UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADEMICO DE INFORMÁTICA
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

ALCEU DEGGERONE

IMPLEMENTAÇÃO DE UM SISTEMA DE MONITORAMENTO UTILIZANDO ZABBIX NA REDE LOCAL DE ENSINO (RLE)

TRABALHO DE CONCLUSÃO DE CURSO

Curitiba - PR
2016

ALCEU DEGGERONE

IMPLEMENTAÇÃO DE UM SISTEMA DE MONITORAMENTO UTILIZANDO ZABBIX NA REDE LOCAL DE ENSINO (RLE)

Monografia apresentada à disciplina de Trabalho de Conclusão de Curso de Bacharelado em Sistemas de Informação da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Fabiano Scriptore de Carvalho, MSc.

Curitiba - PR

2016

Ministério da Educação
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
Câmpus Curitiba
Diretoria de Graduação e Educação Profissional
Departamento Acadêmico de Informática
Coordenação do Curso de Bacharelado em Sistemas de Informação

TERMO DE APROVAÇÃO

“IMPLEMENTAÇÃO DE UM SISTEMA DE MONITORAMENTO UTILIZANDO ZABBIX NA REDE LOCAL DE ENSINO (RLE)”

por

“Alceu Deggerone”

Este Trabalho de Conclusão de Curso foi apresentado no dia **5** de **dezembro** de **2016** como requisito parcial à obtenção do grau de Bacharel em Sistemas de Informação na Universidade Tecnológica Federal do Paraná - UTFPR - Câmpus Curitiba. O(a)(s) aluno(a)(s) foi(ram) arguido(a)(s) pelos membros da Banca de Avaliação abaixo assinados. Após deliberação a Banca de Avaliação considerou o trabalho _____.

_____ **Prof. Fabiano Scriptore de Carvalho** (Presidente - UTFPR/Curitiba)	_____ **Prof. Anelise Munaretto Fonseca** (Avaliador 1 – UTFPR/Curitiba)
_____ **Prof. Luiz Augusto Pelisson** (Avaliador 2 - UTFPR/Curitiba)	_____ **Prof. Leyza Elmeri Baldo Dorini** (Professor Responsável pelo TCC – UTFPR/Curitiba)
_____ **Prof. Leonelo Dell Anhol Almeida** (Coordenador do curso de Bacharelado em Sistemas de Informação – UTFPR/Curitiba)	

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso.”

RESUMO

Este trabalho apresenta uma proposta de implementação de um sistema de gerência que poderá ser utilizado na Rede Local de Ensino (RLE) do Departamento Acadêmico de Informática do campus Curitiba da UTFPR, pois o gerenciamento de redes torna-se cada vez mais complexo. Essa implementação vem para auxiliar o administrador da rede que terá um maior controle das redes de computadores obtendo um melhor desempenho. Realizou-se um breve estudo sobre gerenciamento de redes, descobertas de serviços e o protocolo SNMP. Foram analisadas as ferramentas Cacti e Zabbix, para definir qual delas seria utilizada como ferramenta de monitoramento. Usou-se a ferramenta Nmap, bem como analise das configurações dos elementos de rede para auxiliar no mapeamento da rede, e seus principais serviços do RLE. Finalmente, houve a implantação da ferramenta para monitoramento da rede com o objetivo de auxiliar o administrador a obter informações necessárias para recuperação das falhas que podem ocorrer em seus serviços e elementos de redes, e ainda, para ter uma noção da ocupação da banda das interfaces da rede principal do RLE.
Palavras-chave: SNMP, redes, gerência, Zabbix, Nmap.

ABSTRACT

This monograph presents a brief study on network management, OSI reference model, service discovery and SNMP. Also an analysis of Cacti tools, Zabbix and Nmap, to make a map of the network and its core services for Information Technology department of Paraná Federal Technology University and the implementation of one network monitoring tool purpose assisting the administrator to obtain the necessary information for recovery of failures that can occur in its services and network elements.

LISTA DE FIGURAS

LISTA DE SIGLAS

DAINF	Departamento Acadêmico de Informática
DNS	Domain Name System
HTTP	HyperText Transfer Protocol
ICMP	Internet Control Message Protocol
IP	Internet Protocol
ISO	International Organization for Standarzation
LAN	Local Area Network
MAC	Media Access Control
MIB	Management Information Base
OID	Object Identifier
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
PING	Packet Internet Groper
QoS	Quality of Service
RLE	Rede Local de Ensino
TCP	Transmission Control Protocol
TTL	time to live
UDP	User Data Protocol
UTFPR	Universidade Tecnológica Federal do Paraná
SNMP	Simple Network Management Protocol
VBL	Variable Binding List

SUMÁRIO

1 INTRODUÇÃO

1.1 PROBLEMA

As redes de computadores, por serem parte essencial para todas as empresas, tendem a ficar cada vez maiores e mais complexas (STALLINGS, 2005). Isto implica em um maior número de usuário e de serviços, que facilmente são aglutinados aos seus processos, passando a ser indispensáveis para estas instituições. O impacto disto é que quanto maior e mais complexa for a rede, maiores são as chances de haverem problemas que podem ocasionar uma indisponibilidade parcial ou, até mesmo total.

Uma rede local é a interconexão de diversos dispositivos em uma rede de computadores que concede um meio de troca de informações entre esses dispositivos (STALLINGS, 2005).

Redes complexas, como o próprio nome diz, são muito difíceis de serem gerenciadas somente por pessoas. Com isso é necessário o uso de ferramentas que irão auxiliar em todo o processo de gerenciar. As utilidades desses artefatos são das mais diversas e se fazem cada vez mais necessárias devido a difícil tarefa de garantir a eficiência e coordenação da rede, aliado a junção de equipamentos dos mais diversos fabricantes e as diferentes distribuições das aplicações entre clientes e servidores (STALLINGS, 2005).

O gerenciamento da rede deve ser considerado tanto quanto o projeto da rede (BIRKNER, 2003). O controle e coleta de dados são vitais para um gerenciamento de LAN, assim sendo é indispensável que a rede tenha ferramentas de apoio que auxiliarão nessas tarefas. Com este intuito a *International Organization for Standarzation* (ISO) propôs tópicos centrais, os quais abrangem, principalmente, as áreas funcionais para um bom funcionamento de redes. São elas:

- Gerenciamento de Falhas;
- Gerenciamento de Contabilidade;
- Gerenciamento de Configurações;
- Gerenciamento de Desempenho;
- Gerenciamento de Segurança.

Com isso, não basta somente implementar uma rede conforme a metodologia desejada sem ter ferramentas para fazer o gerenciamento da mesma. Com as ferramentas corretas, um administrador de redes pode fazer o controle da rede de forma mais direta. O plano de gerenciamento permite que possam ser feitas verificações periódicas nas entidades, detectando falhas e defeitos, podendo minimizar os efeitos fazendo a proteção do sistema, propagando a informação de falha ou mau desempenho e realizando testes para poder achar a exata localização da falha (SOARES; LEMOS; COLCHER, 1995).

1.2 JUSTIFICATIVA

Ao longo dos estudos no curso de Bacharelado em Sistemas de Informação (BSI) na Universidade Tecnológica Federal (UTFPR) houve um momento em que surgiu a oportunidade de visitar a unidade do PoP-PR (*Point of Presence* - Paraná) que é um dos pontos de presença do Brasil, que tem como missão prover conectividade à infraestrutura nacional de alto desempenho. Nesta ocasião tivemos a chance de conhecer de perto uma estrutura robusta de rede, a qual era gerenciada por algumas ferramentas que propiciavam um tempo de resposta a erros muito baixos, fazendo com que as falhas não fossem percebidas pelos usuários, na maioria dos casos.

Com o intuito de analisar a aplicabilidade de tais níveis de gerenciamentos na Rede Local de Ensino do Departamento Acadêmico de Informática (RLE-DAINF) este trabalho propõe-se a estudar a viabilidade de aplicação de três protocolos de gerenciamento de redes (ICMP, SNMP e a descoberta de serviços através do protocolo ICMP), fazer o mapeamento da RLE e fazer um cruzamento de funcionalidades dos protocolos com as devidas compatibilidades dos equipamentos da rede em questão.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

O objetivo geral desse trabalho é realizar a implantação de um sistema de gerência de redes de computadores utilizando uma ferramenta de monitoramento (Zabbix) na Rede Local de Ensino (RLE) do Departamento Acadêmico de Informática (DAINF) do campus Curitiba, da Universidade Tecnológica Federal do Paraná (UTFPR).

1.3.2 Objetivos Específicos

- Fazer um estudo sobre os principais protocolos de gerenciamento de redes utilizados atualmente;
- Utilizar as ferramentas *Ping*, *Traceroute* e *Nmap* para a realização do mapeamento da rede de computadores e dos seus serviços;
- Fazer um levantamento de equipamentos da RLE;
- Elaborar o mapeamento físico e lógico da infraestrutura de redes de computadores da RLE;
- Implementar o sistema de monitoramento Zabbix para gerenciar os elementos de redes.

Os objetivos específicos deste trabalho serviram de base para a implantação de ferramenta para monitoramento, realizar um mapa da rede principal e seus serviços que melhor atender nosso objetivo geral

1.4 METODOLOGIA

A metodologia aplicada neste trabalho envolve a organização de um conjunto de atividades, conforme descrito a seguir:

a) **Pesquisas bibliografias relacionadas a fundamentação teórica:**
   Foram realizadas pesquisas com base em livros, artigos de congressos e de revistas técnicas e científicas a respeito do tema do trabalho proposto. A intenção é fazer um levantamento técnico e científico, para que o trabalho fique devidamente fundamentado.
b) **Compreender e comparar os protocolos de gerenciamento de redes:**
   Foi realizado o paralelo entre os protocolos ICMP, SNMP e a descoberta de portas e serviços utilizando ICMP, levantando suas principais características e funções;
c) **Realizar o mapeamento dos equipamentos da Rede Local de Ensino:**
   Realizou-se *in loco* o levantamento da abrangência da rede e dos equipamentos nela inseridos, utilizando as ferramentas *PING*, *Traceroute* e *Nmap*, bem como a compatibilidade dos mesmos com os protocolos estudados.
d) **Apresentar o resultado obtido do mapeamento para o gestor da RLE:**
   Apresentou-se o mapa da rede para o gestor responsável pela Rede Local de Ensino a fim de pontuar os equipamentos e serviços a serem monitorados.
e) **Eleger qual(is) protocolos se adequam melhor na RLE:**
   Pontuar os porquês de tal(is) protocolo(s) terem sido eleitos a ser utilizada no trabalho de acordo com suas prévias avaliações de aplicabilidade.

1.5 ORGANIZAÇÃO DO DOCUMENTO

Este documento é composto por seis capítulos. O capitulo 1 contempla a introdução da proposta, descrevendo a justificativa, os objetivos, a metodologia. No capítulo 2 está sendo descrito o estado a arte do trabalho.

O terceiro capítulo traz o referencial teórico da proposta com temas como: redes de computadores, o modelo relacional OSI, gerenciamento de redes, SNMP e a descoberta de *host* e serviços.

O capitulo 4 apresenta as opções de ferramentas a serem utilizadas e/ou implementadas no projeto.

O capitulo cinco apresenta o desenvolvimento da parte prática com mapeamento de rede, mapeamento de serviços e implantação da ferramenta Zabbix. E por fim, no sexto capítulo encontra-se o referencial bibliográfico.

2 ESTADO DA ARTE

Há vários trabalhos relacionados ao uso do protocolo SNMP e com gerência de redes. Na tese (TAROUCO, 2011) a autora discute as limitações dos modelos tradicionais para o gerenciamento de rede, aprofundando no contexto de redes atuais, que possuem particularidades distintas, e por consequência acabam por demandar requisitos de gerenciamento específicos, os quais não são identificados nas redes tradicionais.

No trabalho (BLACK, 2008), Black comparou nove ferramentas (RRD, ZENOSS, ManageOP Engine, BigBrother4, SpiceWorks, Look@LAN, Zabbix e Nagios) de gerenciamento e monitoração de redes, com parâmentos como performance, facilidade de utilização e necessidade de recursos, com o objetivo de ajudar os gerentes de redes a escolher a ferramenta que mais atende suas necessidades. Os resultados obtidos foram apresentados em uma tabela contendo as principais características que cada ferramenta apresenta ou não.

O artigo (TELTUMDE & MESHRAM, 2012) visa explicar e projetar um *software* de monitoramento de redes que permita que a tarefa de monitorar seja feita de forma fácil e eficiente, discorrendo a respeito dos aspectos que são monitorados manualmente e que devem ser monitorados de forma automatizada.

Já o artigo (SCHÖNWÄLDER & MARINOV, 2011) discute como os protocolos de segurança existentes, que operam acima da camada de transporte e abaixo dos protocolos de aplicação, podem ser usados para assegurar o SNMP. Nele os autores fazem uma análise minuciosa de uma implementação protótipo, comparando o desempenho do SNMPv3 aliado ao uso do SSH, TLS e DTLS com outras versões do SNMP. Também é abordado a diferença entre as várias opções para proteger o SNMP e prover diretrizes na escolha da solução a ser implementada.

3 EMBASAMENTO TEÓRICO

A seguir serão abordados temas relacionados com o referencial teórico, sendo eles: redes de computadores, o protocolo OSI, gerenciamento de redes, protocolo SNMP e a descoberta de hosts e de serviços.

3.1 REDES DE COMPUTADORES

O escopo de redes de computadores contempla diversos tipos de redes, desde as mais comuns até as menos conhecidas. Basicamente elas se diferenciam por terem diferentes objetivos, escalas e tecnologias.

A forma mais comum de uso das redes de computadores é a com a função de interligar computadores pessoais, estações de trabalho e instalações de empresas, com o intuito de trocar informações e compartilhar recursos. As redes locais (LANs) se diferem dos outros tipos de redes por três motivos: tem um tamanho restrito, as tecnologias de transmissão normalmente utilizam como meio de transmissão um cabo que conecta as maquinas, e a possibilidade se de ser implementada em diversas topologias (TANENBAUM, 2003).

Podemos destacar algumas topologias físicas no escopo de LANs e de redes de malhas:

- **Rede em Estrela**: A rede se configura com a figura de um nó central ligado a todos os outros nós e por onde é passado todas as mensagens (Soares & G Colcher, 1995);
- **Rede em Anel**: Os dispositivos são conectados em série, formando um circuito fechado. Eles trabalham como repetidores, até que a mensagem seja retirada da rede pelo nó de destino (Soares & G Colcher, 1995);
- **Rede em Barramento**: Todos os nós se ligam ao mesmo meio de transmissão e cada nó pode ver todas as informações transmitidas (Soares & G Colcher, 1995);
- **Malha Total**: Consiste na comunicação total entre os equipamentos, ou seja, em uma rede com cinco elementos, cada um deles possuirá quatro conexões distintas. Quando um novo nó é inserido este terá que ser ligado a todos os outros, fato que torna este tipo de topologia mais onerosa, com custo e grau de complexidade muito elevada para redes muito grandes, devido a grande quantidade de cabos utilizados e com um alto grau de redundância;

- **Malha Parcial**: Topologia na qual a conexão entre os equipamentos da rede é feita de forma em que todos os nós possam se comunicar, e com um grau de redundância aceitável, porém não é preciso que todos os nós estejam conectados entre si.

3.2 O MODELO REFERENCIAL OSI

O modelo referencial OSI (*Open Systems Interconnection*) trata-se de uma proposta desenvolvida pela ISO com o objetivo de padronizar a nível internacional, os protocolos empregados nas diversas camadas (TANENBAUM, 2003). Ele se dispõe em sete camadas (figura 2) que foram concebidas segundo a aplicação dos seguintes princípios:

- Uma camada deve ser criada onde houver necessidade de outro grau de abstração;
- Cada camada deve executar uma função bem definida;
- A função de cada camada deve ser escolhida tendo em vista a definição de protocolos padronizados internacionalmente;
- Os limites de camadas devem ser escolhidos para minimizar o fluxo de informações pelas interfaces;
- O número de camadas deve ser grande o bastante para que funções distintas não precisem ser desnecessariamente colocadas na mesma camada e, pequeno o suficiente para que a arquitetura não se torne difícil de controlar (TANENBAUM, 2003).

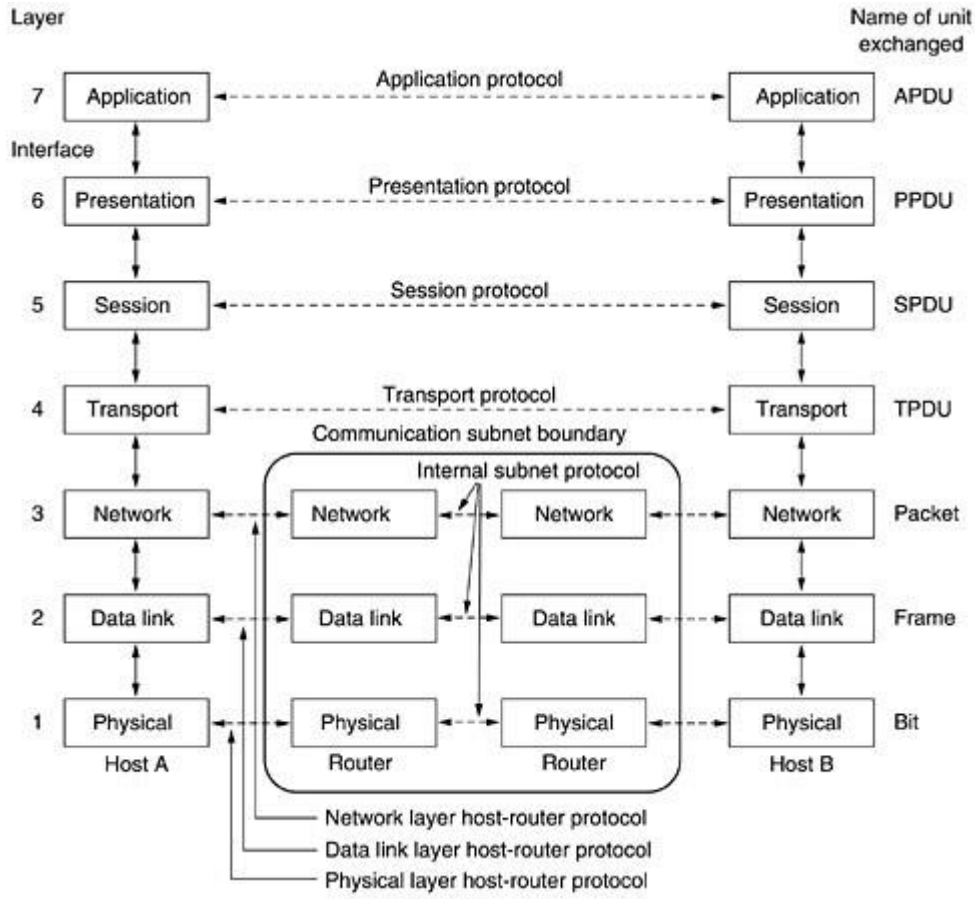

Figura 1 - O modelo de referência OSI
Fonte: (TANENBAUM, 2003).

A seguir serão abordadas as funções e características de cada uma das sete camadas.

3.2.1 Camada Física

Os protocolos da camada Física (camada 1) descrevem os meios mecânicos, elétricos, funcionais e procedimentais para ativar, manter e desativar conexões físicas para transmissão de bits a partir de um dispositivo de rede (Cisco, 2015). Ela fornece os requisitos para transportar, pelo meio físico de rede, os bits que formam o quadro da camada de Enlace de Dados. Essa camada aceita um quadro completo da camada de Enlace de Dados e o codifica como uma série de sinais que serão transmitidos para o meio físico local. Os bits codificados que formam um quadro são recebidos por um dispositivo final ou por um dispositivo intermediário.

A entrega de quadros pelo meio físico local exige os seguintes elementos da camada Física:

- Meio físico e conectores ligados;
- Representação de bits no meio físico;
- Codificação de dados e informações de controle;
- Circuito transmissor e receptor nos dispositivos de rede.

3.2.2 Camada de Enlace de Dados

Os protocolos de camada de Enlace de Dados (camada 2) descrevem métodos para trocar quadros de dados entre dispositivos através de um meio físico comum (Cisco, 2015). A Camada de Enlace realiza dois serviços básicos:

- Permite às camadas superiores acessarem o meio usando técnicas como enquadramento;
- Controla como o dado é colocado sobre o meio e é recebido do meio usando técnicas como o controle de acesso ao meio e detecção de erros.

3.2.3 Camada de Rede

A camada de Rede (camada 3) fornece serviços para trocar pedaços individuais de dados através da rede entre dispositivos finais identificados (Cisco, 2015). Ela fornece serviços para realizar trocas de fragmentos individuais de dados na rede entre dispositivos finais identificados. Para realizar este transporte de uma extremidade à outra, a camada três utiliza quatro processos básicos:

- Endereçamento;
- Encapsulamento;
- Roteamento;
- Desencapsulamento.

3.2.4 Camada de Transporte

A camada de Transporte (camada 4) define os serviços para segmentar, transferir e reunir os dados para comunicações individuais entre dispositivos finais (Cisco CCNA, 2015).

Ela proporciona a segmentação de dados e o controle necessário para reagrupar esses segmentos em fluxos de comunicação. Suas responsabilidades primárias para realizar isto são:

- Rastrear a comunicação individual entre as aplicações nos hosts de origem e destino;
- Segmentar dados e gerenciar cada segmento;
- Reagrupar os segmentos em fluxos de dados de aplicação;
- Identificar as diferentes aplicações.

3.2.5 Camada de Sessão

A camada de Sessão (camada 5) fornece serviços à camada de Apresentação para organizar seu diálogo e para gerenciar a troca de dados (Cisco, 2015). Suas funções criam e mantêm diálogos entre as aplicações de origem e destino. A camada de Sessão lida com a troca de informações para iniciar diálogos, mantê-los ativos e reiniciar sessões interrompidas ou ociosas por um longo período.

3.2.6 Camada de Apresentação

A camada de Apresentação (camada 6) fornece uma representação comum de dados transferidos entre serviços da camada de Aplicação (Cisco, 2015). A camada de Apresentação tem três funções principais:

- Codificação e conversão de dados da camada de Aplicação para garantir que os dados do dispositivo de origem possam ser interpretados pela aplicação adequada no dispositivo de destino;
- Compressão dos dados de forma que eles possam ser descomprimidos pelo dispositivo de destino;
- Criptografia dos dados para transmissão e decodificação de dados quando o destino os recebe.

3.2.7 Camada de Aplicação

A camada de Aplicação (camada 7) fornece os meios para conectividade ponto-a-ponto entre indivíduos na rede humana, usando redes de dados (Cisco, 2015). É a camada que fornece a interface entre as aplicações que utilizamos para comunicação e a rede subjacente pela qual nossas mensagens são transmitidas. Os protocolos da camada de aplicação são utilizados para troca de dados entre programas executados nos hosts de origem e de destino. Há muitos protocolos da camada de Aplicação, e outros novos estão em constante desenvolvimento.

3.3 GERENCIAMENTO DE REDES

Os requisitos de gerenciamento de redes podem ser divididos em cinco principais áreas:

- Gerenciamento de Falhas;
- Gerenciamento de Contabilidade;
- Gerenciamento de Configurações;
- Gerenciamento de Desempenho;
- Gerenciamento de Segurança.

3.3.1 Gerenciamento de Falhas

Para o funcionamento de uma rede é preciso manter correto o sistema globalmente como parcialmente, para quando ocorrer a falha, a correção aconteça o mais breve possível (STALLINGS, 2005). Com base nessa urgência, é preciso seguir algumas regras:

- Definir corretamente o local da falha;
- Encapsular a falha do resto da rede;
- Ajustar a rede para minimizar os impactos;
- Restaurar ou substituir elemento falho.

Para isso é preciso fazer a distinção entre a falha e o erro. Falha é uma anomalia que requer ação do gerente para ser resolvido. A falha apontada pelo mau funcionamento da rede

com alta proporção de erros. (Stallings, 2005). Por exemplo: se um cabo de rede romper, ocasionará uma elevada taxa de erros.

O cliente necessita e espera uma resposta rápida ao problema, para isso faz-se necessário a implementação de serviço de detecção de falhas com diagnóstico ágil e correto.

A redundância é uma alternativa para minimizar os danos causados por falhas. Após a correção da mesma é indispensável garantir que o problema foi solucionado e que problemas similares não ocorrerão (Stallings, 2005).

3.3.2 Gerenciamento de Contabilidade

O gerente de redes precisa saber quanto recurso cada o usuário ou grupo usa da rede por algumas razões.

- Um usuário ou grupo pode estar abusando dos seus privilégios, e com isso causando problemas para os outros;
- Podem estar usando de maneira incorreta a rede, o gerente pode auxiliar para um uso eficaz;
- O gerente poderá planejar o progresso da rede com eficiência, se ele tiver controle da usabilidade do serviço de cada usuário ou grupo.

O gerente de redes precisa saber qual informação pode ser acessada por determinado usuário ou grupo, impondo assim, as limitações. E ainda, determinar a hierarquia das informações propondo um escalonamento de prioridade para cada nó (Stallings, 2005).

3.3.3 Gerenciamento de Configuração

O gerenciamento de configuração é uma grande função dentro de gerenciamento de rede. Tem como seu principal objetivo monitorar a estrutura de rede tanto lógico quanto físico. Para realizar o controle de hardware e software desde que sejam gerenciáveis. Um inventário de equipamentos e programas é mantido e atualizado regularmente (Stallings, 2005).

O gerenciamento de configuração não é apenas a tecnologia para coletar informações a respeito do dispositivo, mas também sobre os processos necessários para suporte e operações de rede (Forouzan, 2007) (Stallings, 2005).

Pode ser resumido como:

- Coleta de *hardware* de dispositivos e de inventário de *software*;
- Gerenciamento de *software* de dispositivos;
- Coleta de configuração do dispositivo, *backup*, visualização, arquivo, comparação;
- Detecção de alterações na configuração, *hardware* ou *software*;

Implementação da mudança de configuração para suportar o gerenciamento de mudanças (Stallings, 2005)

3.3.4 Gerenciamento de Desempenho

A gestão de desempenho envolve a coleta periódica de métricas de qualidade de serviço *QoS* que caracterizam o desempenho do sistema de rede e recursos. Um dos seus objetivos é encontrar o congestionamento da rede e seus pontos de estrangulamento, e minimizá-los. Essas medidas tornam a rede disponível para que o desempenho possa ser mantido em limites aceitáveis. Aplica-se o monitoramento constante da rede em busca de tendências, os parâmetros são monitorados e registrados; estes incluem: taxa de transmissão de dados (taxa de transferência), as taxas de erro, o tempo de inatividade / *uptime*, percentagens de tempo de uso e tempo de resposta para verificar como está a saúde da rede (Stallings, 2005).

3.3.5 Gerenciamento de Segurança

Controla o acesso aos recursos da rede, conforme estabelecido pelas diretrizes de segurança da organização. Estão preocupados principalmente com o acesso autenticado e autorizado à rede, bem como a criptografia de dados; ou seja, controlar todo o acesso e garantir proteção tanto a rede como aos dispositivos individuais, contra o abuso intencional ou acidental, acesso não autorizado e perda de comunicação (Stallings, 2005).

