FABIANO MARIO RODRIGUES FIALHO

UM AMBIENTE DE GERÊNCIA PARA DISPOSITIVOS SENSORES EM REDES DE SENSORES SEM FIO

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae.*

VIÇOSA
MINAS GERAIS - BRASIL
2013

Ficha catalográfica preparada pela Seção de Catalogação e Classificação da Biblioteca Central da UFV

T
F438a
2013

Fialho, Fabiano Mário Rodrigues, 1983-
Um ambiente de gerência para dispositivos sensores em redes de sensores sem fio / Fabiano Mário Rodrigues Fialho. – Viçosa, MG, 2013.
x, 83 f. : il. (algumas color.) ; 29cm.

Inclui anexos.
Orientador: Carlos de Castro Goulart.
Dissertação (mestrado) - Universidade Federal de Viçosa.
Referências bibliográficas: f.69-75.

1. Redes de sensores sem fio. 2. Redes de computadores - Protocolos. I. Universidade Federal de Viçosa. Departamento de Informática. Programa de Pós-Graduação em Ciência da Computação. II. Título.

CDD 22. ed. 004.6

FABIANO MARIO RODRIGUES FIALHO

UM AMBIENTE DE GERÊNCIA PARA DISPOSITIVOS SENSORES EM REDES DE SENSORES SEM FIO

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae.*

APROVADA: 25 de julho de 2013.

Thais Regina de Moura Braga

Mauro Nacif Rocha
(Coorientador)

Carlos de Castro Goulart
(Orientador)

AGRADECIMENTOS

Primeiramente, gostaria de agradecer a Deus, por me dar sabedoria, força e coragem para enfrentar os desafios da vida.

Aos meus pais José Lelis e Vera, que sempre me apoiaram e acreditaram no meu potencial, além de sempre terem a palavra certa para ajudar nos momentos mais difíceis, e por todo o incentivo dado para ingressar e concluir o mestrado.

À minha esposa Lorena, pelo companheirismo, carinho e compreensão pelos momentos ausentes devido à realização do meu mestrado, além de sua grande colaboração para a confecção desta dissertação.

Ao meu filho Arthur, que está por vir, que desde o momento em que eu soube de sua existência, só ganhei mais forças para conseguir realizar este trabalho e vencer os desafios que apareceram.

À minha avó Bilú (*in memorian*) que sempre me acolheu e possibilitou que muitas das minhas conquistas fossem possíveis. Eu tenho certeza de que onde você estiver, está olhando por nós aqui.

Ao meu irmão Leandro, por toda sua amizade e companhia.

Aos meus sogros Zélia e Antônio, por toda a força dada desde o momento em que os conheci.

Ao meu cunhado Rafael, que além da amizade, também contribuiu muito para que fosse possível concluir o mestrado, não economizando esforços para me ajudar nos conteúdos nos quais eu tinha dificuldade, mesmo quando era necessário atravessar a madrugada.

Ao meu orientador Carlos Goulart pela idéia do trabalho de pesquisa que foi realizado, pelo apoio em todos os momentos de dificuldade, pela disponibilidade em me receber o mais rápido possível sempre que houveram dúvidas e acima de tudo pela amizade demonstrada. Sem dúvida nenhuma grande parte deste trabalho só foi possível devido ao seu auxílio.

Aos diretores da DTI, Luiz Carlos e Michelini, por todas as palavras de apoio e incentivo para a realização do mestrado.

Aos colegas de serviço da DTI por compreenderam os momentos de ausência e me ajudarem a vencer esta etapa.

Aos amigos, Waldir Denver e Rafael Machado, por fornecerem todos os equipamentos e ajuda com configuração dos mesmos. Muito deste trabalho também só foi possível devido à ajuda de vocês.

Ao Altino pelos conselhos e presteza em ajudar a resolver qualquer problema que tive durante o período do mestrado.

A todos os demais familiares e amigos pelo apoio em todas as fases da minha vida.

SUMÁRIO

LISTA DE FIGURAS

RESUMO

FIALHO, Fabiano Mario Rodrigues, M.Sc., Universidade Federal de Viçosa, julho de 2013. **Um ambiente de gerência para dispositivos sensores em redes de sensores sem fio**. Orientador: Carlos de Castro Goulart. Coorientadores: Mauro Nacif Rocha e Ricardo dos Santos Ferreira.

As Redes de Sensores sem Fio (RSSF) criam oportunidades para diversas aplicações, tais como monitoramento ambiental, gerenciamento de infraestrutura, segurança pública, entre outras. Mesmo com os nós sensores possuindo pouca capacidade individual, a colaboração entre eles pode realizar a execução de uma tarefa complexa. Devido à grande gama de aplicações, várias pesquisas têm sido realizadas para tornar possível a utilização das RSSFs em situações reais. A escassez de recursos dos nós sensores determinam uma mudança de paradigmas em relação às redes tradicionais. Então, para resolver algumas dessas questões, neste trabalho foi proposto um ambiente de gerência para RSSF, consistindo de uma extensão de uma Base de Informação de Gerência (MIB) e um agente proxy de gerência, além do uso do protocolo de gerência SNMP (*Simple Network Management Protocol*). Tal ambiente permite a gerência remota de uma RSSF através da Internet, a partir de uma estação de gerência usando ferramentas com interface baseada na Web. O agente proxy de gerência faz o mapeamento das mensagens SNMP para mensagens no formato que os nós sensores que não são gerenciáveis via SNMP possam entender. A implementação do modelo proposto foi feita utilizando sensores de baixo custo e com arquitetura livre, tanto de hardware quanto de software, e ferramentas de software livre. Essa implementação permitiu verificar a viabilidade de utilização do modelo proposto para a gerência dos nós sensores disponíveis no mercado atualmente.

ABSTRACT

FIALHO, Fabiano Mario Rodrigues, M.Sc., Universidade Federal de Viçosa, july, 2013. **A management environment for sensor devices in wireless sensor networks**. Adviser: Carlos de Castro Goulart. Co-Advisers: Mauro Nacif Rocha and Ricardo dos Santos Ferreira.

Wireless Sensor Networks (WSN) create opportunities for several applications such as environmental monitoring, infrastructure management, public safety, among others. Even with sensor nodes having small individual capacity, the cooperation among them can carry out the execution of a complex task. Due to the large range of applications, many research works have been done to make it possible to use WSNs in real situations. The limited resources of sensor nodes determine a change on paradigm with respect to traditional networks. To solve some of these issues, this work proposes a management environment for WSN, consisting of an extension of a Management Information Base (MIB), a proxy management agent, and the use of SNMP (Simple Network Management Protocol). Such an environment allows remote management of a WSN through the Internet, using a web based management station. The proxy management agent maps the SNMP messages into messages in the format understandable by sensor nodes, which are not SNMP-manageable.

The implementation of the proposed model was made using low-cost sensors, free hardware and software architecture, and open source tools. The implementation has shown the feasibility of using the proposed model for the management of sensor nodes available on the market today.

1. Introdução

1.1. O problema e sua importância

O gerenciamento de redes pode ser entendido como o processo de controlar uma rede de computadores de tal modo que seja possível maximizar sua eficiência e produtividade [STALLINGS, 1999]. Esse processo engloba um conjunto de funções integradas que podem estar em uma máquina ou em várias, dispersas a milhares de quilômetros, e em diferentes organizações. É importante observar que, com estas funções, pode-se controlar uma rede de computadores e seus serviços, provendo mecanismos de monitoração, análise e controle dos dispositivos e recursos da mesma [HOLANDA FILHO, 1998].

A gerência de rede compreende a monitoração, análise e resolução de eventuais problemas, dentre outras atividades necessárias para a manutenção de uma rede com qualidade de serviços adequada aos objetivos dos sistemas de informação [MELCHIORS, 1999].

Em virtude da relação custo-benefício, a utilização de redes de computadores vem crescendo de forma bastante significativa, atuando simultaneamente em conjunto das tecnologias de telecomunicações e informática. Entretanto, devido ao grande aumento de novos dispositivos que se comunicam através da rede, o monitoramento de redes de computadores é uma tarefa muito complexa, afetando o gerenciamento de redes em diversos aspectos como, por exemplo, a escalabilidade, sobrecarga dos dispositivos e canal de comunicação [STALLINGS, 1999] [KONA; XU, 2002].

Paralelamente, avanços na fabricação de circuitos integrados tornaram possível a integração de tecnologia de micro-sensores, computação de baixo consumo e comunicação sem fio em um sistema compacto [AKYILDIZ et al., 2002]. As redes formadas por esses dispositivos, denominadas Redes de Sensores Sem Fio (RSSFs), criam oportunidades para diversas aplicações, tais como monitoramento ambiental, gerenciamento de infra-estrutura, segurança pública e de ambientes em geral, transporte e controle militar [ESTRIN et al., 2000][SRIVASTAVA et al., 2001]. Embora os dispositivos sensores possuam individualmente pouca capacidade, a colaboração entre eles pode gerar a execução de uma grande tarefa.

Dada a gama vasta de potenciais aplicações, um esforço grande de pesquisa tem sido despendido em direção a tornar as RSSFs possíveis de serem utilizadas na prática. Um dos objetivos tem sido estudar o gerenciamento dessas redes que, principalmente por causa da sua propriedade *adhoc*, já que os elementos que a compõem não necessitam de um concentrador central para implementar a comunicação entre esses nós, e da escassez de recursos, imprimem uma mudança de paradigma em relação às redes tradicionais.

Esta limitação de recursos restringe a capacidade do hardware e da reserva de energia destes elementos. Esse tipo de rede é aplicável em larga escala e em ambientes remotos ou de difícil acesso, o que dificulta e, em alguns casos, impossibilita a manutenção local destes equipamentos. Para se controlar o uso dos recursos visando a melhoria da produtividade da rede, técnicas de gerenciamento podem ser adotadas. Porém, elas devem ser escolhidas com cuidado de forma que os benefícios do gerenciamento não sejam anulados com o consumo extra de recursos [SILVA, 2006].

Para que seja possível a utilização eficiente do verdadeiro potencial das RSSFs, essas devem ser conectadas a redes baseadas no protocolo IP, onde residem a maioria dos recursos de informação existentes.

A integração das RSSFs com redes IP tem sido estimulada por vários fatores. Primeiro, as redes IP permitem o uso da infra-estrutura e recursos de informação existentes. Segundo, as tecnologias baseadas em IP, em conjunto com suas ferramentas de diagnóstico, gerenciamento e delegação, já existem, e foram devidamente testadas e aprovadas. Terceiro, os dispositivos baseados em IP podem ser mais facilmente conectados a outras redes IP sem depender de *gateways* de tradução [CHAUDHRY et al, 2010].

O gerenciamento existente em redes IP, utilizando-se principalmente o SNMP (*Simple Network Management Protocol*) fornece ferramentas de diagnóstico e gerenciamento. Entretanto não podemos implantar o SNMP diretamente nas RSSFs devido à sua limitação de recursos dos dispositivos atuais. Então é essencial termos um sistema de gerência que seja leve o suficiente para ser implantado nas RSSFs e ainda seja compatível com o SNMP [CHAUDHRY et al, 2010].

Outra motivação deste trabalho foi a necessidade de monitoramento ambiental da Mata do Paraíso em Viçosa-MG, utilizando sensores. A mata do paraíso está localizada a aproximadamente 6 km da área central da Universidade Federal de Viçosa e possui uma área de 194 hectares [DEF, 2013]. Este monitoramento é realizado por até

4 nós sensores, e é importante que haja alguma forma de buscar essa informação remotamente, devido à dificuldade de acesso ao local, e para evitar uma grande movimentação no ambiente o que pode inviabilizar o acompanhamento ou a compreensão do fenômeno que se deseja estudar. Devido à distância entre a Mata do Paraíso e a UFV é necessário utilizar alguma rede com capacidade maior de alcance, de forma que os dados possam ser disponibilizados na Internet.

Considerando as questões acima, e o fato de implementações reais de um ambiente de gerência SNMP não terem sido suficientemente testadas na literatura, neste trabalho foi proposto o desenvolvimento de um proxy que fará a intercomunicação entre a rede IP e a RSSF, além da definição de uma Base de Informação de Gerência (*Management Information Base /* MIB). Posteriormente, foi implementado um protótipo onde testes foram conduzidos para verificar o funcionamento e a eficiência do conjunto.

1.2. Hipótese

Este trabalho assume uma hipótese que é possível utilizar o protocolo SNMP (*Simple Network Management Protocol*) para gerenciar redes de sensores sem fio.

1.3. Objetivos

Este trabalho tem como objetivo principal a prototipação de uma RSSF utilizando tecnologia livre de hardware e software, e permitir que esta seja gerenciada utilizando o protocolo SNMP.

Para que o objetivo seja alcançado foi proposta uma extensão da MIB apresentada em [SILVA, 2005] uma vez que essa MIB foi considerada bem abrangente para o universo de RSSFs em geral.

Com o intuito de permitir a validação da MIB foi proposto um ambiente de gerência baseado na arquitetura de um *proxy* SNMP que fará a comunicação entre a rede tradicional TCP/IP e a RSSF e permitirá o gerenciamento da última. A MIB deve ser implementada no *proxy*.

1.4. Organização do trabalho

O trabalho está organizado como se segue. O capítulo 2 apresenta alguns conceitos fundamentais de gerência de redes de dados. Além da fundamentação teórica inicial sobre gerência de redes, é abordado também o protocolo SNMP. No capítulo 2, também é apresentado o conceito de RSSF, identificando os principais elementos da rede, além dos paradigmas necessários para aplicação da gerência da rede de dados em RSSFs.

No capítulo 3 são apresentadas as principais propostas existentes de gerência de RSSF. São apresentadas tanto soluções nas quais é proposto que o gerenciamento seja realizado pela própria RSSF, quanto propostas que são baseadas na utilização do SNMP.

No capítulo 4 é apresentado o modelo do ambiente de gerência utilizado neste trabalho, além da descrição da MIB e dos componentes restantes do modelo que são nó sensor, agente *proxy* e servidor.

No capítulo 5 temos a descrição da implementação do modelo, sendo detalhados cada um dos elementos descritos no capítulo 4, além da forma de funcionamento e comunicação entre eles. Neste capítulo também são descritos a estrutura do ambiente de testes montado e os resultados encontrados.

No capítulo 6 são apresentadas as conclusões de acordo com os resultados obtidos no capítulo 5, além de comentários sobre possibilidades de trabalhos futuros considerando o ambiente utilizado.

2. Gerência de Redes e Redes de Sensores sem Fio

A seguir serão apresentados os conceitos utilizados em diversos campos correlatos a este trabalho, com o objetivo de criar um substrato teórico necessário ao entendimento dos objetivos deste trabalho e indicar a forma como estes conceitos se inserem no contexto da gerência de RSSF.

2.1. RSSF

Avanços na tecnologia de micro-sistemas eletromecânicos (MEMS), na comunicação sem fio e em eletrônica digital tornaram possível o desenvolvimento de nós sensores de baixo custo, baixo consumo e multifuncionais que são pequenos no tamanho e que se comunicam de forma livre em curtas distâncias. Esses pequenos nós sensores, os quais são constituídos por componentes de sensoriamento, processamento de dados e comunicação, alavancaram a idéia de redes de sensores baseadas no esforço colaborativo de um grande número de nós.

Uma rede de sensores pode ser composta por um grande número de nós sensores, os quais são implantados de forma densa, ou seja bem próximos uns dos outros, tanto dentro do fenômeno a ser sensoriado ou muito próximo a ele. A posição dos nós sensores não precisa ser pré-determinada. Isso permite a implantação aleatória em áreas de difícil acesso ou em operações de recuperação de desastre. Por outro lado, esse fato implica que os protocolos e algoritmos das redes de sensores sejam capazes de realizar uma auto organização. Outra característica única das redes de sensores, é que esses possuem um processador interno, então ao invés de enviar o dado bruto, os nós sensores podem utilizar seu processamento para realizar de forma local computações simples e transmitir apenas o que foi requisitado e dados parcialmente processados.

As características descritas anteriormente garantem uma vasta gama de aplicações para a rede de sensores. Algumas das áreas de aplicação são: saúde, militar e segurança. Como exemplo, dados fisiológicos de um paciente podem ser monitorados remotamente por um médico. Os sensores também podem ser utilizados na detecção de agentes químicos estranhos no ar ou na água.

A realização dessas e de outras aplicações das redes de sensores requer técnicas de rede sem-fio *ad hoc*. Embora muitos protocolos tenham sido propostos para rede sem fio *ad hoc*, eles possuem características que não são compatíveis com o comportamento

de uma rede de sensores devido aos seguintes motivos: o número de sensores em uma rede de sensores pode ser muitas vezes maior do que o número de nós em uma rede *ad hoc*, os nós sensores são implantados de forma densa e são suscetíveis a falhas, a topologia da rede de sensores muda com frequência, as rede de sensores utilizam principalmente a difusão para envio de informações enquanto a maioria das rede *ad hoc* são baseadas em comunicações ponto a ponto, os nós sensores são limitados em energia, capacidade computacional e memória.

Uma das restrições mais importantes em uma rede de sensores é o requisito de baixo consumo de energia. Os nós sensores carregam fontes de energia limitadas e muitas vezes insubstituíveis. Assim, enquanto redes tradicionais visam alcançar um alto nível de parâmetros de qualidade de serviço (QoS) como vazão, atraso, entre outros, as redes de sensores devem ter como alvo primariamente a conservação de energia como principal parâmetro de QoS [AKYILDIZ et al., 2002].

Em RSSF, a transmissão dos nós sensores para o destino pode acontecer diretamente ao destino, ou utilizando multi saltos, como exposto nas figuras 2.1 e 2.2 [KARL, 2005].

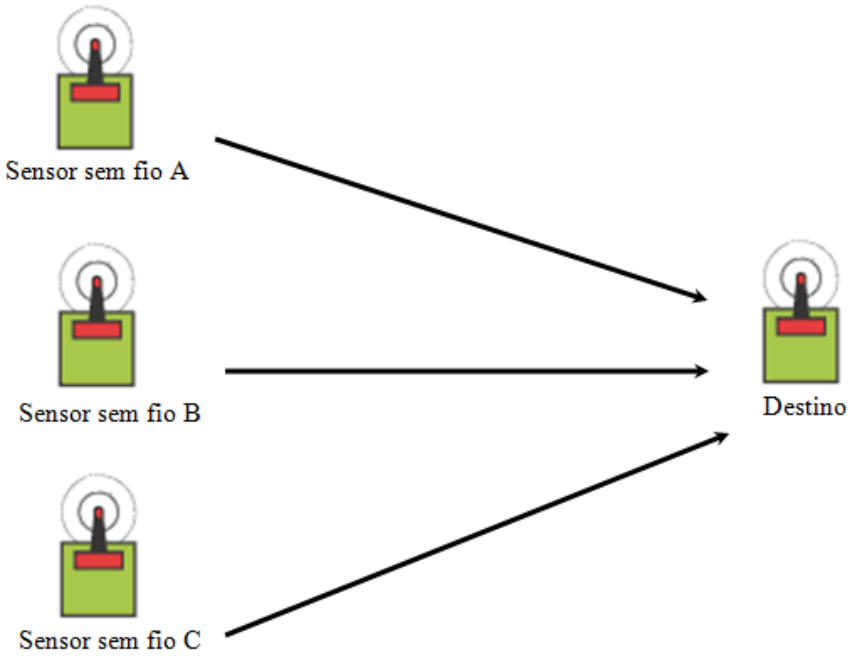

Figura 2.1 - Sensores com envio direto ao destino

A rede multi salto, exemplificada na figura 2.2, também se beneficia da variedade de caminhos possíveis, criados através de roteamento. Novos caminhos

devem também ser gerados no caso de falhas em saltos intermediários, funcionando como rotas redundantes.

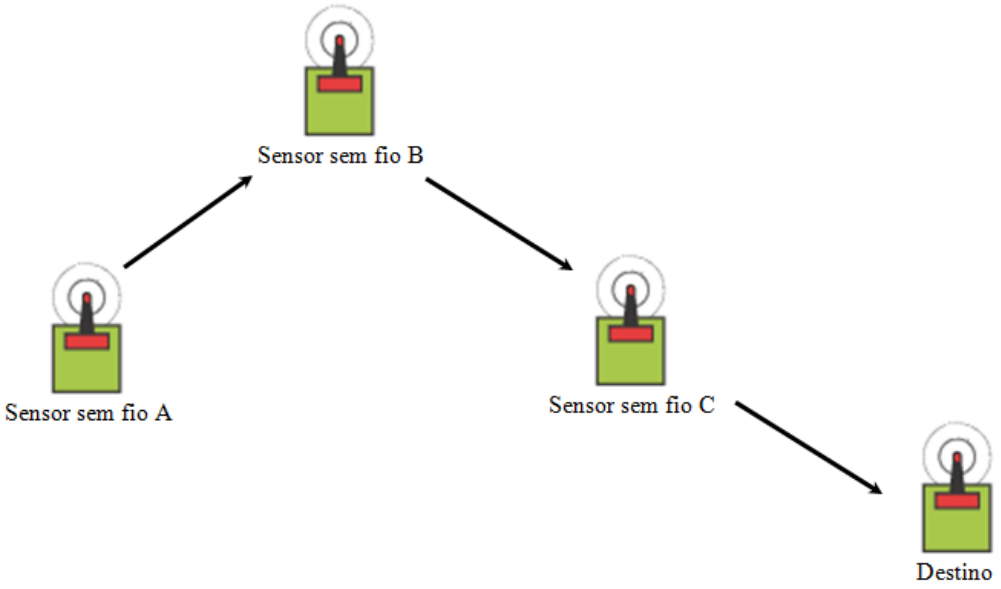

Figura 2.2 - Rede de sensores sem fio implementada com multisalto

Independentemente da forma com que se pretende implementar uma RSSF, outro ponto importante é garantir que o desempenho da rede, em suas funções básicas, deve permanecer inalterado sem ser influenciado pelo crescimento do número de sensores. Essa escalabilidade, no caso de RSSF, reflete diretamente sobre a área de cobertura. Em redes com múltiplos saltos existe a oportunidade inegável de cobrir uma área maior e manter a rede funcionando, evitando interrupções devido à falha de nós sensores. Contudo, os sensores com a inteligência necessária para realizar a tarefa de criação de diversas rotas acabam por ser mais custosos em seu desenvolvimento do que sensores preparados para o envio direto para o destino, caso não haja o interesse nem a necessidade de se economizar energia. A complexidade e o custo acabam sendo grandes problemas neste caso. As RSSFs são dependentes da aplicação e podem apresentar diferentes características.

2.1.1. Caracterização das RSSFs

Uma RSSF pode ser considerada homogênea quando todos os nós que a compõe tem características iguais de hardware, ou seja a mesma configuração de processador,

memória, bateria, transceptor e dispositivo sensor. Em contrapartida, caso a RSSF seja composta por nós com diferentes capacidades, ela é dita heterogênea.

Quanto à organização, uma RSSF pode ser plana, se não existem agrupamentos de nós. Já quando os nós se organizam em grupos a RSSF é dita hierárquica. Neste caso podem haver vários níveis de hierarquia e para cada grupo de nós deve haver um líder. Em redes heterogêneas, os nós de maior capacidade podem assumir a liderança durante todo o tempo de vida da rede. Para o caso de uma rede hierárquica homogênea, um processo para a escolha do líder deve acontecer. Novas escolhas devem ocorrer motivadas por diferentes parâmetros, podendo ser citado, nível mínimo de energia residual do líder atual. Critérios podem ser estabelecidos para determinar quais nós podem votar e quais são elegíveis. Além disso pode ser utilizado um sistema de indicação, onde o primeiro líder é indicado pela entidade global de gerenciamento que possui visão global da rede. Como cada líder tem visão local do seu grupo, este indicará o seu sucessor.

De um modo geral as RSSFs são compostas de pelo menos um ponto de acesso por onde a rede troca informações com o exterior. No caso das redes planas, o ponto de acesso é implementado em um nó sorvedouro (*sink*) também chamado de nó de monitoração. Devido ao alcance dos transceptores de rádio frequência e do consumo de energia na transmissão, os nós em uma rede plana disseminam os dados em direção ao nó sorvedouro utilizando uma comunicação multisaltos. No caso das redes hierárquicas, o ponto de acesso é implementado em estações base que em geral, não tem restrições de processamento, memória ou energia. Nas redes hierárquicas homogêneas, os líderes recebem a informação dos nós do grupo, podem realizar algum tipo de processamento, como fusão, agregação, contagem, entre outros e disseminam a informação resultante utilizando comunicação multisaltos até a estação base. Em alguns casos, se o hardware permitir, o alcance do rádio dos líderes pode ser aumentado para transmitirem com envio direto para o destino (estação base), mas com o aumento do consumo de energia. Para as redes hierárquicas heterogêneas, os nós com maior alcance de rádio, podem enviar os dados do grupo diretamente para a estação base. O fato do consumo de energia aumentar é compensado pelo fato deste possuir uma bateria com maior capacidade.

Uma RSSF é considerada estacionária quando os nós são dispostos sobre a região de monitoração e permanecem no mesmo lugar, Mesmo com este cenário, a topologia é dinâmica, uma vez que com o passar do tempo alguns nós tornam-se inativos devido ao consumo de energia, causando uma alteração na topologia. Por outro

lado, nos casos onde um ou mais nós da rede ou pontos de acesso são móveis, diz-se que a rede é móvel.

Quanto à propagação de informação dos nós para o observador, sendo este o usuário final ou entidade que está interessado nos dados coletados pela RSSF, uma RSSF pode ser contínua, quando os dados são continuamente coletados, processados, e enviados ao observador. A RSSF é considerada reativa quando enviam dados referentes a eventos que ocorrem no ambiente de monitoração. E programada, quando os nós periodicamente se desligam e despertam apenas sob condições pré-estabelecidas ou a intervalos regulares definidos pela aplicação.

Uma RSSF é dita irregular quando os nós estão aleatoriamente distribuídos sobre a área monitorada apresentando diferentes densidades. A rede é balanceada quando apresenta uma distribuição uniforme dos nós. Uma RSSF pode ser densa se houver uma alta densidade de nós por área e esparsa caso contrário. Mesmo que uma rede de sensores seja densa no início, com o tempo ela passará a ser balanceada até se tornar esparsa e morrer, ou seja, ficar impossibilitada de prover serviços dentro da qualidade especificada.

