

*Projeto e Implementação de um Sistema de
Monitoramento de uma Estufa Agrícola Utilizando
Rede Hierárquica de Sensores Sem Fio.*

Hugo Vaz Sampaio

Dezembro / 2016

Projeto de pesquisa de Mestrado em Ciência da Computação

Projeto e Implementação de um Sistema de Monitoramento de uma Estufa Agrícola Utilizando Rede Hierárquica de Sensores Sem Fio

Este documento corresponde ao Projeto de Pesquisa apresentado à Banca Examinadora para defesa de dissertação no curso de Mestrado em Ciência da Computação da Faculdade Campo Limpo Paulista.

Campo Limpo Paulista, 9 de dezembro de 2016.

Hugo Vaz Sampaio

Shusaburo Motoyama (Orientador)

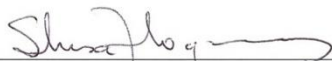
Faculdade Campo Limpo Paulista
Programa de Mestrado em Ciência da Computação

“Projeto e Implementação de um Sistema de Monitoramento de uma Estufa Agrícola Utilizando Rede Hierárquica de Sensores Sem Fio”

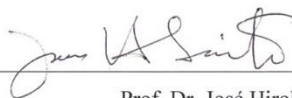
Hugo Vaz Sampaio

Dissertação de Mestrado apresentado ao Programa de Mestrado em Ciência da Computação da Faculdade Campo Limpo Paulista, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Membros da Banca:



Prof. Dr. Shusaburo Motoyama
(Orientador-FACCAMP)



Prof. Dr. José Hiroki Saito
(FACCAMP)



Prof. Dr. Michel Daoud Yacoub
(UNICAMP)

Campo Limpo Paulista, 09 de dezembro de 2016.

FICHA CATALOGRÁFICA

Dados Internacionais de Catalogação na Publicação (CIP)

Câmara Brasileira do Livro, São Paulo, Brasil

Sampaio, Hugo Vaz

Projeto e implementação de um sistema de monitoramento de uma estufa agrícola utilizando rede hierárquica de sensores sem fio / Hugo Vaz Sampaio. Campo Limpo Paulista, SP: FACCAMP, 2016.

Orientador: Prof. Dr. Shusaburo Motoyama

Dissertação (mestrado) – Faculdade Campo Limpo Paulista – FACCAMP .

1. Redes de sensores sem fio. 2. RSSF. 3. Estufa. 4. Zigbee. I. Motoyama, Shusaburo. II. Faculdade Campo Limpo Paulista. III. Título.

CDD-005.621

Agradecimentos

Agradeço aos meus pais por me ensinarem o valor e a importância do estudo para a minha vida.

Agradeço a minha esposa e filho, pelo amor, incentivo e apoio incondicional.

Aos colegas de mestrado que ajudaram direta ou indiretamente, em especial o Paulo e o Sérgio.

Ao Prof. Motoyama, pela orientação, apoio e confiança.

Ao Prof. Osvaldo, pelo auxílio financeiro que permitiu a aquisição dos equipamentos utilizados nesta dissertação.

Agradeço à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pela concessão de bolsa durante todo o período de realização do Curso de Mestrado.

Resumo

Na presente dissertação é proposto um sistema de monitoramento de uma estufa agrícola utilizando rede de sensores sem fio (RSSF). Esse monitoramento tem como principais parâmetros de controle a temperatura e umidade do ar, a umidade da terra e a luminosidade. Para a coleta de dados desses parâmetros é proposta, neste trabalho, uma rede de sensores sem fio em uma forma hierárquica.

Nessa configuração os sensores agregados com todas as funcionalidades de coleta, processamento e transmissão de dados sem fio, constituindo nós sensores, localizam-se no nível mais baixo da hierarquia. Nos níveis intermediários encontram-se os nós roteadores utilizados para transportar os dados dos nós sensores até o nó coordenador. No nível mais alto, está o nó coordenador utilizado para enviar os dados a uma estação base, onde esses dados são armazenados e processados.

Os detalhes do projeto, assim como uma implementação simplificada desse sistema de monitoramento são apresentados ao longo da dissertação. O número total de nós sensores possíveis de acomodar no sistema de monitoramento proposto foi estimado teoricamente em dois diferentes casos. No primeiro caso, considerando o total de número de endereços disponíveis para identificação dos nós sensores e roteadores, uma expressão analítica foi obtida para o cálculo do número de sensores que o sistema pode utilizar. No segundo caso, o número de sensores foi estimado considerando que o tráfego de cada um dos nós sensores tem uma distribuição de probabilidade, assim modelando o sistema hierárquico proposto como uma rede de filas. Consequentemente, foram analisados dois diferentes modelos de rede de filas. No primeiro modelo, o tráfego de realimentação do coordenador para ativação ou desativação dos atuadores não foi levado em conta, e no segundo modelo, o tráfego de realimentação foi considerado. As análises mostraram que dependendo das taxas de pacotes dos nós sensores, um sistema de monitoramento de grande porte poderá ser implementado.

Palavras chaves: Redes de Sensores sem fio, RSSF, estufa, Zigbee

Abstract

A greenhouse monitoring system using hierarchical wireless sensor network (WSN) is presented in this dissertation. The main parameters required to monitor and control a greenhouse are air humidity and temperature, ground moisture and environment lightness. For the data gathering of these parameters, a hierarchical WSN is presented in this work. In this configuration, the sensors, aggregated with all gathering functionalities, processing and wireless data transmission capabilities, denoted as sensor nodes, are in the lowest level. The router nodes are provided in the middle level to transmit data from sensor nodes to a controller named coordinator node. The coordinator node, in the highest level, is used to communicate with a central base, where all data received are stored and processed. The details of the project and a simple implementation of this monitoring system are presented. The total number of sensor nodes possible to accommodate in the proposed monitoring system is estimated for two different cases. In first case, considering the total number of addresses available for identification of the sensor and router nodes, an analytical expression is derived for calculation of number of sensor nodes the system can use. In second case, the number of sensor nodes is estimated considering that traffic of sensor nodes has a distribution and modeling the proposed hierarchical WSN as a network of queues. Two different models of network of queues are analyzed. In first model, the feedback traffic from coordinator for activation or deactivation of the actuator nodes is not taken into account, and in second model, the feedback traffic from coordinator is considered. The analysis showed that depending on the packet rates of sensor nodes, a very large monitoring system can be implemented.

Keywords: Wireless Sensor Networks, WSN, Greenhouse, Zigbee

Glossário

COM – Porta serial de comunicação.

dBm – Decibel milliwatt.

DC – Direct Current.

GND – Entrada Ground ou terra.

IEEE - Instituto de Engenheiros Eletricistas e Eletrônicos.

IoT – *Internet of Things* ou Internet das Coisas

LDR – *Light Dependent Resistor.*

LOG – Arquivo com registro de mensagens recebidas.

MAC – *Media Access Control.*

mAh – miliampere-hora.

MEMS – *Microelectronic mechanical system.*

PAN ID – Identificador de PAN.

PWM – *Pulse Width Modulation.*

RSSF – Redes de Sensores Sem Fio.

RX – Porta serial de Recebimento.

SoC - *System-on-a-chip.* Chip integrado com rádio.

TX – Porta serial de Transmissão.

USB - *Universal Serial Bus.*

WPAN – *Wireless Personal Area Network.*

XCTU – Software proprietário para configurar antenas Xbee.

Índice de Figuras

Figura 1. Parâmetros para monitoramento e controle. Adaptado de Pawlowski et al. (2009).....	17
Figura 2. Gráfico de variação de temperatura °C em 96 horas. Benavente (2010).	19
Figura 3. Gráfico de distância do sinal entre nó coordenador e nó sensor. Benavente (2010).....	19
Figura 4. Estufa agrícola utilizada para testes. Adaptado de Akshay et al (2012).	23
Figura 5. Exemplo de uma RSSF em estufa de larga escala.	25
Figura 6. Rede simplificada para implementação prática.....	27
Figura 7. Exemplo de rede Zigbee.....	30
Figura 8. Tela de configuração do nó coordenador no software XCTU.	31
Figura 9. Conexão física entre o Arduíno Uno e antena Xbee Zigbee.	32
Figura 10. Exemplo de um pacote de Requisição de Transmissão.....	35
Figura 11. Comandos para a criação de um pacote Zigbee.	36
Figura 12. Exemplo de pacote de transmissão para recebido.	36
Figura 13. Tela da ferramenta de geração de pacotes API do XCTU.	37
Figura 14. Conexão física do nó sensor de temperatura e umidade do ar.	38
Figura 15. Conexão física do nó sensor de luminosidade.	39
Figura 16. Conexão física do nó sensor de umidade da terra.	39
Figura 17. Bomba de água utilizada no nó atuador.	40
Figura 18. Comandos da função ligabomba().....	41
Figura 19. Conexão física do nó atuador.	42
Figura 20. Tela do log de pacotes gerado no software XCTU.	43

Figura 21. Exemplo de arquivo de log de pacotes.....	43
Figura 22. Exemplo de uma linha do arquivo leitor de log.	44
Figura 23. Tela do software visualizador de log.....	46
Figura 24. Foto da RSSF desenvolvida.	48
Figura 25. Gráfico de variação de luminosidade.	49
Figura 26. Nó sensor de ar com aquecedor de ar elétrico.....	49
Figura 27. Gráfico de variação de umidade e temperatura do ar.....	50
Figura 28. Sensor de umidade submerso em um copo de água.	50
Figura 29. Gráfico de variação do nível de umidade.....	51
Figura 30. Foto dos nós sensores da RSSF.....	52
Figura 31. Tela do software leitor de log com informações do experimento 2. ..	53
Figura 32. Gráfico de variação de temperatura e umidade do ar em 36 horas. ...	54
Figura 33. Gráfico de variação de luminosidade em 36 horas.	54
Figura 34. Gráfico de variação de umidade da terra.....	55
Figura 35. RSSF com 3 nós sensores e um atuador. Em amarelo o nó sensor de ar, em branco o nó sensor de luminosidade, em vermelho o nó sensor da terra e em preto o nó atuador.	57
Figura 36. Sensor de terra e mangueira da bomba de água.	58
Figura 37. Nó atuador com uma bomba de água.	58
Figura 38. Gráfico de variação de temperatura e umidade do ar.....	59
Figura 39. Gráfico de variação de luminosidade.	60
Figura 40. Gráfico de variação de umidade da terra.....	60
Figura 41. Gráfico detalhado de um acionamento automático da bomba de água.	61
Figura 42. RSSF hierárquica proposta.....	66
Figura 43. Variação no tempo de espera $E\{TC\}$ em milissegundos.	67
Figura 44. Curvas com variação de ρR e ρC	69

Figura 45. Variação ETR de ρC e ρr	70
Figura 46. Tempo total de espera $E\{T\}$ em segundos.	71
Figura 47. Exemplo de uma RSSF hierárquica com feedback em um ramo.....	72
Figura 48. Exemplo de filas em série.	111
Figura 49. Exemplo de filas com realimentação.	111

Índice de Tabelas

Tabela 1. Tempo de vida estimado para o nó sensor. Benavente (2010).	20
Tabela 2. Tabela comparativa entre nós de amostragem fixa e variável. Benavente (2010).....	20
Tabela 3. Tabela comparativa de hardware utilizado em artigos referenciados..	22
Tabela 4. Quantidade de nós da RSSF.....	26
Tabela 5. Quadro com definições de camadas Zigbee.	29
Tabela 6. Detalhes de um pacote tipo requisição de transmissão.....	35
Tabela 7. Detalhes de um pacote tipo recebido.	37
Tabela 8. Exemplo de um pacote de cada tipo de nó sensor.	45
Tabela 9. Comparação de erros de pacotes enviados por cada nó sensor.....	61
Tabela 10. Número total de nós sensores possíveis de acomodar com diferentes taxas de pacotes.	68
Tabela 11. Números totais de nós sensores e atuadores possíveis de acomodar com diferentes taxas de pacotes.....	73

Sumário

Capítulo 1 - Introdução	14
1.1 Objetivos	15
1.2 Contribuições	15
1.3 Organização Deste Trabalho	15
Capítulo 2 – Trabalhos Relacionados	17
Capítulo 3 - Projeto de Uma RSSF Para Estufas Agrícolas de Qualquer Tamanho	23
Capítulo 4 - Implementação Prática da Rede Hierárquica de Sensores Sem Fio	27
4.1 Projeto dos Nós	28
4.1.1 Projeto da Parte do Rádio	28
4.1.2 Projeto da Parte do Hardware	31
4.2 Projeto do Nó Coordenador e do Nó Roteador	32
4.3 Montagem de Pacotes	34
4.4 Nós Sensores	38
4.5 Nó Atuador	40
4.6 Log de Pacotes XCTU	42
4.7 Ferramenta Leitor de Log	44
Capítulo 5 – Resultados Experimentais	47
5.1 Experimento 1 – Teste dos nós sensores	47
5.2 Experimento 2 – Captura de Dados por 36 Horas	51
5.3 Experimento 3 - Captura de Dados por 72 Horas com Atuador	55
Capítulo 6 - Estimção Teórica dos Números de Nós Sensores e Atuadores	62
6.1 Considerando Limite de Endereço	62
6.2 Considerando Rede de Filas	64

6.2.1 Rede de Sensores sem Fio	64
6.2.2 Com Realimentação (Feedback).....	71
Conclusão	74
Referências	75
Apêndices	77
Apêndice A – Algoritmo na linguagem C para o nó sensor do ar.	77
Apêndice B – Algoritmo em linguagem C para o nó sensor do solo.	78
Apêndice C – Algoritmo em linguagem C para o nó sensor de luz.	79
Apêndice D - Algoritmo em linguagem C para o nó coordenador.....	80
Apêndice E - Algoritmo em linguagem C para o nó atuador.	82
Apêndice F – Artigo WCF 2016.....	83
Apêndice G - Artigo 2 WCF 2016.....	92
Apêndice H – Artigo ICCCN 2017 – B1.....	100
Anexos	110
Anexo A – Fundamentação teórica de Teoria de Filas.....	110

Capítulo 1 - Introdução

O avanço tecnológico em sistemas microeletrônicos (MEMS), eletrônicos digitais e comunicação sem fio, permitiram o desenvolvimento de dispositivos de comunicação sem fio, de pequeno porte, baixo custo, baixo consumo de energia e multifuncionais, Akyildiz et al. (2002). Estes dispositivos quando organizados em uma rede sem fio, são denominados nós. Um conjunto de nós forma uma rede denominada Rede de Sensores Sem Fio (RSSF). A utilização de RSSFs está proporcionando inovações em várias áreas de aplicação, tais como, médica, agrícola, controle de tráfego, controle ambiental entre outras.

Na área de agricultura, a RSSF pode ser usada para obter as informações necessárias que uma planta necessita para crescer, como temperatura e umidade do ar, umidade da terra, luminosidade, entre outros. As variações dessas informações são relativamente lentas. Assim, os nós sensores sem fio, alimentados por bateria, podem coletar e enviar essas informações esporadicamente por um longo período. Dessa forma, um sistema de monitoramento agrícola pode, por exemplo, mostrar esses dados a uma equipe de plantão ou automaticamente tomar providências como regar, ligar um sistema de ventilação e deixar o ambiente mais escuro.

Um dos tipos de agricultura que está sendo cada vez mais utilizado, devido à sua eficiência produtiva, é a agricultura em estufas. Esse método busca minimizar a variação do clima e seus efeitos adversos decorrentes de altas ou baixas temperaturas do ar. Além disso, possibilita controlar a qualidade da colheita e minimizar o tempo de crescimento da planta. Com a utilização de uma RSSF dentro de uma estufa, é possível coletar, com precisão, as informações climáticas, permitindo um controle otimizado da plantação.

Na literatura, Kalaivani et al (2011), Pawlowski et al. (2009), Zagade e Kawitkar (2012), encontram-se vários artigos com propostas e implementações de controle de estufa. Entretanto, a maioria desses artigos trata de projetos, simulações e testes iniciais de RSSFs. Não há, ainda, um número expressivo de artigos que tratam de RSSF hierárquica para estufas, onde a RSSF controla centenas ou milhares de nós sensores, denominada neste trabalho de RSSF hierárquica.

1.1 Objetivos

O objetivo principal dessa pesquisa é propor um projeto de sistema de monitoramento para estufas de qualquer tamanho, utilizando RSSF. Adicionalmente, para avaliar, enriquecer e complementar os estudos, visa-se testar o funcionamento do projeto, através de uma implementação prática e o desenvolvimento de um software da parte central do sistema proposto. Posteriormente, será feito um estudo teórico para estimar as limitações de uma RSSF, utilizando teoria de filas.

1.2 Contribuições

Estima-se que o resultado do desenvolvimento do projeto de pesquisa proposto, com a sua implementação prática, propiciará algumas contribuições para a área da agricultura, especificamente para plantações em estufa com o uso de RSSF hierárquica para uso em larga escala.

Dentre essas contribuições, podem-se destacar:

- a) o detalhamento de projeto de uma RSSF para estufas de larga escala, em conjunto com o projeto de hardware dos nós de rede, utilizando antenas Zigbee;
- b) o desenvolvimento de um software, denominado Leitor de Log. Este software processa arquivos de texto que contém o histórico de dados gerados pela RSSF, denominados arquivos de log de dados, e exibe as informações em tabelas e gráficos;
- c) a modelagem de fila de uma rede hierárquica para RSSFs, de forma a permitir escalabilidade, considerando os limites do hardware utilizado.

1.3 Organização Deste Trabalho

Este trabalho está organizado em sete Capítulos. No primeiro capítulo, é feita a introdução ao assunto de RSSF aplicadas em estufas agrícolas, indicando os objetivos, contribuições da dissertação e os resultados esperados. No Capítulo 2 a seguir, são apresentados os principais artigos encontrados na literatura sobre monitoramento de plantações estufas agrícolas com a utilização de RSSF. Os princípios que norteiam o projeto de uma estufa agrícola e a sua configuração hierárquica de RSSF são discutidos no Capítulo 3. No Capítulo 4, é apresentado o projeto de implementação prática

simplificada de uma RSSF apropriada para estufas agrícolas. No Capítulo 5, são apresentados os resultados de quatro experimentos com a RSSF projetada. No Capítulo 6, um estudo teórico de filas é desenvolvido para estimar a quantidade de nós sensores e atuadores possíveis de acomodar em uma RSSF, de modo a permitir escalabilidade do sistema. O Capítulo 7 apresenta as conclusões e indicações de trabalhos futuros. Adicionalmente, fazem parte da dissertação oito Apêndices e um Anexo. Nos apêndices A até E são descritos os algoritmos utilizados na configuração dos nós desenvolvidos. Os apêndices F, G e H são artigos aceitos para publicação, sendo o apêndice H um artigo submetido em conferência internacional.

Capítulo 2 – Trabalhos Relacionados

No *survey* de Kalaivani et al (2011), é demonstrado que há uma grande quantidade de artigos sobre RSSF utilizadas em agricultura de precisão. Nos artigos citados no *survey*, foram realizados experimentos para capturar dados sobre uma grande variedade de parâmetros, e os mais utilizados são: temperatura, velocidade do vento, direção do vento, umidade do solo, PH do solo, identificação de planta, umidade da folha, detecção de ervas daninha. Em plantio em estufas, os parâmetros relacionados ao vento não são considerados visto que a estufa fornece proteção física às plantas.

O artigo de Pawlowski et al. (2009) mostra que plantações em estufas possuem dois grupos de sistemas independentes com diferentes problemas:

- a) Grupo 1: alterações climáticas como temperatura e umidade do ar, vento e radiação solar, em vermelho na Fig. 1. Dentro de uma estufa é possível controlar as alterações climáticas, denominadas Variáveis Controláveis, mostradas em verde na Fig. 1.

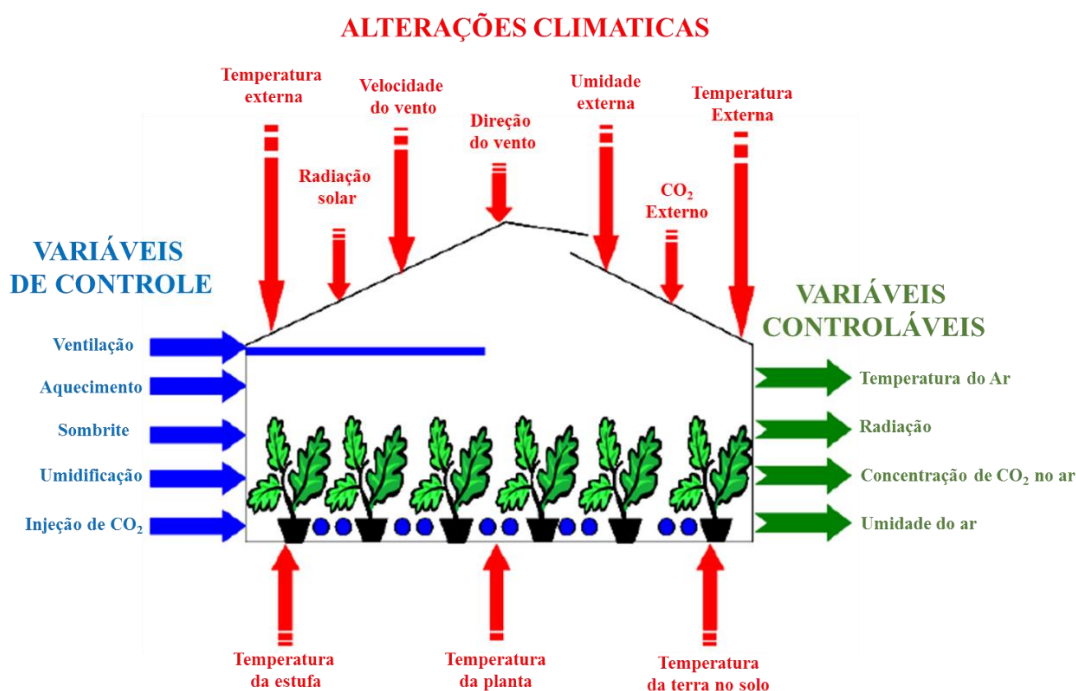


Figura 1. Parâmetros para monitoramento e controle. Adaptado de Pawlowski et al. (2009).

As variáveis controláveis são temperatura/umidade do ar, luminosidade, concentração de CO₂, entre outros, que afetam diretamente no crescimento das plantas. As Variáveis de Controle são utilizadas para alterar as variáveis controláveis, através de

atuadores que podem acionar um sistema de ventilação, aquecimento, umidificação do ar, entre outros, mostrado em azul.

Dessa forma, dentro de uma estufa é possível monitorar e controlar a variação climática de acordo com a necessidade de cada planta.

- b) Grupo 2: nesse grupo, os problemas são variáveis de fertirrigação, ou irrigação com fertilizantes. Informações como temperatura/umidade da terra, composição da terra, entre outros, são utilizadas para decisões como volume e/ou periodicidade da fertirrigação.

No artigo também foi apresentada uma técnica de controle para estufas, baseada em eventos, por meio de simulações com o aplicativo Matlab. Para simular os eventos foram definidos valores de limiares, máximos e mínimos, para as variáveis temperatura/umidade do ar, onde o valor acima ou abaixo do limiar é considerado um evento.

Um problema em RSSF é a vida útil de um nó sensor alimentado por bateria. Benavente (2010) propõe uma solução para economizar energia e prolongar a vida útil da bateria em nós sensores. Em sua dissertação foi desenvolvida uma RSSF para monitoramento ambiental em um vinhedo utilizando intervalo de amostragem variável nos nós sensores. Como exemplo de amostragem variável foram definidos limiares máximo e mínimo para o valor da temperatura coletado pelos sensores. Foram desenvolvidos nós da rede com antenas Zigbee-PRO e sensores de temperatura e umidade relativa do ar, que capturam e transmitem dados para um dispositivo central.

Benavente (2010) realizou alguns experimentos para avaliar a RSSF desenvolvida. O primeiro experimento foi comparar a exatidão dos dados coletados por nove sensores de temperatura do ar, Sensirion SHT11 que tem precisão de 2°C, com o sensor HOBO Pro v2, que tem precisão de 0,2°C.

O primeiro experimento foi executado durante 96 horas ininterruptas, e os dados capturados de cada nó sensor permitiu a comparação da precisão dos valores capturados pelos sensores. Um gráfico foi gerado, demonstrando, em todos os sensores, igual comportamento na variação de temperatura em relação ao tempo, com diferença de até um grau de temperatura, conforme Fig. 2.

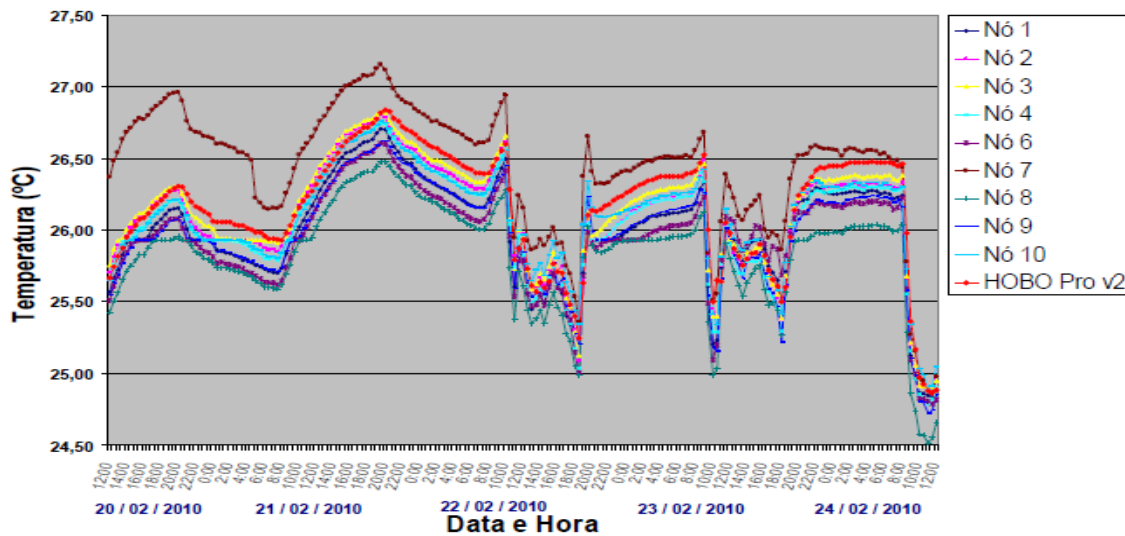


Figura 2. Gráfico de variação de temperatura °C em 96 horas. Benavente (2010).

O segundo experimento testou alcance de conectividade do sinal entre os nós sensores e a estação base. A Fig. 3 mostra um gráfico do resultado das medições de alcance onde a potência de transmissão foi variada entre 0 e -25dBm (decibel milliwatt) para determinar o alcance máximo do sinal. Os nós foram dispostos em diferentes alturas dentro da estufa para comparação.

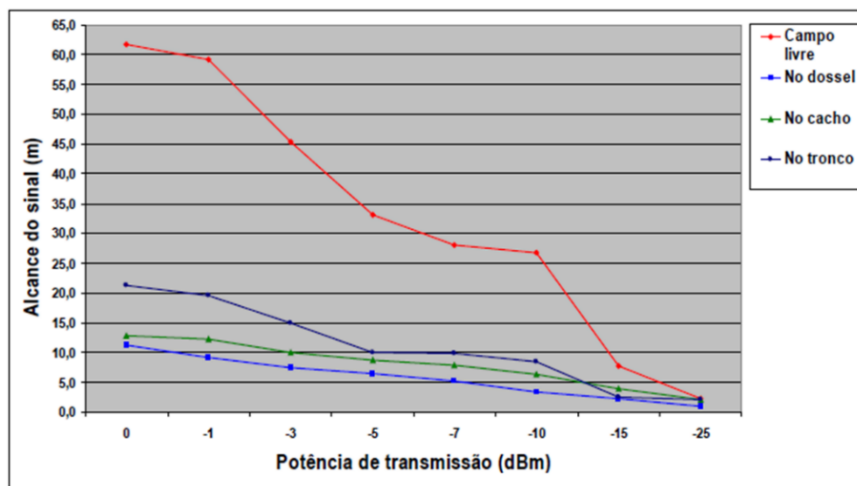


Figura 3. Gráfico de distância do sinal entre nó coordenador e nó sensor. Benavente (2010).

Após o desenvolvimento da RSSF e dos experimentos, Benavente estimou o tempo médio de vida útil de um nó sensor quando alimentado por bateria, considerando dois tipos de baterias com 2100 e 2500 mAh (Miliampere/hora), dois tamanhos de pacotes de dados com 44 e 24 bytes, e intervalos de amostragem de 3, 5, 10, 15, 30 e 60 minutos. A Tabela 1 mostra os resultados dos cálculos indicando o tempo estimado de vida de um nó sensor com valores em horas, dias e meses.

Tabela 1. Tempo de vida estimado para o nó sensor. Benavente (2010).

Intervalo de amostragem	Tempo de vida estimado	Tamanho do pacote de dados			
		44 bytes		24 bytes	
		2.100 mAh	2.500 mAh	2.100 mAh	2.500 mAh
3 min	horas	863	1.028	1.463	1.741
	dias	36	43	61	73
	meses	1,2	1,4	2,0	2,4
5 min	horas	1.350	1.607	2.194	2.612
	dias	56	67	91	109
	meses	1,9	2,2	3,0	3,6
10 min	horas	2.340	2.786	3.510	4.179
	dias	98	116	146	174
	meses	3,3	3,9	4,9	5,8
15 min	horas	3.097	3.687	4.388	5.223
	dias	129	154	183	218
	meses	4,3	5,1	6,1	7,3
30 min	horas	4.578	5.451	5.850	6.965
	dias	191	227	244	290
	meses	6,4	7,6	8,1	9,7
60 min	horas	6.017	7.164	7.020	8.357
	dias	251	298	293	348
	meses	8,4	9,9	9,8	11,6

Com as informações da tabela pode-se comparar um nó com intervalo de amostragem fixo e outro de amostragem variável. Um nó sensor denominado Nó 1, com uma bateria de 2100 mAh, envia pacotes de dados de tamanho 24 bytes, e intervalo de amostragem fixo de 5 minutos. Nestas condições é estimado que a vida útil da bateria será de 2194 minutos.

Um segundo nó, denominado Nó 2, com intervalo de amostragem variável, onde se a temperatura do ar estiver entre 23°C e 26°C o intervalo de coleta de dados será de 30 minutos. Caso a temperatura situe-se fora do intervalo, a coleta de dados será feita a cada 5 minutos. Foi considerado que o nó 2 realizou amostragem de dados a cada 30 minutos durante 75% da vida útil da bateria, e a cada 5 minutos nos demais 25%, desse modo a vida útil da bateria do nó 2 será de 4936 horas, uma sobrevida de mais de 100% comparado ao nó 1, como mostra a Tabela 2.

Tabela 2. Tabela comparativa entre nós de amostragem fixa e variável. Benavente (2010).

Id do nó	Horas	Dias	Meses
Nó 1	2.194	91	3,0
Nó 2	4.936	206	6,9

Em Mohanty e Patil (2013), foram selecionados três parâmetros (luminosidade, temperatura e umidade do ar) para serem monitorados em uma estufa. Esses parâmetros são essenciais para o crescimento de uma planta, e são utilizados para garantir uma

colheita mais produtiva. O artigo apresenta uma RSSF, utilizando antenas MICAz no padrão IEEE 802.15.4, e nós sensores para medir os parâmetros acima. Também são avaliadas a confiança e a habilidade da rede para detectar o microclima em uma estufa.

Foi conduzido um experimento com duração total de quatro horas e coletas de dados realizadas com intervalo de 30 minutos. Os resultados indicam que os dados foram capturados e transmitidos sem erros, mostrando a robustez desta RSSF.

No artigo de Ahonen et al. (2008), foi desenvolvida uma RSSF com nós da rede utilizando placas microcontroladoras com rádio Sensinodes, para verificar a variação de microclima em diferentes alturas em uma estufa. O rádio da placa Sensinodes utiliza protocolo 6LoWPAN, que habilita a transmissão de pacotes IPv6 comprimidos através de redes IEEE 802.15.4. Quatro nós com sensores de luminosidade, temperatura e umidade do ar foram dispostos em diferentes alturas dentro da estufa. O intervalo de captura dos dados foi de quatro minutos durante três horas. Os resultados indicam que a RSSF capturou com sucesso as informações de variação de temperatura e umidade em diferentes alturas da estufa. Como resultado foi verificado que a parte mais alta da estufa apresentou para a temperatura do ar, em média, 2°C a mais que o ponto mais baixo da estufa.

Silva e Fruett (2010) desenvolveram uma RSSF para o monitoramento de temperatura e umidade relativa do ar na Floresta Amazônica. A RSSF utilizou o protocolo IEEE 802.15.4 e consistiu em três nós de rede, sendo dois nós sensores e um nó coordenador. Um experimento foi realizado em uma torre de observação de 51 metros de altura na floresta Amazônica. O experimento durou duas horas e o intervalo de captura dos nós sensores foi de 10 segundos. Um nó sensor foi posicionado na base da torre e o outro no topo, a fim de verificar diferenças na variação de temperatura e umidade do ar acima e abaixo do dossel da floresta. Com os dados obtidos, foi possível observar que abaixo do dossel da floresta a temperatura e umidade do ar mantiveram-se estáveis, enquanto acima do dossel houve variação dos mesmos.

No artigo de Akshay et al (2012) foi desenvolvido uma RSSF para monitorar a variação de temperatura e umidade do ar dentro de uma estufa agrícola. Os nós da rede utilizaram antenas no padrão Zigbee para a comunicação entre os nós sensores e o nó coordenador. A estufa citada no artigo possuía bombas de água para irrigação do solo, sistema de *fogging* (vapor de água) e um sistema de ventilação que poderiam ser acionados caso irregularidades sejam constatadas nos dados coletados. Neste artigo a parte de nós atuadores não foi desenvolvida.

Em Shivasaisomarathi e Akaravam (2010) foi proposto um sistema hierárquico para monitoramento de estufas utilizando RSSF. Este sistema é composto por um centro de monitoramento, nós sensores e equipamentos de controle. Na proposta, os nós sensores foram dispostos dentro de uma estufa, e são responsáveis pela aquisição periódica das informações ambientais e envio destes dados para um centro de controle. O centro de controle analisa os dados obtidos, e toma decisões enviando uma mensagem de controle para um equipamento de controle. Este equipamento irá regular o ambiente, baseado em parâmetros. A RSSF proposta é dividida em clusters, cada cluster é composto por um *cluster head* também denominado nó sink, e nós membros (nós sensores e controle). O nó sink é definido por um algoritmo de eleição de um nó sensor. O centro de controle possui uma antena para onde os nós sink irão encaminhar os pacotes. Todo o controle e gerenciamento da rede, análise e armazenamento dos pacotes, e decisão para ativar equipamentos de controle, são feitos pelo centro de monitoramento.

Em Quan et al (2016) é proposto um sistema de monitoramento para estufas utilizando protocolo Zigbee. Oito nós sensores foram desenvolvidos e distribuídos dentro de uma estufa para capturar informações referentes a temperatura e umidade do ar. Um nó atuador foi desenvolvido para acionar um ventilador caso o valor capturado referente a temperatura do ar ultrapasse um limiar predefinido.

A Tabela 3 mostra uma lista comparativa, com os artigos citados neste capítulo e o hardware utilizado em suas propostas. São mostradas as placas micro controladoras, sensores, antenas e protocolos utilizados no desenvolvimento dos nós da rede em cada artigo.

Tabela 3. Tabela comparativa de hardware utilizado em artigos referenciados.

Artigo	Protocolo	Antena	Placa/Microcontrolador	Sensores		
				Temperatura do Ar	Umidade do ar	Luminosidade
Ahonen et al(2010)	6LoWPan	CC2420	Sensinode	x	x	x
Benavente (2010)	IEEE 802.15.4	CC2420	Micaz	x	x	
Silva e Fruett (2010)	IEEE 802.15.4	Xbee	LPC1114/LPC2378	x	x	
Akshay et al (2012)	Zigbee	Tarang F4	PIC18F452	x	x	
Mohanty e Patil (2013)	IEEE 802.15.4	CC2510	Micaz	x	x	x
Quan et al (2016)	Zigbee	CC2530	CC2530	x	x	

Capítulo 3 - Projeto de Uma RSSF Para Estufas Agrícolas de Qualquer Tamanho

As estufas agrícolas são utilizadas em plantações para fornecer proteção física de variações climáticas como o vento, chuva, granizo, neve, e amenizar variações bruscas de temperaturas e umidades. O excesso de calor e/ou CO₂ pode ser prejudicial para a plantação sendo os controles de temperatura e umidade do ar muito importantes. São importantes, também, controles de luminosidade e irrigação.

