PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS

CEATEC

FACULDADE DE ENGENHARIA ELÉTRICA

FREDERICO SILVEIRA CYRIACO

GERÊNCIA DE REDES DE SENSORES SEM FIO – UMA ABORDAGEM COM *SNMP*

CAMPINAS

2011

FREDERICO SILVEIRA CYRIACO

GERÊNCIA DE REDES DE SENSORES SEM FIO – UMA ABORDAGEM COM *SNMP*

Dissertação apresentada como exigência para obtenção do Título de Mestre em Engenharia Elétrica, ao Programa de Pós-Graduação em Gestão de Redes de Telecomunicações, Pontifícia Universidade Católica de Campinas.

Orientador: Prof. Dr. Omar Carvalho Branquinho

PUC-CAMPINAS

2011

Ficha Catalográfica
Elaborada pelo Sistema de Bibliotecas e
Informação - SBI - PUC-Campinas

t621.382
C997g

Cyriaco, Frederico Silveira.
Gerência de redes de sensores sem fio: uma abordagem com SNMP / Frederico Silveira Cyriaco. – Campinas: PUC-Campinas, 2011.
119p.

Orientador: Omar Carvalho Branquinho.
Dissertação (mestrado) – Pontifícia Universidade Católica de Campinas, Centro de Ciências Exatas, Ambientais e de Tecnologias, Pós-Graduação em Engenharia Elétrica.
Inclui bibliografia.

1. Sistemas de telecomunicações. 2. Redes de sensores sem fio. 3. Comunicações digitais. 4. Sistema de comunicação sem fio. I. Branquinho, Omar Carvalho. II. Pontifícia Universidade Católica de Campinas. Centro de Ciências Exatas, Ambientais e de Tecnologias. Pós-Graduação em Engenharia Elétrica. III. Título.

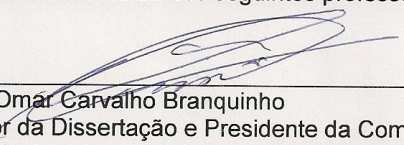
22.ed.CDD – t621.382

FREDERICO SILVEIRA CYRIACO

GERÊNCIA DE REDES DE SENSORES SEM FIO – UMA ABORDAGEM COM SNMP

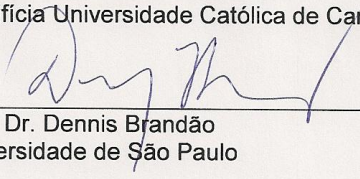
Dissertação apresentada ao Curso de Mestrado Profissional em Gestão de Redes de Telecomunicações do Centro de Ciências Exatas, Ambientais e de Tecnologias da Pontifícia Universidade Católica de Campinas como requisito parcial para obtenção do título de Mestre em Gestão de Redes de Telecomunicações.
Área de Concentração: Gestão de Redes e Serviços.
Orientador: Prof. Dr. Omar Carvalho Branquinho

Dissertação defendida e aprovada em 06 de fevereiro de 2012 pela Comissão Examinadora constituída dos seguintes professores:



Prof. Dr. Omar Carvalho Branquinho
Orientador da Dissertação e Presidente da Comissão Examinadora
Pontifícia Universidade Católica de Campinas

Prof. Dr. Alexandre de Assis Mota
Pontifícia Universidade Católica de Campinas



Prof. Dr. Dennis Brandão
Universidade de São Paulo

Dedico este trabalho aos meus pais, Luiz Claudio Andrade Cyriaco e Eliana Maria Silveira Cyriaco, pois sem este apoio não seria possível concluir deste trabalho.

AGRADECIMENTOS

Ao Prof. Dr. Omar Carvalho Branquinho,

Meu grande incentivador, orientador e mestre.

Ao Mestrando Raphael Montali da Assumpção,

Grande colaborador e companheiro de jornada.

Aos Graduandos Augusto Orlani a Karyna Cardoso,

Pelo empenho na execução de testes e coleta de dados.

Aos Técnicos Juliana Machado e Eduardo Veiga do Laboratório de Meios de Transmissão,

Pelo suporte técnico prestado.

“Obstáculo é aquilo que você enxerga,
quando tira os olhos do seu objetivo.”

Henry Ford

Resumo

CYRIACO, Frederico Silveira. *Gerência de redes sensores sem fio – uma abordagem com SNMP*. 2011. Dissertação (Mestrado em Engenharia Elétrica) – Pontifícia Universidade Católica de Campinas, Centro de Ciências Exatas Ambientais e de Tecnologias, Programa de Mestrado Profissional em Gestão de Redes de Telecomunicações, Campinas, 2011.

A presente dissertação tem como objetivo propor uma abordagem para a gerência remota de redes de sensores sem fio. Para isso, foi inserido no contexto de redes sensores o elemento do tipo *Proxy*, responsável por interconectar duas redes distintas, mantendo a capacidade de comunicação entre estas duas redes, assim como a adaptação de protocolos de comunicação. Neste caso, as redes as quais o *Proxy* se conecta são as redes de sensores sem fio e *IP*. O protocolo de gerência utilizado do lado da rede *IP* foi o *SNMP*, *Single Network Management Protocol*, por ser um protocolo de gerência de redes estável, de código aberto e amplamente utilizado e disseminado em ambientes de rede. Com o objetivo de se combinar as funcionalidades de gerência via *SNMP* com uma rede de sensores sem fio, foi realizada a engenharia da *MIB*, *Management Information Base*, criando-se uma correspondência entre características da rede de sensores sem fio e as funcionalidades de configuração, desempenho e falha, presentes em uma rede de gerência clássica. Com esta *MIB* foi possível elaborar um protótipo, a título de prova de conceito, para a demonstração destas funcionalidades.

Termos de indexação: Rede de sensores sem fio. *SNMP*. Gerência de redes. *Proxy*.

Abstract

CYRIACO, Frederico Silveira. *Wireless Sensor Network Management – an SNMP approach*. 2011. Dissertation (Master in Electrical Engineering) – Pontifícia Universidade Católica de Campinas, Centro de Ciências Exatas Ambientais e de Tecnologias, Programa de Mestrado Profissional em Gestão de Redes de Telecomunicações, Campinas, 2011.

This dissertation aims to propose an approach for remote management of wireless sensor networks. For that was inserted in the context of sensor networks the Proxy type element, responsible for interconnect two different networks, while maintaining the ability to communicate between these two networks, as well as the adaptation of communication protocols. In this case, the Proxy networks which interconnects the networks of wireless sensors and IP. The management protocol used on the side of the IP network is SNMP, Single Network Management Protocol, as a network management protocol stable, open source and widely disseminated and used in network environments. In order to combine the functionality via SNMP management with a network of wireless sensors was performed engineering MIB, Management Information Base, creating a correspondence between the characteristics of wireless sensor network configuration and functionality, performance and failure, present in a classical network management. With this MIB was possible to build a prototype as a proof of concept to demonstrate these features.

Index terms: Wireless Sensor Network. SNMP. Network management. Proxy.

LISTA DE GRÁFICOS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ACK – *Acknowledgment*

ASK – *Amplitude Shift Keying*

CDMA – *Code Division Multiple Access*

CDR – *Call Detail Record*

CSMA – *Carrier Sense Multiple Access*

dB - Decibél

FDMA – *Frequency Division Multiple Access*

FSK – *Frequency Shift Keying*

GFSK – *Gaussian Frequency Shift Keying*

IANA – *Internet Assignment Numbers Authority*

ISM – *Industrial Scientific Medical Band*

LLC – *Logical Link Control*

MACA – *Multiple Access with Collision Avoidance*

MACAW - *Multiple Access with Collision Avoidance for Wireless*

MANNA – *Management Architecture for Wireless Sensor Networks*

MIB – *Management Information Base*

MSK – *Minimum Shift Keying*

OID – *Object Identification*

OOK – *On-off Keying*

PGD – Plano de Gerência de Dados

PGR – Plano de Gerência de RSSF

PWM – *Pulse Width Modulation*

QoS – *Quality of Service*

RSSF – Rede de Sensores Sem Fio

RSSI – *Radio Signal Strength Indicator*

SNMP – *Single Network Management Protocol*

TDMA – *Time Division Multiple Access*

WSN – *Wireless Sensor Network*

WSNMP – *Wireless Single Network Management Protocol*

SUMÁRIO

1 Introdução

Em nosso mundo estamos expostos a uma enorme quantidade de informações sobre importantes grandezas físicas, disponíveis visualmente ou digitalmente, que estão acessíveis por variadas formas, entre elas a internet. Os relatórios climáticos, a monitoração de florestas tropicais, a monitoração de pragas em culturas (GOENSE, 2005), o nível do mar, são exemplos de monitorações realizadas constantemente e disponíveis para a comunidade científica na tentativa de se realizar medições e predições sobre, por exemplo, grandes mudanças no meio ambiente (COPPIN, 2004).

Além das monitorações realizadas diretamente no meio ambiente, há o interesse em se realizar a monitoração sobre o dia a dia das pessoas (MAINWARING, 2002)(MARTINEZ, 2004), nos diversos contextos em que este se insere: vida pessoal, social ou profissional. Por exemplo, através do conceito *Internet of Things*, ou “Internet das Coisas” em tradução livre (ATZORI, 2010), buscam-se formas de interação entre os aspectos humanos, como dados sobre a saúde de um indivíduo (JARA, 2010), sua localização geográfica, seu perfil de condução de veículos automotivos, e sistemas de monitoração que correlacionem os diversos dados e gerem ações (THOMPSON, 2005). As tecnologias que possibilitam a criação da Internet das Coisas envolvem a capacidade de identificação de eventos, a capacidade de coleta de dados do ambiente e a miniaturização de componentes (ITU, 2005). Estes componentes, cada vez menores, apresentam a habilidade de interconexão e interação com o ambiente.

Analisando o contexto de monitoração do meio ambiente e o contexto de Internet das Coisas, a mesma necessidade existe e será de grande impacto na arquitetura destes sistemas: a forma como as informações são passadas adiante até que chegue ao interessado em obter a informação, sendo que podem haver bruscas alterações no ambiente de monitoração e no posicionamento geográfico dos elementos monitorados. Uma forma de se manter uma infraestrutura de monitoração sobre um ambiente ou pessoa, capaz de suportar mobilidade e ter flexibilidade quanto à forma de se enviar os dados coletados é através da rede de sensores sem fio.

Rede de sensores sem fio (RSSF), ou *Wireless Sensor Network* (WSN), é uma rede de elementos dedicados a monitoração e atuação em processos, diretamente em campo ou em ambientes confinados, e que tem como características marcantes a capacidade de utilizar uma quantidade de energia reduzida a ponto de seus elementos sensores funcionarem ininterruptamente por dias ou meses sem a necessidade de manutenção de suas fontes de alimentação (KARL, 2005). As redes de sensores são, em sua concepção, redes centradas em dados, ou seja, o interesse no uso da rede de sensores sem fio está concentrado na monitoração do ambiente. Protocolos de comunicação entre os sensores se encarregam de cuidar da rede sem fio, tirando o foco da gerência da rede e se concentrando na gerência dos dados.

Em um cenário de operação da rede de sensores sem fio no qual existem fatores físicos colaborando para o mau funcionamento da rede, estes protocolos podem ser eficientes no contorno destes problemas, mas não deixam claro aos usuários o que verdadeiramente acontece. Nesse contexto surge a necessidade de gerenciar estes sensores, tanto do ponto de vista da rede de interconexão dos sensores, quanto das grandezas medidas pelos sensores. A rede de sensores sem fio deixa de ser então tratada apenas como centrada em dados. Além de sensores, também podem existir nós na rede de sensores funcionando como atuadores para controle de processos (AKYILDIZ, 2010).

A gerência de elementos de rede, assim como em RSSF, deve ser executada através de comandos enviados aos agentes de gerência, presentes nos diversos nós e deve estar preparada para receber respostas dos agentes. Assim sendo, um protocolo de gerência deve ser compartilhado através de toda a rede, sendo implementado sobre cada um dos elementos de rede (STALLINGS, 1999).

A criação de protocolos de gerência foi fundamental para a uniformidade de mensagens esperada de um sistema de gerência de informações de diversos nós de rede. Em resposta à necessidade de se padronizar as mensagens em uma rede de gerência, vários fabricantes se voltaram para um padrão em especial, o *SNMP* (*Simple Network Management Protocol*).

Inicialmente especificado no fim dos anos 1980, era considerado muito limitado para as tarefas mais críticas, e então foram criadas mais duas variações. A segunda versão, um aprimoramento do padrão *SNMP*, conhecido como *SNMPv2*, provê mais funcionalidades e eficiência que o padrão original. A terceira versão, *SNMPv3*, provê mais segurança que as versões anteriores, incluindo senha e encriptação de dados (STALLINGS, 1999) (UDUPA, 1999). A abordagem da gerência por *SNMP* se baseia na centralização de informações em uma estação de gerência e reconhecimento dos dados através da correspondência com uma base de dados de gerência chamada *MIB*, *Management Information Base*.

Para este trabalho será adotada a abordagem do *SNMPv1* aplicada à rede de sensores sem fio, já que a transmissão de dados de gerência sujeito a políticas de segurança não é do nosso interesse. A implementação de segurança nesta proposta exigiria o uso da versão SNMPv3, adequada às políticas de segurança porém de mais difícil implementação.

1.1 Motivação

A tendência é de que todas as informações estejam disponíveis na Internet. Esta tendência também deve acontecer com as informações coletadas nas RSSF. Portanto, a integração das RSSF com a Internet é um ponto chave para que, de fato, as informações monitoradas e os controles possam ser realizados remotamente. A integração da RSSF com a Internet é portanto um ponto chave a ser avaliado na busca de uma solução adequada. A motivação deste trabalho é definir uma forma de RSSF, não apenas centrada em dados, utilizando um protocolo de gerência de redes para comunicação entre máquinas, suprindo a necessidade de gerência de rede de sensores sem fio e dados gerados pelo sensor. A necessidade de se gerenciar sensores sem fio, devido às características do ambiente, amplia o escopo da gerência centrada em dados e, por tanto, a concepção da gerência de RSSF deve se adequar a este fim.

1.2 Objetivo

Utilizando o arcabouço teórico de gerência de redes de computadores e ferramentas de implementação de *hardware* e *software*, será proposto um método de gerência de RSSF através do protocolo *SNMP*, implementando-se uma adaptação entre a rede de sensores sem fio e a rede de gerência baseada em *IP*, utilizando o protocolo de gerência *SNMP*. Seguindo a metodologia utilizada durante a pesquisa e o desenvolvimento desta abordagem, ao final desta proposta espera-se obter uma metodologia para a engenharia da *MIB* do agente *SNMP*, onde residem todas as informações gerenciadas. Com a implementação desta estratégia as RSSF passaram a ser um elemento de rede de computadores, como são os computadores, roteadores, estações rádio base e vários dispositivos, que são gerenciados pela Internet.

1.3 Organização do trabalho

O trabalho está organizado como se segue. O Capítulo 2 apresenta o conceito de gerência de rede de dados. Além da fundamentação teórica inicial sobre gerência de redes, é abordado também o protocolo *SNMP*.

No Capítulo 3 o conceito de RSSF é apresentado identificando os principais elementos da rede. São identificados neste ponto os paradigmas necessários para a aplicação da gerência de redes de dados na gerência de RSSF. Serão comentados também aspectos relativos à propagação de rádio no sentido de situar aspectos relevantes à gerência de RSSF.

No Capítulo 5 é feito um estudo da aplicação dos paradigmas de gerência de redes para a gerência de RSSF. O trabalho propõe a identificação de dois planos de gerência: rede e dados. A partir desta definição em planos, cria-se uma extensão do arcabouço de gerência de redes de modo a facilitar a identificação dos papéis na gerência de redes, ou seja, a diferenciação entre a gerência de RSSF e a gerência de dados.

No Capítulo 6 é descrita a engenharia da MIB, identificando os aspectos importantes na gerência de uma rede de sensores sem fio. É definida a

organização dos dados da MIB em seus diversos planos, de rede e de dados, criando a diferenciação formal entre os dois planos.

O Capítulo 7 apresenta a metodologia empregada para a avaliação do arcabouço proposto no Capítulo 6. Uma RSSF foi construída para monitoração da intensidade de sinal de radio frequência e temperatura em diversos pontos, sendo que estas informações devem ser encaminhas para uma estação de gerência. A RSSF será monitorada medindo a *RSSI*, ou *Radio Signal Strenght Indicator*. A construção da arquitetura de gerência via agente *Proxy*, considerando os dois planos, utiliza uma plataforma que permite a construção da *MIB* customizada para atender as necessidades da proposta. A metodologia de teste para avaliação da proposta é apresentada.

No Capítulo 8 são obtidos resultados da aplicação dos conceitos do Capítulo 5 e Capítulo 6, através de experimentos criados com o auxílio de uma plataforma de implementação de sensores e de *softwares* de gerência de redes.

No Capítulo 9 é feita uma análise dos resultados verificando a consistência da proposta inicial, através da comparação com outras propostas presentes na literatura.

No Capitulo 10 é apresentada a conclusão a respeito da validade e aplicabilidade do modelo de gerência de RSSF proposto e relacionados alguns trabalhos futuros.

2 Gerência de Redes

A seguir serão revistos os conceitos já utilizados em diversos campos correlatos a este trabalho, com o objetivo de criar um substrato teórico necessário ao entendimento de nossos objetivos e indicar a forma como estes conceitos se inserem no contexto da gerência de redes de sensores sem fio.

2.1 Conceito de Gerência de Rede de Computadores

Do ponto de vista de usuários e administradores de infraestrutura de redes de computadores, a disponibilidade dos serviços presentes nesta estrutura e a possibilidade de monitorá-la a fim de mantê-la funcionando são necessidades constantes. O retorno financeiro a partir do investimento inicial sobre esta rede é alcançado na medida em que as falhas se mantêm em patamares controlados e seus gastos se mantém inferiores ao retorno obtido a partir da venda de serviços prestados, gerando menos reclamações de usuários.

Com o objetivo de realizar este controle de forma organizada e estruturada, a gestão sobre os diversos aspectos de uma rede deve ser aplicada e exercitada pelos vários usuários e administradores (STALLINGS, 1999).

Na Figura 1 tem-se um exemplo de rede de gerência de dados clássica, onde o elemento de rede responsável pela gerência da rede, 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. A estação de gerência de rede se comunica com os diversos elementos através de protocolos de comunicação máquina-máquina, ou seja, entre os agentes presentes nos diversos nós, e a estação de gerência. Isso garante que a administração da estação de gerência seja centralizada e independente da operação dos diversos elementos da rede.

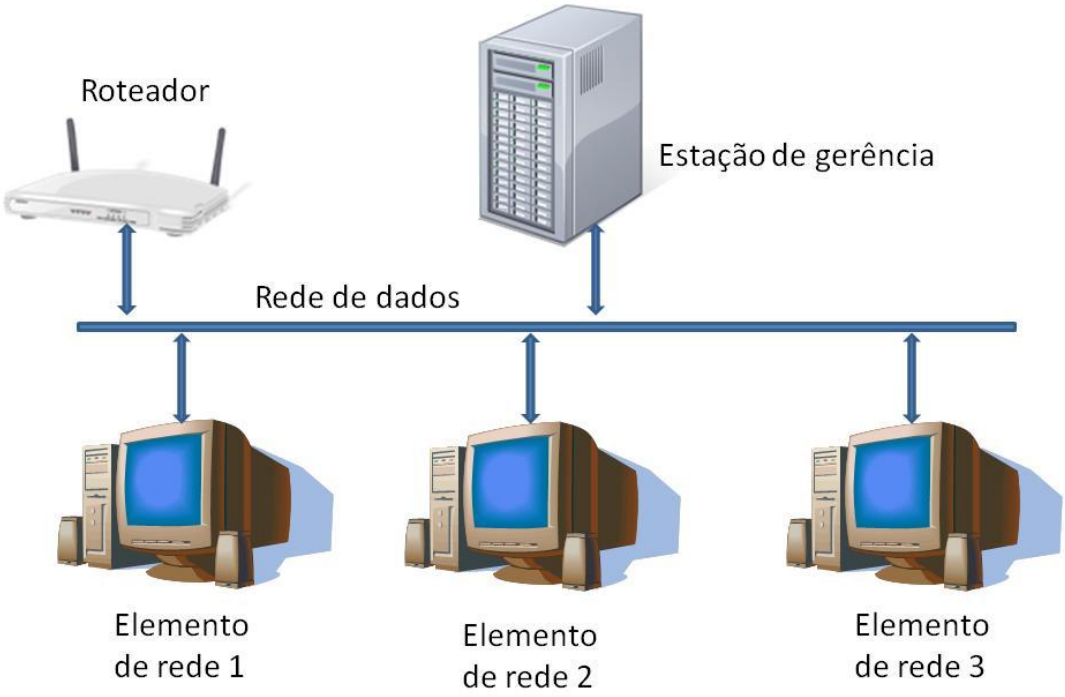


Figura 1 - Típica infraestrutura de rede com estação de gerência.

Conceitualmente, uma gerência de redes deve atender a cinco pilares em que se apoiam quaisquer tecnologias de gerência. Tipicamente, uma gerência de redes deve implementar a gerência de configuração, gerência de desempenho, gerência de falha, gerência de contabilidade, ou *accounting,* e gerência de segurança (STALLINGS, 1999)(UDUPA, 1999). Um sistema de gerência de redes não precisa, obrigatoriamente, implementar todos estes pilares simultaneamente, pois a priori são aspectos independentes uns dos outros. Na verdade, cada um destes aspectos é de interesse de diferentes usuários, ou seja, os usuários interessados em manter a rede em bom estado não são necessariamente os mesmos usuários interessados em segurança de dados.

De forma a esclarecer estes pontos, com relação ao atendimento aos pilares da gerência de redes, a seguir discutimos separadamente cada um destes aspectos, situando-os em casos específicos e introduzindo para cada um dos pilares a correspondência na gerencia de rede de sensores sem fio.

2.1.1 Gerência de configuração

No inicio, ou durante a operação de uma rede, são realizadas inicializações de recursos, físicos ou lógicos, que viabilizam a comunicação entre os nós de uma rede. Além da inicialização, com o tempo são necessários diversos ajustes, seja por simples manutenção ou por alteração das configurações de elementos de rede. A gerência de configuração pode ainda ser responsável pelo desligamento de recursos individuais da rede, visando o isolamento de partes defeituosas da rede em operação.

A configuração inicial da rede pode ter como entrada o planejamento inicial da rede, levando em conta a capacidade esperada e a quantidade de serviços esperada, ou um conjunto de informações historicamente reunidas sobre o comportamento de redes similares, podendo ser aplicadas em uma nova rede. Desta forma um administrador tem referências históricas sobre tipos de configuração a utilizar, melhorando muito a qualidade das redes através do acumulo de experiência.

Porém, os dados iniciais podem permanentemente ser monitorados e alterados, segundo informações recebidas de outras gerências (STALLINGS, 1999). Por exemplo, a gerência de configuração pode realizar uma alteração automaticamente em resposta aos dados de desempenho recebidos da gerência de desempenho. A gerência de configuração pode ainda definir limites de validade, onde valores máximos e mínimos podem ser utilizados pela gerência de falhas para criar eventos.

Em redes de sensores sem fio a gerência de configuração se aplica no *setup* inicial da rede e de sensores, configurando, por exemplo, parâmetros de rádio e limites de medição, ou reconfigurando a rede em caso de perda ou quebra de sensores.

2.1.2 Gerência de desempenho

A gerência de desempenho define um conjunto de indicadores que descrevem a capacidade de um elemento de rede, do ponto de vista de

disponibilidade, tempo de resposta, *throughput*, e utilização (STALLINGS, 1999). O desempenho é influenciado por fatores externos ao elemento da rede, ou do próprio equipamento.

Tipicamente, as condições da rede influenciam diretamente na experiência do usuário, negativamente ou positivamente, assim como os equipamentos em seu conjunto *hardware* e *software* influenciam positivamente ou negativamente, segundo as condições internas dos equipamentos e dos serviços de rede que este equipamento requisita da rede, utilizando suas pilhas de protocolo de rede para se comunicarem.

A gerência de desempenho atua na monitoração e controle da rede com o objetivo de atuar sobre, por exemplo, a banda alocada para *links* de dados, a largura de banda configurada nas portas físicas e indicar a necessidade de links agregados em *load balance* (PHAM, 2004). A gerência de desempenho deve possuir como parâmetros a capacidade máxima da rede a fim de determinar se esse limite está próximo de ser alcançado. Com esse dado a gerência de desempenho pode correlacionar medições de tráfego nos diversos elementos da rede e identificar ofensores à capacidade máxima da rede. Os ofensores criam empecilhos ao desempenho geral da rede uma vez que, por exigência das aplicações executadas por eles, exigem muito *throughput* da rede.

Na Figura 2 tem-se um típico cenário de monitoração de links entre os elementos de rede. A monitoração deve se estender até os roteadores e *switches* da rede, uma vez que estes podem representar um gargalo na rede de dados.

Em termos de redes de sensores sem fio, a gerência de desempenho deve realizar medições de, por exemplo, perdas de pacotes e capacidade dos links de comunicação entre os sensores, de modo a definir as condições de operação de uma dada rede de sensores sem fio.

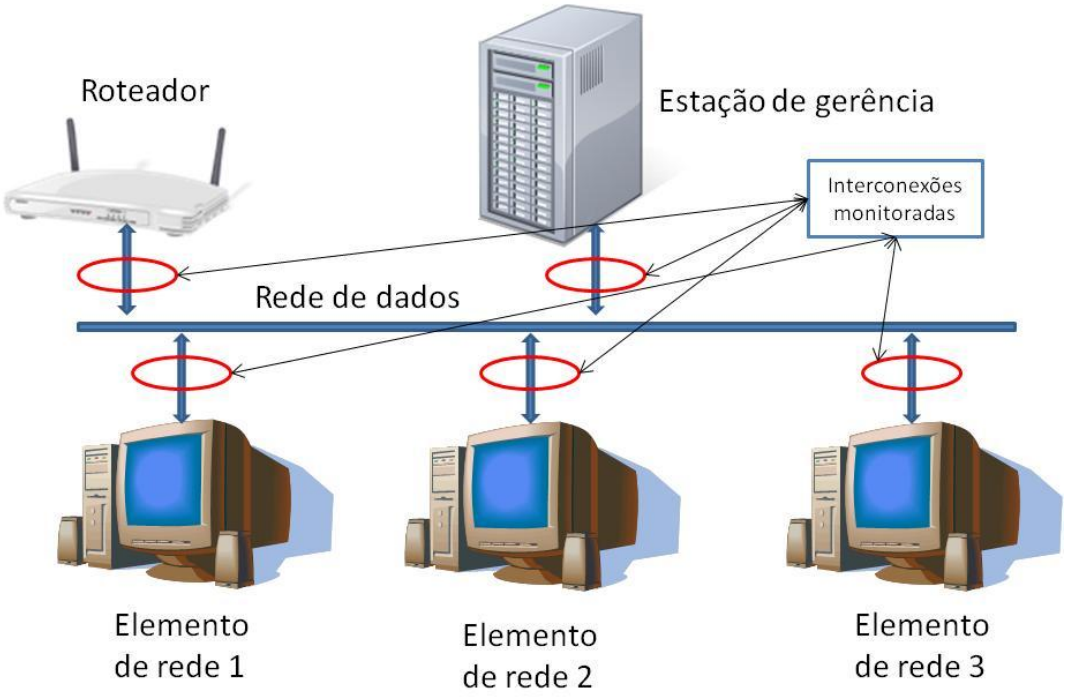


Figura 2 - Rede monitorada para controle de problemas de desempenho.

A gerência de desempenho é, por tanto, uma fonte de consulta importante para tomadas de decisão sobre investimentos financeiros em infraestrutura. A localização precisa de gargalos pode ser o divisor de águas entre a subutilização de recursos da rede ou o uso balanceado permitindo um retorno positivo dos usuários.

2.1.3 Gerência de falhas

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.

