

RAUL PASTRELLO SILVA

Telemetria de estações retransmissoras de TV

Raul Pastrello Silva

Telemetria de estações retransmissoras de TV

Trabalho de Graduação apresentado ao Conselho de Curso de Graduação em Engenharia Elétrica da Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, como parte dos requisitos para obtenção do diploma de Graduação em Engenharia Elétrica.

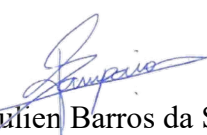
Orientador (a): Prof. Thiago José Michelin
Coorientador (a): Prof. Dr. Leonardo Mesquita

S586t	Silva, Raul Pastrello Telemetria de estações retransmissoras de TV / Raul Pastrello Silva - Guaratinguetá, 2022. 91 f : il. Bibliografia: f. 84-86 Trabalho de Graduação em Engenharia Elétrica – Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, 2022. Orientador: Prof. Dr. Thiago José Michelin Coorientador: Prof. Dr. Leonardo Mesquita 1. Televisão - Estações. 2. Sensoriamento remoto. 3. Transmissão de imagem. I. Título. CDU 528.8
-------	--

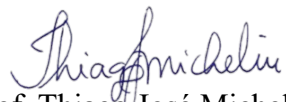
RAUL PASTRELLO SILVA

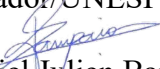
ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO
PARTE DO REQUISITO PARA A OBTENÇÃO DO DIPLOMA DE
“GRADUADO EM ENGENHARIA ELÉTRICA”

APROVADO EM SUA FORMA FINAL PELO CONSELHO DE CURSO DE
GRADUAÇÃO EM ENGENHARIA ELÉTRICA


Prof. Daniel Julien Barros da Silva Sampaio
Coordenador

BANCA EXAMINADORA:


Prof. Thiago José Michelin
Orientador/UNESP-FEG


Prof. Dr. Daniel Julien Barros da Silva Sampaio
UNESP-FEG


Prof. Dr. Samuel Euzédice de Lucena
UNESP-FEG

Julho, 2022

Dedico este trabalho,
de modo especial, à minha família e à minha
namorada, Evelyn, que me deram todo o apoio
para seguir no curso.

AGRADECIMENTOS

Em primeiro lugar agradeço a Deus, fonte da vida e da graça. Agradeço pela minha vida, minha inteligência, minha família e meus amigos;

Ao meu orientador, *Prof. Thiago José Michelin*, que não desistiu do projeto. Sem a sua orientação, dedicação e auxílio, o estudo aqui apresentado seria impossível;

Aos meus pais, *Francisco e Denise*, que apesar das dificuldades enfrentadas, sempre incentivaram meus estudos e me forneceram tudo para terminar o curso de Engenharia;

Aos funcionários e principalmente aos professores da Faculdade de Engenharia do Campos de Guaratinguetá, pela dedicação e comprometimento com a passagem do conhecimento.

“O importante é não parar de questionar. A
curiosidade tem sua própria razão de existir”;
Albert Einstein

RESUMO

O presente trabalho tem como objetivo a criação de um sistema de telemetria e telecomando para as estações retransmissoras de televisão (RTV). Para o desenvolvimento do sistema, foi utilizada a estrutura de uma grande emissora de TV localizada na cidade de São Paulo. Tal emissora possui uma rede de estações retransmissoras em todo estado, que tem por objetivo difundir a programação mesmo aos telespectadores que não estão localizados nos grandes centros. Cada estação retransmissora possui equipamentos capazes de receber, processar e retransmitir o sinal, para uma cidade ou mesmo região. O sistema desenvolvido será capaz de realizar o envio remoto das principais leituras dos equipamentos, sendo apresentadas em uma plataforma gráfica de fácil visualização. Além disso, será possível a execução de comandos remotos para a alteração de certos parâmetros de funcionamento dos equipamentos. O sistema proposto irá diminuir os custos com deslocamentos desnecessários de técnicos ao local, além de auxiliar no gerenciamento da rede de retransmissão de TV, evitando interrupções de sinal nas cidades e facilitando a manutenção dos postos pelos técnicos e engenheiros da emissora.

PALAVRAS-CHAVE: Sistema. Telemetria. Telecomando. TV. RTV. Emissora. Rede. Retransmissoras.

ABSTRACT

This work proposes a telemetry and telecommand system for television relay stations (RTV). For the development of the system, the structure of a large TV station located in the city of São Paulo was used. This station has a network of retransmitting stations throughout the state, whose objective is to spread the programming even to viewers who are not located in large centers. Each relay station has equipment capable of receiving, processing and retransmitting the signal, for a city or even region. The developed system will be able to carry out the remote sending of the main parameters of the equipment, being presented in a graphic platform of easy visualization. In addition, it will be possible to execute remote commands to change certain operating parameters of the equipment. The proposed system will reduce costs with unnecessary displacements of technicians to the site, in addition to assisting in the management of the TV relay network, avoiding signal interruptions in cities and facilitating the maintenance of stations by the station's technicians and engineers.

KEYWORDS: System. Telemetry. Remote control. TV. RTV. Broadcaster. Network. Relays.

LISTA DE ILUSTRAÇÕES

Figura 1 - Concepção da arquitetura TCP/IP.....	18
Figura 2 - Camadas da arquitetura TCP/IP.....	19
Figura 3 - Principais componentes de uma arquitetura de gerenciamento de rede.	24
Figura 4 - Arquitetura do protocolo SNMP.....	25
Figura 5 – Estrutura em árvore da MIB.....	26
Figura 6 - Arquitetura da MIB.....	27
Figura 7 - Estação geradora.	29
Figura 8 - Cabo coaxial com conector BNC.	29
Figura 9 - Exemplo: PMT "200".	31
Figura 10 – Transmissão terrestre – TV Digital.....	33
Figura 11 - Transmissão via satélite – TV Digital.....	34
Figura 12 - Diagrama de retransmissão de sinais.	35
Figura 13 - Antena de recepção de satélite – Banda C.....	36
Figura 14 - Receptor de satélite para aplicações profissionais.....	37
Figura 15 - Funcionamento de um receptor de satélite profissional.....	37
Figura 16 - Transmissor de UHF.....	38
Figura 17 – Torre e antenas de transmissão de uma estação retransmissora de televisão.....	40
Figura 18 - Diagrama de blocos simplificado de um sistema telemetria.....	41
Figura 19 - Sistema de telemetria sem fio.	42
Figura 20 - Esquemático do projeto proposto.	45
Figura 21 - Raspberry Pi Model 3B+.	47
Figura 22 - Agente SNMP corretamente instalado no Raspberry Pi.....	48
Figura 23 - Requisitos para a instalação do Zabbix Server.	53
Figura 24 – <i>Frontend</i> do Zabbix pronto para uso pós instalação bem-sucedida.....	54
Figura 25 - Host do receptor de satélite.....	55
Figura 26 - Host do transmissor UHF.	56
Figura 27 - Criando um item no Zabbix Server.....	57
Figura 28 - Serviço Zabbix Proxy corretamente instalado no Raspberry Pi.	58
Figura 29 - Instância sendo executada na plataforma <i>Amazon Web Services</i>	59
Figura 30 - Definição do nível C/N ou SNR.	60
Figura 31 - Itens para a monitoração do receptor de satélite.....	64
Figura 32 - Itens para a monitoração do transmissor.....	64

Figura 33 - Verificando disponibilidade SNMP dos hosts criados.	65
Figura 34 - Valores dos parâmetros retornados ao servidor (receptor).	65
Figura 35 - Valores dos parâmetros retornados ao servidor (transmissor).	66
Figura 36 - <i>Frontend</i> para a monitoração da estação retransmissora.	67
Figura 37 - Exemplo de comando utilizando a biblioteca PySNMP.	69
Figura 38 - Fluxo de funcionamento do programa desenvolvido.	69
Figura 39 - Exemplo da recepção e envio dos comandos.	70
Figura 40 - Bot de envio de mensagens ao Raspberry Pi.	70
Figura 41 - Resultado da operação no transmissor.	71
Figura 42 - Bancada de testes do sistema.	72
Figura 43 - Bancada de testes do sistema.	72
Figura 44 - Diagrama esquemático da bancada de testes.	73
Figura 45 - Diagrama do sistema de telemetria para instalação em campo.	75
Figura 46 - Interior do abrigo da retransmissora.	76
Figura 47 - Receptor e transmissor da estação retransmissora.	76
Figura 48 - Esquemático do sistema de telemetria montado em campo.	77
Figura 49 - Sistema de telemetria instalado e em funcionamento.	78
Figura 50 - Monitoração do sinal de referência GPS no transmissor durante 1 mês.	79

SUMÁRIO

1	INTRODUÇÃO	13
2	FUNDAMENTAÇÃO TEÓRICA.....	15
2.1	A REDE DE COMPUTADORES E A INTERNET	15
2.2	ARQUITETURA TCP/IP.....	17
2.2.1	A família de protocolos TCP/IP	18
2.2.2	Camadas da arquitetura TCP/IP	19
2.2.2.1	Camada de aplicação	19
2.2.2.2	Camada de transporte	20
2.2.2.3	Camada de rede.....	20
2.2.2.4	Camada de interface de rede.....	20
2.3	GERENCIAMENTO DE REDES	21
2.3.1	Objetivos do gerenciamento de redes	22
2.3.2	Arquitetura de gerenciamento de uma rede de computadores	22
2.4	PROTOCOLO DE GERENCIAMENTO SNMP.....	24
2.4.1	Arquitetura do protocolo SNMP.....	25
2.5	TEORIA SOBRE REDES DE TELEVISÃO	28
2.5.1	Estação geradora	28
2.5.1.1	SDI (<i>Serial digital interface</i>)	29
2.5.1.2	Codificador (<i>encoder</i>).....	30
2.5.1.3	ASI (<i>Asynchronous serial interface</i>)	30
2.5.1.4	TS-MPEG <i>Transport stream</i>	30
2.5.1.5	<i>Program specific information</i> (PSI)	31
2.5.1.6	Multiplexador	31
2.5.1.7	Modulador.....	32
2.5.1.8	ISDB-T (<i>Integrated Services Digital Broadcasting Terrestrial</i>)	32
2.5.1.9	DVB-S2 (<i>Digital Video Broadcasting – Satellite – Segunda geração</i>)	33
2.5.2	Retransmissora (RTV)	34
2.5.2.1	Antena de recepção via satélite.....	35
2.5.2.2	Receptor de satélite.....	36
2.5.2.3	Transmissor.....	37
2.5.2.4	Antena de transmissão	39
2.6	TELEMETRIA	40

2.6.1	Componentes de um sistema de telemetria	41
2.6.2	Sistema de telemetria em estações retransmissoras de TV	42
3	TEMA DE ESTUDO	43
4	DESENVOLVIMENTO.....	46
4.1	RASPBERRY PI	46
4.2	INSTALAÇÃO DO PROTOCOLO SNMP NO RASPBERRY PI	47
4.3	ZABBIX SERVER	48
4.3.1	Funcionalidades do Zabbix.....	49
4.3.2	Arquitetura do Zabbix	50
4.3.2.1	<i>Templates</i> e itens no Zabbix	51
4.3.2.2	Zabbix e SNMP	51
4.3.2.3	Compatibilidade do Zabbix	52
4.3.3	Instalação do Zabbix Server	52
4.3.4	Aquisição do Zabbix Server.....	52
4.3.5	Pré-requisitos para instalação do Zabbix.....	52
4.3.6	Instalação via pacotes	53
4.3.7	Criando <i>host</i> SNMP no Zabbix Server.....	54
4.3.8	Configuração dos templates.....	56
4.3.8.1	Criando um <i>template</i> e adicionando itens.....	56
4.4	INSTALAÇÃO DO ZABBIX PROXY NO RASPBERRY PI.....	57
4.5	PLATAFORMA AMAZON WEB SERVICES (AWS)	58
4.6	OBTENDO DADOS PARA A MONITORAÇÃO DE UM DISPOSITIVO	60
4.6.1	Relação portadora/ruído (C/N)	60
4.6.2	Nível de recepção	61
4.6.3	Presença de sinal de radiofrequência.....	61
4.6.4	Presença de sinal de referência GPS.....	61
4.6.5	Potência direta	62
4.6.6	Potência refletida	62
4.6.7	Presença de sinal ASI na entrada do transmissor	62
4.6.8	Temperatura do modulador	63
4.7	MONITORAÇÃO DOS RECURSOS.....	63
4.8	PLATAFORMA GRAFANA.....	66
4.9	EXECUÇÃO DE COMANDOS VIA TELEGRAM	68
5	IMPLEMENTAÇÃO	72

5.1	PRÉ-IMPLEMENTAÇÃO EM BANCADA	72
5.2	IMPLEMENTAÇÃO EM CAMPO	74
6	ANÁLISE DOS RESULTADOS	79
6.1	MONITORAÇÃO EM BANCADA	79
6.2	MONITORAÇÃO EM CAMPO	81
7	CONCLUSÃO.....	82
	REFERÊNCIAS.....	84
	APÊNDICE A – Código da aplicação para telecomandos via Telegram	87

1 INTRODUÇÃO

Apesar da crescente substituição por serviços de *streaming* a televisão ainda tem suma importância na vida das famílias brasileiras. Segundo o último censo do IBGE, mais de 12 milhões de famílias brasileiras não possuem acesso à *internet* e, nessa realidade, a televisão tem papel fundamental ao trazer entretenimento e informação à população.

Hoje, a televisão é um item comum na casa dos brasileiros. Mas isso só aconteceu porque em 18 de setembro de 1950 nascia a primeira emissora de tevê do país com a primeira transmissão no Brasil, feita pela Rede Tupi de Televisão, a TV Tupi, a primeira estação da América do Sul e a quarta do mundo, de responsabilidade do jornalista e empresário Assis Chateaubriand, do Diários Associados. (CORREIO BRAZILIENSE, 2020)

A televisão ao passar dos anos passou por diversas mudanças, estando hoje presente em mais de 90% dos domicílios brasileiros. Nesse contexto, as estações retransmissoras de sinal têm papel fundamental em difundir o conteúdo das emissoras para os lugares mais isolados do país.

As retransmissoras de televisão (RTVs) começaram a ser implantadas há quase 40 anos e possuem papel fundamental no diz respeito a interiorização da programação de TV no Brasil. Uma retransmissora tem como função permitir que mesmo o público mais distante da estação geradora possa receber o sinal com qualidade, proporcionando ao cidadão entretenimento e informação para seu dia. Sem a retransmissão do sinal, o conteúdo gerado pelas emissoras de televisão ficaria restrito aos grandes centros urbanos.

Como aponta Görgen (2011), essa malha invisível de distribuição de bens simbólicos é mantida até hoje, em parte, graças ao suporte das prefeituras dos pequenos e médios municípios. De acordo com informações do Sistema de Controle de Radiodifusão (SRD) da Agência Nacional de Telecomunicações (Anatel), existem 9.927 estações autorizadas a prestar serviço de retransmissão em todo o Brasil.

Dentro de um número tão expressivo de estações retransmissoras e visto sua importância para as populações locais, é inegável a importância de um monitoramento constante por parte das geradoras de conteúdo (grandes emissoras de TV), garantindo assim que uma cidade ou mesmo toda uma região coberta por seu sinal difundido.

Muitas vezes, a geradora do sinal fica refém da prefeitura ou de telespectadores locais para informar se seu sistema de retransmissão está operando corretamente. Além disso, pode

demorar dias para que essa informação chegue ao setor de engenharia da empresa, acarretando em reclamações acumuladas que podem vir a tornar-se cobranças a todos os funcionários.

Assim sendo, esse trabalho se propõe a criar um sistema de monitoramento remoto, ou telemetria, robusto e de alta disponibilidade, com o objetivo de auxiliar técnicos e engenheiros de televisão a diagnosticar e possivelmente resolver problemas nas estações retransmissoras sem a necessidade do deslocamento de um técnico, diminuindo custos e melhorando o controle sobre a cadeia de distribuição do sinal da geradora do sinal.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 A REDE DE COMPUTADORES E A INTERNET

Os primeiros passos da rede de computadores que conhecemos hoje foram dados no início da década de 1960. Três estudantes – sem que um soubesse do trabalho do outro – trabalharam no estudo da técnica de comutação de pacotes, que é a base para a comunicação entre dois computadores geograficamente isolados.

Leonard Kleinrock, que, na época, era aluno do MIT, desenvolveu o primeiro trabalho a ser publicado sobre a comutação de pacotes. Ele se baseou na teoria de filas e conseguiu provar com elegância o poder da abordagem da comutação de pacotes para fontes intermitentes de sinal.

Em 1964, no Rand Institute, Paul Baran começou sua pesquisa sobre a transmissão da voz humana para fins militares e, ao mesmo tempo, Donald Davies e Roger Scantlebury focavam no mesmo tema no National Physical Laboratory, na Inglaterra.

Esses trabalhos serviram de alicerce para o desenvolvimento das redes de computadores como conhecemos hoje. As pesquisas seguiram, e ainda na década de 60, J. C. R. Licklider e Lawrence Roberts, ambos colegas de Kleinrock no MIT, publicaram o desenho geral daquela que viria a ser a ancestral da Internet pública dos dias atuais, a ARPA-net, sendo essa a primeira rede desenvolvida se baseando em técnicas de comutação de pacotes.

Em 1969, foi instalado o primeiro dispositivo comutador de pacotes, na UCLA (Universidade da Califórnia em Los Angeles), sob a supervisão de Kleinrock. Em 1972, a ARPA-net possuía 15 nós, e o primeiro protocolo fim a fim entre sistemas da rede (chamado de NCP – *network-control protocol*), estava concluído. Com o protocolo disponível para uso, diversas aplicações começaram a ser desenvolvidas, entre elas o e-mail – por Ray Tomlinson, da BBN - em 1972.

A ARPA-net era uma rede isolada, sendo que para conectar-se a ela era necessário estar conectado fisicamente à um dispositivo da rede. Na década de 1970, diversas redes foram surgindo, todas baseando-se na comutação de pacotes. Dentre elas, pode-se citar a ALOHAnet (rede de micro-ondas que interligava universidades das ilhas do Havaí), as redes via rádio da DARPA, a Telenet (rede comercial da BBN baseada na tecnologia ARPA-net), entre outras redes.

Nesse ponto era necessário desenvolver algum protocolo para interconectar as redes, que não paravam de surgir. O primeiro trabalho sobre o tema foi patrocinado pela DARPA (Defense Advanced Research Projects Agency — Agência de Projetos de Pesquisa Avançada de Defesa). Assim, através dos estudos de Vinton Cerf e Robert Kahn foi criada uma rede de redes. A partir deste ponto também começou a se utilizar o termo *internetting*.

Esses princípios de arquitetura foram incorporados ao TCP. As primeiras versões desse protocolo, contudo, eram muito diferentes do TCP de hoje. Elas combinavam uma entrega sequencial confiável de dados via retransmissão por sistema final (que ainda faz parte do TCP de hoje) com funções de envio (que hoje são desempenhadas pelo IP). As primeiras experiências com o TCP, combinadas com o reconhecimento da importância de um serviço de transporte fim a fim não confiável, sem controle de fluxo, para aplicações como voz em pacotes, levaram à separação entre IP e TCP e ao desenvolvimento do protocolo UDP. Os três protocolos fundamentais da Internet que temos hoje — TCP, UDP e IP — estavam conceitualmente disponíveis no final da década de 1970. (KUROSE & ROSS, 2013, Cap. 1, p. 46)

Metcalf e Boggs (baseando-se na ALOHAnet) lançaram as bases para as LANs de PCs de hoje, quando, em 1976, desenvolveram o protocolo Ethernet, com o objetivo de conectar diversos dispositivos, como computadores, impressoras, discos compartilhados etc.

Ao final da década de 1970, cerca de 200 máquinas estavam conectadas à ARPAnet. Ao final da década de 1980, o número de máquinas ligadas à Internet pública, uma confederação de redes muito parecida com a Internet de hoje, alcançaria cem mil. A década de 1980 seria uma época de formidável crescimento. (KUROSE & ROSS, 2013, Cap. 1, p. 46)

No dia 1º de Janeiro de 1983 o protocolo TCP/IP foi adotado como padrão para as máquinas da ARPA-net.

Na transição de 1989 e 1990, Tim Berners-Lee do CERN (European Center for Nuclear Physics — Centro Europeu para Física Nuclear) baseou-se em estudos anteriores sobre hipertextos para criar a World Wide Web, marcando o evento mais importante da década para as redes de computadores e a internet. Berners-Lee e seus companheiros desenvolveram as primeiras versões de HTML, HTTP, um servidor Web e um navegador (browser) — os quatro componentes fundamentais da Web.

No final da década de 90 a internet já dava acesso a centenas de aplicações, entre elas quatro principais: o *e-mail*, a *Web*, serviço de mensagens instantâneas e compartilhamento peer-to-peer de arquivos *mp3*.

A partir dos anos 2000, a distribuição da Internet no mundo caminhou a passos largos. Além da conhecida internet DSL (*Digital Subscriber Line*, a famosa “internet discada”), com o

passar do tempo, diversas empresas começaram a oferecer Internet de alta velocidade para o público em geral.

Esse desenvolvimento e distribuição rápida da internet pelo mundo abriu caminho para o surgimento de sites de vídeos sob demanda, como *YouTube* e *Netflix*, bem como para a propagação das redes sociais e serviços de vídeo conferência. Também possibilitou o acesso e gerenciamento remoto de diversos dispositivos que agora não mais precisariam estar conectados à mesma rede, podendo estar distribuídos em vários lugares do mundo.

2.2 ARQUITETURA TCP/IP

Essa arquitetura foi pensada para interconectar os centros de computação das universidades americanas. A alternativa da época (década de 1960), era a comunicação via linhas telefônicas dedicadas.

