PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS

CENTRO DE CIÊNCIAS EXATAS, AMBIENTAIS E DE TECNOLOGIAS - CEATEC

FACULDADE DE ENGENHARIA ELÉTRICA

JOHN F L MADEIRA

GERÊNCIA DE REDES SENSORES SEM FIO COM SNMP: UMA ABORDAGEM COM PROXY GATEWAY

Campinas

2014

JOHN F L MADEIRA

GERÊNCIA DE REDES SENSORES SEM FIO COM SNMP: UMA ABORDAGEM COM PROXY GATEWAY

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

2014

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

t621.3851
M181g

Madeira, John Franklin Loiola
Gerência de Redes Sensores Sem Fio com SNMP: uma abordagem com Proxy Gateway / John Franklin Loiola Madeira. - Campinas: PUC-Campinas, 2014.
p.

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. Redes de sensores sem fio. 2. Sistemas de computação sem fio. 3. Redes de computação. 4. Redes de computadores - Protocolos. 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 – t621.3851

Pontifícia Universidade Católica de Campinas

Centro de Ciências Exatas, Ambientais e de Tecnologia - CEATEC

Programa de Pós-Graduação

BANCA EXAMINADORA

Presidente e Orientador Prof. Dr.: _____

1^0 Examinador: _____

2^0 Examinador: _____

Campinas, _____ de _____ de 2014

Dedico este trabalho à minha
esposa, Elizabeth, pelo apoio incondicional e
para minha filha, Julia, que me ajudou com
sua alegria perene

.

AGRADECIMENTOS

Ao Prof. Dr. Omar Carvalho Branquinho,

Meu grande incentivador, orientador e mestre.

Ao Prof. Dr. Marcelo Abbade

Pela atenção e paciência

Ao Mestrando Deivis Pirani,

Grande colega e companheiro de jornada.

Aos técnicos Laboratório de Meios de Transmissão, pelo suporte técnico prestado.

À Pontifícia Universidade Católica de Campinas, pela concessão da bolsa de estudos para o Mestrado Profissional em Engenharia Elétrica.

A nossa mais elevada tarefa deve ser a de formar
seres humanos livres que sejam capazes de, por si mesmos,
encontrar propósito e direção para suas vidas.

Rudolf Steiner

RESUMO

MADEIRA, John Franklin Loiola. *Gerência de Redes Sensores Sem Fio com SNMP: Uma Abordagem Com Proxy Gateway*. 2014. 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, 2014.

Este trabalho apresenta uma abordagem para gerência de rede sensores sem fio (RSSF) com utilização do *Simple Network Management Protocol* (SNMP). Nesta proposta, foi introduzido elemento chamado SNMP Proxy Gateway (SPG) que tem objetivo de fazer a interconexão das arquiteturas de rede para gerência sem a necessidade de alteração ou adaptação da pilha de protocolos TCP/IP. Para isso, foi implementado protótipo de gerência da rede através de um conjunto hardware e software com função de fazer a transcrição dos PDUs do SNMP para a RSSF. Um conjunto de experimentos foram executados em laboratório com objetivo de testar a proposta e validar os resultados obtidos.

Palavras chaves: Rede de Sensores Sem Fio, RSSF. *SNMP*. Gerência de redes. *Proxy Gateway*

ABSTRACT

MADEIRA, John F L. Using SNMP for *Wireless Sensor Network Management: An Approach with a Proxy Gateway,* 2014. 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, 2014.

This paper presents an approach to management of wireless sensor network (WSN) with the use of the Simple Network Management Protocol (SNMP). In this proposal, was implemented a new element called SNMP Proxy Gateway (SPG) that has aimed to make the interconnection of management without the need for modification or adaptation of the stack TCP/IP protocols. Thus, a new implementation based in hardware and software were deployed to make the transition of SNMP PDUs for WSN. A set of experiments were made in laboratory to test and validate the proposed results.

Index terms: Wireless Sensor Network, WSN. SNMP. Network *Management. Proxy Gateway.*

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

6LoWPAN	*IPv6 over Low Power Wireless Personal Area Network*
AppServer	*Application Server*
ASN.1	*Abstract Syntax Notation One*
BLIP	*Berkeley IP Implematation*
CMIS	*Common Management Information Services*
CMIP	*Common Management Information Protocol*
CMOT	*Common Management Information Protocol on TCP*
CPU	*Central Processing Unit*
DB	*Data Base*
dBi	*Decibel isotropic*
dBm	*Decibel miliwatt*
DNS	*Domain Name Service*
GUI	*Graphical User Interface*
HTTP	*Hypertext Transfer Protocol*
IAB	*Internet Architecture Board*
ICMP	*Internet Control Message Protocol*
IDE	*Integrated Development Environment*
IoT	*Internet of Things*
IP	*Internet Protocol*
IPv6	*Internet Protocol* versão 6
ISO	*International Standards Organization*
LDR	*Light Dependent Resistor*
LPSiRa	Laboratório de Pesquisa em Sistemas Rádio
M2M	*Machine to Machine*
MAC	*Media Access Control*
MAN	*Metropolitan Area Network*
MIB	*Management Information Base*
ms	Milisegundo ou 10^{-3} segundo
NIC	*Network Inteface Card*
NMS	*Network Management Station*

OID	*Object Identification*
OpenWRT	Projeto *Open Source* de distribuição Linux para sistemas embarcados
OSI	*Open Systems Interconnection*
PAN	*Personal Area Network*
PD	*Propagation Delay*
PDU	*Protocol Data Unit*
ProcD	*Processing Delay*
PTT	*Packet Transmission Time*
PUCCAMP	Pontifícia Universidade Católica de Campinas
RAM	*Random Access Memory*
RF	*Radio Frequency*
RFC	*Request for Comments*
RFID	*Radio-Frequency Identificaion*
RoF	*Radio over Fibre*
RSSF	Rede de Sensores Sem Fio
RSSI	*Radio Signal Strength Indicator*
RTC	*Real Time Clock*
RTT	*Round Trip Time*
SCADA	*Supervisory Control and Data Acquisition*
SGMP	*Simple Gateway Monitoring Protocol*
SM	Single Mode
SMI	*Storage Management Initiative*
SNMP	*Single Network Management Protocol*
SPG	*SNMP Proxy Gateway*
TCP	*Transmission Control Protocol*
UDP	*User Datagram Protocol*
WiFi	*Wireless Fidelity.* Marca registrada da *Wi-FI Alliance*
WSN	*Wireless Sensor Network*
WSNMP	*Wireless Single Network Management Protocol*
XML	*Extensible Markup Language*
ZigBee	Padrão de redes sem fio baseado no IEEE 802.15.4

SUMÁRIO

1. Introdução

As redes sensores sem fio (RSSFs) podem conter desde algumas unidades de nós sensores até milhares de dispositivos distribuídos nas cidades e em grandes regiões metropolitanas. Com isso, surge um grande desafio: A gerência das RSSFs. Esse desafio leva em conta que as RSSFs não são como as redes tradicionais. As restrições de tamanho dos nós sensores, limitação de processamento e memória além do controle do gasto de energia e, ainda, a localização dispersa desses dispositivos, fazem com que a gerência da RSSF seja considerada como fator chave para o sucesso de operação e sua evolução. Por outro lado, é importante que o legado das redes tradicionais seja mantido visando redução de gastos com implantação de novos protocolos e sistemas específicos de gerência.

Um dos grandes pontos críticos das RSSFs é o gasto de energia dos nós sensores (ou dispositivos sensores) devido, em grande parte, às configurações mono-otimizadas dos componentes de hardware e software que formam a RSSF [1]. Ainda de acordo com [1], a forma de gerenciamento das RSSFs, interfere diretamente em seu desempenho global. Quanto ao contexto de eficiência de gerenciamento dos recursos de uma RSSF, o gerenciamento pode ser abordado de duas formas: A primeira abordagem envolve o desenvolvimento de protocolos de comunicação mais eficientes para tratar as especificidades da RSSF sendo necessário, ainda, o uso de mecanismos mais eficientes quanto ao Controle de Acesso ao Meio (MAC), a auto-organização dos nós sensores e sua organização em relação aos nós vizinhos e, além disso, a eficiência na comunicação entre os nós e os gerentes (*clusters heads*). A segunda abordagem trata do impacto quanto às atividades que possam estar gerando desperdício de recursos e que possam aumentar o gasto de energia, tanto no aspecto local do nó sensor, quanto no aspecto global da RSSF envolvendo as características de comunicação de toda a rede. Essa abordagem está diretamente envolvida quanto a forma de como é feita a gerência da RSSF. Nessa proposta de dissertação, terá como foco o aspecto de gerência da RSSF.

Na literatura é bastante conhecido o problema da escassez de energia nas redes de sensores sem fio. Conforme mencionado em [2], numa RSSF há diversos fatores que podem limitar o desempenho dos nós sensores devido a baixa eficiência na utilização dos recursos e no desperdício de energia dos nós da rede. Um desses fatores é a limitação de tamanho do dispositivo sensor, uma vez que estes nós tem o tamanho bastante reduzido, não sendo capazes de suportar fisicamente grandes fontes acumuladoras de energia. Outro fator é que, devido a grande quantidade de nós sensores que uma RSSF pode conter, a substituição ou gerenciamento da fonte de energia para cada dispositivo, torna-se uma tarefa extremamente onerosa e complexa do ponto de vista da gerência da rede. Mesmo com a adoção de fontes de energia alternativas como células de energia solar ou conversores de campo eletromagnético [3], as dimensões reduzidas dos nós sensores limitam bastante o uso dessas alternativas de energia e, dependendo do tipo de aplicação a qual se destina, torna-se inviável sua adoção.

Além disso, outros problemas como o próprio funcionamento dos nós sensores podem criar bastante impacto da gestão da rede [4] sendo de difícil controle se feita sem auxílio de um sistema de gestão da RSSF de forma centralizada. Um exemplo disso é que, dependendo do tipo de topologia adotada, a falha de um nó sensor pode refletir de forma destrutiva em parte segmento de rede, traduzindo-se em isolamento de todo segmento caso ele seja rota para chegar até o *cluster head* da rede o qual, por sua vez, pode detectar de forma incorreta que um segmento está indisponível. Neste caso, há duas formas de resolver o problema: A primeira é através da auto-organização dos nós sensores que podem ocorrer de forma monolítica ou autônoma. A segunda forma é sistêmica onde é avaliado o comportamento de RSSF de uma forma direta envolvendo ações de contra-medidas que podem ser automatizadas por um sistema centralizado de gerência com o propósito de poupar recursos dos nós sensores e do canal de comunicação.

No caso de uma formação monilítica, a tomada de decisão do novo arranjo será feita por cada nó sensor individualmente, deixando que todo o processamento para formação da nova topologia fique restrito aos nós sensores,

no entanto, haverá aumento do consumo do recurso da bateria para tal atividade. Essa abordagem não é vantajosa do ponto de vista de eficiência energética para os nós sensores [5].

Já para a formação sistêmica, a decisão para rearranjo do conjunto é feita pelo sistema de gestão da RSSF o qual tem uma visão global da rede o que facilita a formação no novo arranjo, uma vez que ele contém as informações de cada nó e pode avaliar a melhor as possibilidades de rota para desvio do tráfego de rede e, além disso, evita que o nó sensor tenha gasto de energia desnecessário com o processamento de novas rotas e, consequentemente, racionalizando o canal de comunicação da RSSF.

Naturalmente, sendo a tomada de decisão feita pelo sistema de gerência, é necessária a adoção de contrapartidas para que se possa garantir a sua operacionalidade e a adoção de alguns requisitos fundamentais como, por exemplo, redundância de componentes e segurança no ambiente de gerência, os quais devem ser considerados no planejamento da implantação da RSSF.

A gestão sistêmica da RSSF também pode ser feita de forma manual, onde o gestor da RSSF interpreta a informação através de um catálogo de eventos gerados pelo sistema de gestão onde é apresentado e identificado o problema e uma proposta de solução de contorno é proposta ou, também, pelo próprio sistema de gestão, através do uso de um sistema de supervisão e aquisição de dados (SCADA) utilizando-se, por exemplo, o próprio protocolo SNMP, o qual pode ser usado como vetor da tomada de decisão de forma autônoma pelo sistema de gerência com base em pré-condições previamente configuradas e interpretadas pelo ambiente de gestão que utilizada primitivas do tipo *Get, Set e Trap*.

Nessa dissertação, será considerado a forma sistêmica de gerenciamento das RSSF com uso do protocolo SNMP. Para isso, será apresentado uma *Management Information Base* (MIB) específica para as RSSFs com o objetivo de criar um *framework* para o gerenciamento das particularidades e funcionalidades características da rede sensores sem fio.

1.1 Motivação

O conceito de “cidade conectada” está cada vez mais presente em nosso dia-a-dia. Este conceito é também conhecido por *Internet of Things* (IoT) e é apresentado por [6] como a próxima evolução da Internet a qual mudará a forma que os seres humanos interagem com o meio ambiente. Desta forma, espera-se um impacto ainda maior quando este conceito for consolidado em áreas como educação, clima, segurança, etc. Ainda conforme [6], em 2010 já havia mais dispositivos conectados (12,5 bilhões) do que pessoas no mundo (6,8 bilhões). Em 2020 é estimado 50 bilhões de dispositivos conectados contra uma população aproximada de 7,6 bilhões de pessoas.

Neste cenário, a comunicação entre os dispositivos (M2M) se tornará ainda mais comum, fazendo o monitoramento e controle de processos irem além da telemetria, telecomando e telesupervisão. Isso tornará mais real a comunicação das pessoas com o mundo físico fazendo que se tenha maior liberdade nas interações entre as ‘coisas’ e nas atitudes proativas desses dispositivos. Isso somente é possível pelo uso em larga escala das RSSFs nas cidades e nos centros metropolitanos [5].

