

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**REQUISITOS BÁSICOS PARA A IMPLEMENTAÇÃO
DE UMA REDE DE SENSORES SEM FIO**

TRABALHO DE GRADUAÇÃO

Patrick Pivotto Viera

**Santa Maria, RS, Brasil
2008**

REQUISITOS BÁSICOS PARA A IMPLEMENTAÇÃO DE UMA REDE DE SENSORES SEM FIO

por

Patrick Pivotto Viera

Trabalho de Graduação apresentado ao Curso de Ciência da Computação –
Bacharelado, da Universidade Federal de Santa Maria (UFSM, RS), como
requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação

Orientadora: Prof^ª. Roseclea Duarte Medina

Trabalho de Graduação nº 254

Santa Maria, RS, Brasil

2008

**Universidade Federal de Santa Maria
Centro de Tecnologia
Curso de Ciência da Computação**

A Comissão Examinadora, abaixo assinada,
aprova o Trabalho de Graduação

**REQUISITOS BÁSICOS PARA A IMPLEMENTAÇÃO DE UMA REDE
DE SENSORES SEM FIO**

elaborado por
Patrick Pivotto Viera

como requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação.

Comissão Examinadora:

Roseclea Duarte Medina, Dr^a.
(Presidente/Orientadora)

Raul Ceretta Nunes, Dr. (UFSM)

Antonio Marcos de Oliveira Candia, B.el (UFSM)

Santa Maria, 31 de janeiro de 2008.

AGRADECIMENTOS

Agradeço à minha família pelo apoio e incentivo em todos os momentos de minha vida, principalmente nos mais difíceis, e por tudo o que sou hoje e poderei me tornar no futuro.

À professora Roseclea Duarte Medina, minha orientadora, por ter proposto este tema inovador, acreditando no meu potencial, e por ter se dedicado junto comigo para a realização deste trabalho, auxiliando-me nos momentos em que tive dúvidas, com sua experiência e conhecimentos.

Aos meus colegas, que durante estes quatro anos tornaram-se ótimos companheiros. Com certeza, a amizade construída não acabará ou diminuirá, mesmo com o tempo e a distância que nos separará.

A todos os meus amigos, que estiveram presentes em diversos momentos da minha vida e com os quais também aprendi muito.

RESUMO

Trabalho de Graduação
Curso de Ciência da Computação
Universidade Federal de Santa Maria

REQUISITOS BÁSICOS PARA A IMPLEMENTAÇÃO DE UMA REDE DE SENSORES SEM FIO

AUTOR: PATRICK PIVOTTO VIERA

ORIENTADORA: ROSECLEA DUARTE MEDINA

Data e Local da Defesa: Santa Maria, 31 de janeiro de 2008.

As redes de sensores sem fio surgiram com a finalidade de monitorar um ambiente ou objeto e realizar alguma operação com base nos dados obtidos ou reportá-los para serem tratados ou analisados. Essas redes consistem no uso de uma grande quantidade de dispositivos compactos e autônomos chamados nós sensores, os quais possuem a capacidade de se comunicar entre si. Elas são aplicadas em diversas situações do cotidiano, tais como aplicações médicas, militares, comerciais, ambientais e agrícolas. Neste contexto, este trabalho tem por objetivos definir o tipo de sensor e o sistema operacional a serem utilizados no projeto “Uso de telemetria para transmissão de dados de desempenho de máquinas agrícolas visando o gerenciamento”, que visa tratar os dados obtidos a partir de máquinas agrícolas equipadas por sensores, para monitorar a eficiência da máquina em trabalho. Para isto, é apresentada uma abrangente revisão da literatura sobre este tema recente e que está em expansão. Durante o trabalho, descrevem-se as etapas de definição do equipamento e do sistema operacional, as tecnologias encontradas e vantagens e desvantagens de cada item dentro do seu escopo.

Palavras-chave: rede de sensores sem fio; aplicações em RSSF; pilha de protocolos para RSSF; nós sensores; sistemas operacionais embutidos.

ABSTRACT

Undergraduation Final Work
Computer Science
Federal University of Santa Maria

BASIC REQUIREMENTS FOR IMPLEMENTING A WIRELESS SENSOR NETWORK

AUTHOR: PATRICK PIVOTTO VIERA
ADVISER: PROF. DR^a. ROSECLEA DUARTE MEDINA

Wireless sensor networks were created to track an environment or an object and to perform some operation based on the data or to report them to be treated or examined. These networks consist in the use of a large quantity of compact and autonomous devices called sensor nodes, which have the capacity to communicate among themselves. They are applied in many situations of daily life, such as medical, military, commercial, environmental and agricultural applications. In this context, the goal of this work is to define the type of sensor and operating system to be used in a project, which means to process the data obtained from farm machinery equipped by sensors, to track the efficiency of the machine while at work. To do this, a comprehensive review of literature on this recent and expanding subject is presented. This work also describes the stages of hardware and operating system definition, the technologies found and the advantages and disadvantages of each item within its scope.

Keywords: wireless sensor network; WSN applications; WSN protocols; sensor nodes; embedded operating systems;

LISTA DE FIGURAS

FIGURA 2.1 -	Esquema da constituição física de um nó sensor	14
FIGURA 2.2 -	Máquina agrícola equipada com sensores	15
FIGURA 3.1 -	Módulo MICA2 e sua arquitetura	32
FIGURA 3.2 -	Módulo MICAz e sua arquitetura	33
FIGURA 3.3 -	Módulo IRIS e diagrama	34
FIGURA 3.4 -	Módulo Imote2 e o diagrama da sua estrutura	35
FIGURA 3.5 -	Protótipo do WSN430	36
FIGURA 3.6 -	Plataforma XM2400	37
FIGURA 3.7 -	M2135 Mote e PM2130 <i>Manager</i>	38
FIGURA 3.8 -	BTnode rev3 e sua arquitetura	39
FIGURA 3.9 -	TinyNode 584 e sua placa de expansão	39
FIGURA 3.10 -	Plataforma TelosB e sua arquitetura	40
FIGURA 3.11 -	Etapas do desenvolvimento de uma aplicação no TinyOS	48
FIGURA 3.12 -	Instalação do TinyOS com o Cygwin Bash Shell	49

LISTA DE TABELAS

TABELA 3.1 -	Quadro comparativo entre equipamentos para redes de sensores	41
TABELA 3.2 -	Comparação entre as principais características dos sistemas operacionais	47

SUMÁRIO

1 INTRODUÇÃO	11
2 REVISÃO DE LITERATURA	14
2.1 Considerações iniciais sobre Redes de sensores sem fio	14
2.2 Exemplos de aplicações	15
2.2.1 Aplicações militares	15
2.2.2 Aplicações ambientais	15
2.2.3 Aplicações médicas	16
2.2.4 Aplicações comerciais	16
2.3.5 Aplicações agrícolas	17
2.3 Características e funcionamento das RSSF	18
2.4 Equipamentos de <i>hardware</i>	19
2.5 Pilha de protocolos e camadas	20
2.5.1 Camada física	20
2.5.2 Camada de enlace	21
2.5.3 Camada de rede	21
2.5.3.1 Protocolos de roteamento para redes planas	22
2.5.3.2 Protocolos de roteamento para redes hierárquicas	23
2.5.4 Camada de transporte	23
2.5.4.1 Protocolos de transporte	24
2.6 Padrões para RSSF	25
2.7 Sistemas Operacionais	26
2.7.1 Princípios de projeto de sistemas operacionais para sensores	26
2.7.2 Exemplos de sistemas operacionais em RSSF	28
2.8 Segurança em RSSF	29
2.8.1 Tipos de problemas e ataques	29
3. REQUISITOS BÁSICOS PARA RSSF	31
3.1 Definição do equipamento	31
3.1.1 Requisitos	31
3.1.2 Equipamentos pesquisados	31
3.1.2.1 MICA2	32
3.1.2.2 MICAz	33
3.1.2.3 IRIS	33
3.1.2.4 Imote2	34
3.1.2.5 WSN430	35
3.1.2.6 XM2400	36
3.1.2.7 M2135 Mote	37
3.1.2.8 BTnode	38

3.1.2.9 TinyNode	39
3.1.3 Equipamento escolhido: TELOSB	40
3.2 Definição do sistema operacional	41
3.2.1 Sistemas operacionais pesquisados	42
3.2.1.1 MANTIS	42
3.2.1.2 MagnetOS	43
3.2.1.3 SOS	43
3.2.1.4 YATOS	44
3.2.1.5 RETOS	45
3.2.2 Sistema operacional escolhido: TinyOS	45
4. CONCLUSÃO	50
REFERÊNCIAS BIBLIOGRÁFICAS	51

Capítulo 1

INTRODUÇÃO

Nas últimas décadas, as redes de computadores têm sofrido uma rápida e intensa evolução, devido em grande parte à popularização da Internet. Por consequência, surgiram as redes sem fio, que estão sendo cada vez mais estudadas e utilizadas na busca de novas soluções para a transmissão de dados entre dispositivos remotos. Loureiro et al (2003) explica que, aliado a isso, o avanço tecnológico no desenvolvimento de microprocessadores e da comunicação sem fio permitiu a criação de *chips* contendo vários sensores a serem empregados principalmente no sensoriamento remoto.

Com a finalidade de monitorar e/ou controlar um ambiente e reportar os dados obtidos sobre a área de interesse, foram criadas as redes de sensores sem fio (RSSF), segundo Aioffi (2007). Essas redes consistem em uma grande quantidade de dispositivos compactos e autônomos, chamados nós sensores, os quais possuem a capacidade de se comunicar entre si.

A posição de cada nó pode ser escolhida precisamente ou mesmo ser determinada de forma aleatória pela freqüente dificuldade de acesso aos locais onde os sensores serão colocados, que correspondem a uma região próxima ao fenômeno a ser analisado, e pela capacidade dos nós de se auto-configurarem. A comunicação entre estes nós é feita através de uma rede *ad-hoc* sem fio, um nó transmitindo ao próximo os valores do sensoriamento e este passando os dados ao nó seguinte, como descreve Rezende (2004).

As principais funções desempenhadas pelos nós sensores das RSSF são: determinar o valor de algum parâmetro em um determinado local, como a temperatura ou grau de umidade de um ambiente; detectar eventos e estimar valores; classificar um objeto detectado, por exemplo, identificar se certas células são malignas; e rastrear um objeto (Loureiro et al, 2003).

As redes de sensores são aplicadas em diversas situações do cotidiano, tais como em aplicações médicas para monitorar condições físicas e sinais vitais de pacientes hospitalizados; monitoração da qualidade do ar, analisando temperatura e presença de gases tóxicos em um edifício ou em um bairro de uma cidade; aplicações para o Exército, onde se acompanha, por exemplo, os movimentos de tropas inimigas; na agricultura; na monitoração industrial; em estudos ambientais em geral, como para a detecção de incêndios, entre outros (Cabrini, 2006; Ruiz et al, 2004; Braga, 2004).

A tecnologia das redes de sensores sem fio é relativamente recente e está em expansão. As primeiras pesquisas nesta área tratavam sobre onde elas poderiam ser aplicadas

e quais mecanismos seriam utilizados para reduzir o consumo de energia, principal preocupação devido à difícil manutenção dos sensores utilizados em algumas aplicações, como aponta Barbosa et al (2005). Eram preferidos *hardwares* que consumiam pouca energia. Quanto ao *software*, o enfoque principal foi dado aos protocolos para interconexão dos nós.

Atualmente, mais precisamente a partir de 2005, o objetivo foi consumir menos energia por instrução executada e, em decorrência disto, estão sendo estudadas novas formas de alimentação de energia dos nós sensores, que diferem das antigas baterias não-recarregáveis. Barbosa et al (2005) descreve que na parte do *software* discutem-se principalmente a padronização e a flexibilidade a fim de aumentar a portabilidade das plataformas dentre as aplicações existentes. Também são importantes as pesquisas nas áreas de segurança sobre os dados e da arquitetura de *software* e *middleware*.

A importância deste trabalho se justifica por fazer parte de um projeto maior, chamado “Uso de telemetria para transmissão de dados de desempenho de máquinas agrícolas visando o gerenciamento”, que almeja determinar um sistema de telemetria e tratar os dados obtidos a partir de máquinas agrícolas equipadas por sensores, para monitorar a eficiência da máquina em trabalho.

O presente trabalho teve por objetivos definir o tipo de sensor e o sistema operacional a serem utilizados no projeto citado.

Para escolher o sensor, foi realizada uma pesquisa sobre diferentes tipos de sensores e suas funcionalidades. Também foi feito um estudo comparativo entre os mesmos, a fim de definir o *hardware* mais propício. O passo seguinte seria propor um modelo e parâmetros de configuração que seriam necessários na definição das variáveis e das coletas iniciais dos dados trabalhados.

No aspecto de definição do sistema operacional, realizou-se uma pesquisa sobre os S.O.s utilizados em redes de sensores e, de acordo com o contexto no qual o TG se insere, foi escolhido o sistema que apresentou as melhores condições de uso. Após, foram executadas as etapas de instalação, configuração e validação do S.O. escolhido.

É importante salientar que tais medidas foram analisadas em conjunto, no sentido de verificar a melhor opção considerando as necessidades e limitações do projeto. Portanto, buscou-se a melhor associação de sistema operacional e equipamento de *hardware*, de modo que propiciem condições efetivas de implantação.

O capítulo 2 apresenta uma revisão bibliográfica sobre os principais conceitos utilizados no desenvolvimento deste trabalho. Descrevem-se, entre outros, aplicações,

fundamentos, arquiteturas, equipamentos, sistemas operacionais e características de funcionamento sobre as redes de sensores sem fio.

No terceiro capítulo, são citados os requisitos e necessidades para a definição do equipamento a ser utilizado na rede de sensores sem fio do projeto. Em seguida, são apresentados os principais módulos sensores disponíveis no mercado e suas características. Ainda é feita a descrição do módulo escolhido e a justificativa desta decisão.