3.4 SNMP

Uma das maiores funções de um administrador de rede é coletar informações precisas a respeito dos serviços e da infraestrutura que compõem sua rede. Existem várias ferramentas e opções que facilitam a coleta de processamento deste tipo de informações e boa parte delas utilizam a tecnologia conhecida como SNMP.

Simple Network Management Protocol, também conhecido com SNMP, é um protocolo implementando na 7ª camada do modelo de referência OSI (camada de aplicação), (ilustrado na figura 1) e é a maneira que os servidores e equipamentos podem disponibilizar informações sobre o estado atual. É um canal pelo qual a figura do administrador pode modificar valores pré-definidos.

Existem várias versões do protocolo SNMP, a primeira foi adotada como padrão em 1989 e possuía falhas no quesito de segurança. O SNMPv2 (versão 2) fornece gerenciamento de rede centralizado e distribuído incluindo aprimoramentos na sua estrutura e gerenciamento. O SNMPv3 (versão 3), foi criado para solucionar as questões de segurança da primeira e segunda versão, fornecendo acesso seguro às informações de gerenciamento por meio de autenticação e criptografia de pacotes.

O protocolo SNMP foi criado com o propósito de ser capaz de coletar informações de diferentes sistemas, tanto em suas funções como em suas próprias estruturas. Isso é possível devido ao método e o caminho de consulta das informações relevantes terem sido padronizados.

A arquitetura de gerência utilizando o SNMP conta com quatro elementos básicos (figura 2):

- Os nós gerenciados (agentes);
- As estações de gerenciamento (gerentes);
- As informações de gerenciamento (MIBs);
- O protocolo de gerenciamento (SNMP).

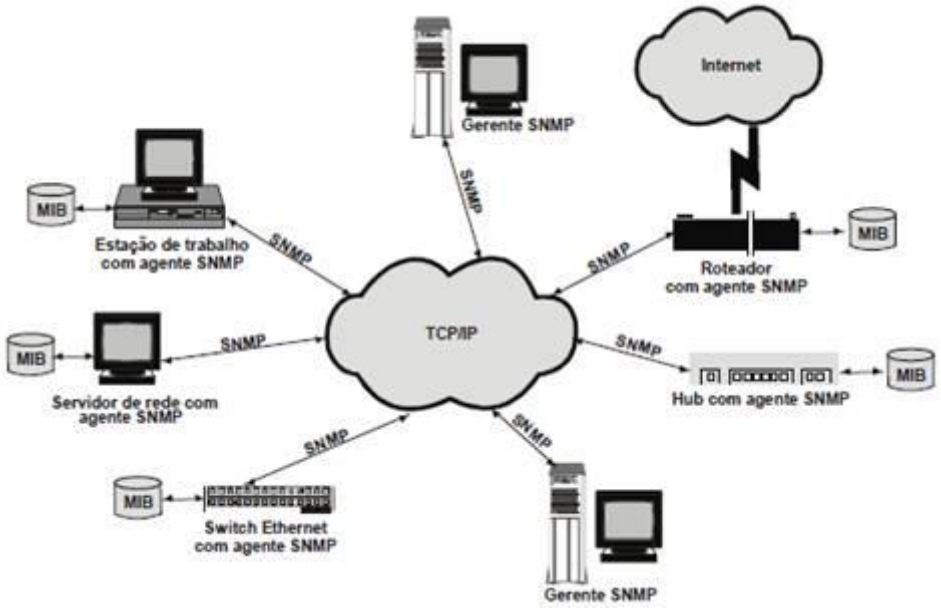

Figura 2 - A estrutura de funcionamento dos Gerentes e Agentes através do protocolo SNMP.
Fonte: (Teleco, s.d.).

Os gerentes SNMP são *software*s executados em uma ou mais estações capazes de realizar tarefas de gerenciamento da rede, sendo responsáveis por enviar *pollings (requests)* às estações agentes e receber as respostas a estes *pollings (responses).* Podendo ainda acessar (*get*) ou modificar (*set*) informações nos agentes e receber, mesmo sem requisição, informações relevantes ao gerenciamento (*traps*).

Os agentes SNMP são instalados nos dispositivos gerenciáveis da rede, que podem ser quaisquer componentes de *hardware* conectados a ela, tais como computadores (*hosts*), impressoras, *hubs*, *switches*, roteadores, entre outros. Os agentes interagem diretamente com a MIB (*Management Information Bases*) e são responsáveis por responder às solicitações feitas pelos gerentes (*pollings*) através de ações (*responses*). Eles também podem enviar, assincronamente, informações (*traps*) aos gerentes, quando ocorre algum problema sério ou um evento relevante para o gerenciamento da rede.

A MIB é um banco de dados lógico que armazena informações estatísticas de configuração e de *status*, relativas a todos os possíveis objetos gerenciáveis da rede. Tais objetos (variáveis) possuem nome, atributos e um conjunto de operações que podem ser realizadas sobre estes objetos, sendo descritas pela linguagem abstrata de definição de tipo de dados ASN.1.

A comunicação entre agentes e gerentes SNMP (figura 3) é feita com a troca de mensagens, sendo cada mensagem representada inteira e independentemente dentro de um pacote UDP (*User Data Protocol*), que é utilizado como protocolo de transporte. Esta

mensagem consiste de um identificador da versão, o nome da comunidade SNMP e a PDU (Protocol Data Uni) na comunicação entre NMS e Agente. Um dos motivos pela escolha de um protocolo que não oferece garantia na comunicação é o fato do UDP ter menos *overhead*, reduzindo assim o impacto do sistema de gerência no desempenho da rede. Por padrão tem-se as portas 161 e 162. A primeira tem como função o envio e recebimento de requests, e a segunda recebe as informações dos agentes, alocados nos equipamentos monitorados.

O tipo da PDU determina o tipo da transação ou operação SNMP a ser realizada. Cada PDU tem um único identificador de *request* que é usado para sua identificação. Os campos *error-status* e *error-index* são utilizados para armazenar informações de erro relativas à PDU. O último e mais importante campo da PDU é a carga útil (*payload*) ou VBL (*Variable Binding List*). Neste estão inclusas todas as variáveis SNMP e seus valores associados. Estas variáveis são as informações propriamente ditas que os gerentes leem, escrevem e relatam. Toda operação SNMP requer uma VBL para especificar precisamente a informação sendo acessada ou modificada.

Os tipos de PDU são:

- ✓ *Get request*: usado para solicitar o valor de uma ou mais variáveis da MIB;
- ✓ *Get-next request*: usado para solicitar os valores de um conjunto sequencial de variáveis da MIB e, após a solicitação do primeiro valor usando o comando *get*, os valores seguintes são solicitados usando este comando;
- ✓ *Set request*: usado para atribuir um valor a uma variável da MIB;
- ✓ *Get response*: usado para enviar resposta aos comandos *get*, *get-next* e *set*;
- ✓ *Trap*: usado para enviar informações de alarme ou eventos significativos.

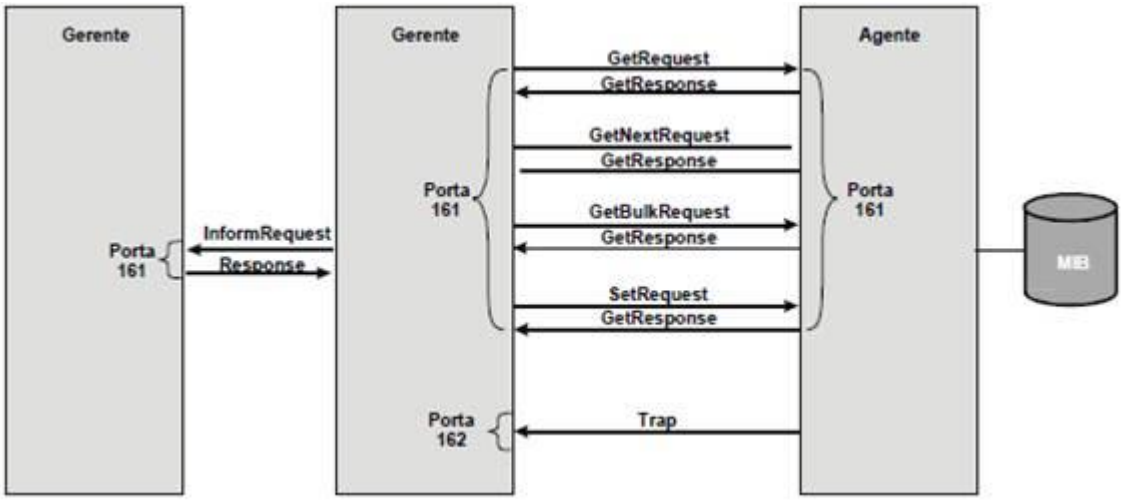

Figura 3 - O fluxo de mensagens com base no modelo gerente/ agente.
Fonte: (Teleco, s.d.).

3.5 DESCOBERTA DE HOSTS UTILIZANDO ICMP

O IP fornece datagramas não confiáveis e não verifica a entrega dos pacotes. Ele foi projetado dessa maneira para fazer uso eficiente dos recursos de rede. O protocolo IP é um serviço de entrega de melhor esforço (*best effort)* que fornece um datagrama de sua origem até o destino final. No entanto ele tem duas deficiências: falta de controle de erros e falta de mecanismos de notificação a erros. O protocolo IP não possui mecanismos de notificação e correção de erros. Caso as mensagens sejam perdidas, descartadas ou o tempo passou do limite e não encontrou o destino final o IP, descarta os datagramas e não avisa que fez esse descarte, pois ele não possui mecanismos de notificação.

O protocolo IP também não tem mecanismos para verificar se um determinado host ou roteador está respondendo e faz-se necessário a verificação dessas informações. O gerente de redes precisa verificar informações do *host* ou do roteador.

O protocolo Internet *Control Message Protocol* (ICMP) foi concebido para compensar essas duas deficiências. É um protocolo auxiliar para protocolo IP (Forouzan, 2007) (Networkservice, 1992).

3.5.1 Tipos de Mensagem

Mensagens ICMP dividem-se em duas: mensagens notificação de erros e consultas.

Notificações de erros informam sobre erros ocorridos em host ou roteador após processar o datagrama IP.

Consultas ocorrem em duplas e elas ajudam o gerente de redes a obter informações sobre os equipamentos. Por exemplo, os nós podem descobrir os vizinhos e os *hosts* podem descobrir informações dos roteadores na rede para ajudar o redirecionamento das mensagens (Forouzan, 2007) (Networkservice, 1981).

3.5.2 Formato da Mensagem

Mensagem ICMP (figura 4) composta no total de 8 *bytes*. O formato muda conforme o tipo de mensagem mas os primeiros 4 *bytes* são constantes:

- *Type*: tipo de mensagem;

- *Code*: Motivo da mensagem;
- *Checksum*: Verifica a integridade do pacote (Forouzan, 2007).

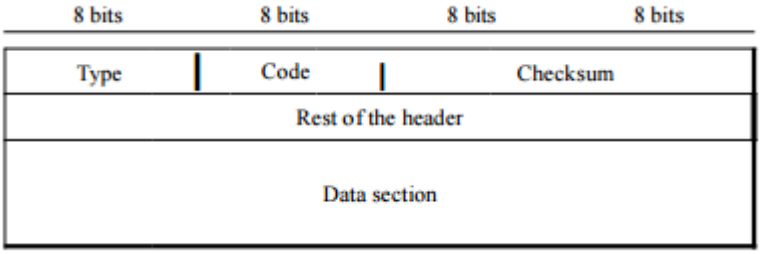

Figura 4 - Mensagem ICMP
Fonte: (Forouzan, 2007).

3.5.3 Notificação de Erros

ICMP é um mecanismo de notificação de erros. Sempre que ocorre um erro no encaminhamento dos pacotes, o ICMP avisará o originador sobre o erro acontecido (figura 5). Uma vez que o IP não tem essa responsabilidade de notificar os erros, é de responsabilidade do ICMP fazer essa notificação, mas ele somente notifica e não corrige os erros.

Existem cinco tipos de mensagens de erros:

- *Destination Unreachable* (destino inalcançável);
- *Source Quench* (contenção de fonte);
- *Time Exceded* (tempo esgotado);
- *Parameter Problems* (problemas de parâmetro);
- *Redirection* (redirecionamento).

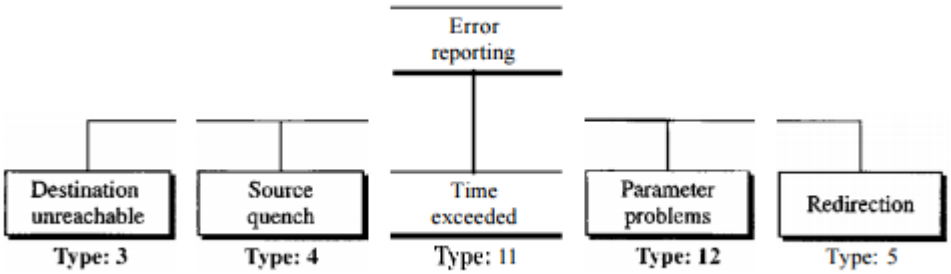

Figura 5 - Tipos de Mensagens
Fonte: (Forouzan, 2007).

Mensagens de erros são compostas por: Cabeçalho IP do pacote original mais 8 bytes de dados, o cabeçalho IP é adicionado para informar o originador da mensagem que estará recebendo a notificação dos erros, os outros 8 bytes de dados são para uso do TCP e UDP para

fornecer o números das portas (figura 6). Esses dados são usados para informar os protocolos superiores TCP e UDP sobre os erros (Forouzan, 2007) (Networkservice, 1981).

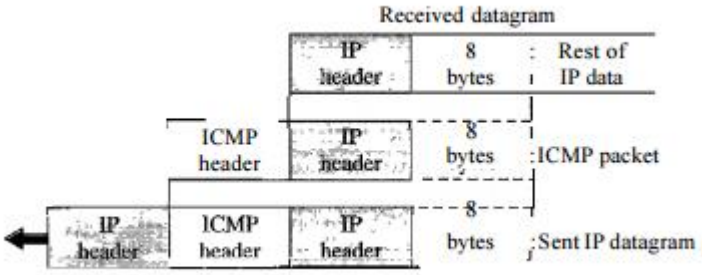

Figura 6 - Cabeçalho
Fonte: (Forouzan, 2007).

3.5.4 Destino Inalcançável

Esta mensagem é usada quando o roteador não pode localizar o destino ou o pacote é descartado, então é enviada uma mensagem de erro "destino inalcançável" para o originador. Percebemos que o envio da mensagem pode ser realizado pelo *host* ou roteador (Morioto, 2008).

3.5.5 Contenção de Fluxo

O protocolo IP não faz controle de fluxo, não tem controle do *host* que envia os pacotes e roteadores que encaminham. Devido à falta desse controle, podem ocorrer problemas com o protocolo IP, podendo acarretar congestionamento no *host* de destino ou no roteador intermediário. O Roteador ou *host* tem um *buffer* máximo para os pacotes a serem encaminhados (roteador) ou processados (*host*), se os pacotes forem recebidos mais rápidos do que podem processar ou encaminhar, são descartados se o *buffer* estiver cheio.

No ICMP foi feito o controle de fluxo para ajudar o protocolo IP, uma vez que o mesmo não faz esse controle. Quando existe o descarte do datagrama por congestionamento junto host ou roteador, ele envia uma mensagem para o originador informando que a mensagem foi descartada e que existe congestionamento em algum ponto, necessitando desacelerar o envio de pacotes para evitar mais perdas (Morioto, 2008) (Networkservice, 1992).

3.5.6 Tempo esgotado

Esta mensagem é enviada quando um datagrama IP com o Campo TTL (*time to live*) igual a zero. O TTL controla essa situação dos saltos entre os roteadores quando um datagrama visita um roteador o campo TTL é decrementado de um, quando esse campo fica igual a zero o roteador descarta o pacote. Este evento é um sintoma que os pacotes estão em *looping* devido a erros na tabela de roteamento, que há enorme congestionamento, ou que os valores TTL estão muito baixo (Morioto, 2008) (Networkservice, 1992).

3.5.7 Problemas de parâmetro

Isto indica que alguns parâmetros do campo de cabeçalho estão corrompidos. Este pode ser visto utilizando a verificação do cabeçalho, pode ocorrer devido a um erro no envio para host ou roteador, então o *host* ou roteador descartará o pacote e informará o originador sobre o problema (Forouzan, 2007).

3.5.8 Redirecionamento

Quando um datagrama precisa ser enviado para outra rede, é necessário saber o IP do próximo roteador ou *host*. Tanto o *host* ou roteador precisam ter a tabela de roteamento para fazer a entrega. No caso dos roteadores, os mesmos possuem tabelas dinâmicas para saber o endereço do próximo salto. Todavia os endereços IP do *host* não fazem parte dessa tabela, isso ocorre para ter eficiência no direcionamento.

O *host*, geralmente, conhece somente um endereço IP, o do *gateway default*, que faz o redirecionamento para redes externas. Porém, pode ocorrer um erro de envio (ilustrado na figura 7): o *host* A deseja se comunicar com o *host* B e envia um datagrama para um roteador R1, o qual consulta sua tabela de roteamento e encaminha o datagrama para o roteador R2, que é o roteador de borda, e ao mesmo tempo envia uma mensagem para o *host* A informando que para destinatários externos o roteador R2 é a melhor escolha (figura 7) (Forouzan, 2007).

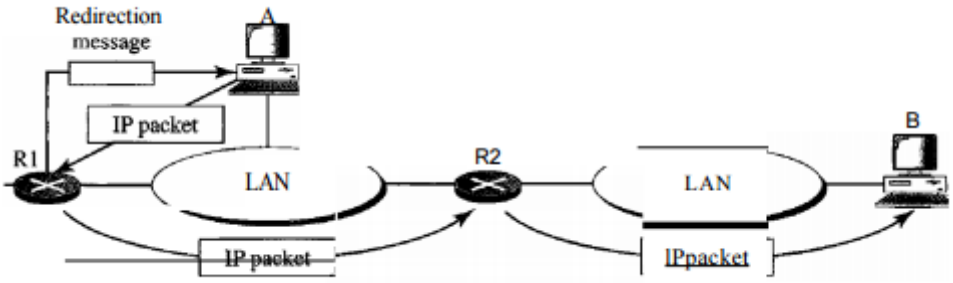

Figura 7 - Redirecionamento
Fonte: (Forouzan, 2007).

3.5.9 Descoberta de Host

Para determinar a rota de *host* ou roteador podemos realizar o rastreamento dos pacotes, para isso usamos duas ferramentas *PING* e *Traceroute.*

Usamos a ferramenta *PING* para determinar se o *host* está ativo e respondendo (figura 8). O *host* de origem envia uma notificação ICMP *eco-request*, caso o destinatário esteja ativo, ele responde com uma notificação ICMP *echo-reply*. O *PING* ajusta o campo "*identifier-field*" nas notificações de *echo-request* e *echo-reply*, este campo inicia-se em zero e incrementa em uma unidade toda vez que uma nova mensagem é enviada, então o *PING* calcula o tempo de resposta marcando o horário de saída e diminuído do horário de chegada para saber o tempo total.

```
$ ping thda.edu
PING thda.edu (153.18.8.1)   56 (84)  bytes of data.
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=O    ttl=62    time=1.91 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=l    ttl=62    time=2.04 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=2    ttl=62    time=1.90 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=3    ttl=62    time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=4    ttl=62    time=1.93 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=5    ttl=62    time=2.00 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=6    tt1=62    time=1.94 ms
64 bytes from tiptoe.:fhda.edu (153.18.8.1): icmp_seq=7   ttl=62    time=1.94 ms
64 bytes from tiptoe.:fhda.edu (153.18.8.1): icmp_seq=8   ttl=62    time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=9    ttl=62    time=1.89 ms
64 bytes from tiptoe.:fhda.edu (153.18.8.1): icmp_seq=lO   ttl=62    time=1.98 ms
```

Figura 8 - PING
Fonte: (Forouzan, 2007).

O *PING* fica enviando mensagens intermitentemente até realizar a ação de interrupção. Após isso, imprime os dados na tela informando o número de pacotes enviados e recebidos, e o tempo total. Com isso determinamos se um host está ativo e respondendo (Forouzan, 2007).

3.5.10 Traceroute

Conforme vimos previamente, o PING é usado para indicar a conectividade entre dois *hosts.* O *Traceroute* (tracert) é um utilitário que nos permite observar o caminho entre esses *hosts*. O *trace* gera uma lista dos saltos que foram bem-sucedidos ao longo do caminho (CISCO CCNA, 2015). Essa ferramenta é usada para determinar rotas de um pacote até seu destino, unido com ICMP podemos mapear uma rota. O *Traceroute* usa duas mensagens do ICMP para determinar a rota percorrida: *time-exceeded* e *destination unreachable* (figura 9).

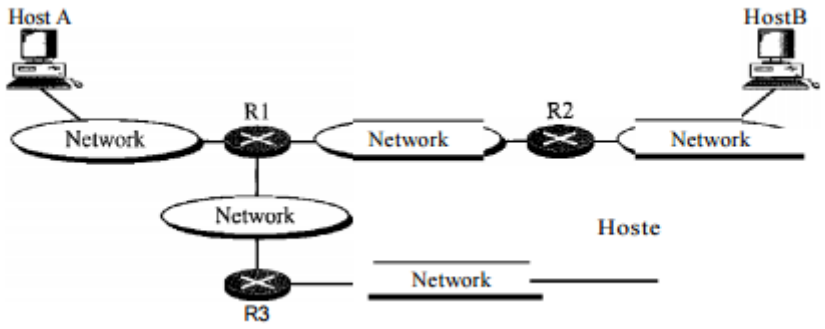

Figura 9 - Traceroute
Fonte: (Forouzan, 2007).

A topologia acima determina que um pacote parte do *host* A para *host* B passando pelo roteador R1 e R2, mas geralmente não sabemos a configuração dessa topologia. O *Traceroute* usa mensagens ICMP e o campo *time-to-live* do IP para determinar a rota.

Etapas para descobrir a rota para R1 e tempo de resposta de A pra R1:

1. Envia pacote de A para B usando UDP, a mensagem é encapsulada pelo IP com o TTL igual a 1 *traceroute* salva o horário de envio;
2. R1 recebe a mensagem e decrementa TTL que fica com o valor zero. R1 envia uma mensagem ICMP de *time-exceeded* para A avisando que TTL está igual a zero e descarta o pacote;
3. O *traceroute* que está em A recebe a mensagem ICMP e usa o endereço IP de destino encapsula ICMP para descobrir o IP de R1 e também diminui o horário

do anterior para determinar o tempo de resposta. Repete as etapas 1,2 e 3 para ter uma média do tempo de resposta;

4. O *traceroute* repete a sequência anterior para determinar o endereço R2, mas dessa vez o TTL é 2, com isso somente o R2 que vai descartar a mensagem e avisa sobre a mensagem ICMP *time-exceeded*;
5. Repete a etapa 4 para descobrir o endereço de B e tempo de resposta, mas quando *host* B recebe a mensagem ele decrementa TTL e não descarta o pacote pois chegou ao seu destino. Nesse momento usa-se uma porta fora do padrão UDP então ele tenta achar uma aplicação com aquela porta ela não existindo descarta o pacote e envia uma mensagem ICMP *destination ureachable* para A. Isso não ocorre nos roteadores pois eles não verificam cabeçalho UDP. Então, *traceroute* registra o IP de destino e o tempo ao a mensagem de *destination ureachable* o *host* A reconhece que a rota foi mapeada, não sendo necessário enviar outros pacotes (Forouzan, 2007).

3.6 DESCOBERTA DE SERVIÇOS

Existem várias formas para fazer a descoberta de serviços em redes de computadores, como por exemplo: DEAP*space*, UPnP, P2PDP, WBEM e escaneamento de portas. Será descrito somente o escaneamento de porta e WBEM. O reconhecimento é considerado a primeira fase para uma tentativa sistemática de localizar, recolher, identificar e registar informações, procurando descobrir o máximo de dados possíveis sobre o *host*.

Esta primeira etapa é considerada uma busca de informação passiva, isto envolve atividades como: a coleta de informações, a determinação do alcance da rede, identificando máquinas ativas, encontrar portas abertas e pontos de acesso, detecção de sistema operacional, serviços de impressão digital e mapear a rede.

O escaneamento de portas tem dois tipos principais:

- Escaneamento por força bruta: realizam *scans* de uma forma agressiva pela leitura de uma porta, dado um intervalo especificado. Ele estabelece uma conexão completa com o *host* para inspecionar se as portas estão abertas. Em virtude do estabelecimento da conexão, é possível detectar a presença das portas abertas. Assim, quando um grande número de pacotes SYN chegarem para solicitar uma conexão a partir de um único

endereço IP em várias portas da máquina de destino, isso indica que um *scanner* de força bruta está analisando as portas abertas (Gadge & Patil, s.d.);

- Escaneamento semiaberto: não estabelece uma conexão completa com o *host*. Eles enviam um único pacote com um sinalizador específico definido para *host,* com base na resposta que pode ser compreendido se as portas estão abertas ou não.
- Existem vários tipos de verificações; padrões para cada varredura podem ser identificados como:

3.6.1 SYN Scans

Este *scan* envia um grande número de pacotes com SYN *flag* para o *host*, desta forma não completa *handshake* triplo após o recebimento do SYN/ACK, o que indica que a porta está aberta, então ele fecha a conexão.

3.6.2 TCP Connect Scan

Este *scan* faz um grande número de conexões que são estabelecidas com o *host* em portas diferentes. O estabelecimento de uma conexão indica que a porta correspondente está aberta, uma vez que a conexão foi estabelecida, a porta é aberta e feita sua identificação, então a conexão é fechada. Assim, se a partir de um determinado *host* um grande número de conexões são estabelecidas em várias portas em um espaço muito curto de tempo, pode-se inferir que, uma varredura TCP *Connect* é feita para determinar várias portas de um *host*.

3.6.3 FIN Scan

Este *scan* envia um grande número de pacotes com apenas o *flag* FIN para o *host* de destino. Se o host responde com um RST, indica que a porta está fechada, e as portas abertas simplesmente ignoram pacotes. Essa verificação pode ser facilmente identificada se há um grande número de pacotes com a sinalização FIN as portas abertas de um *host*.

3.6.4 NULL Scan

Este *scan* envia um grande número de pacotes, sem sinalização definida para um *host* destino, as portas abertas ignoram esses pacotes. Considerando que as portas fechadas respondem de volta com um RST, essa verificação pode ser facilmente identificada se houver um grande número de pacotes sem nenhuma sinalização, podendo determinar as portas que estão abertas.

3.6.5 UDP Scan

Este *scan* envia um grande número de pacotes UDP para *host* destino. Este *scan* não completa *handshake* triplo, assim pode-se determinar se a porta UDP está aberta.

3.6.6 ICMP Scan

Este *scan* inclui o envio de ICMP *echo-request* para os endereços IP específicos para verificar se estão ativos.

Com essas técnicas de escaneamento de portas pode-se determinar quais portas estão abertas, tendo uma ideia dos tipos de serviços que estão contidos em cada *host* da rede para fazer o mapeamento.