Esse projeto foi concebido no Departamento de Defesa dos EUA e é conhecido como modelo *Department of Defense* (DoD). A documentação do projeto foi elaborada pela comunidade acadêmica americana envolvida no desenvolvimento do projeto. A proposta da rede inovou em muitos aspectos os conceitos de rede existentes na época e foi tão bem aceita que perdura até os dias de hoje, mantendo sua concepção original nas linhas gerais. (ELIAS & LOBATO, 2013, Cap. 2, p. 59)

A arquitetura foi inicialmente a solução encontrada pela RAND Corporation na época da Guerra Fria, para um possível caos na comunicação entre autoridades americanas caso houvesse uma guerra nuclear. A ideia principal era relativamente simples: Cada nó da rede era igual em termos das funções que executava, e tinha como princípio “receber e passar adiante” as mensagens.

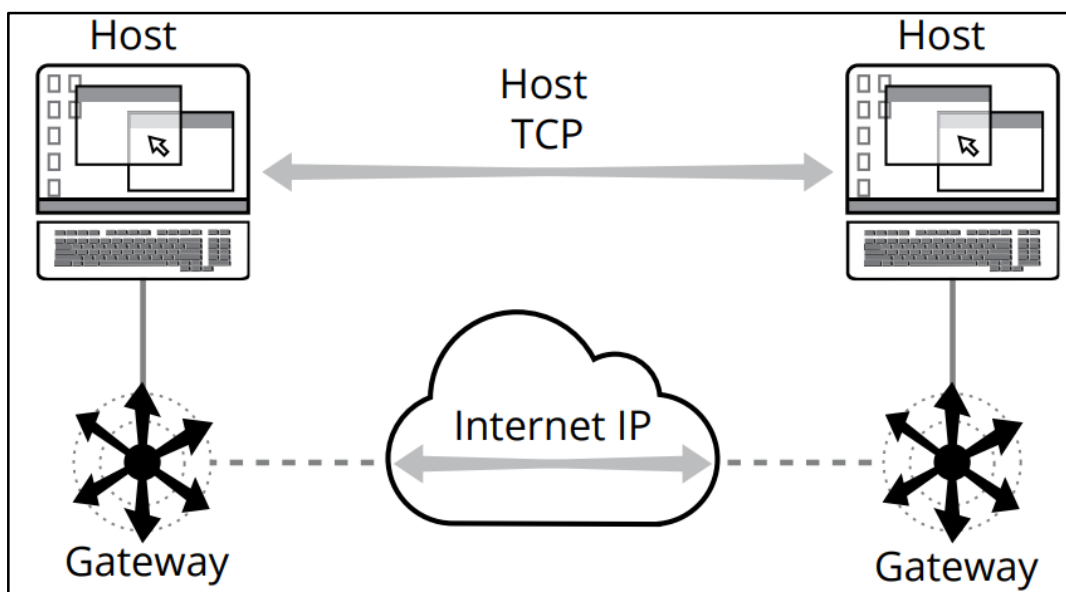
As mensagens seriam divididas em pacotes, cada pacote seria endereçado individualmente. Um pacote deveria trafegar pela rede, saindo de um ponto inicial e chegando à um ponto final, não importando o caminho que fizesse para isso. A rede de testes da arquitetura TCP/IP foi a ARPAnet, sendo as especificações dos protocolos definidas através dos documentos chamados RFCs - Request for Comments (Solicitação de Comentários), que se tornariam os padrões universais.

2.2.1 A família de protocolos TCP/IP

Se define como um conjunto de protocolos que permite a comunicação de dispositivos de diferentes tecnologias e de diferentes fabricantes, além de redes com diferentes tecnologias de acesso. É importante deixar claro que a utilização da família de protocolos TCP/IP não depende da Internet, e qualquer organização que queira interconectar suas diversas redes pode adotar essa família de protocolos.

Os usuários enxergam a inter-rede como uma rede virtual única à qual todos os dispositivos estão conectados, independentemente da forma das conexões físicas. Para isso, uma inter-rede TCP/IP adota um mecanismo de endereçamento universal, baseado em endereços IP, que permite a identificação única de cada dispositivo. (ELIAS & LOBATO, 2013, Cap. 2, p. 59)

Figura 1 - Concepção da arquitetura TCP/IP.



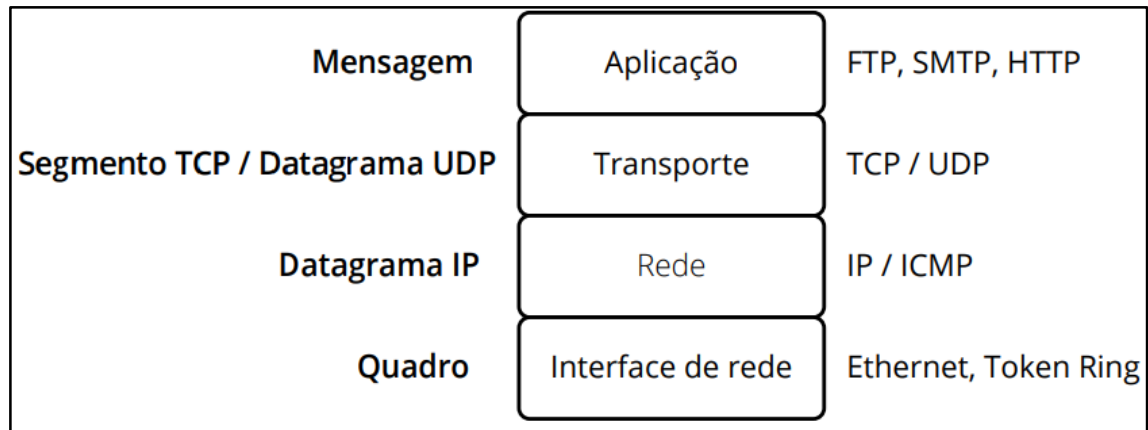
Fonte: Elias & Lobato (2013).

A figura 1 representa de forma simplificada a ideia da família de protocolos TCP/IP. Os dois protocolos principais são o TCP e o IP. O protocolo TCP, que é padronizado pela RFC 793, faz a função da camada de transporte, sendo nomeado pelos projetistas da Internet como protocolo fim-a-fim, ou “Host to Host Protocol”.

O protocolo IP, protocolo inter redes ou Internet Protocol é padronizado pela RFC 791, realizando a função da camada de rede. O gateway é o responsável por conectar o usuário à internet. O gateway pode ser fisicamente representado por um roteador, sendo este definido como um dispositivo que está conectado à duas ou mais redes.

2.2.2 Camadas da arquitetura TCP/IP

Figura 2 - Camadas da arquitetura TCP/IP.



Fonte: Elias & Lobato (2013)

2.2.2.1 Camada de aplicação

A camada de aplicação é onde se encontram as aplicações de rede e seus respectivos protocolos. A camada de aplicação da Internet é composta por diversos protocolos, como:

- *HTTP*: Possibilita a requisição e transferência de documentos pela Web;
- *SMTP*: Possibilita a transferência das mensagens de serviços de e-mail;
- *FTP*: Possibilita a transferência de arquivos entre dois dispositivos fim-a-fim.
- *SNMP*: Protocolo padrão para monitoramento e gerenciamento de redes.

Um pacote de informação deve sair da camada de aplicação de um sistema final e chegar à camada de aplicação de outro sistema final, através de um dos protocolos de aplicação. Esse pacote de informação recebe o nome de mensagem.

Ficará evidente pela definição da arquitetura TCP/IP, que, para que haja a comunicação efetiva entre dois dispositivos distintos são necessárias diversas etapas que são realizadas pelos protocolos de cada camada. Um dos protocolos utilizados para o gerenciamento de redes é o *SNMP (Simple Network Management Protocol)*, que atua na camada de aplicação e será estudado posteriormente nesse trabalho.

2.2.2.2 Camada de transporte

A camada de transporte é responsável por transmitir mensagens (recebidas da camada de aplicação) entre dois itens da rede. Nessa camada existem dois protocolos: TCP e UDP, sendo que qualquer um dos dois é capaz de transportar as mensagens da camada de aplicação. O primeiro, TCP, fornece uma gama mais completa de serviços, como:

- Entrega garantida de mensagens da camada de aplicação ao destino
- Controle de fluxo (compatibilização de velocidades entre receptor e transmissor)
- Controle de congestionamento
- Fragmentação de mensagens longas em segmentos mais curtos

Já o UDP é um protocolo mais enxuto focado em segurança, não fornecendo serviços de controle de fluxo e congestionamento. Um pacote na camada de transporte é chamado de segmento.

2.2.2.3 Camada de rede

Quase sempre denominada apenas camada IP (*Internet Protocol*), uma vez que nesta camada está incluído o famoso protocolo IP. O protocolo IP é responsável por enviar, receber e direcionar os pacotes entre transmissor e receptor. Os pacotes da camada de rede são chamados datagramas.

Um hospedeiro pode ser entendido como qualquer dispositivo conectado à rede. Um dos protocolos da camada de transporte (TCP ou UDP) em um hospedeiro de origem provê a camada de rede um pacote (segmento) e o endereço do destinatário (assim como um e-mail possui seu conteúdo e um endereço de destino). Em seguida, a camada de rede fornece o serviço de entrega do segmento à camada de transporte no hospedeiro de destino.

2.2.2.4 Camada de interface de rede

É a camada responsável por aceitar os datagramas IP e repassá-los utilizando a rede física disponível. Faz a ponte entre a tecnologia da rede física e o protocolo IP.

Geralmente, esta camada inclui o driver de dispositivo no sistema operacional e a respectiva placa de rede, tratando os detalhes de hardware para conexão física com a rede e transmissão de dados no meio físico. Assim, podemos dizer que a camada de interface de rede é basicamente suportada pela própria tecnologia da rede física. (ELIAS & LOBATO, 2013, Cap. 2, p. 64)

2.3 GERENCIAMENTO DE REDES

O gerenciamento, independentemente da área, deverá controlar as atividades e os recursos, bem como assegurar a segurança, confiabilidade e alta disponibilidade de um sistema.

As redes de computadores estão em constante evolução. Assim sendo, torna-se cada dia mais necessário que o ambiente de redes de computadores seja gerenciado, mantendo o mesmo funcional o máximo de tempo possível. Com o aumento da dependência das redes por parte das empresas, decidir não gerenciar os elementos de rede pode ser uma escolha muito arriscada, possivelmente gerando – em algum momento - prejuízos financeiros às corporações.

O administrador de rede deve possuir as ferramentas e o conhecimento necessário para detectar anomalias em dispositivos conectados à uma rede. Isso permite que se controle uma rede reativamente, isto é, realizando ajustes conforme possíveis modificações ocorram no ambiente e tomar decisões que evitem que problemas mais sérios surjam.

As funções básicas do gerenciamento de uma rede devem ser:

- Gerenciamento de falhas: Verifica quando ocorrem falhas e disponibiliza funções de análise e correção.
- Gerenciamento de configuração: Caso ocorram grandes mudanças na rede, deverá entregar ao administrador ferramentas para a configuração dos recursos lógicos e físicos.
- Gerenciamento de desempenho: Deverá fornecer o monitoramento do desempenho da rede, permitindo controlar os fatores que mantêm a qualidade de cada serviço.
- Gerenciamento de contabilização: Investiga quais e quanto dos recursos estão em uso, determinando um custo associado à utilização do mesmo.
- Gerenciamento de segurança: Deve garantir a segurança que previamente foi definida para a rede.

2.3.1 Objetivos do gerenciamento de redes

O gerenciamento de uma rede permite a melhor utilização dos recursos e a diminuição do tempo de indisponibilidade de um serviço e/ou equipamento. Auxilia os gestores a tomarem atitudes que evitem que um dispositivo fique fora de funcionamento, e, caso o mesmo pare de funcionar, também permite obter dados rápidos para formular uma estratégia e resolver o incidente.

A gerência de uma rede se torna mais necessária de acordo com o tamanho e a complexidade da mesma. Dispositivos podem estar conectados próximos ou não da área de atuação do administrador, sendo assim é de extrema importância que haja um controle preventivo dos recursos dos equipamentos em rede.

O gerenciamento de dispositivos em rede pode ser entendido em três fases bem definidas. São elas:

- Obtenção de dados: Processo - automatizado ou não – que tem por objetivo obter métricas do recurso gerenciado.
- Diagnóstico: Através das informações recebidas, os dados são tratados e analisados. Os gestores do sistema monitorado ou o software automatizado devem agora determinar a causa do problema no dispositivo.
- Atuação: Assim que diagnosticado o problema, o administrador do sistema deverá executar uma ação – através das ferramentas disponíveis – sob os recursos monitorados, com a finalidade de sanar os problemas que estão ocorrendo.

2.3.2 Arquitetura de gerenciamento de uma rede de computadores

Essencialmente, o gerenciamento de uma rede de computadores funciona entre dois sistemas interconectados, um remoto que é chamado de hospedeiro, e um local, geralmente chamado de servidor. O hospedeiro é o dispositivo ou sistema que se deseja gerenciar, e o servidor é a central na qual as informações serão recebidas, analisadas e tratadas.

Diz-se que dois computadores estão interconectados se são capazes de trocar informações. A conexão não precisa ser via fio de cobre; fibra óptica, micro-ondas, infravermelho e satélites de comunicação também podem ser usados. As redes vêm em muitos tamanhos, formas[...] (TANENBAUM & D WETHERALL, 2017, Cap. 1, p. 2).

A arquitetura de um sistema de gerenciamento de uma rede de computadores pode ser comparada ao gerenciamento de uma organização humana, possuindo três componentes principais. A primeira é a entidade gerenciadora ou servidor, que em uma organização humana pode ser definida pelo chefe, que deverá coletar e analisar as informações recebidas. O segundo componente são os dispositivos gerenciados ou agentes, que na analogia pode ser representado pelas filiais de uma empresa. E por fim, o terceiro é o protocolo de gerenciamento, que é a forma com a qual as filiais passarão as informações à entidade gerenciadora.

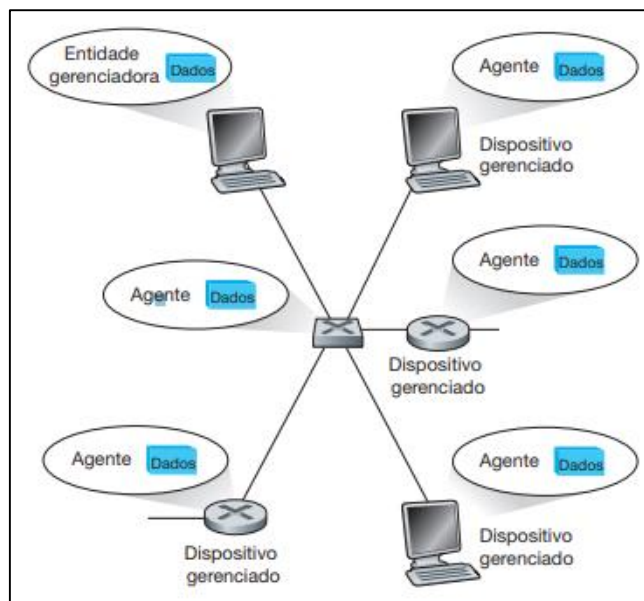
Passando ao universo do gerenciamento de redes, a entidade gerenciadora é um componente que geralmente possui um ser humano em seu circuito. Esse componente é executado em uma central de gerenciamento de rede, que pode ser a sede de uma empresa, por exemplo. Deve coletar e controlar as informações recebidas. Em uma entidade gerenciadora o humano tem em suas mãos as ferramentas para controlar o funcionamento dos recursos da rede.

O dispositivo gerenciado é um equipamento na rede gerenciada.

Um dispositivo gerenciado é um equipamento de rede, pode ser um hospedeiro, um roteador, uma ponte, um hub, uma impressora ou um modem. No interior do dispositivo gerenciado pode haver diversos objetos gerenciados. Estes são, na verdade, as peças de hardware...e os conjuntos de parâmetro de configuração para as peças de hardware e software...Esses objetos gerenciados têm informações associadas a eles que são coletadas dentro de uma Base de Informações de Gerenciamento MIB (*Management Information Base*). (KUROSE; ROSS, 2013, Cap. 9, p. 559)

Já o protocolo de gerenciamento é um componente da arquitetura de gerenciamento que é executado entre a entidade gerenciadora e o agente de gerenciamento dos dispositivos da rede. O protocolo fornece à entidade gerenciadora uma ferramenta com a qual pode investigar o estado de cada dispositivo da rede e inclusive executar ações sobre eles.

Figura 3 - Principais componentes de uma arquitetura de gerenciamento de rede.



Fonte: Kurose & Ross (2013).

2.4 PROTOCOLO DE GERENCIAMENTO SNMP

Conforme explicado no item 2.2.2 deste trabalho, a arquitetura TCP/IP possui camadas, cada uma com seus respectivos protocolos. Na camada de aplicação, existe um protocolo responsável por carregar informações entre o sistema de gestão e os dispositivos gerenciados, o SNMP (*Simple Network Management Protocol* – Protocolo Simples de Gerenciamento de Rede).

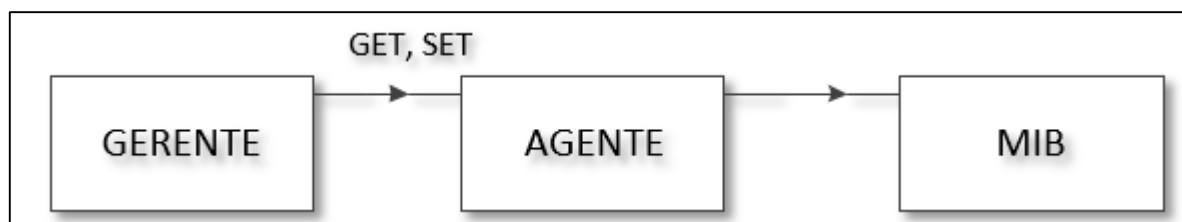
O SNMP é um protocolo de gestão de redes definido pela IETF (Internet Engineering Task Force), que facilita a troca de informações entre dispositivos de uma rede, como impressoras e switches. A partir do protocolo é possível gerenciar e controlar equipamentos de redes, assim detectando possíveis problemas e erros que podem impactar no funcionamento de componentes vitais de uma organização.

Para Kurose & Ross (2013), o SNMP é usado para transmitir informações e comandos entre uma entidade gerenciadora e um agente que os executa em nome da entidade dentro de um dispositivo de rede gerenciado.

2.4.1 Arquitetura do protocolo SNMP

Toda e qualquer aplicação de gerência de redes se baseia no conceito de troca de informações entre um agente e um gerente.

Figura 4 - Arquitetura do protocolo SNMP.



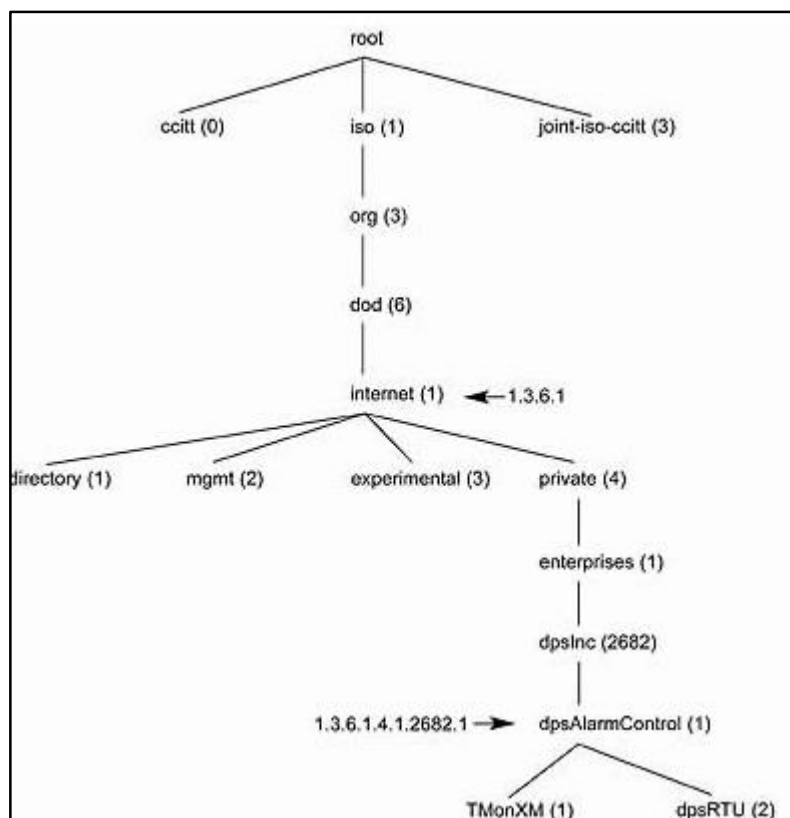
Fonte: Elaborado pelo autor.

Gerente: É executado em sua maioria em centrais de gerenciamento, ou plataformas de gerenciamento, que podem ser executadas em servidores online, por exemplo. Nele, as informações sobre cada dispositivo são recebidas através dos agentes, e por fim processada e apresentada aos administradores para que possam controlar o funcionamento dos objetos gerenciados.

Agente: São os responsáveis por manter as informações de funcionamento dos dispositivos através de suas respectivas MIBs (Management Information Base). O agente também pode agir sobre o objeto, caso o gerente solicite.

MIB: A MIB nada mais é do que um tipo de árvore hierárquica de informações de um determinado objeto gerenciável na rede (como roteadores, computadores, switches etc), desde que estes sejam compatíveis com o protocolo SNMP. As MIBs contém informações de configuração desses dispositivos, como a versão do software que está rodando no equipamento, o espaço livre em disco, o endereço IP, entre outros.

Figura 5 – Estrutura em árvore da MIB.