Além das medições das alterações climáticas, para o controle de uma estufa agrícola são necessários, também, dispositivos que alterem as condições ambientais como ventilador, ar condicionado, aquecedores, entre outros. Esses dispositivos serão chamados de atuadores, neste artigo, e podem ser acionados automaticamente para uma maior precisão no controle da estufa.

Outros aspectos das estufas são o tamanho e infraestrutura, por exemplo, em Ahonen et al. (2008), os testes foram realizados em uma seção de uma estufa de tamanho 18m x 80m. Em Akshay et al (2012), uma estufa de tamanho 10 x 15 metros foi utilizada para testes, mostrada na Fig. 4.

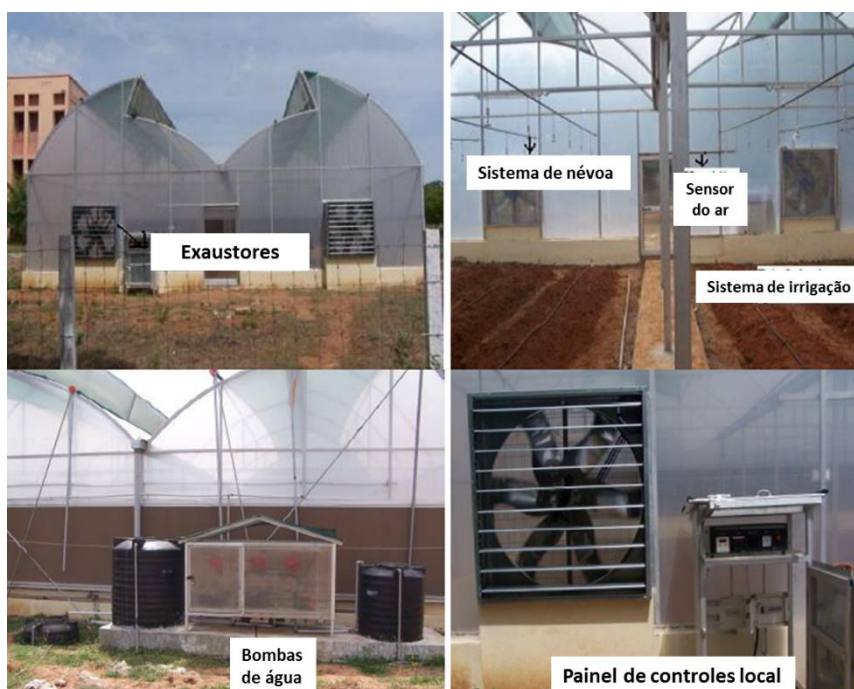


Figura 4. Estufa agrícola utilizada para testes. Adaptado de Akshay et al (2012).

Em visita realizada à empresa Ervas Finas Horticultura (2016), localizada em uma região montanhosa do município de Campo Limpo Paulista-SP, contatou-se a existência de 80 estufas, onde são plantadas ervas aromáticas, flores comestíveis, mini legumes e folhagens especiais. A estrutura de cada estufa é metálica, com dimensões de 6 metros de largura por 48 metros de comprimento, dividida em 16 seções de três metros de comprimento, totalizando 1280 seções. Nesta visita foi possível verificar que no local há um sistema de irrigação automatizado que é acionado em horários fixos diariamente. No local também há um sistema manual de sombrites.

Como pode se observar, as dimensões das estufas podem ter variações razoáveis. Assim, para o projeto proposto neste trabalho, uma RSSF com estrutura de larga escala poderá ser utilizada para monitorar e controlar um conjunto de estufas individuais, ou uma estufa de grandes dimensões dividida em compartimentos individualizados, como mostrado na Fig. 5.

Um nó sensor é definido, neste trabalho, como um dispositivo de rede formado por sensores, processador, memória, rádio e fonte de energia. Um nó sensor pode coletar diferentes tipos de dados provenientes dos sensores conectados, como temperatura do ar, luminosidade, etc. Na Fig. 5 os nós sensores são distribuídos dentro de cada estufa, e coletam os mais variados dados.

A configuração de uma RSSF para estufas de qualquer tamanho, neste trabalho, será organizada em uma forma hierárquica, onde os nós sensores de cada compartimento ou estufa ficam no nível mais baixo e enviam dados para um nó intermediário, denominado roteador (mostrado por símbolo R na Fig. 5), e daí para o nó coordenador que está no nível mais elevado, visto na figura, fora da estufa. Na figura são mostrados somente três níveis de hierarquia, mas, dependendo da necessidade, para distâncias mais longas, o nível intermediário poderá ter mais estágios.

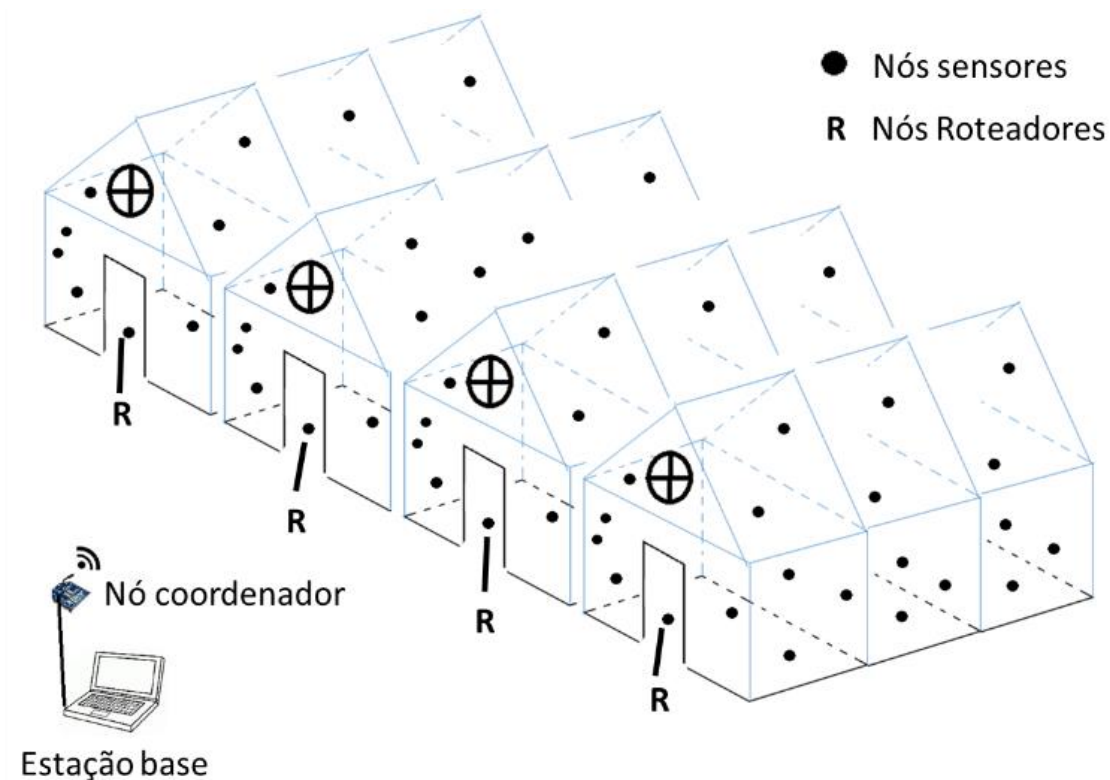


Figura 5. Exemplo de uma RSSF em estufa de larga escala.

O nó coordenador se comunica com a estação base que armazena todos os dados recebidos. Os dados obtidos, pela estação base, podem ser apresentados na forma de tabela ou gráficos. Uma RSSF desenvolvida de forma hierárquica como apresentado, terá como uma de suas limitações a quantidade de nós filhos que o nó coordenador ou os nós roteadores podem ter.

Como exemplo de aplicação de uma RSSF hierárquica é usado como *case* a empresa Ervas Finas Horticultura. A empresa possui 80 estufas de tamanho 6m x 48m, onde cada estufa é dividida em 16 seções de 6m x 3m totalizando 1280 seções. Propõe-se que em cada seção sejam utilizados seis nós de rede, na seguinte composição: um nó roteador, um nó sensor de umidade da terra, um nó sensor de luminosidade, um nó sensor de temperatura/umidade do ar, um nó atuador para irrigação do solo e um nó atuador para controle do sombrite. Adicionalmente um nó coordenador é utilizado para criar e gerenciar a RSSF, totalizando 7681 nós de rede, conforme detalhado na Tabela 4.

Tabela 4. Quantidade de nós da RSSF.

Nós	Tipo	Quantidade de nós
Sensores	luminosidade	1280
	temperatura/umidade do ar	1280
	Umidade da terra	1280
Atuadores	Irrigação	1280
	Sombrite	1280
Roteadores	-	1280
Coordenador	-	1
Total	-	7681

Se um nó sensor de luminosidade capturar dados com intervalo de um minuto, durante um ano, serão gerados:

$$\frac{1 \text{ dado}}{\text{min}} \times 60 \text{ min} \times 24 \text{ horas} \times 30 \text{ dias} \times 12 \text{ meses} = \mathbf{518.400 \text{ dados/ano.}}$$

O nó sensor do solo também irá gerar a mesma quantidade de dados que o nó de luminosidade, porém o nó sensor do ar irá gerar o dobro de dados referentes a temperatura e a umidade do ar totalizando em 1.036.800 dados. Uma base de dados com essas informações possibilita conhecer com profundidade a variação anual de luminosidade, umidade da terra, temperatura e umidade do ar em uma seção de uma estufa, contribuindo para decisões ótimas de controle de sombrite e intervalo de irrigação.

Ampliando-se a análise anterior, com a utilização de 1280 sensores de luminosidade serão gerados 663.552.000 dados referentes à variação de luminosidade observada em todas estufas em um ano. Considerando a utilização dos três tipos de sensores teremos 3840 sensores que irão gerar 2.654.208.000 de dados pertinentes à variação anual de luz, umidade da terra, temperatura e umidade do ar. A tabulação e análise dos dados pode auxiliar em decisões seguras, como o local ideal de plantio conforme as necessidades de luz da planta ou melhor época para plantio.

Assim, um sistema de monitoramento agrícola é proposto com o desenvolvimento e implementação prática de uma RSSF protótipo, assim como o desenvolvimento de um software para a interpretação dos dados obtidos.

Na RSSF hierárquica proposta, os nós sensores, atuador e coordenador fazem parte do processamento dos dados, assim o tamanho do pacote pode ser reduzido, diminuindo o fluxo de pacotes e permitindo maior quantidade de nós na rede. A responsabilidade de armazenamento dos dados foi centralizada no computador.

Capítulo 4 - Implementação Prática da Rede Hierárquica de Sensores Sem Fio.

Para a implementação prática do projeto proposto, uma configuração simplificada de uma RSSF foi utilizada, conforme mostrada na Fig. 6. Na implementação prática dos experimentos 1 e 2 não foram utilizados atuadores. Observa-se na Fig. 6 que o Nó 1 está conectado a um sensor de umidade da terra, o nó 2 está conectado a um sensor de umidade e temperatura do ar, o nó 3 está conectado a um sensor de luminosidade, o nó 4 é um atuador, o nó 5 é um nó roteador. O nó 6 é o Coordenador da rede, que receberá todas as informações e irá transmitir para o computador pela porta Serial/USB (*Universal Serial Bus*).

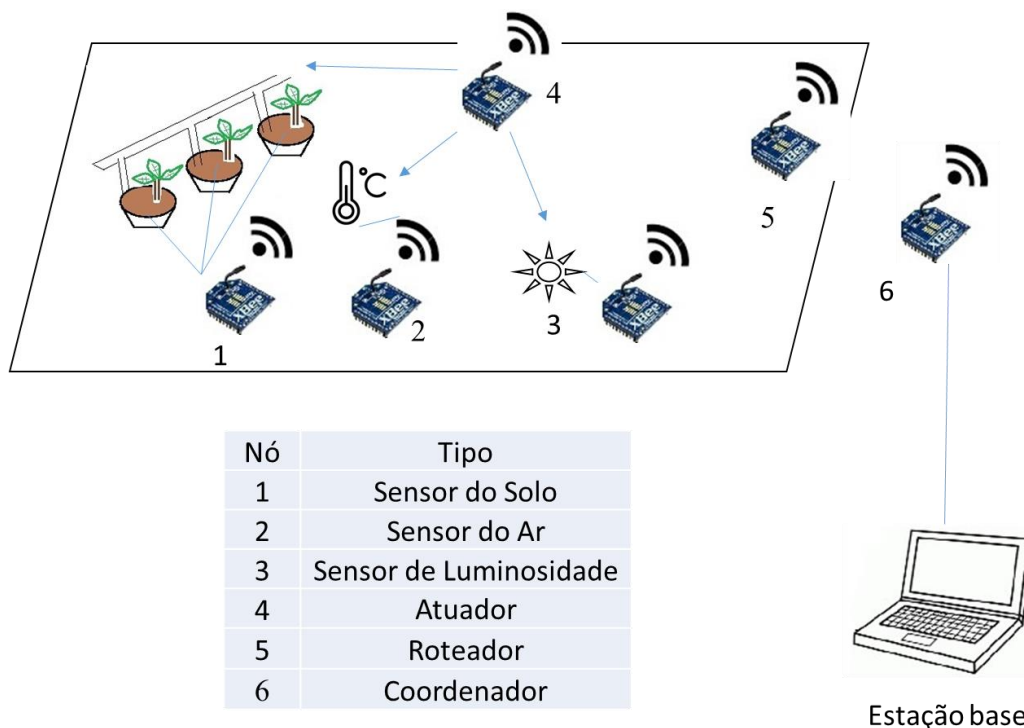


Figura 6. Rede simplificada para implementação prática.

Este projeto de estufa agrícola poderá resolver dois dos problemas observados na empresa Ervas Finas Horticultura que utiliza um sistema automatizado de irrigação do solo em intervalos de tempo pré-configurados, mas não verifica o nível de umidade do solo, o que pode acarretar em excesso de irrigação em alguns pontos, possivelmente prejudicando a raiz das plantas e acarretando desperdício de água.

Um outro problema constatado é o controle sobre a quantidade de luz que as plantas recebem em dias ensolarados. Atualmente, é utilizado um sistema manual, requerendo a ação de uma pessoa para abrir ou fechar manualmente uma cortina de lona, denominada de sombrite, de acordo com a necessidade de cada tipo de planta. Este projeto poderá contribuir para melhorar esse sistema, através da utilização de uma RSSF que coletará dados dos nós sensores de luminosidade, e auxiliará na decisão de abrir ou fechar um sombrite com um nível de precisão bastante superior ao processo atual, permitindo sua automação com a utilização de atuadores.

4.1 Projeto dos Nós

O projeto dos nós pode ser dividido em duas partes, uma parte relativa ao rádio, para a transmissão de dados sem fio, e a outra parte, um hardware que coleta dados de sensores e envia ao rádio, ou aciona dispositivos como uma bomba de água ou motor de passo.

Nesta dissertação o desenvolvimento dos nós protótipos utilizaram peças de baixo custo e fácil disponibilidade no mercado. Para a placa foi escolhido o Arduino Uno, uma placa de prototipação, que possui um microcontrolador ATmega328P. A antena escolhida foi a Xbee Zigbee (2016) que utiliza o protocolo Zigbee (2016). Todos os nós da rede utilizaram uma antena Xbee Zigbee e um Arduino uno. O nó atuador foi desenvolvido utilizando uma bomba de água, alimentada por pilhas, e um motor de passo.

4.1.1 Projeto da Parte do Rádio

As redes de sensores sem fio podem utilizar diferentes protocolos para a comunicação entre os nós da rede. Os artigos da Tabela 3 utilizaram os protocolos IEEE 802.15.4, 6LoWPan e Zigbee.

O padrão de redes IEEE 802.15.4 (2016), denominado WPAN (*Wireless Personal Area Network*), define as camadas Física, com velocidade de transmissão de 250kbps, e de MAC (*Media Access Control*) com limite de endereçamento local em 16bits ou de 0 a 65535 endereços, sendo o endereço 0 reservado para o nó coordenador.

O protocolo 6LoWPan, que significa IPv6 over WPAN, foi desenvolvido baseado no protocolo IEEE 802.15.4. Ele utiliza as camadas Física e de MAC do WPAN, adiciona

um endereço de 64 bits e permite usar pacotes IPv6 comprimidos nas redes IEEE 802.15.4, que habilita a conexão direta de um nó de rede com a internet.

O padrão Zigbee (2016), padrão de rede sem fio de baixa velocidade e baixa utilização de energia, também utiliza das camadas Física e MAC do IEEE 802.15.4. Adicionalmente possui funcionalidades de controle, gerenciamento e segurança da rede conforme a Tabela 5. A Zigbee Alliance (2016), uma aliança entre dezenas de grandes empresas internacionais, foi fundada em 2002, com o objetivo de padronizar e continuamente desenvolver o Zigbee.

Tabela 5. Quadro com definições de camadas Zigbee.

Aplicação	Customizável	Zigbee	Descrição
Aplicação framework	Zigbee	Física	Define operação física, sensibilidade de recebimento, rejeição de canal, potência de saída, número de canais, modulação do chip e especificação de velocidade de transmissão, com máximo 250 kbps operando na banda ISM 2.4 GHz. Ver IEEE 802.15.4 para mais detalhes.
Aplicação profile		MAC	Inclui serviços como retransmissão, confirmação de dados e técnica de detecção de colisões (CSMA-CA).
Rede	IEEE 802.15.4	Rede	Adiciona capacidade de roteamento, permitindo os dados atravessar múltiplos nós da origem ao destino.
MAC		Profile	Define objetos de endereçamento como profiles, clusters e endpoints.
Física		Framework	Camada que provê descobrimento de serviços e dispositivos, e capacidades avançadas de gerência da rede

Em uma rede Zigbee existem três tipos de nós:

(1) *Coordenador* - é responsável pela criação da rede, selecionando um PAN ID (*Personal Area Network ID*) de 16 bits que irá identificar a rede. Dessa forma, várias redes podem funcionar simultaneamente no mesmo local. O coordenador atribui um identificador para os nós filhos, limitado a 16 bits de endereçamento local, totalizando 65535 possíveis nós. O coordenador deve estar sempre ligado para permitir a junção de novos nós e auxiliar no roteamento de dados.

(2) *Roteador* - são nós que devem se juntar a uma rede antes de transmitir, receber ou rotear dados. Permite dispositivos finais se associarem. Auxilia no roteamento de pacotes e expande a área de cobertura da rede. Deve estar sempre ligado e seu uso é opcional.

(3) *Dispositivos finais (End Device)* - são nós que se conectam na rede para transmitir ou receber dados, sempre através de um nó pai. Não pode rotear pacotes ou permitir dispositivos se juntarem a rede. Pode entrar em modo *low power* para conservar energia e possibilitar alimentação por bateria.

Os dispositivos finais podem ser sensores, capturando informações, como temperatura, umidade do ar, ou atuadores, ligando um dispositivo como, uma bomba de água, interruptor de luz, ventilador. A Fig. 7 mostra um exemplo de uma rede Zigbee.

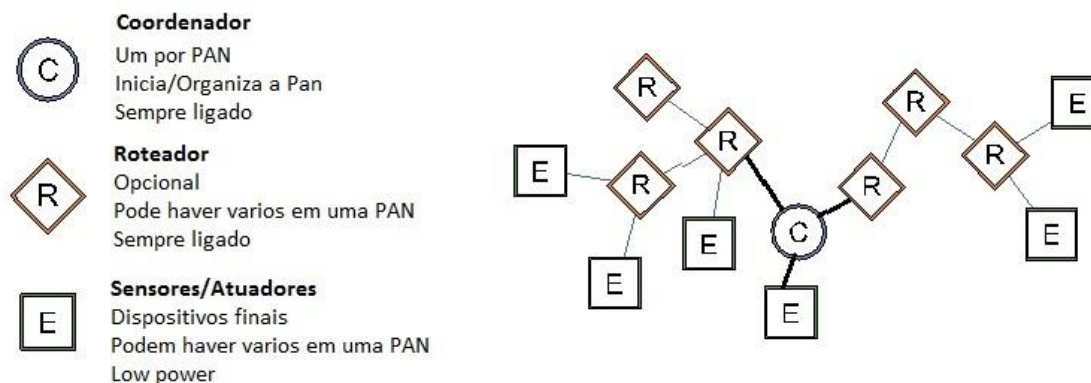


Figura 7. Exemplo de rede Zigbee.

As especificações da Zigbee se mostram adequadas para a sua utilização em estufa agrícola, principalmente pela sua baixa velocidade de transmissão e baixo consumo de energia, além de ser um componente fácil de encontrar no mercado.

Nos artigos referenciados na Tabela 3 foram utilizadas diferentes antenas no desenvolvimento de protótipos. As antenas CC2420, CC2510 e CC2530 são parte de placas do tipo *System-on-a-chip* (SoC). As placas SoC integram em um mesmo chip diferentes funcionalidades como um rádio. As antenas Tarang F4 e Xbee, já descontinuadas, são modelos anteriores à antena Xbee Zigbee (2016) usada nesta dissertação.

Para o rádio foi escolhido um componente que é compatível com a recomendação da Zigbee. O rádio Xbee Zigbee (2016) é fabricado pela empresa Digi (2016) seguindo os padrões Zigbee. Ele possui uma limitação de 14 nós filhos e alcance de sinal em 60 m para ambientes fechados. As antenas são configuradas utilizando o software XCTU (2016), fornecido no site da Digi. Através do XCTU, pode-se definir o modo de funcionamento das antenas (Coordenador, Roteador ou *End Device*), PANID (*Personal Area Network Id*), endereço de destino entre outros.

Na Fig. 8 é exibida a tela de configuração do XCTU para o nó coordenador. Após modificações de parâmetros é necessário efetuar a escrita com o botão *Write*. Foi escolhido arbitrariamente o valor 200 para a PANID, desta forma todos os nós que se juntarem à

rede irão exibir este valor para PANID. Somente um nó coordenador pode operar em cada PANID.

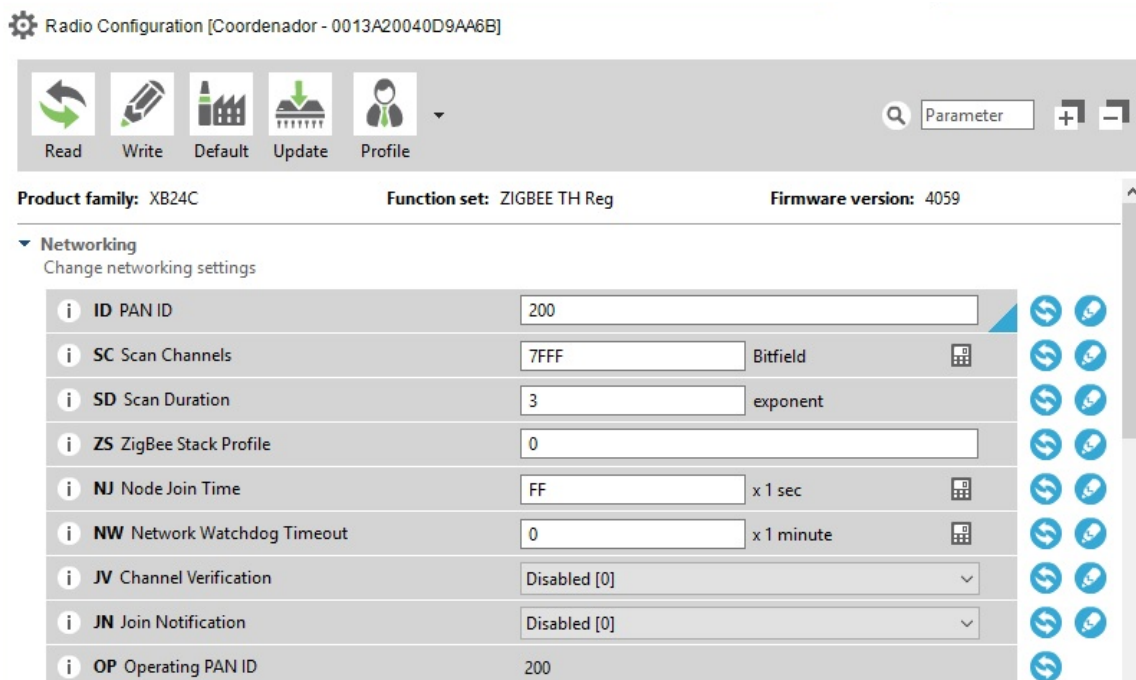


Figura 8. Tela de configuração do nó coordenador no software XCTU.

4.1.2 Projeto da Parte do Hardware

Nos artigos da Tabela 3 foram utilizadas placas microcontroladoras SoC dos modelos Sensinode, Micaz e CC2530.

A placa escolhida, para esta dissertação, foi o Arduíno Uno (2016) que possui microcontrolador ATmega328P com clock de 16 MHz, 32KB de memória flash, 14 portas digitais para input/output, 6 portas analógicas e porta USB.

Os nós sensores utilizaram diferentes tipos de sensores sendo eles: luminosidade, umidade da terra, temperatura e umidade do ar. A captura de dados dos sensores é feita através da leitura da porta analógica em que o sensor está conectado no Arduíno. Para os nós sensores foram utilizados um sensor de temperatura/umidade do ar DHT11, um sensor de luminosidade LDR e um higrômetro sensor de umidade da terra. Também foi desenvolvido um nó atuador com uma bomba de água, com motor elétrico RS-360SH, e um motor de passo 28BYJ-48.

4.2 Projeto do Nó Coordenador e do Nó Roteador

Os nós do tipo coordenador e roteador possuem somente a parte de rádio conectado ao Arduino Uno. Quatro conexões são necessárias entre a antena e o Arduino Uno. São necessários 3.3 volts (V) para o funcionamento da antena Xbee Zigbee e a ligação do *ground* (GND) ou negativo. A porta de transmissão (TX) do Xbee é conectada à porta de recebimento (RX) do Arduino e a porta TX do Arduino é conectada à porta RX da antena. As conexões físicas do roteador e do coordenador são mostradas na Fig. 9.

As portas RX e TX da antena e do Arduino são portas seriais. Para as antenas as taxas da porta serial utilizadas são de 9600 bits/seg ou 115200 bits/seg configuradas na antena. A porta serial do Arduino deve ser programada para funcionar na mesma velocidade da porta serial da antena, com o comando `Serial.begin(9600);` é iniciada a porta serial do Arduino com taxa de 9600 bits/seg.

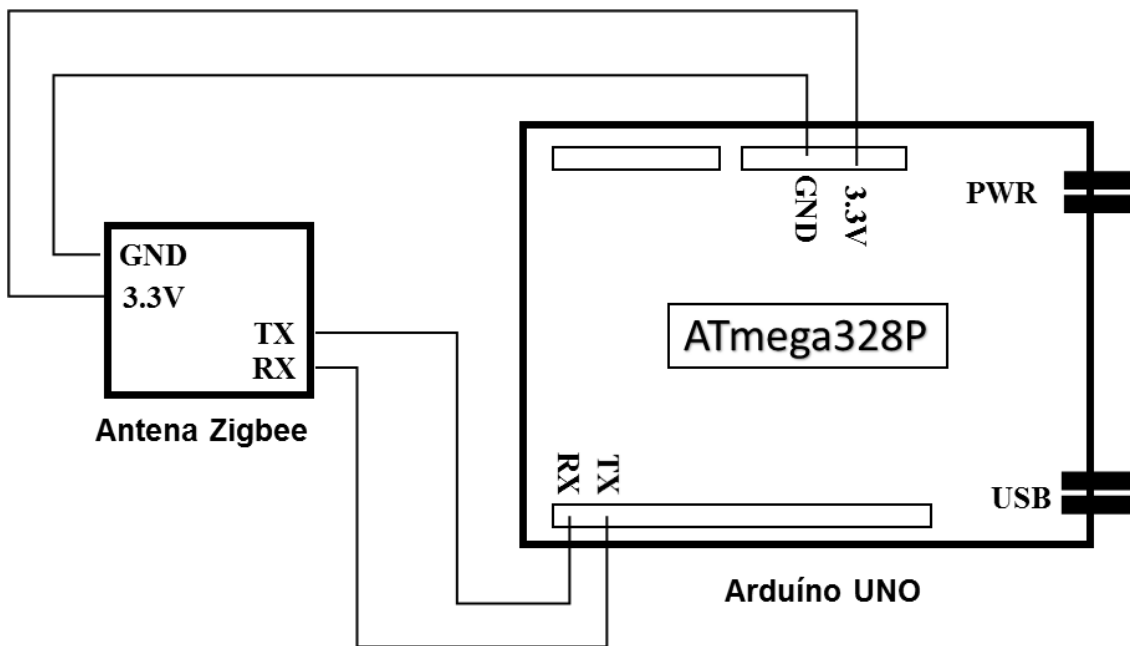


Figura 9. Conexão física entre o Arduino Uno e antena Xbee Zigbee.

Para conectar a antena no Arduino foi utilizada uma placa adaptadora denominada *Shield*, que faz as mesmas conexões físicas da Fig. 9. Esta placa foi utilizada em todos os nós da rede.

O nó coordenador foi utilizado de duas formas nos experimentos:

1. Na primeira forma, utilizada nos experimentos 1 e 2, o microprocessador foi removido do Arduino, para que a porta serial da antena faça conexão direta com a

porta serial/COM USB do computador. Neste caso o Arduíno serviu como um adaptador da antena, dessa forma todos os pacotes recebidos pela antena do nó coordenador poderiam ser interpretados diretamente pelo XCTU, que faz a leitura da porta serial do computador, possibilitando a geração de um arquivo log dados com a informação de pacotes recebidos.

2. Na segunda forma foi utilizado o microprocessador no Arduíno para todos os nós. Todas as mensagens geradas pelo microprocessador são encaminhadas para sua porta TX conectadas na porta RX da antena, então são transmitidas. No caso de uma mensagem chegar na antena do nó atuador ou coordenador, são enviadas por sua porta TX e recebidas na porta RX do microprocessador onde serão processadas.

No caso do nó coordenador os pacotes recebidos serão analisados pelo microprocessador, que decidirá quais dados serão encaminhados para a porta serial/COM ou se uma mensagem será enviada para o nó atuador. Para o nó atuador as mensagens recebidas serão processadas pelo microcontrolador, que poderá acionar um ou mais atuadores.

As antenas funcionam em dois modos diferentes intercambiáveis chamados de AT e API:

- **Modo AT (Transparente)** - é utilizado para conectar apenas dois nós. Este modo é utilizado para os nós sensores que apenas enviam informações para o nó coordenador. Os dados recebidos pela porta serial RX da antena serão automaticamente encapsulados e enviados para o endereço de destino, definido previamente na configuração da antena. Caso o endereço de destino necessite ser modificado, será necessário reconfigurar a antena utilizando o programa XCTU. Utilizando um Arduino Uno, é executado o comando “Serial.print(x);” para enviar o valor de “x” à porta RX da antena. O valor de x será automaticamente encapsulado e enviado para o endereço de destino.
- **Modo API (*Application programming interface*)** – para enviar dados é necessária a montagem manual do pacote. Este modo é utilizado para que um nó possa enviar dados para múltiplos destinos modificando o endereço de destino, entre outras

configurações, nos bytes correspondentes. Com um Arduino Uno foi utilizado o comando “Serial.write(0x7E);” para enviar um byte com valor hexadecimal 7E para a porta RX da antena.

Para esta dissertação, os testes e experimentos de uma RSSF, não foram utilizados nós roteadores devido a curta distância entre os nós da rede.

4.3 Montagem de Pacotes

Um nó sensor, com a antena Xbee em modo AT, envia para o nó coordenador um pacote com um ou mais valores capturado pelos sensores. Primeiramente, o sensor captura um valor e armazena em “x”, então, no microcontrolador do Arduino, é executado o comando “Serial.print(x);” que envia o valor de “x” para a porta de RX da antena. O valor de “x” é enviado, na porta TX do Arduino, como um sinal gráfico imprimível, seguindo os padrões da tabela ASCII, (*American Standard Code for Information Interchange*). A antena recebe o valor de sinal em sua porta RX, encapsula automaticamente o dado e envia para o coordenador. Os pacotes enviados, por qualquer antena, são denominados do tipo “Requisição de Transmissão” com tamanho variado conforme a quantidade de dados a serem enviados.

Por exemplo, um nó sensor de luminosidade captura o valor “1” e armazena na variável “x”. O comando “Serial.print(x);” irá enviar o valor de “x” para a porta TX do microprocessador do Arduino, ou seja, um sinal de valor “1” será enviado para a antena, que fará a leitura do valor de “x” em sua porta RX. A antena irá transformar, neste exemplo, o valor de sinal “1” em hexadecimal com valor 31, que será encapsulado em um pacote e enviado para o nó destino. Neste exemplo, representado na Fig. 10, o pacote tem tamanho de 19 bytes, onde cada byte é representado por um valor hexadecimal. O pacote inicia com o byte delimitador 7E que indica ser um pacote do protocolo WPAN.

Se um valor com dois dígitos for enviado, por exemplo “11”, a antena reconhecerá como dois sinais de valor 1. Assim transformando em dois valores hexadecimais 31 31, ou seja, cada caractere será transformado em um valor hexadecimal separadamente e ocupará um byte de tamanho.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
7E	00	0F	10	00	00	13	A2	00	40	D9	AA	6B	FF	FE	00	00	31	DE

Figura 10. Exemplo de um pacote de Requisição de Transmissão.

A Tabela 6 detalha a formação do pacote Zigbee da Fig. 10. Os campos Pacote ID, Endereços de destinos 64 e 16 bit, raio de broadcast e outras opções são definidas na configuração da antena previamente.

Tabela 6. Detalhes de um pacote tipo requisição de transmissão.

Byte nº	Descrição	Valor Hexadecimal	Significado
1	Delimitador inicial	7E	
2 e 3	Tamanho	00 0F	15 bytes de tamanho
4	Tipo de Frame	10	Requisição de transmissão
5	Frame ID	00	Sem confirmação de recebimento
6 a 13	Endereço destino 64-bit	00 13 A2 00 40 D9 AA 6B	Nó coordenador
14 e 15	Endereço destino 16-bit	FF FE	Envia para todos roteadores
16	Raio de Broadcast	00	limite de saltos para broadcast
17	Opções	00	
18	Dados	31	Sinal com valor 1 da tabela ASCII
19	Checksum	DE	

Para os nós com antenas Xbee no modo API o encapsulamento dos dados enviados para a porta RX da antena não é automático. A criação do pacote será manual, sendo necessário indicar cada byte do pacote, permitindo a modificação do pacote diretamente em sua criação. Para enviar este pacote para outro destinatário os campos endereço de destino 64 bits e checksum deverão ser modificados.

O comando “Serial.write(0x7E);” no microcontrolador do Arduino envia o valor hexadecimal 7E transformado em binário para a porta RX da antena. Assim, o pacote completo é representado pelo código da Fig. 11.

Os pacotes do tipo requisição de transmissão recebidos pelo coordenador são processados, transformados para pacotes do tipo recebido e enviado para a porta serial TX da antena. Os bytes referentes ao endereço de destino são substituídos por bytes referentes ao nó de origem do pacote.

Byte	Comando
1	Serial.write(0x7E);
2	Serial.write(0x00);
3	Serial.write(0x0F);
4	Serial.write(0x10);
5	Serial.write(0x00);
6	Serial.write(0x00);
7	Serial.write(0x13);
8	Serial.write(0xA2);
9	Serial.write(0x00);
10	Serial.write(0x40);
11	Serial.write(0xD9);
12	Serial.write(0xAA);
13	Serial.write(0x6B);
14	Serial.write(0xFF);
15	Serial.write(0xFE);
16	Serial.write(0x00);
17	Serial.write(0x00);
18	Serial.write(0x31);
19	Serial.write(0xDE);

Figura 11. Comandos para a criação de um pacote Zigbee.