4 FERRAMENTAS

4.1 NMAP

NMAP é uma ferramenta livre e *open source* para a descoberta de rede e auditoria de segurança. Muitos sistemas e administradores de rede também acham que é útil para tarefas como inventário de rede, horários de atualização de serviços de gerenciamento e monitoramento de *host* ou *uptime* serviço.

O NMAP utiliza pacotes IP para determinar os serviços que estão disponíveis na rede, que serviços, quais sistemas operacionais (e versões de SO) eles estão executando, que tipo de filtros de pacotes / *firewalls* estão em uso. Ele foi projetado para digitalizar rapidamente grandes redes, mas funciona bem para *host* único.

O NMAP é executado em todos os principais sistemas operacionais de computador, e os pacotes binários oficiais estão disponíveis para os sistemas operacionais Linux, Windows e Mac OS X. Além do clássico de linha de comando executável NMAP, a suíte NMAP inclui um visualizador de interface avançada (NMAP, 2015).

4.2 CACTI

CACTI é uma interface completa para *RRDTool*, que armazena todas as informações necessárias para criar gráficos e preenchê-los com dados em um banco de dados. O *frontend* é completamente PHP. Além de ser capaz de manter gráficos, fontes de dados e *Round Robin* Arquivos em um banco de dados, CACTI lida com a coleta de dados. Há também suporte SNMP a ser utilizado para criar gráficos de tráfego com MRTG.

Fontes de dados:

Para lidar com a coleta de dados, um usuário pode alimentar no CACTI os caminhos para qualquer *script*, juntamente com todos os dados que o usuário terá de "preencher", o CACTI reúne esses dados em um trabalho CRON e preenche um banco de dados *MySQL*.

Fontes de dados também podem ser criadas para que correspondam aos dados reais no gráfico. Por exemplo se um usuário quiser representar graficamente os tempos de *ping* a um *host*, pode-se criar uma fonte de dados utilizando um *script* que pode pingar um *host* e retorna

seu valor em milissegundos. Opções para *RRDTool* como a forma de armazenar os dados que um usuário será capaz de definir qualquer informação adicional de que a fonte de entrada de dados requer, como um *root* para executar *ping*. Uma vez que uma fonte de dados é criada, ela é automaticamente mantida em intervalos de 5 minutos.

Gráficos:

Uma vez que, uma ou mais fontes de dados são definidos, um gráfico *RRDTool* pode ser criado utilizando os dados. O CACTI permite criar vários tipos de gráficos com o *RRDTool*, existindo gráficos padrões para auxiliar o gerente de redes. Uma área de seleção de cores e função de preenchimento automático de texto também ajuda na criação de gráficos para tornar o processo mais fácil. O usuário pode criar gráficos com base em *RRDTool* no CACTI, mas há várias maneiras de mostrar. Junto com uma "visão de lista" padrão e um "modo de visualização", que lembra o *frontend RRDTool 14all*, há uma "exibição de árvore", que permite que o usuário coloque gráficos em uma árvore hierárquica para fins de organização.

Gerenciamento de usuários:

Devido às muitas funções do CACTI, uma ferramenta de gerenciamento baseada no usuário, pode-se adicionar usuários e dar-lhes direitos a determinadas áreas da interface. Isso permitiria criar alguns usuários que podem alterar os parâmetros do gráfico, enquanto outros só podem ver os gráficos. Cada usuário também mantém as suas próprias definições quando se trata de gráficos de visualização (Cacti, 2015).

4.3 ZABBIX

O *Zabbix* é um *software open source* que possui muitas maneiras de monitorar aspectos de uma infraestrutura de TI. Apresenta uma característica de monitoramento distribuída com gerenciamento centralizado enquanto, muitas dessas aplicações têm banco de dados central com o *Zabbix* é possível distribui-lo fazendo níveis de monitoramento. Suas características são:

- Interface *Web* (figura 10);
- Suporte aos sistemas Linux, Solaris, HP-UX, AIX, FreeBSD, OpenBSD, NetBSD, Mac OS X, Windows;
- Agentes nativos para maioria dos Sistemas Operacionais;

- Capacidade de monitorar diretamente SNMP (V1,v2,v3) e dispositivos IPMI;
- Integração com banco de dados MySQL, POSTGRES e Oracle;
- Geração de gráficos;
- Envio de alertas por e-mail e SMS;
- Permite customização (Zabbix, 2015).

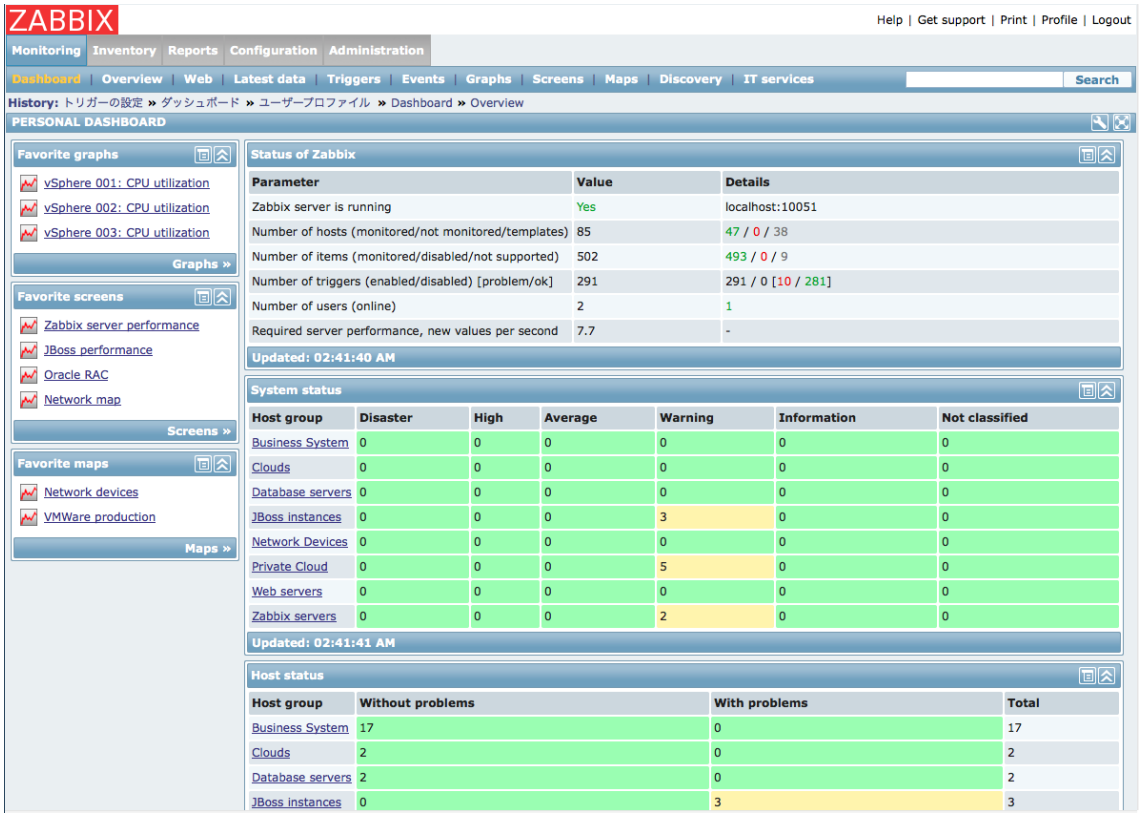

Figura 10 - Interface Web da ferrameta Zabbix.
Fonte: (Zabbix, 2015).

5 DESENVOLVIMENTO

Neste capítulo serão apresentados os comandos utilizados na Rede Local de Ensino, bem como suas saídas, que serão analisadas e utilizadas como base para o mapeamento da rede e serviços.

5.1 MAPEAMENTO DE SERVIÇOS

A Rede Local de Ensino está situada dentro da sub-rede da UTFPR 200.134.10.0/24, e para realizar o mapeamento dos serviços junto ao RLE usou-se a ferramenta NMAP. A seguir serão apresentados os dados coletados com *software*.

Primeiramente, aplicou-se o NMAP na rede 200.134.10.0 com o objetivo de realizar uma varredura (tabela 1).

Starting Nmap 5.00 (http://nmap.org) at 2015-06-03 16:04 BRT
Host marumbi.dainf.ct.utfpr.edu.br (200.134.10.1) is up (0.0010s latency).
MAC Address: 00:0C:29:93:D8:1E (VMware)
Host 200.134.10.2 is up (0.00025s latency).
MAC Address: E8:39:35:91:02:F3 (Unknown)
Host arcaz.dainf.ct.utfpr.edu.br (200.134.10.3) is up (0.00098s latency).
MAC Address: 00:0C:29:B6:83:88 (VMware)
Host thompson.dainf.ct.utfpr.edu.br (200.134.10.5) is up (0.00098s latency).
MAC Address: 00:0C:29:9D:6D:1C (VMware)
Host ns.dainf.ct.utfpr.edu.br (200.134.10.6) is up (0.00097s latency).
MAC Address: 00:0C:29:25:B2:84 (VMware)
Host gohan.dainf.ct.utfpr.edu.br (200.134.10.7) is up (0.00060s latency).
MAC Address: 00:0C:29:57:A2:FD (VMware)
Host 200.134.10.14 is up (0.0015s latency).
MAC Address: 1C:1D:86:16:58:41 (Unknown)
Host 200.134.10.21 is up (0.00012s latency).
MAC Address: 84:2B:2B:7F:55:C5 (Unknown)
Host 200.134.10.24 is up (0.0016s latency).
MAC Address: 68:7F:74:17:37:53 (Unknown)
Host 200.134.10.27 is up (0.00059s latency).
MAC Address: 00:0C:29:BF:64:05 (VMware)
Host seminfo.dainf.ct.utfpr.edu.br (200.134.10.28) is up (0.00012s latency).
MAC Address: E0:69:95:C9:58:6C (Unknown)
Host 200.134.10.31 is up.
Host 200.134.10.32 is up (0.00055s latency).
MAC Address: 00:0C:29:B2:95:5E (VMware)

Host emilias.dainf.ct.utfpr.edu.br (200.134.10.33) is up (0.00066s latency).
MAC Address: 00:0C:29:C6:E0:07 (VMware)
Host 200.134.10.34 is up (0.00080s latency).
MAC Address: 00:0C:29:D2:74:7E (VMware)
Host plone.dainf.ct.utfpr.edu.br (200.134.10.37) is up (0.00095s latency).
MAC Address: 00:0C:29:61:CC:C5 (VMware)
Host 200.134.10.56 is up (0.00097s latency).
MAC Address: 84:2B:2B:7A:C2:07 (Unknown)
Host 200.134.10.61 is up (0.0011s latency).
MAC Address: 00:0C:29:46:36:93 (VMware)
Host 200.134.10.63 is up (0.00053s latency).
MAC Address: 00:0C:29:CA:44:E0 (VMware)
Host 200.134.10.86 is up (0.0010s latency).
MAC Address: 00:0C:29:94:C8:4F (VMware)
Host enya.dainf.ct.utfpr.edu.br (200.134.10.100) is up (0.00056s latency).
MAC Address: E0:DB:55:15:D0:8C (Unknown)
Host dhcp101.dainf.ct.utfpr.edu.br (200.134.10.101) is up (0.00085s latency).
MAC Address: 52:54:00:C1:CC:64 (QEMU Virtual NIC)
Host dhcp102.dainf.ct.utfpr.edu.br (200.134.10.102) is up (0.00077s latency).
MAC Address: 52:54:00:72:12:5C (QEMU Virtual NIC)
Host dhcp105.dainf.ct.utfpr.edu.br (200.134.10.105) is up (0.00090s latency).
MAC Address: 52:54:00:F3:AF:99 (QEMU Virtual NIC)
Host dhcp111.dainf.ct.utfpr.edu.br (200.134.10.111) is up (0.00038s latency).
MAC Address: 00:24:1D:F9:37:6F (Giga-byte Technology Co.)
Host dhcp113.dainf.ct.utfpr.edu.br (200.134.10.113) is up (0.015s latency).
MAC Address: 00:24:73:91:53:D7 (3Com Europe)
Host dhcp121.dainf.ct.utfpr.edu.br (200.134.10.121) is up (0.00015s latency).
MAC Address: E0:69:95:C9:58:D2 (Unknown)
Host dhcp122.dainf.ct.utfpr.edu.br (200.134.10.122) is up (0.0032s latency).
MAC Address: 10:78:D2:DA:12:48 (Unknown)
Host dhcp124.dainf.ct.utfpr.edu.br (200.134.10.124) is up (0.00028s latency).
MAC Address: 20:1A:06:57:27:1B (Unknown)
Host dhcp134.dainf.ct.utfpr.edu.br (200.134.10.134) is up (0.0061s latency).
MAC Address: 10:78:D2:DA:12:40 (Unknown)
Host dhcp138.dainf.ct.utfpr.edu.br (200.134.10.138) is up (0.00098s latency).
MAC Address: E4:11:5B:0B:61:9B (Unknown)
Host 200.134.10.152 is up (0.00065s latency).
MAC Address: 0C:4D:E9:9D:E5:6F (Unknown)
Host 200.134.10.156 is up (0.0013s latency).
MAC Address: A8:20:66:0E:1D:00 (Unknown)
Host 200.134.10.164 is up (0.00036s latency).
MAC Address: 68:5B:35:BF:2B:F7 (Unknown)
Host 200.134.10.169 is up (0.00012s latency).
MAC Address: 84:2B:2B:7B:B6:1E (Unknown)
Host 200.134.10.176 is up (0.051s latency).
MAC Address: 00:24:73:92:34:43 (3Com Europe)

Host 200.134.10.179 is up (0.000091s latency).
MAC Address: 08:60:6E:73:CD:34 (Unknown)
Host 200.134.10.185 is up (0.00021s latency).
MAC Address: DC:0E:A1:01:18:69 (Unknown)
Host 200.134.10.191 is up (0.00087s latency).
MAC Address: 00:0C:29:E5:1D:2D (VMware)
Host 200.134.10.192 is up (0.00020s latency).
MAC Address: 84:2B:2B:7B:26:D0 (Unknown)
Host 200.134.10.196 is up (0.00019s latency).
MAC Address: 10:78:D2:DA:22:8C (Unknown)
Host 200.134.10.197 is up (0.00024s latency).
MAC Address: A8:20:66:0E:22:B6 (Unknown)
Host 200.134.10.251 is up (0.032s latency).
MAC Address: 00:0B:AC:E0:D3:00 (3Com)
Host metcalfe.dainf.ct.utfpr.edu.br (200.134.10.254) is up (0.000059s latency).
MAC Address: 00:06:4F:63:AB:8D (Pro-nets Technology)
Nmap done: 256 IP addresses (44 hosts up) scanned in 1.49 seconds

Tabela 1 - Varredura IP
Fonte: Autoria Própria.

Após o resultado da varredura total da rede (tabela 1) foi realizada uma reunião com o administrador da RLE para definir quais os *hosts* seriam de seu domínio. Como resultado desta reunião obteve-se a seguinte lista:

- *Host* 200.134.10.1;
- *Host* 200.134.10.5;
- *Host* 200.134.10.6;
- *Host* 200.134.10.7;
- *Host* 200.134.10.27;
- *Host* 200.134.10.32;
- *Host* 200.134.10.37.

Em seguida, realizou-se a aplicação do uso do NMAP e cada em cada um dos itens acima listados.

No *Host* 200.134.10.1 foi identificada a porta 53 (*Domain Name System*) aberta. Esta porta é a padrão para o protocolo DNS. Com isso conclui-se que esse *Host* comporta-se como um possível Servidor de DNS, conforme demonstrado no destaque da figura abaixo.

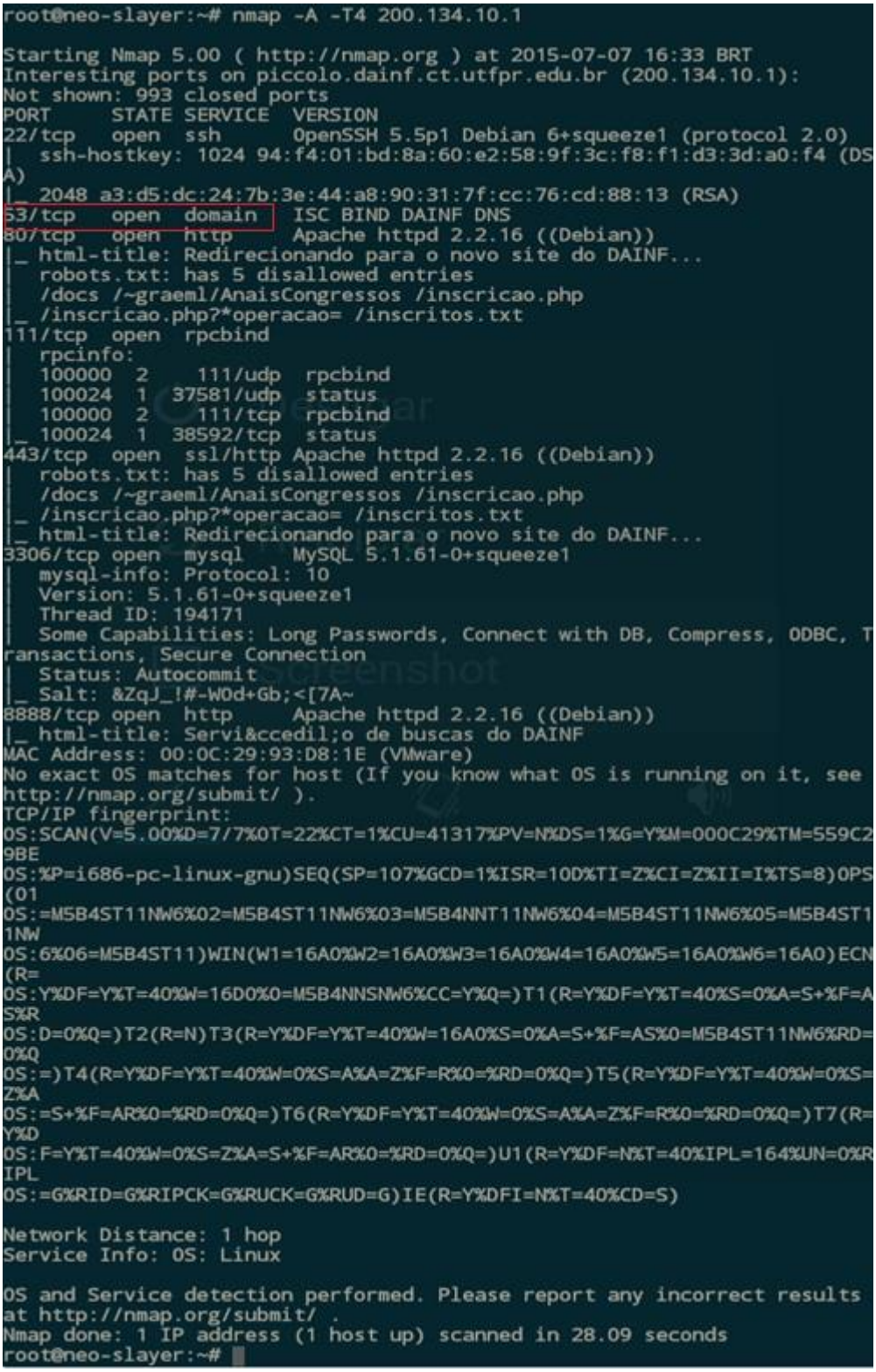

Figura 11- Nmap para host 200.134.10.1
Fonte: Autoria Própria.

No *Host* 200.134.10.5 foram identificadas as seguintes portas abertas: 25 (*Simple Mail Transfer Protocol*); 110 (*Post Office Protocol version 3*); 143 (*Internet Message Access*

Protocol). Portas essas que são, comumente, usadas para servidor de E-mail, conforme o destaque da figura a seguir.

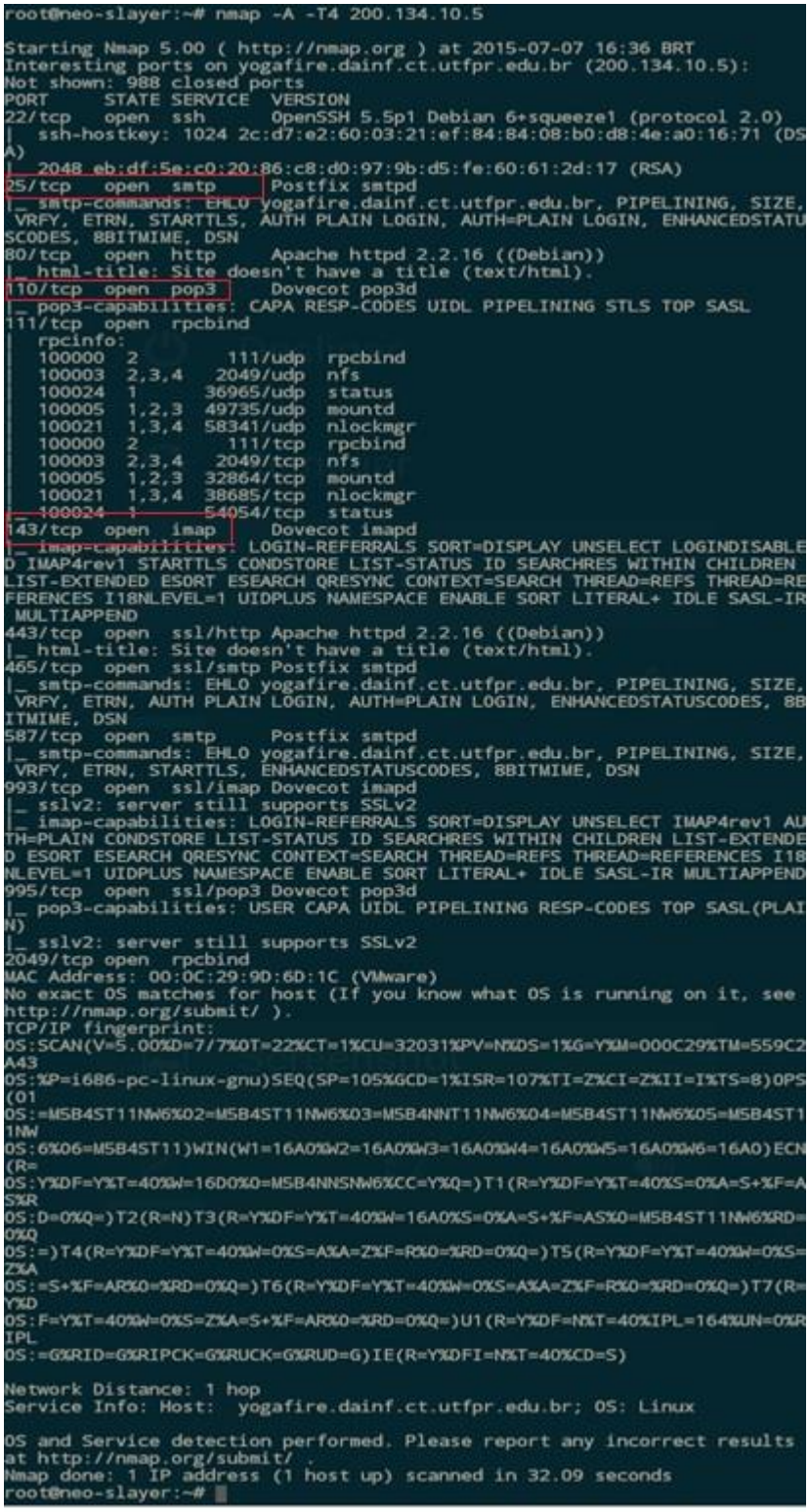

Figura 12 - Nmap para host 200.134.10.5
Fonte: Autoria Própria.

No *Host* 200.134.10.6 foi identificada a porta 53 (*Domain Name System*) aberta. Esta porta é a padrão para o protocolo DNS. Com isso, conclui-se que esse *Host* comporta-se como um possível Servidor de DNS, conforme demonstrado no destaque da figura abaixo.

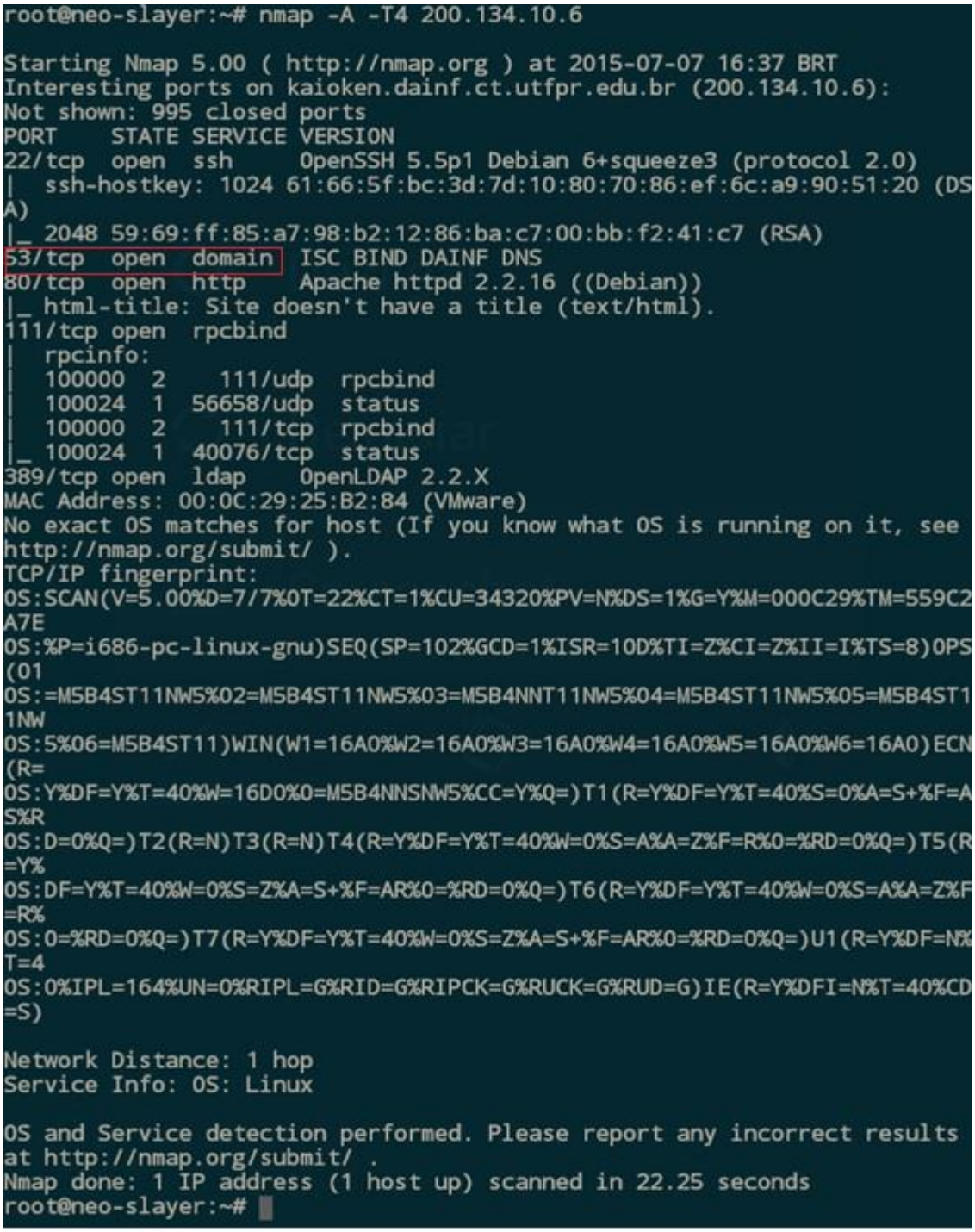

Figura 13 - Nmap para host 200.134.10.6
Fonte: Autoria Própria.

