

**LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO DO RN
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

GEBSON VICTO ALVES FEITOZA

**ANÁLISE E IMPLANTAÇÃO DE FERRAMENTA PARA GERÊNCIA DA
REDE DE COMPUTADORES DA LIGA DE ENSINO DO RN**

*Monografia apresentada ao Curso de Graduação em
Sistemas de Informação da Faculdade Natalense
para o Desenvolvimento do Rio Grande do Norte
como requisito parcial para obtenção do grau de
Bacharel em Sistemas de Informação.*

Orientador:

Exp. Rômulo Fagundes Carneiro.

**NATAL
2011**

GEBSON VICTO ALVES FEITOZA

**ANÁLISE E IMPLANTAÇÃO DE FERRAMENTA PARA GERÊNCIA DA REDE DE
COMPUTADORES DA LIGA DE ENSINO DO RN**

Monografia apresentada ao Curso de Graduação em
Sistemas de Informação da Faculdade Natalense
para o Desenvolvimento do Rio Grande do Norte
como requisito parcial para obtenção do grau de
Bacharel em Sistemas de Informação.

Orientador:

Esp. Romulo Fagundes Cantanhede.

**NATAL
2011**

Catálogo na Publicação – Biblioteca da FARN
Setor de Processos Técnicos

Feitoza, Gebson Vito Alves

Análise e implantação de ferramenta para gerência da rede de computadores da liga de ensino do RN / Gebson Vito Alves Feitoza. - Natal, 2011.

69f.

Orientador: Romulo Fagundes Cantanhede (Esp.)

Monografia (Bacharelado em Sistemas de Informação) – Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte.

1. Gerência de Rede - Monografia. 2. Liga de Ensino RN – Monografia. 3. Projeto de Rede – Monografia. I. Título.

RN/FARN/BC

CDU 004.7

BIBLIOTECA UNI-RN

004.7 F311a - RB: 37285 (M002099)

Título: Análise e implantação de ferramenta ...

Autoria: Feitoza, Gebson Vito Alves

2011, ex. 1

AGRADECIMENTOS

Agradeço primeiramente a Deus todo poderoso criador do universo, detentor de todo o conhecimento, por ter proporcionado esse momento.

Agradecimento de forma especial e com muito carinho aos meus pais que não mediram esforços para que fosse possível manter meus estudos, mesmo a grandes distancias, também por nunca faltar apoio para a tomada de decisões em minha vida.

Gostaria de agradecer de forma muito especial a minha esposa por sempre apoiar-me e dar força em todos os momentos de dificuldades e alegrias.

Agradeço também ao Lineu Paiva administrador de rede da Liga de Ensino do RN, por ser um grande incentivador e colaborador para a realização deste projeto.

Agradeço em especial ao amigo Deógenes e sua família por acolher-me com muito carinho em suas vidas e sua residência.

Agradeço à Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte, a todos os colegas e a todos os meus professores, pelos ensinamentos ministrados em especial ao meu orientador Romulo Fagundes Cantanhede.

E a todos que, de alguma forma, contribuíram na elaboração deste trabalho.

Dedico este trabalho primeiramente aos meus pais pelo amor, carinho e a grande ajuda dada em todos os projetos de minha vida.

A minha esposa por todo amor, paciência e compreensão durante a elaboração deste trabalho.

GIBSON VICTO ALVES FEITOZA

ANÁLISE E IMPLANTAÇÃO DE FERRAMENTA PARA GERÊNCIA DA REDE DE
COMPUTADORES DA LIGA DE ENSINO DO RN

Aprovado em: 11/11/2011

BANCA EXAMINADORA


Prof. Esp. Romulo Figueiredo Carneiro
Orientador


Prof. Msc. Josemar Alves Pinheiro
Membro


Prof. Msc. Antonio Rocha
Membro

Dedico este trabalho primeiramente aos meus pais pelo amor, carinho e o grande apoio dado em todos os projetos de minha vida.

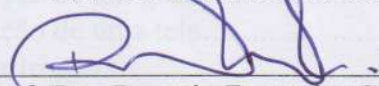
A minha esposa por todo amor, paciência e compreensão durante a realização deste sonho.

GEBSON VICTO ALVES FEITOZA

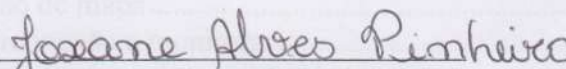
ANÁLISE E IMPLANTAÇÃO DE FERRAMENTA PARA GERÊNCIA DA REDE DE COMPUTADORES DA LIGA DE ENSINO DO RN

Aprovado em: ____/____/____

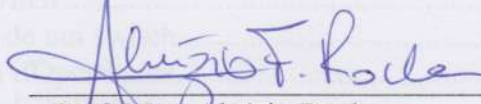
BANCA EXAMINADORA



Prof. Esp. Romulo Fagundes Cantanhede
Orientador



Prof. Msc. Joseane Alves Pinheiro
Membro



Prof. Msc. Aluizio Rocha
Membro

LISTA DE ILUSTRAÇÕES

Figura 1: Gerenciamento SNMP	21
Figura 2: Página HTML onde é possível visualizar os gráficos gerados pelo MRTG.....	22
Figura 3: Gráfico gerado pelo RRDTool exibindo informações de uma CPU.....	23
Figura 4: Principais Componentes do Zabbix	26
Figura 5: Itens de monitoração simples.....	28
Figura 6: Exemplo de gráfico – Demonstrando Entrada (IN) e Saída (OUT) de Dados.....	30
Figura 7: Exemplo de elementos que podem ser inseridos em uma tela.....	31
Figura 8: Gráficos inseridos na tela.....	32
Figura 9: Visão geral de uma tela.....	32
Figura 10: <i>Triggers</i> associados a um <i>switch</i>	33
Figura 11: Condições de limites das <i>triggers</i>	34
Figura 12: Exemplo de mapa.....	35
Figura 13: Página para realizar o upload de um <i>template</i>	36
Figura 14: Adicionar um Host no Zabbix.....	37
Figura 15: Adicionar <i>template</i> ao equipamento	37
Figura 16: Formulário para criação de um gráfico	38
Figura 17: Formulário para criação de uma tela.....	39
Figura 18: Formulário para criar <i>triggers</i>	40
Figura 19: Formulário para criar uma ação	40
Figura 20: Formulário para inserir as condições da ação	41
Figura 21: Formulário para configurar operações da ação	42
Figura 22: Formulário para criação de mapa.....	43
Figura 23: Equipamentos de interconexão a monitorar.....	44
Figura 24: Modelos de switches	45
Figura 25: Tecnologias suportadas por um <i>switch</i>	46
Figura 26: Cadastro de um <i>switch</i>	47
Figura 27: Itens monitorados de um switch.....	54
Figura 28: Expressão do item <i>ifOperStatus</i>	55
Figura 29: Expressão do item <i>ICMPPING</i>	55
Figura 30: Expressão do item <i>HTTP</i>	55
Figura 31: Tela de monitoramento de portas do switch 2924.	57
Figura 32: Gráfico exibindo erro de tráfego.....	58
Figura 33: Composição do e-mail de um alerta.....	58
Figura 34: Ações criadas.	61
Figura 35: Mapa	62

LISTA DE QUADROS

Quadro 1: Trecho de um <i>template</i>	27
Quadro 2: Trecho de um template exemplificando o uso do SNMP	29
Quadro 3: Modelos de concentradores antigos	45
Quadro 4: Modelos de concentradores novos	47
Quadro 5: Pacotes necessário a instalação	48
Quadro 6: <i>Softwares</i> auxiliares.....	48
Quadro 7: Zabbix-server.....	49
Quadro 8: Compilação do <i>software</i>	49
Quadro 9: Configurar interface <i>Web</i>	49
Quadro 10: Configurar arquivos e criar usuário para o SO.....	50
Quadro 11: Criar usuário e <i>database</i> para o Zabbix.....	50
Quadro 12: Configurar autenticação do PostgreSQL	50
Quadro 13: Configurar autenticação do PostgreSQL	51
Quadro 14: Configuração do Zabbix Server.....	51
Quadro 15: Alterar e configura parâmetros do PHP.INI	51
Quadro 16: Acessar interface do Zabbix	52
Quadro 17: Templates Utilizados	53
Quadro 18: Telas Geradas	56
Quadro 19: Script para notificação via <i>e-mail</i>	59
Quadro 20: Arquivo de configuração do ESMTP	60

RRAR:	Round Robin Archive
RDD:	Round Robin Database
RRUTool:	Round Robin Database Tool
RPC:	Request for Comments
RN:	Rio Grande do Norte
SSH:	Secure Shell
SM:	Service de Mensagens curtas
SNMP:	Simple Network Management Protocol
SO:	Sistema Operacional
SQLDB:	Sistema de Gerenciamento de Banco de Dados
TPL:	The PostgreSQL License
TI:	Tecnologia da Informação
WAN:	Wide Area Network

LISTA DE REDUÇÕES

CPU:	Central Processing Unit
CRM:	Customer Relationship Management
DHCP:	Dynamic Host Configuration Protocol
FARN:	Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte
FTP:	File Transfer Protocol
GPL:	General Public License
HTML:	HyperText Markup Language
HTTP:	Hypertext Transfer Protocol
HTTPS:	Hyper Text Transfer Protocol Secure
ICMP:	Internet Control Message Protocol
IDS:	Intrusion Detection System
IEEE:	Institute of Electrical and Electronics Engineers
IETF:	Internet Engineering Task Force
IMAP:	Internet Message Access Protocol
IP:	Internet Protocol
ISO:	International Standards Organization
MAC:	Media Access Control
MIB:	Management Information Base
MRTG:	Multi Router Traffic Grapher
NMS:	Network Management Station
OSI:	Open Systems Interconnection
PHP:	Hypertext Preprocessor
RRA:	Round Robin Archive
RRD:	Round Robin Database
RRDTool:	Round Robin Database Tool
RFC:	Request for Comments
RN:	Rio Grande do Norte
SSH:	Secure Shell
SMS:	Serviço de Mensagens curtas
SNMP:	Simple Network Management Protocol
SO:	Sistema Operacional
SGBD:	Sistema de Gerenciamento de Banco de Dados
TPL:	The PostgreSQL License
TI:	Tecnologia da Informação
WAN:	Wide Area Network

RESUMO

Este documento descreve a análise, o planejamento e a implantação de uma ferramenta para gerência da rede de computadores da Liga de Ensino do Rio Grande do Norte, considerando a utilização de diversos protocolos de gerenciamento, a escolha adequada de equipamentos e a implantação de uma arquitetura de gerenciamento. São abordados os principais aspectos de um projeto de redes, em especial a gerenciabilidade, destacando sua relação com os demais aspectos e definindo, desta forma, sua importância. A implantação de gerenciabilidade em uma rede de computadores pode ser atingida com a utilização de um conjunto de utilitários e protocolos, tais como SNMP e ICMP, controlados por uma ferramenta de gerenciamento de redes de computadores. No estudo de caso da Liga de Ensino do Rio Grande do Norte, a ferramenta escolhida foi o Zabbix, com a qual é apresentada uma proposta para melhorar o nível de gerenciamento da rede, aproveitando os benefícios do uso desse tipo de ferramenta.

Palavras Chaves: Gerenciamento. Redes. Zabbix. Ferramenta. Protocolos.

ABSTRACT

This document describes the analysis, planning and implementation of a management tool for the computer network of the Liga de Ensino do Rio Grande do Norte, considering the use of various management protocols, the choice of appropriate equipment and the implementation of a management architecture. Covers the key aspects of a network design, especially the manageability, highlighting their relationship with the other defining aspects and thus its importance. The implementation of manageability in a computer network can be achieved with the use of a range of tools and protocols such as SNMP and ICMP, controlled by a management tool of computer networks. In the case study of the Liga de Ensino do Rio Grande do Norte, the tool of choice was the Zabbix, which is presented with a proposal to improve the level of network management, taking advantage of the benefits of using this type of tool.

Keywords: Management. Networks. Zabbix. Tools. Protocols.

SUMÁRIO

1	INTRODUÇÃO	7
1.1	OBJETIVOS	7
1.1.1	Objetivos Específicos	8
2	ASPECTOS DE UM PROJETO DE REDE	8
2.1	DISPONIBILIDADE	9
2.2	VIABILIDADE	10
2.3	DESEMPENHO	10
2.4	SEGURANÇA	11
2.5	GERENCIABILIDADE	12
2.6	ESCALABILIDADE	12
2.7	USABILIDADE	13
2.8	ADAPTABILIDADE	13
2.8.1	Relação entre segurança e gerenciabilidade	13
2.8.2	Relação entre disponibilidade e gerenciabilidade	14
2.8.3	Conflito entre desempenho e gerenciabilidade	15
2.8.4	Conflito entre viabilidade e gerenciabilidade.	15
3	FERRAMENTAS E PROTOCOLOS DE GERÊNCIA	16
3.1	LOGS	16
3.2	ICMP	16
3.3	ACESSO: TELNET E SSH	17
3.4	SNMP	17
3.4.1	Elementos do SNMP	19
3.4.2	Operações do SNMP	20
3.5	MRTG	21
3.6	RRD E RRDTOOL	22
4	A FERRAMENTA ZABBIX	24
4.1	ARQUITETURA	25
4.2	PROTOCOLOS, FERRAMENTAS E TERMINOLOGIA	27
4.2.1	Template	27

4.2.2	ICMP	28
4.2.3	SNMP	29
4.2.4	Gráficos	30
4.2.5	Triggers (Gatilhos)	33
4.2.6	Ações	34
4.2.7	Mapas	34
4.3	EXEMPLO DE MONITORAMENTO DE UM EQUIPAMENTO DE REDE	35
4.3.1	Criar ou Importar Templates	36
4.3.2	Inserir Hosts	36
4.3.3	Criar Gráficos	38
4.3.4	Criar Telas	38
4.3.5	Criar Triggers	39
4.3.6	Criar Ações	40
4.3.7	Criar Mapas	42
5	ESTUDO DE CASO	44
5.1	ANÁLISE DA GERENCIABILIDADE DA REDE	44
5.2	IMPLANTAÇÃO DO ZABBIX	48
5.2.1	Instalação	48
5.2.1.1	Apache2, PHP5 e PostgreSQL	48
5.2.1.2	Zabbix	49
5.2.2	Templates utilizados	52
5.2.3	Itens monitorados	53
5.2.4	Principais Triggers	54
5.2.4.1	Falha de uma porta de backbone em um switch	54
5.2.4.2	Perda de conectividade de um equipamento ativo de rede	55
5.2.4.3	Parada de um servidor <i>Web</i>	55
5.2.5	Telas geradas	56
5.2.6	Mecanismos de Alertas e Ações Criadas	58
5.2.7	Mapa	61
6	CONCLUSÃO	63
	ANEXO A - CONFIGURAÇÃO DO SERVIDOR ZABBIX	67
	ANEXO B – COMPOSIÇÃO DE UM TEMPLATE ASSOCIADO A UM SWITCH	69

1 INTRODUÇÃO

Hoje em dia nota-se um grande desenvolvimento das tecnologias de redes de computadores, tanto em equipamentos quanto em softwares. Com esse desenvolvimento, também surgem novas ameaças ao bom funcionamento de toda infraestrutura de TI, tais como novas versões de softwares maliciosos, *hackers*, vírus, *spywares*, *bugs*, entre outros, o que gera uma das grandes preocupações dos administradores de redes – a segurança da rede.

O gerenciamento dos recursos de rede é essencial para atingir os objetivos de segurança exigidos pelas corporações nas redes de computadores modernas. Através de gerenciamento, é possível identificar fontes de problemas e até mesmo perceber a iminência do acontecimento desses problemas, de forma a conseguir evita-los ou minimizar seus impactos.

Analisando esse cenário, observa-se que a segurança obtida pelo bom gerenciamento ainda não é uma das preocupações principais das empresas, embora se perceba que as mesmas estão buscando soluções que proporcionem um maior nível de desempenho e disponibilidade de suas redes, como forma de elevar a qualidade dos serviços oferecidos e a produtividade. Para atingir desempenho e disponibilidade, são perceptíveis atualmente os maiores investimentos em gerência de redes, mas, em grande parcela dos casos, de forma pouco consciente e com pouca ênfase.

Assim uma justificativa para a realização deste projeto é proporcionar o pleno funcionamento dos ativos da rede de computadores, através do monitoramento por uma ferramenta, de gerenciamento de redes centralizada, de fácil configuração e instalação no parque tecnológico da Liga de Ensino do RN de forma a garantir o máximo de disponibilidade da rede de computadores.

1.1 OBJETIVOS

O presente trabalho tem como objetivo analisar e implantar na Liga de Ensino do RN, uma ferramenta de gerenciamento de redes de computadores denominada Zabbix, destacando a análise do impacto na gerenciabilidade em toda a infraestrutura, bem como destacar os principais benefícios na implantação do Zabbix.

1.1.1 Objetivos Específicos

O trabalho tem como objetivos específicos os seguintes itens:

- Analisar se o software corresponde aos seus principais objetivos que o mesmo dispõe.
- Apresentar as dificuldades encontradas durante a implantação e configuração da ferramenta.
- Análise do impacto no desempenho da rede.

2 ASPECTOS DE UM PROJETO DE REDE

Como afirma Oppenheimer (1999 p.5), “o bom projeto de rede deve reconhecer que os requisitos de um cliente encarnam muitas metas do negócio e técnicas, inclusive requisitos de disponibilidade, facilidade de escalonamento, viabilidade, segurança e gerenciamento”. O reconhecimento de que a infraestrutura de rede não é um objetivo-fim, mas um meio, que tem por objetivo fornecer suporte a um negócio ou a um processo produtivo, torna o projeto de redes uma atividade que parte de requisitos de negócio. Oppenheimer relaciona diversas metas de negócio comuns em um projeto de rede. Dentre essas metas, é possível destacar:

- Ampliar a receita e o lucro;
- Melhorar comunicações na empresa;
- Aumentar produtividade;
- Expandir-se para outros mercados;
- Modernizar tecnologias desatualizadas;
- Melhorar segurança em aplicativos e dados de missão crítica;

O projeto de rede precisa captar, com o cliente, requisitos práticos necessários para auxiliar a organização a atingir esses objetivos com o uso de tecnologia, e espalhar esses requisitos em métricas técnicas de projeto de rede, de forma a atendê-los satisfatoriamente.

Dentre as métricas técnicas, Oppenheimer (1999) recomenda a análise de: disponibilidade; escalabilidade; desempenho; segurança; gerenciabilidade; usabilidade; adaptabilidade e viabilidade.

Natural, no posicionamento do cliente, é se encontrar uma forte inclinação às métricas de desempenho e baixo custo. Em uma visão inicial comum do cliente, quanto mais rápida e

eficiente for sua rede, com um custo razoável, melhor para seu negócio. Contudo, uma análise um pouco mais profunda, indica que desempenho não é a métrica mais importante na maioria dos negócios. Em muitas situações, o desempenho suficiente é atendido com pouca aplicação de recursos, que podem ser utilizados em outros aspectos do projeto da rede.