O microcontrolador do Arduíno, do nó coordenador, irá fazer o processamento dos pacotes do tipo recebido na porta RX do microcontrolador. Em cada pacote recebido serão identificados os nós de origem e os dados contidos no mesmo.

A Fig. 12 mostra um pacote de transmissão sendo enviado para o nó coordenador a partir de um nó sensor. O coordenador irá processar este pacote e encaminhar para a porta serial o pacote recebido. A porta serial pode estar conectada a um Arduíno ou computador, que podem processar e extrair os dados dos pacotes recebidos.

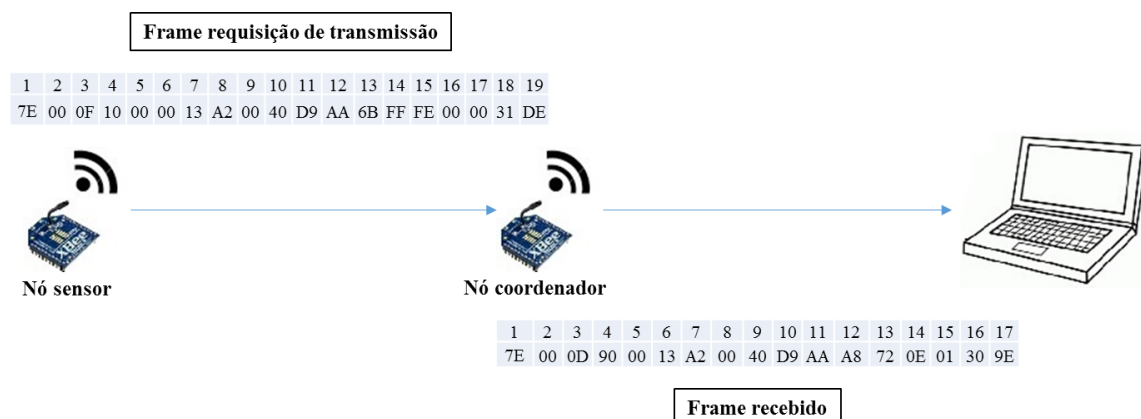


Figura 12. Exemplo de pacote de transmissão para recebido.

O pacote do tipo recebido possui dois bytes a menos que o pacote transmitido. Os campos de Frame ID e Raio de Broadcast somente são utilizados por pacotes de

transmissão. A Tabela 7 mostra o pacote recebido com detalhes, indicando o endereço de origem de 64 e 16 bits, assim como os dados recebidos.

Tabela 7. Detalhes de um pacote tipo recebido.

Byte nº	Descrição	Valor Hexadecimal	Significado
1	Delimitador inicial	7E	
2 e 3	Tamanho	00 0D	13 bytes de tamanho
4	Tipo de Frame	90	Frame recebido
5 a 12	Endereço origem 64-bit	00 13 A2 00 40 D9 AA A8	Endereço nó sensor
13 e 14	Endereço origem 16-bit	72 0E	Endereço local nó sensor
15	Opções	00	
16	Dados	31	Sinal com valor 1 da tabela ASCII
17	Checksum	DE	

O software XCTU possui uma ferramenta para auxiliar na geração de diversos tipos de pacotes API. A Fig. 13 mostra a tela de criação do pacote de requisição de transmissão do exemplo anterior. Os campos Pacote ID, endereços de 16 e 64 bit, raio de broadcast, opção de confirmação de recebimento e os dados, são modificáveis conforme necessidade.

XBee API Frame generator
This tool will help you to generate any kind of API frame and copy its value. Just fill in the required fields.



Protocol: Mode:

Frame type:

Frame parameters:

- Start delimiter: 7E
- Length: 00 0F
- Frame type: 10
- Frame ID:
- 64-bit dest. address:
- 16-bit dest. address:
- Broadcast radius:
- Options:
- RF data:

Generated frame:

7E 00 0F 10 00 00 13 A2 00 40 D9 AA 6B FF FE 00 00 31 DE

Byte count: 19

Figura 13. Tela da ferramenta de geração de pacotes API do XCTU.

4.4 Nós Sensores

Três tipos diferentes de nós sensores foram desenvolvidos: de temperatura e umidade do ar, de luminosidade e de umidade da terra. Para todos os nós sensores a porta de transmissão (TX) do Xbee é conectada à porta de recebimento (RX) do Arduino, e a porta TX do Arduino é conectada à porta RX do Xbee.

Para medir a temperatura e a umidade do ar foi utilizado um sensor do tipo DHT11 (2016), conectado a Arduino Uno em uma porta analógica, e uma antena Xbee Zigbee. O sensor DHT11 possui uma biblioteca para Arduino, que permite que os valores capturados indiquem a umidade do ar em porcentagem (%), e a temperatura do ar em Celsius (C°). A precisão da temperatura é de ± 2 graus, e a precisão da umidade do ar é de $\pm 5\%$. A conexão física do nó sensor de temperatura e umidade do ar é mostrada na Fig. 14.

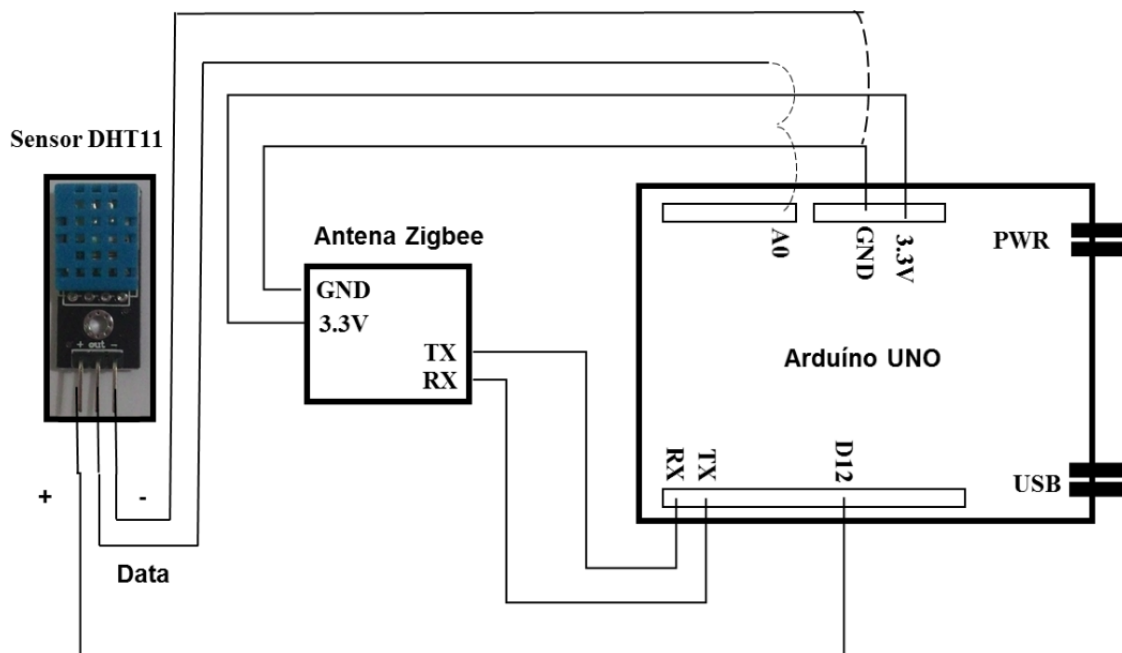


Figura 14. Conexão física do nó sensor de temperatura e umidade do ar.

O nó sensor de luminosidade possui um sensor de luminosidade LDR (*Light Dependent Resistor*) conectado ao Arduino em uma porta analógica e uma antena Xbee Zigbee. Os valores capturados pelo sensor na porta analógica são entre 0 e 1023. Tais valores são convertidos em uma escala de 0 a 9, onde 0 indica menor incidência de luz e 9 maior incidência de luz. A conexão física do nó sensor de luminosidade é mostrada na Fig. 15.

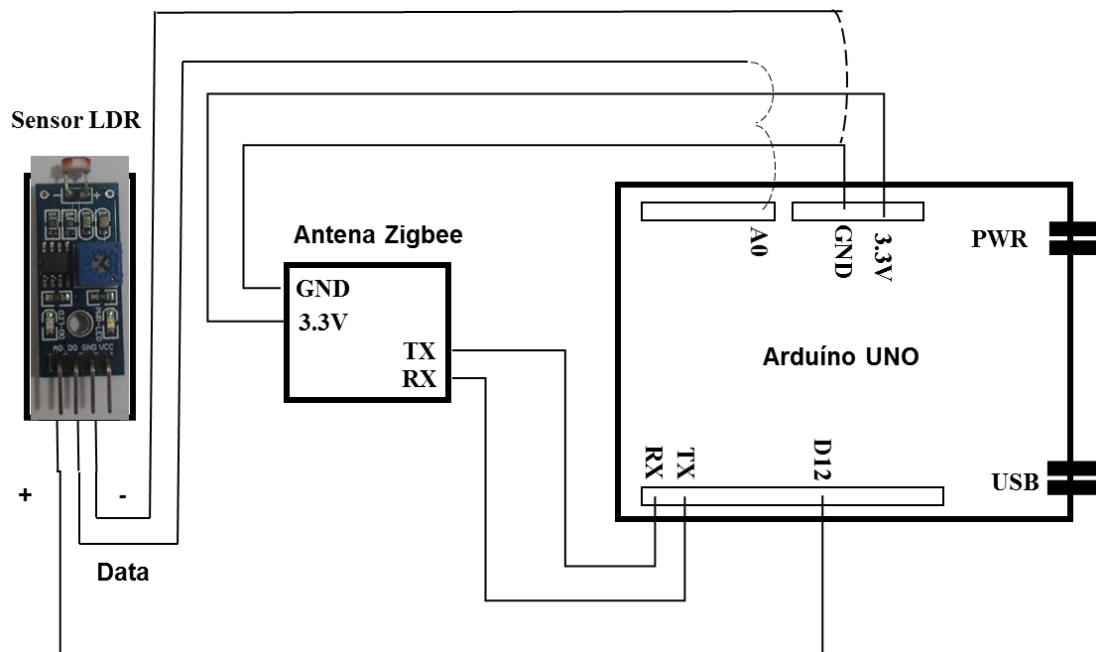


Figura 15. Conexão física do nó sensor de luminosidade.

O nó sensor de umidade da terra, também denominado nó sensor do solo, é composto por um sensor higrômetro que detecta as variações de umidade no solo. Ele é conectado a uma porta analógica do Arduino Uno, que captura valores entre 0 e 1023. Tais valores são convertidos em uma escala de 0 a 9 onde um menor número indica menor umidade. A conexão física do nó sensor de umidade da terra é mostrada na Fig. 16.

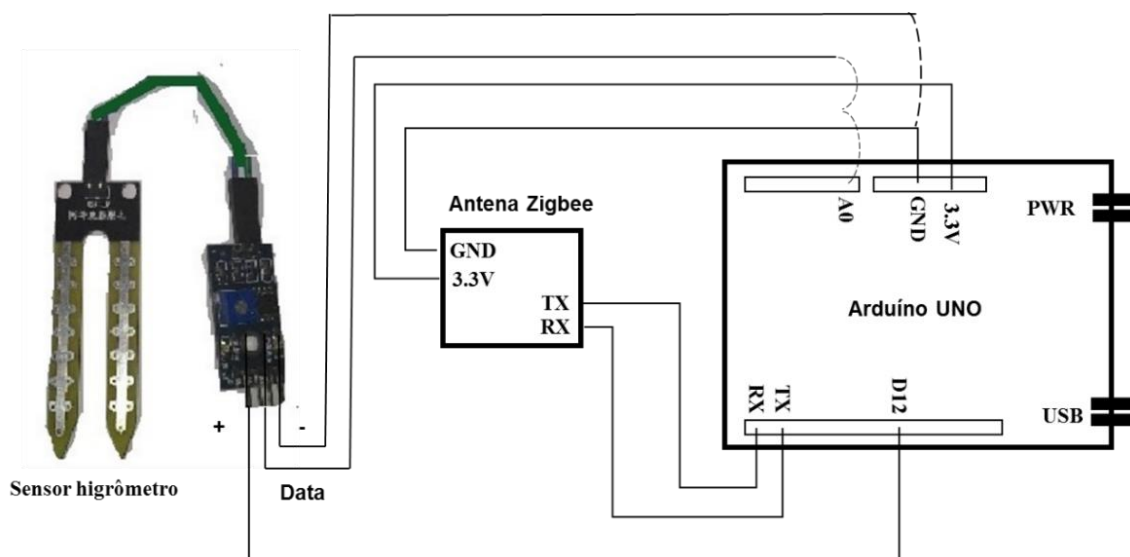


Figura 16. Conexão física do nó sensor de umidade da terra.

4.5 Nó Atuador

Um nó atuador foi desenvolvido com uma bomba de água, um motor de passo e seus respectivos drivers controladores, que são acionados quando valores específicos são capturados pelos nós sensores e encaminhados para o nó atuador.

O microprocessador do nó atuador analisa os pacotes recebidos, identifica o comando embutido no pacote e executa a ligação da bomba de água ou acionamento do motor de passo. Em uma porta digital PWM (*Pulse Width Modulation*) do Arduíno foi conectado um driver de motor DC (*Direct Current* ou corrente contínua). As portas digitais PWM permitem o controle de tensão de saída entre 0 e 5V com o comando `analogWrite(porta, valor);`. Os valores entre 0 e 255 podem ser atribuídos na porta PWM.

O driver de motor DC é responsável pela ativação da passagem de energia entre as pilhas e o motor elétrico da bomba de água. Quando o valor da porta digital PWM for alto (255) o driver permitirá a passagem da energia das pilhas para o motor. Caso seja um valor baixo (0) o driver bloqueia a passagem elétrica entre as pilhas e o motor. No Arduíno é executado o comando “`analogWrite(6, 255);`” para acionar o driver conectado na porta digital 6. Neste driver são conectadas uma fonte de energia (pilhas) e a bomba de água. Para este nó atuador a alimentação elétrica da bomba de água foi realizada utilizando 4 pilhas de 1,5V cada ligadas em série totalizando 6V.

A bomba de água utilizada é um motor elétrico RS-360SH (2016) com um adaptador mostrado na Fig. 17. Este motor funciona entre 3V e 9V com 1,76 ampere-hora em sua máxima eficiência.



Figura 17. Bomba de água utilizada no nó atuador.

Como exemplo um valor capturado pelo nó sensor de umidade da terra será encaminhado para o nó coordenador, que irá processar e comparar este valor com um limiar pré-definido. Caso o valor da umidade da terra esteja abaixo do limiar, uma mensagem será enviada para o atuador acionar a bomba de água por um tempo determinado.

O nó atuador irá executar a função “ligabomba()” se receber uma mensagem do coordenador com valor letra “l” referente a ligar a bomba de água. O exemplo da Fig. 18 mostra os comandos que serão executados pela função, onde o driver de motor DC está conectado a uma porta digital PWM, que será acionada por 5 segundos, ligando a bomba por este mesmo período, e depois desligada.

```
void ligabomba() { //liga a bomba de agua por 5 segundos
  analogWrite(6, 255); // Ativa a porta 6 PWM com 5V
  delay(5000); //espera 5 segundos
  analogWrite(6, 0); // Desativa porta 6 PWM com 0V
}
```

Figura 18. Comandos da função ligabomba().

O outro dispositivo utilizado pelo nó atuador é um motor de passo 28BYJ-48 (2016), com controlador driver ULN2003 (2016). O motor de passo é acionado através do driver controlador. O driver é conectado ao Arduíno em 4 portas digitais que são ligadas e desligadas alternadamente de forma que fazem o motor girar no sentido horário ou anti-horário. A Fig. 19 mostra as conexões físicas do nó atuador.

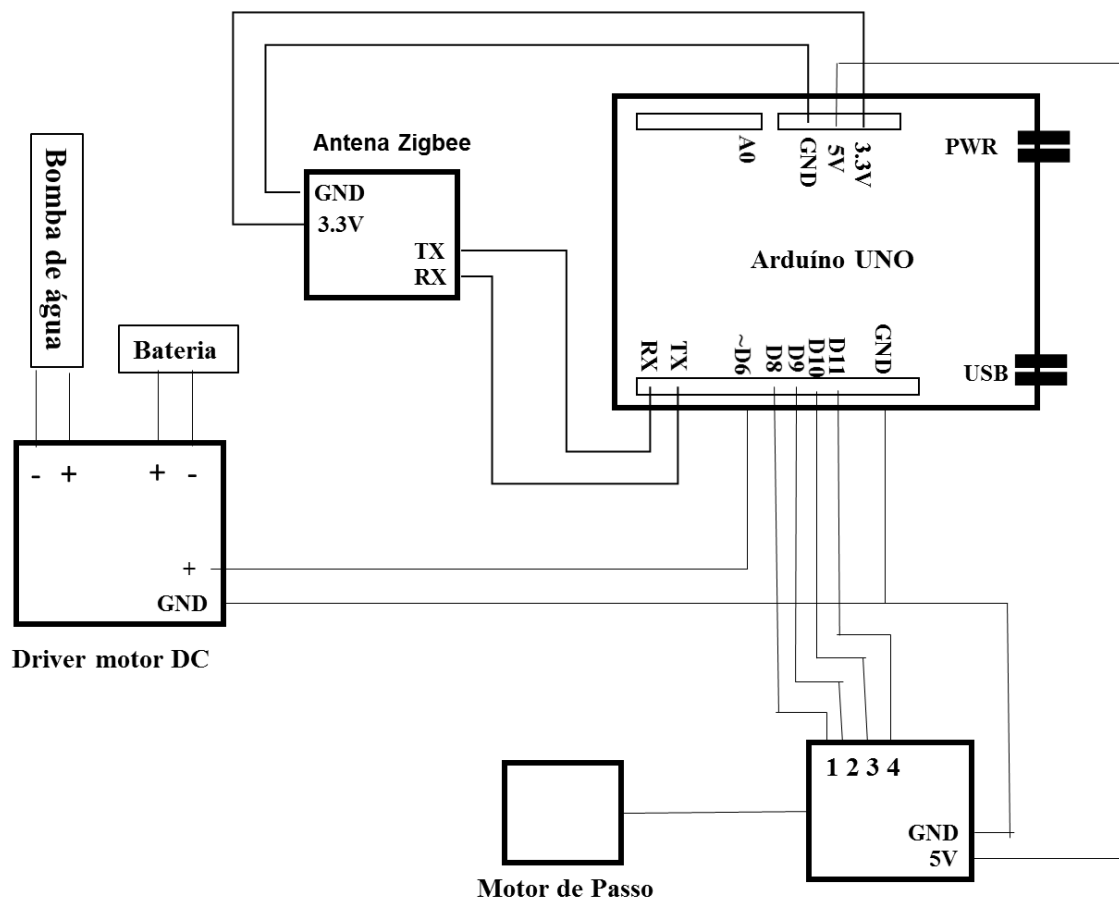


Figura 19. Conexão física do nó atuador.

4.6 Log de Pacotes XCTU

A conexão entre o Arduino e um computador é feita por meio de uma porta COM do computador. A comunicação entre os dispositivos é feita pela porta USB do computador que simula uma porta Serial/COM. No software XCTU há uma ferramenta para visualização e armazenamento dos pacotes recebidos da porta COM, com limite de dois mil pacotes, denominada ferramenta de log. Cada pacote recebido, pelo nó coordenador, é adicionado ao log com o acréscimo de um identificador e do horário de chegada do pacote. A Fig. 20 mostra a tela do log na ferramenta XCTU de pacotes recebidos pelo nó coordenador.

Record date: 06-03-2016 03:29:10.583

Module ID: Coordenador

Module address: 0013A20040D9AA6B

Firmware function: ZIGBEE TH Reg

Firmware version: 4059

Port configuration: COM5 - 9600/8/N/1/N

Frames log

ID	Time	L...	Frame
0	17:49:52.994	13	Receive Packet
1	17:49:53.299	16	Receive Packet
2	17:49:53.600	15	Receive Packet
3	17:49:54.004	13	Receive Packet
4	17:49:54.357	16	Receive Packet
5	17:49:54.610	15	Receive Packet
6	17:49:55.011	13	Receive Packet
7	17:49:55.365	16	Receive Packet
8	17:49:55.617	15	Receive Packet
9	17:49:56.019	13	Receive Packet
10	17:49:56.371	16	Receive Packet
11	17:49:56.573	15	Receive Packet
12	17:49:56.977	13	Receive Packet
13	17:49:57.432	16	Receive Packet

Frame details

Receive Packet (API 1)

7E 00 0D 90 00 13 A2 00 40 D9 AA A8 72 0E 01 37 97

Start delimiter

7E

Length

00 0D (13)

Frame type

90 (Receive Packet)

64-bit source address

00 13 A2 00 40 D9 AA A8

16-bit source address

72 0E

Receive options

Figura 20. Tela do log de pacotes gerado no software XCTU.

Esta ferramenta de log do XCTU permite salvar o log de pacotes em um arquivo de texto com extensão “.log”. A Fig. 21 mostra um exemplo de log gerado e salvo em arquivo. Este arquivo de log contém o histórico dos pacotes recebidos pelo coordenador, e precisa ser processado para que os dados sejam extraídos. Um software denominado **Leitor de Log** foi desenvolvido com esta finalidade.

experimento 1.log - Bloco de notas

Arquivo
Editar
Formatar
Exibir
Ajuda

```

06-21-2016 17:49:52.994,0,RECV,7E000D900013A20040D9AAA8720E013797
06-21-2016 17:49:53.299,1,RECV,7E0010900013A20040D9AC04DCCE01343232317D
06-21-2016 17:49:53.600,2,RECV,7E000F900013A20040D9AC0333C301300D0AB4
06-21-2016 17:49:54.004,3,RECV,7E000D900013A20040D9AAA8720E013797
06-21-2016 17:49:54.357,4,RECV,7E0010900013A20040D9AC04DCCE01343232317D
06-21-2016 17:49:54.610,5,RECV,7E000F900013A20040D9AC0333C301300D0AB4
06-21-2016 17:49:55.011,6,RECV,7E000D900013A20040D9AAA8720E013797
06-21-2016 17:49:55.365,7,RECV,7E0010900013A20040D9AC04DCCE01343232317D
06-21-2016 17:49:55.617,8,RECV,7E000F900013A20040D9AC0333C301300D0AB4
06-21-2016 17:49:56.019,9,RECV,7E000D900013A20040D9AAA8720E013797
06-21-2016 17:49:56.371,10,RECV,7E0010900013A20040D9AC04DCCE01343232317D
06-21-2016 17:49:56.573,11,RECV,7E000F900013A20040D9AC0333C301300D0AB4
06-21-2016 17:49:56.977,12,RECV,7E000D900013A20040D9AAA8720E013797

```

Figura 21. Exemplo de arquivo de log de pacotes.

4.7 Ferramenta Leitor de Log

O arquivo de log gerado pelo XCTU na Fig. 21 exibe as informações na seguinte ordem: mês, dia, ano, hora, minuto, segundo, milissegundo, identificador de ordem de chegada, tipo de pacote e o pacote em hexadecimal. Os pacotes de cada tipo de nó sensor possuem tamanho fixo, sendo os nós sensores do ar, umidade da terra e luminosidade com 20, 19 e 17 bytes respectivamente

Um programa, denominado leitor de log, foi desenvolvido para ler, organizar e processar os dados coletados pelos nós sensores do arquivo de log gerado pelo XCTU. O leitor de log organiza as informações extraídas de um arquivo de log em colunas, sendo data, hora, nó de origem, estado do sensor, umidade do ar e temperatura do ar. Após a abertura de um arquivo de log o programa executa sete passos repetidos à cada linha do arquivo:

- 1 – Lê os 10 primeiros caracteres e copia para a coluna Data.
- 2 – Lê os caracteres 12 ao 19, copia para a coluna hora
- 3 – Ignora os próximos 14 caracteres. Os caracteres seguintes são referentes ao pacote Zigbee.
- 4 – Ignora 20 caracteres. Os quatro caracteres seguintes são referentes ao nó de origem identificados na Tabela 8.
- 5 – Lê 4 caracteres, compara com a Tabela 8 e insere o nome do nó na coluna “Nó”.
- 6 – Ignora seis caracteres. Os próximos caracteres são referentes aos dados.
- 7 – Lê caracteres referentes aos dados de acordo com o tipo de nó de origem e insere na coluna correspondente.

A Fig. 22 mostra um exemplo de uma linha do arquivo leitor de log com as indicações de cada passo realizado pelo software.

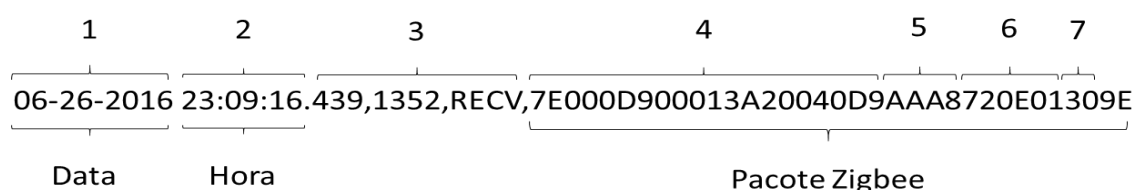


Figura 22. Exemplo de uma linha do arquivo leitor de log.

Os nós sensores, neste trabalho, geram e enviam pacotes de tamanho fixo com uma quantidade fixa de bytes de dados. O pacote do nó sensor de luz possui um byte de informação. O pacote do nó sensor do ar possui quatro bytes de informação. O pacote do nó sensor da terra possui três bytes de dados, porém apenas o primeiro byte, com valor 30, representa o valor do sensor. Os outros dois bytes 0D e 0A são referentes aos comandos pular linha e nova linha

A Tabela 8 mostra três pacotes recebidos dos três tipos de nós sensores. A coluna nó identifica qual o nome do nó de origem. A coluna Pacote exhibe o pacote inteiro recebido. A coluna tamanho identifica o tamanho do pacote em bytes. A coluna endereço mostra os últimos dois bytes do endereço de 64 bits do nó de origem. A coluna Dados identifica a informação embutida no pacote recebido, esses dados são exibidos em valores hexadecimais. Por fim, a coluna Valor mostra o valor dos dados em decimal.

Tabela 8. Exemplo de um pacote de cada tipo de nó sensor.

Nó	Pacote	Tamanho	Endereço	Dados	Valor
Luz	7E000D900013A20040D9AAA8720E013797	17	AAA8	37	7
Ar	7E0010900013A20040D9AC04DCCE01343232317D	20	AC04	34 32 32 31	42 21
Terra	7E000F900013A20040D9AC0333C301300D0AB4	19	AC03	30 0D 0A	0

A Fig. 23 mostra a tela do software leitor de log com colunas de data, hora, nó de origem, estado, umidade do ar em porcentagem (%) e temperatura do ar em graus centígrados (°C). Os valores da coluna “Estado”, referentes aos nós sensores de luz e solo, foram convertidos da escala de 0 a 9 para Escuro e Muita luminosidade, e Seco e Molhado. As informações exibidas são referentes ao arquivo do experimento 1 detalhado na subseção 5.1.

Abrir GraficoTerra Grafico Ar GraficoLuz						
	Data	Hora	Nó	Estado	Umidade %	Temperatura °C
▶	06-21-2016	17:49:52	Luz	7 Muita luminosidade		
	06-21-2016	17:49:53	Ar		42	21
	06-21-2016	17:49:53	Terra	Molhada		
	06-21-2016	17:49:54	Luz	7 Muita luminosidade		
	06-21-2016	17:49:54	Ar		42	21
	06-21-2016	17:49:54	Terra	Molhada		
	06-21-2016	17:49:55	Luz	7 Muita luminosidade		
	06-21-2016	17:49:55	Ar		42	21
	06-21-2016	17:49:55	Terra	Molhada		
	06-21-2016	17:49:56	Luz	7 Muita luminosidade		
	06-21-2016	17:49:56	Ar		42	21

Figura 23. Tela do software visualizador de log.

No programa Leitor de Log também foi desenvolvida uma ferramenta de geração de gráficos, representado por três botões GraficoTerra, Grafico Ar e GraficoLuz. Ao executar os botões GraficoTerra e GraficoLuz são gerados gráficos com a variação do tempo da coluna “Hora” e os valores da coluna “Estado” convertidos na escala de 0 a 9. O botão GraficoAr gera um gráfico com a variação do tempo da coluna “Hora” e os valores das colunas Umidade e Temperatura.

Capítulo 5 – Resultados Experimentais

Três experimentos foram executados para averiguar a confiabilidade e funcionamento de uma RSSF com nós de rede compostos dos equipamentos anteriormente citados. O primeiro experimento foi executado para testar o funcionamento de uma RSSF com a captura de dados utilizando nós sensores de luminosidade, umidade da terra e umidade/temperatura do ar. O segundo testou a confiabilidade e estabilidade de uma RSSF em um período de 36 horas de funcionamento, capturando dados com nós sensores de luminosidade, temperatura/umidade do ar e umidade da terra. O terceiro com duração de 72 horas, visou testar a RSSF com utilização de um nó atuador para verificar o acionamento automático da bomba de água conforme o nível de umidade da terra se torna seco.

Todos os dados capturados foram analisados pelo nó coordenador, que é responsável por encaminhar mensagens de ativação e desativação de dispositivos conectados ao nó atuador, e enviar os dados para o computador para manter o histórico de entrada de dados em um arquivo de log.

5.1 Experimento 1 – Teste dos nós sensores

Em uma sala de laboratório da FACCAMP (Faculdade Campo Limpo Paulista) uma RSSF foi configurada, com três nós sensores e um nó coordenador, para testes iniciais dos nós sensores de luminosidade, temperatura, umidade do ar e umidade da terra. Cada nó sensor foi configurado para capturar dados com intervalo de um segundo por um período de 5 minutos. Os dados capturados pelos nós sensores foram encaminhados para o nó coordenador, que estava conectado a um computador por meio da porta USB, onde foi gerado um log com o programa XCTU.

O arquivo de log criado foi aberto no programa leitor de log e os gráficos de variação de luminosidade, variação de temperatura e umidade do ar e variação de umidade da terra foram gerados.

Não houve necessidade de utilização de um nó roteador para expandir o sinal da rede pois os nós deste experimento encontravam-se próximos fisicamente. Na Fig. 24 é mostrada uma foto da RSSF desenvolvida.

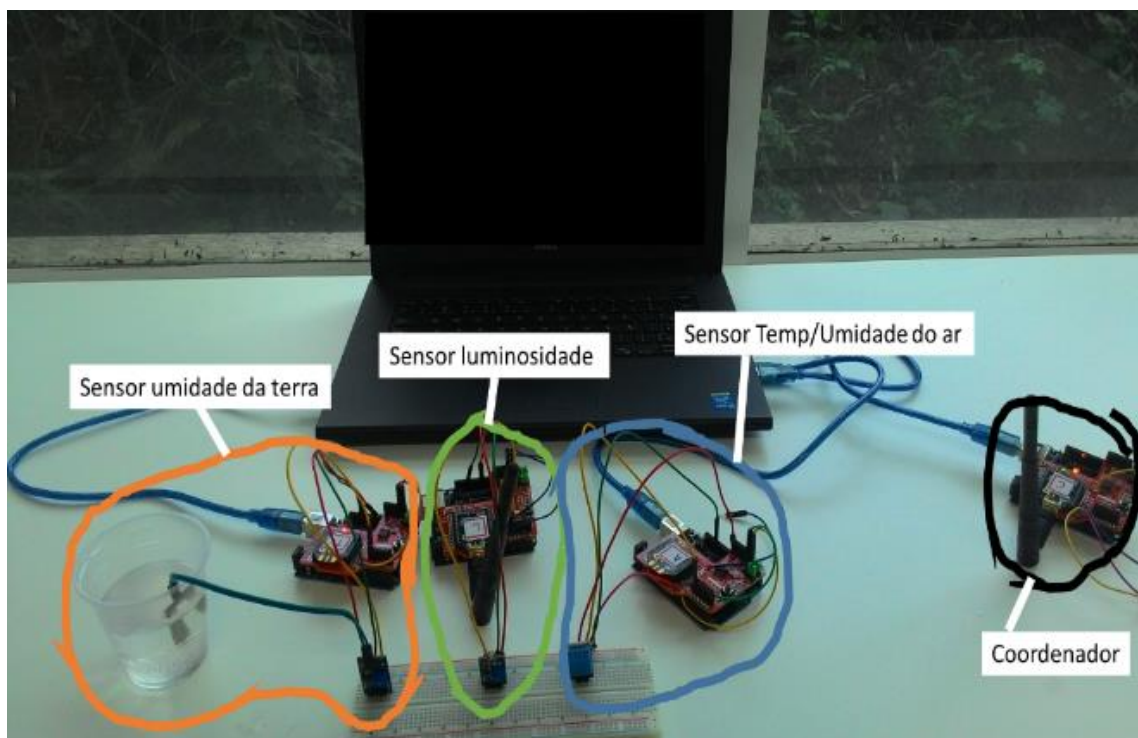


Figura 24. Foto da RSSF desenvolvida.

Durante a captura dos dados foram realizadas simulações de variação de iluminação, temperatura, umidade do ar e umidade da terra. Para o nó sensor de luminosidade foi bloqueada a luz e depois utilizada uma lanterna diretamente no sensor. Com valor inicial de luminosidade igual a 7, os valores abaixo de 7 indicam menor quantidade de luz capturada pelo sensor e os valores acima de 7 indicam maior quantidade de luz, como mostrado no gráfico da Fig. 25.

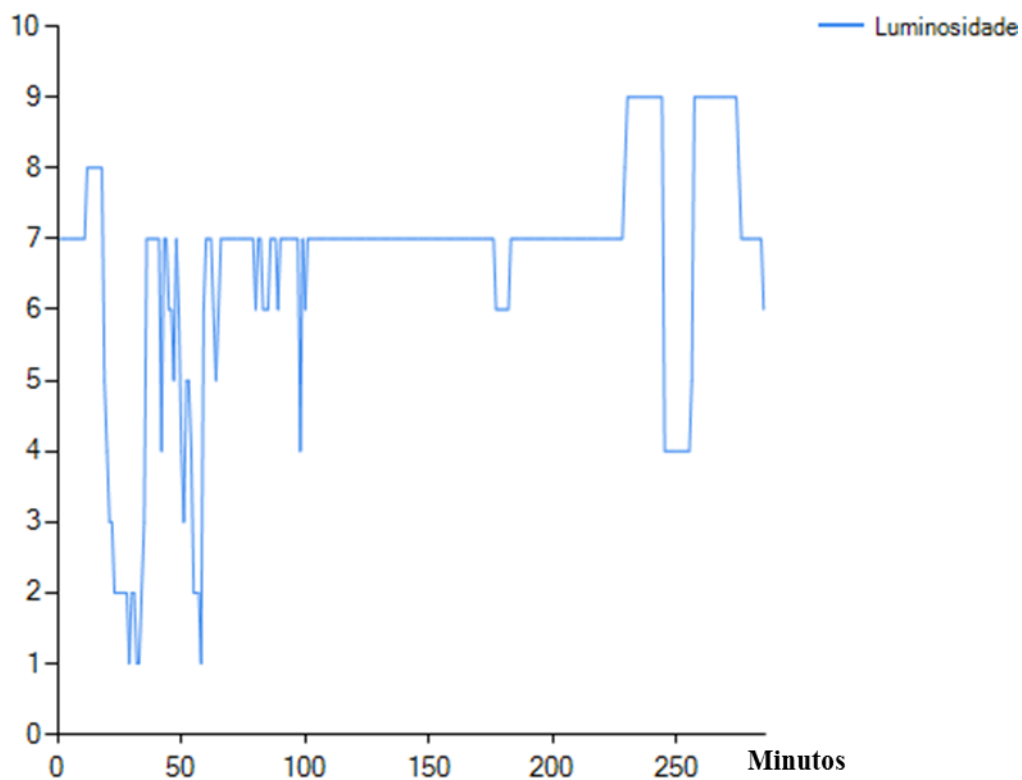


Figura 25. Gráfico de variação de luminosidade.

Foi utilizado um aquecedor de ar elétrico diretamente sobre o nó sensor de temperatura/umidade do ar como mostra a Fig. 26. Conforme o valor da temperatura foi elevado a umidade do ar diminuiu. A Fig. 27 mostra a variação de temperatura/umidade do ar deste experimento.

