -** informa se a análise dos pacotes capturados pelo sistema é realizada de modo *online* ou *offline*;
- **Pacotes capturados -** relaciona quais tipos de pacotes são capturados pelo sistema;
- **Eficiência energética -** verifica se o sistema se preocupa em minimizar o consumo de energia dos nós da rede de monitoramento;
- **Mecanismo de sincronização** - descreve o mecanismo de sincronização utilizado pelo sistema para inserir a marca de tempo nos pacotes capturados;
- **Análise de eventos -** descreve quais informações de monitoramento são obtidas ao se analisar os pacotes capturados; e
- **Ferramenta de visualização -** tipo da ferramenta utilizada para a visualização das informações obtidas.

Tabela 1 – Comparação dos sistemas de monitoramento.

	Inferência	Modo de Análise	Pacotes Capturados	Eficiência Energética
SNTS [Khan et al. 2007]	Não	Offline	Dados + Controle	Não
SNIF [Ringwald and Romer 2007]	Não	Online	Dados + Controle	Não
PIMOTO [Awad et al. 2008]	Não	Online	Dados + Controle	Não
LiveNet [Chen et al, 2008]	Não	Offline	Dados	Não
PMSW [Xu et al. 2011]	Sim	Online	Dados + ACK	Não

Tabela 2 – Comparação dos sistemas de monitoramento (continuação).

	Mecanismo de Sincronização	Análise de Eventos	Ferramenta Visualização
SNTS [Khan et al. 2007]	Ajuste de clock	Diagnóstico de falhas na camada de rede	Desenvolvida pelos autores
SNIF [Ringwald and Romer 2007]	Nenhum	Diagnóstico de falhas *cross-layer*	Desenvolvida pelos autores
PIMOTO [Awad et al. 2008]	Ajuste de clock	Não possui	Wireshark
LiveNet [Chen et al, 2008]	Clock dos sniffers ajustados na implantação da rede	Não possui	Desenvolvida pelos autores
PMSW [Xu et al. 2011]	Ajuste de clock	Análise de falhas e desempenho	Desenvolvida pelos autores

As Tabelas 1 e 2 mostram um quadro comparativo dos sistemas de monitoramento discutidos neste artigo. As principais considerações obtidas observando-se os dados destas tabelas são:

i. Nenhum dos sistemas é energeticamente eficiente;
ii. Apenas o PMSW tem a capacidade de inferir pacotes não capturados;
iii. Nenhum dos sistemas implementa mecanismo de sincronização nos *sniffers* (nós da rede de monitoramento). O Livenet sincroniza os *sniffers* somente na implantação da rede. O SNTS, o Pimoto e o PMSW utilizam uma estratégia de ajuste de *clock*, contudo, esta estratégia não é tão precisa quanto à sincronização dos *sniffers*.
iv. O SNTS, o SNIF e o PWSW analisam os dados capturados com o intuito de detectar eventos de falha ou desempenho da rede monitorada, enquanto que o Pimoto e o Livenet apenas mostram os *traces* dos pacotes capturados; e
v. Apenas o Pimoto exibe as informações de monitoramento em uma ferramenta de gerência de rede utilizada pela comunidade (i.e., wireshark).

3. O Sistema de monitoramento proposto

Conforme mencionado na seção 1, um sistema de monitoramento passivo energeticamente eficiente é importante 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, nós propomos um sistema de monitoramento passivo para RSSF que reduz o consumo de energia da rede de monitoramento. Além disso, o sistema proposto implementa um mecanismo de sincronização nos *sniffers* e disponibiliza as informações de monitoramento através de um agente SNMP (*Simple Network Management Protocol*). A sincronização dos *sniffers* é importante para garantir a precisão dos *timestamps* dos pacotes capturados, possibilitando assim uma análise de dados mais precisa. A disponibilização das informações obtidas com o monitoramento através de um agente SNMP permite integrar o sistema proposto com ferramentas de gerência livres e gratuitas que suportam o protocolo SNMP, tais como Nagios e Net-SNMP.