Do ponto de vista prático e aplicado à tecnologia, as metas de negócio mais comuns, dentre as que implicam em investimentos em infraestrutura de TI, podem ser exemplificadas:

- Sistemas de gestão;
- CRM (Customer Relationship Management);
- Correio eletrônico;
- Comunicação interna;
- *Marketing* institucional;
- e-Business;

As metas acima são críticas e impõem grandes investimentos em infraestrutura. Outras metas de menor importância também são comuns:

- Compartilhamento de arquivos;
- Acesso a terminal;
- Acesso à Internet;

Ao observar as metas críticas, é possível concluir que, em qualquer das metas observadas, disponibilidade e segurança são vitais para o negócio, e assim para o cliente. A seguir serão avaliados cada um dos aspectos técnicos de um projeto de rede, e sem seguida são destacadas as relações entre disponibilidade, segurança e gerenciabilidade.

2.1 DISPONIBILIDADE

De acordo com Oppenheimer, (1999), refere-se à disponibilidade o percentual de tempo em que a rede ou serviço está disponível para seus usuários. Partindo do princípio de que falhas, em hardware ou em cabeamento, são inevitáveis, é possível concluir que um dos requisitos de disponibilidade é a redundância, ou seja, a duplicação de equipamentos e caminhos na rede. Um segundo requisito de disponibilidade é a capacidade de recuperação das falhas.

A disponibilidade está desta forma, vinculada a diversos fatores, como taxas de erros, estabilidade, período de tempo entre falhas e período de tempo de reparação ou recuperação

de falhas. Já de acordo com Souza (2005, p.10), “A disponibilidade é o primeiro passo para medir o desempenho de uma rede, pois determina se a rede está ativa, ou seja, se o tráfego está atravessando a rede.”. Analisando-se esses conceitos percebe-se a importância de garantir o máximo de disponibilidade a rede de computadores.

2.2 VIABILIDADE

A viabilidade é uma meta importante para projetos de redes, pois trata dos custos financeiros, tais como gastos com equipamentos e custos com a operação da rede em si. Nota-se que empresas muitas vezes não estão dispostas a investir o quanto deveriam com certos aspectos da infraestrutura de rede. Mas, para obter uma rede segura, com disponibilidade e desempenho satisfatórios, necessita-se de equipamentos com muitas tecnologias agregadas e componentes redundantes, o que aumenta o custo de implantação da rede.

Ainda há outro aspecto a ser analisado. Conforme Oppenheimer (1999, p.44), “o segundo aspecto mais caro do funcionamento de uma rede, em seguida ao custo dos circuitos WAN's, é o custo de contratar, treinar, e manter o pessoal que irá operar e administrar a rede”. Essa informação é interessante e ilustra o peso do investimento em pessoal para operar e administrar a rede.

A viabilidade da implantação de uma rede passa, então, pelo estudo de quantificação das metas (desempenho, disponibilidade, etc.) e dos entraves (custos financeiros em geral, impedimentos físicos, etc.), e do estabelecimento de relações entre metas e entraves, de forma a atingir um equilíbrio de interesses.

2.3 DESEMPENHO

O desempenho em uma rede pode ser avaliado através de muitas métricas, dentre as quais podem ser destacadas: capacidade, utilização, vazão, carga oferecida, precisão, eficiência, retardo, variação do retardo e tempo de resposta. Como cita Oppenheimer (1999), eficiência é uma medida do quanto uma operação é efetiva em comparação com o custo em esforço, energia, tempo ou dinheiro. Assim, a eficiência acaba se tornando métrica de desempenho para um projeto de rede. Destacam-se também outras métricas diversas de desempenho de rede, elencadas a seguir:

- Utilização, que é a porcentagem da capacidade total disponível em uso;

- Vazão, ou seja, a quantidade de dados transferidos (taxa bit/ unidade de tempo), entre dois nós da rede, sendo um emissor e outro receptor;
- Carga oferecida é, de acordo com Oppenheimer (1999), a soma de todos os dados ou tráfego que todos os nós da rede estão prontos para enviar em um dado momento;
- Retardo, ou atraso, pode ser definido como o tempo médio entre o momento em que um quadro está pronto para ser transmitido a partir de um nó e o momento da entrega em outro lugar na rede;
- Tempo de resposta é definido como o intervalo de tempo entre o pedido de algum serviço de rede e uma resposta ao pedido.

De acordo com Kurose e Ross (2006), a meta do gerenciamento de desempenho é quantificar, medir, informar, analisar e controlar o desempenho de diferentes componentes da rede. Assim, a análise do desempenho deve ser feita, em cada métrica citada anteriormente, para cada componente da rede, de forma a atingir um bom gerenciamento de desempenho.

2.4 SEGURANÇA

Conforme Tanenbaum (2003), os riscos de segurança em redes podem ser divididos em quatro áreas intimamente relacionadas: confidencialidade, autenticidade, não repúdio e integridade. Confidencialidade é relacionada a manter a informação longe de pessoas ou sistemas não autorizados. Autenticidade e autenticação possuem relação com a verificação de se a informação veio da pessoa que diz tê-la enviado. O não repúdio é alcançado quando a informação é validada independente de qualquer contestação, inclusive do seu próprio remetente. Integridade tem relação com a confirmação de originalidade da informação, não tendo esta sido alterada entre sua produção e seu recebimento. Além dessas áreas, tradicionalmente se inclui a disponibilidade como objeto de estudo da segurança.

Obviamente, adicionar métodos de segurança que tratem de todas as áreas citadas tem um impacto sobre o custo de implantação e operação em uma rede. Através de planejamento, é possível dimensionar os riscos e requisitos em cada uma das áreas da segurança, e com isso obter melhores informações gerenciais de como uma implementação de segurança pode impactar o projeto de rede.

Outro exemplo de impacto pode ser extraído do uso de segurança. Como afirma Oppenheimer (1999, p.38) “Normas estritas de segurança também podem afetar a

produtividade de usuários, especialmente se alguma facilidade de uso tiver de ser sacrificada para proteger recursos e dados”. Nota-se que a segurança sempre envolve perdas e ganhos. Por exemplo, perde-se facilidade de uso para obter mais mecanismos de autenticação na rede.

2.5 GERENCIABILIDADE

Conforme Flausino (2004, p.60), a gerenciabilidade “visa atender às necessidades de se projetar uma rede com o intuito de facilitar sua monitoração e gerenciamento, assegurando com isto sua estabilidade permanente de operação”.

O gerenciamento de redes pode ser entendido como o processo de controlar uma rede de computadores de tal modo que seja possível maximizar sua eficiência e produtividade. Tal processo compreende um conjunto de funções integradas que podem estar em uma máquina ou espalhados por milhares de quilômetros, em diferentes organizações e residindo em máquinas distintas. (MENEZES E SILVA, 1998, p.1).

É possível, diante do exposto até aqui, ampliar essa definição para adicionar, como metas de gerência, a segurança e a disponibilidade dos recursos, com algum impacto na viabilidade e no desempenho.

2.6 ESCALABILIDADE

A escalabilidade ou a facilidade de escalonamento, em um projeto de rede tem como objetivo dimensionar o nível de crescimento que um projeto de rede pode suportar, sendo essa uma das metas fundamentais para determinados projetos de redes.

Conforme Oppenheimer (1999, p.18) “As grandes empresas estão acrescentando usuário, aplicativos, instalações adicionais e conexões de rede externas a uma taxa rápida”. Analisando-se essa afirmação nota-se que um bom projeto de rede desenvolvido para uma empresa seja ela qual for à dimensão, deve ser capaz de se adaptar aos possíveis aumentos na utilização e na infraestrutura da rede.

2.7 USABILIDADE

Usabilidade ou facilidade de uso pode ser entendida como a facilidade com a qual usuários da rede podem ter acesso a rede e demais serviços, segundo Oppenheimer (1999), a facilidade de gerenciamento busca tornar mais fácil todo o trabalho dos gerentes de redes, já a facilidade de uso, tem como objetivo facilitar o trabalho dos usuários final.

Nota-se que ao se aprofundar na usabilidade o administrador de rede precisar avaliar, alguns impactos, como por exemplo, o impacto na implementação de uma política de segurança, visto que normas muito restritas podem ter efeitos negativos sobre a usabilidade. Conforme Oppenheimer (1999, p.41), “Você pode maximizar a facilidade de uso distribuindo esquemas de nomenclatura amigável de hosts e métodos de configurações fáceis que fazem uso de protocolos dinâmicos como o DHCP (*Dynamic Host Configuration Protocol*)”.

2.8 ADAPTABILIDADE

Muitos administradores de rede não se atentam as dificuldades ao implantar uma tecnologia sem antes analisar todos os seus impactos ao ambiente computacional, observa-se que um projeto de rede deve-se adaptar-se a inúmeras mudanças que venham a ocorrer como:

- Mudança de tecnologia.
- Mudanças de Protocolos.
- Mudança na prática do negócio.
- Mudança de legislação e metas fiscais.

A facilidade de adaptação conforme Oppenheimer (1999), diz, que um projeto de rede flexível é capaz de se adaptar a padrões de tráfego variáveis e também a requisitos de qualidade de serviço (QoS). Outro importante aspecto da adaptabilidade é a rapidez, com que dispositivos que fazem parte da rede e interligam os nós, conseguem se adaptar a problemas e atualizações.

2.8.1 Relação entre segurança e gerenciabilidade

Como foi visto, a segurança é um dos pilares para um bom projeto de redes. Um primeiro aspecto da relação entre segurança e gerenciabilidade pode ser observado na necessidade de gerenciamento dos próprios métodos de segurança, situados pela arquitetura

de gerenciamento OSI (*Open Systems Interconnection*) no “gerenciamento de segurança”. Conforme Oppenheimer (1999), o gerenciamento da segurança garante a um gerente de redes, por exemplo, manter e distribuir senhas e outras informações com relação à autenticação e autorização, como também atendimento a monitoração e testes das normas de segurança e proteção aplicadas na rede.

Um segundo aspecto da relação está no fato de que, para atingir algumas metas de segurança, é necessário o uso de gerenciamento. Por exemplo, a auditoria, etapa de fundamental importância para metas de segurança, é realizada normalmente através da análise de *logs*, que são registros de ocorrências e comportamentos dos sistemas, e consistem em um método de gerenciamento. A segurança contra falhas passa pela utilização de protocolos de gerenciamento que monitorem o comportamento de um sistema ou equipamento. O comportamento anômalo de um dispositivo ou enlace de rede pode ser verificado com a observação do tráfego, em uma análise quantitativa e/ou qualitativa.

A detecção e a prevenção a ataques também é realizada com métodos de gerenciamento. Um exemplo é a utilização de um IDS (Sistema de detecção de intrusão), que conforme Barbosa (2000, p.5) é um “Sistema composto de hardware e software que trabalham juntos para identificar eventos inesperados que podem indicar que um ataque irá acontecer, está acontecendo ou aconteceu”. Ou seja, uma ferramenta utilizada para detectar e alertar o administrador, sobre ataques, acessos e outras atividades maliciosas acontecendo na rede.

2.8.2 Relação entre disponibilidade e gerenciabilidade

Conforme discutido antes, para uma empresa que busca ter seus serviços de rede sempre disponível, a mesma deve investir na redundância de equipamentos. Mas essa técnica não é a única forma de conseguir boa disponibilidade. Observa-se, então, a relação entre gerenciabilidade e disponibilidade.

Disponibilidade refere-se ao percentual do tempo em que a rede está disponível. Por exemplo, se uma rede foi projetada para ficar 24 horas no ar, e ela parar por pelo menos uma hora, a disponibilidade neste caso já será bastante afetada. A gerência dos recursos permite a visualização do status dos serviços e equipamentos de rede, e assim detectar, isolar, diagnosticar e corrigir problemas. Conforme cita Oppenheimer (1999 p.246), “existe uma variedade de ferramentas para atender a esses requisitos de gerenciamento de falhas, inclusive ferramentas de monitoramento que alertam os gerentes para problemas, analisadores de

protocolos para resolução de falhas e software de *help desk* para documentar problemas e alertar os usuários sobre problemas”.

2.8.3 Conflito entre desempenho e gerenciabilidade

O gerenciamento de uma rede implica na utilização de protocolos e ferramentas para gerência. Esses protocolos e ferramentas, naturalmente, possuem sua parcela de utilização da rede. É possível afirmar que o gerenciamento toma para si parte do desempenho da rede.

Observa-se uma perda de desempenho quando os ativos da rede como *switches*, servidores, entre outros, estão sendo monitorados, enviando informações de monitoramento para um gerente. Isso acontece porque, ao ativar um protocolo de gerência, tal como o SNMP, o equipamento de rede ocupa parte de seu processamento com as atividades do protocolo. Além disso, mensagens de controle ou de dados de gerenciamento serão trocadas entre os dispositivos de rede e os dispositivos de gerenciamento, o que irá gerar tráfego e ocupar parte da banda de transmissão dos enlaces de rede.

2.8.4 Conflito entre viabilidade e gerenciabilidade.

Oppenheimer (1999) recomenda que em geral, você deve colher informações suficientes para se sentir a vontade e tiver certeza de que compreende a extensão e a viabilidade do projeto de desenvolvimento de rede. Analisando essa afirmação, é possível observar que os quesitos de gerenciabilidade, em um projeto de rede, não são normalmente avaliados do ponto de vista de viabilidade. Geralmente, projetos de rede simples levam em conta apenas critérios técnicos de desempenho, robustez, entre outros aspectos, mas dificilmente o aspecto de gerenciabilidade.

É preciso observar que para se realizar gerência de recursos de rede, são necessários investimentos em bons equipamentos, que suportem as tecnologias e protocolos específicos da arquitetura de gerenciamento proposta, pessoas capacitadas, entre outros elementos, visando uma rede de alta qualidade segura e disponível. Esses elementos, no entanto, aumentam o custo de implantação da rede.

3 FERRAMENTAS E PROTOCOLOS DE GERÊNCIA

A gerência de redes está intimamente ligada à qualidade dos serviços oferecidos na rede, visto que a gerência tem como função monitorar e controlar os elementos da rede sejam eles físicos ou lógicos. Analisando a dificuldade de controlar um grande número de equipamentos interligados, sujeitos a vários problemas, surgem os sistemas de gerenciamento de redes e ferramentas, para auxiliar administradores de rede, nos desafios de manter a rede sempre disponível e segura.

De acordo com Lopes, Sauv  e Nicolletti (2003, p.4), um sistema de ger ncia de rede “pode ser definido como uma cole  o de ferramentas integradas para a monitora  o e controle da rede”. No qual s o oferecidos v rios recursos usados para executar quase todas as tarefas da ger ncia da rede.

Muitos protocolos, ferramentas e t cnicas s o utilizados nos equipamentos e sistemas de ger ncia. A seguir   feita uma rela  o dos principais, com uma breve descri  o.

3.1 LOGS

Os registros de atividades do sistema s o chamados de *logs*. Quando os sistemas computacionais sofrem algum tipo de ataque, os intrusos sempre deixam pistas, registradas de alguma forma no pr prio sistema alvo ou em sistemas auxiliares.

Como afirma Solha (1999) “os *logs* podem desempenhar um papel preventivo, na medida em que podem registrar as eventuais tentativas de ataque”, pois os mesmos armazenam tudo o que acontece no sistema como ( ltimas entradas, comandos digitados etc.).

Mais do que registros de seguran a, os *logs* s o capazes de registrar todas as atividades de um dado servi o ou sistema, e assim, formar um reposit rio de informa  es uteis ao gerenciamento.

3.2 ICMP

Conforme   especificado no RFC 792, o protocolo de mensagens de controle da Internet – ICMP (*Internet Control Message Protocol*) – faz parte do protocolo IP (*Internet Protocol*).   utilizado por roteadores que est o conectados em uma rede, onde os mesmos usam o protocolo para comunicar informa  o sobre a pr pria camada de redes.

De acordo com Kurose e Ross (2006), a utilização mais comum para o protocolo ICMP, é informar a ocorrência de erros. Por exemplo, quando um datagrama IP não chega ao seu destino, o protocolo ICMP informa ao remetente sobre o problema. Uma mensagem ICMP é composta por um campo de tipo e outro campo de código, e transporta o conteúdo do cabeçalho e dos primeiros 8 *bytes* do datagrama IP que motivou a criação da mensagem ICMP.

3.3 ACESSO: TELNET E SSH

“Um dos mais importantes problemas que cientistas da computação precisaram resolver foi como possibilitar alguém que estivesse operando um computador a acessar e usar outro, como se ele ou ela estivesse conectada localmente a esta outra máquina.”. Cita França (2005, p.32). Para resolver tal problema, surgiram protocolos como o Telnet e o SSH (*Secure Shell*).

O Telnet é considerado um protocolo de comunicação de dados, ou seja, um conjunto de regras para comunicação entre computadores. De acordo com França (2005, p.33), “A comunicação via Telnet funciona entre diferentes tipos de computadores e sistemas operacionais porque o protocolo é o mesmo para todos os sistemas, Telnet é frequentemente usado para *login* remoto, onde se acessa um computador a partir de outro”.

O SSH, o qual também é considerado um programa e um protocolo de rede, que permite a conexão com outros computadores, tornando possível executar comandos de uma unidade remota, têm suas funcionalidades parecidas com o Telnet. O seu diferencial é possuir suporte a criptografia de dados, evitando que dados importantes como *logins*, senhas, e outros dados sejam roubados.

O SSH apresenta muitas outras vantagens em relação ao Telnet, além da segurança dos dados, o mesmo possui várias implementações como, *ssh*, *openssh*, *ossh* entre outras. E seu suporte técnico e encontrado em muitas listas de discussão, além de instalação e configuração simplificada.

3.4 SNMP

Considerando as necessidades de gerenciamento dos recursos de redes de computadores, além da necessidade de simplificação do processo de gerência, surge o protocolo SNMP (*Simple Network Management Protocol*), que tem como finalidade

proporcionar o gerenciamento dos diversos dispositivos conectados a uma rede de forma simples e prática.

Visto que uma rede de computadores complexa é composta por roteadores, switches e servidores além de diversos outros equipamentos. Nota-se que gerenciar e garantir pleno funcionamento de todos os equipamentos não é uma tarefa das mais simples. Conforme Mauro e Schmidt (2003, p.1), "O SNMP foi lançado em 1988, para atender a necessidade cada vez maior de um padrão para gerenciar dispositivos de IP". Ainda segundo Mauro e Schmidt (2003), o SNMP oferece aos usuários um conjunto de operações que permitem o gerenciamento remoto de dispositivos.