Quanto ao tamanho a RSSF pode ser considerada como pequena, caso esta tenha até cem elementos de rede, média quando possuir entre 100 e 1000 elementos de rede e grande quando possuir mais de 1000 elementos.

Considerando-se a alocação de canal, as RSSFs podem ser estáticas quando a largura de banda é dividida em partes iguais para os nós, podendo estas partes iguais serem referentes à frequência, tempo, código, espaço ou ortogonal. Cada nó recebe uma parte privada da comunicação minimizando a interferência. Já na alocação de canal dinâmica, não existe atribuição fixa de largura de banda, então os nós disputam o canal para comunicação.

Quando se trata do fluxo de informação, as RSSFs podem ser do tipo inundação (*flooding*), quando os nós fazem difusão (*broadcast*) de suas informações para seus vizinhos, que por sua vez fazem difusão desses dados até alcançar o ponto de acesso. Apesar de possuir um alto *overhead*, está imune às mudanças dinâmicas de topologia e a alguns ataques de DoS (*Denial of Service*). Nas RSSFs do tipo *multicast*, os nós fornam grupos e utilizam a comunicação de grupo (*multicast*) para comunicação entre os membros do grupo. Já no tipo *unicast*, os nós podem se comunicar diretamente com o ponto de acesso, usando protocolos de roteamento multisaltos. No modo *gossiping*, os nós sensores selecionam os nós para os quais enviam os dados. Por último temos o

modo *bargaining* onde os nós somente enviam os dados caso o nó destino manifeste interesse, neste caso existe um processo de negociação.

Ao considerarmos o processamento a cooperação entre os nós pode ser feita por infraestrutura, onde os nós sensores executam procedimentos relacionados à infraestrutura da rede como por exemplo, algoritmos de acesso ao meio, roteamento eleição de líderes, descoberta de localização e criptografia. Já na cooperação localizada, os nós sensores executam além dos procedimentos de infra estrutura, algum tipo de processamento local básico, como tradução dos dados coletados pelos sensores baseado na calibração. Finalmente, temos a cooperação por correlação, onde os nós sensores estão envolvidos em procedimentos de correlação de dados como fusão, supressão seletiva, contagem, compressão, multi-resolução e agregação [RUIZ, 2003] [NAKAMURA, 2003].

A seção a seguir faz um paralelo entre a pilha TCP/IP e exemplifica a correlação de cada uma das camadas com uma RSSF.

2.1.2. Pilha de protocolos para RSSF

Para o projeto de uma rede de sensores, o mais conveniente é considerar uma pilha de protocolos que atenda aos objetivos desse tipo de rede. Embora seja possível criar uma pilha de protocolos específica, o mais interessante é considerar uma pilha que mantenha semelhança com a pilha TCP/IP, no que se refere à sua essência. Montar a pilha TCP/IP, mesmo que minimizada, nos sensores não é prático devido às restrições já mencionadas de consumo de energia e espaço em memória. Entretanto, considerando a essência da pilha é possível identificar as funções necessárias a serem executadas. Pensando nesta questão, a plataforma Radiuino [RADIUINO, 2011] propõe uma pilha como mostrada na figura 2.3.

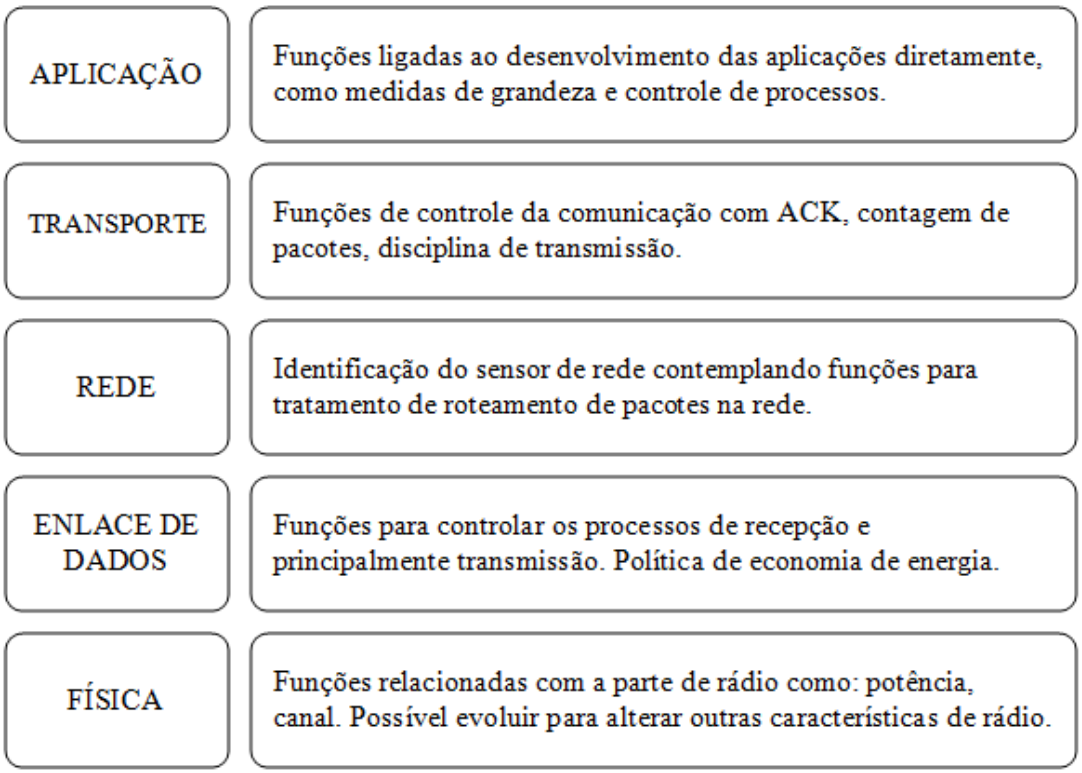

Figura 2.3 - Pilha de protocolos do Radiuino [CYRIACO, 2012]

Na figura 2.3 é possível identificar as funcionalidade básicas para cada uma das camadas. Referências recentes como [AKYILDIZ, 2010] e [DARGIE, 2010] apresentam explicitamente uma estrutura equivalente à pilha TCP/IP para RSSFs. A estratégia de RSSF deve considerar parâmetros e atributos de cada pilha de protocolo [CYRIACO, 2012]. A seguir serão tratadas as características principais de cada camada, excetuando-se a camada de transporte, que raramente é considerada em RSSFs.

2.1.2.1 Camada de Aplicação

Este camada é responsável por abstrair a topologia física da rede para a aplicação a ser atendida pelo sensor, fazendo interface com os processos a serem monitorados ou controlados. Alguns exemplos destes processos são: medida de grandezas e acionamento de dispositivos. Para a medida de grandezas são encontrados transdutores analógicos e digitais. Para o caso de transdutores analógicos é necessário a utilização de circuitos adaptadores para realizar a conversão para o dado digital. Os dados amostrados são tratados e codificados para sua transmissão. Os dados gerados pelas medidas podem necessitar de compressão em função da quantidade de

informações coletadas [AKYILDIZ, 2010]. Outra função da camada de aplicação é tratar do tipo de informação que será transmitida, como por exemplo, dados periódicos. Cada tipo de dado tem seu tratamento específico para atender aos requisitos de qualidade. A consistência das informações obtidas também pode ser verificada pela camada de aplicação. Por exemplo, ao realizar uma sequência de 5 medidas de temperatura, e verificar-se que o valor da terceira medida é muito grande, enquanto as outras permanecem com valores próximos, podemos perceber uma inconsistência de valores dependendo do tipo de ambiente e do intervalo entre as medidas. Para este caso a camada de aplicação pode evitar a transferência destes dados inconsistentes, além de permitir a implementação de algoritmos que checarão a validade dos dados medidos.

Ao se utilizar atuadores pode ser necessário, em alguns casos, verificar se de fato a atuação foi efetiva. Por exemplo, ao constatar toxicidade em algum ambiente, ventiladores devem ser acionados, e portanto seria importante verificar se, de fato, os ventiladores foram acionados, e para tanto seria necessário um sensor que meça a velocidade do ar. Este tipo de verificação é pertinente em processos críticos, mas deve ser considerado em cada acionamento [CYRIACO, 2012].

Para RSSF com múltiplos saltos, no qual vários sensores enviam informação para um destino, existe a possibilidade dos nós perto do destino terem de trafegar um grande número de pacotes. Este efeito pode ser muito ruim, uma vez que estes nós podem rapidamente consumir a energia disponível e simplesmente parar de funcionar. Existem técnicas para minimizar este tipo de efeito com a otimização da transmissão dos dados. A medida de temperatura mais uma vez pode ser considerada como exemplo, já que pode ser possível determinar que os sensores só enviem os dados caso a temperatura ultrapasse um valor ou delta de variação. Outra alternativa seria calcular uma média e somente ela ser enviada. Também podem ser utilizadas formas de agregação dos dados para que os nós intermediários somente transmitam um condensado da informação [KARL, 2005].

2.1.2.2. Camada de Rede

Essa camada é responsável pela identificação do sensor na rede e pelos algoritmos de roteamento. Esse é um dos tópicos mais investigados em redes de sensores na procura de algoritmos apropriados para o roteamento que atendam às peculiaridades dos tipos de redes. Entre os desafios dos protocolos de roteamento

podem ser mencionados consumo de energia, escalabilidade, endereçamento, robustez, topologia e atendimento ao tipo de aplicação.

Segundo [AKYILDIZ, 2010], os protocolos de roteamento podem ser classificados da seguinte forma:

- centrado em dados: não se prende ao endereço do sensor mas aos dados que estão sendo monitorados;
- hierárquico: propõe uma estrutura hierárquica dos nós com diferente funções como sensor final e roteador;
- geográfico: o roteamento é baseado na posição do sensor na área onde está instalada a rede;
- baseado em *QoS*: o roteamento se baseia em critérios de qualidade de serviço oferecido. Podem existir informações que exigem mais qualidade na conectividade, possuindo mais prioridade.

2.1.2.3. Camada MAC

Tradicionalmente a camada 2 é referenciada como camada de enlace ou *data link layer*. Na pilha TCP/IP esta camada possui as funções de *Logical Link Control* (LLC), responsável pela ligação da camada 2 com a camada 3 e a função de controle de acesso ao meio (MAC). Como não é utilizado o LLC em RSSF a camada 2 é denominada, em geral, como MAC.

A camada MAC é responsável por determinar como será a comunicação entre os dispositivos. A RSSF utiliza a banda sem licenciamento ISM (*Industrial Scientific Medical*), na qual existe somente uma banda de frequência utilizada para a comunicação em ambos os sentidos [SM.2180, 2010]. Neste caso a disciplina de comunicação deve ser projetada para evitar colisão.

O canal sem fio em RSSF possui uma característica de ser o mesmo para todos os nós vizinhos, em geral. Ou seja, este canal deve ser compartilhado por todas as estações que estejam próximas o suficiente para que haja interferência entre elas. Neste caso a disputa pelo meio deve ter critérios que evitem ou minimizem a probabilidade de colisões de mensagens transmitidas simultaneamente.

Segundo [DARGIE, 2010] os protocolos MAC pode ser classificados da seguinte forma: protocolos livres de disputa com designação fixa (FDMA, TDMA e

CDMA) ou dinâmica (Polling, Passagem de ficha e baseado em reserva); protocolos baseados em disputa (ALOHA, CSMA, MACA e MACAW).

Os protocolos livres de disputa com designação fixa alocam alguma dimensão para a comunicação. Estas dimensões são frequência, tempo ou código. A comunicação se dá em uma das dimensões de forma exclusiva, não havendo colisão. Na designação dinâmica a primeira possibilidade é a base realizar um *polling* entre os sensores. Neste caso cada sensor responde a uma requisição da base, não existindo, portanto colisão. A outra forma é passando um *token* entre as estações que desejam transmitir. Finalmente é utilizada a estratégia da utilização de *slots* de tempo estáticos, que permite que os nós sensores reservem futuros acessos ao meio baseado na demanda [CYRIACO, 2012].

Os protocolos baseados em disputa são tradicionalmente utilizados em redes sem fio. O mais antigo é o *ALOHA* em que cada nó transmite e aguarda a confirmação de sucesso da transmissão. Com a evolução das técnicas de rádio frequência foi possível desenvolver transceptores que possuem a capacidade de "escutar o meio de comunicação" antes da transmissão. Esta estratégia é chamada de *Carrier Sense Multiple Access* (CSMA). Outra estratégia é o *Multiple Acces with Collision Avoidance* (MACA) com a solicitação de reserva do canal através de uma mensagem curta denominada *Request-To-Send* (RTS) transmitida pela estação que deseja utilizar o canal de transmissão e uma mensagem *Clear-To-Send* (CTS) que autoriza a transmissão. Por último existe a estratégia MACAW que foi desenvolvida para redes locais sem fio, em que a estação que recebe o frame transmitido responde com uma mensagem de confirmação (*Acknowledgement* - ACK), indicando para outras estações que o meio está livre.

2.1.2.4. Camada física

Para que seja possível o processo de comunicação o *frame* a ser transmitido deve sofrer processos que permitam a adaptação ao meio de comunicação, no caso o canal sem fio, bem como parâmetros para serem ajustados de acordo com as necessidades. Alguns desses paramêtros são relacionados a seguir: [KARL, 2005]

- Modulação
- Potência
- Canal de frequência

- Ganho da antena
- Taxa de transmissão

Estas informações são importantes de serem conhecidas em uma comunicação de rádio, uma vez que podem determinar a distância em função do canal de comunicação. A gerência da RSSF deve considerar estas informações para permitir os ajustes adequados para o bom funcionamento da rede [CYRIACO, 2012].

2.2. Gerência de Redes

Organizações investem quantidades significativas de tempo e dinheiro na construção de redes de dados complexas. Ao invés da companhia dedicar um ou mais engenheiros de rede somente para a manutenção, a relação custo-benefício seria melhor se o sistema pudesse cuidar dele mesmo na maior parte do tempo e, no processo, realizar tarefas de rotina para os engenheiros. Este cenário deixaria o engenheiro livre para trabalhar no desenvolvimento futuro da rede [LEINWAND, 1995].

A partir desta necessidade nasceu o conceito de gerenciamento de redes. O gerenciamento de rede é o processo de controlar uma rede de dados complexa para maximizar sua eficiência e produtividade. De acordo com [SAYDAM, 1996]:

"O gerenciamento de rede inclui a disponibilização, a integração e a coordenação de elementos de hardware, software e humanos, para monitorar, testar, consultar, configurar, analisar, avaliar e controlar os recursos da rede, e de elementos, para satisfazer às exigências operacionais, de desempenho e de qualidade de serviço em tempo real a um custo razoável."

Na figura 2.1 têm-se um exemplo de rede de gerência de dados clássica, onde o elemento de rede responsável pela gerência da mesma, ou estação de gerência de rede, é um nó de rede diferente das outras estações de trabalho e de outros elementos de rede, onde se encontram os agentes [CYRIACO, 2012].

O gerente é uma espécie de software que permite a obtenção e o envio de informações de gerenciamento junto aos Objetos Gerenciados, mediante a comunicação com um ou mais agentes [AZAMBUJA, 2001]. As informações devem ser obtidas através de requisições realizadas pelos gerentes aos agentes do sistema, ou então, através de envio automático disparado pelo agente a um determinado gerente (mensagens denominadas traps no SNMP). Normalmente, um gerente está presente em uma estação de gerenciamento de rede.

O agente nada mais é do que um software presente nos dispositivos gerenciados. A principal função de um agente é o atendimento às requisições efetuadas pelo software gerente e o envio automático de informações de gerenciamento ao gerente, indicando a existência de um evento previamente programado. Também é de responsabilidade do agente efetuar a interface entre diferentes mecanismos utilizados na instrumentação das funcionalidades de gerenciamento inseridas em um determinado dispositivo [STALLINGS, 1998].

Ao conjunto de variáveis utilizadas para representar informações estáticas ou dinâmicas vinculadas a um determinado objeto gerenciado, denominamos MIB (*Management Information Base*). Grande parte das funcionalidades de um gerente/agente, destina-se à troca de dados existentes na MIB [STALLINGS, 1998].

O agente é capaz de responder ao gerente consultas padronizadas sobre o conjunto de informações contido na base. De fato, em geral é codificado um arquivo, chamado arquivo de MIB, no qual são relacionadas informações para que o gerente saiba quais são os dados que podem ser solicitados a um agente e também as informações de alerta que poderão ser enviadas do agente para o gerente [RIGANTI, 2005].

Constituída por uma estrutura em árvore contendo as variáveis de gerência de um determinado equipamento, a MIB define para cada variável um identificador único denominado *Object Identifier* (OID), formado por um número inteiro não negativo. Em princípio, todos os objetos definidos em todos os padrões oficiais podem ser exclusivamente identificados. Para localizar uma determinada informação, o identificador da variável que será acessada pelo protocolo de comunicação é representado com o endereço Internet Protocol (IP) do equipamento em conjunto com o identificador do objeto na árvore MIB (OID) [STALLINGS, 1999].

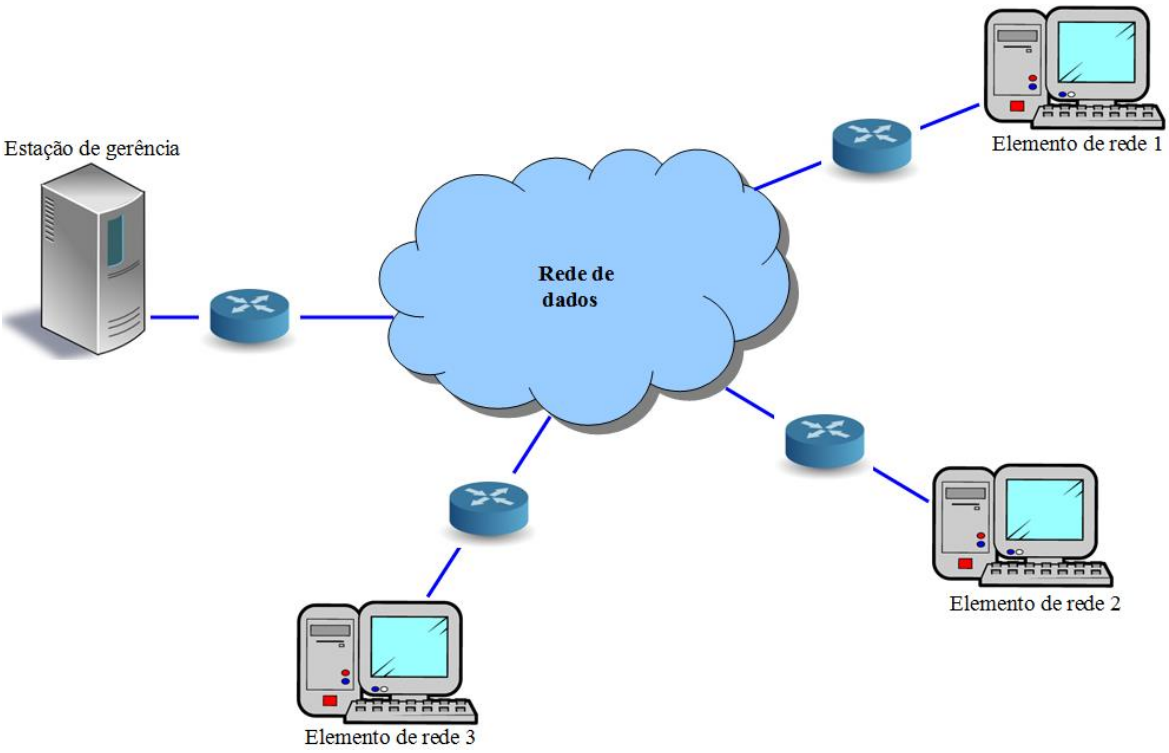

Figura 2.4 - Típica infra-estrutura de rede com estação de gerência

2.2.1. Áreas funcionais de gerenciamento

Para definir melhor o escopo do gerenciamento de redes, o fórum de gerenciamento de rede da ISO (International Organization for Standardization) propôs cinco áreas funcionais, sendo elas, gerência de configuração, gerência de desempenho, gerência de falha, gerência de contabilidade, ou *accounting*, e gerência de segurança [LEINWAND, 1995]. Um sistema de gerência de redes não precisa, obrigatoriamente, implementar todos esses pilares simultaneamente, pois a princípio são aspectos independentes uns dos outros [CYRIACO, 2012].

A seguir serão discutidos separadamente cada uma destas áreas funcionais, situando-as em casos específicos e introduzindo para cada uma delas a correspondência na gerência de rede de sensores sem fio.

2.2.1.1. Gerência de configuração

A gerência de configuração é o processo de obter dados de uma rede e utilizá-los para gerenciar a configuração de todos os dispositivos de rede. Está nela envolvida o

recolhimento de informação sobre a configuração de rede atual, utilizando estes dados para modificar a configuração de rede dos dispositivos, armazenamento dos dados, manutenção de um inventário atualizado e produção de relatórios baseados nos dados.

A gerência de configuração melhora o controle do engenheiro de rede sobre a configuração de dispositivos de rede ao oferecer um rápido acesso a dados vitais de configuração destes dispositivos. Por exemplo, um aspecto de dado de configuração é a configuração atual de cada dispositivo de rede. Ao se desejar adicionar interfaces de rede para um dispositivo em particular, seria desejável saber primeiramente o número de interfaces físicas disponíveis anteriormente no dispositivo, bem como o endereço de rede atribuído a estas interfaces, para auxiliar na configuração do software do dispositivo. Utilizando a gerência de configuração, esta informação pode ser localizada facilmente, uma vez que ela está armazenada em um local conhecido. Ela ainda pode ser responsável pelo desligamento de recursos individuais da rede, visando o isolamento de partes defeituosas da rede em operação.

Uma vez que a informação da gerência de configuração foi obtida, geralmente será necessário atualizá-la, com esta podendo acontecer de forma automática. E como o processo de gerência de configuração permite estas modificações, elas poderiam ser salvas antes de serem enviadas para o dispositivo. Assim o sistema de gerência seria capaz de verificar se as mudanças de configuração foram apropriadas para o dispositivo e avisar antes que aconteça uma má configuração inadvertida do dispositivo [LEINWAND, 1995].

Nas RSSFs a gerência de configuração se aplica na configuração inicial da rede e de sensores, configurando, como exemplo, parâmetros de rádio e limites de medição, ou reconfigurando toda a rede em caso de perda ou quebra de sensores. A operação das RSSFs é dependente do gerenciamento de configuração que inclui funções de planejamento e manutenção da rede. Então, sendo o objetivo da rede monitorar (coletar dados, processar dados e enviar estes dados ao observador) e controlar um ambiente, qualquer problema ou situação não prevista na configuração da rede pode comprometer a qualidade de serviço de uma RSSF.

O gerenciamento de configuração envolve a monitoração e a manutenção do estado da rede. O estado da rede é dependente de vários aspectos, como energia, topologia e conectividade. Assim, pode-se utilizar modelos para descrever esses aspectos. Outros aspectos devem ser considerados no gerenciamento de configuração:

localização dos nós, organização, densidade, estado operacional, estado administrativo, estado de uso da rede, entre outros [LOUREIRO et. al, 2003].

2.2.1.2. Gerência de desempenho

A gerência de desempenho envolve garantir que uma rede de dados permaneça acessível e não congestionada e para tanto verifica indicadores do ponto de vista de disponibilidade, tempo de resposta, *throughput* (vazão), e utilização. O desempenho é influenciado por fatores externos ao elemento da rede, ou do próprio equipamento [STALLINGS, 1999] [LEINWAND, 1995].

Geralmente, as condições de rede influenciam diretamente na experiência do usuário, negativa ou positivamente. Assim como os equipamentos, em seu conjunto *hardware* e *software*, também influenciam, segundo suas condições internas e dos serviços de rede que estes equipamentos requisitam da rede, utilizando suas pilhas de protocolo de rede para se comunicaram.

O principal benefício da gerência de desempenho é auxiliar na redução de sobrecarga e de inacessibilidade da rede provendo um nível consistente de serviço para o usuário. Utilizando-se a gerência de desempenho é possível monitorar a utilização de dispositivos de rede e de *links*, e tais dados podem auxiliar na identificação de tendências, isolamento de problemas de desempenho e, possivelmente, mesmo resolvê-los antes que eles tenham um impacto negativo na qualidade da rede [LEINWAND, 1995].

A gerência de conteúdo envolve quatro passos que se seguem: coleta de dados da utilização atual de dispositivos de rede e *links*, análise dos dados relevantes, definição de limites de utilização, e simulação da rede.

Então para a gerência de desempenho é necessário que se tenha como parâmetros, por exemplo, a capacidade máxima de uma rede a fim de determinar se este limite está próximo de ser alcançado. Com esse dado é possível correlacionar medições de tráfego de diversos elementos da rede e identificados possíveis ofensores à capacidade máxima da rede. Estes ofensores criam empecilhos ao desempenho geral da rede uma vez que, por exigência de possíveis aplicações executadas por eles, exigem muita vazão da rede [CYRIACO, 2012].

Vale lembrar que a qualidade de uma rede de sensores está associada ao tempo de vida residual desta rede. Assim, os dois objetivos principais das redes de sensores

são: disseminar informações coletadas sobre o ambiente monitorado e prolongar o tempo de vida da rede. Vale ressaltar que este propósito é diferente das outras redes sem fio onde o objetivo é prover qualidade de serviço e alta largura de banda. Porém, as redes de sensores herdaram os problemas clássicos das redes sem fio, quais sejam, porcentagem elevada de dados perdidos na comunicação e dificuldade de controle de energia

Na área de desempenho, o desafio diz respeito ao número de parâmetros gerenciados, pois se for grande, o consumo de recursos também aumenta e o tempo de vida da rede pode ser comprometido. Contudo, o monitoramento do desempenho da rede é necessário para garantir a qualidade do serviço entregue pelas redes de sensores. Um exemplo de QoS pode ser observado no intervalo de tempo necessário para se obter a informação sobre um ambiente monitorado. Se o tempo de entrega for excessivo esta informação pode perder o seu valor para a aplicação.

