Avaliação de um Sistema de Monitoramento Passivo para RSSF: Experimentos com Simulador ou *Testbed*?

Fernando P. Garcia[1,2,3], Rossana M. C. Andrade[1,2], Gabriel Braz Goulart[2], Francisco Gonçalves de A. Filho[2], José Neuman de Souza[1,2]

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, rossana@ufc.br, {gabrielgoulart, franciscoalmeida}@great.ufc.br, neuman@ufc.br

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 (a rede alvo). O EPMOSt (**E**nergy-efficient **P**assive **MO**nitoring **S**yst**e**m) é um sistema de monitoramento passivo que usa um mecanismo de eleição de sniffers (nós da rede de monitoramento) para reduzir o consumo de energia da rede de monitoramento, prolongando o seu tempo de vida. Neste artigo, o mecanismo de eleição de sniffers proposto no EPMOSt é avaliado através da realização de experimentos com um simulador de RSSF com o objetivo de comparar com os resultados obtidos a partir de experimentos realizados anteriormente em um testbed. Espera-se com esta comparação analisar estas duas formas de experimentação e propor um direcionamento para a realização de experimentos em RSSF.*

Abstract. *Monitoring systems are important for debugging and analyzing Wireless Sensor Networks (WSNs). In passive monitoring, a monitoring network is deployed in order to capture and analyze packets sent by the network to be monitored (target network). EPMOSt is an **E**nergy-efficient **P**assive **MO**nitoring **S**yst**e**m for WSNs that uses a sniffer election mechanism to reduce the energy consumption of the monitoring network. In this way, the lifetime of the monitoring network is extended. In this paper, the sniffer election mechanism proposed in EPMOSt is evaluated by conducting experiments with a WSN simulator in order to compare with the results obtained from experiments performed in a testbed. We aim at analyzing both ways of WSNs evaluation in order to point out better directions for experiments in WSNs.*

1. Introdução

Os avanços recentes nas áreas de microeletrônica, sensoriamento e comunicação sem fio propiciaram o surgimento e a evolução das RSSF (Redes de Sensores Sem Fio). Aplicações propostas para RSSF incluem detecção sísmica, monitoramento ambiental, casas inteligentes, entre outras. Em geral, as RSSF são compostas por nós sensores de

tamanho reduzido alimentados por baterias e que utilizam comunicação sem fio de pequeno alcance. Além disso, estas redes possuem severas restrições de energia, processamento, memória e largura de banda.

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].

No monitoramento passivo, uma rede de monitoramento adicional é implantada juntamente com a rede que deve ser monitorada (a rede alvo). 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.

Diante deste contexto, em [Garcia et al. 2014] nós propusemos o sistema de monitoramento passivo **EPMOSt** (***E**nergy-efficient **P**assive **MO**nitoring **S**ys**t**em*), cujo principal objetivo é reduzir o consumo de energia da rede de monitoramento. Desta forma, o tempo de vida da rede de monitoramento é prolongado e, consequentemente, a rede alvo é beneficiada pelo monitoramento por mais tempo. O EPMOSt reduz o consumo de energia da rede de monitoramento devido, principalmente, à utilização de um mecanismo de eleição de *sniffers* (nós da rede de monitoramento), o qual 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 o consumo de energia desta rede.

No presente artigo, o mecanismo de eleição de *sniffers* proposto no EPMOSt é avaliado através da realização de experimentos com um simulador de RSSF. Os resultados obtidos através dessas simulações são descritos, analisados e comparados neste artigo com os resultados apresentados em [Garcia et al. 2014], os quais foram obtidos a partir de experimentos realizados com sensores reais (*testbed*). As diferenças entre os resultados alcançados através do *testbed* e da simulação são também discutidas com o intuito de analisar estas duas formas de experimentação e propor um direcionamento para a realização de experimentos em RSSF.

O restante deste artigo está organizado da seguinte forma: A Seção 2 apresenta o sistema de monitoramento EPMOSt. A Seção 3 descreve os experimentos realizados para avaliar o mecanismo de eleição de *sniffers* proposto no EPMOSt, bem como apresenta e discute os resultados obtidos. A Seção 4 aborda alguns trabalhos relacionados. Por fim, as conclusões e trabalhos futuros são apresentados na Seção 5.