Devido a seu grande sucesso, o protocolo SNMP é usado não somente em computadores ou roteadores, mas praticamente em quase todos os equipamentos. O SNMP pode ser utilizado para várias funções, como: auxiliar o trabalho do administrador da rede informando e corrigindo erros ou problemas que são detectados na rede, entre outras como visualização de tráfego e estatísticas que passam na rede ajudando na tomada de decisão do administrador para onde agir e como agir em determinado momento.

Como cita Mauro e Schmidt (2003), A IETF (*Internet Engineering Task Force*) é responsável pela definição de protocolos, padrão que controlam o tráfego na Internet, incluindo também o SNMP. A IETF também publica *Requests for Comments* (RFCs) que são especificações e orientações para os mais variados protocolos existentes.

O SNMP definido na RFC 1157, que engloba três versões, SNMPv1, esta versão foi amplamente implementada em produtos comerciais tão logo foi aceita como padrão para gerenciamento de redes. Em seguida surge a versão SNMPv2, principalmente devido a deficiência de segurança SNMPv1. No ano de 1999 foi publicada a última versão, SNMPv3. Essa versão implementa novos requisitos de segurança, não encontrados em versões anteriores, além de adicionar novas funções, tais como gerenciamento distribuído, formas de tratamento e transporte de dados. Nota-se, porém, que o SNMPv3 ainda continua em desuso. Stange (2008, p.23), afirma que "Isto vem ocorrendo pela complexidade de implementação dessa versão, causada principalmente pelas exigências físicas do equipamento a ser gerenciado.". O suporte a SNMPv3, portanto, é mais caro e tem maior impacto na viabilidade de um projeto de rede.

3.4.1 Elementos do SNMP

Conforme Mello (2000, p16), “O SNMP possui uma característica cliente/servidor, onde o cliente seria o gerenciador da rede e o servidor o agente SNMP, e a base que modela esse agente denomina-se MIB (*Management Information Base*)”. De forma resumida, os agentes estão implantados, geralmente em *software* básico, nos equipamentos ativos de rede. O gerente é um sistema que colhe informações dos agentes, as transformando em informações úteis ao gerenciamento da rede.

Como cita COSTA (2008, p.22) “Um dispositivo gerenciado é um nó de rede que possui um agente SNMP instalado e se encontra em uma rede gerenciada.”. A seguir são detalhados os componentes de uma arquitetura SNMP.

- Agente: equipamentos como *hosts*, *switches*, roteadores entre outros, para que possam ser gerenciados por uma estação de gerenciamento de rede NMS (*Network Management Station*), devem estar equipados com um agente SNMP. Segundo Mello (2000, p.16) “o agente responde a requisições da estação de gerenciamento que pode ser o envio de informações de gerência ou ações sobre as variáveis do dispositivo onde está.”. O agente tem um papel fundamental em todo o processo de gerência da rede, como acessar e disponibilizar informações de gerência, além de alertar situações inesperadas de funcionamento dos dispositivos monitorados, através do envio de *traps* ao gerente. Conforme COSTA (2008, p.22) “Um agente tem o conhecimento das informações de gerenciamento locais e traduz essas informações para um formato compatível com o protocolo SNMP”.
- Gerente: atua como uma interface entre o as aplicações de gerência correntes no NMS e os agentes espalhados pelos dispositivos da rede. Conforme Mello (2000), o gerente tem como função enviar comando aos agentes, solicitando informações sobre variáveis de um objeto gerenciado ou modificando o valor de determinada variável. Ainda segundo Mello (2000), cabe também ao gerente encaminhar à aplicação de gerência os *traps* que porventura sejam enviados pelos agentes.

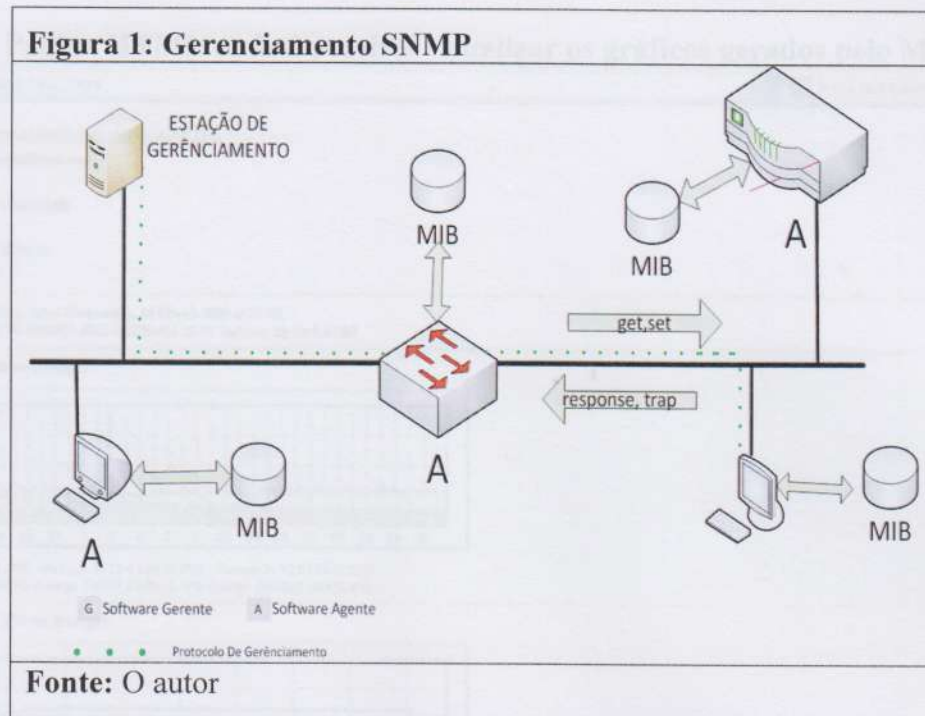
- A MIB, ou seja, base de informações de gerenciamento, como o próprio nome sugere, pode ser considerada uma base de dados na qual são armazenados todos os objetos gerenciados que o agente rastreia, esse objetos no qual estão armazenados na MIB em sua essência são variáveis que representam aspectos de determinado sistema. Conforme Battisti (2000), uma estação de gerenciamento monitora um equipamento recuperando dos mesmos valores de objetos da MIB, sendo possível realizar uma ação sobre o agente, como modificar ou alterar os valores de variáveis específicas.

3.4.2 Operações do SNMP

Para se tornar possível a comunicação entre os agentes e gerentes no SNMP, existem várias operações. A seguir serão apresentadas algumas operações básicas.

- GET: esta operação é realizada pelo gerente sempre que houver a necessidade de recuperar alguma informação do objeto gerenciado;
- GETNEXT: muito parecido com o comando GET, porém de acordo com o que cita Mello (2000), quando realizado comando GetNext, o agente deve ler a próxima instância disponível, na ordem especificada pela MIB, da(s) variável(is) associada;
- SET: esta operação faz requisições a determinados agentes, com a atribuição e/ou alteração de valores de determinadas variáveis especificadas em uma MIB;
- TRAP: Segundo Mello (2000, p.20) “A operação *Trap* difere de todas as outras. Ela é utilizada por um agente SNMP para notificar de forma assíncrona a um gerente algum evento extraordinário que tenha ocorrido no objeto gerenciado”.

É possível observar a interação entre todos os elementos da arquitetura SNMP na figura 1.

Figura 1: Gerenciamento SNMP

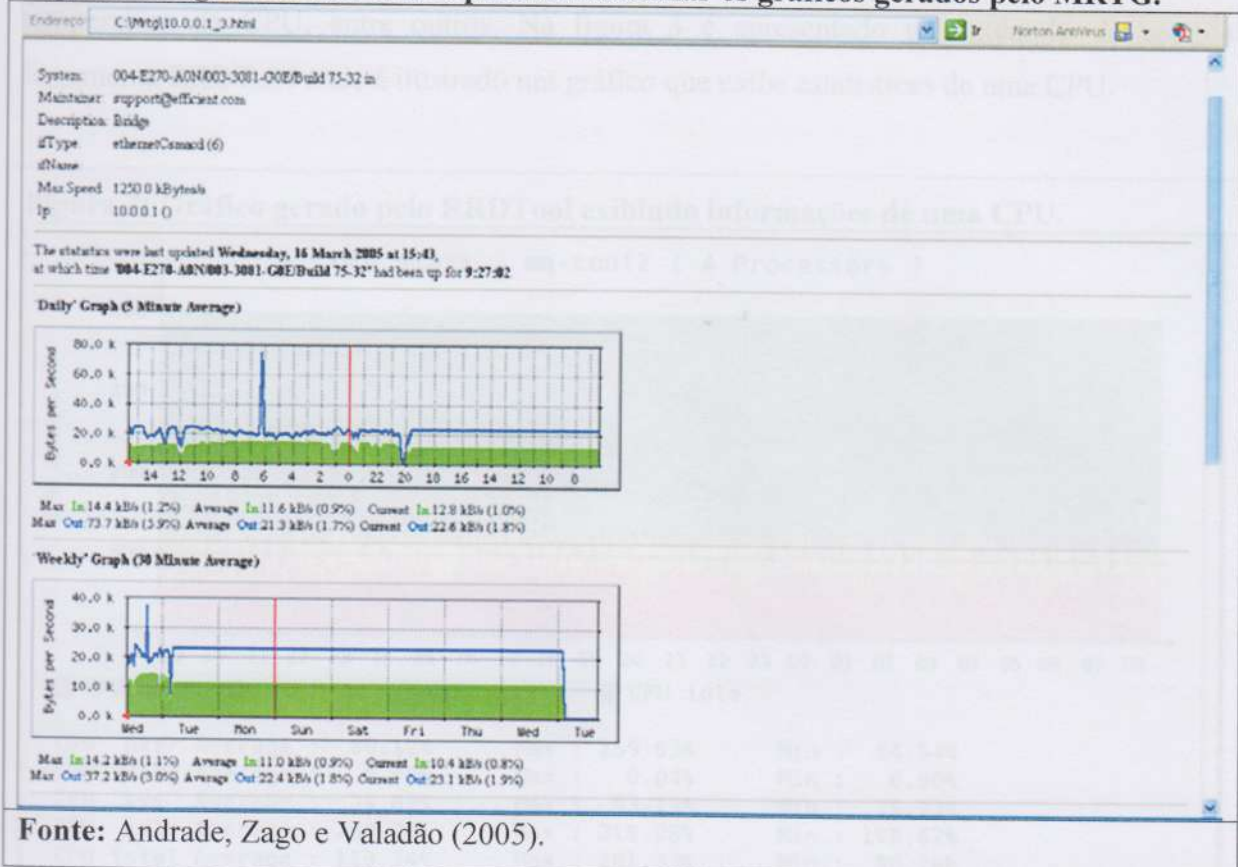
3.5 MRTG

Uma vez coletados dados por um gerente SNMP, é necessário transformar esses dados em informações úteis ao gerenciamento da rede. Algumas ferramentas podem tornar a tarefa de gerenciar a rede um pouco mais fácil, exibindo os dados de forma agradável.

O MRTG (*Multi Router Traffic Grapher*) de acordo com Andrade, Zago e Valadão (2005, p.10), “é uma ferramenta que utiliza o protocolo SNMP para obtenção de informações de tráfego dos mais variados objetos presentes em um ambiente de rede”. Além de realizar análise de tráfego, o mesmo pode monitorar quaisquer dados que sejam coletados via SNMP.

O MRTG gera gráficos que são visualizados em HTML (*Hyper Text Markup Language*). Funciona tanto na plataforma Microsoft Windows como nos sistemas Unix/Linux, foi desenvolvido em *Perl*, e é um *software* livre publicado sob a licença GNU GPL (GNU *General Public License*). O MRTG foi um dos primeiros sistemas a exibir informações de gerência de forma agradável aos administradores. A figura 2 ilustra a página onde são gerados os gráficos gerados pelo MRTG.

Figura 2: Página HTML onde é possível visualizar os gráficos gerados pelo MRTG.



3.6 RRD E RRDTOOL

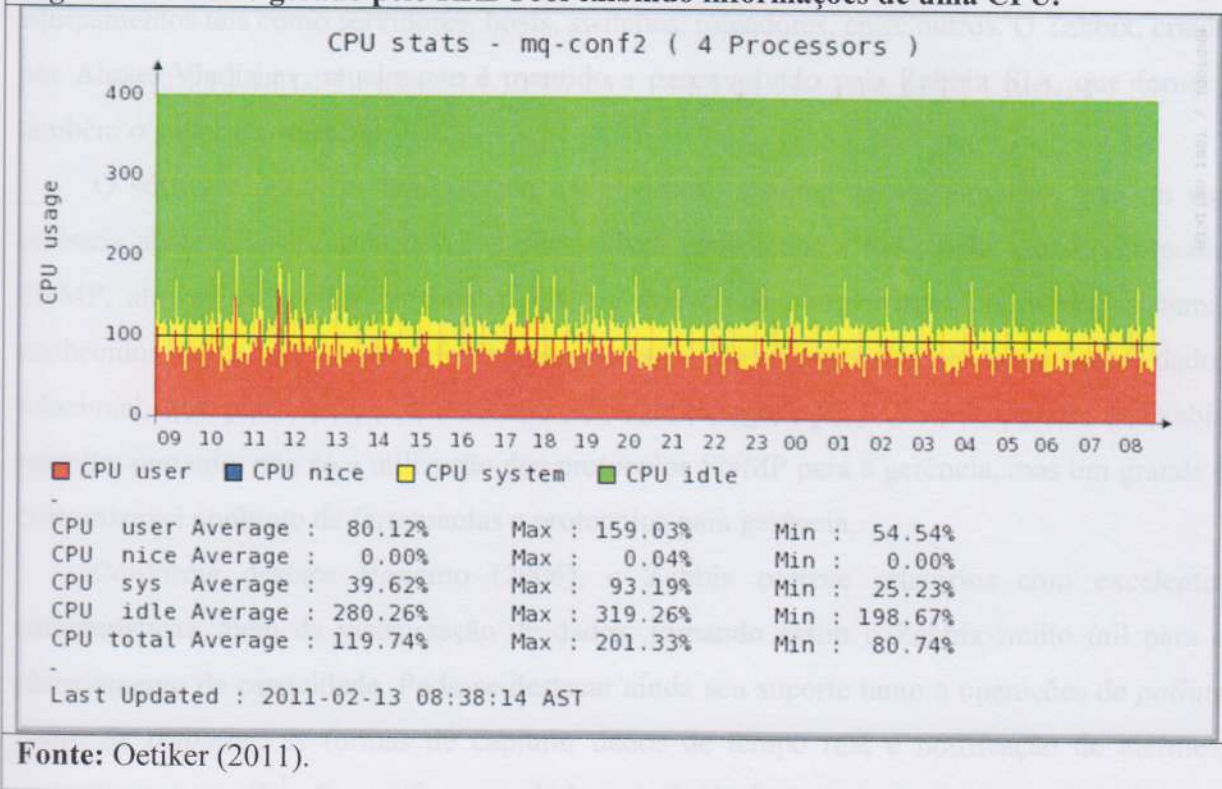
RRD (*Round Robin Database*) é um mecanismo de bancos de dados usualmente utilizado para guardar dados relacionados a sequências de tempo, como banda de rede, temperatura, carga de CPU, dentre outros. O banco de dados Round Robin Database grava dados circularmente, impedindo o crescimento indefinido do banco. Os dados são gravados em arquivos RRA (*Round Robin Archive*).

O RRDTool (*Round Robin Database Tool*), é um conjunto de ferramenta que implementam RRD e tem como principal função a monitoração de dados. Conforme Bonomo (2006, p.18), “RRDTool é uma ferramenta poderosa para monitoração de tráfego, que reduz sensivelmente a carga gerada pela monitoração para a exibição dos gráficos, o que possibilita uma representação visual em tempo real do tráfego.”.

O RRDTool é licenciado sob GNU GPL. De acordo com Black (2008), o mesmo foi desenvolvido com o objetivo de armazenar séries de dados numéricos sobre o estado da rede de computadores, e tem como uma de suas principais funções o armazenamento séries de

dados temporais podendo ser empregado para o armazenamento de outros dados como temperatura da CPU, entre outros. Na figura 3 é apresentado um exemplo do uso da ferramenta RRDTool onde é ilustrado um gráfico que exibe estatísticas de uma CPU.

Figura 3: Gráfico gerado pelo RRDTool exibindo informações de uma CPU.



Fonte: Oetiker (2011).

...podem configurar um e-mail para receber os devidos alertas na ocorrência de algum evento na rede.

Outro ponto forte do Zabbix é sua interface gráfica para Web, onde o usuário tem acesso a todas as configurações do programa. É uma interface simples e intuitiva, permitindo a visualização de todos os serviços e equipamentos monitorados na rede, além de dar acesso a todos os relatórios e estatísticas, assim como também a consultas e análise das informações armazenadas nos bancos de dados.

É possível listar alguns dos recursos para monitoramento oferecidos pelo Zabbix:

- Roda em todas as distribuições Linux/Unix.
- Realiza monitoramento sem a necessidade de um agente instalado.
- Uma excelente interface gráfica para Web.
- Software Open Source distribuído pela licença GPL v2.
- Integração com bancos de dados MySQL, SQLite, PostgreSQL ou Oracle.

4 A FERRAMENTA ZABBIX

O Zabbix é um software de código aberto, publicado sob licença GPL, e consiste em uma ferramenta de gerenciamento de redes, a qual permite monitorar inúmeros serviços e equipamentos que estão ativos na rede, seja uma simples aplicação instalada na rede ou equipamentos tais como servidores, hosts, switches, roteadores, entre outros. O Zabbix, criado por Alexei Vladishev, atualmente é mantido e desenvolvido pela Zabbix SIA, que fornece também o suporte comercial.

O software pode ser considerado um poderoso sistema de monitoração, que em sua essência absorve as informações dos dispositivos conectados a rede, pelo uso do protocolo SNMP, através de *scripts* personalizados, ou através de componentes do próprio sistema, conhecidos como agentes. As informações coletadas são armazenadas em um banco de dados relacional, que pode ser escolhido entre MySQL, PostgreSQL, SQLite e Oracle. O Zabbix permite, portanto, não só a utilização dos protocolos SNMP para a gerência, mas um grande e customizável conjunto de ferramentas e protocolos para gerência.

Conforme destaca Bonomo (2006), o Zabbix oferece relatórios com excelentes características, além da visualização de dados, tornando assim o Zabbix muito útil para o planejamento de capacidade. Pode-se destacar ainda seu suporte tanto a operações de *polling* como de *trapping*, as formas de capturar dados de tempo real e notificação de alarmes, respectivamente. O software fornece ainda um flexível mecanismo de alarmes onde seus usuários podem configurar um *e-mail* para receber os devidos alertas na ocorrência de algum evento na rede.

Outro ponto forte do Zabbix é sua interface gráfica para *Web*, onde o usuário tem acesso a todas as configurações do programa. É uma interface simples e intuitiva, permitindo a visualização de todos os serviços e equipamentos monitorados na rede, além de dar acesso a todos os relatórios e estatísticas, assim como também a consultas e análise das informações armazenadas nas bases de dados.