No mesmo capítulo, a seção seguinte trata sobre a definição do sistema operacional que rodará sobre o equipamento escolhido, apresentado anteriormente. Serão descritos os demais S.O.s pesquisados e suas características e funcionalidades. Após, são feitas considerações sobre o S.O. escolhido, justificando a opção.

Por fim, são apresentadas as considerações finais e conclusões sobre o trabalho realizado.

Capítulo 2

REVISÃO DE LITERATURA

Este capítulo busca apresentar conceitos e características das redes de sensores sem fio relevantes na escolha de um modelo a ser implementado ou já existente.

2.1 Considerações iniciais sobre Redes de sensores sem fio

A RSSF, descrita por Macedo et al (2005), é um caso particular das redes *ad-hoc* sem fio, composta por unidades computacionais autônomas chamadas nós sensores. Estes nós são pequenos dispositivos constituídos de unidade de processamento (processador), memória estática e dinâmica, interface analógica e digital, bateria, módulo de rádio (transceptor) e sensores de diversos tipos, como apresenta Cabrini (2006). A estrutura do nó sensor descrito pode ser visualizada a partir da Figura 2.1 São capazes de medir diversas variáveis, como temperatura, luminosidade, sons, umidade e movimentos de qualquer natureza.

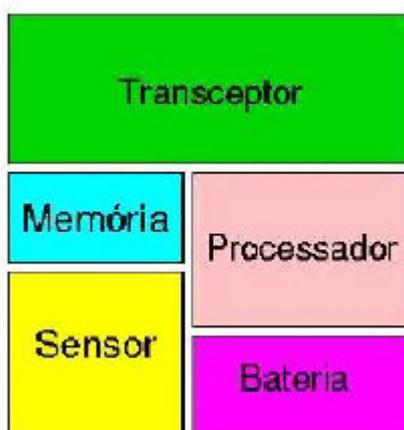


Figura 2.1 – Esquema da constituição física de um nó sensor – Fonte: Loureiro et al (2003, p. 10).

Os nós individualmente possuem pouca capacidade computacional e de energia, mas um esforço colaborativo entre os mesmos permite a realização de uma grande tarefa. Ruiz et al (2004) afirma que os sensores podem ser lançados sobre áreas remotas e, sem a intervenção humana, formar uma rede sem fio que coleta dados sobre o fenômeno de interesse, realiza processamento local e dissemina as informações para um ponto de acesso, como uma estação base.

Através de ondas de rádio, os dados são transmitidos pelos nós, que podem ser empregados desde poucas unidades até milhares numa mesma rede. Essa quantidade, assim como a arquitetura da rede, os tipos de serviços e de dados e os protocolos de comunicação a serem utilizados, dependem diretamente da aplicação a qual se destina.

2.2 Exemplos de aplicações

As redes de sensores sem fio possuem condições de dar suporte a uma extensa gama de aplicações úteis e, gradativamente, estão sendo cada vez mais utilizadas em várias situações do cotidiano com propósitos diversos. Esta seção apresenta exemplos de aplicações onde tais redes são envolvidas.

2.2.1 Aplicações militares

Redes de sensores sem fio podem formar sistemas de inteligência, comunicação, controle e rastreamento, de modo que auxiliem em tarefas militares, como monitoramento de tropas aliadas e forças inimigas, detecção de possíveis ataques químicos, biológicos e nucleares, monitoramento de munição e equipamentos (Stojmenovic, 2005).

Sensores acoplados em viaturas, armas e até nos próprios soldados captam informações relevantes para determinar o melhor conjunto de decisões a serem tomadas. Stojmenovic (2005, p. 26) diz que “Nodos sensores podem ser programados para enviar notificações sempre que um movimento em uma região particular é detectado”. Contudo, os sensores também podem ser programados para que permaneçam totalmente passivos até que um fenômeno particular aconteça.

2.2.2 Aplicações ambientais

Redes de sensores sem fio podem ser usadas para controlar um determinado ambiente, seja ele terrestre, aéreo ou marítimo. Apresenta como vantagens uma longa vida útil, a proximidade do objeto a ser observado, sem afetarem o ambiente em que a rede está inserida, desde que os sensores sejam pequenos o suficiente para se tornarem imperceptíveis, de acordo com Karl (2005).

Algumas dessas aplicações incluem monitoramento de habitat, rastreamento de animais, detecção de queimadas, inundações, abalos sísmicos e presença de poluentes químicos. Um exemplo mais específico, citado por Karl (2005), é o uso de redes de sensores sem fio na observação do crescimento de plantas em perigo. Neste caso, os dados coletados pelos sensores ajudam a determinar os fatores que suportam o crescimento de tais plantas.

2.2.3 Aplicações médicas

Neste tipo de aplicação, nós sensores podem ser implantados em pacientes de hospitais com a finalidade de monitorar sua localização e os respectivos sinais vitais. Mesmo com a liberdade de se deslocar livremente, o paciente continuará sendo supervisionado constantemente. No caso de algum acidente, os médicos e a equipe de profissionais responsáveis serão alertados imediatamente, desde que disponham de um sensor correspondente, conforme aponta Stojmenovic (2005).

As RSSFs assumem grande importância em situações que se necessitam cuidados intensivos devido a uma operação de risco e também podem ser usadas na assistência a idosos e na administração de medicamentos dados a um paciente, observando-se suas informações fisiológicas em tempo real.

2.2.4 Aplicações comerciais

Detecção e rastreamento de veículos, supervisão do fluxo de tráfego, controle no interior de prédios (industriais, comerciais e habitacionais) são algumas das possibilidades com o uso de redes de sensores sem fio (Sohraby et al, 2007).

A utilização de pequenos dispositivos com capacidade de processamento e comunicação é uma solução ao alto custo das câmeras que capturam imagens nas ruas das grandes cidades ou de rodovias. Dessa forma, pode-se obter o número de veículos que trafegam por uma via e estimar a velocidade deles, além de prever possíveis pontos de congestionamento (Rossetto, 2007).

Uma maneira de se reduzir o consumo de energia em prédios e aumentar o conforto dos moradores é a utilização de redes de sensores para controlar em tempo real variáveis como temperatura, ventilação e iluminação. Em regiões propensas a abalos sísmicos, é possível determinar a intensidade do fenômeno físico e o nível de segurança após a ocorrência de um evento dessa natureza (Karl, 2005). Essas são algumas características presentes nos

prédios inteligentes, além de outras, como detecção de incêndios e sistema inteligente de segurança, com dados sobre a movimentação de pessoas.

2.2.5 Aplicações agrícolas

RSSFs podem ser usadas na agricultura de precisão para controle automático da irrigação de uma plantação, por exemplo. Assim, os nós sensores seriam implantados próximos às raízes das plantas, monitorando a taxa de umidade do solo e acionando os dispositivos de irrigação quando necessário, conforme descreve Rossetto (2007).

Pode-se estimar mais precisamente o nível de fertilidade e o tipo do solo a ser cultivado e realizar um controle mais efetivo contra pestes ou pragas que possam prejudicar a plantação (Karl, 2005).

Além disso, a pecuária também pode ser beneficiada pelas redes de sensores. É possível verificar a localização dos animais e a necessidade de tratamento contra infecções e parasitas, por meio de sensores instalados em cada animal.

Outra aplicação em que é possível usar uma rede de sensores é na monitoração do funcionamento de máquinas agrícolas, escopo e objetivo do projeto ao qual este trabalho está vinculado. Pela Figura 2.2, pode-se ter uma noção de um modelo que poderia ser implantado.

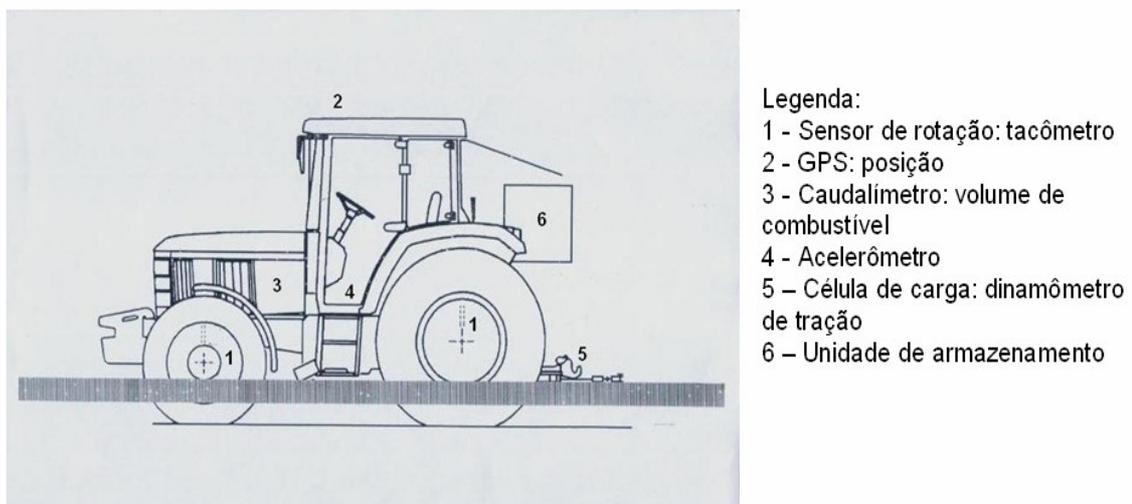


Figura 2.2 – Máquina agrícola equipada com sensores

2.3 Características e funcionamento das RSSF

Nesta seção, discutir-se-ão os princípios de funcionamento e algumas características particulares das RSSFs.

A escalabilidade deve ser prevista pelas arquiteturas e protocolos utilizados, visto que uma grande quantidade de sensores é empregada em uma rede. Em aplicações de monitoramento ambiental de florestas, por exemplo, a rede é composta por dezenas de milhares de nós.

Algumas redes necessitam que os sensores sejam endereçados unicamente, como os que são colocados no corpo humano ou em determinado animal ou equipamento, visto que nesses casos a localização do dado coletado é um fator determinante. Já, as que monitoram um fenômeno externo não necessitam da identificação de cada sensor, pois normalmente o que interessa é apenas o valor de uma determinada variável nessa região, como descreve Loureiro et al (2003).

A mobilidade dos sensores diz respeito a se estes podem se locomover em relação ao sistema dos quais coletam os dados. Sensores colocados na superfície de um oceano com a finalidade de medir o nível de poluição da água são móveis. Um exemplo de sensores considerados estáticos são os utilizados para medir sinais vitais de um paciente (Loureiro et al, 2003).

As RSSF devem ser tolerantes a alguns tipos de falhas, devidos a problemas que têm grande probabilidade de ocorrerem. É muito comum um sensor esgotar sua energia ou parar de funcionar corretamente, como também pode haver falhas na comunicação entre dois ou mais nós (Karl, 2005). Portanto, devem ser utilizados mecanismos de redundância que tratem esses problemas (Rossetto, 2007).

Uma das características mais importantes é a auto-organização da rede. Isso significa que sensores podem tornar-se ativos ou inativos automaticamente de acordo com a necessidade. Por exemplo, sensores se omitirem quando já há outros atuando na mesma região de cobertura, para economizar energia. Ou a entrada de novos sensores em uma rede após certo tempo ou quando for notado que algum sensor está danificado (Loureiro et al, 2003).

O tempo de vida de uma rede de sensores está ligado à qualidade de serviço e à energia disponível. Um exemplo de como aumentar o tempo de vida é a troca de fontes de energia tradicionais (baterias) por sensores que se utilizam da energia solar, embora essa fonte seja limitada. Em condições ideais, o tempo de vida dessa rede seria infinito, segundo Karl (2005). Por outro lado, quando se necessita de uma maior qualidade de serviço, mais energia é

consumida pelos nós e isso diminui o tempo útil da rede. Inerentemente, o tempo de vida de uma rede de sensores sem fio depende do tipo da aplicação para a qual ela foi projetada.

2.4 Equipamentos de *hardware*

Nesta seção, serão apresentadas algumas plataformas que vêm sendo adotadas na construção de RSSFs, para utilização comercial ou acadêmica.

A plataforma de nós sensores Mica Motes é uma das mais usadas em redes de sensores sem fio. Possui conexão para sensores de diversas naturezas: iluminação, temperatura, barométrico, aceleração, acústico, magnético, entre outros. O sensor Mica2 (Crossbow, 2008) apresenta baixo consumo de energia e, conseqüentemente, longa vida útil da bateria. A taxa de transmissão de dados é 38.4 Kbps e possui um transceptor multicanal de rádio que opera na frequência de 868 ou 916 MHz. O sensor oferece suporte para reprogramação *wireless* e adota o TinyOS como sistema operacional. Algumas das aplicações desenvolvidas são: monitoramento ambiental, redes sem fio de larga escala e computação distribuída.

A plataforma de código aberto TelosB (Crossbow, 2008) foi desenvolvida para permitir experimentos e estudos em laboratório à comunidade de pesquisadores. Possui recursos como: programação via USB, rádio com antena integrada *onboard* compatível com IEEE 802.15.4, taxa de transmissão de dados relativamente alta, a 250 Kbps. Além disso, possui um micro-controlador do modelo TI MSP430 com 10 KB de memória RAM e sensores integrados de temperatura, luminosidade e umidade opcionais. Também utiliza o sistema operacional TinyOS.

Os nós sensores do projeto MicroAMPS (UAMPS, 2004) possuem uma política de gerenciamento de energia que permite ao sensor adaptar seu consumo de energia às características do ambiente onde se encontra e dos recursos que ele dispõe. Portanto, é indicado para aplicações cujo ambiente apresenta muitas variações (Ruiz et al, 2004). O alcance fica entre 10 e 100 metros e a taxa chega a até 1 Mbps, em transmissões ponto a ponto, operando o rádio transceptor LMX3162 a uma frequência de 2,45 GHz

Cabe citar também o BEAN (*Brazilian Energy-Efficient Architectural Node*), o único sensor existente no meio acadêmico desenvolvido no Brasil, pelo projeto *Sensor-Net* (SENSORNET, 2004). Por enquanto, existem apenas protótipos desse sensor, mas o projeto continua em expansão e, segundo o site do projeto, pretende-se lançá-lo comercialmente.