Por outro lado, as características das RSSFs tornam essas redes singulares. O uso limitado de recursos e sua aplicação em ambientes remotos e, muitas vezes, hostis evidenciam isso [7]. Devido a essas características, o correto gerenciamento dos recursos da RSSF passa pela gerência dos componentes que formam a RSSFs: os nós sensores, o canal de comunicação e da própria rede em si, avaliando seu comportamento, operação e expansão. [8].

Conforme mencionado por [9], a gerência inteligente leva em conta a administração de três recursos básicos: energia, memória e processamento. Esses são recursos críticos para operação de qualquer RSSF e, por isso, essas

redes não devem ser consideradas como as redes tradicionais fazendo com que o sistema de gerência seja um ponto importante seu desempenho.

1.2 Objetivo

Conforme discutido anteriormente, devido as suas singularidades, as RSSFs devem ser vistas como redes com características próprias. No entanto, é importante fazer com que essas redes utilizem componentes que possam fazer a 'tradução' entre a RSSF e a rede TCP/IP. Um componente chave para essa função é proposto aqui: *SNMP Proxy Gateway* (SPG).

Desta forma, foi desenvolvido um *gateway* como um sistema embarcado de baixo custo baseado numa plataforma Linux utilizando-se *scripts* em linguagem de programação Lua. Sua função é fazer a interpretação dos valores dos transdutores dos nós sensores para Identificadores de Objetos (OIDs) que possam ser lidos ou configurados pelas primitivas do SNMP descrito na RFC 1157 [10] usando as primitivas *Get*, *Set e Trap*.

Também é proposto uma *Management Information Base* (MIB), conforme descrito na RFC 1213 [11] específica para as RSSFs o qual tem a função de fazer a interpretação dos OIDs para objetos amigáveis do ponto de visto do usuário.

1.3 Organização

Este trabalho está organizado da seguinte forma: Na seção 2 serão discutidos alguns trabalhos relacionais ao tema. Na seção 3 é apresentada uma breve descrição dos componentes dos sistemas. Na seção 4 é apresentado conceitos aplicados no desenvolvimento do trabalho e a construção da MIB. Na

seção 5 são apresentados componentes individuais da proposta. Na seção 6 são apresentados os experimentos e resultados obtidos. E, por fim, na seção 7 é apresentado as conclusões finais.

2. Trabalhos Relacionados

Há na literatura vários trabalhos que exploram o uso do SNMP seguindo os cinco pilares clássicos da gerência definido pela *Open Systems Interconnection* (OSI) e apresentado por [12]: Gerência de Falhas, Gerência de Contabilidade, Gerência de Configuração, Gerência de Desempenho e Gerência de Segurança. De forma geral, as pesquisas têm como desafio principal apresentar mecanismos para melhorar uma ou mais dessas cinco áreas.

Em [13], por exemplo, é proposto uma implementação de um protocolo baseado no *Protocol Data Unit* (PDU) do SNMP versão 1 (SNMPv1) para redes sensores móveis. Essa implantação é baseada numa distribuição hierárquica de rede com formação de c*lusters* o qual realiza associações em grupos de nós sensores com objetivo de reduzir a troca de mensagens entre gerente (*clusters head*) e agentes ou nós sensores. Uma variação dessa implementação também pode ser vista em [14] para redes sensores em regiões metropolitanas com uso da tecnologia radio sobre fibra (RoF). Este tipo de implementação tem reflexo na Gerencia e Desempenho, pois, minimiza o trafego de mensagens entre os dispositivos gerenciados. Também reflete nas áreas como a Gerência de Falhas porque diminui a perda de pacotes, consequentemente, aumenta a vida útil da bateria do nó sensor e, ainda, na Gerência de Configuração porque possibilita melhor precisão na coleta de informação para a gerência da rede promovendo o acesso à dados dos objetos gerenciáveis.

Outras propostas, como as discutidas por [15] e [16], apresentam modelos de gerência utilizando SNMP. O objetivo é melhorar a eficiência da RSSF adotando-se unicamente o uso da pilha TCP/IP nos dispositivos gerenciáveis para acesso aos identificadores de objeto (OIDs). No entanto esses modelos exigem uma implemantação mais complexa e, muitas vezes, necessitam rodar a pilha TCP/IP nos nós sensores. O problema dessa abordagem é que isso implica em embarcar toda a pilha do TCP/IP nos dispositivos gerenciáveis fazendo com que se tenha um desperdício quanto ao

processamento e capacidade de memória, uma vez que as subseções da pilha também são carregadas. No entanto, o carregamento completo da pilha TCP/IP nem sempre é possível devido a limitação de recursos desses dispositivos. Conforme mencionado por [8], por exemplo, o carregamento dos protocolos TCP e ICMP é desnecessário para o SNMP que causa aumento no cosumo de energia dos nós sensores e impactam diretamente a desempenho sistêmico da RSSF. Ainda segundo [8], o protocolo de comunicação consome aproximadamente 75% de energia dos nós sensores de uma RSSF. Essa é uma das razões para a não adoção do uso direto da pilha TCP/IP nos dispositivos que compõem a RSSF.

Mais recentemente [17] propõe um sistema de gerência chamado *WSN Management System*. Nesta proposta, é discutida a gestão centralizada dos componentes gerenciáveis através dos relacionamentos dos objetos focando em três áreas: Configuração, Desempenho e Falha. Como indicado por [17], esse método diminui a perda de pacotes e o tempo de resposta ou *Round Trip Time* (RTT) porque permite aos administradores tomar decisões assertivas quanto a classificação dos dados da rede (intensidade de sinal, perda de pacotes, nível da bateria, etc) e de dados do usuário obtidas pelo transdutores de medição como temperatura, luminosidade, entre outros. Apesar desses benefícios como apontados em [17], essa implementação ainda envolve embarcar a pilha TCP/IP para os nós sensores o que acarreta problemas já apontados.

Como alternativa, há propostas sobre o uso do protocolo IPv6 com adaptações para implementação para as RSSFs. Essa abordagem usa o 6LoWPAN (*IPv6 over Low Power Wireless Personal Area Network*) definido pela RFC 4919 [18] e modificada para atender o padrão de redes sensores ZigBee (IEEE 802.15.4) conforme RFC 4944 [19]. No entanto, o uso do 6LoWPAN traz algumas questões relacionadas ao desempenho que ainda são bastantes incipientes como, por exemplo, o excesso de fragmentação causado a introdução de um novo *layer* para adaptar do *datagrama* IPv6 de 1.280 *Bytes* para o padrão 802.15.4 de 127 *Bytes* o que é, consequentemente, traduzido em um custo computacional para os nós, implicando em maior consumo de energia.

Outra questão é a real necessidade de se manter a adoção do *Internet Protocol* (IP) o qual tem a função exclusiva de roteamento na camada de rede, uma vez que nem sempre há necessidade de roteamento por pacotes usando o protocolo IP em uma *Personal Area Network* (PAN) a qual pode ser uma característica típica de uma RSSF [20] [21].

Por outro lado, em [22] por exemplo, é proposto uma nova abordagem clamada *b*6LoWPAN. A proposta baseia-se numa implementação feita pela Universidade de Berkeley, conhecida como *Berkeley IP Implematation* (Blip) para uso com a plataforma *TinyOS*. Essa implementação introduz várias técnicas como compressão de cabeçalho, roteamento ponto-a-ponto e descobrimento de rota por vizinhança. No entanto as questões ainda persistem como, por exemplo, aumento de processamento e consumo de memória devida fragmentação e remontagem dos *datagramas*.

Desta forma, a proposta apresentada aqui, não tem pretensão de fazer nenhum grande alteração nas camadas nas baixas do modelo de referência OSI e sim, ao contrário, manter todas as características dessas camadas preservando sua integridade operacional, porém, associando a forma especifica de comunicação da RSSF através da integração com um *Proxy Gateway*. Essa proposta será discutida mais à frente. Antes, serão apresentados breves conceitos dos requisitos básicos para formação desse ambiente.

3. O SNMP e a MIB

O SNMP é um protocolo de camada de aplicação que compõe a pilha de protocolos TCP/IP e é atualmente utilizado amplamente como ferramenta na gerência de rede. Ele permite a troca de informações entre os dispositivos gerentes e os agentes ou dispositivos gerenciados.

A proposta apresentada aqui é manter a simplicidade quanto a implementação do SNMP. Isso é traduzido na adoção de um SNMP Proxy Gateway (SPG) cuja função principal é fazer a integração entre a rede TCP/IP e a redes sensores sem fio (RSSF). Para isso, serão apresentados a seguir conceitos básicos dos componentes essenciais para entendimento dessa proposta.

3.1 Simple Network Management Protocol - SNMP

Segundo [12], as redes se tornam cada vez maiores, complexas e heterogêneas fazendo com que o custo relacionado a gerência aumente a medida que sua estrutura cresça. Essa mesma abordagem pode ser aplicada para as RSSFs, onde há necessidade de ser feito controle de custos relacionados ao seu crescimento e o uso de ferramentas de gerência que possam ser capazes de interoperar em sistemas de fornecedores distintos. Este é o ambiente ideal para uso do SNMP.

O protocolo SNMP foi, inicialmente, uma recomendação do *Internet Architecture Board* (IAB) apresentada em 1988 como uma solução de curto prazo para um padrão de gerência de redes para a Internet. Assim, uma proposta foi apresentada como primeira versão ao *International Standards Organization* (ISO) de um framework de gerência. Em 1989, o comitê apresentou o padrão que se tornaria definitivo. Esse padrão era conhecido como *Common*

Management Information Services/Common Management Information Protocol (CMIS/CMIP) ou, ainda, como *Common Management Information Protocol on TCP* (CMOT) e, desta forma, surgiu o predecessor do SNMP chamado de *Simple Gateway Monitoring Protocol* (SGMP) que não era totalmente compatível com o SNMP.

A partir daí, um grupo de trabalho foi formado para definir uma estrutura de gerenciamento de informação (SMI) com base na recomendação do ISO/SMI o qual serviu como ponto de partida para desenvolvimento de uma convenção abstrata de nomes para uso na Internet. Desta forma, surgiu a *Management Information Base* (MIB). A primeira versão da MIB surgiu em 1988 através da RFC 1065 e da RFC 1066. A ideia inicial é que essa especificação da MIB fosse compatível com ambas as propostas de protocolo de gerência SNMP e SGMP, facilitando a migração dos sistemas que utilizariam o SNMP para o SGMP. No entanto, as expectativas de uso pela comunidade para o novo padrão SGMP não foram completamente atendidas devido a incompatibilidade entre os dois frameworks e, assim, o padrão SGMP foi abandonado.

Em 1990, o IAB lançou o padrão SNMP baseado no SMI/MIB com o *status full recomendation* para a comunidade. Essa recomendação ficou conhecida como SNMPv1 o qual gerou a primeira especificação do protocolo conhecida como RFC 1156. A Figura 1 mostra os campos básicos do protocolo SNMP desta versão.

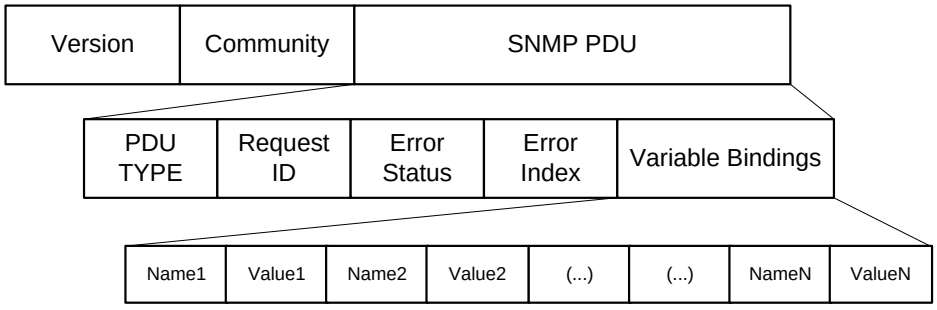

Figura 1: Datagrama SNMP

Devida a sua concepção simples o protocolo usa, essencialmente, apenas dois comandos básicos em sua operação. Esses componentes básicos são chamados de *Protocol Data Unit* (PDU). O primeiro componente básico é o *Get* que é usado para obter valores do objeto gerenciado a partir da estação de gerência. O segundo componente básico é o *Set* que é utilizado para alterar valores do dispositivo gerenciado a partir da estação de gerência. Desta forma, a estação de gerência pode ter controle sobre os dispositivos gerenciados de uma forma centralizada e coordenada.

Há um terceiro tipo de PDU que é utilizado em requisições não solicitadas pela estação de gerência: *Trap*. O *Trap* é um tipo de PDU que ocorre quando o dispositivo gerenciado reconhece uma anomalia ou alguma situação de falha é detectada e, desta forma, o próprio dispositivo gerenciado tem autonomia para enviar informações à estação de gerência mesmo sem que essa informação ter sido solicitada. Este tipo de PDU tem um formado do *datagrama* diferente. A Figura 2 mostra o PDU do tipo *Trap*.