É possível listar alguns dos recursos para monitoração oferecidos pelo Zabbix:

- Roda em todas as distribuições Linux/Unix.
- Realiza monitoramento sem a necessidade de um agente instalado.
- Uma excelente Interface gráfica para *Web*.
- Software *Open Source* distribuído pela licença GPL v2.
- Integração com banco de dados MySQL, SQLite, PostgreSQL ou Oracle.

- Suporte nativo ao protocolo SNMP.
- Controle de acesso e permissões dos usuários.
- Notificação de eventos por *e-mail*.
- Monitoramento em tempo real.
- *Log* de auditoria.

Além disso, ao Zabbix proporciona outras grandes vantagens, aos seus usuários, como:

- Uma baixa curva de aprendizado.
- Implantação de baixo custo.
- Configuração e Instalação simples.
- Sistema de monitoramento centralizado. Onde todas as informações são armazenadas em uma base de dados relacional.

4.1 ARQUITETURA

O Zabbix é dividido em três funções, que o tornam muito próximo à própria arquitetura do SNMP. O sistema é formado pelas subdivisões a seguir.

- Servidor Zabbix.
- Agente Zabbix.
- Interface do Zabbix.

O servidor Zabbix foi desenvolvido em linguagem C. Observa-se que o servidor é de grande responsabilidade, pois é através do mesmo que se torna possível a coleta e o armazenamento de todos os dados que estão sendo monitorados.

De acordo com Filho (2010), o Agente Zabbix tem como função transmitir informações que foram coletadas dos sistemas operacionais no qual estão instalados para o servidor. O mesmo segue hospedado na máquina rodando como um serviço, e no momento em que recebe alguma solicitação do servidor, o agente a processa e retorna os dados solicitados, como por exemplo: consumo dos recursos de processador, memória, temperatura do processador e outros vários. Como já afirmado, o Zabbix oferece suporte para várias plataformas, disponibilizando seus agentes para vários sistemas operacionais.

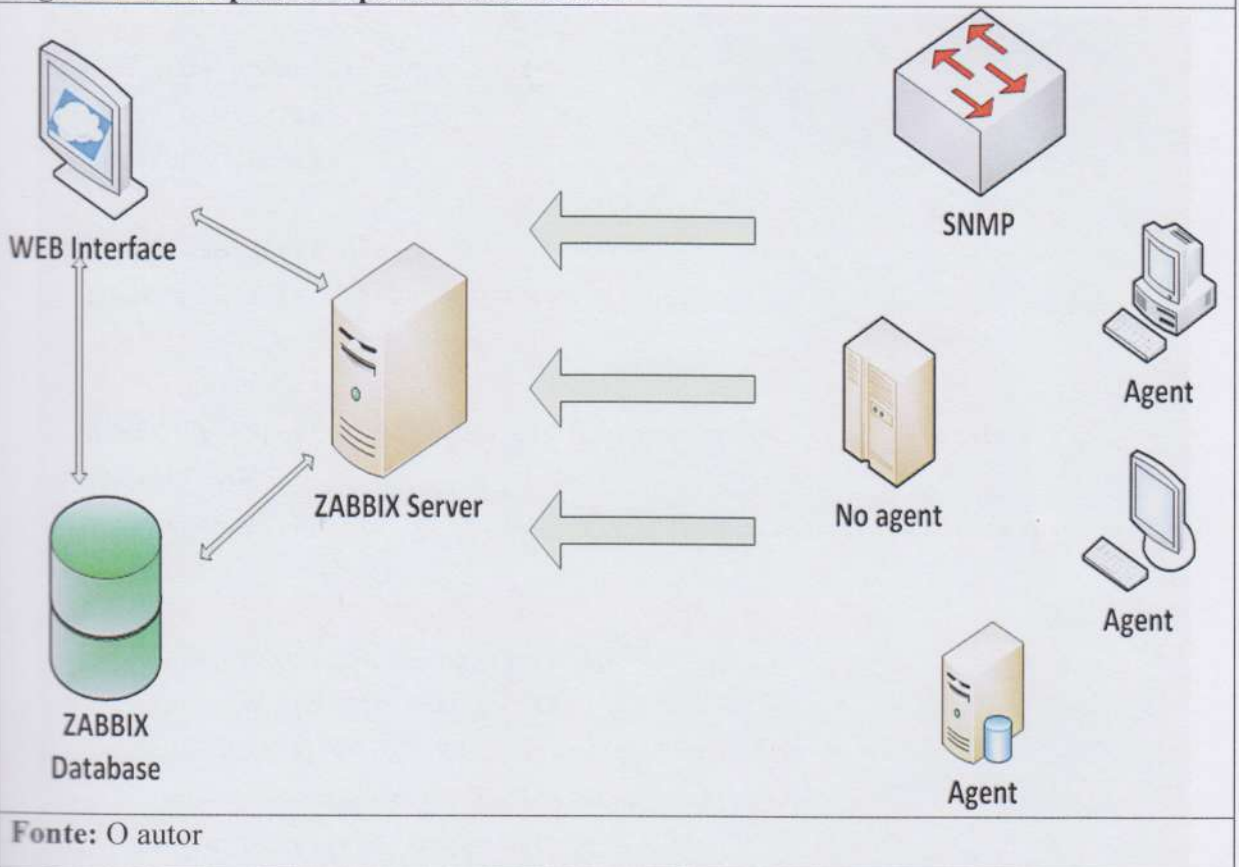
O Zabbix possui uma interface para *Web* bastante simples, e de excelente utilidade. A mesma, desenvolvida em linguagem PHP, torna possível que o administrador tenha total

acesso para configurar o sistema, selecionar agentes, aplicar *scripts*, configurar alarmes, e, principalmente, exibir relatórios, gráficos e telas sobre o estado dos recursos de rede.

É possível perceber que os agentes do Zabbix possuem função semelhante a um agente SNMP, assim como o servidor Zabbix e sua interface correspondem ao Sistema de Gerenciamento de Redes (NMS) da arquitetura SNMP. Os agentes do Zabbix são de uso opcional, podendo o administrador usar os agentes SNMP em substituição, ou utilizar as duas formas de monitoramento, ou até mesmo criar sua própria forma de monitoramento.

Observa-se, na figura 4, que o Zabbix apresenta uma versátil arquitetura e distribuição de componentes.

Figura 4: Principais Componentes do Zabbix



4.2 PROTOCOLOS, FERRAMENTAS E TERMINOLOGIA

4.2.1 Template

Um *template* ou molde é um modelo de dados onde são especificadas as operações e as informações que serão utilizadas no monitoramento. O template, é escrito em formato XML (*Extensible Markup Language*), pode ser aplicado a um ou mais equipamentos, de forma a especificar a captura de dados para aquele tipo de equipamento. No quadro 1, é exibido um trecho de template, o qual contém metadados e outros dados que definem os parâmetros a serem gerenciados em um *switch* por exemplo.

Quadro 1: Trecho de um *template*

```
<hosts>
<host name="Template_3COM_2924">
<ip>127.0.0.1</ip>
<port>10050</port>
<groups>
<group>Templates</group>
<group>3Com Switch 2924 SFP-PLUS</group>
</groups>
<trigger>
<description>Port 1 on {HOSTNAME} Status Change</description>
<type>0</type>
<expression>{Template_3COM_2924:ifOperStatus.1.diff(0)}=1</expression>
</trigger>
<item type="4" key="ifOutOctets.13" value_type="3">
  <snmp_oid>ifInOctets.24</snmp_oid>
  <snmp_port>161</snmp_port>
  <description>Port 24 Status</description>
  <snmp_oid>ifOperStatus.24</snmp_oid>
  <snmp_port>161</snmp_port>
</item>
<graph name="Port 25 Traffic" width="900" height="200">
```

Fonte: O autor

Conforme estar ilustrado no quadro 1, percebe-se como é formada a sintaxe de um *template*. Pode-se definir grupos para um determinado modelo de equipamento (*host*) ou qualquer seja o ativo monitorado, Os parâmetros de monitoramento são definidos em itens

como o “ifOutOctets” que monitora os bytes de entrada de determinada porta de um *switch*. Além deles, também é possível definir: *triggers*, que são condições de limite para o disparo de uma ação; e os gráficos que serão gerados para determinados itens, com o objetivo de monitorar o tráfego que passa nas portas dos ativos rede monitorados.

4.2.2 ICMP

O Protocolo ICMP pode ser utilizado no Zabbix basicamente para mensagens de *echo* (ping). Através dessas mensagens é possível testar se um equipamento está disponível em uma rede IP. Em um ambiente de rede complexo, deve-se destacar algum esforço em manter as mensagens de ICMP liberadas para transitar entre o sistema de gerenciamento de redes (NMS) e todos os equipamentos monitorados.

No Zabbix, os ativos de redes podem se utilizar de *templates* configurados com vários itens de monitoramento. Dentre esses itens, é possível adicionar algum monitoramento através de SNMP, ICMP além de diversos outros tipos de monitoramento.

Figura 5: Itens de monitoração simples

STANDARD ITEMS		Type	Simple check
Key	Description		
ftp<,port>	Checks if FTP server is running		
ftp_perf<,port>	Checks if FTP server is running		
http<,port>	Checks if HTTP (WEB) server		
http_perf<,port>	Checks if HTTP (WEB) server		
icmppingloss[<ip>,<count>,<interval>,<size>,<timeout>]	Returns percentage of lost ICMP		
icmpping[<ip>,<count>,<interval>,<size>,<timeout>]	Checks if server is accessible		
icmppingsec[<ip>,<count>,<interval>,<size>,<timeout>,<type>]	Returns ICMP ping response		
imap<,port>	Checks if IMAP server is running		
imap_perf<,port>	Checks if IMAP server is running		
ldap<,port>	Checks if LDAP server is running		
ldap_perf<,port>	Checks if LDAP server is running		

Fonte: O autor

A figura 5 apresenta alguns dos tipos de checagem disponíveis para uso na criação de um novo item de monitoramento, com os quais é possível saber se determinados serviços estão em execução como HTTP, FTP, IMAP, ICMP, além de outros. A figura 5 lista apenas

os tipos de monitoramento de checagem simples, excluindo os que utilizam protocolos mais complexos como o SNMP ou os mecanismos proprietários do Zabbix.

O ICMP, listado na figura através dos itens `icmpping`, `icmppingloss` e `icmppingsec`, recebe como parâmetros o endereço IP de destino do teste (equipamento a ser testado), e outros parâmetros comuns das mensagens de *echo* do ICMP, como o tamanho em *bytes* de cada mensagem, a quantidade de mensagens, etc. O resultado obtido a partir de uma checagem é “sim” ou “não”, “0” ou “1”. O `icmpping` é implementado, no Zabbix, com o utilitário `FPING`, disponível para download na Internet.

4.2.3 SNMP

Zabbix utiliza o SNMP através de bibliotecas para o sistema operacional e de utilitários que realizam as operações do protocolo. Essas operações são definidas para o Zabbix através de *templates*. Vale salientar que para cada modelo de equipamento é necessário um novo *template* com seus parâmetros específicos. Isso acontece porque cada equipamento possui uma MIB específica, que pode ou não ter identificadores de objetos (OID) padronizados. Caso a MIB não possua OIDs bem conhecido, é necessário utilizar algum método para descobrir os OIDs do equipamento específico. No quadro 2, observa-se um pequeno trecho, como exemplo, de um *template* associado a um *switch* 3COM 2924 ilustrando o uso do SNMP através do parâmetro `<snmp_oid>`.

Quadro 2: Trecho de um *template* exemplificando o uso do SNMP

```
<host name="Template_3COM_2924">
<item type="4" key="ifInOctets.24" value_type="3">
  <description>Port 24 Traffic IN</description>
  <snmp_oid>ifInOctets.24</snmp_oid>
  <snmp_port>161</snmp_port>
</item>
<item type="4" key="ifOperStatus.24" value_type="1">
  <description>Port 24 Status</description>
  <snmp_oid>ifOperStatus.24</snmp_oid>
  <snmp_port>161</snmp_port>
</item>
```

Fonte: O autor

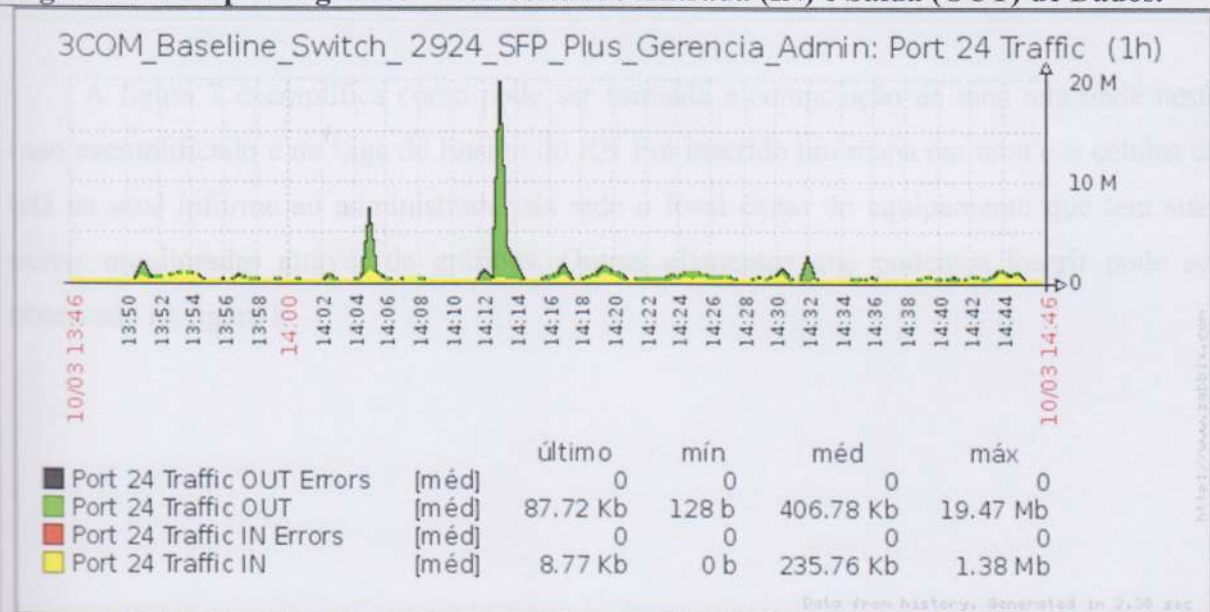
É possível observar a criação de dois itens de monitoramento, com acesso a objetos SNMP através de OIDs obtidos na MIB do equipamento. No primeiro item, o objeto ifInOctets.24 é acessado, enquanto no segundo item o objeto ifOperStatus.24 é acessado.

4.2.4 Gráficos

Um dos grandes diferenciais do Zabbix em relação a outros sistemas de monitoramento de redes, sem dúvida é a geração de gráficos. Além de complexos e com vários parâmetros, esses gráficos são gerados a partir de um módulo PHP GD que é uma biblioteca, que permite realizar operações de criação e manipulações com imagens de forma rápida e dinâmica, a partir de código PHP.

As informações contidas em um gráfico são armazenadas no banco de dados PostgreSQL na forma de um *buffer* circular, proporcionando ao administrador da rede uma melhor visualização e clareza dos itens monitorados. Pode-se visualizar melhor como é composto um gráfico analisando a figura 6 que exemplifica o comportamento de um gráfico no Zabbix.

Figura 6: Exemplo de gráfico – Demonstrando Entrada (IN) e Saída (OUT) de Dados.

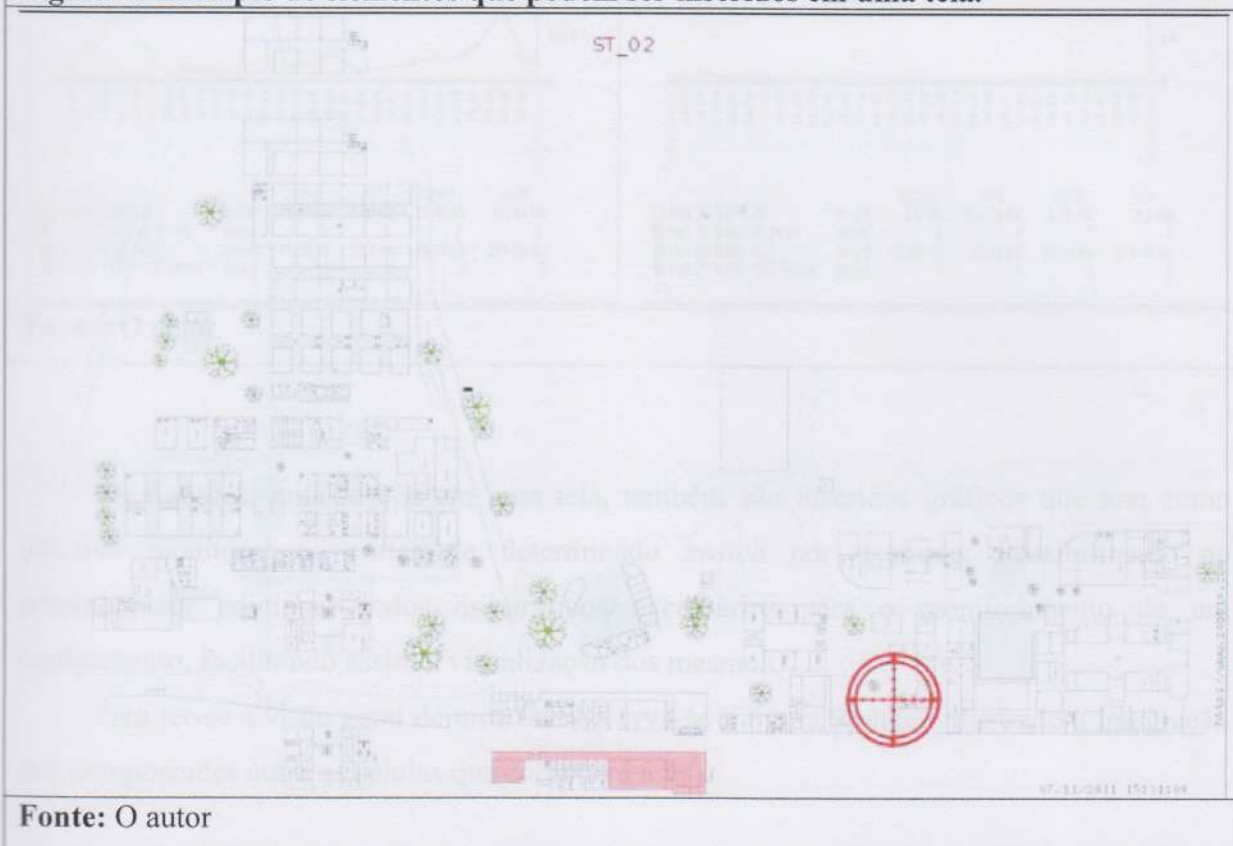


Fonte: O autor

O Zabbix oferece a opção de criar telas, que é basicamente uma tabela dividida por células, que podem ser preenchidas por um gráfico, mapa, entre outros elementos,

proporcionando ao administrador a personalização da visualização do monitoramento como observa-se na figura 7.