2. O EPMOSt

Em [Garcia et al. 2014] nós propusemos o sistema de monitoramento EPMOSt. A Figura 1 mostra a visão geral do 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 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* localizado no servidor. O servidor 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 de nós, quantidade de pacotes enviados e recebidos por cada nó, entre outras). Estas informações são armazenadas em uma MIB (*Management Information Base*) para serem acessadas por um agente SNMP (*Simple Network Management Protocol*).

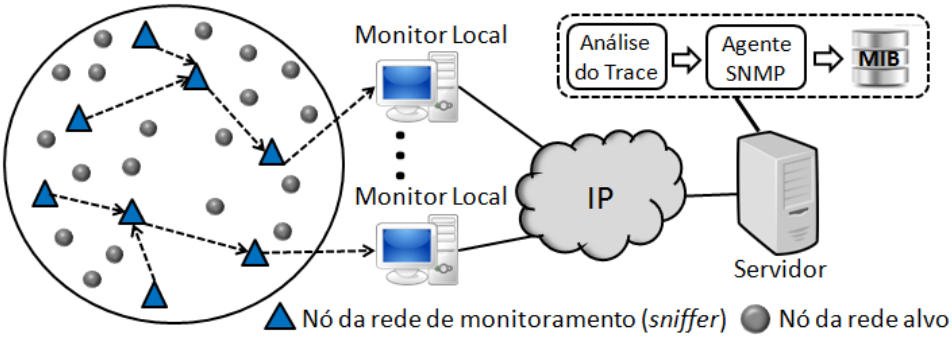

Figura 1. Visão geral do EPMOSt (adaptado de [Garcia et al. 2014]).

A Figura 2 mostra o diagrama de atividades UML (*Unified Modeling Language*) do EPMOSt.

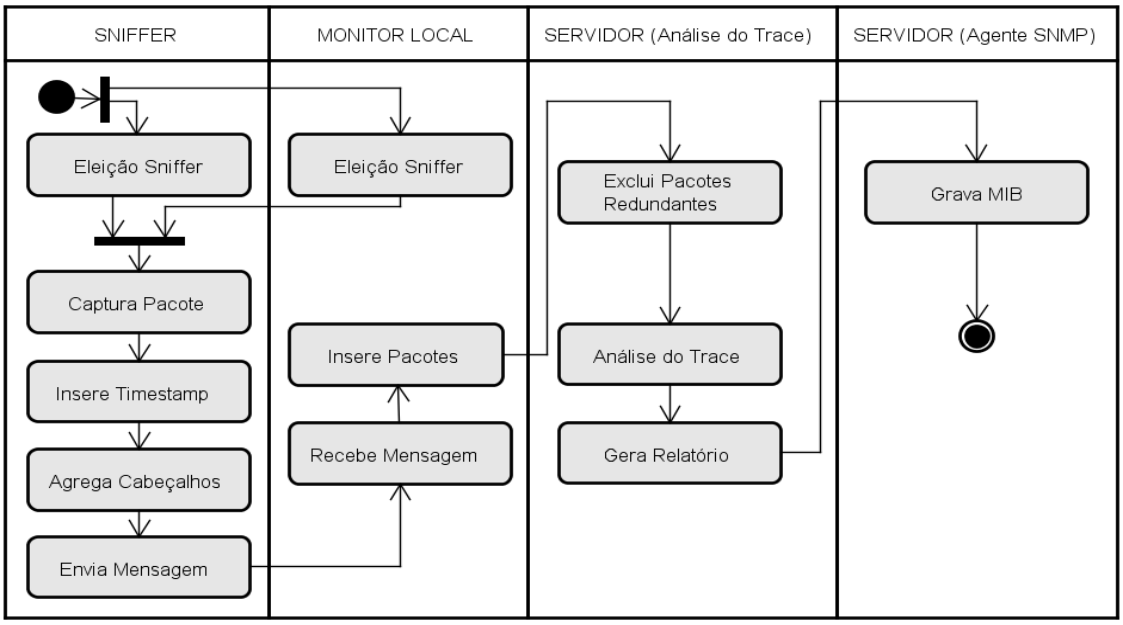

Figura 2. Diagrama de atividades do EPMOSt (adaptado de [Garcia et al. 2014]).

Após a implantação da rede de monitoramento, os *sniffers* e o monitor local iniciam um mecanismo (**Eleição Sniffer**) para eleger quais nós da rede alvo terão seus pacotes capturados por quais *sniffers*. Este mecanismo de eleição é executado quando um *sniffer* captura pela primeira vez um pacote de um determinado nó da rede alvo e leva em consideração o RSSI (*Received Signal Strength Indicator*), que indica o nível de potência do sinal recebido.

