



**Universidade Federal de Campina Grande**

**Centro de Engenharia Elétrica e Informática**

Curso de Graduação em Engenharia Elétrica

ÉRICO CASTRO DE ALBUQUERQUE MÉLO

**REDES DE SENSORES SEM FIO APLICADAS AO  
MONITORAMENTO DE UMA MALHA DE ATERRAMENTO**

Campina Grande, Paraíba  
Outubro de 2016

ÉRICO CASTRO DE ALBUQUERQUE MÉLO

REDES DE SENSORES SEM FIO APLICADAS AO  
MONITORAMENTO DE UMA MALHA DE ATERRAMENTO

*Trabalho de Conclusão de Curso submetido à  
Unidade Acadêmica de Engenharia Elétrica da  
Universidade Federal de Campina Grande como  
parte dos requisitos necessários para a obtenção  
do grau de Bacharel em Ciências no Domínio da  
Engenharia Elétrica.*

Área de Concentração: Processamento de Informação

Orientador:

Professor Edmar Candeia Gurjão, D. Sc.

Campina Grande, Paraíba  
Outubro de 2016

ÉRICO CASTRO DE ALBUQUERQUE MÉLO

# REDES DE SENSORES SEM FIO APLICADAS AO MONITORAMENTO DE UMA MALHA DE ATERRAMENTO

*Trabalho de Conclusão de Curso submetido à Unidade Acadêmica de Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.*

Área de Concentração: Processamento de Informação

Aprovado em        /        /

**Professor Avaliador**  
Universidade Federal de Campina Grande  
Avaliador

**Professor Edmar Candeia Gurjão, D. Sc.**  
Universidade Federal de Campina Grande  
Orientador, UFCG

Dedico este trabalho à minha família.

## AGRADECIMENTOS

Agradeço a essa Deus pelo dom da vida e por todas as bênçãos que recebo diariamente.

Agradeço a essa Instituição pela minha acolhida e pelas condições oferecidas, que me permitiram concluir este trabalho.

Agradeço também à minha mãe, Maria Neide, e a meu pai, José Nilson, por terem se esforçado tanto para me proporcionar uma boa educação, por terem me alimentado com saúde, força e coragem, as quais que foram essenciais para superação de todas as adversidades ao longo desta caminhada.

Agradeço também a toda minha família, que com todo carinho e apoio, não mediu esforços para eu chegar a esta etapa da minha vida.

Agradeço a todos os meus amigos que sempre estiveram ao meu lado.

Agradeço ao meu orientador, Edmar Candeia Gurjão, pela paciência e tempo dedicado as sugestões.

Enfim, agradeço a todos que de alguma forma, passaram pela minha vida e contribuíram para a construção de quem sou hoje.

*“What's money? A man is a success if he gets up  
in the morning and goes to bed at night and in  
between does what he wants to do.”*

Bob Dylan.

## RESUMO

Este trabalho tem como propósito apresentar um protocolo de comunicação a ser implementado em uma rede de sensores sem fio utilizada para o monitoramento de uma malha de aterramento. O protocolo proposto é baseado no padrão IEEE 802.15.4 e se encarrega de organizar a o tráfego de informação entre os nós sensores e a central de processamento. Além disso, o protocolo possui um mecanismo que possibilita que as leituras dos nós sensores sejam levadas à uma mesma base temporal. Um protótipo de rede de sensores sem fio foi criado para testar o protocolo proposto.

**Palavras-chave:** 802.15.4, Aterramento, RSSF.

## ABSTRACT

This work presents a communication protocol designed to be implemented in a wireless sensor network for sensing voltage levels in different points of a grounding mesh. The protocol is based in the IEEE 802.15.4 standard and is responsible to organize the traffic of information between the sensor nodes and the processing central. The protocol also defines a scheme that helps the network to the data read by the sensor nodes to a common time base. A prototype of the wireless sensor network was created to rest the proposed protocol.

**Keywords:** XBee, 802.15.4, Grounding, WSN.



# LISTA DE FIGURAS

Figura 1. Representação dos subsistemas de um nó sensor (SAUSEN, 2008).....	15
Figura 2. Visão geral da organização de uma RSSF (SOARES, 2012).....	15
Figura 3. Topologias em estrela e ponto a ponto (IEEE, 2011).....	16
Figura 4. Rede em árvore de <i>clusters</i> (IEEE, 2011).....	17
Figura 5. Topologias possíveis com o protocolo 802.15.4 (VIKA CONTROLS, 2016).....	18
Figura 6. Topologia de uma rede XBee (VIKA CONTROLS, 2016).....	18
Figura 7. Topologia de uma rede DigiMesh (VIKA CONTROLS, 2016).....	19
Figura 8. Topologia de uma rede XBee WiFi (VIKA CONTROLS, 2016).....	19
Figura 9. Visão superior dos módulos XBee (DIGI INTERNATIONAL, 2016).....	20
Figura 10. Estrutura de um quadro API (DIGI INTERNATIONAL, 2016).....	21
Figura 11. Estrutura do pacote <i>TX Request</i> (DIGI INTERNATIONAL, 2016).....	22
Figura 12. Estrutura do pacote <i>TX Status</i> (DIGI INTERNATIONAL, 2016).....	22
Figura 13. Estrutura do pacote <i>RX Packet</i> (DIGI INTERNATIONAL, 2016).....	22
Figura 14. Estrutura de uma malha de aterramento simples.....	23
Figura 15. Valores de leituras atingindo o limiar em instantes diferentes.....	24
Figura 16. Representação em visão superior da rede de sensores sem fio.....	26
Figura 17. Preenchimento da memória circular antes da reorganização das amostras.....	32
Figura 18. Preenchimento da memória circular depois da reorganização das amostras.....	33
Figura 19. Protótipo de nó sensor.....	33
Figura 20. Nós de medição e sincronização.....	34
Figura 21. Nó coordenador montado com Arduino e XBee com o uso de <i>shield</i> .....	34
Figura 22. Visualização dos pacotes nos NM e em NC.....	35
Figura 23. Broadcast recebido pelos nós sensores.....	35
Figura 24. Quadro de requisição provindo do coordenador.....	35
Figura 25. Pacote recebido pelo coordenador com valores lidos pelos nós de medição.....	36
Figura 26. Resultados das medições nos nós sensores.....	36

## LISTA DE TABELAS

Tabela 1. Pinos utilizados no projeto dos nós sensores da rede (DIGI INTERNATIONAL, 2016). .....	20
Tabela 2. Parâmetros importantes dos nós sensores. ....	27
Tabela 3. Lista de funções do Arduino utilizadas (ARDUINO, 2016).....	28
Tabela 4. Numeração dos bits do registrador adcsra (GAMMON, 2012). ....	30
Tabela 5. Descrição dos bits do registrador adcsra (GAMMON, 2012).....	31

# SUMÁRIO

1	Introdução.....	12
1.1	Motivação .....	12
1.2	Objetivos.....	13
1.3	Estrutura do Trabalho .....	13
2	Fundamentação Teórica.....	14
2.1	Sistemas de aterramento .....	14
2.2	Redes de Sensores Sem Fio .....	14
2.3	Protocolo IEEE 802.15.4 .....	16
2.4	Dispositivos XBee .....	17
2.4.1	Modo de operação transparente.....	20
2.4.2	Modo de operação API ( <i>Application Programming Interface</i> ) .....	20
2.5	Protocolos de Sincronização .....	22
3	Material e Métodos.....	23
3.1	Elementos da Rede.....	24
3.1.1	Nós de Sincronização (NS).....	25
3.1.2	Nós de Medição (NM).....	25
3.1.3	Nó Coordenador (NC) .....	25
3.2	O Funcionamento da Rede .....	26
3.3	Montagem experimental .....	27
3.3.1	Hardware .....	27
3.3.2	Firmware.....	28
3.3.2.1	Nó Coordenador .....	29
3.3.2.2	Nó de Medição .....	30
3.3.2.3	Nó de sincronização .....	33
4	Resultados e Discussões .....	33
5	Conclusão .....	37
	Referências .....	38

# 1 INTRODUÇÃO

Em subestações de energia são utilizados sistemas de aterramento na forma de malha que são de fundamental importância para garantir a segurança de instalações elétricas assim como das pessoas em suas proximidades (CASTRO, 2016). Entretanto, fatores ambientais de ação corrosiva podem prejudicar a integridade da malha de aterramento, de modo que se faz necessário monitorar continuamente seu funcionamento para evitar maiores prejuízos (LONG, 2012).

Os avanços recentes na tecnologia de sistemas micro-eleto-mecânicos, comunicação sem fio e eletrônica digital possibilitaram o desenvolvimento de nós sensores de baixo custo e capazes de se comunicar sem fio a curtas distâncias. Nós sensores podem ser utilizados para sensoriamento contínuo e detecção de eventos (AKYILDIZ, 2002). Estes equipamentos sensores podem ser utilizados na forma de Redes de Sensores Sem Fio (RSSF), um tipo especial de rede que normalmente é constituída de uma grande quantidade de nós, e que deve possuir mecanismos para autoconfiguração e adaptação em caso de falhas de comunicação e perda de nós. Isto significa que algoritmos distribuídos tradicionais, como protocolos de comunicação, precisam ser revistos. Os desafios e considerações de projeto de RSSF vão muito além das redes tradicionais (LOUREIRO, 2003).

## 1.1 MOTIVAÇÃO

Vários sistemas têm sido propostos para monitorar as condições dos sistemas de aterramento. A maioria destes sistemas se utiliza de sensores instalados nas pontas das hastes de aterramento. Os sensores são responsáveis por medir valores de tensão e posteriormente transmiti-los para uma central de processamento.