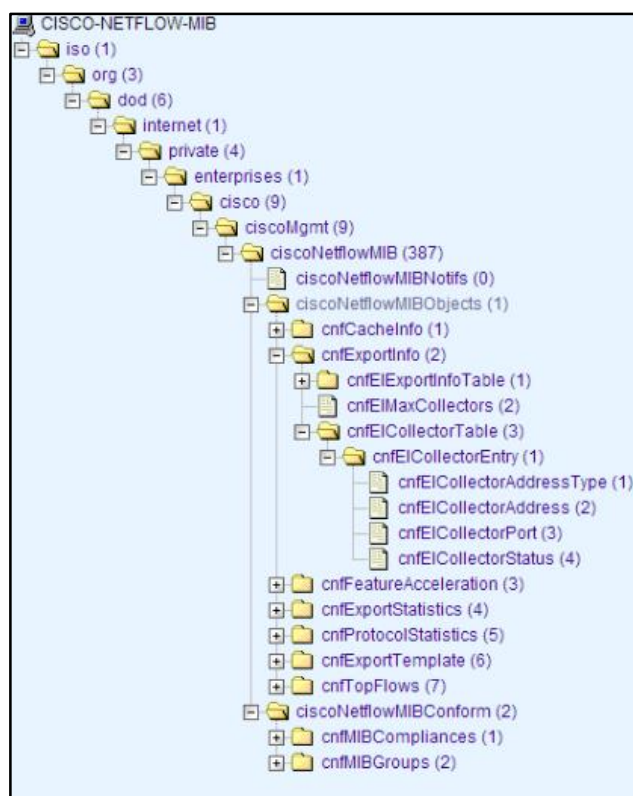
Fonte: DPS Telecom (2022).

Quando um gerente SNMP deseja saber o valor de um determinado parâmetro de um dispositivo da rede, como o nome do equipamento, o tempo de funcionamento ou até mesmo sua temperatura, ele irá montar um pacote que inclui um identificador único referente ao objeto de interesse.

Cada parâmetro ou característica de um dispositivo gerenciável SNMP possui um identificador único de objeto (OID), que consiste em números separados por pontos decimais que apontam para o valor de um parâmetro específico do agente SNMP.

O Protocolo SNMP não define quais informações (quais variáveis) um sistema de gestão deve oferecer. Esse dispõe de um design extensível no qual a informação disponível é definida pelas bases de informação do gerenciamento (MIBs). As MIBs descrevem a estrutura de gestão dos dados de um subsistema de um dispositivo por meio do uso de denominações hierárquicas contendo os identificadores dos objetos (OID). A grosso modo, cada OID identifica uma variável que pode ser lida ou escrita via SNMP. (UFRJ, 2022).

Figura 6 - Arquitetura da MIB.



Fonte: TELCO Manager (2018).

Logo, para acessar por exemplo o objeto “cnfElCollectorAddressType”, presente na figura 6 acima, seria utilizada a OID desse objeto “1.3.6.1.4.1.9.9.387.1.2.3.1.1”, que é seu identificador único.

O agente SNMP irá receber a requisição e buscar por essa OID na base de informações do dispositivo no qual ele está instalado. Caso a OID seja encontrada, um pacote de resposta é montado e enviado com o valor atual do parâmetro solicitado. Se, por outro lado, a OID não for encontrada, o pacote é montado e enviado com uma resposta de erro, identificado que esse parâmetro específico não pode ser monitorado.

O protocolo SNMP possui basicamente duas operações: Get (Obter), Set (Definir). A primeira operação permite que o gerente recupere informações de um objeto gerenciado informando a OID referente à informação que deseja obter.

Já a segunda – Set – permite que a entidade gerenciadora defina o valor de um parâmetro dentro dos objetos MIB do dispositivo gerenciado.

Geralmente, o SNMP utiliza as portas 161 para o agente e 162 para a entidade gerenciadora. O gerente pode enviar as solicitações para o agente a partir de qualquer porta disponível (porta de origem), tendo como destino a porta 161 do dispositivo gerenciado.

O protocolo SNMP, ainda, faz uso de um tipo de serviço de autenticação e controle de acesso as variáveis MIB. A partir das comunidades (SNMP Community), cada mensagem enviada do gerente ao agente será identificada, definindo seus limites de acesso aos subconjuntos dos objetos de uma MIB, podendo especificar também o modo de acesso (READ-ONLY – apenas leitura ou READ-WRITE – leitura e escrita).

O protocolo SNMP é amplamente utilizado pelos fabricantes e gestores de rede, pois é robusto e bem definido. Com a gerência de rede via SNMP é possível agilizar a análise e tomada de decisões facilitando e muito a vida dos analistas.

2.5 TEORIA SOBRE REDES DE TELEVISÃO

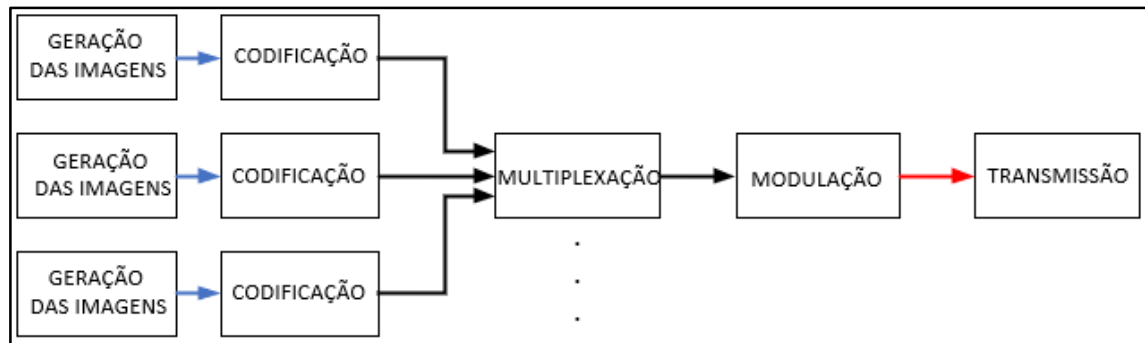
O objeto de estudo deste trabalho é o gerenciamento e monitoramento remoto da rede de estações retransmissoras de TV de uma empresa do ramo localizada na cidade de São Paulo. Para começar a entender sobre a problemática citada no início deste trabalho, se faz necessário um estudo sobre o que é, e como funciona uma rede de emissora de televisão.

As emissoras de televisão funcionam em rede, por meio do Sistema de Transmissão e Recepção de Televisão Broadcasting. Tal rede é subdividida em 2 partes: Estação geradora e retransmissoras.

2.5.1 Estação geradora

É a estação principal, onde se concentra todo o aparato técnico para geração da programação. Neste local os programas de TV são gravados e editados, além de ser gerada a programação ao vivo. Aqui também fica localizada toda a estrutura técnica de distribuição do sinal, que, após sua produção, deverá ser codificado, multiplexado, modulado e transmitido, por fim chegando à casa dos telespectadores. A figura 7 mostra através de um diagrama simples a cadeia de geração e distribuição de sinais dentro de uma estação geradora de TV.

Figura 7 - Estação geradora.



Fonte: Elaborado pelo autor.

2.5.1.1 SDI (*Serial digital interface*)

É o padrão utilizado pelas emissoras de televisão para conectar câmeras e dispositivos de edição. É utilizado para transmitir sinais digitais de vídeo sem compressão ou codificação, podendo ou não conter áudio na mesma cadeia de bits.

Para a transmissão, são utilizados cabo coaxiais e conectores BNC. Com essa tecnologia é possível transmitir em resolução HD em 60 *fps* (frames por segundo), a uma taxa de 1,485Gbit/s de dados. A tecnologia SDI oferece baixa latência na transmissão de dados, além de praticamente não sofrer perdas de pacotes em cabos de até 100 metros.

Figura 8 - Cabo coaxial com conector BNC.



Fonte: Haivision (2022)

Por transmitir sinais não codificados, a tecnologia SDI é utilizada para quando se necessita imagens em tempo real, sem atrasos, uma vez que se elimina a necessidade de um codificador (e decodificador) para a transmissão dos sinais. Por isso, é muito aplicada nas emissoras de televisão para as conexões de câmeras à mesa de corte em transmissões ao vivo, por exemplo.

2.5.1.2 Codificador (*encoder*)

É o responsável por receber o sinal serial SDI com a programação da emissora e em seguida fazer a codificação e compressão do sinal no padrão MPEG-2 ou MPEG-4. A saída do codificador será um sinal contendo um pacote de informações do programa específico.

2.5.1.3 ASI (*Asynchronous serial interface*)

É um protocolo para a transmissão de vídeo codificado em MPEG Transport Stream. Um sinal ASI opera em uma taxa fixa máxima de 270Mbits/s e pode carregar em uma mesma transmissão diversos programas em SD (Standard Definition) ou HD (High Definition) e comporta os padrões de compressão MPEG-2 ou MPEG-4.

Os dados transmitidos por um sinal ASI são divididos em pacotes de dois tamanhos possíveis: 188 bytes ou 204 bytes. A transmissão em pacotes de 188 bytes consiste no sinal puro sem correções. Após passar por processos e códigos de correções de erros, os pacotes se estendem para 204 bytes.

Os cabos e conectores são os mesmos da tecnologia SDI, sendo utilizado o cabo coaxial e o conector BNC.

2.5.1.4 TS-MPEG *Transport stream*

É um formato digital para transmissão e armazenamento de vídeo, áudio e outros tipos de dados. É utilizado para sistemas broadcast como o DVB (*Digital Video Broadcasting*) e o IPTV (*Internet Protocol Television*). Uma *Transport Stream* carrega a informação de vídeo e áudio já comprimidos e encapsulados na forma de pacotes, cada um com o tamanho limitado à 188 ou 204 bytes. Cada pacote representa um programa contido na *Transport Stream*.

2.5.1.5 Program specific information (PSI)

Devido a sua capacidade de carregar diversos programas e serviços em um único fluxo de bits, a decodificação da *Transport Stream* é um pouco mais complexa. Deve-se informar ao decodificador quais informações devem ser decodificadas, e isso é feito através das tabelas PSI. As tabelas PSI mais relevantes são:

- PAT (*Program Association Table*): Essa tabela lista todos os programas disponíveis na Transport Stream
- PMT (*Program Map Table*): Contém a informação sobre um programa específico. Cada programa possui sua PMT.

Cada PMT possui um conjunto de *Packet Identifier* ou PID que, como o próprio nome já diz, são identificadores dos pacotes de áudio, vídeo ou serviços de um determinado programa. Uma PID sempre faz referência à sua PMT, ou seja:

- A PID 201 se refere ao áudio 1 da PMT 200. (Exemplo)

Figura 9 - Exemplo: PMT "200".

Vídeo 1 PID 200	Áudio PID 201	Vídeo 2 PID 203	Dados PID 204	Vídeo 1 PID 200	Vídeo 1 PID 200	Vídeo 2 PID 203	Vídeo 1 PID 200	Vídeo 2 PID 203	Dados PID 204	Vídeo 2 PID 203	Vídeo 1 PID 200
--------------------	------------------	--------------------	------------------	--------------------	--------------------	--------------------	--------------------	--------------------	------------------	--------------------	--------------------

Fonte: Hemmlepp (s.d)

2.5.1.6 Multiplexador

Responsável por receber diversos fluxos de sinais ou Transport Streams e os multiplexar, formando um único fluxo de dados chamado BTS (Broadcast Transport Stream). Sua saída será uma informação serial transmitida no protocolo ASI.

2.5.1.7 Modulador

Irá receber a informação digital serial codificada e comprimida vinda do multiplexador e modular utilizando técnicas de modulação avançadas, de forma a transmitir a maior quantidade de símbolos utilizando o mínimo de banda possível. O tipo de modulação utilizada pela emissora vai depender da maneira na qual o sinal é irradiado.

Após a codificação e modulação do sinal contendo a programação da emissora, deve-se transmitir para o público em geral. Há basicamente duas formas para se fazer isso, uma via transmissão terrestre e outra via satélite.

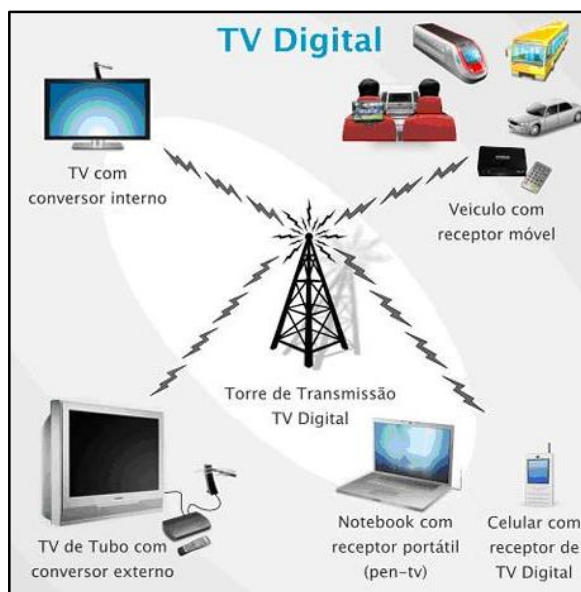
2.5.1.8 ISDB-T (*Integrated Services Digital Broadcasting Terrestrial*)

A transmissão terrestre no Brasil segue o padrão ISDB-Tb, que foi criado seguindo o padrão ISDB-T japonês. Segundo Teleco (2022), no Brasil, o modo de transmissão é a radiodifusão terrestre, com o uso da modulação BST-OFDM e operando em canal de 6 MHz.

A modulação BST-OFDM (*Band-segmented Transmission Orthogonal Frequency Division Multiplexing*) é uma evolução da modulação COFDM e é utilizada no sistema ISDB-T. Essa modulação trabalha com segmentação de banda, dividindo a largura de 6 MHz em 13 segmentos, e, dependendo da transmissão escolhida, utiliza um ou mais segmentos para cada camada, podendo transmitir até três feixes de dados simultâneos com modulações diferentes entre si (Teleco, 2022).

O padrão ISDB-Tb consiste em um sistema de radiodifusão, na qual uma antena é instalada no ponto mais alto da cidade e gera uma onda eletromagnética omnidirecional em formato circular, no qual seu diâmetro atinge por volta de 40Km. Por essa razão, a transmissão terrestre se restringe a atingir o público localizado na cidade na qual o *head-end* está localizado, e algumas vezes para cidades próximas.

Figura 10 – Transmissão terrestre – TV Digital



Fonte: Wikidot (2011).

Muitas emissoras contam com *links* de satélite para a distribuição do sinal por todo estado ou até mesmo país. O padrão seguido para a transmissão de TV digital via satélite no Brasil é o DVB-S2.

2.5.1.9 DVB-S2 (*Digital Video Broadcasting – Satellite* – Segunda geração)

Criado para ser o sucessor do sistema de codificação e demodulação DVB-S de televisão digital (apresentado em 1994), foi desenvolvido em 2003 por um consórcio de empresas participantes do projeto Digital Video Broadcasting, e ratificado em 2005 pelo ETSI (*European Telecommunications Standards Institute*). Seu desenvolvimento coincidiu com a introdução da HDTV e novos codecs de vídeo, como o H.264 (MPEG-4 AVC). O sistema DVB-S2 apresentou diversas melhorias em relação ao seu antecessor (DVB-S), dentre elas:

- Aceita vídeos com compressão MPEG-4 e técnicas de modulação QPSK, 16APSK ou 32APSK;
- Conta com avançado esquema de codificação;
- Sistemas de modulação que permitem alocar dinamicamente a largura de banda, mudando os parâmetros de transmissão.

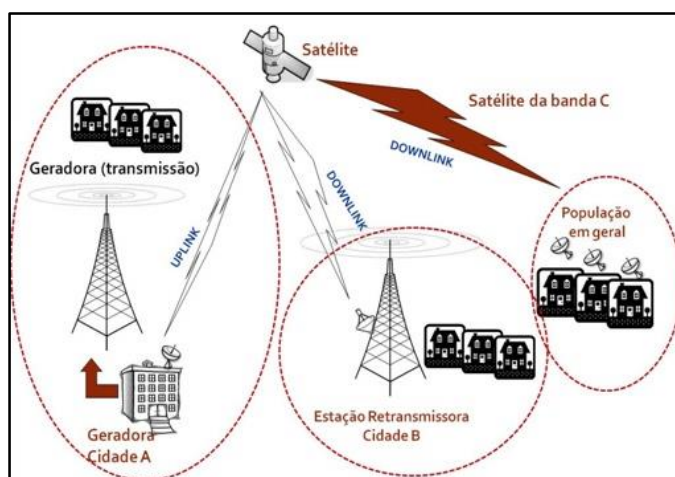
O documento de apresentação do padrão DVB-S2 afirma que o mesmo apresenta um ganho de desempenho de cerca de 30% em relação ao DVB-S, considerando a mesma potência de transmissão e largura de banda de satélite.

Os satélites geoestacionários ficam a cerca de 30.000 Km de altitude e basicamente funcionam como refletores, recebendo o sinal irradiado da Terra pela antena de transmissão da emissora (com todo conteúdo da programação) e transmitindo de volta para o planeta.

As antenas de transmissão via satélite das emissoras de televisão no Brasil emitem ondas eletromagnéticas de frequência de banda C, cujo espectro fica na faixa de 4 a 8 GHz e tem a vantagem de uma maior estabilidade em condições climáticas adversas. As antenas de transmissão em banda C devem estar alinhadas com satélites que recebem sinais na mesma faixa de frequência.

O sinal de volta ou descida é então captado pelas retransmissoras, que vão receber esse sinal a partir de uma antena de recepção, processá-lo e novamente transmitir para seu público local via transmissão terrestre ISDB-T. É possível também que esse sinal seja recebido diretamente pelo telespectador final, caso este possua os equipamentos de recepção em sua residência (uma antena de recepção e o decodificador).

Figura 11 - Transmissão via satélite – TV Digital.



Fonte: Abert (2013)

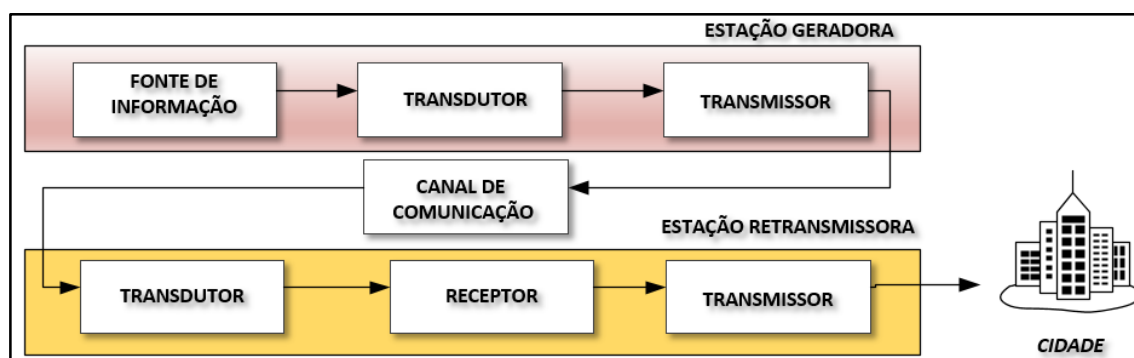
2.5.2 Retransmissora (RTV)

É um serviço complementar, que tem como objetivo retransmitir, simultaneamente, os sinais da estação geradora de televisão para o público em geral, em lugares onde o sinal irradiado pela geradora não alcance. Se resume, em geral, a um receptor, transmissor e

equipamentos acessórios destinados a complementar o sinal da estação geradora de televisão. O sinal vindo da estação geradora pode ser captado via *links* de satélite ou micro-ondas.

Após a compreensão do caminho no qual o sinal gerado pela emissora de TV percorre até ser irradiado ao satélite, fica mais fácil entender como uma estação retransmissora de TV funciona e qual o seu propósito. A figura 12 detalha na forma de diagrama de blocos os fundamentos de uma retransmissão de sinal.

Figura 12 - Diagrama de retransmissão de sinais.



Fonte: Elaborado pelo autor.

2.5.2.1 Antena de recepção via satélite

Segundo Carvalho e Badinhan (2011), uma antena de recepção pode ser definida como um dispositivo transdutor que transforma a energia eletromagnética irradiada em corrente de RF para ser entregue ao receptor. Portanto, sua função é primordial em qualquer comunicação em que exista radiofrequência.

Nas estações retransmissoras ou afiliadas, antenas de recepção de satélite estão propriamente alinhadas com algum satélite que retransmite o sinal em frequências de banda C (varia de 4GHz à 8GHz). A maioria das antenas de recepção das estações retransmissoras e afiliadas no Brasil possuem de 2,6 a 3 metros de diâmetro.

As antenas de recepção possuem um componente que fica localizado em uma haste no centro da antena chamado LNB (*Low Noise Blockconverter*, ou conversor de baixo ruído) que basicamente converte e amplifica os sinais eletromagnéticos concentrados na antena para sinais elétricos, capazes de ser interpretados pelo receptor de satélite. Também é responsável por mudar a frequência da portadora do sinal recebido a partir do oscilador local localizado neste componente, convertendo de banda C para banda L (frequência de micro-ondas, na faixa de 950 a 2150 MHz), que poderá ser interpretada pelo receptor de satélite.

Figura 13 - Antena de recepção de satélite – Banda C.



Fonte: Elaborado pelo autor.

2.5.2.2 Receptor de satélite

Utilizado para recepções DVB-S2, o receptor ou IRDs (*Integrated Receiver Decoder*) é um dispositivo eletrônico capaz de capturar a informação digital transmitida em sinais de rádio na faixa de 950 à 2150MHz (conhecida como banda L), amplificar, de-modular e enviar o sinal codificado para o transmissor das estações retransmissoras, que irá modular o sinal e transmitir para a cidade através da antena de retransmissão. O receptor também poderá decodificar o sinal recebido, transformando o conteúdo para o formato original compatível com a TV do telespectador, podendo exibir o conteúdo do canal.

Possuem basicamente dois tipos, para diferentes aplicações: O receptor de satélite doméstico e o profissional. O doméstico, como próprio nome já diz, é voltado para a recepção de sinais locais, isto é, para os consumidores que possuem antenas parabólicas em suas residências (muito utilizado em regiões remotas). Esse tipo de receptor possui funções limitadas, sendo conhecidos também como *set-up-boxes*.