A Figura 1 mostra a visão geral do sistema de monitoramento proposto, onde uma rede de monitoramento é 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*, gera um arquivo de *trace* (*trace* local) com as informações dos pacotes capturados e envia o *trace* local, através de uma rede IP, para o monitor global. O monitor global recebe os *traces* de um ou mais monitores locais e gera um único *trace* (*trace* global), que é analisado para se obter diversas informações sobre a rede alvo.

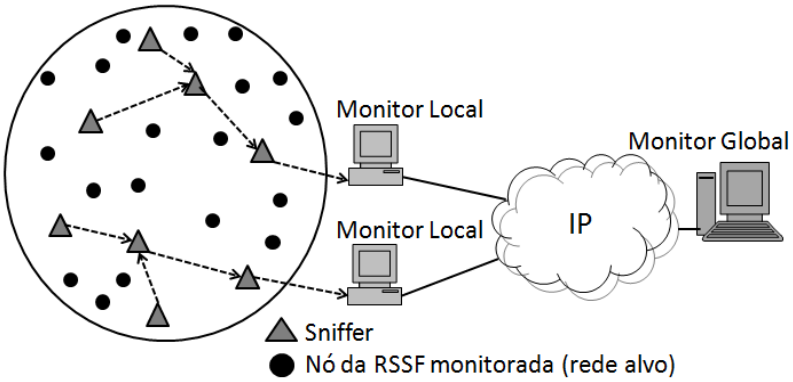


Figura 1 – Visão geral do sistema de monitoramento proposto.

A Figura 2 mostra o diagrama de atividades do sistema proposto, no qual os pacotes de um determinado nó da rede alvo são capturados por apenas um *sniffer* com o intuito de evitar a transmissão de pacotes duplicados e, consequentemente, reduzir o consumo de energia da rede de monitoramento. Para tanto, faz-se necessário implementar 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 4.1.

Ao **capturar um pacote** da rede alvo, o *sniffer* **insere um *timestamp*** neste pacote. Para garantir a precisão dos *timestamps*, faz-se necessário utilizar um mecanismo para a sincronização dos *sniffers* (**Sinc Sniffers**). Após capturar alguns pacotes, o *sniffer* pode utilizar um mecanismo para agregar os cabeçalhos (**Agrega *Headers***) 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.

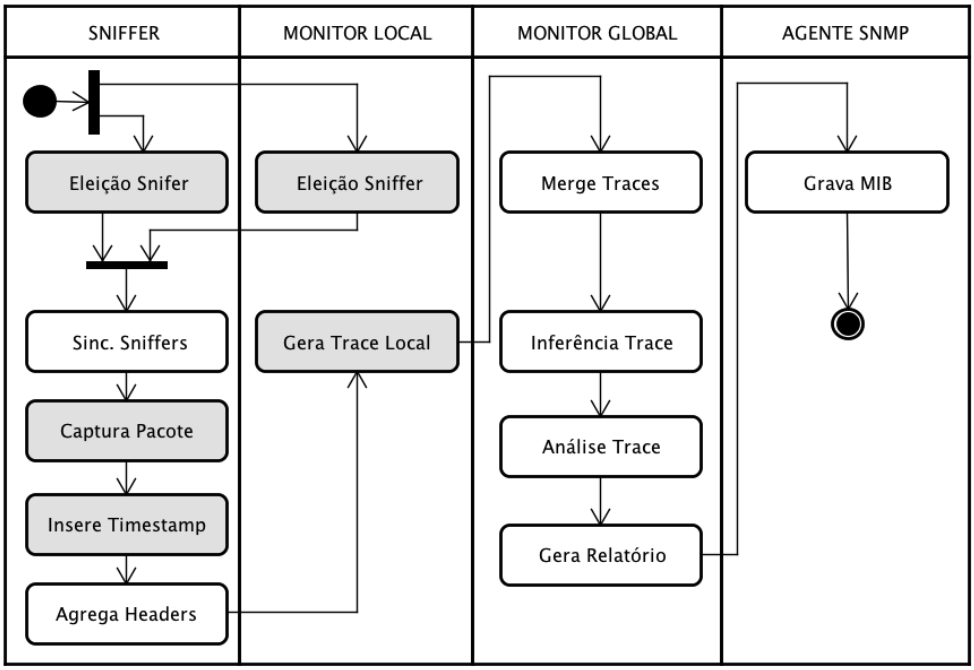


Figura 2 – Diagrama de atividades do sistema de monitoramento proposto.