2.5 Pilha de protocolos e camadas

Numa rede de sensores, o protocolo é responsável por gerenciar a comunicação entre os nós sensores e também entre a rede e o mundo exterior. Sua importância é fundamental para evitar e corrigir congestionamentos, balanceando a carga de informação gerada. A eficiência do protocolo de comunicação varia de acordo com a topologia da rede e sua aplicação.

Nesta seção, são apresentados alguns dos protocolos de comunicação desenvolvidos especificamente para RSSFs e algumas características das camadas de comunicação.

2.5.1 Camada física

A tarefa da camada física é a transmissão de mensagens entre sensores. Ela é responsável por selecionar as frequências que serão utilizadas, gerar a portadora, detectar, modular e codificar o sinal. O projeto da camada física deve levar em conta o meio de transmissão (Rossetto, 2007). A comunicação entre os sensores da rede pode utilizar sinais ópticos, sinais infravermelhos ou sinais de rádio-frequência.

A comunicação óptica tem como principal vantagem o consumo de energia menor que nas outras tecnologias. Porém, esta comunicação exige que o transmissor e o receptor estejam alinhados, além de ser sensível às condições atmosféricas.

A comunicação através de sinais infravermelhos também exige uma linha de sinal entre o transmissor e o receptor. Por esta limitação, as redes com comunicação por infravermelho são pouco utilizadas.

A comunicação através de sinais de rádio-frequência é a mais comum. A principal vantagem é que neste tipo de rede o transmissor e o receptor não precisam estar alinhados. O receptor só precisa estar suficientemente próximo do transmissor, para ser capaz de receber os sinais de rádio com um nível de potência aceitável (Rezende, 2004). Como desvantagens, a comunicação por rádio é sensível a ruídos provocados por aparelhos que operam na mesma faixa de frequência.

2.5.2 Camada de enlace

A camada de enlace é capaz de identificar os nós sensores da rede, visto que as redes de sensores sem fio não exigem a definição prévia de uma infra-estrutura. Além do controle

de acesso ao meio (MAC), esta camada realiza as tarefas de controle de erros, detecção de quadros e multiplexação do fluxo de dados (Rezende, 2004).

O controle de acesso ao meio é responsável pelo estabelecimento da comunicação *multihop*, como forma de organizar a rede e estabelecer rotas. Outra função do MAC é a distribuição dos meios de transmissão entre os sensores que fazem parte da rede. Para isso, existem diversos protocolos, sendo mais comum o CSMA, também utilizado no padrão *Ethernet*, com um mecanismo que evita colisões.

O controle de erros também é muito importante, pois permite a retransmissão de dados que não foram recebidos corretamente, geralmente em função de erros de transmissão. Um dos protocolos mais utilizados é o ARQ (*Automatic Repeat Request*). No ARQ, o receptor envia mensagens de reconhecimento do tipo ACK quando recebe um pacote corretamente e NAK quando detecta a perda de um pacote. Desta forma, o transmissor é informado dos erros que ocorrem na transmissão, e pode reenviar os pacotes nos quais houve erros (Karl, 2005).

2.5.3 Camada de rede

A camada de rede é responsável pelo roteamento de dados entre os sensores, ou seja, encontrar um caminho entre o nó de origem e o destino de uma mensagem. Os protocolos de roteamento utilizados devem suportar a comunicação *multihop* e devem buscar sempre o uso mais eficiente possível da energia do sensor. Além disso, devem tratar a redundância dos dados transmitidos, já que os dados dos nós fontes, em geral, referem-se a um fenômeno em comum, como é descrito por Ruiz et al (2004).

A escolha da rota que será utilizada para transmissão de uma mensagem entre um sensor e o nó (sink) que se comunica com o mundo exterior pode utilizar um dos seguintes critérios: maior energia disponível, em que a rota possui a maior soma possível das energias disponíveis em todos os sensores; menor consumo de energia, onde o consumo de energia na transmissão de pacotes entre sensores é o menor possível; menor número de saltos, em que a rota é calculada de acordo com o número de sensores pelo qual o pacote tem que passar até chegar ao nó sink.

Os protocolos são geralmente classificados de acordo com a arquitetura da rede. Quanto à configuração da sua organização, uma RSSF pode ser caracterizada como hierárquica ou plana. A hierárquica é uma rede em que os nós estão organizados em grupos chamados *clusters*, cada um deles com um líder que poderá ser eleito pelos nós comuns, sendo que os grupos podem organizar hierarquias entre si, de acordo com Ruiz et al (2004).

Daí a origem do nome. Já, numa rede plana, os nós não estão organizados em grupos. Consequentemente não existe uma hierarquia.

2.5.3.1 Protocolos de roteamento para redes planas

Na difusão direcionada, o nó que deve transmitir nomeia os dados usando um par de atributos, que descrevem a tarefa a ser desempenhada. O nó *sink* propaga quais atributos quer receber. Os nós vizinhos repassam essa informação, que, ao passar pelos nós, preenchem um campo correspondente à distância percorrida. Ao chegar a um nó que contenha o atributo de interesse, este o envia através do caminho do gradiente até a estação base, não havendo identificação prévia dos nós na rede.

Outro algoritmo de roteamento para redes planas é o SAR (*Sequential Assignment Routing*), que realiza roteamento *multihop* utilizando tabelas, pelas quais faz uma seleção de múltiplos caminhos, evitando *overhead* em caso de falha. Ele cria diversas árvores, sendo que a raiz de cada uma delas é vizinha da estação base. A escolha do caminho a ser utilizado é baseada de acordo com fatores como os recursos de energia disponíveis, QoS e a prioridade do pacote a ser enviado. As árvores formam os múltiplos caminhos e evitam os nós problemáticos, podendo um nó pertencer a várias árvores distintas, o que lhe permite enviar suas mensagens por uma entre várias árvores disponíveis.

Há também o protocolo SPIN (*Sensor Protocols for Information via Negotiation*), que utiliza um conjunto de protocolos de negociação para disseminar as informações de um nó sensor para todos os outros nós da rede. O roteamento é feito com base na energia disponível em cada nó (Ruiz et al, 2004). Os três tipos de mensagens disseminadas são o ADV (*Advertisement*), que avisa quando um nó possui dados para compartilhar; REQ (*Request*), mensagem que um nó vizinho envia a outro sensor quando o dado compartilhado o interessa; e DATA, quando o primeiro nó responde à requisição com uma mensagem de dados.

2.5.3.2 Protocolos de roteamento para redes hierárquicas

O LEACH (*Low-Energy Adaptative Clustering Hierarchy*) é um protocolo que tem por objetivo minimizar o custo de energia dos nós das RSSFs. Ele utiliza o agrupamento dos nodos (*cluster*) de forma que sempre exista um nodo principal (*cluster-head*) em cada grupo, sendo que a eleição de quem será o nodo principal é feita com base nos nós com maior energia disponível, como aponta Vidal (2007). A função do nodo principal é agregar os dados

dos nodos do mesmo grupo, tratar as informações redundantes e enviá-los à estação base, de onde os dados serão finalmente processados. Para resolver o problema da falta de energia, há uma rotação dos *cluster-heads* dentro dos *clusters*. A cada período de tempo, os nós se alternam nessa função, havendo um gasto de energia mais uniforme entre os nós e evitando também que a perda de um *cluster-head* leve à inutilização da rede. A comunicação entre os nós comuns e o *cluster-head* é feita através de TDMA.

O algoritmo LEACH-C (*LEACH-Centralized*) tem os mesmos princípios de funcionamento do LEACH, porém o nodo principal do grupo é centralizado e eleito uma só vez. O motivo da criação do LEACH-C é que o uso do LEACH não garante necessariamente um melhor desempenho, já que a cada rodada um novo nodo principal é escolhido, assim tendo-se sempre um desempenho distinto, de acordo com Vidal (2007). Se um dos nós escolhidos tiver um mau desempenho energético, isto afetará a média geral. No LEACH-C, cada nó manda para a estação base informações sobre sua localização física e quantidade de energia disponível. A escolha do nodo principal se dá com base nos que tem mais energia disponível. Desta forma, garante-se uma dissipação de energia mais uniforme pela área de distribuição dos nós.

2.5.4 Camada de transporte

Diferentemente das redes tradicionais, as redes de sensores sem fio normalmente não exigem um protocolo de transporte na comunicação entre os sensores da rede, visto que a maioria das aplicações admite a perda de dados. Porém, existem aplicações específicas nas quais a entrega confiável de dados é importante. Tais aplicações são serviços com funções de desligamento e gerenciamento de nós, em que o uso de um protocolo de transporte é essencial.

Embora o protocolo TCP seja amplamente utilizado na Internet e ofereça transmissão de dados de forma confiável, ordenada e com recuperação de perda de pacotes, seu uso não é recomendado em RSSFs (Karl, 2005; Sohraby, 2007). Uma das causas é o excesso de mensagens enviadas para começar a comunicação entre dois pontos. Além disso, a perda de alguns segmentos que não são indispensáveis gera uma retransmissão das informações, tornando a taxa de transmissão mais baixa e aumentando o consumo de energia dos sensores, principalmente quando o nó transmissor encontra-se longe do nó *sink*.

Diversos fatores são levados em consideração na escolha de protocolos para a camada de transporte, como conservação de energia, controle de tráfego, confiabilidade na

disseminação dos dados, segurança e gerência dos nós, como indica Sohraby (2007). Portanto, esses protocolos devem ser simples, mas capazes de entregar os dados corretamente.

2.5.4.1 Protocolos de transporte

CODA (*Congestion Detection and Avoidance*) é um protocolo que detecta o congestionamento da rede de acordo com a taxa de utilização de cada *buffer* e o tráfego nos canais. O nó que detectar o congestionamento manda uma mensagem ao seu vizinho para reduzir sua taxa de transmissão. Quando isso acontece, um determinado *bit* do pacote enviado é alterado. Assim que o *sink* receber essa informação, ele envia uma mensagem a todos os nós para que reduzam suas taxas. Quando se atinge um nível razoável, ele manda outra mensagem permitindo o aumento da taxa de transmissão dos nós (Sohraby, 2007).

O protocolo ESRT (*Event-to-sink Reliable Transfer*) é usado em RSSFs baseadas em eventos e o reconhecimento é feito na estação-base através do recebimento de mensagens de vários sensores, Ruiz et al (2004). Assume-se um reconhecimento confiável quando uma quantidade satisfatória de sensores envia mensagens. A estação-base calcula o estado de operação da rede de acordo com os níveis de congestionamento e confiabilidade. De acordo com esse estado, o protocolo repassa aos nós a taxa de envio de dados desejada, evitando consumo excessivo de energia e processamento nos sensores.

O PSFQ (*Pump Slowly, Fetch Quickly*) utiliza a confirmação ponto a ponto na correção local de erros e se adapta a diferentes condições da rede, sendo, por isso, indicado para as redes de sensores, que possuem altas taxas de erros. Possui 3 operações: *push*, correspondente ao envio de fragmentos de uma mensagem ao nó seguinte, dentro do caminho até o destinatário; *fetch*, que acusa que algum fragmento não foi recebido por um nó intermediário dentro do tempo esperado; *report*, que indica ao emissor o recebimento completo da mensagem (Ruiz et al, 2004).

2.6 Padrões para RSSF

O objetivo de se desenvolver padrões para as camadas de protocolos das redes de sensores sem fio é garantir a interoperabilidade entre os dispositivos que formam a rede. Além disso, eles podem reduzir o risco de integrar comunicações *wireless* proprietárias (Rossetto, 2007). A tendência é que, com o uso cada vez mais frequentes dessas tecnologias, melhorem-

se o alcance e a confiabilidade das RSSFs. Serão vistos alguns dos padrões das tecnologias de comunicação sem fio.

A família de padrões IEEE 1451 tem como objetivo facilitar a integração de sensores e atuadores de diferentes fabricantes. Os principais resultados obtidos foram a criação do TEDS (*Transducer Electronic Data Sheet*), que contém informações relativas ao sensor, como o fabricante, faixa de medição, parâmetros de desempenho, etc.; *Address Logic*, que especifica as formas de endereçamento do sensor na rede; e o NCAP (*Network Capable Application Processor*), que faz a interface do sensor com a rede utilizada pelo sistema, como descreve Rezende (2004). Assim, o tipo de dispositivo utilizado não interfere em como a comunicação na rede é feita.

Segundo Callaway (2003 apud ROSSETTO, 2007), o padrão IEEE 802.15.4 especifica as camadas físicas e de enlace para dispositivos com complexidade reduzida, baixo custo, consumo de energia limitado e conectividade via rede sem fio. As aplicações atendidas são bem variadas e algumas técnicas ajudam a reduzir o consumo de energia, como o anexo de um *byte* ao quadro recebido pela camada física, a fim de indicar a qualidade do canal de comunicação e controlar a energia do nó transmissor.

Para ditar as especificações das camadas de rede, transporte e aplicação de dispositivos que operam no padrão IEEE 802.15.4, foi criado o *ZigBee Alliance*, um consórcio formado por diversos fabricantes e provedores de serviços. A idéia é que o padrão seja global, atendendo às necessidades da comunicação sem fio entre sensores com taxas de transmissão médias e altas, com pouco alcance e baixo custo de implementação (Rossetto, 2007). Exemplos práticos do uso dessa tecnologia são periféricos de computadores pessoais sem necessidade de cabos e controle remoto de equipamentos domésticos.