A gerência de falhas, no entanto, pode também fazer uso de informações de alarmes para indicar um problema futuro. Por exemplo, uma estação de gerência pode contabilizar alarmes indicativos de perdas de pacotes, comum em redes complexas, e definir como uma falha quando as perdas de pacotes alcançarem um determinado patamar, configurado via gerência de configuração.

Similarmente, a estação de gerência poderia indicar uma falha na rede, indicativa de interrupção de link de dados, que teria um comportamento similar, porém neste caso sendo uma falha detectada diretamente do sistema gerenciado. No primeiro caso a gerência de falhas realizou uma correlação de informações e tomou uma decisão baseada em critérios de qualidade. No segundo caso a gerência de falhas recebeu a informação de falha no sistema diretamente. Na Figura 3 são identificados cada um dos casos.

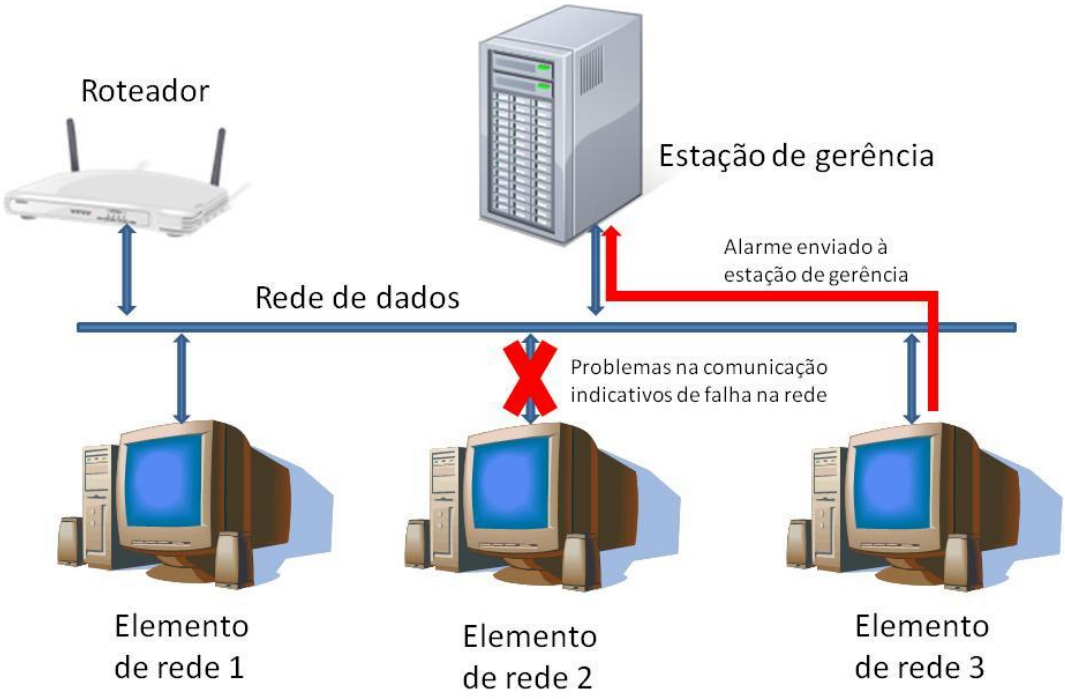


Figura 3 - Alarmes e falhas.

O papel da gerência de falhas, portanto, está também ligada diretamente a disponibilidade dos serviços prestados pela rede de dados. Através da gerência de falhas a causa de um comportamento anômalo deve ser localizada lógica e fisicamente 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. Todo o processo de registro de reclamações de usuários finais, detecção do problema, a correção da falha e registro de correções é conduzida pela gerência de falhas e centralizada na estação de gerência operada pelos administradores da rede.

Em uma rede de sensores sem fio 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.

2.1.4 Gerência de segurança

Voltada para outros aspectos de uma rede, a gerência de segurança esta ligada ao controle de acessos de usuários aos recursos desta rede e acesso a informações sigilosas, seguindo protocolos rígidos (BELOSTOTSKY, 1997)(STAMATELOPOULOS, 1997). O controle sobre senhas de rede e seu tempo de validade, privilégios de usuário sobre recursos da rede e distribuição de certificados digitais são exemplos de controles realizados pela gerência de segurança com o objetivo de segregar os recursos e separar usuários e administradores por níveis de responsabilidade, como ilustrado na Figura 4.

Além de implementar toda uma política de tratamento de informações sigilosas, a gerência de segurança se responsabiliza também pelo monitoramento e garantia de aplicação desta políticas. O meio mais comumente utilizado é a geração de registros de atividade textuais, ou *Logs*, e a permanente auditoria destes registros em busca de evidências de atividades que estejam em desacordo com os níveis de sigilo desejados.

Em resumo, a gerência de segurança pode ser definida como um conjunto de práticas de utilização de recursos de rede, voltadas para a manutenção da confiabilidade da infraestrutura da rede e do conteúdo que trafega pelos elementos desta rede, ou que fica simplesmente armazenado nos elementos de rede.

A gerência de segurança em rede de sensores sem fio pode ser utilizada na restrição ao acesso a determinados sensores, permitindo ou não permitindo a leitura das medições realizadas pelos sensores.

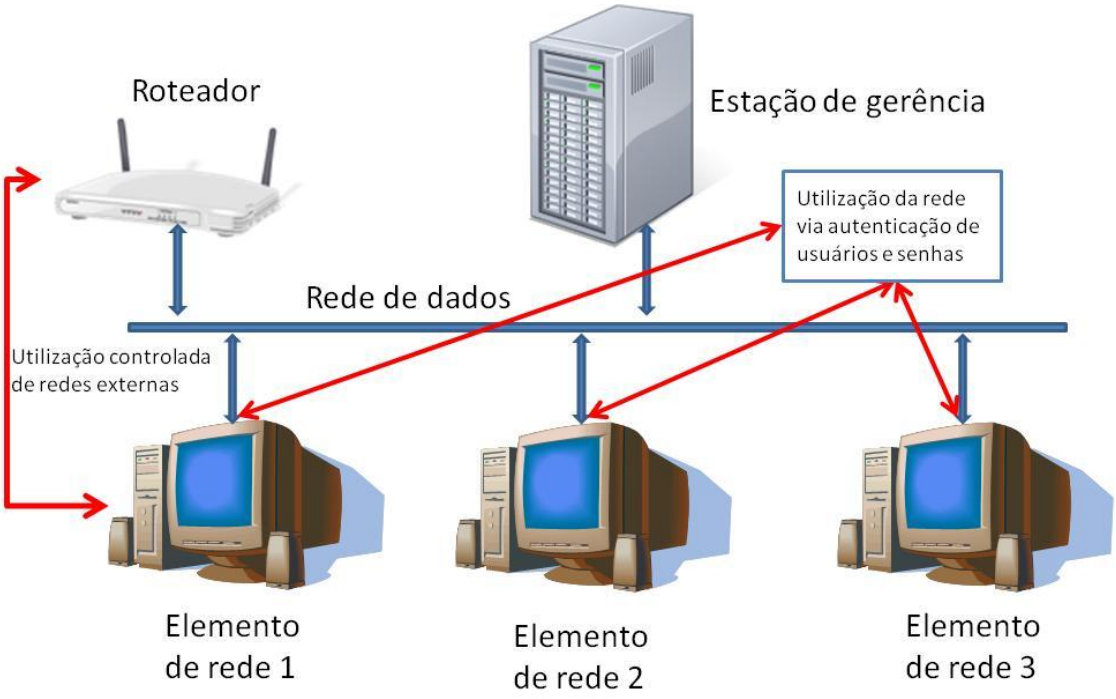


Figura 4 - Acesso controlado à rede.

2.1.5 Gerência de Contabilidade (*accounting*)

A gerência que cuida de cobrar pelos serviços prestados por uma rede fica a cargo da gerência de contabilidade, ou *accounting*. Na Figura 5 tem-se um exemplo de como os elementos de rede estão dispostos e como se interagem para que a funcionalidade contabilidade seja utilizada, neste caso na contabilização de registros detalhados de chamadas telefônicas, ou *Call Detail Record* (*CDR*).

Independente do serviço que esteja sendo disponibilizado, a estação de gerência fica a cargo da geração de cobranças e monitoração de atividades, podendo ser utilizada como base de dados de conhecimento da operadora para o oferecimento de novos serviços aos clientes.

Do ponto de vista dos administradores da rede, a gerência de contabilidade é também uma importante fonte de informação sobre o retorno do investimento realizado sobre a rede e sobre a necessidade de futuros investimentos, principalmente em capacidade e tecnologias. Através da gerência

de contabilidade os administradores monitoram o uso da rede, detectando o uso indevido de recursos ou mesmo a subutilização dos mesmos.

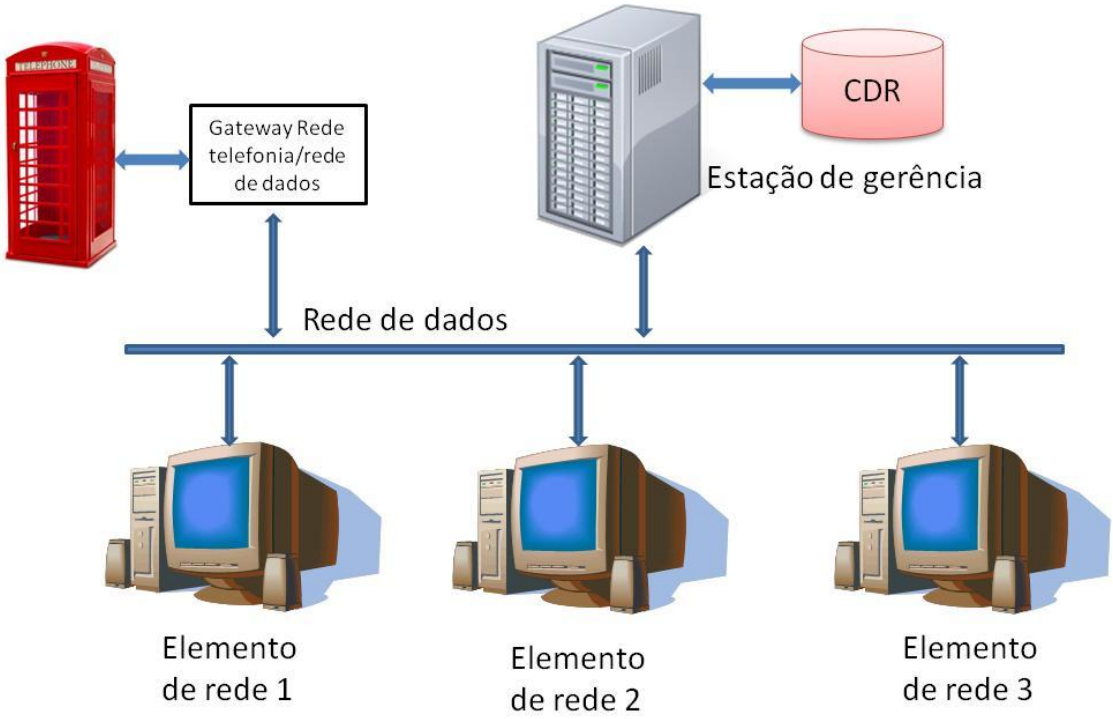


Figura 5 - Exemplo genérico do cenário de coleta de dados de contabilidade para uma rede de telefonia (PEIXOTO, 2005).

2.2 SNMP

O protocolo *SNMP,* ou *Single Network Management Protocol*, é um conjunto de especificações de gerência de rede, caracterizado por ser um protocolo de comunicação entre máquinas. A implementação deste tipo de protocolo se baseia no princípio de interação entre estação de gerência e agentes de gerência, através de um protocolo de comunicação.

A estação de gerência, que concentra os dados dos agentes, é um elemento independente da rede, ou seja, sua presença ou ausência não reflete diretamente no funcionamento da rede. Em um nível mais alto, deve possuir aplicações que implementem a interface homem-máquina e possuam funcionalidades aderentes aos pilares de gerência de redes: configuração, desempenho, falhas, segurança e contabilidade. Já os agentes são os correspondentes da estação de gerência, presentes nos elementos de rede, e que

reúnem informações em estruturas de dados. Estas estruturas de dados possuem uma correspondência com uma base de dados, presente na estação de gerência, denominada *MIB*. Portanto, a MIB é um elemento chave para a gerência das redes, uma vez que estarão nela as informações pertinentes para verificar o bom funcionamento das redes.

De modo a obter o estado atual dos agentes, ou modificar certos parâmetros, a estação de gerência lê ou modifica os dados dos agentes através de funcionalidades implementadas pelo protocolo de gerência *SNMP*. As funcionalidades básicas do protocolo *SNMP* são o *SET,* que modifica parâmetros ou dados nos agentes, o *GET*, que lê parâmetros ou dados dos agentes, e o *TRAP*, que são enviados a estação de gerência sem que uma requisição tenha sido disparada por esta estação. A estação de gerência, baseada nas informações presentes na *MIB*, utiliza estas funcionalidades para então realizar a gerência da rede.

A leitura de dados nos agentes, ou atuação sobre os mesmos através de configurações, é realizada a partir da estação de gerência executando-se comandos via protocolo SNMP. Estes comandos de gerência são enviados aos agentes juntamente com a identificação do objeto que se pretende controlar. Esta identificação é chamada de *Object Identification*, ou *OID*. A *OID* é uma maneira de se organizar de forma hierárquica todos os objetos presentes no agente *SNMP* e que são visíveis aos comandos de *GET* e *SET*. Desta forma, a estação de gerência envia ao agente *SNMP* um *GET + OID* para obter objetos, e *SET + OID + Novo Valor* para modificar objetos, ou parâmetros, no agente SNMP.

Além do envio de comandos a *OID* tem papel fundamental no significado dos *TRAPS* enviados pelos agentes. Cada um dos *TRAPS* deve ser diferenciado por diferentes *OID's*, assim como os objetos sujeitos aos comandos *GET* e *SET*. Desta forma a estação de gerência é capaz de identificar cada um dos eventos em sua camada de aplicação.

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*, *Address Translation Group*, *IP Group*, *ICMP Group*, *TCP Group*, *UDP*

Group, *EGP Group*, *Transmission Group* e *SNMP Group*). Cada um destes 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* pode não ser implementado, por características das máquinas ou por se tratarem de redes proprietárias que possuem outros tipos de protocolos e não o TCP/IP. Neste caso, um recurso pode ser utilizado para adaptar uma rede apropriada para o *SNMP* e estas máquinas, chamado *Proxy Agent*. Este recurso será amplamente explorado neste trabalho, com o objetivo de implementar as funcionalidades básicas do *SNMP* a RSSF.

3 RSSF

A Rede de Sensores sem Fio (RSSF) tem como objetivo genérico sustentar uma infraestrutura de sensores sem fio destinados a mensurar grandezas físicas e atuadores destinados a controlar grandezas físicas em grandes espaços e transmitir esses dados a um destino (KARL, 2005). A RSSF é uma rede centrada em dados, e não em endereços. Isto é, o objetivo maior de uma RSSF não é obter medidas em locais exatos, e sim o comportamento do ambiente como um todo. Um exemplo típico é a medição do conforto térmico em ambientes, utilizando diversos sensores. A média das temperaturas em diversos pontos é levada em consideração, sendo um parâmetro de utilização sobre eventuais atuações sobre o ambiente.

Em RSSF, aspectos como consumo de energia e possibilidades de formar redes ponto a ponto, ponto multiponto, com ou sem multi salto, podem ser levados em consideração e demonstram o quão heterogênea e distribuída a RSSF pode ser, como nas Figuras 6 e 7 (KARL, 2005).

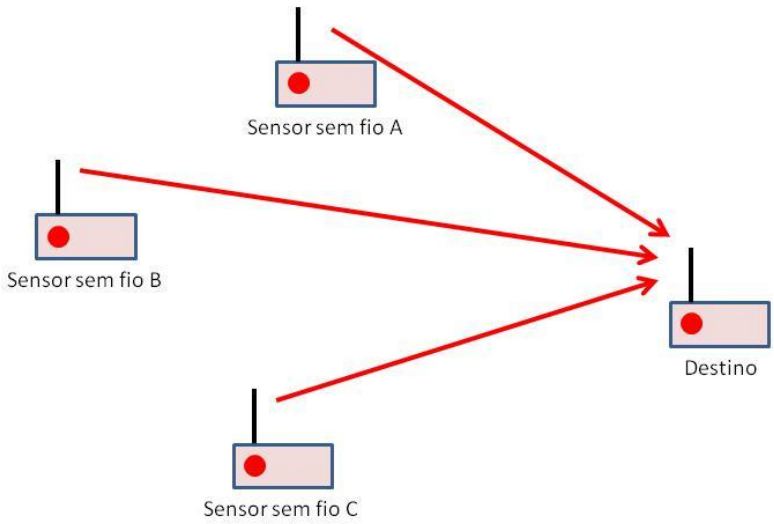


Figura 6 - Sensores enviando diretamente ao destino.

A rede multi salto, exemplificada na Figura 7, também se beneficia da variedade de caminhos possíveis, criados através de roteamento. Novos

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

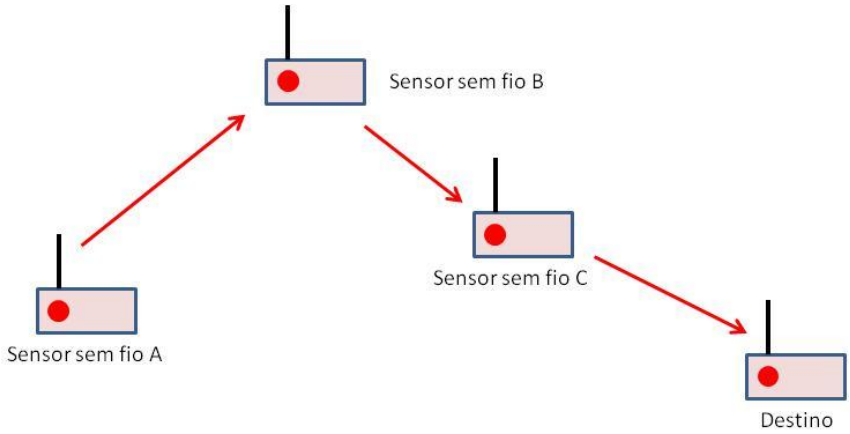


Figura 7 - Rede RSSF implementada com multi salto.

Independente da forma com que se pretende implementar uma RSSF, outro ponto importante é garantir que independentemente do crescimento do número de sensores o desempenho da rede, em suas funções básicas, deve permanecer inalterado. Esta escalabilidade, no caso de redes RSSF, reflete diretamente sobre a área de cobertura. Em redes com múltiplos saltos existe a oportunidade clara de cobrir uma área maior e manter a rede funcionando, evitando interrupções de transmissão em caso de falhas em sensores. Porém os sensores com a inteligência necessária para a criação de diversas rotas acabam por serem 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 ofensores neste caso.

Para o projeto de uma rede de sensores o mais conveniente é considerar uma pilha de protocolo que atenda aos objetivos deste 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 em função, principalmente, do consumo e espaço em memória, tornando o sensor um elemento complexo. Considerando, entretanto, a essência da pilha é possível identificar as funções necessárias a serem

executadas. Neste sentido, a plataforma Radiuino (RADIUINO, 2011) propõe uma pilha como mostrado na Figura 8.

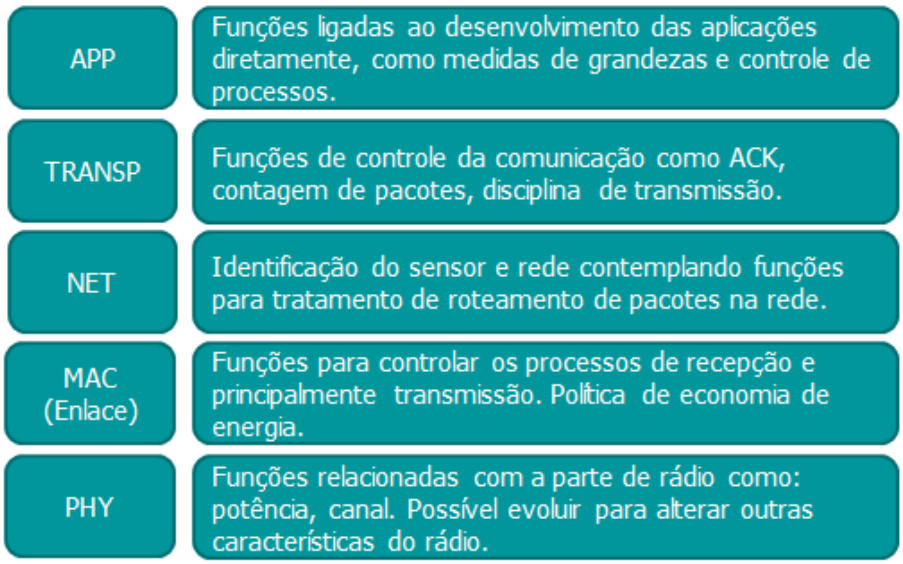


Figura 8 - Pilha de protocolo do Radiuino

Pela figura é possível identificar as funcionalidades básicas para cada uma das camadas. As referências mais recentes (AKYILDIZ, 2010)(WALTENEGUS, 2010) apresentam explicitamente a estrutura equivalente à pilha *TCP/IP*. A estratégia de gerência de RSSF deve considerar parâmetros e atributos de cada camada da pilha de protocolo. Os próximos itens tratam das características principais de cada camada.

3.1 Camada de Aplicação

O papel desta camada é abstrair a topologia física da rede para a aplicação a ser atendida pelo sensor, fazendo interface com os processos a serem monitorados e/ou controlados. Exemplos destes processos são: medida de grandezas, acionamento de dispositivos, etc. Para a medida de grandezas são encontrados transdutores analógicos e digitais. No caso dos transdutores analógicos é necessária a utilização de conversor analógico para digital, sendo necessário o projeto de circuitos adaptadores. Os dados amostrados são tratados e codificados para sua transmissão. Atualmente existem transdutores que já

fornecem as grandezas medidas na forma digital, sendo bem mais conveniente seu uso em função de não ser necessário o desenvolvimento de circuitos adaptadores. Os dados gerados pelas medidas podem necessitar de compressão em função do tamanho dos dados coletados (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 medidas também pode ser checada pela camada de aplicação. Por exemplo, a medida de uma temperatura que na última medida foi de 25 graus e na medida atual é de 200 graus pode mostrar uma inconsistência na medida, dependendo do tipo de ambiente e do intervalo de medida. Este tipo de verificação é função da camada de aplicação, em que pode ser evitada a transmissão de dados inconsistentes e criando algoritmos para checar a validade dos dados medidos.

Utilizando-se atuadores em algum processo se torna necessário, em alguns casos, checar se de fato foi efetiva a atuação. Por exemplo, se um atuador aciona um ventilador para circulação de ar em um ambiente tóxico seria importante a verificação se de fato o sistema foi acionado da forma correta. Portanto, um sensor que meça a velocidade do ar pode ser importante para garantir que de fato o processo foi acionado adequadamente. Logicamente este tipo de checagem é pertinente em processos críticos, mas deve ser considerado em cada acionamento.

Para RSSF que possui múltiplos saltos, no qual vários sensores enviam informações para um destino (*many-to-one*), existe a possibilidade dos nós perto do destino terem que trafegar grande número de pacotes. Este efeito pode ser desastroso, 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. Um exemplo simples seria a medida de temperatura em uma área. Os sensores estariam programados para enviar os dados somente quando a temperatura ultrapassar um valor ou um delta de variação. Também pode ser feita uma média da temperatura e somente se enviada esta média. Podem ser utilizadas também formas de agregação dos dados para que os nós intermediários somente transmitam um condensado da informação (KARL, 2005).

3.2 Camada de Transporte

A confiabilidade da transmissão do pacote deve ser provida pela camada de transporte, que utiliza estratégias para garantir que os dados transmitidos entre nós estejam íntegros. Uma função simples atribuída à camada de transporte é a contagem de pacotes transmitidos. Com esta estratégia os nós podem controlar a perda de pacote nos processos de transmissão. Outra função desta camada é confirmar o recebimento, através do *ACK*, de um pacote pelo nó receptor para o nó transmissor. Esta é uma função do *TCP* na pilha *TCP/IP*, que está na camada de transporte. Outra função desta camada seria o controle de congestionamento através de processos de controle de transmissão (AKYILDIZ, 2010). Funções de multiplexação e demultiplexação podem ser previstos na camada de transporte para atender diferentes aplicações.

Certamente o emprego de alguma técnica para controle da comunicação deve ser adotada considerando as limitações do sensor. Duas limitações são a pouca energia disponível e a memória necessária para guardar informações que podem ser retransmitidas. A utilização de alguma técnica de controle deve ser condicionada ao tipo de informação a ser transportada. Por exemplo, existem protocolos de controle de transmissão adequados para comunicação em tempo real e confiável (AKYILDIZ, 2010). Estes protocolos devem cuidar das questões de congestionamento da rede e o atendimento de atrasos compatíveis com a aplicação em questão.

3.3 Camada de Rede

Esta camada é responsável pela identificação do sensor na rede e pelos algoritmos de roteamento. Este é um dos tópicos mais investigado em rede de sensores na procura de algoritmos apropriados para o roteamento que atendam as peculiaridades dos tipos de redes. Entre os desafios dos protocolos de roteamento podem ser mencionados os seguintes:

- Consumo de energia

- Escalabilidade
- Endereçamento
- Robustez
- Topologia
- Atendimento ao tipo de aplicação

Os protocolos de roteamento podem ser classificados da seguinte forma (AKYILDIZ, 2010):

- *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 diferentes 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 do serviço oferecido. Podem existir informações que exigem mais qualidade na conectividade, possuindo mais prioridade.

3.4 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 *IP*, 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 pela disciplina da comunicação entre os dispositivos. Em sistemas mais complexos, como sistemas celulares, existe uma banda reservada para a transmissão da base e outra banda, com igual largura, para a transmissão do móvel. A RSSF utiliza a banda sem licenciamento *ISM* (*Industrial Scientificand Medical*), na qual existe somente uma banda de frequência utilizada para 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 colisão das mensagens transmitidas simultaneamente. O mecanismo utilizado pelos nós sensores que permite o compartilhamento do canal é chamado de controle de acesso ao meio (*MAC*), sendo seu projeto determinante para o sucesso da comunicação entre sensores ou entre o sensor e a base.

Os protocolos *MAC* podem ser classificados da seguinte forma (WALTENEGUS, 2010):

I) Protocolos livres de disputa
- Designação fixa
  - *FDMA*
  - *TDMA*
  - *CDMA*
- Designação dinâmica
  - *Polling*
  - *Token passing*
  - *Reservation-Based*

II) Protocolos baseados em disputa
- *ALOHA*
- *CSMA*
- *MACA*
- *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 os nós sensores reservem futuros acessos ao meio baseado na demanda.

Os protocolos baseados em disputa são os tradicionalmente utilizados em RSSF. O mais antigo é o *ALOHA* em que cada nó transmite e aguarda uma confirmação do sucesso da transmissão. Com a evolução das técnicas rádio 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 Access 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 transmitir e uma mensagem *Clear-to-Send* (*CTS*) que autoriza a transmissão. Finalmente 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 (*Acknowlegement – ACK*), indicando para outras estações que o meio está livre.

3.5 Camada Física

Para que seja possível o processo de comunicação do 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 destes processos/parâmetros estão relacionados a seguir (KARL, 2005):

- Modulação
- Potência
- Canal de frequência
- Ganho de antena
- Taxa de transmissão

Estas informações são importantes de serem conhecidas em uma comunicação rádio, uma vez que podem determinar a distância em função do canal de comunicação, que será tratado no próximo item. A gerência da RSSF

deve considerar estas informações para permitir os ajustes adequados para o bom funcionamento da rede.

3.6 Propagação de sinais

Com o objetivo de caracterizar o meio de transmissão utilizado neste estudo, neste caso a transmissão sem fio, a seguir é feito um esclarecimento a respeito da propagação de sinais de rádio, com ênfase nas perdas por atenuação no espaço livre e em modelos de atenuação em função da distância em relação ao transmissor e obstáculos situados no ambiente.