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 um pacote de confirmação (ACK) para o monitor local e inicia 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 RSSI. 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**. 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**.

Com a utilização deste mecanismo de eleição, 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 redundantes através da rede de monitoramento e, consequentemente, reduzindo o consumo de energia desta rede.

Após a execução do mecanismo de eleição, os *sniffers* iniciam a **captura de pacotes**, 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. 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 é opcional e tem como objetivo reduzir a quantidade de dados enviados pela rede de monitoramento e, consequentemente, reduzir o consumo de energia desta rede.

O monitor local **recebe as mensagens** de monitoramento enviadas pelos *sniffers* e **insere os pacotes** capturados em um arquivo de *trace* localizado no servidor. O servidor extrai as 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. 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, ferramentas de gerência baseadas no SNMP podem exibir as informações.

3. Experimentos

Em [Garcia et al. 2014] foram realizados experimentos em um *testbed* para avaliar o mecanismo de eleição de *sniffers* proposto no EPMOSt. Este *testbed* utilizou, como *sniffers* e como nós da rede alvo, nós MicaZ. A plataforma MicaZ possui como principais características: microprocessador ATMEGA128L, 4KB de memória RAM, 128KB de memória ROM e transceptor de rádio frequência CC2420 [Crossbow 2014].

No presente trabalho foram realizadas simulações com os mesmos cenários utilizados no *testbed* com o intuito de comparar os resultados obtidos nas simulações

com os resultados obtidos a partir dos experimentos realizados com sensores reais. O simulador COOJA [Osterlind et al. 2006] foi escolhido por ser um simulador de RSSF 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.

3.1. Descrição

As simulações foram realizadas com os mesmos cenários utilizados no *testbed* descrito em [Garcia et al. 2014]. A Figura 3 ilustra um exemplo de cenário utilizado para a realização das simulações. 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 envia um pacote de dados para o nó sorvedouro. 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 (com até três saltos). A estação base envia os pacotes recebidos dos *sniffers* para o monitor local.

Foram realizados experimentos com ***N*** *sniffers* distribuídos na área monitorada. Para cada valor de ***N*** (3, 5, 7, 9 e 11), dois tipos de experimentos foram realizados: "com eleição" e "sem eleição". No experimento "com eleição", os *sniffers* executam o mecanismo de eleição descrito na Seção 2. 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.

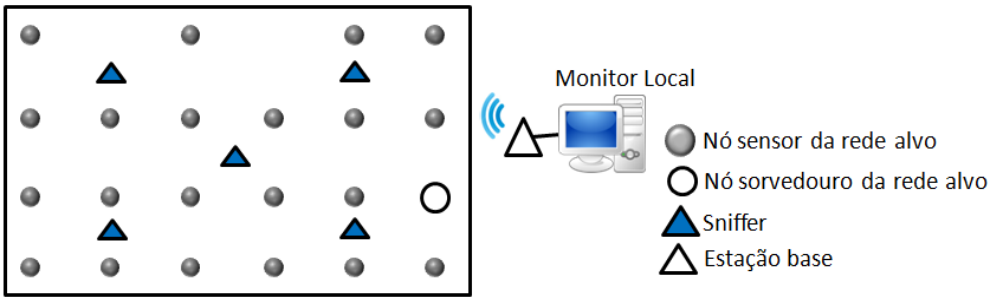

Figura 3. Cenário utilizado nas simulações.

Os experimentos realizados neste trabalho não utilizam a agregação de cabeçalhos, pois têm como principal objetivo avaliar o mecanismo de eleição de *sniffers*. Neste caso, para cada pacote capturado, o *sniffer* envia uma mensagem para o monitor local contendo o seu endereço (*sniffer address*), uma marca de tempo (*timestamp*) e o pacote capturado, conforme mostrado na Figura 4. O cabeçalho (*header*) de 05 bytes é inserido pelo protocolo de enlace da plataforma MicaZ.

Header	Sniffer Address (1 byte)	Timestamp (2 bytes)	Pacote Capturado

Figura 4. Formato do pacote enviado pelos sniffers.

Para a avaliação dos experimentos foram utilizadas as métricas definidas em [Garcia et al. 2014], a seguir: porcentagem de pacotes distintos capturados pela rede de monitoramento (*%PCapturados*) e energia consumida pela rede de monitoramento na transmissão dos pacotes capturados (*Et*).