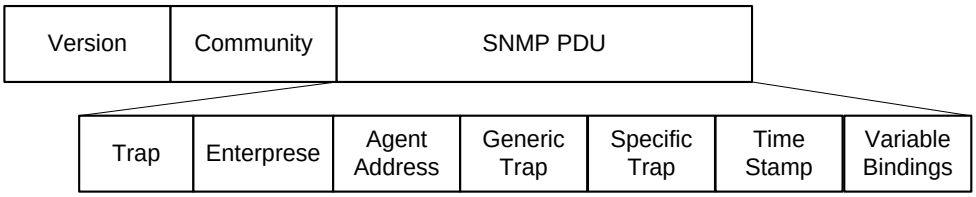

Figura 2: PDU do tipo Trap

A segunda versão (SNMPv2) foi lançada em 1993 com foco em prover melhoria nos mecanismos para uma nova base de informação de objetos gerenciáveis. Isso é descrito pela MIB-2 na RFC 1213. Outra importante contribuição foi a especificação para introduzir um mecanismo para autenticação provendo melhor grau de segurança ao protocolo o que, diferentemente da versão 1, não era previsto. Essa especificação foi descrita na RFC 1441 e RFC 1442. No entanto, esse mecanismo de segurança era vulnerável e facilmente manipulado com o avanço da tecnologia de chaves criptográficas [12]. Com isso houve necessidade de aprimorar o protocolo para incorporar um mecanismo de

segurança mais robusto e confiável. Essa melhoria seria incorporada de versão seguinte.

Assim, o grupo de trabalho que lançou a terceira versão do SNMP (SNMPv3) teve a preocupação com a implantação de segurança o que não era, até então, foco principal das versões anteriores. Essa nova característica inclui mecanismos de autenticação e privacidade para os dados. As especificações de operação da nova versão do protocolo podem ser encontradas nas RFCs 3410 e 3418 e as novas implementações de segurança estão descritas na RFC 3414, que descreve individualmente modelo de segurança de usuário, e na RFC 3415, que disponibiliza o modelo de controle de acesso para essa nova versão.

Todas as versões do protocolo usam a mesma estrutura modular. Isso facilita a compatibilidade com o padrão anterior tornando-se possível que futuros grupos de trabalho possam aprimorar o protocolo sem alterações do legado. O projeto de desenvolvimento modular, também tem a vantagem de deixar toda a documentação atualizada sem a necessidade de alterações de outras partes da documentação.

A comunicação entre a estação de gerência e os dispositivos gerenciados é feita através do *User Datagram Protocol* (UDP) como protocolo da camada de transporte da pilha TCP/IP. Não é objeto aqui detalhar como esse processo de comunicação é feito porque ele é transparente para a camada de aplicação. Maiores detalhes sobre esse processo da camada de transporte é discutido por [12].

De forma sintética, um dos campos importantes é o *variable bindings*. Esse campo contém informação quanto ao valor da variável a ser requisitada ou modificada pelo gerente para o dispositivo gerenciado nos casos dos PDUs do tipo *Get* e *Set,* respectivamente. No caso do PDU do tipo *Trap*, ele é responsável pelo envio da codificação de erro reconhecido do dispositivo gerenciado ao gerente da rede. Outro campo importante é o campo *agent address* que contém o endereço do *Object Identifier* (OID) a qual deve ser associado pela MIB. Os detalhes de como essa associação é feita na MIB serão discutidos a seguir.

3.2 *Management Information Base – MIB*

Todas as informações recolhidas e manipuladas pelo sistema de gerência são definidas através da *Management Information Base* (MIB). Os dispositivos gerenciados executam tarefas de gerenciamento da rede através de requisições feitas pela estação de gerência de rede. Os dispositivos gerenciados contém partes menores de informações de gerência chamados objetos gerenciáveis. Esses objetos são descritos usando um conjunto de configurações indicados pela *Abstract Syntax Notation One* (ASN.1) e, assim, cada objeto gerenciável contém um nome, uma síntese de informação e um código de descrição.

A descrição completa do ASN.1 está além do escopo deste trabalho mas, de forma resumida, as maiores vantagens do seu uso são:

1. Especificamente projetada para comunicação entre dispositivos, independente de plataforma;

2. È extensível e pode ser usada para descrever a grande maioria dos objetos que formam a rede;

3. Uma vez que um termo é definido, ele pode ser reutilizado fazendo-se blocos para construção de outros objetos. Esse é um ponto importante porque ajuda na estruturação da MIB que será vista mais à frente.

Uma MIB é uma base de dados virtual que contém as informações dos objetos gerenciados que descreve o tipo de informação de cada dispositivo gerenciado conforme o padrão ASN.1 mencionado anteriormente. Do ponto de vista da MIB, cada objeto gerenciável é associado a um identificador de objeto (OID) que determina seu posicionamento e seu nome único dentro da MIB. Um OID, nada mais é, do que uma sequência numérica obtida através do mecanismo de varredura da árvore de endereços para identificação de um objeto específico.

Esse processo é análogo a formação de um endereço ou um caminho, a partir da raiz (root) para localização de um componente específico numa estrutura *Domain Name Service* (DNS). A Figura 3 mostra a estrutura de árvore e seus respectivos ramos aplicados na proposta desta dissertação.

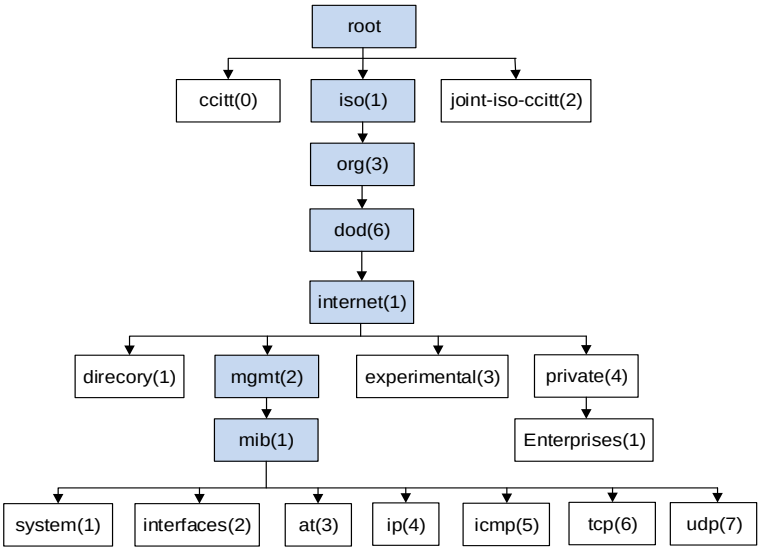

Figura 3: Arvore para formação do OID

Conforme mencionado anteriormente, a árvore segue um processo hierárquico de identificação de seus ramos. Esse mecanismo é iniciado pelo identificador raiz e depois segue o padrão pai-filho até chegar ao objeto desejado.

Por padrão, abaixo do *label* Iso(1), o ISO convencionou uma sub-árvore para org(3) para uso por organismos nacionais e internacionais. Logo depois, o *label* dod(6) foi associado para o Departamento de Defesa dos EUA. O IAB reservou o ramo logo abaixo, chamado de internet(1), para ficar sob sua administração. Desta forma, o OID de identificação 1.3.6.1 ou ainda iso.org.dod.internet, identifica o ramo da árvore com o *label* Internet. Todos os demais objetos gerenciáveis devem ficar abaixo desse ramo. Mais à frente, será discutido sobre a construção de uma MIB aplicada especificamente às redes sensores sem fio.

4. Gerência para Redes Sensores Sem Fio

As RSSFs estão se tornando cada vez mais importantes porque abre grandes possibilidades para aplicações que envolvam a *Internet of Things* (IoT) conforme mencionado em [23]. As RSSFs podem ser aplicadas em ambientes perigosos e de difícil acesso, onde as condições de funcionamento não de aplicariam para as redes tradicionais. Geralmente, esse tipo de rede consiste em um grande número de nós sensores densamente distribuídos sobre uma região de interesse para coleta de informação ou controle específico de componentes físicos do ambiente.

Tipicamente esses dispositivos tem baixa taxa de transporte de dados que está na ordem de quilo bits por segundo (kbps) e, devido ao seu tamanho reduzido, são implementados protocolos ou mecanismos de comunicação capaz de racionalizar o consumo de energia fazendo, por exemplo, com que o nó fique em estado de hibernação enquanto não está transmitindo.

Alternativamente, outra técnica para poupar bateria dos nós sensores, é a adoção de roteamento por múltiplos saltos fazendo com que a mensagem chegue ao destino através da construção de rota através de vários nós sensores. Há, também, restrições quanto ao uso recursos de memória e de processamento, o que limita seu uso individual em aplicações mais complexas que exijam manipulação de informação local.

A Figura 4 mostra uma RSSF enquanto que a Figura 5 apresenta o nó sensor utilizado neste trabalho o qual é composto tipicamente por um sistema de alimentação, processador, conjunto de transdutores e um transceptor de rádio frequência (RF).

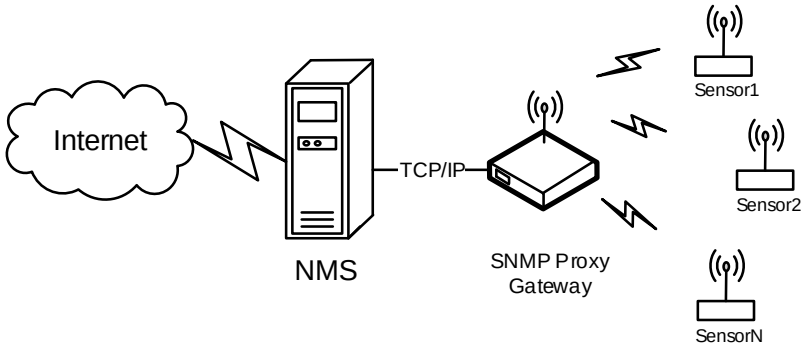

Figura 4: Visão simplificada dos componentes da RSSF

Por outro lado, há diversos trabalhos publicados que mostram as RSSFs sendo utilizadas em vários tipos de situações ajudando na coleta de grande volume de informações em milhares de pontos de monitoramento. Uma RSSF pode, por exemplo, fazer o controle de microclima de uma determinada região para que pesquisadores e usuários possam analisar as informações coletadas ajudando na avaliação e tomada de decisão. Esse mesmo conceito pode ser levado para controle e monitoramento de água usada na agricultura, atividades sismológicas e vulcânicas no interior do solo, controle de fluxo de veículos numa determinada região metropolitana e assim por diante. Atualmente, as RSSFs ainda estão presentes nas redes para automação residencial, RFID, segurança por satélite etc.

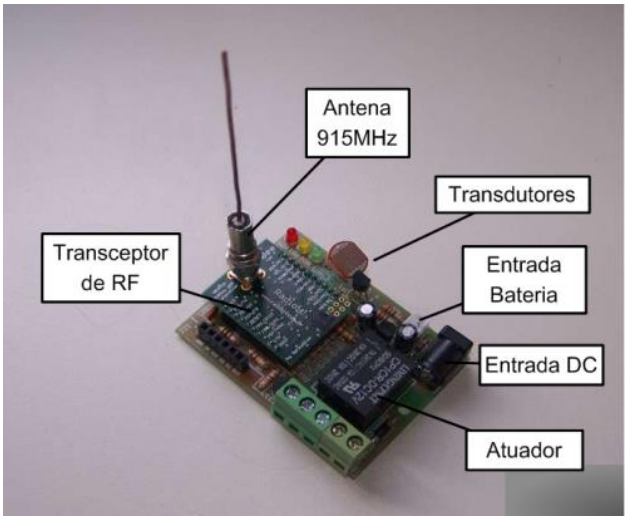

Figura 5: Nó sensor

Por isso, é importante que um sistema de gerência, neste contexto, seja desenvolvido para facilitar a operação e a consequente evolução da RSSF. Desta forma, uma nova abordagem de gerência foi proposta nesse trabalho envolvendo a uso do SNMP como facilitador do plano de gerência para integração com a RSSF. A vantagem dessa abordagem é que, além de ser de baixo custo, mantém o legado e a compatibilidade com as redes tradicionais TCP/IP.

A integração entre essas camadas é feita por um componente de interconexão chamado aqui de SNMP *Proxy Gateway* (SPG). A Figura 6 mostra um diagrama de conexão dos componentes do sistema de gerência da RSSF baseado no SPG. Neste caso, optou-se pelo uso de uma RSSF (ou *Wireless Sensor Network* - WSN) baseada em plataforma Arduino com rádio transceptor de 915 MHz. A razão disso é a facilidade de uso dos componentes de uma plataforma aberta a qual possibilita a programação dos componentes diretamente através de uma interface amigável (IDE), onde é possível manipular parâmetros como nível de potência de transmissão, modulação de sinal, escolha do canal de comunicação e largura de banda.

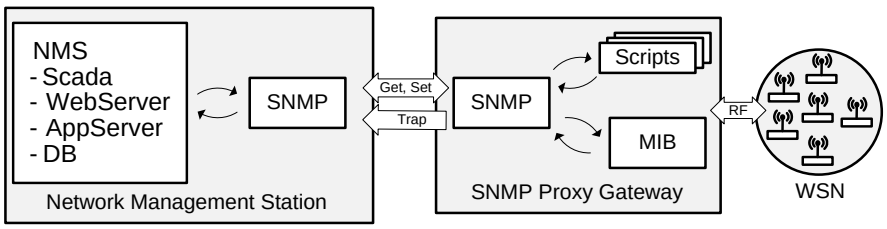

Figura 6: Diagrama de blocos do sistema

Basicamente, a proposta é dividida em dois sub-sistemas. O primeiro é o *Network Management Station* (NMS), responsável pela gerência centralizada dos objetos gerenciáveis (OIDs) da RSSF. A NMS é, por sua vez, composta por um conjunto de aplicações específicas para monitoramento, armazenamento dos dados para uso no apoio a tomada de decisão. O segundo componente é o

SPG que contém um agente SNMP. Este componente permite a comunicação com o gerente através das primitivas *Get*, *Set* e *Trap* padrão do protocolo SNMP. Isso é feito através de um conjunto de *scripts* e uma MIB especificamente desenvolvida para essa atividade. Mais à frente, será detalhado o mecanismo de funcionamento desses componentes.