3.6.1 Modelo do Espaço Livre

O modelo mais simples e que serve de base para muitos modelos é o do espaço livre. Neste modelo não são considerados obstáculos entre as antenas e nenhum efeito do ambiente. Na Figura 9 tem-se um esquema bastante simples sobre um cenário de propagação de sinais de rádio, entre duas estações. Os elementos necessários para descrever o comportamento de sinais de rádio estão relacionados com as características dos equipamentos transmissor e receptor e com as características do meio ambiente que serve como canal de transmissão.

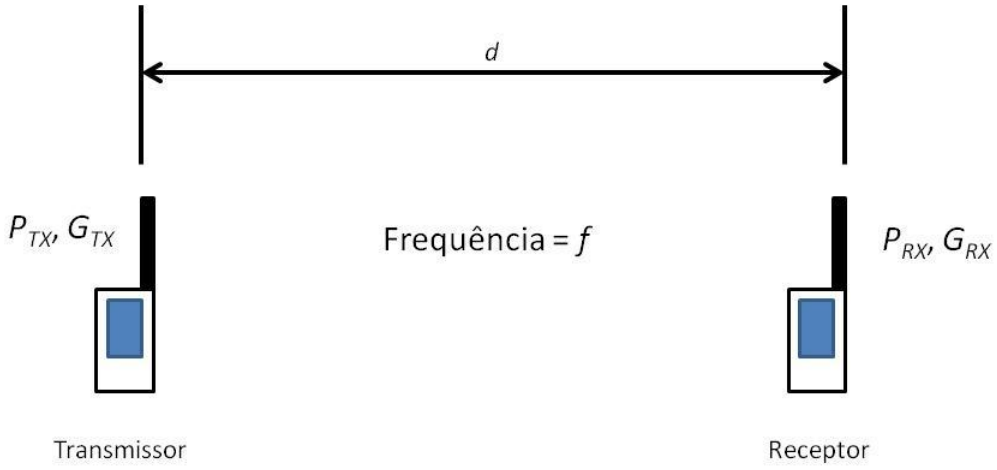


Figura 9 - Caracterização dos elementos presentes na propagação de sinais.

O transmissor está a uma distância *d* do receptor e ambos estão sintonizados em uma frequência *f* de propagação. O transmissor utiliza a uma

potência P_{TX} e sua antena possui um ganho de G_{TX}. Do lado receptor, este possui um ganho de antena de G_{RX} e potência de recepção P_{RX}. A propagação, nestas condições, pode ser considerada em espaço livre e é descrita pela fórmula de Friis (RAPPAPORT, 2002), na Expressão (1):

$$P_{RX} = P_{TX}.G_{TX}.G_{RX}.\left(\frac{\lambda}{4\pi d}\right)^2 \quad (1)$$

Esta fórmula pode ser apresentada também na seguinte forma na Expressão (2):

$$P_{RX} = \frac{P_{TX}.G_{TX}.G_{RX}}{\left(\frac{4\pi d}{\lambda}\right)^2} = \frac{P_{TX}.G_{TX}.G_{RX}}{L_{EL}} \quad (2)$$

Onde L_{EL} é a atenuação no espaço livre. Para fins práticos se utiliza o decibel (dB) para as grandezas envolvidoas, como na Expressão (3):

$$P_{RX_{dBm}} = P_{TX_{dBm}} + G_{TX_{dBi}} + G_{RX_{dBi}} - L_{EL_{dB}} \quad (3)$$

3.6.2 Modelo de propagação *Log-Distance*

O modelo do espaço livre apresentado no item anterior é bastante simples, carecendo de refinamentos para considerar também os efeitos do ambiente na atenuação do sinal. O modelo de propagação de sinais de rádio *Log-Distance* é muito utilizado na caracterização de ambientes indoor, e se baseia no cálculo da potência recebida em um receptor a partir de uma medida de referência a uma distância conhecida d_0 em relação ao transmissor e de características físicas do meio de transmissão (RAPAPPORT, 2002). Na Figura 10 identificam-se os elementos necessários na exemplificação do modelo de propagação *Log-Distance*. Neste caso uma medição de referência é realizada e determinada a uma distância conhecida d_0.

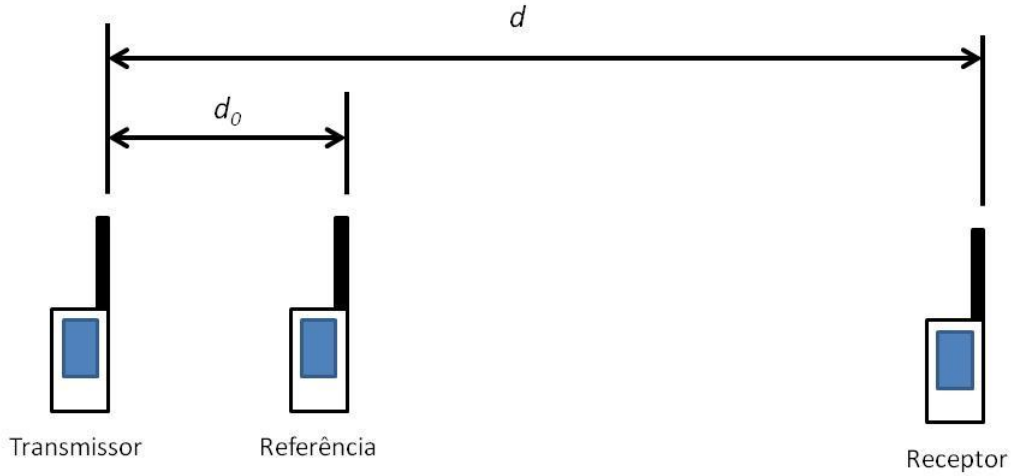


Figura 10 - Referência utilizada para o cálculo da potência recebida.

Neste caso, a potência recebida a uma distância *d* do transmissor é dada pela Expressão (4):

$$\frac{P_r(d_o)}{P_r(d)} = \left(\frac{d}{d_o}\right)^{\beta} \tag{4}$$

Onde $P_r(d_0)$ é a potência recebida a uma distância d_0, $P_r(d)$ é a potência recebida a uma distância *d* e β é o fator de atenuação que define o ambiente. Modificando esta expressão para a unidade dBm, na Expressão (5):

$$\left|\frac{P_r(d)}{P_r(d_o)}\right|_{dB} = -10.\beta.\log\left(\frac{d}{d_o}\right) \tag{5}$$

Até a distância de referência d_0 define-se a perda em espaço livre. Com relação à distância *d*, a perda é calculada em (5) levando-se em consideração apenas características do meio de transmissão. A expressão total que se obtém para o cálculo, em dBm, da potência recebida em um ponto genérico $d > d_0$ é:

$$P_{RX}(d) = P_{TX} + G_{TX} + G_{RX} - L_T \tag{6}$$

Onde L_T é a atenuação total sofrida, descrita na Expressão (7):

$$L_T = 10.\log\left(\frac{4\pi.d_0}{\lambda}\right)^2 + 10.\beta.\log\left(\frac{d}{d_o}\right) \tag{7}$$

A partir da Expressão (7) verificamos que existe uma forte dependência da atenuação total no meio em relação ao tipo de ambiente em que ocorre a propagação do sinal. Em distâncias maiores que a de referência, onde ocorre a maior parte da influencia do ambiente sobre a propagação, o fator β caracteriza o ambiente, influenciando a atenuação total.

4 Propostas Existentes de Gerência de RSSF

A questão gerência de RSSF ainda é um termo em aberto. Muitas propostas consideram uma autonomia total das RSSF na sua organização e de certa forma indicando a não necessidade de gerência deste tipo de rede. Entretanto, esta visão é distorcida, uma vez que sempre 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 RSSF para as mais diversas finalidades. Entretanto, a maioria das RSSF estará operando 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 vertente de investigar formas factíveis e práticas para gerência de RSSF e sua integração com a Internet. Existem algumas abordagens que sugerem a utilização de alternativas para a gerência de RSSF, na tentativa de se criar paradigmas de gerência centrados em arquiteturas de auto-organização e autogestão (XIAO, 2007).

4.1 WSNMP

Uma proposta é a arquitetura de gerência de RSSF do tipo *WSNMP*, ou *Wireless Single Netwok Management Protocol* (ALAM, 2008). Nesta proposta foi implementada uma solução de gerência hierárquica. Os agentes de gerência seguem uma implementação distribuída por toda a rede 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 funcionalidade de *Intermediate Network Manager*, como na Figura 11.

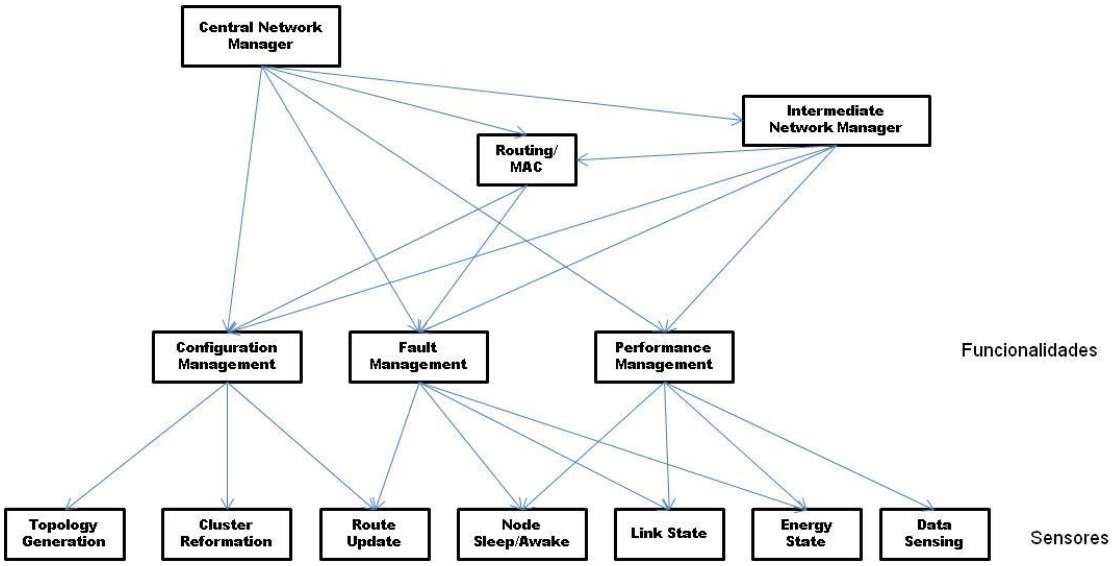


Figura 11 - Arquitetura WSNMP (ALAM, 2008).

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

A gerência de configuração do modelo de gerência *WSNMP*, ou *Configuration Management*, 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 da reformulação de clusters, geração de topologia e atualização de rotas.

- Reformulação de *clusters* (*Cluster Reformation*):

  Um *cluster* é uma reunião de sensores, interconectados, que estão hierarquicamente sujeitos a um *cluster head*. Um *cluster head* é um dos sensores presentes em uma rede *WSN*, configurado para agir como concentrador de todo o *cluster* de sensores e passar a diante os dados através da rede *WSN*.

  Um cluster é formado e reformulado a partir dos dados coletados da rede de sensores. A reformulação envolve o conhecimento do estado de energia dos sensores, uma vez que a função de um sensor no cluster determina o consumo de energia que este terá. Um *cluster head* é

um sensor que gasta mais energia e por isso outros sensores podem ser eleitos *cluster head*, conforme o estado da energia.

- Geração de Topologia (*Topology Generation*):

  A geração de topologias trata da definição das interconexões entre os sensores de um determinado cluster. Através da geração de topologias as interconexões entre o *cluster head* e os sensores pertencentes ao cluster, assim como interconexões entre sensores, são definidas e guardadas pela Gerência de Configuração para reutilização em futuras configurações.

- Atualização de rotas:

  As rotas são definidas pela Gerência de Configuração de modo a definir o fluxo de informação através do cluster. A atualização de rotas esta logicamente sujeita a Gerência de Configuração, pois a geração da topologia e a formação, e reformulação, de clusters são funções da Gerência de Configuração.

A Gerência de Falhas, ou *Fault Management*, é responsável por determinar, através de um mecanismo de detecção de falhas, se um sensor deve ou não ser considerado com 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 dormindo ou acordados, estado de um link de comunicação e estado da energia em um sensor para definir uma falha.

- Estados dos sensores (*Node Sleep/Awake*):

  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 também pode sofrer a transição entre estados dormindo e acordado caso haja a necessidade de economia de energia.

- Estado do Link (*Link State*):

  Os sensores e o cluster head devem estar permanentemente trocando informações. Caso isso não se verifique, em se tratando de sensores reconhecidamente 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.

- Estado de energia (*Energy State*):

  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. Por exemplo, a monitoração do estado da bateria dos sensores é desempenhada pela Gerência de Falhas, gerando evento de falhas por nível crítico de energia e monitorando a carga total, respectivamente.

A Gerência de Desempenho, ou *Performance Management*, é responsável por monitorar a rede WSN de modo a manter o consumo de energia o mais otimizado possível. A monitoração do estado dos links, energia, sensores dormindo e acordados e detecção de dados na rede, é realizada pela Gerência de Desempenho com esse objetivo.

- Detecção de Dados (*Data Sensing*):

  Um sensor deve, de tempos em tempos, se comunicar com outros sensores ou com o *cluster head* para garantir que um *link* esteja em funcionando. 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 identificamos um nível de complexidade que não condiz com a implementação de sensores simples para medições em campo. Para a realização do protocolo *WSNMP* é necessária a existência de uma pilha de protocolos em cada um dos sensores.

4.2 MANNA

Outra proposta feita para a gerência de RSSF é a arquitetura de gerência de sensores *MANNA, Management Arquitecture for Wireless Sensor Networks* (RUIZ, 2003). O objetivo central desta arquitetura de gerência é definir a

rede 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 (gerências de configuração, gerência de falha, gerência de desempenho, gerência de segurança e gerência de accounting), a arquitetura *MANNA* sugere a criação 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, *sensing*, processamento e comunicação) como na Figura 12.

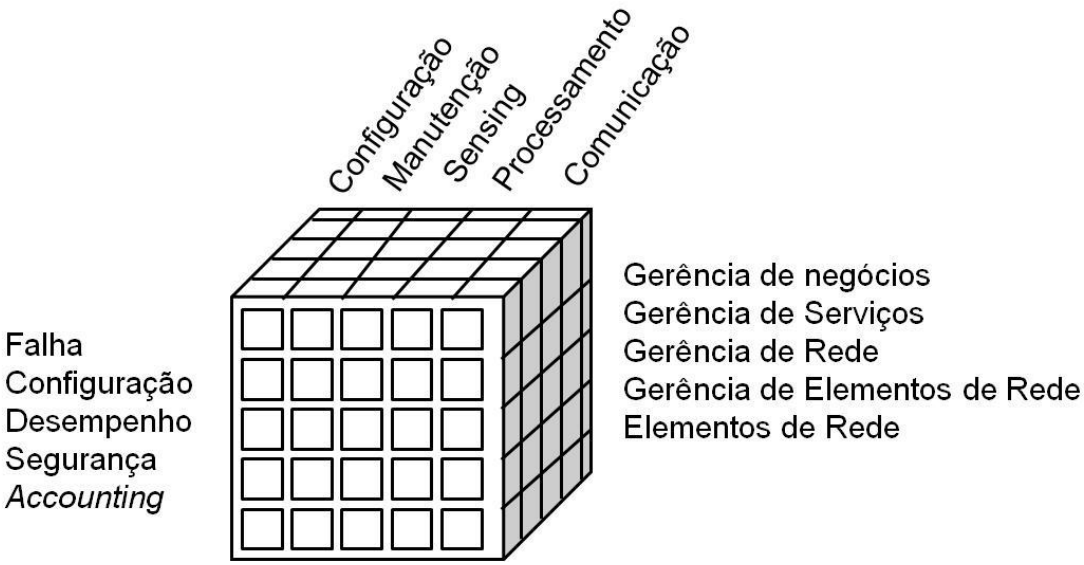


Figura 12 - Arquitetura de gerência MANNA (RUIZ, 2003).

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 em *MANNA*, que não destrói totalmente os conceitos anteriores de gerência de redes de dados, mas propõe uma extensão para a arquitetura clássica de gerência de rede de dados.

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 as diversas situações. Esta extensão foi fundamental para a proposta de *MANNA*, pois segundo defende o autor uma gerência centralizada, como *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 uma gerência centralizada e por isso desqualificando o protocolo *SNMP*.

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

A rede WSN é tipicamente orientada a monitoração de dados e, portanto, as arquiteturas de gerência propostas até então refletem este paradigma. No entanto, a gerência da rede WSN permanece importante do ponto de vista operacional, pois estende a abrangência dos objetos a serem gerenciados, enriquecendo em detalhes a visão da gerência, podendo inclusive ser dividida em grupos distintos de usuários. Esse foco será utilizado na engenharia da MIB para a rede de sensores sem fio, onde a gerência de rede e de dados serão separadas formalmente.

4.3 Propostas baseadas na integração com SNMP

Na literatura existem algumas alternativas para a gerência de RSSF utilizando o protocolo *SNMP*, sendo realizada a integração deste protocolo com a RSSF de diversas formas. Uma tentativa de se implementar o *IPv6* em redes de sensores sem fio é a técnica mais recente, visando o reaproveitamento da pilha de protocolo *IP* e assim adaptar um protocolo de gerência de redes mais facilmente (CHAUDHRY, 2010). Tendo acesso a um protocolo já conhecido, engenheiros e desenvolvedores de sistemas 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 normatizada pelo *IETF* através do *6LoWPAN*, ou *IPv6 over Low Power Wireless Personal Area Network* (COLITTI, 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 de traduções via Proxy, por exemplo.

Outra proposta alternativa é a utilização de agentes intermediários de gerência de RSSF conectados a um *Proxy* através de um protocolo proprietário, por exemplo, o *LiveNCM*, ou *LiveNode 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 tem-se um *Proxy* utilizando não só o protocolo *SNMP*, mas também implementado uma camada intermediária para a adaptação da gerência via *SNMP* à RSSF, via o protocolo proprietário *LiveNCM*.

A seguir será proposta uma arquitetura de gerência de RSSF que utiliza uma interface de integração com uma gerência através de um Agente *Proxy*, que implementa diretamente as pilhas de comunicação da RSSF e *SNMP* em um único *hardware*. Esta integração permite que a RSSF seja considerada um nó de rede TCP/IP, disponibilizando dados de gerência da RSSF e monitorações através de um único método de gerência, via comandos e eventos *SNMP*.

5 Uma Proposta para a Gerência de RSSF com *SNMP*

Em uma típica gerência de redes, o protocolo pelo qual se traduz toda a informação importante e pertinente à gerência é implementado sobre uma pilha de protocolo, que é compartilhada por todos os elementos da rede em questão. Nesta proposta a RSSF seria mais um elemento da rede TCP/IP, como tantos outros dispositivos, que também são gerenciados. Portanto, é natural que as estratégias largamente utilizadas para gerência de redes sejam também apropriados para gerência de RSSF. Interessante notar que as RSSF se caracterizam como redes para comunicação Máquina-Máquina (M2M). O protocolo de gerência mais adequado para a gerência de redes M2M é o SNMP (AN870, 2009), largamente utilizado atualmente em função de seu desenvolvimento ter se dado dentro da Internet.

O protocolo *SNMP*, *Simple Network Management Protocol*, foi implementado sobre a pilha da rede *IP* visando a gerência de grandes redes de computadores, nos quais a pilha *IP* é uma constante presença (STALLINGS, 1999). A vantagem que se extrai 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. Por exemplo, a 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 de usuário, e é contabilizado da mesma forma.

Esta característica da rede de gerência baseada no protocolo *SNMP* se baseia em um recurso importante deste protocolo que é a definição dos objetos de gerência, ou *Object Identification* (*OID*), que nada mais são que uma representação simbólica para uma determinada informação presente na rede, e através destes objetos a gerência trata as informações de forma unificada. No protocolo *SNMP*, essas informações são traduzidas pela base de dados de objetos *MIB*, ou *Management Information Base*. Ou seja, baseando-se na *MIB*, a gerência é capaz de discernir os diversos objetos e a informação que cada um possui.

Neste momento outra característica das redes de gerência deve ser introduzida. Com o objetivo de se implementar um determinado protocolo de

gerência através de uma rede, a pilha de protocolo deve ser compartilhada pelos elementos da rede. Essa pilha se faz presente nestes elementos através de agentes de gerência. Os agentes, que podem ser partes do *software* do elemento gerenciado ou equipamentos em separado (*probes*), interagem com o servidor de gerência atendendo a requisições do tipo *GET* ou *SET*, ou simplesmente informando sobre eventos seguindo uma determinada política, alarmes ou *TRAPS*. Essa política pode ser definida por limiares ultrapassados, como por exemplo, um número de retransmissões de pacotes, ou eventos booleanos *true* e *false*, por exemplo, *true* para fonte de energia *backup* presente e *false* para fonte de energia *backup* ausente. Na ocorrência de algum destes eventos, o agente direciona à estação de gerência um *TRAP*. Tipicamente o *TRAP* possui informações sobre o evento, a estação em que ocorreu o evento, o momento de ocorrência e em alguns casos, para evitar a inundação da rede com *TRAPS*, a periodicidade do evento. Além do evento reportado, o agente também pode ser implementado no sentido de reportar quando tal evento não mais ocorre, através de um novo *TRAP* indicando a interrupção do evento, comumente chamado de “limpeza dos alarmes”.

O problema que se apresenta então é como introduzir o conceito de gerência de redes em uma RSSF, satisfazendo as características de uma rede de sensores sem fio. O paradigma deve se adaptar às necessidades de uma rede de sensores sem fio, visando a integração com ferramentas de gerência já concebidas, por motivos práticos, e deve ser viável para implementação das funcionalidades presentes no protocolo de gerência *SNMP* (*GET*/*SET*/*TRAP*).

De uma forma bem direta, a implementação de uma pilha de protocolo de gerência no sensor implica em dificuldades computacionais relevantes. Se levarmos em consideração a presença do agente em cada um dos sensores, implicaria em memória e processamento extra, em detrimento da autonomia do sensor. É preciso então utilizar uma estratégia que aproveite a estrutura da rede de sensores sem fio, utilizando-a como rede de transporte, sem que grandes implementações sejam necessárias diretamente nos sensores. A estratégia que será seguida é a abordagem do Agente *Proxy*, ou elemento mediador *Gateway*, que se localiza entre a rede de sensores e a rede de gerência (Figura 13). A construção deste Agente *Proxy* levará em consideração, por um lado, os paradigmas de gerência de redes *IP*, representada pelo protocolo de gerência

SNMP, e por outro lado pela implementação de uma pilha de protocolo voltada para as especificidades dos sensores sem fio. O conjunto *software* e *hardware* será abordado em detalhes na descrição da metodologia utilizada para a implementação do protótipo do Agente *Proxy*.

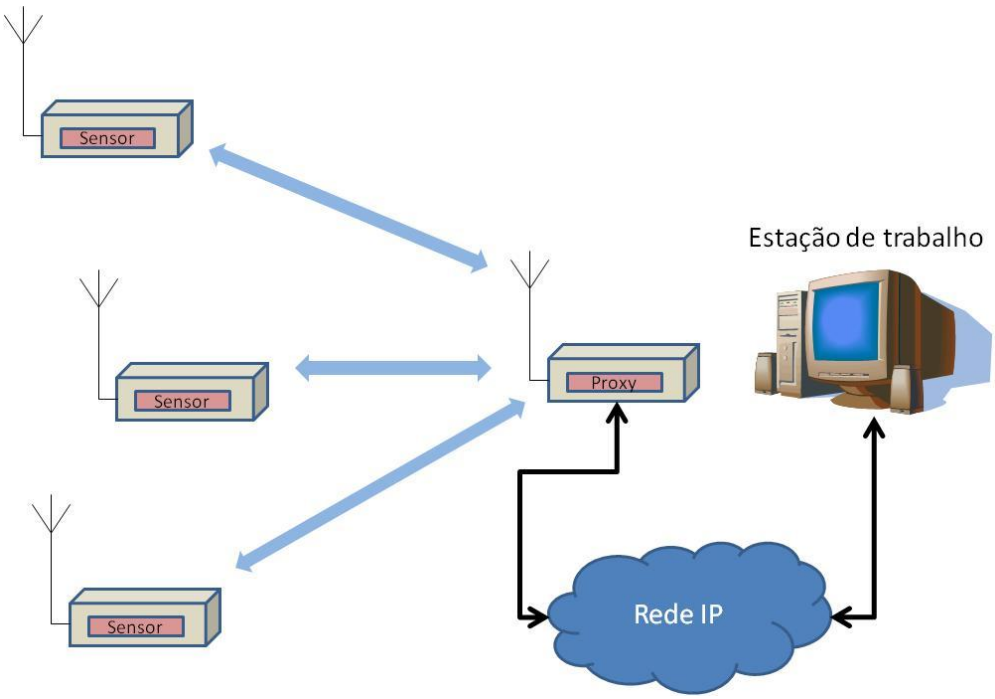


Figura 13 - Modelo Proxy conceitual.

A Figura 12 representa um exemplo de utilização conceitual do Agente *Proxy*, não estando limitado a este tipo de topologia de rede. A visão da estação de gerência com relação a rede de sensores sem fio é de um elemento de rede *IP*, sendo gerenciado pelo conjunto de *MIB's* 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.

5.1 Planos de Gerência de uma RSSF

De modo a se criar uma base de dados coerente com a proposta de estações de gerências, diferenciadas de acordo com o interesse de diferentes usuários, é importante que se definam planos de gerência distintos, onde cada um dos papéis da gerência de RSSF é definido separadamente. Esta organização contribui para a engenharia da *MIB*, separando mais claramente os objetos de interesse de um usuário e de outro, assim como a visibilidade dos objetos, ou seja, se são apenas de leitura (*GET*), leitura e escrita (*GET*/*SET*) ou eventos (*TRAPS*).

5.1.1 Plano de gerência da rede (PGR)

No plano de gerência de rede RSSF, ou PGR, definem-se os parâmetros de rede entre as estações transmissora e receptora, não importando o papel que cada uma das estações desempenha na rede, se sensor ou agente *Proxy*. Esses parâmetros são capazes de indicar para uma estação de gerência se um sensor está em um estado de operação normal, do ponto de vista de conectividade, ou inviável tanto para a gerência quanto pela confiabilidade das medições por ele feitas.

O PGR deve implementar as funcionalidades de Configuração (*GET* e *SET SNMP* com variáveis de leitura e escrita), desempenho (*GET SNMP* com variáveis apenas de leitura) e falhas (*TRAP SNMP* gerado por correlação ou mesmo enviado diretamente dos sensores), de modo a oferecer uma interface com estações de gerência.

O parâmetro chave nesta arquitetura é a potência de sinal na recepção, ou *RSSI*, que é medido nos sentidos *uplink* (*RSSIu*), do sensor para o gateway, e *downlink* (*RSSId*), do gateway para o sensor, adotando a referência do agente *Proxy*.

Criando-se regras de desempenho no comportamento em ambientes diversificados, *indoor* e *outdoor* da RSSF, podem-se definir níveis de *thresholds* e ações sobre a rede como aumento ou diminuição da potência de transmissão via

comandos do protocolo *SNMP*. A gerência pode ordenar uma reconfiguração da potência de transmissão em algum dos nós da RSSF, via comandos *SET SNMP*, de modo a ajustar uma melhor condição de recepção em nós vizinhos. A potência de transmissão, portanto, é mais um parâmetro a ser levado em consideração no plano de gerência da RSSF, uma vez que se relaciona diretamente com a *RSSI* nas duas direções. A potência de transmissão é um parâmetro que deve ser independente do elemento, sensor ou agente *Proxy*, e que pode ser ajustado segundo as características do meio.