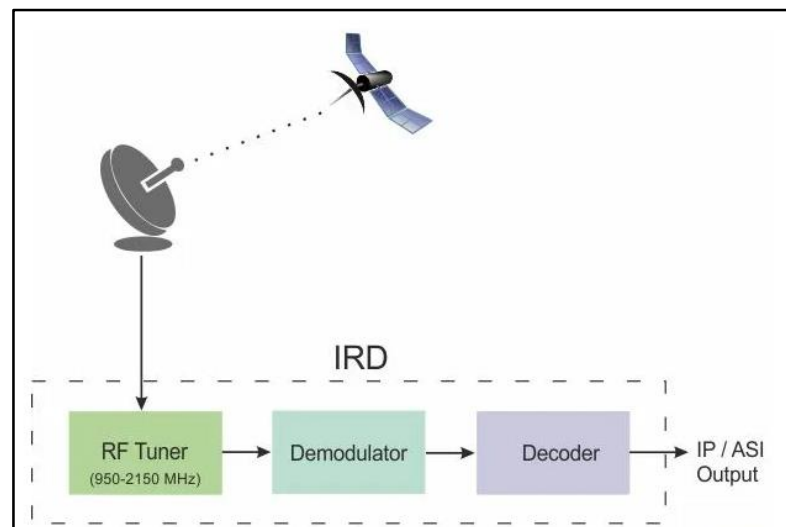
O outro tipo é o receptor de satélite profissional. Ele possui funções avançadas utilizadas na recepção dos sinais, e muitas vezes múltiplas entradas de antenas de satélite. Pode receber também sinais via protocolos de rede e possui saídas em diferentes formatos, como ASI, IP, HDMI e SDI. É comumente utilizado em emissoras de TV para a retransmissão de sinais a cidades fora da área de cobertura da transmissão terrestre. Nesses sistemas o sinal é recebido via satélite, amplificado, de-modulado, e posteriormente retransmitido para os telespectadores.

Figura 14 - Receptor de satélite para aplicações profissionais.



Fonte: Tecsys do Brasil (2022).

Figura 15 - Funcionamento de um receptor de satélite profissional.



Fonte: Headend Info (s.d)

2.5.2.3 Transmissor

As ondas de rádio são formadas pela interação de um campo magnético e um campo elétrico, originando uma onda eletromagnética irradiada por uma antena. Essa onda é gerada por uma corrente de radiofrequência proveniente de um transmissor, em uma frequência de operação que depende do tipo de sistema de comunicação.

Ainda, para Carvalho e Badinhan (2011) o transmissor fornece a potência necessária para amplificar o sinal elétrico, a fim de que ele percorra longas distâncias, uma vez que sua energia vai se perdendo ao longo da transmissão pelo canal de comunicação (fios elétricos ou espaço livre) até ao receptor.

Um equipamento transmissor é constituído basicamente por cinco partes:

- Fonte de alimentação: Fonte de energia utilizada pelo equipamento para poder realizar as etapas necessárias de transmissão do sinal;
- Oscilador: Geração da onda portadora que carregará a informação através do ar;
- Modulador: Através da variação de algum parâmetro da onda portadora (frequência ou amplitude) irá gerar o sinal modulado que será transmitido;
- Amplificador de radiofrequência: Incrementará a potência do sinal modulado para que possa ser transmitido;
- Circuito de casamento de impedâncias: Irá ajustar a impedância de saída do transmissor à da linha de transmissão, de forma a evitar ondas estacionárias.

Figura 16 - Transmissor de UHF.



Fonte: Tecsys do Brasil (2022).

2.5.2.4 Antena de transmissão

A antena de transmissão tem o papel de irradiar o sinal modulado vindo do transmissor.

Uma antena funciona da seguinte maneira: o transmissor produz o sinal da informação na forma de corrente alternada (corrente de radiofrequência). Ao circular na antena de transmissão, a corrente de RF produz uma onda eletromagnética a seu redor, que se irradia pelo ar. Ao atingir uma antena receptora, a onda eletromagnética induz nela uma pequena corrente elétrica, cujas oscilações acompanham o movimento da onda. Essa corrente é muito mais fraca do que a presente na antena transmissora, sendo amplificada no receptor (Carvalho e Badinhan, 2011, cap. 6).

O dimensionamento de uma antena é feito de acordo com o comprimento de onda λ , sendo esse definido pela faixa de frequência de operação do sistema de rádio. Do ponto de vista elétrico, uma antena pode ser vista como uma impedância possuindo componentes resistivos, capacitivos e indutivos, que devem possuir (dentro da faixa de operação) o mesmo valor da linha de transmissão, caso contrário serão geradas ondas estacionárias.

As ondas estacionárias são chamadas também potência refletida ou reversa, as quais geram perdas de energia na transmissão, causando problemas na comunicação.

A *Ultra High Frequency* (UHF) é uma faixa de frequências que se situa entre 300 MHz e 3 GHz, sendo nesse espectro transmitidos os sinais de televisão terrestre. O transmissor monitorado nesse trabalho irá irradiar os sinais de rádio dentro destas frequências.

As antenas de transmissão geralmente estão suspensas a partir de uma torre de aço de forma que alcancem a altitude necessária para irradiar o sinal para toda a cidade. A figura 17 mostra a torre de transmissão da estação retransmissora utilizada como base para este trabalho. Uma torre de transmissão pode sustentar diversas antenas de transmissão, cada uma irradiando o sinal de uma emissora de TV diferente.

Figura 17 – Torre e antenas de transmissão de uma estação retransmissora de televisão.



Fonte: Elaborado pelo autor.

2.6 TELEMETRIA

Os componentes que integram os modernos sistemas de diferentes áreas da sociedade muitas vezes estão fisicamente localizados distantes um dos outros. Assim sendo, se tornou necessário encontrar formas de transmitir informações de funcionamento de sistemas remotos sem que seja necessário que o administrador se dirija ao local de instalação para verificar a integridade dos componentes.

Telemetria é o processo de gravação e transmissão de dados de uma fonte remota para um sistema central com o intuito de monitorar e analisar recursos. A palavra é derivada do grego *tele* = remoto, e *metron* = medida. Os dados podem ser transmitidos de diferentes formas entre o dispositivo monitorado e sistema central, sendo via radiofrequência, infravermelho, ultrassom, GSM, satélite ou cabo.

Os primeiros sistemas de telemetria foram desenvolvidos com o objetivo de monitorar as redes de distribuição de energia, tendo sido instalado pela primeira vez em Chicago no ano de 1912. Foram utilizadas linhas telefônicas para transmitir os dados de operação de diversas usinas para uma central.

O meio de transmissão de dados dependerá da aplicação da telemetria, uma vez que ela é utilizada em diferentes áreas, como meteorologia, medicina, redes de computadores e, como exposto nesse trabalho, em sistemas televisivos.

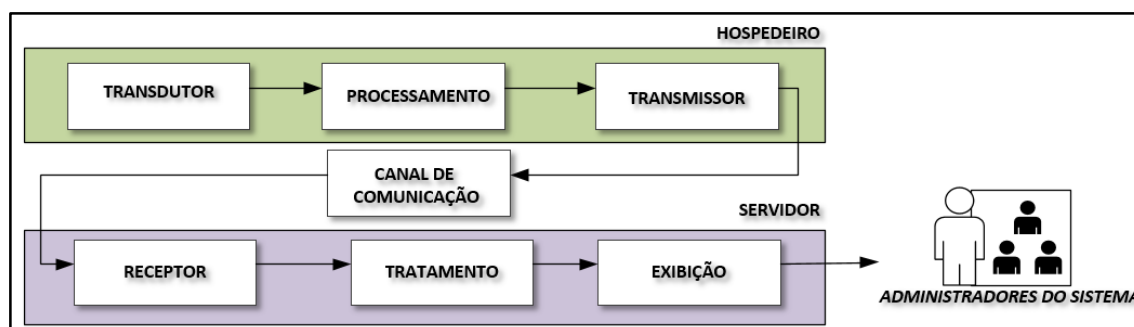
A telemetria é muito utilizada na área aeroespacial, por exemplo. É empregada na Estação Espacial Internacional (ISS), para a monitoração da condição física dos astronautas, bem como do funcionamento dos complexos sistemas instalados na estação. Seu primeiro uso na área foi através do satélite russo Sputnik, lançado em 1957.

A telemetria oferece informações que podem ajudar na detecção de travamentos ou problemas nos sistemas de um dispositivo monitorado. Pode também informar qual recurso está sendo mais utilizado, além de fornecer uma melhor visualização da performance sem ser necessário dirigir-se ao local remoto.

2.6.1 Componentes de um sistema de telemetria

Essencialmente, um sistema de tele supervisão ou telemetria funciona entre um sistema remoto e um local. O sistema remoto deve colher as informações dos dispositivos monitorados e transmiti-las ao sistema central, que irá tratar e exibir os dados para os administradores do sistema.

Figura 18 - Diagrama de blocos simplificado de um sistema telemetria.

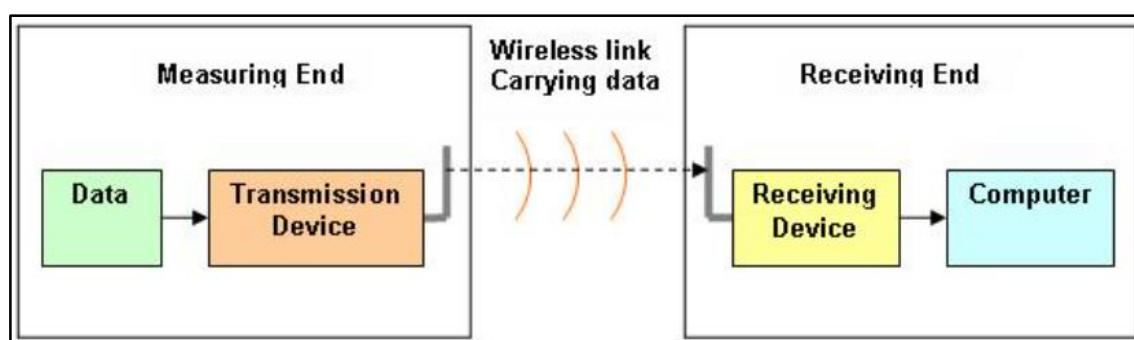


Fonte: Elaborado pelo autor.

Entre esses dois pontos, há um ou mais dispositivos de entrada, chamados de sensores, responsáveis pela coleta, processamento e transmissão dos dados. Os sensores são compostos por transdutores, que possuem como saída dados medidos que, após processados, são chamados de sinais. Os sinais são transmitidos ao servidor por um meio de transmissão que pode ser à fio ou wireless, via rádio frequência, ultrassom ou micro-ondas.

Os sistemas de telemetria sem fio são os que melhor se adaptam a localidades remotas, uma vez que as informações são transmitidas do hospedeiro ao servidor via sinais de radiofrequência, sem a necessidade de cabos. A rede móvel 3G/4G é largamente utilizada para essa finalidade, uma vez que possui alta cobertura em quase todo território nacional e altas velocidades de transmissão. É possível conectar um dispositivo à internet a partir de redes 3G/4G via modem USB e um chip móvel. A figura 19 ilustra o funcionamento de um sistema de telemetria sem fio.

Figura 19 - Sistema de telemetria sem fio.



Fonte: Wireless Technology Advisor (s.d)

2.6.2 Sistema de telemetria em estações retransmissoras de TV

A transmissão de uma variável medida nas estações retransmissoras é uma importante função nos sistemas de uma rede de televisão, uma vez que em muitos casos a programação de uma emissora é distribuída a todo estado ou mesmo a todo país. Ao possuir um sistema de telemetria, as emissoras podem obter dados de um transmissor localizado à centenas de quilômetros em uma estação retransmissora responsável por levar o sinal de TV para uma cidade inteira.

Sem um sistema confiável de telemetria, uma emissora de televisão que transmite sua programação em várias cidades fica refém de telespectadores regionais para informar se o sinal está sendo transmitido com qualidade naquela região ou mesmo se está sendo transmitido.

3 TEMA DE ESTUDO

É possível imaginar a seguinte situação:

Um telespectador da emissora A (localizada em São Paulo) faz uma reclamação na central de relacionamento da empresa, informando que, ao sintonizar o canal desta emissora na cidade de Guaratinguetá (localizada a cerca de 220Km de São Paulo), se depara com um aviso de que não há sinal. Os técnicos da emissora A fazem o possível para orientar o telespectador e resolver o problema via telefone, porém não chegam a uma conclusão.

Caso a emissora A possua um sistema de telemetria instalado em sua estação de retransmissão na cidade de Guaratinguetá, será possível verificar todos os parâmetros de recepção e retransmissão do sinal nessa cidade, garantindo que a cidade esteja ou não sendo atendida.

Caso a emissora A não possua um sistema de telemetria, irá se deparar com uma situação um pouco incômoda: Deve acreditar que o relato do telespectador é verdadeiro e a cidade de Guaratinguetá está fora do ar, sendo necessário deslocar um técnico (que está em São Paulo) para averiguar o problema, correndo o risco de que o problema seja apenas na televisão ou na antena do telespectador, e não na cadeia de retransmissão.

O sinal da emissora A é transmitido para cerca de 200 cidades em todo território do estado de São Paulo. São cidades de pequeno, médio e grande porte que oferecem aos seus habitantes a programação da emissora através das estações retransmissoras que funcionam como ponte para que o sinal chegue aos mais distantes locais.

Manter o controle sobre tantos equipamentos torna-se um problema, tendo em vista as distâncias e a falta de confiabilidade das informações de pessoas que não possuem conhecimento técnico, como telespectadores. Além disso, deslocar um técnico até a estação retransmissora implica em altos custos para a TV.

Pensando em resolver o problema relatado, desenvolveu-se um sistema de telemetria que será instalado em caráter piloto em uma estação retransmissora de TV localizada no interior do estado de São Paulo. A estação em questão retransmite sinais vindos da geradora, uma emissora de televisão localizada na capital do estado.

O sistema de telemetria terá como objetivo monitorar e gerenciar os parâmetros de funcionamento do receptor de satélite e do transmissor de UHF, tendo como premissa auxiliar nas decisões dos técnicos e engenheiros da emissora de televisão, fazendo com que problemas relacionados à falta de sinal para os telespectadores da cidade atendida pela retransmissora sejam resolvidos mais rapidamente.

O município escolhido como piloto para o projeto se estende por 322,2 km² e contava com 120.858 habitantes no último censo, sendo uma importante cidade de sua região. A escolha desta cidade para o projeto foi devido ao fato de que o local no qual está instalado a estação retransmissora possui bom sinal de internet móvel, e também por ser o local mais próximo à estação geradora, facilitando o deslocamento caso seja necessária alguma modificação física no projeto.

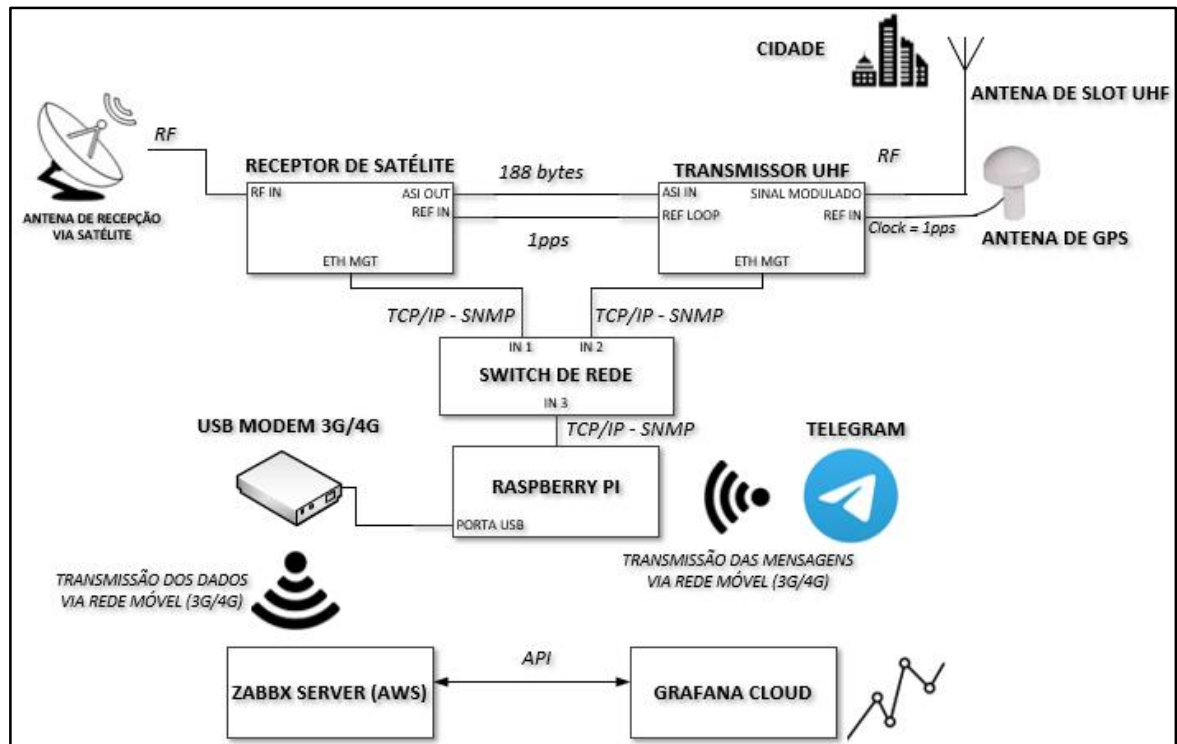
O sistema de telemetria instalado na estação retransmissora terá como componente principal o microcomputador Raspberry Pi, que estará conectado a duas redes: Uma local, na qual estarão conectados também os dispositivos monitorados, e uma global, ou seja, a internet, que será o meio de transmissão entre os dados coletados localmente e um servidor online. A internet será fornecida ao Raspberry Pi a partir de um modem USB que possui um chip de dados móveis e não necessita de configurações adicionais para seu funcionamento.

O servidor online estará hospedado de forma gratuita em uma instância criada na plataforma *Amazon Web Services* da empresa Amazon. Neste servidor estará instalada a plataforma de monitoração Zabbix, que será responsável por receber todos os dados coletados remotamente. O último passo será a criação de uma dashboard de monitoração na plataforma Grafana Cloud, que contará com fácil acesso via navegador e visualização de alto nível na forma de gráficos.

O sistema contará com um módulo ativo, isto é, que poderá atuar e alterar o valor dos parâmetros dos dispositivos. Para isso, será utilizado o aplicativo de mensagens Telegram que irá enviar as mensagens de texto para o Raspberry Pi. O microcomputador Raspberry Pi estará executando uma ferramenta de software desenvolvida na linguagem de programação Python, que irá receber as mensagens através da biblioteca Telepot, e em seguida convertê-las em comandos SNMP, através da biblioteca PySNMP. Por fim, a aplicação encaminhará esses comandos aos dispositivos, que irão interpretar e executar as instruções.

De acordo com as especificações citadas acima foi desenvolvido o sistema de telemetria seguindo o esquemático mostrado na figura 20:

Figura 20 - Esquemático do projeto proposto.



Fonte: Elaborado pelo autor.

4 DESENVOLVIMENTO

Para o desenvolvimento do projeto em questão foram utilizadas as seguintes ferramentas:

- Microcomputador Raspberry Pi Model 3 B+
- Protocolo SNMP
- Extensão Zabbix Proxy
- Servidor Zabbix
- Instância EC2 (AWS)
- Plataforma *Grafana Cloud*
- Aplicativo de mensagens Telegram

Nos tópicos seguintes serão detalhados os processos de instalação e configuração das ferramentas.

4.1 RASPBERRY PI

Os parâmetros dos equipamentos monitorados neste projeto serão enviados ao servidor online a partir de um microcomputador conectado à internet, chamado Raspberry Pi. Segundo a (Raspberry Pi Foundation, 2022), o Raspberry pode ser definido como:

O Raspberry Pi é um computador de baixo custo do tamanho de um cartão de crédito que se conecta a um monitor de computador ou TV e usa um teclado e mouse padrão. É um pequeno dispositivo que permite que pessoas de todas as idades explorem a computação e aprendam a programar em linguagens como *Scratch* e *Python*. Ele é capaz de fazer tudo o que você espera de um computador desktop, desde navegar na Internet e reproduzir vídeos de alta definição, até fazer planilhas, processamento de texto e jogos. (Raspberry Pi Foundation, 2022).

A versão do Raspberry Pi utilizado neste projeto será a 3B+, a qual possui configurações básicas, porém mais que suficientes para atender a solução proposta. Possui interfaces USB e Ethernet que serão utilizadas no decorrer do desenvolvimento do projeto. O sistema operacional desenhado para a plataforma é chamado de Raspbian OS e os passos para a instalação estão disponíveis no site oficial da plataforma <raspberrypi.com>, na sessão software.

Tabela 1 – Especificações técnicas do microcomputador Raspberry Pi

Item	Tecnologia
Processador	Cortex-A53 Quad-Core
<i>Clock</i> do processador	1.4 GHz
Memória RAM	1GB
Adaptador Wifi	802.11 b/g/n/AC 2.4GHz e 5GHz

Fonte: FilipeFlop (2022).

Figura 21 - Raspberry Pi Model 3B+.



Fonte: Raspberry (2022)

O Raspberry Pi funcionará como parte do transmissor do sistema de telemetria, respondendo as solicitações do servidor com dados de leitura dos equipamentos, salvos no banco de dados local.

No Raspberry Pi está instalado uma extensão do Zabbix chamada Zabbix Proxy, sendo essa capaz de enviar as informações ao servidor. Tanto o servidor quanto o proxy serão abordados nos itens 4.3 e 4.4 deste trabalho.

4.2 INSTALAÇÃO DO PROTOCOLO SNMP NO RASPBERRY PI

O primeiro passo para começar a monitorar os parâmetros de um dispositivo via SNMP será instalar o protocolo em um Raspberry Pi. Esse minicomputador será responsável por colher as informações dos dispositivos conectados à rede local, armazenar temporariamente em um banco de dados e repassar esses dados ao servidor online.

Após a obtenção e instalação do sistema operacional, deverão ser seguidos alguns passos para a correta configuração do protocolo SNMP no Raspberry Pi. Para tal feito, foi utilizado o tutorial disponível em português no site pplware, que até o momento da escrita deste trabalho está disponível através do link: <<https://pplware.sapo.pt/tutoriais/dica-como-instalar-o-snmp-no-raspberry-pi/>>

Ao finalizar a instalação do agente SNMP no Raspberry Pi, será necessária agora a configuração do Zabbix *Proxy* no minicomputador, sendo esse procedimento detalhado no próximo tópico do trabalho.