O gerenciamento de desempenho pode disponibilizar funções que permitem à aplicação definir a métrica de qualidade. Isto poderá influenciar na densidade de nós, exposição, quantidade de energia dissipada, entre outros.

Um mecanismo para a implantação de qualidade de serviço nas redes de sensores é atribuir diferentes níveis de importância às informações. Por exemplo, um sensor na floresta coleta informações de temperatura de 25º C na primavera, que está na faixa de valores esperados. Se fosse 50º C na mesma situação, isto seria uma informação mais relevante. Para informações ou pacotes de maior importância, a rede deve fazer um esforço maior para entregá-lo. Isto é, a energia gasta deve variar com a importância dos dados.

Em geral, o gerenciamento de desempenho inclui os seguintes grupos de conjunto de funções: garantia de qualidade, monitoramento, controle e análise do desempenho. O processo de gerenciamento de QoS inicia com a detecção da degradação e finaliza com a eliminação causa primária do problema, passando por estágios intermediários de investigação e análise [LOUREIRO et. al, 2003].

2.2.1.3. Gerência de falhas

A gerência de falhas é o processo de localizar e corrigir problemas ou falhas. Das muitas tarefas envolvidas na gerência de redes, a gerência de falhas é provavelmente a mais importante. Consiste em identificar a ocorrência de uma falha em

uma rede de dados, isolamento da causa da falha e sua posterior correção, caso seja possível. A gerência de falhas aumenta a confiabilidade de uma rede ao prover ferramentas para detectar problemas rapidamente e iniciar o processo de recuperação.

A gerência de falhas implementa, em uma estação de gerência a capacidade de reunir informações sobre comportamentos anômalos atuais de uma rede, pela observação de eventos, ou alarmes, e a detecção de falhas. Ela também pode ser utilizada na identificação de problemas futuros, utilizando os alarmes [CYRIACO, 2012].

Para que seja realizada a identificação de um problema, eventos devem ser transmitidos por um dispositivo de rede quando acontecer uma condição de falha, podendo ser citado como exemplos a falha de um link, a reinicialização de um dispositivo ou a falta de resposta de um host [LEINWAND, 1995].

O papel da gerência de falhas está ligado diretamente à disponibilidade dos serviços prestados pela rede de dados. Através da gerência de falhas a causa de um comportamento anômalo deve ser localizada física e logicamente na rede de dados. Ações corretivas devem ser imediatamente disparadas, seja por vias próprias, enviando comandos remotos ao sistema defeituoso, seja por processos acordados entre equipes de manutenção que recebem o relatório de falhas e se deslocam fisicamente para realizar a correção do problema.

A falha não será uma exceção, mas uma ocorrência normal em uma RSSF. Em todo momento sensores podem falhar em decorrência da falta de energia. Assim o gerenciamento de falhas deve prover funções que permitam detectar nós em que o nível de energia é insuficiente para a execução de atividades.

A rede deve ser tolerante a falhas no sentido de resolver os problemas de topologia e conectividade decorrentes das falhas dos nós, resolver o problema da cobertura e da exposição. Assim, o gerenciamento de falhas deve prover funções de correção de anormalidades incluindo funções de manutenção da rede. Este processo pode incluir a ativação de nós, disposição de novos nós, alterações na topologia da rede, funções de verificação do mapa de energia, funções de verificação da área de cobertura, entre outros [LOUREIRO et. al, 2003].

A gerência de falhas atua no envio de informações de perdas de pacote e indisponibilidade de sensores, utilizando-se a perda de comunicação com um determinado sensor e gerando um evento ou alarme, indicativo de falha. Através de

relatórios elaborados pela gerência de falhas, ações podem ser tomadas no sentido de se recuperar a rede [CYRIACO, 2012].

2.2.1.4. Gerência de segurança

A gerência de segurança envolve a proteção de informações sensíveis localizadas em dispositivos ligados a uma rede de dados ao controlar os pontos de acesso para aquelas informações. Pode-se definir como informação sensível qualquer dado que uma organização queira assegurar, tais como dados de pagamentos, contas de consumidores, e cronogramas de pesquisa e desenvolvimento. A gerência de segurança permite a proteção de informação ao limitar o acesso a *hosts* e dispositivos de rede para usuários (tanto de dentro quanto de fora da organização) e ao notificar o responsável sobre falhas de segurança existentes, ou sobre tentativas de tentar causar falhas. [LEINWAND, 1995].

O controle sobre senhas de rede e seu tempo de validade, privilégios de usuário sobre recursos de rede e distribuição de certificados digitais são exemplos de controles realizados pela gerência de segurança com o objetivar de segregar os recursos e separar usuários e administradores por níveis de responsabilidade.

Além da implementação de políticas de tratamento de informações sigilosas, a gerência de segurança também se responsabiliza pelo monitoramento e garantia da aplicação destas políticas. Geralmente o meio mais utilizado é a geração de registros de atividades textuais, ou *logs*, e a permanente auditoria destes registros em busca de evidências de atividades que estejam em desacordo com os níveis de sigilos desejados [CYRIACO, 2012].

As RSSFs podem empregar um grande número de nós sensores comunicando e desenvolvendo padrões irregulares de processamento distribuído *adhoc* que por sua vez podem produzir informação de alta qualidade com consumo minimizado de recurso. Para prover confidencialidade, integridade e autenticação, esquemas de segurança deverão ser adotados, se a aplicação demandar tal requisito. Essas funcionalidades de segurança são difíceis de disponibilizar devido à natureza não estruturada da rede, a conectividade intermitente e a limitação de recursos.

O controle de acesso aos dados também representa uma funcionalidade importante para as redes de sensores. Em aplicações militares, o alcance das transmissões pode ser reduzido de forma implícita, isto é os nós sensores devem possuir

tamanho reduzido e apresentar um alcance de transmissão pequeno para reduzir a probabilidade de detecção da rede pelo inimigo [LOUREIRO et. al, 2003].

2.2.1.5. Gerência de contabilidade

A gerência de contabilidade consiste na mensuração da utilização de recursos de rede por usuários a fim de que se estabeleça métricas, verificar quotas, determinar custos e cobrar os usuários, além de coletar estatísticas da rede para facilitar a tomada de decisões sobre a alocação de recursos de rede. O uso da gerência de contabilidade ajuda no entendimento do comportamento de usuários em uma rede de dados e pode auxiliar o responsável pela rede a influenciar o comportamento dos usuários de modo que se obtenha um uso mais otimizado dos recursos de rede [LEINWAND, 1995].

Do ponto de vista dos administradores, a gerência de contabilidade também é muito importante como uma fonte de informação sobre o retorno do investimento realizado sobre a rede e sobre a necessidade de investimentos futuros, principalmente em capacidade e tecnologias. Então, utilizando a gerência de contabilidade é possível monitorar o uso da rede e detectar o abuso da utilização de recursos ou a subutilização dos mesmos [CYRIACO, 2012].

Em RSSFs, a contabilização pode prever funções de custo que representam o consumo de energia por nó ou por componente do nó. Estas funções podem ser utilizadas para traçar o comportamento da rede a até mesmo inferir sobre o comportamento dos nós [LOUREIRO et. al, 2003].

A gerência de contabilidade em redes de sensores sem fio também pode ser utilizada na detecção da banda utilizada para envio das informações entre os sensores, e posterior detecção de quão sobrecarregada a rede está e o que deve ser feito para melhorar o seu desempenho.

2.2.2. O protocolo SNMP e Base de Informação de Gerência

O SNMP (*Simple Network Management Protocol)* é um protocolo da camada de aplicação desenvolvido para gerenciar dispositivos de rede que implementam a arquitetura TCP/IP (Internet). A implementação deste tipo de protocolo se baseia no princípio de interação entre a estação de gerência, que é responsável por enviar as mensagens SNMP de consulta/atribuição, e agentes de gerência, que respondem às

consultas ou geram mensagens espontâneas (*traps)* em situações extraordinárias, através de um protocolo de comunicação.

A estação de gerência, que concentra os dados dos agentes, é um elemento independente da rede. Ela deve possuir aplicações que implementam as funcionalidades das 5 áreas funcionais de gerência. No dispositivo gerenciado reside um agente que junta as informações do dispositivo gerenciado no qual ele está em funcionamento. Essa informação que é obtida e organizada é definida na MIB. A estação pode requisitar informações ao agente, então o agente obtém estas informações e responde ao pedido enviando a informação apropriada utilizando mensagens SNMP.

A Base de Informação de Gerência (MIB) é um armazém de informações virtuais que contém os objetos gerenciados que descrevem as informações do dispositivo gerenciado. Estes objetos gerenciados são descritos utilizando um subconjunto limitado da ASN.1 (*Abstract Syntax Notation One*), que é uma notação formal utilizada para descrever dados transmitidos por protocolos de telecomunicação, independentemente da linguagem de implementação e da representação física desses dados [ITU-T. 2013]. Cada objeto gerenciado possui um nome, sintaxe e codificação. Para cada objeto gerenciado na MIB é assinalado um identificador de objeto, que funciona como se fosse um nome. A sintaxe de um objeto se refere à estrutura de dados abstrata a qual o tipo de objeto corresponde. Por exemplo, um objeto pode ser do tipo *integer* ou *octet string*. Nem todos os tipos da ASN.1 são permitidos na definição da sintaxe de um objeto gerenciado. Os tipos que podem ser utilizados são limitados por uma questão de simplicidade. A codificação dos objetos gerenciados diz respeito ao modo como os objetos são representados, quando eles estão sendo transmitidos em uma rede utilizando a sintaxe dos objetos. As regras básicas de codificação (BER - *Basic Encoding Rules*) da ASN.1 são utilizadas para codificação.

Um identificador de objeto (OID) é uma lista de números que percorre uma árvore global da sua raiz até o objeto específico, similar a um endereço ou caminho. Para cada objeto é assinalada uma *string* chamada de descritor de objeto, que é utilizado para facilitar a referência ao mesmo utilizando uma palavra de mais fácil entendimento para humanos. A árvore começa em um nó raiz sem nome com múltiplas sub-árvores [STEENKAMP, 2012]. A figura 2.5 mostra parte da árvore de OID.

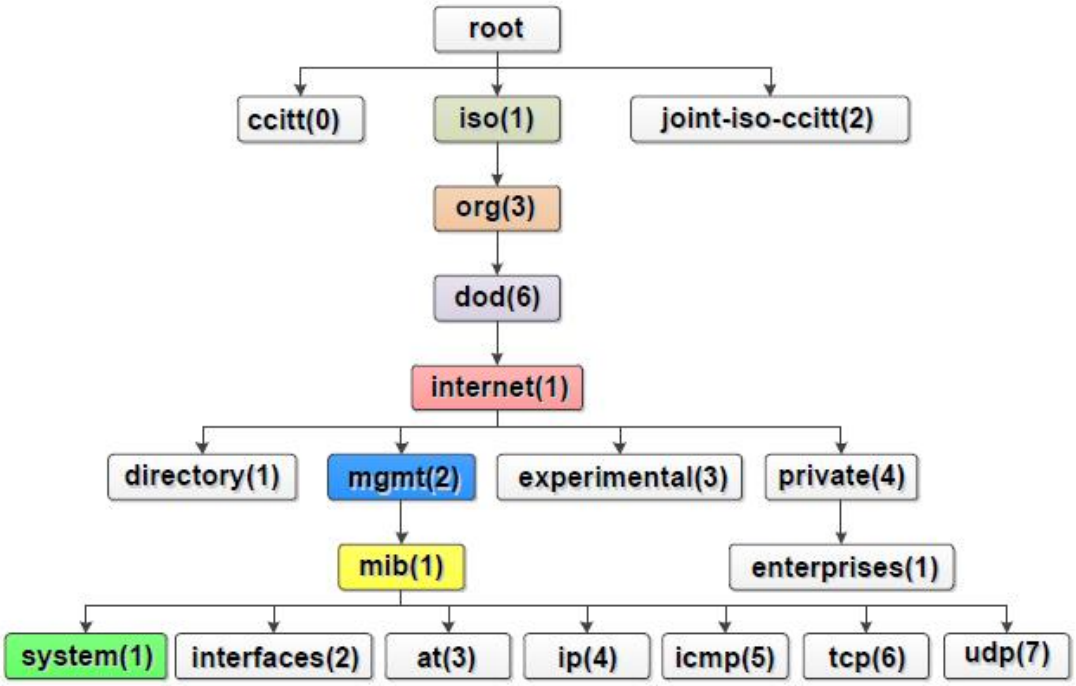

Figura 2.5 - Árvore de OID [STEENKAMP, 2012]

2.2.2.1. SNMPv1

O *Simple Network Management Protocol* versão 1 (SNMPv1) foi a versão inicial do SNMP, especificada na RFC 1157 [CASE et al, 1990]. Junto com a RFC 1155, que define a SMI descrevendo como objetos gerenciados são definidos na MIB, e RFC 1156, que define objetos presentes na MIB a RFC 1157 forma a primeira aparição do SNMP. O SNMP é utilizado para transmitir informações de gerenciamento entre as estações de gerência de rede e os agentes rodando nos elementos da rede. Ele trata todo o gerenciamento como a leitura ou alteração de valores de variáveis. Isto essencialmente limita o número de comandos utilizados no SNMP para dois, uma operação para obter o valor de um dispositivo gerenciado (get) e uma operação para alterar um dado valor em um dispositivo gerenciado (set). Um agente de gerência de rede também é capaz de enviar informações não solicitadas para uma estação de gerência (traps). Um grupo consistindo de um número arbitrário de estações de gerência de rede e agentes SNMP é chamado de comunidade SNMP. Cada comunidade é identificada por uma *string* e no SNMPv1 ela é utilizada como uma forma rudimentar de autenticação.

O SNMP não requer serviço de datagrama confiável e portanto utiliza o UDP (*User Datagram Protocol*) como meio de transporte. Cada mensagem SNMP é composta pelo identificador da versão, senha e o PDU(*Protocol Data Unit*) que contém os padrões de mensagem SNMP, como mostrado na figura 2.6. Uma mensagem SNMP é recebida na porta 161 e as mensagens de trap são enviadas para a porta 162. O SNMP especifica cinco PDUs: GetRequest, GetNextRequest, GetResponse, SetRequest e Trap.

Figura 2.6 - Estrutura de pacote SNMPv1 [STEENKAMP, 2012]

As PDUs GetRequest, GetNextResquest, GetResponse and SetRequest possuem a mesma estrutura básica, enquanto a Trap se diferencia das anteriores, como será mostrado posteriormente. A figura 2.7 mostra a estrutura básica do PDU do SNMPv1. O RequestID é utilizado para identificar mensagens individuais e detectar mensagens duplicadas. ErrorStatus e ErrorIndex proveem informações sobre erros que possam ter ocorrido. O ErrorStatus indica que erro ocorreu enquanto ErrorIndex pode dar informações adicionais sobre qual variável na lista de variáveis na PDU causou o erro. Quando não existe erro o valor de ErrorStatus é zero. O VarBindList é uma lista que faz o pareamento entre o nome da variável e seu respectivo valor. Entretanto algumas PDU utilizam somente o nome da variável, como exemplo o GetRequest. Nestes casos a parte do valor da ligação é desconsiderada. Embora ela não seja utilizada é recomendado que ela ainda seja codificada utilizando o valor nulo da ASN.1.

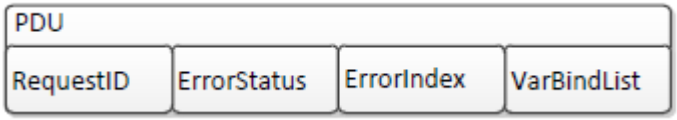

Figura 2.7 – PDU do SNMPv1 [STEENKAMP, 2012]

O GetRequest é gerado quando uma estação de gerência de rede deseja recuperar informação a partir de um dispositivo gerenciado. O GetRequest contém o nome da variável na forma do OID, correspondendo ao objeto gerenciado ao qual se deseja obter o valor. O dispositivo gerenciado responde ao pedido com um GetResponse que é

idêntico ao GetRequest com o valor da variável adicionado. O SetRequest é gerado quando uma estação de gerência deseja definir ou alterar um valor de um objeto gerenciado em um dispositivo gerenciado. O SetRequest contém o nome da variável do objeto gerenciado bem como o novo valor da variável. Após receber o SetRequest o dispositivo gerenciado responde com um GetResponse que contém o nome da variável e seu novo valor. Um dispositivo gerenciado responde a um GetNextRequest com um GetResponse que contém o nome da variável e o valor da variável localizada imediatamente após à variável solicitada na árvore. Um dos usos do GetNextRequest é para recuperar objetos compostos, uma tabela de roteamento por exemplo, onde deve ser usada uma mensagem GetRequest e várias mensagens GetNextRequest.

Os dispositivos gerenciados também são capazes de enviar mensagens não solicitadas para as estações de gerência de rede. Isto pode ser visto como um dispositivo gerenciado enviando um GetResponse sem que este tenha recebido um GetRequest. Este tipo de mensagem é gerada ao enviar o Trap. A estrutura do Trap é diferente dos demais PDUs do SNMPv1 como pode ser vista na figura 2.8. O campo *enterprise* contém a OID da entidade ou organização que está gerando o trap. O campo *agent-addr* contém o endereço IPv4 do agente SNMP que gerou a mensagem. Caso o agente não utilize o IPv4 este campo recebe o valor 0.0.0.0 conforme indicado por [THALER, 2002]. O campo *generic-trap* indica o tipo de trap que foi gerado, com o campo *specific-trap* sendo utilizado para traps específicos diferentes dos tipos especificados no *generic-trap*. O campo *time-stamp* contém a quantidade de tempo há que o agente está funcionando ininterruptamente. O campo *VarbindList* é utilizado da mesma forma como explicado anteriormente. As PDUs utilizadas no SNMPv1 e seus respectivos campos são documentadas na RFC editado por [CASE et al., 1990] [STEENKAMP, 2012].

Figura 2.8 – PDU da trap do SNMPv1 [STEENKAMP, 2012]

2.2.2.2. SNMPv2

Em 1993, a segunda versão do SNMP foi definida e construída baseada na primeira versão do protocolo[Case et al, 1993]. Uma segunda versão do documento da

estrutura de informação de gerência (SMIv2), e a RFC 2578 [MCGLOGHRIE et al, 1999b] também foi publicada para a utilização com o SNMPv2. Também foi publicado um documento descrevendo a base de informação de gerência (MIB) para o SNMPv2 através da RFC 3418 [PRESUHN et al, 2002c]. Como na SNMPv1, o UDP foi escolhido como preferencial para ser o protocolo de transporte, embora outros tipos sejam possíveis [PRESUHN et al, 2002b].

O modelo administrativo para o SNMPv2 [GALVIN et al, 1993a] descreve o comportamento dos componentes do SNMPv2. Debaixo do SNMPv2 cada elemento utiliza protocolos de privacidade e autenticação únicos como definido pelo documento de protocolos de segurança para o SNMPv2 [GALVIN et al, 1993b]. Esta característica adicional de segurança é uma das diferenças entre o SNMP versão 1 e versão 2.

O documento de protocolos de segurança define um protocolo de autenticação que garante que uma mensagem enviada por um elemento seja corretamente identificado como sido originada por ele. Também garante que a mensagem recebida por um dispositivo é a mesma que foi enviada. Isto é feito utilizando um protocolo de autenticação *digest* e utiliza o MD5 com 128 bits [GALVIN et al, 1993b].

O documento de protocolos de segurança também especifica um protocolo de privacidade que garante que uma mensagem enviada de um dispositivo SNMPv2 para outro, não possa ser lido por um terceiro. A privacidade é garantida utilizando o *Data Encryption Standard (DES)* no modo *Cipher Block Chaining mode* [GALVIN et al, 1993b].

Em contraste às propostas de adicionais de segurança do SNMPv2 foi publicado um memorando que descreve o SNMPv2 baseado em senha, também conhecido como SNMPv2c [CASE et al, 1996a]. O protocolo aboliu qualquer característica de segurança adicionada no SNMPv2 e utiliza um *framwork* administrativo utilizando senha que foi baseado no SNMPv1. Entretanto o SNMPv2c utiliza os novos tipos de PDU e de código de erros definidos para o SNMPv2 [PRESUHN et al, 2002c]. Esta versão teve um suporte maior dentro da IETF (Internet Enginnering Task Force) mesmo com um nível de segurança menor [CASE et al, 2002].

Existem outras variantes do SNMPv2 que são SNMPv2u [MCCLOGHRIE, 1996][WATERS, 1996] and SNMPv2* [SNMPv3WG, 2002][CASE et al, 2002]. Estas versões, ao contrário do SNMPv2c, implementaram os recursos adicionais de segurança mas não obtiveram um consenso de suporte dentro do IETF [CASE et al, 2002].

O SNMPv2 oferece alguma melhoras em relação ao SNMPv1 mas algumas das funcionalidades permanecem as mesmas na versão dois. As operações Set, Get e GetNext são as mesmas da versão 1, mas algumas novas operações foram adicionadas na versão 2. Estas alterações incluem a requisição GetBulk e operação Inform. A operação Trap ainda está presente na versão dois, mas utiliza um formato de mensagem diferente. O SNMPv2 também adicionou tipos expandidos de dados como contador de 64 bits.

A operação GetBulk oferece um modo eficiente de buscar grandes blocos de informação. Também pode ser utilizada de um modo similar ao GetNext para percorrer o conteúdo de tabelas. Quando uma mensagem de Trap SNMPv2 é enviada não existe nenhuma confirmação de recebimento. A operação Inform permite a troca de informação entre estações de gerência, possibilitando implementar sistemas de gerência hierárquicos e/ou distribuídos [LEINWAND, 1995].

2.2.2.3. SNMPv3

A terceira versão do SNMP é derivada e foi construída tendo como base as versões anteriores do protocolo. O grupo de trabalho envolvido com o SNMPv3 teve como objetivo principal a definição de adições de segurança e administração para o ambiente de gerência do SNMP, as quais tornariam a comunicação de dados de gerência segura. Esta segurança inclui mecanismos de autenticação e privacidade. O grupo de trabalho (WG) do IETF para o SNMPv3 ficou com a tarefa de criar um padrão único e um conjunto de documentos para providenciar a segurança necessária, algo que o grupo de trabalho do SNMPv2 não conseguiu completar. O SNMPv3WG então ficou responsável por produzir um conjunto único de documentos baseados nos conceitos do SNMPv2u e SNMPv2* [CASE et al, 2002].

O SNMPv3 continuou a utilizar uma arquitetura modular. Todas as três versões do arcabouço padrão de gerência compartilham a mesma estrutura básica. Isto fez com que seja possível a utilização de rascunhos de padrões do SNMPv2 onde possível, fazendo com que o SNMPv3WG se focasse na tarefa principal que era segurança, sem que fosse necessário "reinventar a roda" [CASE et al, 2002]. Este projeto modular também tinha o benefício de que os documentos podiam ser atualizados ou trocados se necessário sem afetar os outros documentos ou módulos. O SNMPv3 também utiliza o

SMIv2 do SNMPv2 [MCCLOGHRIE et al, 1999b;c;a] como uma linguagem de definição de dados para a descrição de módulos da MIB.

Especificações para a operação do SNMPv3 podem ser encontradas das RFC 3410 até 3418. Estes documentos incluem uma RFC de protocolos de operação [PRESUHN et al, 2002c] que é baseada em atualizações e no documento de operações de protocolo do SNMPv2[CASE et al, 1996b], assim construindo sobre e reutilizando padrões já definidos. A RFC de mapeamento de transporte do SNMPv2 também foi reutilizado de uma forma atualizada, RFC 3417 [PRESUHN et al, 2002b]. A RFC 2576 [FRYE et al, 2000] descreve a coexistência entre o SNMPv3, SNMPv2 e SNMPv1.

As RFC 3414 [BLUMENTHAL et al, 2002] e RFC 3415 [WIJNEN et al, 2002], individualmente descreveram o modelo de segurança baseado em usuário (USM – *User-Based Security Model*) e um modelo de controle de acesso baseado em visões (VCAM – *View-based Access Control Model*) para o SNMPv3. O USM provê segurança para as mensagens SNMP enquanto o VCAM controla o acesso às informações de gerência. A RFC 3414 descreve a utilização do HMAC-MD5-96 e HMAC-SHA-96 como protocolos de autenticação. Ele também descreve o uso do CDC-DES que é protocolo simétrico de encriptação. O memorando também afirma ser possível substituir ou suplementar estes protocolos no futuro [STEENKAMP, 2012].

2.2.2.4. A MIB no contexto de gerência atual

O exemplo mais comumente encontrado em estações de gerência SNMP é a gerência de rede IP via MIB-II [MACCLOGHRIE, 1991]. Esta MIB está dividida em dez subgrupos de objetos de gerência *(System Group, Interfaces Group, Addres Translation Group, IP Group, ICMP Group, TCP Group, UDP Group, EGP Group, Transmission Group e SNMP Group)*. Cada um deste grupos oferece um conjunto de objetos de gerência, que são parâmetros indicativos do comportamento de uma rede IP em suas diversas camadas. Estes grupos de objetos podem ser tratados separadamente, na identificação de padrões comuns de uma camada, ou correlacionados, de modo a identificar padrões de comportamento que interferem em diferentes camadas.

Em certos casos o protocolo SNMP não pode ser implementado, por características das máquinas ou por se tratarem de redes que possuem recursos escassos, sejam eles de processamento, armazenamento ou fonte de energia. Neste caso, um recurso pode ser utilizado para adaptar uma rede apropriada para o SNMP a estas

máquinas chamado Pr*oxy Agent*. Este recurso será amplamente explorado neste trabalho, com o objetivo de implementar as funcionalidades básicas do SNMP em RSSFs [CYRIACO, 2012].

Na figura 2.2 podemos observar o grupo interfaces e dentro dele temos os objetos *ifInOctet* e *ifOutOctet* que possuem respectivamente informações do número de bytes recebidos e transmitidos por uma dada interface de rede de um dispositivo qualquer. Informações estas que podem ser utilizadas para gerar um gráfico com a utilização de um *link*, por exemplo.