A tecnologia *Bluetooth*, também desenvolvida em consórcio, possui o intuito de realizar a conexão entre dispositivos eletrônicos sem fio e computadores. Opera na faixa de frequência de 2.4 GHz e permite taxa de comunicação entre os dispositivos de 1 Mbps, com alcance de até 100 metros. Algumas restrições do seu uso em RSSFs são a energia gasta pela arquitetura de mestre-escravo e a possibilidade de poucos dispositivos conectados à rede, em detrimento aos vários sensores utilizados nas redes de sensores, além do nível de segurança apresentado ser questionável (Loureiro et al, 2003).

A família de protocolos IEEE 802.11, por sua vez, foi desenvolvida para descrever a tecnologia das redes sem fio (*Ethernet*) e opera a altas taxas de transmissão de dados, o que acaba elevando consideravelmente o consumo de energia. Outro fator que restringe seu uso em redes de sensores é deixar usuários independentes compartilharem um canal em comum de

forma competitiva (Karl, 2005). Por isso, a tecnologia *ZigBee* é a que mais se destaca no momento, em relação às outras.

2.7 Sistemas Operacionais

A função básica de um sistema operacional é controlar o *hardware* e prover uma abstração entre este e o *software* aplicativo. Como as aplicações em RSSFs normalmente são centradas nos dados e os nós sensores possuem recursos limitados, como processador lento e pouca memória e energia disponível, os sistemas operacionais usados nesse tipo de rede diferem dos convencionais em diversos aspectos.

Existe um conjunto de características desejáveis para sistemas operacionais dos sensores. Sohraby (2007) cita algumas delas: ser compacto, devido à memória reduzida; suportar aplicações que necessitam coleta e disseminação das informações em tempo real; possuir mecanismos eficientes de gerenciamento de recursos, para garantir aos processos tempo suficiente da *CPU*; suportar gerenciamento de energia a fim de aumentar a vida útil dos sensores, por exemplo, fazendo o processo dormir enquanto o sistema estiver ocioso.

A seguir, serão vistas algumas características e dificuldades existentes na implementação, bem como alguns dos sistemas operacionais mais utilizados em redes de sensores sem fio.

2.7.1 Princípios de projeto de sistemas operacionais para sensores

A tarefa principal de qualquer sistema operacional é gerenciar os recursos de *hardware* disponíveis na máquina. Sistemas operacionais de sensores não são exceção a esta regra. O sistema operacional provê serviços abstratos, tais como leitura de sensores, envio e recebimento de dados através de ondas de rádio, e utilização de temporizadores (Stojmenovic, 2005, p. 177).

O gerenciamento de *hardware*, segundo Stojmenovic (2005), pode ser implementado como uma biblioteca de chamadas de funções, visto que o processador normalmente opera nos modos de usuário e supervisor e não existe uma unidade de gerenciamento de memória. Entretanto, essas chamadas de função são uma janela para usuários que acessam o *hardware* diretamente. Tal esquema não impede que os usuários realizem ataques, afetando a segurança e confiabilidade do sistema.

Outros dois problemas tratados pelo sistema operacional são o escalonamento e a sincronização de múltiplas tarefas. Ele deve atuar sobre a alocação do processador entre os processos e garantir exclusão mútua quando necessário. Uma das maneiras de realizar a coordenação das tarefas é restringir o usuário a uma única tarefa, através da execução do código da aplicação diretamente no micro-controlador (Stojmenovic, 2005). Assim, a coordenação de múltiplas tarefas é feita individualmente em cada aplicação. Devido à alta complexidade de tratar isso, sugere-se que o mecanismo de coordenação de tarefas seja implementado no sistema operacional apenas se as aplicações forem realmente utilizar esse recurso.

A restrição de recursos dos nós sensores é um ponto crítico na análise dos requisitos exigidos e afeta diretamente o desenvolvimento dos sistemas operacionais. A primeira restrição é a escassez de memória de dados, que implica em um possível comportamento imprevisível da rede. A engenharia de software é fundamental nesse caso, pois as técnicas que reduzem o consumo de memória impõem limites nos serviços que o sistema operacional pode oferecer. Já a memória de programas não possui restrições severas, pois utiliza-se a memória *flash* para armazenar a imagem de um programa, mais barata e mais difícil escrever neste tipo de memória por acidente (Stojmenovic, 2005). Quanto à utilização do processador, no envio e recebimento de dados entre os sensores, ele não é sobrecarregado. O que pode ocupar mais intensamente o processador são tarefas como agregação, transformação e criptografia de dados.

Talvez o maior problema das redes de sensores seja o consumo de energia dos nós. Para diminuir o consumo da bateria e aumentar o tempo de vida do sensor, os sistemas operacionais implementam mecanismos que desligam alguns componentes enquanto estes não forem necessários. O gerenciamento de energia pode ser de forma implícita, ou seja, o sistema operacional decide sozinho quando deve desativar um componente; ou de forma explícita, em que as aplicações avisam ao sistema operacional os recursos que serão utilizados, facilitando a determinação de quais componentes podem permanecer ativos (Stojmenovic, 2005).

Outra característica interessante a ser prevista é a manutenção da rede, após a implantação dos sensores. Stojmenovic (2005) aponta que, no caso da existência desse recurso, o sistema operacional deve prover uma interface para que o administrador da rede possa realizar a reprogramação dos nós remotamente, isto é, alterar o *software* presente em cada nó.

Também é importante ressaltar que os nós sensores não rodam aplicações completas, apenas partes de uma aplicação distribuída que se conecta ao usuário pelo nó sorvedouro

(*sink*). Portanto, o sistema operacional deve ser otimizado para realizar interação com outras máquinas.

2.7.2 Exemplos de sistemas operacionais em RSSF

Dos sistemas operacionais para redes de sensores mais comuns serão brevemente descritos alguns que apresentaram características interessantes ou pontos favoráveis à sua utilização em relação aos demais, sendo estes: Contiki, eCos, Mate, OSPM, Eyes OS, SenOS, PicOS (Karl, 2005; Sohraby, 2007).

O sistema operacional TinyOS (TINYOS, 2008) foi desenvolvido especialmente para sensores com poucos recursos e permite às aplicações acessarem diretamente o *hardware*. Para suportar aplicações concorrentes nos dispositivos usando pouco processamento e pouca quantidade de memória, utiliza um modelo baseado em eventos. Quando o processador está ocioso, ele mantém o processo em estado de dormência. Já quando o processador torna-se disponível, é feito um escalonamento das tarefas a serem executadas pelo escalonador já implementado. Rossetto (2007, p. 15) afirma que “o TinyOS pode ser definido como um *framework* de programação para redes de sensores, com um conjunto de componentes que permite construir um sistema operacional específico para cada aplicação”. Podem ser apontadas como principais vantagens desse S.O.: pouco código e pequena quantidade de dados necessários; alta velocidade na propagação de eventos e na troca de contexto.

O Mantis (MANTIS, 2007) é um sistema operacional baseado em UNIX, *multithread*, com filas de prioridade e utiliza o mecanismo de semáforo para a sincronização entre os processos. O objetivo disto é oferecer facilidade aos programadores, que já são familiarizados com a linguagem de programação C e com algumas características comuns aos sistemas operacionais tradicionais. Segundo Stojmenovic (2005), o Mantis também apresenta um *framework* que permite um alto nível de integração entre *drivers* de dispositivos, protocolos de rede e o próprio S.O. As principais vantagens do Mantis são: a abstração de problemas de controle e mecanismos de tolerância a falhas isolados das aplicações.

Outro sistema operacional comum em RSSFs é o MagnetOS (MAGNETOS, 2008), caracterizado por adaptar os recursos disponíveis dos nós às mais diversas aplicações e apresentar conservação de energia. Alguns objetivos são oferecer abstração às aplicações e possuir escalabilidade para redes de grande porte. Constitui-se de uma *Java virtual machine*, com componentes estáticos e dinâmicos. Os estáticos traduzem as instruções das aplicações para o nível de *byte-code*, mantendo a semântica original. Sohraby (2007, p. 278) explica que

“os componentes dinâmicos são usados para monitoramento de aplicações, criação de objetos, invocação e migração de código”. Outros recursos presentes nesse S.O. são dois algoritmos que ajudam a controlar e reduzir o consumo de energia na rede, aumentando o tempo de vida útil, e uma interface aos programadores para posicionamento explícito de objetos.

2.8 Segurança em RSSF

Devido à natureza deste tipo de rede, que muitas vezes se encontra em ambientes hostis, a segurança deve receber atenção especial. As redes de sensores apresentam certa vulnerabilidade quanto a ataques de segurança, pois a forma de transmissão de dados presente permite que todos os receptores dentro do raio de alcance do sinal captem a informação. Por isso, mecanismos de proteção física, sobre a comunicação e sobre os dados são de vital importância muitas vezes. Antes de se implementar tais mecanismos, é necessário saber que nível de segurança é desejado.

Alguns requisitos básicos são exigidos para tal. A confidencialidade dos dados significa que uma rede vizinha ou qualquer outra entidade, mesmo que intercepte os dados transmitidos dentro de uma rede, não consegue lê-los. Uma das formas de garantir isso é através da criptografia, mantendo uma chave secreta para acesso dos dados. A autenticação sobre os dados garante que os mesmos se originaram de uma fonte correta ou confiável. Pereira (2003) descreve que a integridade de dados assegura ao receptor que o dado recebido não foi alterado durante seu trânsito. Outra característica comum é a disponibilidade, que garante que os dados estarão acessíveis no instante que uma entidade autorizada desejar realizar alguma operação.

2.8.1 Tipos de problemas e ataques

Um dos problemas mais comuns em RSSFs é a captura de nós, facilitada pelo posicionamento dos sensores em locais pouco protegidos. Quando acontece esse ataque, um nó é removido da rede e tem sua chave criptográfica descoberta. Então, ele é reprogramado e inserido novamente, podendo comprometer o funcionamento da rede. Para contornar isso, podem-se usar protocolos que enviem a mesma mensagem por mais de um caminho, evitando a dependência de um nó ou replicar o estado da rede e usar o consenso para detectar

inconsistências, como sugere Rossetto (2007). Uma outra solução é programar os nós para que, logo que sejam retirados da rede, apaguem sua memória e seus dados criptografados.

A negação de serviço pode ser feita para desabilitar serviços ou comprometer o funcionamento dos fornecedores de algum serviço. Isso pode ser obtido através da destruição de nós ou da sobrecarga do servidor, por meio de múltiplas requisições do mesmo serviço. Tais ataques podem ocorrer em várias camadas.

Na camada física, além da remoção indevida de um nó da rede, pode ocorrer também a obstrução. Ela acontece quando são emitidos sinais na mesma frequência usada pelos nós da rede, causando uma interferência. Isso é facilmente obtido posicionando dispositivos próximos à rede, de modo que enviem sinais de rádio na mesma banda de frequência. Quanto maior for o número de atacantes e quanto mais próximo estiverem do nó *sink*, acarretará maiores danos a toda a rede (Karl, 2005).

Na camada de enlace, uma das táticas é causar a colisão em alguns *bits* da mensagem transmitida. Isso poderá resultar na perda de um pacote inteiro, sem que haja um grande esforço por parte do atacante. Outra forma de danificar os serviços é a exaustão, que consiste em solicitar constantemente o acesso ao meio, ou seja, um nó enviar diversos pacotes ao mesmo destino exigindo resposta (Karl, 2005).

A camada de rede também pode ser atacada de diversas maneiras. Uma das formas é impedir que o nó repasse as mensagens recebidas ou fazê-lo enviar para caminhos incorretos. Os atacantes podem se infiltrar, comportando-se como nós normais. Eles fingem possuir boas rotas de comunicação, mas acabam destruindo os pacotes de dados recebidos, ocasionando inclusive um congestionamento na rede (Karl, 2005).

Os ataques à camada de transporte incluem armadilhas como aplicar números de seqüências errados a pacotes, causando inúmeras retransmissões até que a conexão seja encerrada por um dos participantes. Karl (2005) indica que em redes de sensores implantadas a fim de detectar um evento no ambiente, um nó malicioso pode gerar alguns dados indicando tal evento, acionando automaticamente os demais nós da rede.

Capítulo 3

REQUISITOS BÁSICOS PARA RSSF

Este capítulo versa sobre o tema principal deste trabalho, que é a definição do equipamento e do sistema operacional, considerados os requisitos básicos para a implantação de uma rede de sensores sem fio.

3.1 Definição do equipamento

Esta seção aborda os requisitos e necessidades para a definição do equipamento a ser utilizado na rede de sensores sem fio do projeto. Em seguida, serão apresentados os principais módulos sensores disponíveis para aquisição no mercado e suas características. Ainda será descrito o módulo escolhido e será justificada esta decisão.

3.1.1 Requisitos

O projeto de pesquisa do qual este trabalho faz parte prevê a implantação de uma rede de sensores sem fio para captar os dados de máquinas agrícolas em funcionamento, a fim de melhor gerenciá-las. Portanto, o equipamento deve ser robusto, de preferência compacto, e capaz de realizar medições a partir de diversos tipos de variáveis, tanto da máquina quanto do ambiente no qual se está operando.

Aliado a isso, a escassez de recursos financeiros do projeto requer um equipamento que apresente baixo custo na aquisição e manutenção. Então este pode ser considerado o maior desafio na escolha do módulo sensor com o qual será trabalhado mais adiante.

3.1.2 Equipamentos pesquisados

Esta seção trata sobre os equipamentos necessários para a implantação de uma rede de sensores sem fio. De acordo com as características e funcionalidades apresentadas por cada um dos itens pesquisados, serão descritas as vantagens e desvantagens dos equipamentos.

3.1.2.1 MICA2

A plataforma de nós sensores Mica Motes é uma das mais usadas em redes de sensores sem fio. Possui conexão para sensores de diversas naturezas: iluminação, temperatura, barométrico, aceleração, acústico, magnético, entre outros. O sensor MICA2 (Crossbow, 2008), comercialmente chamado MPR400CB, apresenta baixo consumo de energia e, conseqüentemente, longa vida útil da bateria. A taxa de transmissão de dados é de apenas 38.4 Kbps e possui um transceptor de rádio que pode operar na frequência de 868 ou 916 MHz, com alcance de aproximadamente 150 metros.