No *Host* 200.134.10.7 foi identificada a porta 53 (*Domain Name System*) aberta. Esta porta é a padrão para o protocolo DNS. Com isso, conclui-se que esse *Host* comporta-se como um possível Servidor de DNS, conforme demonstrado no destaque da figura abaixo.

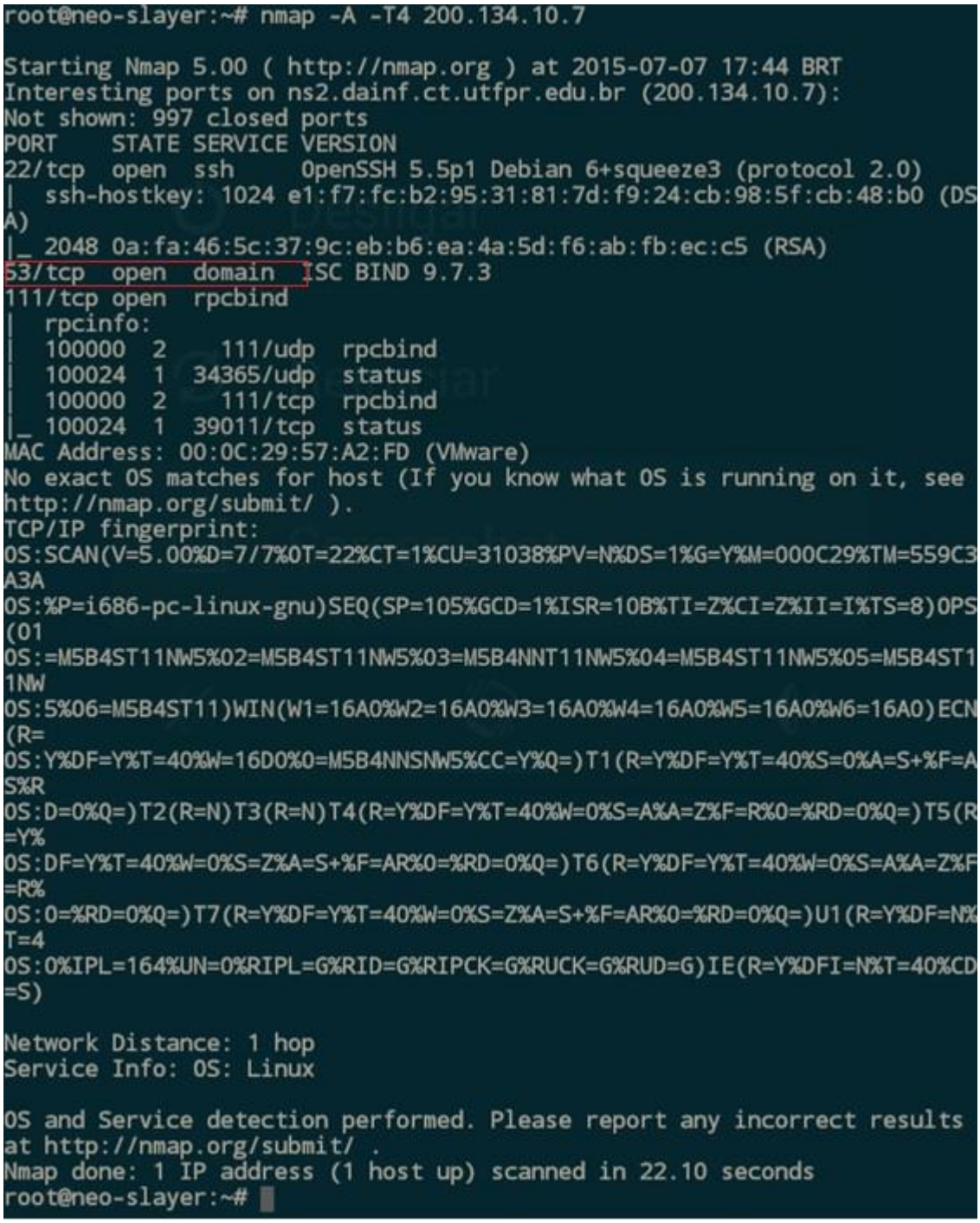

Figura 14 - Nmap para host 200.134.10.7
Fonte: Autoria Própria.

Para o *host* 200.134.10.27 não foram identificas portas que poderiam representar qual serviço que estava presente.

```
root@neo-slayer:~# nmap -A -T4 200.134.10.27

Starting Nmap 5.00 ( http://nmap.org ) at 2015-07-07 16:38 BRT
Interesting ports on 200.134.10.27:
Not shown: 997 closed ports
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 6.0p1 Debian 4+deb7u2 (protocol 2.0)
|  ssh-hostkey: 1024 09:08:3e:2c:64:9d:62:ff:fa:5a:df:bc:33:6f:06:4a (DS
A)
|_ 2048 94:82:a3:1f:a1:fd:c7:db:e8:6d:22:81:a6:63:d1:a0 (RSA)
80/tcp  open  http    Apache httpd 2.2.22 ((Debian))
|_ html-title: Redirecionar
111/tcp open  rpcbind
|  rpcinfo:
|  100000  2,3,4    111/udp  rpcbind
|  100024  1      38094/udp  status
|  100000  2,3,4    111/tcp  rpcbind
|_ 100024  1      60814/tcp  status
MAC Address: 00:0C:29:BF:64:05 (VMware)
No exact OS matches for host (If you know what OS is running on it, see
http://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=5.00%D=7/7%OT=22%CT=1%CU=30013%PV=N%DS=1%G=Y%M=000C29%TM=559C2
AB3
OS:%P=i686-pc-linux-gnu)SEQ(SP=FE%GCD=1%ISR=109%TI=Z%CI=I%II=I%TS=8)OPS(
O1=
OS:M5B4ST11NW6%O2=M5B4ST11NW6%O3=M5B4NNT11NW6%O4=M5B4ST11NW6%O5=M5B4ST11
NW6
OS:%O6=M5B4ST11)WIN(W1=3890%W2=3890%W3=3890%W4=3890%W5=3890%W6=3890)ECN(
R=Y
OS:%DF=Y%T=40%W=3908%O=M5B4NNSNW6%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS
%RD
OS:=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=
Y%D
OS:F=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=
R%O
OS:=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T
=40
OS:%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=
S)

Network Distance: 1 hop
Service Info: OS: Linux

OS and Service detection performed. Please report any incorrect results
at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.70 seconds
root@neo-slayer:~#
```

Figura 15 - Nmap para host 200.134.10.27
Fonte: Autoria Própria.

No *Host* 200.134.10.32, foram identificadas as portas 3306 e 5432 como abertas, que é o padrão para servidores de banco de dados *mysql* e *postges*, respectivamente. Com isso, conclui-se que o *Host* comporta-se como um servidor de Banco de Dados, conforme o destaque da figura abaixo.

```
root@neo-slayer:~# nmap -A -T4 200.134.10.32

Starting Nmap 5.00 ( http://nmap.org ) at 2015-07-07 16:33 BRT
Interesting ports on 200.134.10.32:
Not shown: 994 closed ports
PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 5.5p1 Debian 6+squeeze5 (protocol 2.0)
|  ssh-hostkey: 1024 20:30:0e:c1:9f:9b:68:ca:f2:06:b9:59:0d:e0:63:01 (DS
A)
|_ 2048 05:f5:74:78:48:24:21:a7:5b:58:c0:b7:df:bd:e2:a8 (RSA)
111/tcp  open  rpcbind
|  rpcinfo:
|  100000  2    111/udp  rpcbind
|  100024  1  55479/udp  status
|  100000  2    111/tcp  rpcbind
|_ 100024  1  38325/tcp  status
1521/tcp open  oracle-tns Oracle TNS Listener 10.2.0.1.0 (for Linux)
3306/tcp open  mysql      MySQL 5.1.73-1-log
|  mysql-info: Protocol: 10
|  Version: 5.1.73-1-log
|  Thread ID: 187047
|  Some Capabilities: Long Passwords, Connect with DB, Compress, ODBC, T
ransactions, Secure Connection
|  Status: Autocommit
|_ Salt: 89W*iVQtthOQ'oy31:pG
5432/tcp open  postgresql PostgreSQL DB (Portugese)
8080/tcp open  http       Oracle XML DB Enterprise Edition httpd
|_ html-title: ORACLE DATABASE 10g EXPRESS EDITION LICENSE AGREEMENT Let
ter-S...
MAC Address: 00:0C:29:B2:95:5E (VMware)
No exact OS matches for host (If you know what OS is running on it, see
http://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=5.00%D=7/7%OT=22%CT=1%CU=41821%PV=N%DS=1%G=Y%M=000C29%TM=559C2
989
OS:%P=i686-pc-linux-gnu)SEQ(SP=102%GCD=1%ISR=109%TI=Z%CI=Z%II=I%TS=8)OPS
(O1
OS:=M5B4ST11NW6%02=M5B4ST11NW6%03=M5B4NNT11NW6%04=M5B4ST11NW6%05=M5B4ST1
1NW
OS:6%06=M5B4ST11)WIN(W1=16A0%W2=16A0%W3=16A0%W4=16A0%W5=16A0%W6=16A0)ECN
(R=
OS:Y%DF=Y%T=40%W=16D0%0=M5B4NNSNW6%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=0%A=S+%F=A
S%R
OS:D=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%0=%RD=0%Q=)T5(R
=Y%
OS:DF=Y%T=40%W=0%S=Z%A=S+%F=AR%0=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F
=R%
OS:0=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%0=%RD=0%Q=)U1(R=Y%DF=N%
T=4
OS:0%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD
=S)

Network Distance: 1 hop
Service Info: OS: Linux

OS and Service detection performed. Please report any incorrect results
at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.30 seconds
root@neo-slayer:~#
```

Figura 16 - Nmap para host 200.134.10.32
Fonte: Autoria Própria.

No *Host* 200.134.10.37, as portas que foram identificadas como abertas foram as 80 e 81. Isso não identifica o serviço que está em execução, porém observa-se o serviço *Plone*, que é um gerenciador de conteúdo (Figura 17).

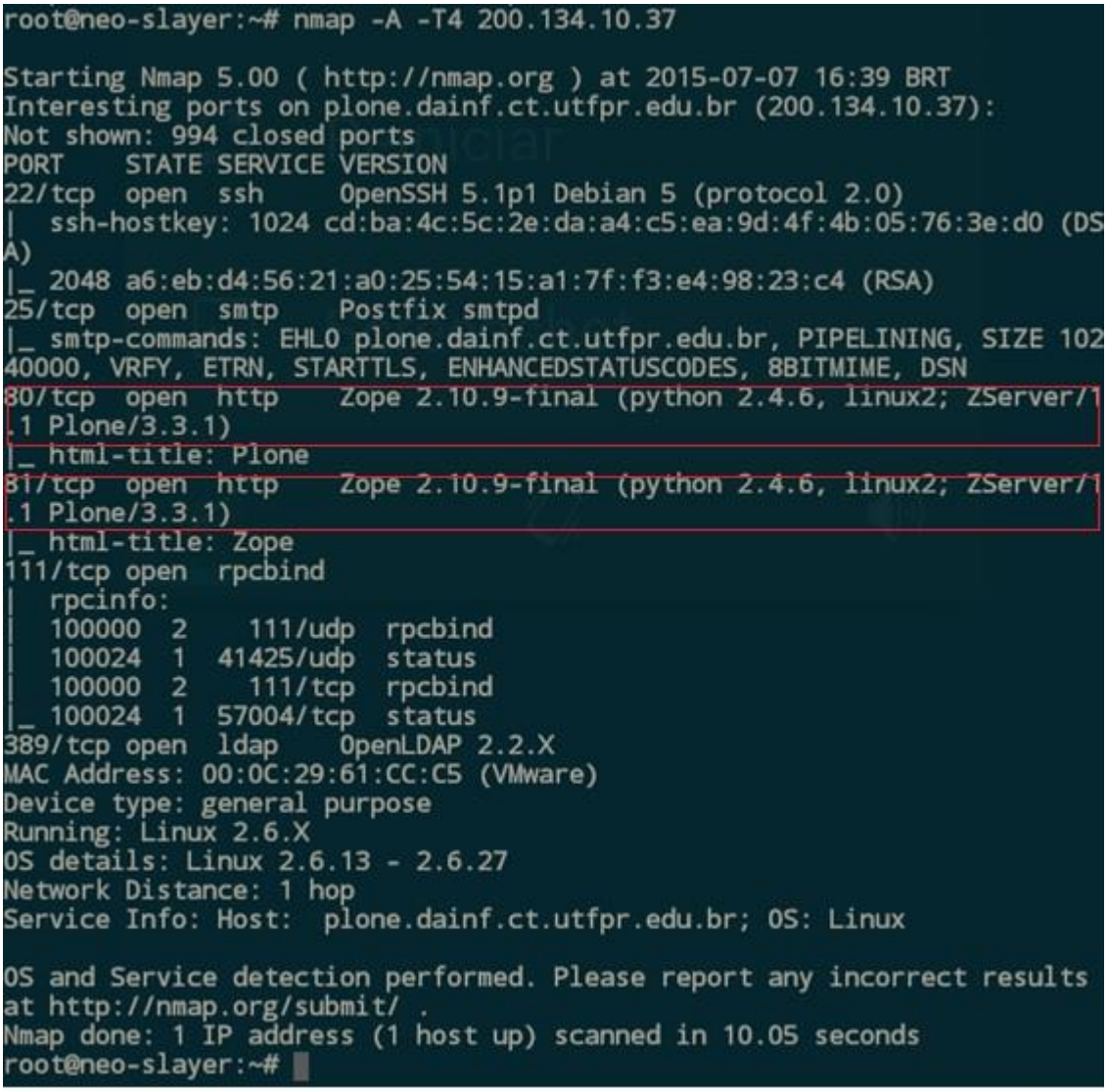

Figura 17 - Nmap para host 200.134.10.37
Fonte: Autoria Própria.

Em uma segunda reunião com o gerente da RLE, discutiu-se qual serviço que estava em execução para cada *Host* identificado acima, chegando na seguinte lista:

- 200.134.10.1, servidor DNS primário;
- 200.134.10.5, servidor de e-mail;
- 200.134.10.6, servidor DNS secundário;
- 200.134.10.7, servidor de DNS reverso;
- 200.134.10.27, servidor *Moodle*;
- 200.134.10.32, servidor de Banco de Dados para os alunos;

- 200.134.10.37, servidor para site DAINF;

5.2 MAPEAMENTO DE HOSTS

Na elaboração do mapeamento de *host* foram utilizadas as ferramentas NMAP e *Traceroute* partindo do servidor *Slayer*, que possui três interfaces de rede (figura 18).

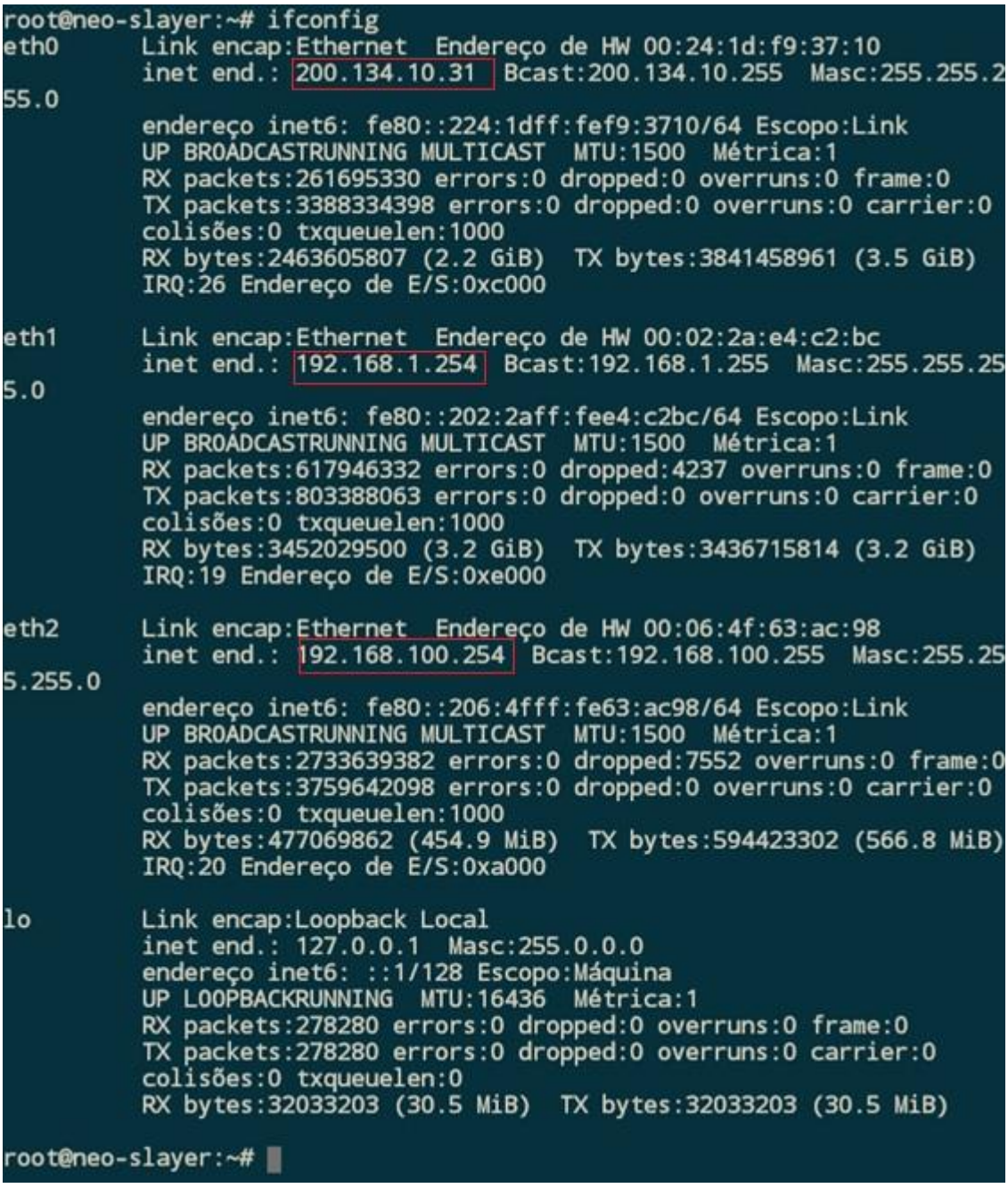

Figura 18 - Interfaces de Rede do Servidor Slayer
Fonte: Autoria Própria.

Para descobrir os saltos de um pacote de dentro da rede da RLE até a rede externa, utilizando o *Traceroute* até o Google (Figura 19). O primeiro salto é para o servidor *Slayer*, conforme IP 192.168.1.254 identificado na (Figura 18), comportando-se como um roteador. Com isso, descobriu-se a figura do servidor *Klingon* com IP 200.134.10.254, sob o domínio do RLE possui duas interfaces de rede (Figura 19).

```
    traceroute to www.google.com (216.58.222.36), 30 hops max, 60 byte packets
 1  192.168.1.254 (192.168.1.254)  0.126 ms  0.119 ms  0.111 ms
 2  metcalfe.dainf.ct.utfpr.edu.br (200.134.10.254)  2.379 ms  2.367 ms  2.816 ms
 3  * * *
 4  master.cefetpr.br (200.17.97.33)  2.802 ms  2.795 ms  2.780 ms
 5  ge-1-3-r2.pop-pr.rnp.br (200.19.74.109)  2.755 ms  2.748 ms  2.750 ms
 6  200.238.139.9 (200.238.139.9)  10.003 ms  6.908 ms  6.904 ms
 7  200.143.254.129 (200.143.254.129)  0.693 ms  0.613 ms  0.610 ms
 8  sp2-pr-oi.bkb.rnp.br (200.143.252.61)  8.297 ms  8.302 ms  8.297 ms
 9  sp-sp2.bkb.rnp.br (200.143.253.37)  7.470 ms  7.476 ms  7.470 ms
10  * * *
11  209.85.254.11 (209.85.254.11)  8.868 ms  8.820 ms  9.178 ms
12  209.85.143.21 (209.85.143.21)  8.766 ms  8.938 ms  8.929 ms
13  gru09s17-in-f4.1e100.net (216.58.222.36)  8.527 ms  8.619 ms  8.571 ms
```

Figura 19 - Traceroute de um computador na rede 192.168.1.0/24 para o Google.
Fonte: Autoria Própria.

```
200.134.10.254 - P
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri May 13 17:33:14 2016 from 179.154.54.250
root@klingon:~# ifconfig
eth0      Link encap:Ethernet  Endereço de HW 00:02:2a:e4:c6:53
          inet end.: 200.134.25.253  Bcast:200.134.25.255  Masc:255.255.255.0
          endereço inet6: fe80::202:2aff:fee4:c653/64 Escopo:Link
          UP BROADCASTRUNNING MULTICAST  MTU:1500  Métrica:1
          RX packets:349798583 errors:0 dropped:212061 overruns:0 frame:0
          TX packets:312869850 errors:0 dropped:0 overruns:0 carrier:0
          colisões:0 txqueuelen:1000
          RX bytes:1163303395 (1.0 GiB)  TX bytes:786502922 (750.0 MiB)
          IRQ:16 Endereço de E/S:0xce00

eth1      Link encap:Ethernet  Endereço de HW 00:08:54:2c:ca:57
          inet end.: 200.134.10.254  Bcast:200.134.10.255  Masc:255.255.255.0
          endereço inet6: fe80::208:54ff:fe2c:ca57/64 Escopo:Link
          UP BROADCASTRUNNING MULTICAST  MTU:1500  Métrica:1
          RX packets:313649011 errors:0 dropped:34320 overruns:0 frame:0
          TX packets:341414086 errors:0 dropped:0 overruns:0 carrier:0
          colisões:0 txqueuelen:1000
          RX bytes:1164985292 (1.0 GiB)  TX bytes:2798538321 (2.6 GiB)
          IRQ:18 Endereço de E/S:0xf000
```

Figura 20 - Interfaces de rede do Servidor Klingon.
Fonte: Autoria Própria.

Em outra reunião com o administrador da RLE verificou-se a existência de três *switches*: o primeiro situado na rede 192.168.100.0/24 (figura 21), já os outros dois estão na rede 200.134.10.0/24 (figura 22) e (figura 23).

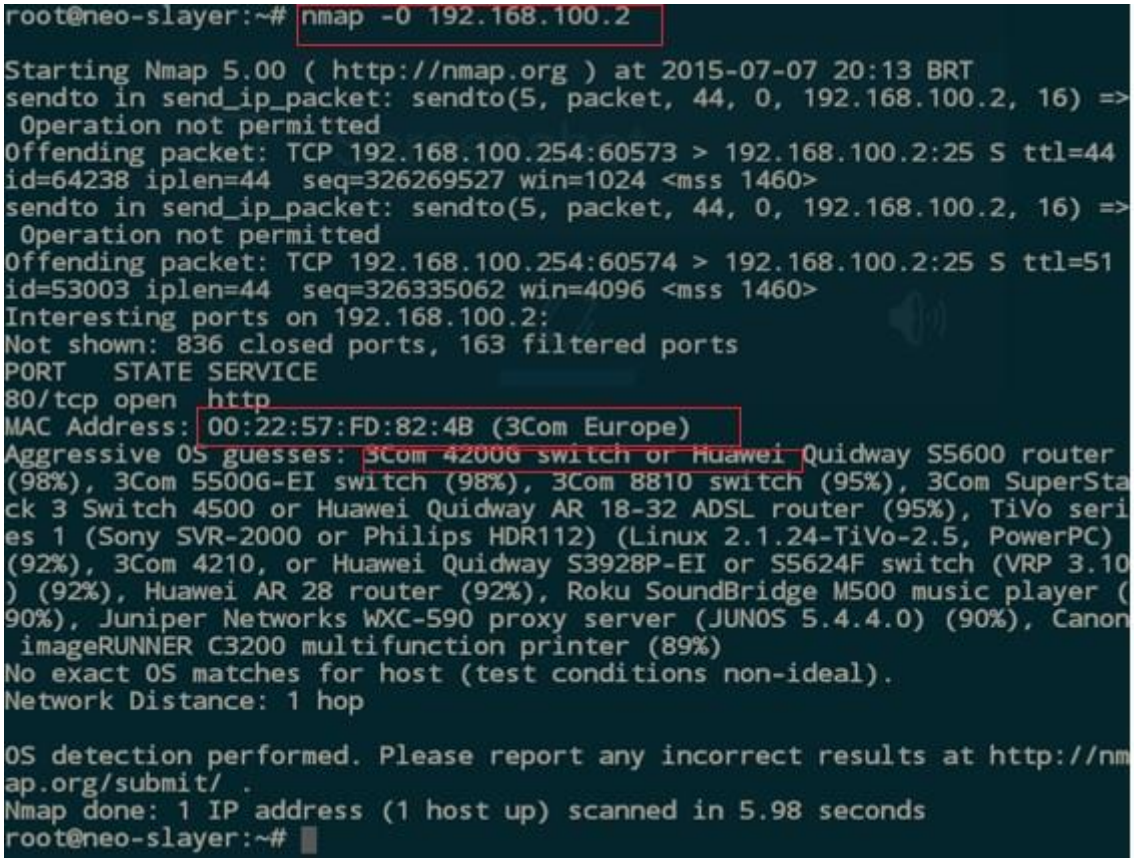

Figura 21 - Switch RLE
Fonte: Autoria Própria.