Ainda no plano de gerência de RSSF, os sensores devem se conectar a um Agente *Proxy* através de um canal específico de comunicação rádio, contido em uma faixa de frequência de operação comum aos dois elementos, sendo novamente um parâmetro de configuração onde se necessita de visibilidade de leitura e escrita (*GET*/*SET*). Pode-se optar por um único canal, quando o Agente *Proxy* se comunica com um sensor apenas por vez, ou por múltiplos canais, para que mais de um sensor seja acessado por vez. Os sensores também devem ser programados com identificações únicas, como um endereço definido na camada de rede da arquitetura da rede RSSF.

É possível também utilizar o plano de gerência de rede RSSF para armazenar dados sobre as características do meio em que os sensores se encontram. Através de uma prévia calibração podemos obter o valor de beta do modelo de propagação *Log-distance* ou *Shadowing* e através deste disparar alarmes, no caso *TRAPS*, preditivos de acordo com alterações no meio e possíveis degradações de sinal, com a possível perda de pacotes. O método de obtenção do beta, da seção de Análise Geral, será usado para comprovação na seção de metodologia, onde os experimentos são caracterizados.

5.1.2 Plano de gerência de Dados (PGD)

No plano de gerência de dados, ou PGD, se definem as variáveis de interesse relacionadas diretamente ao processo monitorado, ou seja, as grandezas que efetivamente são mensuradas utilizando-se transdutores especializados, como por exemplo, temperatura e pressão. A variedade ou precisão das medidas fica relacionada diretamente com os transdutores e

circuitos conversores analógico-digitais, ficando por conta dos sensores o tratamento inicial dos dados, adaptando os dados digitalizados aos frames de comunicação ou ainda realizando pré-processamento dos dados antes do envio para o Agente *Proxy*.

No PGD se aplicam as funcionalidades de configuração, desempenho e falha, assim como na PGR. A configuração é realizada por comandos *GET* e *SET SNMP*, o desempenho é mensurado a partir de comandos *GET SNMP* e as falhas são enviadas a estação de gerência através de *TRAPS SNMP*.

Do ponto de vista do PGD nossa principal preocupação é em obter dados, o que não impede que funcionalidades de atuação sobre os sensores sejam criadas para realizar, por exemplo, recalibração de transdutores sem que seja necessária a remoção do sensor para calibração. Os atuadores podem ser ainda a principal funcionalidade de um sensor, onde são especializados em atuar sobre o processo monitorado.

5.2 Arquitetura de Gerência via Agente Proxy

De forma a criar um elo entre a rede RSSF e a rede de gerência uma abordagem deve ser utilizada de forma a adaptar as características do protocolo de gerência às características da rede RSSF. Uma vez que o protocolo de gerência escolhido é o protocolo *SNMP*, as características deste protocolo devem ser estendidas à rede RSSF, sendo capaz de responder satisfatoriamente às requisições de uma estação de gerência.

A abordagem do Agente *Proxy* é bastante aderente a esta necessidade pela flexibilidade como pode ser implementada, tanto por *hardware* como por *software*. Devemos nos lembrar de que o Agente *Proxy* é um ponto de convergência de vários sensores e desempenha um papel fundamental na gerência nos planos PGR e PGD. Sendo assim, o nível de complexidade de implementação está em um nível mais elevado que os sensores.

A estrutura interna das pilhas de protocolo do Agente *Proxy*, onde ocorre o processo de tradução dos protocolos é mostrado a seguir na Figura 14.

Neste modelo são definidas, do ponto de vista de protocolo, as partes funcionais do Agente *Proxy*. O interesse principal é que, através de uma estação de gerência de rede, os sensores sejam gerenciados através de um protocolo aberto.

A questão do protocolo está ligada a simplicidade na implementação do *firmware* e disponibilidade de plataformas já desenvolvidas para a pilha de protocolo *SNMP*, necessários ao desenvolvimento das camadas no Agente *Proxy*. Outro aspecto da escolha do protocolo aberto é a questão do desenvolvimento da estação de gerência. Com a utilização de um protocolo aberto o usuário pode escolher, em uma longa lista, a aplicação que melhor lhe convir como aplicação de gerência.

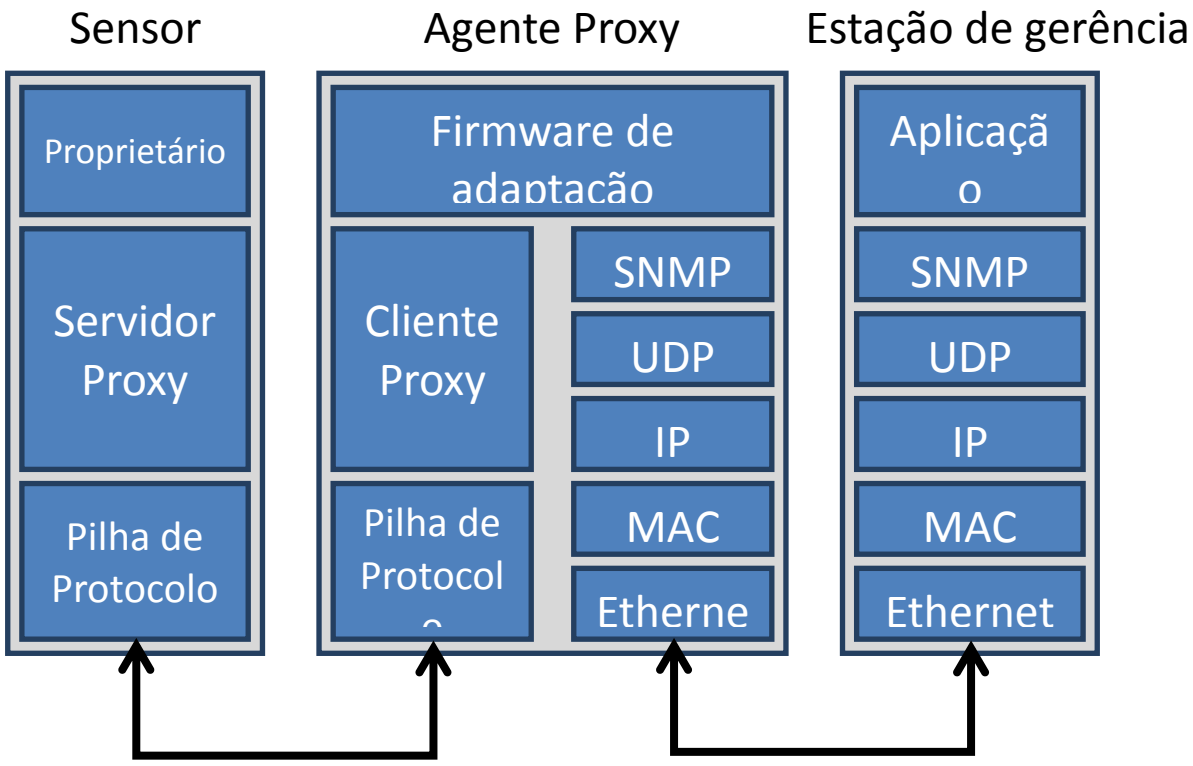


Figura 14 - Proxy entre a rede RSSF e a rede IP.

Além de proporcionar uma forma de tradução transparente à gerência da RSSF, o Agente *Proxy* deve agir de forma a ser o ponto de acesso ou atuação, além de transmissor de eventos. Uma forma de se isolar a RSSF da rede de gerência é implementar no Agente *Proxy* a capacidade de *Polling* de dados, ou seja, em intervalos programados e temporizados o Agente *Proxy* realiza a

varredura da RSSF por parâmetros vinculados à *OID's*. A temporização do *Polling* pode ser também um parâmetro de configuração da RSSF. A velocidade em que é feito o *Polling* pode ser determinante na longevidade dos sensores, uma vez que *Pollings* muito rápidos mantém disponíveis dados mais atualizados sobre a RSSF, mas os sensores precisam responder a mais requisições, gastando mais energia.

A estrutura das várias camadas da RSSF deve estar replicada no Agente *Proxy* a fim de, na camada mais alta do Agente *Proxy*, termos os dados gerados pelo *firmware* do sensor recuperados, assim como dados gerados pelo *firmware* de adaptação possam ser passados ao *firmware* do sensor pelo mesmo mecanismo.

Com relação à estação de gerência, esta se comunica com o Agente *Proxy* utilizando *SNMP*. Em nosso caso, o Agente *Proxy* implementa a pilha *SNMP*.

Caracterizando a estação de gerência, esta é uma entidade física, geralmente representada por uma infraestrutura de servidores ou de um único servidor, e recebe toda a informação de gerência vinda dos nós intermediários, ou Agente *Proxy* no nosso caso específico, ou envia comandos para a RSSF a fim de reconfigurar algum parâmetro. Este nó da rede de gerência é o elemento mais alto no nível hierárquico da gerência de redes, visto que é o elemento que possui uma característica única entre todos os outros 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 dos Agentes *Proxy*, através de um determinado protocolo, para uma linguagem humanamente compreensível. No caso de rede de sensores sem fio, informações do tipo potência de sinal e canal de transmissão são informações presentes neste tipo de rede, mas que necessitam de tradução para os padrões da linguagem humana.

6 Engenharia da MIB

A *MIB* 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 medido. 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 anterior na *OID*.

A construção da *MIB* deverá levar em conta a relação entre as funcionalidades pertinentes à rede RSSF e os parâmetros de interesse aos usuários da gerência de RSSF (HARITSA, 1993)(SHÖNWÄLDER, 2005). Isso se concretiza separando-se o conjunto de objetos gerenciáveis existentes em nossa rede e relacionando diretamente com as possíveis funcionalidades (TYNAN, 2005). Esses objetos gerenciáveis devem obedecer também aos critérios de visibilidade, onde alguns parâmetros devem ser apenas de leitura, ou aplicáveis apenas ao comando *GET SNMP*, ou de escrita e leitura, ou aplicáveis aos comandos *GET* e *SET SNMP*. Além dos parâmetros com critério de visibilidade, outro conjunto de parâmetros pertence ao conjunto dos eventos, ou *TRAPS*, e será utilizado para indicar um estado através da correlação entre parâmetros de configuração e desempenho.

6.1 Abrangência da MIB

Na RSSF, abordaremos as funcionalidades de gerência de configuração, gerência de desempenho e gerência de falhas, identificando as grandezas pertinentes e comentando sobre a que comandos *SNMP* estão relacionados. As gerências de segurança e de accounting, presentes no modelo clássico de gerência de redes, não serão abordadas neste estudo, pois a segurança e a contabilização nos sensores não será o foco deste trabalho, mas podem ser abordados em trabalhos futuros. Fará também parte da modelagem dos dados da *MIB* e separação explícita dos planos de gerência de RSSF e dados, de modo a implementar uma especialização neste nível nas *OID´s* de gerência.

Considerando os conceitos de PGD e PGR, descritos na proposta de gerência de RSSF, é definida uma base de informações que representa o conjunto de grandezas de interesse à gestão da RSSF. Este conjunto de informações recebe o nome de *MIB*, e é representada por um conjunto de dados divididos em níveis hierárquicos organizados.

De modo a representar uma determinada grandeza, a *MIB* disponibiliza essa informação através de *OID,* que deve não só determinar que grandeza está sendo disponibilizada, mas também a hierarquia a que esta grandeza pertence. Este conceito de hierarquia é importante para que se mantenha o padrão na formação das *OID´*s e se possa reunir o maior número possível de variáveis em uma única organização.

A ideia de organização hierárquica em gerência de redes, de uma forma geral, se justifica também na forma em que se modela a base de dados, em nosso caso a *MIB*. Levando-se em conta a consideração inicial sobre a divisão em dois planos de gerência, a modelagem da *MIB* deve refletir este paradigma, de forma a oferecer aos usuários e administradores de rede a segregação de funções em uma rede de gerência, para refletir os planos.

O conjunto de *OID´s* possui um ponto de partida, em nosso caso um conjunto de *OID´s* que estão reunidas sob a hierarquia de *OID´s* privadas. Esse ponto hierárquico inicial é, segundo o padrão de *OID´s* para *SNMP*:

iso.org.dod.internet.private.enterprises.

Esta hierarquia inicial possui uma representação numérica própria, que representa cada um dos níveis hierárquicos. Em representação numérica, esta representação hierárquica é definida como *1.3.6.1.4.1*, ou de uma forma mais clara, *iso(1).org(3).dod(6).internet(1).private(4).enterprises(1)*. O mapeamento entre uma forma e outra está na *MIB*, onde todas as correspondências estão apropriadamente representadas.

A definição da nossa MIB utiliza a seguinte estrutura:

1.3.6.1.4.1.23955.2, onde os seis primeiros níveis representam a hierarquia inicial, da qual ramificam os níveis hierárquicos da arquitetura de

gerência de RSSF. O nível hierárquico sete, representado pelo valor 23955, representa o nível hierárquico PUC-Campinas (SCHIMIDT, 2007) e o nível hierárquico oito, representado pelo valor 2, representa o ponto no qual se ramificam todas as *OID´s* relecionadas a gerência de RSSF. Em especial esta hierarquia foi requisitada pela PUC-Campinas ao órgão *IANA*, *Internet Assigned Numbers Authority*, que registra e gerencia a propriedade sobre as identificações de objetos, diferenciando-os por fabricante e tecnologia, mas que compartilham o mesmo protocolo.

6.2 Plano de gerência da RSSF

A partir desta organização inicial são definidos seus respectivos níveis hierárquicos, a qual plano de gerência pertence uma determinada grandeza. Dessa forma cria-se uma base formalmente hierarquizada, com a possibilidade de se definir sensores com papéis bem definidos na RSSF. Caso um determinado sensor seja capaz de múltiplas medições com grandezas distintas em planos de gerência distintos, se aplicará a este sensor mais de uma *OID* diferenciando-se cada uma delas pela identificação de plano de gerência e grandeza. A *OID* deve, também, possuir uma atribuição específica para tipos distintos de sensores, quando forem sensores ou Agente *Proxy*.

Na hierarquia de gerência, no plano de RSSF, devem-se considerar as funcionalidades configuração, desempenho e falha no projeto de uma *MIB* para gerência de RSSF, organizada hierarquicamente. Em seguida é feita a proposta para o plano de gerência de dados, com possíveis grandezas de interesse em nossa proposta.

6.2.1 Funcionalidade de Configuração para RSSF

As grandezas relacionadas à funcionalidade de configuração são grandezas relacionadas tanto ao comando *GET* quanto ao comando *SET* do *SNMP*, ou seja, são grandezas ajustadas ou pesquisadas através da estação de gerência utilizando-se o comando de gerência *SET* e *GET* do protocolo *SNMP*, respectivamente. De certa forma o conjunto de parâmetros relacionados com a configuração também deve ser visível ao comando *GET*, pois um relatório de

parâmetros pode ser gerado inicialmente para a confirmação dos parâmetros já configurados e planejamento de alterações. Em resumo, são parâmetros de visibilidade de leitura e escrita. A configuração é importante, pois define o comportamento da RSSF do ponto de vista funcional. A seguir estão relacionadas grandezas de configuração, importantes a gerência de rede, pois são parâmetros diretamente vinculados com o comportamento da rede:

- Sensibilidade

A sensibilidade está relacionada com o nível máximo de erro de pacote, ou *Package Error Rate* (*PER*) admissível pelo sistema. Em um dado cenário de transmissão, similar ao da Figura 9, eventualmente, por ação de ruídos impostos ao sinal transmitido, ocorrerá erro na recepção. A taxa de erro de recepção de *bits*, em uma transmissão digital, recebe o nome de *BER*. A definição de PER (RAPAPPORT, 2002), conhecendo o nível de BER, é:

$$PER = 1 - (1 - BER)^N \tag{9}$$

Onde *N* é o número de *bits* do pacote.

O ruído presente em um sistema de transmissão via rádio P_{Noise}(dBm) é a composição do ruído térmico $P_{Term\text{-}Noise}$com a figura de ruído $P_{Noise\text{-}Figure}$, da seguinte forma:

$$P_{Noise}(dBm) = P_{Term-Noise}\,(dBm) + P_{Noise-Figure}(dB) \tag{10}$$

Onde o ruído térmico é definido como:

$$P_{Term-Noise} = 10.\log\left(\frac{KTB}{10^{-3}}\right)[dBm] \tag{11}$$

K é a constante de Boltzmann, *T* é a temperatura e *B* é a largura de banda.

Um dado valor de PER corresponde a relação entre uma potência limite e o ruído total. Esta potência é a sensibilidade e diz-se que para uma determinada PER, existe uma relação sinal-ruído:

$$SNR = \left(\frac{Sensibilidade}{P_{Noise}}\right) \tag{12}$$

Em termos de configuração, este parâmetro pode ser útil na definição de um limite máximo para a PER.

- Potência de transmissão:

  A potência de transmissão em RSSF pode ser um limitante em relação ao alcance e o tempo de vida de baterias, no caso de regime de trabalho autônomo. Assim sendo, é de interesse da gerência manter um controle sobre este parâmetro, realizando comandos *GET*/*SET*.

- Canal de transmissão:

  Na RSSF, o canal de transmissão representa a frequência em que está sendo transmitido um sinal de RF. Em termos de *SET* e *GET* o canal de transmissão pode tanto ser o valor da frequência propriamente dita como também uma indexação para um dado valor de frequência do canal.

- Rota

  A rota, em RSSF, representa um registro de uma conexão entre dois sensores em uma tabela de roteamento. De forma a utilizar este princípio na gerência de RSSF, os comandos *GET* e *SET* podem ser utilizados para descobrir o próximo salto de um sensor ou configurar o próximo salto, respectivamente. Problemas relacionados com uma rota podem ser reportados através de *TRAPS*, realizando-se a correlação de informações obtidas dos sensores, por exemplo, a interrupção na comunicação com algum sensor, e a última tabela existente.

- Ganho da antena

  O ganho da antena representa um parâmetro em decibéis (dBi). É bastante conveniente que este parâmetro seja configurado, uma vez que é utilizado no planejamento da RSSF.

- Atenuação do cabo:

  No sistema implementado fisicamente, pode haver a necessidade de inclusão de cabeamento entre o rádio e a antena. Dependendo do comprimento e tipo de cabo a atenuação pode se tornar

um problema, do ponto de vista de projeto, inserindo perdas na potência transmitida e recebida. Sendo assim, este parâmetro se torna de interesse da gerência de RSSF, uma vez que pode ser inserido via comando *SET*.

- Banda:

  A frequência, na RSSF, é a frequência base de operação do rádio presente em cada um dos sensores. Por exemplo, os rádios dos sensores estão, por *default*, programados para a frequência base de 915 MHz, subdividido em canais. Assim sendo, utilizando o comando *SET* de canais via indexação, o *SET* da frequência base altera a frequência de transmissão ao longo de todos os canais.

- Temporização de *polling* de sensores

  A comunicação entre os sensores e o Agente *Proxy* pode ser dispendiosa do ponto de vista energético para os sensores, assim como o tráfego entre o Agente *Proxy* e a internet também pode ser dispendioso para o Agente *Proxy* por ser um *hardware* mais simples. Desta forma, realizar o *polling* dos sensores a partir do Agente *Proxy* pode ser uma técnica viável para balancear o peso da gerência no sistema. Utilizando o *SET* do tempo de *polling* define-se, portanto, o intervalo de tempo em que uma requisição é válida até que outra possa ser feita. Definindo valores de tempo de *polling* grandes, diminui-se bastante o tráfego na RSSF, mas a gerência se torna lenta. Valores de *polling* rápidos aumentam o tráfego na RSSF, mas tornam a gerência mais rápida.

- Tipo de nó

  Na RSSF, um sensor pode ser o equipamento que está de fato realizando uma medida ou também pode ser um simples repetidor. Esta funcionalidade pode ser definida via comando *SET* do *SNMP* para que a gerência busque por medições de sensores que realmente são equipamentos de medida e não repetidores.

Na Figura 15 verifica-se a representação gráfica do ramo da *MIB* responsável pela definição dos objetos de gerência de configuração, do plano de rede.

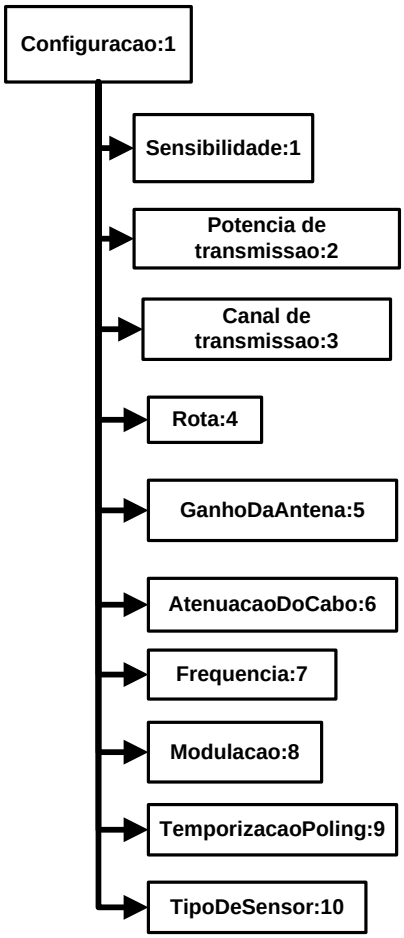


Figura 15 - *MIB* de configuração, plano de rede.

6.2.2 Funcionalidade de Desempenho para RSSF

A funcionalidade de desempenho para a RSSF define o conjunto de variáveis que influenciam na disponibilidade da rede e no desempenho que estes terão. São variáveis de visibilidade de leitura apenas, ou seja, estão disponíveis apenas para comandos *GET SNMP*. Através destas variáveis o Agente *Proxy* mantém um conjunto de dados para futuras correlações com os dados de configuração.

- *RSSI*

  Radio Signal Strenght Indicator, ou *RSSI*, é a medida de potência do sinal de recepção e é medida em *dBm*. Através da monitoração da *RSSI* a gerência de RSSF pode tomar decisões a respeito de problemas de transmissão e necessidade de reconfiguração da potência de transmissão dos sensores vizinhos. Para a gerência de RSSF, esse parâmetro deve estar visível como um parâmetro apenas de leitura.

- *RSSI* média

  A *RSSI* média pode ser monitorada para que eventuais decisões a respeito de estado de um link entre duas estações de rádio sejam tomadas com bases estatísticas e não imediatas. Com este princípio espera-se que eventuais reconfigurações de potência de rádio sejam necessárias se o comportamento da *RSSI* no tempo esteja fora de um limite, onde se pode considerar como sendo o limite de degradação por ruído e aumento de *PER*. Essa medida pode ser usada como base na correlação com valores definidos como limites e *TRAPS* são disparados como aviso.

- *RSSI* desvio padrão

  Através do cálculo do desvio padrão para um universo de medidas de *RSSI* de um determinado sensor, é possível concluir sobre o comportamento da *RSSI* com relação ao ambiente em que o sensor esta inserido. A variação da *RSSI*, com Agente *Proxy* e sensores não móveis, em um ambiente com poucas mudanças físicas, deve ser mínima. Já em ambientes em que existe grande movimentação de pessoas, por exemplo, a influência sobre a *RSSI* será muito maior por questão de obstáculos físicos móveis. Através desta leitura a *MIB* deve estar preparada para avisar, via correlação e *TRAPS*, uma constante mudança no ambiente de medida.

- Beta médio

  O fator Beta, como discutido anteriormente, é um fator dependente do ambiente e que pode sofrer variações de acordo com

obstáculos físicos que possam oferecer perda ou reflexão de rádio. Sendo assim, para uma situação em que o Agente *Proxy* e os sensores estão imóveis, em ambientes com mínima modificação, o Beta deverá se manter o mesmo e sofrer variações à medida que o sensor se afasta do Agente *Proxy*, ou de outro sensor. Monitorando-se o Beta médio pode-se avaliar sobre eventuais problemas de propagação em um ambiente e utilizar este dado para reconfigurações de potência de transmissão ou até mesmo alteração física dos sensores.

- *PER* (*Packet Error Rate*)

  A medição de PER é utilizada para a avaliação da transmissão entre duas estações. Sendo assim pode ser utilizada no julgamento, e eventual reconfiguração, de sensibilidade.

Na Figura 16 verifica-se a representação gráfica do ramo da *MIB* responsável pela definição dos objetos de gerência de desempenho, do plano de rede.

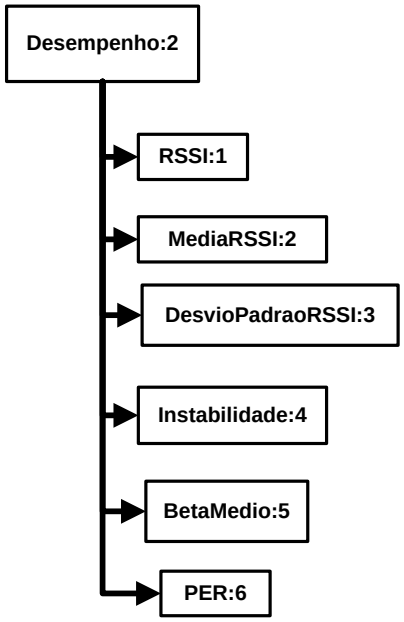


Figura 16 - *MIB* de desempenho, plano de rede.

6.2.3 Funcionalidade de Falhas para RSSF

A funcionalidade de falhas para RSSF é responsável por correlacionar dados de configuração e de desempenho e criar eventos através de comparações. As comparações, de uma forma genérica, podem basear-se nos limites definidos por configuração, onde variáveis de desempenho são constante mente comparadas com valores máximos e mínimos.

- Limiar de *RSSI* médio

  Em uma RSSF um dado valor de RSSI médio pode não ser viável para recepção, segundo parâmetros próprios dos equipamentos. Assim sendo um RSSI médio alcançar um determinado valor pode ser considerado um evento, neste caso um evento indicador de falha ou falha iminente. Esse tipo de evento pode ser disparado segundo um critério de valor e ser definido por características próprias de cada RSSF, correlacionando-se a RSSI média e interrupções de comunicação.

- Limiar de *RSSI* desvio padrão

  Caso as medições de RSSI com o tempo produzam um desvio padrão elevado, um evento pode ser gerado com o objetivo de avisar a gerência que várias modificações estão ocorrendo no ambiente, que podem ser prejudiciais ao desempenho do rádio.

- Sensor não encontrado

  Um evento pode ser disparado quando simplesmente um sensor não respondeu a uma requisição. Neste caso diz-se que o sensor não fora encontrado e um evento de falha de transmissão para este sensor é enviado para a gerência de RSSF. Esse evento possui uma conotação bastante genérica, uma vez que pode haver muitos aspectos envolvidos na falha de comunicação. O evento de retorno de comunicação que pode ser representado pelo mesmo *TRAP*, também pode ser utilizado de forma genérica após a normalização da operação.

Na Figura 17 verifica-se a representação gráfica do ramo da *MIB* responsável pela definição dos objetos de gerência de falhas, do plano de rede.

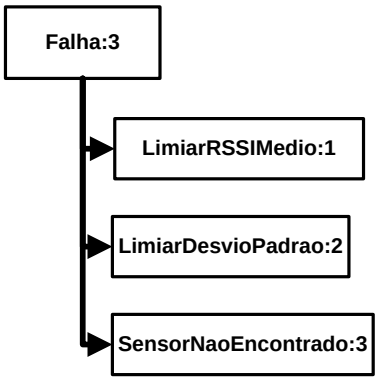


Figura 17 - *MIB* de falhas, plano de rede.

6.3 Plano de Gerencia de dados

Com relação ao plano de dados, uma abordagem similar é utilizada, porém as definições de parâmetros ficam em aberto, uma vez que uma grande quantidade de medidas pode ser utilizada, além de eventuais tratamentos sobre os sinais que podem gerar mais parâmetros, como por exemplo, o cálculo de médias e de desvio padrão.

6.3.1 Funcionalidade de Configuração para Dados

A caracterização de ambientes através do controle de, por exemplo, temperatura é de interesse tanto do ponto de vista de conforto humano como também do pessoal responsável pela manutenção da RSSF, uma vez que a temperatura pode ser um problema em ambientes abertos ou fechados. A integridade de equipamentos eletrônicos fica constantemente sendo atacada por modificações de temperatura e a vida útil de baterias é serialmente comprometida nesta situação (VRUDHULA, 2003), o que justifica a necessidade de se monitorar constantemente a temperatura e suas variações.