Um dos sistemas que visam analisar as condições da malha de aterramento é proposto em CASTRO,2016 no contexto de um projeto de pesquisa e desenvolvimento com a CEPISA – Eletrobrás Piauí Distribuidora. Este projeto de conclusão de curso propõe um protocolo de comunicação entre os nós sensores e a central de comunicação a ser utilizado no sistema de monitoramento da malha de aterramento.

## 1.2 OBJETIVOS

Este trabalho tem por objetivos:

- Propor um protocolo de comunicação a ser utilizado em uma rede de sensores sem fio com a finalidade de monitorar as condições de malhas de aterramento subestações.
- Propor um algoritmo de sincronização que permita trazer os resultados das medições para uma mesma referência temporal.
- Montagem e configuração de uma rede de sensores sem fio, baseada em módulos XBee 802.15.4 para teste do protocolo.
- Desenvolvimento do *firmware* a ser embarcado nos microcontroladores presentes nos nós sensores da rede.

## 1.3 ESTRUTURA DO TRABALHO

No desenvolvimento deste trabalho é feita inicialmente uma revisão bibliográfica a respeito de sistemas de aterramento, de redes de sensores sem fio, padrão IEEE 802.15.4, dispositivos XBee e protocolos de sincronização. Em seguida são apresentados os materiais e métodos utilizados para desenvolver e testar o protocolo de comunicação para a rede de sensores sem fio. Em seguida são apresentados os resultados dos experimentos realizados e uma análise dos mesmos. Finalmente, estão explicitadas algumas conclusões a respeito do projeto.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentados os embasamentos teóricos que deram suporte ao estudo. O texto aqui apresentado é resultado de uma ampla revisão bibliográfica a respeito do tema.

### 2.1 SISTEMAS DE ATERRAMENTO

Sistemas de aterramento têm por finalidade prover um caminho de baixa impedância para o escoamento de correntes de surto, tanto na frequência industrial como provenientes de descargas atmosféricas, além de fornecer uma referência para todo o sistema elétrico. Logo, os sistemas de aterramento são de fundamental importância para garantir a segurança de instalações elétricas, assim como das pessoas em suas proximidades (CASTRO, 2016).

A Absorção de corrente elétrica pelo sistema de aterramento dá-se por meio de uma resistividade que o solo apresenta para este aterramento em especial. Existem vários sistemas de aterramento variando do mais simples, constituído de uma haste cravada verticalmente em um solo homogêneo, até uma malha de aterramento.

Além de atender às finalidades básicas de um sistema de aterramento, as malhas de aterramento utilizados em subestações de energia elétrica são responsáveis por uniformizar o potencial elétrico na superfície da subestação prevenindo a geração de tensões capazes de provocar choque elétrico em pessoas ou animais.

### 2.2 REDES DE SENSORES SEM FIO

Uma rede de sensores sem fio (RSSF) pode ser descrita como um arranjo de nós sensores que, de maneira cooperativa, supervisionam um ambiente (BURATTI, 2009). Os nós sensores ficam em contato direto com o ambiente monitorando uma região de interesse, sendo responsáveis por coletar, enviar e processar dados.

Os nós sensores geralmente possuem uma quantidade limitada de energia, que determina o tempo de vida do dispositivo e conseqüentemente da rede. Por isso, a

eficiência energética é um conceito importante em redes de sensores sem fio. Outras características importantes são o baixo custo, fácil programação, um projeto de hardware expansível, e pequeno tamanho.

Cada nó sensor da RSSF é organizado em 4 subsistemas (SAUSEN, 2008): Um subsistema de sensoriamento, um subsistema de processamento, um subsistema de comunicação e um subsistema de energia (Figura 1).

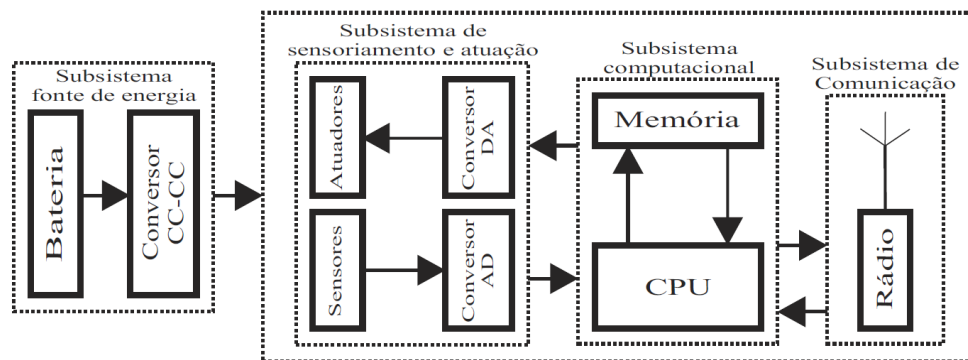


Figura 1. Representação dos subsistemas de um nó sensor (SAUSEN, 2008).

Uma rede de sensores sem fio é composta basicamente por: um conjunto de nós sensores, rede de comunicação sem fio que os conecta; uma estação central para organizar as informações obtidas, além de recursos computacionais para extrair informação dos dados obtidos do meio (SOUSA, 2009). A troca de informações entre a rede de sensores e a estação central é realizada por meio de um nó sorvedouro (*sink node*), que deve receber os dados dos demais nós da rede e repassá-los a um computador (SOARES, 2012). Uma visão geral da organização de uma rede de sensores sem fio pode ser observada na Figura 2.



Figura 2. Visão geral da organização de uma RSSF (SOARES, 2012)

## 2.3 PROTOCOLO IEEE 802.15.4

Este protocolo foi criado com a finalidade de estabelecer um padrão para redes PAN (*Personal Area Networks*), que possibilite a construção de redes de sensores sem fio com baixo custo, baixa taxa de transmissão e baixo consumo energético (VALLE, 2012). As redes PAN são controladas por um nó coordenador, que é capaz de decidir quais são os nós que podem transmitir com prioridade.

O padrão 802.15.4 define o comportamento das camadas física e de enlace.

Para a camada física este protocolo define Offset-QPSK como técnica de modulação. A taxa de dados é definida como 250 kbps e frequência é 2,4 GHz.

As duas topologias aceitas neste padrão são ponto a ponto e estrela. Na topologia ponto a ponto, um nó é a fonte e o outro é o sorvedouro e eles trocam informações diretamente. Na topologia em estrela, um nó central é responsável por receber pacotes de todos os outros nós e encaminha-los para seus destinos finais (VALLE, 2012). As topologias em estrela e ponto a ponto estão ilustradas na Figura 3.

Uma outra possibilidade para este padrão é uma árvore de *-clusters*, que é um caso especial da topologia ponto-a-ponto (IEEE, 2011). A topologia de Arvore de clusters é ilustrada na Figura 4

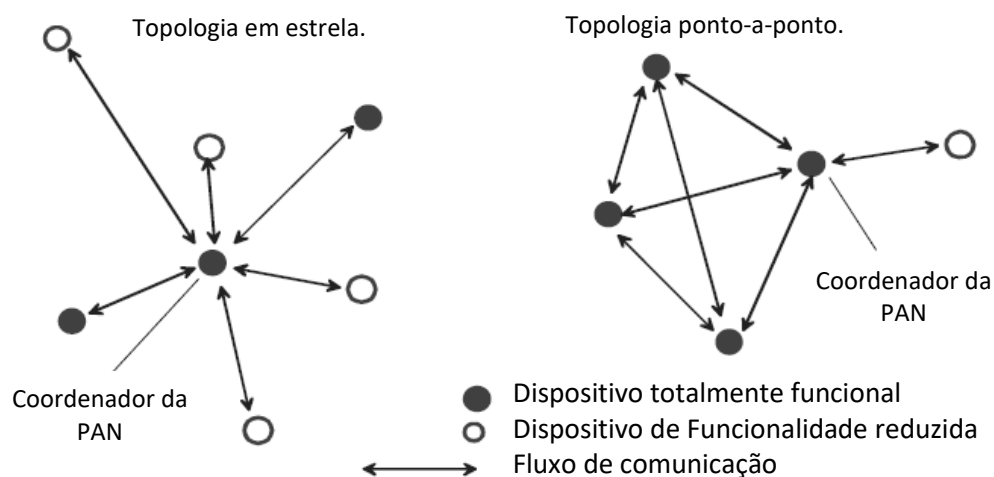


Figura 3. Topologias em estrela e ponto a ponto (IEEE, 2011).



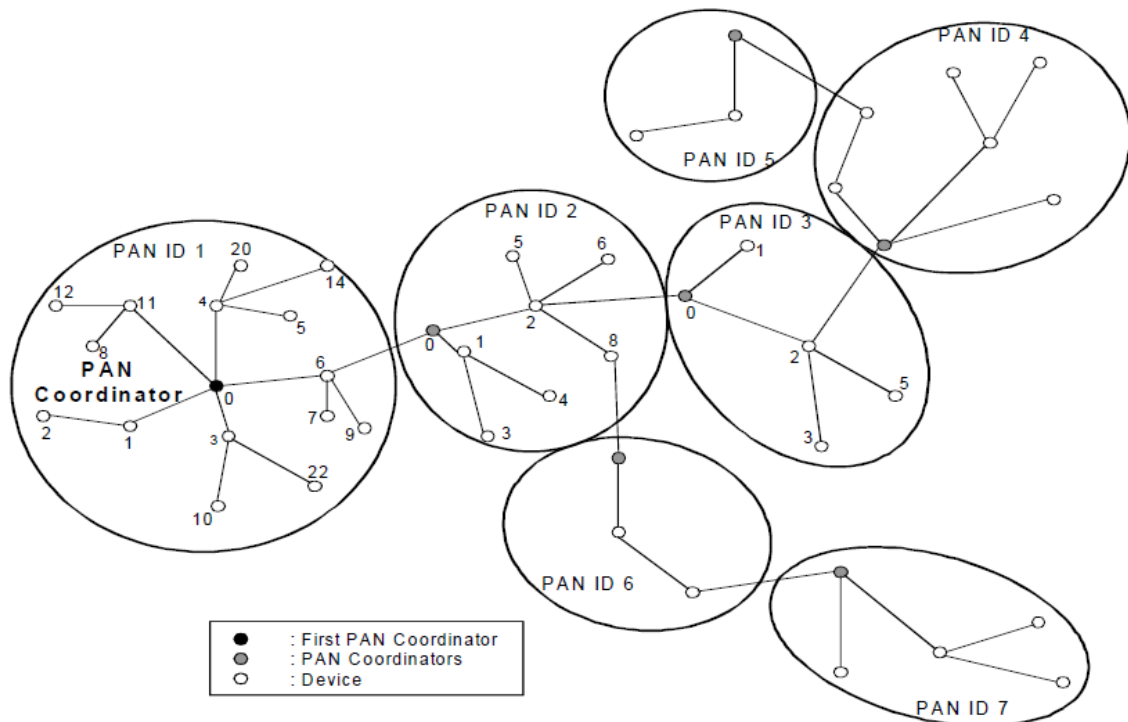


Figura 4. Rede em árvore de *clusters* (IEEE, 2011).