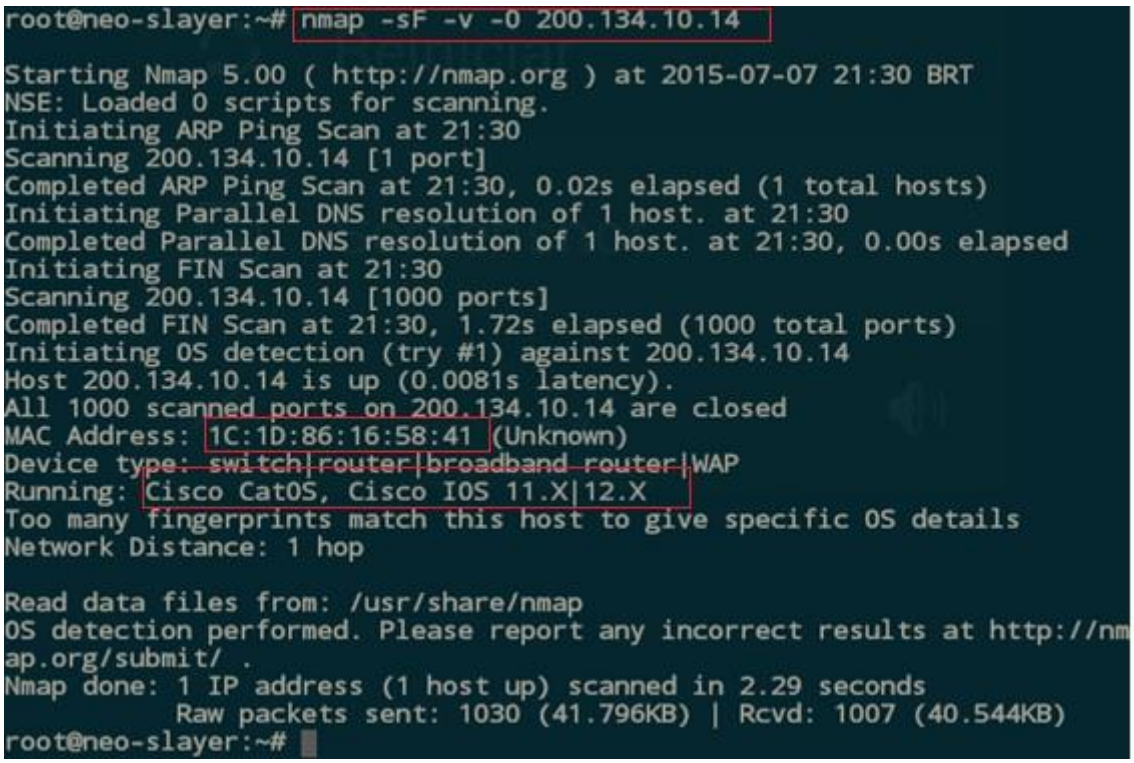

Figura 22 - Switch CGR
Fonte: Autoria Própria.

```
Starting Nmap 6.40 ( http://nmap.org ) at 2015-09-03 21:20 BRT
Nmap scan report for 200.134.10.176
Host is up (0.013s latency).
Not shown: 254 open|filtered protocols
PROTOCOL STATE SERVICE
1        open  icmp
6        open  tcp
MAC Address: 00:24:73:92:34:43 (3Com Europe)

Nmap done: 1 IP address (1 host up) scanned in 8.91 seconds
root@wilian-Inspiron-3442:~#
```

Figura 23 - Switch DAINF
Fonte: Autoria Própria.

Em uma segunda etapa, ao analisar a tabela MAC (figura 24) do *switch* 192.168.100.2, identificou-se a conexão com o *host* 192.168.100.254 por meio do endereço físico (*MAC Address)*, provando que estavam ligados fisicamente.

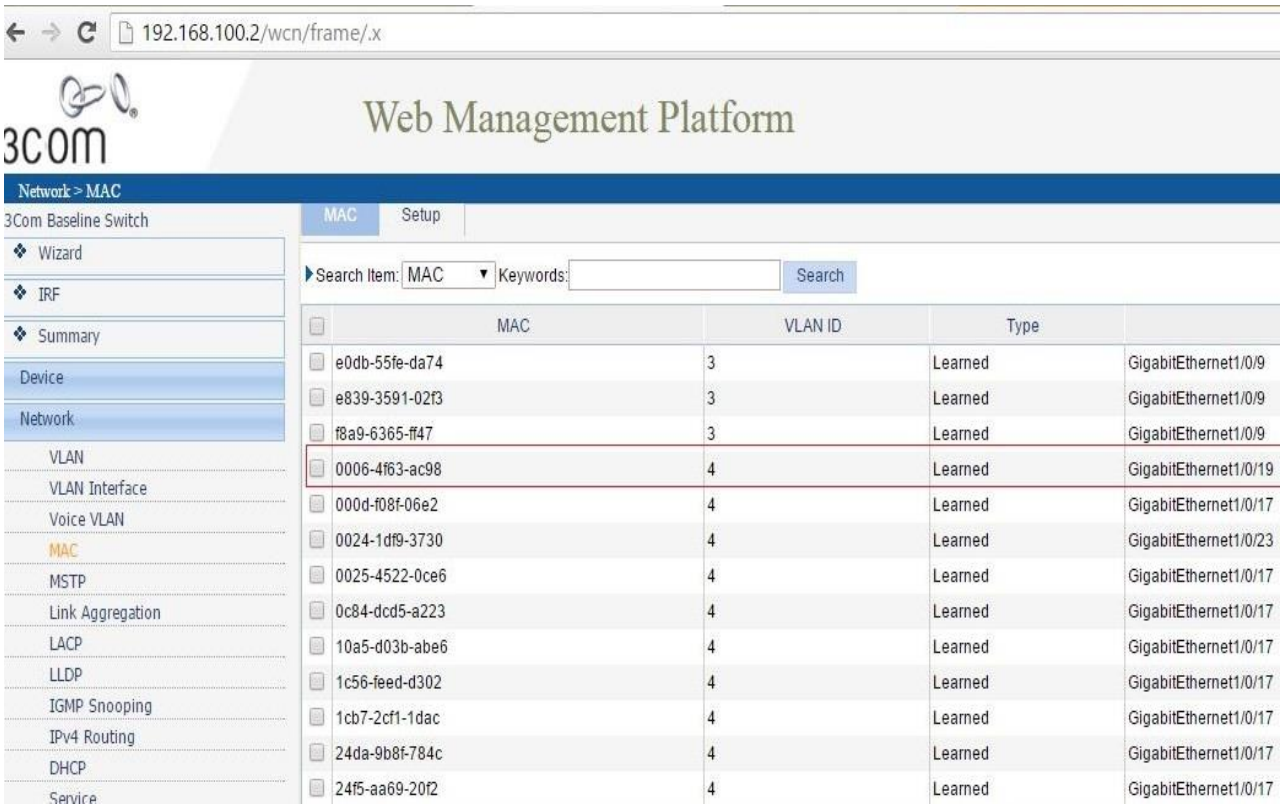

Figura 24 - Switch RLE-MAC
Fonte: Autoria Própria.

Em seguida, ao verificar o *host* 200.134.10.176, identificou-se que o mesmo estava conectado ao *switch* 200.134.10.14 via *MAC Adress* (figura 25) e ao *Slayer* 200.134.10.31, por meio do endereço físico (*MAC Address)* (figura 26).

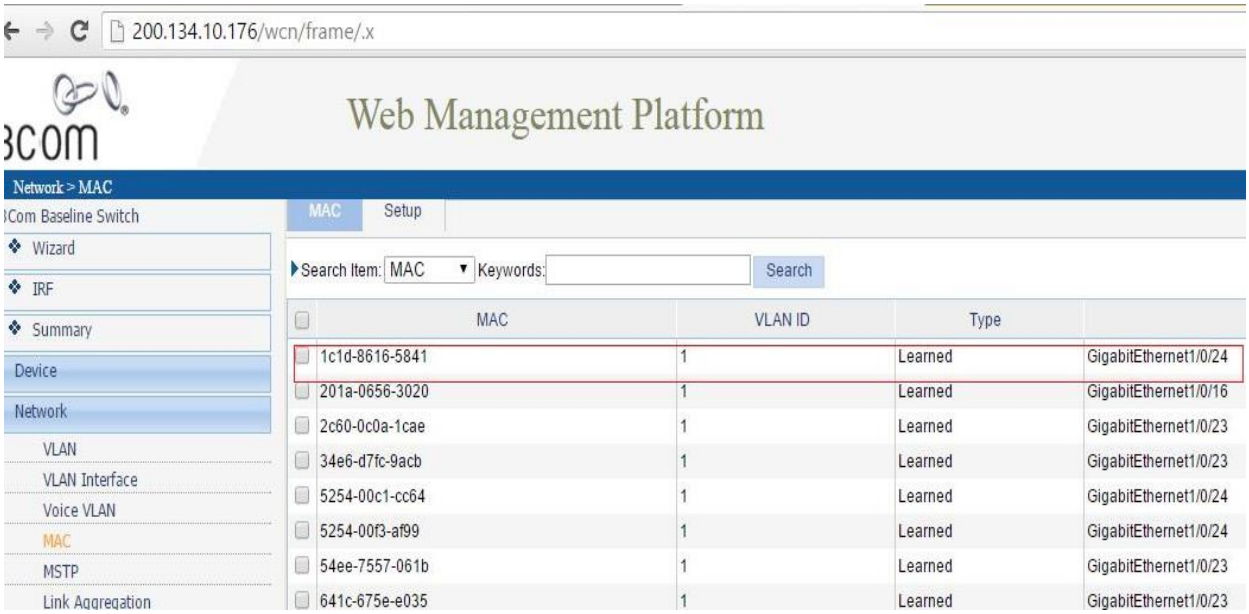

Figura 25 - Switch DAINF-MAC
Fonte: Autoria Própria.

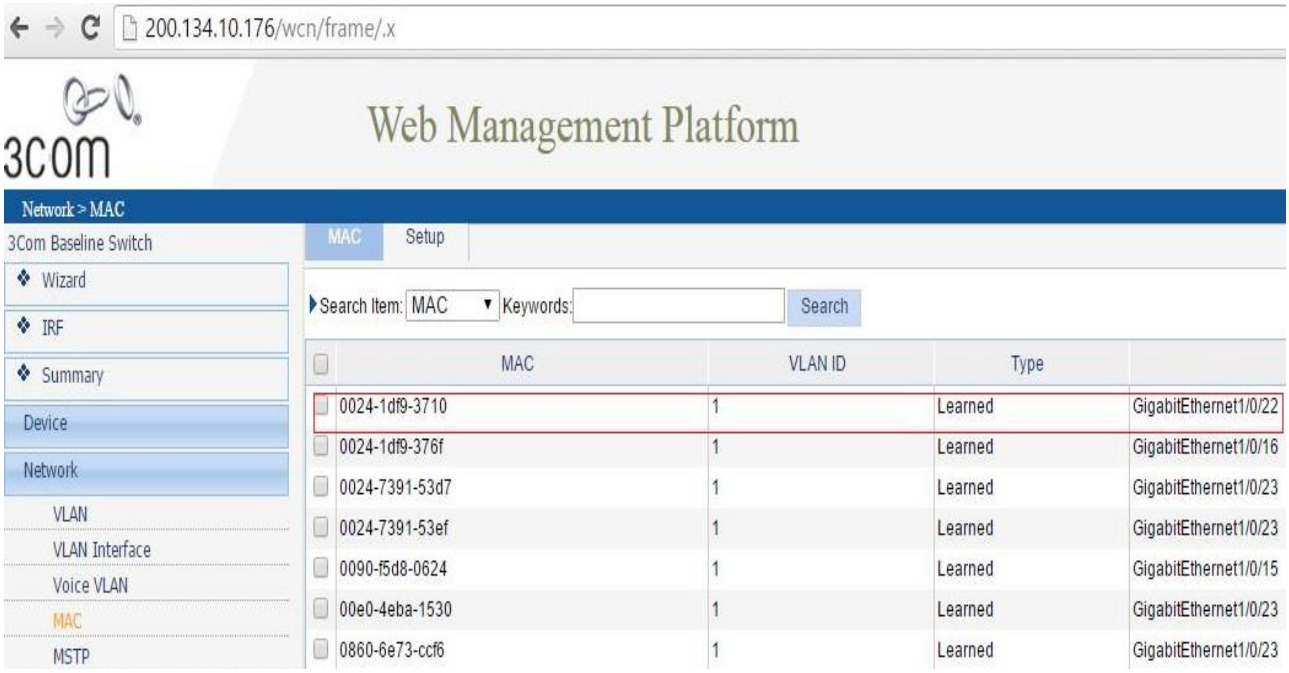

Figura 26 - Switch RLE-MAC
Fonte: Autoria Própria

Ao analisar o *switch* 200.134.10.14, constatou-se que o mesmo estava conectado ao *Klingon* IP 200.134.10.254, pois seu endereço físico (*MAC Address)* está contido na sua tabela (figura 27).

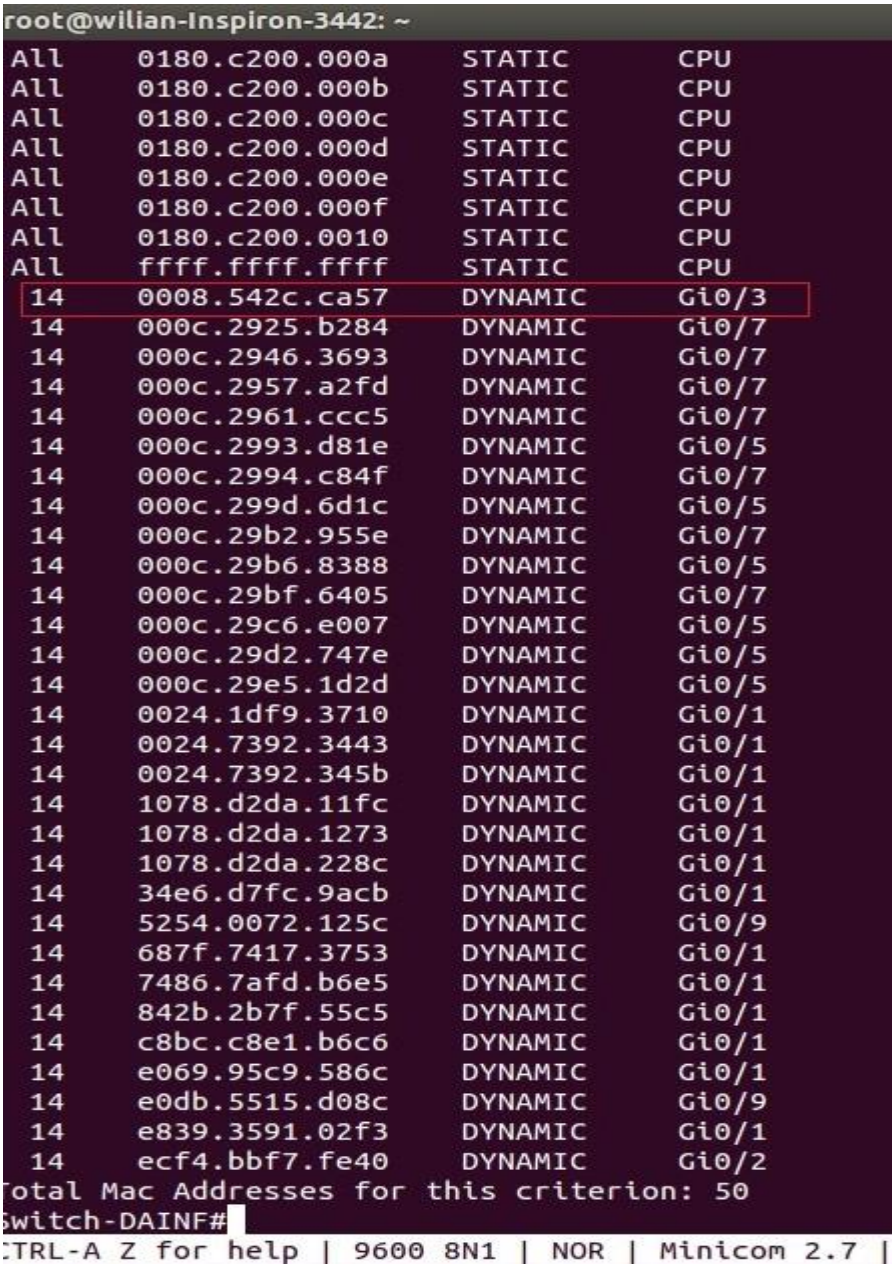

Figura 27 - Switch CGR
Fonte: Autoria Própria.

Analisando o servidor CGR IP:200.134.25.1, verifica-se que estava conectado ao *switch* 200.134.10.14 (figura 28). Este servidor contém máquinas virtuais usadas na RLE.

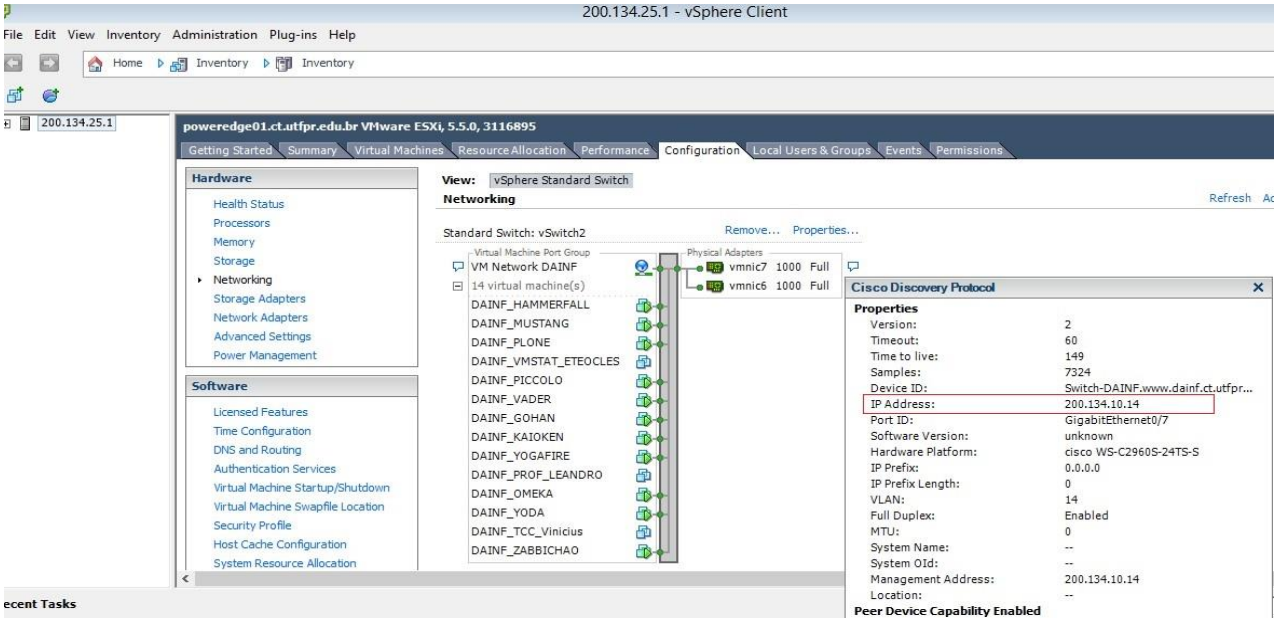

Figura 28 Servidor RLE
Fonte: Autoria Própria.

Mapeamento lógico da Rede Local de Ensino (RLE):

Após a análise da rede com as ferramentas NMAP e Tracerout foi possível realizar o mapa lógico da rede. (Figura 29)

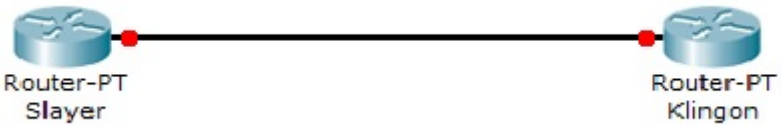

Figura 29 - Mapeamento Lógico
Fonte: Autoria Própria.

Para o mapeamento físico foi utilizada log dos equipamentos ferramentas NMAP e Traceroute e após a análise definiu-se que a rede estava conforme a representação a seguir (figura 30).

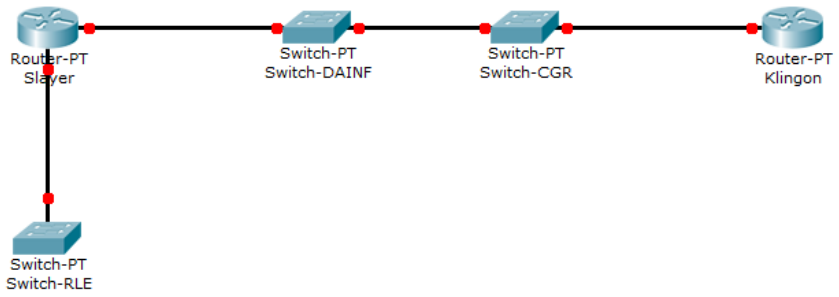

Figura 30 - Mapeamento físico
Fonte: Autoria Própria.

5.3 MONITORAMENTO

Para monitorar, os *hosts* foram divididos em quatro grupos: RLE, DAINF, CGR e MOODLE. Segue listagem do que cada *host* contém:

- RLE contém switch do RLE;
- DAINF contém, servidor virtual de e-mail, servidor site do DAINF e *switch* DAINF;
- CGR contém *switch* CGR;
- *MOODLE* contém servidor virtual do *MOODLE*.

A partir disso, foram analisadas as interfaces dos *switches* para definir quais seriam monitoradas quanto a sua disponibilidade e tráfego.

- Grupo RLE: *switch* RLE
  - GigabitEthernet1/0/18;
  - GigabitEthernet1/0/19;
  - GigabitEthernet1/0/20;
  - GigabitEthernet1/0/21;
  - GigabitEthernet1/0/22;
  - GigabitEthernet1/0/24.

Essas são as interfaces que devem ser monitoradas, pois estão conectadas aos switches das salas (figuras 31 e 32).

IP Address	MAC Address	VLAN ID	Port
192.168.106.181	e069-95c8-ee64	5	GigabitEthernet1/0/18
192.168.106.214	e069-95c9-0a32	5	GigabitEthernet1/0/18
192.168.106.215	e069-95d7-ed5c	5	GigabitEthernet1/0/18
192.168.106.235	e069-95d7-86a3	5	GigabitEthernet1/0/18
192.168.100.254	0006-4f63-ac98	4	GigabitEthernet1/0/19
192.168.107.16	0001-6c7a-d958	7	GigabitEthernet1/0/20
192.168.107.27	0024-1df9-37dc	7	GigabitEthernet1/0/20
192.168.107.38	001f-d0e4-e473	7	GigabitEthernet1/0/20
192.168.107.48	0024-7392-91a1	7	GigabitEthernet1/0/20
192.168.107.49	0024-7392-2a91	7	GigabitEthernet1/0/20
192.168.107.51	0024-1df9-5980	7	GigabitEthernet1/0/20
192.168.107.74	0024-1df9-3604	7	GigabitEthernet1/0/20
192.168.107.95	001f-d0e4-da05	7	GigabitEthernet1/0/20
192.168.107.108	0024-1df9-3793	7	GigabitEthernet1/0/20
192.168.107.110	0024-1df9-5965	7	GigabitEthernet1/0/20
192.168.107.117	001f-d0e5-b7d4	7	GigabitEthernet1/0/20
192.168.107.129	5442-49ad-4497	7	GigabitEthernet1/0/20
192.168.107.166	0024-1df9-36dd	7	GigabitEthernet1/0/20
192.168.107.194	0024-1df9-377c	7	GigabitEthernet1/0/20
192.168.107.212	0024-1df9-3798	7	GigabitEthernet1/0/20

Figura 31 - Interfaces Switch Rle
Fonte: Autoria Própria.

IP Address	MAC Address	VLAN ID	Port
192.168.204.231	c80a-a9f4-65a5	6	GigabitEthernet1/0/21
192.168.204.243	9048-9aef-325f	6	GigabitEthernet1/0/21
192.168.204.246	7486-7af5-49d4	6	GigabitEthernet1/0/21
192.168.204.249	b8af-6788-98af	6	GigabitEthernet1/0/21
192.168.108.50	20fd-f1be-bc3d	9	GigabitEthernet1/0/22
192.168.108.145	001f-d0e4-e983	9	GigabitEthernet1/0/22
192.168.100.5	0024-1df9-3730	4	GigabitEthernet1/0/23
192.168.109.10	20fd-f1be-bfa3	8	GigabitEthernet1/0/24
192.168.109.245	001f-d0e5-b3d4	8	GigabitEthernet1/0/24

Figura 32 - Interfaces Switch Rle
Fonte: Autoria Própria

- Grupo DAINF: *switch* DAINF
    - GigabitEthernet1/0/24
    - GigabitEthernet1/0/22

Essas são interfaces que estão conectadas ao *switch* CGR e a *Slayer*, (figura 33 e 34).

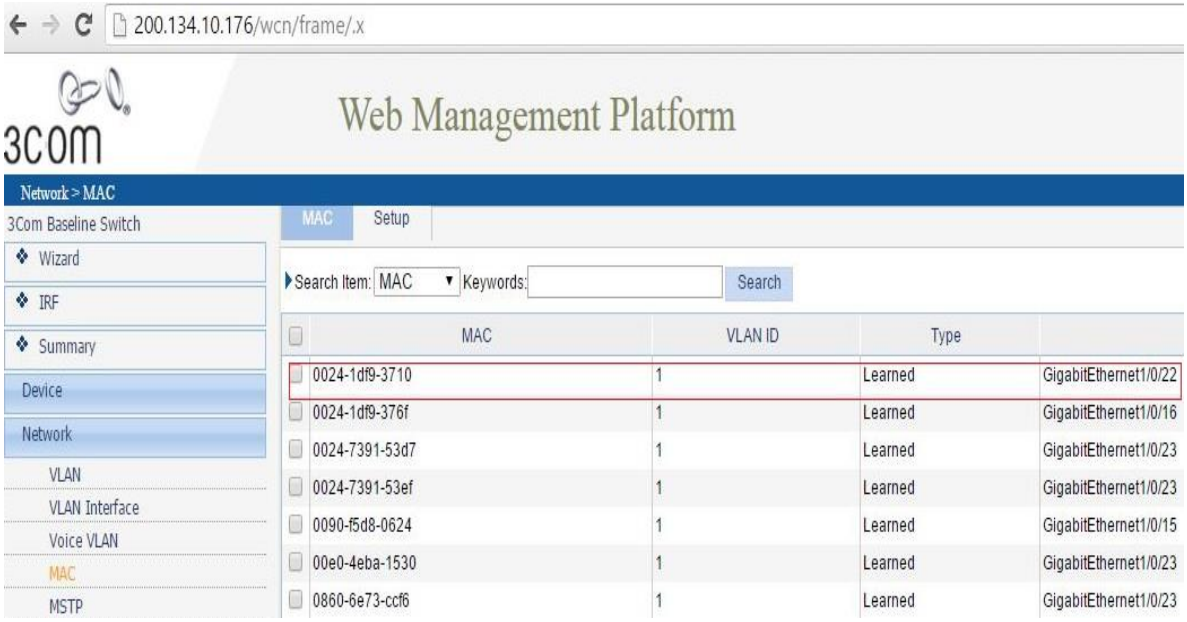

Figura 33 - Interfaces Switch DAINF
Fonte: Autoria Própria.