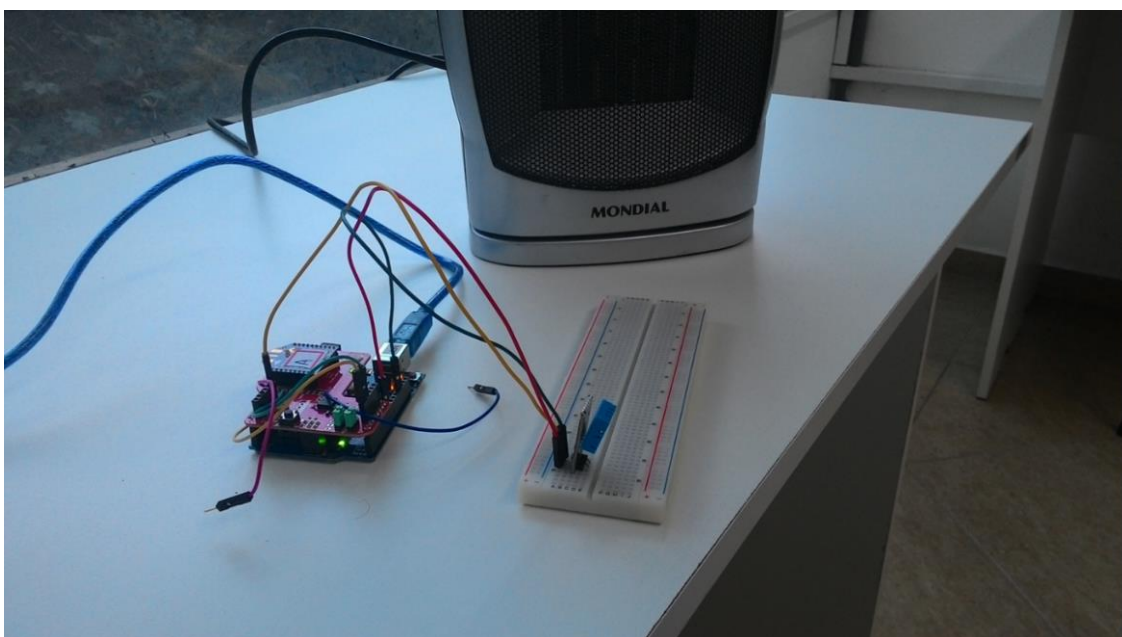


Figura 26. Nó sensor de ar com aquecedor de ar elétrico.

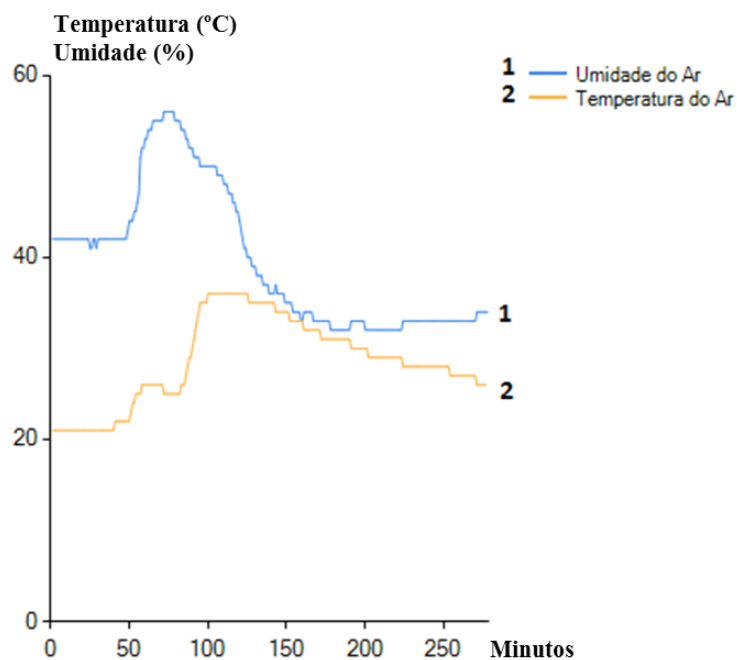


Figura 27. Gráfico de variação de umidade e temperatura do ar.

O nó sensor de umidade da terra estava inicialmente conectado com o sensor higrômetro submerso em um copo de água, para simular terra molhada indicando valor 6 conforme Fig. 28. O sensor foi removido e inserido no copo para simular terra seca ou molhada. Um gráfico com a variação do nível de umidade do sensor é mostrado na Fig. 29.

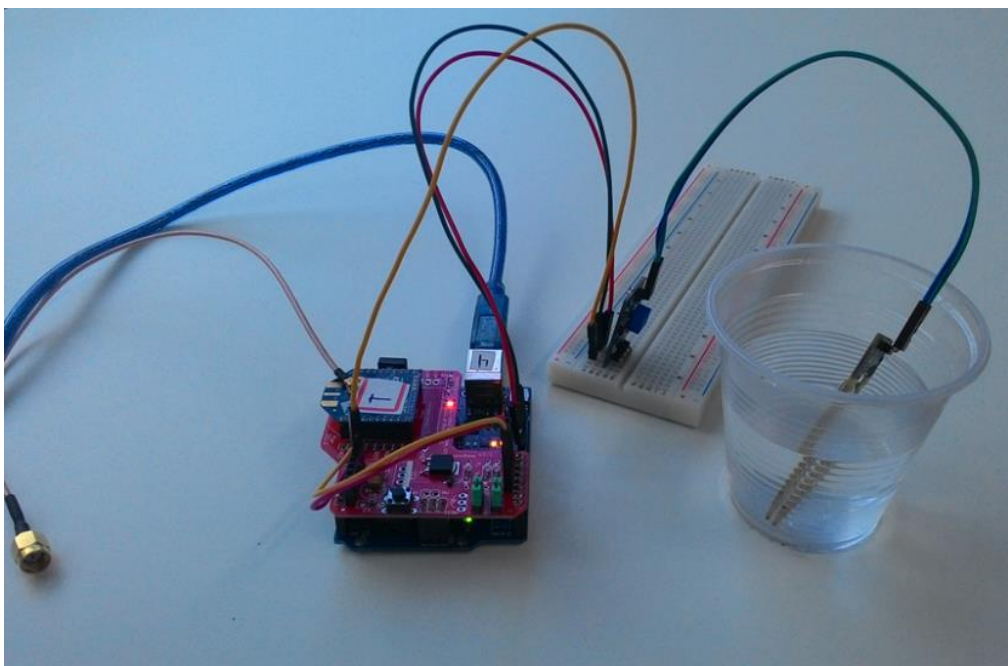


Figura 28. Sensor de umidade submerso em um copo de água.

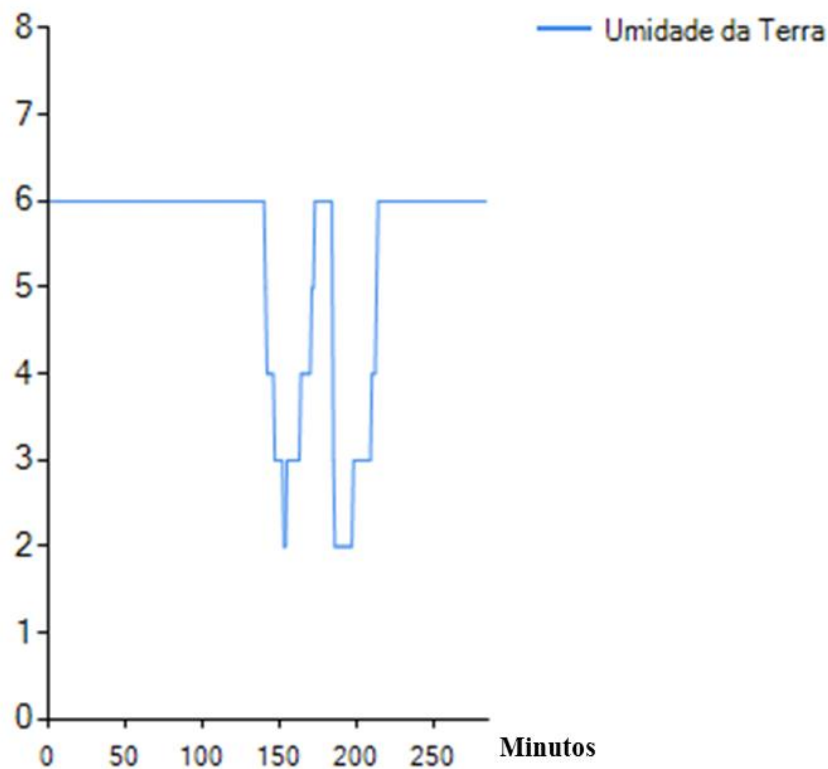


Figura 29. Gráfico de variação do nível de umidade.

Esse experimento demonstrou o funcionamento de uma RSSF capaz de capturar dados reais do ambiente, por meio de nós sensores, com intervalo de amostragem de 1 segundo.

5.2 Experimento 2 – Captura de Dados por 36 Horas

Foi desenvolvida uma RSSF com um nó coordenador e três nós sensores em ambiente controlado para verificar o comportamento da rede em longos períodos de funcionamento. Este experimento foi realizado em ambiente externo. Os nós sensores são de luminosidade, umidade/temperatura do ar e umidade da terra já descritos anteriormente. Neste experimento os valores da escala do nó de umidade da terra foram alterados, onde o valor 5 representa molhado e o valor 7 representa seco. Esta alteração ocorreu no programa do microcontrolador para testar outra forma de representação dos valores apresentados.

A Fig. 30 mostra uma foto dos nós sensores da RSSF demarcados por três cores diferentes. A área demarcada por amarelo é o nó sensor de temperatura/umidade do ar. A cor vermelha demarca o nó de umidade da terra. O azul é o nó sensor de luminosidade.

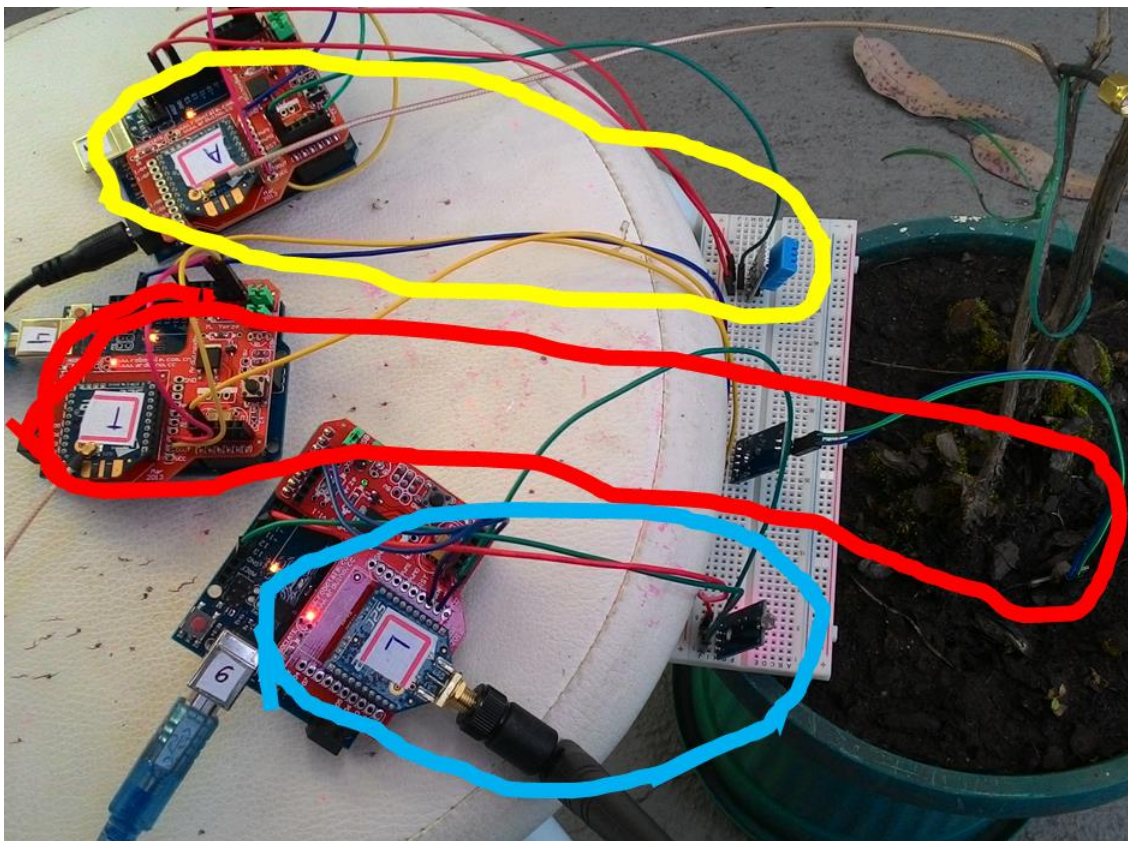


Figura 30. Foto dos nós sensores da RSSF.

Os nós sensores foram configurados para capturar os dados dos sensores com intervalo de um minuto por um período de 36 horas. Os dados capturados foram encaminhados para o nó coordenador, que transferiu as informações para um computador usando a porta Serial/USB. Os dados capturados foram salvos em um arquivo de log.

O arquivo de log criado foi aberto no programa leitor de log e os gráficos de variação de luminosidade, variação de temperatura e umidade do ar e variação de umidade da terra foram gerados.

O arquivo de log foi aberto no software leitor de log, sendo exibido em forma de tabela conforme a Fig. 31, que mostra as informações recebidas dos nós sensores, indicando data, hora, nó de origem, estado da terra ou luz, umidade do ar em % e temperatura do ar em °C

Abrir GraficoTerra Grafico Ar GraficoLuz						
	Data	Hora	Nó	Estado	Umidade %	Temperatura °C
▶	06-26-2016	23:09:16	Luz	0 Baixa luminosidade		
	06-26-2016	23:09:20	Terra	Umida		
	06-26-2016	23:10:06	Ar		63	13
	06-26-2016	23:10:16	Luz	0 Baixa luminosidade		
	06-26-2016	23:10:20	Terra	Umida		
	06-26-2016	23:11:06	Ar		63	13
	06-26-2016	23:11:16	Luz	0 Baixa luminosidade		
	06-26-2016	23:11:19	Terra	Umida		
	06-26-2016	23:12:06	Ar		63	13
	06-26-2016	23:12:16	Luz	0 Baixa luminosidade		
	06-26-2016	23:12:19	Terra	Umida		
	06-26-2016	23:13:06	Ar		63	13
	06-26-2016	23:13:16	Luz	0 Baixa luminosidade		
	06-26-2016	23:13:19	Terra	Umida		
	06-26-2016	23:14:06	Ar		63	13
	06-26-2016	23:14:16	Luz	0 Baixa luminosidade		
	06-26-2016	23:14:19	Terra	Umida		
	06-26-2016	23:15:06	Ar		63	13

Figura 31. Tela do software leitor de log com informações do experimento 2.

A Fig. 32 mostra um gráfico com a variação de umidade e temperatura do ar durante o período de captura. É possível observar por volta do minuto 800 houve um pico na temperatura e uma queda na umidade devido a incidência direta da luz solar no sensor DHT11. Em um segundo momento por volta do minuto 1800 houve erro nos valores de temperatura capturados.

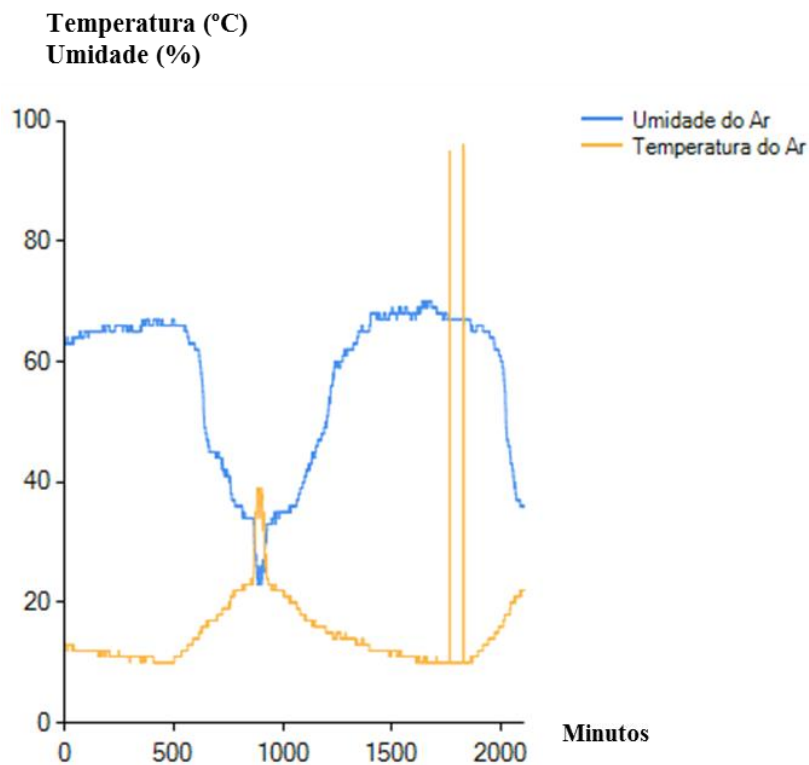


Figura 32. Gráfico de variação de temperatura e umidade do ar em 36 horas.

A Fig. 33 mostra um gráfico com a variação de luminosidade. Os valores acima de 0 indicam captura de luz. Por volta do minuto 1200 é possível observar um pequeno pico de luz com valor 3. Esse pico é referente a uma lâmpada que foi acesa no período noturno, permitindo a verificação dos nós sensores e acompanhamento do funcionamento da rede.

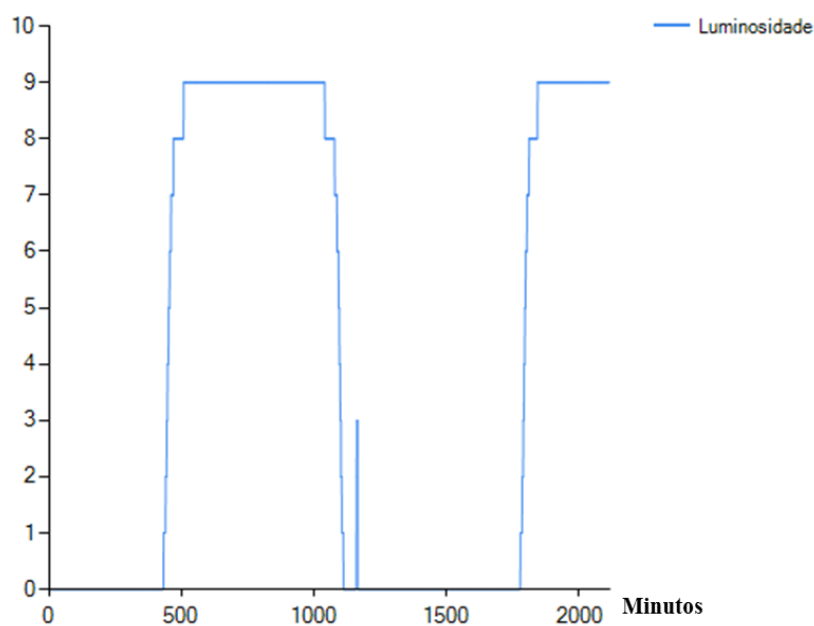


Figura 33. Gráfico de variação de luminosidade em 36 horas.

Na Fig. 34 é mostrado um gráfico com a variação de umidade da terra. Inicialmente a terra estava molhada indicando valor 5, por volta do minuto 600 a terra foi irrigada novamente e naturalmente drenada pelos furos do vaso. Por volta do minuto 1800 a terra se apresenta seca. Essas informações podem auxiliar na decisão de irrigar a terra no momento em que se torna seca.

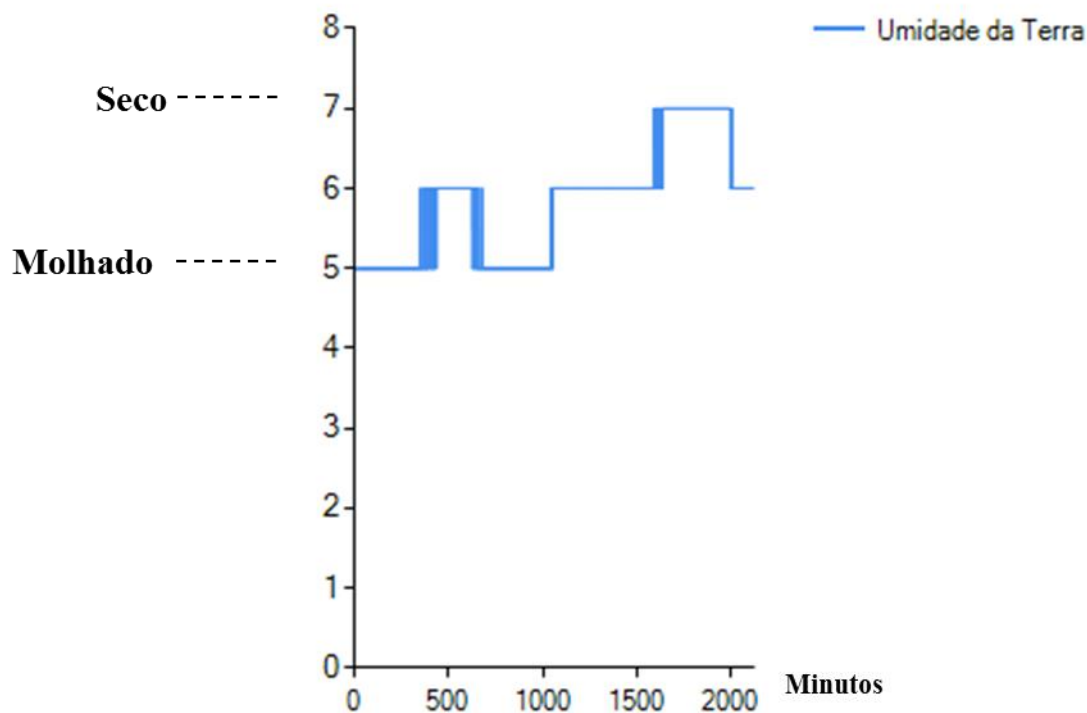


Figura 34. Gráfico de variação de umidade da terra.

5.3 Experimento 3 - Captura de Dados por 72 Horas com Atuador.

No 3º experimento foi desenvolvida uma RSSF para verificar a variação da umidade da terra de um vaso com uma planta e automaticamente acionar uma bomba de água caso a terra se torne seca. Os nós sensores foram configurados para capturar dados a cada um minuto por 72 horas. É esperado que cada nó sensor envie 4320 mensagens neste intervalo de tempo.

Foram utilizados um nó coordenador, três nós sensores e um nó atuador. Os três nós sensores são de luminosidade, umidade da terra, umidade e temperatura do ar. Um nó atuador foi criado com uma bomba de água e um motor de passo.

Nesse experimento, com atuador, o nó coordenador foi configurado com um algoritmo que analisa os pacotes recebidos para determinar sua origem, processar os dados e enviar uma mensagem para o nó atuador acionar a bomba de água, caso a terra esteja seca. A execução deste programa impossibilita a conexão direta da antena Xbee com o programa XCTU, afetando o recebimento das informações e geração de log.

Um programa alternativo denominado PLX-DAQ foi utilizado para fazer a leitura dos dados recebidos na porta COM do computador, visando a criação de um arquivo de log. O PLX-DAQ é uma macro executada dentro de um arquivo excel que habilita o recebimento de informações da porta COM. A conexão da porta COM é limitada pela macro com a velocidade de 9600b/s, logo todas as antenas e Arduínos tiveram suas portas seriais configuradas na mesma velocidade.

Após iniciar a conexão do nó coordenador na porta COM com o PLX-DAQ, foi executado o comando `Serial.println("LABEL ,Hora ,Luz, UmidadeAr ,TemperaturaAr ,UmidadeTerra");`. Este comando inicia a 1ª linha no excel, e nomeia as colunas Luz, UmidadeAr, TemperaturaAr e UmidadeTerra. A separação das colunas é delimitada pela vírgula “,”.

O nó coordenador analisa cada pacote recebido dos nós sensores e envia as informações para o PLX-DAQ de acordo com o nó de origem. É utilizado o comando `Serial.println("DATA,TIME,");` para enviar informações para o PLX-DAQ. Cada vez que este comando é executado uma linha é preenchida no excel.

A Fig. 35 mostra a RSSF desenvolvida para este experimento. Uma caixa de madeira foi utilizada como suporte dos nós da rede ao redor de um vaso com uma planta. O vaso possui furos em sua parte inferior para drenar o excesso de água da terra.

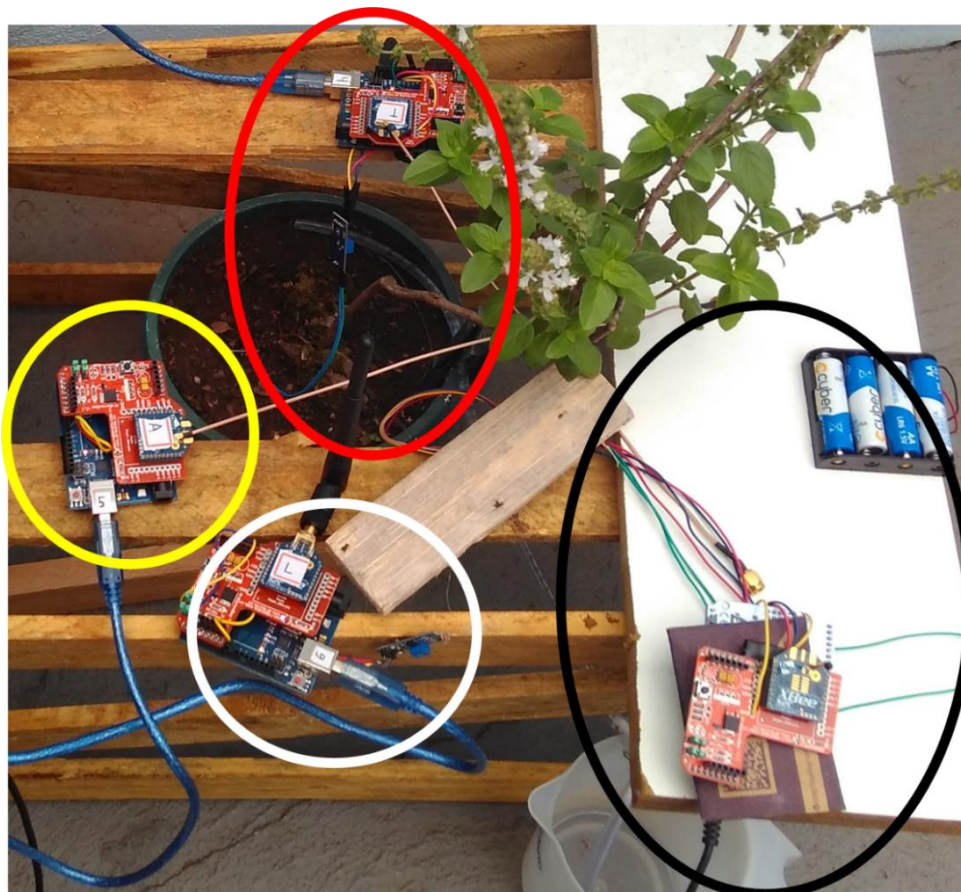


Figura 35. RSSF com 3 nós sensores e um atuador. Em amarelo o nó sensor de ar, em branco o nó sensor de luminosidade, em vermelho o nó sensor da terra e em preto o nó atuador.

O sensor higrômetro foi inserido na terra do vaso para capturar os valores referentes a umidade da terra e enviar para o nó coordenador. Se o valor capturado pelo sensor for igual ou menor que 2, o coordenador enviará uma mensagem para o nó atuador ativar a bomba de água, que irá irrigar a terra com uma mangueira, como visto na Fig. 36. A Fig. 37 mostra o nó atuador conectado no driver DC, que está conectado à bomba de água, e as pilhas para alimentar a bomba de água.



Figura 36. Sensor de terra e mangueira da bomba de água.

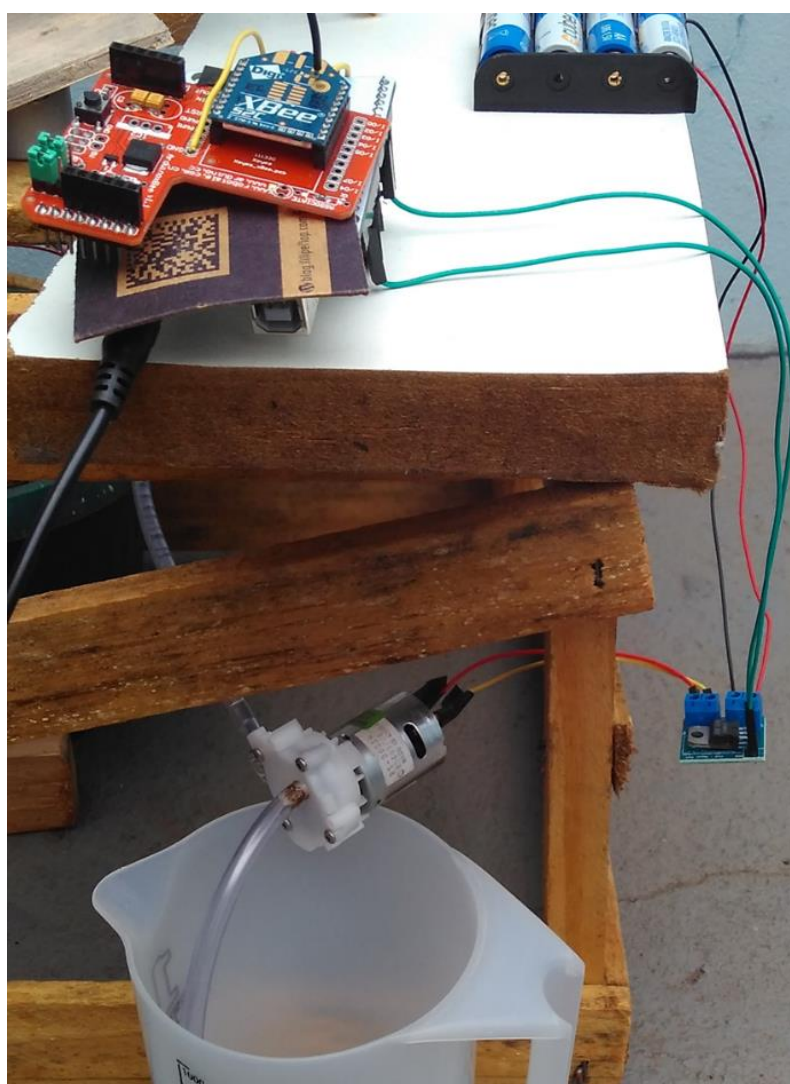


Figura 37. Nó atuador com uma bomba de água.

Após o final do experimento os dados preenchidos no excel foram filtrados conforme o nó de origem. Foram gerados 3 gráficos indicando a variação de temperatura e umidade do ar, luminosidade e umidade do solo.

A Fig. 38 mostra o gráfico de variação de temperatura e umidade do ar. A variação da temperatura é representada pela linha azul e a variação da umidade do ar é representada pela linha laranja. Alguns valores errados foram capturados em três ocasiões. Por volta dos minutos 1000, 1900 e 2800, é possível observar picos nos valores de temperatura e umidade. No total 4242 pacotes foram enviados para o nó coordenador.

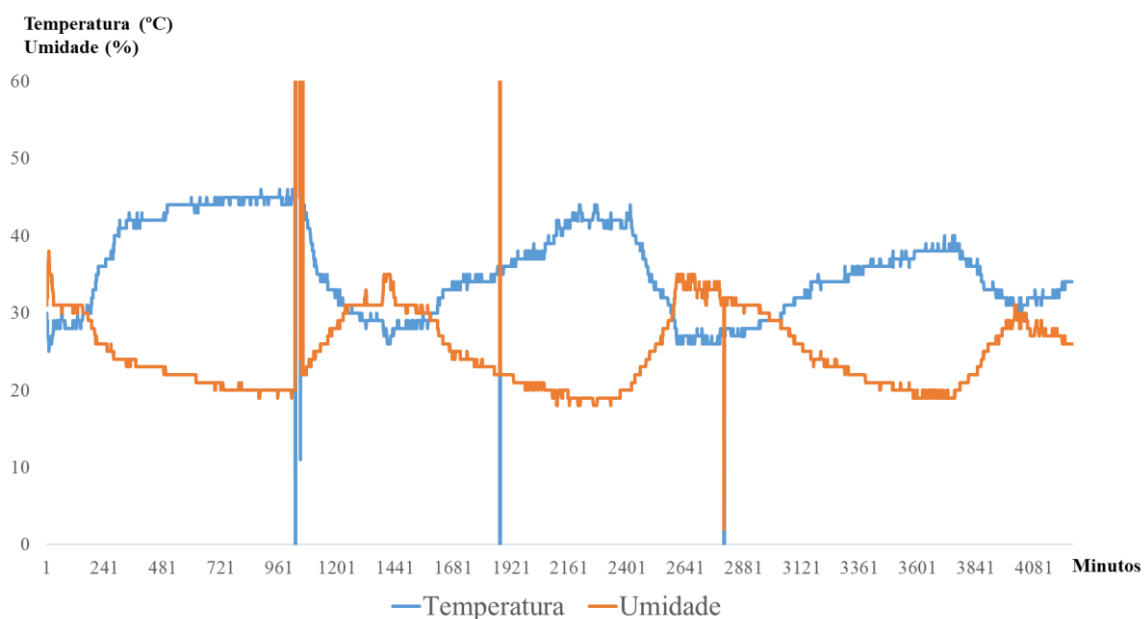


Figura 38. Gráfico de variação de temperatura e umidade do ar.

A Fig. 39 mostra o gráfico de variação de luminosidade. No gráfico os valores estipulados são 9 para muita luz e 0 para escuro. A captura e envio dos valores pelo nó sensor se mostrou instável no período noturno onde muitos pacotes apresentaram erros ou não foram enviados ao nó coordenador. No total 2832 pacotes foram enviados para o nó coordenador a partir do nó sensor de luminosidade.

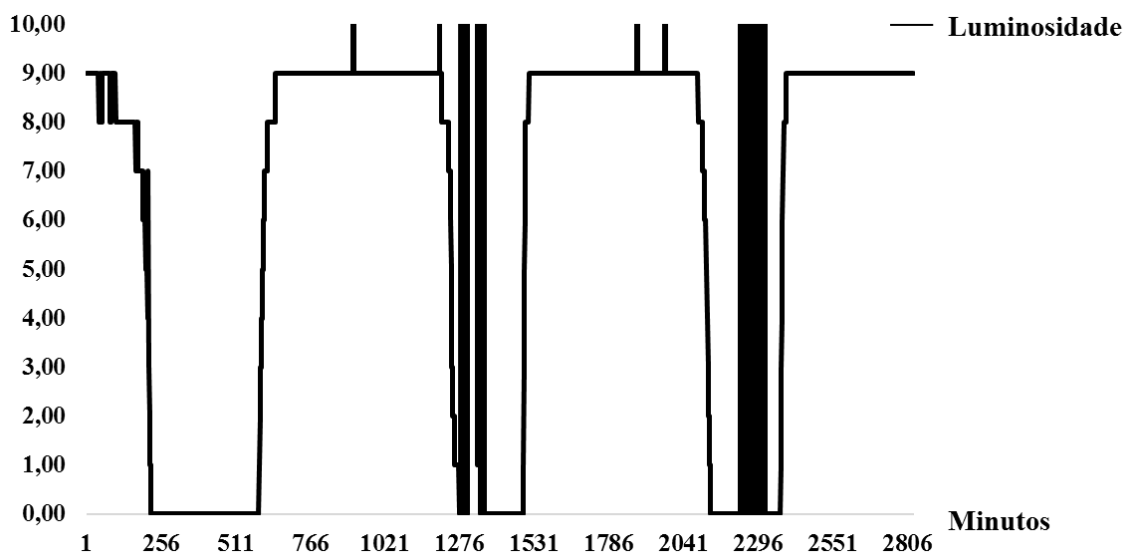


Figura 39. Gráfico de variação de luminosidade.