4.1 Gerência da Rede e Gerência dos Dados para as RSSF

São vários os termos que devem ser definidos numa MIB. Esses termos vão desde elementos mais amplos que tratam da gerência da RSSF como, por exemplo, uma classe de objetos para uma determinação função como leitura de intensidade de sinal de RF, características da antena (ganho, tipo, polarização, etc.), até elementos específicos sobre os dados de usuário, temperatura, luminosidade, etc.

Desta forma, como mencionado anteriormente, uma nova MIB foi proposta também com o objetivo de fazer a segregação de duas categorias em relação a gerência: dados de gerência da RSSF e dados de rede do ponto de vista do usuário. O principal objetivo disso é manter os dados de gerência da rede distintos dos dados do usuário com base nos cinco pilares da gerência. A Figura 7 apresenta o diagrama mencionado nessa abordagem.

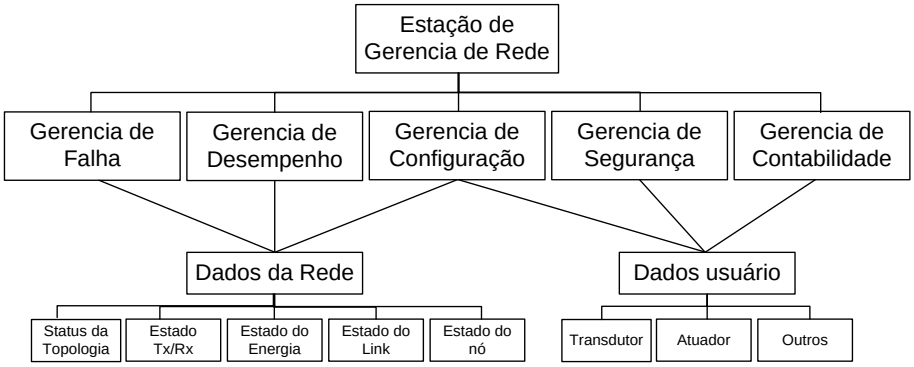

Figura 7: Relacionamento entre as áreas de gerência

Para melhor desempenho quanto a gerência de uma RSSF, é importante que se tenha uma distinção clara sobre os dados da rede e dados de usuário. Isso é importante para separar a informação relacionada aos aspectos comportamentais da rede e, consequentemente, ajudar outros aspectos como a racionalização do canal de comunicação e no apoio a tomada de decisão.

De toda a forma, independentemente da técnica utilizada para comunicação e racionalização dos recursos de uma RSSF, a gerência dos componentes que formam a rede passa a ter papel de destaque neste contexto, uma vez que gerência dos dados de uma RSSF passa a ir além do simples monitoramento convencional da rede.

4.2 Engenharia da MIB

Para cada elemento dentro de uma MIB é dado, como mencionado anteriormente, um identificador de objeto (OID). Um OID é um número único que descreve cada elemento dentro do universo da rede SNMP. Cada OID é associado a um rótulo (*label*) para fazer sentido ao usuário final. Essa tradução entre o OID e o *label* do objeto é feito também pela MIB.

Para a proposta apresentada nesse trabalho, é apresentado a sequência para o OID pai 1.3.6.1.4.1.23955.1.1.1. Isso significa que este endereço é a designação para o conjunto de nós sensores que são administrados pelo primeiro controlador SPG, ou seja, *label* SPG(1).

Abaixo dessa hierarquia é obtido um conjunto de nós filhos que são os nós sensores associados ao ramo SPG(1). A Figura 8 mostra a árvore completa associada ao primeiro nó pai 1.3.6.1.4.1.23955.1.1.1 que contém consequentemente 4 nós filhos 1.3.6.1.4.1.23955.1.1.1.101, 1.3.6.1.4.1.23955.1.1.1.102, 1.3.6.1.4.1.23955.1.1.1.103 e

1.3.6.1.4.1.23955.1.1.1.104. Desta forma, o nó pai passa a ser um *gateway* padrão daquele ramo da RSSF que contém uma quantidade de ‘n’ nós filhos.

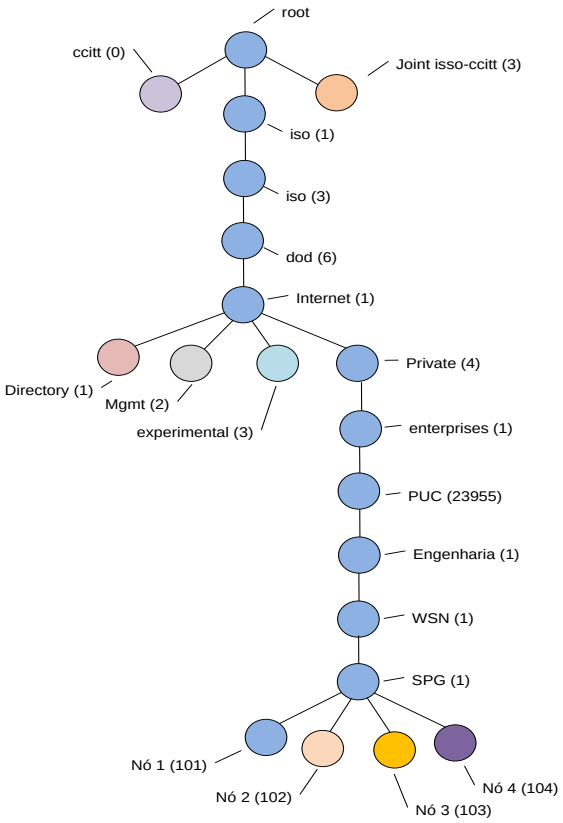

Figura 8: Árvore de identificador de objetos (OID)

Assim, cada objeto gerenciado é associado à um OID para fazer a identificação dos dados dos objetos que estão dentro de uma mensagem (PDU) do SNMP. Quando um dispositivo envia um PDU ele está, na verdade, enviando uma sequência de números no campo *variable bindings* com os respectivos valores da mensagem do objeto gerenciado. Neste caso, um objeto gerenciado pode ser um transdutor de temperatura, o *status* de um atuador (ligado/desligado), uma determinada ação (alterar valor) e assim por diante. A Figura 9 mostra a perspectiva a nível dos objetos gerenciáveis propostos neste trabalho.

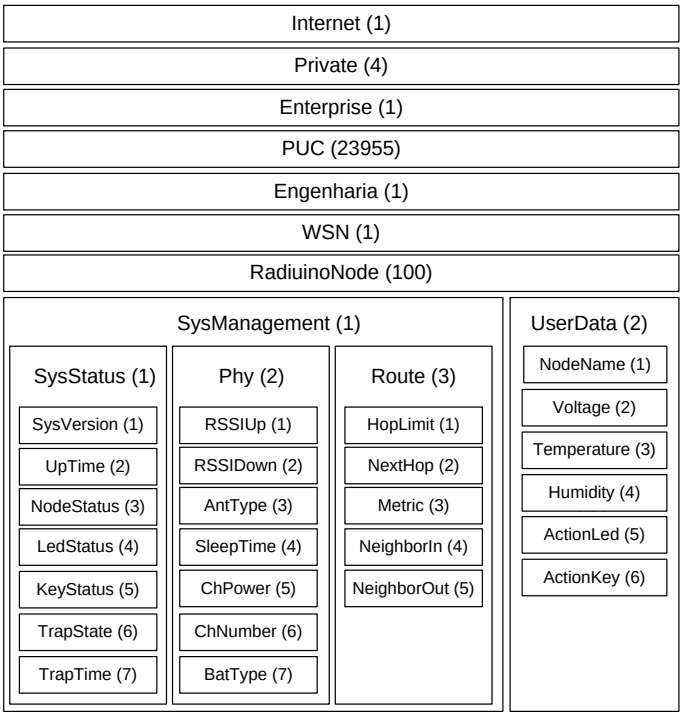

Figura 9: MIB para Rede Sensor Sem Fio

Conforme mostrada na figura acima, um exemplo de um OID é a sequência numérica 1.3.6.1.4.1.23955.1.1.1.100.1.1.5 cujo o *label* refere-se a *KeyStatus* (5), onde é obtido o valor de uma chave de um atuador do nó sensor cujo um OID é 1.3.6.1.4.1.23955.1.1.1.100. Percebe-se a hierarquia para os dados de sistema *SysManagement*(1) e dados do usuário definido por *UserData*(2), fazendo-se com que se tenha a distinção de dois tipos de informações conforme discutidos no item anterior.

Para traduzir essa estrutura a nível de implementação, é apresentado o código mostrado na Figura 10, o qual representa parte da descrição que foi desenvolvida para MIB. Essa descrição segue a definição da ASN.1

```
-- Implementation of the SENSOR group this
-- is just a example for PUCCCAMP

SensorName OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..255))
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
```

```
            "Name of node from your WSN."
    ::= { sensores 1 }

Transdutores OBJECT-TYPE
    SYNTAX  Transdutores
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
            "Valores dos transdutores."
    INDEX   { ipRouteDest }
    ::= { SensorName 1 }

Transdutores ::=
    SEQUENCE {
        Temp
            INTEGER,
        Lum
            INTEGER,
        VBat
            INTEGER,
        VFont
            INTEGER,
        RSSIDown
            INTEGER,
        RSSIUP
            INTEGER,
        RedLed
            INTEGER,
        YellowLed
            INTEGER,
        GreenLed
            INTEGER,
        KeyStatus
            INTEGER
    }
END
```

Figura 10: Parte do código ANS.1 da MIB aplicada

O código é iniciado com alguns campos e definições obrigatórios como *OBJECT-TYPE*, que contém o nome ou *label* do nó sensor, *SYNTAX* que contém informações de tamanho dos dados. O campo *ACCESS* contém a definição do tipo de acesso referente ao objeto e, por fim, os dois últimos campos *Status* e *Description* com informações de afinidade ao qual o objeto é proposto. Logo depois dos campos obrigatórios, é informado a entrada referente ao objeto gerenciado em relação a árvore que, neste caso, trata-se da entrada ::= { sensores 1 }, a qual é utilizada para identificar o nó pai com o *label* ‘SensorName 1’ e os filhos respectivos associados e esse sensor. Em sequência um conjunto de dados pode ser obtidos pela entrada Transdutores ::= onde cada entrada tem sua própria definição do tipo de dados que, neste caso, é *integer*.

Uma vez feito o mapeamento do objeto gerenciado pela MIB, a informação é obtida através de conjunto de *scripts* desenvolvidos para fazer a leitura dos transdutores dos nós sensores para o seu respectivo objeto gerenciado. Esse conjunto de *scripts* serão detalhados mais a frente.

O importante neste momento é apenas ter em mente que, após feita a coleta da informação dos nós sensores por esse conjunto de *scripts*, a informação é associada diretamente aos respectivos OIDs que, por sua vez, encaminha a informação para o NMS através das primitivas do SNMP. A partir deste ponto, é proposto uma abordagem de tratamento desses dados através da NMS. Sua proposta principal é servir como sistema de gerência centralizado para uma ou conjunto de RSSFs.

5. Implementação do NMS e do SPG

Neste capitulo serão apresentados o SPG e a NMS passando pelos seus componentes principais e o sistema de comunicação desenvolvido para gerenciamento da RSSF através do protocolo SNMP. O sistema tem dois componentes principais. A Figura 11 mostra o diagrama de blocos para visão geral desses componentes

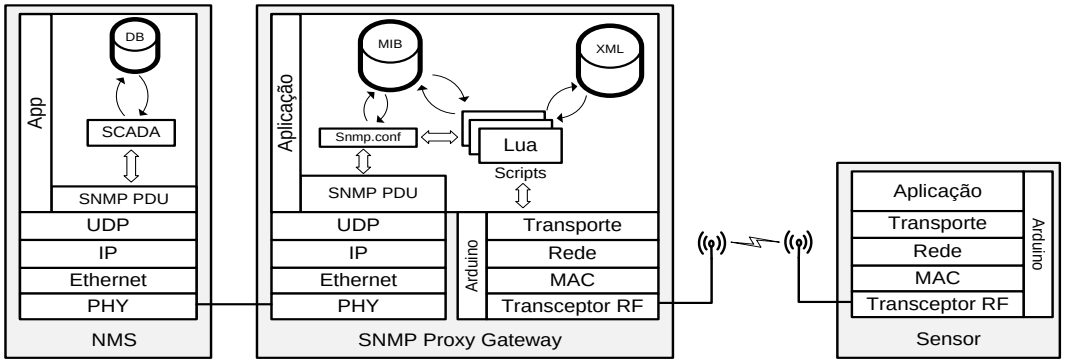

Figura 11: Visão geral dos principais componentes do sistema

A seguir, será detalhado cada componente de forma separada.

5.1 A *Network Management Station* - NMS

O primeiro componente principal é a Network Management Station (NMS) responsável pela administração centralizada das RSSFs e de todos os objetos gernciáveis sob sua administração. Nessa proposta, uma NMS pode conter um ou mais SPG sob seu domínio de gerência conforme mostrado na Figura 12 através de um modelo hierárquico. Esse modelo envolve a gerência dos recursos da rede, definição de rotas, gerenciamento de recursos dos nós sensores (bateria e processamento) e assim dois diante.