Também pode ser observado na figura 2.2 o grupo experimental e enterprise. O grupo experimental é utilizado para o desenvolvimento de protocolos e MIBs experimentais, as quais se tenha intenção de testar para posteriormente ser definida como padrão. Portanto a MIB definida neste trabalho será inserida no grupo experimental. Posteriormente a MIB pode ser movida para o grupo enterprise que contém MIBs padronizadas e registradas especificamente por uma companhia ou organização [LEINWAND, 1995].

2.3. Gerenciamento de RSSF

Por se tratar de um assunto tão abrangente e de um sistema com características tão particulares, a definição de uma arquitetura de gerenciamento para as redes de sensores sem fio exige o estudo de vários contextos nas diferentes áreas funcionais e níveis de gerenciamento. O gerenciamento da rede de sensores sem fio também prevê a utilização de modelos que representem o estado da rede, como por exemplo, mapa de energia, topologia, conectividade, e modelos não determinísticos, e abstração de fases para o seu ciclo de vida, ou seja, estabelecimento da rede, manutenção, sensoriamento do ambiente, processamento e comunicação.

O projeto de uma rede de sensores sem fio é dependente da aplicação e das características envolvidas com essas redes. Assim, o gerenciamento deve considerar os aspectos genéricos envolvidos com o estabelecimento e a manutenção da rede e com o sensoriamento, o processamento e a comunicação dos dados. Isto significa que o gerenciamento deverá encontrar as possíveis similaridades existentes e para estas, propor uma lista de funções de gerenciamento. Nas redes de sensores sem fio, o gerenciamento de energia costuma ser um dos principais aspectos a ser considerado já que a longevidade da rede depende da sua utilização racional. Isto implica na limitação

da capacidade de processamento e na redução da largura de banda utilizada para transmissão.

Uma discussão importante do ponto de vista de gerenciamento diz respeito ao comportamento imprevisível da rede, principalmente devido ao ambiente onde a rede está inserida. Consequentemente, os resultados obtidos por uma aplicação específica poderão ser diferentes a cada execução. Outra consequência da imprevisibilidade é a possível ociosidade dos nós por um período longo de tempo.

As redes de sensores são ditas de aplicação específica (*application-specific)* já que são desenvolvidas para uma determinada função. As atividades de planejamento, desenvolvimento e manutenção ocorrem em função do objetivo da rede. Assim, as aplicações que utilizam redes de sensores assumem e executam as próprias funções de gerenciamento que tendem a ser específicas ao invés de propósito geral. Além disso, as redes de sensores são desenvolvidas sem uma separação lógica entre as funcionalidades de gerenciamento (serviços e funções de gerenciamento) e as funcionalidades da rede. Talvez isso seja consequência da não utilização de uma arquitetura de gerenciamento para RSSFs que permita a superposição da rede lógica da gerência sobre a rede física de aplicação específica [LOUREIRO et. al, 2003].

3. Propostas existentes de gerência de RSSF

A questão de gerência de RSSF ainda é um tema em aberto. Algumas propostas consideram uma autonomia total das RSSF em sua organização indicando uma não necessidade de gerência deste tipo de rede. No entanto, esta visão é distorcida, uma vez que na maioria dos casos será necessária a gerência deste tipo de rede para avaliar o seu funcionamento. Em [AKYILDIZ, 2010] esta questão é tratada esclarecendo que existem diversos tipos de RSSFs para as mais variadas finalidades. O que é observado na maioria dos casos é que as RSSFs operam em ambientes habitados, nos quais em boa parte existe energia para alimentar os sensores. Este cenário permite propor técnicas de gerência para um controle mais efetivo do desempenho deste tipo de rede. Este trabalho se enquadra nesta ideia de investigar formas factíveis e práticas para gerência de RSSF e sua integração com a internet. Existem algumas abordagens que sugerem utilizar alternativas para a gerência de RSSF, na tentativa de criar paradigmas de gerência centrados em arquiteturas de auto-organização e auto-gestão [XIAO, 2007].

3.1. WSNMP

Uma proposta é a arquitetura de gerência de RSSF do tipo WSNMP ou *Wireless Sensor Network Management Protocol* [ALAM, 2008]. Nela foi implementada uma proposta de gerência hierárquica. Os agentes de gerência seguem uma implementação distribuída por toda a RSSF, mas o nó responsável por reunir informações e enviar a uma gerência centralizada é um nó de rede com a função de criar *clusters*, ou um conjunto de sensores que se organizam geograficamente, se vinculam e centralizam as informações. Este nó recebe o nome de *cluster head* e implementa a função de Gerente intermediário de rede (*Intermediate Network* Manager), como pode ser visto na figura 3.1. O WSNMP parte de alguns requisitos como se seguem:

- a rede deve ser homogênea, ou seja todos os nós devem ter o mesmo poder de processamento e mesma faixa de alcance e transmissão;
- o CSMA/CA é utilizado como protocolo MAC;
- a rede pode ser tanto direcionada a eventos, quando os nós dentro do raio do evento geram tráfego, quanto periódica quando os sensores geram dados periodicamente a uma taxa fixa.

O WSNMP implementa um gerente central que está no nível mais alto da hierarquia e está localizado no *sink* e um gerente intermediário em cada *cluster head*, enquanto os agentes são implementados em cada nó sensor. Os gerentes intermediários são utilizados para distribuir as funções de gerência e coletar e organizar os dados de gerência. Eles não se comunicam entre si e trabalham de forma independente. Eles executam funções de gerência baseados nos estados de sua rede local ao passo que o gerente central tem o conhecimento global do estado da rede e junta o conhecimento global das camadas subjacentes e gerentes intermediários.

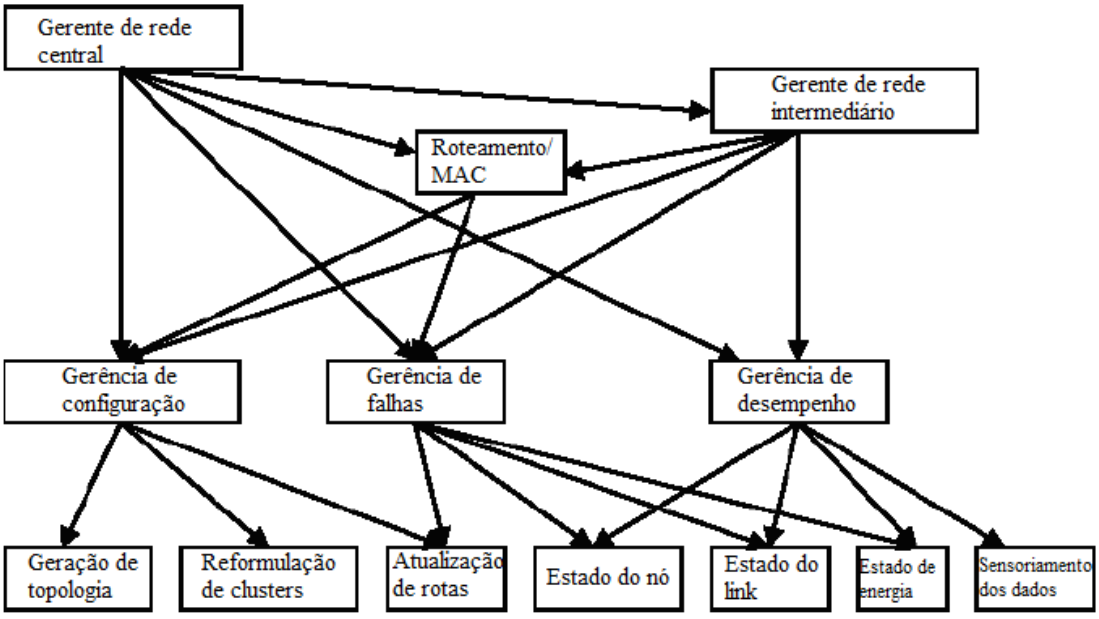

Figura 3.1 - Arquitetura do WSNMP [ALAM, 2008]

Nesta arquitetura parte da gerência está implementada diretamente nos sensores, que possuem as funcionalidades de configuração, desempenho e falha presentes. Cada uma desta funcionalidades atua nos sensores, modificando os seus estados de funcionamento.

A gerência de configuração do modelo de gerência WSNMP, trata de coletar dados sobre a rede e agir sobre esta, principalmente no que diz respeito à formação de clusters, ou união de um grupo de sensores que se interconectam. A gerência de configuração atua sobre a rede através reformulação de clusters, geração de topologia e atualização de rotas.

Um *cluster head* geralmente consome mais energia que um nó sensor normal uma vez que ele tem que passar adiante os dados, não apenas dos membros mas também

dos *cluester heads*. Então para balancear o consumo de energia é necessário trocar o *cluster head* periodicamente, mecanismo conhecido como reformulação de *clusters*
O gerente central possui a informação completa da topologia e ao analisar o pacote recebido ele pode encontrar o nível aproximado de energia de cada nó sensor.

A geração de topologia trata da definição das interconexões entre os sensores de um determinado cluster. Através da geração de topologia as interconexões entre o *cluster head* e os sensores pertencentes ao cluster, assim como as interconexões entre sensores, são definidas e guardadas pela Gerência de Configuração para reutilização em futuras configurações. As rotas são definidas pela Gerência de Configuração de modo a definir o fluxo de informação através do cluster.

A Gerência de Falhas é responsável por determinar, através de um mecanismo de detecção de falhas, se um sensor está ou não em estado de falha. Esse mecanismo é o resultado da correlação entre as informações recebidas pela Gerência de Falhas. A Gerência de Falhas utiliza informações de atualização de rotas, estado dos sensores, podendo ser inativo ou ativo, estado de um link de comunicação e estado de energia em um sensor para definir uma falha.

O estado de um sensor deve ser conhecido pela gerência de falhas para que uma eventual falha de comunicação possa ser corretamente interpretada. Um sensor pode também sofrer uma transição de estados caso haja necessidade de economia de energia.

Os sensores e o cluster head devem estar permanentemente trocando informações. Caso isso não aconteça. em se tratando de sensores que estejam acordados, o *link state* pode avisar a gerência de falhas sobre um problema de comunicação com um determinado sensor. A partir da detecção desta falha são realizadas alterações na rede, através da Gerência de Configuração, com o objetivo de contornar a falha.

A Gerência de Falhas determina a permanente monitoração da energia presente nos sensores, retendo esta informação e disponibilizando para a Gerência de Configuração quando necessário.

A Gerência de Desempenho, ou *Performance Management*, é responsável por monitorar a RSSF de modo a manter o consumo de energia o mais otimizado possível. O monitoramento dos estados dos *links*, energia, sensores inativos e detecção de dados na rede, é realizado pela Gerência de Desempenho com este objetivo.

Um sensor deve, de tempos em tempos, se comunicar com outros sensores ou com o *cluster head* para garantir que um *link* esteja em funcionamento. Porém, é importante também que seja monitorada a existência de dados sendo enviados na rede.

Pacotes de dados a serem enviados podem determinar, eventualmente, a transição de estado acordado ou dormindo em sensores vizinhos, de modo a garantir uma rota de comunicação para o sensor.

Neste tipo de arquitetura identifica-se um nível de complexidade que não condiz com a implementação de sensores simples para medições em campo. Para a utilização do protocolo WSNMP é necessária a existência de uma pilha de protocolos em cada um dos sensores, além disso o gerente central demanda um hardware de nível de processamento e memória maiores, devido à todas as informações que são tratadas e armazenadas nele.

O WSNMP, como exposto anteriormente, apresenta uma proposta de gerência para uma RSSF hierárquica, que não é o caso deste trabalho. Além disto, não possibilita a implementação do SNMP para realizar a gerência, o que acontece neste trabalho.

3.2 MANNA

Outra proposta para a gerência de RSSF é o arcabouço de gerência de sensores *MANNA, Management Architecture for Wireless Sensor Networks* [RUIZ et. at, 2003]. O objetivo principal desta arquitetura de gerência é definir a RSSF como sendo auto gerenciável e tendo a característica de ser auto organizável, ou seja, ela cria automaticamente as rotas de comunicação entre os sensores. Além das áreas funcionais de uma gerência de redes tradicional (configuração, falhas, desempenho, segurança e contabilidade), a arquitetura *MANNA* sugere a utilização de níveis de gerência (gerência de negócio, gerência de serviços, gerência de rede, gerência de elementos de rede e elementos da rede), e uma terceira dimensão, chamada de funcionalidades da RSSF (configuração, manutenção, senosoriamento, processamento e comunicação) como ilustrado na figura 3.2.

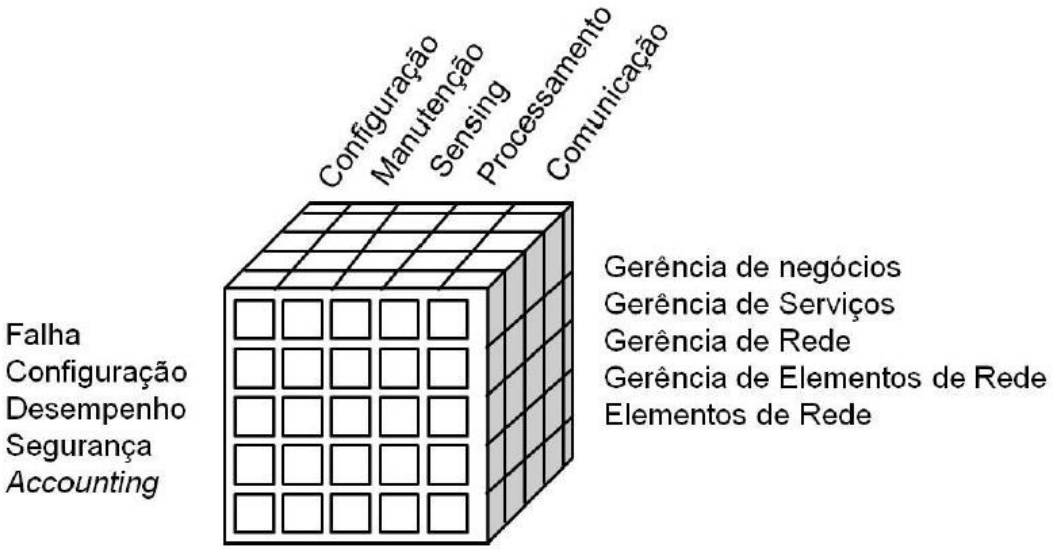

Figura 3.2 - Arquitetura de gerência MANNA [RUIZ, 2003]

É através deste conjunto de funcionalidades que a arquitetura propõe uma forma de auto-organizar os sensores por configuração, baseada no sensoriamento realizado por cada um dos sensores.

A auto-organização presente em *MANNA* propõe mecanismos de correlação em diversas grandezas de modo que os sensores possam tomar decisões sobre a organização da rede, principalmente sobre rotas. Isso explica a arquitetura de gerência tridimensional *MANNA*, que não abandona os conceitos anteriores de gerência de redes de dados, mas apresenta uma extensão para a arquitetura clássica.

Os parâmetros dos planos de gerência no universo auto-organizável, como em *MANNA*, se correlacionam e agem sobre a rede de modo a adaptá-la a diversas situações. Esta extensão foi fundamental para a proposta de *MANNA*, pois segundo defendem os autores, uma gerência centralizada como o SNMP não poderia ser utilizada para a gerência distribuída desta arquitetura. A gerência *MANNA* utiliza a monitoração da rede para tomar decisões internamente, não necessariamente tendo uma interface com gerência centralizada.

O arcabouço proposto em *MANNA* é um trabalho conceitual e este trabalho apresenta uma implementação de uma das propostas conceituais do *MANNA*.

Independente da forma de gerência proposta, o objetivo das arquiteturas *WSNMP* e *MANNA* é caracterizar a forma tradicional de gerência de redes, do tipo centralizado, como sendo inviável para a RSSF. A justificativa é que a gerência de redes

tradicional é uma arquitetura centralizada, em que os usuários requisitam informações diretamente aos elementos de rede através de agentes de gerência, e para se implementar uma gerência capaz de manter uma infra-estrutura auto organizável o protocolo implementado deve possuir características de interação entre máquinas, tornando a RSSF orientada a monitoração de dados e escondendo as características da rede. Com isso espera-se que o comportamento da rede, do ponto de vista topológico, não necessite de interação com os usuários da rede. Ela irá se definir da melhor forma possível, por algoritmos próprios [GEORGEFF, 2004] [DEB, 2001].

3.3. Propostas baseadas na integração com SNMP

Na literatura existem algumas propostas de gerência de RSSF utilizando o protocolo SNMP, sendo realizada a integração deste protocolo com a RSSF de diversas formas. Um tentativa de se implementar o *IPv6* em redes de sensores sem fio é uma das técnicas mais recentes, visando o reaproveitamento da pilha arquitetura *TCP/IP* e assim adaptar um protocolo de gerência de redes mais facilmente [CHAUDHRY, 2010]. Tendo acesso a um protocolo já conhecido, engenheiros e desenvolvedores podem adaptar as características de uma rede *IPv6*, através da gerência, às necessidades de monitoramento de uma RSSF. A integração do *IPv6* a uma RSSF é normatizado pelo *IETF* através do *6LoWPAN* ou *IPv6 over Low Power Wireless Personal Area Network* [COLLITI, 2011]. Através desta técnica pretende-se alcançar os sensores através de endereços *IPv6* e implementar a pilha de protocolo de gerência nos próprios sensores, sem a utilização da traduções via *Proxy*, essa proposta permite a utilização do SNMP para a gerência da RSSF, uma vez que os sensores são enxergados como elementos de uma rede *IP*, mas para a sua implementação os sensores devem possuir uma capacidade maior de hardware, permitindo que a arquitetura *TCP/IP* seja implantada diretamente nos mesmos. Então apesar de permitir a gerência da RSSF utilizando o SNMP, o *6LowPAN*, apresenta uma abordagem diferente à utilizada nesse trabalho uma vez que os sensores podem ser alcançados diretamente sem a necessidade de utilização de um agente *proxy*.

Outra alternativa é a utilização de agentes intermediários de gerência de RSSF conectados a um *Proxy* através de um protocolo proprietário, como o LiveNCM, ou *Live Node Non Invasive*, *Context-Aware and Modular Management Tool* [JACQUOT, 2010]. Nesta arquitetura um agente intermediário se comunica com um *proxy*,

responsável por criar a interface entre a RSSF e o protocolo LiveNCM. O mesmo agente intermediário se comunica com um agente SNMP externo que implementa a interface entre o agente intermediário e a gerência de RSSF via rede *TCP/IP*. Neste caso têm-se um *proxy* utilizando não só o protocolo SNMP, mas também implementando uma camada intermediária para a adaptação da gerência via SNMP à RSSF. O LiveNCM apresenta uma proposta semelhante à utilizada nesse trabalho, utilizando um agente *proxy*, mas com a diferença do protocolo de comunicação ser proprietário.

Outra proposta analisada está em [CYRIACO, 2012]. Nela também foi proposta a utilização de um agente *proxy* para fazer a interface entre a RSSF e a rede *IP*. Este agente implementa diretamente as pilhas de comunicação de RSSF e SNMP em um único hardware. Deste modo a RSSF foi considerada um nó da rede *TCP/IP*, disponibilizando dados de gerência da RSSF e monitorações através de um único método de gerência via comandos e eventos SNMP. Também foi proposta uma MIB que se baseou nos planos de gerência de rede (PGR) e plano de gerência de dados (PGD). O PGR contém informações como nível de sinal, canal e potência de transmissão, já o PGD contém dados como temperatura e temperatura média.

Essa última proposta é bem próxima à estratégia usada nesse trabalho, diferenciando na implementação do agente proxy, já que o mesmo foi montado como um hardware embarcado rodando Linux, e neste trabalho foi utilizado um computador com o mesmo sistema instalado além de um módulo de comunicação para gerenciar a RSSF. Outra diferença diz respeito às MIBs utilizadas, uma vez que em [CYRIACO, 2012] a MIB foi definida para um ambiente bem específico, que era o do trabalho. Enquanto que neste trabalho, foi proposta uma extensão de uma MIB já existente e que tentou englobar ao máximo informações possíveis de serem gerenciadas em uma RSSF.

4. Projeto de gerência SNMP para RSSFs

Para que fosse possível a conclusão do trabalho, as etapas descritas a seguir foram realizadas. Em um primeiro momento foi avaliado um projeto de RSSF montado para realizar o monitoramento da Mata do Paraíso em Viçosa-MG. Esse projeto adotou a estrutura de agente *proxy*, mas não possuía padronização para disponibilização das informações via SNMP. Além disso só era possível realizar consultas referentes aos valores observados pelos sensores.

Este trabalho propõe que a RSSF seja considerada mais um elemento da rede TCP/IP, como tantos outros dispositivos, que também são gerenciados. Portanto, é natural que as estratégias largamente utilizadas para a gerência de redes sejam também apropriadas para a gerência de RSSF.

O protocolo SNMP, foi implementado visando a gerência de grandes redes de computadores, nos quais a pilha *IP* é uma constante presença [STALLINGS, 1999]. Uma grande vantagem de um cenário deste tipo é que uma dada informação, mesmo que extraída de equipamentos distintos e diferentes, pode ser tratada de uma forma única por uma estação de gerência de redes. Desta forma, uma informação de perda de pacotes na interface de rede de um servidor de uma rede *IP* tem o mesmo significado da informação de perda de pacotes na interface de uma estação do usuário e é contabilizado da mesma forma.

Essa característica da rede de gerência baseada no protocolo SNMP tem como base um recurso importante desse protocolo que é a definição dos objetos de gerência ou OID (*Object IDentifier* ou Identificador de Objeto), que nada mais são do que uma representação simbólica para uma determinada informação presente na rede, e através desses objetos a gerência trata as informações de forma unificada. No protocolo SNMP estas informações são traduzidas pela Base de Informação de Gerência (MIB).

A questão então é como introduzir o conceito de gerência de redes em uma RSSF, satisfazendo as características de uma rede de sensores sem fio, visando a integração com ferramentas de gerência já concebidas, por motivos práticos, devendo ser viável para a implementação das funcionalidades presentes no protocolo de gerência SNMP.

De uma forma bem direta, a implementação de uma pilha de protocolos de gerência no sensor implica em dificuldades computacionais relevantes. Se considerarmos a presença de um agente em cada um dos sensores, implicaria em

memória e processamento extra, em detrimento à autonomia do sensor. Então é necessário utilizar uma estratégia que aproveite a estrutura da RSSF, utilizando-a como rede de transporte, sem que grandes implementações sejam necessárias diretamente nos sensores. Então a estratégia seguida neste trabalho foi a utilização do Agente *Proxy*, ou *Gateway*, que se localiza entre a rede de sensores e a rede de gerência. A visão da estação de gerência, ou servidor, com relação a RSSF é de um elemento de rede IP, sendo gerenciado pelo conjunto de MIBs aplicáveis ao modelo de rede de sensores. O conjunto de sensores é encarado como uma única entidade, que representa a monitoração de um espaço físico, mas que é gerenciado a partir de um único ponto.

A estrutura básica do modelo proposto pode ser caracterizada por uma Base de Informação de Gerência (MIB), um *gateway* (proxy), um nó sensor e um servidor. Os nós fazem consulta aos sensores e transmitem essas informações ao *gateway*. O *gateway* é responsável por organizar o fluxo de informações oriundas da rede de sensores e formatar esses dados de maneira que eles possam ser consultados via SNMP, consultas essas que serão realizadas pelo servidor. Além disso, o servidor vai gerar gráficos com estes valores consultados. Para que seja possível a utilização do protocolo SNMP é necessário a existência de uma MIB, que contenha as informações referentes aos nós que podem ser consultadas. Essa MIB fica armazenada no *gateway*. A figura 4.1 mostra uma representação do sistema proposto neste trabalho.

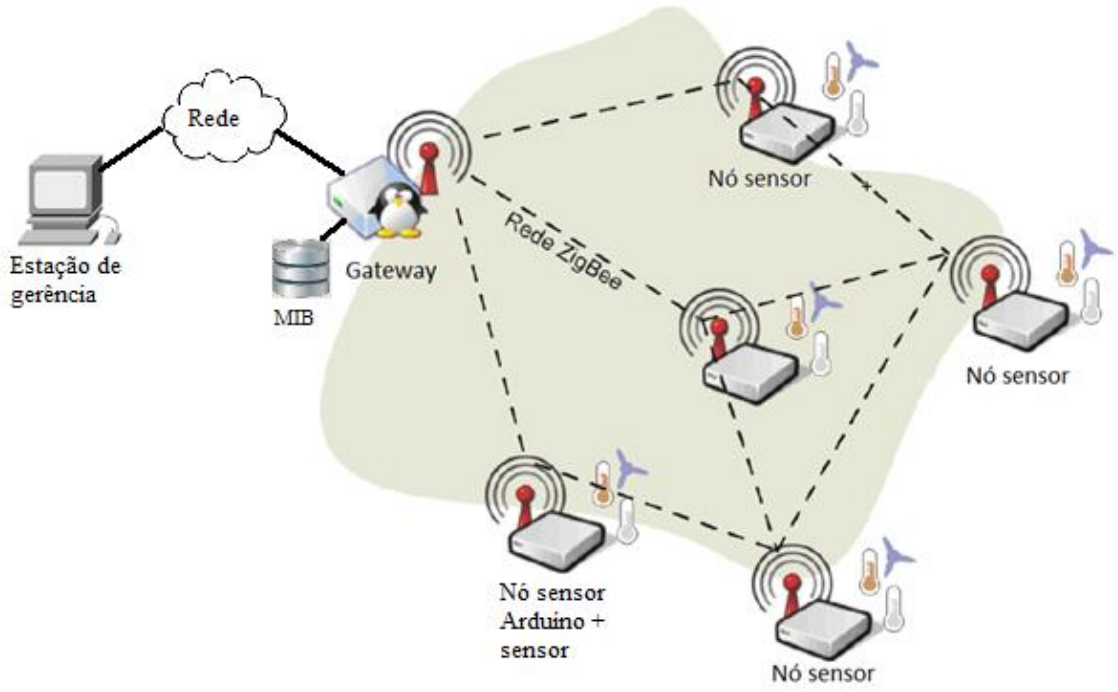

Figura 4.1 - Modelo simplificado do ambiente de testes.