A Fig. 40 mostra o gráfico de variação de umidade da terra no período de 72 horas. No início do experimento a terra estava seca e o higrômetro capturou valor 2. A terra foi manualmente irrigada e após, o valor 6 foi capturado pelo sensor, indicando que a terra estava molhada. No decorrer do experimento a bomba de água foi automaticamente acionada 12 vezes. Por volta dos minutos 1600, 2800 e 3300 visualizam-se períodos em que valores 2 ou menor foram capturados repetidamente. Estes períodos indicam que o nó atuador não recebeu pacotes do nó coordenador para ativar a bomba de água. O nó sensor do solo enviou 4136 pacotes para o nó coordenador neste experimento. Os valores capturados acima de 6 são considerados pacotes com erro.

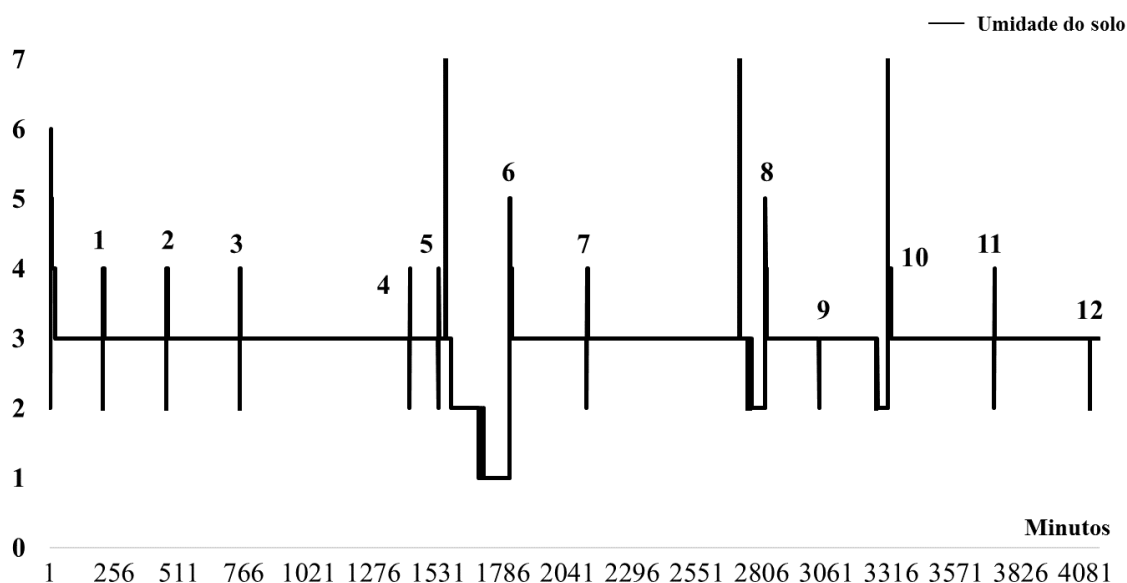


Figura 40. Gráfico de variação de umidade da terra.

A Fig. 41 mostra uma visão detalhada do acionamento automático número 1 da Fig. 40. No instante do minuto 211 o valor de umidade do solo capturado foi 2, assim o nó atuador recebeu uma mensagem do nó coordenador para acionar a bomba de água. O nó atuador acionou a bomba de água por 5 segundos e no minuto 212 o valor de umidade do solo capturado foi 4, indicando que houve variação na umidade do solo.

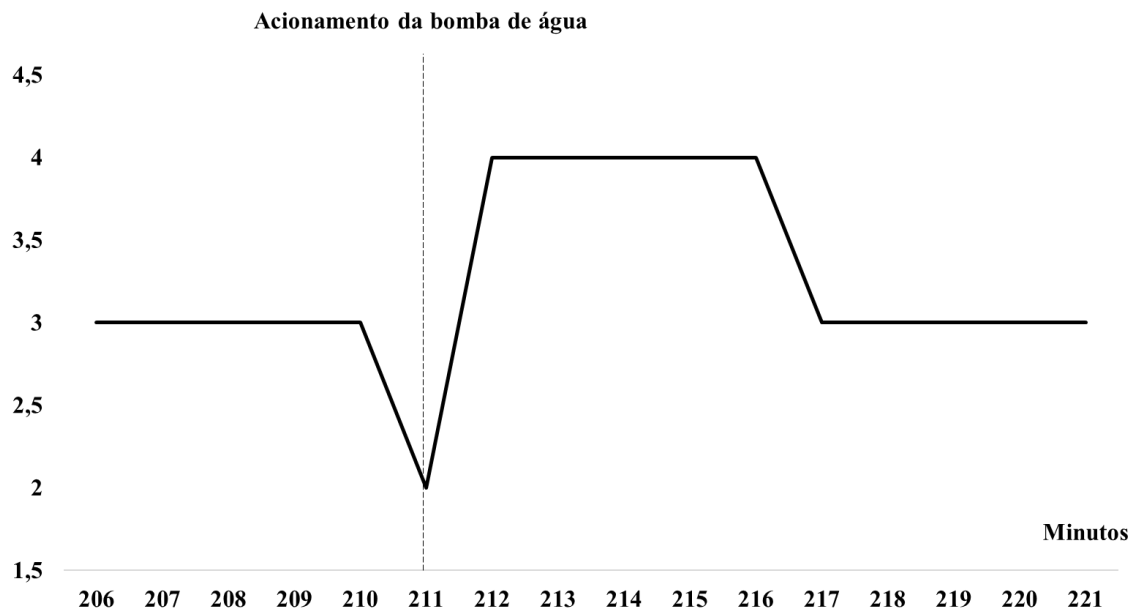


Figura 41. Gráfico detalhado de um acionamento automático da bomba de água.

No decorrer deste experimento ocorreram erros de rede onde não foi possível identificar a raiz do problema. A Tabela 9 mostra o percentual de pacotes perdidos para os três tipos de nós sensores, considerando um total de 4320 de pacotes esperados para cada nó sensor.

Tabela 9. Comparação de erros de pacotes enviados por cada nó sensor.

Nó sensor	Pacotes recebidos	Erro (%)
Ar	4242	1,81
Solo	4136	4,26
Luz	2832	34,44

Capítulo 6 - Estimação Teórica dos

Números de Nós Sensores e Atuadores

A estimação teórica do número de nós sensores e atuadores é realizada, em um primeiro momento na subseção 6.1, considerando apenas as limitações das antenas Xbee utilizadas nos nós desenvolvidos nesta dissertação.

Na subseção 6.2 são feitos cálculos teóricos para estimar a quantidade de nós sensores e atuadores possíveis de acomodar utilizando teoria de redes de filas com aplicação dos teoremas de Burke e Jackson, detalhados no Anexo A.

6.1 Considerando Limite de Endereço

A RSSF hierárquica adotada neste trabalho possui uma estrutura de árvore, com um nó coordenador no nível mais alto denominado nó zero. O nó coordenador tem um limite de N nós filhos conectados diretamente a ele. Os nós diretamente conectados ao coordenador estão no nível 1. Cada nó no nível 1 pode ter N filhos e repete-se em cada nível. Esses níveis correspondem aos níveis intermediários e são utilizados para rotear pacotes para o nó coordenador. Os nós nos níveis intermediários são denominados nós roteadores. Os nós nos níveis mais baixos são denominados nós dispositivos finais, podendo ser nós sensores ou atuadores.

É assumido que o protocolo Zigbee tem um total de endereços locais denominado Z_{MAX} . O número de nós roteadores nos níveis intermediários são denominados R, e cada nó roteador pode ter N nós filhos. O número de nós roteadores conectados diretamente aos nós sensores é denominado R_s e também pode ter N nós filhos. O número total de nós sensores, no nível mais baixo, é denominado T_s . Assim podemos escrever como:

$$Z_{Max} = T_s + R_s + R \quad (1)$$

O valor de R_s é dado pelo total de nós sensores, dividido pelo número de nós filhos N em um roteador:

$$R_S = \frac{T_S}{N} \quad (2)$$

O valor de R é dado pela soma de roteadores intermediários em todos os níveis acima R_S . O primeiro nível acima de R_S é definido pelo total de roteadores no nível de R_S , dividido pelo número de nós filhos N que um roteador pode ter. É assumido que todos os roteadores podem acomodar até N nós filhos. Logo pode ser escrito:

$$R = \frac{R_S}{N} + \frac{R_S}{N^2} + \frac{R_S}{N^3} \dots \quad (3)$$

Substituindo (2) e (3) em (1), é obtido:

$$Z_{Max} = T_S + \frac{T_S}{N} + \frac{T_S}{N^2} + \frac{T_S}{N^3} \dots = T_S \left(1 + \frac{1}{N} + \frac{1}{N^2} + \frac{1}{N^3} \dots \right) \quad (4)$$

Seja n, o número de níveis na estrutura hierárquica. Neste caso, a soma entre parêntesis é uma série geométrica finita, e a Eq. 4 pode ser reescrita como:

$$Z_{Max} = T_S \left(\sum_{k=0}^{n-1} \left(\frac{1}{N} \right)^k \right) = T_S \left(\frac{1 - \left(\frac{1}{N} \right)^n}{1 - \frac{1}{N}} \right) \quad (5)$$

Para um número muito grande de níveis, isto é, n tendendo a infinito, a Eq. 5 pode ser escrita como:

$$Z_{Max} = \frac{T_S * N}{N - 1} \quad (6)$$

Como exemplo numérico é suposto que a antena Xbee Zigbee tem um limite de N=14 nós filhos e o protocolo Zigbee tem um máximo de $Z_{Max} = 65535$ endereços locais. Aplicando esses valores em (5), pode ser encontrado o número máximo de nós sensores T_{Smax} , e o total de roteadores R_{max} .

$$T_{S_{\max}} = \frac{65535 * 13}{14} \cong 60854 \text{ nós sensores} \quad (7)$$

$$R_{\max} = Z_{\max} - T_{S_{\max}} = 4681 \text{ nós roteadores} \quad (8)$$

6.2 Considerando Rede de Filas

A análise feita no subitem 6.1 considerou apenas o número máximo de endereços disponíveis. Nesta subseção será analisado o número de nós sensores possíveis de acomodar, pressupondo as características de tráfego dos nós sensores utilizando teorias de filas.

Na rede hierárquica adotada neste trabalho, como mostrado na subseção 6.1, os nós sensores, roteadores e coordenador constituem uma configuração em árvore. Considerando que os roteadores e coordenador possuem *buffers* para armazenar temporariamente os pacotes que chegam de cada nó, a configuração de árvore pode ser modelada como uma rede de filas.

Na subseção 6.2.1 é feita a estimativa de nós sensores possíveis de acomodar em uma RSSF hierárquica quando são variados os intervalos de envio de pacotes pelos nós sensores e os limites de carga para o nó coordenador e nós roteadores.

Na subseção 6.2.2 é recalculada a estimativa de nós sensores considerando que parte dos pacotes que saem do nó coordenador é enviada de volta para os nós atuadores como realimentação ou *feedback*.

6.2.1 Rede de Sensores sem Fio

A Fig. 42 mostra a RSSF hierárquica proposta com configuração em árvore sendo analisada como rede de filas. A Fig. 42 mostra que os nós sensores estão anexados aos roteadores de níveis mais baixos com *buffers*. No estágio intermediário apenas um nível de roteadores com buffers é mostrado e o coordenador com *buffer* no nível mais alto. Para a análise do sistema de filas as seguintes suposições são feitas:

- apenas um nível de nós roteadores no estágio intermediário.
- o tráfego gerado pelos nós sensores possuem uma distribuição Poisson.

- λ_s pcts/s é o tráfego parcial gerado pelos nós sensores e enviado para cada nó roteador no nível mais baixo em que são diretamente conectados.

- λ_R pcts/s é o tráfego parcial gerado pelos nós roteadores filhos conectados diretamente aos nós roteadores do nível 1.

- λ_C pcts/s é o tráfego chegando no nó coordenador gerado pelos nós roteadores filhos.

- $1/\mu$ é o tamanho do pacote.

- C é a capacidade do link da antena Zigbee em b/s.

- μ_C , μ_{Rx} e μ_{Rxx} são os tempos de serviço de pacote do nó coordenador, nós roteadores do nível 1 e nós roteadores no nível mais baixo, respectivamente. O indexador “x” pode assumir valores de 1 até o limite de nós filhos N .

A rede de filas na Fig. 42 pode ser analisada usando o teorema de Burke (veja Anexo A). Para calcular o número de nós sensores que o sistema pode acomodar, é considerado que a limitação de processamento está localizada na porta serial da antena Xbee Zigbee com valor $C = 115200$ bps.

A antena do nó coordenador Xbee Zigbee (Z_C) pode ter $N=14$ nós filhos. Cada um dos 14 nós roteadores (R) conectados ao coordenador possuem as filas R_1 a R_{14} . Cada roteador pode ter 14 filhos e assim sucessivamente. O trajeto formado por cada nó roteador do nível R , e seus nós filhos, é denominado um ramo da árvore, neste trabalho. Os nós sensores são configurados para enviar pacotes com tamanho médio de $1/\mu = 20$ bytes. Logo, a taxa de atendimento na saída do coordenador será $\mu_C = 115200/20 \times 8 = 720$ pacotes por segundo.

No caso da Fig. 42, uma fila M/M/1 pode ser utilizada em cada uma das filas, logo a fórmula (9) pode ser aplicada. O tempo médio gasto em cada fila é $E\{T\}$, considerando o tempo de espera no *buffer* mais o tempo de serviço.

$$E\{T\} = \frac{1}{\mu - \lambda} \quad (9)$$

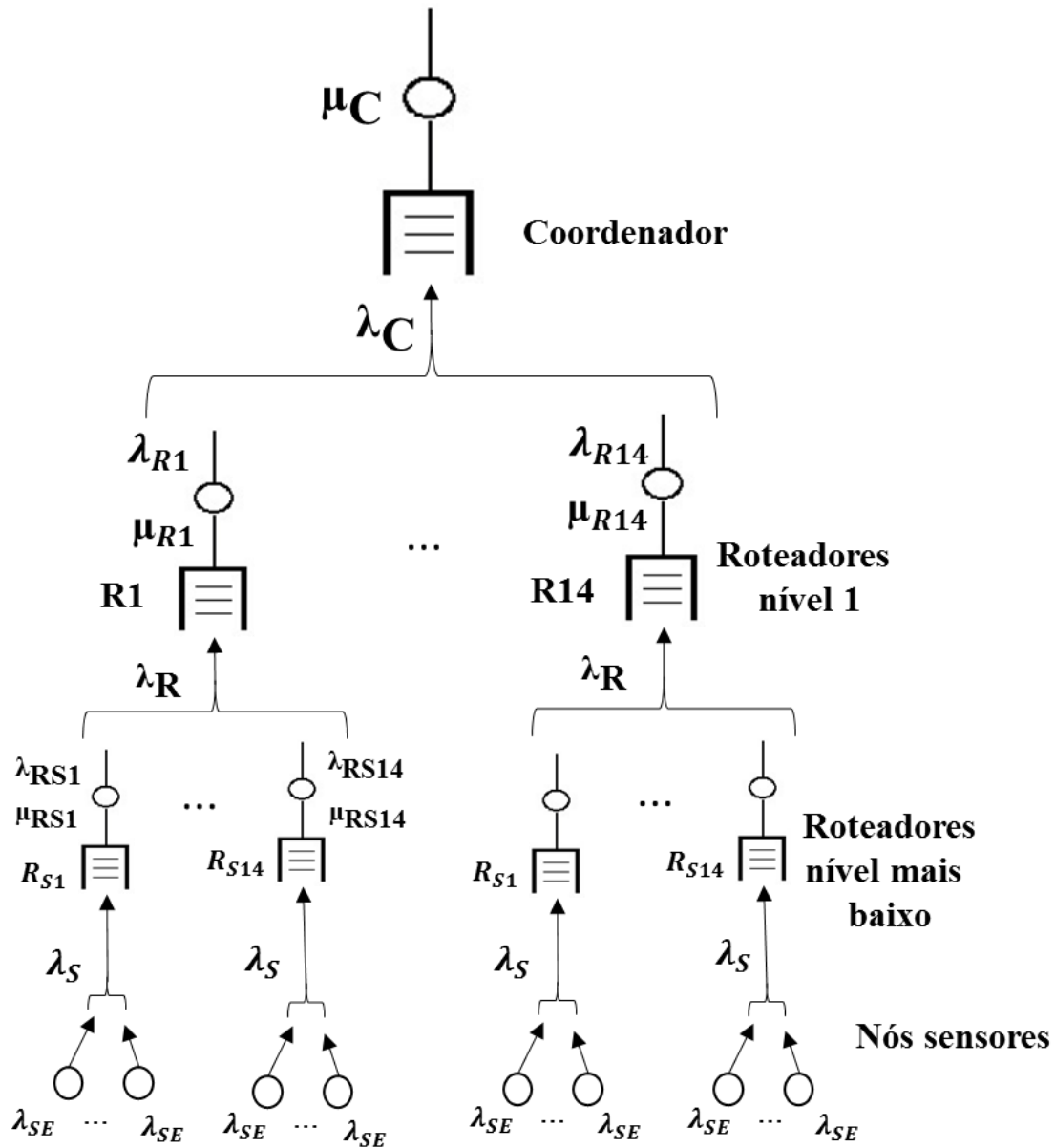


Figura 42. RSSF hierárquica proposta.

A Fig. 43 mostra um gráfico da variação de tempo média $E\{T_C\}$, em que um pacote irá ficar no coordenador, em milissegundos, enquanto ρ_C varia de 0.1 a 0.9 com $\mu_C = 720$ pcts/s. Pode ser observado que $E\{T_C\}$ tem uma inclinação abrupta para cima após $\rho_C = 0.7$. Assim, para assegurar a estabilidade do sistema, o valor de ρ_C é limitado a 0.5.

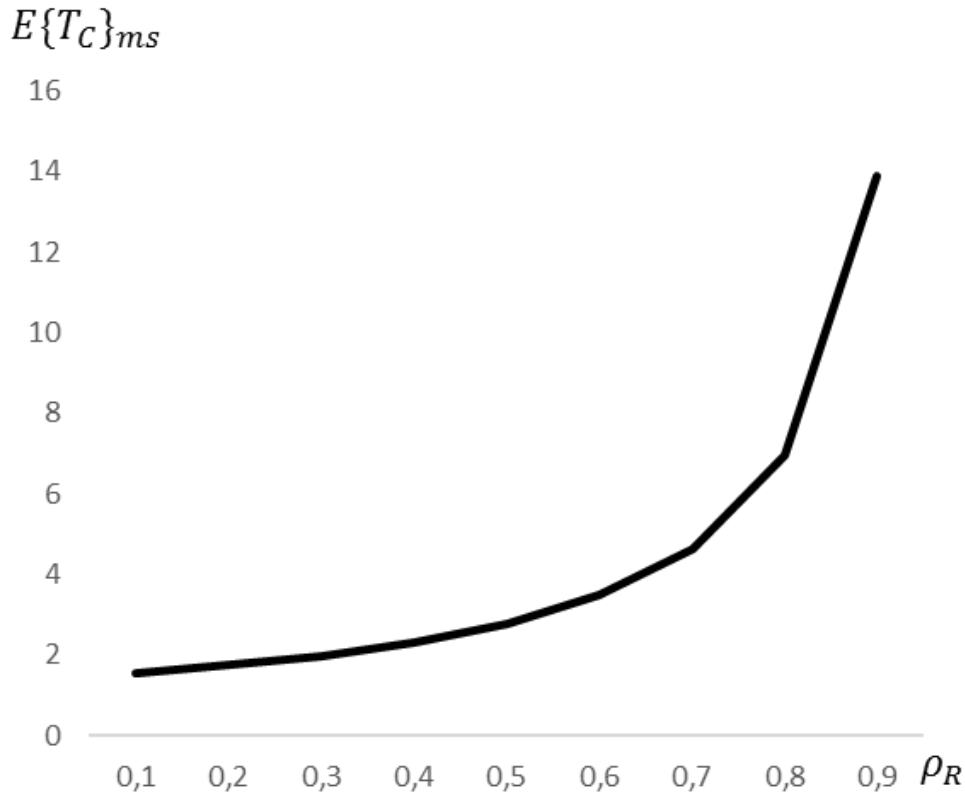


Figura 43. Variação no tempo de espera $E\{T_C\}$ em milissegundos.

Com $\rho_C = 0.5$ e $\mu_C = 720$ pcts/s o valor de λ_C será 360 pcts/s. A taxa de pacotes λ_C é o total aplicado como entrada no nó coordenador. Considerando que o nó coordenador pode ter até $N=14$ nós filhos e que os pacotes estão chegando com a mesma distribuição nos nós roteadores nível 1, a taxa de saída de cada nó roteador será:

$$\lambda_{R1} = \frac{\lambda_C}{14} = \frac{360}{14} = 25.71 \text{ pcts/s} \quad (10)$$

Este também será o valor da taxa de saída em μ_{R1} do nó roteador R_1 . Supondo que a carga no nó roteador R_1 , ρ_{R1} , é igual a 0.5, então λ_R será 12.85 pcts/s. O tempo médio de espera no roteador R_1 , neste caso, será:

$$E\{T_{R1}\} = \frac{1}{25,71 - 12,85} = 0,0778s \text{ or } 77ms \quad (11)$$

Assumindo o mesmo raciocínio para o nó roteador R_{S1} no nível mais baixo, os seguintes valores são encontrados:

$$\lambda_{RS1} = \frac{\lambda_R}{14} = \frac{12.85}{14} \approx 0.92 \text{ pcts/s} \quad (12)$$

$$\lambda_S = 0.5 \times \lambda_{RS1} = 0.46 \text{ pcts/s} \quad (13)$$

$$E\{T_{RS1}\} = \frac{1}{0.92 - 0.46} = 2.1 \text{ s} \quad (14)$$

Supõe-se que cada nó sensor gere uma taxa média de pacotes de λ_{SE} . Finalmente, com λ_S dividido por λ_{SE} , será possível estimar quantos nós sensores o Sistema de monitoramento pode acomodar.

A Tabela 10 mostra o número de nós sensores possíveis de acomodar para diferentes taxas de pacote dos nós sensores.

Tabela 10. Número total de nós sensores possíveis de acomodar com diferentes taxas de pacotes.

λ_{SE}	λ_S/λ_{SE}	Total de nós sensores no ramo R1	TS
1 pct/s	-	-	-
1 pct/5s	2,3	32,2	451
1 pct/10s	4,6	64,4	902
1 pct/60s	27,6	386,4	5410
1 pct/300s	138	1932	27048

O número de nós sensores na Tabela 10 são encontrados considerando que a carga de 50% em ρ_C . Muitas outras considerações podem ser feitas, para a carga, para encontrar um número adequado de nós sensores para diferentes aplicações em estufas.

A Fig. 44 exibe diferentes valores de tempo médio de espera $E\{T\}$, em função de ρ_R , considerando ρ_C como parâmetro. Utilizando as curvas apresentadas no gráfico da Fig. 44 os valores mais convenientes para $E\{T\}$, ρ_C e ρ_R podem ser escolhidos para diferentes aplicações.

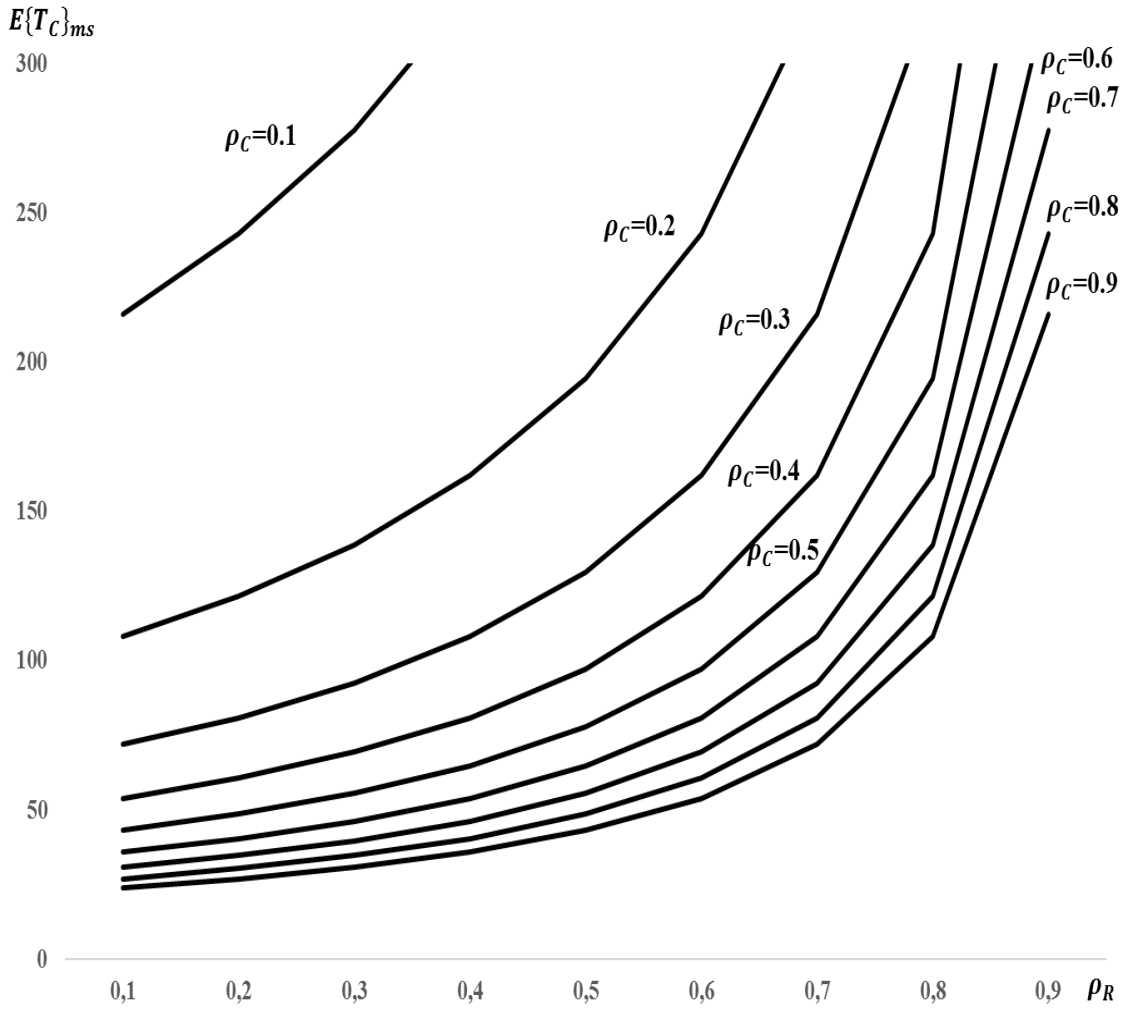


Figura 44. Curvas com variação de ρ_R e ρ_C

Na Fig. 45 é mostrado um gráfico, em milissegundos, da variação do tempo médio de espera, em um roteador do nível 1 $E\{T_R\}$, quando ambas cargas do nó coordenador e nó roteador no nível 1, ρ_C e ρ_r respectivamente, variam entre 0.1 e 0.9. Os valores de $E\{T_R\}$ antes de 0.3 e depois de 0.7 mostram uma variação abrupta na curva o que pode indicar instabilidade no sistema se as cargas forem abaixo de 0.3 ou acima de 0.7. Quando ρ_C e ρ_r estão em 0.5 o sistema irá funcionar em sua máxima eficiência com 77ms de espera no roteador.

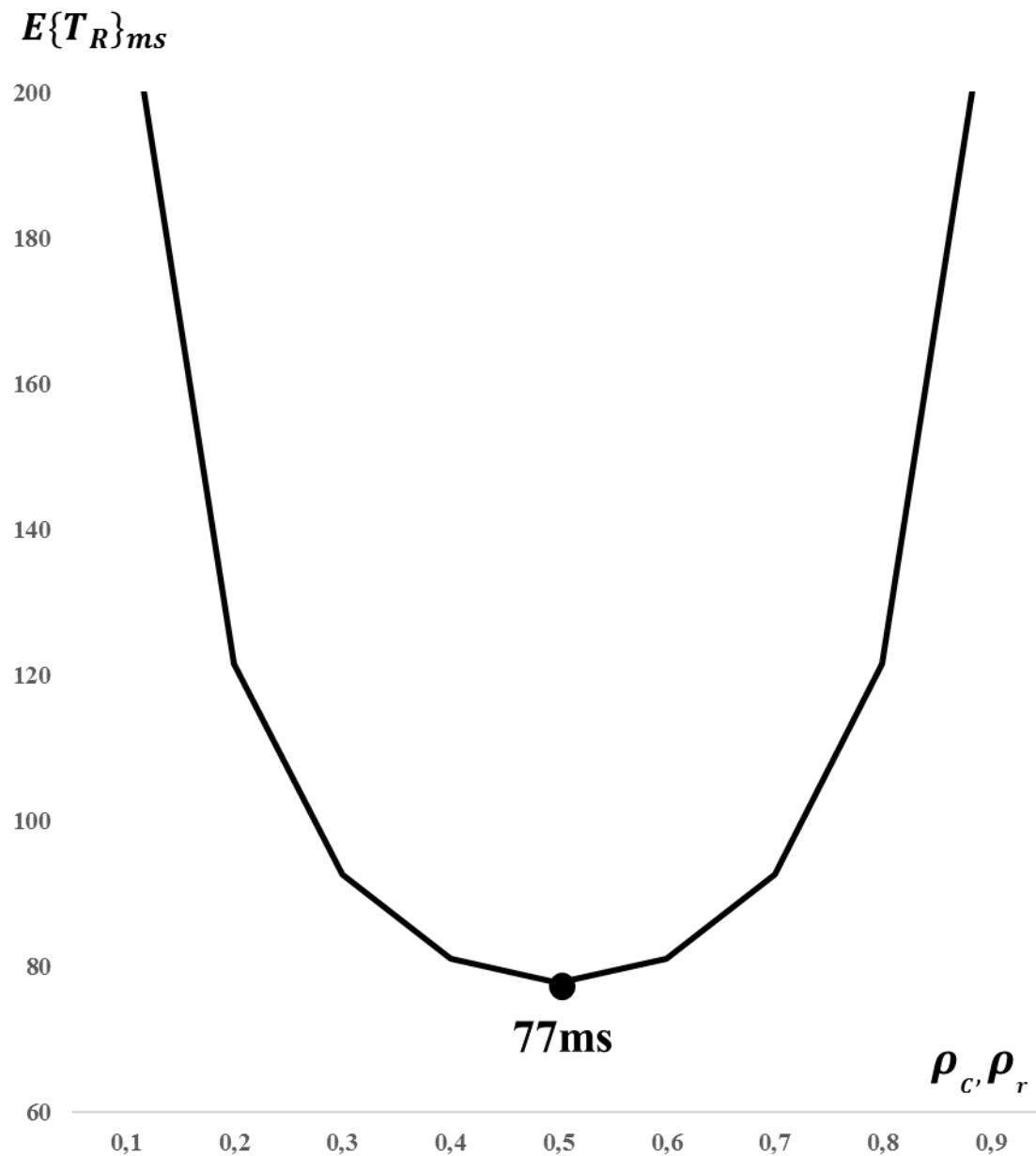


Figura 45. Variação $E\{T_R\}$ de ρ_c e ρ_r .

Na Fig. 46 uma curva com a soma dos tempos de espera nos três níveis é apresentada, sendo $E\{T\} = E\{T_C\} + E\{T_{R1}\} + E\{T_{RS1}\}$ em segundos. O valor de ρ no eixo x é associado aos valores de ρ_c, ρ_r e ρ_{RS} . Por exemplo, para $\rho = 0.5$, os valores de $\rho_c = \rho_r = \rho_{RS} = 0.5$, o total de tempo de espera será $E\{T\} \approx 2.2s$. Os tempos de espera entre $0.6 < \rho < 0.8$ aparentam ter valores mais baixos, mas como já visto, se $\rho > 0.6$ o sistema pode se tornar instável.

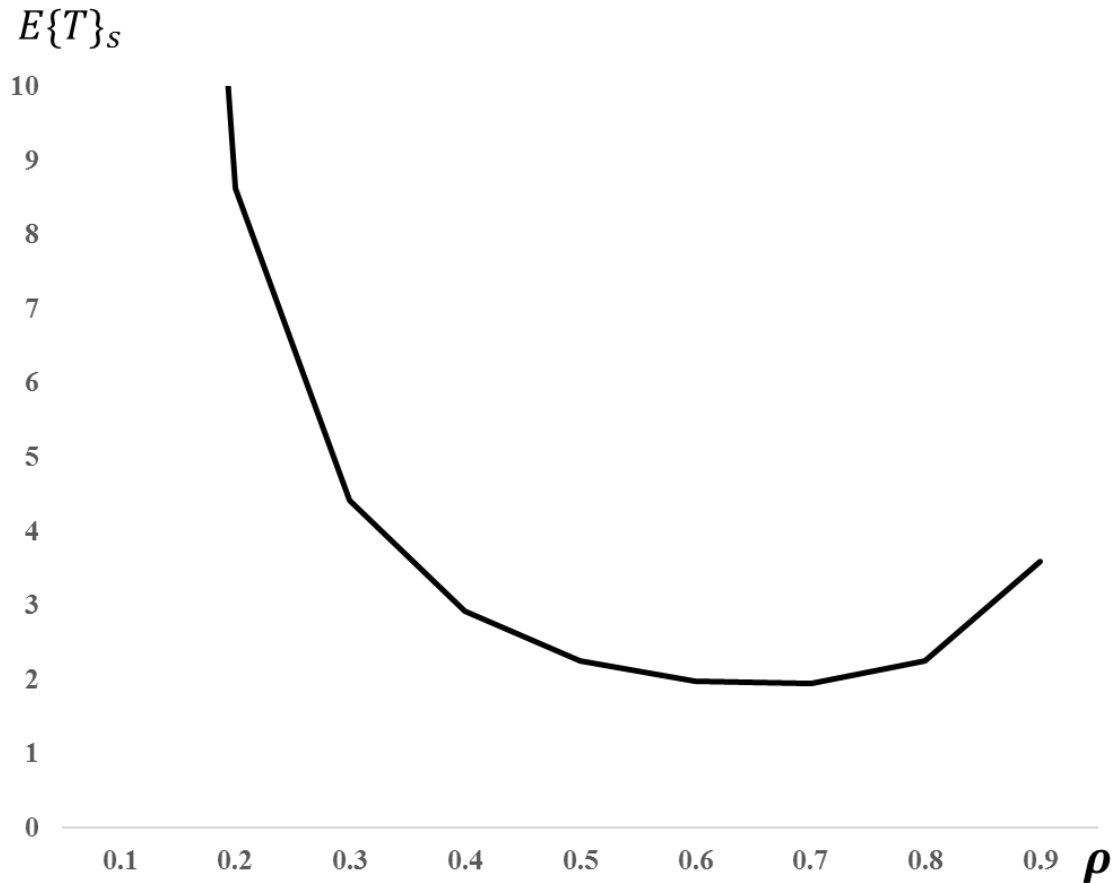


Figura 46. Tempo total de espera $E\{T\}$ em segundos.

6.2.2 Com Realimentação (*Feedback*)

A análise feita em 6.2.1 considera que no nível mais baixo, da RSSF proposta, há apenas nós sensores. Um sistema de monitoramento e controle para uma estufa agrícola utiliza atuadores para alterar as condições ambientais, assim parte dos pacotes será enviado para os nós atuadores. Este tráfego é denominado Realimentação ou Feedback.

Neste caso, supõe-se que o nó coordenador envie parte do tráfego de saída de volta para os nós de dispositivos finais para a ativação ou desativação de atuadores mostrado na Fig. 47. A figura mostra apenas um ramo da rede, que será usada para análise. A mesma análise pode ser usada para os outros ramos da rede como uma rede de filas. Neste caso o teorema de Jackson em teoria de redes de filas pode ser aplicado (veja Anexo A), logo a mesma fórmula M/M/1 da equação (9) pode ser usada, neste caso, para cada fila considerando o tráfego de realimentação ou *feedback*.