Este padrão possibilita dois mecanismos de controle de acesso ao meio, que são com farol (*beacon*) e sem farol. Quando o modo de operação sem farol é utilizado, o acesso ao meio é controlado por CSMA/CA (VALLE, 2012). O modo operacional com farol utiliza uma combinação de CSMA/CA e TDMA.

Este padrão permite o uso de esquemas opcionais de retransmissão baseados em confirmações e *timeouts* para prover confiabilidade à rede (VALLE, 2012). O lado receptor envia confirmação após receber o pacote correto. Caso o transmissor não receba uma confirmação, o pacote é transmitido novamente um número limitado de vezes.

## 2.4 DISPOSITIVOS XBEE

Módulos XBee são constituídos de um microcontrolador e um transceptor utilizado para comunicação remota entre nós sensores de redes de sensores sem fio. Os módulos XBee, que anteriormente eram produzidos pela *MaxStream*, são atualmente fabricados pela *Digi International*.

Os módulos XBee são comercializados nas seguintes versões:

- XBee Série 802.15.4
- XBee ZigBee

- XBee DigiMesh
- XBee WiFi

Os módulos do modelo **XBee Série 802.15.4** permitem a construção de redes ponto-a-ponto e ponto-multiponto (estrela), não possuindo funções de roteamento. Quando são utilizados módulos deste modelo, qualquer nó da rede pode operar em modo de baixo consumo (*sleep mode*). As topologias possíveis com o modelo 802.15.4 estão ilustradas na Figura 5.

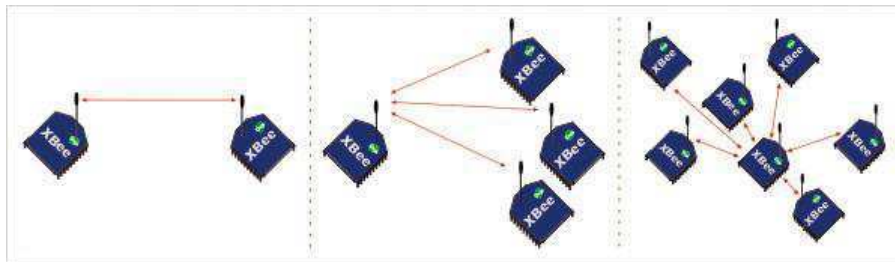


Figura 5. Topologias possíveis com o protocolo 802.15.4 (VIKA CONTROLS, 2016).

O modelo **XBee ZigBee** possui como camadas física e de enlace o protocolo 802.15.4, acima da qual opera o padrão ZigBee, que estabelece três tipos de dispositivos na rede: Coordenador, Roteadores e *End Devices*. Coordenador e roteadores possuem capacidade de roteamento de dados pela rede. *End Devices* não roteiam dados, porém podem operar em modo de baixo consumo de energia. A topologia em malha do modelo ZigBee é ilustrada na Figura 6.



Figura 6. Topologia de uma rede XBee (VIKA CONTROLS, 2016).

O **XBee DigiMesh** funciona com o protocolo proprietário da *Digi International*. Possui como camada física o protocolo 802.15.4. Assim como o ZigBee, o DigiMesh é destinado para redes em malha (*mesh*), porém todos os nós da rede podem rotear dados e também operar em modo de baixo consumo de forma sincronizada. A topologia do modelo DigiMesh é ilustrada na Figura 7.

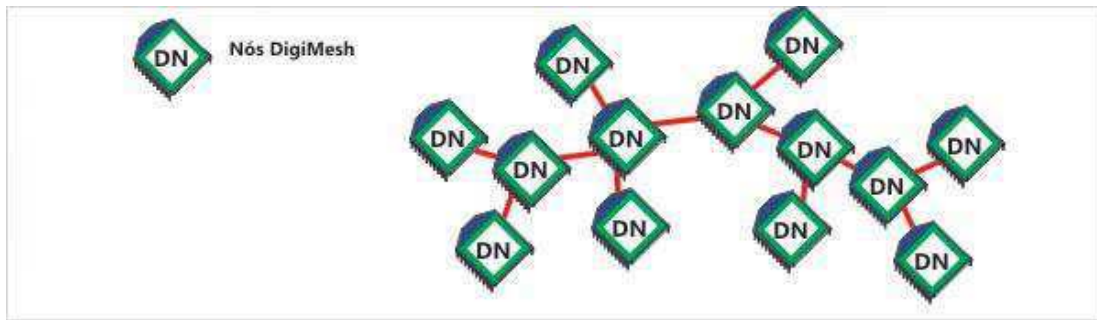


Figura 7. Topologia de uma rede DigiMesh (VIKA CONTROLS, 2016).

O XBee Wi-Fi utiliza o padrão IEEE 802.11 mundialmente conhecido e difundido. Os módulos XBee com protocolo Wi-Fi comunicam-se diretamente com *Access Points* Wi-Fi e facilitam a integração entre dispositivos na internet e redes ethernet locais. Não possui a funcionalidade de roteamento de dados comum aos protocolos ZigBee e DigiMesh. A Topologia de uma rede XBee com protocolo Wi-Fi está ilustrada na Figura 8.



Figura 8. Topologia de uma rede XBee WiFi (VIKA CONTROLS, 2016).

O modelo escolhido para ser utilizado neste trabalho foi o XBee 802.15.4, pois este é capaz de assumir a topologia desejada, na qual um nó sensor pode se comunicar diretamente com qualquer outro componente da rede, diferentemente do que acontece no padrão ZigBee que se utiliza de roteadores.

Os módulos XBee do modelo utilizado têm potência de transmissão de 1 mW (0 dBm), alcance de até 30m em ambiente fechado e de até 100m em ambientes abertos com visada direta. A versão XBee-PRO tem potência de transmissão de 60 mW (18 dBm), com alcance de até 100 m em ambientes fechados e 1,5 km em ambientes abertos com visada direta. Os módulos XBee tem sensibilidade no receptor de -92 dBm e os módulos XBee-PRO -100 dBm. A taxa de transmissão deste módulo é de 250 Kbps.

Os módulos XBee possuem dois tipos possíveis de endereço. Um de 16 bits e um de 64 bits. O endereço de 64 bits é o número de série do módulo e não pode ser alterado. O outro possui 16 bits e é definido através da alteração do parâmetro MY.

Os módulos XBee possuem dois modos de operação. São eles O modo de operação transparente e o modo API.

#### 2.4.1 MODO DE OPERAÇÃO TRANSPARENTE.

Por padrão os módulos XBee operam em modo transparente. Neste modo de operação, os módulos atuam como simples substitutos para uma linha serial. Ou seja, todos os dados recebidos pelo pino DI, que é a entrada de dados serial são enfileirados em um *buffer* no módulo transmissor; transmitidos por radiofrequência e recebidos na saída do pino DO, que é a saída serial dos módulos. Uma vista da disposição dos pinos dos módulos XBee pode ser observada na Figura 9. Os pinos utilizados na comunicação entre os nós sensores é ilustrada na Tabela 1.

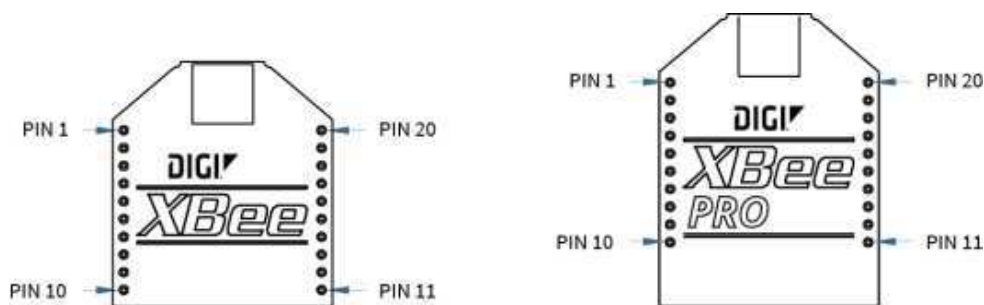


Figura 9. Visão superior dos módulos XBee (DIGI INTERNATIONAL, 2016).

Tabela 1. Pinos utilizados no projeto dos nós sensores da rede (DIGI INTERNATIONAL, 2016).

PIN	Nome	Direção	Função
1	VCC	-	Fonte de potência
2	DOUT	Output	Entrada serial de dados
3	DIN/CONFIG	Input	Saída serial de dados
10	GND	-	Terra

Este modo de operação não permite identificar a origem dos dados recebidos, nem a potência do sinal recebido. Também não são disponíveis pacotes de status nem confirmação.

