

MODELAGEM, IMPLEMENTAÇÃO E SIMULAÇÃO DE UM PROTOCOLO DE COMUNICAÇÃO EM TEMPO-REAL PARA UMA REDE DE SENSORES SEM FIO

J. P. C. M. OLIVEIRA ¹, S. S. SOUSA ¹, P. R. C. DE ARAÚJO ¹, A. W. O. RODRIGUES ¹, A. L. C. DE ARAÚJO ¹

¹ Instituto Federal de Educação, Ciência e Tecnologia do Ceará, IFCE
Av. 13 de Maio, 2081, Benfica, Fortaleza-CE, CEP:60040-215, Brasil
paolo.oliveira@ifce.edu.br

RESUMO

Rede de Sensores Sem Fio (RSSF) é um conjunto de nós sensores organizados em uma rede cooperativa (HILL et al., 2000), com cada nó possuindo capacidade de processamento, memória, transceptor de RF e baterias. O sensoriamento de cada nó sensor e a comunicação entre eles são utilizados para monitorar e atuar sobre os mais diversos ambientes. O objetivo deste trabalho é modelar, implementar e simular um protocolo de comunicação em tempo-real para uma Rede de Sensores Sem Fio. Neste trabalho, são apresentadas três abordagens: a modelagem da topologia a ser utilizada pela RSSF; a implementação da topologia da RSSF utilizando as plataformas Arduino, como nó sensor, e Raspberry PI, como nó sink ou gateway; e a simulação da RSSF utilizando o simulador OPNET (RIVERBED, 2014). Como resultado, são apresentadas as simulações no OPNET para comprovar a escalabilidade da aplicação, bem como as respostas recebidas pelo nó sink (Raspberry PI) para comprovar a corretude do protocolo desenvolvido.

PALAVRAS-CHAVE (PALABRAS-CLAVE): Protocolos de Comunicação, Rede de Sensores Sem Fio, Sistemas de Tempo-Real, Simulação de Redes.

ABSTRACT

Wireless Sensor Network (WSN) is a collection of sensor nodes organized into a cooperative network (HILL et al., 2000), and each node with processing power, memory, RF transceiver and batteries. The sensing of each sensor node and the communication between them are used to monitor and act on the most diverse environments. The objective of this paper is to model, simulate and implement a real-time communication protocol to a Wireless Sensors Network. In this paper, three approaches are presented: the modeling of the topology to be used by the WSN; the implementation of the WSN topology using the Arduino platform, as a sensor node, and Raspberry PI, as a sink node or gateway; and the simulation of the WSN using the OPNET simulator (RIVERBED, 2014). As a result, this work presents the OPNET simulations to demonstrate the scalability of the application, as well as the responses received by the sink node (Raspberry PI) to prove the correctness of the protocol.

PALAVRAS-CHAVE: Communication Protocols, Wireless Sensor Network, Real-Time Systems, Network Simulation.

1 INTRODUCTION

As RSSFs vêm revolucionando a obtenção e processamento de informações em todas as áreas

da sociedade. Entretanto, devido a sua natureza e a suas aplicações, corrigir erros ou trocar baterias depois da instalação da rede de sensores costuma

ser proibitivo. Assim, são requeridos algoritmos de roteamento e/ou de comunicação, técnicas para auto-organização dos dispositivos e protocolos eficientes para garantir um menor uso de energia da rede de sensores, ampliando assim a sua vida útil. Além da redução do consumo de energia da rede, essas técnicas e algoritmos possibilitam o suporte à latência, bem como atribuir precisão, tolerância a falhas, escalabilidade e disposição dos sensores através da rede.

Devido as suas características, as RSSFs estão sendo utilizadas em diversas áreas da sociedade. Entre elas: medicina, militar, segurança, energia, monitoramento ambiental, alerta de catástrofes, entre outras. Segundo Singh e Pesch (2010), as RSSFs estão sendo muito importantes em aplicações de comunicação sem fio industriais, tais como controle de processos, manufaturas, entre outras, que requerem sistemas de alto desempenho. Muitas dessas áreas requerem aplicações confiáveis e com restrição de tempo. Uma aplicação com restrição de tempo em sua resposta pode ser entendida como uma aplicação de tempo-real, a qual deve responder a um estímulo de entrada em um tempo finito e especificado. Singh e Pesch (2010) concluem que em aplicações industriais, a troca de dados deve ser rápida, confiável e determinística, com latência da ordem de centenas de microssegundos utilizando-se uma rede de sensores de baixa potência.