4.1. Descrição da MIB

A MIB (*Management Information Base* ou Base de Informação de Gerência) pode ser representada como uma árvore de objetos, onde as ramificações indicam os diversos níveis, que se especializam a medida que a OID se define. O item final da OID, ou folha, deve ser sempre o objeto a ser monitorado. Caso uma estação de gerência omita a parte final de uma OID em uma busca, esta receberá do agente uma lista referente a todos os objetos pertencentes ao nível superior da OID [CYRIACO, 2012].

Para a definição da MIB buscou-se abranger o máximo possível de informações pertinentes a uma RSSF, e neste contexto a MannaMIB proposta em [SILVA, 2005] foi considerada como um interessante ponto de partida para a definição da MIB proposta neste trabalho.

A MIB proposta utilizou os sete grupos propostos na MannaMIB, sendo eles: energia, topologia, transceptor, processador, sensor, administração e hierarquia. Os grupos foram mantidos devido ao fato de englobar os principais componentes e informações possíveis de serem gerenciadas em uma RSSF. Em adição a esses sete grupos foi criado o grupo Traps, uma vez que o MannaMIB não cobriu a utilização de *traps*, e esse tipo de recurso foi considerado importante para a gerência de RSSFs, uma vez que é necessária a detecção de falhas de forma rápida nas redes de sensores, permitindo que ações sejam tomadas antes que nó sensor pare de funcionar. Além disso, o intervalo de consulta em uma RSSF pode ser grande, e uma situação crítica deve ser identificada o mais rápido possível. Cada um desses grupos também pode ser chamado de sub-árvore.

Com relação aos objetos de cada grupo, grande parte foi mantido como proposto no MannaMIB, enquanto alguns outros foram alterados ou adicionados de acordo com a necessidade. Os objetos que foram alterados ou adicionados estão em destaque nas figuras de cada classe a seguir.

As características específicas das RSSFs, vistas na seção 1, fazem com que seja necessária a definição de objetos que devem ser controlados pelo sistema de gerenciamento. A definição da MIB utilizada para redes tradicionais não é suficiente e pode até ser considerada inadequada, possuindo objetos que não são necessários para RSSFs. Portanto foi proposta a MIB a seguir, baseada no MannaMIB que contém objetos relacionados com RSSFs, que podem ser gerenciados. Estes objetos gerenciáveis devem obedecer aos critérios de visibilidade, onde alguns parâmetros

devem ser apenas de leitura, ou aplicáveis somente ao comando GET SNMP, ou de escrita e leitura, ou aplicáveis aos comandos GET e SET SNMP. A MIB foi dividida hierarquicamente em grupos, como se segue: Energia, Topologia, Transceptor, Processador, Sensor, Administração, Hierarquia e Traps. Para a definição da MIB foi utilizada a sintaxe SMIv2. Os objetos e grupos estão detalhados no Anexo I deste trabalho.

Propõe-se o posicionamento da MIB WSN-MIB-UFV na sub-árvore *Experimental* representada pela OID 1.3.6.1.3, ou de uma forma mais detalhada, *iso(1) .org(3).dod(6).internet(1).experimental(3).*

O grupo energia, definido em [SILVA, 2005] para representar os possíveis tipos de fonte de energia utilizados nós sensores, tem algumas funcionalidades relacionadas à fonte de energia. Usando objetos deste grupo é possível medir desde o tipo de fonte de energia utilizado (bateria, solar, eólica, dentre outros), passando por informações de configuração (marca da bateria - nome do fabricante), além de objetos usados na gerência de falha e de desempenho (Voltagem residual). O objeto voltsFonteEnergia, traz a informação da quantidade de volts que está sendo gerada pela fonte de energia, considerando neste caso que a bateria do sensor possa ser recarregada utilizando energia solar, por exemplo. Já com o objeto voltagemResidual, é possível obter a quantidade de volts sendo gerada pela bateria da nó sensor, o objeto VoltagemTotal contém a quantidade máxima de volts entregue pela bateria, enquanto que a Voltagem de Operação define a quantidade mínima de volts necessária para o funcionamento do nó sensor, informação esta que pode ser utilizada para detectar a eminência de cessar o funcionamento do mesmo.

Na figura 4.2 pode-se visualizar a representação da sub-árvore energia.

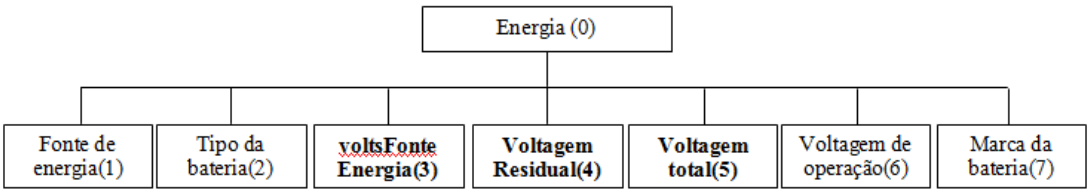

Figura 4.2- Sub-árvore Energia.

O grupo Topologia possui informações referentes à disposição física dos nós sensores. Com os objetos deste grupo é possível saber informações como a localização geográfica do nó sensor, considerando latitude, longitude e altitude (Coordenadas X, Y

e Z), além de informações de configuração (É móvel, velocidade, direção, tipo de descoberta de localização), também possui um objeto que pode ser utilizado na gerência de falhas (Vizinhos). Na figura 4.3 está representada a sub-árvore topologia.

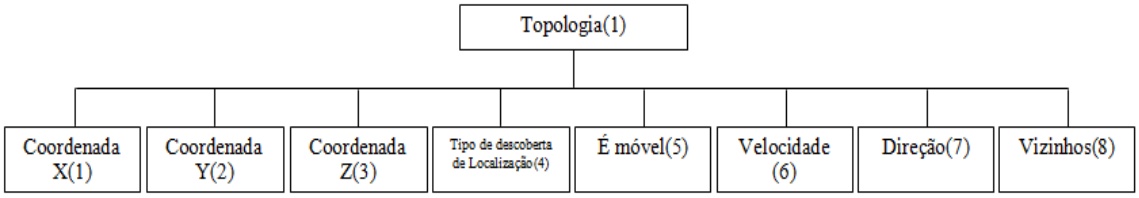

Figura 4.3 - Sub-árvore topologia.

O grupo Transceptor traz informações sobre o transmissor utilizado no nó sensor. Os objetos definidos neste grupo provêem informações pertinentes à gerência de configuração (Estado operacional, Canal de transmissão e tipo), além de informações para a gerência de falhas (rssi). Na figura 4.4 está a representação gráfica da sub-árvore transceptor. Os objetos RSSI e Canal de Transmissão não constavam na MannaMIB e trazem respectivamente informações sobre potência de sinal de recepção medida em dBm, e do canal utilizado para a transmissão dos pacotes.

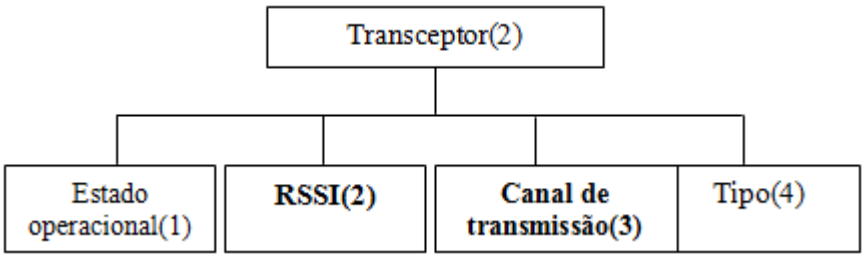

Figura 4.4 - Sub-árvore transceptor.

O grupo Processador é responsável por conter as informações do hardware utilizado como base de processamento do nó sensor, e apesar de possuir o nome processador, contém informações que vão além do chip de processamento utilizado (estado operacional do processador, frequência, consumo por instrução, mips e tipo de processador), como os objetos de memória RAM e ROM livres e totais. Estes objetos podem ser utilizados tanto na gerência de falhas quanto na de configuração. A sub-árvore processador está representada na figura 4.5.

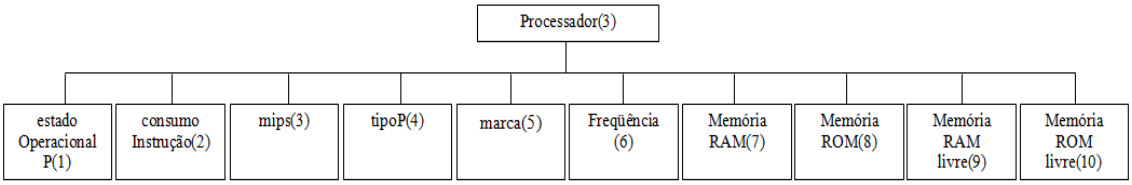

Figura 4.5 - Sub-árvore processador.

O grupo Sensor contém as informações referentes aos sensores presentes em cada um dos nós da RSSF. Utilizando seus objetos é possível gerenciar informações de configuração (quantidade de sensores, consumo, última calibração, marca do sensor, unidade de medida) além de informações de falha (taxa de erro) e também o valor obtido pelo sensor, podendo ser de diversos tipos (temperatura, umidade, luminosidade, entre outros). Para a implementação deste grupo ele foi considerado uma tabela, já que em um mesmo nó posso ter mais de um sensor. Cada sensor é indexado (indice Sensor), e a partir deste índice obtêm-se as informações específicas à cada sensor. Abordagem semelhante é utilizada com o grupo interfaces presente na figura 2.2. Na figura 4.6 está representada a sub-árvore sensor.

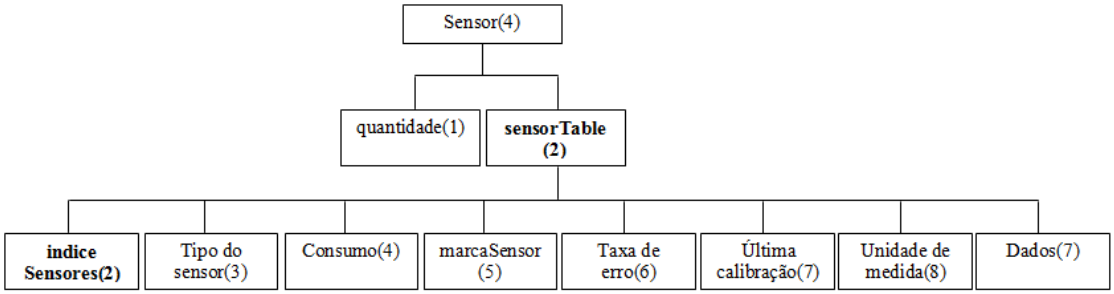

Figura 4.6 - Sub-árvore sensor.

O grupo Administração traz informações administrativas referentes aos nós sensores, e possibilita a obtenção de informações tanto da gerência de configuração (estado administrativo, é lider) quanto informações da gerência de contabilidade (dados enviados e recebidos e mensagens de gerência enviadas e recebidas). Pode-se visualizar a sub-árvore administração na figura 4.7.

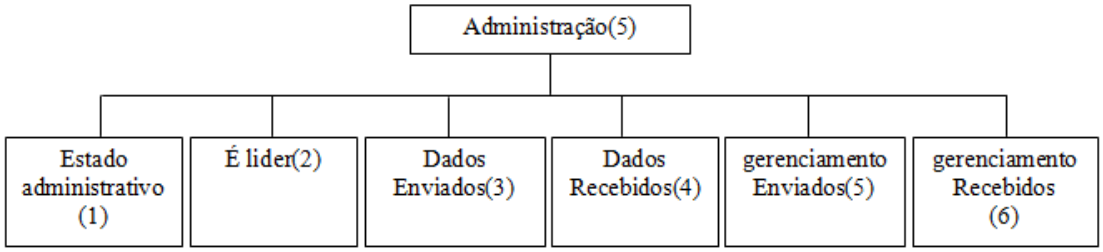

Figura 4.7 - Sub-árvore administração.

O grupo Hierarquia é responsável por conter informações sobre como a rede de sensores está organizada e fornece informações do grupo ao qual o sensor pertence (identificador do grupo, todos integrantes do grupo, tipo de formação do grupo, integrantes do grupo ativos e reserva e nível de hierarquia) informações estas que podem ser utilizadas na gerência de configuração e de falhas. Este grupo também foi considerado como uma tabela, uma vez que em uma RSSF, podem existir vários grupos hierárquicos. Os objetos quantidade e groupTable foram adicionados para permitir esta representação em tabela. O primeiro indica a quantidade de grupos hierárquicos presente na RSSF, enquanto que groupTable contém as informações de cada grupo. Na figura 4.8 está representada a sub-árvore hierarquia.

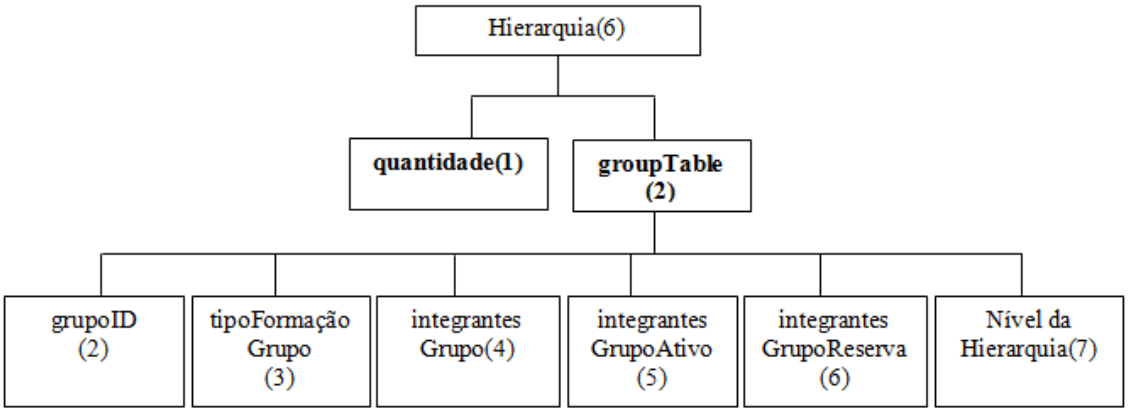

Figura 4.8 - Sub-árvore hierarquia.

Por último têm-se o grupo Traps, elemento não presente na MIB proposta em [SILVA, 2005] e que foi definida como um objeto gerenciável, para padronizar o envio de eventos da WSN-MIB-UFV. A RFC 1215 [ROSE, 1991] padroniza a definição das traps como parte da MIB. Contém os seguintes objetos Id, limite inferior, média e limite superior. A ID é utilizada para identificar de que se trata a trap, e os objetos limite

inferior, média e limite superior podem ou não serem utilizados de acordo com a necessidade. Por exemplo, para o acaso de medir o nível mínimo de sinal, seria definida uma ID para este evento, e um valor para o objeto limite inferior. Nas figuras 4.9 e 4.10 são mostradas respectivamente a sub-árvore Traps e todas as sub-árvores da MIB WSN-MIB-UFV.

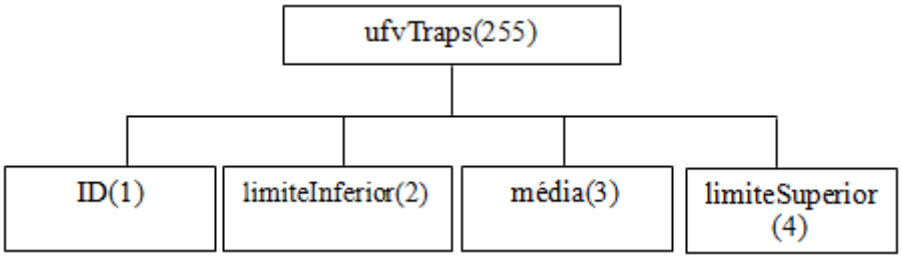

Figura 4.9 - Sub-árvore Traps.

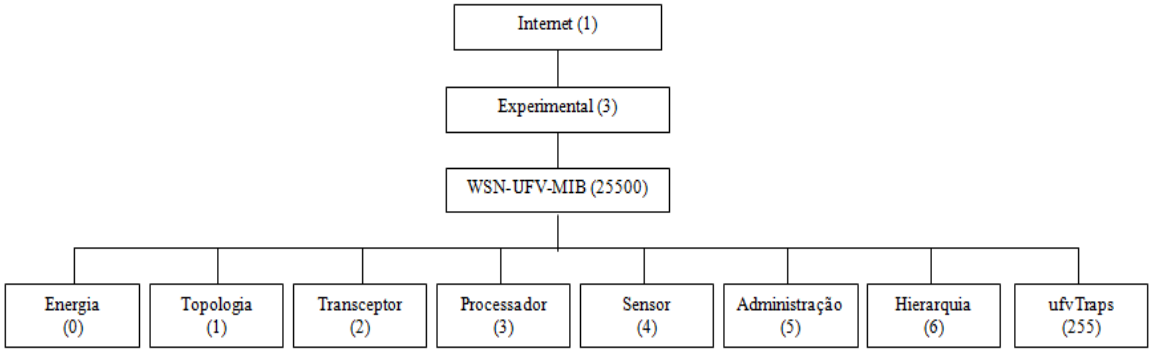

Figura 4.10 - Representação da MIB

4.2. Descrição dos Elementos do Projeto

O nó é o elemento responsável pela obtenção da medida do fenômeno a ser sensoriado, e para tanto possui sensores de tipos variados. Também deve possuir uma interface de comunicação para enviar as informações coletadas, a qual deve ser a mesma presente no agente *proxy*.

O uso do SNMP requer que todos os agentes bem como as estações de gerência suportem UDP e IP. Isto limita a gerência direta de alguns dispositivos e exclui outros, que não possuem qualquer suporte a qualquer parte da pilha de protocolos *TCP/IP*. Além disso existem vários pequenos sistemas que implementam o *TCP/IP* para a

utilização através de suas aplicações, mas para os quais não é desejável adicionar o peso do SNMP, da lógica de agentes e da manutenção da MIB.

Para acomodar estes dispositivos que não implementam o SNMP, o conceito de proxy foi definido. Desse modo um agente SNMP age como um *proxy* para um ou mais dispositivos, ou seja o agente SNMP representa os dispositivos que estão atrás do *proxy* [STALLINGS, 1998].

O *gateway*, ou agente *Proxy*, é responsável por receber as informações dos sensores, e portanto deve possuir uma interface de comunicação do mesmo tipo do nó, organizá-las e disponibilizá-las para consulta via SNMP.

O agente *proxy* SNMP faz com que seja possível o monitoramento e controle de elementos de rede que não possuem o protocolo de transporte ou de gerência. Além disto permite que os elementos a serem gerenciados através do *proxy*, possuam configurações inferiores de hardware, já que não há a necessidade de implementação de protocolos complexos e a necessidade de espaço de armazenamento maior para o armazenamento da MIB.

Por último, o servidor se comunica com o *gateway*, utilizando a pilha de protocolos TCP/IP, e é responsável por realizar a consulta às informações dos sensores utilizando consultas(GET), enviar configurações para os nós sensores (SET) ou receber eventos gerados pelos sensores (TRAP). Além disso, a informação consultada é organizada e armazenada em gráficos para facilitar a sua análise.

O servidor é considerado como nível mais alto na hierarquia da gerência de redes, visto que é o elemento que possui uma característica única entre todos os elementos da rede de gerência, que é a propriedade de tradução dos dados na camada de aplicação. Basicamente ele se refere às aplicações responsáveis por traduzir as informações recebidas do agente *proxy*, através de um determinado protocolo, para uma linguagem humanamente compreensível. Ao pensarmos na RSSF, informações como potência de sinal e canal de transmissão são informações deste tipo de rede, mas que necessitam de tradução para os padrões da linguagem humana.

5. Implementação e testes

Levando-se em consideração o caráter prático escolhido para a abordagem do objeto pesquisado, neste caso a definição de uma MIB para RSSF e a implementação em um sistema real de gerenciamento de elementos rede sem fio com agentes do tipo *proxy*, foram utilizados no trabalho protótipos didáticos e desenvolvimento de procedimentos de testes, para verificar a correção da implementação e avaliar o desempenho do conjunto proposto no trabalho.

Utilizando a documentação de fabricantes e gerando protótipos baseados em micro controladores, *software* embarcado e rádio transceptores, chegou-se à arquitetura dos nós sensores. Buscando a simplificação dos protótipos, um desenvolvimento financeiramente viável e para facilitar a replicação do experimento, optou-se por soluções *open source* (código aberto) tanto de software quanto de hardware.

5.1. Implementação do protótipo

O ambiente de testes no qual a MIB foi implementada e testada é composto pelos elementos descritos a seguir.

5.1.1. Nó sensor

Os nós foram implementados considerando-se a idéia de utilização de tecnologia barata e eficaz. Foi utilizado como referência para confecção dos nós o ambiente de desenvolvimento Arduíno. Essa plataforma de hardware livre é composta por um micro-controlador de placa única, uma linguagem de programação baseada primariamente em C++ e um ambiente de desenvolvimento integrado (IDE) [ARDUÍNO, 2013].

Cada nó pode conter inúmeros sensores e o tipo utilizado depende diretamente da informação que se deseja coletar. A maior parte dos sensores responde através de um sinal elétrico a um estímulo, isto é, convertem o fenômeno sensoriado em um sinal elétrico. Nesse caso, podem ser chamados de transdutores. O transdutor é geralmente composto por um elemento sensor e uma parte que converte a reação sensoriada em um sinal elétrico.

A comunicação utilizou como base o *ZigBee* que foi construído para ser um complemento do padrão IEEE 802.15.4 [IEEE, 2003], o qual define as camadas físicas e

de controle de acesso ao meio (MAC) para redes de áreas pessoais (PAN) de baixo custo e baixa taxa de transmissão.

O *ZigBee* padroniza as camadas superiores da pilha de protocolo. A Camada de Rede é a responsável pela organização e promoção de roteamento em uma rede com vários nós. A Camada de Aplicação provê um *framework* para o desenvolvimento e comunicação de aplicações distribuídas. A Camada de Aplicação compreende o *framework* de aplicação, os objetos de dispositivos ZigBee (ZDO) e a sub-camada de aplicação (APS). O *framework* de aplicação pode possuir até 240 objetos de aplicação (APO), que são módulos de aplicação desenvolvidos pelo usuário que fazem parte da aplicação ZigBee. O ZDO provê serviços que permitem que os objetos de aplicação se descubram e se organizem em uma aplicação distribuída. A APS oferece uma interface para serviços de dados e de segurança para o APO e ZDO [BARONTI, et al, 2007]. Uma visão geral da pilha de protocolos ZigBee é mostrada na figura 5.1.

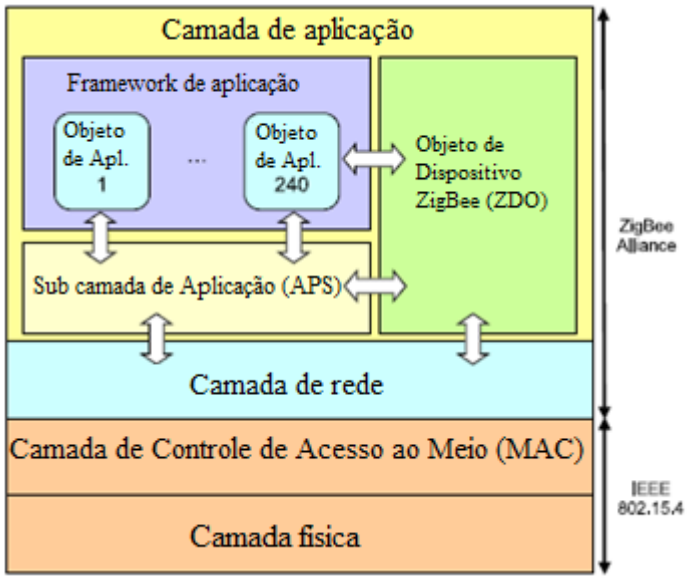

Figura 5.1 - Arquitetura funcional de camadas e pilha de protocolos do ZigBee

Foi utilizado o módulo *ZigBee* chamado *XBee* [DIGI, 2013]. O mesmo foi configurado no modo API, sendo assim alterada sua configuração padrão, qual seja o modo de operação transparente. O modo API é baseado em quadros e assim permite a manipulação dos pacotes, o que torna possível, inclusive enviar comandos de configuração remotamente. Outras vantagens do modo API são: a não necessidade de se entrar em modo de comandos, e inutilizar o rádio temporariamente, para modificar o destinatário de suas transmissões; Receber informações sobre o sucesso ou falha de cada

pacote de Rádio Frequência (RF) transmitido; Identificar o endereço do remetente de cada pacote recebido, sem a necessidade de configurações adicionais.

O hardware utilizado neste trabalho foi montado para um projeto final do curso de graduação em Ciência da Computação da UFV do aluno Waldir Denver Muniz Meireles Filho, e este nó sensor está representado na figura 5.2. O esquema da ligação entre o Arduíno e o rádio XBee está representado na figura 5.3.

Neste sensor foram implementados um sensor de luminosidade que utiliza uma entrada analógica do micro controlador, um divisor de tensão para a bateria, que também utiliza uma porta analógica e um sensor RHT03 para realizar o sensoriamento de temperatura e umidade. Maiores informações sobre este sensor e como pode ser realizada sua instalação no Arduíno podem ser encontradas em [RHT03, 2013].