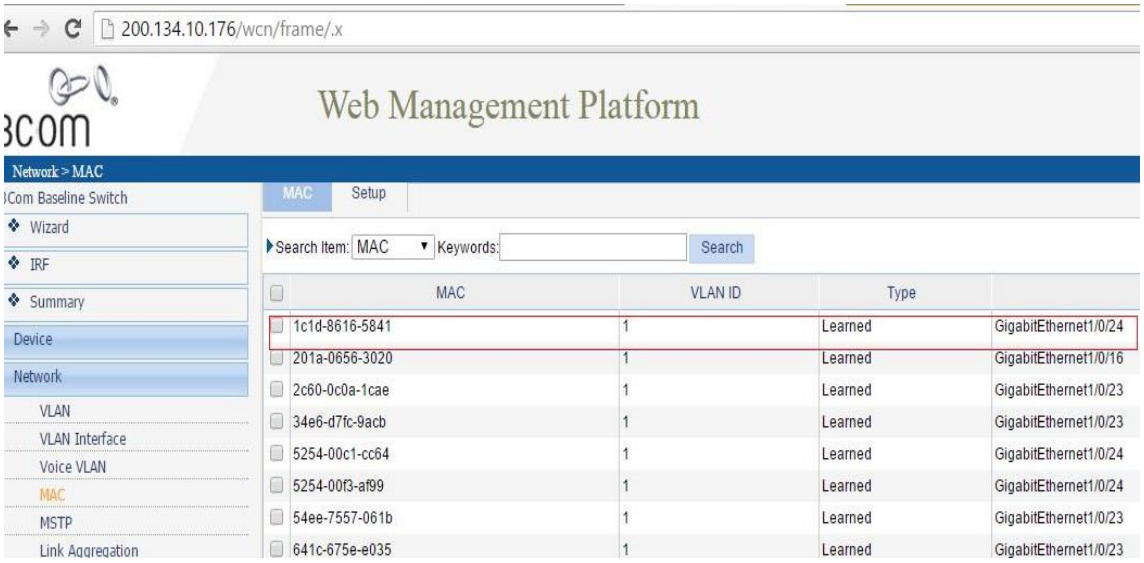

Figura 34 Interfaces Switch DAINF
Fonte: Autoria Própria.

- Grupo CGR: switch CGR
    - GI0/1;
    - GI0/3;
    - GI0/5;
    - GI0/7.

Para definir quais interfaces seriam monitoradas, foram analisados os endereços físicos (*MACs)* destacados na figura a seguir, que são dos hosts 200.134.10.1, 200.134.10.5, 200.134.10.6, 200.134.10.7, 200.134.10.27, 200.134.10.32, 200.134.10.37, 200.134.10.254, respectivamente, conforme visto no mapeamento de serviços. Para esse grupo de IPs estão contidas as máquinas virtuais do DAINF e seu *gateway*.

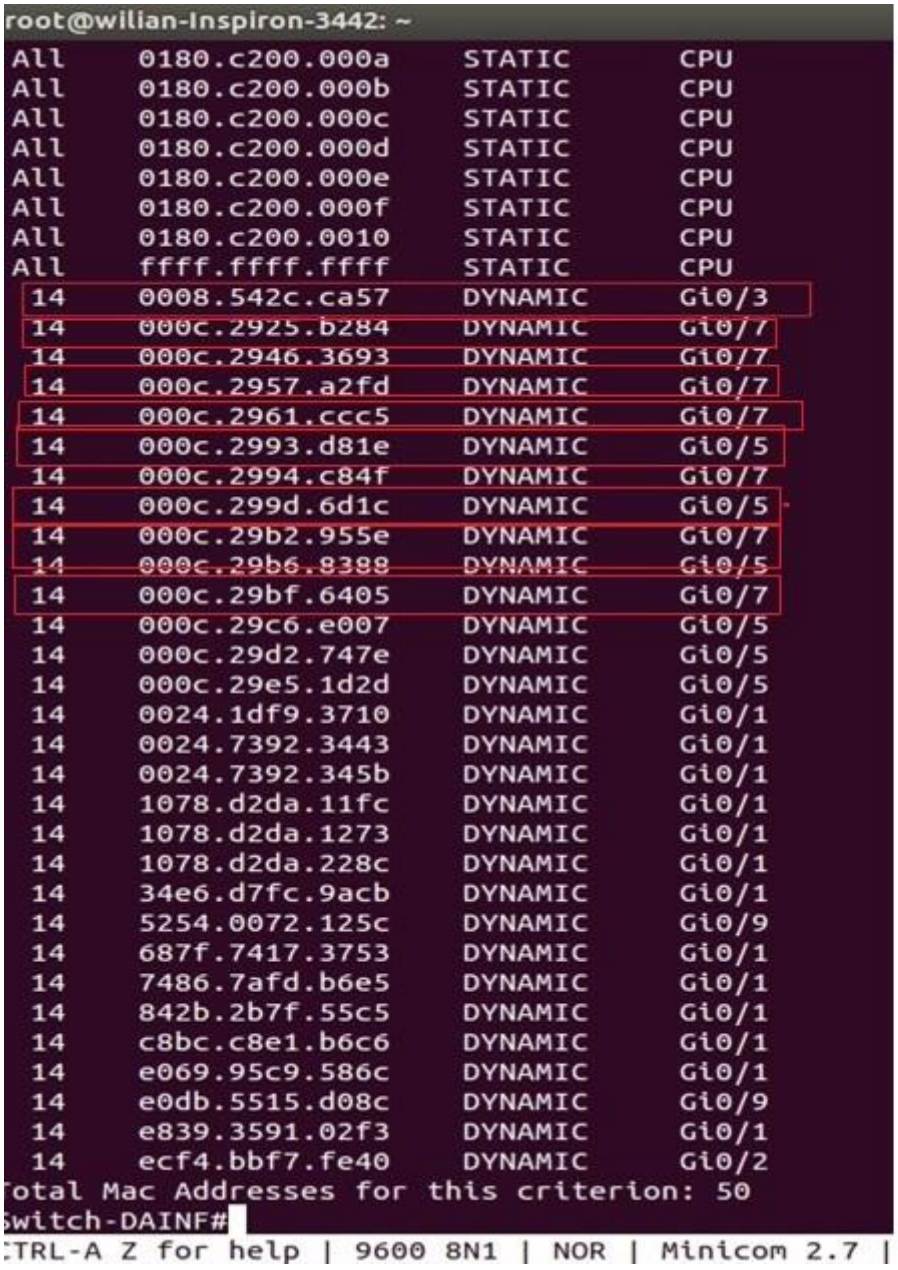

Figura 35 Interfaces Switch CGR
Fonte: Autoria Própria.

Para construir o monitoramento das interfaces, através da ferramenta *Zabbix*, foi necessário seguir algumas etapas:

- Etapa 1: Criação do Grupo RLE;

- Etapa 2: Criar e associar o *host switch*-RLE ao grupo;
- Etapa 3: Associar um *template* ao grupo.

A seguir, tem-se a configuração para uma das interfaces monitoradas, que foram replicadas para as outras.

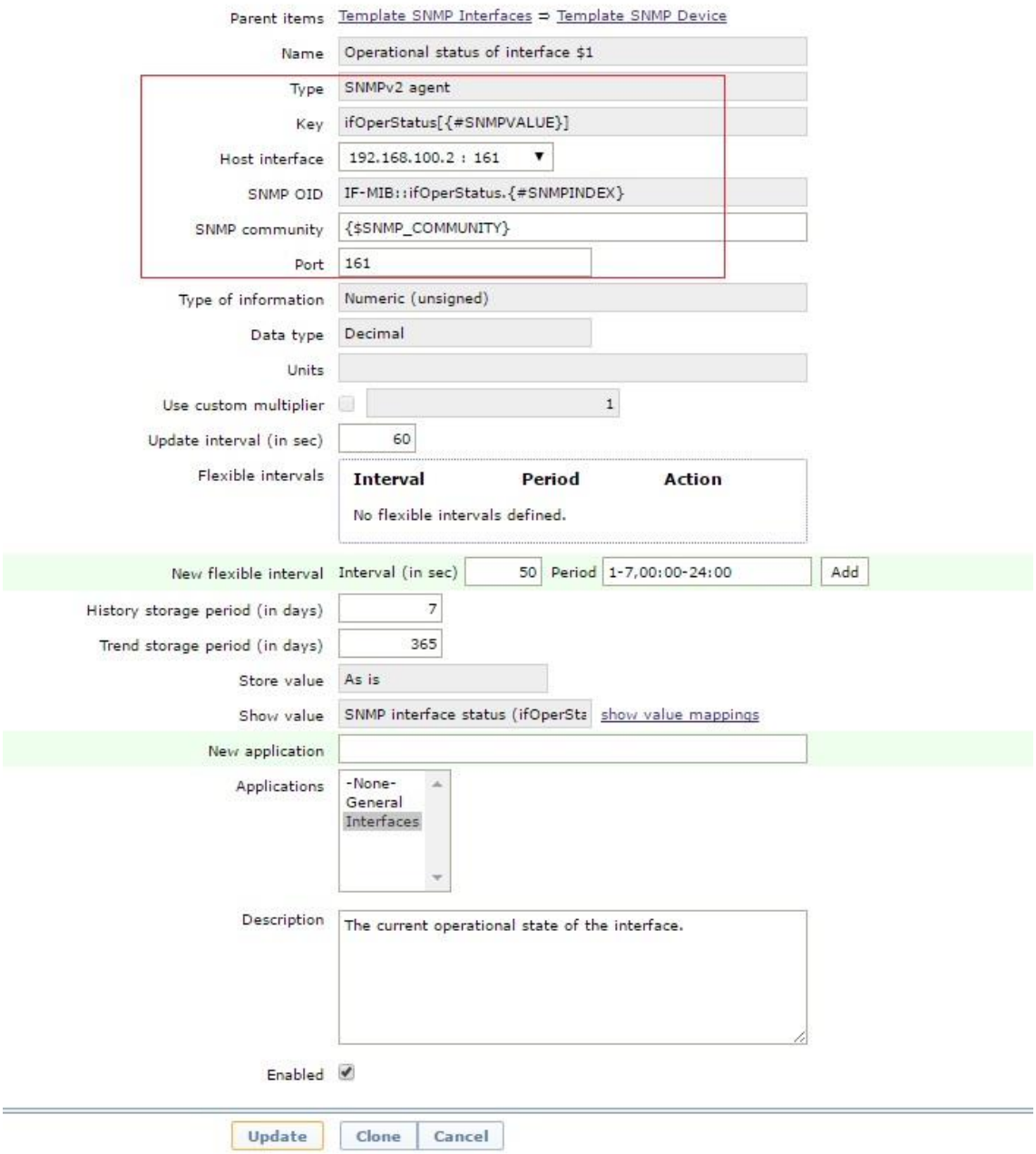

Figura 36 - Configuração para as Interfaces-Switch RLE
Fonte: Autoria Própria.

No quadro em destaque da figura acima, tem-se alguns itens que foram configurados para a realização do monitoramento das interfaces, segundo a disponibilidade:

- *Type*: Versão *SNMP* usada para monitoramento;

- *Key*: ifOperStatus[{#SNMPVALUE}], nome único para identificação, onde "{#SNMPVALUE}" comporta-se como se fosse uma variável:
- *Host* interface: IP do *Host* a ser monitorado:
- SNMP OID: OID 1.3.6.1.2.1.2.2.1.8 que determina o status da porta monitorada a qual foi substituída, IF-MIB::ifOperStatus.{#SNMPINDEX}, para facilitar a compreensão, onde .{#SNMPINDEX} também comporta-se como se fosse uma variável:
- SNMP *community*: comunidade que está configurada o *host* monitorado, onde {$SNMP_COMMUNITY} comporta-se como se fosse uma variável obtendo a comunidade do *host* monitorado:
- *Port*: porta que está configurada no *host* para obter as informações SNMP.

Configuração para o tráfego de entrada das interfaces:

- *Key*: De ifOperStatus[{#SNMPVALUE}] para ifInOctets[{#SNMPVALUE}];
- SNMP OID: De IF-MIB::ifOperStatus.{#SNMPINDEX} para IF-MIB::ifInOctets.{#SNMPINDEX}, essa OID determina o tráfego de entrada para interface.

Configuração para o tráfego de saída:

- *Key*: ifOutOctets[{#SNMPVALUE}];
- SNMP OID : IF-MIB::ifOutOctets.{#SNMPINDEX}.

Configuração das *Triggers* (figura 37), que é composta por:

- {Switch RLE:ifOperStatus[{#SNMPVALUE}].last()}=2, onde:
  - *Switch* RLE, é o *host* a ser monitorado;
  - ifOperStatus[{#SNMPVALUE}].last(), recebe o último valor da interface; e
  - =2 é a constate a ser comparada.

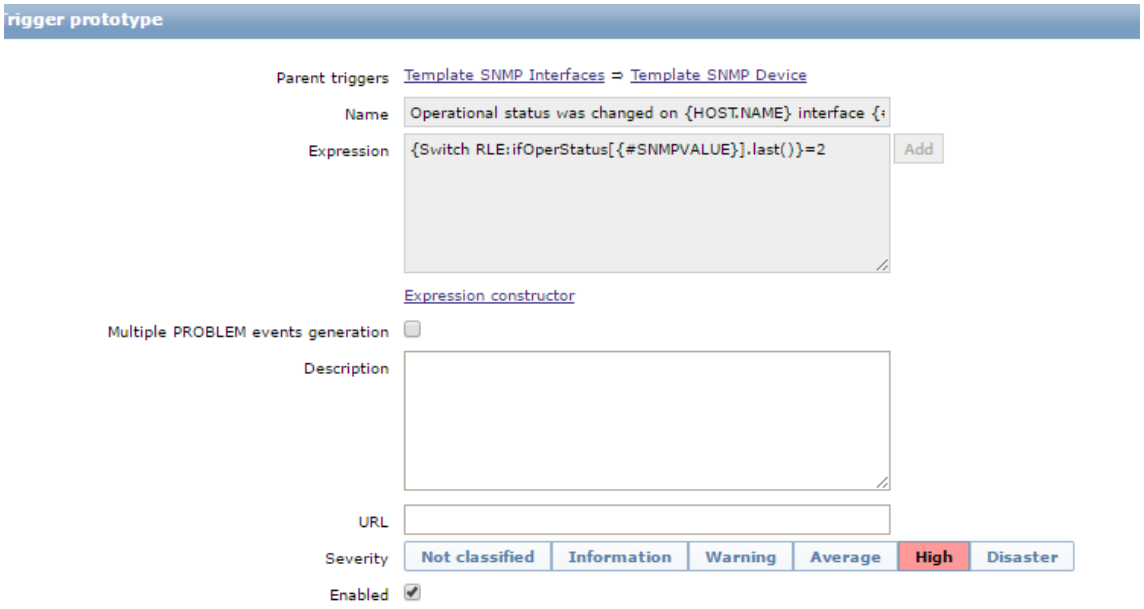

Figura 37 - Configuração das Triggers-Switch RLE
Fonte: Autoria Própria.

A expressão "{Switch RLE:ifOperStatus[{#SNMPVALUE}].last()}=2" quer informar que, se uma interface do host 192.168.100.2 tiver um valor igual a 2 ocorrerá um aviso *high*.

Para o monitoramento do *switch* DAINF utilizou-se a mesma configuração citada anteriormente:

- Etapa 1: Criação do Grupo DAINF;
- Etapa 2: Criar e associar o *host switch*-DAINF ao grupo;
- Etapa 3: Associar um *template* ao grupo.

Segue a configuração para uma das interfaces monitoradas que foi replicada para as outras.

No quadro em destaque da figura abaixo, temos alguns itens que foram configurados para a realização do monitoramento das interfaces, segundo a sua disponibilidade.

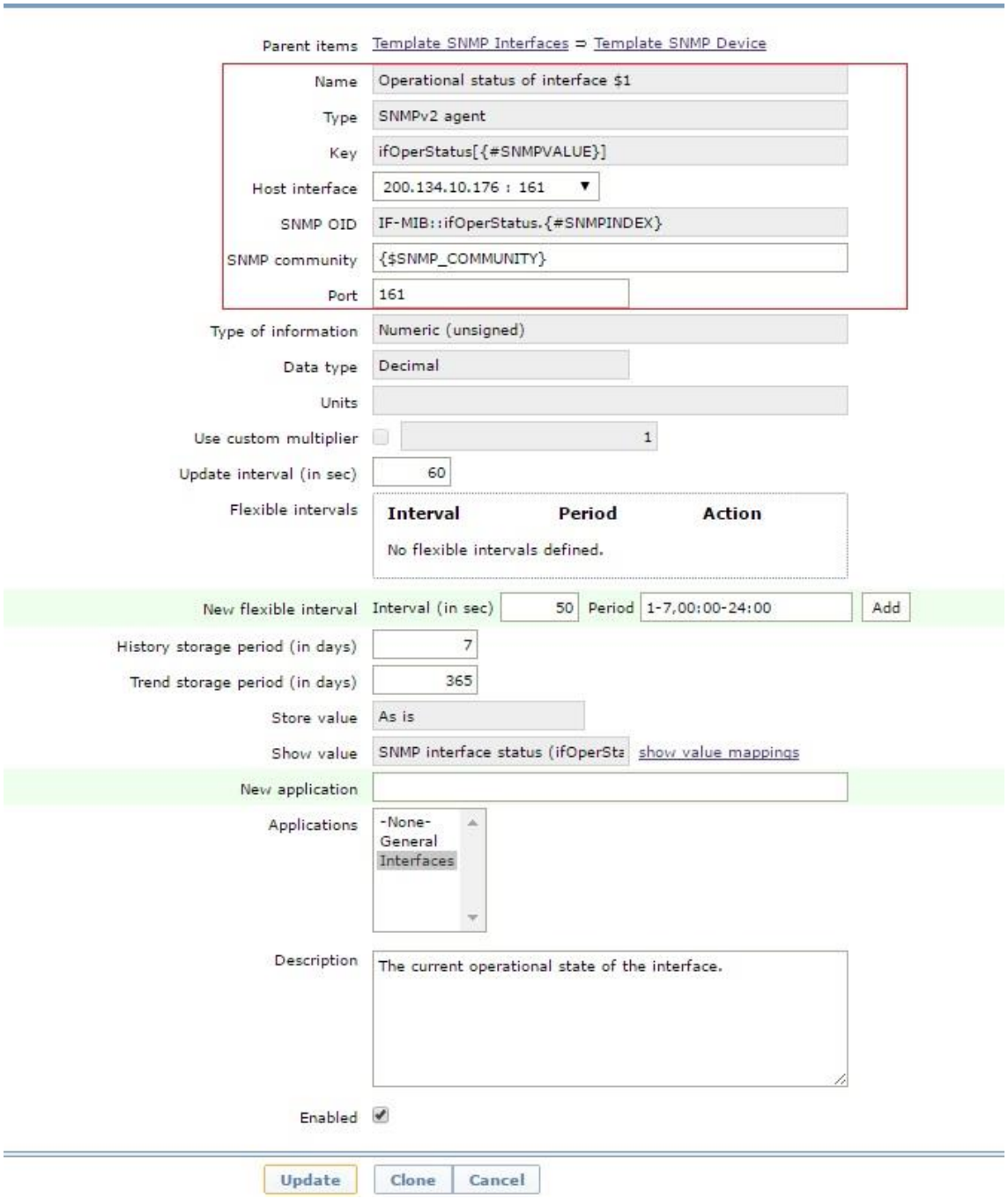

Figura 38 Configuração para interfaces-Switch DAINF
Fonte: Autoria Própria.

Onde:

- *Type* : Versão *SNMP* usada para monitoramento;
- *Key*: ifOperStatus[{#SNMPVALUE}], nome único para identificação, sendo que "{#SNMPVALUE}" comporta-se como se fosse uma variável;
- *Host* interface: IP do *Host* a ser monitorado;

- SNMP OID: OID 1.3.6.1.2.1.2.2.1.8 que determina o status da porta monitorada que foi substituída, IF-MIB::ifOperStatus.{#SNMPINDEX}, para facilitar a compreensão, onde .{#SNMPINDEX} também comporta-se como se fosse uma variável;
- SNMP *community*: comunidade que está configurada ao *host* monitorado, onde {$SNMP_COMMUNITY} comporta-se como se fosse uma variável obtendo a comunidade do *host* monitorado;
- Port: porta que está configurada no *host* para obter as informações SNMP.

Para o tráfego de entrada das interfaces foram configuradas com alteração nos itens a seguir:

- *Key*: De ifOperStatus[{#SNMPVALUE}] para ifInOctets[{#SNMPVALUE}];
- SNMP OID: De IF-MIB::ifOperStatus.{#SNMPINDEX} para IF-MIB::ifInOctets.{#SNMPINDEX} essa OID determina o tráfego de entrada para interface.

Já para tráfego de saída a configuração ficou conforme a seguir:

- *Key*: ifOutOctets[{#SNMPVALUE}];
- SNMP OID: IF-MIB::ifOutOctets.{#SNMPINDEX}.

O próximo passo foi a configuração das *Triggers* (Figura 39), que é composta por:

- {Switch RLE:ifOperStatus[{#SNMPVALUE}].last()}=2, onde:
  - *Switch* DAINF, é o *host* a ser monitorado;
  - ifOperStatus[{#SNMPVALUE}].last(), recebe o último valor da interface;
  - =2 é constate a ser comparada.

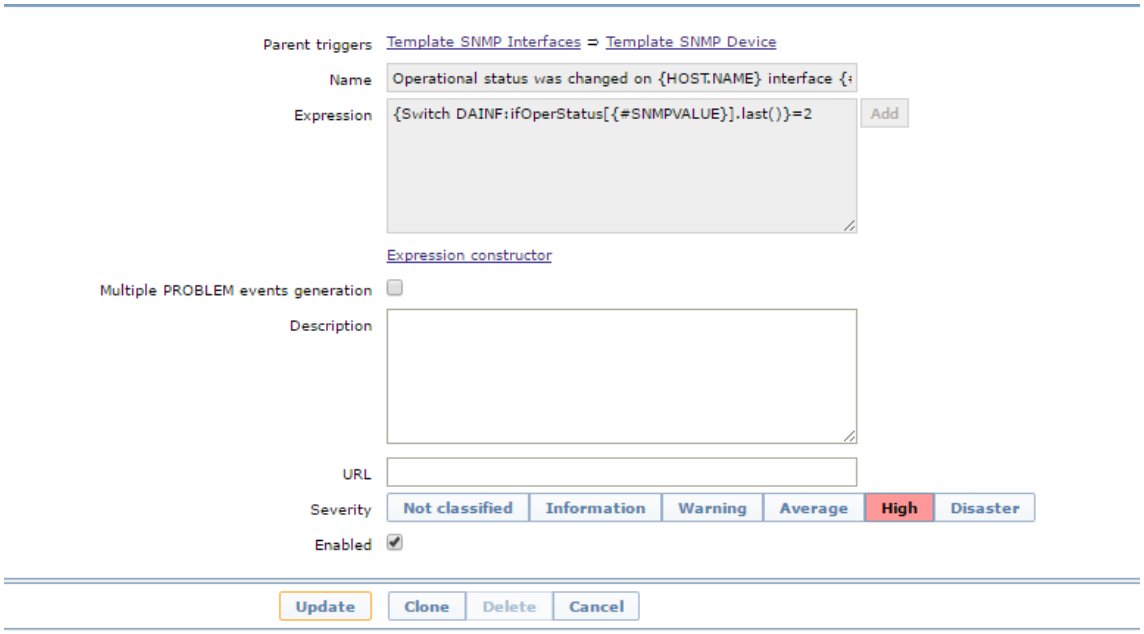

Figura 39 Configuração das Triggers-Switch DAINF
Fonte: Autoria Própria.

A expressão "{Switch DAINF:ifOperStatus[{#SNMPVALUE}].last()}=2"quer informar que, se uma interface do *host* 200.134.10.176 tiver um valor igual a 2, ocorrerá um aviso *high*.

Para o site do DAINF, o monitoramento se deu através do protocolo *HTTP*, a partir disso, gerou-se uma *trigger* (figura abaixo), que dispara somente se a resposta da requisição ao site http://www2.dainf.ct.utfpr.edu.br/ for diferente de 200.

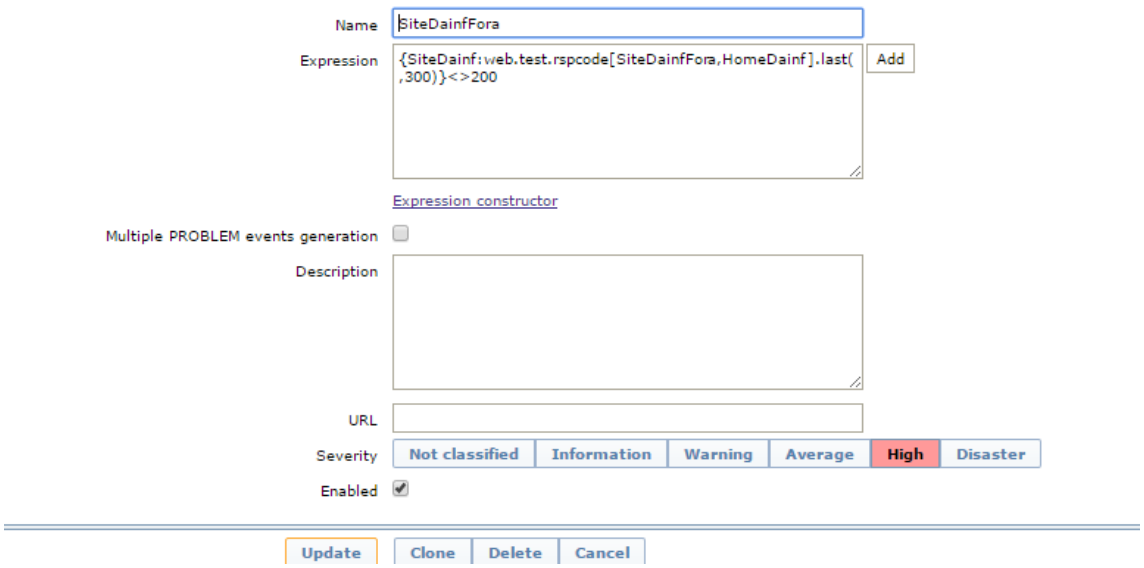

Figura 40 - Configuração da Trigger para Site DAINF
Fonte: Autoria Própria.

Para o *host* e-mail, que está no grupo DAINF, foram configurados os itens a seguir:

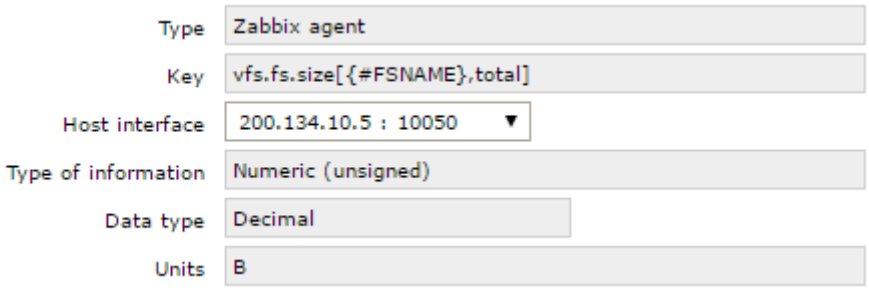

Figura 41 - Configuração Espaço para HD
Fonte: Autoria Própria.

- *Type*: agente *Zabbix* instalado no servidor de e-mail;
- *Key*: vfs.fs.size[{#FSNAME},total] essa configuração para total de espaço no *HD;*
- *Host* Interface: *host* a ser monitorado;

A configuração da *trigger* para esse host deu-se conforme a figura abaixo:

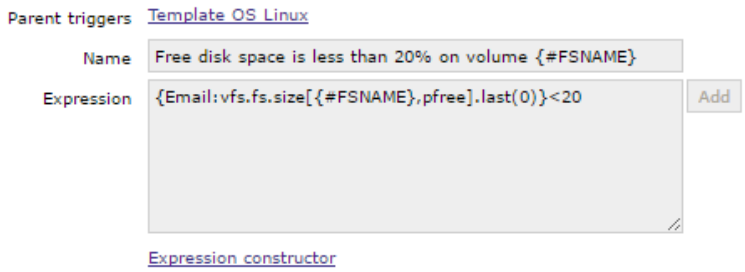

Figura 42 Configuração da Trigger para Site DAINF
Fonte: Autoria Própria.