Neste trabalho, modelamos uma RSSF não hierárquica através de sua representação em uma topologia radial. Singh e Pesch (2010) optaram por utilizar uma rede de sensores hierárquica com topologia em árvore, cuja comunicação em tempo-real se baseia em TDMA (Time Division Multiple Access). Oh et al. (2012) propõem um protocolo de serviço de tempo-real e confiável para uma RSSF não hierárquica com o conceito de topologia de trilhas paralelas geográficas. Mazinani et al. (2012) descrevem um protocolo de roteamento distribuído que possibilita selecionar o próximo nó sensor de uma rede não hierárquica (no caminho entre o nó fonte e o nó sink) com base em uma abordagem modular juntamente com uma política de roteamento para situações

especiais. O protocolo de roteamento apresentado por Mazinani et al. (2012) visa atender restrições de tempo na comunicação da RSSF. No trabalho apresentado por Yoo et al. (2010), os autores descrevem um algoritmo de escalonamento de mensagens para uma RSSF com topologia estrela. Esse algoritmo permite selecionar a próxima mensagem a ser enviada baseada em seu tamanho para atender às restrições de tempo na comunicação.

Na topologia radial adotada neste trabalho, os nós sensores estão organizados por setores (raios), o que facilita a utilização de um protocolo de comunicação com restrição de tempo. Nesse protocolo, deve haver uma comunicação em que “grupos de sensores” devem enviar seus pacotes de mensagens em determinados slots (porções) de tempo. Isso é um dos fatores que caracteriza uma comunicação em tempo-real. Ou seja, garantir que um pacote de mensagens de um sensor ou de um grupo de sensores chegue ao nó sink em um tempo determinado. Neste trabalho, foi utilizada a técnica TDMA para prover restrição de tempo na comunicação da rede de sensores.

Para uma melhor validação do protocolo proposto neste trabalho, foi implementada uma RSSF (com topologia radial) utilizando kits arduino UNO, como nós sensores, e arquitetura Raspberry PI, como nó sink ou gateway. A escolha do kit Arduino baseou-se em sua característica de ser uma plataforma aberta, além de ser programado em uma linguagem de programação baseada na linguagem C. A arquitetura Raspberry PI foi escolhida devido a algumas características importantes, tais como: seu poder de processamento; facilidade de programação, por já possuir os compiladores C e Python; por possuir sistema operacional embarcado; e ter preço reduzido. A programação dos nós sensores é baseada na linguagem C. Entre outros motivos, por ser a linguagem base para o Arduino, além de ser uma linguagem bastante conhecida e utilizada. Já o nó sink (Raspberry PI) foi programado em linguagem Python. Entre outros motivos, por já ser a linguagem nativa que acompanha o pacote de instalação do Raspbian, sistema operacional do Raspberry PI. A comunicação utilizada pela RSSF

é baseada no protocolo ZigBee, que atende ao padrão IEEE 802.15.4. Conforme apresentado pelos autores Gungor e Hancke (2009), e Agha et al. (2009), muitas aplicações industriais requerem baixa taxa de transferência, baixa potência e pontualidade (restrição de tempo). Assim, o padrão de comunicação IEEE 802.15.4 é um dos melhores candidatos para RSSF industriais, segundo os autores Choi e Kim (2008), e Chen et al. (2010). Os autores Anbagi, Erol-Kantarci and Mouftah (2013), Brak e Essaaidi (2012), Gaouda (2013), Lu, Li e Wu (2011), e Vo et al. (2013) têm usado o padrão de comunicação ZigBee devido a características de auto-organização, autorreconfiguração e segurança na comunicação oferecidas pelo protocolo.

Este trabalho está dividido em algumas seções. Na seção Materiais e Métodos, são apresentados os materiais utilizados nesse trabalho, bem como a metodologia empregada para obtenção dos resultados. Em Resultados e Discussão, são mostrados os dados obtidos na validação do sistema, bem como os dados da simulação utilizando o simulador OPNET 14.5 (RIVERBED, 2014). Por fim, e encerrando o trabalho, a conclusão e as referências são apresentadas.

1.1 MOTIVAÇÕES

Em muitas aplicações residenciais, comerciais ou industriais, há a necessidade de restrição de tempo nas atividades realizadas por essas aplicações. Um bom exemplo são os sistemas modernos de distribuição, transmissão e geração de energia elétrica que utilizam o conceito de Smart Grid, o qual está relacionado à interconexão, comunicação e controle de Dispositivos Eletrônicos Inteligentes (DEI), segundo os autores Zhang e Bhatt (2010), Qian et al (2012) e Brak Essaaidi (2012). Os sistemas elétricos que utilizam o conceito de Smart Grid comunicam-se utilizando, em sua maioria, um destes dois protocolos de comunicação: DNP3 e IEC61850. Esses protocolos possuem restrição “rígida” de tempo na comunicação. Assim, uma RSSF a ser introduzida em uma Smart Grid teria

obrigatoriamente que ter um protocolo de comunicação com restrição de tempo. Ou seja, um protocolo de comunicação em tempo-real.