#### 2.4.2 MODO DE OPERAÇÃO API (*APPLICATION PROGRAMMING INTERFACE*)

Neste modo de operação todos os dados que entram e saem dos módulos estão contidos em quadros (*frames*). Isso aumenta o nível de interação entre as aplicações e a forma como os módulos se comunicam, pois estes frames possuem informações que podem ser úteis a aplicações específicas.

A comunicação API é feita por meio de pacotes que têm formato pré-definido. Estes pacotes têm cabeçalhos cujos campos podem ser alterados pela aplicação. Isso permite às aplicações desenvolver funções de roteamento, que não são cobertas pelo padrão IEEE 802.15.4.

O modo de operação API permite a criação de aplicações que necessitem do endereço de origem. Permite ainda que o endereço de destino das mensagens seja modificado se a necessidade de alterar as configurações dos módulos. O modo API também dá acesso ao nível de potência do sinal recebido.

A estrutura dos pacotes API está disponível na Figura 10. O primeiro byte delimita o início do pacote. Os bytes 2 e 3 são responsáveis por explicitar seu tamanho. Os Bytes de 4 a  $n$  são diferentes para cada tipo de pacote. O byte  $n+1$  é um *checksum* para verificar erros

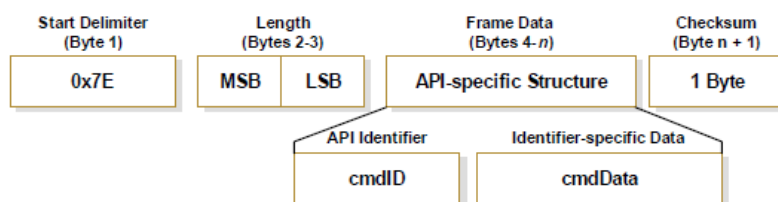


Figura 10. Estrutura de um quadro API (DIGI INTERNATIONAL, 2016).

Para utilizar o modo API é necessário conhecer ao menos três tipos de pacote:

- *TX Request*,
- *TX Status*;
- *RX Packet*.

*TX request* é o tipo de pacote que é entregue pela aplicação do nó transmissor para ser enviado para o destino. Tem como base a estrutura comum aos quadros API. O pacote *TX Status* é gerado automaticamente para comunicar o sucesso ou a falha numa transmissão. O pacote *RX Packet* é o padrão para o modo como os quadros são entregues à aplicação de destino as Figuras de 11 a 13 tem mais informações sobre estes pacotes.

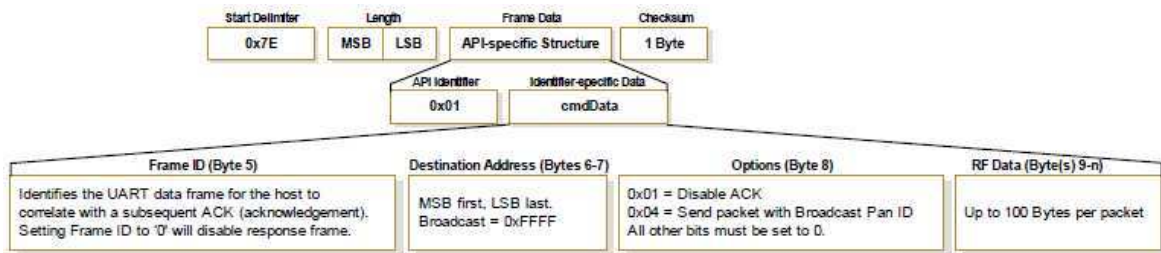


Figura 11. Estrutura do pacote *TX Request* (DIGI INTERNATIONAL, 2016).

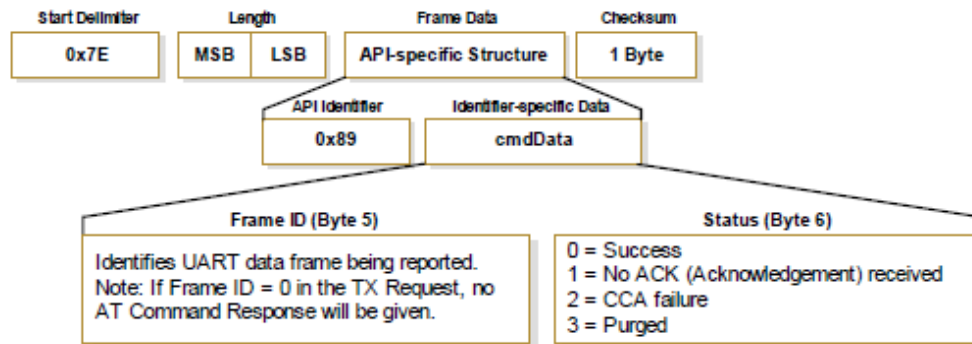


Figura 12. Estrutura do pacote *TX Status* (DIGI INTERNATIONAL, 2016).

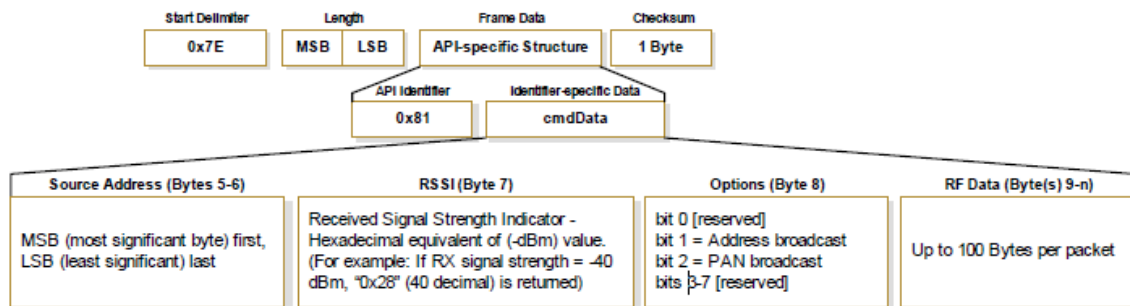


Figura 13. Estrutura do pacote *RX Packet* (DIGI INTERNATIONAL, 2016).

## 2.5 PROTOCOLOS DE SINCRONIZAÇÃO

Existem aplicações no campo de RSSF nas quais o tempo é um fator importante. Os nós sensores são capazes de medir tempo localmente. Porém devido à desvios de fase, e taxas de variação de frequência aleatórias nos osciladores, os valores de tempo aferido acabam sendo diferentes em cada nó sensor da rede (KARL, 2007).

Sincronização é um fator importante em RSSF já que estas têm como propósito monitorar fenômenos físicos. Mesmo osciladores de nós sensores de mesmo modelo tem diferentes taxas de variação de frequência em seus osciladores de modo que eventualmente sairão de sincronismo, daí a importância de algoritmos de sincronização.

### 3 MATERIAL E MÉTODOS

Em subestações de energia elétrica o sistema de aterramento empregado é a malha de aterramento. Este sistema é caracterizado por um conjunto de hastes enterradas no solo e interconectadas por meio de cabos. A configuração de uma malha de aterramento simples composta de 16 hastes conectadas por cabos de cobre é ilustrada na Figura 14.

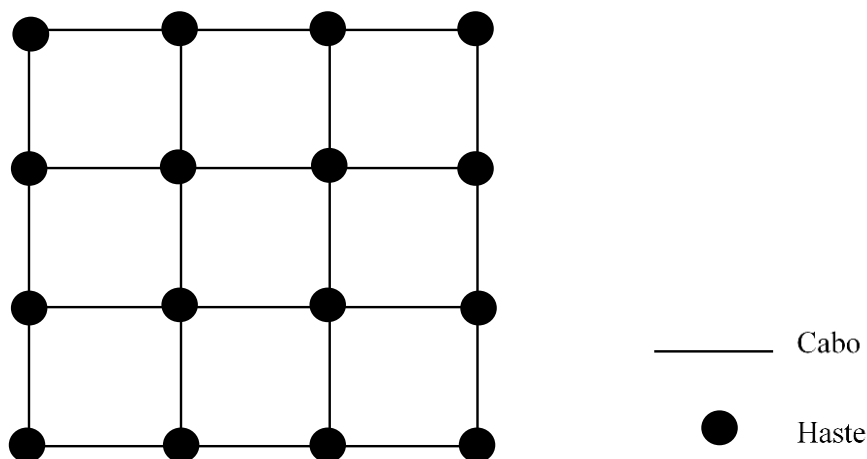


Figura 14. Estrutura de uma malha de aterramento simples.

As malhas de aterramento têm como objetivo prover um caminho de baixa resistência para correntes de manobra e de surtos atmosféricos, além de diminuir variações de tensão na superfície da subestação que poderiam gerar tensões de passo e de toque perigosos. Entretanto fatores ambientais de ação corrosiva podem vir a prejudicar a integridade da malha de aterramento, de modo que se faz necessário monitorar continuamente seu funcionamento para evitar maiores prejuízos (LONG, 2012).

Castro (2016) propõe um sistema de aquisição de dados para monitorar o funcionamento da malha de aterramento. Para contribuir para a solução do problema de monitoramento da malha de aterramento, este trabalho propõe um protocolo de comunicação específico para a rede de sensores sem fio a ser utilizada para reunir as informações aferidas pelos em cada ponto de medição.

O propósito do sistema de monitoramento é identificar a ocorrência da descarga automaticamente, medir os valores de tensão na malha de aterramento por ela causados, e enviar o resultado das medições a uma central de processamento.

Para tanto os nós sensores são equipados com conversores A/D que permitem o monitoramento dos níveis de tensão nos pontos monitorados. Quando os níveis de tensão ultrapassam um determinado limiar a descarga é caracterizada.

Porém diferenças de resistividade causadas por defeitos na malha podem fazer com que os níveis de tensão variem substancialmente dentro da malha, de modo que o momento que a tensão lida por nós sensores em pontos diferentes da malha atinge o limiar estabelecido em momentos diferentes (Figura 15). Logo, observou-se a necessidade de criar um esquema que permita trazer as amostras lidas pelos nós sensores para uma mesma referência temporal.