Em termos de conforto térmico, o agente deve possuir uma inteligência tal que mudanças bruscas de temperatura ou valores acima ou abaixo de valores tolerados devem gerar eventos. Isso expande o conceito de apenas leituras de

temperatura para um nível de gerência de dados, onde se observam patamares e não apenas valores. Estes patamares devem possuir visibilidade de leitura e escrita, ou seja, devem ser configurados para os valores tolerados em cada sistema. Assim se mantém a flexibilidade esperada do sistema de gerência.

Nessa linha de pensamento, observamos certas semelhanças na engenharia do plano de gerência de RSSF. Grandezas físicas precisam ser monitoradas, patamares necessitam ser respeitados e eventos podem ser gerados através da correlação entre medidas e patamares. Isso verifica o princípio de formação adotado na engenharia de *MIB* para o PGR, propagado para a engenharia de *MIB* para o PGD. Utilizando este princípio, pode-se construir uma estrutura hierárquica condizente com a engenharia de *MIB*, utilizando a mesma linha de pensamento utilizada para o plano de gerência de RSSF.

No caso da grandeza temperatura, podem-se definir variáveis de limite para o ambiente de medidas.

- Temperatura de Trabalho Inferior

  A Temperatura de Trabalho Inferior é responsável por configurar o limite inferior para temperatura do processo sendo monitorado, ou seja, abaixo deste limite a estação de gerência precisa ser avisada através de eventos correlacionados. Esta variável é bastante genérica, tendo um significado que se aplica a qualquer caso de medidas de temperatura.
- Temperatura de Trabalho Superior

  A Temperatura de Trabalho Superior é responsável por configurar o limite superior para a temperatura do processo sendo monitorado.

Na Figura 18 verifica-se a representação gráfica do ramo da *MIB* responsável pela definição dos objetos de gerência de configuração, do plano de dados.

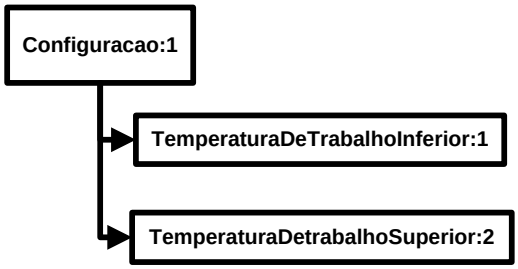


Figura 18 - *MIB* de configuração, plano de dados.

6.3.2 Funcionalidade de Desempenho para Dados

A funcionalidade de desempenho para dados fica bastante sujeita a precisão e calibração dos instrumentos de medida acoplados nos sensores. Considerando que para um conjunto de componentes foi realizada uma calibração e implementação da curva de resposta do transdutor na programação do sensor, podemos utilizá-lo como instrumento de medida. Para o caso de temperatura:

- Temperatura

  A medida de temperatura é o valor medido, já calibrado, no transdutor e convertido para valores digitais, se o sensor for analógico.
- Temperatura Média

  A Temperatura média é um valor calculado periodicamente para a temperatura depois de um número de medidas.

- Desvio Padrão de Temperatura

  O desvio padrão pode ser útil na detecção de comportamentos inesperados no ambiente, como grandes variações de temperatura.

Na Figura 19 verifica-se a representação gráfica do ramo da *MIB* responsável pela definição dos objetos de gerência de desempenho, do plano de dados.

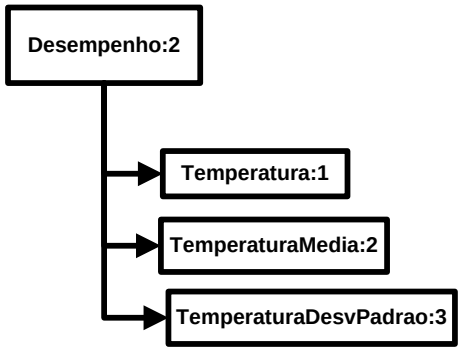


Figura 19 - *MIB* de desempenho, plano de dados.

6.3.3 Funcionalidade de Falhas para Dados

Com relação à Funcionalidade de falhas para dados, considera-se o principio da correlação de configurações e desempenho. Os eventos criados nesta funcionalidade são:

- Limiar de Temperatura Inferior

  É um evento criado a partir da correlação entre a configuração do limite inferior e a medida de temperatura.
- Limiar de Temperatura Superior

  É um evento criado a partir da correlação entre a configuração do limite inferior e a medida de temperatura.

Na Figura 20 verifica-se a representação gráfica do ramo da *MIB* responsável pela definição dos objetos de gerência de falhas, do plano de dados.

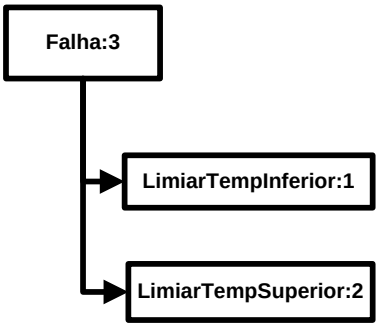


Figura 20 - *MIB* de falhas, plano de dados.

6.4 Relacionamento Plano versus Parâmetros

Com estas duas construções fica clara a relação entre os planos de gerência e as funcionalidades cobertas na engenharia da *MIB*. Na Tabela 1 está definida a matriz de relacionamento entre os planos de gerência e os parâmetros de gerência.

O interessante é que, com este nível de relacionamento entre variáveis e planos de gerência pode-se aplicar a correlação de variáveis de diferentes planos com o objetivo de avaliar cenários mais complexos, como o caso da influência da temperatura na longevidade da bateria dos sensores.

Tabela 1 - Matriz de relacionamento entre plano e parâmetros.

Matriz	Configuração	Desempenho	Falha
Plano de gerencia de RSSF	Sensibilidade, Potencia de Transmissão, Canal de transmissão, Rota, Ganho da Antena, Atenuação do Cabo, Frequencia, modulação, Temporização de polling, Tipo de Sensor	RSSI, Média de RSSI, Desvio Padrão de RSSI, Instabilidade, Beta Medio, PER	Limiar de RSSI Médio, Limiar de Desvio Padrão, Sensor Não Encontrado.
Plano de gerência de Dados	Temperatura de trabalho Superior, Temperatura de trabalho Inferior	Temperatura, Temperatura Média, Temperatura Desvio Padrão	Limiar de Temperatura Superior, Limiar de Temperatura Inferior

Além desta matriz, é possível a construção de um modelo para esta arquitetura como na Figura 21, onde se descrevem as relações entre planos de gerência e funcionalidades.

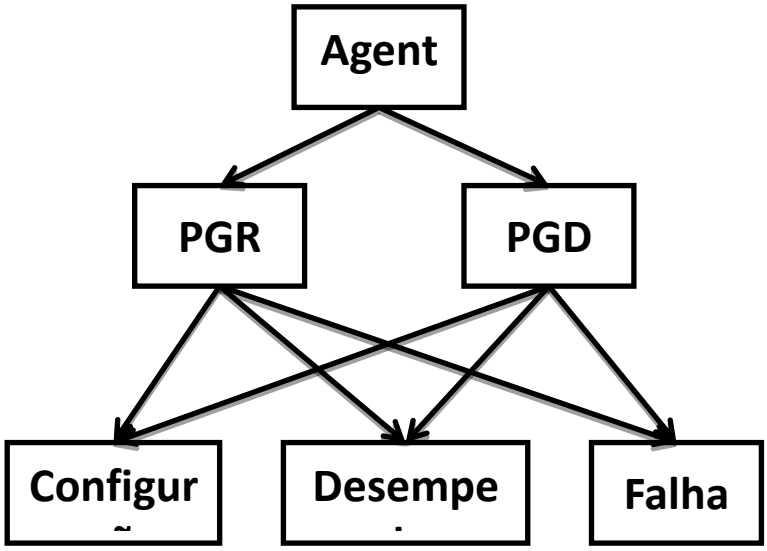


Figura 21 - Planos e funcionalidades

O objetivo deste diagrama é definir as funcionalidades de gerência existentes no Agente *Proxy* e os planos a que cada uma delas pertence. As funcionalidades de Configuração, Desempenho e Falhas estão presentes tanto no Plano de Gerência de RSSF como no Plano de Gerência de Dados, sendo formalmente diferenciadas através da estrutura de *MIB* criada para esta proposta. As relações existentes entre os planos e as funcionalidades são mapeadas somente no agente *Proxy*, não havendo implementação destes conceitos diretamente nos sensores. Sendo o Agente *Proxy* uma entidade centralizadora, hierarquicamente superior, os mecanismos de gerência nos dois planos ficam uniformizados.

7 Metodologia

Levando-se em consideração o caráter prático escolhido para a abordagem do objeto pesquisado, neste caso a caracterização de um sistema real de gerenciamento de elementos de rede sem fio com agentes do tipo *Proxy*, trabalharemos com protótipos didáticos e desenvolvimento de procedimentos de testes.

Utilizando a documentação de fabricantes e gerando protótipos baseados em micro controladores, *software* embarcado e rádio transceptores, chegaremos a uma arquitetura inicial, capaz de sintetizar os pontos mais importantes de nossa pesquisa, em busca de agentes *SNMP* embarcados capazes de enviar os dados necessários a gerência da rede.

Buscando a simplificação dos protótipos e buscando um desenvolvimento mais ágil e financeiramente viável, optamos por soluções *open source*, tanto de *software* como de *hardware* abertos, presentes no mercado, baseados em micro controladores Atmel AVR 8 bits, de baixo custo e linguagem de programação C/C++, com ambiente de desenvolvimento simplificado e utilização de *API*'s também *open source*. O kit de desenvolvimento utilizado é composto pela *IDE* do projeto *open source* Arduino (ARDUINO, 2011) e pela *API* do projeto Radiuino (RADIUINO, 2011). Nosso universo de possibilidades abrange interfaces Ethernet, USB e serial e serão exploradas em sua totalidade em nossos experimentos.

A abordagem que utilizaremos para o protótipo inicial é a de sensores que respondem a estímulos externos, em uma rede simples, puramente de sensores. Essa rede será composta de sensores e uma base ou concentrador, com protocolo de comunicação por nós definida, criando assim uma disposição do tipo ponto multiponto (Figura 22). Esta disposição ponto multiponto se justifica pelo fato de que o frame de comunicação entre os nós da RSSF não possui implementação para roteamanto e múltiplos saltos. Em seguida será realizada a integração com o elemento Agente *Proxy*, que fará o papel do tradutor da pilha de protocolo *SNMP* para o protocolo de comunicação que criamos para a nossa rede

sem fio (Figura 23). Este protótipo final será utilizado nos experimentos de validação.

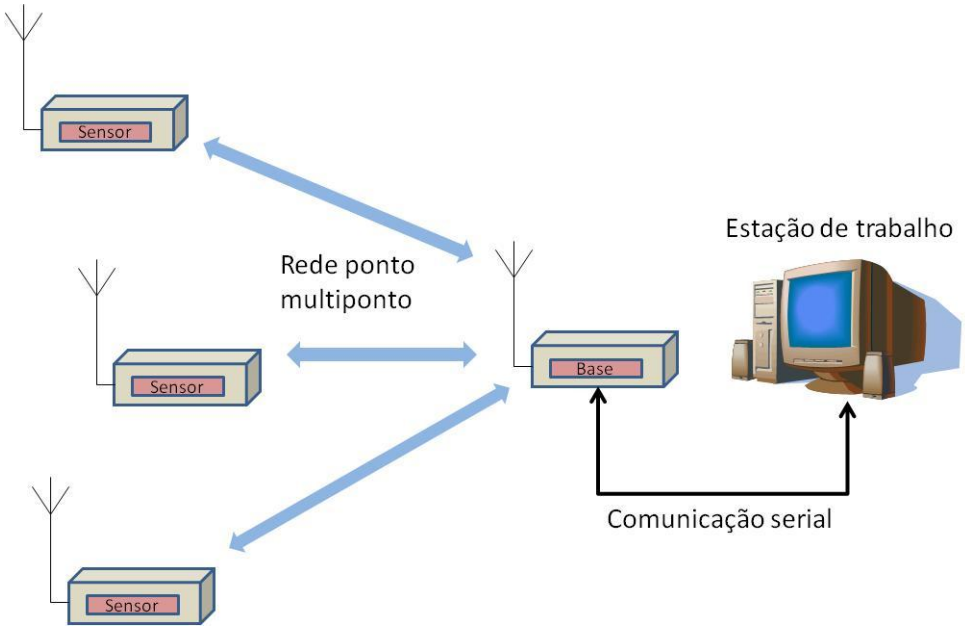


Figura 22 - Protótipo inicial com comunicação serial.

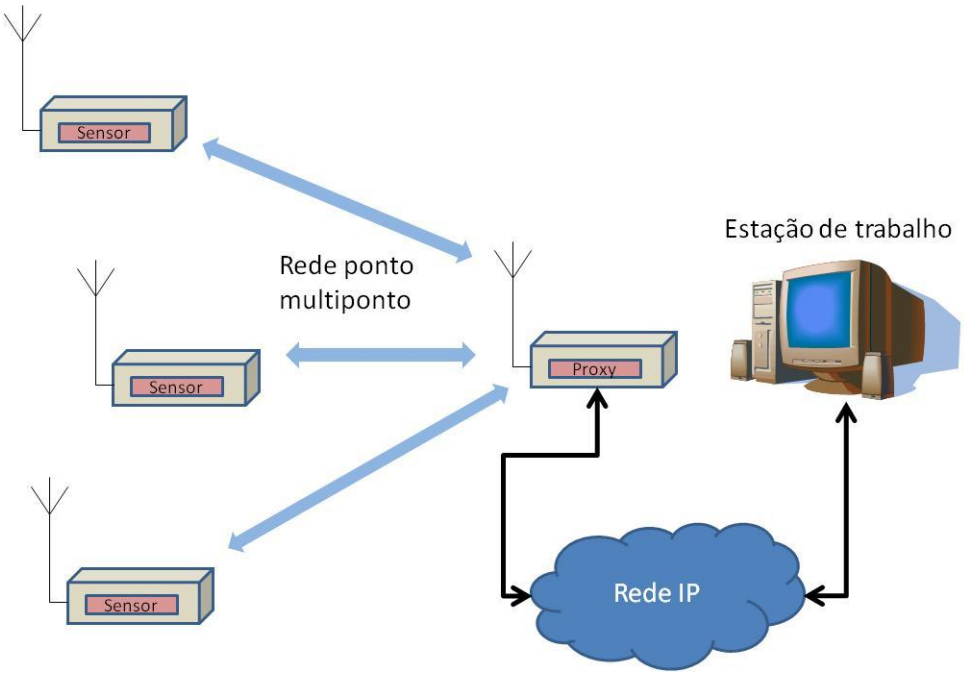


Figura 23 - Rede RSSF integrada a uma rede de gerência SNMP.

Com esta infraestrutura física serão realizados os testes para validar o principio de gerência de RSSF através de um elemento Agente *Proxy*, utilizando o protocolo *SNMP*. A gerência proposta subdivide as funcionalidades em dois planos distintos, e por isso, os testes serão realizados para cobrir os dois casos.

7.1 RSSF

A RSSF é formada por sensores, construídos a partir de plataformas de desenvolvimento de *hardware* e *software*, e um frame de comunicação entre os sensores, com o comprimento de 52 bytes, como na Figura 24. Neste frame é possível inserir as grandezas utilizadas nos experimentos a partir dos sensores e recuperar as informações no Agente *Proxy*. Com Este mesmo frame é possível enviar dados a partir do Agente *Proxy* na direção dos sensores.

O frame projetado para a rede RSSF possui 52 bytes de comprimento, satisfazendo a necessidade de propagação dos dados dos sensores e também para caber no buffer de transmissão dos módulos de rádio dos sensores, sem a necessidade de fragmentação em diversas partes.

Phy				MAC				Net				Transp			
RSSI_DLINK	LQI_DLINK	RSSI_ULINK	LQI_ULINK	TBD	TBD	TBD	TBD	DST_ID	DST_NID	SRC_ID	SRC_NID	COUNT	TBD	TBD	TBD
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

AD0			AD1			AD2			AD3			AD4			AD5		
AD0_W	AD0_H	AD0_L	AD1_W	AD1_H	AD1_L	AD2_W	AD2_H	AD2_L	AD3_W	AD3_H	AD3_L	AD4_W	AD4_H	AD4_L	AD5_W	AD5_H	AD5_L
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33

IO0			IO1			IO2			IO3			IO4			IO5		
IO0_W	IO0_H	IO0_L	IO1_W	IO1_H	IO1_L	IO2_W	IO2_H	IO2_L	IO3_W	IO3_H	IO3_L	IO4_W	IO4_H	IO4_L	IO5_W	IO5_H	IO5_L
34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51

Figura 24 - Frame de comunicação RSSF.

O Frame da RSSF, projetado a partir da plataforma RADIUINO, possui os seguintes campos:

- Phy – Conjunto de 4 bytes, correspondentes a camada física da RSSF, onde se encontram as informações de potência de recepção do sinal RSSI_DLINK e RSSI_ULINK e os indicadores de qualidade de sinal LQI_DLINK e LQI_ULINK.

- MAC – Conjunto de 4 bytes, correspondentes a camada de controle de acesso ao meio, onde atualmente não possui funcionalidades definidas mas que poderá ser utilizada pelo RADIUINO na implementação de políticas de acesso ao meio.
- Net – Conjunto de 4 bytes, correspondestes a camada de rede da RSSF, onde se localizam o endereço de destino para envio do frame e endereço de origem, ou de onde o frame partiu.
- Transp – Conjunto de 4 bytes, correspondentes a camada de transporte da RSSF, onde está disponível um contador de pacotes com um byte de comprimento e mais 3 bytes de uso geral.
- Aplicação:
    - AD0 a AD5 – Conjunto de 18 bytes, correspondentes a camada de aplicação da RSSF, onde estão disponíveis os bytes inferior (ADx_L) e superior (ADx_H) dos conversores analógico-digitais, no total de 6. Esta disponível também, para cada conversos, um byte de configuração (ADx_W).
    - IO0 a IO5 – Conjunto de 18 bytes, correspondentes a camada de aplicação da RSSF, onde estão disponíveis os bytes inferior (IOx_L) e superior (IOx_H) das entradas e saídas digitais. Além destes bytes há um terceiro disponível para usos gerais (IOx_W).

Os sensores foram construídos com microcontroladores Atmel, família AVR de microcontroladores de 8 bits, modelo ATMEGA328, integrado na plataforma de desenvolvimento Arduino. A seguir, as características principais desta plataforma (ARDUINO, 2011):

- Microcontrolador: ATMEGA328
- Tensão de operação: 5V
- Tensão de alimentação: 7V – 12V (recomendado), 6V – 20V (limite)
- Entrada / Saída digitais: 14 pinos (sendo 6 pinos disponíveis para *PWM*)
- Entradas analógicas: 6 pinos
- Memória *Flash*: 32KB, com 512 Bytes para *bootloader*
- *SRAM*: 2KB
- *EEPROM*: 1KB

- *Clock*: 16MHz

Acoplado ao microcontrolador existe um módulo de rádio que opera na faixa de 915MHz, fabricado pela Texas Instruments, modelo CC1101 (SWRS061G, 2012). As características principais deste circuito integrado são:

- Alta sensibilidade: -112 dBm em 1,2 kBaud, 868 MHz, PER de 1%.
- Baixo consumo: 14,7 mA em RX, 1,2 kBaud, 868 MHz.
- Potência de saída: até 12 dBm para todas as frequências suportadas
- Taxa de transmissão programável: 0,6-600 kbps.
- Bandas de freqüência: 300-348 MHz, 387-464 MHz e 779-928 MHz.
- Modulação: 2-FSK, 4-FSK, GFSK, MSK, OOK e ASK.
- Rápido tempo de inicialização: 240 ms do sono para RX ou modo de TX.

Além destas características o CC1101 exige uma quantidade reduzida de componentes externos, não havendo a necessidade de implementação de filtros externos ao circuito integrado. A configuração utilizada para o CC1101, tanto nos sensores quanto no Agente *Proxy*, foi:

- Potência de saída: +10 dBm
- Modulação: GFSK
- Canal: Independente por rádio (0 – Proxy, 1 – Sensor 1, 2 – Sensor 2, etc.)

O ATMEGA328 possui seis conversores A/D e pode-se observar que o frame de comunicação da RSSF contempla o transporte de todos eles, subdividindo-os em byte superior (H) e byte inferior (L). Neste mesmo frame estão disponíveis informações sobre camada física (Phy), MAC, camada de rede (Net) e camada de transporte (Transp). Estes dados estão disponíveis para utilização, mas não serão utilizados nesta proposta.

Exemplos de sensores da RSSF são mostrados na Figura 25. Com estes sensores podem-se realizar experimentos nos planos de RSSF e de dados, dependendo da programação e dos transdutores utilizados. Estes sensores estão também equipados com transdutores de temperatura, modelo LM35.

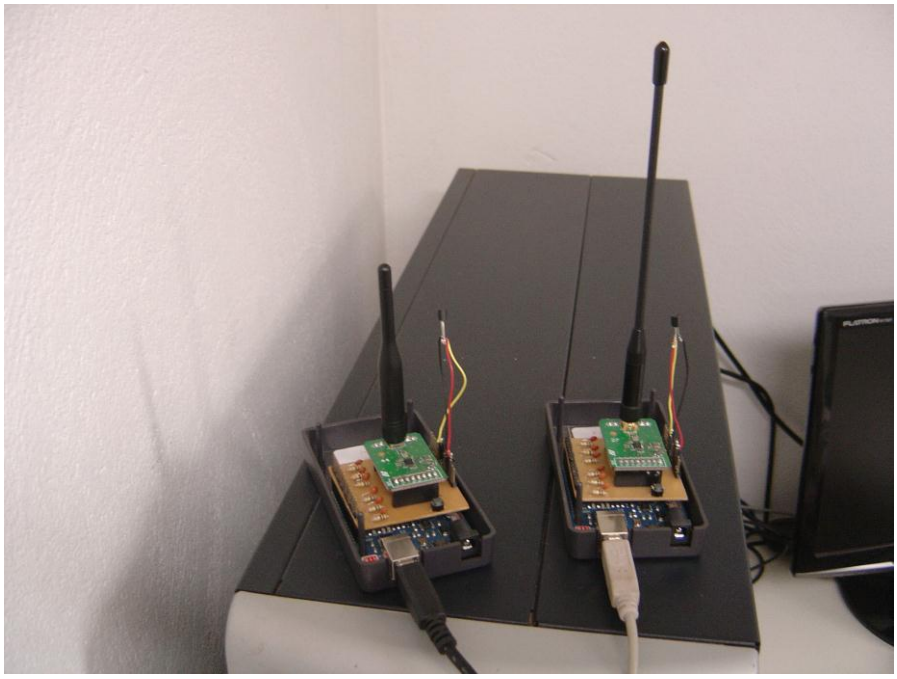


Figura 25 - Sensores da RSSF.

Utilizando o mesmo firmware de programação é possível implementar o sensor em hardware mais elaborado, com uma variedade maior de transdutores disponíveis. Na Figura 26 a seguir tem-se um sensor sem fio, já com transdutores de intensidade luminosa e temperatura, controle por chaveamento, pronto para receber o firmware básico desenvolvido para os sensores sem fio.

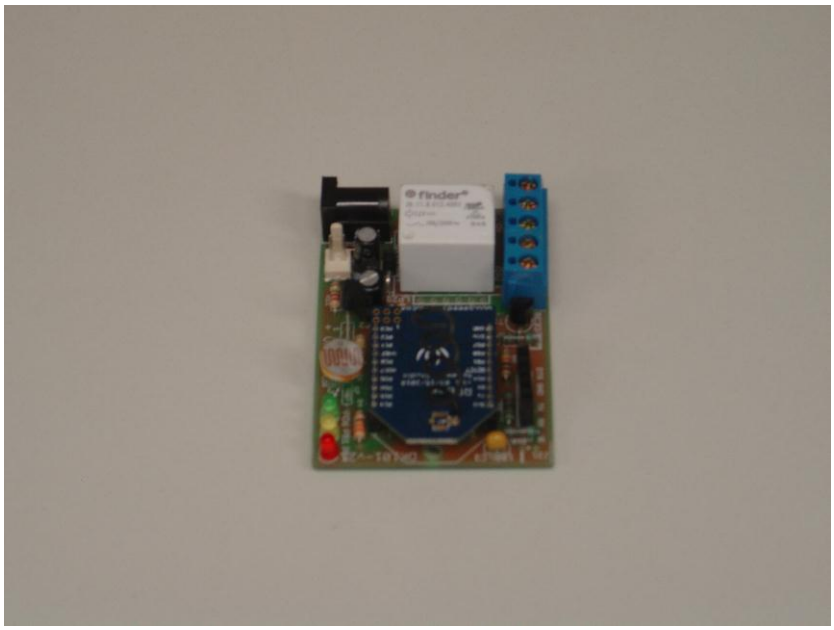


Figura 26 - Sensor sem fio com transdutores internos.

7.2 *Agente Proxy*

O módulo Agente *Proxy* foi construído seguindo o princípio de engenharia da *MIB*, no que diz respeito ao seu *software*, com similaridades com os sensores, pois possui um módulo de rádio similar aos sensores, possuindo apenas um *firmware* diferente. No caso do Agente *Proxy*, a separação é inteiramente física, estando os dois módulos interconectados via TX/RX disponíveis, como na Figura 27.

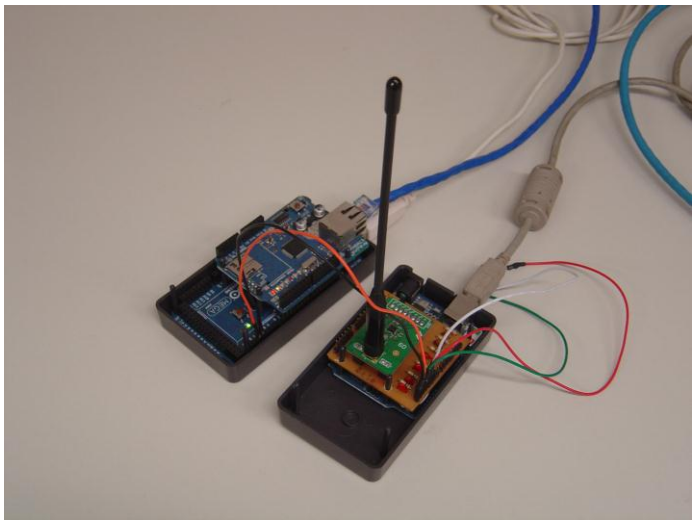


Figura 27 - Configuração de hardware do Agente Proxy.

Na Figura 28 tem-se o módulo de rádio do Agente *Proxy*, onde verificamos as similaridades com os sensores. Tanto o microcontrolador como o rádio acoplado é do mesmo fabricante e modelo.

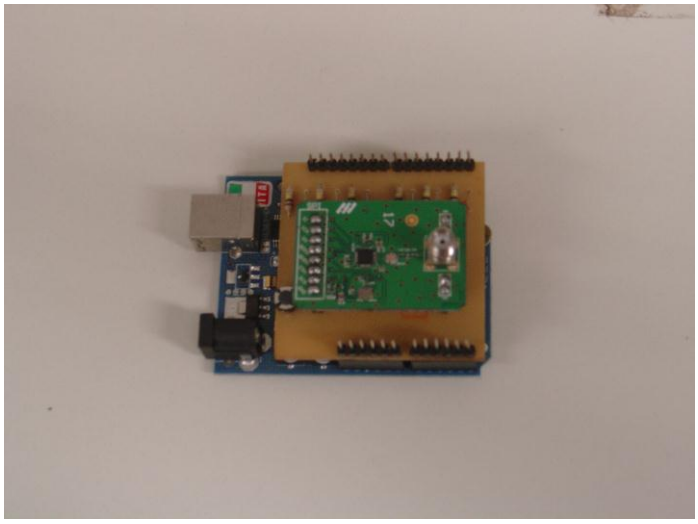


Figura 28 - Módulo de rádio do Agente Proxy.