A porcentagem de pacotes distintos capturados pela rede de monitoramento (*%PCapturados*) é determinada pela Equação 1, onde *Pcapturados* representa a

quantidade total de pacotes distintos capturados pela rede de monitoramento e *PenvAlvo* representa a quantidade total de pacotes enviados pelos nós sensores da rede alvo. Na plataforma MicaZ, os pacotes redundantes podem ser detectados analisando-se o campo DSN (*Destination Sequence Number*) presente no cabeçalho dos pacotes enviados pelos nós sensores. O valor de DSN é incrementado pelo nó de origem a cada pacote enviado [Crossbow 2014]. Portanto, se dois ou mais pacotes possuem o mesmo endereço de origem e o mesmo DSN, significa que se trata do mesmo pacote.

$$\%Pcapturados = 100 * Pcapturados/PenvAlvo \quad (1)$$

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]. Neste modelo, a energia consumida na transmissão (*Et*) é determinada pela Equação 2, 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.

$$Et = Psent\ x\ Plength\ x\ TB\ x\ It\ x\ V \quad (2)$$

A quantidade de pacotes enviados pelos *sniffers* é determinada pela Equação 3. 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 [Crossbow 2014]. Nos experimentos realizados, cada pacote enviado pelos *sniffers* tem tamanho (*Plength*) de **23** bytes, sendo **05** bytes de *header* e **18** bytes da mensagem de monitoramento (vide Figura 4). Substituindo-se estes valores e a Equação 3 na Equação 2, obtém-se a Equação 4.

$$Psent = Pcapturados + Predundantes \quad (3)$$

$$Et = 38{,}42\ x\ 10^{-6}\ x\ (Pcapturados + Predundantes) \quad (4)$$

3.2. Resultados e Discussão

Para cada valor de ***N*** (quantidade de *sniffers*) e para cada tipo de experimento ("com eleição" e "sem eleição") foram realizadas 10 simulações com duração de 15 minutos. Os resultados mostrados nos gráficos das Figuras 5 e 6 referem-se aos valores médios das 10 simulações realizadas com intervalo de confiança de 95%. Os resultados referentes aos experimentos com *testbed* foram extraídos de [Garcia et al. 2014].

A Figura 5 mostra a energia consumida pela rede de monitoramento em função da quantidade de *sniffers*. Pode-se observar que, tanto no *testbed* quanto no simulador, 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. Quando 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 observar também na Figura 5 que as curvas dos experimentos realizados com o simulador e com o *testbed* apresentam comportamentos similares. No entanto, existe uma diferença entre os valores representados por cada curva. Isto acontece porque no simulador, diferentemente do *testbed*, a potência das interfaces de rádio dos *sniffers* não é reduzida quando os *sniffers* ficam mais próximos uns dos outros. Portanto, no simulador, a quantidade de nós da rede alvo que estão na área cobertura de cada *sniffer* não é alterada quando a quantidade de *sniffers* aumenta.

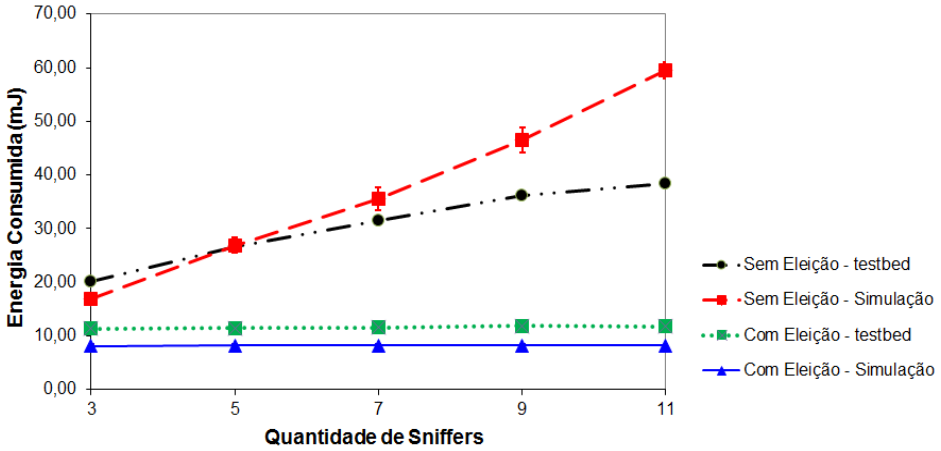

Figura 5. Energia consumida pela rede de monitoramento X quantidade de sniffers.