Figura 5.2 – Nó sensor utilizado no trabalho

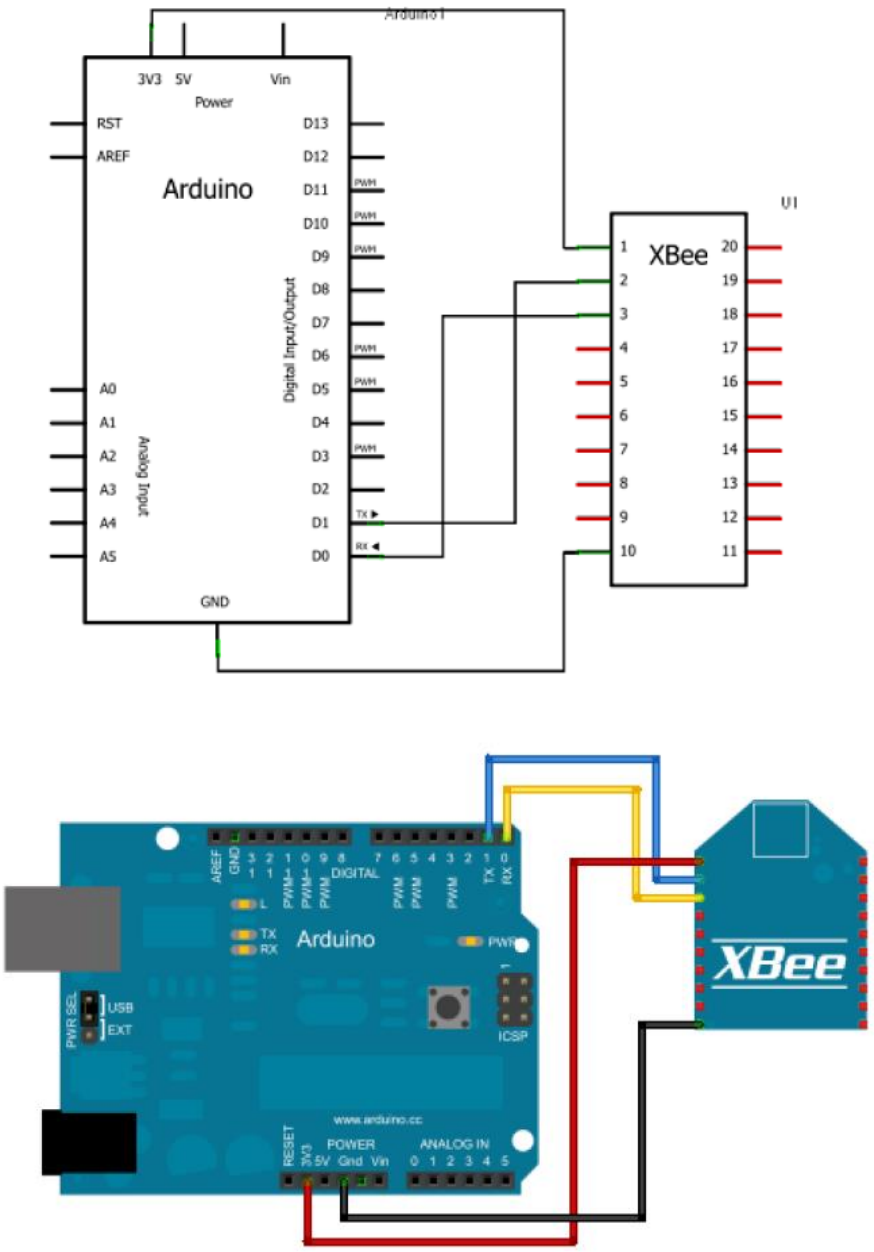

Figura 5.3 – a)Representação da ligação entre o rádio e o Arduíno b)Detalhe da ligação na prática.

5.1.2. Gerenciamento

O protocolo de gerência de redes utilizado foi o SNMP, que descreve uma metodologia para o gerenciamento de dispositivos numa rede de computadores. Para tanto, a especificação precisa ser implementada. O projeto Net-SNMP [NET-SNMP, 2013] é uma implementação livre do protocolo de gerenciamento SNMP, distribuída

através da licença BSD [BSD, 2013]. Dentre os conjuntos de utilitários disponibilizados pelo Net-SNMP destaca-se um extenso agente SNMP com muitas funcionalidades.

A MIB WSN-MIB-UFV foi definida, utilizando o padrão SNMPv2-SMI. As SMIs (*Structure of Management Information*) são regras utilizadas para definir objetos que podem ser acessados por um protocolo de gerência de redes [SLOMAN, 1994]. Como a ferramenta Net-SNMP não possui a WSN-MIB-UFV, foi necessária a criação do código fonte responsável por determinar como os dados são obtidos para serem preenchidos na MIB. Este código foi escrito na linguagem C. O método escolhido para disponibilizar a WSN-MIB-UFV foi a compilação do código em um subagente que se comunica com o Net-SNMP através do recurso AgentX, definido na RFC 2741 [DANIELE et al, 2000], a qual propõe um ambiente padronizado para agentes SNMP extensivos. A RFC 2741 define entidades de processamento chamadas agente mestre e subagente, além de um protocolo chamado AgentX, utilizado para a comunicação entre eles, além dos elementos do procedimento utilizado pelo agente extensível para processar mensagens do protocolo SNMP. Portanto, o AgentX permite que consultas sejam feitas ao sub-agente através do agente principal que é o Net-SNMP.

Na estação de gerência foi utilizado o Cacti, que é um software que utiliza o Net-SNMP [CACTI, 2013]. O Cacti é fundamentalmente uma ferramenta Web, que utiliza prioritariamente gráficos para exibir as informações gerenciadas. Pode-se definir a requisição, armazenamento e difusão das informações como as três tarefas básicas que caracterizam seus princípios de operação e que serão utilizadas para gerar gráficos com as informações consultadas nos agentes. Na figura 5.4 está mostrado o modelo do funcionamento básico do Cacti.

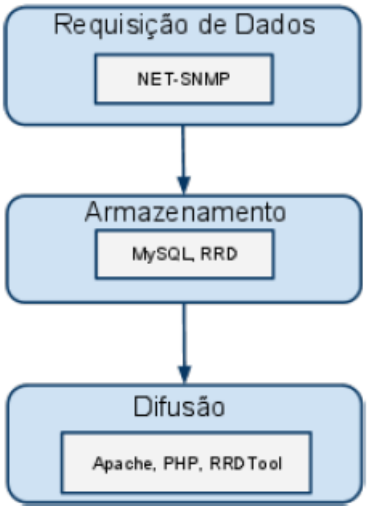

Figura 5.4 - Funcionamento básico do Cacti

5.1.3. SNMP *Sensor Driver (*snmpSD)

Além da utilização dos elementos citados anteriormente, foi preciso desenvolver um novo sub-sistema de software chamado SNMP *Sensor Driver* (snmpSD), que faz o mapeamento da RSSF para o Agente SNMP instalado no *gateway*. O snmpSD é fundamental para o funcionamento dessa metodologia, pois os sensores, em sua essência, não são elementos gerenciáveis. E após a agregação do componente de gerenciamento adicional provido por esse novo elemento, é possível comunicar os sensores ao Gerente SNMP.

O conjunto Arduíno, rádio Zigbee e elemento sensor formam um nó sensor, capaz de ler as variáveis do ambiente, analisá-las e transmitir esta informação, se necessário. O dado, quando transmitido pelo rádio do nó é então recebido pelo rádio ZigBee "Coordenador", entregue no formato de um pacote via porta serial para a placa principal do *gateway* onde está instalado o snmpSD. Essa informação é então armazenada juntamente com os dados de todos os outros sensores da rede.

Também faz parte do snmpSD uma interface de consulta que fornece acesso indireto aos dados dos sensores. E é dessa forma que o Agente SNMP, que também reside no *gateway*, interage com a RSSF e mapeia os sensores na MIB.

Foi desenvolvido um software de controle do nó escrito em C++ utilizando a IDE de desenvolvimento do Arduíno, onde está implementada toda a lógica do nó. Este software é responsável pela sincronização do relógio do sensor, consultar os elementos sensores, fazer o processamento local dos dados e transmitir as informações quando necessário. O software também manipula os quadros que devem ser entregues ao rádio ou que são recebidos por ele. Quando o arduíno é ligado, o software inicia um sistema de configuração do hardware. Entre esses processos destacam-se: a) configuração dos pinos do arduíno como entrada e saída; b) inicialização das variáveis; c) configuração da porta serial e; d) montagem de um sistema de arquivos simples para armazenamento dos dados dos sensores. Depois disso, o sistema precisa sincronizar-se com o coordenador da RSSF. Então um pacote especial é enviado para o coordenador requisitando dados de configuração, sendo que uma das informação recebidas é a hora do sistema. A etapa seguinte consiste em esperar que o pacote de configuração chegue até a estação. Para isto, o sistema fica monitorando o buffer de entrada de dados. Caso seja recebido qualquer pacote diferente do pacote de configuração, uma nova solicitação é feita. Quando o pacote de configuração é recebido de volta, ele é desmontado e as

informações necessárias para sua configuração são extraídas. Na figura 5.5 pode ser vista a lógica de processos implementadas nesse software.

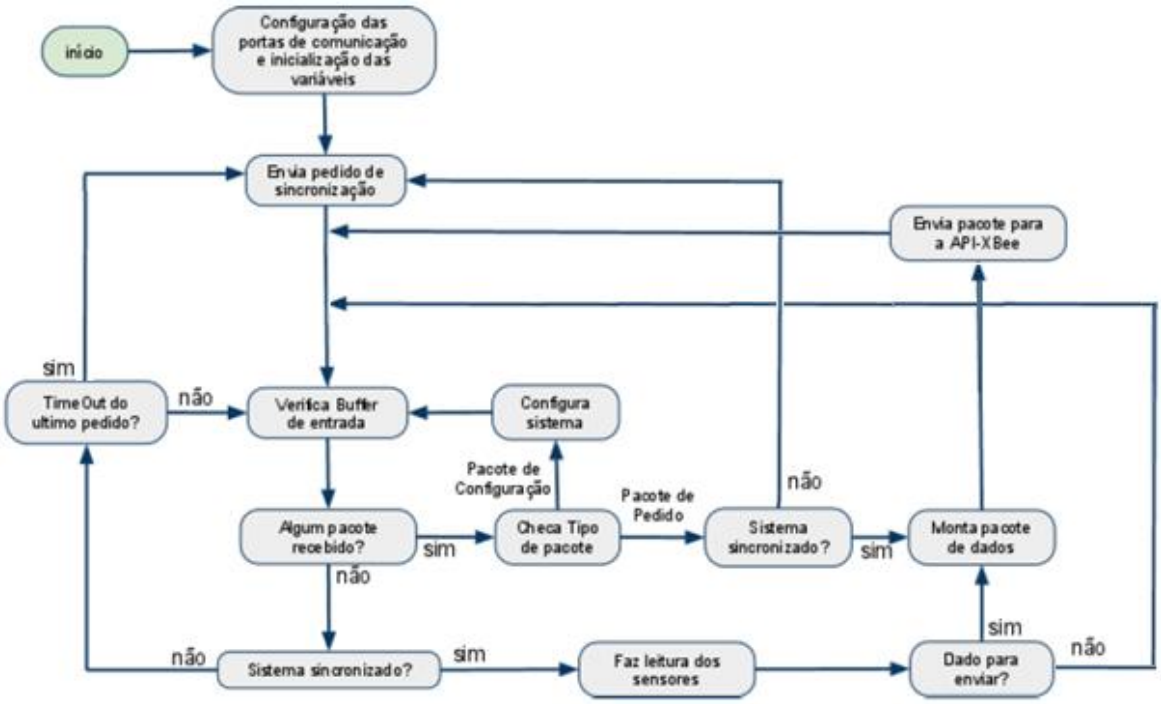

Figura 5.5 - Diagrama de Estados do Software Executado pelo Nó

5.1.4. Agente *proxy* (*gateway)*

O *gateway* da rede de sensores é composto por um computador rodando o sistema operacional Linux e um XBee configurado como coordenador da rede de sensores. As informações de todos os sensores da rede são então entregues a esse equipamento. Essas informações são tratadas pelo *driver* de sensores e disponibilizadas pelo Agente SNMP *Sensor Driver* (snmpSD).

O snmpSD foi dividido em dois componentes de software para realizar as tarefas de gerência, através do contato direto com nós da RSSF, e a tradução dessas interações para elementos externos. O componente de gerência do snmpSD utiliza um arquivo pré-configurado no padrão XML para descrever os detalhes dos nós, o mapa XML está detalhado no Anexo II, e um exemplo de configuração do mesmo está exposto na figura 5.6. Baseado nessas informações, ele entra em contato com todos os elementos físicos que deverá monitorar e monta um banco de dados volátil que representa o estado instantâneo da rede. Assim, os elementos da estação física existente em campo possuem