A diferença marcante fica por conta do módulo que opera como agente *SNMP* da RSSF. Neste caso optou-se por um *hardware* com mais recursos, possuindo mais memória e opções de entrada e saída. Em nosso caso, a interconexão entre os módulos é feita a partir de uma porta serial TX/RX extra presente no módulo agente, representado na Figura 29. Para a construção do Agente Proxy foi utilizado também o kit Arduino, diferenciando apenas pelo modelo do microcontrolador utilizado, um ATMEGA2560.

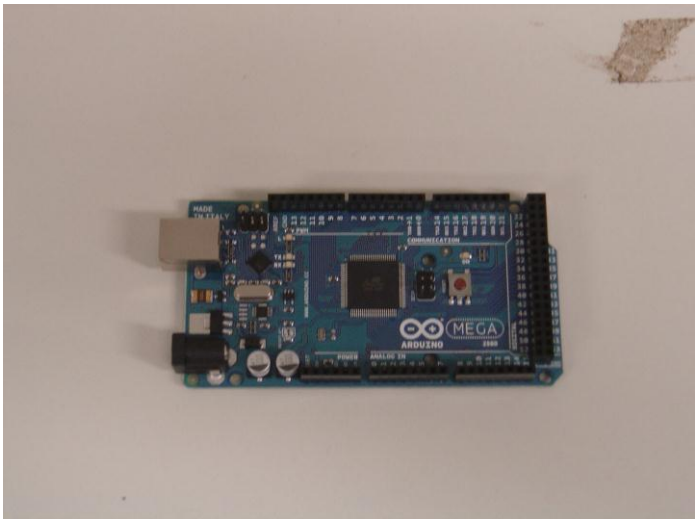


Figura 29 - Módulo de agente Proxy.

A seguir os dados principais da plataforma composta pelo ATMEGA2560 (ARDUINO, 2011):

- Microcontrolador: ATMEGA2560
- Tensão de operação: 5V
- Tensão de alimentação: 7V – 12V (recomendado), 6V – 20V (limite)
- Entrada / Saída digital: 54 pinos (sendo 14 pinos disponíveis para *PWM*)
- Entradas analógicas: 16 pinos
- Memória Flash: 256KB, com 8KB utilizados para *bootloader*
- SRAM: 8KB
- EEPROM: 4KB
- Clock: 16MHz

Na Figura 30 tem-se uma disposição dos sensores e do Agente Proxy para uma comparação de tamanho entre estes elementos.

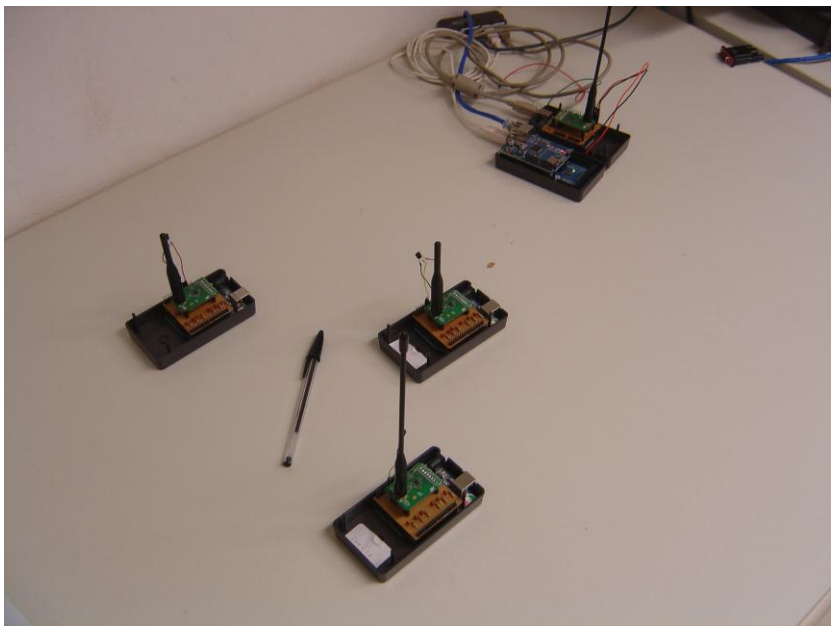


Figura 30 - Sensores e Agente Proxy, comparativamente.

7.3 Software de Gerência

A estação de gerência deve possuir, na camada de aplicação, um recurso capaz de através do conhecimento do protocolo de comunicação, interpretar o processo monitorado. Neste caso, a estação de gerência pode ser construída utilizando-se um aplicativo dedicado a monitoração e atuação em processos do tipo SCADA (*Supervisory Control and Data Acquisition*), como por exemplo o ScadaBR (SCADA, 2011), ou então se utilizando um *MIB browser* capaz de enviar comandos ao agente e receber *TRAPS* de eventos, como pro exemplo o *iReasoning MIB browser* (IREASONING, 2011). A escolha por um software de gerência pode ser guiada pelo objetivo que se determina para a gerência. Aplicativos, como é o caso do *MIB Browser,* são bastante utilizados para a monitoração de dados de gerência de rede e investigação. O ScadaBR é mais indicado para a monitoração de processos, por isso, em nosso caso, seria indicado como um candidato a gerência de dados da rede. Estes dois ambientes podem coexistir na mesma rede ou mesmo na mesma estação de gerência. Mas, como o objetivo da engenharia de *MIB* para a RSSF foi de separar dois planos distintos de gerência, é interessante que usuários tenham segregados fisicamente os recursos de gerência disponíveis.

Na Figura 31 tem-se a configuração física da estação de gerência, capaz de executar tanto aplicativos dedicados para a monitoração de processos como *MIB browsers*.

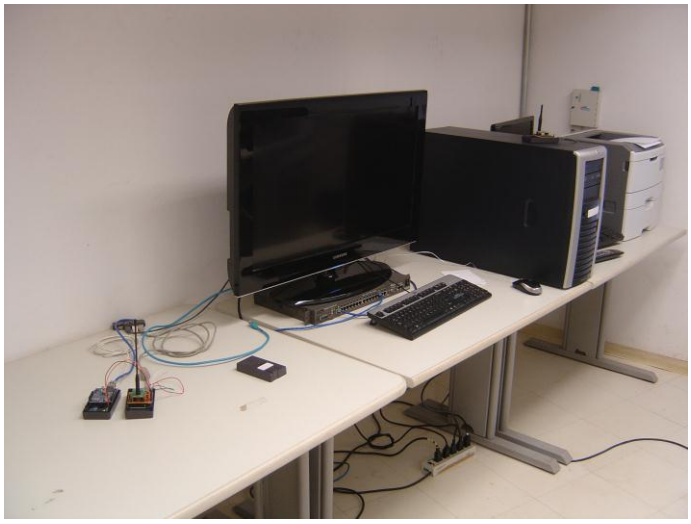


Figura 31 - Estação de gerência.

Na Figura 32 é possível ver uma típica tela de *MIB browser*, onde os dados principais são inseridos, como o endereço IP do agente, a *OID* que se pretende buscar dados, uma tela de resultados e uma hierarquia de *OID*. Na Figura 33 pode-se observar a monitoração sendo feita através do relacionamento, feito pelo software ScadaBR, entre medidas e posicionamentos geográficos, úteis na monitoração de processos.

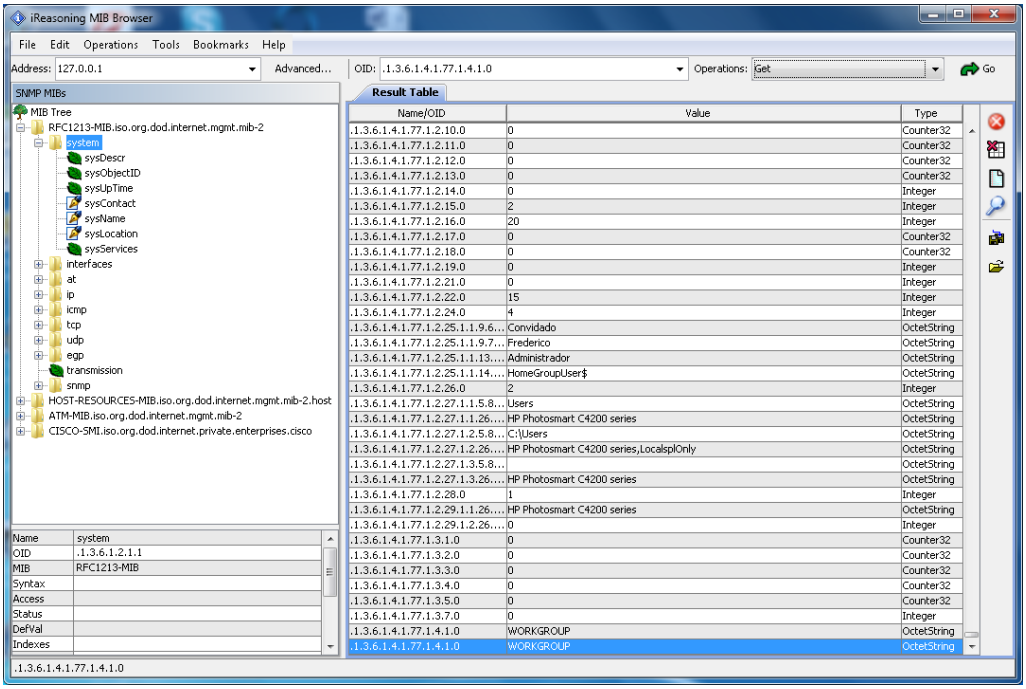


Figura 32 - Tela típica de um MIB browser.

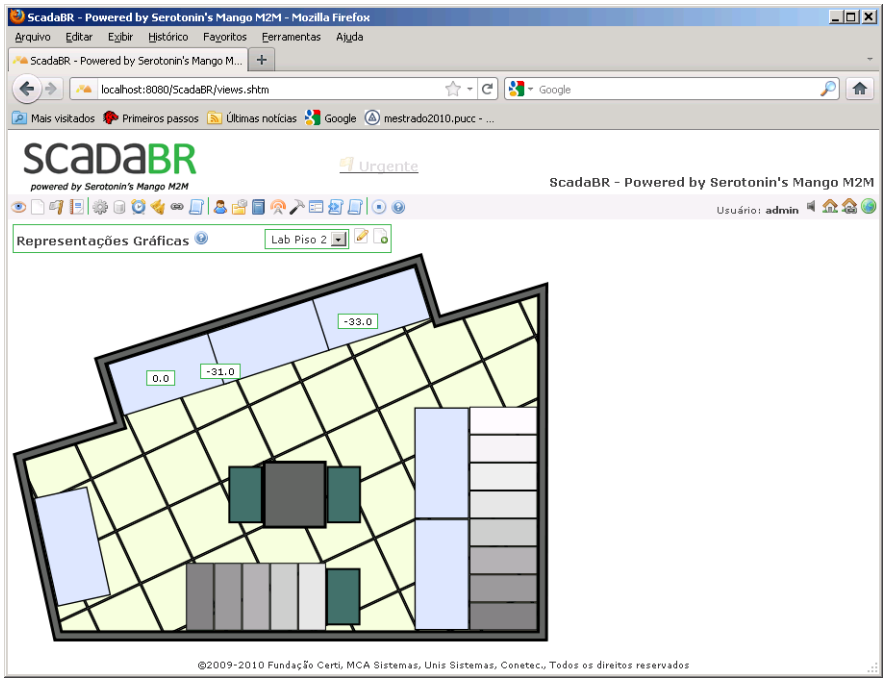


Figura 33 - Exemplo de gerência via software ScadaBR.

7.4 Parâmetros de Gerência

Para nossos testes, no plano de gerência de RSSF utilizaremos a medida de *RSSI*, sua média e desvio padrão, para avaliações quanto ao meio de transmissão e determinadas condições de uso dos sensores e utilizaremos os limiares de *RSSI* para a verificação da resposta do agente a correlações. Estes parâmetros foram escolhidos por estarem relacionados diretamente com a avaliação da propagação de rádio e por serem as grandezas melhor mensuradas pelo protótipo.

Para o plano de gerência de dados foi escolhida a temperatura por ser o protótipo de mais simples implementação e por ter já sido realizada a engenharia para esta grandeza.

7.5 Metodologia de Teste

O planejamento dos testes a serem realizados visa à busca por evidências sobre a validade da proposta, demonstrando a aplicabilidade em sistemas reais. Por tanto, serão realizados experimentos com o protótipo desenvolvido em laboratório com o intuito de simular situações reais de operação dos sensores em interação com um Agente *Proxy*. Visando o acordo com o princípio de gerência subdividido em planos de gerência, os testes devem ser definidos de acordo em este princípio. Sendo assim, os testes serão divididos em aplicabilidade a cada plano de gerência, além de um teste definido para a aplicação em *survey* de ambientes.

Em situações de operação contínua os sensores devem monitorar a condição de operação dos seus rádios informando para a rede certos parâmetros. A potência de sinal medida na recepção desempenha um papel fundamental na determinação da qualidade do sinal, uma vez que sinal e ruído devem obedecer a uma relação favorável à regeneração dos sinais transportados via rádio, sem distorção ou perdas.

Neste sentido algumas condições de propagação de sinais podem ser avaliadas e a arquitetura de gerência proposta deve se mostrar como uma ferramenta interessante, justamente aos interessados em manter boas condições de utilização de uma RSSF.

Através dos testes serão verificadas as condições de propagação em um meio, verificando seu comportamento e enumerando possíveis ações a serem realizadas de modo a configurar da melhor forma a rede. O *survey*, ou vistoria, via medições de *RSSI* através da gerência de RSSF é aplicado neste caso para a caracterização da propagação. A medição de *RSSI* de sensores a diversas alturas do solo, ou de um pavimento, visa a avaliação dos diversos casos relacionados com a instalação dos sensores em campo e as características de propagação nestes casos, verificando as melhores condições e identificando configurações otimizadas. Correlacionando estes dados espera-se que o agente *Proxy* realmente aja criando eventos relacionados com as causas, avisando a estação de gerência sobre possíveis problemas.

Um teste específico para a medição da influência dos danos causados a antena dos sensores ou do Agente *Proxy* sobre a *PER* também foi definido, verificando-se o comportamento através da gerência de RSSF.

Com relação ao plano de dados serão feitas diversas leituras em um ambiente fechado com temperatura controlada e arrefecimento forçado, verificando a média final.

8 Resultados de Testes

A seguir são apresentados alguns resultados de testes realizados, com o objetivo de validar os conceitos propostos, utilizando o Agente *Proxy* desenvolvido a partir dos conceitos discutidos durante a engenharia de *MIB*. Os testes foram separados de acordo com o foco nos planos de gerência, RSSF ou dados. Desta forma é possível demonstrar que o funcionamento da gerência pode ser tratado de forma separada por planos, dependendo do foco dos usuários, e ao mesmo tempo validar a proposta de gerência de RSSF via *SNMP* para rede e dados.

8.1 Plano de Rede - Teste de SURVEY

No plano de rede foram obtidos resultados experimentais para a validação dos princípios de gerência de rede aplicáveis a RSSF, utilizando a técnica de *survey* em ambiente semi-confinado com poucos obstáculos. O teste de *survey* visa a caracterização de um ambiente, realizando um estudo do perfil de propagação por este ambiente, em diversos pontos de medida, medindo o desempenho do sensor em cada caso. Com o resultado do *survey*, pode-se verificar a aplicabilidade de algum modelo matemático. O teste em ambiente densamente povoado, com sensores dispostos a diversas alturas do pavimento visa medir o desempenho dos sensores em diversas condições de instalação.

Para o experimento de *survey* considerou-se um espaço semi-confinado como o da Figura 34. Este espaço é um corredor externo a uma casa, ao ar livre, sendo limitado por três paredes até uma distância de 15 metros e possui uma descontinuidade de aproximadamente 4 metros ao final do corredor, onde há o contorno da casa.

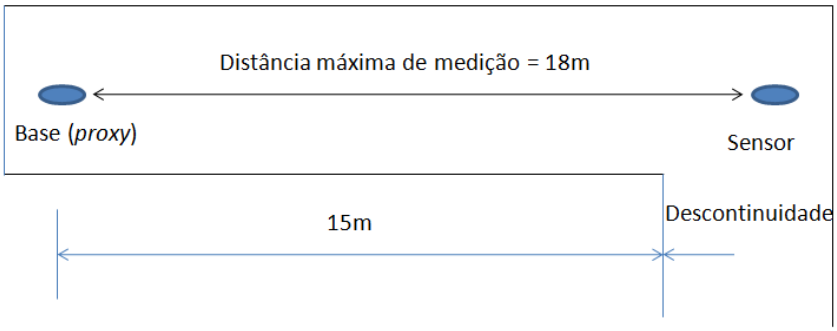


Figura 34 - Espaço físico do experimento de *survey*.

Foram realizadas 30 medidas de *RSSI* para *uplink* e mais 30 medidas de *RSSI* para *downlink* a cada 1 metro, no total de 18 metros, sendo o primeiro metro considerado como a medida de referência de *RSSI* no modelo *Log-Distance*, aplicável neste caso. Sensor e Agente *Proxy* estavam a 50 cm do solo. Tanto o sensor quanto o agente operavam a uma potência de sinal de +10 dBm e com antenas de ganho 0 dBi.

O comportamento da intensidade de sinal nas diversas distâncias pode ser verificado na Figura 35, onde cada um dos pontos de *uplink* e *downlink* representam a média das suas respectivas 30 medidas. Além dos valores médios de *RSSI* para cada distância, foi calculado também o beta para cada um dos pontos e um beta médio que caracteriza o ambiente do experimento. Os valores de beta para cada uma das distancias estão mostrados na Figura 36. O beta médio foi calculado a partir dos diversos valores de Beta em cada um dos pontos de medida. Na Figura 35 podemos verificar que, a partir do valor médio de beta, é possível obter um perfil de propagação aproximado para os diversos pontos de medida do experimento. Observa-se também que a maioria dos pontos se encontra dentro do intervalo de média de beta, mais ou menos três desvios padrão, ou intervalo de confiança de 99%, considerando que o universo de medidas do experimento esteja em uma distribuição normal, de acordo com o Teorema do Limite Central.

No primeiro ponto de medida, o mais próximo do agente *Proxy* (1 metro), foram obtidas medições de *RSSI* de *uplink* e *downlink* com potência abaixo do esperado. Devido a proximidade dos transceptores e a potência

utilizada, houve saturação dos receptores, registrando assim valores inexatos de potência de recepção (Figura 35).

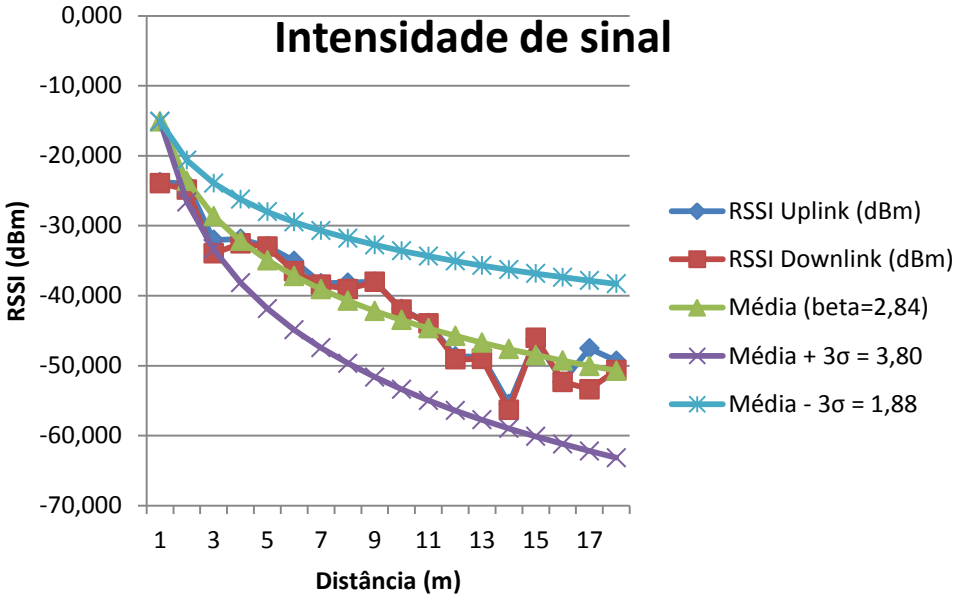


Figura 35 - Comportamento em local semi-confinado.

Uma vez que as potências de uplink e downlink se mostraram muito próximas durante grande parte do experimento, foi utilizado apenas um dado para o cálculo, no caso o uplink. Em situações em que as potências de uplink e downlink tenham diferenças, o valor de beta para um mesmo ambiente não será único.

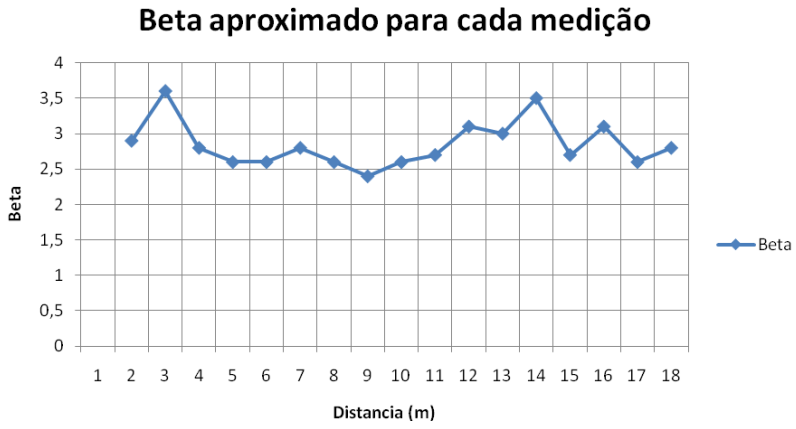


Figura 36 - Comportamento do fator Beta no ambiente do experimento.

8.2 Plano de rede - Monitoração de propagação

Outro experimento realizado, em laboratório, foi a medição de *RSSI* em ambiente com intensa movimentação de pessoas (Figura 33), com sensores em diversas alturas em relação ao pavimento superior (0, 52, 97 e 140 cm), além de uma medição com relação a um sensor um andar abaixo, localizado a 0 cm do pavimento inferior. Os sensores foram programados para transmitir a +10 dBm. O objetivo deste teste é de averiguar o nível de atenuação que obstáculos em movimento causam na propagação de sinais nas condições de operação, medindo o desempenho do sensor. Estes dados são importantes na caracterização do ambiente, verificando se existem condições de operação da gerência de RSSF em cenários de alta complexidade de propagação.

Neste experimento foi considerado o sensor a uma distância fixa de 4,60 m, porém a distâncias diferentes do solo e com movimentação intensa de pessoas, executando trabalhos diversos no laboratório, não relacionados com o experimento, e objetos no meio de propagação. O Agente *Proxy* estava localizado em um local fixo a 74 cm do pavimento superior e foram coletadas 100 amostras de *RSSI* para cada caso. Obtivemos os seguintes resultados, Figuras 37 a 41, para os diversos casos, utilizando a estação de gerência e a *MIB* de gerência de RSSF para realizar as coletas:

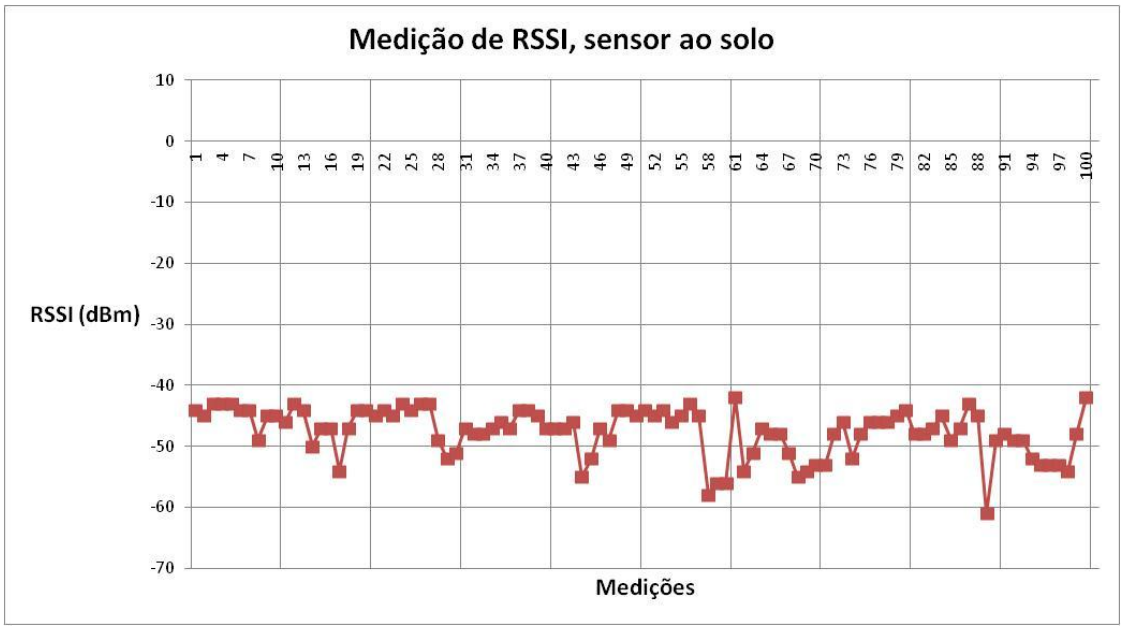


Figura 37 – Medição a altura de 0 cm do pavimento superior.

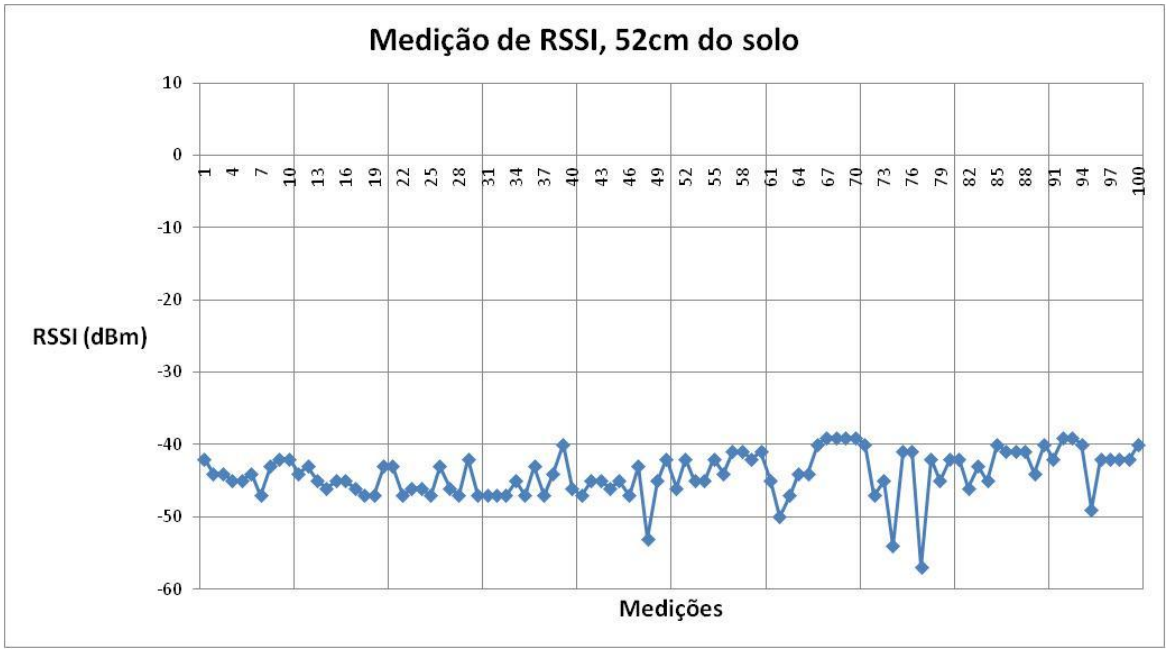


Figura 38 - Medição a altura de 52 cm do pavimento superior.