Figura 22 - Agente SNMP corretamente instalado no Raspberry Pi.

```
● snmpd.service - Simple Network Management Protocol (SNMP) Daemon.
   Loaded: loaded (/lib/systemd/system/snmpd.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2021-06-25 14:21:00 -03; 16s ago
```

Fonte: Elaborado pelo autor.

4.3 ZABBIX SERVER

Criado por Alexei Vladishev e atualmente mantido pela Zabbix SIA, é uma solução grátis e open source de monitoramento distribuído. É um software que monitora diversos parâmetros de uma rede, a saúde e a integridade dos servidores. É uma excelente plataforma para a visualização gráfica, e permite a criação de alertas capazes de agilizar a solução de problemas.

Zabbix é acessado via plataforma web, por isso, permite que o status da rede e dos servidores possam ser acessados de qualquer localização. Graças a sua escalabilidade, pode ser aplicado tanto a pequenas quanto a grandes corporações.

O Zabbix foi desenvolvido para suportar diversos SGBD's (Sistema de Gerenciamento de Banco de dados), para armazenar e coletar dados e configurações. Ainda, segundo (Zabbix, 2022) as funcionalidades do software Zabbix são:

- Fácil configuração de novos servidores e dispositivos;
- Monitoramento distribuído com uma interface web centralizada;
- Servidor disponível para as mais diversas distribuições Linux;
- Agentes disponíveis para os mais diversos sistemas operacionais;
- Sistema de autenticação de usuário altamente segura;
- Permissões de usuário flexíveis;

- Notificação via e-mail de eventos pré-definidos;
- Visualização a nível empresarial dos recursos monitorados;
- Sistema de auditoria de logs.

4.3.1 Funcionalidades do Zabbix

De acordo com Zabbix (2022) o Zabbix é uma solução de monitoração integrada, que provê diversos recursos de monitoração em um único pacote.

A coleta de dados: A coleta de dados dos dispositivos monitorados pode ser feita via agentes ou proxy, com intervalos de coleta configuráveis (de 10 em 10 minutos, por exemplo). A plataforma Zabbix suporta o protocolo SNMP e todas suas funcionalidades.

Alertas altamente configuráveis: É possível configurar alertas e enviá-los a um ou vários destinatários, agilizando a resolução de possíveis problemas.

Gráficos em tempo real: Quaisquer recursos numéricos monitorados podem ser monitorados a partir de gráficos altamente customizáveis.

Recursos monitorados: O front-end da plataforma Zabbix fornece ao gestor do sistema uma visão de altíssimo nível dos recursos monitorados, facilitando a análise e tomada de decisão ao gerenciar os recursos dos dispositivos.

Histórico e armazenamento de dados: Todos os dados coletados pelo servidor são armazenados em banco de dados. O tempo de armazenamento dos dados é configurável e a plataforma possui ferramentas de limpeza interna para dados muito antigos.

Configuração simplificada: As configurações feitas para realizar o monitoramento dos dispositivos serão detalhadas posteriormente neste trabalho. Na plataforma Zabbix, todo elemento monitorado é chamado de host, e este começa a ser monitorado assim que seus parâmetros são definidos. Para cada host é possível criar um perfil de monitoração (template), que irá facilitar o processo quando criado outros hosts para dispositivos iguais.

API Zabbix: Uma API (*Application Programming Interface*) basicamente é uma funcionalidade oferecida por terceiros que pode ser usada em aplicações sem a necessidade de novas codificações. Segundo Zabbix (2022) a API Zabbix fornece interface programável para atualizações em massa, integração com ferramentas de terceiros e outros recursos.

Pronto para ambientes complexos: Através do Zabbix Proxy (que será explicado oportunamente neste trabalho) é possível estender as funcionalidades do Zabbix Server sem

necessitar de grande poder de processamento. Tal solução é perfeita para a monitoração remota de dispositivos.

4.3.2 Arquitetura do Zabbix

A arquitetura do Zabbix é dividida em três camadas, que são: a aplicação, banco de dados e interface web. A camada de aplicação é representada pelo servidor, que é o componente principal da plataforma de monitoração.

Servidor Zabbix é o componente central da solução e, em ambientes centralizados, os agentes enviam os dados coletados (sobre integridade, disponibilidade e estatísticos) para ele. Em ambientes descentralizados o envio dos dados é feito para um componente intermediário: o *proxy*. (ZABBIX, 2022, Cap. 1).

Ainda, de acordo com Zabbix (2022) o Zabbix proxy pode coletar dados de desempenho e disponibilidade em nome do servidor Zabbix.

O Zabbix Proxy possui um banco de dados separado e é a solução ideal para a monitoração centralizada de localidades geograficamente dispersas e para redes gerenciadas remotamente. É um host responsável por fazer a coleta de informações de agentes remotos. O Zabbix Proxy coleta os dados, os consolida na forma de pacotes e transmite para o Zabbix Server. É uma extensão do Zabbix que não necessita de um hardware poderoso para funcionar, inclusive podendo ser instalado no Raspberry Pi, como será o caso deste projeto.

O *Zabbix Proxy* é um processo que pode receber dados de um ou mais dispositivos monitorados e enviar ao *Zabbix Server*, basicamente ele funciona em nome do *Zabbix Server* (na visão do agente monitorado, o *proxy* passa a ser o *Zabbix Server*). Todos os dados recebidos são armazenados temporariamente (*bufferizados*) e transferidos ao *Zabbix Server* a qual o *Zabbix Proxy* pertence, sendo excluídos na sequência do armazenamento temporário do *Proxy*. A utilização deste componente é opcional, mas normalmente é muito benéfica pois distribui a carga de monitoração normalmente atribuída ao *Zabbix Server*. (Zabbix, 2022, Cap. 6)

A camada de banco de dados se encarrega de armazenar as informações recebidas pela camada de aplicação, e fazer a ponte com a camada de exibição ou interface web.

Todas as informações de configuração da monitoração são armazenadas no banco de dados, tanto o servidor quanto a interface *web* do Zabbix interagem com o SGBD. Por exemplo, quando você utiliza a interface *web* (ou a API) para adicionar itens, eles são salvos em uma tabela do SGBD. Em paralelo a isso, o *Zabbix Server* irá buscar na tabela de itens a lista de itens que deverão ser monitorados. (ZABBIX, 2022, Cap. 2)

A última camada, o *front-end*, realiza a exibição dos dados armazenados no banco de dados aos administradores. Zabbix (2022) afirma que a solução vem com uma interface *web* para acesso rápido e a partir de qualquer dispositivo.

Os benefícios da utilização do Zabbix como sistema de gerenciamento são várias, entre elas: é *open-source*, o tempo de aprendizagem é reduzido, o sistema de monitoramento é centralizado e salvo em banco de dados, a economia relacionada a tempos de inatividade de servidores ou dispositivos de rede, o suporte ao protocolo SNMP etc. Organizações de todos os tamanhos e ao redor de todo o mundo confiam no Zabbix como sua plataforma de monitoramento primária.

4.3.2.1 *Templates* e itens no Zabbix

Um *template* é um conjunto de entidades que pode ser associadas de forma fácil e conveniente a vários hosts. As entidades são geralmente itens que estão sendo monitorados. Os itens se resumem a recursos do dispositivo gerenciado. (ZABBIX, 2022, Cap. 2).

Para Zabbix (2022) os itens são a forma que o Zabbix utiliza para receber dados de um host. Uma vez configurado um host, será necessário adicionar itens para iniciar a coleta de dados.

Como na vida real vários hosts são idênticos ou muito similares, é natural que exista um conjunto de entidades (itens, triggers, gráficos...) que possam ser usados para vários hosts. Com o uso de templates tal processo é simplificado ao simplesmente associar um host a um template, com isso o Zabbix já irá copiar todo o perfil de monitoração necessário para o host.

Outro benefício do uso de templates é que se for necessária a modificação de um determinado perfil de monitoração (por exemplo: adicionar uma nova métrica de monitoração em todos os servidores monitorados) isso poderá ser feito no nível do template, de forma que todos os hosts associados serão alterados em conjunto.

4.3.2.2 Zabbix e SNMP

Existem ao menos 5 formas de comunicação entre o agente ou proxy Zabbix e o Zabbix Server. Para este trabalho será utilizada a comunicação via protocolo SNMP, que é capaz de gerenciar qualquer dispositivo que utiliza desse protocolo para seu gerenciamento. O

dispositivo SNMP com o Zabbix Agent ou Zabbix Proxy instalado utiliza a porta 161 e o protocolo UDP para se comunicar com o servidor Zabbix.

4.3.2.3 Compatibilidade do Zabbix

O Zabbix Server não é compatível com o Windows, porém pode ser instalado em diversas outras plataformas, como as distribuições Linux e o MacOS. No caso do trabalho em questão, o Zabbix Server será instalado em um ambiente de computação em nuvem da AWS (*Amazon Web Services*), economizando recursos e agregando versatilidade ao projeto. O processo de configuração do servidor em nuvem será descrito no item 4.5 deste trabalho.

4.3.3 Instalação do Zabbix Server

O Zabbix Server é o componente principal do software Zabbix. Ele é responsável por receber os dados dos agentes ou proxies e reportar dados que informem a disponibilidade e integridade dos sistemas.

A instalação do Zabbix Server será feita sobre o sistema operacional Ubuntu em sua versão 18.04 LTS, que estará sendo executado em uma instância da plataforma AWS, conforme descrito no item 4.5 deste trabalho. Todos os pacotes instalados serão obtidos do repositório oficial Zabbix: <<https://repo.zabbix.com/>>

4.3.4 Aquisição do Zabbix Server

Os pacotes oficiais no formato DEB são disponibilizados no repositório Zabbix <<http://repo.zabbix.com>>. Após a instalação do repositório oficial na distribuição que irá receber a instalação do Servidor Zabbix, este estará disponível para ser instalado a partir do gerenciador de pacotes. (ZABBIX, 2022, Cap. 3).

4.3.5 Pré-requisitos para instalação do Zabbix

Para instalar o Zabbix existem requisitos de memória (128MB) e de armazenamento (256MB disponíveis em disco). Entretanto, a quantidade de memória e de disco, obviamente, dependerá da quantidade de hosts e de parâmetros monitorados. (ZABBIX, 2022, Cap. 2)

O Zabbix Server e, especialmente, seu banco de dados pode exigir quantidade significativa de recursos da CPU dependendo da quantidade de parâmetros monitorados. (ZABBIX, 2022, Cap. 2)

A figura 23 ilustra exemplos de configuração de hardware para os sistemas CentOS e RedHat. Pode-se considerar que os requisitos são equivalentes para o sistema Ubuntu.

Figura 23 - Requisitos para a instalação do Zabbix Server.

Nome	Plataforma	CPU/Memória	SGDB	Hosts Monitorados
<i>Small</i>	CentOS	Virtual Appliance	MySQL InnoDB	100
<i>Medium</i>	CentOS	2 CPU cores/2GB	MySQL InnoDB	500
<i>Large</i>	RedHat Enterprise Linux	4 CPU cores/8GB	RAID10 MySQL InnoDB ou PostgreSQL	>1000
<i>Very large</i>	RedHat Enterprise Linux	8 CPU cores/16GB	RAID10 rápido MySQL InnoDB ou PostgreSQL	>10000

Fonte: Zabbix (2022)

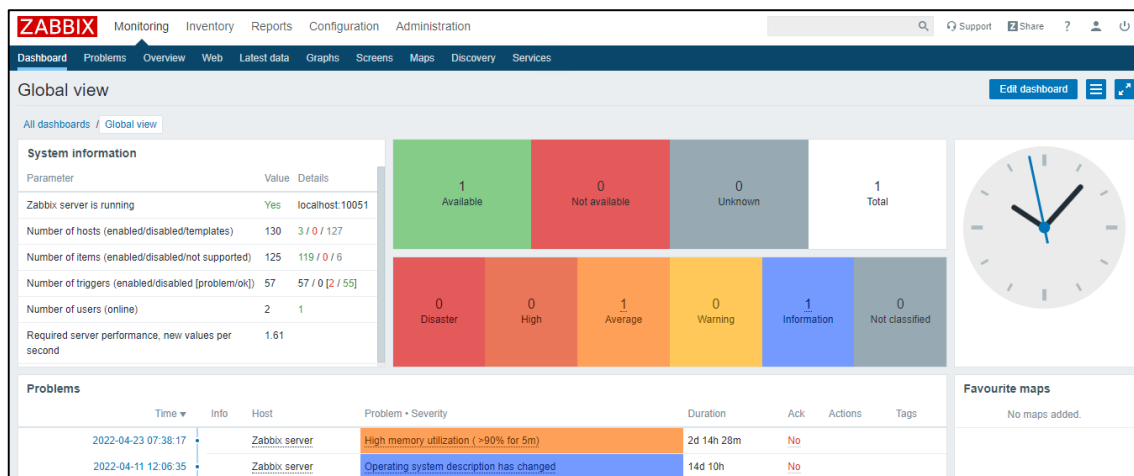
4.3.6 Instalação via pacotes

O download e instalação dos pacotes deverão ser feitos a partir do terminal do sistema operacional, com usuário que possua privilégios de root. Os pacotes necessários para a instalação do Zabbix Server são:

- ***zabbix-server-mysql***: Responsável pela criação do banco de dados do servidor;
- ***zabbix-frontend-php***: Responsável pela configuração do *front-end* do Zabbix Server.
- ***zabbix-apache-conf***: Responsável pela configuração do *back-end* do servidor Zabbix.
- ***zabbix-sql-scripts***: Responsável pela configuração do banco de dados do servidor Zabbix;
- ***zabbix-agent***: Responsável pela configuração de monitoração dos recursos do Zabbix Server.

O tutorial detalhado de instalação do Zabbix Server para diversas plataformas está detalhado no site oficial do Zabbix e, até o momento da escrita deste trabalho encontra-se disponível em: <<https://www.zabbix.com/br/download>>

Figura 24 – *Frontend* do Zabbix pronto para uso pós instalação bem-sucedida.



Fonte: Elaborado pelo autor.

4.3.7 Criando *host* SNMP no Zabbix Server

Um host pode ser entendido por um dispositivo a ser monitorado dentro da plataforma Zabbix. Pode possuir diversas interfaces, como: Agente, JMX, IPMI e no caso deste trabalho, SNMP.

A criação de hosts é uma das primeiras ações de monitoramento no Zabbix. Por exemplo, se você quer monitorar alguns parâmetros no servidor "x", você deve primeiro criar um host chamado, digamos, "Servidor X" e então pode começar a adicionar itens de monitoramento neste novo host. (Zabbix, 2022, Cap. 2)

Antes de configurar o host em si, é necessário criar um grupo de hosts. Nele, ficarão organizados todos os hosts relacionados a esse grupo. Após a configuração do grupo é possível criar os *hosts* que serão anexados aos dispositivos que desejam ser monitorados.

O tutorial para a criação do grupo e dos *hosts* na interface *web* do Zabbix estão disponíveis na documentação oficial da plataforma, disponível, até o momento da escrita deste trabalho, em: <<https://www.zabbix.com/documentation/4.4/pt/manual/config/hosts/host>>

O grupo de hosts criados para esse trabalho será nomeado com o nome da cidade na qual se localiza a estação retransmissora escolhida para o projeto piloto. Serão criados dois hosts nos quais estarão configurados os parâmetros de monitoração dos dispositivos: Um anexado à monitoração do receptor de satélite e outro ao transmissor.

A interface escolhida será a SNMP (*Simple Network Management Protocol*). O protocolo estará sendo executado em um Raspberry Pi que por sua vez irá colher informações dos

dispositivos nas estações retransmissoras. A opção de monitoramento por proxy será utilizada com o nome do proxy configurado no item 4.4. A figura 25 mostra a configuração do host do receptor de satélite da estação retransmissora de TV. Já a figura 26 ilustra o host do transmissor UHF da estação retransmissora.

Figura 25 - Host do receptor de satélite.

The screenshot shows a web-based configuration interface for a satellite receiver host. The form includes the following sections:

- Host name:** A text field containing "ITATIBA - RX Tecsys TS8200".
- Visible name:** An empty text field.
- Groups:** A dropdown menu showing "ITATIBA" with a search box below it containing "type here to search" and a "Select" button.
- Agent interfaces:** A table with headers "IP address", "DNS name", "Connect to", "Port", and "Default". Below the table is an "Add" button.
- SNMP interfaces:** A table with headers "IP", "DNS", and "Port". The first row contains "192.168.0.101", "IP", and "161". There is a "Use bulk requests" checkbox checked, a "Remove" button, and an "Add" button.
- JMX interfaces:** An "Add" button.
- IPMI interfaces:** An "Add" button.
- Description:** A text area containing "Monitoração do receptor de satélite da estação retransmissora de Itatiba".
- Monitored by proxy:** A dropdown menu showing "ITATIBA".
- Enabled:** A checked checkbox.
- Buttons:** "Update", "Clone", "Full clone", "Delete", and "Cancel".

Fonte: Elaborado pelo autor.

Figura 26 - Host do transmissor UHF.

The screenshot shows the Zabbix Host configuration page for a UHF transmitter. The form includes the following fields and sections:

- Host name:** ITATIBA - TX Tecsys TS6200
- Visible name:** (empty)
- Groups:** ITATIBA (selected), with a search box and a Select button.
- Agent interfaces:** A table with columns for IP address, DNS name, Connect to, Port, and Default. An Add button is present.
- SNMP interfaces:** A table with columns for IP address, DNS name, Connect to, Port, and Default. The first row shows IP: 192.168.0.100, Port: 161. There is a checkbox for "Use bulk requests" and a Remove button.
- JMX interfaces:** An Add button.
- IPMI interfaces:** An Add button.
- Description:** Monitoração do transmissor de satélite da estação retransmissora de Itatiba.
- Monitored by proxy:** ITATIBA (selected).
- Enabled:** Checked.
- Buttons:** Update, Clone, Full clone, Delete, Cancel.

Fonte: Elaborado pelo autor.

4.3.8 Configuração dos templates

Segundo Zabbix (2022), a configuração de um template requer que você primeiro crie uma definição geral do *template* (podemos chamar de definição externa) para depois criar suas entidades (definição interna), que podem ser items, triggers, graphs etc.

4.3.8.1 Criando um *template* e adicionando itens

O passo a passo para a configuração de um template na interface web do Zabbix está descrito na documentação oficial, disponível no site da plataforma: <<https://www.zabbix.com/documentation/4.4/pt/manual/config/templates/template>>

Para este projeto serão criados dois *templates*: Um para o receptor de satélite e outro para o transmissor de UHF. Em cada *template* estão definidos como serão monitorados cada métrica dos dispositivos. Para cada métrica, deve existir um Item correspondente, configurado com os parâmetros corretos para a monitoração. O tutorial para a criação de Itens de monitoração na plataforma web do Zabbix é descrito também na documentação oficial.

A figura 27 demonstra, por exemplo, a tela de configuração do item que irá informar se há ou não presença de sinal ASI na entrada no transmissor de satélite:

Figura 27 - Criando um item no Zabbix Server.

The screenshot shows the Zabbix Server configuration page for creating a new item. The form includes the following fields and options:

- Name:** ASI INPUT
- Type:** SNMPv2 agent
- Key:** .1.3.6.1.4.1.41913.1.3.4.1.0
- SNMP OID:** .1.3.6.1.4.1.41913.1.3.4.1.0
- SNMP community:** public
- Port:** 161
- Type of information:** Numeric (unsigned)
- Units:**
- Update interval:** 1m
- Custom intervals:**

Type	Interval	Period	Action
Flexible	Scheduling	50s	1-7,00:00-24:00
- History storage period:** Do not keep history / Storage period: 90d
- Trend storage period:** Do not keep trends / Storage period: 365d
- Show value:** As is
- New application:**
- Applications:** -None-
- Populates host inventory field:** -None-
- Description:**
- Enabled:** ☒

Buttons at the bottom: Update, Clone, Delete, Cancel.

Fonte: Elaborado pelo autor.

4.4 INSTALAÇÃO DO ZABBIX PROXY NO RASPBERRY PI

O Zabbix Proxy será executado no Raspberry Pi e servirá como ponte entre os dispositivos monitorados e o servidor Zabbix. Ele fará a primeira coleta das métricas dos dispositivos, as armazenará em um banco de dados local SQLite.

SQLite é uma biblioteca de código aberto (*open source*) desenvolvido na linguagem C que permite a disponibilização de um pequeno banco de dados na própria aplicação, sem a necessidade de acesso a um SGBD separado. A estrutura de banco junto com a aplicação é denominada de “banco de dados embutido” e é indicada para aplicações de pequeno porte, que utilizam poucos dados. (PORTAL GSTI, 2022).

Em seguida o Zabbix Proxy enviará essas informações ao servidor Zabbix para o qual está configurado.

Para a instalação da extensão Zabbix Proxy no Raspberry Pi foi utilizado o pacote oficial Zabbix contendo todas as dependências para a instalação do proxy e, além disso, o pacote de instalação do banco de dados SQLite. Ambos se encontram – até o momento da escrita deste trabalho - disponíveis no repositório oficial Zabbix: <<https://repo.zabbix.com/>>

O passo a passo para a instalação dos pacotes e configuração do Zabbix Proxy no Raspberry Pi é descrito no site SBCode.net, e se encontra disponível, até o presente momento, em: <<https://sbcode.net/zabbix/proxy-install/>>

Figura 28 - Serviço Zabbix Proxy corretamente instalado no Raspberry Pi.

```
● zabbix-proxy.service - Zabbix Proxy
   Loaded: loaded (/lib/systemd/system/zabbix-proxy.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2022-05-04 15:17:13 BST; 5 days ago
```

Fonte: Elaborado pelo autor.

4.5 PLATAFORMA AMAZON WEB SERVICES (AWS)

A *Amazon Web Services* (AWS) é a plataforma de nuvem mais adotada e mais abrangente do mundo, oferecendo mais de 200 serviços completos de datacenters em todo o mundo. Milhões de clientes, incluindo *as startups* de crescimento mais rápido, grandes empresas e os maiores órgãos governamentais, estão usando a AWS para reduzir seus custos, ficarem mais ágeis e inovarem mais rapidamente. (AMAZON, 2022)

Uma instância consiste em um computador rodando na nuvem. A partir de uma instância, toda a responsabilidade pelo processamento e alocação de memória é repassada à computadores pertencentes a empresa que oferece o serviço. Nestes computadores é possível instalar um sistema operacional, executar servidores virtuais e plataformas de monitoração, sendo necessário para o usuário apenas possuir uma máquina local com acesso à internet para gerenciar sua instância.