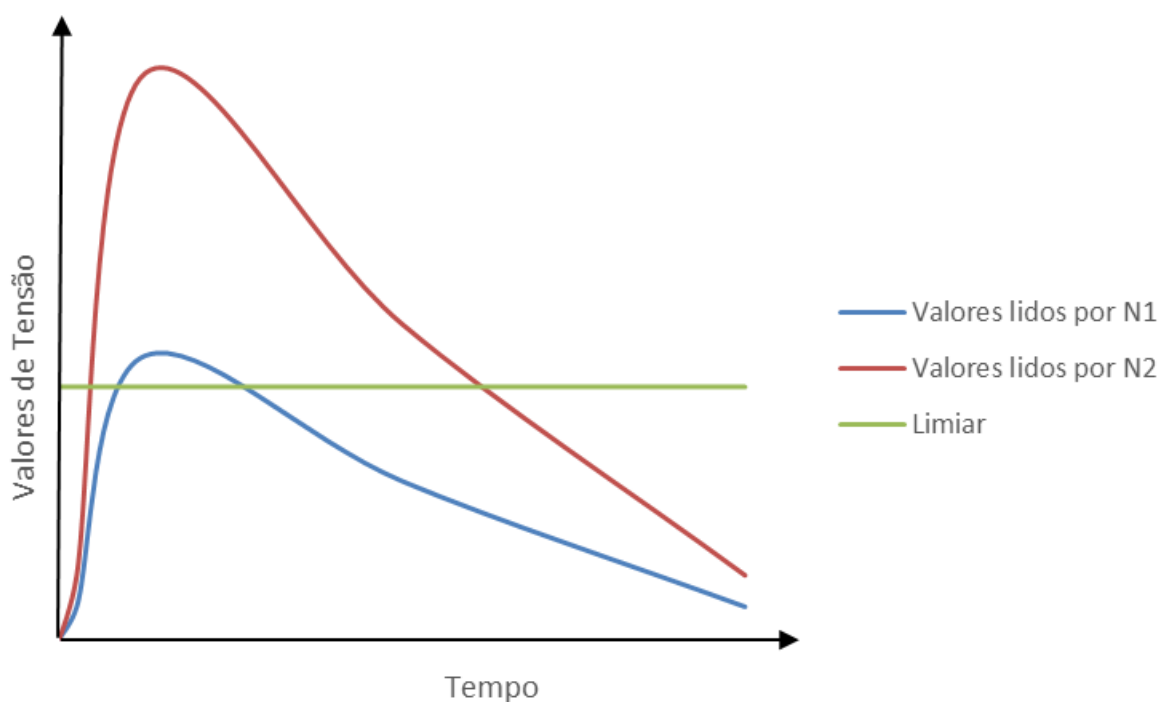


Figura 15. Valores de leituras atingindo o limiar em instantes diferentes.

Chegou-se à conclusão de que a identificação do instante da descarga de maneira individual por cada nó da rede não permitiria a reconstrução da cronologia do evento de maneira adequada. A alternativa adotada foi a utilização de um tipo de nó sensor com função específica de prover uma referência temporal para os demais nós da rede.

Portanto são utilizados três tipos de nós sensores na rede: Nós de Sincronização (NS), Nós de Medição (NM) e o Nó Coordenador (NC). Em todos os nós da rede são utilizados módulos transceptores XBee 802.15.4.

### 3.1 ELEMENTOS DA REDE



Os módulos XBee que foram utilizados tem implementado o protocolo IEEE 802.15.4 que é responsável por endereçamento, detecção de erros, controle de acesso ao meio (MAC) e confirmação e reenvio de pacotes. Foi utilizado o endereçamento de 16 bits.

### 3.1.1 NÓS DE SINCRONIZAÇÃO (NS)

Os nós de sincronização são responsáveis por detectar a ocorrência do evento de interesse e enviar um pacote *broadcast* que servirá de referência temporal para os demais sensores.

### 3.1.2 NÓS DE MEDIÇÃO (NM)

Os nós de medição são responsáveis por medir os valores de tensão em diversos pontos da malha de aterramento. Estes nós têm por finalidade coletar valores de tensão nos pontos de medição para que posteriormente se possa realizar um levantamento da comportamento da malha quando submetida a uma descarga atmosférica. Eles devem enviar ao coordenador da rede os resultados de suas medições juntamente com o intervalo de tempo entre a detecção da descarga e a recepção do broadcast.

### 3.1.3 NÓ COORDENADOR (NC)

O coordenador da rede atua como um nó sorvedouro, que por sua vez é um dispositivo encarregado reunir todas as informações colhidas pelos nós de medição e encaminha-las a uma central na qual estas serão analisadas. Para que a comunicação se dê de maneira organizada, minimizando o número de choques na troca de pacotes, os nós de sensores não devem enviar seus pacotes para o coordenador logo de imediato. Ao invés disso, o coordenador que ficará responsável por enviar pacotes de requisição individuais a cada membro da rede, que deverão ser respondidos com as informações colhidas pelos mesmos.

### 3.2 O FUNCIONAMENTO DA REDE

Os nós sensores são posicionados nas hastes da malha de aterramento como na Figura 16. É utilizado o modo de operação sem farol do padrão IEEE.802.15.4, portanto o controle de acesso ao meio é realizado por um CSMA/CA. Isto implica que não é possível a mais de um nó sensor transmitir simultaneamente.

Os nós de sincronização são espalhados em diferentes pontos da rede de modo a garantir que mesmo havendo defeitos na malha de aterramento, a descarga seja detectada com sucesso por algum deles.

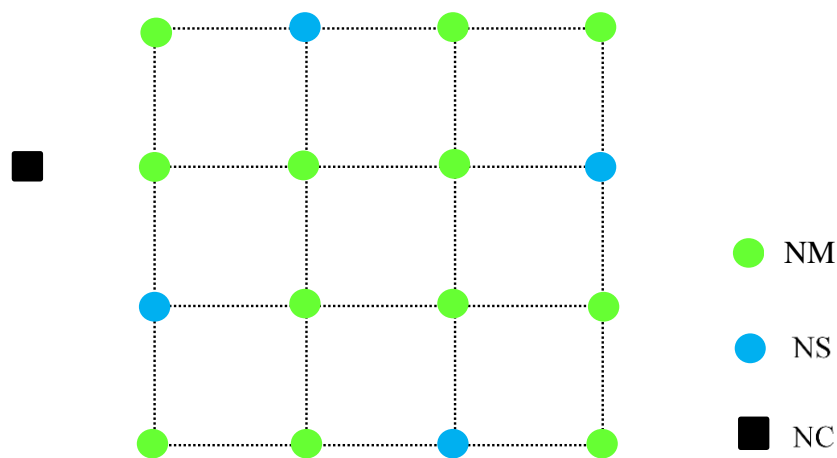


Figura 16. Representação em visão superior da rede de sensores sem fio.

Os nós de medição estão continuamente lendo valores de tensão dos pontos da malha onde estão alocados, e guardando estes valores lido em uma *memória circular*. No momento da descarga os nós de medição detectam o aumento de tensão localmente e armazenam quantidades predefinidas de valores de amostras anteriores e posteriores à descarga.

Simultaneamente um nó de sincronização identifica o aumento no valor de tensão e enviam uma mensagem *broadcast* para os demais membros da rede. Então os nós de medição deverão medir o intervalo de tempo decorrido entre a identificação local da descarga e a recepção da mensagem de sincronização, e armazenar o valor de sua duração em uma variável.

O nó coordenador também recebe a mensagem do NS e, por meio desta, é então informado da ocorrência da descarga e pode começar a enviar requisições individuais a cada nó NM. Quando solicitados pelo NC, os NM devem enviá-lo os valores lidos

juntamente com o valor correspondente ao intervalo de tempo entre a detecção local da descarga e a chegada da mensagem de sincronização.

### 3.3 MONTAGEM EXPERIMENTAL

Nesta sessão serão descritos os elementos de *hardware* e *firmware* utilizados para o desenvolvimento do protótipo de rede de sensores sem fio utilizado para testar o protocolo desenvolvido.

#### 3.3.1 HARDWARE

Os nós sensores, da rede de sensores sem fio, foram implementados utilizando rádios XBee 802.15.4 em conjunto com microcontroladores do tipo Arduino Uno. Os terminais de transmissão e recepção, TX e RX respectivamente, do microcontrolador foram diretamente ligados aos terminais de recepção e transmissão do módulo XBee para fazer a transmissão dos dados. Os microcontroladores processam as mensagens trocadas pelos membros da rede permitindo a tomada de decisões por parte dos mesmos.

Os parâmetros dos módulos XBee são responsáveis por especificar o modo de funcionamento dos módulos. Os parâmetros que precisam ser modificados para a montagem da rede estão descritos na Tabela 2.

Tabela 2. Parâmetros importantes dos nós sensores.

Parâmetro	Descrição
MY	Define o valor o endereço de 16 bits do módulo.
DL	Endereço de destino para comunicação. No modo API o endereço de destino pode ser especificado pela aplicação.
AP	Define o modo de operação da rede. As opções são API[0], API[1] e API[2].

O endereço de destino das mensagens DL do coordenador deve ser programado em zero, pois este nó irá se comunicar com vários nós sensores, e o destino será especificado sempre pela aplicação.

Os nós sensores assim como o coordenador da rede, são capazes de enviar e transmitir mensagens, porém ao passo que o coordenador que precisa se comunicar com cada sensor da rede, os nós sensores são configurados para se comunicarem exclusivamente com o coordenador, que irá encaminhar as informações para uma central.

Os módulos XBee foram programados no modo API no qual todas as informações contidas nos cabeçalhos das mensagens trocadas são entregues pela porta serial. Estas informações serão processadas e utilizadas pelos microcontroladores conectados aos módulos a fim de organizar a comunicação.