1.2 CONTRIBUIÇÕES

A contribuição principal deste trabalho é permitir que uma RSSF possa ser utilizada em uma aplicação que exija restrição de tempo em sua atividade, tal como em uma Smart Grid. Como exemplo, o ajuste dos tempos de consulta e resposta na RSSF em relação aos tempos exigidos pelos protocolos DNP3 e IEC61850 possibilitará integrar uma RSSF a um sistema elétrico que utilize o conceito de Smart Grid. A possibilidade de ajuste dos frames e slots de tempo do protocolo de comunicação apresentado neste trabalho permitirá integrar a RSSF em qualquer aplicação de tempo-real, desde que haja compatibilidade com os tempos de resposta da aplicação.

2 TOPOLOGIA DA REDE DE SENSORES SEM FIO

2.1 ARQUITETURA DA RSSF

Para implementação prática, foi construída uma rede de sensores sem fio com um Raspberry PI, representando o nó Sink ou gateway, e 5 kits Arduino UNO, representando os nós sensores. Para comunicação, foi utilizado um rádio transceiver XBEE S2 (protocolo Zigbee) em cada nó sensor e no nó sink (LAKHE, 2012). Os rádios foram configurados para comunicarem em broadcast. Ou seja, permitindo a comunicação de um rádio para todos os seus vizinhos. Essa pequena rede foi disposta em uma topologia radial, como apresentada na Figura 1.

2.2 MODELAGEM DA TOPOLOGIA DA RSSF

A implementação do protocolo de comunicação teve como base a modelagem de uma RSSF com topologia radial, com grupo de sensores dispostos como se tivessem em um mesmo raio de uma circunferência. Assim, a

comunicação entre os nós sensores acontece na direção do raio mais externo para o raio mais interno, até as mensagens alcançarem o nó sink, que fica no centro da circunferência ou no raio mais interno. Ou seja, um nó sensor de um raio ou camada mais externa envia seu pacote de dados para um nó sensor de uma camada mais interna, e assim sucessivamente, até que esse pacote chegue ao nó sink. Uma boa analogia seria considerar uma propagação de ondas em sentido inverso, de fora para dentro.

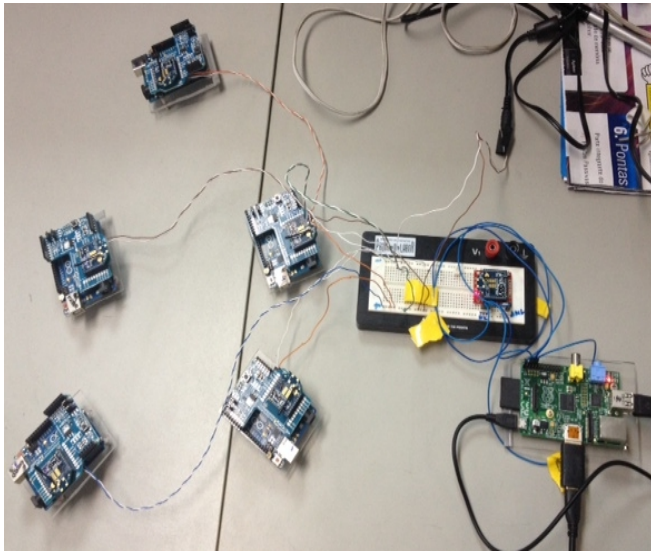


Figura 1: Topologia da RSSF representada pelos kits Arduino e o Raspberry PI

Em nossa rede de sensores, apresentada na Figura 1, dispomos 3 nós sensores (kit Arduino UNO) no raio mais externo, 2 nós (kit Arduino UNO) no raio intermediário, e, mais internamente, o nó sink, representado pelo Raspberry PI.

2.3 PROTOCOLO DE COMUNICAÇÃO DA RSSF

O protocolo de comunicação foi implementado considerando o nó sink como o coordenador de toda a comunicação da rede. Assim, ele envia em broadcast, a cada intervalo de 5 segundos, um pacote de controle para toda a rede. Cada intervalo de 5 segundos representa um slot de tempo para aplicação, e um conjunto de K (onde K é o número de raios) slots de tempo

representa um frame de tempo para a aplicação. Então, o nó Sink envia uma palavra de controle, representando o início do slot de tempo 0 para os nós do raio mais externo. Ao receberem essa palavra de controle, os nós do raio mais externo enviam seus pacotes de dados para os nós do raio mais interno. Os nós do raio mais interno recebem esses dados e ficam aguardando o próximo slot de tempo. Esse procedimento acontecerá sucessivamente até que as mensagens de todos os nós sensores cheguem ao nó sink.

Na Figura 2, é apresentada uma topologia de RSSF não hierárquica com forma radial. Observa-se que o sentido de propagação de mensagens é do raio mais externo para o raio mais interno, e que os nós sensores estão alocados em raios concêntricos.