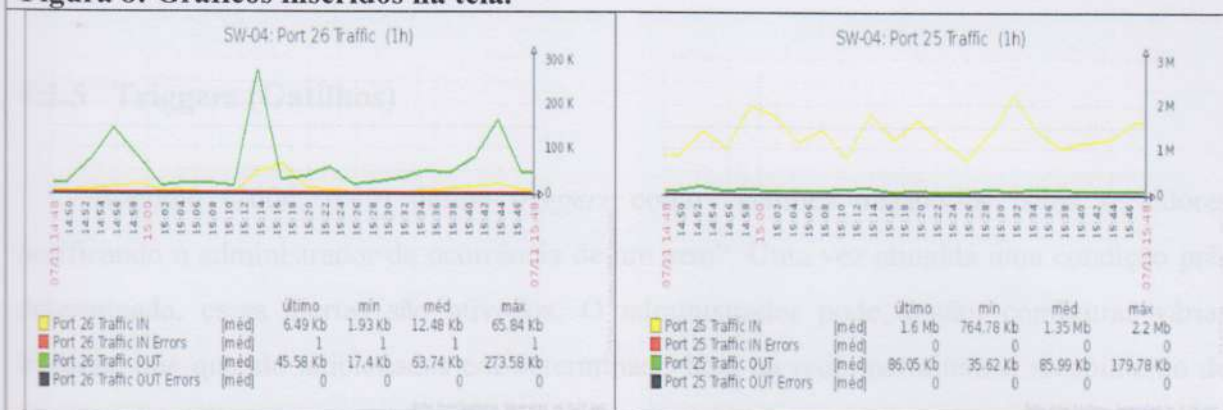
Figura 7: Exemplo de elementos que podem ser inseridos em uma tela.



Fonte: O autor

A figura 7 exemplifica como pode ser formada a composição de uma tela onde neste caso exemplificado e na Liga de Ensino do RN Foi inserido um mapa em uma das células da tela na qual informa ao administrador da rede o local exato do equipamento que tem suas portas monitoradas através de gráficos. Outros elementos que podemos inserir pode ser observado na figura 8.

Figura 8: Gráficos inseridos na tela.



Fonte: O autor

Na figura 8, nota-se que em uma tela, também são inseridos gráficos que tem como objetivo monitorar as portas de determinado *switch* por exemplo. Possibilitando ao administrador adicionar todos os gráficos necessários para o monitoramento de um equipamento, facilitando assim a visualização dos mesmos.

Para ter-se a visão geral de uma tela observa-se a figura 9 onde é ilustrado a integração dos componentes entre as células que compõem a tela.

Figura 9: Visão geral de uma tela.



Fonte: O autor

4.2.5 Triggers (Gatilhos)

Bonomo (2006, p.49) define *triggers* como “alarmes disparados pelos servidores notificando o administrador da ocorrência de um erro”. Uma vez atingida uma condição pré-determinada, esses alertas são ativados. O administrador pode, então, configurar várias triggers, que quando adicionadas em determinado host na rede, permitem o recebimento de alertas sobre todas as ocorrências de erro do host.

Figura 10: Triggers associados a um switch

CONFIGURAÇÃO DAS TRIGGERS			
TRIGGERS			
Mostrando 1 a 27 de 27 encontrado(s)			
Lista de templates		Aplicações (0)	
☐	Risco	Status	Nome ↑
☐	Médio (Average)	Ativo	Port 1 on {HOSTNAME} Status Change
☐	Médio (Average)	Ativo	Port 2 on {HOSTNAME} Status Change
☐	Médio (Average)	Ativo	Port 3 on {HOSTNAME} Status Change
☐	Médio (Average)	Ativo	Port 4 on {HOSTNAME} Status Change
☐	Médio (Average)	Inativo	Port 5 on {HOSTNAME} Status Change
☐	Médio (Average)	Inativo	Port 6 on {HOSTNAME} Status Change
☐	Médio (Average)	Inativo	Port 7 on {HOSTNAME} Status Change
☐	Médio (Average)	Inativo	Port 8 on {HOSTNAME} Status Change
☐	Médio (Average)	Inativo	Port 9 on {HOSTNAME} Status Change
☐	Médio (Average)	Inativo	Port 10 on {HOSTNAME} Status Change

Fonte: O autor

Na figura 10 observam-se algumas *triggers* que estão associados a um *switch*. A associação é feita ligando uma trigger diretamente a um item, ou seja, para cada trigger tem-se um item associado, pode-se analisar também nesta figura qual o risco de cada *trigger* bem como seu status.

Na figura 11, é possível observar as condições de limite estabelecidas em cada *trigger*. Uma condição é geralmente estabelecida sobre o resultado de uma função (um cálculo) realizado sobre o valor obtido em um item.

Figura 11: Condições de limites das *triggers*

CONFIGURAÇÃO DAS TRIGGERS		Criar trigger
Grupo: Todos	Host: 3COM_Baseline_Switch_2226_P	[Esconder triggers inativas]
Plus_Bib_Fam	DNS: -	Porta: 10050 Status: Monitorado Disponibilidade: Detectado
Expressão	Erro	
{3COM_Baseline_Switch_2226_Plus_Bib_Fam:ifOperStatus.1.diff(0)}=1	✓	
{3COM_Baseline_Switch_2226_Plus_Bib_Fam:ifOperStatus.2.diff(0)}=1	✓	
{3COM_Baseline_Switch_2226_Plus_Bib_Fam:ifOperStatus.3.diff(0)}=1	✓	
{3COM_Baseline_Switch_2226_Plus_Bib_Fam:ifOperStatus.4.diff(0)}=1	✓	
{3COM_Baseline_Switch_2226_Plus_Bib_Fam:ifOperStatus.5.diff(0)}=1	✓	
{3COM_Baseline_Switch_2226_Plus_Bib_Fam:ifOperStatus.6.diff(0)}=1	✓	
{3COM_Baseline_Switch_2226_Plus_Bib_Fam:ifOperStatus.7.diff(0)}=1	✓	
{3COM_Baseline_Switch_2226_Plus_Bib_Fam:ifOperStatus.8.diff(0)}=1	✓	
{3COM_Baseline_Switch_2226_Plus_Bib_Fam:ifOperStatus.9.diff(0)}=1	✓	
{3COM_Baseline_Switch_2226_Plus_Bib_Fam:ifOperStatus.10.diff(0)}=1	✓	
{3COM_Baseline_Switch_2226_Plus_Bib_Fam:ifOperStatus.11.diff(0)}=1	✓	

Fonte: O autor

4.2.6 Ações

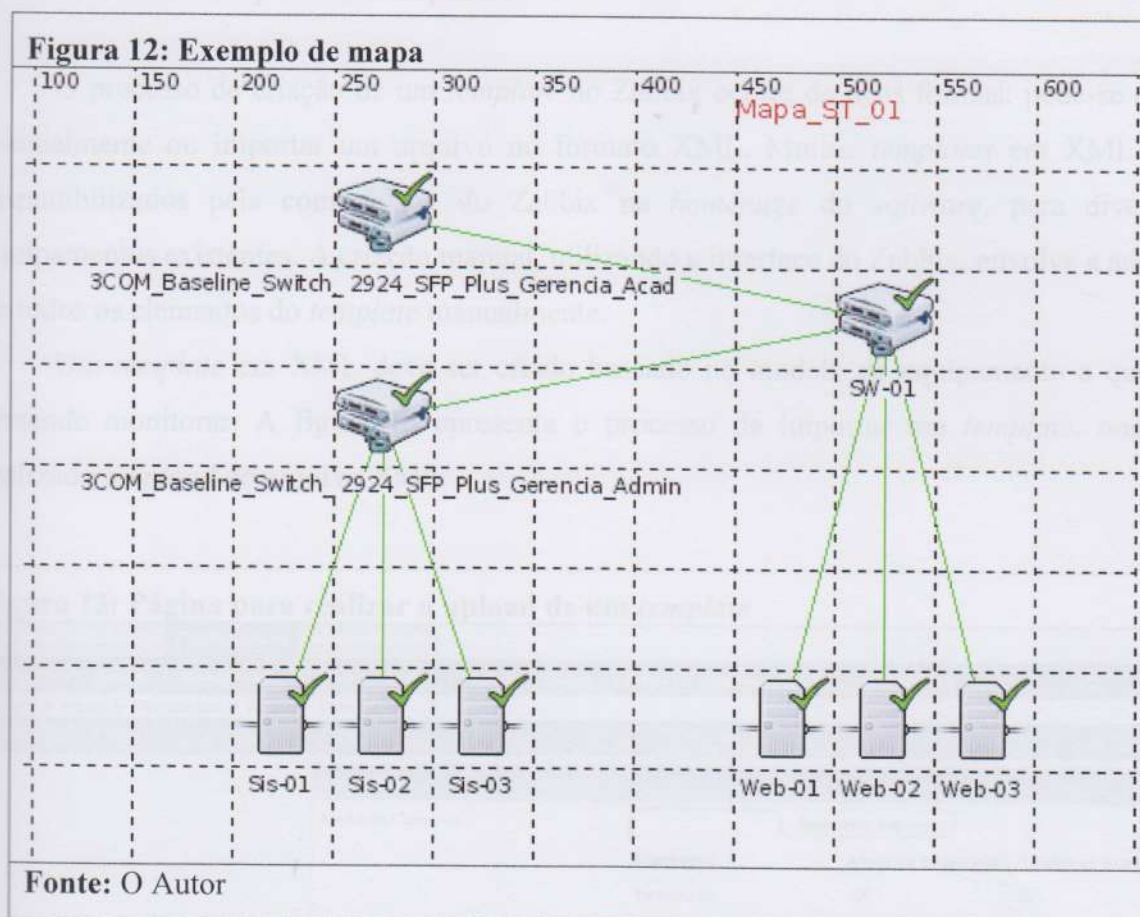
Um recurso que o servidor Zabbix disponibiliza é criação de ações, que são acionadas a partir de algum alerta disparado, ou seja, na mudança de estado (disparo) de uma *trigger* ou um grupo de *triggers*.

Quando uma determinada ação é ativada, pode ser enviada uma notificação para um usuário ou para um grupo de usuários, através de *e-mail*, como também via *jabber* que é um protocolo utilizado para troca de mensagens instantâneas. Além disso, é possível executar um comando remoto ou um *script* local no equipamento que ocasionou o alarme, por exemplo.

4.2.7 Mapas

Mapa é outro recurso que o Zabbix oferece e que proporciona uma melhor visualização dos *hosts* cadastrados em seu banco de dados. Através dos mapas torna-se possível simular todo o desenho da rede envolvida e monitorada, inserindo todos os ativos gerenciados

(switches, máquinas, servidores, entre outros), conectando esses equipamentos em posições específicas do mapa proporcionando um maior nível de semelhança com rede física. Um exemplo de um map é ilustrado na figura 12.



Como apresentado na figura 12, os mapas são de grande utilidade, sendo possível visualizar o conjunto de hosts monitorados pelo Zabbix. De acordo com a ativação de *triggers*, o que destaca um problema em um equipamento. O mapa é atualizado, em tempo real, exibindo o status de toda a rede. Assim, através dos mapas, tem-se uma visão geral de como anda a “saúde” da rede.

4.3 EXEMPLO DE MONITORAMENTO DE UM EQUIPAMENTO DE REDE

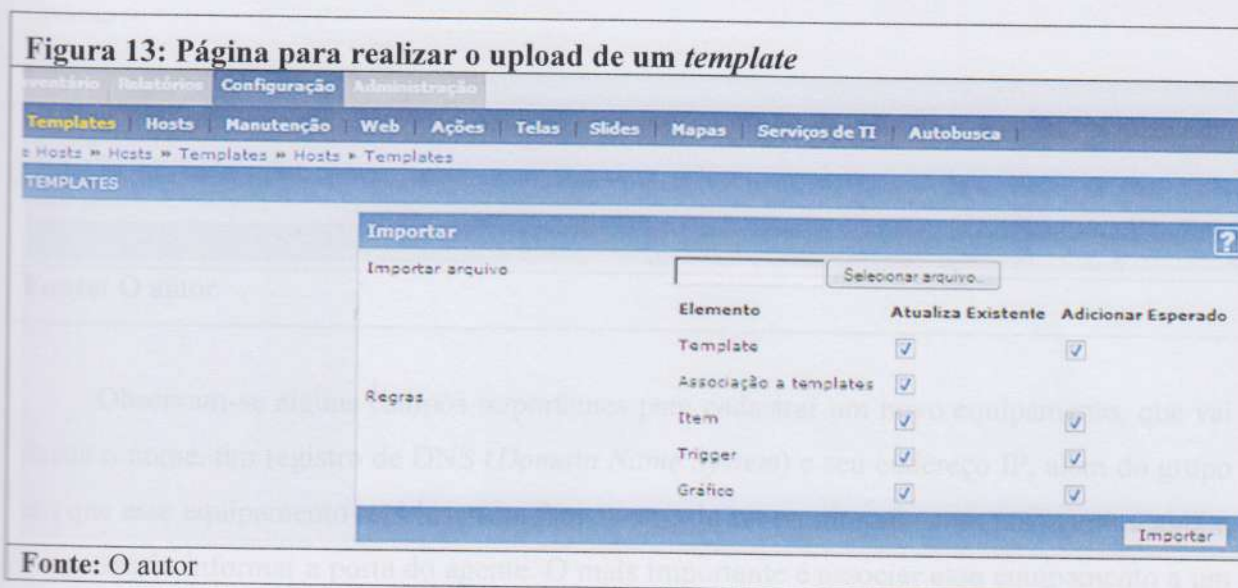
Com todos os recursos de monitoramento especificados anteriormente, é possível seguir uma linha de raciocínio para estabelecer o monitoramento de uma rede de computadores

utilizando o Zabbix. Os passos para realizar o processo estão descritos neste tópico, que parte da criação de um *template* de equipamento e chega nos produtos finais do monitoramento, que são os elementos visuais e os alertas.

4.3.1 Criar ou Importar Templates

O processo de criação de um *template* no Zabbix ocorre de duas formas: pode-se criar manualmente ou importar um arquivo no formato XML. Muitos *templates* em XML são disponibilizados pela comunidade do Zabbix na *homepage* do *software*, para diversos equipamentos existentes. A criação manual, utilizando a interface do Zabbix, envolve a adição de todos os elementos do *template* manualmente.

Um *template* em XML deve ser criado baseado no modelo de equipamento a que se pretende monitorar. A figura 13 apresenta o processo de importar um *template*, onde é realizado o *upload* do arquivo XML.



4.3.2 Inserir Hosts

Afigura 14 exemplifica o processo de inserir um novo equipamento a ser monitorado pelo servidor Zabbix. Um equipamento é, para o Zabbix, um *Host*.

Figura 14: Adicionar um Host no Zabbix

Monitoramento | Inventário | Relatórios | **Configuração** | Administração

Grupos de Hosts | Templates | **Hosts** | Manutenção | Web | Ações | Telas | Slides | Mapas | Serviços de TI | Autobusca

Histórico: Fila » Dashboard » Hosts » Dashboard » Hosts

CONFIGURAÇÃO DE HOSTS

Host

Nome:

Grupos:

Nos grupos:

Outros grupos:

Novo grupo:

Nome DNS:

Endereço IP:

Conectado a:

Porta do agente Zabbix:

Monitorado por proxy:

Status:

User IPMI: ☐

Salvar Cancelar

Fonte: O autor

Observam-se alguns campos importantes para cadastrar um novo equipamento, que vai desde o nome, um registro de DNS (*Domain Name System*) e seu endereço IP, além do grupo em que esse equipamento será inserido. Para o caso de ser monitorado com um agente Zabbix, é necessário informar a porta do agente. O mais importante é associar esse equipamento a um *template*, que tenha sido previamente criado ou importado. Desta forma, todos os itens, *triggers* e gráficos do *template* serão aplicados ao novo *Host*.

Figura 15: Adicionar *template* ao equipamento

Associado aos templates

Não há templates associados

Adicionar Desassociar Desassociar e limpar

Fonte: O autor

Analisando-se a figura 15 percebe-se que para associar um *template* ao *Host* basta clicar na opção adicionar, escolher o template de acordo com o modelo do equipamento, e por fim salvar o novo *Host*.

4.3.3 Criar Gráficos

Se os gráficos não estão no *template*, é necessário criá-los pela interface. Para criar um novo gráfico, escolhe-se o *Host* onde o gráfico será adicionado e, em seguida, clica-se em “Criar Gráfico”.

Figura 16: Formulário para criação de um gráfico

Gráfico	
Nome	
Largura	900
Altura	200
Tipo do gráfico	Normal
Mostrar horário comercial	☐
Mostrar triggers	☐
Linha Percentil (Esquerda)	☐
Linha Percentil (Direita)	☐
Valor MÍNIMO no eixo Y	Calculado
Valor MÁXIMO no eixo Y	Calculado
Itens	Adicionar
Visualizar Salvar Cancelar	

Fonte: O autor

Na figura 16, percebe-se que em um gráfico são inseridas informações como o nome, as dimensões e o tipo de gráfico, que varia de acordo com a necessidade. O elemento principal do gráfico é a coleção de itens que serão apresentados no mesmo. Os itens recuperam os dados que são exibidos no gráfico.

4.3.4 Criar Telas

O efeito do monitoramento pode ser sentido através da visualização ou do recebimento de alertas. No caso de visualização, os gráficos são geralmente organizados em telas.

Conforme visto anteriormente, telas são tabelas organizacionais de elementos gráficos que tem o objetivo de proporcionar ao administrador a personalização da visualização do monitoramento conforme citado anteriormente.

Figura 17: Formulário para criação de uma tela

Fonte: O autor.

As telas podem ser criadas aleatoriamente, conforme exibe a figura 17. Para criar uma tela são necessárias informações simples, como nome e disposição das linhas e colunas. Em seguida são arranjados elementos gráficos na tela recém-criada.

4.3.5 Criar Triggers

Para o caso de alertas, são necessárias as condições em que os alertas são gerados. Essas condições são especificadas em *triggers*, que são associadas a um item de monitoramento de um *host*. Dependendo do valor do item, ou de um cálculo sobre ele, a *trigger* é acionada.

Para adicionar uma *trigger*, deve ser editado o *host*, onde será visível o botão “Criar Trigger”. Observa-se, na figura 18, que uma *trigger* é composta com itens comuns como nome, e também com informações mais complexas, como no caso da expressão, que é uma função que será executada sobre o valor do item da *trigger*, e que definirá se a *trigger* será ativada ou não. Outras informações da *trigger* incluem uma definição do nível de risco e um status (inativo ou ativo).