Os módulos XBee que foram utilizados tem implementado o protocolo IEEE 802.15.4 que é responsável por endereçamento, detecção de erros, controle de acesso ao meio (MAC) e confirmação e reenvio de pacotes.

Programando os microcontroladores presentes nos membros da rede podemos implementar os protocolos que devem funcionar nas camadas acima do protocolo IEEE 802.15.4. Para maior eficiência, este protocolo é orientado para suprir especificamente as demandas do projeto.

### 3.3.2 FIRMWARE

Para programar o Arduino, é utilizada a linguagem de programação C/C++. O Arduino permite que sejam utilizadas diversas funções prontas que facilitam bastante a programação do mesmo. Na Tabela 3 estão listadas algumas das funções utilizadas.

Tabela 3. Lista de funções do Arduino utilizadas (ARDUINO, 2016)

Função	Descrição
<code>void setup()</code>	A esta função é chamada apenas uma vez. Geralmente utilizada para inicializar variáveis
<code>void loop()</code>	O <i>loop</i> é um laço. As operações declaradas nesta parte do programa são repetidas indefinidamente enquanto o microcontrolador estiver em funcionamento.
<code>Serial.begin()</code>	Estabelece a taxa de bits por segundo da comunicação serial.
<code>Serial.available()</code>	Retorna a quantidade de bytes disponíveis na porta serial.
<code>Serial.print()</code>	Imprime dados pela porta serial no formato ASCII.
<code>Serial.write()</code>	Escreve dados binários na porta serial. Pode receber como parâmetros valores, <i>strings</i> ou <i>buffers</i> acompanhados de seu comprimento.
<code>Serial.read()</code>	Lê os dados que entram na porta serial.
<code>delay()</code>	Pausa a execução do programa por uma quantidade de milissegundos estabelecida por meio de parâmetros.
<code>millis()</code>	Retorna o número de milissegundos passados desde o início da execução do programa no formato de um inteiro sem sinal.

A troca de informações entre o módulo XBee e o Arduino se dá por meio de uma comunicação serial. Ao receber os dados pelo pino RX, estes são armazenados em um buffer com capacidade para 64 bytes. Se durante uma comunicação não houver espaço suficiente para armazenar um novo pacote, haverá perda de dados.

Utiliza-se a função *Serial.read()* para ler os bytes que estão na porta serial. À medida que os bytes são lidos, mais espaço será liberado no buffer da porta serial.

Depois de formados, os dados dos pacotes saem do Arduino pelo pino TX e que deve estar conectado ao pino RX de um módulo XBee. Para enviar os dados é utilizada a função *Serial.write()*, que tem como parâmetros o um vetor e seu comprimento. É importante que os elementos do vetor utilizado não tenham tamanho maior que um byte. Para tanto o formato escolhido para o buffer é do tipo *uint8\_t* (Inteiro sem sinal de 8 bits).

Para que a comunicação seja bem-sucedida, é preciso que os pacotes sejam montados corretamente. Em outras palavras, todos os campos dos pacotes devem ser preenchidos com valores coerentes. Com isso em vista, pensou-se em criar uma função que recebesse como parâmetros o destinatário do pacote, a mensagem a ser transmitida e o seu tamanho. Esta função é responsável por preencher todos os campos do pacote, calculando automaticamente o tamanho do pacote e o *checksum*.

Os microcontroladores do tipo Arduino foram programados de acordo com o tipo de nó sensor a que pertenciam, NC, NM e NS.

### 3.3.2.1 NÓ COORDENADOR

No modelo estabelecido para esta rede de sensores sem fio, o coordenador é o responsável por recolher as informações adquiridas por todos os nós sensores membros da rede e entrega-la a uma central de dados na qual estas informações serão analisadas.

O NC deverá estar disponível para receber dados a qualquer momento. Para tanto, o coordenador deve verificar continuamente a existência de informação disponível na porta serial. O coordenador lê a informação na porta serial até encontrar o caractere 0x7E, que é um sinalizador que delimita o início do pacote recebido. Os dois bytes que seguem 0x7E são responsáveis por informar o tamanho do pacote. Estes três bytes precisam ser armazenados em uma variável auxiliar, para que de posse do valor do tamanho total do pacote a ser recebido, o coordenador possa criar um vetor com o comprimento necessário para armazenar o mesmo. Em seguida o conteúdo da variável auxiliar é transferido para o vetor recém-criado assim como o restante do pacote que esperava para ser lido na porta

serial. Em seguida o coordenador irá analisar o pacote recebido. Caso o este contenha a informação de que ocorreu uma descarga, NC deverá enviar requisições a cada nó de medição um pacote no qual solicitará que o nó sensor envie os dados colhidos no período correspondente à descarga detectada. Isto é feito desta forma para evitar um excesso de choque de pacotes.

### 3.3.2.2 NÓ DE MEDIÇÃO

Os nós de medição assim como o coordenador da rede, são capazes de enviar e transmitir mensagens. Ao contrário dos coordenadores que devem se comunicar com cada sensor da rede, os nós sensores são configurados para se comunicarem exclusivamente com o coordenador, que irá encaminhar as informações para uma central. Além disso, os nós sensores precisam estar continuamente lendo a entrada de um conversor analógico digital a fim permitir a detecção de descargas, que é a finalidade da rede.

Os microcontroladores do modelo Arduino Uno que possuem a portas capazes de realizar um processo de conversão analógico-digital. Estas portas serão utilizadas para a entrada dos valores de tensão proporcionais aos medidos na malha de aterramento com a finalidade de monitorar o funcionamento da mesma.

O tempo necessário para realizar uma conversão analógico-digital, cerca de 104 us, seria suficiente para realizar centenas de outras operações. Utilizando uma interrupção é possível aproveitar este tempo para que o Arduino continue realizando outras operações. Para utilizar uma interrupção é preciso primeiro escrever uma Rotina de Interrupção, que será chamada quando ocorre a quando a interrupção ocorrer. Em seguida é necessário especificar para o processador quando a interrupção deve ocorrer (GAMMON, 2012)

O registrador ADCSRA é responsável por iniciar e habilitar a conversão AD, assim como especificar o divisor de frequência necessário ao ADC. Abaixo estão especificados cada bit deste registrador e suas funções. As Tabelas 4 e 5 descrevem o funcionamento deste registrador.

Tabela 4. Numeração dos bits do registrador ADCSRA (GAMMON, 2012).

Bit	7	6	5	4	3	2	1	0
ASCSRA	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0

Tabela 5. Descrição dos bits do registrador ADCSRA (GAMMON, 2012).

Bit	Descrição
<b>ADEN</b> ( <i>ADC Enable</i> )	Para habilitar o conversor AD, este bit deve possuir nível lógico 1.
<b>ADSC</b> ( <i>ADC Start Conversion</i> )	A cada vez que o nível lógico 1 é estabelecido neste bit, o conversor AD inicia o processo de conversão. Quando a o processo de conversão termina, o hardware traz o valor do bit de volta para zero. Este bit pode ser utilizado como um <i>flag</i> para verificar se o processo de conversão se completou.
<b>ADFR</b> ( <i>ADC Free Running Select</i> )	Este bit pode ser usado para alternar ADC para o modo <i>free running</i> ou modo de conversão individual. Para habilitar o modo <i>free running</i> este bit deve ter valor 1. Para o modo de conversão individual, se estabelece o valor deste bit em 1.
<b>ADIF</b> ( <i>ADC Interrupt Flag</i> )	Este é um bit de interrupção que é modificado para 1 pelo hardware uma vez que a conversão está completa e os registradores de dados (ADCL e ADCH) são atualizados. A interrupção de conversão AD é executada se ADIE e I-bit em SREG estiverem com valor 1. ADIE é o bit 3 do registrador ADCSRA e SREG é o <i>Status Register</i> . O <i>Status Register</i> contém o resultado da última instrução aritmética executada. I-bit é o sétimo bit do Status e conhecido como <i>Global interrupt enable</i> .
<b>ADIE</b> ( <i>ADC Interrupt Enable</i> )	Quando este bit é estabelecido como um e o I-bit em SREG também, a interrupção de <i>ADC Conversion Complete</i> é ativada.
<b>ADPS</b> ( <i>ADC Prescaler Select Bits</i> )	<b>ADC Prescaler Select Bits (ADPS):</b> Os bits <b>ADPS2</b> , <b>ADPS1</b> e <b>ADPS0</b> são usados para a frequência do <i>clock</i> do circuito. Para que o conversor AD funcione com sua resolução máxima, ele precisa receber frequência de 50 kHz a 200 kHz de acordo com o <i>datasheet</i> ; mas o <i>clock</i> do sistema irá geralmente fornecer uma frequência mais elevada (8 MHz, 10 MHz, 16 MHz etc.). Para reduzir esta frequência para valores adequados utilizam-se <i>ADC prescaler bits</i> .

Portanto, para realizar a conversão analógico-digital, o nó sensor utiliza uma rotina de interrupção que é chamada quando utilizando os bits do registrador ASCSRA. A conversão ocorrerá de modo contínuo até o momento em que haja informações a serem transmitidas. Neste momento uma nova conversão não será iniciada até que os dados anteriormente colhidos sejam devidamente enviados.

Os dados lidos por meio do conversor analógico-digital devem ser armazenados, pois caso seja detectada uma descarga será necessário enviar dados colhidos em instantes anteriores a mesma. Como o sistema deve funcionar por tempo indeterminado e a capacidade de armazenamento do Arduino é limitada, faz-se necessário utilizar um mecanismo que permita ao sistema manter os valores das últimas amostras, e permita a liberação do espaço ocupado por amostras colhidas em a mais tempo, que não são mais

relevantes. Este problema pode ser resolvido utilizando uma técnica de programação chamada de memória circular ou memória em anel.