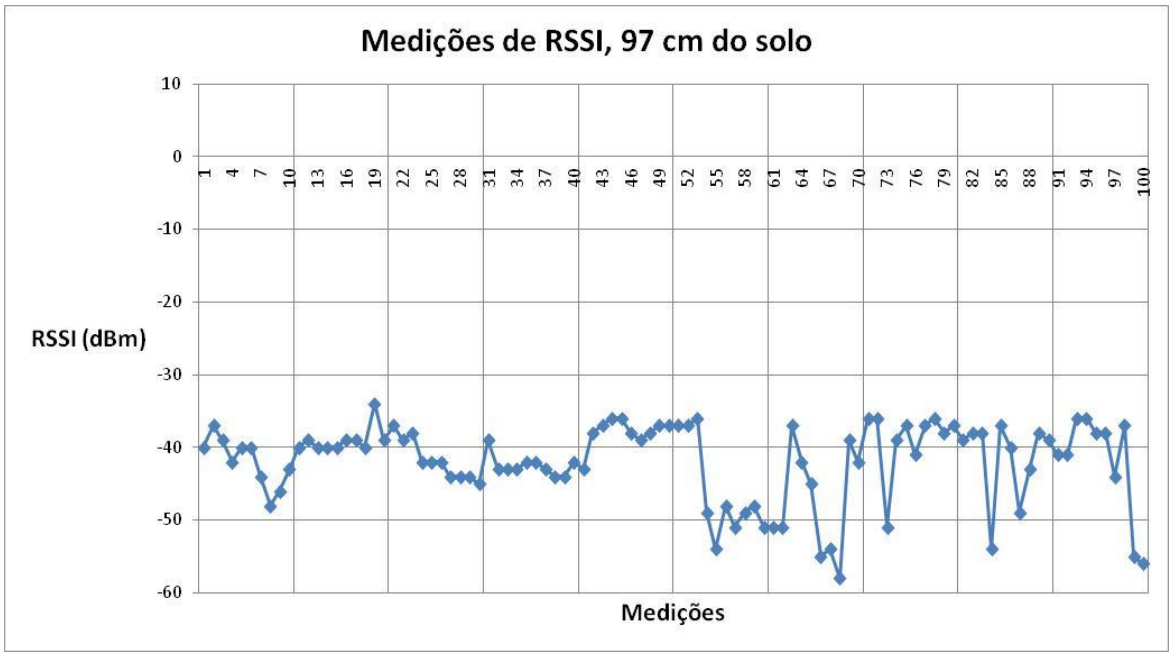


Figura 39 - Medição a altura de 97 cm do pavimento superior.

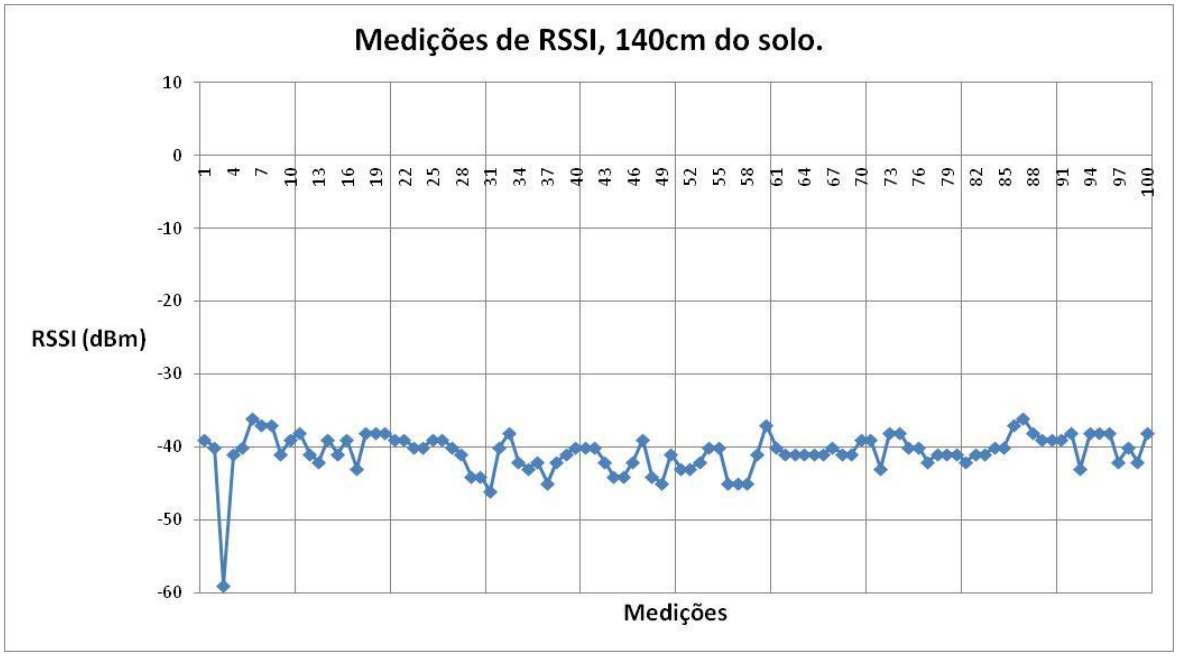


Figura 40 - Medição a altura de 140 cm do pavimento superior.

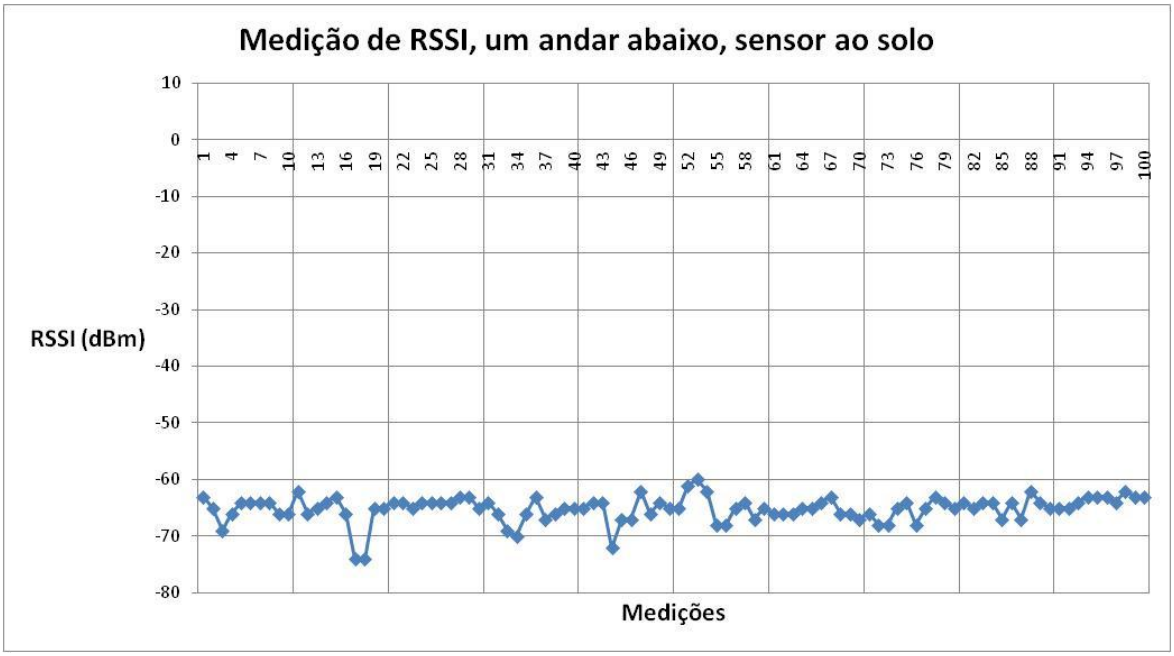


Figura 41 - Medição a altura de 0 cm do pavimento inferior (andar abaixo).

Juntamente com estes dados obtivemos o seguinte resumo estatístico aproximado com relação ao universo de medições de *RSSI* (Tabela 2):

Tabela 2 - Resumo estatístico das medições de RSSI, sensores em diversas alturas.

Posição (cm)	Média (dBm)	Desvio Padrão
0	-46,18	3,94
52	-43,05	3,19
97	-39,72	5,44
140	-40,04	2,85
Andar abaixo	-64,59	2,28

8.3 Plano de Rede - Monitoração de eventos por TRAPS

De modo a se verificar a utilização do Agente *Proxy* como um monitorador de eventos foi criado um experimento para se relacionar um parâmetro pré-configurado com um valor de leitura. Baseado em uma regra de correlação, o Agente *Proxy* envia um *TRAP SNMP* de modo a avisar a uma estação de gerência sobre o evento causador.

Neste experimento foi verificada a possibilidade de se enviar *TRAPS* no caso de a potência de sinal recebida, lida via comando *GET SNMP*, for menor que um valor pré-estabelecido, por exemplo, via um comando *SET SNMP*.

Na Figura 42 tem-se uma sequência de leituras de RSSI em um sensor, ao qual foi imposta uma perda na recepção do sinal, através da remoção da antena. A antena foi recolocada para que a interrupção no envio dos TRAPS fosse verificada. No total foram realizadas seis medidas de potência sem a antena do sensor.

Name/OID	Value	Type	IP:Port
.1.3.6.1.4.1.23955.2.1.1001.2.1001.0	-28	OctetString	192.168.2.6...
.1.3.6.1.4.1.23955.2.1.1001.2.1001.0	-28	OctetString	192.168.2.6...
.1.3.6.1.4.1.23955.2.1.1001.2.1001.0	-29	OctetString	192.168.2.6...
.1.3.6.1.4.1.23955.2.1.1001.2.1001.0	-29	OctetString	192.168.2.6...
.1.3.6.1.4.1.23955.2.1.1001.2.1001.0	-29	OctetString	192.168.2.6...
.1.3.6.1.4.1.23955.2.1.1001.2.1001.0	-28	OctetString	192.168.2.6...
.1.3.6.1.4.1.23955.2.1.1001.2.1001.0	-86	OctetString	192.168.2.6...
.1.3.6.1.4.1.23955.2.1.1001.2.1001.0	-86	OctetString	192.168.2.6...
.1.3.6.1.4.1.23955.2.1.1001.2.1001.0	-86	OctetString	192.168.2.6...
.1.3.6.1.4.1.23955.2.1.1001.2.1001.0	-86	OctetString	192.168.2.6...
.1.3.6.1.4.1.23955.2.1.1001.2.1001.0	-86	OctetString	192.168.2.6...
.1.3.6.1.4.1.23955.2.1.1001.2.1001.0	-86	OctetString	192.168.2.6...
.1.3.6.1.4.1.23955.2.1.1001.2.1001.0	-29	OctetString	192.168.2.6...
.1.3.6.1.4.1.23955.2.1.1001.2.1001.0	-29	OctetString	192.168.2.6...
.1.3.6.1.4.1.23955.2.1.1001.2.1001.0	-29	OctetString	192.168.2.6...
.1.3.6.1.4.1.23955.2.1.1001.2.1001.0	-29	OctetString	192.168.2.6...
.1.3.6.1.4.1.23955.2.1.1001.2.1001.0	-29	OctetString	192.168.2.6...

Figura 42 - Captura de medições de RSSI com e sem perdas.

A remoção da antena causou uma diminuição da potência de sinal recebida de -29 dBm para -86 dBm, perfazendo uma perda de 57 dB. Para a pré-configuração foi considerada uma potência de -40 dBm, então para sinais a baixo desta potência é gerado um *TRAP SNMP*. Na Figura 43 têm-se os *TRAP SNMP* desmontados via um analisador de protocolos. O Agente *Proxy* é configurado para o IP 192.168.2.64 e a estação de gerência é configurada com o IP 192.168.2.60.

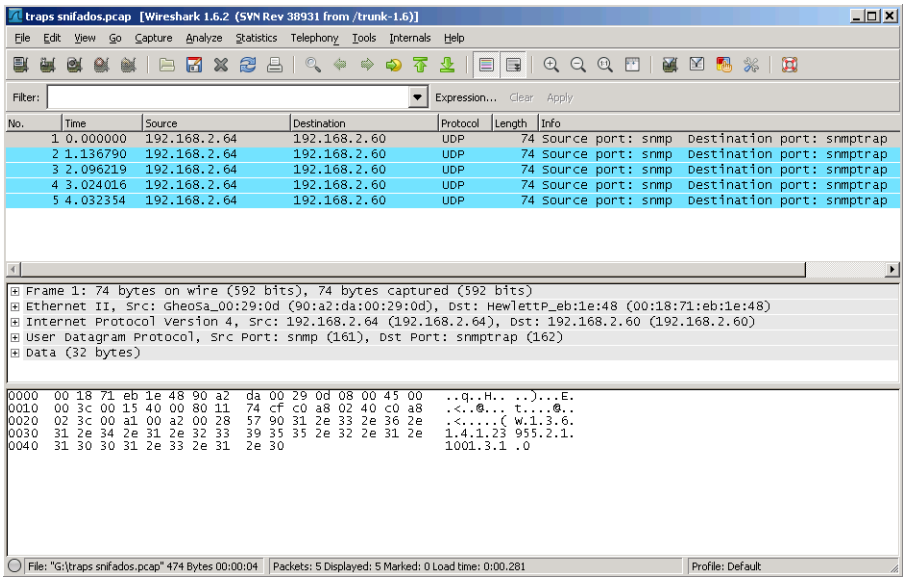


Figura 43 - Analise dos pacotes de TRAP SNMP enviados pelo Agente Proxy.

O *TRAP SNMP* difere do comando *GET* e *SET SNMP* pela porta *UDP*, sendo a primeira a porta 162 e a segunda a porta 161. O analisador já filtra por protocolo e porta, sendo assim está mostrada apenas o fluxo de *TRAPS* para cada requisição de *RSSI* feita ao sensor sem fio.

8.4 Plano de Rede - Influência de danos na antena.

Em situações onde os sensores são colocados ou lançados em lugares inóspitos, podem-se esperar que danos físicos ocorram. As consequências podem ser imediatas ou futuras, como por exemplo, na medição da potência de recepção (*RSSI*). Desta forma, foi monitorada a RSSI, neste caso de *uplink*, verificando-se a existência de perdas de pacote na rede sem fio.

Como cenário de testes foi definido uma construção inicial dos sensores com suas antenas padrão de 174 mm e foram feitas também medições com antenas menores, simulando um dano físico, de 84 mm. Foram realizadas 100 medições consecutivas para cada um dos casos escolhidos. O mesmo cenário de testes foi repetido para condições de propagação diferentes, em nosso caso entre andares distintos do mesmo prédio.

Na Tabela 3 estão os resultados obtidos dos testes. As medidas que causaram erro foram excluídas e a média recalculada com os valores válidos.

Tabela 3 - Testes de medição de *RSSI* com simulação de danos na antena.

Cenário	Medição de RSSI de uplink médio (dBm)	PER(%)
Sensor e Base, antenas intactas	-33,7	0
Sensor com antena danificada	-34,3	1
Sensor e Base com antenas danificadas	-41,1	4
Sensor e Base, antenas intactas, andares distintos	-61,4	4
Sensor com antena danificada, andar distinto	-63,9	15
Base com antena danificada, andar distinto	-64,9	20
Sensor e Base com antenas danificadas, andar distinto	-64	15

8.5 Plano de Dados - Monitoração de ambiente.

Para o plano de dados foi realizado um estudo de conforto térmico em um ambiente de temperatura controlada, realizando a monitoração de temperatura durante um intervalo de tempo de duas horas, aproximadamente, entre os horários 18:12 e 20:39, do dia 10 de Novembro, no laboratório LP-Sira, PUCCAMP. Neste intervalo um total de 13.000 amostras foram registradas, utilizando o agente *Proxy* e a *MIB* do plano de gerência de dados. O resultado das medidas está mostrado na Figura 44.

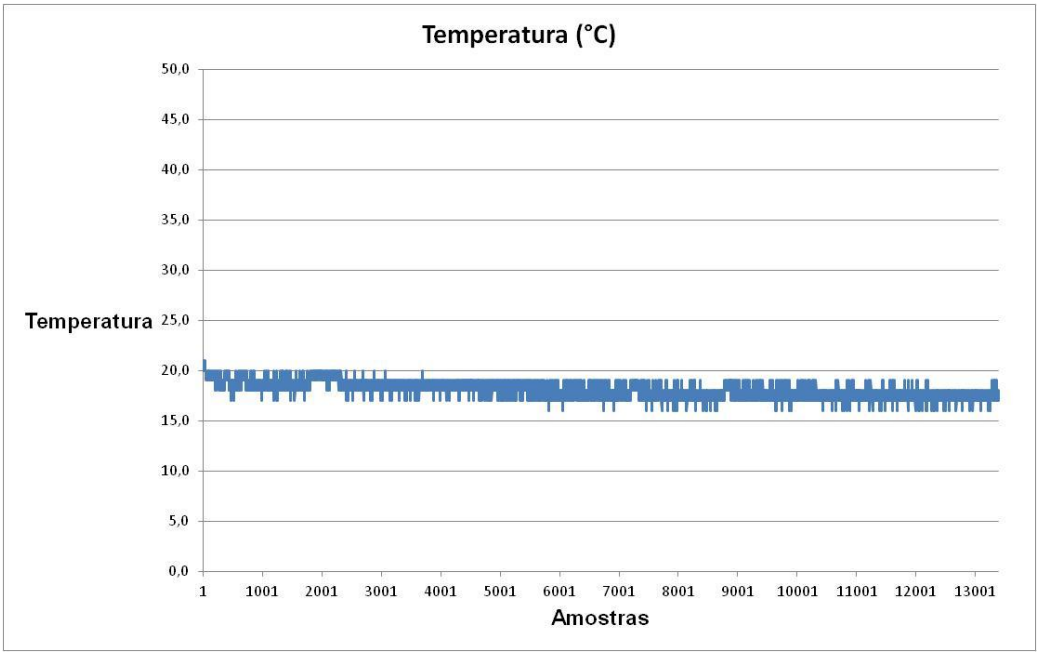


Figura 44 - Perfil de temperatura, ambiente controlado.

Para este teste foi considerado o ajuste, não aferido, do equipamento de ar-condicionado em 17°C e a temperatura medida média por todo o local de medida. Observa-se que a tendência foi de ligeira queda em direção a temperatura ajustada, entre os horários de medida, com o esfriamento do ambiente interior e o pôr do sol.

O valor médio medido para este experimento foi de, aproximadamente, 18°C, com um desvio padrão de 0,0067.

9 Análise e Comparação dos Resultados

9.1 Resultados

A partir dos resultados obtidos dos experimentos utilizando a abordagem de gerência de RSSF, pode-se observar a concordância com os aspectos mais importantes de uma rede gerência via Agente *Proxy*. Com os experimentos de medição de potência de sinal de recepção, ou *RSSI*, este parâmetro importante pôde ser caracterizado utilizando o princípio de gerência via Agente *Proxy*, mantendo a rede de comunicação entre os sensores e o Agente *Proxy* com sua implementação isolada e independente da implementação do agente *Proxy*.

Operacionalmente, a gerência de RSSF se mostra transparente aos administradores e usuários dos recursos da RSSF. Essa é uma vantagem proporcionada pelo uso 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 Agente *Proxy* abstrai todas as especificidades da RSSF e traduz para uma linguagem universal, baseada em identificações e variáveis com visibilidades distintas.

Utilizando o agente *Proxy*, uma estação de gerência tem a visibilidade dos sensores, a partir dos dois planos de gerência distintos, podendo se concentrar na monitoração independente de grandezas vinculadas a um dos planos, como nos experimento de *survey*, medição de desempenho em diversos cenários de instalação de sensores e medição do conforto térmico em ambiente confinado. O mesmo conjunto de *hardware* e *software* pôde ser utilizado em diversos experimentos, com objetivos distintos, coletando dados a partir de diferentes *OID's*.

Alguns aspectos, detectáveis nos experimentos, puderam ser demonstrados. No experimento de *survey*, por exemplo, foi possível detectar a mudança de ambiente a certa distância do Agente *Proxy*, onde houve uma variação de RSSI em relação ao esperado no modelo *Log-Distance* e uma divergência entre os valores de *RSSI* de *uplink* e *downlink*. Esse comportamento

define uma modificação brusca de ambiente, alterando a propagação de sinais (ANDERSEN, 1995).

No teste com sensores a diversas distâncias do solo e em andares distintos observa-se uma influência clara do posicionamento dos sensores e da movimentação de pessoas sobre o desempenho do sensor, estando à mesma distancia do Agente *Proxy* (FANIMOKUM, 2003), além da influência dos diferentes pavimentos onde se encontravam o sensor e a Agente *Proxy*. Comparando os dois casos, verifica-se que as perdas impostas por um pavimento foram maiores que as perdas impostas pelo fluxo de pessoas, combinada com a instalação dos sensores em alturas distintas.

9.2 Comparação

Com relação a comparação com outras arquiteturas propostas na literatura sobre redes de sensores sem fio, a proposta apresentada por este trabalho visa obter uma prova de conceito contra a opinião de que uma gerência de rede de sensores sem fio deve ser galgada em fornecer funcionalidades de gerência entre os sensores, de modo a implementar a auto organização e a auto gestão. Com esse tipo de gerência as grandezas são tratadas internamente na rede de sensores e não possuem uma interface definida claramente com sistemas de gerência 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 e dominado tecnicamente por um vasto público de administradores e usuários de rede, garantindo assim uma melhor aceitação.

9.3 Avaliação da Proposta

Com relação a eficiência da gerência de RSSF baseada em agentes *Proxy*, esta está intimamente ligada com a construção interna do Agente *Proxy*, em seu *hardware* e *software*. Além das limitações de *hardware* impostas pela necessidade de miniaturização dos circuitos existe a limitação de *software*, definida pelas opções da linguagem de programação utilizada e compiladores.

Como existe um número de funcionalidades definidas neste modelo, através da engenharia da *MIB*, a eficiência da proposta pode ser afetada por estes limitantes, uma vez que uma ou outra funcionalidade pode ser excluída de forma a criar mais espaço de execução do *firmware*.

A eficiência da proposta também pode ser afetada pelo projeto dos frames da RSSF, uma vez que deve haver espaço suficiente nos frames para o tráfego das informações de gerência definidas na engenharia da *MIB*, nas duas direções, Agente *Proxy* para sensor e sensor para Agente *Proxy*. Nesta proposta foi criada, por exemplo, uma rede ponto-multi ponto, pois o frame de RSSF não previu a formação de rotas entre vários sensores.

A eficiência da proposta está ligada também com a engenharia da *MIB* desenvolvida. O método utilizado para relacionar as diversas grandezas gerenciadas foi o mesmo, tanto no PGR quanto no PGD, criando uma interface com estações de gerência para os dois planos, diferente de outras propostas onde parâmetros de rede ficam confinados nos algoritmos de auto-organização.

10 Conclusão

A utilização de agentes do tipo *Proxy* na gerência de sensores sem fio se mostrou útil na caracterização de ambientes e como base de conhecimento para alterações nas configurações de gerência, ou as características do ambiente de medidas, de modo a obter resultados satisfatórios e utilizar correlações de resultados para criar eventos indicativos de problemas na rede. Estes são conceitos presentes no arcabouço de gerência de redes de uma forma genérica e fica aqui demonstrado que é possível aplicar estes conceitos em redes de sensores.

O universo de medidas possíveis de serem realizadas não se resume ao conjunto de variáveis aqui descritas. Na tentativa de criar uma prova de conceito em gerência de sensores, um grupo de variáveis foi selecionado para o modelamento da proposta, o que não impede que muitas outras medidas sejam feitas, utilizando as interfaces presentes nos sensores sem fio. As limitações que encontramos na confecção dos sensores, no que diz 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 a gerência de sensores sem fio, como por exemplo sensores capazes de executas diversas *threads*, ou fluxos de execução de código simultaneamente.

A gerência de RSSF, neste trabalho, se baseou na diferenciação de planos de gerência, o que facilitou na engenharia de *MIB* do sistema de gerência. No entanto, podem existir casos em que o conjunto de variáveis presentes em um único plano não basta para indicar um determinado fenômeno, podendo criar oportunidade para trabalhos futuros abordando este tema. A utilização de um protocolo de gerência de redes como o *SNMP* se mostrou neste caso uma alternativa interessante, pois trata os dois planos da mesma forma. O fluxo das informações dos dois planos seguem os mesmos princípios, existindo uma entidade que envia comandos para o agente, este agente responde aos comandos e ao mesmo tempo envia eventos a gerência. A diferenciação é garantida pela engenharia da *MIB*.

Um exemplo seria o desgaste de unidades de bateria, comum em sensores sem fio, catalisado por condições atmosféricas. É conhecido que as variações de temperatura influenciam a autonomia de baterias causando diminuição na sua longevidade (VRUDHULA, 2003). Em termos de RSSF, essa característica é importantíssima e deve ser monitorada. Neste caso têm-se variáveis de planos distintos agindo e determinando uma situação de falha em um futuro não previsto. Assim sendo, regras de correlação envolvendo temperatura, do PGD, e potência de transmissão do rádio do PGR podem ser executadas em um agente *Proxy* para se manter, dentro de parâmetros possíveis, o consumo de energia nos sensores.

Levando-se em consideração a arquitetura utilizada, uma evolução pertinente seria a integração de Agentes *Proxy* em redes *Mesh*, ou totalmente interligada, com algoritmos de roteamento implementados e configuráveis a partir da gerência.

Referências

AKYILDIZ, I. et al, "*Wireless Sensor Networks*", *John Wiley & Sons, 2010.*

ALAM, M. et al, "*WSNMP: a Network Management Protocol for Wireless Sensor Networks*", *Kyung Hee University, Feb. 2008.*

ANDERSEN, J. B. et al, "*Propagation Measurements and Models for Wireless Communications Channels*", *IEEE Communications Magazine, January 1995.*

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

ARDUINO, http://www.arduino.cc, 2011

ATZORI, L. et al, "*The Internet of Things: a survey*", *Elsevier, Computer Networks, 2010.*

BELOSTOTSKY, E. B., "*Secure SNMP*", *Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, February 6, 1997.*

CHAUDHRY, S. A. et al, "*EMP: A Network Management Protocol for IP-Based Wireless Sensor Networks*", *International Conference on Wireless and Ubiquitous Systems, October 2010, Sousse, Tunisia*.

COLITTI, W. et al, "*Integrating Wireless Sensor Networks with the Web*", *The 10th International Conference on Information Processing in Sensor Networks (IPSN), 2011.*

COPPIN, P. R. et al, "*Digital change detection methods in ecosystem monitoring: a review*", *International Journal of Remote Sensing, 2004.*

DEB, B. et al, "*A Topology Discovery Algorithm for Sensor Networks with Applications to Network Management*", *Tech. Rep. DCS-TR-441, Rutgers University, 2001.*

FANIMOKUM, A. et al, "*Effects of Natural Propagation Environments on Wireless Sensor Network Coverage Area*", *Proceedings of the 35th Southeastern Symposium on System Theory, 2003.*

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.*

GOENSE, D. et al, "*Wireless Sensor Networks for Precise Phytophthora Decision Support*", *Agro technology and Food Innovations BV, The Netherlands, 2005.*

HARITSA, J. et al, "*Design of the MANDATE MIB*", *Institute for Systems Research, University of Maryland, College Park, MD, USA, 1993.*

IREASONING, http://ireasoning.com, 2011

ITU Internet Reports*, "The Internet of Things"*, *Executive Summary, International Telecommunication Union, 2005.*

JACQUOT, A et al, "*A New Management Method for Wireless Sensor Networks*", *9th IEEE IFIP Annual Mediterranean Ad Hoc Network Workshop, Juan Les Pins, France, 2010*.

JARA, A. J. et al, "*A Pharmaceutical Intelligent Information System to Detect Allergies and Adverse Drugs Reations Based on Internet of Things*", *Department Information and Communications Engineering, Computer Science Faculty, University of Murcia, Murcia, Spain, 2010.*

KARL, H. et al, "*Protocols and Architectures for Wireless Sensor Networks*", *John Wiley and Sons Ltd, West Sussex, England, 2005.*

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

MAINWARING, A. et al, "*Wireless Sensor Networks for Habitat Monitoring*", *WSNA´02, September 28, 2002, Atlanta, Georgia, USA.*

MARTINEZ, K. et al, "*Environmental Sensor Networks*", *IEEE Computer, 2004.*

PARIDIS, L. et al, "*A Survey of Fault Management in Wireless Sensor Network*", *Journal of Network and Systems Management, Vol. 15, No. 2, June 2007.*

PEIXOTO, J. C., “Contabilização de chamadas em uma infraestrutura de Voz sobre IP (VoIP)”, Laboratório de Voz sobre IP, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil, 2005.