O monitor local recebe as mensagens de monitoramento enviadas pelos *sniffers*, **gera o trace local** com todos os pacotes capturados e envia o *trace* para o monitor global 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 monitor global.

O monitor global recebe os *traces* enviados pelos monitores locais e faz o *merge* destes *traces* (**Merge Traces**), gerando um único *trace* global. A partir do *trace* global, o monitor global usa mecanismos de inferência (**Inferência Trace**) para inferir alguns pacotes que não foram capturados pelos *sniffers*. [Xu et al. 2011] propõem algoritmos para inferir pacotes não capturados pelos *sniffers*, que podem ser utilizados na implementação do sistema proposto.

Em seguida é realizada a **análise do trace** para se obter informações sobre a rede alvo (descoberta de topologia, perda de pacotes, morte e reinicialização de nós, etc.). Estas informações são utilizadas para **gerar um relatório** que será exibido para o usuário do sistema e são também **gravadas em uma MIB** (*Management Information Base*) 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.

O sistema proposto 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 monitor global os períodos de tempo em que a rede alvo será monitorada. O monitor global 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*. Ao receber um comando *off*, o *sniffer* comuta para o modo *sleep*, reduzindo assim o consumo de energia em até 95% [Jurdak et al. 2008]. Periodicamente, o *sniffer* comuta para o modo de recepção para verificar se o monitor local enviou um comando *on*. Se receber um comando *on*, o *sniffer* permanece no estado de recepção e reinicia o processo de captura dos pacotes, e caso contrário retorna ao modo *sleep*.

Em resumo, o sistema de monitoramento passivo proposto neste trabalho tem como principal objetivo prolongar o tempo de vida da rede de monitoramento, podendo tornar possível o monitoramento de RSSF *in situ* durante um longo período de tempo ou até mesmo durante todo o seu tempo de vida.

4. Implementação de módulos do sistema de monitoramento proposto

Os módulos pintados com a cor cinza na Figura 2 (**Eleição Sniffer**, **Captura Pacote**, **Insere Timestamp** e **Gera Trace Local**) foram implementados, e serão discutidos e validados neste artigo para mostrar a eficiência energética do sistema de monitoramento proposto. O restante do sistema proposto ainda está em fase de desenvolvimento.

4.1. Sniffers

Nesta implementação, a rede de monitoramento utiliza como *sniffers* nós da plataforma MicaZ, desenvolvida pela *Crossbow Technology*. Esta plataforma foi escolhida por ser muito utilizada em aplicações de RSSF de uma maneira geral e por ser utilizada em outros trabalhos de RSSF do grupo de pesquisa ao qual este trabalho está vinculado ([Rocha et al. 2012] e [Cavalcante et al. 2012]). A aplicação de monitoramento embarcada nos *sniffers* foi desenvolvida utilizando a linguagem de programação nesC e executa sobre o sistema operacional TinyOS.

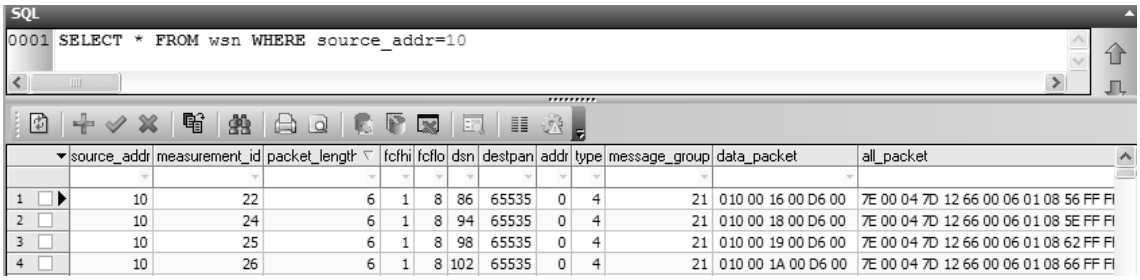
Após a implantação (*deploy*) da rede de monitoramento, os *sniffers* e o monitor local iniciam o mecanismo **Eleição Sniffer** para eleger quais nós da rede alvo terão seus pacotes capturados por quais *sniffers*. Este mecanismo é executado quando um *sniffer* captura pela primeira vez um pacote de um determinado nó da rede alvo e leva em consideração o nível de potência do sinal recebido (RSSI). 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 a potência do sinal recebido. 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. 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 potência de sinal. Caso $\mathbf{S_Y}$ esteja recebendo o sinal de **A** com maior potência 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**. Porém, se $\mathbf{S_X}$ estiver recebendo o sinal de **A** com maior potência do que $\mathbf{S_Y}$, 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**. Desta forma, o mecanismo de eleição garante que apenas um *sniffer* captura os pacotes enviados por um determinado nó da rede alvo, evitando assim a transmissão de pacotes capturados redundantes através da rede de monitoramento.