Memória circular ou Buffer circular é uma estrutura de memória do tipo em que os primeiros dados a serem inseridos são os primeiros a serem retirados (*first in, first out*). Quando novos valores são adicionados a fila, os primeiros valores a serem inseridos vão sendo substituídos. Deste modo os valores ficam armazenados na mesma ordem em que são inseridos.

A memória em anel que foi criada tem base em um simples vetor. Para fazer com que o vetor funcione como uma memória circular foram utilizadas contadores e um marcador de posição. Um dos contadores é serve para percorrer as posições do vetor antes da detecção do evento esperado. As especificações do projeto pedem que sejam mantidas as  $n$  de amostras anteriores ao evento, e sejam colhidas outras  $m$  amostras posteriores. Para tanto um segundo contador é utilizado para contar as amostras após o evento. A posição na qual ocorre o evento é guardada em outra variável.

Na Figura 17 está uma representação do funcionamento de uma memória circular. As posições de cor verde representam as amostras coletadas antes que ocorresse o evento de interesse. A variável posição irá armazenar a posição do vetor na qual os valores lidos ultrapassam o limiar estabelecido. Após a detecção, uma determinada quantidade de amostras é colhida, representada acima pela cor azul.

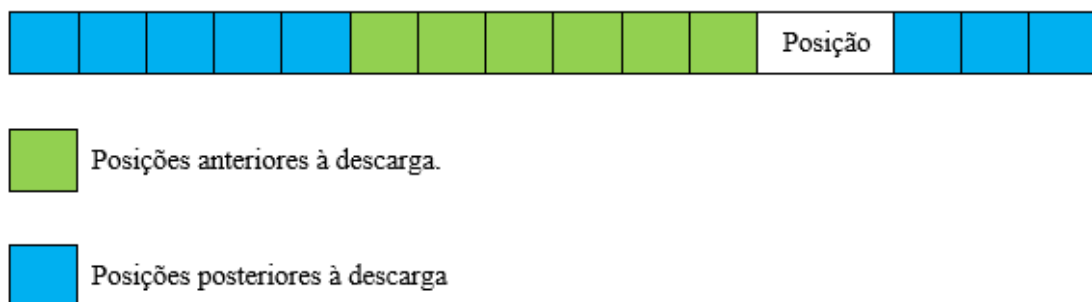


Figura 17. Preenchimento da memória circular antes da reorganização das amostras.

Em seguida, utilizamos nosso conhecimento sobre a posição do vetor em que foi detectado o evento de interesse, a quantidade de amostras anteriores, e a quantidade de amostras posteriores para rearranjar os valores agora de forma cronológica (Figura 18).



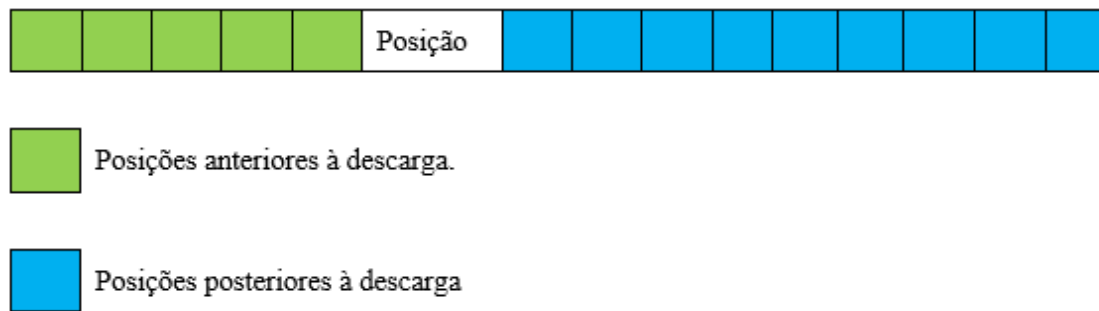


Figura 18. preenchimento da memória circular depois da reorganização das amostras.

### 3.3.2.3 NÓ DE SINCRONIZAÇÃO

Os nós NS são capazes de enviar mensagens no modo *broadcast*. Eles são responsáveis por monitorar continuamente a malha de aterramento até que seja detectada a descarga. Diferentemente dos nós de medição, estes sensores não precisam armazenar amostras para enviá-las ao coordenador da rede.

## 4 RESULTADOS E DISCUSSÕES

Para testar o protocolo proposto foi montado um protótipo da rede proposta em laboratório. A montagem dos nós sensores que compõe a rede pode ser observada na Figura 19.

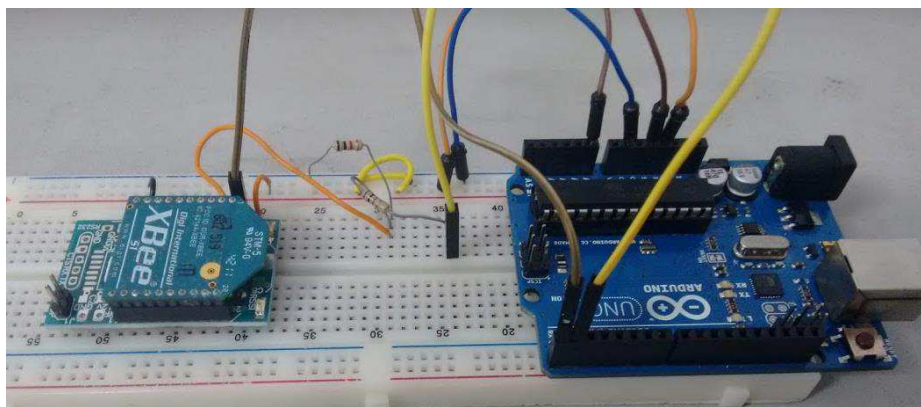


Figura 19. Protótipo de nó sensor

Os rádios XBee também podem ser conectados aos Arduinos por meio de *shields* que fazem as conexões necessárias de modo compacto.

Foram montados 3 nós de medição, 1 nó de sincronização e um nó coordenador. Os nós de medição foram configurados com os endereços MY iguais a 07, 08 e 10. Os nós sensores e o nó de sincronização podem ser observado na Figura 20.

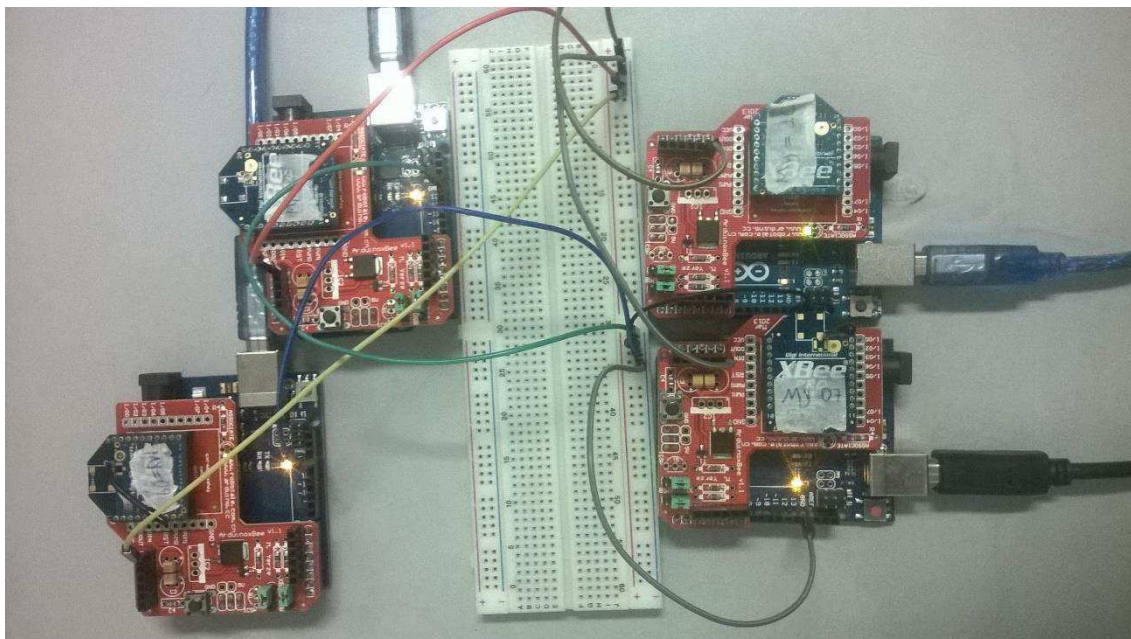


Figura 20. Nós de medição e sincronização.

O nó de sincronização foi configurado com MY igual a 09 e o coordenador possui MY igual a 11. O nó coordenador pode ser visualizado na Figura 21.

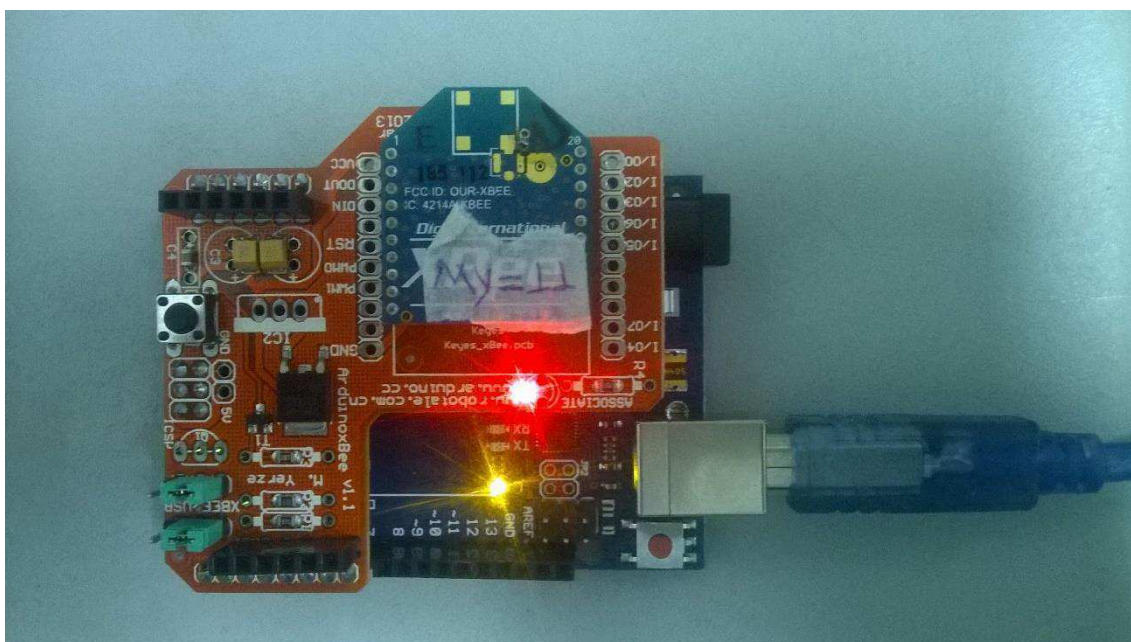


Figura 21. Nó coordenador montado com Arduino e XBee com o uso de *shield*.