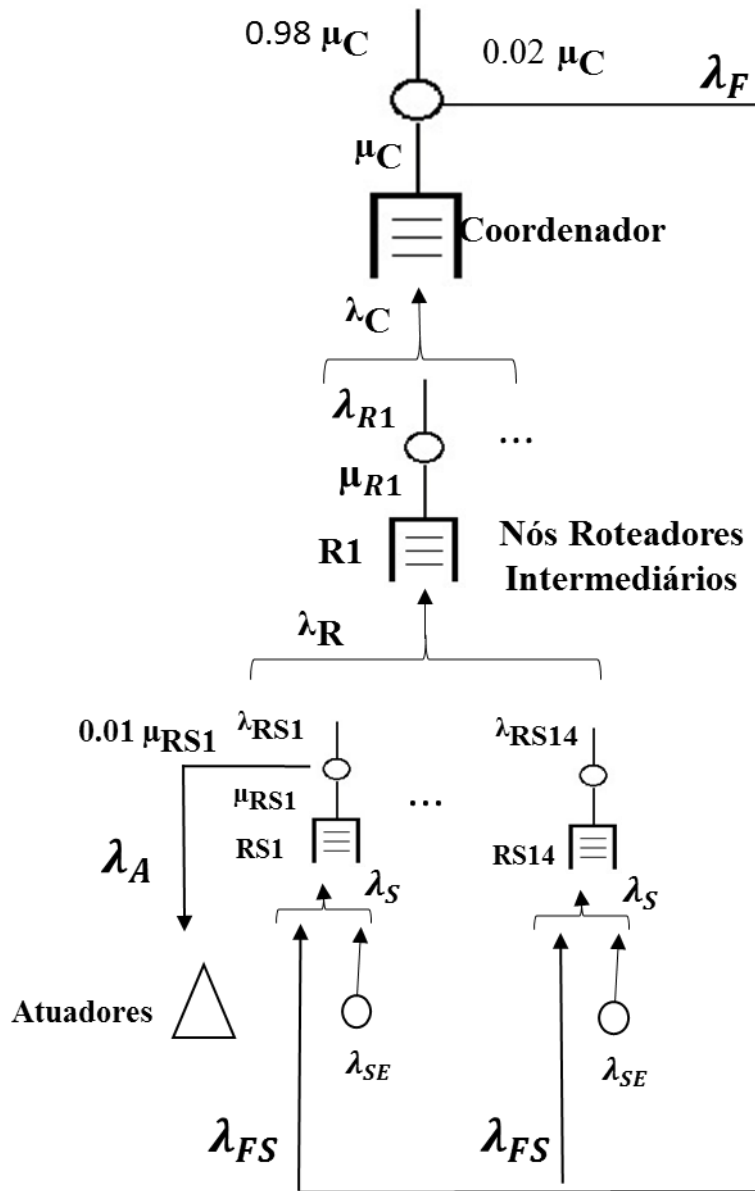


Figura 47. Exemplo de uma RSSF hierárquica com feedback em um ramo.

Para cálculos numéricos, foi considerado que apenas 2% do tráfego do coordenador seja retornado, distribuído igualmente, para os nós roteadores de nível mais baixo R_{Sx} . Também é considerado que os pacotes de feedback do nó coordenador serão retransmitidos pelos nós roteadores intermediários sem a necessidade de armazenagem dos pacotes. Logo, a taxa dos pacotes de *feedback* é $\lambda_F = 0.02\mu_C = 0.02 * 720 = 14.4\text{pct/s}$.

É assumido que em cada roteador R_{Sx} receberá igualmente o tráfego de feedback λ_F , do coordenador, assim, já que os nós roteadores intermediários podem ter até $N = 14$ filhos, e os de nível mais baixo também podem ter até $N = 14$ filhos, o tráfego de feedback λ_{FS} como entrada de R_{S1} será:

$$\lambda_{FS} = \frac{14.4}{14*14} \approx 0,07 \text{ pct/s} \quad (15)$$

$$\lambda_S = \lambda_{SE} + \lambda_{FS} \quad (16)$$

Também é assumido que 1% do tráfego de saída de um roteador R_{SX} será usado para ativação ou desativação de um nó atuador, como mostrado na Fig. 46. Assim, este tráfego será:

$$\lambda_A = 0.01\mu_{RS1} = 0.01 * 0.92 = 0,0092\text{pct/s} \quad (17)$$

$$\lambda_S = 0.5(\mu_{RS1} + \lambda_A) \approx 0.46\text{pct/s} \quad (18)$$

O tráfego total é λ_S , assim, se assumirmos uma estimativa de tráfego para cada nó sensor λ_{SE} , será possível calcular quantos nós sensores e atuadores esta RSSF hierárquica pode acomodar.

$$\lambda_{SE} = \lambda_S - \lambda_{FS} \approx 0,39 \text{ pct/s} \quad (19)$$

A Tabela 11 mostra o número de nós sensores e atuadores, possíveis de acomodar, para diferentes intervalos de taxas de envio de pacotes dos nós sensores, e o tráfego de *feedback* para nós atuadores, e considerando que as cargas de ρ_C e ρ_{RSx} são 50%.

Os valores definidos para as cargas do nó coordenador, nós roteadores e feedback na configuração da RSSF da Fig. 46 indicam que não é possível haver nós sensores se o intervalo entre pacotes for $\lambda_{SE} = 1\text{pct/s}$. Os nós atuadores deverão receber, em média, um pacote a cada 100 segundos. Não será possível alocar nós atuadores se a taxa média de pacotes $\lambda_A < 1\text{pct}/100\text{s}$.

Tabela 11. Números totais de nós sensores e atuadores possíveis de acomodar com diferentes taxas de pacotes.

λ_{SE}	λ_S/λ_{SE}	Total de nós sensores no ramo R1	TS	λ_A	Total de nós atuadores no ramo R1	Total de nós atuadores
1 pct/s	-	-	-	-	-	-
1 pct/5s	1,95	27,3	382	-	-	-
1 pct/10s	3,9	54,6	764	-	-	-
1 pct/60s	23,4	327,6	4586	-	-	-
1 pct/100s	39	546	7644	1	14	196
1 pct/300s	117	1638	22932	3	42	588

Conclusão

Um sistema de monitoramento de RSSF hierárquica para aplicação em estufas de tamanhos variados foi proposto nesta dissertação. O desenvolvimento e implementação do sistema proposto foram detalhados, e três experimentos foram executados em ambiente controlado. Os resultados indicaram que o sistema está operando corretamente nas condições testadas.

Análises teóricas foram feitas considerando duas situações diferentes para estimar o número de nós sensores que o sistema proposto pode acomodar. Na primeira situação a restrição de número de endereços locais, do protocolo Zigbee, foi considerado. É demonstrado que uma rede Zigbee pode acomodar um grande número de nós sensores e nós roteadores. Na segunda situação, o número de nós sensores possíveis de acomodar foi estimado considerando as características de tráfego, dos nós sensores, e teoria de rede de filas com dois casos analisados. No primeiro caso não há tráfego de realimentação do sistema. Os resultados mostraram que, dependendo das taxas de envio dos pacotes dos nós sensores, um grande número de nós sensores podem ser usados. No segundo caso, o tráfego de *feedback* do nó coordenador foi levado em consideração, mostrando que o *feedback*, na maioria dos casos, não afeta o número de nós sensores, mas pode ser significativo se a taxa de *feedback* for similar ou maior que a taxa de pacotes dos nós sensores.

Para trabalhos futuros, o sistema de monitoramento proposto será testado em experimentos de campo em uma estufa real. Além disso, diferentes modelos de tráfego dos nós sensores serão considerados para estimação de nós sensores possíveis de acomodar. Como continuação do desenvolvimento dos nós, poderá ser agregado uma placa solar como fonte de energia assim como a adição de funcionalidades para permitir que a RSSF participe da Internet das Coisas (IoT). Para a continuação do desenvolvimento do software Leitor de Log poderá ser adicionado a função de comunicação com a porta COM permitindo o recebimento de dados em tempo real, além de habilitar o acesso da RSSF direto com a internet.

Referências

- Ahonen, T., Virrankoski, R., Elmusrati, M. (2008), “Greenhouse Monitoring with Wireless Sensor Network”, *Mechtronic and Embedded Systems and Applications (MESA)*
- Akyildiz I.F., Su W., Sankarasubramaniam Y., Cayirci E. (2002), Wireless sensor networks: a survey. *Computer Networks*, p. 393–422.
- Arduíno Uno (2016), disponível em <www.arduino.cc/en/Main/ArduinoBoardUno>, acessado em 16/06/2016.
- Baterias 9v (2016), disponível em <<http://www.powerstream.com/9V-Alkaline-tests.htm>>, acessado em 05/09/2016.
- Benavente, J.C.C. (2010). *Monitoramento ambiental de vinhedos utilizando uma rede de sensores sem fio que coleta dados com um intervalo de amostragem variável*. Dissertação de Mestrado.
- DHT11 (2016), disponível em <<http://www.micropik.com/PDF/dht11.pdf>>, acessado em 05/07/2016.
- Digi (2016), disponível em <<http://www.digi.com/>>, acessado em 16/06/2016.
- Ervas Finas Horticultura (2016), disponível em <<http://ervasfinasnet.com.br>>, acessado em 28/06/2016.
- Kalaivani, T., Allirani, A., Priya, P. (2011). “A Survey on Zigbee Based Wireless Sensor Networks in Agriculture”, 3rd International Conference on Trendz in Information Sciences & Computing (TISC2011), p 85-89.
- Keshtgari, Manijeh e Deljoo, A. (2012). A wireless sensor network solution for precision agriculture based on zigbee technology. *Wireless Sensor Network*.
- Khedo, Kavi Kumar, H. M. R. T. M. Z. (2014). PotatoSense: A wireless sensor network system for precision agriculture. *IST-Africa Conference Proceedings*.
- Kulkarni, Raghavendra V., F. A. e. K. G. V. (2011). Computational intelligence in wireless sensor networks: A survey. *IEEE Communications Surveys & Tutorials*, p. 68–96.

- Loureiro, AAF e Nogueira, J. (2003). Redes de sensores sem fio. *XXI Simpósio Brasileiro de Redes de Computadores*, p. 179–226.
- Mohanty, N.R, Patil, C.Y. (2013). “Wireless Sensor Networks Design for Greenhouse Automation”, *International Journal of Engineering and Innovative Technology (IJEIT)*, vol 3, p. 257-262.
- Oliveira, Luís M. e Rodrigues, J. J. (2011). Wireless Sensor Networks: a Survey on Environmental Monitoring. *Journal of Communications*, p. 143–151.
- Pawlowski, A., Guzman, J. L., Rodríguez, F., Berenguel, M., Sánchez, J. e Dormido, S. (2009). “Simulation of Greenhouse Climate Monitoring and Control with Wireless Sensor Network and Event-Based Control”, *Sensors*, vol 9, p. 232-252.
- Pilhas AA(2016), disponível em <https://pt.wikipedia.org/wiki/Pilha_AA>, acessado em 05/09/2016.
- RS-360SH (2016), disponível em <http://www.robotstorehk.com/motors/doc/rs_360sh.pdf>, acessado em 05/09/2016.
- Shivasaisomarathi, Akavaram Saispurthyreddy. (2010). Design of Greenhouse Control System Using Wireless Sensor Networks. *International Journal of Sciences: Basic and Applied Research (IJSBAR)*
- Silva, Marcel S. e Fruett, F. (2010). Rede de sensores sem fio de baixo custo para monitoramento ambiental. *XVIII Congresso Brasileiro de Automática*, p. 2351–2356.
- Vieira, Marcos, C. C. S. D. (2003). Survey on wireless sensor network devices. *ETFA*.
- Xbee Zigbee (2016), disponível em <<http://www.digi.com/products/xbee-rf-solutions/rf-modules/xbee-zigbee>>, acessado em 16/06/2016.
- XCTU (2016), disponível em <<http://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu>>, acessado em 16/06/2016
- Zigbee (2016), disponível em <<http://www.zigbee.org/zigbee-for-developers/network-specifications/zigbeepro/>>, acessado em 05/07/2016.
- Zigbee Alliance (2016), disponível em <<http://www.zigbee.org/>>, acessado em 28/06/2016.
- Zagade S.U., Kawitkar R.S. (2012), Wireless Sensor Network for Greenhouse. *International Journal of Science and Technology*.

Apêndices

Os apêndices A-E são referentes aos algoritmos, em linguagem C, utilizados para a configuração dos nós da RSSF desenvolvida nesta dissertação.

Apêndice A – Algoritmo na linguagem C para o nó sensor do ar.

```
void loop() { // executa o algoritmo em loop.
  capturaDados () { // Captura valores do sensor.
    DHT.read11(0); // Leitura do sensor de
    //temperatura/umidade do ar na porta analógica 0.
    temperatura = (int)DHT.temperature; // Conversão
    //para número inteiro.
    umidade = (int)DHT.humidity; // Conversão para
    //número inteiro.
  }
  enviarDados() { //Envia dados para porta Serial da
  //antena Xbee Zigbee.
    Serial.print(umidade); // envia valor da umidade
    //do ar.
    Serial.print(temperatura); // envia valor de
    //temperatura do ar.
  }
}
```

Algoritmo 1. Algoritmo em linguagem C para captura e envio de dados do nó sensor do ar.

Apêndice B – Algoritmo em linguagem C para o nó sensor do solo.

```
void loop() { //Executa algoritmo em loop.
  digitalWrite(12, 1); //Liga sensor conectado na porta digital 12.
  delay(1000); //Espera 1 segundo.
  umidade = analogRead(0); //Leitura de valores do sensor na
  porta analógica 0.
  if (umidade <= 100) {
    terra = 9;
  } else if ((umidade > 100) && (umidade <= 200)) {
    terra = 8;
  } else if ((umidade > 200) && (umidade <= 300)) {
    terra = 7;
  } else if ((umidade > 300) && (umidade <= 400)) {
    terra = 6;
  } else if ((umidade > 400) && (umidade <= 500)) {
    terra = 5;
  } else if ((umidade > 500) && (umidade <= 600)) {
    terra = 4;
  } else if ((umidade > 600) && (umidade <= 700)) {
    terra = 3;
  } else if ((umidade > 700) && (umidade <= 800)) {
    terra = 2;
  } else if ((umidade > 800) && (umidade <= 900)) {
    terra = 1;
  } else if (umidade > 900) {
    terra = 0;
  }
  Serial.println(terra); //Envia valor de terra e pula uma linha
  //para porta serial da antena.
  digitalWrite(12, 0); //Desliga sensor.
  delay(29000); //Espera 29 segundos
}
```

Algoritmo 2. Algoritmo em linguagem C para captura e envio de dados do nó sensor da terra.

Apêndice C – Algoritmo em linguagem C para o nó sensor de luz.

```
void loop() {
    ligar();
    capturaDados();
    enviarDados();
    hibernar();
}
void ligar(){
    digitalWrite(12,1);
    delay(900);
}
void capturaDados() {
    luz = analogRead(0);
    if (luz <= 100) {
        Luz = 9;
    } else if ((luz > 100) && (luz <= 200)) {
        Luz = 8;
    } else if ((luz > 200) && (luz <= 300)) {
        Luz = 7;
    } else if ((luz > 300) && (luz <= 400)) {
        Luz = 6;
    } else if ((luz > 400) && (luz <= 500)) {
        Luz = 5;
    } else if ((luz > 500) && (luz <= 600)) {
        Luz = 4;
    } else if ((luz > 600) && (luz <= 700)) {
        Luz = 3;
    } else if ((luz > 700) && (luz <= 800)) {
        Luz = 2;
    } else if ((luz > 800) && (luz <= 900)) {
        Luz = 1;
    } else if (luz > 900) {
        Luz = 0;
    }
}
void enviarDados() {
    analogWrite(10,0);
    mySerial.print(Luz);
    Serial.print(Luz);
}
void hibernar(){
    digitalWrite(12,0);
    analogWrite(10,168);
    delay(59000);
}
```

Algoritmo 3. Algoritmo em linguagem C para captura e envio de dados do nó sensor de luz.

Apêndice D - Algoritmo em linguagem C para o nó coordenador.

```
void loop()
{
  pular();
}
void pular() {
  if (Serial.available() >= 19) { //verifica se há 19
//ou mais bytes na porta serial.
    if (Serial.read() == 0x7E) { //compara se o
//byte lido é um identificador zigbee.
      for (int i = 0; i < 10; i++) {
        pula = Serial.read(); //descarta os próximos
//10 bytes
      }
      antena();
    }
}
void antena() {
  x = Serial.read(); //Leitura do byte de endereço
//do nó
  if (x == 0x04) { //nó sensor de temperatura e
//umidade do ar
    for (int i = 0; i < 3; i++) {
      byte pula = Serial.read(); //descarta os
//próximos 3 bytes.
    }
    ar = Serial.read(); //leitura do 1o byte de dados
//umidade do ar
    ar2 = Serial.read(); //2o byte de dados -
//umidade do ar
    temp = Serial.read(); //3o byte de dados -
//temperatura do ar
    temp2 = Serial.read(); //4o byte de dados -
//temperatura do ar
    ar = (ar - 48) * 10;
    ar2 = ar2 - 48;
    aux = ar + ar2;
    temp = (temp - 48) * 10;
    temp2 = temp2 - 48;
    aux2 = temp + temp2;
    Serial.print(",");
    Serial.print(aux);
    Serial.print(",");
    Serial.println(aux2);
  }
  if (x == 0xA8) { //nó sensor de luz
    for (int i = 0; i < 3; i++) {
      byte pula = Serial.read();
    }
    luz = Serial.read();
    byte pula = Serial.read();
    luz = luz - 48;
    Serial.print("DATA,TIME,");
    Serial.print(luz);
    Serial.println();
    sombrite();
  }
}
```

```
if (x == 0x03) { //nó sensor de umidade da terra
  for (int i = 0; i < 3; i++) {
    byte pula = Serial.read();
  }
  terra = Serial.read();
  terra = terra - 48;
  Serial.print("DATA,TIME,");
  Serial.print(",");
  Serial.print(",");
  Serial.print(",");
  Serial.println(terra);
  //Serial.print(",");
  if (terra <= 2) {
    ligabomba();
  }
} // fim do loop()
```

<pre> void ligabomba() { //envia "l" para atuador Serial.write(0x7E); Serial.write(0x00); Serial.write(0x0F); Serial.write(0x10); Serial.write(0x00); Serial.write(0x00); Serial.write(0x13); Serial.write(0xA2); Serial.write(0x00); Serial.write(0x40); Serial.write(0xD9); Serial.write(0xAA); Serial.write(0x60); Serial.write(0xFF); Serial.write(0xFE); Serial.write(0x00); Serial.write(0x00); Serial.write(0x6C); Serial.write(0xAE); } </pre>	<pre> void sombrite() { if (luz <= 4) { abresombrite(); } else if (luz == 9) { fechasombrite(); }} void abresombrite() { // envia "a" para o atuador Serial.write(0x7E); Serial.write(0x00); Serial.write(0x0F); Serial.write(0x10); Serial.write(0x00); Serial.write(0x00); Serial.write(0x13); Serial.write(0xA2); Serial.write(0x00); Serial.write(0x40); Serial.write(0xD9); Serial.write(0xAA); Serial.write(0x60); Serial.write(0xFF); Serial.write(0xFE); Serial.write(0x00); Serial.write(0x00); Serial.write(0x61); Serial.write(0xB9); Serial.println(); } </pre>	<pre> void fechasombrite() { //envia "f" para o atuador Serial.write(0x7E); Serial.write(0x00); Serial.write(0x0F); Serial.write(0x10); Serial.write(0x00); Serial.write(0x00); Serial.write(0x13); Serial.write(0xA2); Serial.write(0x00); Serial.write(0x40); Serial.write(0xD9); Serial.write(0xAA); Serial.write(0x60); Serial.write(0xFF); Serial.write(0xFE); Serial.write(0x00); Serial.write(0x00); Serial.write(0x66); Serial.write(0xB4); Serial.println(); } </pre>
--	--	--

Algoritmo 4. Programa do nó coordenador.

Apêndice E - Algoritmo em linguagem C para o nó atuador.

<pre>void loop() { pular(); } void pular() { if (Serial.available() >= 17) { if (Serial.read() == 0x7E) { for (int i = 0; i < 14; i++) { byte pula = Serial.read(); } atuar(); } } } void atuar() { valor = Serial.read(); byte pula = Serial.read(); if (valor == 0x61) { // byte lido é igual a "a" abresombrite(); } if (valor == 0x66) { // byte lido é igual a "f" fechasombrite(); } if (valor == 0x6C) { // byte lido é igual a "l" ligabomba(); } }</pre>	<pre>void abresombrite() { Serial.println("Abriu Sombrite"); if (i == 0) { //verifica se o sombrite está fechado. passo.step(1024) ; i++; } } void fechasombrite() { Serial.println("Fechou Sombrite"); if (i == 1) { //verifica se o sombrite está aberto passo.step(-1024) ; i--; } } void ligabomba() { analogWrite(3, 255); Serial.println("ligou a bomba"); delay(5000); //espera 5 segundos analogWrite(3, 0); }</pre>
---	--

Algoritmo 5. Algoritmo em linguagem C para o nó atuador.

Apêndice F – Artigo WCF 2016

XII WCF 2016 <xiiwcf2016@easychair.org>

📧 19 de ago ☆



para mim ▾

Prezado(a) Hugo Sampaio

Comunicamos que seu artigo Projeto e Implementação de um Sistema de Monitoramento de uma Estufa Agrícola de Larga Escala Utilizando Rede de Sensores Sem Fio submetido ao XII Workshop de Computação da FACCAMP foi ACEITO como artigo LONGO.

Algumas informações importantes:

1- Lembramos que o prazo de submissão da versão final é o dia 30 de agosto de 2016 com um MÁXIMO de OITO (8) PÁGINAS.

2.- Por favor, considere as sugestões dos revisores. Os comentários foram incluídos no final deste e-mail.

3. A versão final do artigo deve adequar-se ao padrão de trabalhos do WCF, em quantidade de páginas e formato, definidos na chamada de trabalhos em

<http://www.cc.faccamp.br/workshop.php?ano=2016>

Atenciosamente,

Comité de Organização.

Projeto e Implementação de um Sistema de Monitoramento de uma Estufa Agrícola de Larga Escala Utilizando Rede de Sensores Sem Fio

Hugo Vaz Sampaio, Shusaburo Motoyama

Faculdade Campo Limpo Paulista (FACCAMP)

Programa de Mestrado em Ciências da Computação

{hvazsampaio@gmail.com, shumotoyama@gmail.com}

Resumo. É proposto, neste artigo, um sistema de monitoramento de uma estufa agrícola de larga escala utilizando rede de sensores sem fio (RSSF). Para o monitoramento de uma estufa, os principais parâmetros de controle são temperatura e umidade do ar, umidade da terra, luminosidade, etc. Para a coleta de dados desses parâmetros, é proposta, neste trabalho, uma rede de sensores sem fio em uma forma hierárquica. Nessa configuração, os sensores, agregados com todas as funcionalidades de coleta, processamento e transmissão de dados sem fio, constituindo nós sensores, ficam no nível mais baixo. Nos níveis intermediários, utilizados para transportar os dados dos sensores para longas distâncias, ficam os nós denominados roteadores, e por fim no nível mais alto fica o nó coordenador, utilizado para enviar os dados a uma estação base, onde esses dados são processados. Os detalhes do projeto, assim como uma implementação simplificada desse sistema de monitoramento são apresentados. Os testes preliminares feitos em experimentos em laboratório mostram que o sistema está funcionando adequadamente.

Abstract. A large scale greenhouse monitoring system using wireless sensor network (WSN) is proposed in this paper. The main parameters required to monitor and control a greenhouse are air humidity and temperature, ground moisture and environment lightness. For the data gathering of these parameters a hierarchical WSN is proposed in this work. In this configuration the sensors, aggregated with all gathering functionalities, processing and wireless data transmission capabilities, denoted as sensor nodes, are on the lowest level. In the middle level, router nodes are used to transport data from sensor nodes to the coordinator node. In the highest level, the coordinator node is provided to send all received data to a central base, where data is stored and processed. The details of the project and a simple implementation of this monitoring system are presented. The preliminary tests carried out in laboratory experiments are showing that the system is operating properly.

1. Introdução

O avanço tecnológico de rede de sensores sem fio (RSSF) está proporcionando inovações em várias áreas de aplicação, tais como, médica, agrícola, controle de tráfego, controle ambiental entre outras. Na área de agricultura a RSSF pode ser usada para obter as informações necessárias que uma planta necessita para crescer, como temperatura e

umidade do ar, umidade da terra, luminosidade, entre outros. As variações dessas informações são relativamente lentas, assim, os sensores sem fio, alimentados por bateria, podem coletar e enviar essas informações esporadicamente, de tal modo que um sistema de monitoramento agrícola poderia mostrar esses dados a uma equipe de plantão ou automaticamente tomar providências como, por ex., regar, ligar sistema de ventilação, deixar o ambiente mais escuro, etc.

Um dos tipos de agricultura que está sendo cada vez mais utilizado devido a sua eficiência produtiva é a agricultura em estufas. As plantações em estufas buscam minimizar a variação do clima e seus efeitos adversos causados pelas altas ou baixas temperaturas do ar. Desta forma, é possível controlar a qualidade da colheita e minimizar o tempo de crescimento da planta. Com a utilização de uma RSSF dentro de uma estufa é possível coletar, com precisão, as informações climáticas, permitindo um controle otimizado da plantação.

Várias propostas e implementações de controle da estufa são apresentadas na literatura Pawlowski et al. (2009). A maioria desses artigos trata do projeto e teste de uma única estufa. Não há, entretanto, muitos artigos que tratam de uma estufa de larga escala, que controlam centenas de sensores. O objetivo deste artigo é propor um projeto de sistema de monitoramento para estufas de larga escala, utilizando RSSF. Além disso, tem como objetivo testar o funcionamento, através de uma implementação prática e o desenvolvimento de software da parte central do sistema proposto.

Este trabalho está organizado em 6 Seções. Na Seção 2 a seguir, são apresentados os principais artigos encontrados na literatura sobre monitoramento de plantações agrícolas com a utilização de RSSF. Os principais princípios que norteiam o projeto de uma estufa agrícola de larga escala e a sua configuração apropriada de RSSF, são discutidos na Seção 3. Na Seção 4, é apresentado o projeto de implementação prática simplificada de uma RSSF apropriada para estufas agrícolas. Na Seção 5, são apresentados os resultados de um experimento com a RSSF projetada. Finalmente, na Seção 6, as principais conclusões e trabalhos futuros são apresentados.

1. Trabalhos relacionados

Segundo Pawlowski et al. (2009) plantações em estufas possuem dois grupos de sistemas independentes com diferentes problemas. No 1º grupo a variação climática como temperatura/umidade do ar, luminosidade, concentração de CO₂, entre outros, afetam diretamente no crescimento das plantas. Dentro de uma estufa, a variação climática deve ser monitorada e controlada de acordo com a necessidade de cada planta. Os problemas do 2º grupo são variáveis de fertirrigação, ou irrigação com fertilizantes. Informações como temperatura/umidade da terra, composição da terra, entre outros, são utilizadas para decisões como volume e/ou periodicidade da fertirrigação. Foi desenvolvida uma técnica de controle para estufa baseada em eventos por meio de simulações com o aplicativo Matlab. Foram definidos limiares máximos e mínimos para as variáveis temperatura/umidade do ar, onde o valor acima ou abaixo do limiar é considerado um evento.

Um problema em RSSF é a vida útil de um nó sensor alimentado por bateria. Benavente (2010) propõe uma solução para economizar energia e prolongar a vida útil da

bateria em nós sensores. Utilizando intervalo de amostragem variável em nós sensores, foi desenvolvida uma RSSF para monitoramento ambiental em um vinhedo. Como exemplo de amostragem variável foram definidos limiares máximo e mínimo para o valor da temperatura coletado pelo sensor. O intervalo de coleta de dados será de 15 minutos, se estiver entre 10°C e 30°C. Caso a temperatura situe-se fora do intervalo, a coleta de dados será feita a cada 5 minutos. Foram desenvolvidos nós da rede com antenas Zigbee-PRO e sensores de temperatura e umidade relativa do ar, que capturam e transmitem dados para um dispositivo central.

Em Mohanty e Patil (2013), foram selecionados três parâmetros (luminosidade, temperatura e umidade do ar) que devem ser monitorados em uma estufa para garantir uma colheita mais produtiva. O artigo apresenta uma RSSF, utilizando antenas MICAz no padrão IEEE 802.15.4, e sensores para medir os parâmetros acima. Também são avaliadas a confiança e a habilidade da rede para detectar o microclima em uma estufa. Foi conduzido um experimento com duração total de quatro horas e coletas de dados realizadas com intervalo de 30 minutos. Os resultados indicam que os dados foram capturados e transmitidos sem erros, mostrando a robustez desta RSSF.

No artigo de Ahonen et al. (2008), foi desenvolvida uma RSSF com antenas Sensinodes para verificar a variação de microclima em diferentes alturas em uma estufa. Essas antenas utilizam protocolo 6LoWPAN, que habilita a transmissão de pacotes IPv6 comprimidos através de redes IEEE 802.15.4. Foram dispostos em diferentes alturas, dentro da estufa, quatro nós com sensores de luminosidade, temperatura e umidade do ar. Os dados foram capturados com intervalo de quatro minutos durante três horas. Os resultados indicam que a RSSF capturou com sucesso informações de variação de temperatura e umidade em diferentes alturas da estufa.

2. Projeto de uma RSSF para estufa agrícola de larga escala

As estufas agrícolas são utilizadas em plantações para fornecer proteção física de variações climáticas como o vento, chuva, granizo, neve, e amenizar variações bruscas de temperaturas e umidades. O excesso de calor e/ou CO₂ pode ser prejudicial para a plantação sendo os controles de temperatura e umidade do ar muito importantes. São importantes, também, controles de luminosidade e irrigação.

Além das medições das alterações climáticas, para o controle de uma estufa agrícola são necessários, também, dispositivos que alterem as condições ambientais como ventilador, ar condicionado, aquecedores, entre outros. Esses dispositivos serão chamados de atuadores, neste artigo, e podem ser acionados automaticamente para uma maior precisão no controle da estufa.

Outro aspecto das estufas é o tamanho, por exemplo, em Ahonen et al., os testes são realizados em uma seção de uma estufa de tamanho 18m x 80m. No cultivo em estufas da empresa Ervas Finas Horticultura (2016), localizada em Campo Limpo Paulista-SP, em uma região montanhosa, há 80 estufas, onde são plantadas ervas aromáticas, flores comestíveis, mini legumes e folhagens especiais. A estrutura de cada estufa é metálica, com dimensões de 6 metros de largura por 48 metros de comprimento, dividida em 16 seções de três metros de comprimento, totalizando 1280 seções.

Como pode se observar as dimensões das estufas podem ter variações razoáveis. Assim, para o projeto proposto, neste artigo, uma estrutura de larga escala será composta de um conjunto de estufas individuais, ou uma estufa de grandes dimensões será dividida em compartimentos individualizados, como mostrado na Fig. 1. Um nó sensor é definido, neste artigo, como dispositivo de rede formado por sensores, processador, memória, rádio e fonte de energia. Um nó sensor pode coletar diferentes tipos de dados como temperatura do ar, luminosidade, etc. Na Fig. 1 os nós sensores são distribuídos dentro de cada estufa, e coletam os mais variados dados.

A configuração de uma RSSF para uma estufa de larga escala proposta, neste trabalho, será organizada em uma forma hierárquica. Nessa estrutura hierárquica, os nós sensores de cada compartimento ou estufa ficam no nível mais baixo, e enviam dados para um nó intermediário, denominado de roteador (mostrado por símbolo R na Fig. 1), e daí para o nó coordenador que está no nível mais elevado, visto na Fig.1, fora da estufa. Na figura são mostrados somente três níveis de hierarquia, mas, dependendo da necessidade, para distâncias mais longas, o nível intermediário poderá ter mais estágios.

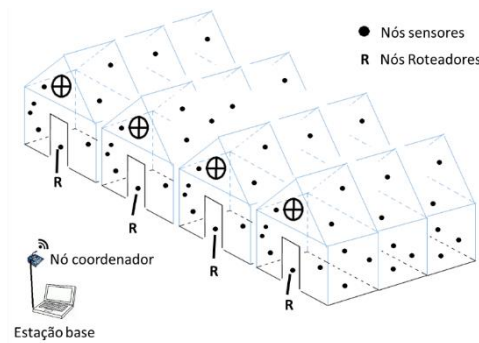


Figura 1. Exemplo de uma RSSF em estufa de larga escala

O nó coordenador se comunica com a estação base que armazena todos os dados recebidos. Os dados obtidos, pela estação base, podem ser apresentados na forma de tabela ou gráficos. A seguir, para testar a parte central do sistema de monitoramento da estufa agrícola foi feita uma implementação prática.

3. Implementação prática

Para a implementação prática do projeto proposto, uma configuração simplificada de uma RSSF foi utilizada, conforme mostrada na Fig. 2. Nesta implementação prática não foram utilizados atuadores.

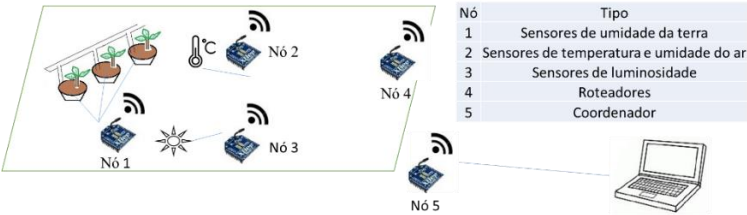


Figura 2. Rede simplificada para implementação prática.

Observa-se pela Fig. 2 que no Nó 1 está conectado com um sensor de umidade da terra, no nó 2 um sensor de umidade e temperatura do ar, no nó 3 um sensor de luminosidade e o nó 4 é um roteador intermediário. O Nó 5 é o Coordenador da rede, que receberá todas as informações e irá transmitir para o computador pela porta USB.

4.1 Projeto do Nó sensor

O projeto do nó sensor pode ser dividido em duas partes, uma parte relativa ao rádio, para a transmissão de dados sem fio, e a outra parte, um hardware que coleta os dados do sensor e envia ao rádio.

4.1.1 Projeto da parte do rádio

Para o rádio foi escolhido um componente que é compatível com a recomendação da Zigbee (2016). A Zigbee Alliance (2016), uma aliança entre dezenas de grandes empresas internacionais, foi fundada em 2002, com o objetivo de padronizar e continuamente desenvolver o Zigbee, padrão de rede sem fio de baixa velocidade e baixa utilização de energia.

O padrão de redes PAN de baixa transmissão IEEE 802.15.4 define a camada Física e de MAC utilizada pela Zigbee, com velocidade de transmissão de 250Kbps e endereçamento local de 16bits, ou de 0 a 65535 endereços, sendo o endereço 0 reservado para o coordenador.

As especificações da Zigbee se mostram adequadas para a sua utilização em estufa agrícola, principalmente pela sua baixa velocidade de transmissão e baixo consumo de energia, além de ser um componente fácil de encontrar no mercado.

O rádio utilizado, neste projeto, é o Xbee Zigbee (2016). O rádio Xbee Zigbee é fabricado pela empresa Digi seguindo os padrões Zigbee. Possui uma limitação de 14 nós filhos e alcance de sinal em 60 m para ambientes fechados. As antenas são configuradas utilizando o software XCTU (2016), fornecido no site da Digi. Através do XCTU, pode-se definir o modo de funcionamento das antenas (Coordenador, Roteador ou Nós sensores), PANID, endereço de destino entre outros.

4.1.2 Projeto da parte do hardware

Para esta parte foi utilizado o kit do Arduíno Uno (2016). O Arduíno Uno é uma placa com microcontrolador ATmega328P. Possui clock de 16 MHz, 32KB de memória flash, 14 portas digitais para input/output, 6 portas analógicas e porta USB. Para cada nó sensor foi inserido um algoritmo no microcontrolador para captura e envio de dados. A captura de dados dos sensores é feita através da leitura de uma ou mais portas analógicas.

4.2 Nós sensores desenvolvidos

Os três tipos diferentes de nós sensores são desenvolvidos: de temperatura e umidade do ar, de luminosidade e de umidade da terra.

No nó sensor de temperatura e umidade do ar é utilizado o sensor de temperatura e umidade do ar do tipo DHT11 (2016), conectado a Arduino Uno em uma porta analógica, e uma antena Xbee Zigbee. O sensor DHT11 possui uma biblioteca para Arduino, onde os valores capturados indicam a umidade do ar em porcentagem (%), e a temperatura do ar em Celsius (C°). A precisão da temperatura é de ± 2 graus, e a precisão da umidade do ar é de $\pm 5\%$.