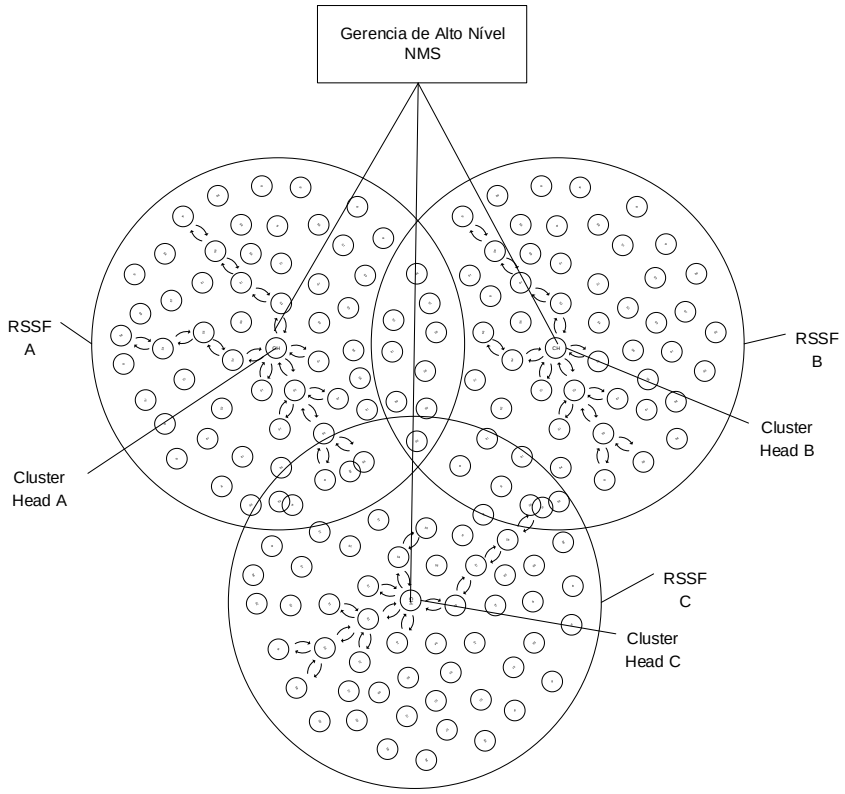

Figura 12: Modelo hierárquico de gerência para RSSFs

Além disso, A NMS é responsável pela gerência dos dados de usuário como a geração de relatórios analíticos, controle de ações proativas e reativas, controle de acesso aos componentes, disponibilização dos dados através de console gráfico por protocolo HTTP (web), etc. É função também da NMS disponibilizar o console de gerência para administração dos recursos da rede para os administradores da RSSF e para usuários.

A NMS proposta neste trabalho é baseada em um sistema de plataforma aberta seguindo o conceito de gerência para controle, supervisão e aquisição de dados, SCADA. Esse conceito é amplamente utilizado para automação de sistemas de comunicação máquina-a-máquina (M2M) e consolidado na indústria. A plataforma adotada foi o SCADABR. A escolha deste

ambiente foi devido a estar baseada em plataforma de software livre (Linux) e adotar de projetos *open source* como OpenWRT e Apache.

A base do sistema é suportada por um servidor de aplicação do projeto Apache Tomcat (AppServer), que tem a função de tradução dos componentes da plataforma desenvolvidos em Java™ numa *interface* amigável para o usuário sobre um servidor *web* (WebServer). O sistema ainda faz uso de um repositório de dados (DB) que armazena o conteúdo adquirido pelo processo de monitoramento. Além de possibilitar o armazenamento dos dados, estes podem ser utilizados na geração de relatórios estáticos ou dinâmicos, alteração de estado de um ou mais componentes e mapeamento dos objetos da RSSF através da interface gráfica de usuário ou *Graphical User Interface* (GUI).

A adoção de uma GUI é uma importante ferramenta na ajuda da classificação e categorização do tipo de dado que pode ser provido pela RSSF. Neste caso, é adotada uma distinção de um lado, dados da rede: referência a definição de rotinas para tratamento de falhas, análise dos atributos de desempenho e disponibilidade dos itens de configuração da RSSF; e dados de usuário: destinado para tratamento por regras de negócio como contabilidade e segurança. O fato de usar uma interface centralizada baseada no modelo de gerência proposto, é importante para manter uma categorização hierárquica do ponto de vista de gerência, para que os dados possam ser classificados ou categorizados de acordo com a afinidade da área conforme os pilares clássicos de gerência discutidos anteriormente.

No caso da leitura de um transdutor de temperatura de um nó sensor, por exemplo, o mais importante para o usuário final, é saber qual o valor da temperatura sem importar como essa operação de leitura da temperatura no nó sensor é realizada. Neste caso, a leitura é feita apenas lendo-se o valor referente ao respectivo OID que é associado pela MIB ao transdutor de temperatura de um nó sensor específico.

A seguir, será discutido como esse mecanismo funciona no SPG e de que forma ele foi implementado para fazer a integração do sistema de gerência baseado em TCP/IP para à RSSF.

5.2 O SNMP Proxy Gateway

Para a implementação do SPG do ponto de vista do modelo de camadas, foi feito uma equalização do modelo de referência OSI ao modelo desenvolvido para a plataforma Arduino adotada pelo SPG. O objetivo disso é facilitar o entendimento do modelo de múltiplas camadas para melhorar o desempenho pela programação intercamadas. A Figura 13 mostra o modelo adotado. Por esse modelo é possível, por exemplo, fazer o *overlayer* entre camadas para propor melhorias de desempenho sistêmico. Isso pode ser adotado, também, para futuros trabalhos de pesquisa sobre o tema. Neste instante, não é escopo de discussão neste trabalho.

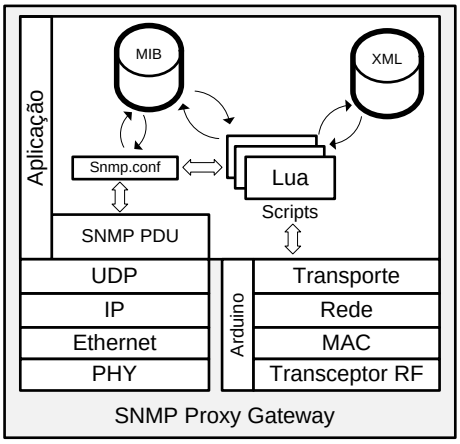

Figura 13: Componentes básicos do SPG

Em relação as interfaces de comunicação de rede, há basicamente dois tipos. A primeira interface é a Ethernet que faz a comunicação com a rede TCP/IP onde a suíte do protocolo SNMP é executada. A segunda interface é a interface aérea com o transceptor de RF a qual tem a função de comunicação com a RSSF. Alternativamente, há uma terceira interface que pode de utilizada. Essa interface segue o padrão WiFI e pode ser utilizada, por exemplo, para

facilitar o uso e a aplicação do SPG em qualquer tipo de ambiente. A razão disso é que, devido ao conjunto ser bastante compacto, pode-se fazer o gerenciamento remoto da NMS através de uma conexão sem fio e ainda oferecer uma implantação compacta e de fácil instalação. Adicionalmente, na plataforma também há portas do tipo Ethernet para conexão *wired* caso necessário o qual permite múltiplas entradas Ethernet. Assim, o SPG pode oferecer uma administração totalmente remota sem fio para a RSSF. Um protótipo para essa plataforma é apresentada na Figura 14.

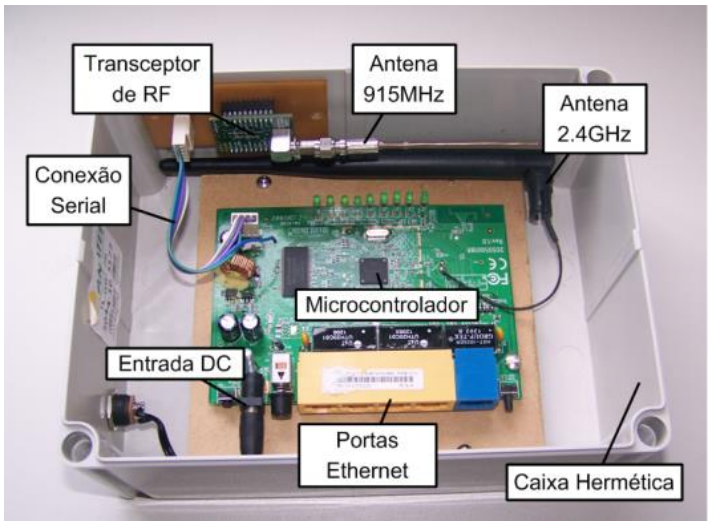

Figura 14: SNMP Proxy Gateway

Na Tabela 1 é apresentada as característica do hardware adotado para construção do SPG.

Tabela 1: Característica de Hardware do SPG	
CPU	Atheros AR9330
Clock CPU	400 MHz
RAM	32 MB
Flash	4 MB
NIC ETH	4x1
Porta Serial	Sim
Fonte Alimentação	9 VDC / 0,6A
Dimensões (LxCxA)	174x118x33mm

Para uso dessa plataforma como gateway entre as redes, foi utilizado um transceptor de RF da Texas Instrument modelo CC 1101. Na Tabela 2 é apresentada as característica do hardware desse transceptor.

Tabela 2: Característica de hardware do transceptor de RF	
CPU	AVR Atmega328
Clock CPU	8 MHz
RAM	2 kB
Flash	32 kB Flash
RTC	32768 kHz (±10 ppm)
Transceptor	CC1101 RF
Faixas de Frequências	902 a 907,5 MHz e 915 a 928 MHz
Modulação	2FSK (configurável)
Potência de TX	Até +10 dBm
Sensibilidade RX	Até -112 dBm (com ~1% de PER)
Regulamentação	FCC, Anatel e Austrália

A seguir, será discutido sobe a implementação do SPG como componente de interconexão das redes TCP/IP e RSSF passado pela construção dos scripts responsáveis pela montagem e encaminhamento do pacotes de gerência.

5.3 Implementação Sistêmica

A RSSF utilizada como protótipo desse trabalho, como mencionado anteriormente, é baseada numa estrutura *open source* e na plataforma Arduino. Assim, foi desenvolvido um pacote especifico para comunicação da RSSF.

Esse pacote é formado por 64 *bytes*, divididos em 12 *bytes* de cabeçalho e 52 *bytes* para *payload*. Cada *byte* tem uma posição especifica e cada posição possui determinados valores de acordo com sua finalidade. A Figura 15 mostra o mapa do pacote da RSSF utilizada nos testes dessa dissertação.

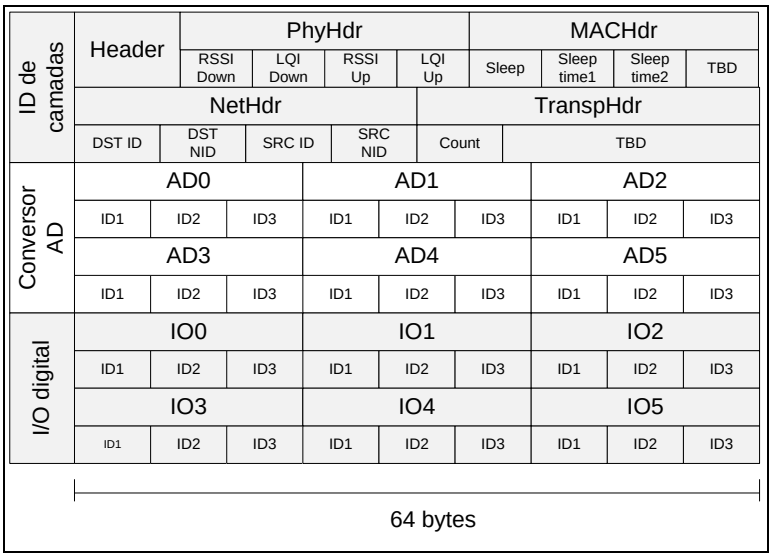

Figura 15: Mapa de bytes do pacote da RSSF

Os principais campos desse pacote são: Para os campos de identificação de rede, o campo DST_ID é responsável pelo armazenamento do endereço de destino do pacote dentro da RSSF. O campo SRC_ID é responsável pelo endereço de origem do pacote. Depois há seis campos para identificação dos conversores analógico/digital, de AD0 ao AD5, e mais seis campos de entrada/saída digital nomeados, de IO0 ao IO5. Há ainda, o *header* para identificação e manipulação da camada física nos nós sensores como potência de sinal de descida e subida (*downlink* e *uplink*) e tempo de hibernação do sensor (*sleep time*).