O módulo possui memória EEPROM de 4 KB, uma memória Flash com 128 KB, na qual ficam armazenados os programas aplicativos desenvolvidos, e outra memória Flash de 512 KB, que armazena os dados de até 100.000 medições. O sensor oferece suporte para reprogramação *wireless* e adota o TinyOS como sistema operacional. Algumas das aplicações desenvolvidas são: monitoramento ambiental, redes sem fio de larga escala e computação distribuída.

Embora o MICA2 apresente condições favoráveis para seu uso, como consumo reduzido de bateria e o fato de permitir alta escalabilidade, – é capaz de gerenciar mais de mil nós em uma única rede – outro fator foi relevante para este não ser a melhor opção. Percebe-se pela Figura 3.1 que o módulo por si só é compacto. Mas, para que seu funcionamento seja efetivo, o mesmo requer uma placa de interface serial/USB com o computador e placas com os sensores propriamente ditos, os quais capturarão os dados. Isto acaba elevando drasticamente o custo do equipamento completo, extrapolando as limitações financeiras do projeto.

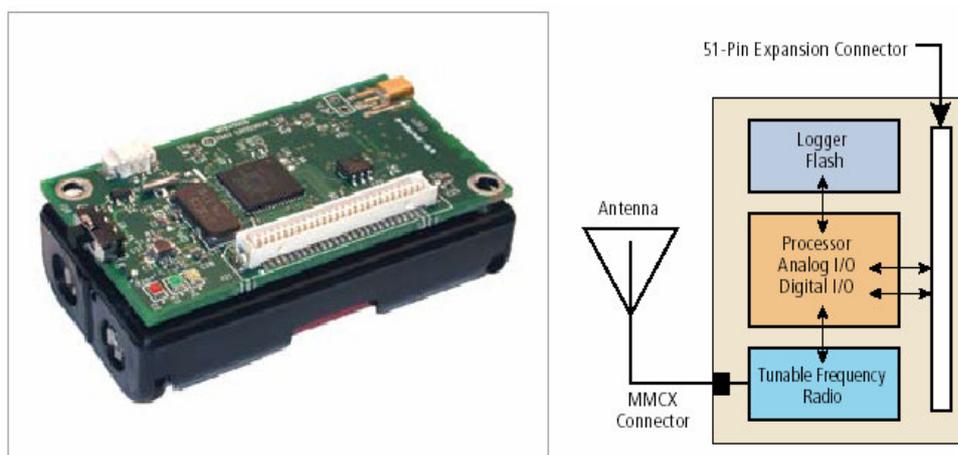


Figura 3.1 – Módulo MICA2 e sua arquitetura – Fonte: CROSSBOW (<http://www.xbow.com>)

3.1.2.2 MICAz

O módulo MICAz, também comercializado pela Crossbow (2008) pelo nome MPR2400CA, é da família Mica Motes e pode ser considerado o sucessor do MICA2, apresentado anteriormente. Possui taxa de transmissão de dados bem maior que o seu antecessor: 250 Kbps. A frequência do módulo de rádio pode variar entre 2,4 e 2,48 GHz e é capaz de captar sinais ao seu redor com raio de até 300m. Também gerencia sensores de diversas naturezas e apresenta escalabilidade de mais de mil nós.

A memória principal também é de apenas 4 KB, com a memória Flash de 128 KB; e o sistema operacional suportado é o mesmo: TinyOS. Outras melhorias apresentadas são: a capacidade do transceptor de rádio ser mais resistente a interferências e a conseqüente segurança sobre os dados que isso acarreta.

O fator determinante para que o uso do MICAz no projeto fosse rejeitado caracteriza-se pelo mesmo problema do MICA2. A versão padrão do módulo não contém a placa de sensores necessária para a realização dos testes e a placa de interface com o computador. Observa-se no esquema da Figura 3.2 a ausência das placas acessórias exigidas.

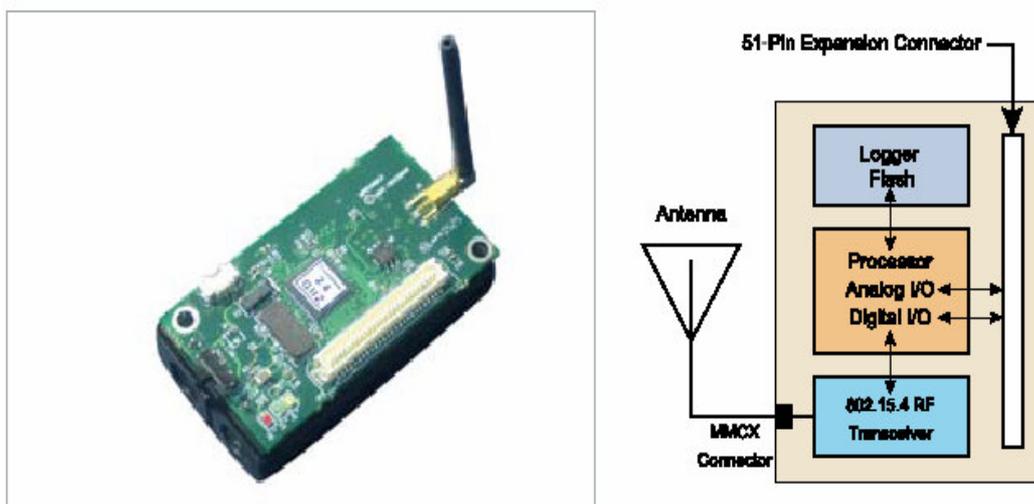


Figura 3.2 – Módulo MICAz e sua arquitetura – Fonte: CROSSBOW (<http://www.xbow.com>)

3.1.2.3 IRIS

Outro componente da família Mica Motes é o IRIS (Crossbow, 2008), modelo XM2110CA. Suas características são bastante semelhantes às do MICAz, inclusive, com mesma taxa de transmissão de dados e banda de frequência de rádio. Como os modelos

anteriores, suporta o sistema TinyOS e é totalmente compatível com os acessórios da família Mica Motes.

É indicado para aplicações que exigem alta velocidade na transmissão de dados, como sons, vídeos e vibrações. As principais vantagens são: o alcance de até 500 metros entre os nós sem a necessidade de amplificar o sinal; e a memória RAM de 8 KB, praticamente o dobro da existente nos modelos anteriores.

Na Figura 3.3 a seguir, é mostrado o equipamento descrito e um esquema demonstrativo. Pelas mesmas razões dos outros módulos sensores, não foi o preferido para ser utilizado no projeto.

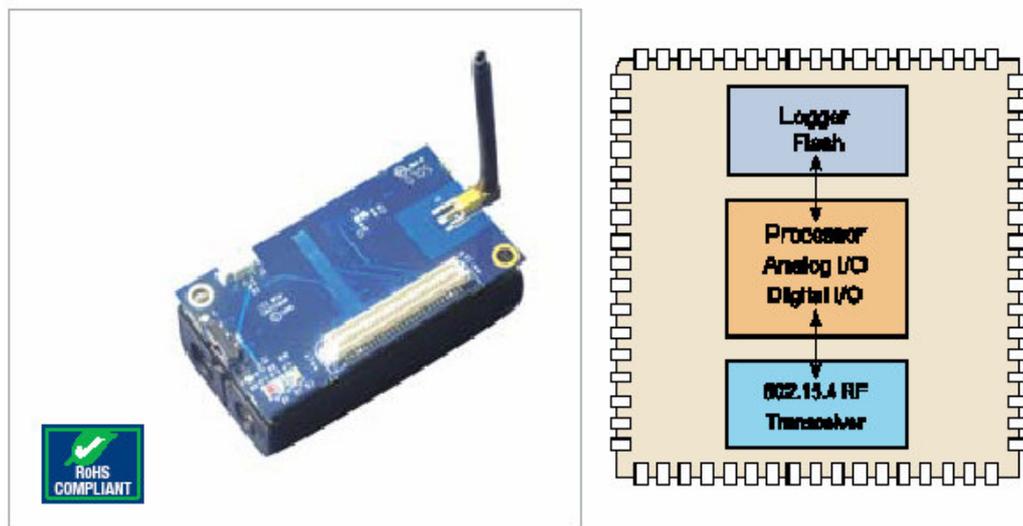


Figura 3.3 – Módulo IRIS e diagrama – Fonte: CROSSBOW (<http://www.xbow.com>)

3.1.2.4 Imote2

O Imote2 (Crossbow, 2008), ou IPR2400, é um dos modelos mais modernos e avançados do fabricante. A plataforma inclui o processador de baixo consumo PXA271 XScale®, com frequência entre 13 e 416 MHz, além de um co-processador *wireless* MMX DSP. Possui incríveis 256 KB de memória SRAM, 32 MB de memória Flash e 32 MB de SDRAM, os quais podem ser visualizados na Figura 3.4.

Também possui rádio e antena integrada que opera em 16 canais na faixa de 2,4 GHz e a uma taxa de 250 Kbps, que permite um alcance de 30 metros com linha de visada, isto é, sem obstáculos entre os nós. Outra característica interessante é o suporte a diversos sistemas operacionais, tais como TinyOS, Linux e SOS, além de outros *softwares* de código aberto.

O Imote2 é recomendado para aplicações comerciais como: processamento de imagens

digitais, análise e monitoração industrial, monitoração sísmica e de vibrações; ou até projetos de pesquisa acadêmica. Estes incluem sensores de aceleração nos três eixos, temperatura, umidade e luminosidade; interface USB, com a qual é possível recarregar a bateria, entre outros inúmeros recursos.

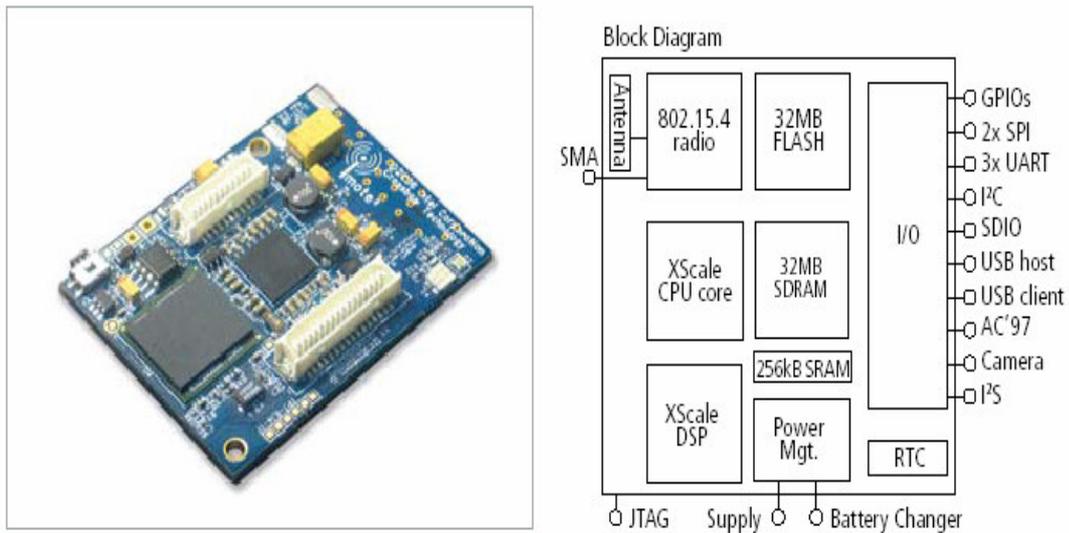


Figura 3.4 – Módulo Imote2 e o diagrama da sua estrutura – Fonte: CROSSBOW (<http://www.xbow.com>)

3.1.2.5 WSN430

Outro módulo sensor que merece destaque é o WSN430 (WorldSens, 2008). Ele foi desenvolvido com o objetivo de oferecer suporte a aplicações e implantação de redes de sensores autônomas de larga escala, com uma plataforma de baixíssimo consumo e a baixo custo, permitindo recarga da bateria, uma longa vida útil do sensor e rápida recuperação do modo *standby*.

Algumas das características do equipamento são: o transceptor, que opera em diversas faixas de frequência entre 315 e 915 MHz, com taxa de transmissão de dados a 500 Kbps, podendo ser ajustada; antena integrada e conector para antena externa; o micro-controlador TI MSP430 a 8 MHz, com 10 KB de RAM e memória Flash programável de 48 KB; e memória Flash de 1 MB para dados.

Outros recursos disponíveis são dois fotodiodos integrados, um sensor de temperatura embutido, um conector serial e conector para cartões de expansão com sensores, o que pode ser visto na Figura 3.5. O WSN430 também oferece suporte a diversos sistemas operacionais, como o Contiki, o FreeRTOS e o TinyOS.