"{Email:vfs.fs.size[{#FSNAME},pfree].last(0)}<20" caso o espaço livre em disco é esteja inferior a 20% gera um alarme de *warning*.

O monitoramento do switch CGR seguiu o mesmo padrão das anteriores:

- Etapa 1: Criação do Grupo CGR;
- Etapa 2: Criar e associar o *host switch*-CGR ao grupo;
- Etapa 3: Associar um *template* ao grupo.

Segue a configuração para uma das interfaces monitoradas que foram replicadas para as outras conforme a figura a seguir:

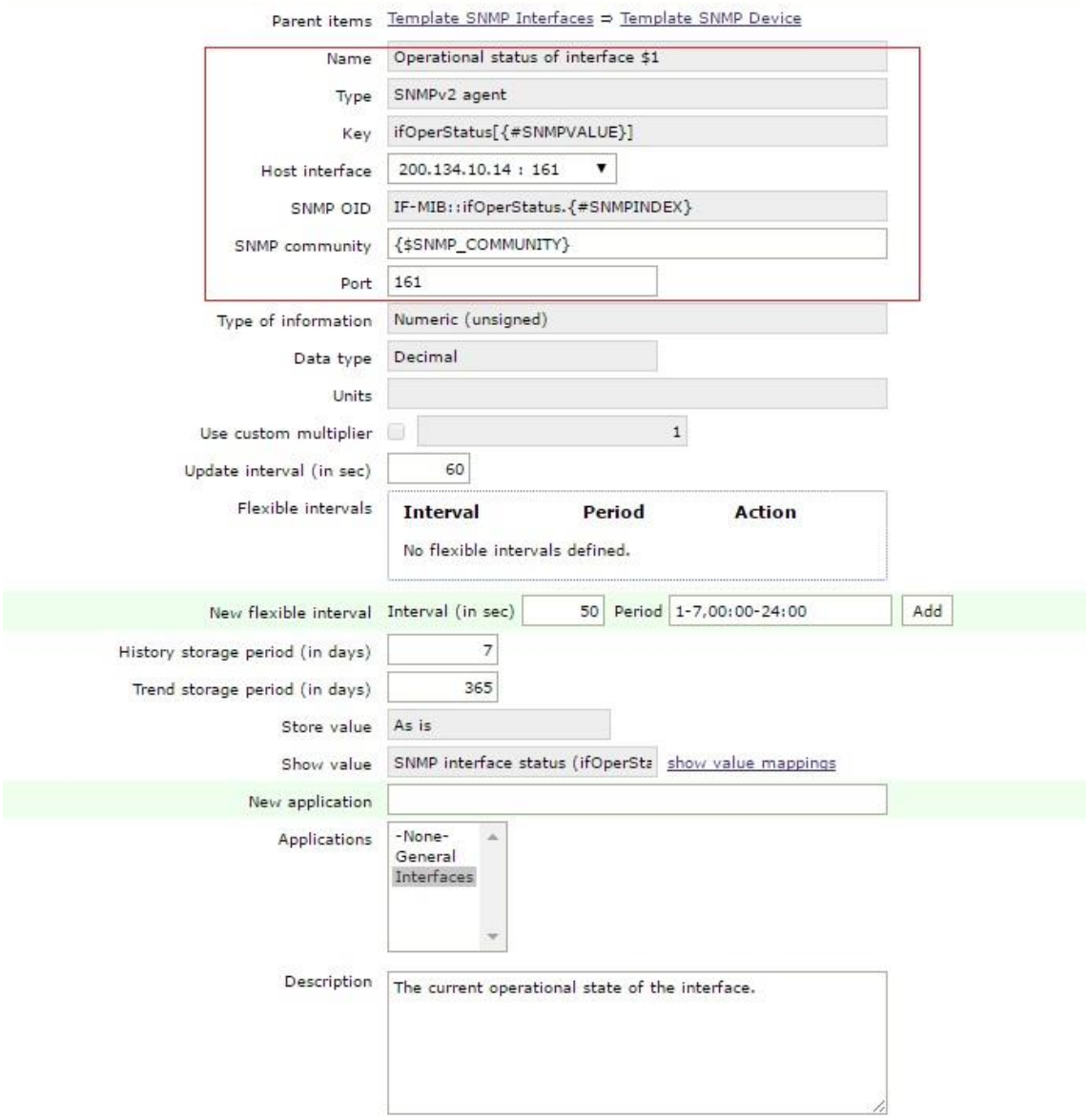

Figura 43 Configuração para interfaces-Switch CGR
Fonte: Autoria Própria.

- *Type* : Versão *SNMP* usada para monitoramento;
- *Key*: ifOperStatus[{#SNMPVALUE}], nome único para identificação, onde "{#SNMPVALUE}" comporta-se como se fosse uma variável;
- *Host* interface: IP do *Host* a ser monitorado;
- SNMP OID: OID 1.3.6.1.2.1.2.2.1.8 que determina o status da porta monitorada que foi substituída, IF-MIB::ifOperStatus.{#SNMPINDEX}, pra facilitar a compreensão em que .{#SNMPINDEX} comporta-se como se fosse uma variável;

- SNMP *community*: comunidade que está configurada ao *host* monitorado, onde {$SNMP_COMMUNITY} comporta-se como se fosse uma variável obtendo a comunidade do *host* monitorado;
- *Port*: porta que está configurada no *host* para obter as informações SNMP.

Para o tráfego de entrada das interfaces foram configuradas alteração nos itens a seguir:

- *Key*: De ifOperStatus[{#SNMPVALUE}] para ifInOctets[{#SNMPVALUE}];
- SNMP OID: De IF-MIB::ifOperStatus.{#SNMPINDEX} para IF-MIB::ifInOctets.{#SNMPINDEX} essa OID determina o tráfego de entrada

Já para o tráfego de saída a configuração ficou conforme segue:

- *Key*: ifOutOctets[{#SNMPVALUE}];
- SNMP OID : IF-MIB::ifOutOctets.{#SNMPINDEX}.

O próximo passo foi a configuração das *Triggers* (figura 44), que é composta por:

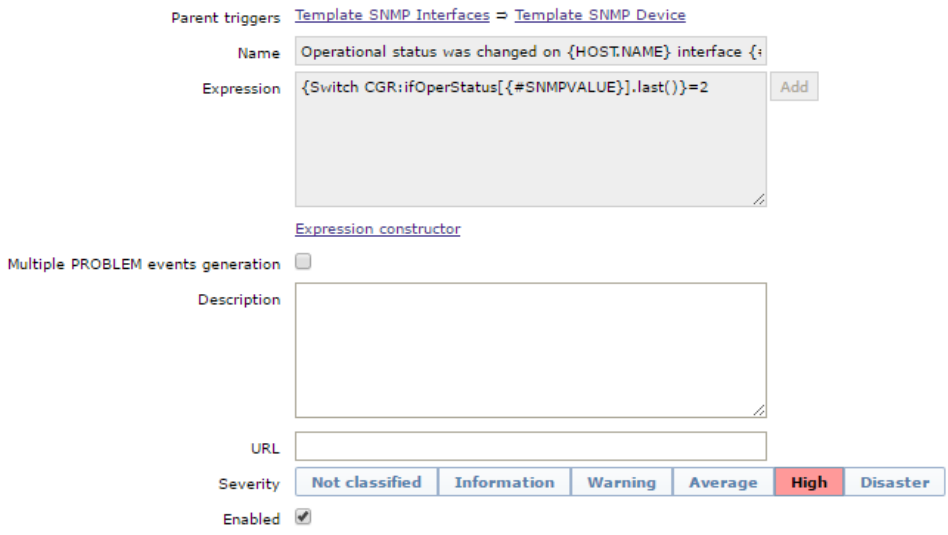

Figura 44 - Configuração das Triggers para Switch CGR
Fonte: Autoria Própria.

{Switch RLE:ifOperStatus[{#SNMPVALUE}].last()}=2, onde:

- *Switch* CGR, *host* a ser monitorado;
- ifOperStatus[{#SNMPVALUE}].last(), recebe o último valor da interface ;
- =2 constate a ser comparada.

A expressão "{Switch CGR:ifOperStatus[{#SNMPVALUE}].last()}=2" informa se uma interface do *host* 200.134.10.14 tiver um valor igual a 2 resultando em um aviso *high*.

Por fim, o *Moodle* teve a mesma configuração do site do DAINF. Seu monitoramento se deu por meio do protocolo *HTTP*, a partir disso, gerou-se uma *trigger* que dispara somente se a resposta da requisição ao site http://moodle.dainf.ct.utfpr.edu.br/ for diferente de 200 (Figura 45).

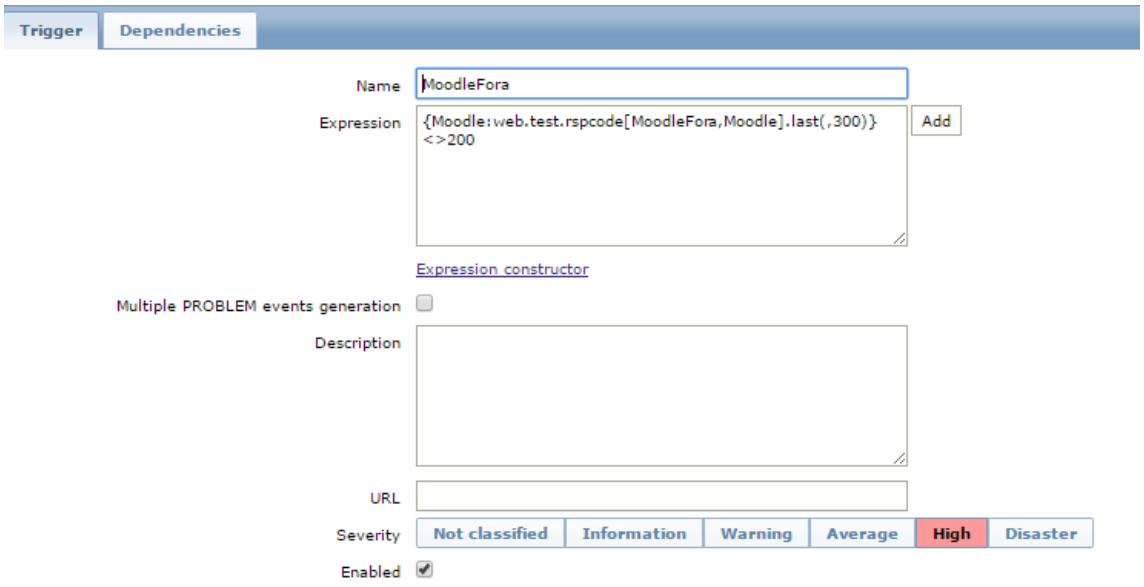

Figura 45 Configuração da Trigger para Moodle
Fonte: Autoria Própria.

Após as configurações, temos algumas telas para destacar na ferramenta *zabbix*. A primeira delas é uma visão geral (Figura 46).

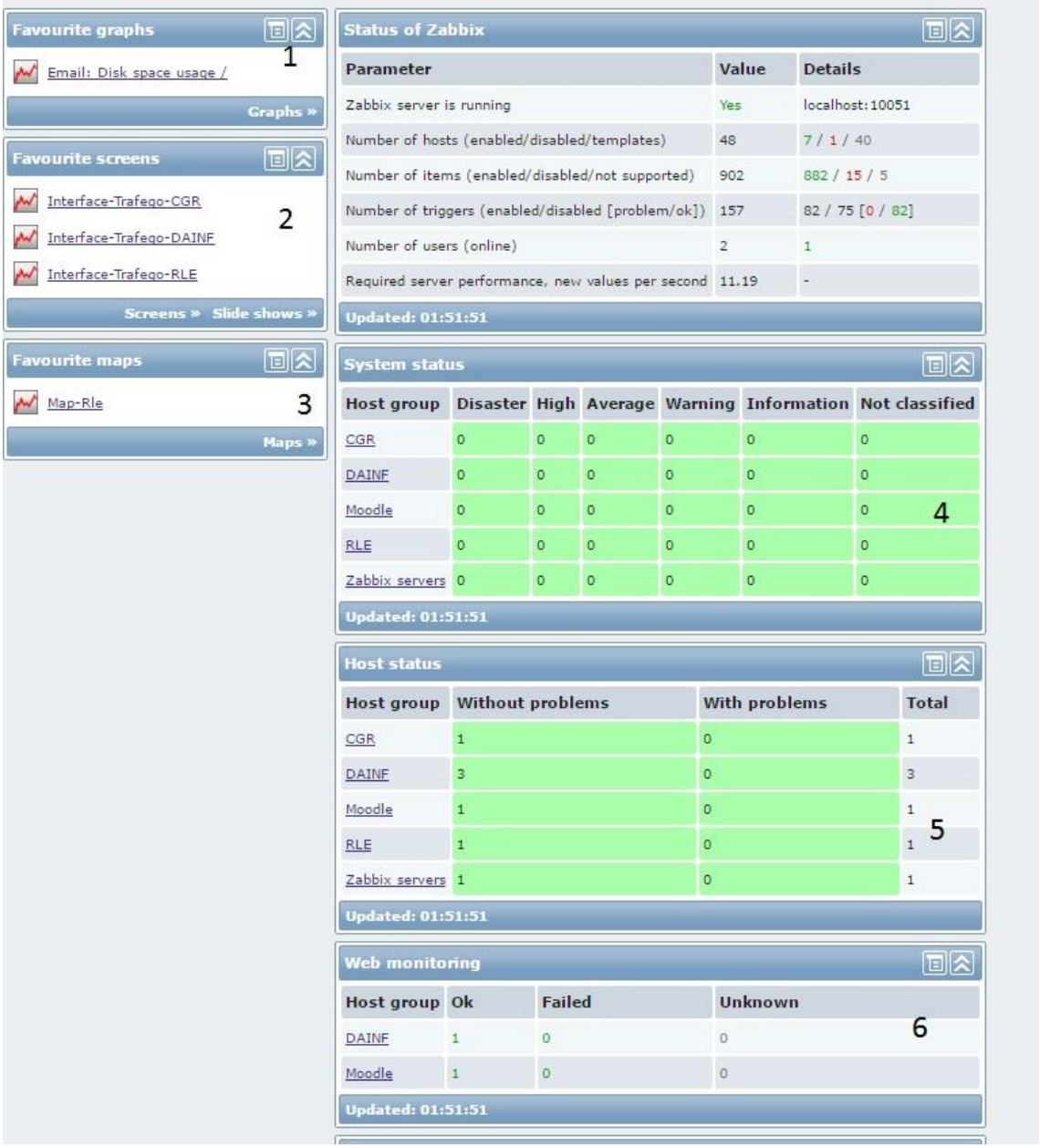

Figura 46 - Tela Inicial Zabbix
Fonte: Autoria Própria.

O item 1 destacado na figura acima é o uso do disco para o servidor de e-mail. Já o item 2, são os links para as telas de monitoramento de tráfego. O item 3 é o mapa parcial da rede principal do RLE, pois os *switches* das salas e a *Klingon* não são monitorados. Os itens 4 e 5 são visões gerais dos tipos de alarmes e se existe algum *host* com problema na rede. O item 6 é a visão para as páginas *Web* monitoradas.

Já destacado anteriormente, temos os tráfegos das interfaces divididas de acordo com o seu grupo. Para o CGR, ficaram conforme as figuras 47 e 48:

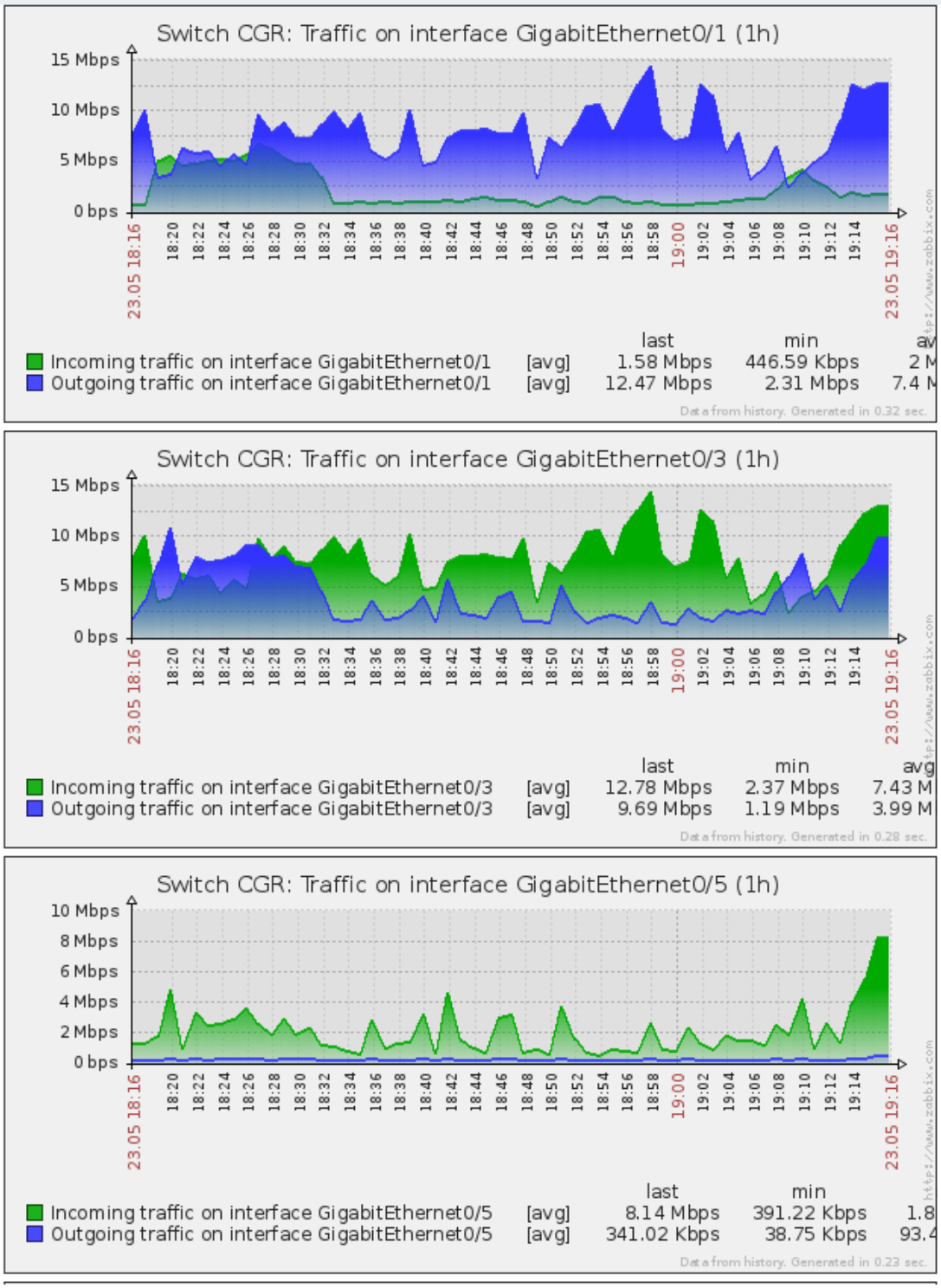

Figura 47 - Tráfego Interfaces Switch CGR
Fonte: Autoria Própria.

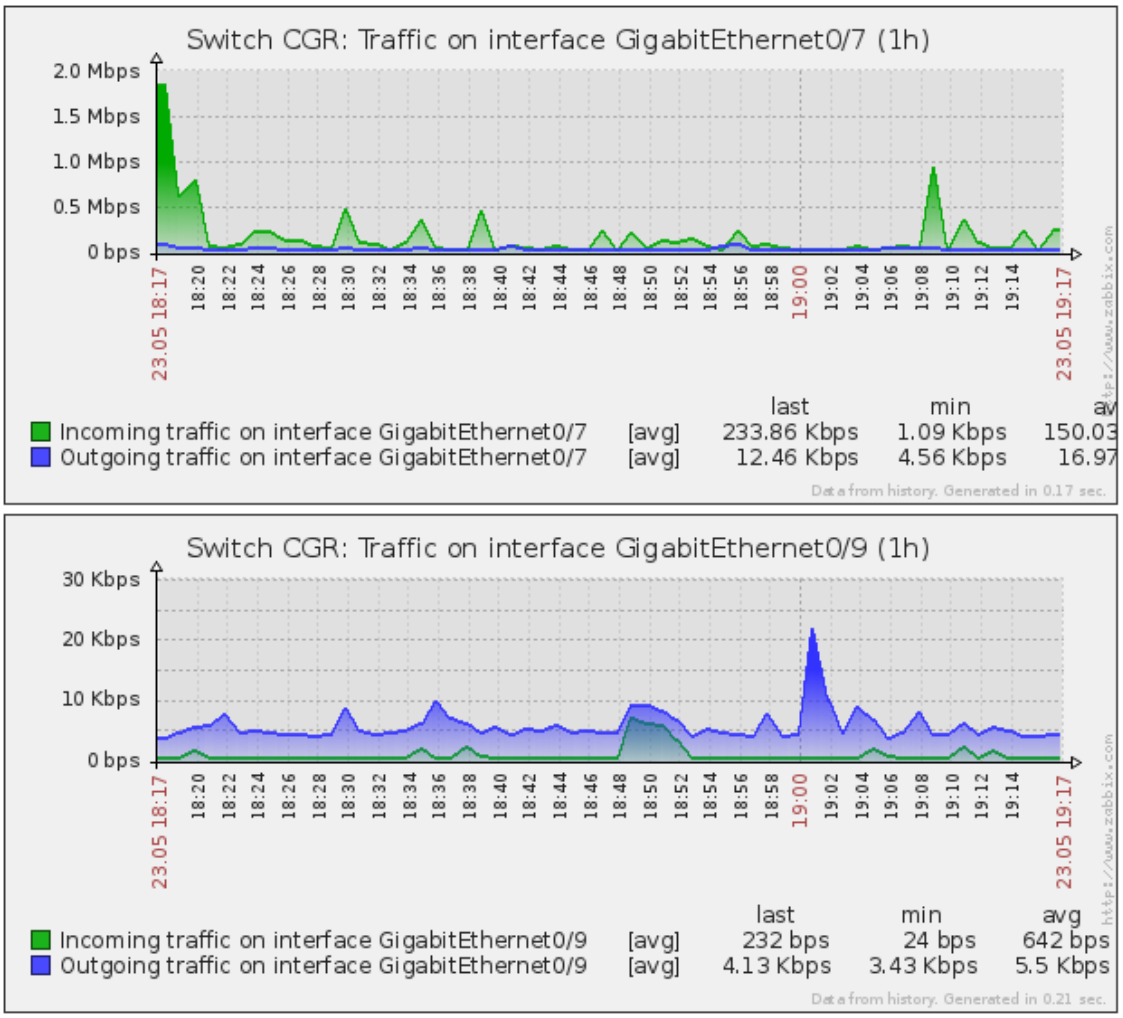

Figura 48 - Tráfego Interfaces Switch CGR
Fonte: Autoria Própria.

Para o grupo DAINF, as interfaces monitoradas segundo o tráfego foram as seguintes (Figura 49):

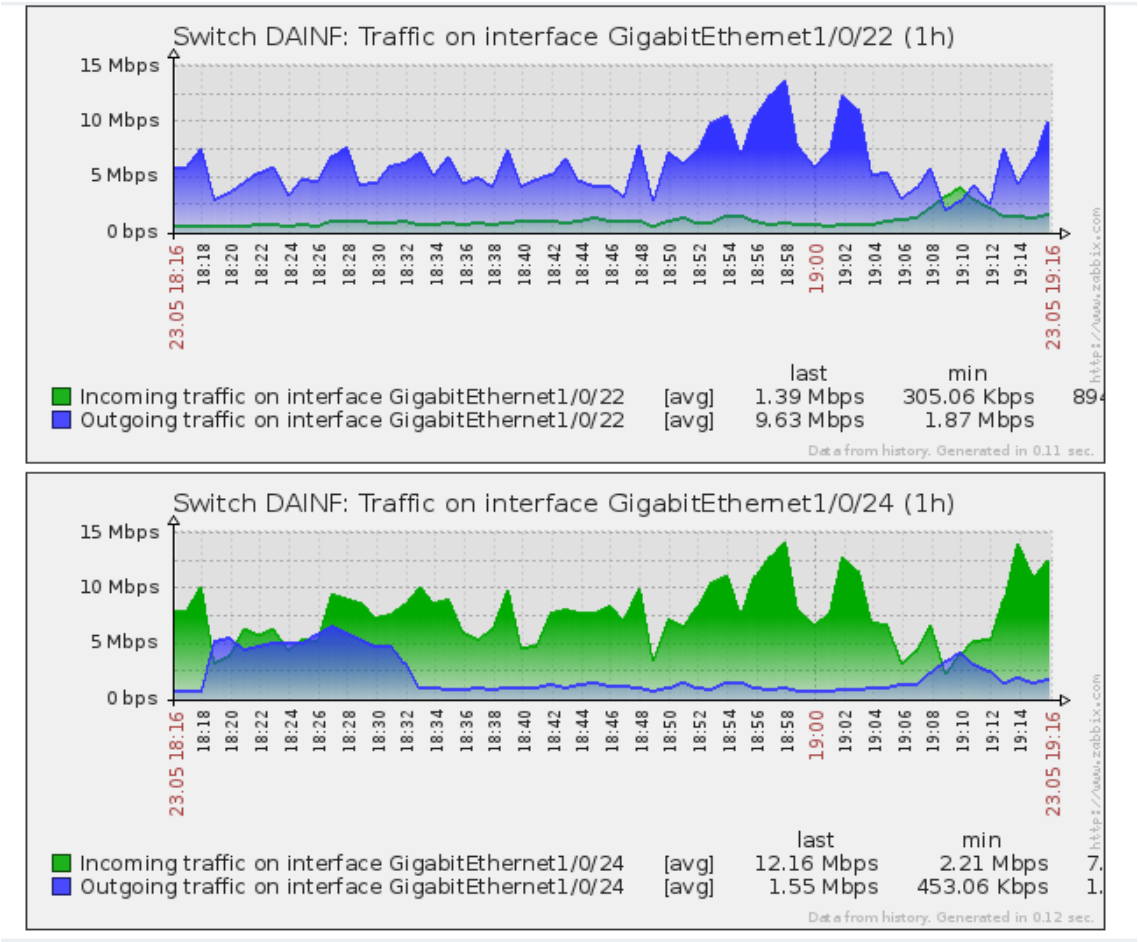

Figura 49 - Tráfego Interfaces Switch DAINF
Fonte: Autoria Própria.