Pode-se perceber ainda na Figura 5 que no simulador, 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, 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 devido a não redução da potência das interfaces de rádio dos *sniffers*.

A Figura 6 mostra a porcentagem de pacotes distintos capturados pela rede de monitoramento em função da quantidade de *sniffers*. Pode-se observar que no *testbed*, para os tipos de experimentos "sem eleição" e "com eleição", a porcentagem de pacotes capturados aumenta quando a quantidade de *sniffers* 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 no *testbed*, a porcentagem de pacotes capturados é um pouco maior do que nos 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*.

Pode-se observar também na Figura 6 que no simulador, para os tipos de experimentos "sem eleição" e "com eleição", 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*. Acredita-se que esta diferença pode ser atribuída a 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 sem fio e dispositivos com interfaces *Bluetooth*).

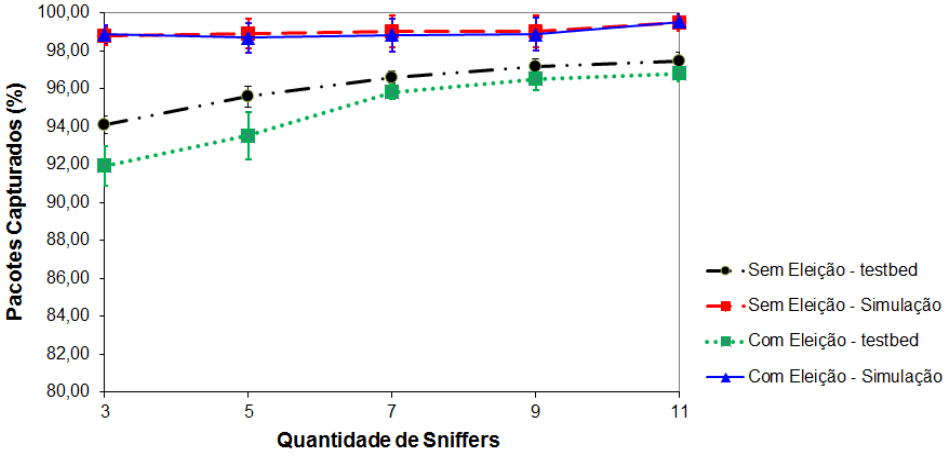

Figura 6. Pacotes capturados pela rede de monitoramento X quantidade de sniffers.

Os resultados obtidos através da realização de experimentos tanto com sensores reais da plataforma MicaZ quanto com o simulador COOJA demonstram que o mecanismo de eleição de *sniffers* proposto no 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.

Os resultados apresentados nesta seção confirmam que um simulador (e.g., COOJA) pode ser utilizado para avaliar o comportamento do mecanismo de eleição de *sniffers* proposto no EPMOSt, facilitando assim eventuais evoluções da proposta visto que o uso de um *testbed* implica em dificuldades adicionais de configuração do ambiente, de tempo e de custo. Além disso, o conhecimento das eventuais diferenças entre os resultados obtidos através do simulador e do *testbed* permite que o simulador possa ser utilizado para avaliar o comportamento do mecanismo de eleição de *sniffers* em cenários com grande quantidade nós, o que seria mais complexo com a utilização de sensores reais.

4. Trabalhos Relacionados

Diversos trabalhos propõem sistemas de monitoramento passivo propostos especificamente para RSSF, entre os quais destacam-se: SNIF [Ringwald and Romer 2007], Pimoto [Awad et al. 2008], LiveNet [Chen et al, 2008], PMSW [Xu et al. 2011] e EPMOSt [Garcia et al. 2014].

No SNIF [Ringwald and Romer 2007], cada *sniffer* possui duas interfaces de rádio, sendo uma usada para capturar os pacotes enviados pelos nós da rede alvo e outra para enviar os pacotes capturados para o nó sorvedouro, onde as informações extraídas dos pacotes são analisadas. O consumo de energia da rede de monitoramento e a porcentagem de pacotes capturados não foram avaliados.