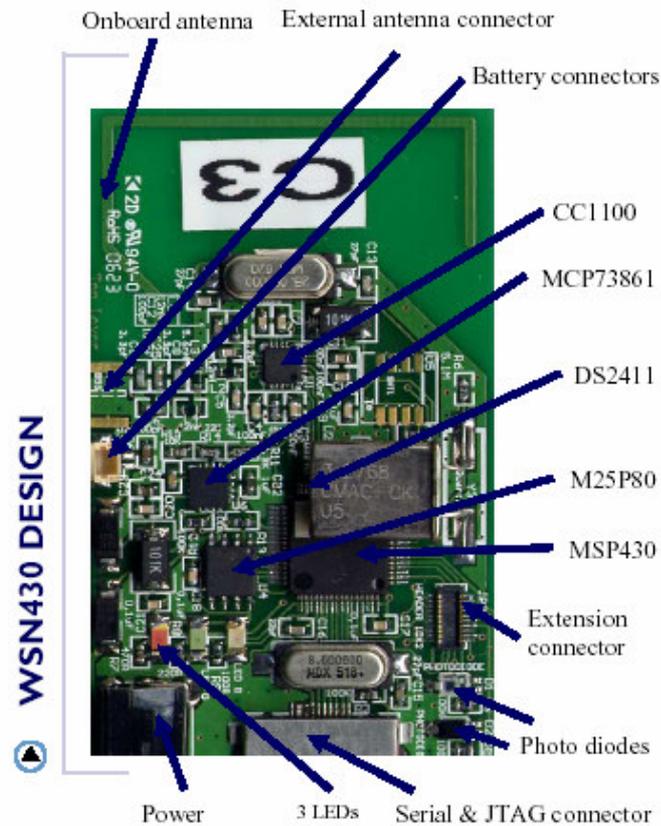


Figura 3.5 – Protótipo do WSN430 – Fonte: WORLDSSENS (<http://www.worldsens.net>)

3.1.2.6 XM2400

A placa XM2400 e sua correspondente XM900 (S3C Inc, 2007) operam na frequência de 2,4 GHz e 900 MHz, respectivamente. Elas foram desenvolvidas com o intuito de oferecer prototipação de aplicações para sensores sem fio de uma maneira prática e adequada, ao mesmo tempo em que provêem produção em alta escala a um baixo custo, com consumo reduzido de energia e permitem customização para aplicações específicas.

Os módulos possuem interface para sensores, antena interna ou externa e suportam as topologias estrela, *mesh* e híbrida. Permitem um alcance de 150 metros com linha de visada e 30 metros em lugares fechados. As indicações de uso são para aplicações de monitoração e sensoriamento remoto; controle do processo industrial; gerenciamento de energia e monitoração na agricultura.

Mas, como pode se observar na Figura 3.6, a plataforma XM2400 não possui os sensores integrados para realização dos testes e medições, além de suas dimensões não serem tão reduzidas quanto à dos outros módulos pesquisados. Outro problema encontrado foi o fato do fabricante não especificar o sistema operacional suportado.



Figura 3.6 – Plataforma XM2400 – Fonte: S3C INC (<http://www.s3cinc.com>)

3.1.2.7 M2135 Mote

O M2135, da linha SmartMesh-XT™ (Dust Networks, 2007), é um produto voltado para aplicações comerciais em ambientes industriais e garante desempenho, escalabilidade e confiabilidade. Utiliza um rádio com frequência de 2,4 GHz, alcance de aproximadamente 400 metros em ambientes externos e taxa de dados de 250 Kbps. Os componentes empregados são de baixo consumo e o equipamento provê coordenação na rede, a fim de tornar mais eficiente o uso da energia disponível.

O módulo inclui interface digital, analógica ou serial. Ele funciona em conjunto com o PM2130 *Manager* (Dust Networks, 2007), uma placa compacta com configurações semelhantes ao módulo descrito e que oferece os serviços de gerenciamento, configuração e conexão com o mundo externo para a rede de M2135 Motes. Segundo o fabricante, não é necessário o desenvolvimento de *softwares* para o funcionamento do produto, visto que ele já vem programado. Entretanto, não foram apresentadas as informações sobre a memória e o micro-controlador incluído na placa.



Figura 3.7 – M2135 Mote e PM2130 *Manager* – Fonte: DUST NETWORKS (<http://www.dust-inc.com>)

3.1.2.8 BTnode

O BTnode rev3 (BTnodes, 2007) é uma plataforma autônoma de computação e comunicação *wireless* baseada na tecnologia *Bluetooth*, com um rádio de baixo consumo – o mesmo do Mica2 – que funciona na banda de frequências entre 433 e 915 MHz e um microcontrolador. A plataforma é mais utilizada para prototipação em pesquisas sobre redes de sensores distribuídos.

As memórias existentes no módulo são: EEPROM de 4 KB, SRAM de 64+180 KB e outra memória Flash ROM de 128 KB. Possui uma porta de extensão integrada, que permite conexão com placas de sensores diversos e placa FPGA. Em termos de software, é compatível com o sistema operacional TinyOS e com o BTnut System Software e seus exemplos de programas, escritos em linguagem C. Outra característica é que possui ferramentas que facilitam sua utilização em máquinas cujos sistemas instalados sejam Windows, Linux, MacOS ou BSD.

Na Figura 3.8, é mostrado o BTnode rev3 e o esquema de sua arquitetura. Como os demais módulos sensores, necessita equipamentos de *hardware* acessórios para a realização de testes e medições.

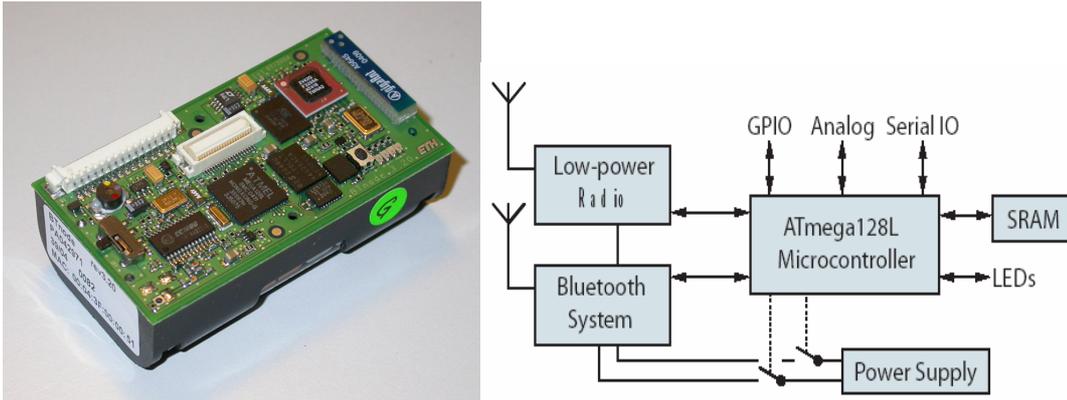


Figura 3.8 – BTnode rev3 e sua arquitetura – Fonte: BTNODES (<http://www.btnode.ethz.ch>)

3.1.2.9 TinyNode

Outra plataforma importante é o TinyNode 584 (Shockfish, 2006), o qual provê uma forma simples e confiável de comunicação entre sensores e atuadores. De acordo com o fabricante, sua bateria dura aproximadamente 5 anos. Utiliza o transceptor de rádio Xemics® XE1205, na frequência de 868 MHz, e um micro-controlador de 8 MHz.

Algumas das características apresentadas são: interfaces analógica, serial e digital; rápida recuperação do modo *standby*; antena e sensor de temperatura integrados; taxa de dados de até 152,3 Kbps; facilidade para integrar diversos tipos de sensores; alcance de 200 metros em ambientes abertos. Sua imagem é vista na Figura 3.9, juntamente com a placa acessória que permite diversas conexões com outros dispositivos e sensores de vários tipos.

Além disso, possui memória RAM de 10 KB, 48 KB de espaço para os programas e 512 KB de memória Flash. Por fim, é importante ressaltar que este equipamento foi otimizado para rodar com o sistema TinyOS e pode ser configurado para funcionar com baixa largura de banda e um maior alcance, ou com um pequeno alcance e com largura de banda maior.

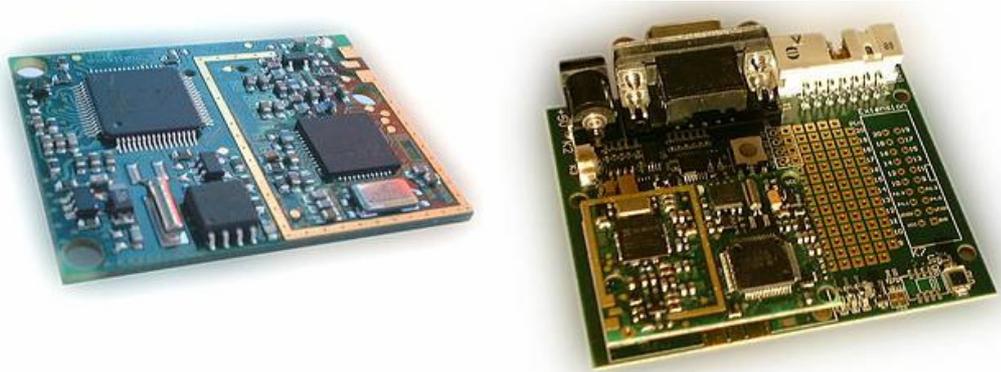


Figura 3.9 – TinyNode 584 e sua placa de expansão – Fonte: SHOCKFISH (<http://www.shockfish.com/>)

3.1.3 Equipamento escolhido: TELOSB

Após uma abrangente pesquisa sobre módulos sensores, optou-se pelo TELOSB (Crossbow, 2008). A plataforma de código aberto TelosB Mote TPR2420CA foi desenvolvida para permitir experimentos e estudos em laboratório à comunidade de pesquisadores acadêmicos e em geral. Possui recursos como: programação via USB, rádio com antena integrada compatível com IEEE 802.15.4, taxa de transmissão de dados relativamente alta, a 250 Kbps. Além disso, possui um micro-controlador do modelo TI MSP430 com 10 KB de memória RAM, memória Flash de 48 KB para programas e memória Flash de 1 MB para armazenar dados de medições.

O alcance, em ambientes fechados, varia de 20 a 30 metros; e em ambientes externos, de 75 a 100 metros. A plataforma apresenta consumo de energia reduzido e rápida recuperação do modo *standby*, o que torna mais longa a vida útil da bateria. Assim como os demais produtos da Crossbow, também utiliza o sistema operacional TinyOS, nas versões 1.1.11 ou subseqüentes.

O que o diferencia de todos os outros módulos encontrados no mercado é o conector USB embutido na placa e os sensores integrados de temperatura, luminosidade e umidade. Tais componentes podem ser visualizados na Figura 3.10. Isso permite ao pesquisador realizar testes iniciais sem a necessidade de adquirir outros acessórios, como placas de extensão, que elevam bastante o preço final do produto. Para testes específicos, podem-se adquirir sensores opcionais que capturem diversos tipos de variáveis do ambiente e integrá-los ao módulo.

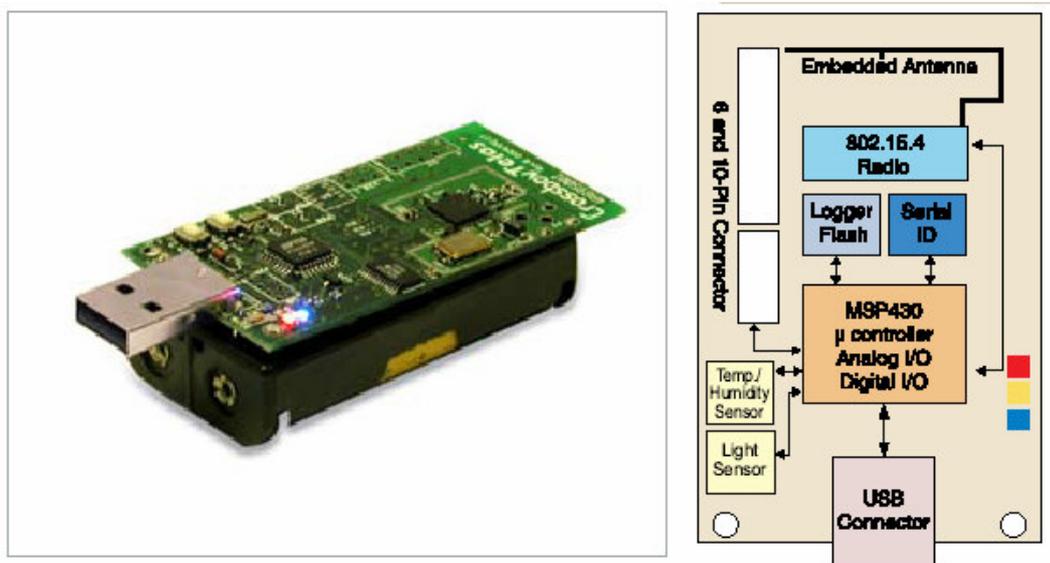


Figura 3.10 – Plataforma TelosB e sua arquitetura – Fonte: CROSSBOW (<http://www.xbow.com>)

Na tabela 3.1, é mostrado um comparativo das principais características do *hardware* de cada módulo pesquisado. Os campos não preenchidos são devido aos casos em que a informação não se encontrava disponibilizada no site dos fabricantes. Essas informações correspondem aos equipamentos sem os acessórios exigidos. Nota-se que a plataforma TelosB é a única que dispõe de uma lista aceitável de sensores integrados.

Tabela 3.1 – Quadro comparativo entre equipamentos para redes de sensores

Modelo	Frequência	RAM	Flash serial	Taxa de transmissão	Alcance	Sensores embutidos
TELOSB	2.4 – 2.48 GHz	10 KB	1 MB	250 kbps	100 m	Luz Temperatura Umidade
IRIS	2.4 – 2.48 GHz	8 KB	512 KB	250 kbps	> 300 m	NÃO POSSUI
MICA2	868/916 MHz	4 KB	512 KB	38.4 kbps	150 m	NÃO POSSUI
MICAz	2.4 – 2.48 GHz	4 KB	512 KB	250 kbps	100 m	NÃO POSSUI
Imote2	2.4 – 2.48 GHz	32 MB	32 MB	250 kbps	30 m	NÃO POSSUI
WSN 430	315/433/868 /915 MHz	10 KB	1 MB	500 kbps	-	Temperatura
XM2400	900 MHz / 2.4 GHz	-	-	-	150 m	NÃO POSSUI
M2135	2.4 – 2.48 GHz	-	-	250 kbps	400 m	NÃO POSSUI
BTnode	413/915 MHz	64 + 180 KB SRAM	128 KB	76.8 kbps	-	NÃO POSSUI
TinyNode	868 MHz	10 KB	512 KB	152,3 kbps	200 m	Temperatura

3.2 Definição do sistema operacional

Esta seção trata sobre a definição do sistema operacional que rodará sobre o equipamento escolhido, apresentado anteriormente. São descritos os S.O.s pesquisados e suas características e funcionalidades. Após, são feitas considerações sobre o S.O. escolhido, justificando esta decisão.