O nó sensor de luminosidade possui um LDR (Resistor Dependente de Luz) conectado ao Arduino em uma porta analógica, e uma antena Xbee Zigbee. Os valores capturados na porta analógica são entre 0 e 1023. Foi dividido em dez escalas de aproximadamente 100 unidades representados por números de 0 a 9 onde um menor número indica menor incidência de luz.

O nó sensor de umidade da terra é composto por um sensor que detecta as variações de umidade no solo, conectado a uma porta analógica do Arduino Uno, indica valores entre 0 e 1023. Foi dividido em dez escalas de aproximadamente 100 unidades representados por números de 0 a 9 onde um menor número indica menor umidade.

Para os projetos do nó roteador e do nó coordenador, como esses nós possuem somente a parte de rádio, foram utilizados os rádios da Xbee Zigbee.

4. Experimento realizado.

Em sala de laboratório da FACCAMP (Faculdade Campo Limpo Paulista) uma RSSF foi configurada para testar o funcionamento de nós sensores de luminosidade, temperatura/umidade do ar e umidade da terra. Cada nó sensor capturou dados com intervalo de um segundo por um período de 5 minutos. Os dados capturados foram encaminhados para o nó coordenador, que por meio da porta USB foi gerado um log com o programa XCTU.

Foi desenvolvido um software para leitura do log gerado com visualização dos dados em tabela indicando o nó sensor de origem e os valores capturados. A Fig. 3a mostra a tela do software com colunas de data, hora, nó de origem, estado, umidade do ar em % e temperatura do ar em °C. Na Fig. 3b é mostrada uma foto da RSSF desenvolvida.

Durante a captura dos dados foram realizadas simulações de variação de iluminação, temperatura/umidade do ar e umidade da terra. Para o nó sensor de luminosidade foi bloqueada a luz e depois utilizada uma lanterna diretamente no sensor. Os valores abaixo de 7 indicam menor quantidade de luz capturada pelo sensor, e os valores acima de 7 indicam maior quantidade de luz, como mostrado na Fig. 4a.

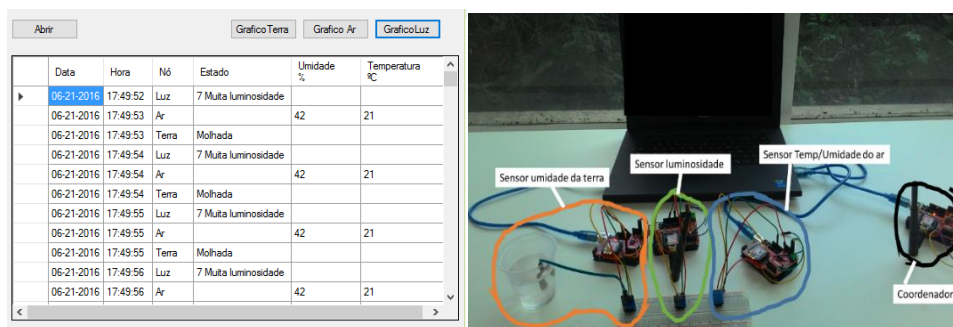


Figura 3. (a) Tela do software visualizador de log. (b) Foto da RSSF desenvolvida.

Foi utilizado um aquecedor de ar elétrico diretamente sobre o nó sensor de temperatura/umidade do ar. A Fig. 4b mostra a variação de temperatura/umidade do ar. O nó sensor de umidade da terra estava inicialmente com o sensor submerso em um copo de água para simular terra molhada. O sensor foi removido do copo para simular terra seca e o valor capturado, como mostrado na Fig. 4c

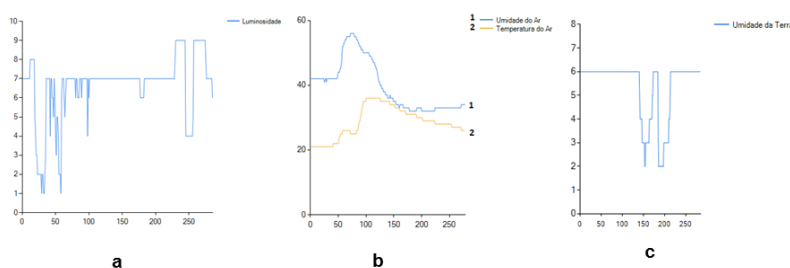


Figura 4. Gráficos resultantes do experimento em laboratório. (a) gráfico de variação de luminosidade. (b) gráfico de variação de temperatura/umidade do ar. (c) gráfico de variação de umidade da terra.

5. Conclusão e Trabalhos futuros.

O sistema de monitoramento de uma estufa agrícola de larga escala proposto, neste artigo, foi baseado em uma rede de sensores sem fio configurada em uma forma hierárquica. O projeto dessa configuração, assim como uma implementação simplificada foram detalhados. Os testes de um experimento em laboratório mostraram que a rede proposta funciona adequadamente, coletando os dados dos sensores e o software desenvolvido mostrou, em gráficos, as variações dos dados capturados.

Para futuros trabalhos, o sistema de monitoramento será desenvolvido com nós sensores e atuadores. Um experimento deverá ser realizado para mostrar realimentação na rede através de eventos, acionando o atuador de acordo com limiares definidas para os tipos de dados capturados. Esse experimento em laboratório também será realizado utilizando bateria para os nós sensores, dessa forma sendo possível calcular a vida útil de um nó sensor projetado. Para se ter resultados mais conclusivos do sistema projetado, são necessários testes em campo, em uma estufa real, que serão feitos, na medida do possível, na estufa da empresa Ervas Finas Horticultura.

Referências

- Kalaivani, T., Allirani, A., Priya, P. (2011). “A Survey on Zigbee Based Wireless Sensor Networks in Agriculture”, 3rd International Conference on Trendz in Information Sciences & Computing (TISC2011), p 85-89.
- Benavente, J.C.C. (2010). *Monitoramento ambiental de vinhedos utilizando uma rede de sensores sem fio que coleta dados com um intervalo de amostragem variável*. Dissertação de Mestrado.
- Pawlowski, A., Guzman, J. L., Rodríguez, F., Berenguel, M., Sánchez, J. e Dormido, S. (2009). “Simulation of Greenhouse Climate Monitoring and Control with Wireless Sensor Network and Event-Based Control”, Sensors, vol 9, p. 232-252.
- Mohanty, N.R, Patil, C.Y. (2013). “Wireless Sensor Networks Design for Greenhouse Automation”, International Journal of Engineering and Innovative Technology (IJEIT), vol 3, p. 257-262.
- Ahonen, T., Virrankoski, R., Elmusrati, M. (2008), “Greenhouse Monitoring with Wireless Sensor Network”, Mechatronic and Embedded Systems and Applications (MESA)
- Arduíno Uno (2016), disponível em <www.arduino.cc/en/Main/ArduinoBoardUno>, acessado em 16/06/2016.
- Xbee Zigbee (2016), disponível em <<http://www.digi.com/products/xbee-rf-solutions/rf-modules/xbee-zigbee>>, acessado em 16/06/2016.
- XCTU (2016), disponível em <<http://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu>>, acessado em 16/06/2016
- Ervas Finas Horticultura (2016), disponível em <<http://ervasfinasnet.com.br>>, acessado em 28/06/2016.
- Zigbee (2016), disponível em <<http://www.zigbee.org/zigbee-for-developers/network-specifications/zigbeepro/>>, acessado em 05/07/2016.
- Zigbee Alliance (2016), disponível em <<http://www.zigbee.org/>>, acessado em 28/06/2016.
- DHT11 (2016), disponível em <<http://www.micropik.com/PDF/dht11.pdf>>, acessado em 05/07/2016.

Apêndice G - Artigo 2 WCF 2016

----- Mensagem encaminhada -----

De: "XII WCF 2016" <xiiwcf2016@easychair.org>

Data: 19/08/2016 6:37 PM

Assunto: XII WCF 2016. Decisão sobre o artigo 10

Para: "Paulo Nietto" <pnietto@gmail.com>

Cc:

Prezado(a) Paulo Nietto

Comunicamos que seu artigo O Uso do Algoritmo de Agrupamento Hierárquico Divisivo DIANA em uma Rede de Sensores Sem Fio Aplicada à Agricultura submetido ao XII Workshop de Computação da FACCAMP foi ACEITO como artigo LONGO.

Algumas informações importantes:

1- Lembramos que o prazo de submissão da versão final é o dia 30 de agosto de 2016 com um MÁXIMO de OITO (8) PÁGINAS.

2.- Por favor, considere as sugestões dos revisores. Os comentários foram incluídos no final deste e-mail.

3. A versão final do artigo deve adequar-se ao padrão de trabalhos do WCF, em quantidade de páginas e formato, definidos na chamada de trabalhos em

<http://www.cc.faccamp.br/workshop.php?ano=2016>

O Uso do Algoritmo de Agrupamento Hierárquico Divisivo DIANA em uma Rede de Sensores Sem Fio Aplicada à Agricultura

Paulo R. Nietto, Hugo V. Sampaio

Faculdade Campo Limpo Paulista – FACCAMP
Campo Limpo Paulista – SP

{pnietto,hvazsampaio}@cc.faccamp.br

Abstract. *Wireless Sensor Networks (WSF) use sensor nodes to obtain information about the characteristics of the environment. The information collected helps on decisions to activate actuators such as, the irrigation frequency or control the air temperature inside a greenhouse. The data can also be used as input for divisive hierarchical clustering algorithms, and based on the data clustering, it is possible to derive conclusions. This article describes the main steps to achieve the proposed objectives.*

Resumo. *Uma rede de sensores sem fio (RSSF) usa nós sensores para obter informações sobre as características do domínio de sua aplicação. Tais informações auxiliam na decisão de atuadores, como por exemplo definir a frequência de irrigação ou controlar a temperatura do ar de uma estufa. Os dados captados por RSSF podem também servir de entrada a algoritmos de agrupamento hierárquicos divisivos para que, com base no agrupamento de tais dados, seja possível derivar conclusões sobre eles. O artigo descreve uma proposta de trabalho e os principais passos para atingir os objetivos propostos.*

1. Introdução

O avanço tecnológico de rede de sensores sem fio (RSSF) está proporcionando inovações em várias áreas de aplicação, tais como, médica, agrícola, controle de tráfego, controle ambiental entre outras. Na área de agricultura a RSSF pode ser usada para obter as informações necessárias que uma planta necessita para crescer, como temperatura e umidade do ar, umidade da terra, luminosidade, entre outros. As variações dessas informações são relativamente lentas, assim, os sensores sem fio, alimentados por bateria, podem coletar e enviar essas informações esporadicamente, de tal modo que um sistema de monitoramento agrícola poderia informar tais dados a uma equipe de plantão ou automaticamente tomar providências como, por ex., regar, ligar sistema de ventilação, deixar o ambiente mais escuro, etc.

Em várias taxonomias propostas na literatura, com o objetivo de organizar os algoritmos de agrupamento (ver Xu *et al.* (2007), Jain (2010), Theodoridis & Koutroumbas (2009)), uma categoria sempre presente é a dos chamados algoritmos

hierárquicos que, via de regra, produzem um conjunto de agrupamentos aninhados, organizados como uma árvore hierárquica, que pode ser visualizada como um dendrograma. Os algoritmos hierárquicos, por sua vez, se subdividem em dois grupos: (1)

hierárquicos aglomerativos, que se caracterizam por iniciar o processo com um agrupamento em que cada um dos dados é considerado um grupo do agrupamento e (2) hierárquicos divisivos, que se caracterizam por iniciar o processo com um agrupamento tendo apenas um grupo, aquele com todos os dados fornecidos e que, a cada passo, divide um dos grupos do agrupamento, até que cada grupo contenha um dado (ou então, existam apenas k grupos, em que k é fornecido como parâmetro), em uma abordagem caracterizada como *top-down*.

Este artigo descreve os passos já realizados relativos a um projeto de pesquisa voltado à investigação do uso do algoritmo de agrupamento hierárquico divisivo DIANA que tem por objetivo agrupar os dados captados por RSSF para que seja possível identificar similaridades e diferenças nos dados e, como consequência, seja possível derivar conclusões sobre eles. A Seção 2 contextualiza as RSSF. Na Seção 3 é apresentada uma descrição geral do algoritmo de agrupamento hierárquico divisivo DIANA para, então, apresentar seu pseudocódigo. A Seção 4 apresenta o experimento preliminar. Por fim, a Seção 5 descreve os próximos passos para a continuação e finalização do projeto de pesquisa.

2. Rede de Sensores Sem Fio

As redes de sensores sem fio (RSSF) são constituídas por um conjunto de dispositivos interligados denominados nós de rede. Um nó é composto por microprocessador, memória, rádio e fonte de energia. Uma RSSF possui um nó coordenador que cria e gerencia a RSSF e, nós sensores (um nó de rede com 1 ou mais sensores) para coleta de dados do domínio ao qual estão incorporados [Yick *et al.* 2008, Culler *et al.* 2004].

Este artigo apresenta uma RSSF desenvolvida com padrão Zigbee [Zigbee 2016], contendo um nó coordenador e três nós sensores. Cada nó utiliza uma antena Xbee Zigbee [Xbee Zigbee 2016], conectada a um Arduíno Uno [Arduino Uno 2016]. A RSSF foi desenvolvida em ambiente controlado para coletar dados ambientais de luminosidade, temperatura/umidade do ar e umidade da terra, através de nós conectados a sensores. Para cada nó sensor foi inserido um algoritmo no microcontrolador para captura e envio de dados. Cada sensor é conectado ao Arduino Uno através de uma porta analógica e a captura de dados é realizada através da leitura dessas portas.

O rádio Xbee Zigbee utilizado neste projeto possui uma limitação de 14 nós filhos e alcance de sinal em 60 metros para ambientes fechados. As antenas são configuradas utilizando o software XCTU [XCTU 2016], fornecido no site da Digi [Digi 2016]. Através do XCTU podemos definir o modo de funcionamento das antenas (Coordenador, Roteador ou *End Device*), PANID (*Personal Area Network Id*), endereço de destino, entre outros.

O Arduíno Uno, utilizado em cada nó, é constituído de uma placa com microcontrolador ATmega328P que possui clock de 16 MHz, 32KB de memória flash, 14 portas digitais para *input/output*, 6 portas analógicas e 1 porta USB. Uma antena Xbee Zigbee é conectada em cada Arduino Uno para transmissão dos dados captados.

Os sensores de temperatura e umidade do ar, do tipo DHT11 [DHT11 2016], possui uma biblioteca para Arduíno, onde os valores capturados indicam a umidade do ar em porcentagem (%), e a temperatura do ar em graus centígrados (C°). A precisão da temperatura é de ± 2 graus, e a precisão da umidade do ar é de $\pm 5\%$.

O nó sensor de luminosidade possui um LDR (*Light Dependent Resistor*), onde os valores capturados são entre 0 e 1023 e tais valores foram convertidos em uma escala de 0 a 9, onde 0 indica menor incidência de luz e 9 maior incidência de luz. O sensor de detecção de variações de umidade do solo captura valores entre 0 e 1023, onde tais valores foram convertidos em uma escala de 0 a 9, sendo 0 completamente molhado e 9 completamente seco.

Os nós sensores foram configurados para capturar os dados dos sensores com intervalo de um minuto por um período de 24 horas. Os dados capturados foram encaminhados para o nó coordenador, que transferiu as informações para um computador usando a porta Serial/USB. Cada nó sensor capturou 1440 dados, totalizando 5760 dados.

3. O Algoritmo DIANA

O algoritmo *Divisive ANALysis* (DIANA), é um algoritmo divisivo baseado na proposta de Macnaughton-Smith *et al.*, (1964), que possui a complexidade $O(N^2 \log N)$. Como descrito em Kaufman & Rousseeuw (1990), é uma proposta de um procedimento para algoritmos hierárquicos divisivos, em que algumas das funções empregadas podem ser customizadas, na dependência da aplicação considerada. Devido à possibilidade de customização, tal esquema pode dar origem a diferentes “instanciações”, muitas vezes consideradas novos algoritmos.

Segue uma descrição informal do algoritmo. No processo iterativo conduzido pelo DIANA, o t -ésimo agrupamento produzido tem t grupos em que $t = 1, \dots, N$, onde N é a quantidade de dados.

Considere que G seja um grupo já formado e que o objetivo seja dividi-lo em dois grupos de tal maneira que os dois grupos resultantes possuam a maior dissimilaridade possível entre eles. Inicialmente o algoritmo busca identificar um dado em G cuja dissimilaridade média com relação aos dados restantes seja máxima. O dado com dissimilaridade máxima é retirado de G e inserido em um novo grupo criado nesse momento, chamado tempG . Na sequência, para cada dado $x \in G$, é calculada a média dos valores de dissimilaridade de x , com todos os demais dados de G , i.e., média dos valores $\text{diss}(x,y)$, $y \in G$. De maneira análoga, é calculada a média dos valores de dissimilaridade de x com os dados pertencentes a tempG , ou seja, a média dos valores $\text{diss}(x,z)$ $z \notin G$ (i.e., $z \in \text{tempG}$).

Se para cada $x \in G$:

- $D(x) = (\text{média diss}(x,y), y \in G, x \neq y) - (\text{média diss}(x,z), z \in \text{tempG})$ for negativa, então tempG não receberá mais nenhum dado de G .
- Caso contrário, o dado $x \in G$ que produzir o valor máximo para a diferença $D(x) = (\text{média diss}(x,y), y \in G, x \neq y) - (\text{média diss}(x,z), z \in \text{tempG})$ é escolhido e retirado de G e, então, inserido em tempG .

O procedimento é repetido até que cada dado seja o único em um grupo ou, então até que um critério de parada seja satisfeito. O Algoritmo 3.1 descreve um pseudocódigo do algoritmo DIANA.

```

procedure DIANA (X, AGi)
Input: X = {P1, P2, ..., PN} %N dados a serem agrupados
Output: AGi

1. begin
2. t ← 1
3. AG1 ← {X} % agrupamento inicial
4. G ← X
5. repeat
6. maxDiss ← maiorDissimilaridade(G) % encontrar elemento mais dissimilar
7. G ← G - {maxDiss}
8. tempG ← {maxDiss}
9. repeat
10. Para cada dado x pertencente a G encontre:
11. D(x) = (média diss(x,y), x ∈ G e x≠y) - (média diss(x,z) z ∈ tempG)
12. Encontre o dado x para qual a diferença D(x) é a maior.
13. if D(x) > 0 then
14. begin
15. tempG ← tempG ∪ {x}
16. G ← (G - {x})
17. end
18. until todos valores D(x) ≤ 0
19. t ← t+1
20. Gi ← tempG
21. AGi ← (AGt-1 - {G}) ∪ {G, Gi}
22. G ← grupoMaiorDiametro(AGi) % encontrar grupo com maior diâmetro
23. until cada grupo contenha um único dado
24. return AGi
25. end procedure

```

Algoritmo 3.1. Pseudocódigo do algoritmo DIANA.

4. Experimento Preliminar

O experimento preliminar foi realizado utilizando a RSSF descrita na Seção 2 e o algoritmo DIANA, descrito na Seção 3. O algoritmo DIANA foi adotado nesse trabalho com objetivo de verificar sua eficácia em agrupar dados advindos do domínio da agricultura. O algoritmo K-Means é utilizado nesse experimento para comparação de resultados, por ser o algoritmo mais conhecido e utilizado para a tarefa de agrupamento de dados e, também, para apoio aos diversos outros algoritmos que possuem alto custo computacional. O conjunto de dados utilizado consiste de 1440 dados descritos por 4

atributos, onde cada atributo corresponde ao valor capturado por um dos sensores no mesmo minuto.

Os resultados obtidos foram avaliados utilizando dois índices de validação interna: o índice de Dunn [Dunn 1974], apresentado na Eq. (4.1) e o índice Silhouette [Rowsseew 1987], apresentado na Eq. (4.2). A Tabela 4.1 descreve a notação utilizada nos índices de validação. Índices de validação interna são métodos adequados para a avaliação quantitativa do resultado de um agrupamento, uma vez que tal avaliação do agrupamento resultante de um algoritmo utiliza somente quantidades e características inerentes ao conjunto de padrões.

Tabela 4.1 Notação utilizada nos índices de validação.

Notação	Significado
P_i	I-ésimo padrão
G_i	I-ésimo grupo
NG	Número de grupos
$d(P_i, P_j)$	Distância entre dois padrões distintos
$ G_i $	Quantidade de dados do i-ésimo grupo

$$Dunn = \min_{i=1, \dots, NG} \left\{ \min_{j=i+1, \dots, NG} \left(\frac{d(G_i, G_j)}{\max_{K=1, \dots, NG} \text{diam}(G_K)} \right) \right\}, \quad (4.1)$$

$$d(G_i, G_j) = \min_{P_i \in G_i, P_j \in G_j} \{d(P_i, P_j)\}; \quad \text{diam}(G_i) = \max_{P_i, P_j \in G_i} \{d(P_i, P_j)\}$$

$$Silhouette = \frac{1}{NG} \sum_i \left\{ \frac{1}{|G_i|} \sum_{P_i \in G_i} \frac{b(P_i) - a(P_i)}{\max[b(P_i), a(P_i)]} \right\}, \quad (4.2)$$

$$a(P_i) = \frac{1}{|G_i| - 1} \sum_{P_j \in G_i} d(P_i, P_j); \quad b(P_i) = \min_{j, j \neq i} \left[\frac{1}{|G_j|} \sum_{P_j \in G_j} d(P_i, P_j) \right]$$

Devido à versão do algoritmo K-Means utilizada ser sensível à escolha dos centroides iniciais (que são escolhidos de forma aleatória), o algoritmo K-Means é executado cinco vezes e, tanto a média dos cinco resultados quanto o melhor dos cinco resultados obtidos são utilizados para comparação.

4.1 Resultado do Agrupamento dos Dados

A Tabela 4.2 ilustra o resultado dos índices de validação Dunn e Silhouette nos agrupamentos resultantes dos algoritmos DIANA e K-Means com o valor do parâmetro de número de grupos 2 (*ie.*, $K=2$). Apesar do algoritmo DIANA não requerer o número de grupos como parâmetro, esse parâmetro pode ser utilizado como critério de parada.

Tabela 4.2 Resultados dos agrupamentos no conjunto de padrões captados por RSSF com K=2.

Algoritmo	Dunn	Silhouette
DIANA	0,58	0,51
K-Means (melhor execução)	0,08	0,59
K-Means (média das execuções)	0,08	0,59

O algoritmo K-Means induziu o mesmo agrupamento em suas 5 execuções. O índice de Dunn indica que o agrupamento induzido pelo algoritmo DIANA é superior ao induzido pelo algoritmo K-Means. Entretanto, segundo o índice Silhouette, o algoritmo K-Means apresenta performance superior ao algoritmo DIANA. Se considerar que o índice de Dunn apresentou um valor 0,5 superior para agrupamento resultante DIANA e o índice Silhouette apresentou um valor 0,08 acima para o agrupamento resultante do K-Means, pode-se concluir que nesse experimento, o algoritmo DIANA produziu um agrupamento mais representativo em relação ao agrupamento produzido pelo algoritmo K-Means.

5. Comentários Finais e Próximas Etapas

O trabalho até então desenvolvido, que envolveu testes de captura dos dados através da RSSF e o uso de tais dados como entrada ao algoritmo DIANA, continuará com o desenvolvimento e aplicação de uma RSSF na área de agricultura para que os dados captados pela RSSF sejam utilizados como entrada ao algoritmo DIANA e, através dos agrupamentos resultantes do algoritmo, seja possível identificar similaridades e diferenças nos dados e, como consequência, derivar conclusões.

Referências

- Arduíno Uno (2016). Disponível em <www.arduino.cc/en/Main/ArduinoBoardUno> [Acessado em 16/06/2016].
- Culler, D., Estrin, D. and Srivastava, M. (2004) “Overview of Sensor Networks”. *IEEE Computer Magazine*, v. 37, no. 8, pp. 41–48.
- DHT11 (2016). Disponível em <<http://www.micropik.com/PDF/dht11.pdf>> [Acessado em 05/07/2016].
- Digi (2016). Disponível em <<http://www.digi.com/>> [Acessado em 28/06/2016].
- Dunn, J. (1974) “Well Separated Clusters and Optimal Fuzzy Partitions”. *Journal of Cybernetics*, v. 4, pp. 95–104.
- Jain, A. K. (2010) “Data clustering: 50 years beyond K-means”. *Pattern Recognition Letters*, v. 31, pp. 651–666.
- Kaufman, L. & Rousseeuw, P.J. (1990) “Finding Groups in Data: An Introduction to Cluster Analysis”. USA: John Wiley & Sons, Inc.
- Rowsseew, P. (1987) “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis”. *Journal of Computational and Applied Mathematics*, v. 20, pp. 53–65

- Theodoridis, S. & Koutroumbas, K. (2009) “Pattern Recognition”, 4th edition. USA: Elsevier.
- Xbee Zigbee (2016). Disponível em <<http://www.digi.com/products/xbee-rf-solutions/rf-modules/xbee-zigbee>> [Acessado em 16/06/2016].
- Xu, H., Xu, D., Lin, E. (2007) “An applicable hierarchical clustering algorithm for content-based image retrieval”, In: Proc. of The International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications, MIRAGE’07, pp.82–92.
- Yick, J., Mukherjee, B. and Ghosal, D. (2008) “Wireless Sensor Network Survey”. *Computer Networks*, v.52, no. 12, pp. 2292–2330.
- Zigbee (2016). Disponível em < <http://www.zigbee.org/>> [Acessado em 28/06/2016].

Apêndice H – Artigo ICCCN 2017 – B1



waset.org

WORLD ACADEMY OF
SCIENCE, ENGINEERING
AND TECHNOLOGY

ACCEPTANCE LETTER

November 06, 2016

Master of BA/A/S Hugo Sampaio
FACCAMP
Brazil

Herewith, the international scientific committee is happy to inform you that the peer-reviewed draft paper code 17DK060085 entitled (Design and Implementation of a Greenhouse Monitoring System Using Hierarchical Wireless Sensor Network by Hugo Sampaio, Shusaburo Motoyama) has been accepted for oral presentation as well as inclusion in the conference proceedings of the ICCCN 2017 : 19th International Conference on Computer Communication Networks to be held in Copenhagen, Denmark during June, 11-12, 2017. The high-impact conference papers will also be considered for publication in the special journal issues at <http://waset.org/Publications>.

Conference Registration and Writing Formatted Paper:

1. Conference registration documents should be submitted to:
<http://waset.org/apply/2017/06/copenhagen/ICCCN?step=2>
2. Word Template File should be Downloaded at
<http://waset.org/downloads/template.docx>
3. Latex Style File should be Downloaded at <http://waset.org/downloads/latex.zip>
4. Copyright Transfer Statement Document should be Downloaded at
waset.org/publications/copyright?paperCode=17DK060085

Design and Implementation of a Greenhouse Monitoring System Using Hierarchical Wireless Sensor Network

Hugo Sampaio, Shusaburo Motoyama
Master Program in Computer Science, Faculdade Campo Limpo Paulista, FACCAMP
hvazsampaio@gmail.com, shumotoyama@gmail.com

Abstract—A greenhouse monitoring system using hierarchical wireless sensor network (WSN) is presented in this paper. The main parameters required to monitor and control a greenhouse are air humidity and temperature, ground moisture and environment lightness. For the data gathering of these parameters, a hierarchical WSN is presented in this work. In this configuration, the sensors, aggregated with all gathering functionalities, processing and wireless data transmission capabilities, denoted as sensor nodes, are in the lowest level. The router nodes are provided in the middle level to transmit data from sensor nodes to a controller named coordinator node. The coordinator node, in the highest level, is used to communicate with a central base, where all data received are analyzed. The details of the project and a simple implementation of this monitoring system are presented. The total number of sensor nodes possible to accommodate in the proposed monitoring system is estimated for two different cases. In first case, considering the total number of addresses available for identification of the sensor and router nodes, an analytical expression is derived for calculation of number of sensor nodes the system can use. In second case, the number of sensor nodes is estimated considering that traffic of sensor nodes has a distribution and modeling the proposed hierarchical WSN as a network of queues. Two different models of network of queues are analyzed. In first model, the feedback traffic from coordinator for activation or deactivation of the actuator nodes is not taken into account, and in second model, the feedback traffic from coordinator is considered. The analysis showed that depending on the packet rates of sensor nodes, a very large monitoring system can be designed.

Keywords— large scale, greenhouse, wireless sensor network, ZigBee.

I. INTRODUCTION

THE continuous advance in Wireless Sensor Networks (WSN) is making possible innovations in many different areas such as medical, traffic control, environment control, agriculture, among others.

In agriculture WSNs can be used to obtain the necessary information that a plant needs to grow such as air temperature and humidity, ground humidity and luminosity level. The variation of these parameters are relatively slow, so that data can be collected sporadically with wireless sensor nodes powered by batteries. The collected data is sent to a central node to be processed and stored. An agricultural monitoring system could show this data to an expert team or automatically take

measures such as irrigating the plants, activating a ventilation system or controlling a shading system.

One of the types of agriculture commonly used is planting in greenhouses. The greenhouses offer physical protection to plants against external agents, and allow climate control inside the greenhouse. The result is a more efficient production and easy to handle the crop.

Using a WSN inside a greenhouse, it is possible to collect, with precision, all climate information and control an automated greenhouse.

Many proposals and implementations to control a greenhouse are presented in literature [1]-[7]. Most of the papers show projects and tests in one greenhouse. However, there are few papers proposing WSNs for large scale greenhouses, where a WSN controls hundreds of sensor nodes. In [7], a WSN for large scale greenhouse is proposed based on clusters and cluster heads. In this paper, a WSN based on hierarchical structure is proposed. In this structure, the sensor nodes are divided into groups and in each group a router is provided. The sensor nodes are in lowest level of hierarchy and the routers are in the middle level and can have many stages. In the highest level a coordinator is provided.

This paper is divided into seven sections. In Section 2, main papers related to this work are presented. The design principles and the model of experimental prototype are discussed in Section 3. In Section 4, the details of the hardware implementation of nodes are shown. The tests and results of the experiments done with the developed WSN are presented in Section 5. In Section 6, the theoretical estimation of number of sensor nodes considering limitation of addresses and taking into account the traffic characteristic of sensor nodes and queuing theory is carried out. Finally, the conclusions and future work are presented in Section 7.

II. RELATED WORK

In [1], different applications using Zigbee WSN for precision farming is discussed in a survey. A large variety of parameters are identified for controlling such as air temperature and humidity, soil humidity and PH, wind speed and direction, identification of plants and detection of weeds.

In [2], the luminosity, soil humidity, air temperature and humidity parameters are selected to be monitored inside a greenhouse. A WSN is presented using MICAz antennas with

IEEE 802.15.4 standard and sensor nodes. A 4-hour experiment was conducted to collect data with interval of 30 minutes.

In [3] it is shown that planting in a greenhouse has two groups of independent systems with different problems. In the first group, the climate variables parameters such as air temperature and humidity, solar radiation, CO₂ concentration and luminosity are identified as main parameters. If the values of these parameters are too high or too low, the crop may be damaged. Inside a greenhouse it is possible to control the climate variables through the control variables. The control variables are used to alter the climate variables, using actuators that can turn on the heating system, activate the ventilation system etc. In the second group, the main parameters are the fertigation variables. These variables such as soil humidity, composition and pH, are used for decisions of volume and periodicity of the fertigation. External climate cannot be controlled, thus the greenhouse will provide protection to plants from wind, dust, snow or any abrupt variations that could damage the crop. Fig. 1 shows an example of a greenhouse with examples of control variables, external climate and climate variables as presented in [3].

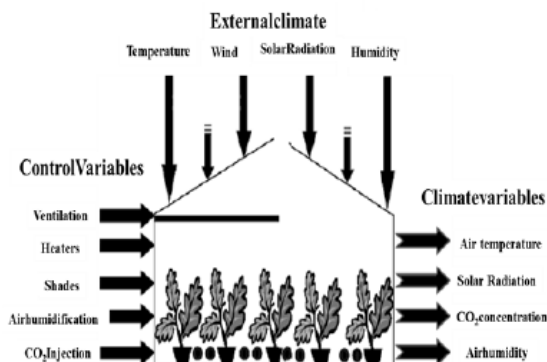


Fig. 1 Examples of greenhouse variables and parameters as shown in [3].

In [4], a WSN was developed with Sensinodes antennas, in order to verify the microclimate variation at different heights in a greenhouse. These antennas use 6LoWPAN protocol, which enables the transmission of compressed IPv6 data packages on a WPAN. 4 sensor nodes each with luminosity, air temperature and humidity sensors were used. An experiment was made for 3 hours with interval of collected data of four minutes.

In [5], a WSN was developed with two nodes with Sensirion sensors to monitor the variation of air temperature and humidity parameters of the Amazon forest. One node was on the ground level and the other node above the forest canopy. An experiment was made for 2 hours with interval of collected data of 10 seconds.

In [6], a WSN was developed to monitor the variation of air temperature and humidity inside of a greenhouse. The nodes used Tarang F4 antennas with Zigbee protocol. No experiment was reported in the paper.

In [7], a WSN monitoring system for large scale greenhouse using hierarchical structure is proposed. The proposed nodes are designed using CC2420 antennas, IEEE 802.15.4 compliant

RF transceivers, and powered by solar energy. The proposed structure is based on clusters. The sensor nodes are scattered in a cluster and a node is elected as cluster head through an algorithm. The data gathered from sensor nodes are sent to a cluster head which in cooperation with other cluster heads route the data to a base station and then to a monitoring center. The monitoring center will receive all data, store, analyze and decide if an actuator needs to be activated.

In this paper, a hierarchical monitoring system for greenhouse of any size is proposed. In the proposed system, the sensor nodes are divided into groups and each group has a router to gather the data from sensor nodes. The sensor nodes are in the lowest level of hierarchy and in the middle level, many router nodes, arranged also in a hierarchical configuration, can be used to transport data from sensor nodes to the coordinator node. In the highest level, the coordinator node is provided to send all received data to a central base, where data are stored and processed. This hierarchical structure is very convenient because it can be used the Zigbee capabilities. A network using Zigbee can be deployed in a hierarchical form.

III. DESIGN PRINCIPLES AND EXPERIMENTAL PROTOTYPE

The greenhouses are used to provide physical protection to plants against climate variations such as wind, rain, snow, or abrupt variations of temperatures. High temperatures and CO₂ concentration as well as low humidity could be harmful to plants, so controlling these variables is very important. Other important variables are luminosity intensity and soil irrigation.

In order to control a greenhouse some devices are used to alter the control variables, for example to turn on a fan, air conditioner, heater, shades, and CO₂ injections among others. These devices are called actuators nodes in this paper, and can be automatically activated for a higher precision of greenhouse control.

Another aspect of greenhouses is the size. The size of greenhouse used for tests in [2] is 18 m x 80 m. A firm named Ervas Finas Horticultura, located on a mountain region in Campo Limpo Paulista-SP, Brazil, uses 80 greenhouses for different plantations, such as spicy herbs, baby vegetables and edible flowers. Each greenhouses structure is metallic, with width of 6 meters and length of 48 meters. The greenhouse is divided into sections of 3 meters of length. Each greenhouse has 16 sections, totaling 1280 sections.

As it can be noticed the greenhouse dimensions can vary on size, and the use of a hierarchical WSN will allow scalability of the network as the number of sensor and actuator nodes grow. In this paper a large size greenhouse is considered a group of small greenhouses, or a large greenhouse divided into smaller sections as shown on Fig. 2.

A sensor node is defined, in this paper, as a device composed of sensors, processor, radio and power source. A sensor node can capture different types of data such as air temperature and humidity, luminosity, soil conditions and etc. In Fig. 2 the sensor nodes are distributed inside each greenhouse to collect different data.

In this paper, the WSN configuration for a greenhouse is organized hierarchically. In this hierarchical network the sensor nodes are at the lowest level, and sends data to an intermediate

node named router node. The router nodes will relay the data to a coordinator node located at the highest level, as can be seen outside of the greenhouse in Fig. 2. The figure shows only three levels of hierarchy, but more routers levels at intermediate stage can be added to increase area coverage as needed.