Durante o processo de monitoramento, 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 explicado anteriormente. Após capturar um pacote, o *sniffer* insere uma marca de tempo (*timestamp*) e envia o pacote capturado para o monitor local através da rede de monitoramento utilizando roteamento *multihop* (o pacote é encaminhado do nó de origem ao nó de destino passando por nós intermediários).

4.2. 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.). O monitor local comunica-se com a rede de monitoramento através de uma estação base. Conforme explicado na seção 4.1, o monitor local executa o mecanismo de eleição juntamente com os *sniffers*. Além disso, o monitor local recebe os pacotes enviados pelos *sniffers* e **gera o trace local** com os pacotes da rede alvo capturados. Nesta implementação, o trace local é armazenado em uma tabela do banco de dados MySQL [MySQL 2012]. O MySQL foi utilizado neste trabalho por ser um sistema de gerenciamento de banco de dados bastante difundido e ser um software livre com licença GPL (*General Public License*). 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*). Para exemplificar, a Figura 3 exibe alguns pacotes enviados pelo nó da rede alvo cujo endereço (*source_addr*) é 10.


SQL

```
0001 SELECT * FROM wsn WHERE source_addr=10
```

	source_addr	measurement_id	packet_length	fcfhi	fcflo	dsn	destpan	addr	type	message_group	data_packet	all_packet
1	10	22	6	1	8	86	65535	0	4	21	010 00 16 00 D6 00	7E 00 04 7D 12 66 00 06 01 08 56 FF FI
2	10	24	6	1	8	94	65535	0	4	21	010 00 18 00 D6 00	7E 00 04 7D 12 66 00 06 01 08 5E FF FI
3	10	25	6	1	8	98	65535	0	4	21	010 00 19 00 D6 00	7E 00 04 7D 12 66 00 06 01 08 62 FF FI
4	10	26	6	1	8	102	65535	0	4	21	010 00 1A 00 D6 00	7E 00 04 7D 12 66 00 06 01 08 66 FF FI

Figura 3 – Exibição de pacotes capturados.

5. Experimentos

Esta seção descreve e analisa os experimentos realizados para validar os módulos do sistema de monitoramento proposto cujas implementações foram descritas na seção 4.

5.1. Descrição dos experimentos

Os experimentos foram realizados utilizando 28 nós MicaZ com sistema operacional TinyOS. A plataforma MicaZ possui como principais características: microprocessador

ATMEGA128L, 4KB de memória RAM, 128KB de memória ROM e transceptor de rádio frequência CC2420.

A Figura 4 mostra o cenário utilizado para a realização dos experimentos. A rede alvo é composta por 22 nós, sendo 21 nós sensores e 01 nó sorvedouro. Os nós sensores executam uma aplicação que a cada minuto mede a temperatura do ambiente e envia para o nó sorvedouro através de roteamento *multihop*. 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 05 sniffers e 01 estação base. Os *sniffers* capturam os pacotes enviados pelos nós da rede alvo e enviam para a estação base. A estação base envia os pacotes recebidos dos *sniffers*, através de um cabo USB, para um notebook que executa a aplicação do monitor local. A aplicação do monitor local recebe os pacotes enviados pela estação base e armazena em um banco de dados MySQL. É importante ressaltar que a estação base tem como função principal intermediar o envio de pacotes entre os *sniffers* e o monitor local, e não captura nenhum pacote da rede alvo.