3.2.1 Sistemas operacionais pesquisados

Dos sistemas operacionais mais comuns para redes de sensores, serão brevemente descritos alguns que apresentaram características interessantes ou pontos favoráveis à sua utilização.

3.2.1.1 MANTIS

O MANTIS (MANTIS, 2007) é um sistema operacional de código aberto baseado em UNIX, para plataformas de redes de sensores sem fio. Ele usa a estrutura *multithread*, com filas de prioridade e utiliza o mecanismo de semáforo para garantir exclusão mútua e sincronização entre os processos de E/S (entrada e saída). Essas funções são executadas ocupando menos de 500 *bytes* de memória RAM e 14 KB de memória Flash (Sohraby, 2007). O núcleo do sistema é implementado em linguagem C, assim como a API para ambientes de desenvolvimento do Windows e Linux.

O objetivo disto é oferecer facilidade aos programadores, que já são familiarizados com a linguagem de programação C e com as características comuns aos sistemas operacionais tradicionais. De acordo com o site do desenvolvedor, o MANTIS (MOS) é suportado por diversas plataformas, entre elas: MICA2, MICAz e TELOS. Outra característica encontrada é um escalonador que ajuda a reduzir o consumo de energia quando o nó sensor está dormente.

Os três principais componentes do MOS são as camadas COM, DEV e NET. A camada DEV oferece uma interface para acessar dispositivos periféricos, tais como sensores e memória Flash. A camada COM provê interface aos dispositivos de comunicação, como o transceptor do rádio. Por sua vez, a camada NET permite que os protocolos de rede sejam abstraídos da lógica do programa. Ela também permite que múltiplos protocolos de roteamento sejam registrados e implementados (MANTIS, 2007).

Segundo Stojmenovic (2005), o MANTIS também apresenta um *framework* que permite um alto nível de integração entre *drivers* de dispositivos, protocolos de rede e o próprio S.O. As principais vantagens do MANTIS são: a abstração de problemas de controle e mecanismos de tolerância a falhas isolados das aplicações.

3.2.1.2 MagnetOS

Outro sistema operacional comum em RSSFs é o MagnetOS (MAGNETOS, 2008), desenvolvido com o intuito de suportar facilmente aplicações para redes de sensores e fornecer abstração às mesmas. É caracterizado por adaptar os recursos disponíveis dos nós às mais diversas aplicações, além de apresentar conservação de energia e possuir escalabilidade para redes de grande porte.

Constitui-se de uma *Java virtual machine* sobre um conjunto de nós heterogêneos, com componentes estáticos e dinâmicos. Os estáticos traduzem as instruções das aplicações para o nível de *byte-code*, mantendo a semântica original. Sohraby (2007, p. 278) explica que “os componentes dinâmicos são usados para monitoramento de aplicações, criação de objetos, invocação e migração de código”. Isto é chamado de serviço de particionamento de código pelos criadores do sistema e tal operação preserva as interfaces das classes (MAGNETOS, 2008).

Outros recursos presentes nesse S.O. são dois algoritmos – *NetPull* e *NetCenter* – que ajudam a controlar e reduzir o consumo de energia na rede, aumentando o tempo de vida útil, e uma interface aos programadores para posicionamento explícito de objetos, ou seja, distribuição dos componentes das aplicações pelos nós da rede, de forma dinâmica. Também é necessária a realocação destes componentes de acordo com possíveis alterações na rede.

Em suma, “MagnetOS provê uma imagem única do sistema sobre uma rede *ad hoc* usando particionamento transparente de código como um mecanismo e *NetPull* e *NetCenter* como políticas de migração de código. O sistema resultante executa aplicações de maneira eficiente, adaptativa, escalável, de propósito geral e independente da plataforma.” (MAGNETOS, 2008).

3.2.1.3 SOS

O SOS (SOS 2.x, 2007) é um sistema operacional construído para funcionar com os módulos desenvolvidos pelo grupo NESL - *Networked & Embedded Systems Laboratory* (NESL, 2008). O núcleo do sistema implementa troca de mensagens, memória dinâmica e carga de módulos. Esses módulos de *software* são usados dinamicamente para criar um sistema que suporta adição, modificação e remoção dinâmica de serviços de rede.

A motivação principal deste S.O. é permitir configuração dinâmica, isto é, modificar o *software* aplicativo individualmente nos nós da rede depois que esta estiver implantada e

ativa. Assim, a rede pode ser atualizada, de acordo com a necessidade de se adicionar novos módulos aos sensores ou remover módulos inativos ou desnecessários (SOS 2.x, 2007).

Outro objetivo é fornecer ao programador serviços comuns em sistemas operacionais tradicionais, aumentando a flexibilidade e confiabilidade dos aplicativos e do sistema como um todo e permitindo a implantação de sistemas heterogêneos. O SOS também suporta plataformas de outros fabricantes, como boa parte da família Mica Mote e Moteiv.

Algumas das vantagens apresentadas são: programas escritos em linguagem C, com compilador padrão e depurador GDB; suporte do *kernel* para coleção de lixo da memória e escalonamento de tarefas com prioridade; suporte a simulações em computadores, usando aplicativo específico.

3.2.1.4 YATOS

O YATOS, sigla de “*Yet Another Tiny Operating System*” (Almeida et al, 2004), é um sistema operacional desenvolvido especificamente para nós sensores de redes de sensores sem fio. Foi inicialmente criado para o nó sensor BEAN, sendo o primeiro sistema operacional brasileiro para tal tipo de rede. O YATOS atende aos requisitos impostos pelas RSSFs, mapeia eventos em tarefas, é dirigido por eventos e fornece uma interface de programação de aplicativos (API).

Outro fator é que ele reúne características importantes para o projeto de um S.O., as quais não se encontram totalmente em praticamente nenhum dos S.O.s existentes. Segundo Almeida et al (2004), são elas:

- O sistema é dirigido por eventos, ou seja, pode entrar no modo de menor consumo de energia sempre que possível. Ao invés de realizar espera ocupada, a tarefa pode dormir e ser acordada por um evento que notifica a liberação do recurso, o que reduz drasticamente o consumo de energia.
- Ocupa o mínimo da memória de um nó sensor, sobrando mais espaço para o código das aplicações e os dados coletados.
- Consome pouca energia, visto que possui mecanismos que permitem ao processador mudar o modo de operação de energia, como diminuir a frequência do rádio quando uma computação estiver sendo realizada.
- Oferece multitarefa baseada em prioridade, se for necessária a execução de duas ou mais tarefas distintas simultaneamente.

- O S.O. é modular, facilitando a manutenção e o desenvolvimento de novos componentes e a gerência dos mesmos.
- O YATOS foi desenvolvido em C, com o intuito de facilitar o uso e a integração com aplicações para os programadores.

3.2.1.5 RETOS

O RETOS (RETOS, 2007) é um sistema operacional desenvolvido pelo grupo MOBED (MOBED, 2007), da Universidade de Yonsei, para redes de sensores sem fio. O S.O. fornece: uma interface de programação *multithread*; resiliência, ou robustez, do sistema, que permite ao sensor funcionar eficientemente, livre de erros das aplicações, sem *hardware* extra; expansão do *kernel* com re-configuração dinâmica; e abstração de rede orientada para RSSFs, ou seja, garante ambientes de programação independentes para desenvolvedores do núcleo do sistema, da camada de rede ou de aplicativos.

A robustez do sistema é alcançada utilizando-se duas técnicas: checagem de código da aplicação, que usa um *software* para impedir as aplicações do usuário de acessar parte da memória fora dos limites estipulados; e modo de operação duplo, que utiliza uma pilha para aplicações do usuário e outra pilha para tratamento de interrupções e chamadas de sistema (CHA et al, 2007).

Algumas das técnicas usadas para implementar *multithreading* são: reduzir a memória usada pelo *kernel*; reduzir o consumo de energia segundo um temporizador variável; realizar escalonamento de tarefas pela ocorrência de eventos. Por último, a abstração da rede é suportada por uma arquitetura de protocolos apropriada que permite o mínimo de modificação às outras camadas e o máximo de funcionalidade dos módulos de rede (CHA et al, 2007).

3.2.2 Sistema operacional escolhido: TinyOS

O sistema operacional TinyOS (TINYOS, 2008) foi desenvolvido especialmente para sensores com poucos recursos e permite às aplicações acessarem diretamente o *hardware*, quando isso é necessário. As duas propostas iniciais eram garantir fluxo de dados concorrente entre os dispositivos e prover componentes modularizados com pouco processamento e sem sobrecarga de armazenamento, ou seja, com o mínimo de requerimento de *hardware*.

A linguagem utilizada é o nesC – “*Network Embedded Systems C*” -, com sintaxe semelhante ao C, mas com alguns recursos a mais para suportar a estrutura e o modelo de

execução das aplicações do TinyOS. Ela foi criada para substituir a linguagem C, utilizada nas primeiras versões do S.O., permitir uma verificação estrita do código em tempo de compilação e facilitar o desenvolvimento dos componentes (Stojmenovic, 2005). Outra causa é a otimização, visto que o TinyOS não suporta alocação dinâmica de memória.

Para suportar aplicações concorrentes nos dispositivos usando pouco processamento e pouca quantidade de memória, utiliza um modelo baseado em eventos. Quando o processador está ocioso, ele mantém o processo em estado de dormência. Quando o processador torna-se disponível, é feito um escalonamento das tarefas a serem executadas pelo escalonador.

Na grande maioria dos sistemas operacionais existentes, aloca-se uma pilha separada para cada tarefa em execução, onde serão armazenados registros e variáveis locais. Contudo, prevendo os poucos recursos disponíveis, os projetistas do TinyOS optaram por um modelo de execução que determina que as múltiplas tarefas de uma aplicação compartilhem a mesma pilha. Assim, uma tarefa deve ser executada por completo antes de liberar o processador e a pilha da memória a outra tarefa. Nesse caso, a tarefa deve gravar seu estado na memória global antes que os tratadores de evento façam o escalonamento (Stojmenovic, 2005).

O TinyOS oferece suporte a dezenas de plataformas e placas de sensores. É amplamente usado na comunidade acadêmica para simulações, desenvolvimento e testes de algoritmos e protocolos. Muitos grupos atuam em conjunto para melhorar ainda mais esse sistema e estabelecer padrões em redes de sensores, num ambiente de código livre.

Rossetto (2007, p. 15) afirma que “o TinyOS pode ser definido como um *framework* de programação para redes de sensores, com um conjunto de componentes que permite construir um sistema operacional específico para cada aplicação”.

As principais desvantagens apresentadas são as dificuldades iniciais nas etapas de instalação e configuração e a complexidade a qual é exposto o desenvolvedor da aplicação. Isto significa que é necessário ao programador escrever seu código em relativo “baixo nível”. Porém, podem ser citadas diversas vantagens que justificam a utilização deste S.O.: pouco código e pequena quantidade de dados necessários; permite uma modularidade eficiente; alta velocidade na propagação de eventos e na troca de contexto.

Além dos fatores já citados, o principal motivo pela escolha deste sistema operacional é a total compatibilidade e suporte às aplicações desenvolvidas para o módulo TelosB (Crossbow, 2008) - o equipamento definido para ser usado – e outros do mesmo fabricante, que possam despertar interesse para aquisição no futuro.

Por último, ele ainda se adapta aos recursos existentes no nodo sensor no qual está instalado, de forma que, mesmo o TelosB não sendo o módulo que possui mais recursos de

hardware, pode ser utilizado perfeitamente na rede de sensores sem fio a ser implantada, com a melhor relação custo/benefício encontrada.

A tabela 3.2 mostra as principais características apresentadas pelos sistemas operacionais pesquisados neste trabalho.

Tabela 3.2: Comparação entre as principais características dos sistemas operacionais

SISTEMA OPERACIONAL	CARACTERÍSTICAS/VANTAGENS
TINYOS	<ul style="list-style-type: none"> - orientado a eventos - código simplificado - linguagem de programação NesC - modularidade eficiente - gerência de energia - escalonador de tarefas do tipo fila sem prioridade (FIFO – <i>First In First Out</i>) - interface para diversos tipos de sensores
MANTIS (MOS)	<ul style="list-style-type: none"> - multitarefa - abstração da complexidade de controle e divisão de tempo - isolamento de tolerância a falhas das aplicações - facilidade de programação através da linguagem C - suporte a várias plataformas - baixo uso da memória pelo S.O.
MAGNETOS	<ul style="list-style-type: none"> - orientado a eventos - particionamento transparente de código - migração de código - mecanismo eficiente de conservação de energia - escalabilidade - independência da plataforma
SOS	<ul style="list-style-type: none"> - orientado a eventos - memória dinâmica - carga de módulos de <i>software</i> - configuração dinâmica da rede após implantação - facilidade de programação através da linguagem C - escalonamento de tarefas com prioridade - coleção de lixo da memória
YATOS	<ul style="list-style-type: none"> - multitarefa e orientado a eventos - baixo consumo de energia - ocupação mínima da memória - modularidade - facilidade de programação através da linguagem C
RETOS	<ul style="list-style-type: none"> - multitarefa - robustez do sistema, separando o <i>kernel</i> das aplicações dos usuários - abstração aos programadores dos detalhes entre as várias camadas da rede - escalonamento de tarefas de acordo com a ocorrência de eventos - expansão do <i>kernel</i> com re-configuração dinâmica

A Figura 3.11 explicita as etapas envolvidas no desenvolvimento de uma aplicação no sistema operacional TinyOS, a partir da linguagem nesC.