A Amazon permite que os usuários da plataforma Amazon Web Services (AWS) criem instâncias sem custo algum desde que não ultrapassem alguns limites estabelecidos de processamento e uso de banda. Para este projeto foi utilizado a instância EC2 (Elastic Compute

Cloud), que consiste em um servidor instalado na nuvem capaz de rodar aplicações na infraestrutura da Amazon Web Services (AWS).

O EC2 suporta a instalação de diversos sistemas operacionais para servidor como o Windows Server, CentOS, Debian e o Ubuntu, sendo este último o escolhido para a execução deste trabalho.

O *Amazon Elastic Compute Cloud* (Amazon EC2) oferece uma capacidade de computação escalável na Nuvem da *Amazon Web Services* (AWS). O uso do Amazon EC2 elimina a necessidade de investir em hardware inicialmente, portanto, você pode desenvolver e implantar aplicativos com mais rapidez. É possível usar o Amazon EC2 para executar quantos servidores virtuais forem necessários, configurar a segurança e as redes e gerenciar o armazenamento. O Amazon EC2 permite aumentar ou reduzir a escala para lidar com alterações nos requisitos ou com picos em popularidade, reduzindo sua necessidade de prever o tráfego. (AMAZON, 2022)

Para hospedar o servidor Zabbix foi criada uma instância na plataforma de computação em nuvem *Amazon Web Services* (AWS) com o objetivo de fornecer o acesso ao servidor de qualquer lugar do mundo, sem a necessidade de ocupação de hardware adicional, de forma gratuita.

Para isso foi escolhida a instância EC2 da plataforma AWS que fornece recursos suficientes para a execução do Zabbix Server. O sistema instalado no servidor em nuvem é o Ubuntu, em sua versão 18.04 LTS. A máquina rodando em nuvem possui um IP único associado a ela, o qual será utilizado posteriormente para o acesso ao servidor Zabbix.

Os passos seguidos para a criação e configuração da instância estão disponíveis na página de documentação oficial da Amazon, a partir do link: <https://docs.aws.amazon.com/pt_br/AWSEC2/latest/UserGuide/concepts.html>

Figura 29 - Instância sendo executada na plataforma *Amazon Web Services*.

Name	ID de instância	Estado da inst...	Tipo de inst...	Verificação de status	Status do al...	Zona de dispon...	DNS IPv4 público
-	i-05f2229b9cc7edacf	Executando	t2.micro	2/2 verificações aprovadas	Sem alar...	sa-east-1c	ec2-18-228-217-122.sa...

Instância: i-05f2229b9cc7edacf

Detalhes	Segurança	Redes	Armazenamento	Verificações de status	Monitoramento	Tags
----------	-----------	-------	---------------	------------------------	---------------	------

Resumo da instância

ID de instância i-05f2229b9cc7edacf	Endereço IPv4 público 18.228.217.122 endereço aberto	Endereços IPv4 privados 172.31.46.99
Endereço IPv6 -	Estado da instância Executando	DNS IPv4 público ec2-18-228-217-122.sa-east-1.compute.amazonaws.com endereço aberto

Fonte: Elaborado pelo autor.

4.6 OBTENDO DADOS PARA A MONITORAÇÃO DE UM DISPOSITIVO

O primeiro passo para começar a monitorar um dispositivo é determinar os parâmetros de interesse para a monitoração. O arquivo *mib* de um dispositivo fornece uma lista de parâmetros muito completa, porém na maioria das vezes apenas alguns destes são interessantes para os gestores do sistema.

Neste projeto serão monitorados dois dispositivos em cada estação retransmissora: Um receptor de satélite e um transmissor de UHF. Para o receptor de satélite, os parâmetros de interesse são descritos nas seções seguintes.

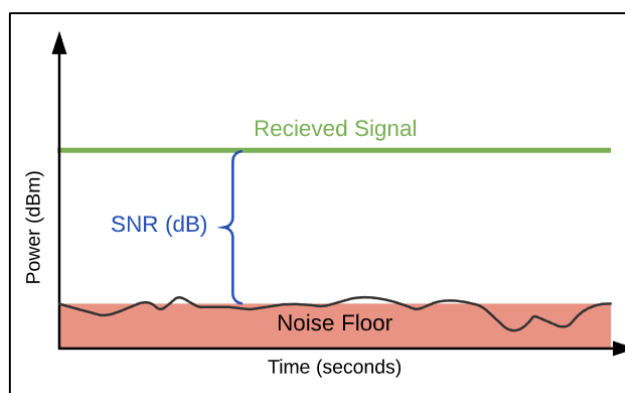
4.6.1 Relação portadora/ruído (C/N)

Também chamado de SNR (*Signal-to-noise ratio*), é a medida da intensidade da onda portadora recebida por um receptor em relação à intensidade do ruído recebido por esse mesmo receptor. Altos valores de C/N (medidos em decibéis) provém melhor qualidade na recepção do sinal digital, em relação a baixos níveis do mesmo parâmetro.

Quando a intensidade do ruído introduzido a um sinal é muito alto, o receptor começa a ter dificuldades em diferenciar o que é informação do que é “sujeira”. Em transmissões digitais, altos níveis de C/N podem indicar baixas taxas de erros na transmissão dos *bits*.

Segundo Teleco (2022) a relação C/N é a relação da potência do sinal pela potência do ruído. Quanto maior a potência do ruído, menor será a relação C/N.

Figura 30 - Definição do nível C/N ou SNR.



Fonte: Cisco Meraki (2022).

O parâmetro C/N é medido em decibéis (dB) através de uma relação entre a potência da portadora de um sinal desejado e o total de ruído recebido. Sendo V_C e V_N os níveis da tensão quadrática (RMS) do sinal da portadora e do ruído, respectivamente, pode-se definir a relação C/N como:

$$C/N_{dB} = 20 \log_{10} \left(\frac{V_C}{V_N} \right)$$

4.6.2 Nível de recepção

Medido em dBm (decibéis por miliwatts) indica a potência do sinal que sai da antena de recepção e chega ao receptor de satélite. Em geral essa potência é muito baixa, em torno de -50dBm, ou $1 \times 10^{-8} W$.

$$-50dBm = 10 \log_{10} \left(\frac{P_1}{1mW} \right)$$

$$P_1 = 10^{-8}$$

$$P_1 = 0,00000001W$$

4.6.3 Presença de sinal de radiofrequência

Indica de forma binária (1 para sim e 0 para não) se o receptor de satélite está sendo alimentado pelo sinal de radiofrequência vindo da antena de recepção de satélite. Para o caso proposto neste trabalho, a frequência do sinal recebido pela antena está na faixa da banda C, o qual é convertido para a banda L através do LNB e que, por fim, chega ao receptor de satélite via cabo coaxial.

4.6.4 Presença de sinal de referência GPS

Os satélites de GPS devem estar devidamente sincronizados entre si, para garantir sua precisão ao localizar objetos ou pessoas. Essa sincronização é feita via relógios atômicos extremamente precisos instalados em cada satélite de GPS. Assim sendo, os satélites de GPS são capazes de transmitir à Terra um sinal pulsado de *1pps* (um pulso por segundo), que pode

ser captado por uma antena de GPS local, transmitido a um receptor e ser utilizado como referência, ou *clock*. A referência é necessária para diversos processos lógicos dentro dos equipamentos eletrônicos, e para o receptor de satélite não é diferente.

Para o transmissor de satélite, os parâmetros de interesse para a monitoração são descritos nas próximas seções deste trabalho.

4.6.5 Potência direta

É a potência do sinal de radiofrequência do transmissor que é repassado à linha e posteriormente à antena de transmissão. Essa potência é a que efetivamente será irradiada pela antena via ondas eletromagnéticas e é dimensionada no projeto de cobertura de uma cidade. Em geral, quanto maior a cidade, mais potência o transmissor deve irradiar para atendê-la completamente. Medido em Watts (W).

4.6.6 Potência refletida

O transmissor gera o sinal modulado em rádio frequência e o transmite a uma certa potência à antena. Quando a linha e a antena de transmissão não estão com o correto casamento de impedâncias, ocorre a aparição de uma onda estacionária, também chamada de potência refletida ou reversa, que volta através da linha até o transmissor. Caso a potência refletida for muito alta, pode até mesmo queimar a saída do transmissor. Medido em Watts (W).

A impedância da antena na faixa de operação deve ter o mesmo valor apresentado pela linha de transmissão à qual a antena está ligada. Caso contrário, ocorrerá descasamento de impedâncias entre a linha de transmissão e a antena, provocando perdas de energia, devido a reflexões. Essa energia refletida, somada com a energia incidente, gera na linha de transmissão uma onda estacionária, prejudicando a comunicação. (Carvalho e Badinhan, 2011, cap. 6).

4.6.7 Presença de sinal ASI na entrada do transmissor

Indica de forma binária (1 para sim e 0 para não) se o transmissor está sendo alimentado pelo sinal codificado ASI vindo da saída do receptor de satélite.

4.6.8 Temperatura do modulador

Como o transmissor trabalha sob condições de alta corrente principalmente nos componentes responsáveis pela modulação do sinal, é importante monitorar a temperatura destes componentes eletrônicos, para o caso de um curto-circuito, por exemplo. Medida em °C.

4.7 MONITORAÇÃO DOS RECURSOS

Após o entendimento dos recursos que serão monitorados no sistema, agora será necessário buscar dentro da lista *.mib* (*Management Information Base*), os endereços das *OIDs* (*Object Identifier*) referentes aos recursos que deseja-se monitorar. O arquivo *.mib* pode ser baixado no site do fabricante ou, em alguns casos, na página de configuração do equipamento.

O projeto piloto de monitoração foi desenvolvido para dispositivos de uma fabricante brasileira de equipamentos do ramo de televisão e, para a obtenção da MIB, foi necessário abrir a página de configuração dos equipamentos (receptor e transmissor) via web browser. Para tal feito, foram realizados os seguintes passos:

- Conexão do equipamento ao computador via cabo de rede;
- Verificação do endereço IP do equipamento e configuração do computador para que os dois estejam na mesma rede;
- Acessar a página de configuração do equipamento digitando o IP do equipamento no navegador.
- Encontrar a seção SNMP e realizar o download da lista de informações de gerenciamento (MIB);

Dependendo do equipamento e marca, a forma de obtenção da lista pode variar.

Basicamente, podemos pensar em um banco de dados de objetos gerenciados e rastreados pelos agentes onde qualquer tipo de status ou dados estatísticos pode ser acessado pelo gerente. Isso recebe o nome de *Management Information Base* (MIB). (SOARES et al, 2022).

A tabela 2 mostra as *OIDs* referentes aos recursos que serão monitorados no receptor e no transmissor da estação retransmissora:

Tabela 2 - OIDs dos parâmetros a serem monitorados.

Equipamento	Parâmetro monitorado	OID
TRANSMISSOR	Presença de sinal ASI	.1.3.6.1.4.1.41913.1.3.4.1.0
	Potência direta	.1.3.6.1.4.1.41913.1.2.4.3.0
	Potência refletida	.1.3.6.1.4.1.41913.1.2.4.4.0
	Referência	.1.3.6.1.4.1.41913.1.2.3.1.1.0
	Temperatura do modulador	.1.3.6.1.4.1.41913.1.2.2.10.1.1.0
RECEPTOR	Relação C/N	.1.3.6.1.4.1.41913.3.1.3.1.8.0
	Nível de recepção	.1.3.6.1.4.1.41913.3.1.3.1.9.0
	Estado do sintonizador de RF	.1.3.6.1.4.1.41913.3.1.3.1.7.0

Fonte: Elaborado pelo autor.

A partir das OIDs obtidas, agora será necessário criar os itens no Zabbix Server para que a monitoração seja possível. O processo de criação de itens está descrito no item 4.3.8 deste trabalho. A figura 31 mostra os itens para monitoração do receptor de satélite.

Figura 31 - Itens para a monitoração do receptor de satélite.

Name ▼	Triggers	Key
RX TECSYS TS8200: Tune Status		.1.3.6.1.4.1.41913.3.1.3.1.7.0
RX TECSYS TS8200: Nível de recepção		.1.3.6.1.4.1.41913.3.1.3.1.9.0
RX TECSYS TS8200: C/N		.1.3.6.1.4.1.41913.3.1.3.1.8.0

Fonte: Elaborado pelo autor.

Já a figura 32 mostra os itens criados para a monitoração do transmissor.

Figura 32 - Itens para a monitoração do transmissor.

Name ▼	Triggers	Key
TX TECSYS TS6200: Temperatura modulador		.1.3.6.1.4.1.41913.1.2.2.10.1.1.0
TX TECSYS TS6200: Referência		.1.3.6.1.4.1.41913.1.2.3.1.1.0
TX TECSYS TS6200: Potência reversa		.1.3.6.1.4.1.41913.1.2.4.4.0
TX TECSYS TS6200: Potência direta		.1.3.6.1.4.1.41913.1.2.4.3.0
TX TECSYS TS6200: ASI INPUT		.1.3.6.1.4.1.41913.1.3.4.1.0

Fonte: Elaborado pelo autor.

Quando terminada a configuração de todos os itens que se deseja monitorar, é importante verificar se os mesmos estão respondendo via SNMP. Para isso, acessar a aba Hosts no servidor Zabbix e verificar se os dispositivos configurados disponíveis (em verde).

Figura 33 - Verificando disponibilidade SNMP dos hosts criados.

☐	Name ▲	Interface	Templates	Status	Availability
☐	RX Tecsyst TS8200	192.168.0.101: 161	RX TECSYS TS8200	Enabled	ZBX SNMP JMX IPMI
☐	TX Tecsyst TS6200	192.168.0.100: 161	TX TECSYS TS6200	Enabled	ZBX SNMP JMX IPMI

Fonte: Elaborado pelo autor.

Após essa configuração, o Zabbix Server estará configurado para a monitoração dos parâmetros desejados. Assim sendo, já é possível verificar através do próprio servidor as métricas dos dispositivos configurados na plataforma em tempo real e com alta precisão. Para tal feito, acessar a aba *monitoring* e, após, clicar em *latest data*.

Figura 34 - Valores dos parâmetros retornados ao servidor (receptor).

Host	Name	Last value
RX Tecsyst TS8...	- other - (3 items)	
	C/N .1.3.6.1.4.1.41913.3.1.3.1.8.0	13.5 dB
	Nível de recepção .1.3.6.1.4.1.41913.3.1.3.1.9.0	-39 dBm
	Tune Status .1.3.6.1.4.1.41913.3.1.3.1.7.0	2

Fonte: Elaborado pelo autor.

Figura 35 - Valores dos parâmetros retornados ao servidor (transmissor).

TX Tecsys TS6... - other - (5 Items)	
☐	ASI INPUT 1.3.6.1.4.1.41913.1.3.4.1.0 0
☐	Potência direta 1.3.6.1.4.1.41913.1.2.4.3.0 11
☐	Potência reversa 1.3.6.1.4.1.41913.1.2.4.4.0 0.5
☐	Referência 1.3.6.1.4.1.41913.1.2.3.1.1.0 1
☐	Temperatura modulador 1.3.6.1.4.1.41913.1.2.2.10.1.1.0 34.0

Fonte: Elaborado pelo autor.

Finalizada a configuração e validação do Zabbix Server no projeto, já é possível montar a interface gráfica para monitoração, a qual será feita via plataforma Grafana, sendo esse processo explicado na próxima seção.

4.8 PLATAFORMA GRAFANA

O Grafana é uma plataforma de monitoração em tempo real, para visualização e monitoração de métricas, essencialmente na forma de gráficos. Os gráficos da plataforma são exibidos em painéis, e podem ser alimentados por diversos tipos de banco de dados, tanto pagos como gratuitos. Um grupo de painéis formam uma *dashboard*, que pode reunir dados de diferentes fontes de dados.

A plataforma também está disponível na nuvem através do Grafana Cloud, podendo ser acessada através do site grafana.com. O Grafana e o Grafana Cloud oferecem uma plataforma de monitoração altamente personalizável e com alta disponibilidade, facilitando a análise dos dados por parte dos gestores do sistema e, no caso da segunda solução, sem a necessidade de hospedar a solução localmente.

A fonte de dados da *dashboard* criada neste projeto para a monitoração das estações retransmissoras será o Zabbix Server. Toda comunicação será feita através da API disponibilizada dentro da própria plataforma do Grafana.

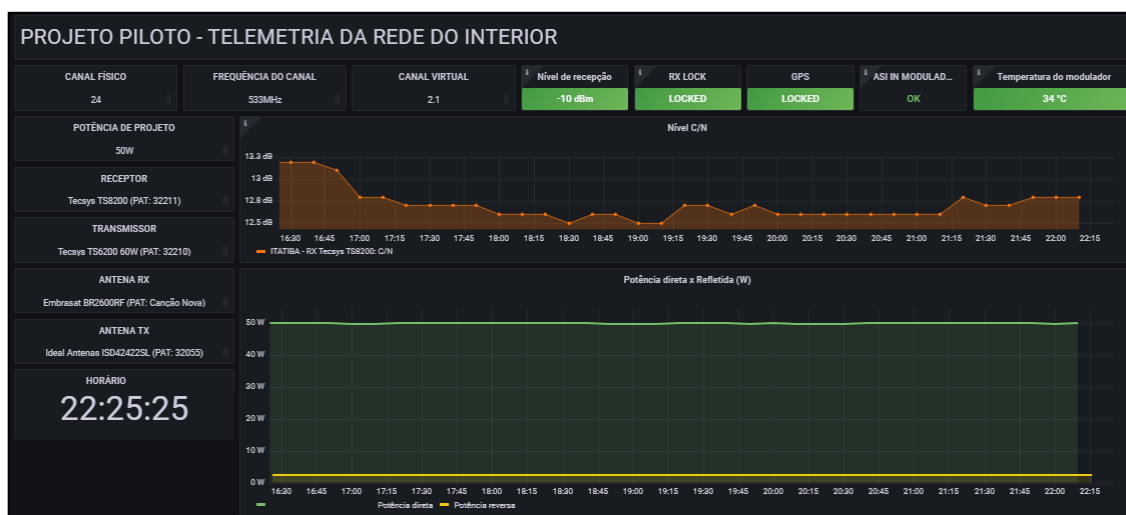
Segundo Basso (2020), a integração entre as ferramentas Grafana e Zabbix é realizada através de um plugin que deve ser instalado no Grafana; ele é responsável por permitir a criação de um *datasource* que irá consumir a API do Zabbix.

Após fazer o login no Grafana, será necessário instalar e configurar a integração do Grafana com o Zabbix Server. Para isso, na plataforma, acessar a aba de plug-ins (em configurações), buscar por Zabbix e fazer a instalação. O passo a passo para a configuração do plugin de integração entre Zabbix e Grafana está disponível – até o momento da escrita deste trabalho – no site do desenvolvedor Alexander Zobnin: <<https://alexanderzobnin.github.io/grafana-zabbix/configuration/>>

Após a instalação do plugin será possível consumir os dados monitorados pelo servidor Zabbix, através das dashboards da plataforma Grafana. Para tal feito seguir o tutorial também descrito no site do desenvolvedor, disponível em: <<https://alexanderzobnin.github.io/grafana-zabbix/guides/gettingstarted/>>

A figura 36 mostra a *Dashboard* de monitoração da estação retransmissora de TV, devidamente configurada com as métricas desejadas:

Figura 36 - *Frontend* para a monitoração da estação retransmissora.



Fonte: Elaborado pelo autor.

4.9 EXECUÇÃO DE COMANDOS VIA TELEGRAM

Segundo Telegram (2022), o *Telegram* é um aplicativo de mensagens móvel e desktop baseado em nuvem com foco em segurança e velocidade.

Com a finalidade de envio de comandos simples aos equipamentos monitorados, foi desenvolvida uma ferramenta de software utilizando a linguagem de programação Python, que irá receber comandos de texto a partir do aplicativo móvel *Telegram* e realizar o mapeamento das OIDs nos dispositivos via protocolo SNMP, atuando sobre os mesmos e podendo fazer resets, alteração de potência (do transmissor de UHF) e algumas outras alterações no sistema. Esta ferramenta será executada pelo Raspberry Pi.

Python é uma linguagem de programação de alto nível [...]. Por ser uma linguagem de sintaxe relativamente simples e de fácil compreensão, ganhou popularidade entre profissionais da indústria tecnológica que não são especificamente programadores, como engenheiros, matemáticos, cientistas de dados, pesquisadores e outros. (KRIGER, 2022).

Para a programação deste sistema foram utilizadas três ferramentas. A primeira é o interpretador dos códigos escritos em Python e as outras duas são as bibliotecas: Telepot e PySNMP. Os pacotes do ambiente Python para Raspberry Pi podem ser encontrados no site oficial: <<https://www.python.org/downloads/>>

A biblioteca PySNMP realiza o envio dos comandos através da porta ethernet do Raspberry Pi para os dispositivos monitorados (receptor e transmissor). Esta biblioteca necessita de três parâmetros para o envio do comando para o dispositivo conectado. São eles:

- IP do dispositivo de destino;
- OID do parâmetro que se deseja alterar;
- Novo valor do parâmetro;

Dentro da biblioteca PySNMP será utilizada a função `setsnmp`, que irá realizar o envio dos novos parâmetros aos equipamentos através da operação Set disponibilizada pelo protocolo SNMP.

Figura 37 - Exemplo de comando utilizando a biblioteca PySNMP.

IP DO DISPOSITIVO	OBJECT IDENTIFIER (OID)	NOVO VALOR
<pre>setsnmp('192.168.0.101', '.1.3.6.1.4.1.41913.3.1.3.6.2.0', Integer(1))</pre>		

Fonte: Elaborado pelo autor.