Já para o grupo RLE, as interfaces mostradas (Figuras 50 e 51):

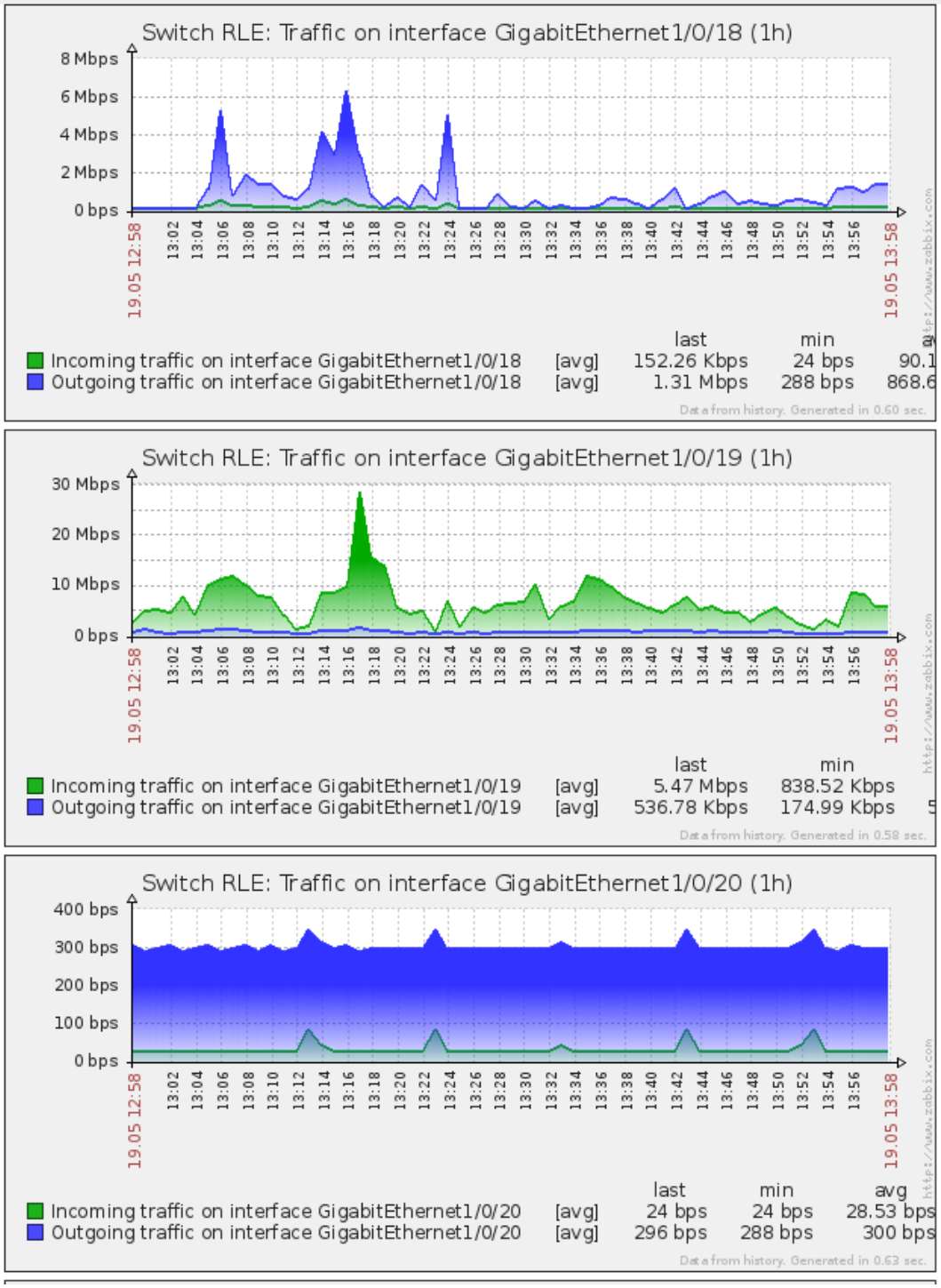

Figura 50 - Tráfego Interfaces Switch RLE
Fonte: Autoria Própria.

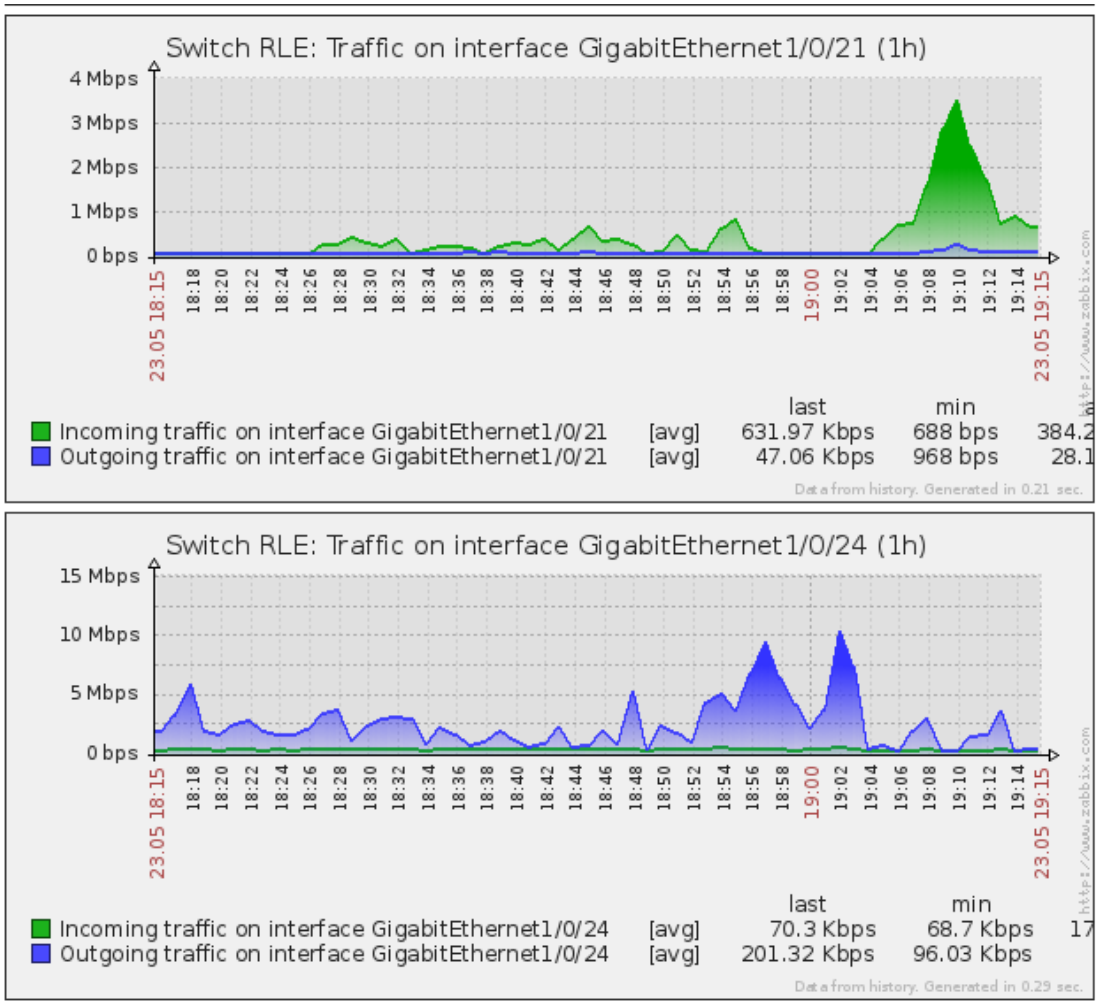

Figura 51 - Tráfego Interfaces Switch RLE
Fonte: Autoria Própria.

Mapa parcial da rede principal do RLE.

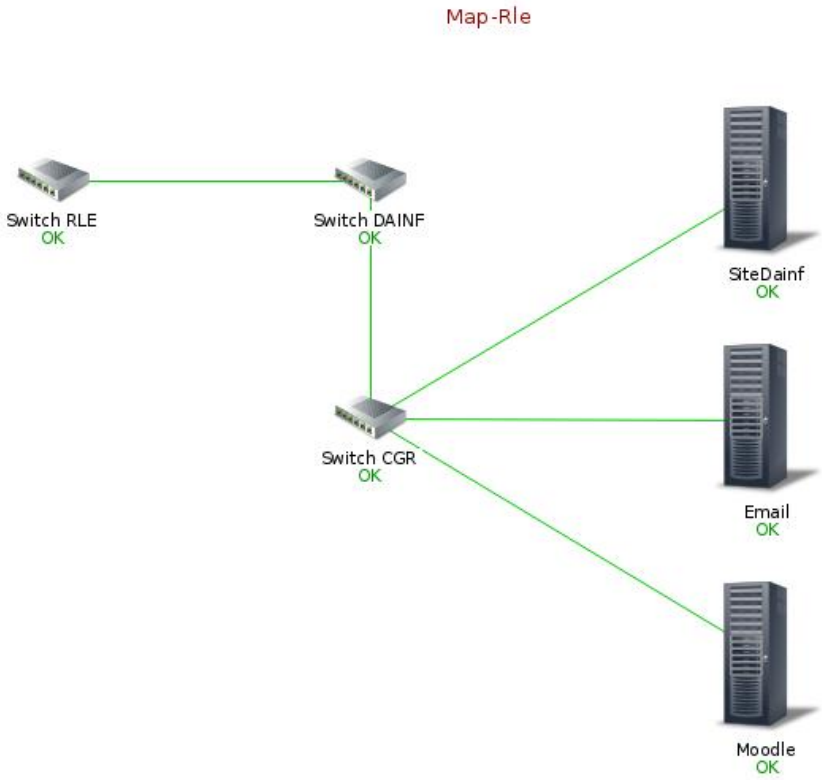

Figura 52 - Mapa Parcial Rede RLE
Fonte: Autoria Própria

Na figura a seguir, tem-se um gráfico para monitorar o uso do disco do servidor de e-mail:

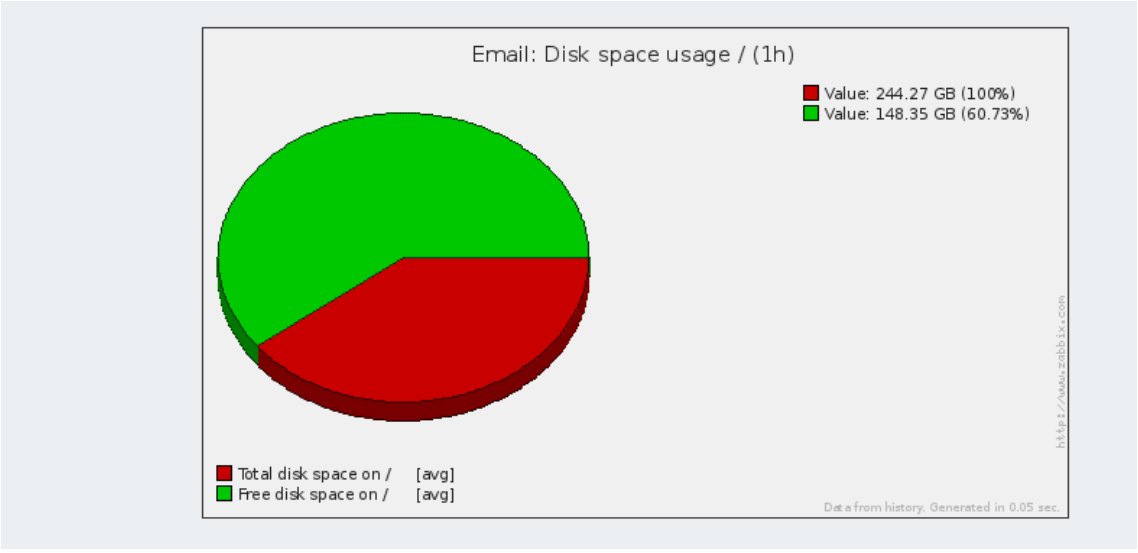

Figura 53 - Espaço HD
Fonte: Autoria Própria.

Tem-se com isto uma visão de alarmes para cada um dos *hosts*, segundo suas *triggers* destacadas anteriormente.

Na imagem abaixo, temos a tela de alarmes para o *switch* RLE.

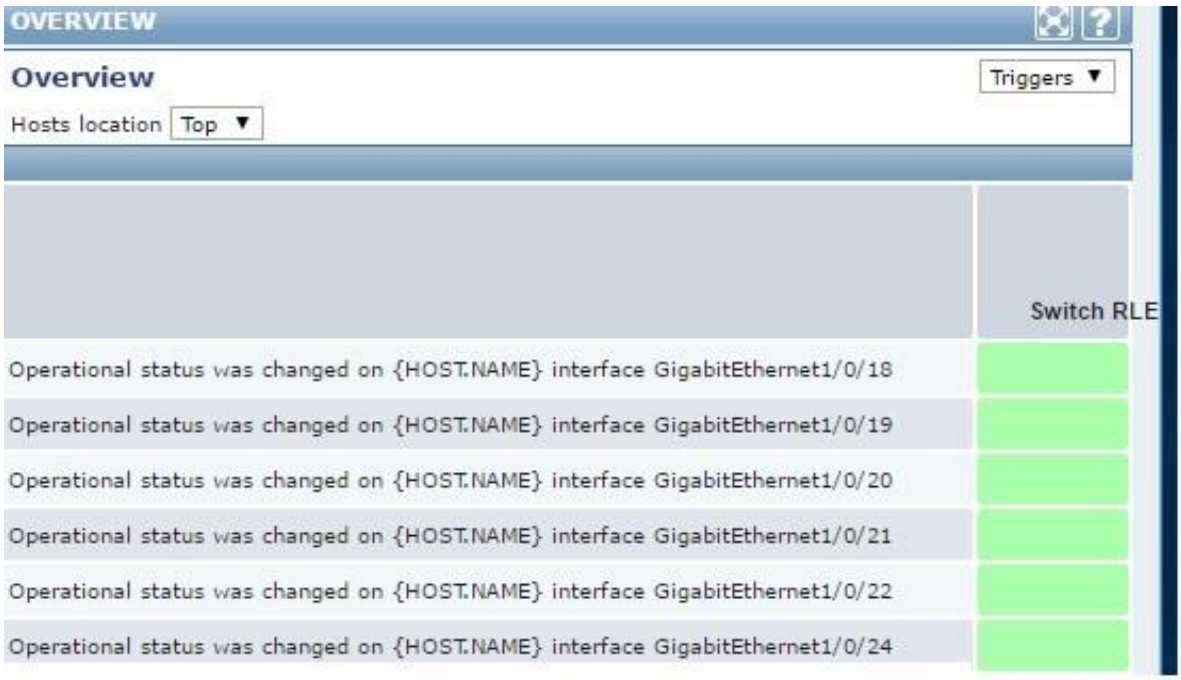

Figura 54 - Tela Triggers Switch RLE
Fonte: Autoria Própria.

A seguir, tem-se a tela de alarmes referentes ao grupo DAINF, conforme as *triggers* descritas anteriormente (Figura 55).

Figura 55 - Tela Triggers Servidor Email
Fonte: Autoria Própria.

Para grupo CGR, temos as seguintes telas de alarmes, definidos de acordo com as *triggers* definidas anteriormente (Figura 56)

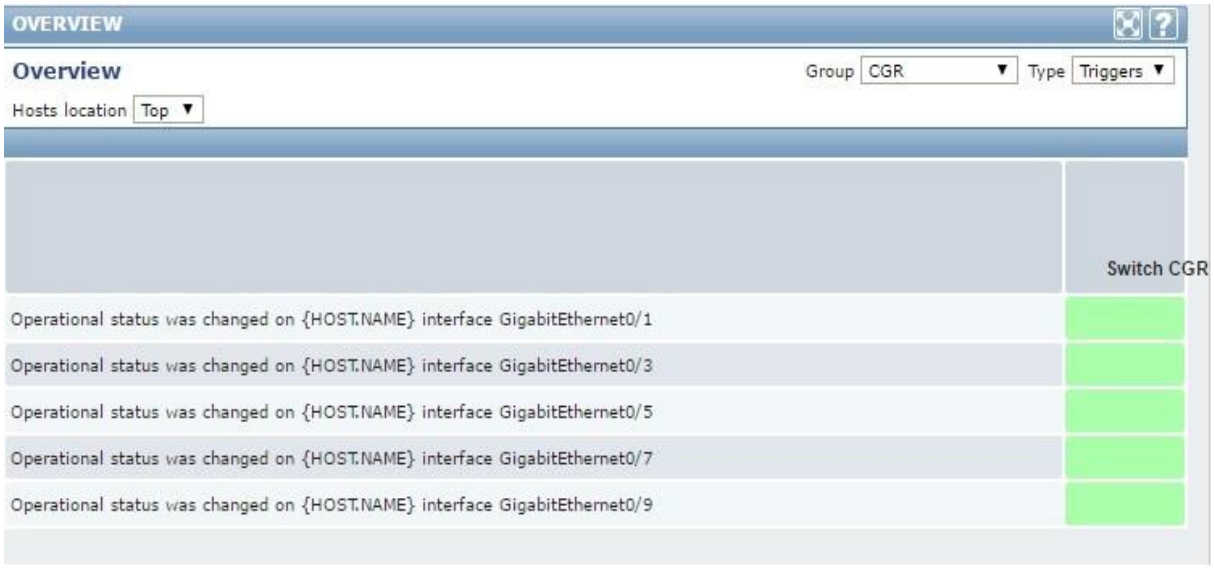

Figura -56 Tela Triggers Switch CGR
Fonte: Autoria Própria.

6 CONCLUSÃO

A monografia teve como proposta a implementação do sistema de gerenciamento de redes de computadores na Rede Local de Ensino (RLE). Com esta implementação, o administrador de redes pode verificar o status dos elementos de redes em tempo real, possibilitando com isto um melhor diagnóstico de eventuais problemas que possam afetar a rede. Na implementação, verificou-se alguns pontos de vulnerabilidade, como por exemplo o gateway default da RLE é um computador pessoal (PC) fazendo o papel de um roteador de borda. Isto não inviabiliza a infraestrutura, pois a rede está funcionando, mas não é a melhor prática pois um roteador desempenha um melhor papel no roteamento de pacotes IPs do que um computador pessoal. Para a implementação do sistema, houve a necessidade de serem estudados vários conceitos teóricos, como o gerenciamento de redes, seus principais pontos de atenção, o protocolo SNMP, ferramentas de gerência (Zabbix, Cacti), sistemas operacionais e serviços.

Dentro dessa realidade, a realização da monografia dividiu-se em três etapas: (1) descobrir os serviços; (2) mapear a rede; e, (3) implantar o monitoramento via zabbix.

Etapa 1: Para identificar quais serviços estavam sobre domínio do gerente de redes ocorreu uma breve reunião e foi identificado o seguinte *range* 200.134.10.0/24. Após isso, utilizou-se ferramenta NMAP para fazer a varredura na rede e identificar todos os IPs ativos na rede. Em nova reunião, definiu-se quais os *hosts* ele administrava chegando a lista abaixo.

- *Host* 200.134.10.1;
- *Host* 200.134.10.5;
- *Host* 200.134.10.6;
- *Host* 200.134.10.7;
- *Host* 200.134.10.27;
- *Host* 200.134.10.32;
- *Host* 200.134.10.37.

Executou-se novamente a ferramenta NMAP para identificar os serviços que foram: DNS, E-mail, DNS, DNS, Mysql, Postares, Plone. Para o *host* 200.134.10.27 não foi possível identificar o serviço. Então, realizou-se uma nova reunião para apresentar os resultados e o administrador da rede afirmou que estavam corretos os resultados dos serviços e *hosts* verificados.

Após isso iniciou-se a etapa 2: Para esta etapa necessitava definir o número de saltos que a rede tinha até a saída para CGR, então utilizou-se o *traceroute* para determinar que existem duas maquinas que comportam-se como roteadores. Em nova reunião com administrador da rede para definiu-se quais eram os elementos da camada dois que estavam conectados a eles e quais eram os elementos que estavam entre os roteadores. Com isso, chegou-se aos seguintes elementos: *switch* RLE, *switch* DAINF, *switch* CRG. Logo após, ao analisar a tabela *MAC* de cada elemento para determinar como estavam conectados, concluiu-se que o *switch* RLE concentrava todos os *switchs* das salas conectadas a ele, via as seguintes portas:

- GigabitEthernet1/0/18;
- GigabitEthernet1/0/20;
- GigabitEthernet1/0/21;
- GigabitEthernet1/0/22;
- GigabitEthernet1/0/24.

O mesmo estava conectado ao roteador slayer via GigabitEthernet1/0/19. Então o próximo elemento verificado foi o *switch*-DAINF, que foi analisado a tabela *MAC* e determinou-se que esse estava conectado ao *switch*-RLE via porta GigabitEthernet 1/0/22 e ao *switch*-CGR via porta GigabitEthernet 1/0/24. O próximo passo foi analisar o *switch*-CGR, que após a análise da tabela *MAC* determinou-se estar conectado ao *host* de saída *Klingon* identificando a rede principal para departamento de informática.

Finalmente, a etapa três iniciou-se com uma reunião para determinar quais eram os principais problemas e foram os seguintes: a internet das salas deixava de funcionar pois os alunos deligavam o switch; a ferramenta moodle e site do dainf ficavam inativos; o servidor de e-mail tinha problemas de capacidade e gerava inconsistências, nenhum computador da rede conseguia navegar na internet. Com a intenção de apresentar rápida solução para esses problemas instalou-se a ferramenta de monitoramento *zabbix* no servidor administrado pelo CRG.

Para ajudar na resolução do problema de falta de internet para todo departamento e definir se a falha estava em alguma conexão física da rede principal definida na etapa 2, aplicou-se o monitoramento por *SNMP* nas interfaces do *switch*-RLE, *switch*-DAINF e *switch*-CGR, caso ocorra algum problema irá gerar um alarme na ferramenta de monitoramento.

Já para o problema do *moodle*, utilizou-se o monitoramento do protocolo http, caso tenha alguma resposta diferente da 200 irá gerar um alarme de inconsistência na aplicação *moodle*.

Para o site do DAINF, fez-se o uso do mesmo monitoramento do *moodle*.

Para o problema do desligamento do *switch* das salas, monitora-se as interfaces do *switch*-RLE definida na etapa 2. Caso haja o desligamento dos *switchs* das salas, será gerado um alarme na ferramenta *zabbix* de porta indisponível. Não irá identificar qual *switch* foi desligado, mas sim que existe algum problema físico com a interface ligada ao *switch*-RLE.

Por último, foi instalado o agente *SNMP* no servidor de e-mail. Agente esse que fica monitorando a capacidade do disco rígido, gerando um alarme caso essa capacidade atinja 80% do total, para que o administrador tome as providencias necessárias.

Ao fazer a topologia lógica e física da rede identificou-se que fisicamente os servidores que estavam alocadas as máquinas virtuais não estavam sobre o domínio da gerência do RLE com isso gerava alguns problemas para administrador rede, pois no processo de desenvolvimento percebeu-se algumas manutenções nos mesmos e o gerente não foi avisado ou consultado se poderia ser realizado. E por fim a máquina virtual disponibilizada para execução desse projeto estava alocada nesses servidores então caso tenha alguma manutenção o zabbix ficará indisponível perdendo temporariamente a visibilidade dos alarmes da rede.

O *zabbix* proporcionou um ganho para o administrador pois agora tem uma ferramenta para ajudar na identificação dos problemas que podem vir a existir na rede.

REFERÊNCIAS

BIRKNER, M. (2003). *Projeto de Interconexão de Redes-Cisco Internetwork Design.* (P. E. Brasil, Ed.) São Paulo.

BLACK, L. T. (2008). *Comparação de Ferramentas de Gerenciamento de Redes.* Acesso em 09 de 05 de 2015, disponível em http://lume.ufrgs.br/bitstream/handle/10183/15986/000695315.pdf?sequence=1

Cacti. (2015). Acesso em 25 de 5 de 2015, disponível em http://www.cacti.net/

Cisco. (2015). Acesso em 2015 de 06 de 11, disponível em https://www.netacad.com/

DMTF. (s.d.). Acesso em 26 de 4 de 2015, disponível em http://www.dmtf.org/standards/wbem

Forouzan, B. (2007). *Data Communications and Networking* (4 ed.). New York: McGraw-hill.

Gadge, J., & Patil, A. A. (s.d.). Acesso em 28 de 4 de 2015, disponível em http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4772622

GlobalSim. (21 de 08 de 2013). *World Class Training Simulators*. Fonte: GlobalSim: http://www.globalsim.com/

IETF. (s.d.). Acesso em 15 de 04 de 2015, disponível em https://tools.ietf.org/html/rfc792

Linux.die. (s.d.). Acesso em 20 de 5 de 2015, disponível em http://linux.die.net/man/1/snmpwalk

Linux.die. (s.d.). Acesso em 16 de 5 de 2015, disponível em http://linux.die.net/man/8/traceroute

Medeiros, F. A., Alonço, A. S., Balestra, M. R., Dias, V. O., & Landerhal, M. L. (Novembro de 2008). Utilizacao de um veiculo aereo nao-tripulado em atividades de imageamento georeferenciado.

Microsoft. (s.d.). Acesso em 16 de 5 de 2015, disponível em https://support.microsoft.com/en-us/kb/217014/pt-br

Morioto, C. (2008). *Redes Guia Prático* (2 ed.). GDH Press e Sul Editores.

Nagios. (s.d.). Acesso em 25 de 5 de 2015, disponível em http://nagios-br.com/nagios-xi

Networkservice. (s.d.). Acesso em 13 de 4 de 2015, disponível em http://www.networksorcery.com/enp/rfc/rfc816.txt

Networkservice. (1981). Acesso em 14 de 4 de 2015, disponível em www.networksorcery.com/enp/rfc/rfc1349.txt

Networkservice. (1992). Acesso em 14 de 4 de 2015, disponível em http://www.networksorcery.com/enp/rfc/rfc816.txt

NMAP. (2015). Acesso em 15 de 5 de 20015, disponível em https://nmap.org/

Schönwälder, J. (2011). Schönwälder, Jürgen. *IEEE Transactoin on network and service managemen, 8*, 52-54.

Soares, L. F., & G Colcher. (1995). *Redes de Computadores das LANs,MANs e WANs as Redes ATM* (2 ed.). Rio de Janeiro: Campus.

Stallings, W. (2005). *Redes e Sistemas de Comunicação de Dados* (5 ed.). Rio de Janeiro: Elsevier.

TANENBAUM, A. (2003). *Redes de Computadores* (4 ed.). (Elsevier, Ed.) São Paulo.

Teleco. (s.d.). Acesso em 5 de 5 de 2015, disponível em http://www.teleco.com.br/pdfs/tutorialgmredes2.pdf

Teltumde, & Prakash. (2012). Management of networks using SNMP. *Iternational Journal of Engineering Innovations and research, 1*, 135.

Traceroute. (2015). Acesso em 16 de 5 de 2015, disponível em http://www.traceroute.org/

Wbemsolutions. (s.d.). Acesso em 25 de 4 de 20015, disponível em http://www.wbemsolutions.com/tutorials/snia/SMI/Technical/smis-overview.html

Williamson, A., Lombardi, D. A., Folkard, S., Stutts, J., Courtney, T. K., & Connor, J. L. (15 de Novembro de 2009). The link between fatigue and safety.

Zabbix. (08 de 06 de 2015). Fonte: http://zabbixbrasil.org/?page_id=59

Zabbix. (2016). *https://www.zabbix.com/documentation/1.8/manual/advanced_snmp.*

ANEXO

Topologia da rede do DAINF.