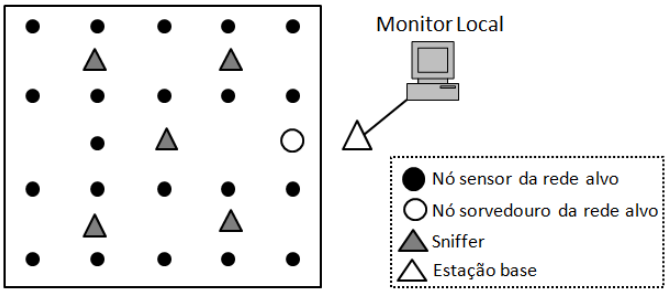


Figura 4 – Cenário utilizado nos experimentos.

Foram realizados dois tipos de experimentos: “Com Eleição” e “Sem Eleição”. No experimento “Com Eleição”, os sniffers executam a aplicação descrita na seção 4.1, onde é implementado o mecanismo de eleição proposto neste trabalho.

No experimento “Sem Eleição”, os sniffers não possuem nenhum mecanismo de eleição e capturam todos os pacotes dos nós da rede alvo que estão na área de cobertura dos seus rádios. Em seguida, os sniffers enviam os pacotes capturados para o monitor local, que então armazena no banco de dados. Vale ressaltar que esta é a estratégia utilizada por todos os sistemas de monitoramento descritos na seção 2.

Para a avaliação dos experimentos, foram definidas as seguintes métricas: quantidade de pacotes enviados pela rede alvo (*PenvAlvo*), quantidade de pacotes distintos capturados (*Pcapturados*), quantidade de pacotes não capturados (*PnãoCapturados*), quantidade de pacotes redundantes capturados (*Predundantes*) e energia consumida na transmissão dos pacotes capturados (*Et*).

No sistema proposto, assim como em todos os cinco sistemas analisados na Seção 2, os *sniffers* “escutam” todos os pacotes que trafegam nas suas interfaces de rádio. Portanto, a energia consumida na recepção de pacotes no sistema proposto é similar à energia consumida nos sistemas analisados, e por isso não foi utilizada como métrica de avaliação.

A quantidade total de pacotes enviados pelos 21 nós sensores 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*.

$$PenvAlvo = \sum_{i=1}^{21} (contMediçãoFinal\_i - contMediçãoInicial\_i + 1) \quad (1)$$

A quantidade de pacotes distintos do nó *i* capturados pela rede de monitoramento (*PcapturadosNó_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 é obtida através da Equação 2. Portanto, a quantidade de pacotes não capturados pela rede de monitoramento é determinada pela Equação 3.

$$Pcapturados = \sum_{i=1}^{21} PcapturadosNó\_i \quad (2)$$

$$PnãoCapturados = PenvAlvo - Pcapturados \quad (3)$$

Dois ou mais pacotes são considerados redundantes quando possuem o mesmo endereço do nó e mesmo contador de medição de temperatura. Portanto, a quantidade de pacotes redundantes do nó *i* (*PredundantesNó_i*) é determinada verificando-se quais pacotes deste nó possuem o mesmo contador de medição. Logo, a quantidade total de pacotes redundantes capturados é obtida através da Equação 4.

$$Predundantes = \sum_{i=1}^{21} PredundantesNó\_i \quad (4)$$

Para calcular a energia consumida pelos *sniffers* 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]. 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.

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

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]. Nos experimentos realizados, cada pacote enviado pelos *sniffers* tem tamanho (*Plength*) de 23 bytes, sendo 07 bytes de *header* e 16 bytes referentes ao pacote enviado pelo nó da rede alvo. Substituindo-se estes valores na Equação 5, obtém-se a Equação 6.

$$Et = 38.42 \times 10^{-6} \times Psent \quad (6)$$

A quantidade de pacotes enviados pelos *sniffers* é determinada pela Equação 7. Então, a energia consumida pelos *sniffers* na transmissão dos pacotes capturados da rede alvo é determinada pela Equação 8.