Figura 18: Formulário para criar triggers

Trigger ?

Nome: Port 1 on (HOSTNAME) Status Change

Expressão (Alterar método de entrada): {3COM_Baseline_Switch_2226_Plus_Eto_FamiiLastChange.1.Last(0)} Adicionar

A trigger depende de: Não há dependências definidas

Nova dependência: Adicionar

Geração de Evento: Normal

Risco: Alto (High)

Comentários:

URL:

Inativo: ☐

Salvar Cancelar

Fonte: O autor

4.3.6 Criar Ações

Uma ação é o comportamento específico de um alerta. Ela pode ser ativada por uma ou mais *triggers*, utilizando qualquer combinação lógica entre elas. As ações são adicionadas em seu *menu* específico, conforme a figura 19.

Figura 19: Formulário para criar uma ação

Monitoramento | Inventário | Relatórios | **Configuração** | Administração

Grupos de Hosts | Templates | Hosts | Manutenção | Web | **Ações** | Telas | Slides | Mapas | Se

Histórico: Configuração de telas » Hosts » Configuração das triggers » Dashboard » Configuração de ações

CONFIGURAÇÃO DE AÇÕES

Ação

Nome:

Origem do evento: **Triggers** ▼

Ativar escalonamentos: ☐

Assunto padrão: {TRIGGER.NAME}; {TRIGGER.STATUS}

Mensagem padrão: {TRIGGER.NAME}; {TRIGGER.STATUS}
Last value: {ITEM.LASTVALUE}

Mensagem da recuperação: ☐

Status: **Ativo** ▼

Salvar Cancelar

Condições da ação

Condições: Nenhuma condição definida

Nova

Operações da Ação

☐ Detalhes **Ação**

Nenhuma ação definida

Nova

Fonte: O autor

Na interface apresenta conforme figura 19, pode-se definir o nome, o assunto e uma mensagem padrão. Em cada uma dessas informações, podem ser utilizados macros que trazem dados dos itens e das *triggers* utilizadas na ação.

A condição da ação, exibida na figura 20, é baseada em uma lógica envolvendo, normalmente, *triggers*. Por exemplo, um determinado aviso de *e-mail* pode ser disparado quando um *switch* não responde por ICMP *echo*, ou seja, se um switch parar de funcionar seja por perda de conectividade ou qualquer outra falha, bem quando ocorre uma mudança de status em uma de suas portas monitoradas.

Figura 20: Formulário para inserir as condições da ação

Condições da ação

Tipo do cálculo: **E/OU** ▼ (A ou B)

Condições:

(A) ☐ Trigger = "3COM_Baseline_Switch_2226_Plus_Bib_Farn:{HOSTNAME} Disponibilidade"

(B) ☐ Trigger = "3COM_Baseline_Switch_2226_Plus_Bib_Farn:Port 8 on {HOSTNAME} Status Change"

Remover selecionado(s)

Nova condição

Aplicação ▼ = ▼

Adicionar Cancelar

Fonte: O autor

Figura 22: Formulário para criação de mapa

Configuração | Administração

Manutenção | Web | Ações | Telas | Slides | **Mapas** | Serviços de TI | Autobusca

Configuração de mapas de rede » Grupos de Hosts » Mapas de rede

New system map ?

Nome

Largura

Altura

Imagem de fundo

Destaque de ícone ☐

Marcar elementos nas mudanças de status das triggers ☐

Expandir problemas singulares ☐

Tipo do texto do ícone

Localização do texto do ícone

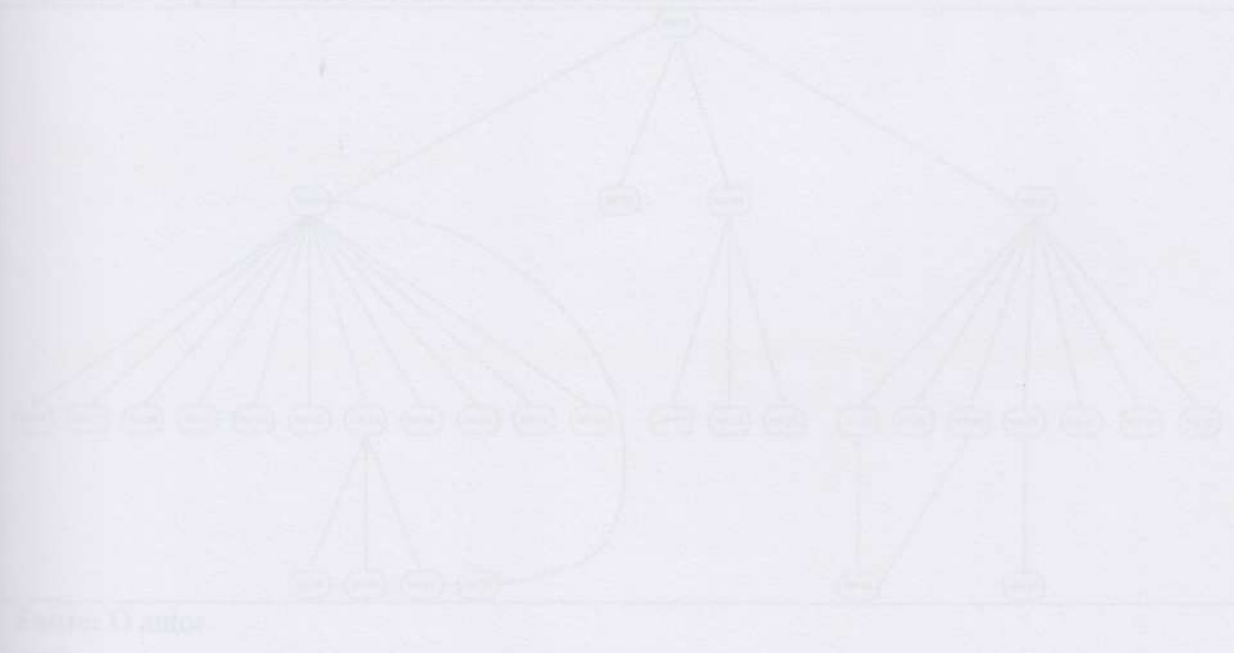
Exibição de problemas

Salvar Cancelar

Fonte: O autor

Observa-se, na figura 22, que para criar um mapa necessita-se de informações simples como nome, suas dimensões, uma imagem de fundo opcional.

Figura 23: Equipamentos de interconexão a monitorar

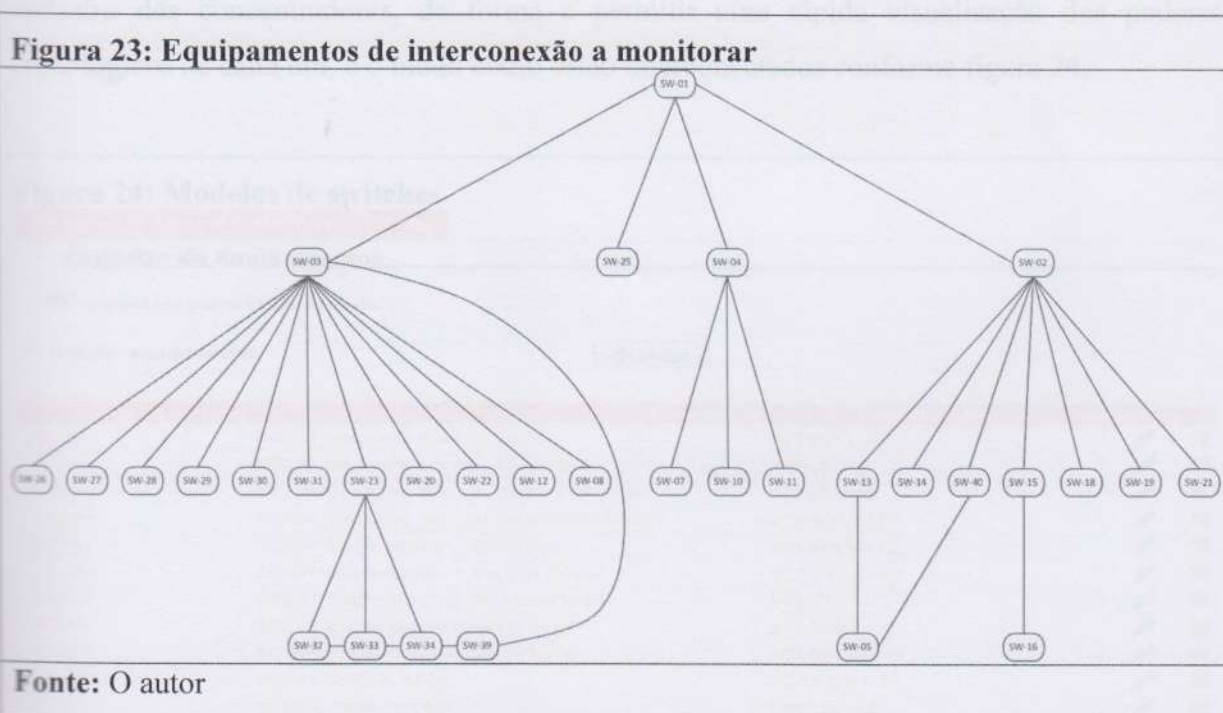


5 ESTUDO DE CASO

Para aumentar o nível de gerenciabilidade da rede de computadores da Liga de Ensino do RN, foi proposta e implementada uma solução de gerenciamento utilizando Zabbix. Neste capítulo são apresentadas a análise inicial, a escolha de equipamentos e a configuração do *software*.

5.1 ANÁLISE DA GERENCIABILIDADE DA REDE

A rede corporativa da Liga de Ensino do RN é formada majoritariamente de equipamentos comutadores (*switches*), totalizando um total inicialmente de 40 *switches* monitorados pelo Zabbix, dando um total geral de 70 ativos de redes monitorados entre *switches*, *access points* e servidores *web*. Caracteriza, assim, uma rede de *bridges*, com *switches* como equipamentos de interconexão e diversos meios e tecnologias de ligação entre eles. Assim, um dos objetivos principais para a gerência dos recursos de rede é gerenciar os *switches* da rede. A figura 23 mostra um grafo que ilustra os equipamentos de interconexão de *backbone* da rede em questão.



O quadro 3 mostra os tipos de equipamentos existentes na rede.

Quadro 3: Modelos de concentradores antigos

Marca	Modelo	PartNumber
3Com	3COMBaseline Switch 2024	3C16471-US
3Com	3COM Baseline Switch 2226-SFP Plus	3CBLSF26
3Com	3Com Baseline Switch 2924-SFP Plus	3CBLSG24
3Com	3COMSuperStack 3 Baseline	3C16465B-US
3Com	3COMSuperStack 3 Switch 3226	3CR17500-91
3Com	3COMSuperstack 3 Switch 4226T	3C17300A
3Com	3COMSuperStack Switch 1100	3C16950-US
3Com	3COMSuperStack Switch 3300	3C16985B-US
3Com	3COM Switch 4500	3CR17561-91
D-Link	D-Link DES-1016D	DES-1016D
D-Link	D-Link DES-1024D	DES-1024D
Encore	Switch Hub Encore 8 portas	ENH908-NWY
Extreme	Extreme Summit X450e-24p	Summit 16142

Fonte: O autor

Na listagem do quadro 3, é possível observar que alguns equipamentos não possuem suporte a SNMP. Para identificar essa questão, bem como auxiliar em outros aspectos gerenciais, a equipe da Gerência de Redes e Sistemas da FARN desenvolveu um sistema para cadastro dos concentradores, de forma a permitir uma rápida visualização dos poderes tecnológicos de cada um, e o modo como estão interconectados conforme figura 24.

Figura 24: Modelos de switches

Bridges ▾ Manter Cadastros ▾ Logs				
Cadastro de Equipamentos				
 Cadastrar novo equipamento  Buscar equipamento Buscar				
Marca	Modelo	Part Number	Editar	
3Com	3COM Baseline Switch 2024	3C16471-US		
3Com	3COM Baseline Switch 2226-SFP Plus	3CBLSF26		
3Com	3Com Baseline Switch 2924-SFP Plus	3CBLSG24		
3Com	3COM Baseline Switch 2928-SFP Plus	3CRBSG2893		
3Com	3COM SuperStack 3 Baseline	3C16465B-US		
3Com	3COM SuperStack 3 Switch 3226	3CR17500-91		
3Com	3COM Superstack 3 Switch 4226T	3C17300A		
3Com	3COM SuperStack Switch 1100	3C16950-US		
3Com	3COM SuperStack Switch 3300	3C16985B-US		
3Com	3COM Switch 4500	3CR17561-91		
D-Link	D-Link DES-1016D	DES-1016D		
D-Link	D-Link DES-1024D	DES-1024D		
Encore	Switch Hub Encore 8 portas	ENH908-NWY		
Extreme	Extreme Summit X450e-24p	summit: 16142		

Fonte: O autor

Na figura 24 é possível visualizar todos os modelos de equipamentos ao qual fazem parte da rede. Já na figura 25 observam-se todas as tecnologias suportadas por um modelo de equipamento usado na rede da Liga de Ensino do RN. No caso é ilustrado um *switch* 3Com 2924. Visualizando a figura 25 observa-se as várias tecnologias suportadas, como HTTP, HTTPS (*Hyper Text Transfer Protocol secure*), VLAN, além de todas as versões do protocolo SNMP entre outras tecnologias.

Figura 25: Tecnologias suportadas por um switch

Bridges ▾	Manter Cadastros ▾	Logs
Cadastro de Equipamentos		
 Editar  Excluir		
ID		
3		
Marca		
3Com		
Modelo		
3Com Baseline Switch 2924-SFP Plus		
Part Number		
3CBLSG24		
Quantidade de Portas		
28		
Tecnologias Suportadas		
• HTTP - Acesso com HTTP • HTTPS - Acesso com HTTPS • IEEE 802.1Q - VLAN • RSTP - Rapid Spanning Tree Protocol • SNMPv1 - Simple Network Management Protocol Versão 1 • SNMPv2 - Simple Network Management Protocol Versão 2 • SNMPv3 - Simple Network Management Protocol Versão 3		
Fonte: O autor		

No sistema, também estão cadastrados cada switch instalado da rede de computadores. A figura 26 exibe o cadastro de *switches*, na qual cada equipamento possui, em seu cadastro, informações tais como: nome segundo a norma EIA/TIA 606, endereço IP, endereço MAC (*Media Access Control*), sala de telecomunicações onde o mesmo encontra-se instalado, além de informações referentes ao modelo do equipamento.

Figura 26: Cadastro de um switch

Bridges ▾ **Manter Cadastros** ▾ **Logs**

Cadastro de Bridges

 **Editar**  **Excluir**

ID
1

Nome
SW-01

Contato
Lineu Paiva

IPv4
10.1.1.253

MAC
00:04:96:34:A7:62

Equipamento
summit 16142
Extreme Summit X450e-24p

Sala de Telecomunicações
ST-01
Gerência de Informática FARN

Fonte: O autor

Dada a necessidade de gerenciar a infraestrutura de redes da Liga de Ensino, foram adquiridos novos equipamentos e eliminados equipamentos antigos, os quais não apresentavam suporte a SNMP e a outras tecnologias essenciais. Com os novos equipamentos adquiridos, juntamente com os *switches* gerenciáveis os quais já estavam na rede da Liga de Ensino do RN, todo o parque de concentradores da rede da Liga de Ensino passou a possuir suporte a gerenciamento de rede conforme estão dispostos na listagem do quadro 4.

Quadro 4: Modelos de concentradores novos

Marca	Modelo	PartNumber
3Com	3COM Baseline Switch 2226-SFP Plus	3CBLSF26
3Com	3Com Baseline Switch 2924-SFP Plus	3CBLSG24
3Com	3COM Switch 4500	3CR17561-91
3Com	3COMSuperStack Switch 3300	3C16985B-US
3Com	3COMSuperStack 3 Switch 3226	3CR17500-91
3Com	3COMSuperstack 3 Switch 4226T	3C17300A
3Com	3COMBaseline Plus Switch 2928	3CRBSG2893
Extreme	Extreme Summit X450e-24p	Summit 16142

Fonte: O autor

Além dos concentradores, alguns equipamentos servidores também podem passar por gerência através de SNMP. A Liga de Ensino do RN, tendo a maioria de servidores utilizando sistemas GNU/Linux, teve o suporte a SNMP ativado através da instalação da biblioteca e dos utilitários SNMPD.

5.2 IMPLANTAÇÃO DO ZABBIX

5.2.1 Instalação

Como estamos instalando o servidor do Zabbix em um sistema operacional GNU/Linux, todo o processo de instalação deve ser realizado com permissões de administrador. Uma vez estando como administrador do sistema, O próximo passo é instalar pacotes que serão utilizados para tratar as tecnologias utilizadas pelo Zabbix, evitando assim problemas durante o processo de instalação conforme é apresentado no quadro 5.

Quadro 5: Pacotes necessário a instalação

```
sudo su
apt-get install make g++ gccautoconf bzip2 ca-certificates fping
libcurl3-gnutls libgd2-xpm libiksemel3 libt1-5 libxpm4
```

Fonte: O Autor

5.2.1.1 Apache2, PHP5 e PostgreSQL

O Zabbix necessita de outros *softwares* para o seu funcionamento. Sua interface roda em um servidor *Web* com suporte à linguagem PHP5. O banco de dados utilizado será o PostgreSQL, que é atualmente um dos SGBD relacionais mais robustos e está disponível em licença TPL (The PostgreSQL License), uma licença *Open Source*. Para instalar os *softwares* executa-se os comandos ilustrados no quadro 6.

Quadro 6: *Softwares* auxiliares

```
apt-get install apache2 php5 libapache2-mod-php5 php5-gd php5-pgsql
postgresql-server-dev-8.4 postgresql libpq5 libpq-dev

apt-get install libiksemel-dev libiksemel3 libldap-2.4-2 libldap2-dev
ldap-utils libcurl4-gnutls-dev libsnmp-devsnmpsnmpd
```

Fonte: O Autor

5.2.1.2 Zabbix

Depois de instalados todos os *softwares* e dependências necessárias para que o Zabbix funcione corretamente, é realizado o *download* do Zabbix, primeiramente é criado um diretório `/sources` e em seguida é realizado o *download* do software, logo depois é descompactado o arquivo conforme é apresentado no quadro 7.

Quadro 7: Zabbix-server

```
mkdir /sources
cd /sources
wget http://server/project/zabbix/ZABBIX%20Latest%20Stable/1.8.4/zabbix-1.8.4.tar.gz
tar -zxvf zabbix-1.8.4.tar.gz
cd zabbix-1.8.4/
```

Fonte: O Autor

Neste ponto configura-se, compila-se e instala-se o código fonte do *software*. Note que não é necessário um comando “make” sem alvo. Neste ponto devem ser habilitados os recursos necessários ao servidor Zabbix, tais como suporte a SNMP e PostgreSQL. Além disso, devem ser compilados o servidor e o agente do Zabbix, conforme demonstrado no quadro 8.