PySNMP é uma implementação de mecanismo SNMP desenvolvido para a linguagem Python. Possui mecanismo SNMP totalmente funcional capaz de atuar em funções de Agente/Gerente/Proxy, interconectando versões do protocolo SNMP v1/v2c/v3 sobre IPv4/IPv6 e outros transportes de rede. (PySNMP, 2022).

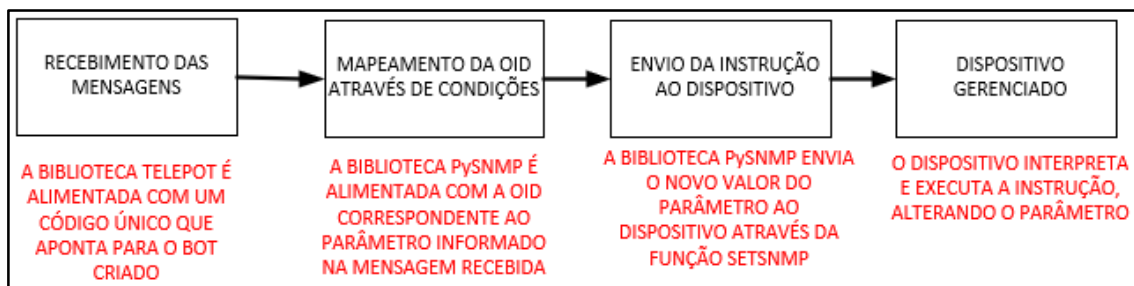
O passo a passo completo para a instalação da biblioteca PySNMP está disponível na documentação oficial, que, até o momento da escrita deste trabalho, se encontra em: [<https://pysnmp.readthedocs.io/en/latest/>](https://pysnmp.readthedocs.io/en/latest/)

A segunda biblioteca instalada no Raspberry Pi é chamada Telepot. Segundo Telepot (2022), o Telepot ajuda você a criar aplicativos para a API do Telegram Bot. Funciona em Python 2.7 e Python 3. Para Python 3.5+.

Essa API permite conectar *bots* ao nosso sistema. *Telegram Bots* são contas especiais que não requerem um número de telefone adicional para configurar. Essas contas servem como uma interface para o código em execução em algum lugar do seu servidor. (Telegram.org, 2022)

Será responsável por realizar a conexão entre o Raspberry Pi e o Telegram, recebendo as mensagens de texto enviadas pelo aplicativo e podendo enviar uma resposta. Cada mensagem de texto recebida será mapeada em um comando SNMP específico.

Figura 38 - Fluxo de funcionamento do programa desenvolvido.



Fonte: Elaborado pelo autor.

A figura 39 demonstra um exemplo da recepção e análise de um comando de texto (reinício do transmissor de UHF) enviado pelo aplicativo Telegram, e, por fim, a realização da operação Set do protocolo SNMP em um dispositivo conectado ao Raspberry Pi.

Figura 39 - Exemplo da recepção e envio dos comandos.

```
if command == '/resettx': #recepção do comando enviado pelo aplicativo
    bot.sendMessage(chat_id,"Reiniciando o TX, aguarde...") #envio de resposta ao usuário
    setsnmp('192.168.0.100','1.3.6.1.4.1.41913.1.1.7.1.1.0',Integer(1)) #envio de comando SNMP ao dispositivo
    time.sleep(90)
    bot.sendMessage(chat_id,"TX reiniciado com sucesso!") #Informe de procedimento concluído
```

Fonte: Elaborado pelo autor.

Após a programação do sistema, os arquivos foram salvos localmente no microcomputador Raspberry Pi, que irá interpretá-los através do compilador Python instalado no mesmo. O código completo do sistema encontra-se disponível no apêndice A do presente trabalho.

O tutorial para criação do bot responsável pelo envio de comandos de texto do aplicativo ao Raspberry Pi está disponível no site oficial da plataforma Telegram: <<https://core.telegram.org/bots#3-how-do-i-create-a-bot>>

A figura 40 ilustra o funcionamento do bot. Já a figura 41 mostra o resultado obtido com o comando, que, no caso, é o desligamento do sinal de rádio frequência da saída do transmissor, levando potência para zero.

Figura 40 - Bot de envio de mensagens ao Raspberry Pi.



Fonte: Elaborado pelo autor.

Figura 41 - Resultado da operação no transmissor



Fonte: Elaborado pelo autor.

A partir do sistema ativo montado, será possível não só monitorar os recursos como também atuar sobre os dispositivos que compõem a estação retransmissora, aumentando o leque de ferramentas para a manutenção dos sistemas abordados neste trabalho.

5 IMPLEMENTAÇÃO

5.1 PRÉ-IMPLEMENTAÇÃO EM BANCADA

Com o objetivo de assegurar a disponibilidade e estabilidade do sistema de telemetria na estação retransmissora, foi inicialmente montado um setup de testes, de forma a monitorar, estudar e melhorar o funcionamento do sistema. A bancada de testes foi montada no laboratório de radiofrequência da emissora de televisão utilizada como base para esse trabalho, como mostram as figuras 42 e 43.

Figura 42 - Bancada de testes do sistema.



Fonte: Elaborado pelo autor.

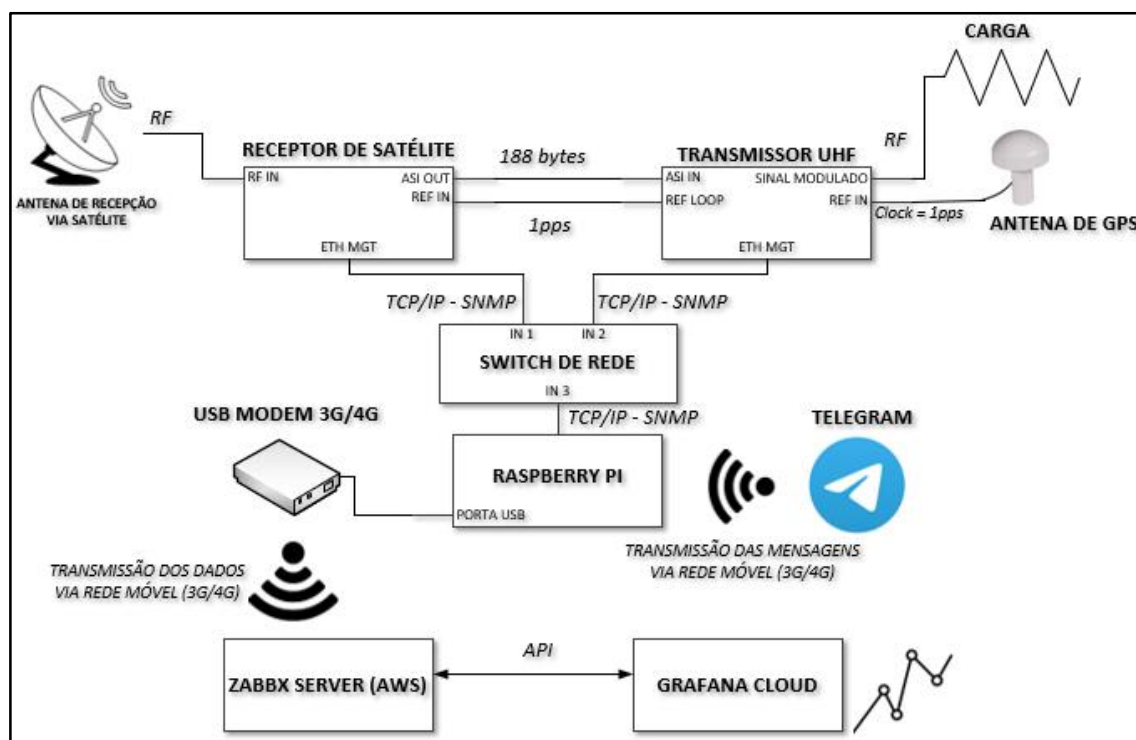
Figura 43 - Bancada de testes do sistema.



Fonte: Elaborado pelo autor.

A figura 42 mostra o receptor de satélite e o transmissor de UHF, que são os dois dispositivos a serem monitorados. Como foi mostrado no item 4.3.7 deste trabalho, foram criados dois hosts no servidor Zabbix, nos quais estão configuradas as métricas (através dos templates e itens), que serão monitoradas nestes dois equipamentos. A figura 44 mostra o esquemático da bancada de testes.

Figura 44 - Diagrama esquemático da bancada de testes.



Fonte: Elaborado pelo autor.

O satélite recebe os sinais irradiados da Terra contendo a programação da emissora de TV e os retransmite de volta para a Terra. O sinal chega na forma de ondas eletromagnéticas, na frequência de banda C. O conversor instalado na antena modifica a frequência do sinal e o amplifica de forma a poder ser recebido pelo receptor. O receptor de satélite demodula o sinal de radiofrequência, o transformando em sinal digital codificado *ASI*, podendo assim ser amplificado e modulado outra vez pelo transmissor. O passo final é transmitir o sinal de rádio frequência agora em UHF (podendo variar de 300MHz à 3GHz), através da linha, chegando à antena.

Os equipamentos de recepção e transmissão estão referenciados através de uma antena de GPS que fornece aos mesmos um sinal de 1pps (um pulso por segundo).

Tanto o receptor quando o transmissor possui uma porta de gerência Ethernet na qual estará conectado a placa Raspberry Pi, que irá solicitar as informações dos parâmetros dos equipamentos através do protocolo SNMP. A rede local de gerenciamento é composta também por um switch simples, responsável por interconectar os dispositivos através do protocolo TCP/IP.

Segundo CISCO (2022) um switch permite que os dispositivos conectados compartilhem informações e conversem entre si.

Pode-se observar no esquema, ainda, que a carga está absorvendo a potência na saída do transmissor. A única modificação que deverá ser feita ao instalar o sistema na estação retransmissora será a troca da carga por uma antena de transmissão, que irá cumprir o papel de irradiar o sinal modulado.

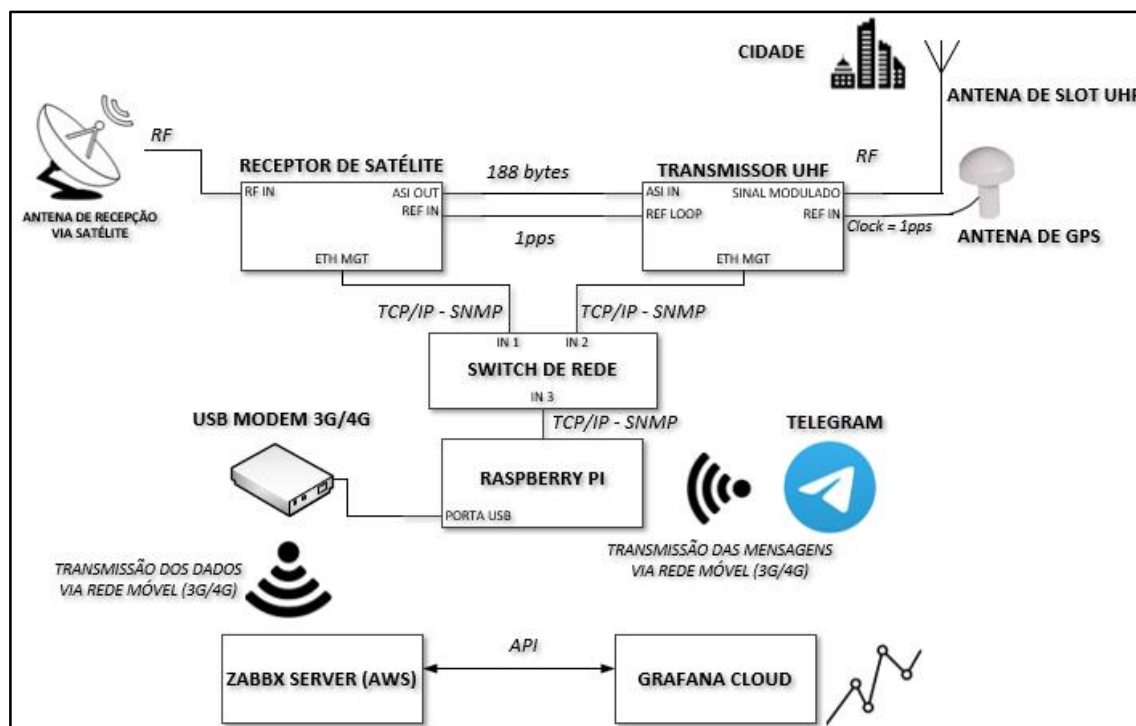
Ao que se refere ao envio dos comandos aos dispositivos via *Telegram*, não houve a necessidade de se instalar uma estrutura adicional no sistema montado, uma vez que o programa está sendo executado pelo Raspberry Pi e utiliza a mesma rede móvel 3G para o envio das mensagens.

5.2 IMPLEMENTAÇÃO EM CAMPO

Com o objetivo de estudar e validar o funcionamento do sistema de telemetria em campo, utilizou-se a estação retransmissora de uma cidade do interior de São Paulo como piloto para o projeto. Para o local foram levados os mesmos equipamentos detalhados no item anterior, substituindo a carga resistiva pela antena de transmissão.

A figura 45 mostra o esquemático do sistema de telemetria instalado na estação retransmissora piloto.

Figura 45 - Diagrama do sistema de telemetria para instalação em campo.



Fonte: Elaborado pelo autor.

A instalação do sistema de telemetria em um ambiente real é muito importante para a validação do projeto, uma vez que em campo os equipamentos estão sujeitos a uma série de contratempos como: Ausência de sinal de internet móvel (o que impossibilita o envio das métricas ao servidor), superaquecimento (visto que todos os equipamentos estarão alocados dentro de um abrigo fechado com pouca circulação de ar), descargas elétricas (que podem danificar inclusive os equipamentos monitorados), umidade etc.

A figura 46 mostra a parte interna do abrigo no qual se localizam os equipamentos responsáveis pela retransmissão do sinal. Já a figura 47 ilustra os dois equipamentos que serão monitorados pelo sistema de telemetria desenvolvido.

Figura 46 - Interior do abrigo da retransmissora.



Fonte: Elaborado pelo autor.

Cada conjunto de equipamentos é responsável por receber e retransmitir o sinal de uma emissora de televisão que atua na cidade em questão.

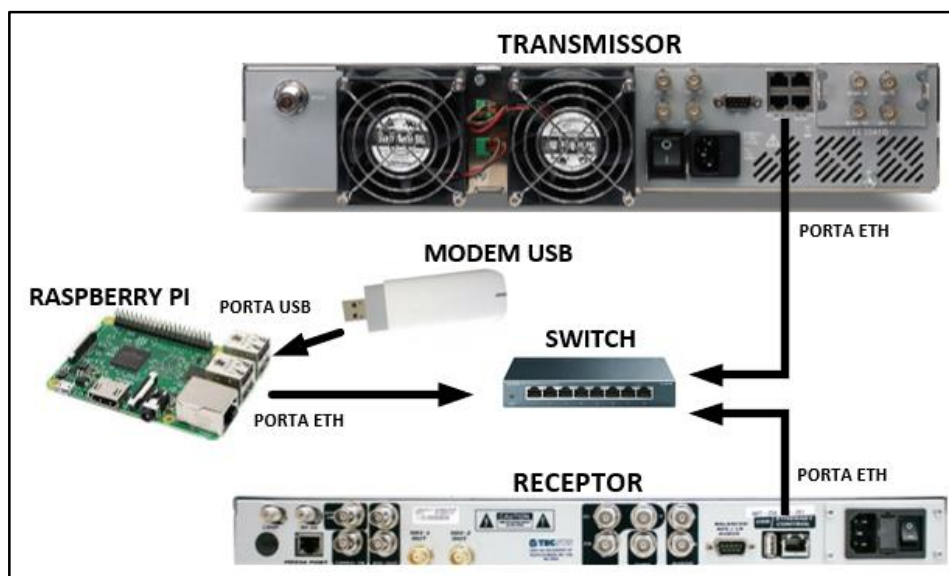
Figura 47 - Receptor e transmissor da estação retransmissora.



Fonte: Elaborado pelo autor.

Após a chegada na estação retransmissora, verificou-se que os equipamentos que seriam monitorados estavam em pleno funcionamento. Criou-se a rede local de troca de informações via protocolo SNMP, conectando o receptor e o transmissor ao Raspberry Pi via switch de rede.

Figura 48 - Esquemático do sistema de telemetria montado em campo.

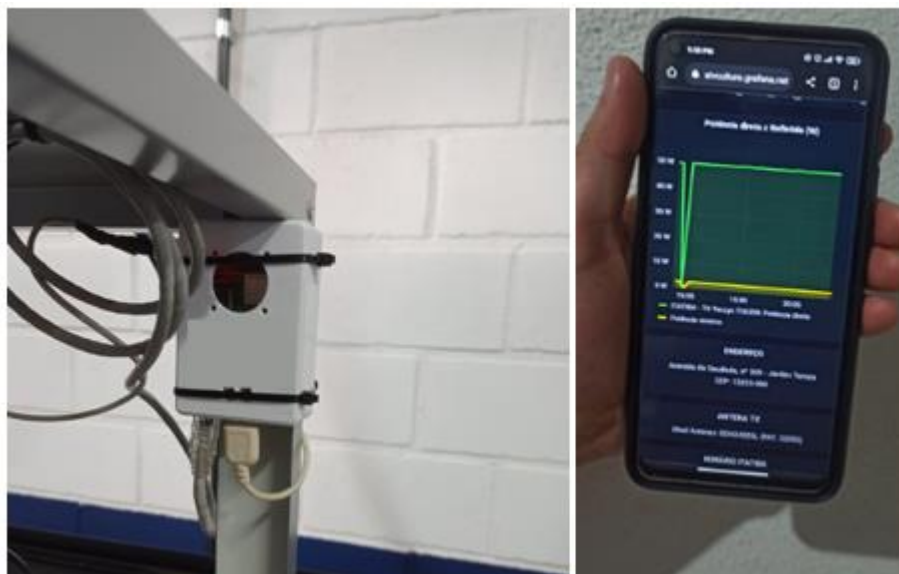


Fonte: Elaborado pelo autor.

Em seguida foi verificado que a qualidade da rede móvel dentro do abrigo era boa, encontrando inclusive o sinal 4G. Assim sendo, não houve dificuldades em conectar o Raspberry Pi à internet através do modem USB (ilustrado na figura 48), um passo necessário para o envio de informações ao servidor Zabbix.

A figura 49 demonstra o sistema de telemetria instalado na estação retransmissora, colhendo informações dos equipamentos e enviando à plataforma de monitoração. O sistema de envio de comandos também foi ativado em campo utilizando a mesma estrutura física montada para a monitoração dos recursos.

Figura 49 - Sistema de telemetria instalado e em funcionamento.



Fonte: Elaborado pelo autor.

6 ANÁLISE DOS RESULTADOS

6.1 MONITORAÇÃO EM BANCADA

O sistema montado em bancada foi cuidadosamente monitorado durante 1 (um) mês, através dos dados informados no Zabbix Server e na plataforma Grafana. Considerando a proposta colocada em discussão, pode-se dizer que o sistema atendeu todos os requisitos esperados. Apresentou grande disponibilidade e estabilidade, ofereceu fácil visualização aos técnicos e engenheiros da emissora utilizada como base para esse trabalho, possibilitando um diagnóstico mais prático de algum problema relacionado ao funcionamento do sistema.

A gráfico da figura 50 dá uma visão geral do funcionamento do sistema de telemetria durante o mês de monitoração em bancada, baseando-se na monitoração do sinal de referência (GPS) no transmissor através da plataforma Grafana. Assumindo que esse parâmetro não deve sofrer variações devido a própria confiabilidade do serviço de geolocalização, é possível afirmar que as quedas visualizadas no gráfico são, em sua totalidade, causadas por falhas no sistema de telemetria projetado, com exceção de uma motivada por falta de energia no laboratório.

Figura 50 - Monitoração do sinal de referência GPS no transmissor durante 1 mês.



Fonte: Elaborado pelo autor.

Pode-se verificar ao analisar o gráfico que a monitoração começa no dia 24 de abril e que a queda mais acentuada ocorre no dia 25 de abril, a qual dura cerca de 3 dias, até o dia 28. Essa queda se deve a um problema de energia ocorrido na sede da TV, motivada por uma tentativa de roubo de cabos elétricos na parte externa da emissora. Apenas os sistemas vitais da TV seguiram alimentados pelo gerador, e o laboratório de radiofrequência não é um deles.

Como é mostrado no gráfico, no dia 28 de abril a energia foi reestabelecida e os dados de telemetria passaram a ser novamente enviados ao servidor. Inicialmente os dados de referência informam “*unlocked*” pois o sinal de GPS não havia sido restaurado totalmente. Rapidamente (por isso o pico visto no gráfico) a referência foi encontrada e os dados retornaram ao estado “*locked*”.

Ocorreram algumas quedas pontuais devido a travamentos do servidor que hospeda o Zabbix na plataforma AWS. Tais quedas podem ser visualizadas no gráfico da figura 50 e variam de 6 a 16 horas. Especificamente, são causadas por falhas na verificação (feitas pela própria inteligência da plataforma) da integridade do servidor. É uma falha já discutida em fóruns e que não há uma solução efetiva por parte da empresa que oferece o serviço, sendo a recomendação para a resolução do problema um simples reinício da instância, o que foi feito e realmente funcionou.

Assim sendo, é possível dizer que o sistema montado em bancada respondeu da forma esperada e atendeu as demandas citadas no item 3 deste trabalho. Utilizando o método gráfico, foi possível calcular uma porcentagem de 94% de disponibilidade do sistema. Para chegar a esse número foi utilizada a fórmula descrita abaixo:

$$Disponibilidade \% = \left(1 - \frac{\text{horas sem dados de telemetria}}{\text{horas totais monitoradas}}\right) * 100$$

$$Disponibilidade \% = \left(1 - \frac{41}{720}\right) * 100$$

$$Disponibilidade = 94,31\%$$

A parte ativa do sistema de telemetria, ou seja, o envio de comandos via mensagens de texto, teve uma performance muito boa no que se diz respeito à interpretação dos comandos, e não houve momentos de indisponibilidade notados. O tempo para a execução dos comandos é variável, depende do tipo de parâmetro alterado.