sua representação lógica no *Driver*. Os eventos e alterações são pré gerenciados nesse elemento antes de serem comunicadas ao Gerente SNMP. A figura 5.7 exemplifica como é feita esse representação abstrata.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <config>
  - <sockets numreg="1">
      <socket timeOutReconect="5" panId="0" idhH="0" idhL="0" baudrate="9600" port="/dev/ttyUSB0" type="XBee2" name="Xbee2_1"/>
    </sockets>
  - <stations numreg="1">
    - <station name="NODE2" socket="Xbee2_1">
      - <sensors numreg="10">
          <sensor type="arduino" name="umidade" dataType="int" list="false" latencyUpdate="5" sintax="HM01"/>
          <sensor type="arduino" name="radiacao" dataType="int" list="false" latencyUpdate="5" sintax="RD01"/>
          <sensor type="arduino" name="temperatura" dataType="int" list="false" latencyUpdate="5" sintax="TP01"/>
          <sensor type="arduino" name="relogio" dataType="int" list="false" latencyUpdate="3" sintax="TIME"/>
          <sensor type="arduino" name="bateria" dataType="int" list="false" latencyUpdate="5" sintax="VT02"/>
          <sensor type="arduino" name="solar" dataType="int" list="false" latencyUpdate="5" sintax="VT03"/>
          <sensor type="xBee" name="RSSI" dataType="int" list="false" latencyUpdate="3" sintax="DB"/>
          <sensor type="xBee" name="NOME" dataType="string" list="false" latencyUpdate="3" sintax="NI"/>
          <sensor type="xBee" name="Adrres" dataType="int" list="false" latencyUpdate="3" sintax="SL"/>
          <sensor type="xBee" name="Canal" dataType="int" list="false" latencyUpdate="3" sintax="DB"/>
        </sensors>
      </station>
    </stations>
</config>
```

Figura 5.6 - Exemplo de configuração utilizada no arquivo XML.

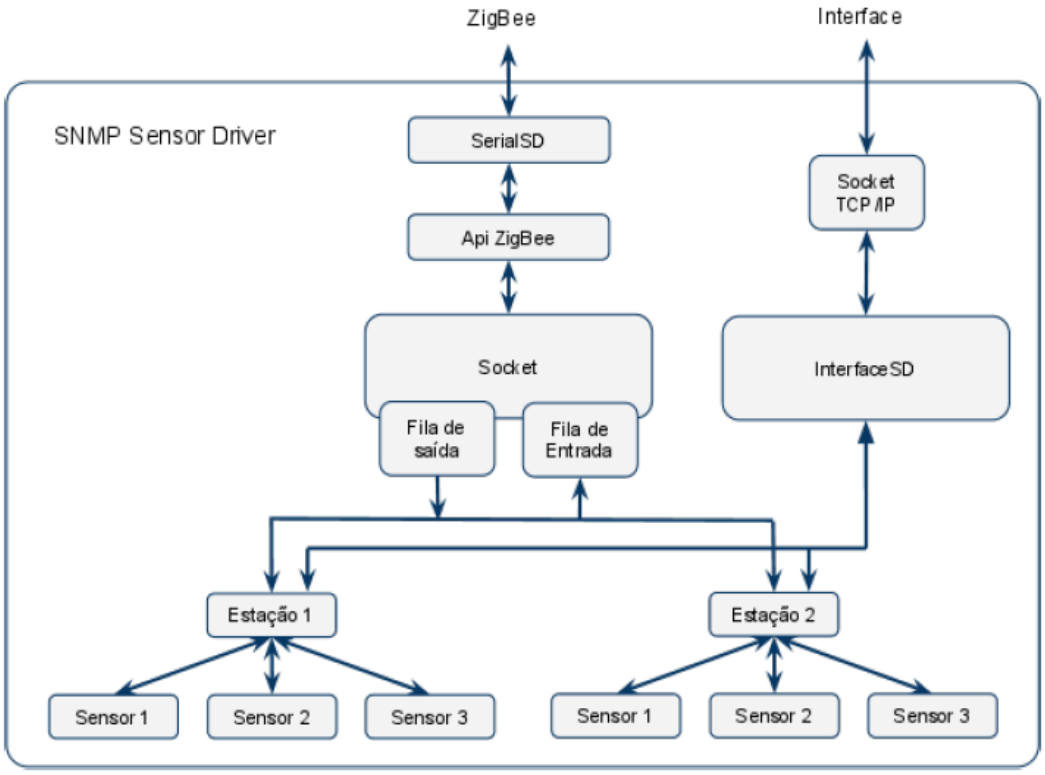

Figura 5.7 - Diagrama dos Componentes do snmpSD

Uma característica importante desse software é sua divisão em camadas hierárquicas. Cada camada usa as funções da própria camada ou da camada inferior, para esconder a complexidade do processo e transparecer as operações. Essa técnica foi baseada nos princípios de operação da pilha TCP/IP.

Os pacotes recebidos ou enviados devem passar pelas camadas do sistema, adicionando ou removendo informações nestes pacotes a fim de se adequar a camada que ele se encontra.

Quando uma consulta deve ser feita a um nó sensor na estação em campo, ela deverá passar através de cada camada abstrata antes de chegar finalmente ao dispositivo físico. Tudo começa adicionando um comando de consulta (*Token*) juntamente com o nome do sensor ao qual ele deve ser entregue. Por exemplo, para a consulta de temperatura é enviado o *token* TP01, para saber o relógio do nó sensor o *token* utilizado é TIME. O sensor então entrega esta informação para a representação abstrata da estação ao qual este pertence. E em seguida a camada do *socket*, após receber estes dados de cada sensor, junta os pacotes e o marca com o endereço daquela estação na rede ZigBee.

Antes de ser entregue ao rádio de transmissão, o pacote deve ser formatado no padrão que o *ZigBee* entende. Depois disso, o pacote será recebido e entregue corretamente ao rádio destinatário. O processo é exemplificado na figura 5.8 que mostra a montagem de um pacote para transmissão. O tamanho do pacote vai variar dependendo da quantidade de consultas simultâneas a serem feitas a cada nó sensor. Para o caso de consulta a mais de um nó sensor, os pacotes serão montados em separado.

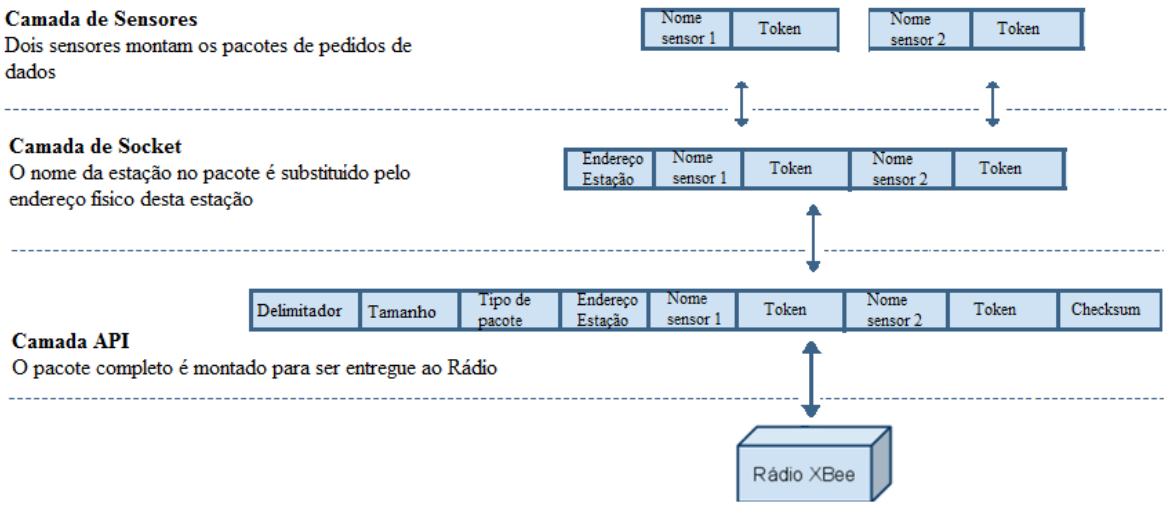

Figura 5.8 - Montagem de pacotes de Consulta de Dados

Todo este processo é realizado no sentido inverso do lado do receptor a fim de extrair as requisições direcionadas individualmente a cada sensor corretamente. Quando

a estação responde à solicitação enviando o pacote com a leitura, o quadro de resposta deve seguir o processo ao contrário, removendo a informação dos pacotes até receber o dado de cada sensor e armazená-lo na representação abstrata. A seguir estão listados os componentes do snmpSD.

O SerialSD é um componente que implementa um Buffer. Ele controla todos os dados que chegam à porta serial. Assim tudo que sai ou entra é armazenado provisoriamente no SerialSD. Ele ainda fornece uma interface de configuração da porta serial e faz sinalização da informação contida na porta como o tamanho do buffer no momento. O SerialSD é importante também para manter uma compatibilidade entre outros componentes que fazem acesso à porta serial executando tarefas que são comuns entre eles.

A API (*Application Programming Interface*) do Arduíno Xbee-Arduino [ARDUINO, 2013] foi utilizada para extrair as informações recebidas contidas no buffer da porta serial. Toda informação enviada ou recebida pelo rádio XBee está formatada na forma de quadros. Cabe a este software extrair as informações recebidas contidas no buffer da porta serial.

A biblioteca Xbee-Arduino possibilita a comunicação com Arduino de XBees no modo de API, com suporte tanto para Série 1 (802.15.4) como para Série 2 (ZB Pro / ZNet). Na figura 5.9 está exemplificado um quadro do *ZigBee*.

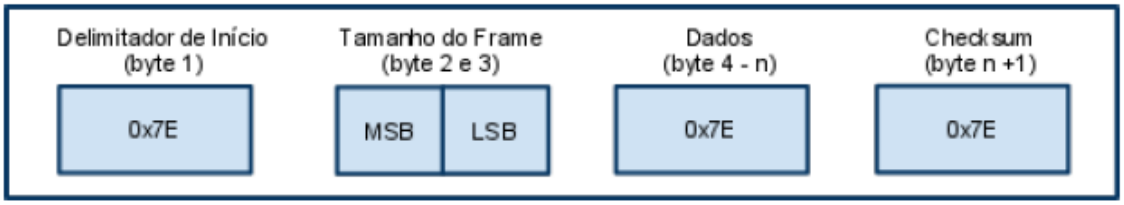

Figura 5.9 - Formato do quadro utilizado pelo ZigBee

O socketSD é o componente que tem acesso ao meio de comunicação. Toda informação entre as estações de sensores e o *gateway* da rede passa por este componente. Aqui as informações recebidas pela API Xbee são entregues informando o endereço da estação que enviou o pacote.

Este componente mantém uma tabela onde estão cadastradas todas as estações em campo com seus endereços e nome, sendo que o nome é necessário devido ao fato do XML ser configurado com esta informação. Esta informação é importante, pois todos os pacotes que chegam ao componente para ser entregues à estação estão

marcados com o nome da mesma. Como o XBee necessita do endereço do rádio para entregar os pacotes, fica sob responsabilidade do socketSD a tarefa de saber os endereços.

O socketSD começa se configurando a partir do arquivo XML citado anteriormente, que informa a ele a porta que se encontra o hardware de transmissão de dados assim como suas configurações para comunicação. A seguir o componente as configura. A próxima etapa do processo é se conectar ao hardware e testar a conexão enviando um pacote de teste ao hardware. Se tudo correr bem, o hardware de transmissão responderá com uma mensagem, caso contrário o socketSD esperará alguns segundos e tentará novamente mais tarde. Caso a conexão seja completada, a próxima tarefa será conseguir os endereços das estações que se encontram em campo. Assim um pacote especial é enviado para todos os sensores da rede que deverão responder com um pacote contendo o seu nome e endereço. O sistema possui uma fila de entrada e saída de dados. Na fila de saída estão os pacotes que devem ser entregues para as estações em campo e na fila de entrada, estão os pacotes recebidos pelas estações.

O sistema, depois de conectado, testa se existe algum pacote na fila de saída para ser enviado. Depois de enviar um pacote, se houver, a próxima etapa será tratar os pacotes que chegam. Nesse caso, existem três tipos de pacotes que podem ser tratados pelo socketSD. O pacote de "*Node Identificator*" é o pacote enviado pelas estações informando o seu nome e endereço. Este pacote será tratado pelo próprio socketSD, extraindo esta informação e atualizando sua tabela de endereços. Os pacotes de dados são divididos em consulta ao rádio e consulta ao arduíno. Estes então são tratados, e colocados na fila de entrada de pacotes recebidos. Caso o pacote recebido não seja de nenhum tipo citado acima, então este é descartado. A figura 5.10 está representado o diagrama de fluxo de socketSD.

O componente de software estação é usado para formar uma estrutura de dados que representa a estação em campo. Uma das funções contidas na estação é monitorar a fila de entrada do *socket* a que ele está associado e verificar se há algum pacote que lhe pertence. Assim o pacote é retirado da fila e tratado e repassado para cada sensor pertencente a ele.

O sensor é o componente final da estrutura de dados. É nele onde as informações dos sensores em campo são guardadas e disponibilizadas para consulta. Além de guardar os dados, o sensor possui funcionalidades de sincronia com o sensor em campo. Nele é configurada a periodicidade de atualização que uma estação deve manter para o

sensor. Estas configurações são feitas a partir de um arquivo XML carregado na inicialização do sistema. Nesta configuração também é informado o *Token* (um comando) passado à estação para requisitar o dado do sensor. Alguns dados necessitam ser guardados em forma de lista. Esta informação também é configurada. Neste caso o sensor mantém esta lista de dados.

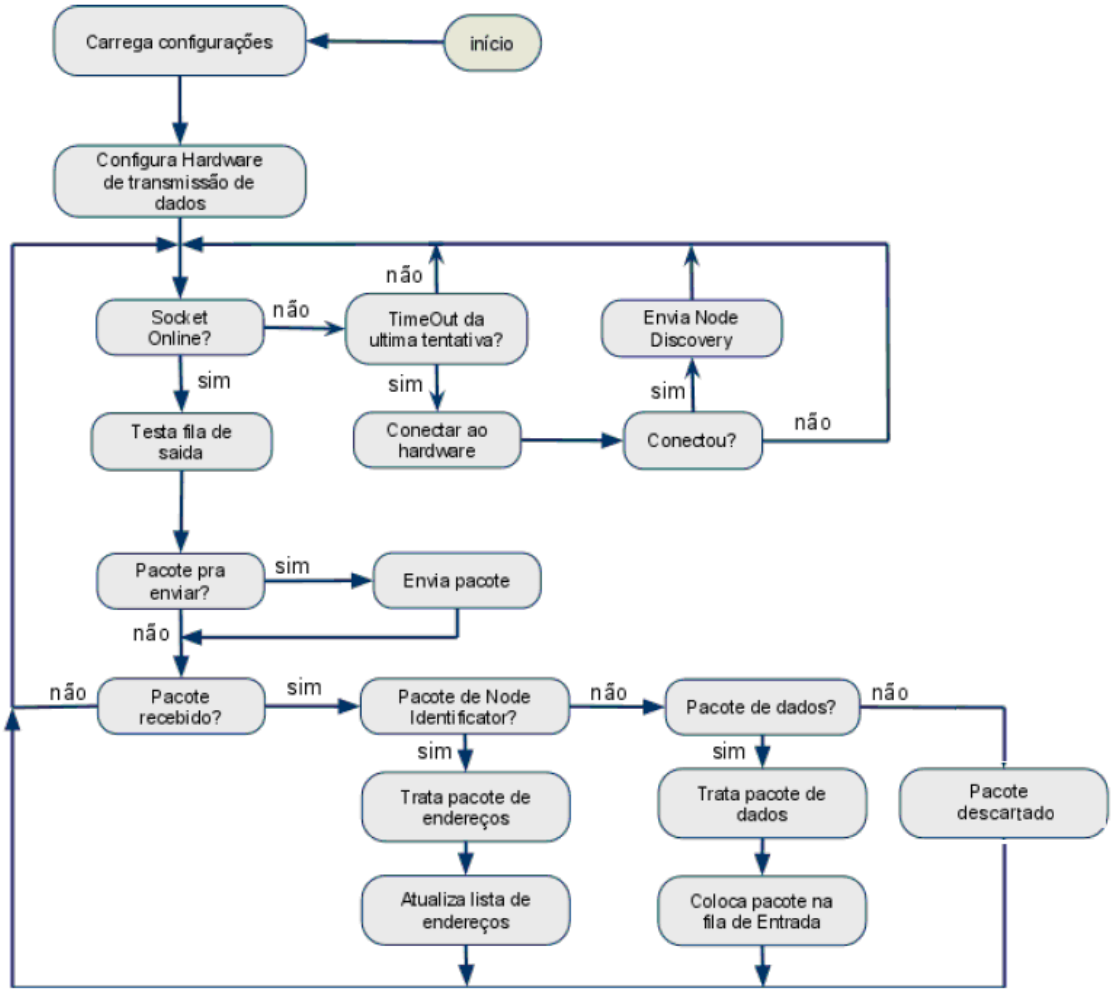

Figura 5.10 - Diagrama de fluxo do socketSD

Também no *gateway* existe a interfaceSD que é um software simples utilizado para criar uma interface de comunicação entre o Agente SNMP e o snmpSD. Toda interação externa feita com o Sensor Driver deve utilizar este software. Ele apresenta um conjunto de instruções que torna possível extrair informações de sensores e das estações monitoradas pelo Sensor Driver. O agente SNMP utiliza esse componente do *Driver* para inserir os dados dos sensores na MIB. Além das funções de consulta, esse componente também é usado para configurar o snmpSD. Segue um exemplo de uma

consulta feita pelo interfaceSD: interface getsensor <Nome do nó> bateria. Neste caso será lido o valor da tensão produzida pela bateria do nó o qual tiver seu nome especificado.

Para que as informações dos sensores possam ser transmitidas via protocolo SNMP, elas devem ser descritas como Recursos Gerenciáveis, representados como objetos no agente SNMP. Uma base de dados com os valores atualizados dos sensores em campo deve ser mantida e disponibilizada sempre que a agente SNMP requisitar. Assim um sistema de software foi desenvolvido para fornecer esta informação.

Os três sistemas mostrados na figura 5.11 são instalados no mesmo computador. O Agente SNMP e o snmpSD permanecem em funcionamento enquanto o computador estiver ligado. Quando um dado de sensor é requisitado, o Agente SNMP executa o InterfaceSD que se conecta via socket TCP/IP ao SNMP Sensor Driver para coletar a informação contida neste. Apos o dado ser entregue ao SNMP, o InterfaceSD desconecta-se do SNMP Sensor Driver e então ele se fecha.

Figura 5.11 - Interação entre os softwares do gateway

5.1.5. Servidor

O software de difusão e armazenamento de dados (Cacti) foi instalado em outro computador com o sistema Linux, e é configurado para consultar o Agente SNMP em intervalos pré-definidos solicitando o valor dos sensores. No Cacti, estes dados são armazenados e indexados conforme a data e hora da consulta.

5.2. Testes

O planejamento dos testes realizados visou a busca por evidências sobre a validade da proposta, demonstrando a aplicabilidade em sistemas reais. Para atingir tal objetivo, foram realizados experimentos com os sensores descritos neste trabalho com o intuito de simular situações reais de operação dos sensores em interação com um Agente Proxy.

Para realizar os testes foi utilizado o ambiente ilustrado na figura 5.12.

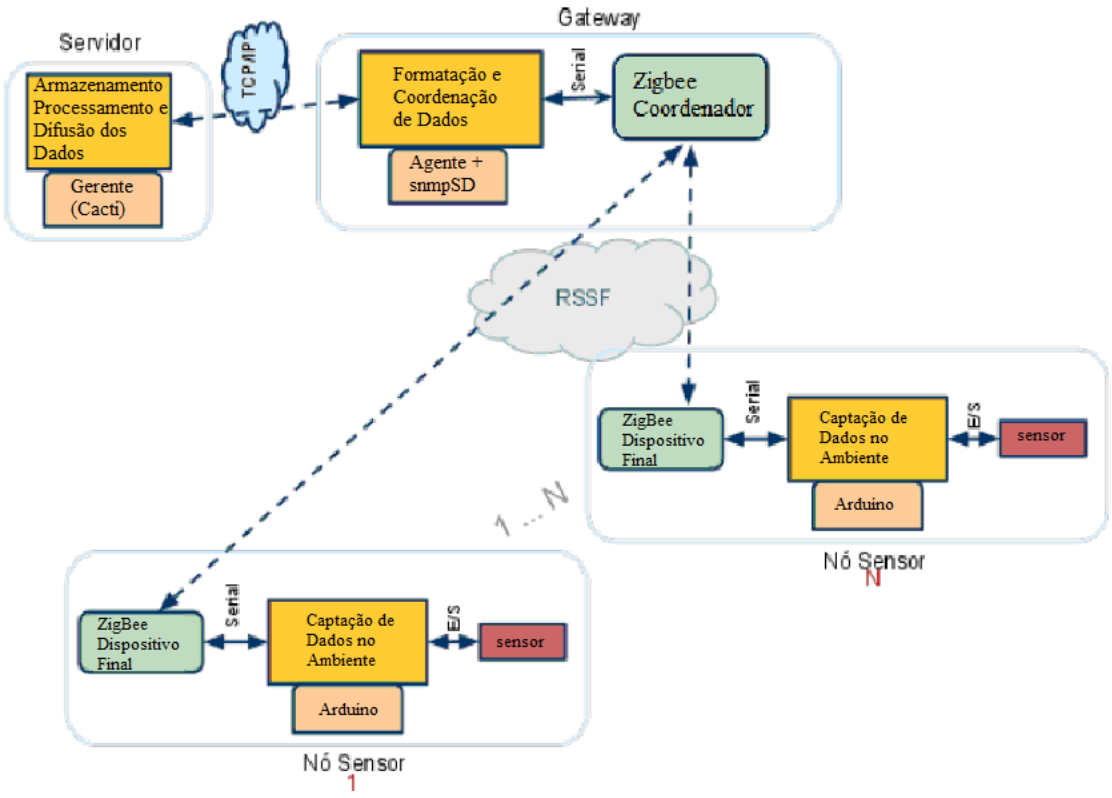

Figura 5.12 - Modelo do ambiente de testes.

Como o objetivo é simular situações reais de operação dos sensores, foi definido que a comunicação entre o servidor que contém o Cacti e o *gateway*, que conterá o agente SNMP seja feita utilizando o protocolo IEEE 802.11. Este tipo de configuração, permite que no mundo real, o servidor fique até a quilômetros de distância do *gateway*, utilizando a internet. Esse último por sua vez, devido à limitação de alcance da rede ZigBee (IEEE 802.15) deve ficar próximo aos nós sensores.

Para a realização dos testes foi utilizado um nó sensor equipado com sensores de temperatura, umidade e luminosidade, portanto estas informações foram mapeadas pelo Cacti. Além dessas informações foram mapeadas a energia residual da bateria, o nível de sinal do nó, e a quantidade de Volts gerada pela fonte de energia alternativa do nó sensor, como exemplo um painel solar. O ambiente utilizado para testes foi uma sala da Diretoria de Tecnologia da Informação da UFV, onde existe movimentação de pessoas, e vários sinais de rede 802.11.

O nó sensor também foi configurado para enviar alertas (*traps*) em caso de mudança do valor coletado para um valor menor do que o mínimo definido. Então foi realizado um teste com o valor do nível de sinal do nó, onde este foi afastado do proxy SNMP, fazendo com que o valor fique abaixo do pré-determinado como valor mínimo, causando a geração do alerta informando sobre esta situação. Esse valor mínimo pode ser alterado pelo comando SNMPSET. A trap foi configurada para enviar a mensagem sinal baixo toda vez que este evento acontecer.

Na figura 5.13 vemos uma sequência de pacotes capturada utilizando a ferramenta wireshark [WIRESHARK, 2013]. Nesta captura foram filtrados os pacotes direcionados para a estação de gerência que possui o endereço IP 192.168.1.123, e também pode-se notar o comando get realizado, com o posterior envio da trap do agente para o gerente devido ao fato do valor da RSSI estar abaixo de -90 dBm, sendo este o valor mínimo determinado.

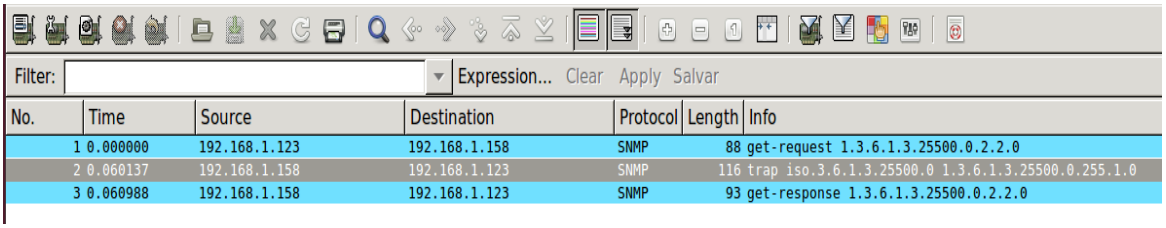

Figura 5.13 – Captura da *trap*

Em seguida o Cacti foi instalado e configurado para realizar consultas via SNMP e disponibilizar as informações em gráficos. O mapeamento foi realizado utilizando as OIDs definidas para os elementos temperatura, umidade, luminosidade, carga restante na bateria e nível de sinal e fonte de energia. Como exemplo, as informações referentes à temperatura são acessadas através da OID 1.3.6.1.3.25500.0.4.2.2.9.0, ou de uma forma detalhada, *iso(1).org(3).dod(6).internet(1).experimental(3).ufv(25500).sensor(4). indice (2). temperatura(2).dados(9)*. O zero no final foi necessário devido ao fato da WSN-MIB-UFV proposta ter sido toda registrada no sub-agente representada no modo escalar [ZOHO, 2012].

A figura 5.14 mostra o gráfico de luminosidade gerado pelo Cacti. Nele pode-se perceber que no período da noite o seu valor ficou bem próximo de zero, e já com o amanhecer, o mesmo começou a subir. Também pode-se notar nele um aumento no valor da luminosidade por volta das 18 horas, momento este onde já havia anoitecido. Isto aconteceu devido ao fato de ter sido acesa a lâmpada da sala onde o sensor se

encontra. Vale ressaltar que no gráfico de luminosidade mostrado, não está sendo apresentada nenhuma leitura antes das 18 horas, devido ao fato do nó sensor ter sido instalado próximo deste horário.

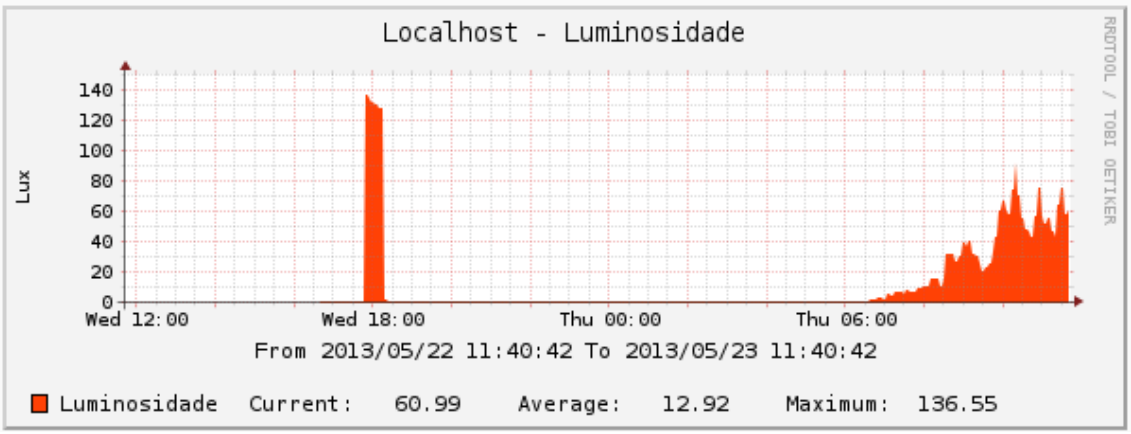

Figura 5.14 - Gráfico de luminosidade gerado pelo cacti

Nas figuras 5.15, 5.16 e 5.17 temos respectivamente os gráficos de nível de sinal (RSSI), temperatura e umidade.

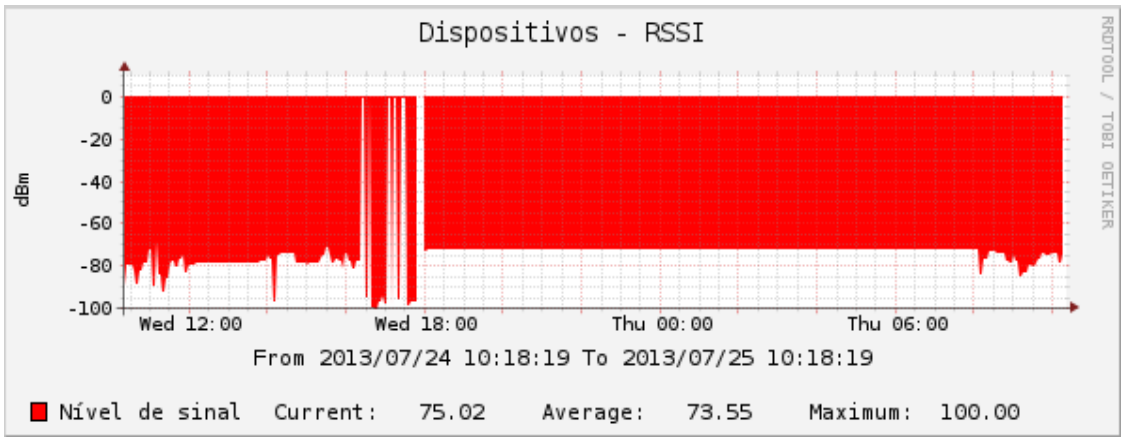

Figura 5.15 - Gráfico de nível de sinal (RSSI) gerado pelo cacti

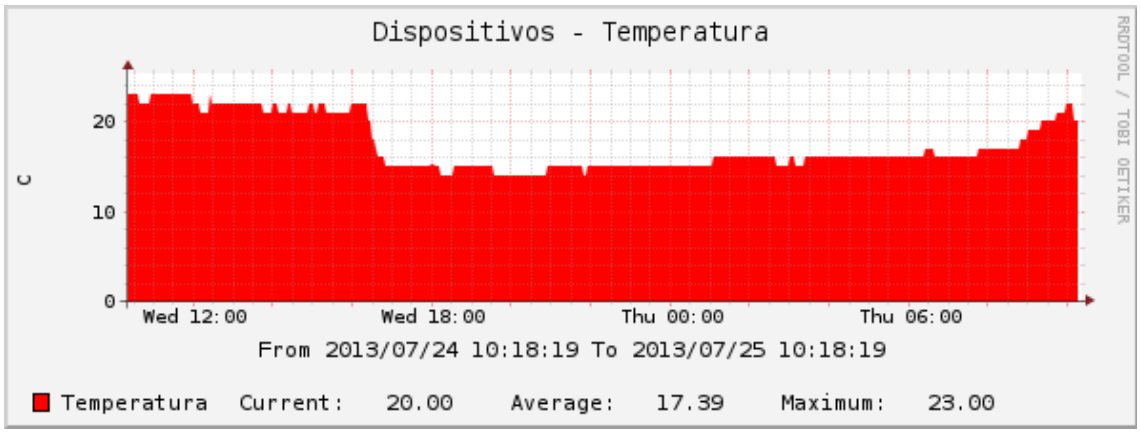

Figura 5.16 - Gráfico de temperatura gerado pelo cacti

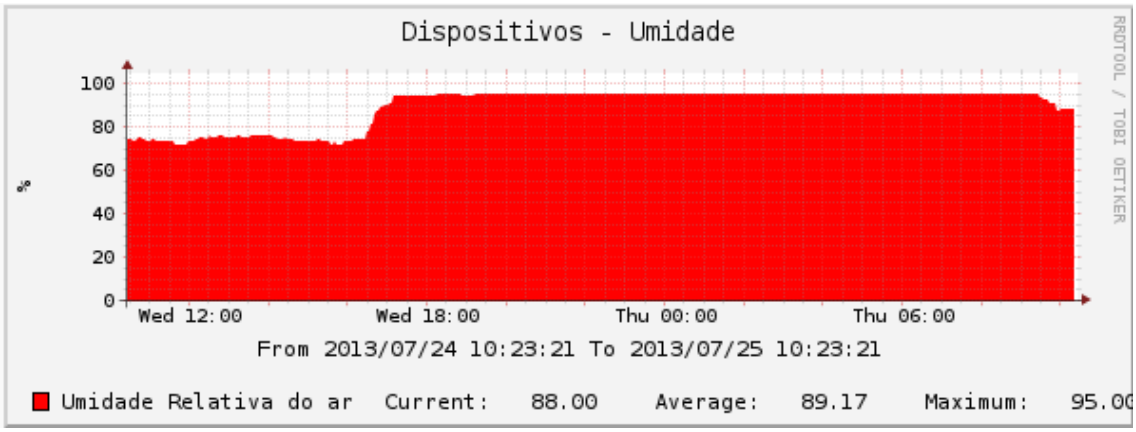

Figura 5.17 - Gráfico de umidade gerado pelo cacti

O servidor pode ser acessado através do seguinte endereço http://200.235.177.203:8080/cacti e para o acesso deve ser informado o usuário guest e a senha guest2013.

6. Conclusões e trabalhos futuros

6.1. Conclusões

Neste trabalho foi implementado um sistema de gerenciamento de redes de sensores sem fio utilizando o protocolo SNMP, um agente proxy e uma MIB estendida, à partir da MIB proposta em [SILVA, 2005].

O objetivo foi mostrar que é possível utilizar o protocolo SNMP para a gerência de RSSF e, devido à limitação de recursos dos nós sensores, é difícil implementar toda a estrutura necessária para a utilização do SNMP diretamente nos nós sensores. Assim, foi definido um agente *proxy* que ficou responsável por fazer a intercomunicação com os dispositivos sensores na RSSF fazendo o mapeamento dos comandos SNMP para comandos executáveis pelos nós sensores.

A extensão da MIB proposta em [SILVA, 2005] foi necessária para poder incluir algumas funcionalidades consideradas importantes para o gerenciamento de um RSSF. Dentre estas funcionalidades, destaca-se o gerenciamento de falhas através do grupo Traps. Como os eventos de falha são comuns em RSSFs a possibilidade de contabilizar, organizar e classificar as *traps* geradas pelos dispositivos sensores pode melhorar o entendimento sobre as falhas ocorridas em uma aplicação. A MIB estendida seguiu o formato de definição padronizado e foi incluída inicialmente no ramo de MIBs experimentais.

Para a realização da implementação proposta nesse trabalho foram utilizadas soluções livres tanto de *hardware* quanto *software*, reduzindo o custo e facilitando assim a implementação da solução e uma possível reprodução deste experimento por outros interessados.

Operacionalmente, a gerência de RSSF se mostra transparente aos administradores e usuários dos recursos da RSSF. Esta é uma vantagem proporcionada pela utilização de um protocolo aberto e por uma MIB de gerência capaz de ser aplicada, independentemente do tipo de sensores. A camada de adaptação presente no *proxy* abstrai todas as especificidades da RSSF e traduz para uma linguagem universal, baseada em identificações e variáveis com visibilidade distintas.

Utilizando o agente *proxy*, uma estação de gerência tem a visibilidade dos sensores, a partir de todas as informações definidas na MIB, podendo se concentrar no monitoramento independentemente das grandezas veiculadas aos diferentes valores

possíveis de serem consultadas na MIB. O mesmo conjunto de *hardware* e *software* pode ser utilizado em diversos experimentos, com objetivos distintos, coletando dados a partir de diferentes *OID*s.

Então ao compararmos outras arquiteturas propostas na literatura sobre redes de sensores sem fio, a proposta implementada neste trabalho visa obter uma prova de conceito contra a opinião de que uma gerência de RSSF deve ser alcançada fornecendo funcionalidades de gerência entre os sensores, de modo a implementar a auto organização e a auto gestão. Nesse tipo de gerência as grandezas são tratadas internamente na rede de sensores e não possuem uma interface definida claramente com os sistemas de gerência de fora da rede.

Dessa forma, pode-se dizer que a arquitetura proposta aqui provê uma interface definida com sistemas de gerência já existentes, utilizando um protocolo já definido (SNMP) e dominado tecnicamente por um considerável público de administradores e usuários de rede, facilitando assim a sua aceitação.

6.2. Trabalhos futuros

Na tentativa de criar uma prova de conceito em gerência de sensores, um grupo de variáveis foi selecionado, e embora houvesse a tentativa de englobar o máximo possível de funcionalidades para uma rede de sensores, possivelmente outras variáveis podem ser adicionadas ao modelo com vistas a deixá-lo mais completo. As principais limitações encontradas na confecção dos sensores, no que dizem respeito a *hardware* e *software*, foram financeiras e tecnológicas. Porém, investimentos em novos micro controladores e transdutores podem ajudar a provar novos conceitos em relação à gerência de redes de sensores sem fio.

Além disto, não foi possível implementar no software do agente *proxy* a contagem de mensagens enviadas e recebidas para e de cada nó sensor, portanto o objeto referente a esta informação na MIB não foi preenchida, podendo esta ser uma funcionalidade a ser implementada futuramente.

Outra funcionalidade que não foi possível implementar, foi uma *trap* que avisaria quando ocorresse mudanças bruscas na temperatura, podendo ser utilizada por exemplo para identificar ocorrências de incêndio, alertando imediatamente a estação de gerência.

Todos os objetos gerenciados passíveis de terem seu valor alterados utilizando o comando SET do SNMP, ficaram armazenados no agente *proxy*, e o mais interessante seria que estas alterações fossem realizadas no nó sensor.

Outra questão diz respeito às *traps*, neste trabalho elas são geradas dentro do sub agente SNMP criado. E portanto só serão geradas quando acontecer um consulta ao nó sensor. O ideal seria que o próprio nó sensor verifique os parâmetros desejados dentro de um dado período de tempo, que também poderia ser um valor configurado via SET dentro do mesmo.

Finalmente, seria interessante a implementação de um agente SNMP próprio com todas as funcionalidades e específico para a RSSF. Esse agente já viria com o conteúdo da MIB embutido, não sendo necessária a realização de sua importação. Além disso, esse agente seria o único requisito de instalação, excluindo assim a necessidade de instalação do NET-SNMP.

Referências Bibliográficas

AKYILDIZ, I.; WELLIAN S.; YOGESH S.; CAYIRCI, E. “A survey on sensor networks”. IEEE Communications Magazine, 40(8):102–114. 2002.

AKYILDIZ, I.; MEHMET C. V. “Wireless Sensor Networks”, John Wiley & Sons, 2010.

ALAM, M.; MAMUM-OR-RASHID, M.; HONG, C. “WSNMP: a Network Management Protocol for Wireless Sensor Networks”, Kyung Hee University, Feb. 2008.

ARDUINO. Em http://www.arduino.cc. Acesso em: 01/03/2013.

ARDUINO/XBEE. Em http://www.arduino.cc/en/Main/ArduinoXbeeShield. Acesso em: 03/03/2013.

AN870. "An SNMP Agent for the Microchip TCP/IP Stack", Application Notes, Microchip Technology Inc, 2009.

AZAMBUJA, M. C. “PSWeM: Desenvolvimento e Implementação de Uma Ferramenta Baseada na Web para Gerênciamento de Redes ao Nível de Serviço”. Porto Alegre: PUCRS. Dissertação de Mestrado apresentada ao departamento de Ciência da Computação, 2001.

BARONTI, P.; PILLAI, P.; CHOOK, V. W. C.; CHESSA, S.; GOTTA, A.; HU, Y. F., " Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards", Computer Communications, Volume 30, Issue 7, Maio 2007, Páginas 1655-1695.

BLUMENTHAL, U; WIJNEN, B. “User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)”, RFC 3414, Internet Engineering Task Force, Em http://www.rfc-editor.org/rfc/rfc3414.txt. 2002.

BSD. Em https://www.ibm.com/developerworks/community/blogs/752a690f-8e93-4948-b7a3-c060117e8665/entry/conhecendo_as_licen_C3_A7as_bsd12?lang=en. Acesso em 05/07/2013.

CACTI. The Cacti Group. Em http://www.cacti.net/. Acesso em: 17/03/2013.

CASE, J.; Fedor, M.; Schoffstall, M.; Davin, J. “A Simple Network Management Protocol (SNMP), RFC 1157, Internet Engineering Task Force. 1990.

CASE, J.; McCloghrie, K.; Rose, M; Waldbusser, S. “Introduction to version 2 of the Internet-standard Network Management Framework”, RFC 1441, Internet Engineering Task Force . 1993.

CASE, J.; MCCLOGHRIE, K.; ROSE, M.; WALDBUSSER, S. “Introduction to Community-based SNMPv2”, RFC 1901, Internet Engineering Task Force, Em http://www.rfc-editor.org/rfc/rfc1901.txt. 1996a.

CASE, J.; MCCLOGHRIE, K.; ROSE, M.; WALDBUSSER, S. “Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)”, RFC 1905, Internet Engineering Task Force, Em http://www.rfc-editor.org/rfc/rfc1905.txt. 1996b.

CASE, J.; MUNDY, R.; PARTAIN, D.; STEWART, B. “Introduction and Applicability Statements for Internet Standard Management Framework”, RFC 3410, Internet Engineering Task Force, Em http://www.rfc-editor.org/rfc/rfc3410.txt. 2002.

CHAUDHRY, Shafique A.; BOYLE, George; WEIPING Song; SREENAN, Cormac; , "EMP: A Network Management Protocol for IP-based Wireless Sensor Networks", International Conference on Wireless and Ubiquitous Systems, Sousse, Tunisia. Outubro. 2010.

COLITTI, W.; STEENHAUT, K.; DE CARO, N. “Integrating Wireless Sensor Networks with the Web”, The 10th International Conference on Information Processing in Sensor Networks (IPSN), 2011.

CYRIACO, Frederico S., "Gerência de redes de sensores sem fio: uma abordagem com SNMP." Campinas: PUC Campinas. Dissertação de Mestrado apresentada ao Centro de Ciências Exatas, Ambientais e de Tecnologias, 2012.

DARGIE, W.; POELLABAUER, C. “Fundamentals of Wireless Sensor Networks: Theory and Practice”, John Wiley & Sons, 2010.

DANIELE, M.; WIJNEN, B.; ELLISON, M.; FRANCISCO, D. “Agent Extensibility (AgentX) Protocol Version 1”, RFC 2741, Network Working Group, Em http://www.rfc-base.org/txt/rfc-2741.txt. 2000.

DEB, B.; BHATNAGAR, S.; NATH, B. “A Topology Discovery Algorithm for Sensor Networks with Applications to Network Management”, Tech. Rep. DCS-TR-441, Rutgers University, 2001.

DEF. Departamento de Engenharia Florestal, UFV, Em: http://www.def.ufv.br/infraEstruturaMataParaiso.php. Acesso em 06/07/2013.

DIGI. DIGI International Inc. XbeeTM Product Manual. Disponível em http://ftp1.digi.com/support/documentation/90000982_A.pdf. Acesso em: 01/03/2013.

ESTRIN, D.; GOVINDAN, R.; HEIDEMANN, J. “Embedding the Internet”. Communications of the ACM, 43(5):39–41. (Special issue guest editors). 2000.

FRYE, R.; LEVI, D.; ROUTHIER, S.; WIJNEN, B. “Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework, RFC 2576, Internet Engineering Task Force, Em http://www.rfc-editor.org/rfc/rfc2576.txt. 2000.

GALVIN, J.; MCCLOGHRIE, K. “Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2)”, RFC 1445, Internet Engineering Task Force, Em http://www.rfc-editor.org/rfc/rfc1445.txt. 1993a.

GALVIN, J.; MCCLOGHRIE, K. “Security Protocols for version 2 of the Simple Network Management Protocol (SNMPv2)”, RFC 1446, Internet Engineering Task Force, Em http://www.rfc-editor.org/rfc/rfc1446.txt. 1993b.

GEORGEFF, I. “A Distributed Topology Discovery Algorithm for Wireless Sensor Networks”, School of Computer Science and Software Engineering, The University of Western Australia, Perth, Australia, 2004.

HOLANDA FILHO, R. “SAGRES: Um Sistema Baseado em Conhecimento para Apoio à Gerência de Falhas em Redes de Computadores”. Dissertação de Mestrado (Ciência da Computação) Universidade Federal do Ceará (UFC), Fortaleza, 1998.

IEEE. Institute of Electrical and Electronics Engineers, Inc., IEEE Std. 802.15.4-2003 "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)", Nova Iorque. 1º de Outubro , 2003.

ITU-T, Introduction to ASN.1, Em http://www.itu.int/ITU-T/asn1/ introduction/index.htm. Acesso em 10/07/2013.

JACQUOT, A.; CHANET, J.; KUN, M. H.; DE SOUSA, G.; MONIER, A. “A New Management Method for Wireless Sensor Networks”, 9th IEEE IFIP Annual Mediterranean Ad Hoc Network Workshop, Juan Les Pins, France. 2010.

KARL, H.; WILLIG, A. “Protocols and Architectures for Wireless Sensor Networks”, John Wiley and Sons Ltd, West Sussex, England, 2005.

KONA, Manoj K.; XU, Cheng-Zhong A Framework for Network Management using Mobile Agents - Department of Electrical and Computer Engineering Wayne State University, Detroit, MI 48202. 2002.

LEINWAND, Allan, CONROY, K. F. “Network Management A practical perspective”. 1995.

LOUREIRO, A. A. F., NOGUEIRA, J. M. S., RUIZ, L. B., MINI R. A. F., NAKAMURA, E. F., FIGUEIREDO, C. M. S. “Redes de sensores sem fio”. XXI Simpósio Brasileiro de Redes de Computadores. 2003.

MACCLOGHRIE, K.; ROSE, M. “Management Information Base for Network Management of TCP/IP-based internets: MIB-II”, RFC1213, Network Working Group, IETP, 1991.

MCCLOGHRIE, K. “An Administrative Infrastructure for SNMPv2”, RFC 1909, Internet Engineering Task Force, Em http://www.rfc-editor.org/rfc/rfc1909.txt. 1996.

MCCLOGHRIE, K.; PERKINS, D.; SCHOENWAELDER, J.; CASE, J.; MCCLOGHRIE, K.; ROSE, M.; WALDBUSSER, S. “Conformance Statements for SMIv2”, RFC 2580, Internet Engineering Task Force, Em http://www.rfc-editor.org/rfc/rfc2580.txt . 1999a.

MCCLOGHRIE, K.; PERKINS, D.; SCHOENWAELDER, J.; CASE, J.; MCCLOGHRIE, K.; ROSE, M.; WALDBUSSER, S. “Structure of Management Information Version 2 (SMIv2)”, RFC 2578, Internet Engineering Task Force, Em http://www.rfc-editor.org/rfc/ rfc2578.txt . 1999b.

MCCLOGHRIE, K.; PERKINS, D.; SCHOENWAELDER, J.; CASE, J.; MCCLOGHRIE, K.; ROSE, M.; WALDBUSSER, S. “Textual Conventions for SMIv2”, RFC 2579, Internet Engineering Task Force, Em http://www.rfc-editor.org/rfc/rfc2579.txt. 1999c.

MELCHIORS, C. Raciocínio Baseado em Casos Aplicado ao Gerenciamento. Dissertação de Falhas em Redes de Computadores (Mestrado em Ciência da Computação) Universidade Federal do Rio Grande do Sul. Porto Alegre. 1999.

NAKAMURA, F. G.; “Planejamento Dinâmico para controle de cobertura e conectividade em Redes de Sensores sem Fio planas”. Belo Horizonte: UFMG. Dissertação de mestrado apresentada ao Departamento de Ciência da Computação, 2003.

NET-SNMP Community. Em http://net-snmp.sourceforge.net/. Acesso em: 17/03/2013.

PRESUHN, R.; CASE, J.; MCCLOGHRIE, K.; ROSE, M; WALDBUSSER, S. “Transport Mappings for the Simple Network Management Protocol (SNMP)”, RFC 3417, Internet Engineering Task Force. 2002b.

PRESUHN, R.; CASE, J.; MCCLOGHRIE, K.; ROSE, M; WALDBUSSER, S. “Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)”, RFC 3416, Internet Engineering Task Force. 2002c.

RADIUINO, Em http://www.radiuino.cc, 2011.

RHT03, Em http://labdegaragem.com/profiles/blogs/tutorial-como-utilizar-o-sensor-de-temperatura-e-umidade-rht03. Acesso em: 07/07/2013.

RIGANTI, A. “Progetto e realizzazione di un sistema di monitoraggio per reti eterogenee basato su protocollo SNMP”. Dissertação de Mestrado (Engenharia de Telecomunicações) Universita di Pisa, Itália, 2005.

ROSE, M. "A convention for Defining Traps for use with the SNMP", RFC 1215, Network Working Group. Em http://www.ietf.org/rfc/rfc1215.txt. 1991.

RUIZ, L., B.; “Maná: Uma arquitetura para gerenciamento de Redes de Sensores sem Fio”. Belo Horizonte: UFMG. Dissertação de doutorado apresentada ao Departamento de Ciência da Computação, 2003.

RUIZ, L. B.; NOGUEIRA, J. M.; LOUREIRO, A. A. F. “MANNA: A Management Architecture for Wireless Sensor Networks”, IEEE Communications Magazine, February, 2003.

SAYDAM, T.; MAGEDANZ, T., “From Networks and Network Management into Service and Service Management”. Journal of Networks and System Management, volume 4. 1996.

SILVA, Fabrício A.; RUIZ, Linnyer B.; BRAGA, Thais R. M.; NOGUEIRA, José M. S.; LOUREIRO, Antonio A. F., "MannaNMP: Um Protocolo de Gerenciamento para Redes de Sensores Sem Fio.", X Workshop de Gerência e Operação de Redes e Serviços, 2005, Fortaleza. Páginas 68-79

SILVA, Fabrício A., "Avaliação de Abordagens de Gerenciamento para Redes de Sensores Sem Fio". Belo Horizonte: UFMG. Dissertação de Mestrado apresentada ao departamento de Ciência da Computação, 2006.

SLOMAN, Morris S., Network and Distributed Systems Management. 1994.

SM.2180, “Impact of industrial, scientific and medical (ISM) equipment on radio communication services”, ITU-R, SM Series, Spectrum Management, 2010.

SNMPv3WG. “SNMP Version 3 (SNMPv3)”, Em http://www.ietf.org/wg/concluded/snmpv3.html, SNMP Version 3 Working Group. 2002.

SRIVASTAVA, M. B.; Muntz, R. R.; Potkonjak, M. “Smart kindergarden: sensor-based wireless networks for smart developmental problem-solving enviroments”. In Mobile Computing and Networking, pages 132–138. 2001.

STALLINGS, William. "SNMP and SNMP v2: The infrastructure for network management," IEEE Commun. Mag., vol. 36, no. 3, pp. 37–43, Mar. 1998

STALLINGS, William, "SNMP, SNMPv2, SNMPv3, and RMON 1 and 2", 3rd edition, Addison-Wesley, 1999.

STEENKAMP, Leon de T. "Wireless sensor network monitoring using the Simple Network Management Protocol". Dissertação de mestrado. Cape Peninsula University of Technology, Cape Town. 2012

THALER, D. "FW: ipv6IfStateChange traps in RFC2465", Mailing list, Em http:// www.ops.ietf.org/lists/v6ops/v6ops.2002/msg00269.html. 2002.

WATERS, G. "User-based Security Model for SNMPv2", RFC 1910, Internet Engineering Task Force, Em http://www.rfc-editor.org/rfc/rfc1910.txt. 1996.

WIJNEN, B.; PRESUHN, R.; MCCLOGHRIE, K. "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 3415, Internet Engineering Task Force, Em http://www.rfc-editor.org/rfc/rfc3415.txt. 2002.

WIRESHARK. Em http://www.wireshark.org/. Acesso em 06/07/2013.

XIAO, Y.; KRISHNA, R; SUN, B.; DU, X.; HU, F.; GALLOWAY, M. "A survey of key management schemes in wireless sensor networks", Department of Computer Science, University of Alabama, Tuscaloosa, AL , USA, May 10, 2007.

ZOHO, 2012. WEBNMS. Em http://www.webnms.com/cagent/help/snmp/c_snmp_testagent.html. Acesso em: 20/05/2013.

Anexo I

Neste anexo está descrita mais detalhadamente a extensão da MannaMIB proposta neste trabalho, chamada WSN-MIB-UFV apresentada na seção 4.1.

A classe Energia(0), diz respeito às características de energia de um sensor, e possui os seguintes objetos identificados:

- Fonte de energia(1): responsável por armazenar qual o tipo de gerador de energia está sendo utilizado no sensor. Tipo de dados: INTEGER. Possíveis valores são: bateria(1), solar(2), eólica(3), corrente alternada(4). O controle sobre este objeto pode ser realizado via comandos GET e SET;
- Tipo da bateria(2): para o caso de se ter a bateria como fonte de energia, indica qual o tipo está sendo utilizado. Tipo de dados: INTEGER.Possíveis valores: AA(0) e lítio(1). O controle sobre o objeto pode ser realizado via comandos GET e SET;
- voltsFonteEnergia(3): Indica em volts(V), a voltagem fornecida pela fonte de alimentação de energia. Tipo de dados: OCTET STRING. Aceita somente o comando GET;
- Voltagem Residual(4): Quantidade de energia restante na fonte de energia, medida em Volts(V). Tipo de dados: OCTECT STRING. Aceita somente o comando GET;
- Voltagem Total(5): Capacidade total de energia da fonte de energia, medida em Volts(V). Tipo de dados: OCTET STRING. Aceite somente o comando GET;
- Voltagem de operação(6): Indica a voltagem mínima de operação do nó sensor em Volts. Tipo de dados: OCTET STRING. Aceita somente o comando GET;
- Marca da bateria(7): Indica a marca da bateria. Tipo de dados: OCTET STRING. O seu valor pode ser apenas consultado via comando GET.

A sub-árvore Topologia(1), traz informações referentes à disposição física dos sensores, e possui os seguintes objetos identificados:

- Coordenada X(1): Indica a latitude em graus decimais do nó sensor. Tipo de dados: OCTET STRING. O seu valor pode ser lido e definido utilizando os comandos GET e SET;

- Coordenada Y(2): Indica a longitude em graus decimais do nó sensor. Tipo de dados: OCTET STRING. O objeto pode ser controlado utilizando os comandos GET e SET;
- Coordenada Z(3): Indica a altitude em metros do nó sensor. Tipo de dados: OCTET STRING. O seu valor pode ser lido e alterado utilizando os comandos GET e SET;
- Tipo de descoberta de Localização(4): Indica o método utilizado para se descobrir a localização do sensor. Tipo de dados: INTEGER. Possíveis valores: GPS(0) e Beacon(1). É um objeto do tipo leitura e escrita, aceitando os comandos GET e SET;
- É móvel(5): Indica se o nó é móvel ou não. Tipo de dados: INTEGER. Possíveis valores: sim(1) e não(2). Aceita os comandos GET e SET;
- Velocidade(6): Indica a velocidade de movimentação do sensor em metros/segundo, caso o sensor esteja em movimento. Tipo de dados: OCTET STRING. O objeto pode ser gerenciado via comandos GET e SET;
- Direção(7): Indica a direção de movimentação do sensor. Tipo de dados: INTEGER. Possíveis valores: Leste (0), Oeste (1), Norte (2), Sul (3), Sudeste (4), Nordeste (5), Noroeste (6), Sudoeste (7). O objeto pode ser gerenciado via comandos GET e SET;
- Vizinhos(8): Contém um sequência com as IDs dos vizinhos do nó sensor. Tipo de dados: DisplayString. O seu valor pode ser lido via o comando GET.

A sub-árvore Transceptor(2) traz informações do transmissor utilizado no sensor e possui os seguintes objetos identificados:

- Estado operacional(1): Indica o estado operacional do rádio no momento. Tipo de dados: INTEGER. Possíveis valores: Ativo (0), Dormindo (1), Inativo (2). O objeto pode ser gerenciado via comandos GET e SET;
- rssi(2): Indica a potência do sinal de recepção do rádio em dBm. Tipo de dados: OCTET STRING, Para a gerência este objeto deve estar visível como somente leitura, aceitando apenas comandos GET;
- Canal de transmissão(3): Indica o canal utilizado para a transmissão. Tipo de dados: Integer32. O seu valor pode ser apenas lido via comando GET;

- Tipo(4): Indica o tipo de transceptor utilizado. Tipo de dados: INTEGER, Valores possíveis: RF (1), Optico (2), Laser (3). Este objeto pode ser gerenciado via comandos GET e SET;

A sub-árvore Processador(3) é responsável por conter as informações do hardware utilizado como base do sensor e possui os seguintes objetos gerenciados:

- estadoOperacionalP(1): Indica o estado operacional do processador no momento. Tipo de dados: INTEGER. Possíveis valores: Ativo (1), Dormindo (2), Inativo (3). Este objeto pode ter o seu valor lido e alterado via comandos GET e SET;
- consumoInstrução(2): Indica qual é o consumo por instrução do processador em Ampere-Hora. Tipo de dados Integer32. Este objeto pode ser gerenciado via comandos GET e SET;
- mips(3): Indica o MIPS(Milhões de Instruções por Segundo) do processador. Tipo de dados: integer32. Este objeto está configurado com a visibilidade de leitura e escrita, aceitando os comandos GET e SET;
- tipoP(4): Indica o tipo do processador. Tipo de dados: DisplayString. Aceita os comandos GET e SET;
- marca(5): Indica a marca do processador. Tipo de dados: DisplayString. Pode ser gerenciado utilizando os comandos GET e SET;
- Freqüência(6): Indica a freqüência de operação do processador em Hz. Tipo de dados: INTEGER. O objeto gerenciado pode ter seu valor lido e alterado via comandos GET e SET;
- Memória RAM(7): Indica a quantidade de memória RAM em Kb. Tipo de dados: Integer32. Aceita os comandos GET e SET;
- Memória ROM(8): Indica a quantidade de memória ROM em Kb. Tipo de dados: Integer32. Pode ser gerenciado utilizando os comandos GET e SET;
- Memória RAM livre(9): Indica a quantidade de memória RAM livre em Kb. Tipo de dados: INTEGER. Com relação à visibilidade este objeto é de escrita e leitura, aceitando os comandos GET e SET;
- Memória ROM livre(10): Indica a quantidade de memória ROM livre em Kb. Tipo de dados: INTEGER. Pode ser gerenciado utilizando os comandos GET e SET.

A sub-árvore Sensor(4) contém as informações referentes aos sensores e possui os seguintes objetos gerenciados:

- quantidade(1): Indica a quantidade de sensores que existe no nó. Tipo de dados: INTEGER. Este objeto é do tipo somente leitura, podendo ser utilizado apenas o comando GET para recuperar o seu valor;
- sensorTable(2): Uma tabela contendo as informações gerenciadas de cada sensor presente no nó. Esta tabela é composta pelos seguintes objetos:
    - indiceSensores(2): Contém o índice do sensor. Tipo de dados: Integer32. O seu valor pode ser apenas lido via comando GET.
    - Tipo do sensor(3): Indica qual o tipo de sensor está sendo utilizado. Tipo de dados: DisplayString. Valores possíveis: Temperatura, Umidade, Luz, Acelerômetro, entre outros. Aceita somente o comando GET;
    - Consumo(4): Indica qual é o consumo do sensor em Watts. Tipo de dados: DisplayString. Pode ser gerenciado utilizando o comando GET;
    - marcaSensor(5): Indica a marca do sensor. Tipo de dados: DisplayString. A visibilidade deste objeto é de somente leitura, aceitando o comando GET;
    - Taxa de erro(6): Indica a taxa de erros das medidas: DisplayString. Pode ser gerenciado utilizando o comando GET;
    - Última calibração(7): Indica a data em que foi realizada a última calibração do sensor. Tipo de dados: TimeTicks. Este objeto pode ser gerenciado utilizando os comandos GET e SET;
    - Unidade de medida(8): Indica a unidade de medida do sensor. Tipo de dados: DisplayString. O valor deste objeto pode ser recuperado utilizando o comando GET;
    - Dados(7): Contém o valor do fenômeno sensoriado. Tipo de dados: OCTET STRING. Pode ser gerenciado utilizando apenas o comando GET.

A sub-árvore Administração(5) contém informações administrativas sobre os sensores e possui os seguintes objetos gerenciados:

- Estado administrativo(1): Indica o estado do sensor. Tipo de dados: INTEGER. Valores possíveis: (0)Desbloqueado, (1)Bloqueado. O seu valor pode ser lido e alterado via comandos GET e SET;
- É lider(2): Indica se o sensor é líder. Tipo de dados: INTEGER. Valores possíveis: sim(1) e não(2). O objeto possui visibilidade de leitura e escrita;

- dadosEnviados(3): Indica a quantidade de mensagens de dados enviadas pelo nó. Tipo de dados: INTEGER. O seu valor pode ser obtido através do comando GET;
- dadosRecebidos(4): Indica a quantidade de mensagens de dados recebidas pelo nó. Tipo de dados: INTEGER. Aceita somente o comando GET;
- gerenciamentoEnviados(5): Indica a quantidade de mensagens de gerenciamento enviadas pelo nó. Tipo de dados: INTEGER. O seu valor pode ser obtido via comando GET;
- gerenciamentoRecebidos(6): Indica a quantidade de mensagens de gerenciamento recebidas pelo nó. Tipo de dados: INTEGER. Em termos de visibilidade é do tipo somente leitura.

A sub-árvore Hierarquia(6) é responsável por conter informações sobre como a rede de sensores está organizada e possui os seguintes objetos gerenciados:

- quantidade(1): Indica a quantidade de grupos que existe na RSSF. Tipo de dados: INTEGER. Este objeto é do tipo somente leitura, podendo ser utilizado apenas o comando GET para recuperar o seu valor;
- groupTable(2): Uma tabela contendo as informações gerenciadas de cada grupo hierárquico presente na RSSF. Esta tabela é composta pelos seguintes objetos:
  - grupoID(2): É utilizado para identificar o grupo de sensores unicamente. Tipo de dados: INTEGER. É do tipo somente leitura.
  - TipoFormaçãoFrupo(3): Indica qual o tipo de formação do grupo. Tipo de dados: INTEGER. Possíveis valores: Centralizado(0), Distribuído (1). Pode ter o seu valor lido e alterado utilizando os comandos GET e SET;
  - integrantesGrupo(4): Contem uma lista com os IDs dos integrantes do grupo. Tipo de dados: DisplayString. Pode ser gerenciado utilizando o comando GET;
  - integrantesGrupoAtivo(5): Contém a lista dos nós integrantes do grupo que estão em operação. Tipo de dados: DisplayString. O tipo de dados é somente leitura;
  - integrantesGrupoReserva(6): Contém a lista dos nós integrantes do grupo que estão fora de operação. Tipo de dados: DisplayString. Aceita somente o comando GET;

- Nível de Hierarquia(7): Indica o nível da hierarquia deste nó. Tipo de dados: INTEGER. Quanto à visibilidade é do tipo somente leitura.

Por último têm-se a sub-árvore ufvTraps(255) é responsável por conter a trap definida na MIB WSN-MIB-UFV, e possui os seguintes objetos gerenciados:

- ID(1): Contém um código para identificar do que se trata a *trap*. Este objeto é do tipo trap;
- limiteInferior(2): Contém o nível mínimo a partir do qual a *trap* será gerada. Tipo de dados: INTEGER. Quanto à visibilidade é do tipo leitura e escrita;
- média(3): Contém o valor médio de medida. Tipo de dados: INTEGER. O seu valor pode ser lido e alterado;
- limiteSuperior(3): Contém um valor máximo, que vai gerar um *trap* quando o valor monitorado ultrapassar este limite. Tipo de dados: INTEGER. Aceita os comandos SNMP GET e SET.

Anexo II

Neste anexo está detalhado o esquema do arquivo XML utilizado para a configuração da comunicação do agente *proxy* com os nós sensores, além da definição da periodicidade da consulta.

A seguir está apresentado o esqueleto do XML.