Quadro 8: Compilação do *software*

```
./configure --prefix=/usr/local/zabbix --enable-agent --enable-server --with-jabber --with-libcurl --with-pgsql --with-net-snmp --with-ldap
Makeinstall
```

Fonte: O Autor

O próximo passo é preparar sua interface para a *Web*, copiando os arquivos fontes PHP para um diretório onde o servidor *Web* possa encontrá-los. Copia-se o conteúdo da pasta `frontend/php/` para `/var/www/zabbix`. Em seguida são copiados os arquivos de configuração padrão de acordo com o quadro 9.

Quadro 9: Configurar interface *Web*

```
cp -r frontends/php/ /var/www/Zabbix
cp -r misc/conf/ /usr/local/zabbix/etc
```

Fonte: O Autor

Logo após, são configurados os *scripts* de inicialização do servidor e agente Zabbix no diretório `/etc/init.d`. Para realizar algumas alterações no *script* do servidor, edita-se o arquivo através do comando **nano /etc/init.d/zabbix-server** copia-se o conteúdo, conforme Anexo A.

Após inserir os arquivos de inicialização, os mesmos devem ser configurados para iniciar automaticamente no sistema operacional, no momento do *boot*, nos níveis especificados nos próprios scripts de inicialização. Neste momento também é criado um usuário no sistema operacional para ser usado tanto pelo zabbix-server quanto pelo zabbix-agent conforme é ilustrado no quadro 10.

Quadro 10: Configurar arquivos e criar usuário para o SO

```
update-rc.dzabbix-server defaults
update-rc.dzabbix-agent defaults

adduser zabbix
```

Fonte: O Autor

Cria-se um usuário no banco de dados PostgreSQL para o Zabbix, conforme quadro 11.

Quadro 11: Criar usuário e *database* para o Zabbix

```
su postgres
psql
create user zabbix with password 'senha';
create database zabbix with owner zabbix encoding 'UTF8';
\q
```

Fonte: O Autor

A configuração de segurança do banco de dados deve ser feita, para controlar que máquinas terão acesso ao PostgreSQL, e como o usuário do banco de dados terá acesso ao banco de dados. Neste caso, o usuário do banco de dados é “zabbix” e o nome do banco de dados também é “zabbix”, com acesso local (via usuário do SO) e host (via TCP/IP) com senhas md5 conforme é ilustrado no quadro 12. Após a configuração reinicializa-se o PostgreSQL para aplicar as configurações realizadas.

Quadro 12: Configurar autenticação do PostgreSQL

```
nano /etc/postgresql/8.4/main/pg_hba.conf
localzabbixzabbix md5
hostzabbixzabbix 127.0.0.1/32 md5

/etc/init.d/postgresqlrestart
```

Fonte: O Autor

Neste ponto é criado um esquema para o banco de dados do Zabbix, e em seguida o mesmo será populado com os dados iniciais do software como esta disposto no quadro 13. Os modelos de *scripts* SQL usados são fornecidos com o Zabbix.

Quadro 13: Configurar autenticação do PostgreSQL

```
psql-U zabbixzabbix-e < create/schema/postgresql.sql
psql -U zabbixzabbix -e <create/data/data.sql
exit
```

Fonte: O Autor

No arquivo de configuração `zabbix-server.conf`, devem ser editadas as diretivas de acordo como quadro 14, que incluem a porta do servidor que o Zabbix irá escutar, bem como o usuário e a senha do banco.

Quadro 14: Configuração do Zabbix Server

```
nano /usr/local/zabbix/etc/zabbix_server.conf
ListenPort=10051
DBUser=zabbix
DBPassword=senha
```

Fonte: O Autor

O PHP necessita de algumas configurações para permitir a utilização de mais recursos pelo Zabbix. Edita-se o arquivo “`php.ini`” com os seguintes valores atribuídos nos itens destacados no quadro 15. Após essa configuração, reinicia-se o `apache2` para se aplicar as configurações realizadas. Logo após é necessário

Quadro 15: Alterar e configura parâmetros do PHP.INI

```
nano /etc/php5/apache2/php.ini
date.timezone = America/Fortaleza
max_execution_time = 300
max_input_time = 300
post_max_size = 16M

/etc/init.d/apache2 restart

chown www-data:www-data -R /var/www/zabbix/
```

Fonte: O Autor

O Zabbix gera um arquivo no qual é preciso alterar o proprietário. Devem ser configuradas permissões para o Apache poder interpretar a aplicação, principalmente no arquivo `zabbix.conf.php` conforme está disposto no quadro 15.

Por fim, utilizando-se de um navegador, acesse o Zabbix através do endereço `http://ip-do-servidor/zabbix` e realize as configurações necessárias. O usuário e senha padrão da interface são listados no quadro 16.

Quadro 16: Acessar interface do Zabbix

```
login=Admin
senha=Zabbix
```

Fonte: O Autor

5.2.2 Templates utilizados

Para que um determinado equipamento (agente) consiga fornecer dados de gerenciamento para o servidor, geralmente é necessária a existência de um *template*, que em sua essência é um molde, que será usado por equipamentos em comum. Por exemplo, se existem switches do modelo 3COM4500 a serem monitorados, então cria-se o molde para esse modelo de equipamento. Não é necessário criar um novo *template* para todos os *switches* desse modelo na rede, e sim reutilizar o *template* já pronto. Observa-se assim uma grande vantagem do uso de *templates*, poupando tempo e trabalho.

O próprio Zabbix já fornece vários *templates* em sua documentação, disponíveis para *download*, inclusive modelos do fabricante 3COM, que, conforme destacado anteriormente, domina o parque de *switches* da Liga de Ensino do RN. Porém, pode surgir a necessidade de monitorar equipamentos que ainda não possuem um *template* pronto, e assim é necessário criar um novo *template* para aquele modelo de equipamento.

No processo de criação de um novo *template*, pode-se utilizar algumas técnicas como:

- Uso de um editor de texto para edição do *template* em XML;
- Utilizar-se da sintaxe de *templates* já existentes;
- Uso do *snmpwalk* que é uma aplicação nativa do protocolo SNMP;
- Substituição, no *template*, dos atributos de SNMP pelos obtidos através do *snmpwalk*;

O *snmpwalk*, que foi utilizado para realizar buscas e obter informações da MIB do equipamento a ser monitorado, envia requisições SNMP ao equipamento. Essas informações coletadas serão analisadas e inseridas no *template*, substituindo os valores das chaves dos itens. Observa-se a importância do *snmpwalk*, visto que cada modelo de equipamento pode ter uma árvore (MIB) de informações diferente.

Em um *template* pode-se inserir diversos parâmetros e itens a serem monitorados. No *template* de um *switch*, por exemplo, pode-se monitorar itens como tráfego de entrada e de saída de uma determinada porta, bem como o status desta porta, entre vários outros itens. Nesse mesmo *template* define-se também *triggers*, além dos gráficos dos itens monitorados.

Em geral, *templates* de *switch* possuem a mesma estrutura e sintaxe. O código de um *template* criado para um *switch* é apresentado conforme o Anexo B.

Na Liga de Ensino do RN foram criados alguns *templates* para os modelos de equipamentos que não tiveram *templates* disponibilizados pela comunidade do Zabbix. O quadro 17 exibe os templates que foram obtidos com a comunidade do Zabbix e os que foram criados por ocasião deste trabalho.

Quadro 17: Templates Utilizados

Template	Modelo de Equipamento Utilizado	Origem
Template_3com_4500	3COM Switch 4500	Comunidade Zabbix
Template_3COM_2226_PLUS	3COMBaseline Switch 2226 Plus	Criado
Template_3COM_2226_SFP_PLUS	3COM Baseline Switch 2226-SFP Plus	Criado
Template_3COM_2924	3Com Baseline Switch 2924-SFP Plus	Criado
Template_3COM_4226T	3COMSuperstack 3 Switch 4226T	Criado
Template_3COM_3300	3COMSuperStack Switch 3300	Criado
Template_3COM_3226	3ComSuperStack 3 Switch 3226	Criado
Template_3COM_2928_SFP_PLUS	3Com Baseline Switch 2928-SFP Plus	Criado

Fonte: O Autor

5.2.3 Itens monitorados

Os itens que compõem um *template* podem ser monitorados (ativados) ou não. No estudo de caso da Liga de Ensino do RN, foram definidos itens de maior relevância para monitorar via gráficos e por notificações via e-mail. A seguir é apresentado cada item e sua aplicação.

- *Traffic IN*: é um item que tem como função coletar o tráfego de entrada de determinada porta do switch, ou seja, é um contador de bytes. Ao serem coletados, esses dados podem ser apresentados via gráficos e telas, configurados no Zabbix;
- *Traffic IN Errors*: neste item é coletada a quantidade de bytes com erros que trafegam entrando pela porta do switch em questão;
- *Traffic OUT*: monitora a quantidade de bytes de saída de determinada porta do switch;
- *Traffic OUT Errors*: neste item é coletada a quantidade de bytes de saída com erros que são transmitidos pela porta específica do switch.
- *IfOperStatus*: neste é possível indicar o estado operacional de uma interface.

- ICMPPING: através deste item é possível verificar se um ativo da rede, por exemplo, um *switch* ou servidor, está acessível por ICMP *ping*.
- HTTP: neste item é verificado se o HTTP, ou seja, o servidor *Web* está rodando e aceitando as conexões.

A figura 27 exibe os itens monitorados para um switch, onde é possível observar uma descrição da operação realizada para coleta dos dados e o status do item.

Figura 27: Itens monitorados de um switch

Port 26 Last Changed		ifLastChange.26	Agente SNMPv2	Inativo
Port 26 Speed		ifspeed.26	Agente SNMPv2	Inativo
Port 26 Status	Triggers (1)	ifOperStatus.3718	Agente SNMPv2	Ativo
Port 26 Traffic IN		ifInOctets.26	Agente SNMPv2	Ativo
Port 26 Traffic IN Errors		ifInErrors.26	Agente SNMPv2	Ativo
Port 26 Traffic OUT		ifOutOctets.26	Agente SNMPv2	Ativo
Port 26 Traffic OUT Errors		ifOutErrors.26	Agente SNMPv2	Ativo
sysName		sysName	Agente SNMPv2	Ativo
Verificar Disponibilidade do Switch	Triggers (1)	icmpping[{IPADDRESS},2,60,32,252]	Monitoração simples	Ativo

Fonte: O autor

5.2.4 Principais Triggers

Conforme descrito anteriormente, as *triggers* são condições de limite, baseadas no valor atual gerado por um item, sobre o qual é realizado um cálculo que define se o limite foi atingido ou não. Como resultado, uma trigger dispara uma ação de monitoramento, por exemplo, o envio de uma mensagem de *e-mail*.

Desta forma, foram definidas algumas *triggers* para acionar os administradores quando as seguintes condições de limite ocorrem.

5.2.4.1 Falha de uma porta de backbone em um switch

O item utilizado para coletar o status de uma porta é o `ifOperStatus`. No resultado da coleta desse item, é aplicada uma função para saber a diferença entre o status atual e o status anterior. Caso haja mudança de status, a *trigger* é ativada. A figura 28 mostra a expressão da *trigger*.

Figura 28: Expressão do item ifOperStatus

Trigger "Port 23 on {HOSTNAME} Status Change" ?	
Nome	Port 23 on {HOSTNAME} Status Change
Expressão (Alterar método de entrada)	{Template_3COM_2928_SFP_PLUS;ifOperStatus.23.diff(0)}=1 Adicionar
Fonte: O autor	

5.2.4.2 Perda de conectividade de um equipamento ativo de rede

Já O item utilizado para verificar a conectividade de equipamento ativo na rede é o ICMPPING. No resultado da coleta desse item, é aplicada uma função que retorna a diferença absoluta entre o valor atual e o valor anterior. Caso haja mudança de disponibilidade do equipamento, a *trigger* é ativada. A figura 29 mostra a expressão da *trigger*.

Figura 29: Expressão do item ICMPPING

Trigger "{HOSTNAME} Disponibilidade" ?	
Nome	{HOSTNAME} Disponibilidade
Expressão (Alterar método de entrada)	2928SFPPLUS:icmping[{IPADDRESS},2,60,32,252].abschange(0)=0 Adicionar
Fonte: O autor	

5.2.4.3 Parada de um servidor Web

O item utilizado para verificar a conectividade de equipamento ativo na rede é o HTTP. No resultado da coleta desse item, é aplicada uma função para saber o ultimo (mais recente) valor. Caso haja parada no servidor Web, a *trigger* é ativada. A figura 30 mostra a expressão da *trigger*.

Figura 30: Expressão do item HTTP

Trigger "Template Webservers : Estado do {HOSTNAME}" ?	
Nome	Estado do {HOSTNAME}
Expressão (Alterar método de entrada)	{Web-03:http.80.last(0)}=0 Adicionar
Fonte: O autor	

5.2.5 Telas geradas

O Zabbix, conforme foi mencionado anteriormente, disponibiliza o recurso para gerar gráficos dos itens monitorados. Esses gráficos são dispostos para o administrador em formas de telas. O quadro 18 lista todas as telas geradas para a implementação do Zabbix na Liga de Ensino do RN. Percebe-se que foi gerada uma tela para cada *switch* cadastrado.

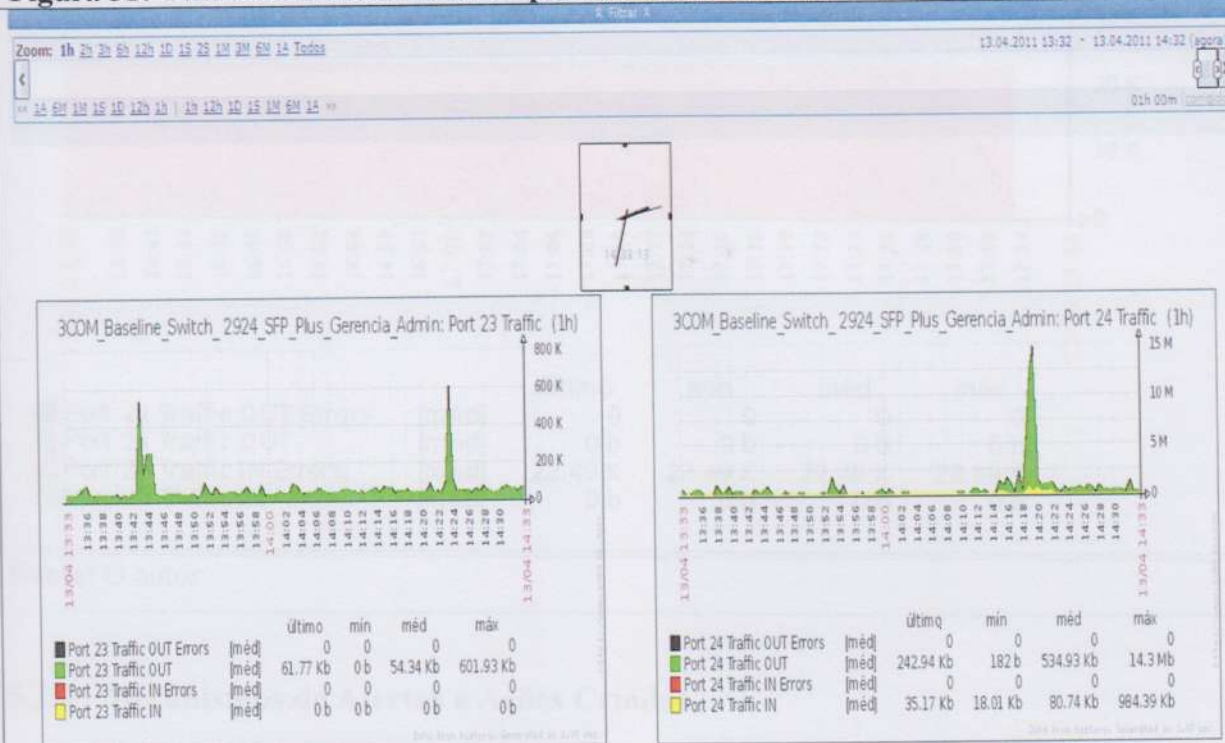
Quadro 18: Telas Geradas

Tela	Modelo de Equipamento Utilizado
Tela_Baseline_Switch_2226_Plus_Bib_Farn	3COMBaseline Switch 2226 Plus
Tela_Baseline_Switch_2226_Plus_LAB05	3COMBaseline Switch 2226 Plus
Tela_Baseline_Switch_2226_SFP_Plus_BL_Jupter	3COM Baseline Switch 2226-SFP Plus
Tela_Baseline_Switch_2226_SFP_Plus_BL_Marte	3COM Baseline Switch 2226-SFP Plus
Tela_Baseline_Switch_2226_SFP_Plus_BL_Terra1	3COM Baseline Switch 2226-SFP Plus
Tela_Baseline_Switch_2226_SFP_Plus_Clara_Camarao_Acad	3COM Baseline Switch 2226-SFP Plus
Tela_Baseline_Switch_2226_SFP_Plus_Clinicas_Admin	3COM Baseline Switch 2226-SFP Plus
Tela_Baseline_Switch_2226_SFP_Plus_Contabilidade_ED	3COM Baseline Switch 2226-SFP Plus
Tela_Baseline_Switch_2226_SFP_Plus_Coord_Cursos	3COM Baseline Switch 2226-SFP Plus
Tela_Baseline_Switch_2226_SFP_Plus_LAB02A	3COM Baseline Switch 2226-SFP Plus
Tela_Baseline_Switch_2226_SFP_Plus_LAB02B	3COM Baseline Switch 2226-SFP Plus
Tela_Baseline_Switch_2226_SFP_Plus_LAB04	3COM Baseline Switch 2226-SFP Plus
Tela_Baseline_Switch_2226_SFP_Plus_LAB05	3COM Baseline Switch 2226-SFP Plus
Tela_Baseline_Switch_2226_SFP_Plus_Professores_ED	3COM Baseline Switch 2226-SFP Plus
Tela_Baseline_Switch_2924_SFP_Plus_Gerencia_Acad	3COM Baseline Switch 2924-SFP Plus
Tela_Baseline_Switch_2924_SFP_Plus_Gerencia_Admin	3COM Baseline Switch 2924-SFP Plus
Tela_SuperStack_3_4226T_Contab_RH	3COMSuperstack 3 Switch 4226T
Tela_SuperStack_3_Switch_3226_LAB3	3COMSuperStack 3 Switch 3226
Tela_Superstack_3_Switch_4226T_LAB05	3COMSuperstack 3 Switch 4226T
Tela_SuperStack_3_Switch_4500_Sec_Geral	3COM Switch 4500
Tela_SuperStack_Switch_3300XM_ClaraCamarao_Admin	3COMSuperStack Switch 3300
Tela_SuperStack_Switch_3300_Clinicas_Acad	3COMSuperStack Switch 3300
Tela_Switch_4500_Adm_redes	3COM Switch 4500
Tela_Baseline_Switch_2226_SFP_Plus_Sec_HC	3COM Baseline Switch 2226-SFP Plus

Fonte: O autor

Em cada tela são inseridos gráficos para cada *switch* monitorado. Para cada *switch*, são inseridos gráficos, que correspondem as suas portas de *backbone*. Foi implantado um padrão de localização das portas de *backbone*, onde ficou definido que serão localizadas nas ultimas portas dos *switches*, conforme é ilustrado na figura 31.