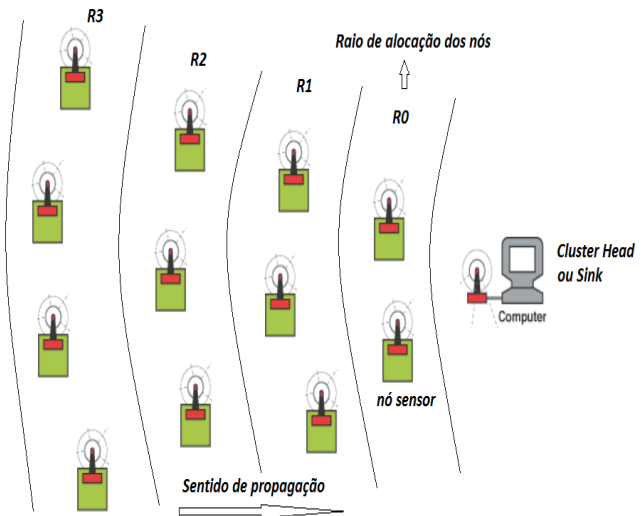


Figura 2: Topologia de RSSF não hierárquica com forma radial

Na Figura 3, é apresentada a estrutura de um frame com seus respectivos slots de tempo. Vale ressaltar que o nó sink é responsável pela transmissão do pacote de controle de sincronização. Ou seja, do pacote de controle de tempo do frame com seus respectivos slots de tempo.

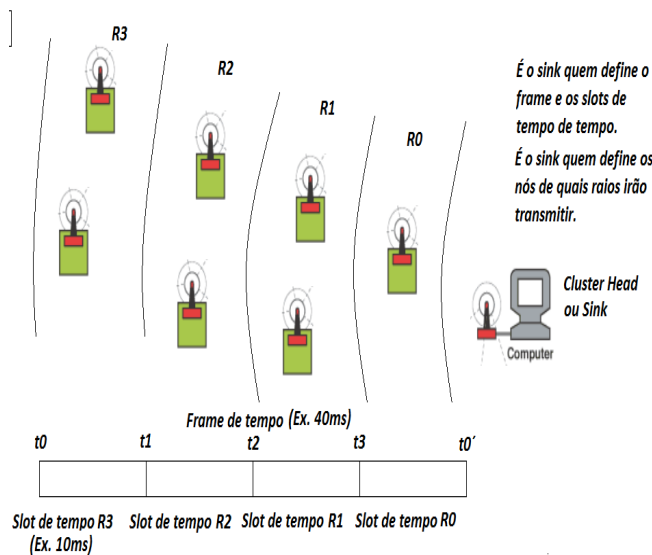


Figura 3: Estrutura de frames com seus respectivos slots de tempo

Na Figura 4, apresenta-se a comunicação de pacotes entre os nós que estão em raios diferentes. A ideia é a comunicação pela propagação de pacotes dos nós de um raio mais externo para nós de um raio mais interno, até chegar ao nó sink. Ou seja, uma comunicação por propagação de ondas no sentido inverso. Na Figura 4, observa-se apenas uma retransmissão de pacotes se o buffer do nó destino já estiver cheio. Ou seja, a cada raio os pacotes são agregados e transmitidos para os nós do raio mais interno. Se o buffer do nó destino já estiver cheio, o pacote recebido é apenas retransmitido.

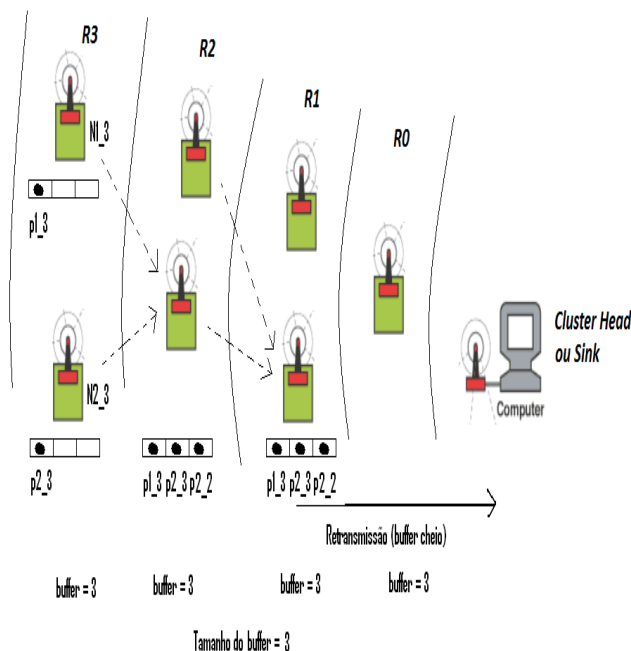


Figura 4: Comunicação de pacotes entre os nós que estão em raios diferentes

Na Figura 5, é mostrado um cálculo para o tamanho do buffer baseado no número de raios existentes na topologia. Esse cálculo permite determinar o tamanho do buffer dos nós sensores, a fim de que não haja tantas retransmissões e todos os pacotes consigam chegar ao nó sink sempre no mesmo frame de tempo, o que garantiria o atendimento do deadline da aplicação de tempo-real. O tamanho do buffer, em bytes, é dado pela Equação 1.

$$\text{tam_buffer} = (R \times 2) + 3(R - 1) \quad \text{Equação 1}$$

Observa-se que o tamanho do buffer é proporcional ao raio (grupo) em que o nó sensor está inserido.