```
<config>
<sockets numreg=>
      <socket name=
                  type=
                  port=
                  baudrate =
                  idhL =  idhH =
                  panId=
            timeOutReconect =  />
</sockets>
<stations numreg=>
      <station name= socket = >
            <sensors numreg=>
               <sensor name= sintax= latencyUpdate = list =
type =  dataType = />

            </sensors>
      </station>
</config>
```

O detalhamento das *tags* utilizadas está apresentada a seguir:

- config: Indica o início e o término das configurações;
- sockets: Indica a quantidade de rádios que serão configurados, de acordo com o valor definido em numreg;
- socket name: Indica o nome que será atribuído ao rádio XBee;
- socket type: Especifica o tipo de rádio que está sendo utilizado;
- socket port: Recebe o caminho para o acesso à porta do rádio dentro do linux;
- socket baudrate: Indica a taxa de transmissão de dados em *bits* por segundo;
- socket idhH e socket idhL: Indica respectivamente o endereço alto e baixo para acesso ao dispositivo dentro da rede ZigBee;
- socket panID: Contém o valor de qual grupo hierárquico o rádio faz parte;
- timeOutReconect: Indica o valor em minutos, para reiniciar o rádio em caso de falta de comunicação;
- stations: Diz respeito a quantidade de informações que posso extrair de cada nó sensor, sendo este valor definido pelo numreg;
- sensor name: Indica o nome do dado que quero recuperar do nó sensor;
- sensor syntax: Especifica o token que será enviado para o nó sensor;
- sensor latencyUpdate: Recebe a periodicidade de consulta do dado no nó sensor, este valor é medido em segundos;
- sensor list: Pode receber *true* ou *false* para indicar se os dados coletados devem ser armazenados em lista;
- sensor type: Indica se a consulta será realizada o rádio ou para o arduíno;

- sensordataType: Contém o tipo de dados da informação coletada, podendo receber valores como int e string.