A comunicação com a RSSF é feita pelo SPG através de um conjunto de *scripts* desenvolvidos especificamente para essa função. Para isso, é necessário que seja formado o *payload* com um determinado estado de bit em cada posição. Este “estado de *bit*” é armazenado pelo sistema numa posição de memória para uso na construção do pacote de dados a ser transmitido para a RSSF. Para isso, foi feito uma implementação usando-se XML. A função do XML é ser utilizado como base de transição para armazenar o “estado de *bit*” antes que o pacote seja enviado para a RSSF. Assim, o *script* é executado pelo PDU do protocolo SNMP o qual gera dinamicamente o pacote e o envia para o nó

sensor da RSSF. Na Figura 16, está descrito parte do arquivo XML o onde é armazenado o estado dos dados.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Pacote>
        <Byte nome="RSSIDown">
                <Posicao>ID0</Posicao>
                <Valor>0</Valor>
        </Byte>
        <Byte nome="LQIDown">
                <Posicao>ID1</Posicao>
                <Valor>0</Valor>
        </Byte>
        <Byte nome="RSSIUp">
                <Posicao>ID2</Posicao>
                <Valor>0</Valor>
        </Byte>
        <Byte nome="LQIUp">
                <Posicao>ID3</Posicao>
                <Valor>0</Valor>
        </Byte>
        <Byte nome="Sleep">
                <Posicao>ID4</Posicao>
                <Valor>0</Valor>
</Byte>
(...)
        </Byte>
        <Byte nome="IO5_H">
                <Posicao>ID50</Posicao>
                <Valor>0</Valor>
        </Byte>
        <Byte nome="IO5_L">
                <Posicao>ID51</Posicao>
                <Valor>0</Valor>
        </Byte>
</Pacote>
```

Figura 16: Pacote de dados da RSSF em XML

Este XML contém basicamente 52 ‘*tags’* onde cada *tag* faz referência a uma posição de memória do pacote. Quando o sistema é iniciado, todas as posições do campo “valor” estão com zero. A medida que o sistema é executado, novos valores são preenchidos nesses campos os quais são, então, armazenados com o seu respectivo último “estado de *bit*” conforme a solicitação é executada pela NMS. Como exemplo, a primeira posição do pacote é o *byte* a qual a *tag* tem o nome de ‘*RSSIDown’*. Nessa *tag,* também é fornecida sua posição em relação ao pacote de dados e o seu respectivo valor do tipo decimal. A conversão decimal para binário é executada pelo *script* de transmissão e será visto mais à frente. Este mesmo mecanismo é adotado até a posição 52, onde o nome da *tag* é ‘IO5_L’ e sua posição é rotulada como ‘ID51’ e tem seu valor ‘0’

(zero). Uma vez definido o arquivo de dados XML, onde contém todas as informações do pacote a ser enviado para a RSSF, o *script* então executa as primitivas do SNMP para manipular esse arquivo e, assim, é construído um pacote de dados para envio para a RSSF.

Nessa implementação, foi utilizado o NetSNMP baseado numa distribuição Linux chamada OpenWRT. Essa é uma distribuição customizada e tem a vantagem de poder ser implementada em sistemas embarcados devido ao seu baixo consumo de memória e CPU, o qual, é capaz de rodar o NetSNMP com até 60% menos uso de recursos se comparado com sistemas fazem a implementam completa do protocolo SNMP. Para que seja possível que o PDU do SNMP possa fazer a execução do *script*, é necessário alterar a entrada do parâmetro ‘config exec’ no arquivo de configuração snmp.conf indicando o *path* com a opção ‘option prog’ e os respectivos parâmetros de execução.

Quanto a implementação dos scripts, eles foram totalmente desenvolvido em linguagem de programação Lua 5.1. Essa linguagem foi inicialmente desenvolvida pela PUC do Rio de Janeiro e já é bastante usada no meio acadêmico para desenvolvimento de prototipagem de sistema embarcados devido a ela ser extremamente robusta e de baixo consumo de recursos de memória da CPU. Seu interpretador consome menos que 270 *kbytes* de memória. Além disso, Lua é uma linguagem totalmente interpretada de fácil execução fazendo com que se torne ideal para rodar em sistemas embarcados.

A seguir, é apresentada parte do código desenvolvido na Figura 17.

```
#!/usr/bin/lua

-- Faz o include para o parser XML
dofile("/scripts/LuaXML/xml.lua")
dofile("/scripts/LuaXML/handler.lua")

-- Faz a leitura do arquivo de dados XML (pacote Radiuino)
local filename = "/scripts/pacote/pacote.xml"
local xmltext = ""
local f, e = io.open(filename, "r+w")

-- Passa o conteudo do arquivo para uma string
if f then
  xmltext = f:read("*a")
else
```

```
  error(e)
end

-- Funcao de parser XML
local xmlhandler = simpleTreeHandler()
local xmlparser = xmlParser(xmlhandler)
xmlparser:parse(xmltext)

-- Abre a serial em mode de escrita
serial=io.open("/dev/ttyATH0","w+d")

-- Faz a leitura do pacote de dados e escreve na serial
pct = ""
for k, p in pairs (xmlhandler.root.Pacote.Byte) do
        pct = pct .. string.char(tonumber(p.Valor))
end

serial:write(pct)

-- Forca limpar a serial
serial:flush()

-- Espera pelo pacote de 52 bytes de resposta
bytes = serial:read(52)

-- Verifica o pacote recebido
if bytes == nil or not bytes or not (string.len(bytes) == 52) then
 print("Erro ao receber o pacote")
 serial:flush()
else

-- Leitura do RSSIDown
RSSIDown = string.byte (bytes, 1)
if RSSIDown > 128 then
RSSIDown = ((RSSIDown - 256) / 2.0) - 74
 else
  RSSIDown =(RSSIDown / 2.0) - 74
  end

-- Leitura do RSSIUp
RSSIUp = string.byte (bytes, 3)
if RSSIUp > 128 then
RSSIUp = ((RSSIUp - 256) / 2.0) - 74
 else
  RSSIUp =(RSSIUp / 2.0) - 74
  end

-- Leitura da tensao da fonte
ad2t = string.byte (bytes, 23) -- tipo de transdutor: tensao da fonte
ad2h = string.byte (bytes, 24) -- Alto
ad2l = string.byte (bytes, 25) -- Baixo
 AD2 = ad2h * 256 + ad2l
VFonte = (0.003223 * AD2) * 11

-- Leitura da tensao da fonte auxiliar (bateria ou fonte externa solar)
ad3t = string.byte (bytes, 26) -- tipo de transdutor: tensao (bateria)
ad3h = string.byte (bytes, 27) -- Alto
ad3l = string.byte (bytes, 28) -- Baixo
 AD3 = ad3h * 256 + ad3l
 VBat = (0.003223 * AD3) * 11

-- Leitura da Temperatura
ad0t = string.byte (bytes, 17) -- tipo de transdutor: temperatura
ad0h = string.byte (bytes, 18) -- Inteiro
ad0l = string.byte (bytes, 19) -- resto
 AD0 = ad0h * 256 + ad0l
 Vout = 0.003223 * AD0
 Temp = (Vout * 100) - 53

-- Leitura do LDR (transdutor de luminosidade)
ad1t = string.byte (bytes, 20) -- tipo de sensor - no caso está medindo LDR
ad1h = string.byte (bytes, 21) -- alto
```

```
ad1l = string.byte (bytes, 22) -- baixo
 LDR = ad1h * 256 + ad1l

-- Imprime resultados
print (RSSIDown)
print (RSSIUp)
print (VFonte)
print (VBat)
print (Temp)
print (LDR)