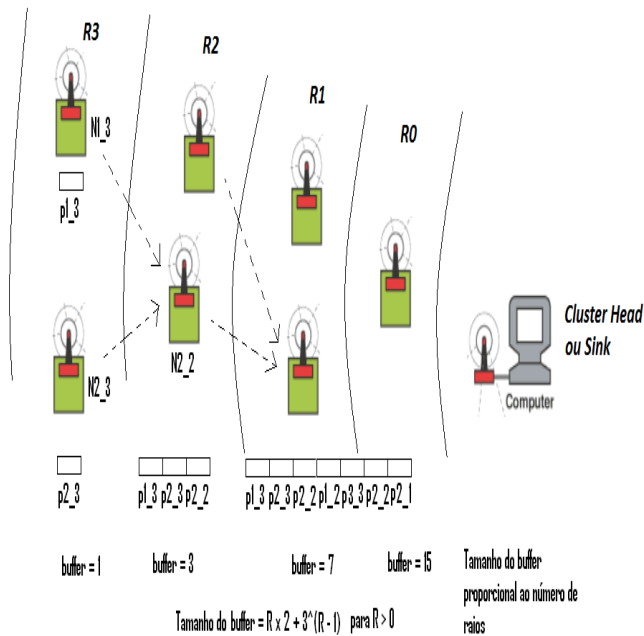


Figura 5: Cálculo para o tamanho do buffer baseado no número de raios existentes na topologia.

Na Figura 6, apresentam-se os deadlines para cada pacote de dados. Os deadlines são utilizados para seleção de qual pacote deve ser transmitido primeiro. Ou seja, o deadline define a prioridade de envio do pacote. O pacote de dados com menor deadline deve ser transmitido prioritariamente. Como a comunicação é em tempo-real, todos os deadlines devem ser atendidos.

Na Figura 7, é apresentada a topologia da RSSF na forma radial. A comunicação deve acontecer do nó de raio externo para um nó de raio interno.

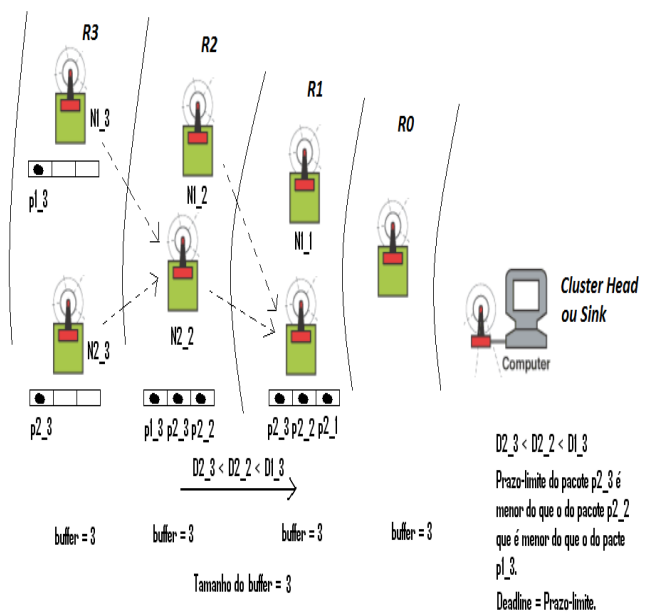


Figura 6: Deadline para cada pacote de dados

Assim, o nó sink envia em modo broadcast um pacote de controle para toda a rede. Esse pacote contém uma mensagem de início de frame e de identificador do raio corrente. Ao receber esse pacote, os nós do último raio transmitem seus pacotes, dentro de seu slot de tempo, para os nós de raio mais interno. E esse procedimento acontece de raio em raio até que as mensagens cheguem ao nó sink. Em caso de nó com buffer cheio, as mensagens devem ser somente retransmitidas

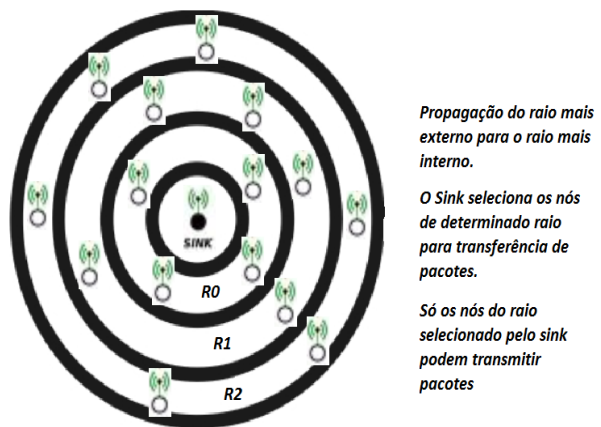


Figura 7: Topologia da Rede de Sensores Sem Fio na forma radial

Na próxima seção, são apresentados os resultados das validações e simulações do protocolo, bem como uma discussão dos dados apresentados.