No Pimoto [Awad et al. 2008], a rede alvo é subdividida em ilhas de monitoramento. Em cada ilha é implantado um *sniffer*, que é responsável por capturar os pacotes enviados pelos nós da rede alvo da sua ilha e enviá-los para seu *gateway* usando um rádio *Bluetooth*. O *gateway* envia os pacotes capturados para um servidor, que analisa e exibe os pacotes capturados. No Pimoto, o consumo de energia da rede de monitoramento e a porcentagem de pacotes capturados também não foram avaliados.

No LiveNet [Chen et al. 2008], os pacotes capturados pelos *sniffers* são armazenados em uma memória *flash*. Após a captura dos dados, os *sniffers* são manualmente recolhidos e os registros dos pacotes capturados são transferidos para um computador, onde são analisados. No LiveNet, foram realizados experimentos utilizando *testbed* com nós das plataformas MicaZ e TmoteSky para avaliar a porcentagem de pacotes capturados em função da quantidade de *sniffers*. Os resultados obtidos apresentaram comportamento similar aos resultados mostrados na Figura 6 (sem eleição – *testbed*), pois a porcentagem de pacotes capturados também aumentou quando a quantidade de *sniffers* aumentou. Entretanto, o consumo de energia da rede de monitoramento não foi avaliado.

No PMSW [Xu et al. 2011], cada *sniffer* captura os pacotes enviados pelos nós da rede alvo que estão na sua área de cobertura e envia os pacotes capturados para o seu *gateway*. Ao receber os pacotes capturados, o *gateway* cria um arquivo de *trace* e o envia para o servidor. O servidor analisa os *traces* para extrair informações sobre o comportamento da rede alvo. No PMSW, foram realizados experimentos utilizando *testbed* com nós da plataforma TelosB para avaliar a porcentagem de pacotes capturados em função da quantidade de *sniffers*. Os resultados obtidos também apresentaram comportamento similar aos resultados mostrados na Figura 6 (sem eleição – *testbed*). Entretanto, o consumo de energia da rede de monitoramento também não foi avaliado.

Nos sistemas de monitoramento SNIF, Pimoto, LiveNet e PMSW, 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, e, em seguida, enviam todos os bytes dos pacotes capturados. Portanto, os resultados dos experimentos "sem eleição" apresentados neste artigo podem representar o comportamento destes quatro sistemas de monitoramento.

Em [Garcia et al. 2014] foram realizados experimentos utilizando *testbed* com nós da plataforma MicaZ para avaliar o consumo de energia da rede de monitoramento e a porcentagem de pacotes capturados em cenários com até 11 *sniffers*. Para cada cenário, foram realizados dois tipos de experimento: (*i*) utilizando o mecanismo de eleição de *sniffers* proposto no EPMOSt; (*ii*) sem utilizar o mecanismo de eleição de *sniffers*. Entretanto, assim como os demais trabalhos discutidos nesta seção, não foram utilizadas simulações para avaliar o sistema de monitoramento proposto, o que é o diferencial do presente artigo bem como a análise comparativa dos dois modos de experimentação.

5. Conclusões e Trabalhos Futuros

Em [Garcia et al. 2014] foram realizados experimentos em um *testbed* para avaliar o mecanismo de eleição de *sniffers* proposto no sistema de monitoramento EPMOSt. No presente artigo foram realizados experimentos utilizando o simulador COOJA nos

mesmos cenários utilizados no *testbed* com o objetivo de comparar os resultados obtidos nas simulações com os resultados obtidos a partir dos experimentos realizados com sensores reais. As diferenças entre os resultados alcançados através do *testbed* e da simulação também foram discutidas com o intuito de analisar estas duas formas de experimentação.

Apesar de apresentarem algumas diferenças, os resultados obtidos através do *testbed* e do simulador demonstraram que o mecanismo de eleição de *sniffers* 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 contribuições apresentadas neste trabalho trazem perspectivas interessantes para futuras pesquisas e destacamos três direções principais a seguir. Primeiramente, um simulador (e.g., 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 sua escalabilidade. Em seguida, poderão ser realizadas novas simulações para avaliar também a porcentagem de pacotes capturados pelos *sniffers* em função do tráfego gerado pelos nós da rede alvo. Finalmente, poderão ser realizados novos experimentos utilizando o simulador para avaliar os tempos de vida da rede de monitoramento e da rede alvo, com o intuito de 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.

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.

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.

Crossbow (2014) "MPR-MIB Users Manual - Crossbow Technology", http://www-db.ics.uci.edu/pages/research/quasar/, Novembro.

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.

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.

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.

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

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.