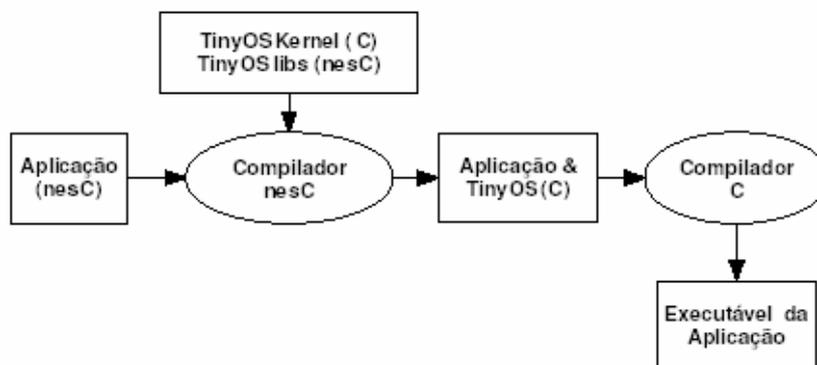


Figura 3.11 – Etapas do desenvolvimento de uma aplicação no TinyOS. – Fonte: Ruiz et al (2004, p. 40).

A documentação da versão 1.1 do TinyOS é vasta e esclarecedora, visto que é o S.O. mais utilizado nesse tipo de rede, em projetos de mais de 500 grupos de pesquisa e companhias, segundo o site oficial (TINYOS, 2008). A versão 2.0 ainda está em fase de aprimoramento e tende a ser mais robusta, eficiente, flexível e simples para dar suporte a novas plataformas.

Tentou-se instalar, num primeiro momento, a versão 1.1.0 do TinyOS no sistema *Windows*. Para isto, são necessários alguns acessórios, como o *Graphviz*, que permite a representação e visualização abstrata de grafos e redes; e os pacotes de configuração do Java. Nesse caso, também é indispensável o aplicativo *Cygwin*, um emulador que permite rodar uma aplicação *Linux* dentro do *Windows*.

Entre os problemas encontrados, está o pacote *tools* que seria extraído durante o processo de instalação do S.O. Após o término da instalação a partir dos pacotes do tipo *rpm*, notou-se que o pacote citado não havia sido extraído de forma correta. Isto aconteceu novamente, mesmo com a instalação de outra versão dos pacotes. Pode-se verificar o problema na Figura 3.12.

Como alternativa, encontrou-se o Xubuntu (TOILERS, 2007), lançado em julho de 2007. Ele simplifica a instalação do TinyOS usando um *live CD* do *Linux* e consiste na agregação do sistema *Xubuntu Debian* e do TinyOS 2.x, incluindo todos seus pacotes e repositórios das versões anteriores do S.O. escolhido.

```

/opt/tyinos-1.x
bash: /opt/tyinos-1.x/tools/java/javapath: No such file or directory
Patrick@PATRICK ~
$ cd ../../
Patrick@PATRICK /
$ ls
Thumbs.db  bin  cygwin.bat  cygwin.ico  etc  home  lib  opt  tmp  usr  var
Patrick@PATRICK /
$ cd tmp
Patrick@PATRICK /tmp
$ rpm --ignoreos -ivh tinyos-tools-1.1.0-1.cygwin.i386.rpm
Preparing... ##### [100%]
package tinyos-tools-1.2.2-1 (which is newer than tinyos-tools-1.1.0-1)
is already installed
Patrick@PATRICK /tmp
$ cd ../opt/tyinos-1.x/
Patrick@PATRICK /opt/tyinos-1.x
$ ls
CUS  ChangeLog  INTEL-LICENSE.txt  Makefile  apps  beta  contrib
Patrick@PATRICK /opt/tyinos-1.x
$ Makefile
all: not found
cd: can't cd to tools
$
$
Patrick@PATRICK /opt/tyinos-1.x
$

```

Figura 3.12 – Instalação do TinyOS com o Cygwin Bash Shell

Capítulo 4

CONCLUSÃO

A tecnologia das redes de sensores sem fio é relativamente recente e as pesquisas nessa área estão em processo de expansão e aperfeiçoamento, pois estas surgem como uma solução interessante em diversas situações do cotidiano, devido às condições de implantação, uso e manutenção e a relação custo/benefício.

Este trabalho exigiu uma pesquisa extensa e abrangente sobre os conceitos, fundamentos e características das redes de sensores sem fio. Foram abordados, inicialmente, os seguintes tópicos: aplicações, equipamentos, pilhas de protocolos, padrões, sistemas operacionais e segurança.

A etapa seguinte consistiu em uma pesquisa mais concentrada e aprofundada sobre os equipamentos de *hardware* necessários para a implantação de uma rede de sensores, com a finalidade de definir o módulo/plataforma que mais se adequasse aos requisitos e às limitações do projeto.

Ainda no desenvolvimento do trabalho, pesquisaram-se especificamente os sistemas operacionais mais apropriados para dar suporte a qualquer tipo de aplicação que possa fazer uso da estrutura da rede proposta. Assim, foi escolhido o S.O. mais propício para funcionar em conjunto com o equipamento definido.

Pode-se concluir, ao final deste trabalho, que o mesmo cumpriu com os objetivos propostos e oferecerá uma contribuição importante ao projeto “Uso de telemetria para transmissão de dados de desempenho de máquinas agrícolas visando o gerenciamento”. Facilitará o processo inicial de implementação da rede, pois abordou temas relevantes e indispensáveis para tal. Portanto, caberá aos executores do projeto se preocupar mais com a definição do sistema de telemetria e a parte de realização dos testes, para que a tese seja confirmada.

REFERÊNCIAS BIBLIOGRÁFICAS

AIOFFI, W. M.; MATEUS, G. R. **Otimização do controle de densidade com sorvedouros móveis em redes de sensores.** In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 25., 2007, Belém, PA. **Anais...** 2007. 1 CD-ROM.

ALMEIDA, V. C. de; VIEIRA, L. F. M.; VITORINO, B. A. D.; VIEIRA, M. A. M.; NACIF, J. A.; FERNANDES, A. O.; SILVA, D. C. da; COELHO, C. N. Jr. **Sistema Operacional YATOS para Redes de Sensores sem Fio.** 2004. Disponível em: <<http://homepages.dcc.ufmg.br/~lfvieira/yatos.html>> - Último acesso em Janeiro de 2008.

BARBOSA, T. M. de A.; SENE, I. G. Jr.; CARVALHO, H., S.; DA ROCHA, A. F.; NASCIMENTO, F. A. de O. **Arquitetura de software para redes de sensores sem fios: a proeminência do middleware.** In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 25., 2005, São Leopoldo, RS. **Anais eletrônicos...** São Leopoldo: UNISINOS, 2005. Disponível em: <<http://www.sbc.org.br/bibliotecadigital/download.php?paper=135>> - Último acesso em Novembro de 2007.

BRAGA, T. R. M. **Aplicações para RSSFs** – Disponível em: <<http://homepages.dcc.ufmg.br/~thaisrb/sensores.html>> - Último acesso em Novembro de 2007.

BTNODES – **A Distributed Environment for Prototyping Ad Hoc Networks.** – Disponível em <<http://www.btnode.ethz.ch>> - Último acesso em Dezembro de 2007.

CABRINI, F. H.; FLORINDO, I. R.; KOFUJI, S. T. **Monitoramento de temperatura utilizando rede de sensores sem fio em ambientes controlados.** – Disponível em: <http://www.pad.lsi.usp.br/humanlab/trabalhos/Contecsi_2006_01.pdf> - Último acesso em Novembro de 2007.

CHA, H.; CHOI, S.; JUNG, I.; KIM, H.; SHIN, H.; YOO, J.; YOON, C. **RETOS: Resilient, Expandable, and Threaded Operating System for Wireless Sensor Networks.** Abril de 2007. Disponível em: <<http://retos.yonsei.ac.kr/publications/ipsn2007cha.pdf>> - Último acesso em Janeiro de 2008.

CROSSBOW – **Wireless Sensor Networks.** Disponível em: <<http://www.xbow.com>> - Último acesso em Janeiro de 2008.

DUST NETWORKS – **Embedded Wireless Sensor Networking for Monitoring and Control.** Disponível em <<http://www.dust-inc.com/>> - Último acesso em Dezembro de 2007.

KARL, H.; WILLIG, A. **Protocols and architectures for wireless sensor networks.** Chichester, West Sussex, England: Wiley, 2005. 507 p.

LOUREIRO, A. A. F.; NOGUEIRA, J. M. S.; RUIZ, L. B.; MINI, R. A. de F.; NAKAMURA, E. F.; FIGUEIREDO, C. M. S. **Redes de sensores sem fio.** In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 21., 2003, Natal, RN. **Anais...** 2003. 1 CD-ROM.

MACEDO, D. F.; CORREIA, L. H. A.; SANTOS, A. L. dos; LOUREIRO, A. A. F.; NOGUEIRA, J. M. S. **Avaliando aspectos de tolerância a falhas em protocolos de roteamento para redes de sensores sem fio.** In: WORKSHOP DE TESTES E TOLERÂNCIA A FALHAS, 6., 2005, Fortaleza, CE. **Anais eletrônicos...** Fortaleza: UFC, 2005. Disponível em: <<http://twiki.im.ufba.br/pub/WTF/ArtigosAceitos/8895.pdf>> – Último acesso em Novembro de 2007.

MAGNETOS – **The MagnetOS Operating System.** Disponível em: <<http://www.cs.cornell.edu/People/egs/magnetos/>> - Último acesso em Janeiro de 2008.

MANTIS - **MANTIS: HomePage.** Disponível em: <<http://mantis.cs.colorado.edu/>> - Último acesso em Janeiro de 2008.

MOBED – **Mobile & Embedded System.** Setembro de 2007. Disponível em: <<http://mobed.yonsei.ac.kr/>> - Último acesso em Janeiro de 2008.

NESL – **Networked & Embedded Systems Laboratory.** Disponível em: <<http://nesl.ee.ucla.edu/>> - Último acesso em Janeiro de 2008.

PEREIRA, M. R.; AMORIM, C. I. de; CASTRO, M. C. S. de. **Tutorial sobre Redes de Sensores.** Cadernos do IME UERJ- Série Informática – Vol. 14 - Junho 2003 - Disponível em <<http://www.ime.uerj.br/cadernos/cadinf/vol14/>> - Último acesso em Novembro de 2007.

RETOS - **Resilient, Extensible, and Threaded OS for WSN.** Junho de 2007. Disponível em: <<http://retos.yonsei.ac.kr/>> - Último acesso em Janeiro de 2008

REZENDE, J. F. **Redes de sensores sem fio.** – Março de 2004. Disponível em: <<http://www.gta.ufrj.br/~rezende/cursos/eel879/trabalhos/rssf1/>> - Último acesso em Novembro de 2007.

ROSSETTO, S.; COSTA, F. C. A. da; SACRAMENTO, V.; ALEXANDRE, L.; NEVES, R.; PINHEIRO, S. **Redes de sensores sem fio: visão geral e um exemplo de implementação.** In: ESCOLA REGIONAL DE REDES DE COMPUTADORES, 5., 2007, Santa Maria. **Anais...** Santa Maria: UFSM, 2007. 1 CD-ROM.

RUIZ, L. B.; CORREIA, L. H.; VIEIRA, L. F. M.; MACEDO, D. F.; NAKAMURA, E. F.; FIGUEIREDO, C. M. S.; VIEIRA, M. A. M.; BECHELANE, E. H.; CAMARA, D.; LOUREIRO, A. A. F.; NOGUEIRA, J. M. S.; SILVA, D. C. Jr. **Arquiteturas para redes de sensores sem fio.** In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 22., 2004, Gramado, RS. **Anais...** 2004. 1 CD-ROM.

S3C INC – **Sensors Silicon Systems.** Disponível em: <<http://www.s3cinc.com>> - Último acesso em Dezembro de 2007.

SENSORNET – **Arquitetura, Protocolos, Gerenciamento e Aplicações em Redes de Sensores Sem Fio.** Abril de 2004. Disponível em: <<http://www.sensornet.dcc.ufmg.br/pt/index.php>> - Último acesso em Janeiro de 2008.

SHOCKFISH – **Shockfish SA, TinyNode** – Março de 2006 – Disponível em: <<http://www.shockfish.com/>> - Último acesso em Janeiro de 2008.

SHOREY, R.; ANANDA, A.; CHAN, M. C.; OOI, W. T. **Mobile, wireless, and sensor networks**: technology, applications, and future directions. Hoboken, New Jersey: Wiley, 2006. 456 p.

SOHRABY, K.; MINOLI, D.; ZNATI, T. **Wireless sensor networks**: technology, protocols, and applications. Hoboken, New Jersey: Wiley, 2007. 307 p.

SOS 2.x – **SOS Embedded Operating System** – Disponível em: <<https://projects.nesl.ucla.edu/public/sos-2x/doc/>>- Último acesso em Janeiro de 2008.

STOJMENOVIC, I. **Handbook of sensor networks**: algorithms and architectures. Hoboken, New Jersey: Wiley, 2005. 531 p.

TINYOS – **TinyOS Community Fórum: An open-source OS for the networked sensor regime**. Disponível em: <<http://www.tinyos.net/>> - Último acesso em Janeiro de 2008.

TOILERS - **XubunTOS < Public < TWiki**. Disponível em: <<http://toilers.mines.edu/Public/XubunTOS>> - Último acesso em Janeiro de 2008.

UAMPS – **MIT PROJECT: μ -Adaptative Multi-domain Power aware Sensors**. Disponível em: <<http://www-mtl.mit.edu/researchgroups/icsystems/uamps/>> - Último acesso em Janeiro de 2008.

VIDAL, R. S.; FARIA, T. G. **Análise dos protocolos LEACH e LEACH-C para as redes de sensores sem fio** (2007). In: ESCOLA REGIONAL DE REDES DE COMPUTADORES, 5., 2007, Santa Maria. **Anais...** Santa Maria: UFSM, 2007. 1 CD-ROM.

WORLDSSENS – **Development framework for sensor networks**. Disponível em: <<http://worldsens.citi.insa-lyon.fr/>> - Último acesso em Janeiro de 2008.