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

$$Et = 38.42 \times 10^{-6} \times (Pcapturados + Predundantes) \quad (8)$$

5.2. Resultados e discussão

Para cada tipo de experimento foram realizados 10 experimentos com duração de 15 minutos. Os resultados mostrados na Tabela 3 referem-se aos valores médios dos 10 experimentos realizados com intervalo de confiança de 95%.

Tabela 3 – Resultados dos experimentos.

Tipo de experimento	PenvAlvo	Pcapturados	PnãoCapturados	Predundantes	Et (mJ)
Com eleição	318 ± 2	298 ± 4	20 ± 3	0	11.44±0.16
Sem eleição	320 ± 5	306 ± 5	14 ± 1	403 ± 10	27.23±0.54

Pode-se observar na Tabela 3 que ao se utilizar o mecanismo de eleição proposto neste trabalho, a quantidade de pacotes redundantes capturados pela rede de monitoramento é zero, pois cada nó da rede alvo tem seus pacotes capturados por apenas um *sniffer*. Entretanto, 20 dos 318 pacotes enviados pela rede alvo não são capturados pela rede de monitoramento, o que corresponde a 6.28%.

Quando não é utilizado o mecanismo de eleição dos *sniffers*, a quantidade de pacotes não capturados é de apenas 4.38% (14 pacotes), pois o mesmo pacote pode ser capturado por mais de um *sniffer*, reduzindo assim a probabilidade de não capturá-lo. No entanto, foram capturados 403 pacotes redundantes, que corresponde a uma média de 1.26 (403/320) pacotes redundantes para cada pacote da rede alvo capturado.

Pode-se observar também na Tabela 3 que quando não é utilizado o mecanismo de eleição, cada *sniffer* consome, a cada 15 minutos, em média 27.23 mJ para a transmissão dos pacotes capturados. Quando o mecanismo de eleição é utilizado, este consumo de energia é de apenas 11.44 mJ. Isto significa uma redução de 58% da energia consumida pelos *sniffers* para a transmissão dos pacotes capturados, prolongando assim o tempo de vida da rede de monitoramento.

6. Conclusões e trabalhos futuros

Neste trabalho, inicialmente nós analisamos e comparamos cinco sistemas de monitoramento passivo propostos para RSSF: SNTS, SNIF, Pimoto, LiveNet e PMSW. Entretanto, nenhum destes sistemas de monitoramento passivo se preocupa em reduzir o consumo de energia da rede de monitoramento. Diante deste contexto, nós propomos um sistema de monitoramento passivo energicamente eficiente para RSSF. Neste artigo, nós apresentamos e validamos os módulos já implementados do sistema. Experimentos foram realizados na plataforma MicaZ e os resultados demonstraram que o mecanismo de eleição utilizado no nosso sistema reduz em até 58% a energia consumida pelos *sniffers* para a transmissão dos pacotes capturados, prolongando assim o tempo de vida da rede de monitoramento. Como trabalhos futuros teremos a implementação e a validação dos demais módulos do sistema proposto. Além disso, nós pretendemos alterar o mecanismo de eleição para levar em consideração, além da potência do sinal recebido, o nível de energia da bateria dos *sniffers* e a quantidade de nós monitorados por cada *sniffer*, com o intuito de balancear o consumo de energia dos *sniffers* e evitar que alguns *sniffers* tenham sua energia esgotada bem antes de outros.

Agradecimentos

Este trabalho é um resultado parcial do projeto UbiStructure financiado pelo CNPq (MCT / CNPq 14/2011 - Universal) sob o número de protocolo 481417/2011-7.

Referências

Awad, A., Nebel, R., German, R. and Dressler, F. (2008) "On the need for passive monitoring in sensor networks" In: IEEE Euromicro Conference on Digital System Design Architectures, Methods and Tools.

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.

Hänninen, 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.

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.

Khan. M. M. H., Luo, L., Huang, C. and Abdelzaher, T. (2007) "SNTS: sensor network troubleshooting suite" In: 3rd IEEE International Conference on Distributed Computing in Sensor Systems, Springer, Berlin.

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.

MySQL (2012) "MySQL", http://mysql.org. Novembro.

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

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.

Wireshark (2012) "Wireshark", http://www.wireshark.org, Outubro.

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.