3 RESULTADOS EXPERIMENTAIS

Para validação do protocolo de comunicação, foram realizadas consultas na RSSF pelo Raspberry PI. As mensagens recebidas pela serial do Raspberry através de uma aplicação em Python foram convertidas para um arquivo .txt, apresentado na Figura 8. O protocolo iniciou uma consulta (enviou uma palavra de controle) para os sensores do raio mais externo, incluindo em um campo da mensagem o identificador desse raio. Essa consulta indica também início do frame e do slot de tempo do raio mais externo. Uma segunda consulta (indicando o segundo slot de tempo) foi enviada para os nós do raio ou camada intermediária. Na Figura 8, observa-se a recepção das mensagens pelo nó sink, no nosso caso o Raspberry PI. Nota-se que as mensagens dos nós mais externos 1, 2 e 3, chegaram com o mesmo *timestamp*. Ou seja, no mesmo slot de tempo dos sensores desse raio. Assim também foi observado com as mensagens dos nós sensores da camada intermediária, que chegaram com o *timestamp* 2. Ou seja, no slot de tempo referente ao raio intermediário. Com esse teste observou-se a correteude na execução do protocolo de comunicação, bem como o atendimento da restrição de tempo na entrega de mensagens ao nó sink. Não foram observados erros ou falhas na recepção de mensagens dos nós sensores pelo nó sink. Isso se deve a alguns motivos: ao tamanho reduzido da rede, com apenas 5 nós sensores; as condições do ambiente, sem ruídos ou interferência de RF; a visada e distância entre os rádios; e um pequeno atraso entre a transmissões dos nós sensores que estão no mesmo raio.

Para validar a escalabilidade do protocolo de comunicação em tempo-real, foi realizada uma simulação no simulador OPNET modeler 14.5, considerado um bom simulador para rede de sensores sem fio (HAMMOODI et al., 2009).

```

recepcao_rasp - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
Node ID --> 1
temperatura - 25
umidade - 65
corrente - 100m
tensão - 2
timestamp - 1
Node ID --> 2
temperatura - 24
umidade - 65
corrente - 100m
tensão - 2
timestamp - 1
Node ID --> 3
temperatura - 25
umidade - 64
corrente - 100m
tensão - 2
timestamp - 1
Node ID --> 4
temperatura - 24
umidade - 65
corrente - 100m
tensão - 2
timestamp - 2
Node ID --> 5
temperatura - 25
umidade - 64
corrente - 100m
tensão - 2
timestamp - 2

```

Figura 8: Arquivo .txt da recepção de mensagens pelo Raspberry PI

Foi criado um cenário tipo campus de 10km x 10km com 15 nós sensores e um nó sink. Cada nó sensor é constituído de um módulo processador (CPU, memória e IO) e um transceiver. O protocolo foi programado no OPNET em linguagem C incluindo as *Kernels Procedures*, funções para comunicação e gerenciamento de dados no OPNET, nos nós sensores e no nó sink. Na Figura 9, apresenta-se a configuração da RSSF no OPNET modeler 14.5.