end
```

Figura 17: Script em linguagem Lua

Na execução do *script*, inicialmente, é feito a inclusão do ParserXML através da função '*dofile*()'. Depois é criado algumas constantes a qual é passado o estado inicial do *path* onde está o arquivo de dados XML através da constante *local filename*. Logo após é feito a execução do ParserXML através do condição '*if*/*then'* e parâmetro *xmlparser*:*parse*(*xmltext*). Assim a porta de comunicação serial o qual contém o transceptor de RF na faixa de 915 MHz é preparada para receber o pacote com parâmetros de leitura e escrita. A leitura das posições correspondente a cada *byte* do pacote de dados é lida através do laço com o parâmetro *xmlhandler.root.Pacote.Byte*. Percebe-se que, neste momento, é feito a conversão decimal/binária através da função *string.char(tonumber(p.Valor))*. Logo depois, o pacote formado é enviado para o transceptor de RF através do parâmetro *serial:write(pct)*.

Quando o pacote chega ao transceptor de RF, o pacote é reconhecido por um conjunto de códigos desenvolvidos em plataforma Arduino, que envia o pacote para o destino dentro da RSSF a qual está designado o canal de comunicação. O pacote é enviado conforme orientação do campo DST_ID mencionado anteriormente. Caso seja utilizado um protocolo de roteamento da RSSF, o próximo sensor que receber o pacote faz a alteração do campo SRC_ID e o envia ao próximo nó até que o pacote chegue ao seu destino final. Quando o pacote chega ao seu destino final, é feita a leitura e/ou alteração dos campos AD e IO conforme seja necessário e, novamente, o pacote é remontado e retransmitido a sua origem no SPG, Novamente, o *script* faz a leitura do novo estado do pacote alterando seus valores no arquivo de dados XML e o envia

novamente através do PDU SNMP até a estação de gerência. Esse mesmo mecanismo é adotado para todas as RSSFs que participam do domínio de gerência da NMS e que utilizam do SPG como plataforma.

Com isso, foi coberto o processo de implantação do SPG, passando pela apresentação das características do hardware e software adotados, utilização de ferramentas e de linguagens de programação e pelo mecanismo de construção e encaminhamento do pacote para o gerenciamento da RSSF em conjunto com o protocolo SNMP.

A seguir, serão apresentados alguns resultados através de experimentos executados em laboratório que visam testar a implantação da proposta com o uso integral do sistema de gerência.

6. Experimentos e Resultados

Os experimentos foram executados no Laboratório de Pesquisa em Sistemas Rádio (LPSiRa) da Pontifícia Universidade Católica de Campinas (PUCCAMP) utilizando a proposta apresentada em conjunto com o protótipo do o sistema de gerência.

O primeiro experimento tem como objetivo validar a funcionalidade do sistema. Para isso foi montado um cenário envolvendo um nó sensor com transdutores associados às OIDs como componentes gerenciáveis da NMS e o SPG como integrador das redes. O segundo experimento teve como objetivo fazer análise do tipo de dado coletado na RSSF em relação ao sistema de gerência num ambiente de rede metropolitana. Neste caso, foi montado um cenário de uma RSSF composta de 6 nós sensores integrada à uma *Metropolitan Area Network* (MAN). A seguir serão discutidos os detalhes de cada experimento.

6.1 Experimento 1

Para o primeiro experimento, foi preparado em ambiente de laboratório com o objetivo de validar sua funcionalidade sistêmica. O ponto de avaliação é saber se uma requisição feita pela NMS através de um PDU a um transdutor do nó sensor é respondida com seu respectivo tempo de resposta. Com base nisso, um nó sensor na faixa de 915 MHz foi configurado na mesma canalização da interface aérea do transceptor de RF instalado no SPG. Os parâmetros utilizados para comunicação da base de RF (SPG) e o nó sensor estão descritos na Tabela 3.

Foi utilizado *pooling* como técnica de acesso ao meio na RSSF. O tempo de *pooling* é parametrizável na NMS e, neste experimento, este tempo foi configurado para 15 segundos entre as requisições. Isso significa que a cada 15 segundos uma nova requisição é feita pela NMS e enviada para o SPG através

de um PDU do tipo Get. Uma vez que a requisição seja recebida pelo SPG, o conjunto de *scripts* fazem o processamento e encaminhamento do pacote para o nó sensor para a leitura dos transdutores. O pacote é, então, devolvido ao SPG fazendo o caminho inverso. Este processo é repetido a cada 15 segundos.

Tabela 3: Parâmetros de configuração de RF

Transceptor	Texas CC 1101
Faixa de Frequência	915 MHz (ISM)
Portadora	919,2 MHz
Modulação	2FSK
Potência de Tx	+10 dBm
Sensibilidade de Rx	- 95 dBm
Corrente de pico	14,7 mA
Tipo de Antena,	Omnidirecional
Ganho da Antena	2 dBi
Taxa	9,6 kbps

No SPG foi instalado o protocolo NetSNMP e a *interface* Ethernet foi configurada para operar no mesmo segmento de rede TCP/IP da estação de gerência (NMS). A Figura 18 mostra o diagrama do cenário montado.

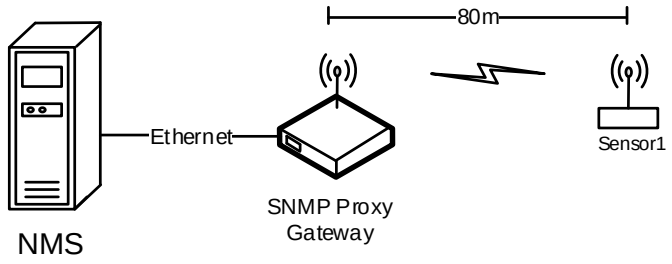

Figura 18: Cenário do experimento 1

Para cada transdutor do nó sensor testado, é associado um OID como componente gerenciável através do PDU do protocolo SNMP proposto pela MIB da Figura 9. Na Tabela 4 é mostrada essa associação. Foram feitas baterias de

testes para leitura dos transdutores para avaliação da resposta do sistema. Desta forma, foi estudado o tempo de resposta médio ou *Round Trip Time* (RTT) para cada requisição.

Tabela 4: Associação dos objetos gerenciáveis da RSSF	
.1.3.6.1.4.1.23955.1.1.1.100.1.2.1	RSSIUp
.1.3.6.1.4.1.23955.1.1.1.100.1.2.2	RSSIDown
.1.3.6.1.4.1.23955.1.1.1.100.1.2.7	Tensão da Bateria
.1.3.6.1.4.1.23955.1.1.1.100.2.2	Tensão da Fonte
.1.3.6.1.4.1.23955.1.1.1.100.2.3	Temperatura Local (LDR)
.1.3.6.1.4.1.23955.1.1.1.100.2.3	Luminosidade Local

O RTT foi testado em duas situações distintas: A primeira fazendo-se a coleta de um OID local no SPG, ou seja, apenas no segmento de rede TCP/IP. Essa primeira avaliação é usada apenas como parâmetro de referência e comparação com o resultado do RTT. A segunda forma é a execução completa da solicitação de gerência, ou seja, a NMS faz a solicitação de um dado ao nó sensor específico. Isso faz com que a coleta dos valores no nó sensor, seja executado usando-se a implementação do sistema através do SPG.

No SPG, o OID local testado foi 1.3.6.1.2.1.1.1 (SysDescr). Esse OID traz informação da descrição do sistema. No caso do OID da RSSF, foi testado o OID 1.3.6.1.4.1.23955.1.1.1.100.1.2.2 que traz a informação da intensidade de sinal (RSSIDown) do nó sensor.

Como resultado, para ambos os casos, houve resposta do OID requisitado pela NMS com tempo de resposta bastante característico. No caso da resposta do SysDescr o tempo médio de resposta foi de aproximadamente 0,3 ms enquanto que para o RSSIDown do nó sensor foi de aproximadamente 200 ms com base no ambiente preparado para este experimento. Uma explicação para essa diferença, está no tempo de preparação e encaminhamento no pacote do segmento da RSSF. Isso pode ser explicado conforme o cálculo do RTT dado a seguir

$$RTT = 2 * (PTT + PD + ProcD) \quad (1)$$

Onde:

PTT = *Packet Transmission Time*. É a quantidade de *bits* (tamanho do pacote) transmitido em razão da taxa do canal de transmissão em *bit/s*.

PD = *Propagation Delay*. Refere-se ao tempo em que o pacote leva para percorrer a distância do transmissor ao receptor em razão da velocidade de propagação do meio.

ProcD = *Processing Delay*. Valor estimado, pois, é dependente do contexto do ambiente de execução. Trata-se do tempo gasto para preparação e montagem do pacote e seu encaminhamento para o meio de transmissão. Essa estimativa deve ser calculada em cada sistema o qual o pacote é construído.

Para o cálculo do RTT total, deve-se considerar o RTT nos dois segmentos distintos: na rede Ethernet e na RSSF. Assim, o RTT_{total} é dado pela soma do RTT no segmento de rede Ethernet mais o RTT para a RSSF. Assim:

$$RTT_{total} = RTT_{ETH} + RTT_{RSSF} \quad (2)$$

Fazendo-se uma análise do RTT_{ETH}, é percebido que ele tem pouca influência no valor do RTT_{total} pois, considerando que o pacote padrão *Ethernet* contém 1.526 *bytes*, a taxa de transmissão é de 100 Mbps e a velocidade do meio (cobre) é de 180.000 km/s e, ainda, que a distância entre o transmissor e receptor é muito pequena (cerca de 10 m) o qual pode ser desprezível para efeitos de cálculo, e que o tempo de ProcD também pode ser considerado desprezível no segmento *Ethernet* (>> zero) devido alta capacidade de processamento do transmissor e do receptor para formação e encaminhamento do pacote, tem-se então que, desta forma, o cálculo teórico estimado do RTT_{ETH} com base em (1) é $RTT_{ETH} = 2*(PTT_{ETH})$. Assim, $RTT_{ETH} =$

2*(1.526*8bit/100*106bit/s) = 0,24 ms. Este valor está próximo do valor médio obtido pelo teste para o segmento *Ethernet*.

Já para o cálculo do RTT_{RSSF}, diferentemente de RTT_{ETH}, percebe-se que o valores de PTT_{RSSF} e do PD_{RSSF} influenciam o valor total de RTT_{RSSF}, pois, o pacote tem 64 bytes e a taxa de transmissão é de 9,6 kbps. Já para o PD_{RSSF} pode ser considerado como desprezível porque a velocidade do meio (ar) é de aproximadamente 300.000 km/s e a distância (80 m) entre o transmissor e o receptor é considerada pontual e desprezível (>> zero). Já para o $ProcD_{RSSF}$ foi calculado em 100 ms devido ao processo de formação do pacote e encaminhamento para o nó sensor. Essa estimativa foi obtida fazendo-se a execução local dos scripts no SPG. Assim, para o RTT_{RSSF}, temos que RTT_{RSSF} = 2*(64*8bit/9,6*10^3bit/s) + 100ms = 206,6ms. Ou seja, o RTT_{total} sofre uma grande influência do tempo de resposta da RSSF no ambiente testado. Uma razão disso, é o tempo de construção e encaminhamento do pacote na RSSF, pois, a execução dos *scripts* é, devido a sua característica de implementação, interpretado pelo compilador e não compilado como em C/C++ ou *assembler*. Assim, percebe-se que valor para RTT_{RSSF} é diretamente influenciado pelo $ProcD_{RSSF}$.

Desta forma, sinteticamente, é percebido que os OIDs respondem as requisições da NMS com destino à RSSF atendendo ao requisito de funcionalidade sistêmica, porém, o tempo de resposta é diretamente dependente do tempo de processamento e encaminhamento do pacote na RSSF. No entanto, mais importante nessa análise é saber não houve necessidade de reconstrução ou adaptação de nenhuma camada do protocolo TCP/IP ou mesmo necessidade de embarcar parte dessa pilha do nó sensor como apresentados em outros estudos como em [18], [19] e [20].

Em resumo, é sistema é funcional mas com desempenho inferior se comparado com sistema de gerência tradicional com o uso do SNMP. Por outro lado, é importante mencionar que, devida as característica de baixa necessidade de performance nas RSSF, o resultado inicial obtido é satisfatório numa avaliação obtida como prova de conceito. Uma evolução natural da proposta

pode ser esperada com o desenvolvimento de trabalhos futuros o qual pode ter como objetivo a melhoria da implementação em termos de desempenho sistêmico.

Outra análise quanto a funcionalidade sistêmica, é em relação a possibilidade de categorização das informações coletadas dos transdutores pelos OIDs, uma vez que o dado coletado tem a característica singular de prover a informação única para cada OID a ele associado ao transdutor. Assim, foram realizadas coletas dos dados pela estação de gerência considerando sua configuração padrão para sistemas supervisórios e aquisição de dados com o objetivo de hierarquizar a informação em gerência da rede e gerência dos dados de usuário.

Para isso, é necessário que OID efetue a coleta dos dados utilizando o campo *variable bindings* de forma unidimensional conforme padrão do protocolo e obedecer assim, o nível de organização hierárquica da gerência da RSSF. Isso é possível porque a saída do conjunto de scripts desenvolvidos fazem a associação através do encadeamento da saída de cada função, por exemplo, a saída da função *print* (RSSIDown) devolve o valor do seu respectivo OID. Desta forma, as informações foram reunidas e categorizadas através da GUI da NMS. O console principal dessa ferramenta é mostrado na Figura 19.