PHAM, P. P. et al, “*Increasing the network performance using multi-path routing mechanism with load balance*”, *Elsevier B. V. Ad Hoc networks 2 (2004), 433-459.*

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

RAPPAPORT, T. S., “*Wireless Communications, Principles and Practice*”, *2nd Edition, Prentice Hall Communications, Engineering and Emerging Technologies Series, 2002.*

RUIZ, L. B. et al, “*MANNA: A Management Architecture for Wireless Sensor Networks*”, *IEEE Communications Magazine, February, 2003.*

RUIZ, L. B. et al, “*On the design of a self-managed wireless sensor network*”, *IEEE Communications Magazine, July 2005.*

RUIZ, L. B. et al, “*Service Management in Wireless Sensors Network*”, *Department of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.*

SCADA, http://www.scadabr.com.br, 2011

SCHIMIDT, C. R. , “Desenvolvimento de uma gerência *SNMP* para dispositivos de redes totalmente ópticas.”, Dissertação, programa de Mestrado Profissional em Gestão de Redes de Telecomunicações, Pontifícia Universidade Católica de Campinas, 2007.

SHÖNWÄLDER, J. , “*Characterization of SNMP MIB Modules*”, *International University Bremen, Bremen, Germany, 2005.*

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

SOHRABI, K. et al, “*Near Ground Wideband Channel Measurement in 800 – 1000 MHz*”, *Electrical Engineering Department, UCLA, Los Angeles, CA, USA, 1999.*

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

STAMATELOPOULOS, F. et al, "*System Security Management via SNMP*", *Network Management and Optimal Design Lab, Department of Electrical and Computer Engineering, National Technical University of Athens, Greece, 1997.*

SUNDMAEKER, H. et al, "*Vision and Challenges for Realizing the Internet of Things*", *Cluster of European Research Projects on the Internet of Things, European Commission – Information Society and Media DG, Brussels, Belgium, 2009.*

SWRS061G, "*CC1101 – Low-Power Sub-1 GHz RF Transceiver*", *Texas Instruments Incorporated, 2012.*

THOMPSON, C. W., "*Smart Devices and Soft Controllers*", *Published by the IEEE Computer Society, IEEE Internet Computing, February 2005.*

TYNAN, R. et al, "*Agents for Wireless Sensor Network Power Management*", *Proceeding of the 2005 International Conference on Parallel Processing Workshops, ICPPW´05, Department of Computer Science, University College Dublin Belfield, Dublin, Ireland, 2005.*

UDUPA, D. K., "*TMN – Telecommunications Management Network*", *McGraw-Hill Telecommunications, 1999.*

VRUDHULA, S. et al, "*Battery Modeling for Energy-Aware System Design*", *IEEE Computer Society, December 2003.*

WALTENEGUS, "*Fundamentals of Wireless Sensor Networks: Theory and Practice*", *John Wiley & Sons, 2010.*

XIAO, Y. et al, "*A survey of key management schemes in wireless sensor networks*", *Department of Computer Science, University of Alabama, Tuscaloosa, AL , USA, May 10, 2007.*

Anexo A – Código base para o agente *Proxy.*

Anexo o código base do agente Proxy. Este código pode ser alterado sem prévio aviso. Com base neste código, agentes customizados podem ser desenvolvidos livremente.

```
#include <Streaming.h>
#include <Ethernet.h>
#include <SPI.h>
#include <MemoryFree.h>
#include <Agentuino.h>
#include <Flash.h>

#define RSSI_RESPONSE 52
#define RSSI_REQUEST52

bytereceiveFromBase[RSSI_RESPONSE]; // Recepcao dos dados da base.
intrssi;// Valor de rssi.
intretSensor;
intvetorCorrec[5] = {74, 74, 74}; //parametros de correção de calibração dos sensores.

// Dados de rede da placa ethernet.
static byte mac[] = {0x90, 0xA2, 0xDA, 0x00, 0x29, 0x0D}; // MAC address do shield
static byte ip[] = {192, 168, 2, 64};                  // IP utilizado pela gerencia
static byte gateway[] = {192, 168, 2, 1};              // Gateway utilizado pela rede de gerencia
static byte subnet[] = {255, 255, 255, 0};             // Mascara da subrede

static char sysDescr[] PROGMEM    = "1.3.6.1.2.1.1.1.0";          // read-only  (DisplayString)
static char sysObjectID[] PROGMEM = "1.3.6.1.2.1.1.2.0";          // read-only  (ObjectIdentifier)
static char sysUpTime[] PROGMEM   = "1.3.6.1.2.1.1.3.0";          // read-only  (TimeTicks)
static char sysContact[] PROGMEM  = "1.3.6.1.2.1.1.4.0";          // read-write (DisplayString)
static char sysName[] PROGMEM     = "1.3.6.1.2.1.1.5.0";          // read-write (DisplayString)
static char sysLocation[] PROGMEM = "1.3.6.1.2.1.1.6.0";          // read-write (DisplayString)
static char sysServices[] PROGMEM = "1.3.6.1.2.1.1.7.0";          // read-only  (Integer)

//sensor 1
//Desempenho
static char rssiUp1[]      PROGMEM = "1.3.6.1.4.1.23955.2.1.1001.2.1001.0";
static char rssiDown1[]    PROGMEM = "1.3.6.1.4.1.23955.2.1.1001.2.1002.0";
static char mediaRssiUp1[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1001.2.1003.0";
static char mediaRssiDw1[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1001.2.1004.0";
static char stdDevRssiU1[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1001.2.1005.0";
static char stdDevRssiD1[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1001.2.1006.0";

//sensor 2
//Desempenho
static char rssiUp2[]      PROGMEM = "1.3.6.1.4.1.23955.2.1.1002.2.1001.0";
static char rssiDown2[]    PROGMEM = "1.3.6.1.4.1.23955.2.1.1002.2.1002.0";
static char mediaRssiUp2[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1002.2.1003.0";
static char mediaRssiDw2[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1002.2.1004.0";
static char stdDevRssiU2[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1002.2.1005.0";
static char stdDevRssiD2[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1002.2.1006.0";

//sensor 3
```

```
//Desempenho
static char rssiUp3[]      PROGMEM = "1.3.6.1.4.1.23955.2.1.1003.2.1001.0";
static char rssiDown3[]    PROGMEM = "1.3.6.1.4.1.23955.2.1.1003.2.1002.0";
static char mediaRssiUp3[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1003.2.1003.0";
static char mediaRssiDw3[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1003.2.1004.0";
static char stdDevRssiU3[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1003.2.1005.0";
static char stdDevRssiD3[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1003.2.1006.0";

// Alguns dados de configuracao, alguns estão como read-only!!!
static char locDescr[]      = "PUCCAMP - Radiuino Agent";  // read-only (static)
static char locObjectID[]   = "1.3.6.1.3.2009.0";          // read-only (static)
static uint32_t locUpTime   = 0;                           // read-only (static)
static char locContact[20]  = "FredericoCyriaco";          // should be stored/read from EEPROM -
read/write (not done for simplicity)
static char locName[20]     = "PUCCAMP";                   // should be stored/read from EEPROM -
read/write (not done for simplicity)
static char locLocation[20] = "Campinas, SP";


// should be stored/read from EEPROM - read/write (not done for simplicity)
static int32_t locServices  = 7;                           // read-only (static)
static char rssiup[5];                                     // rssi da posicao 5 no vetor receiveFromBase
static char rssidown[5];                                   // rssi da posicao 4 no vetor receiveFromBase

uint32_tprevMillis = millis();
charoid[SNMP_MAX_OID_LEN];
SNMP_API_STAT_CODES api_status;
SNMP_ERR_CODES status;

// Paote recebido com o resultado SNMP.
// Desse ponto em diante se desenvolve o agente Proxy!!!
voidpduReceived()
{
  SNMP_PDU pdu;

api_status = Agentuino.requestPdu(&pdu);

if (pdu.type == SNMP_PDU_GET ||
pdu.type == SNMP_PDU_GET_NEXT ||
pdu.type == SNMP_PDU_SET &&
pdu.error == SNMP_ERR_NO_ERROR &&
api_status == SNMP_API_STAT_SUCCESS)
  {
pdu.OID.toString(oid);

if (strcmp_P(oid, rssiUp1) == 0 ||
strcmp_P(oid, rssiUp2) == 0 ||
strcmp_P(oid, rssiUp3) == 0)
   {
if (pdu.type == SNMP_PDU_SET)
     {
pdu.type = SNMP_PDU_RESPONSE;
pdu.error = SNMP_ERR_READ_ONLY;
}
else
{
requestRSSI(oid, 500); // pergunta por informacoes de RSSI do sensor, com timeout.

if (receiveFromBase[2] >= 128)
       {
```

```
rssi = ((receiveFromBase[2] - 256) >> 1) - vetorCorrec[retSensor - 1];
       }
else
       {
rssi = (receiveFromBase[2] >> 1) - vetorCorrec[retSensor - 1];
       }

if (strcmp(rssiup, "ERR") != 0)
       {
itoa((int)rssi, rssiup, 10);
       }

status = pdu.VALUE.encode(SNMP_SYNTAX_OCTETS, rssiup);
pdu.type = SNMP_PDU_RESPONSE;
pdu.error = status;
     }
   }
else if (strcmp_P(oid, rssiDown1) == 0 ||
strcmp_P(oid, rssiDown2) == 0 ||
strcmp_P(oid, rssiDown3) == 0)
   {
if (pdu.type == SNMP_PDU_SET)
     {
pdu.type = SNMP_PDU_RESPONSE;
pdu.error = SNMP_ERR_READ_ONLY;
}
else
{
requestRSSI(oid, 500); // pergunta por informacoes de RSSI do sensor, com timeout.

if (receiveFromBase[0] >= 128)
       {
rssi = (( receiveFromBase[0] - 256) >> 1) - vetorCorrec[retSensor - 1];
       }
else
       {
rssi = (receiveFromBase[0] >> 1) - vetorCorrec[retSensor - 1];
       }

if (strcmp(rssidown, "ERR") != 0)
       {
itoa((int)rssi, rssidown, 10);
       }

status = pdu.VALUE.encode(SNMP_SYNTAX_OCTETS, rssidown);
pdu.type = SNMP_PDU_RESPONSE;
pdu.error = status;
     }
   }
else if (strcmp_P(oid, sysName) == 0)
   {
if (pdu.type == SNMP_PDU_SET)
     {
status = pdu.VALUE.decode(locName, strlen(locName));
pdu.type = SNMP_PDU_RESPONSE;
pdu.error = status;
     }
else
     {
status = pdu.VALUE.encode(SNMP_SYNTAX_OCTETS, locName);
```

```
pdu.type = SNMP_PDU_RESPONSE;
pdu.error = status;
     }
   }
else if (strcmp_P(oid, sysContact) == 0)
   {
if (pdu.type == SNMP_PDU_SET)
     {
status = pdu.VALUE.decode(locContact, strlen(locContact));
pdu.type = SNMP_PDU_RESPONSE;
pdu.error = status;
     }
else
     {
status = pdu.VALUE.encode(SNMP_SYNTAX_OCTETS, locContact);
pdu.type = SNMP_PDU_RESPONSE;
pdu.error = status;
     }
   }
else if (strcmp_P(oid, sysLocation) == 0)
   {
if (pdu.type == SNMP_PDU_SET)
     {
status = pdu.VALUE.decode(locLocation, strlen(locLocation));
pdu.type = SNMP_PDU_RESPONSE;
pdu.error = status;
     }
else
     {
status = pdu.VALUE.encode(SNMP_SYNTAX_OCTETS, locLocation);
pdu.type = SNMP_PDU_RESPONSE;
pdu.error = status;
     }
   }
else if (strcmp_P(oid, sysServices) == 0)
   {
if (pdu.type == SNMP_PDU_SET)
     {
pdu.type = SNMP_PDU_RESPONSE;
pdu.error = SNMP_ERR_READ_ONLY;
     }
else
     {
status = pdu.VALUE.encode(SNMP_SYNTAX_INT, locServices);
pdu.type = SNMP_PDU_RESPONSE;
pdu.error = status;
     }
   }
else
   {
pdu.type = SNMP_PDU_RESPONSE;
pdu.error = SNMP_ERR_NO_SUCH_NAME;
}
Agentuino.responsePdu(&pdu);
}

Agentuino.freePdu(&pdu);
}

void setup()
```

```
{
Serial.begin(9600);
Serial1.begin(9600); // Link do agente com a base RFBee

Ethernet.begin(mac, ip);
api_status = Agentuino.begin();

if (api_status == SNMP_API_STAT_SUCCESS)
  {
Agentuino.onPduReceive(pduReceived);
delay(10);
    Serial <<F("SNMP Agent Initalized...") <<endl;
return;
  }

delay(10);
  Serial <<F("SNMP Agent Initalization Problem...") << status <<endl;
}

void loop()
{
Agentuino.listen();

if (millis() - prevMillis> 1000)
  {
prevMillis += 1000;
locUpTime += 100;
  }
}

voidrequestRSSI(char* pOIDsensor, int timeout)
{
bytebuf[RSSI_REQUEST];
charbufferSensor[5];
charbufferServico[5];

intretServico;

memset(buf, 0x00, sizeof(buf));

memset(bufferSensor,'\0', sizeof(bufferSensor));
memset(bufferServico,'\0', sizeof(bufferServico));

strncpy(bufferSensor, &pOIDsensor[22], 4);
strncpy(bufferServico, &pOIDsensor[29], 4);

retSensor = atoi(bufferSensor) - 1000;
retServico = atoi(bufferServico) - 1000;

buf[8] = (byte)retSensor;
buf[10] = 0;

memset(rssiup,'\0', sizeof(rssiup));
memset(rssidown,'\0', sizeof(rssidown));

Serial1.write((byte *)buf, RSSI_REQUEST);
Serial1.flush();

delay(timeout); // Espera por uma resposta durante 'timeout' milisegundos.
```

```
if(Serial1.available() > 0)
  {
for(inti = 0; i< RSSI_RESPONSE; i++)
    {
receiveFromBase[i] = Serial1.read();
    }
  }
else
  {
memcpy(rssiup, "ERR", 4);
memcpy(rssidown, "ERR", 4);
}
}
```

Anexo B – Código com emissor de TRAPS.

```
#include <Streaming.h>
#include <Ethernet.h>
#include <SPI.h>
#include <MemoryFree.h>
#include <Agentuino.h>
#include <Flash.h>
#include <math.h>

#define RSSI_RESPONSE 52
#define RSSI_REQUEST  52
#define SIZE 10

byte receiveFromBase[RSSI_RESPONSE]; // Recepcao dos dados da base.
int rssi;
float rssi_W_up[3][SIZE];
float rssi_W_dw[3][SIZE];
float media_UP;
float media_DW;
int pow_Up_index;
int pow_Dw_index;
int retSensor;
int vetorCorrec[3] = {74, 74, 74};

static byte mac[] = {0x90, 0xA2, 0xDA, 0x00, 0x29, 0x0D}; // MAC address do shield
static byte ip[] = {192, 168, 2, 64};                  // IP utilizado pela gerencia
static byte gateway[] = {192, 168, 2, 1};              // Gateway utilizado pela rede de gerencia
static byte subnet[] = {255, 255, 255, 0};             // Mascara da subrede

static byte destip[] = {192, 168, 2, 60};

static char sysDescr[]    PROGMEM = "1.3.6.1.2.1.1.1.0";  // read-only  (DisplayString)
static char sysObjectID[] PROGMEM = "1.3.6.1.2.1.1.2.0";  // read-only  (ObjectIdentifier)
static char sysUpTime[]   PROGMEM = "1.3.6.1.2.1.1.3.0";  // read-only  (TimeTicks)
static char sysContact[]  PROGMEM = "1.3.6.1.2.1.1.4.0";  // read-write (DisplayString)
static char sysName[]     PROGMEM = "1.3.6.1.2.1.1.5.0";  // read-write (DisplayString)
static char sysLocation[] PROGMEM = "1.3.6.1.2.1.1.6.0";  // read-write (DisplayString)
static char sysServices[] PROGMEM = "1.3.6.1.2.1.1.7.0";  // read-only  (Integer)

//sensor 1
//Desempenho
static char rssiUp1[]      PROGMEM = "1.3.6.1.4.1.23955.2.1.1001.2.1001.0";
static char rssiDown1[]    PROGMEM = "1.3.6.1.4.1.23955.2.1.1001.2.1002.0";
static char mediaRssiUp1[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1001.2.1003.0";
static char mediaRssiDw1[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1001.2.1004.0";
static char stdDevRssiU1[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1001.2.1005.0";
static char stdDevRssiD1[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1001.2.1006.0";

//sensor 2
//Desempenho
static char rssiUp2[]      PROGMEM = "1.3.6.1.4.1.23955.2.1.1002.2.1001.0";
static char rssiDown2[]    PROGMEM = "1.3.6.1.4.1.23955.2.1.1002.2.1002.0";
static char mediaRssiUp2[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1002.2.1003.0";
static char mediaRssiDw2[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1002.2.1004.0";
static char stdDevRssiU2[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1002.2.1005.0";
static char stdDevRssiD2[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1002.2.1006.0";
```

```
//sensor 3
//Desempenho
static char rssiUp3[]      PROGMEM = "1.3.6.1.4.1.23955.2.1.1003.2.1001.0";
static char rssiDown3[]    PROGMEM = "1.3.6.1.4.1.23955.2.1.1003.2.1002.0";
static char mediaRssiUp3[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1003.2.1003.0";
static char mediaRssiDw3[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1003.2.1004.0";
static char stdDevRssiU3[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1003.2.1005.0";
static char stdDevRssiD3[] PROGMEM = "1.3.6.1.4.1.23955.2.1.1003.2.1006.0";

static char locDescr[]      = "PUCCAMP - Radiuino Agent";  // read-only (static)
static char locObjectID[]   = "1.3.6.1.3.2009.0";          // read-only (static)
static uint32_t locUpTime   = 0;                           // read-only (static)
static char locContact[20]  = "Frederico Cyriaco";         // should be stored/read from EEPROM -
read/write (not done for simplicity)
static char locName[20]     = "PUCCAMP";                   // should be stored/read from EEPROM -
read/write (not done for simplicity)
static char locLocation[20] = "Campinas, SP";

// should be stored/read from EEPROM - read/write (not done for simplicity)
static int32_t locServices  = 7;                           // read-only (static)
static char rssiup[5];                                     // rssi da posicao 5 no vetor receiveFromBase
static char rssidown[5];                                   // rssi da posicao 4 no vetor receiveFromBase

uint32_t prevMillis = millis();
char oid[SNMP_MAX_OID_LEN];
SNMP_API_STAT_CODES api_status;
SNMP_ERR_CODES status;

// Pacote recebido com o resultado SNMP.
void pduReceived()
{
  SNMP_PDU pdu;

  api_status = Agentuino.requestPdu(&pdu);

  if (pdu.type == SNMP_PDU_GET ||
      pdu.type == SNMP_PDU_GET_NEXT ||
      pdu.type == SNMP_PDU_SET &&
      pdu.error == SNMP_ERR_NO_ERROR &&
      api_status == SNMP_API_STAT_SUCCESS)
  {
    pdu.OID.toString(oid);

    if (strcmp_P(oid, rssiUp1) == 0 ||
        strcmp_P(oid, rssiUp2) == 0 ||
        strcmp_P(oid, rssiUp3) == 0 ||
        strcmp_P(oid, mediaRssiUp1) == 0 ||
        strcmp_P(oid, mediaRssiUp2) == 0 ||
        strcmp_P(oid, mediaRssiUp3) == 0)
    {
      if (pdu.type == SNMP_PDU_SET)
      {
        pdu.type = SNMP_PDU_RESPONSE;
        pdu.error = SNMP_ERR_READ_ONLY;
      }
      else
      {
        requestRSSI(oid, 500); // pergunta por informacoes de RSSI do sensor, com timeout.

        if (receiveFromBase[2] >= 128)
```

```
      {
        if(pow_Up_index < SIZE)
        {
           rssi = ((receiveFromBase[2] - 256) >> 1) - vetorCorrec[retSensor - 1]; // calculo da rssi

           if(rssi < -40)
             Udp.sendPacket("1.3.6.1.4.1.23955.2.1.1001.3.1.0", destip, 162);

           rssi_W_up[retSensor - 1][pow_Up_index] = 0.001 * pow(10, (rssi/10)); // transforma em
Watt
           pow_Up_index++; // incrementa indexacao
        }
        else
        {
          pow_Up_index = 0; // reseta indexacao
        }
      }
      else
      {
        if(pow_Up_index < SIZE)
        {
           rssi = (receiveFromBase[2] >> 1) - vetorCorrec[retSensor - 1]; // calculo da rssi

           if(rssi < -40)
             Udp.sendPacket("1.3.6.1.4.1.23955.2.1.1001.3.1.0", destip, 162);

           rssi_W_up[retSensor - 1][pow_Up_index] = 0.001 * pow(10, (rssi/10)); // transforma em
Watt
           pow_Up_index++; // incrementa indexacao
        }
        else
        {
          pow_Up_index = 0; // reseta indexacao
        }
      }

      if (strcmp(rssiup, "ERR") != 0)
      {
        itoa((int)rssi, rssiup, 10);
      }

      if (strcmp_P(oid, rssiUp1) == 0 ||
          strcmp_P(oid, rssiUp2) == 0 ||
          strcmp_P(oid, rssiUp3) == 0)
      {
        status = pdu.VALUE.encode(SNMP_SYNTAX_OCTETS, rssiup);
        pdu.type = SNMP_PDU_RESPONSE;
        pdu.error = status;
      }
      else
      {
        for(int index = 0; index < SIZE; index++)
        {
          media_UP += rssi_W_up[retSensor - 1][index];
          media_UP = media_UP/SIZE;
        }

        status = pdu.VALUE.encode(SNMP_SYNTAX_OCTETS, rssiup);
        pdu.type = SNMP_PDU_RESPONSE;
        pdu.error = status;
```

```
    }
  }
}
else if (strcmp_P(oid, rssiDown1) == 0 ||
         strcmp_P(oid, rssiDown2) == 0 ||
         strcmp_P(oid, rssiDown3) == 0)
{
  if (pdu.type == SNMP_PDU_SET)
  {
    pdu.type = SNMP_PDU_RESPONSE;
    pdu.error = SNMP_ERR_READ_ONLY;
  }
  else
  {
    requestRSSI(oid, 500); // pergunta por informacoes de RSSI do sensor, com timeout.

    if (receiveFromBase[0] >= 128)
    {
      if(pow_Dw_index < SIZE)
      {
        rssi = (( receiveFromBase[0] - 256) >> 1) - vetorCorrec[retSensor - 1];
        rssi_W_dw[retSensor - 1][pow_Dw_index] = 0.001 * pow(10, (rssi/10));
        pow_Dw_index++;
      }
      else
      {
        pow_Dw_index = 0;
      }
    }
    else
    {
      if(pow_Dw_index < SIZE)
      {
        rssi = (receiveFromBase[0] >> 1) - vetorCorrec[retSensor - 1];
        rssi_W_dw[retSensor - 1][pow_Dw_index] = 0.001 * pow(10, (rssi/10));
        pow_Dw_index++;
      }
      else
      {
        pow_Dw_index = 0;
      }
    }

    if (strcmp(rssidown, "ERR") != 0)
    {
      itoa((int)rssi, rssidown, 10);
    }

    status = pdu.VALUE.encode(SNMP_SYNTAX_OCTETS, rssidown);
    pdu.type = SNMP_PDU_RESPONSE;
    pdu.error = status;
  }
}
else if (strcmp_P(oid, sysName) == 0)
{
  if (pdu.type == SNMP_PDU_SET)
  {
    status = pdu.VALUE.decode(locName, strlen(locName));
    pdu.type = SNMP_PDU_RESPONSE;
    pdu.error = status;
```

```
    }
    else
    {
      status = pdu.VALUE.encode(SNMP_SYNTAX_OCTETS, locName);
      pdu.type = SNMP_PDU_RESPONSE;
      pdu.error = status;
    }
  }
  else if (strcmp_P(oid, sysContact) == 0)
  {
    if (pdu.type == SNMP_PDU_SET)
    {
      status = pdu.VALUE.decode(locContact, strlen(locContact));
      pdu.type = SNMP_PDU_RESPONSE;
      pdu.error = status;
    }
    else
    {
      status = pdu.VALUE.encode(SNMP_SYNTAX_OCTETS, locContact);
      pdu.type = SNMP_PDU_RESPONSE;
      pdu.error = status;
    }
  }
  else if (strcmp_P(oid, sysLocation) == 0)
  {
    if (pdu.type == SNMP_PDU_SET)
    {
      status = pdu.VALUE.decode(locLocation, strlen(locLocation));
      pdu.type = SNMP_PDU_RESPONSE;
      pdu.error = status;
    }
    else
    {
      status = pdu.VALUE.encode(SNMP_SYNTAX_OCTETS, locLocation);
      pdu.type = SNMP_PDU_RESPONSE;
      pdu.error = status;
    }
  }
  else if (strcmp_P(oid, sysServices) == 0)
  {
    if (pdu.type == SNMP_PDU_SET)
    {
      pdu.type = SNMP_PDU_RESPONSE;
      pdu.error = SNMP_ERR_READ_ONLY;
    }
    else
    {
      status = pdu.VALUE.encode(SNMP_SYNTAX_INT, locServices);
      pdu.type = SNMP_PDU_RESPONSE;
      pdu.error = status;
    }
  }
  else
  {
    pdu.type = SNMP_PDU_RESPONSE;
    pdu.error = SNMP_ERR_NO_SUCH_NAME;
  }
  Agentuino.responsePdu(&pdu);
}
Agentuino.freePdu(&pdu);
```

```
}

void setup()
{
  Serial.begin(9600);
  Serial1.begin(9600); // Link do agente com a base RFBee

  Ethernet.begin(mac, ip);
  api_status = Agentuino.begin();

  if (api_status == SNMP_API_STAT_SUCCESS)
  {
    Agentuino.onPduReceive(pduReceived);
    delay(10);
    Serial << F("SNMP Agent Initalized...") << endl;
    return;
  }

  delay(10);
  Serial << F("SNMP Agent Initalization Problem...") << status << endl;
}

void loop()
{
  Agentuino.listen();
  if (millis() - prevMillis > 1000)
  {
    prevMillis += 1000;
    locUpTime += 100;
  }
}

void requestRSSI(char* pOIDsensor, int timeout)
{
  byte buf[RSSI_REQUEST];
  char bufferSensor[5];
  char bufferServico[5];

  int retServico;

  memset(buf, 0x00, sizeof(buf));

  memset(bufferSensor,'\0', sizeof(bufferSensor));
  memset(bufferServico,'\0', sizeof(bufferServico));

  strncpy(bufferSensor, &pOIDsensor[22], 4);
  strncpy(bufferServico, &pOIDsensor[29], 4);

  retSensor = atoi(bufferSensor) - 1000;
  retServico = atoi(bufferServico) - 1000;

  buf[8] = (byte)retSensor;
  buf[10] = 0;

  memset(rssiup,'\0', sizeof(rssiup));
  memset(rssidown,'\0', sizeof(rssidown));

  Serial1.write((byte *)buf, RSSI_REQUEST);
  Serial1.flush();
```

```
  delay(timeout); // Espera por uma resposta durante 'timeout' milisegundos.

  if(Serial1.available() > 0)
  {
    for(int i = 0; i < RSSI_RESPONSE; i++)
    {
      receiveFromBase[i] = Serial1.read();
    }
  }
  else
  {
    memcpy(rssiup, "ERR", 4);
    memcpy(rssidown, "ERR", 4);
  }
}
```