No caso do reinício do transmissor, o processo demorou cerca de 2 minutos para o reestabelecimento de todos os módulos e, para a alteração de potência não mais que 30 segundos. A interface do aplicativo é amigável e fácil de operar, facilitando no momento de executar um comando.

6.2 MONITORAÇÃO EM CAMPO

As capacidades e limitações do sistema de telemetria desenvolvido foram postos à prova durante 15 dias na estação retransmissora de TV escolhida como piloto para o projeto. Ao instalá-lo em um ambiente de campo, conforme foi descrito no item 5.2 deste trabalho, pôde-se verificar que, apesar da confiabilidade das ferramentas utilizadas, é necessário desenvolver sistemas redundantes para que não haja falhas irreparáveis na monitoração, obrigando o deslocamento de um técnico para colocar a telemetria outra vez em funcionamento.

Problemas com variação de energia podem ocasionar um desligamento do sistema e, nesse caso, não se saberia se o problema ocorreu nas ferramentas de monitoração (*Raspberry Pi* e/ou rede móvel), ou mesmo se foi realmente uma queda de energia. Para solucionar esse problema deve-se alimentar o microcomputador com uma bateria externa que possa garantir o envio de dados, mesmo que haja falta de rede elétrica.

Também se mostrou necessária a instalação de um outro modem USB conectado a uma rede de operadora móvel diferente, uma vez que é relativamente normal que ocorram quedas ou variações no sinal móvel. O modem backup deverá estar conectado ao *Raspberry* em outra porta USB e assumir como principal caso seja percebido falta de conectividade à internet.

Apesar das melhorias que deverão ser feitas no sistema, em nenhum momento o mesmo tornou-se inoperante por muito tempo, apenas ocorreram momentos de ausência de sinal de internet móvel, impossibilitando o envio de informações através do proxy. Após o restabelecimento do sinal – que pode demorar algumas horas – os dados voltaram a ser transmitidos ao Zabbix Server.

É de se ressaltar que o sistema de comandos via Telegram obteve desempenho análogo ao da montagem em ambiente controlado. Praticamente não ocorreram momentos de indisponibilidade, somente em ocasiões de ausência de sinal de internet móvel. Todos os comandos foram executados e respondidos ao operador sem falhas.

7 CONCLUSÃO

O objetivo geral deste trabalho foi desenvolver um sistema de telemetria para a monitoração remota de estações retransmissoras de TV, utilizando a estrutura de uma emissora localizada na cidade de São Paulo. Nesse sentido, foi realizado um estudo das ferramentas de monitoramento que mais se adaptam ao escopo do projeto, modelando cada uma para atender a demanda proposta.

A base do sistema de telesupervisão é o protocolo simples de gerenciamento de redes, ou SNMP, que se adaptou muito bem ao sistema, e, dentro do contexto do trabalho, o mesmo é insubstituível, devido à compatibilidade com os dispositivos monitorados.

A plataforma Zabbix contribuiu de forma a oferecer um ecossistema altamente flexível e robusto de forma gratuita, que centralizou as informações enviadas pelos dispositivos remotos. Já o Zabbix Proxy estendeu as funcionalidades do servidor e ofereceu uma solução simples e confiável de monitoramento remoto.

Na ponta do sistema de telesupervisão encontra-se a plataforma de visualização de dados em forma gráfica, o Grafana. Nela foram concatenados e filtrados os dados enviados pelo servidor Zabbix e por fim apresentados a nível profissional na forma de gráficos que facilitam a análise. Tal plataforma se adaptou muito bem ao projeto proposto uma vez que é leve, rápida e versátil, podendo ser acessada de um simples smartphone, de qualquer lugar do mundo.

Na concepção do projeto havia um limitador relacionado à conectividade à internet, responsável por enviar os dados ao servidor online. A solução proposta de utilizar um modem USB e um chip móvel atendeu à demanda atual, visto que a cobertura das operadoras de telefonia móvel na região na qual foi instalado o sistema é satisfatória. Deve ser salientado que a qualidade do sinal para conexão móvel pode deteriorar-se dependendo do local e cidade na qual o sistema foi instalado. Cada estação retransmissora deve ser analisada de forma individual e o projeto poderá sofrer alterações para melhor atender cada cidade.

A plataforma desenvolvida é altamente escalável uma vez que a ponte entre os equipamentos monitorados e o servidor se trata de um computador conectado à internet – um Raspberry Pi - o que facilita o desenvolvimento de outras ferramentas de monitoração, como foi o caso da plataforma de envio de comandos via aplicativo de mensagens Telegram.

A telemetria ativa desenvolvida se portou muito bem ao ser instalada na estação retransmissora, ofereceu aos responsáveis técnicos da televisão um controle que antes não havia, podendo agora – por exemplo - realizar o reinício dos equipamentos, o que muitas vezes pode resolver um problema de falta de sinal em uma determinada cidade. Além disso, o sistema

programado no Raspberry é altamente moldável, podendo ser reprogramado caso haja a necessidade de inserção de novos comandos.

Como sugestão para melhoria futura do sistema, coloca-se a automatização dos comandos através da plataforma Zabbix, unificando os sistemas passivos e ativos e respondendo a mudanças repentinas no valor de parâmetros alterando-os para o valor de controle, e também enviando mensagens de alerta aos responsáveis técnicos da estação retransmissora de TV. Além disso, a troca do servidor hospedado na AWS também melhoraria a confiabilidade do sistema, uma vez que houve alguns travamentos na instância. Existem outras empresas no mercado que oferecem serviços similares e de alta qualidade, como a Google e a Microsoft, porém nenhuma das duas oferece a hospedagem gratuita para instâncias de pequeno porte.

O protótipo desenvolvido atendeu às expectativas, entregando uma plataforma simples e de fácil visualização, porém com dados importantíssimos para os responsáveis da manutenção de estações retransmissoras de televisão. Os dados informados irão garantir que uma estação está em pleno funcionamento, e a cidade em questão está sendo coberta pelo sinal irradiado da TV, independentemente de informações passadas por telespectadores, que, como explicado anteriormente, podem não representar a realidade. Isso poderá representar uma redução de custos de deslocamento de um técnico até a cidade. Ainda há a possibilidade de, caso haja algum problema na estação, executar um comando remoto reiniciando ou alterando algum parâmetro nos dispositivos, o que, em muitas ocasiões, é o suficiente para sanar a avaria detectada. Caso seja necessário o deslocamento do técnico, o sistema de monitoramento irá facilitar na escolha das ferramentas necessárias para realizar a manutenção.

Tendo em vista a importância da televisão para o público brasileiro e o papel fundamental que as estações retransmissoras possuem, pode-se dizer que o projeto desenvolvido irá aprimorar a manutenção das mesmas, aumentando a disponibilidade dos canais de televisão e levando conteúdo e programação aos telespectadores das cidades mais longínquas, muitos desses que ainda tem a TV como sua principal fonte de lazer.

REFERÊNCIAS

AMAZON WEB SERVICES. **User guide for Linux instances**. [Seattle, WA]: Amazon Web Services, [2022]. Disponível em: https://docs.aws.amazon.com/pt_br/AWSEC2/latest/UserGuide/concepts.html. Acesso em: 30 abr. 2022.

SBCODE. **Install and configure Zabbix proxy**. [S.l.]: SBCode, [2021]. Disponível em: <https://sbcode.net/zabbix/proxy-install/>. Acesso em: 22 fev. 2022.

BASSO, P. **Integrando Zabbix e Grafana**. [S.l.]: Medium, 2020. Disponível em: <https://medium.com/zabbix-brasil/integrando-zabbix-e-grafana-d46de4d1526d#:~:text=A%20integra%C3%A7%C3%A3o%20entre%20as%20ferramentas>. Acesso em: 30 abr. 2022.

CISCO. **Como funciona um switch?**. [São Francisco, CA]: Cisco Systems, [2022]. Disponível em: https://www.cisco.com/c/pt_br/solutions/small-business/resource-center/networking/network-switch-how.html. Acesso em: 21 abr. 2022.

ELIAS, G.; LOBATO, L. C. **Arquitetura e Protocolos de Rede TCP-IP**. 2. ed. Rio de Janeiro: RNP/ESR, 2013.

GÖRGEN, J. Redes de televisão e prefeituras: uma dominação consentida. *In*: COMPOLÍTICA – CONGRESSO DA ASSOCIAÇÃO BRASILEIRA DE PESQUISADORES DE COMUNICAÇÃO E POLÍTICA, II, 2007, Belo Horizonte. **Anais [...]**. Belo Horizonte: Compolítica, 2007. Disponível em: http://www.compolitica.org/home/wp-content/uploads/2011/01/sc_pc-james.pdf. Acesso em: 10 fev. 2022.

GOMES DE CARVALHO, Á.; DA COSTA BADINHAN, L. F. **Habilitação técnica em eletrônica**: Telecomunicações. São Paulo: Fundação Padre Anchieta, 2011.

IZEL, A. **Há 70 anos, a televisão foi inaugurada no Brasil; relembre a história**. [Brasília, DF]: Correio Braziliense, 2020. Disponível em: <https://www.correio braziliense.com.br/diversao-e-arte/2020/09/4876275-ha-70-anos-a-televisao-foi-inaugurada-no-brasil-relembre-a-historia.html#:~:text=Mas%20isso%20s%C3%B3%20aconteceu%20porque>. Acesso em: 17 abr. 2022.

KRIGER, D. **O que é Python, para que serve e por que aprender?**. [Curitiba, PR]: Kenzie Academy, [2022]. Disponível em: <https://kenzie.com.br/blog/o-que-e-python/>. Acesso em: 11 mar. 2022.

KUROSE, J. F.; ROSS, K. W. **Redes de computadores e a internet**: uma abordagem top-down. 6º ed. São Paulo: Pearson Education do Brasil, 2013.

ZABBIX. **Manual do Zabbix**. [Rīga: Rīgas rajons]: Zabbix SIA, [2022]. Disponível em: <https://www.zabbix.com/documentation/current/pt/manual>. Acesso em: 3 maio 2022.

Martinelli Hemmlepp, P. **Transporte MPEG-2**. [s.d]. Disponível em: <https://slideplayer.com.br/slide/2521253/>. Acesso em: 4 mar. 2022.

TELCOMANAGER. **O que é SNMP?**. [Rio de Janeiro, RJ]: Telcomanager, [2022]. Disponível em: <https://www.telcomanager.com/blog/o-que-e-snm/>. Acesso em: 21 fev. 2022.

PORTAL GSTI. **O que é SQLite**. [S.l]: Portal GSTI, [2022]. Disponível em: <https://www.portalgsti.com.br/sqlite/sobre/#:~:text=SQLite%20%C3%A9%20uma%20biblioteca%20de>. Acesso em: 16 maio 2022.

TELEGRAM. **Perguntas frequentes**. [Dubai, D]: Telegram, [2022]. Disponível em: <https://telegram.org/faq/br#p-o-que-e-telegram-o-que-faco-aqui>. Acesso em: 1 jun. 2022.

PINTO, P. **Dica: Como instalar o SNMP no Raspberry Pi**. [S.l]: Pplware, 2017. Disponível em: <https://pplware.sapo.pt/tutoriais/dica-como-instalar-o-snm-no-raspberry-pi/>. Acesso em: 8 maio 2022.

TECSYS BRASIL. **Receiver Decoder: TS 7200 HD**. [São José dos Campos, SP]: Tecsys do Brasil, [s.d]. Disponível em: <https://www.tecsysbrasil.com.br/receiver-ts7200hd/>. Acesso em: 9 abr. 2022.

HAIVISION. **Serial Digital Interface (SDI): Video Streaming Definition**. [Chicago, IL]: Haivision, [2022]. Disponível em: <https://www.haivision.com/resources/streaming-video-definitions/serial-digital-interface-sdi/>. Acesso em: 25 maio 2022.

CISCO MERAKI. **Signal-to-Noise Ratio (SNR) and Wireless Signal Strength**. [São Francisco, CA]: Cisco Meraki, 2020. Disponível em: [https://documentation.meraki.com/MR/WiFi_Basics_and_Best_Practices/Signal-to-Noise_Ratio_\(SNR\)_and_Wireless_Signal_Strength](https://documentation.meraki.com/MR/WiFi_Basics_and_Best_Practices/Signal-to-Noise_Ratio_(SNR)_and_Wireless_Signal_Strength). Acesso em: 20 mar. 2022.

READ THE DOCS. **SNMP library for Python 4.4 documentation: SNMP library for Python**. [S.l]: Read the Docs, [2018]. Disponível em: <https://pysnmp.readthedocs.io/en/latest/>. Acesso em: 10 jun. 2022.

SOARES A. *et al.* **Management Information Bases (MIBs)**. [Rio de Janeiro, RJ]: UFRJ, [s.d]. Disponível em: https://www.gta.ufrj.br/grad/10_1/snm/mibs.htm. Acesso em: 11 mar. 2022.

TANENBAUM, A. S.; WETHERALL, D. **Computer networks**. 5º ed. Vancouver, B.C: Langara College, 2017.

TELEGRAM. **Telegram APIs**. [Dubai, D]: Telegram, [2022]. Disponível em: <https://core.telegram.org/api>. Acesso em: 7 jun. 2022.

READ THE DOCS. **Telepot 12.7 documentation: Introduction**. [S.l]: Read the Docs, [2018]. Disponível em: <https://telepot.readthedocs.io/en/latest/>. Acesso em: 12 jun. 2022.

TECSYS BRASIL. **Transmissor TS6200 baixa e média potência**. [São José dos Campos, SP]: Tecsys do Brasil, [s.d]. Disponível em: <https://www.tecsysbrasil.com.br/receiver-ts7200hd/>. Acesso em: 29 abr. 2022.

TELECO. **TV Digital II: Modos de Transmissão e Modulação.** [São Paulo, SP]: Teleco, [2022]. Disponível em: https://www.teleco.com.br/tutoriais/tutorialtvdconsis2/pagina_3.asp. Acesso em: 6 mar. 2022.

DPS TELECOM. **Understanding the MIB.** [Fresno, CA]: DPS Telecom, [2022]. Disponível em: <https://www.dpstele.com/snmp/tutorial/mib.php>. Acesso em: 27 mar. 2022.

RASPBERRY PI FOUNDATION. **What is a Raspberry Pi?**. [Cambridge, Cambridgeshire]: Raspberry Pi Foundation, [s.d]. Disponível em: <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>. Acesso em: 20 abr. 2022.

HEADEND INFO. **What is IRD? How IRDs works? In digital headend system & features?**. [S.l]: Headend Info, [s.d]. Disponível em: <https://headendinfo.com/ird/>. Acesso em: 10 maio 2022.

WIRELESS TECHNOLOGY ADVISOR. **Wireless Telemetry Systems:** Deliver information from difficult locations. [S.l]: Wireless Technology Advisor, [2019]. Disponível em: <https://www.wireless-technology-advisor.com/wireless-telemetry-systems.html>. Acesso em: 12 maio 2022.

APÊNDICE A – Código da aplicação para telecomandos via Telegram

```
#FUNÇÃO RESPONSÁVEL POR CONTROLAR O ACESSO DOS USUÁRIOS, VISTO QUE OS
BOTS NO TELEGRAM SÃO PÚBLICOS.

# -*- coding: utf-8 -*-

id_a = [XXXXXXX] #IDENTIFICADOR ÚNICO CHAT_ID> UTILIZADO PARA LIMITAR
ACESSO

def retornar_logins():

    return id_a #RETORNA A LISTA DE CHAT_IDs

def login(chat_id):

    if chat_id in id_a:

        return 1 #RETORNA TRUE CASO O USUÁRIO QUE ACESSE O BOT POSSUA SEU
CHAT_ID NA LISTA

    else:

        return 0 #RETORNA FALSE CASO O USUÁRIO QUE ACESSE O BOT NÃO
POSSUA SEU CHAT_ID NA LISTA
```

```
#FUNÇÃO QUE RECEBE AS MENSAGENS DO APP E VERIFICA SE O USUÁRIO POSSUI
REQUISITOS PARA USAR O BOT E, CASO POSSUA, ALIMENTA A FUNÇÃO QUE REALIZA
OS COMANDOS COM A MENSAGEM RECEBIDA.
```

```
# -*- coding: utf-8 -*-
```

```
#IMPORTANDO BIBLIOTECAS NECESSÁRIAS
from pynmp.hlapi import *
from pynmp import debug
import time
from datetime import datetime
import telepot
from telepot.loop import MessageLoop
from login import login
from login import retornar_logins
from comandos import handle
```

```
def inicio():
```

```
    for id in retornar_logins(): #VERIFICANDO SE O ID DO USUÁRIO ESTÁ NA
LISTA DE ACESSO
```

```
        bot.sendMessage(id,"Serviço iniciado.\nTelemetria de volta ao
funcionamento normal em: "+datetime.now().strftime("%d/%m/%Y %H:%M"))
```

```
def telemetria_rede_do_interior(msg):
```

```
    chat_id = msg['chat']['id']
    command = msg['text']
```

```
    if login(chat_id):
```

```
        handle(chat_id, command) #EXECUTANDO COMANDOS CASO O ID DO
USUÁRIO ESTIVER NA LISTA
```

```
    else:
```

```
        bot.sendMessage(chat_id, 'Acesso negado!') #CASO O ID DO USUÁRIO
NÃO ESTIVER NA LISTA
```

```
bot = telepot.Bot('XXXXXXXXXXXXX') #INFORMANDO O TOKEN FORNECIDO PELO BOT
```

```
inicio()
```

```
MessageLoop(bot, telemetria_rede_do_interior).run_as_thread()
#FUNCIONAMENTO CONTÍNUO
```

```
while 1:
    time.sleep(10)
```

```

#FUNÇÃO RESPONSÁVEL POR ANALISAR A MENSAGEM ENVIADA PELO USUÁRIO,
ALIMENTANDO A FUNÇÃO setsnmp COM OS PARÂMETROS NECESSÁRIOS PARA A
EXECUÇÃO DOS COMANDOS.

# -*- coding: utf-8 -*-

#IMPORTANDO AS BIBLIOTECAS NECESSÁRIAS
from pysnmp.hlapi import *
from pysnmp import debug
import time
import random
from datetime import datetime
import telepot
from telepot.loop import MessageLoop
import subprocess
from pyasn1.type.univ import Integer
from pysnmp.proto.rfcl902 import *
from get_snmp import getsnmp
from set_snmp import setsnmp

def handle(chat_id, command): #RECEBENDO A MENSAGEM DO APLICATIVO
    TELEGRAM

    #CASO A MENSAGEM ATENDA A CONDIÇÃO, SERÁ EXECUTADO O COMANDO
    elif command == '/rfoff':
        bot.sendMessage(chat_id,"Desligando RF, aguarde...") #INFORMANDO
        O USUÁRIO

        setsnmp('192.168.0.100','.1.3.6.1.4.1.41913.1.2.2.15.20.3.0',Integer(1))
        #EXECUTANDO O COMANDO NO DISPOSITIVO VIA PROTOCOLO SNMP
        time.sleep(10)
        bot.sendMessage(chat_id,"RF Desligado com sucesso!") #INFORMANDO
        O USUÁRIO

        elif command == '/rfon':
            bot.sendMessage(chat_id,"Ligando RF, aguarde...")

            setsnmp('192.168.0.100','.1.3.6.1.4.1.41913.1.2.2.15.20.3.0',Integer(0))
            time.sleep(10)
            bot.sendMessage(chat_id,"RF ligado com sucesso!")

            elif command == '/resettx':
                bot.sendMessage(chat_id,"Reiniciando o TX, aguarde...")

                setsnmp('192.168.0.100','.1.3.6.1.4.1.41913.1.1.7.1.1.0',Integer(1))
                time.sleep(90)
                bot.sendMessage(chat_id,"TX reiniciado com sucesso!")

                elif command[0:4] == '/pot':
                    bot.sendMessage(chat_id,"Alterando potência para
                    "+str(command[4:])+ "W, aguarde...")

                    setsnmp('192.168.0.100','.1.3.6.1.4.1.41913.1.2.4.1.0',OctetString(str(co
                    mmand[4:])))

```

```

        time.sleep(10)
        bot.sendMessage(chat_id,"Potência alterada com sucesso!")

    elif command == '/rxreset':
        bot.sendMessage(chat_id,"Reiniciando o receptor, aguarde...")

setsnmp('192.168.0.101','.1.3.6.1.4.1.41913.3.1.3.6.2.0',Integer(1))
    time.sleep(90)
    bot.sendMessage(chat_id,"Receptor reiniciado com sucesso!")

    elif command == '/consumo':
        bot.sendMessage(chat_id,str(subprocess.check_output('vnstat -s -m
-i usb0', shell=True)))

    else:
        bot.sendMessage(chat_id, "Comando inválido...")

bot = telepot.Bot('XXXXXXXXXXXXXXXXXXXXXXXXXXXX') #DEFININDO O TOKEN
INFORMADO NA CRIAÇÃO DO BOT PARA A RECEPÇÃO DAS MENSAGENS

```

```

#FUNÇÃO RESPONSÁVEL POR REALIZAR A CONEXÃO COM O DISPOSITIVO SNMP,
ENVIANDO A OID E O NOVO VALOR DO PARÂMETRO.

# -*- coding: utf-8 -*-
# IMPORTANDO BIBLIOTECAS NECESSÁRIAS
from pysnmp.hlapi import *
from pysnmp import debug
import time
import random
from datetime import datetime
import telepot
from telepot.loop import MessageLoop
import subprocess
from pyasn1.type.univ import Integer
from pysnmp.proto.rfc1902 import *

#FUNÇÃO QUE IMPLEMENTA A FUNCIONALIDADE SET DO PROTOCOLO SNMP
def setsnmp(host,oid,valor):

    errorIndication, errorStatus, errorIndex, varBinds = next(
        setCmd(SnmpEngine(),
            CommunityData('private', mpModel=1), #DEFININDO A COMUNIDADE
            UdpTransportTarget((host,161)), #DEFININDO O DESTINATÁRIO
            ContextData(),
            ObjectType(ObjectIdentity(oid), valor) #INFORMANDO O
PARÂMETRO E NOVO VALOR
        )
    )

```