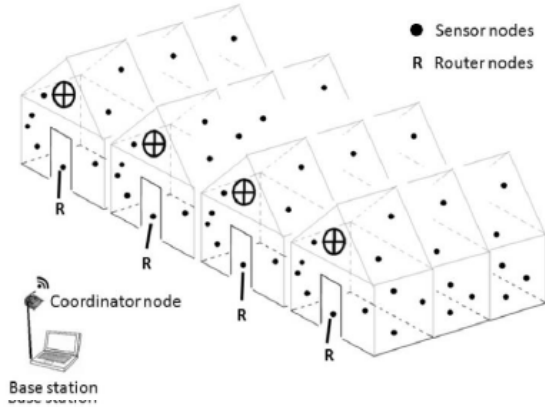


Fig 2. Proposed hierarchical WSN for greenhouse of any size.

A prototype of WSN is developed to demonstrate the main operation of proposed monitoring system. Fig. 3 shows the WSN developed. It has only one section with three sensor nodes, an actuator node, a router node and a coordinator node. The developed sensor nodes are luminosity, air temperature and humidity, and soil humidity. The actuator node has a step motor and a pump for irrigation. The router node is used to expand the network area coverage and numbers of sensor nodes. This configuration can be replicated for all sections, making it possible to monitor and control the entire greenhouse. All collected data is sent to a coordinator node then to a base station where data will be stored and analyzed.

To implement this prototype WSN all nodes are developed using Arduino Uno and Xbee Zigbee Antennas. Arduino Uno is a board with microprocessor ATmega328p that has 32 KB of flash memory and 16 MHz clock speed. The microprocessor can control up to 14 digital I/O pins and 6 analog pins on the board.

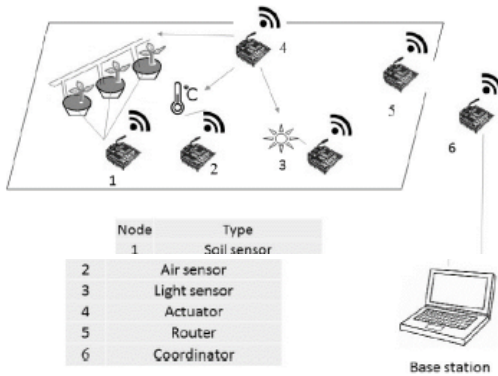


Fig 3. Example of WSN in one section of a greenhouse

The radio used in the nodes is an Xbee Zigbee antenna based on Zigbee with a Freescale microprocessor. The physical layer and media access control layer of Zigbee are based on IEEE 802.15.4 standard. Zigbee has low power, low transmission rate of 250 Kbps, and adds routing capabilities and security mechanisms.

A Zigbee network has three types of nodes. There must be only one Zigbee Coordinator (Z_C). The coordinator builds and maintains the network, distributing local 16 bit addresses, limited to 65535 possible nodes as the coordinator uses local address 0. This node must always be powered.

Zigbee Routers (Z_R) are used to expand the signal range and allow more nodes to connect to the network. Router nodes must be always on. Actuator nodes need to receive data from the coordinator so it must always be on.

Zigbee End Devices (Z_{ED}) are nodes that can enter on a sleep state while are not transmitting or receiving data. Sensor nodes will capture data, send it to the coordinator node, and then enter a sleep state.

IV. NODES DEVELOPMENT

The coordinator node is implemented using an Arduino Uno connected to an Xbee Zigbee antenna. The transmit (TX) port of the Arduino is connected to receive (RX) port of the antenna, and the TX port of the antenna is connected to the RX port of the Arduino. The antenna is powered by the Arduino with 3.3 volts (V), as seen in Fig. 4.

A program uploaded into the Arduino ATmega328P microprocessor reads continuously the serial port RX of the Arduino. This serial port shows incoming data received from the antenna. Each packet (frame) that the coordinator receives will be analyzed and processed and sent to the USB serial port of a computer.

In the computer a program named PLX-DAQ will read the USB serial port, defined as COM port, and store the data in an Excel sheet. This program is a macro that runs inside an Excel file, and limits the serial baud at 9600 b/s.

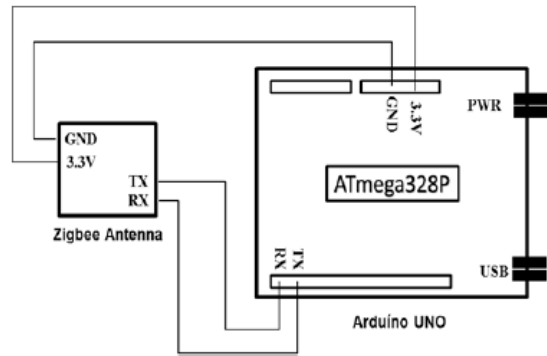


Fig. 4 Physical connections of the coordinator node.

A sensor is connected to the Arduino in an analog port in order to collect data, and is powered by a digital port. A program uploaded into the microprocessor collects data every 1 minute and send to the antenna. The antenna will forward a

packet with the data to the coordinator node. Fig. 5 shows the physical connections of a sensor node. Using different sensors, different sensor nodes can be implemented. For example, the air temperature and humidity sensor node, denominated air sensor node, uses a DHT11 air sensor. This sensor has a specific C library for Arduino. This library will convert the analog values captured of temperature to Celsius (C) scale, and values of humidity to percent (%) scale. The precision of this sensor is $\pm 2^\circ\text{C}$ and $\pm 5\%$ humidity.

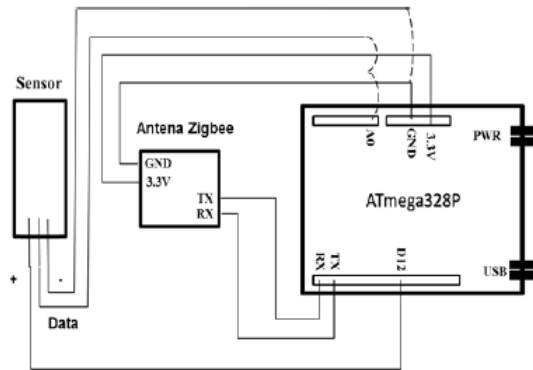


Fig. 5 Physical connections of a generic sensor node.

The physical connections of an actuator node are shown in Fig. 6. In this example, a water pump and a step motor are being controlled. The water pump is used in response to the soil sensor node that has a soil hygrometer sensor connected to an analog port. The microprocessor runs a program that reads the analog port and captures the sensor values from 0 to 1023. These values are converted to a scale from 0 to 9. A lower value means dry soil, and a higher value means wet soil. If the value indicates that the soil is dry, the coordinator node will send a packet to the actuator to turn on the water pump.

The used water pump motor is RS-360SH and requires 1.76A to run. This water pump works between 3V and 9V. Four 1.5V 2500mA batteries were connected in series providing 6V and 2500mA to the motor. The electric current that flows from the batteries to the water pump needs to be controlled. To do this a DC motor driver is used. This driver connects the batteries, the motor and to a digital PWM (*Pulse Width Modulation*) port of Arduino. The digital PWM port allows to control the output voltage between 0V and 5V.

The step motor is used in response to the luminosity sensor node, denominated light sensor node which has a Light Dependent Resistor (LDR) sensor. In this node the microprocessor runs a program that reads the analog port and captures the sensor values from 0 to 1023. These values are converted on a scale from 0 to 9, where 0 means dark. A value of 9 indicates high amount of light. Some plants prefer a shaded location to grow, so, if the luminosity level is high, the coordinator will send a message to the actuator node to activate a stepper motor to move a screen to create a shadow on the light sensor. The used 5V stepper motor 28BYJ-48 is connected to the Arduino through an ULN2003A driver. This driver allows

control of electric current up to 500mA enabling the stepper motor to run. The stepper motor requires 500mA to run.

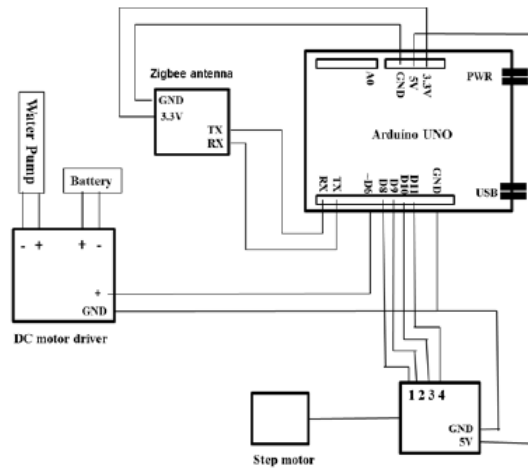


Fig 6 Physical connections of the actuator node

V. TESTS AND RESULTS

To test the developed prototype WSN an experiment is made. The prototype WSN is tested for 72 hours in an external controlled environment. Each sensor node is configured to capture and send data every 1 minute to the coordinator node. A total of 4320 messages are expected per sensor.

All data are gathered and stored in real time in an Excel file, using PLX-DAQ macro to read the COM port of the computer and display it at the screen.

The hygrometer is inserted in the soil of a plant pot to determine when the soil becomes dry. Initially the soil sensor captured value 2 meaning the soil was dry. Water is poured on the pot until it overflowed and drained through the holes under the pot. At this time the soil sensor captured value 6, indicating that the soil was completely wet. The low value for dry, and the high value for wet will vary from the composition of the soil.

Every one minute the soil sensor captured the data and sent to the coordinator. The coordinator is configured to send a message to the actuator node when the soil value received is under 3. The actuator receives this message and activates the water pump for 5 seconds.

Fig. 7 shows the soil humidity variation curves. All values captured above 6 are considered error values. The water pump was automatically activated 12 times on this experiment. Repetitive values of 2 or below indicates that the water pump was not activated when the soil was dry. These are due network errors where the actuator node didn't receive a message from the coordinator.

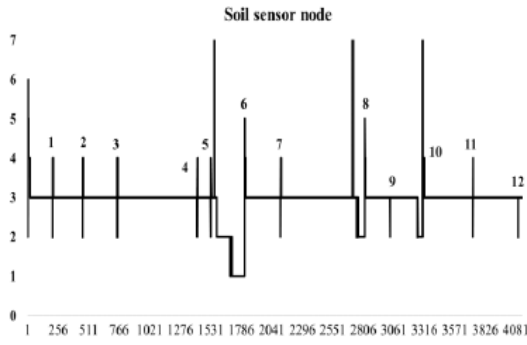


Fig. 7 Soil humidity variation curves.

Fig. 8 shows the curves of the air sensor node. Two lines draw the variation of air temperature and humidity. When the values of temperature rises, the values of humidity lowers. A few packet errors occurred around minutes 1100, 1900 and 2900.

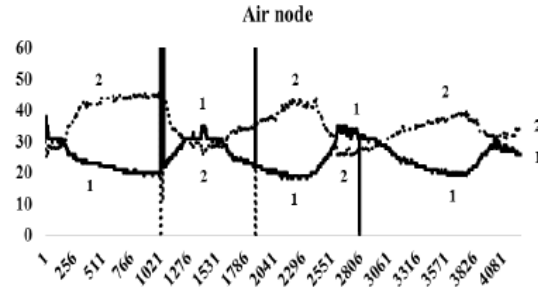


Fig. 8 Air temperature and humidity variation curves.

Fig. 9 shows the variation curves of the light sensor. The scale of curves is divided from 0 to 9, and the 0 value represents night time, and 9 daytime. All values above 9 that are out of scale are considered sensor errors. Most of light sensor errors are concentrated during night time. The errors have unknown source and requires further investigation with realtime monitoring of the WSN.

The coordinator node is configured to send a message to the actuator node if the luminosity level is equal to 9, to open the screen, or below 4 to close the screen. If the message is to close the screen, the actuator node will turn the stepper motor to the right and create shadow on the LDR sensor. If the message is to open the screen, the actuator node will turn the stepper motor to the left removing the screen from the sensor.

The LDR sensor used on this prototype turned out to be too sensitive, as luminosity values did not alter with or without the screen.

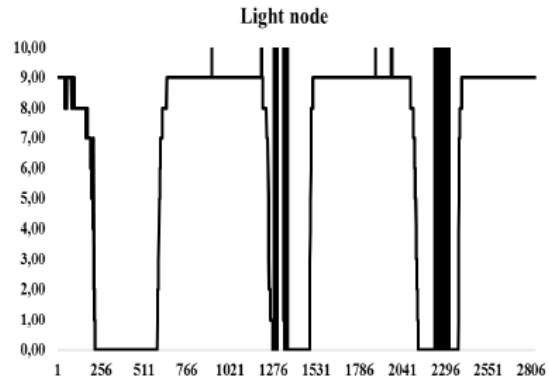


Fig. 9 Light variation curves.

Table 1 shows the total of packets the coordinator node received from each sensor node. The experiment was designed for each sensor node to capture and send data every minute for 72 hours, totaling 4320 packets per sensor node. The light sensor node had 34,44% packet loss, while the air and sensor nodes had 1,81% and 4,26% respectively. These network errors occurred with unknown source and did not alter the results of the experiment.

Table 1 Total packets received by the coordinator per sensor node

Sensor node	Packets received by the coordinator	Packet loss
Air	4242	1,81%
Soil	4136	4,26%

VI. THEORETICAL ESTIMATION OF NUMBER OF SENSOR NODES

A. Considering Limitation of Addresses

The hierarchical WSN adopted in this work has a coordinator at the highest level denominated zero node. The coordinator has a limit of N nodes directly connected to it. These nodes directly connected to the coordinator are in level 1. Each node at level 1 may have N children, and this repeats for each level. These levels correspond to the intermediate level and are used to route the packets to the coordinator. The nodes in this intermediate level are denoted routers. The nodes in the lowest levels are sensor nodes. Thus, the proposed WSN configuration has tree like structure.

The following assumptions are made. The Zigbee has a total number of Z_{MAX} addresses, the number of router nodes in intermediate level is R and each router node has N children. The number of router nodes directed connected to sensor nodes is R_s and it can also have N children. The total number of sensor nodes, at lowest level, is T_s . Thus, it can be written

$$Z_{Max} = T_s + R_s + R \quad (1)$$

The value of R_s is given by the total number of sensors nodes divided by the number of children nodes, N , in a router.

$$R_s = \frac{T_s}{N} \quad (2)$$

The value of R is given by the sum of routers in all levels above R_s . The first level above R_s is defined by the total number of routers in R_s level divided by the number of children nodes a router can have. It is assumed that all routers can accommodate up to N children nodes. Thus, it can be written

$$R = \frac{R_s}{N} + \frac{R_s}{N^2} + \frac{R_s}{N^3} \dots \quad (3)$$

Substituting (2) and (3) into (1), it obtains

$$Z_{Max} = T_s + \frac{T_s}{N} + \frac{T_s}{N^2} + \frac{T_s}{N^3} \dots = T_s \left(1 + \frac{1}{N} + \frac{1}{N^2} + \frac{1}{N^3} \dots\right) \quad (4)$$

The summation in the parenthesis is a geometric series, thus, it can be written as

$$Z_{Max} = \frac{T_s * N}{N-1} \quad (5)$$

Numerical example:

The Zigbee has maximum number of addresses of 65535 and the Xbee Zigbee antenna has a limit of $N=14$ children nodes. Applying these values in (5), it can find the maximum number of sensor nodes (T_{Smax}), and the total number of routers (R_{max}).

$$T_{Smax} = \frac{65535 * 13}{14} \cong 60854 \text{ sensor nodes}$$

$$R_{max} = Z_{Max} - T_{Smax} = 4681 \text{ router nodes}$$

B. Considering Network of Queues

The analysis made in the subsection A considered only the maximum number of addresses available. In this subsection, it is analyzed the number of sensor nodes possible to accommodate considering the traffic characteristics of sensor nodes and using queuing theory.

The hierarchical network adopted in this work, as shown in subsection A, the sensor nodes, routers and coordinator constitute a tree configuration. Considering that the routers and coordinator have buffers to store temporally the packets arriving in each of the nodes, the tree configuration can be modeled as network of queues.

Case 1. Without Feedback

The configuration of network of queues, in this case, will be as shown in Fig. 10. The figure shows that the sensor nodes are attached to the lowest level routers with buffers. In the intermediate stage, only a level of routers with buffers are shown and the coordinator with a buffer is in the highest level.

The following assumptions are made for the analysis. It is assumed only a level in the intermediate stage. The traffic generated by the sensor nodes are assumed to be Poisson

distribution. λ_s pcts/s is the partial traffic generated by the sensors nodes and sent to each router node of lowest level in which they are directly connected. λ_R pcts/s is the partial traffic generated by children router nodes directly connected to the router node of level 1. λ_C pcts/s is the traffic arriving to the coordinator generated by children router nodes of level 1. $1/\mu$ is packet length and C is the link capacity of Zigbee antenna. μ_C , μ_{R1} and μ_{R14} are the packet service rates of coordinator, router of level 1 and router of lowest level, respectively. The x index can assume a value of 1 up to the number of children nodes.

The network of queues of Fig. 10 can be analyzed using Burke theorem. The Burke theorem establishes that output of each queue in tandem with Poisson input and negative exponential service is also Poisson and can be considered as M/M/m independent queues.

To calculate the number of sensor nodes the system can accommodate, it is considered that processing limitation is located at Xbee Zigbee serial antenna, limited to $C = 115200$ bps.

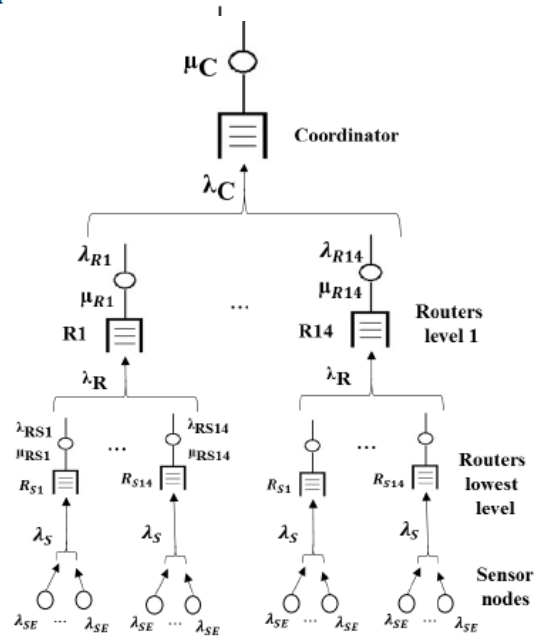


Fig. 10 Proposed hierarchical WSN.

The Xbee Zigbee coordinator (Z_C) can have $N=14$ children nodes. Each of 14 router nodes (R) connected to the coordinator has a queue (R_1 to R_{14}). Each router can have 14 children nodes and so on. The path formed by each router node of R and its children nodes is denoted as a branch in this paper.

The sensor nodes are set to send packets with average size of $1/\mu = 20$ bytes. So the service rate at output of coordinator will be $\mu_C = 115200/20 \times 8 = 720$ packets per second.

In the case of Fig. 10, an M/M/1 queue can be used in each of queues, thus the formula

$$E\{T\} = \frac{1}{\mu_C - \lambda} \quad (6)$$

can be applied. $E\{T\}$ is the average time spent in the queue (waiting time plus service time)

The load ρ is defined, in general, as ratio between incoming packet rate and output packet rate in this paper. Thus

$$\rho = \frac{\lambda}{\mu} \quad (7)$$

Fig. 11 shows a plot of the variation of the average time $E\{T_C\}$, a packet will stay at the coordinator, in milliseconds, while ρ_C varies from 0.1 to 0.9 with $\mu_C = 720$ pcts/s. It can be observed that $E\{T_C\}$ has a sharp rise after $\rho_C = 0.7$. To ensure the system stability, the value of ρ_C is limited to 0.5.

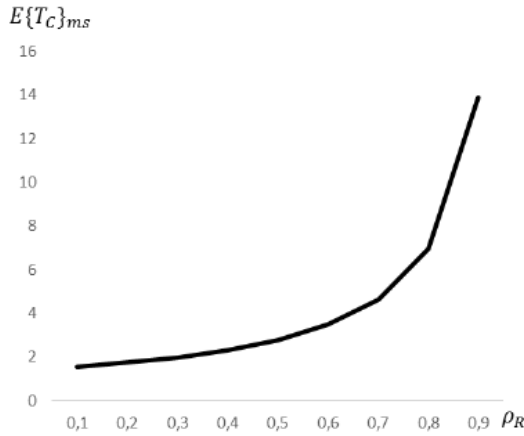


Fig 11. Variation of waiting time $E\{T_C\}$, in milliseconds, at the coordinator node.

With $\rho_C = 0.5$ and $\mu_C = 720$ pcts/s the value for λ_C will be 360 pcts/s. This packet rate is the total applied to the input of the coordinator. Since the coordinator can have up to $N = 14$ children nodes and assuming that the packets are coming with same distribution from router nodes of level 1, the output packet rate of each router node is

$$\lambda_{R1} = \frac{\lambda_C}{14} = \frac{360}{14} = 25.71 \text{ pcts/s}$$

This is also the output rate μ_{R1} of router R_1 . Assuming that the load of router R_1 , ρ_{R1} , is equal to 0.5, then λ_R will be 12.85 pcts/s. The average waiting time at the router R_1 , in this case, will be

$$E\{T_{R1}\} = \frac{1}{25.71 - 12.85} = 0.0778 \text{ s or } 77 \text{ ms}$$

Assuming the same reasoning for the router R_{S1} of lowest level, the following values are found.

$$\lambda_{RS1} = \frac{\lambda_R}{14} = \frac{12.85}{14} \approx 0.92 \text{ pcts/s}$$

$$\lambda_S = 0.5 \times \lambda_{RS1} = 0.46 \text{ pcts/s}$$

$$E\{T_{RS1}\} = \frac{1}{0.92 - 0.46} = 2.1 \text{ s}$$

Suppose that each sensor node generates an average packet rate of λ_{SE} . Finally, with λ_S divided by λ_{SE} , it is possible to estimate how many sensors can be accommodating in the monitoring system.

Tab. 2 shows the numbers of sensors nodes possible to accommodate for different packet rates of sensor node.

Table 2 Total number of sensor nodes possible to accommodate for different packet rates.

λ_{SE}	λ_S/λ_{SE}	Total sensor nodes on branch R1	TS
1 pcts/s	-	-	-
1 pcts/5s	2,3	32,2	451
1 pcts/10s	4,6	64,4	902
1 pcts/60s	27,6	386,4	5410
1 pcts/300s	138	1932	27048

The numbers of sensors nodes in Tab. 2 are found considering that the load ρ_C is 50%. Many others considerations can be made for the load to find others numbers of sensor nodes suitable for different greenhouse application.

Fig. 12 shows the different values of average time $E\{T\}$ in function of ρ_R considering ρ_C as parameters. By using the curves presented in Fig. 12 the most convenient values for $E\{T\}$, ρ_C and ρ_R can be chosen for different applications.

$E\{T\}$, ρ_C and ρ_R can be chosen for different applications.

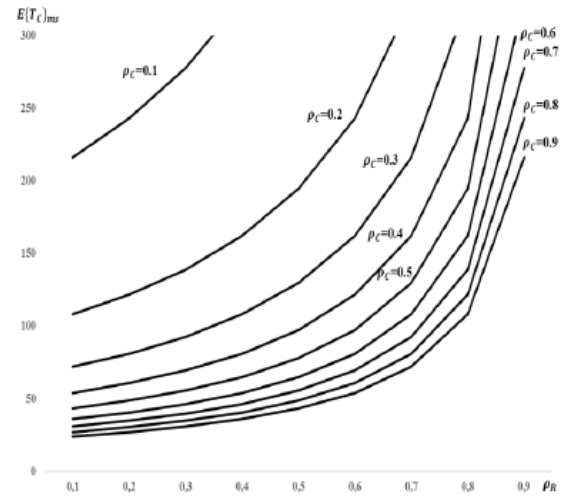


Fig. 12. Variation of ρ_R and ρ_C

In Fig. 13 the sum of waiting times of three levels, $E\{T\} = E\{T_C\} + E\{T_{R1}\} + E\{T_{RS1}\}$ in seconds, is shown. The value of ρ in the x axes is associated to same values of ρ_C, ρ_R and ρ_{RS} . For example, for $\rho = 0.5$, it means that the values of $\rho_C = \rho_R = \rho_{RS} = 0.5$, and the total waiting time $E\{T\} \approx 2.2$ s. The waiting times between $0.6 < \rho < 0.8$ seem to have lower values, but, as already seen, if $\rho > 0.6$ the system can become unstable.

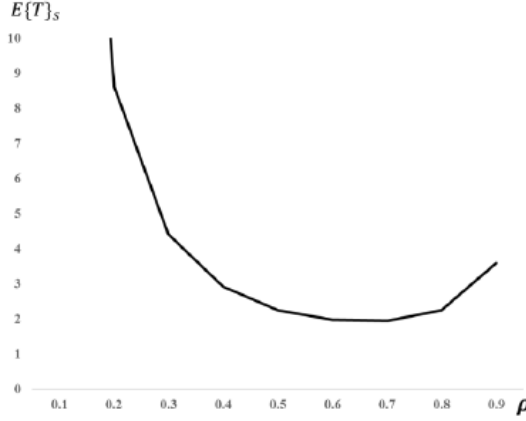


Fig. 13. Total $E\{T\}$ on all three levels

Case 2. With Feedback

In this case, it is assumed that the coordinator will send a part of output traffic back to sensor nodes for the activation or deactivation of actuators as shown in Fig. 14. The figure shows only a branch of network that will be used for analysis. The same analysis can be used for other branches of network of queues. In this case the Jackson theorem of network of queues can be applied. The Jackson theorem establishes that in a network of queues with feedback having negative exponential service, an external input obeying Poisson distribution and an output of network, each queue can be treated as M/M/1 queue. Thus, the same M/M/1 formula of Eq. 6 can be used, in this case, for each queue taking into account the traffic of feedback.

For numerical calculation, it is considered that only two percent of coordinator traffic is returned equally to the lowest level routers R_{Sx} . It is also considered that the feedback packets from the coordinator node will be relayed by the intermediate routers without needing for queuing. Thus, the feedback packet rate is $\lambda_F = 0.02\mu_C = 0.02 * 720 = 14.4 \text{pct/s}$.

It is assumed that each router R_{Sx} will receive equally the feedback traffic λ_F , from coordinator, thus, since the intermediate router can have up to $N = 14$ children and the lowest router can also have up to $N = 14$ children, the feedback traffic λ_{FS} at the input of R_{S1} will be

$$\lambda_{FS} = \frac{14.4}{14+14} \approx 0.07 \text{pct/s and}$$

$$\lambda_S = \lambda_{SE} + \lambda_{FS} \quad (8)$$

It is also assumed that one percent of output traffic of a router R_{Sx} is used for activation or deactivation of an actuator, as

shown in Fig. 14. Thus, this traffic will be $\lambda_A = 0.01\mu_{RS1} = 0.01 * 0.92 = 0.0092 \text{pct/s}$, and $\lambda_S = 0.5(\mu_{RS1} + \lambda_A) \approx 0.46 \text{pct/s}$.

λ_S is the total traffic, thus, by assuming an estimative of traffic for each sensor node λ_{SE} , it is possible to calculate how many sensor and actuator nodes this hierarchical WSN can accommodate.

Tab. 3 shows the numbers of sensors and actuator nodes, possible to accommodate, for different packet rates of sensor nodes and feedback traffic, and considering that the loads ρ_C and ρ_{RSx} are 50%.

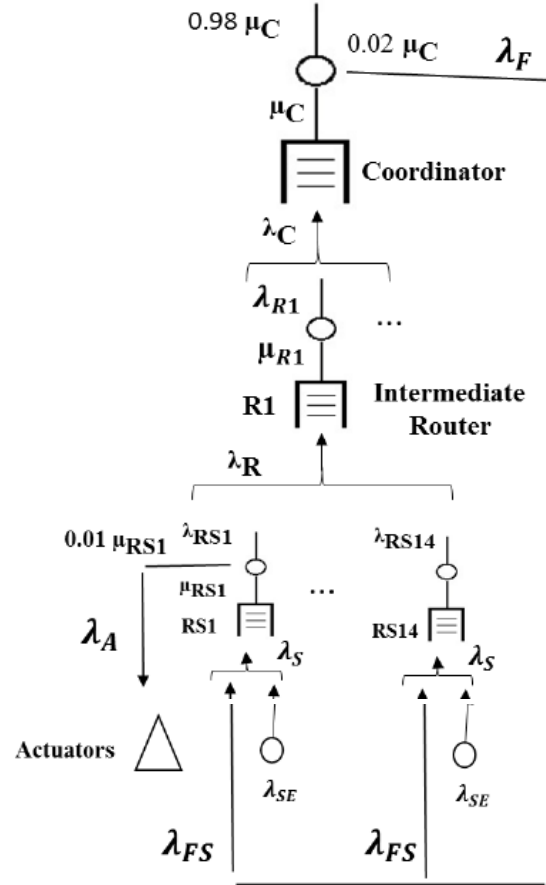


Fig 14. Example of hierarchical WSN with feedback in one branch.

In this example, it is not possible to have sensor nodes with $\lambda_{SE} = 1 \text{pct/s}$. The actuator nodes will receive, in average, 1 packet every 100 second. It is not possible to allocate actuators with $\lambda_A < 1 \text{pct/100s}$.

Table 3 Total number of sensor and actuator nodes possible to accommodate for different packet rates.

λ_{SE}	λ_S/λ_{SE}	Total sensor nodes on branch R1	TS	λ_A	Total actuator nodes on branch R1	Total Actuators
1 pct/s	-	-	-	-	-	-
1 pct/5s	1,95	27,3	382	-	-	-
1 pct/10s	3,9	54,6	764	-	-	-
1 pct/60s	23,4	327,6	4586	-	-	-
1 pct/100s	39	546	7644	1	14	196
1 pct/300s	117	1638	22932	3	42	588

VII. CONCLUSIONS AND FUTURE WORK

A hierarchical WSN monitoring system for greenhouse of any size was proposed in this paper. The design and implementation of proposed system were detailed and some experiments were carried out in laboratory. The results indicated that system is operating well in the tested conditions.

To estimate the number of sensor nodes the proposed system can accommodate, theoretical analyses were carried out considering two different situations. In first situation, the restriction of number of addresses provided by Zigbee was taking into account, showing that very large number of sensor nodes and routers can be accommodated. In the second situation, the number of sensor nodes possible to accommodate was estimated considering the traffic characteristics of sensor nodes and queuing theory. It was shown that proposed hierarchical WSN can be modeled as network of queues, and two cases were analyzed. In first case, the feedback traffic from coordinator for activating or deactivating the actuators was not considered. The results showed that depending on the packet rates of sensor nodes, very large number of sensor nodes can be used. In second case, the feedback traffic from coordinator was taken into account, showing that feedback traffic, in most of cases, does not affect the number of sensor nodes, but it can be significant if the feedback traffic rate is similar or greater than packet rate of sensor nodes.

In future work, the proposed hierarchical monitoring system will be tested in field experiments in real greenhouse. Furthermore, different models of sensor node traffic will be considered for estimation of number of sensor nodes.

REFERENCES

- [1] T. Ahonen, R. Virrankoski, M. Elmusrati, "Greenhouse Monitoring with Wireless Sensor Network", *Mechtronic and Embedded Systems and Applications (MESA)*, 2008.
- [2] M. S. Silva, F. Fruett, "Rede de sensores sem fio de baixo custo para monitoramento ambiental", *XVIII Congresso Brasileiro de Automática*, 2010, pp. 2351-2356.
- [3] N. R. Mohanty, C. Y. Patil, "Wireless Sensor Networks Design for Greenhouse Automation", *International Journal of Engineering and Innovative Technology (IJEIT)*, 2013, vol. 3, pp. 257-262.
- [4] A. Pawlowski, J. L. Guzman, F. Rodríguez, M. Berenguel, J. Sánchez, S. Dormido, "Simulation of Greenhouse Climate Monitoring and Control with Wireless Sensor Network and Event-Based Control", *Sensors*, 2009 vol. 9, pp. 232-252.
- [5] J. C. C. Benavente, "Monitoramento ambiental de vinhedos utilizando uma rede de sensores sem fio que coleta dados com um intervalo de amostragem variável", *Master's dissertation*, 2010.
- [6] T. Kalaivani, A. Allirani, P. Priya, "A Survey on Zigbee Based Wireless Sensor Networks in Agriculture", *3rd International Conference on Trends in Information Sciences & Computing (TISC)*, 2011, pp 85-89.
- [7] Shivasaisomarathi, A. Saispurthyreddy, "Design of Greenhouse Control System Using Wireless Sensor Networks", *International Journal of Sciences: Basic and Applied Research (IJSBAR)*, 2010.

Anexos

Anexo A – Fundamentação teórica de Teoria de Filas

A notação de processos de filas mais utilizada atualmente foi proposta por Kendall, em 1953. Representado por $X/Y/Z/W$ onde X e Y representam as distribuições do intervalo de tempo entre chegadas e do tempo de serviço, respectivamente. Os campos X e Y podem assumir os seguintes valores:

M – Distribuição exponencial negativa.

G – Distribuição geral.

D – Determinístico.

O campo Z representa o número de servidores em na fila e o campo W representa outras características do sistema, tais como comprimento da fila (*buffer*). Nesta dissertação todas as filas foram consideradas como $M/M/1$ com *buffer* infinito.

O tempo médio gasto em cada fila é $E\{T\}$, considerando o tempo de espera mais o tempo de serviço.

$$E\{T\} = \frac{1}{\mu - \lambda} \quad (9)$$

A carga ρ (rô) é definida nesta dissertação, em geral, como a razão entre a taxa de entrada de pacotes e a taxa de saída de pacotes, representado em (20):

$$\rho = \frac{\lambda}{\mu} \quad (20)$$

Em Teoria de filas são considerados dois tipos de Redes de Filas:

1 – Redes Fechadas

Em Redes Fechadas os pacotes não chegam à rede nem saem dela. Há um número fixo especificado de pacotes percorrendo a rede indefinidamente.

2 – Redes Abertas

Em Redes Abertas pacotes chegam à rede vindos de fontes externas, são atendidos e deixam a rede. As redes podem ser com ou sem realimentação.

a) Sem Realimentação:

Em redes sem realimentação pode-se utilizar o teorema de Burke. Este teorema estabelece que a saída de cada fila em tandem com entrada Poisson e serviço exponencial negativo é também Poisson, e pode ser considerado como filas independentes do tipo M/M/m. Em teoria de redes de filas a palavra Tandem designa duas filas em série, cadeia ou cascata, como na Fig. 48.

Numa fila em série com taxa de chegada λ e tempo de serviço exponencial $1/\mu$, cada nó da cadeia pode ser considerado independente dos demais e calcular o tempo de atraso para cada nó. O tempo de atraso total será a soma dos tempos de atraso de cada nó.

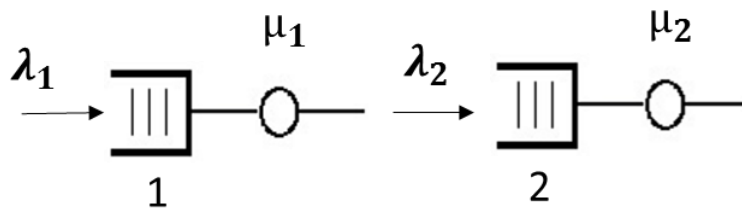


Figura 48. Exemplo de filas em série.

b) Com Realimentação:

O teorema de Jackson estabelece que em uma rede de filas com realimentação, também denominado *feedback*, tendo serviço exponencial negativo, um fluxo de pacotes de fonte externa obedecendo uma distribuição Poisson e saída da rede, cada fila pode ser tratada como uma fila M/M/m. Para o caso desta dissertação as filas foram tratadas como M/M/1. A taxa de *feedback* é representada por γ (gama) como na Fig. 49.

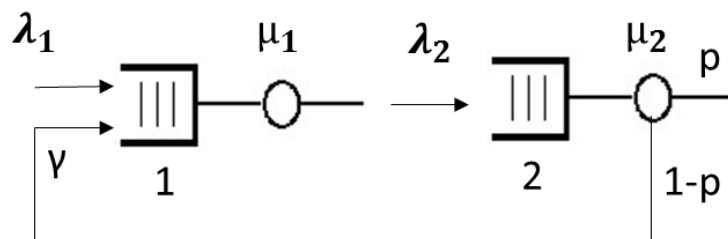


Figura 49. Exemplo de filas com realimentação.