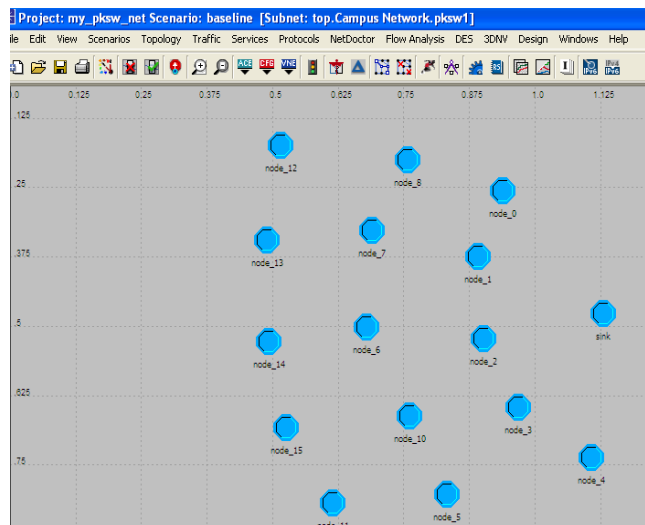


Figura 9: Topologia da RSSF simulada no OPNET modeler 14.5

Na Figura 10, apresentamos os sinais no

transceiver do nó sink. Na Figura superior, observam-se as consultas realizadas pelo nó sink a cada 10 segundos, representado pelo sinal no transmissor do rádio desse nó. Nesse caso, o slot de tempo tem um intervalo de 10 segundos. Ou seja, a cada 10 segundos o nó sink consulta os sensores de cada camada/raio, os quais tem que responder em seu slot de tempo. Na Figura inferior, representando o receptor do rádio do nó sink, observa-se um número maior de pacotes recebidos em cada slot de tempo. Esse resultado deve-se a quantidade de mensagens enviadas pelos nós sensores de cada camada/raio. Pela simulação, foi observada uma média de 3 a 4 pacotes recebidos a cada slot de tempo. Foi observada uma perda de pacotes em cada slot de tempo, que se deve à transmissão simultânea de mensagens pelos rádios dos nós sensores que estão utilizando o mesmo canal de comunicação. Essa simulação atendeu a necessidade de se provar que o protocolo de comunicação apresenta corretude também em RSSFs maiores, suportando assim a escalabilidade.

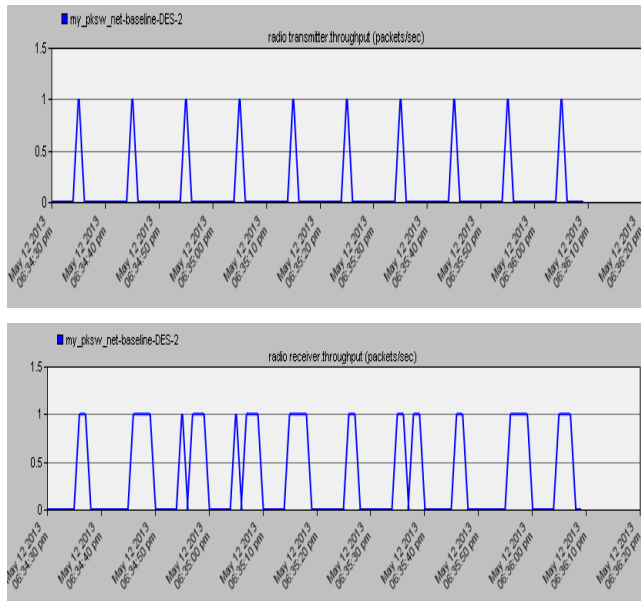


Figura 10: Resultado da simulação de uma RSSF com 15 nós sensores no OPNET 14.5

4 CONSIDERAÇÕES FINAIS

4.1. CONCLUSÕES

A implementação de um protocolo de comunicação para uma RSSF não é uma tarefa trivial, visto a necessidade de contornar problemas relevantes, tais como interferência no meio de transmissão, área de cobertura, falta de linha de visada, entre outros problemas. Assim, foram verificados ou evitados todos esses problemas para que o protocolo funcionasse a contento, e que também conseguisse apresentar as características de tempo-real na comunicação, requisito importante para uma RSSF de tempo-real.

Os resultados obtidos pela validação do protótipo apresentado neste trabalho foram animadores, com características, que se melhoradas, podem possibilitar sua utilização em aplicações críticas no tempo. Outra vantagem desse protótipo é a utilização de uma arquitetura aberta e de baixo custo, como a arquitetura Arduino. A utilização de uma arquitetura Raspberry, como o nó sink, possibilita que esse sistema execute tarefas que exijam maior poder de processamento, como, por exemplo, processamento de imagens. Vale ressaltar que para aplicações mais complexas, como processamento de imagem, os nós sensores devem ter maior poder de processamento e mais memória, que não é o caso do kit Arduino UNO utilizado neste trabalho.

Quanto à validação do protótipo, os resultados mostraram uma corretude na recepção das mensagens e um tempo de resposta esperado para o sistema. Para testar o protocolo de comunicação para uma rede de sensores maior, ou seja, validar o protocolo quanto à escalabilidade, foi necessária a utilização do simulador OPNET. O resultado da simulação mostrou a corretude do protocolo também para redes maiores.

4.2. PROPOSTAS PARA TRABALHOS FUTUROS

Como trabalho futuro, será desenvolvido um modelo de RSSF com topologia hierárquica

em que os nós sensores não serão mais agrupados por setores ou raios. Esse modelo foi pensado para aplicações de Smart Grid, devido ao agrupamento de equipamentos e a restrição de tempo na comunicação desses. Nesse caso, a RSSF será dividida por clusters, com cada cluster possuindo um líder, chamado de cluster head. Cada cluster terá seu frame de tempo para comunicação, com cada nó sensor, dentro do cluster, possuindo um slot de tempo desse frame para sua comunicação.

AGRADECIMENTOS

Esse projeto teve o apoio da Companhia de Energia do Pará (CELPA) e da ANEEL (Agência Nacional de Energia Elétrica).