Uma das portas analógicas dos Arduinos dos NM dos NS foram conectadas a um ponto comum por meio de fios e uma *protoboard*. Utilizou-se um arranjo com um botão para aplicar o sinal que simularia a descarga. Quando o valor lido ultrapassa o limiar

estabelecido, a rede o protocolo é acionado. O resultado da aplicação de um sinal superior ao limiar é observado na Figura 22. O pacote de sincronização está marcado de azul, os pacotes de requisição estão marcados de vermelho, e um pacote com a as amostras colhidas pelo sensor 07 está marcado de amarelo.

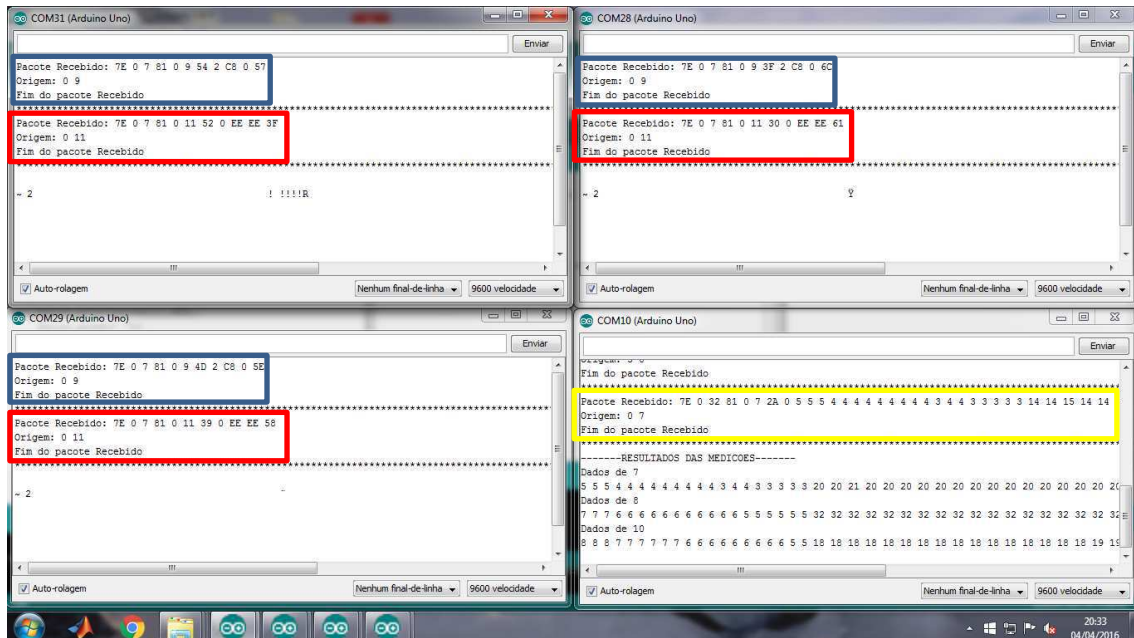


Figura 22. Visualização dos pacotes nos NM e em NC.

Pode-se observar nos monitores seriais o comportamento da rede. Colocando os pacotes recebidos em ordem é possível entender melhor o seu funcionamento. Primeiramente, ao detectar uma variação no sinal aplicado, o nó de sincronização, endereço 09, envia uma mensagem broadcast para todos os membros da rede (Figura 23).

```
Pacote Recebido: 7E 0 7 81 0 9 54 2 C8 0 57
Origem: 0 9
Fim do pacote Recebido
*****
```

Figura 23. Broadcast recebido pelos nós sensores.

Em seguida o coordenador enviou mensagens a cada nó da rede, requerendo que fossem enviados os valores lidos (Figura 24).

```
*****
Pacote Recebido: 7E 0 7 81 0 11 52 0 EE EE 3F
Origem: 0 11
Fim do pacote Recebido
*****
```

Figura 24. Quadro de requisição provindo do coordenador.

Os nós de medição respondem enviando os valores lidos. Parte de um pacote de resposta dos nós de medição ao coordenador está ilustrado na Figura 25.

```
Pacote Recebido: 7E 0 32 81 0 7 2A 0 5 5 5 4 4 4 4 4 4 4 4 4 3 4 4 3 3 3 3 3 14 14 15 14 14 1
Origem: 0 7
Fim do pacote Recebido
*****
```

Figura 25. Pacote recebido pelo coordenador com valores lidos pelos nós de medição

E por fim o nó coordenador organiza os valores recebidos e os exibe em forma de lista. A lista está ilustrada na Figura 26.

```
*****
-----RESULTADOS DAS MEDICOES-----
Dados de 7
5 5 5 4 4 4 4 4 4 4 4 4 3 4 4 3 3 3 3 3 20 20 21 20 20 20 20 20 20 20 20 20 20 20
Dados de 8
7 7 7 6 6 6 6 6 6 6 6 6 5 5 5 5 5 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32
Dados de 10
8 8 8 7 7 7 7 7 7 6 6 6 6 6 6 6 6 5 5 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 19
```

Figura 26. Resultados das medições nos nós sensores.

Os valores recebidos pelo coordenador podem então ser entregues a uma aplicação que irá utilizar os valores amostrados juntamente com os intervalos de tempo para que os valores de tempo medidos.

## 5 CONCLUSÃO

A principal contribuição desse trabalho foi o desenvolvimento do protocolo de comunicação a ser utilizado pelos nós da rede de aquisição de dados para monitoramento de uma malha de aterramento. O Protocolo foi definido e testado qualitativamente.

Fica comprovado que o protocolo de comunicação permite aos nós de medição membros da rede entregar os dados lidos ao coordenador de modo eficiente e evitando colisões e perda de pacotes.

Ainda como trabalhos futuros, sugere-se a criação de uma interface gráfica que plote as amostras lidas pelos nós sensores. A interface deverá utilizar os valores dos intervalos de tempo medidos pelos nós sensores que serão utilizados pelo *software* para recriar a ordem cronológica do evento.

Como trabalhos futuros, são sugeridos novos testes realizados com microcontroladores que possuam conversores A/D com taxas de amostragens mais elevadas, que permitam amostrar sinais de alta frequência, como no caso de descargas atmosféricas.

## REFERÊNCIAS

AKYILDIZ, I. F. et al **Wireless sensor network: a survey**. [S.l.]: [s.n.], v. Computer networks 38.4, 2002.

ARDUINO. Arduino - Home. **Arduino - Home**, 2016. Disponível em: <<https://www.arduino.cc/>>. Acesso em: 12 Agosto 2016.

BURATTI, C. et al **An Overview on Wireless Sensor Networks Technology and Evolution**. Sensors, v. 9, 2009.

CASTRO, M. S. D. **Sistema de Aquisição de Dados de Malhas de Aterramento Submetidas a Descargas Atmosféricas**. Campina Grande: UFCG, 2016.

GAMMON, N. Gammon Software Solutions. **Gammon Forum**, 2012. Disponível em: <<http://gammon.com.au/interrupts>>. Acesso em: 24 set. 2016.

IEEE. **IEEE Standard 802.15.4**. [S.l.]. 2011.

KARL, H. et al **Protocols and architectures for wireless sensor networks**. [S.l.]: John Wiley & Sons, 2007.

LONG, X. et al **Online monitoring of substation grounding grid conditions using touch and step voltage sensors**. [S.l.]: IEEE Transactions on Smart-Grid., 2012.

LOUREIRO, A. A. et al **Redes de sensores sem fio**. Simpósio Brasileiro de Redes de Computadores (SBRC). [S.l.]: [s.n.]. 2003.

SAUSEN, P. S. et **Gerenciamento integrado de energia e controle de topologia em redes de sensores sem fio**. Universidade Federal de Campina Grande. Grande-Campina Grande, Paraíba, Brasil. 2008.

SOARES, S. A. F. **Rede de sensores sem fio para localização e monitoramento de pequenos ruminantes**. Universidade Federal do Vale do São Francisco. Juazeiro-BA. 2012.

SOUSA, M. P. **Diversidade Cooperativa Adaptativa Aplicada**. Universidade Federal de Campina Grande. Campina Grande-Paraíba-Brasil. 2009.

VALLE, O. T. et al **Expansion of the available use classes in IEEE 802.15.4 networks for usage in industrial environment**. 2012 International Symposium on Wireless Communication Systems (ISWCS). [S.l.]: IEEE. 2012. p. 133-135.

VIKA CONTROLS. Vika Controls - Módulos XBee. **Vika Controls - Soluções em Instrumentação**, 2016. Disponível em: <<http://vikacontrols.com.br/produtos/telemetria/modulos-xbee/>>. Acesso em: 22 out. 2016.