Neste caso é apresentado um console de *front-end* do sistema de supervisão e aquisição de dados sendo aplicado para a gerência de uma RSSF através do protocolo SNMP.

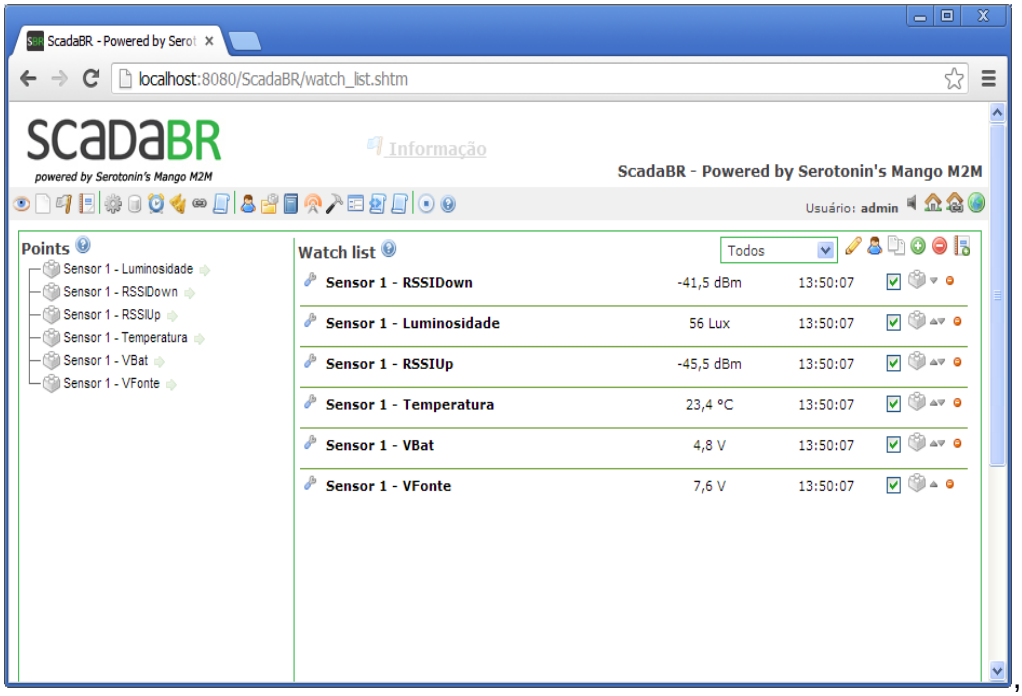

Figura 19: Console gráfico da ferramenta de gerencia baseada em SCADA

Na tela da Figura 19 são mostrados valores coletados pelo PDU tipo GET do nó sensor 1, Por exemplo, o valor do OID RSSIDown medido nó sensor é de -41 dBm (dado de gerência da rede) e o valor do OID de temperatura é 23,4 ºC (gerência do usuário). Esse mesmo mecanismo pode ser adotado para qualquer nó sensor que forma a RSSF podendo ser tratado pela estação de gerência e, consequentemente, pelo usuário final.

6.2 Experimento 2

O objetivo deste experimento é testar o sistema de gerência em um ambiente metropolitano analisando as respostas dos PDUs requisitados através da NMS.

O protocolo de acesso ao meio utilizado foi baseado na técnica de roteamento e *pooling*. Nessa técnica, o sinal de RF é enviado pelo transceptor

de RF do SPG e transmitido para a RSSF e, assim, somente o nó sensor de destino, com base do conteúdo do campo DST_ID, efetua o recebimento do pacote. Os demais nós sensores pertencentes a mesma canalização de RF, fazem a operação de descarte do pacote. Essa operação é, assim, repetida até que o pacote seja entregue ao nó sensor de destino. Neste caso, todos os pacotes percorrem a rota completa até o ultimo nó da RSSF. A Figura 20 mostra como os nós sensores foram dispostos no ambiente do laboratório para execução do experimento.

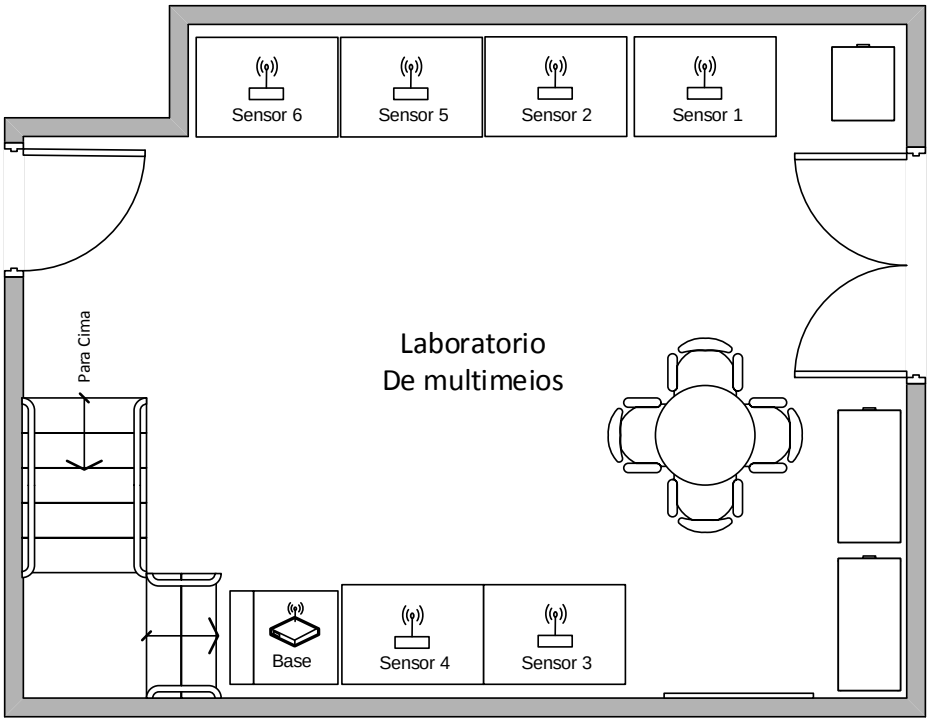

Figura 20: Disposição dos sensores no laboratório LPSiRa

Neste cenário, uma RSSF foi preparada utilizando um conjunto de 6 nós sensores dispostos de forma dispersa e não equidistante entre eles dentro do LPSiRa. O sensor 1 faz a função de *cluster head* onde todo o tráfego que chega à RSSF, através da base, passa por ele antes de ser transmitido ao próximo nó. Obviamente, pela limitação da quantidade de nós sensores, haverá somente uma rota disponível a ser percorrida, ou seja, o fluxo de dados flui do

sensor 1 ao sensor 6. Além disso, a RSSF foi conectada à um sistema de Rádio sobre Fibra (RoF) para emulação de uma *Metropolitan Area Network* (MAN). O objetivo de uso deste cenário numa MAN é emular o funcionamento do sistema de gerência numa rede geograficamente distribuída. Neste caso, o sistema de RoF é formado por um conjunto de fibras ópticas *single mode* (SM), o qual é alocado uma fibra óptica para transmissão (Tx) e outra para recepção (Rx) do sinal recebido pelo transceptor de RF. O comprimento linear total da bobina óptica é de 4 km.

A saída da *interface* aérea do SPG foi conectada a entrada do circulador através *pigtail* para desmembrar em sinal de Tx e Rx e entrar do sistema de RoF. Na ponta oposta do sistema de RoF há outro circulador cuja função é refazer a operação, ou seja, agregar o sinal de Tx e Rx na mesma canalização de RF dentro da faixa de 915 MHz.

A segunda interface do SPG está conectada a interface com porta *Ethernet* a qual faz parte do mesmo segmento de rede TCP/IP da NMS. Assim, uma vez o sistema conectado, as coletas dos dados de gerência dos nós sensores podem ser feitas através do console da NMS através SPG. Desta forma, foi preparado o cenário conforme mostrado na Figura 21.

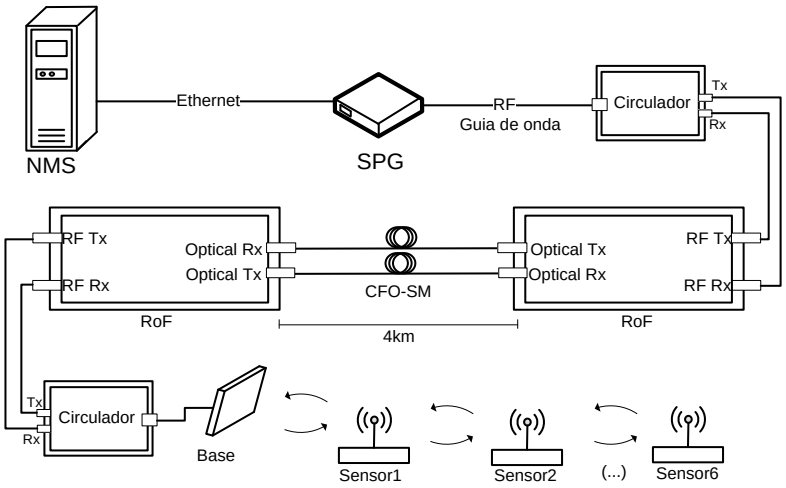

Figura 21: Cenário do experimento 2

Foram feitas diversas baterias de testes com duração variando entre 10 e 100 minutos. Foram coletados dados de nível de potência de descida (RSSIDown) e nível de potência de subida de sinal (RSSIUp). Os resultados dos testes serão discutidos a seguir.

Após algumas baterias de testes, observou-se que os resultados obtidos dos transdutores da RSSF de teste, utilizando-se o SPG, são semelhantes aos resultados obtidos diretamente dos PDUs do protocolo SNMP, sem o uso do SPG o que, inicialmente, está coerente com o esperado em termos de funcionalidade. Também foi possível fazer a coleta de forma individual das informações tanto em nível de gerência da rede quanto em nível de gerência dos dados como intensidade de sinal e temperatura, respectivamente, corroborando com a organização de gerência o qual está aderente com o pilares de gerência.

Para os dados de gerência da rede foi gerado um gráfico comparativo sobre as informações de potência de sinal (RSSI) entre os nós sensores que participaram do experimento proposto. Um exemplo deste tipo de gráfico está apresentado na Figura 22.

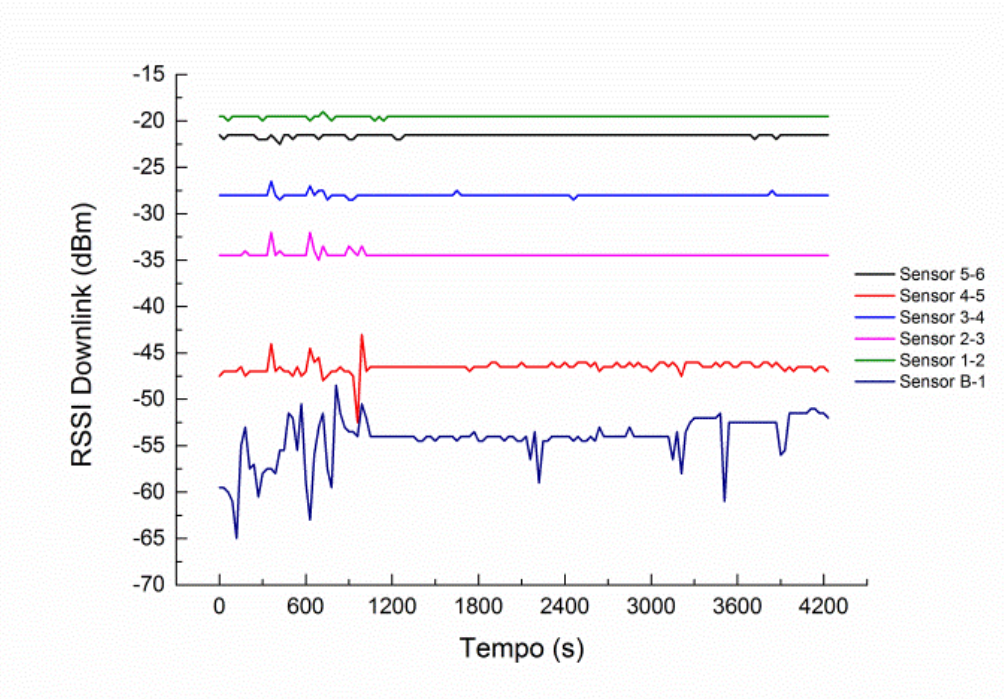

Figura 22: Intensidade de sinal de RF dos nós da RSSF

Pelo gráfico, é apresentado o nível de potência de sinal, em dBm, transmitido de um sensor A para um sensor B (RSSI Downlink) obtidos durante uma amostra de execução do teste por um período de aproximadamente 90 minutos de forma contínua. Devido a formação aleatória dos nós sensores no laboratório, percebe-se diferentes níveis de intensidade de sinal. Por exemplo, o nível médio de intensidade de sinal transmitido da base ao sensor 1 é de aproximadamente -55 dBm, enquanto o nível médio de intensidade de sinal transmitido do sensor 1 para o sensor 2 é de aproximadamente -20 dBm. Essa diferença é observada devido as distâncias entre os sensores indicando que o sensor 2 está, comparativamente, mais próximo ao sensor 1 do que este de sua base e, de fato, essa condição é comprovada pela dispersão dos nós sensores no laboratório conforme indicado no diagrama de disponibilização dos nós sensores no laboratório.

Finalmente, para um contexto mais abrangente, um benefício importante dessa característica de sistema de gerência, é que ela também pode ser explorada pelo gestor da RSSF para que se possa fazer a reorganização do conjunto de nós sensores uma vez que se tem a visão global da rede. Neste caso, por exemplo, a decisão de eleger um novo *cluster head* ou adoção de uma diferente rota para acesso a base pode ser provida através dos PDUs de gerência à RSSF.

A decisão de reeleição de um *cluster head* com o objetivo de fazer uma rota alternativa ao nó de destino pode ser tomada através da estação de gerência alterando-se as métricas de rotas dos campos DST_ID e enviado pelo PDU do protocolo SNMP.

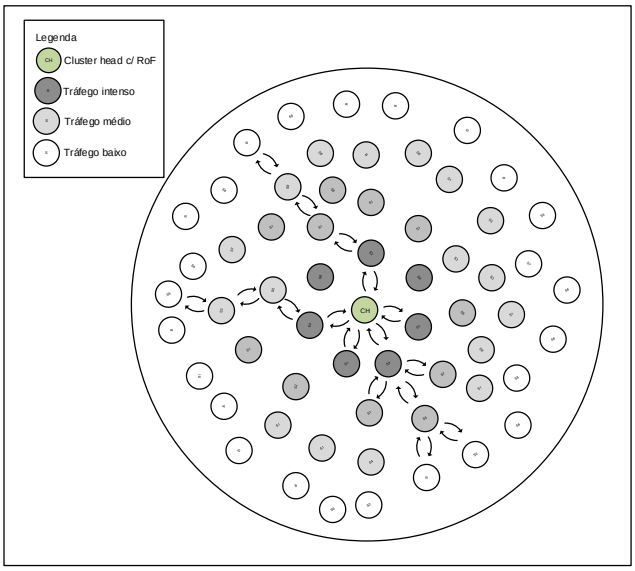

Figura 23: Exemplo de RSSF com múltiplas rotas

A Figura 23 mostra um exemplo de uma RSSF com possibilidade de múltiplas rotas onde essa situação pode ser aplicada. Já há trabalhos avançados desenvolvidos e em andamento sobre tema.

Desta forma, com a apresentação dos experimentos, observou-se que um sistema de gerência aplicado às RSSFs com o uso do protocolo SNMP, é possível através de um sistema que faça a interligação entre as duas redes. A vantagem deste tipo de implementação, além do baixo custo, é que não há necessidade de alteração das características fundamentais de nenhuma das redes. Assim, é possível manter cada rede com suas peculiaridades, além de manter o legado das redes TCP/IP e oferecer um sistema de gerência integrada para as RSSFs.

7. Conclusão

Neste trabalho foi proposta uma implementação para sistema e gerência para RSSF com base no protocolo SNMP. A vantagem dessa abordagem é que ela mantém o legado do sistema de gerência adotado para as redes TCP/IP tradicionais sem a necessidade de alteração ou adequação da pilha para uso nos nós sensores. Dois experimentos foram aplicados para validação da funcionalidade e de sua aplicação em ambiente de redes metropolitanas. Nesses experimentos foram coletos dados de RSSI entre nós sensores da RSSF para validar os resultados.

Desta forma, há indicação de viabilidade de gerência da RSSF através do protocolo SNMP com um sistema para integração através de um SNMP Proxy Gateway, mantendo-se as características básicas das duas redes: a redes TCP/IP com uso integral de sua pilha de protocolos e a RSSF, mesmo com a limitação de recursos como processamento e memória que são características das RSSFs.

Como perspectiva para trabalhos futuros, é importante que o sistema proposto seja testado com maior número de nós sensores. Isso é importante para refletir maior tráfego de dados na RSSFs o que torna mais realista para ambientes de redes metropolitanas que, naturalmente, têm uma maior densidade de dispositivos gerenciáveis.

Bibliografia

[1] W. Dargie e C. Poellabauer, Fundamentals of Wireless Sensor Networks, Theory and Pratice, United Kingdom: John Wiley & Sons Ltd, 2010.

[2] D. P. Bavirisetti, N. P. Mandru e S. Khara, “Optimal Power Management In Wireless Sensor Networks For Enhanced Life Time,” *Journal of Global Research in Computer Science,* p. Volume 3 no. 4, Abril 2012.

[3] L. H. A. Correia, D. F. Macedo, A. L. Santos, J. M. Nogueira e A. A. F. Loureiro, “Uma Taxonomia para Protocolos de Controle de Acesso ao Meio em Redes de Sensores Sem Fio,” 2005.

[4] W. L. Lee, A. Datta e R. Cardell-Oliver, Network Management in Wireless Sensor Networks, 2006.

[5] L. B. Ruiz, T. R. M. Braga, F. A. Silva, H. P. Assunção, J. M. A. Nogueira e A. A. F. Loureiro, “On the Design of a Self-Managed Wireless Sensor Network,” *Self-Organization In Networks Today - IEEE Communications Magazine,* pp. 95-102, 2005.

[6] D. Evans, “The Internet of Things: How the Next Evolution of the Internet,” *Cisco Internet Business Solutions Group,* Abril 2011.

[7] M. M. Alam, M. Mamun-Or-Rashid e C. S. Hong, “WSNMP: A Network Management Protocol for Wireless Sensor Networks,” Gyeonggi, 2008.

[8] A. Jacquot, J.-P. Chanet, K. M. Hdu, G. D. Sousa e A. Monier, “A New Management Method for Wireless Sensor Networks,” *IEEE*

IFIP Annual Mediterranean Ad Hoc Networking Worshop, 23-25 Junho 2010.

[9] S. Gowrishankar, T. G. Basavaraju, D. Manjaiah e S. K. Sarkar, “Issues in Wireless Sensor Networks,” em *WCE - World Congress on Engineering*, London, U.K., 2008.

[10] J. Case, M. Fedor, M. Schoffstall e J. Davin, “A simple network management protocol,” em *RFC 1157*, 1990.

[11] K. McCloghrie e M. Rose, “Management information base for network management of tcp/ip-based internets: MIB-II,” em *RFC 1213*, 1991.

[12] W. Stallings, SNMP, SNMPv2, SNMPv3, and RMON 1 and 2, Addison-Wesley, 1999.

[13] W. Chen, N. Jain e S. Singh, “Anmp: Ad hoc network management protocol,” *IEEE Journal,* 1999.

[14] M. Hossen, B. Jang, K. Kim e Y. Park, “Extension of Wireless Sensor Network By Employing RoF-based 4G Network,” 2009.

[15] C. Shen, C. Jaikaeo, C. Srisathapornphat e H. Huang, “The guerrilla management architecture for ad hoc networks,” *MILCOM ,* 2002.

[16] A. Sethi, D. Zhu e P. Kalyanasundaram, “Shaman - an environment for distributed management applications,” *Integrated Network anagement Proceedings,* 2001.

[17] Y. Ma, J. Chen, Y. Huang e M. Lee, “An Efficient Management System for Wireless Sensor Networks,” *MDPT Journal,* 2010.

[18] K. Ushalnagar, N. Montenegro e G. Schumacher, “IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement and Goals,” em *RFC 4919*, 2007.

[19] N. Montenegro, G. Kushalnagar, N. Hui e J. Culler, “Transmission of IPv6 Packets over IEEE 802.15.4 Networks,” em *RFC 4944*, 2007.

[20] A. Ludovici, A. Calveras e J. Casademont, “Forwarding Techniques for IP Fragmented Packets in a Real 6LoWPAN Network,” em *Wireless Network Group (WNG)*, Mòdul, 2011.

[21] R. Hummen, J. Hiller, H. Wirtz, M. Henze, H. Shafagh e K. Wehrle, “6LoWPAN Fragmentation Attacks and Mitigation Mechanisms,” em *Communication and Distributed Systems, RWTH Aachen University*, Germany, 2013.

[22] L. Steenkamp, “Wireless sensor network monitoring using the Simple Network Management Protocol,” em *Centre for Instrumentation Research, Cape Peninsula University of Technology*, 2012.

[23] D. Evans, “The Internet of Things. How the Next Evolution of the Internet is Changing Everything,” *Cisco White Paper,* Abril 2011.

[24] f. S. Cyriaco, “Wireless Sensor Network managemant Using the SNMP Paradign,” Campinas, 2012.