REFERÊNCIAS

AGHA K. A. et al. Which wireless technology for industrial wireless sensor networks? The development of OCARI technology. *IEEE TRANS. IND. ELECTRON.*, VOL. 56, NO. 10, PP. 4266–4278, OCT. 2009.

ANBAGI I. Al-; EROL-KANTARCI M.; and MOUFTAH H. A. Delay mitigation scheme for wsn-based smart grid substation monitoring. IN *WIRELESS COMMUNICATIONS AND MOBILE COMPUTING CONFERENCE (IWCMC)*, 2013, 9TH INTERNATIONAL, pages 1470–1475, July 2013.

BRAK M. E.; ESSAAIDI M. Wireless sensor network in home automation network and smart grid. IN *COMPLEX SYSTEMS (ICCS)*, 2012 INTERNATIONAL CONFERENCE ON, PAGES 1–6, NOV. 2012.

CHEN F. et al. Simulation study of IEEE 802.15.4 LR-WPAN for industrial applications. *WIRELESS COMMUN. MOBILE COMPUT.*, VOL. 10, NO. 5, PP. 609–621, MAY 2010.

CHOI D.; KIM D. Wireless fieldbus for networked control systems using LR-WPAN.

INT. J. CONTROL AUTOM. SYST, VOL. 6, NO. 1, PP. 119–125, FEB. 2008.

GAOUDA A. Adaptive partial discharge monitoring system for future smart grids. IN *INDUSTRIAL ELECTRONICS SOCIETY, IECON 2013 – 39TH ANNUAL CONFERENCE OF THE IEEE*, pages 4982–4987, Nov 2013.

GUNGOR V. C.; HANCKE G. P. Industrial wireless sensor networks: Challenges, design principles, and technical approaches. *IEEE TRANS. IND. ELECTRON.*, VOL. 56, NO. 10, PP. 4258–4265, OCT. 2009.

HAMMOODI, I. S. et al. A Comprehensive Performance Study of OPNET Modeler For ZigBee Wireless Sensor Networks. *THIRD INTERNATIONAL CONFERENCE ON NEXT GENERATION MOBILE APPLICATIONS, SERVICES AND TECHNOLOGIES (NGMAST)*, 2009.

HILL, J. et al. System Architecture Directions for Networked Sensors. *ASPLOS*, November 2000.

LAKHE, P. R. Wireless Sensor Network Using Zigbee. *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH AND APPLICATIONS (IJERA)*, NATIONAL CONFERENCE ON EMERGING TRENDS IN ENGINEERING & TECHNOLOGY, 2012.

LU C.-W.; LI S.-C.; and WU Q. Interconnecting zigbee and 6lowpan wireless sensor networks for smart grid applications. IN *SENSING TECHNOLOGY (ICST)*, 2011 FIFTH INTERNATIONAL CONFERENCE ON, pages 267–272, Nov 2011.

MAZINANI S. M. et al. An Energy-Efficient Real-Time Routing Protocol for Differentiated Data in Wireless Sensor Networks, *17TH INTERNATIONAL CONFERENCE ON ENGINEERING OF COMPLEX COMPUTER SYSTEMS*, IEEE, 2012.

OH, S.; YIM Y.; LEE J.; PARK H. and KIM S. A Reliable Communication Strategy for Real-Time Data Dissemination in Wireless Sensor Networks. 26TH IEEE INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION NETWORKING AND APPLICATIONS, 2012.

QIAN C. et al. Cognitive transmission based on data priority classification in wsns for smart grid. IN GLOBAL COMMUNICATIONS CONFERENCE (GLOBECOM), 2012 IEEE, PAGES 5166–5171, DEC. 2012.

RIVERBED

<http://www.riverbed.com/products/performance-management-control/opnet.html>

SINGH, J; PESCH, D. An Energy Efficient TDMA Scheduling Scheme for Large-Scale WSN. 2ND INTERNATIONAL CONFERENCE ON COMPUTER ENGINEERING AND TECHNOLOGY, IEEE, 2010.

VO M.-T.; NGUYEN M.-T.; NGUYEN T.-D.; LE C.-T.; and HUYNH H.-T. Towards residential smart grid: A practical design of wireless sensor network and mini-web server based low cost home energy monitoring system. IN ADVANCED TECHNOLOGIES FOR COMMUNICATIONS (ATC), 2013 INTERNATIONAL CONFERENCE ON, pages 540–545, Oct 2013.

YOO S. et al. Guaranteeing Real-Time Services for Industrial Wireless Sensor Networks With IEEE 802.15.4. IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 57, NO. 11, NOVEMBER, 2010.

ZHANG F. L. P.; BHATT N. Next-generation monitoring, analysis, and control for the future smart control center. IN SMART GRID, IEEE TRANSACTIONS ON, VOL. 1, PAGES 186–192, 2010.