Figura 31: Tela de monitoramento de portas do switch 2924.



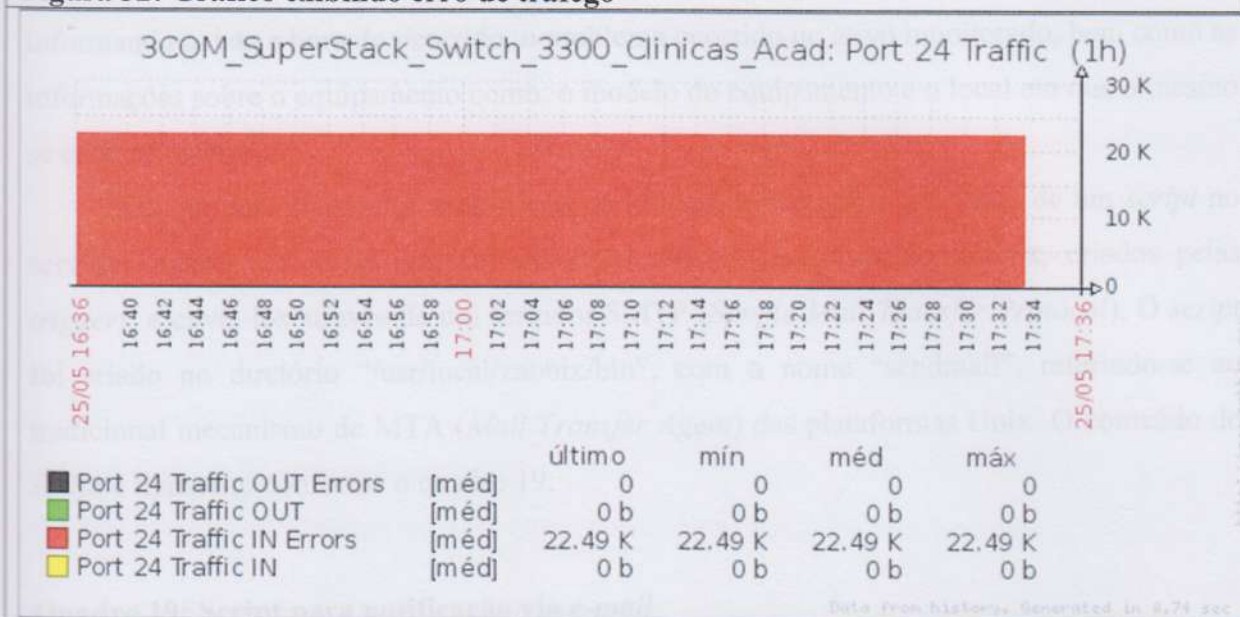
Fonte: O autor

Percebe-se, na figura 31, que as portas monitoradas do *switch* em questão são a 23 e 24, ou seja, as últimas que aquela modelo de *switch* suporta. Nota-se ainda como a criação da tela torna prática a análise e visualização dos dados coletados, que são definidos nos itens monitorados.

Com as informações visualizadas na tela, o usuário pode analisar a quantidade de tráfego que está entrando ou saindo de determinada porta, como também o tráfego com erros, seguindo uma linha de tempo. Por exemplo, se o administrador da rede quer saber a quantidade de tráfego de entrada em determinada porta no período das 14:00 às 15:00 horas, basta observar e coletar os dados do período em questão e realizar os cálculos.

Outra utilização dos gráficos exibidos nas telas é a detecção de erros. Na figura 32, é possível observar uma porta que apresentou comportamento anormal, com alta taxa de erros. A detecção desse comportamento seria bastante difícil, não fosse a verificação visual através de um gráfico.

Figura 32: Gráfico exibindo erro de tráfego



Fonte: O autor

5.2.6 Mecanismos de Alertas e Ações Criadas

A ocorrência de algumas falhas na rede devem ser alertadas ao administrador. Para isso, foram definidas algumas situações em que o administrador será avisado com alertas. Tais alertas serão enviados por *e-mail*, quando alguma das principais *triggers*, listadas no item 5.2.3, for acionada. Pode-se visualizar o conteúdo do e-mail enviado ao administrador da rede conforme figura 33.

Figura 33: Composição do e-mail de um alerta.

Data: 2011.11.03 - 06:59:30

Ocorreu uma mudança na disponibilidade do SW-34, de endereço IP

Ultimo valor de disponibilidade: 1 (PROBLEM)

value: 1

Equipamento:

3COM Baseline Switch 2226 SFP Plus

ST-21: Bloco Terra - Sala 06

Atenciosamente,

Sauron

FARN

Fonte: O autor

Conforme estar disposto na figura 33, no e-mail enviado ao administrador da rede é informando a data e hora do ocorrido, o problema ocorrido no ativo monitorado, bem como as informações sobre o equipamento como: o modelo do equipamento e o local em que o mesmo se encontra instalado.

Para que seja possível a notificação por *e-mail*, é necessária a criação de um *script* no servidor Zabbix. O *script* tem como função receber os avisos do Zabbix, criados pelas *triggers*, e enviá-los através de um servidor SMTP (*Simple Mail Transfer Protocol*). O *script* foi criado no diretório “/usr/local/zabbix/bin”, com o nome “sendmail”, referindo-se ao tradicional mecanismo de MTA (*Mail Transfer Agent*) das plataformas Unix. O conteúdo do *script* é transcrito conforme o quadro 19.

Quadro 19: Script para notificação via e-mail

```
#!/bin/bash

FROM="zabbix@farn.br"
MAILTO=$1
SUBJECT=$2
BODY=$3

function mailAviso() {
    DATA=`date +%d-%m-%Y\ %T`
    DATALOG=`date +%d-%m-%Y-%T`

    #cabecalho
    /bin/echo "to: $MAILTO" > /tmp/tempmail.txt
    /bin/echo "subject: $SUBJECT" >> /tmp/tempmail.txt
    /bin/echo -e "from: $FROM\n\n" >> /tmp/tempmail.txt

    DATALOG=`date +%d-%m-%Y-%T`

    #cabecalho
    /bin/echo "to: $MAILTO" > /tmp/tempmail.txt
    /bin/echo "subject: $SUBJECT" >> /tmp/tempmail.txt
    /bin/echo -e "from: $FROM\n\n" >> /tmp/tempmail.txt

    #corpo da mensagem
    /bin/echo "$BODY" >> /tmp/tempmail.txt
    /bin/echo -e "\n ZABBIX - FARN\n" >> /tmp/tempmail.txt

    #envia
    /bin/cat /tmp/tempmail.txt | /usr/bin/esmtp -t
    /bin/cp /tmp/tempmail.txt /tmp/tempmail.$DATALOG
    /bin/rm -rf /tmp/tempmail.txt
}

# a linha abaixo executa a funcao
mailAviso
```

Fonte: O autor

O *script* possui uma função para montagem e envio da mensagem. A função é chamada ao final do próprio *script*. Nota-se como é composta uma mensagem: o próprio servidor Zabbix (uma *trigger*) disponibiliza os três parâmetros básicos que compõem uma mensagem, que são o destino, assunto e o corpo da mensagem. No *script* eles são chamados respectivamente de *MAILTO*, *SUBJECT* E *BODY*. Uma vez recebidos esses parâmetros, os mesmos são inseridos em um arquivo chamado *tempmail.txt*, onde fica contida a mensagem que será enviada para o destinatário através de um servidor SMTP.

Na Liga de Ensino do RN, para o envio da mensagem, não foi preciso instalar nenhum servidor de *e-mail* completo junto ao Zabbix, pois a instituição já possui um serviço de *e-mail* completo. No servidor do Zabbix, foi instalado e configurado apenas o ESMTP (*Extended SMTP*), que, de acordo com os desenvolvedores, é um MTA apenas para operação de *relay*, com uma sintaxe compatível com a do *sendmail*, baseado nas bibliotecas *libESMTP* e que suporta os principais mecanismos de autenticação. O ESMTP é comumente utilizado para realizar *relay* autenticando em um servidor SMTP existente. A configuração do *software* foi realizada no arquivo “*/etc/esmtprc*”, conforme ilustrado no quadro 20.

Quadro 20: Arquivo de configuração do ESMTP

```
# Config file for ESMTP sendmail

# The SMTP host and service (port)
hostname=mail.farn.br:25

# SECURITY WARNING: Do NOT set username and password in the system wide
# configuration file unless you are the only user of this machine. See
# esmtprc(5).

# The user name
# um endereço de email para autenticar
username=meuzabbix@farn.br

# The password
# a senha do email especificado
password=minhasenha

# Whether to use Starttls
#starttls=disabled

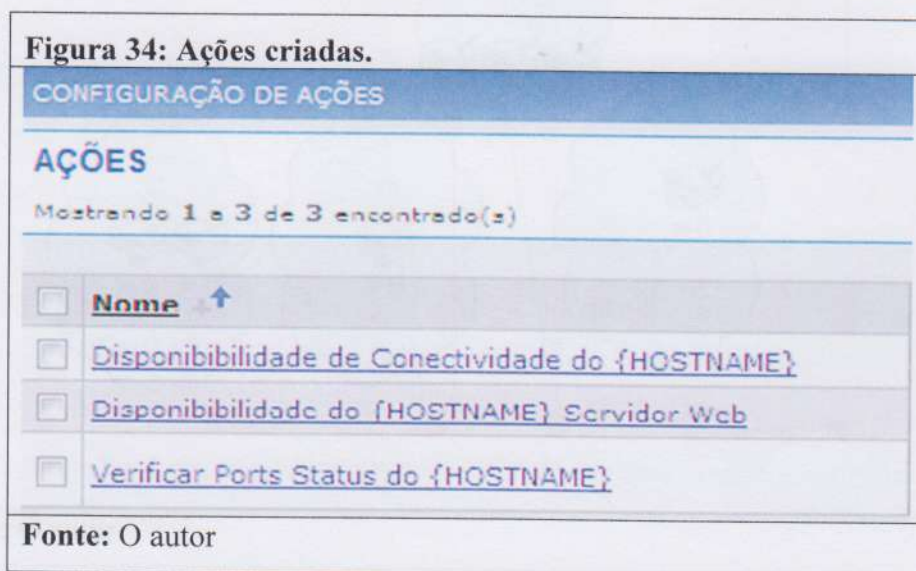
# The certificate passphrase
#certificate_passphrase=

# The Mail Delivery Agent
#mda=""
```

Fonte: O autor

No arquivo ilustrado acima, é configurado o usuário e senha do servidor ESMTP habilitando assim, o envio de *e-mails*.

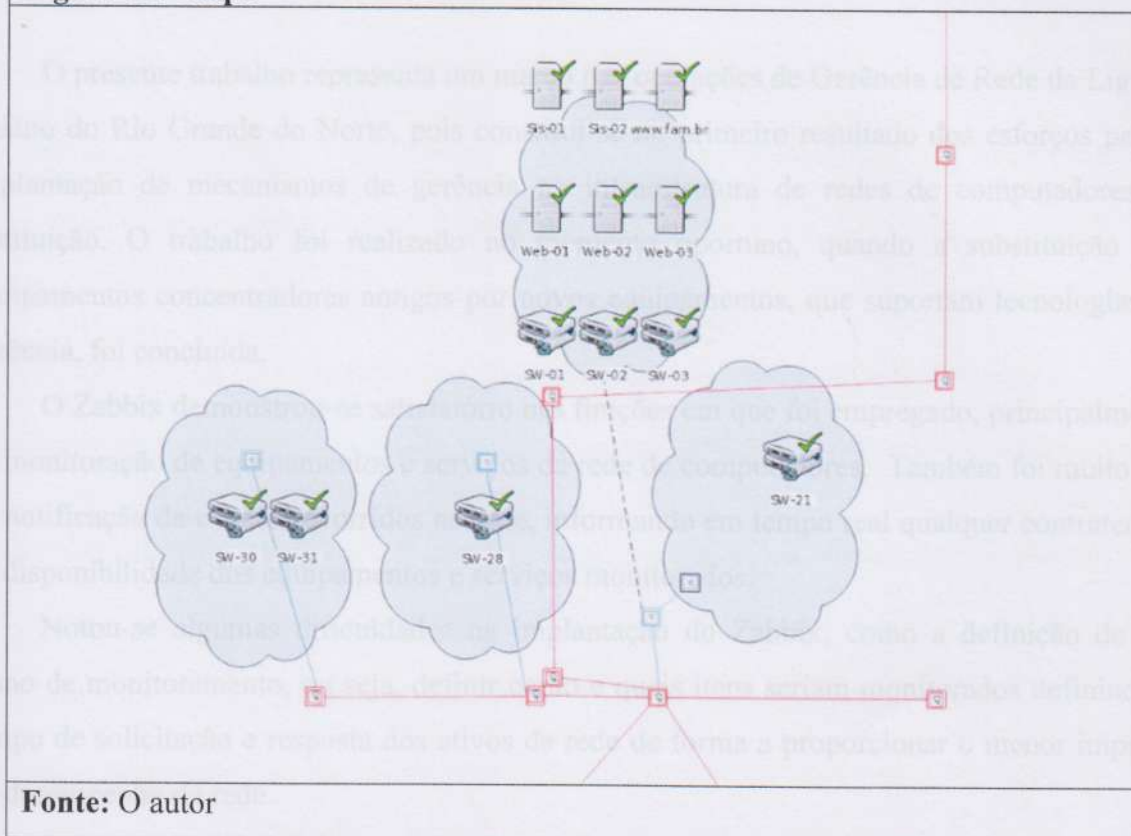
O envio de *e-mail* especificado no *script* acontecerá quando uma *trigger* for acionada, o que dispara uma Ação do Zabbix. Na figura 34 são listadas todas as ações criadas para enviar alertas ao administrador.



Na figura 34, percebe-se que foram definidas ações para alertar sobre disponibilidade de portas de *backbone* de *switches*, disponibilidade de conectividade IP de *switches* e disponibilidade de servidores *Web*. Tais casos foram considerados os casos críticos, onde é necessário que o administrador tenha conhecimento imediato do problema e providencie mais brevemente uma solução, de forma a reduzir o tempo de indisponibilidade na rede de computadores.

5.2.7 Mapa

Na liga de ensino do RN foi criado um mapa para auxiliar aos administradores de rede na monitoração e visualização de possíveis problemas que possam surgir na rede de computadores. Inicialmente foi elaborado um mapa para monitorar a principal sala de telecomunicações. No mapa estão sendo monitorados todos os *switches* e servidores *Web* que fazem parte da sala de telecomunicação além de outros equipamentos localizados no parque tecnológico da Liga de ensino conforme é ilustrada na figura 35.

Figura 35: Mapa**Fonte:** O autor

Como está disposto na figura 35, no mapa são apresentados os equipamentos e servidores Webs ligados a esses equipamentos. Nota-se que todos estão funcionando adequadamente como ilustra o servidor Zabbix inserido uma seta de cor verde.

O impacto do uso de SNMP e ICMP no monitoramento não foi considerado grande, pois, neste primeiro momento, apenas alguns itens e alguns equipamentos essenciais foram incluídos no processo de gestão. Nota-se também uma excelente melhoria na qualidade da gerenciabilidade da rede da Liga de Esportes do RN.

Por fim a partir da implementação do Zabbix foi notória a ajuda na detecção e resolução de problemas na rede da Liga de Esportes, por exemplo, detecção de problema em portas com defeito ou perda de pacotes de um switch, bem como na identificação de áreas pontuais que apresentavam perdas de pacotes constantemente.

6 CONCLUSÃO

O presente trabalho representa um marco nas operações de Gerência de Rede da Liga de Ensino do Rio Grande do Norte, pois constitui-se no primeiro resultado dos esforços para a implantação de mecanismos de gerência na infraestrutura de redes de computadores da instituição. O trabalho foi realizado no momento oportuno, quando a substituição dos equipamentos concentradores antigos por novos equipamentos, que suportam tecnologias de gerência, foi concluída.

O Zabbix demonstrou-se satisfatório nas funções em que foi empregado, principalmente na monitoração de equipamentos e serviços da rede de computadores. Também foi muito útil na notificação de eventos ocorridos na rede, informando em tempo real qualquer contratempo na disponibilidade dos equipamentos e serviços monitorados.

Notou-se algumas dificuldades na implantação do Zabbix, como a definição de um plano de monitoramento, ou seja, definir como e quais itens seriam monitorados definindo o tempo de solicitação e resposta dos ativos de rede de forma a proporcionar o menor impacto no desempenho da rede.

A rotina da administração de redes passa a incluir a vistoria diária da interface do Zabbix, para, observar e analisar o status do gerenciamento proporcionado pelo Zabbix, visando estar sempre atenta a possíveis falhas na rede de computadores da Liga de ensino do RN, garantindo maior disponibilidade de todos os recursos ligados a redes de computadores.

O impacto do uso de SNMP e ICMP nos monitoramentos não foi consideravelmente grande, pois, neste primeiro momento, apenas alguns itens e alguns equipamentos essenciais foram incluídos no processo de gerência. Notou-se também uma excelente melhora no que parte da gerenciabilidade da rede da Liga de Ensino do RN.

Por fim a partir da implantação do Zabbix foi notória a ajuda na detecção e resolução de problemas na rede da Liga de Ensino, por exemplo, detecção de problema em portas com defeitos ou perda de pacotes de um *switch*, bem como na identificação de *access points* que apresentavam perdas de pacotes constantemente.

GALLIANO FILHO, Adilson. Avaliação da Ferramenta Zabbix. 2010. Disponível em:

http://www.pgqia.puc-pr.br/~jamilhoz/RSS/TCR55019VAdilhoz_20Galliano%20-

200Arligo.pdf, Acesso em: 15 dez. 2010.

REFERÊNCIAS

- ANDRADE, G.; ZAGO, F.; VALADÃO, M. **Gerência de Redes Utilizando MRTG**. 2005. Disponível em: <http://www.inatel.br/biblioteca/component/docman/doc_download/3659-gerencia-de-redes-utilizando-mrtg>. Acesso em: 10 jan. 2011.
- BARBOSA, André S. **Sistemas de Detecção de Intrusão**. 2000. Disponível em: <<http://www.lockabit.coppe.ufrj.br/downloads/academicos/IDS.pdf>>. Acesso em: 05 jan. 2011.
- BATTISTI, Gerson. **Modelos de Gerenciamento para Infra-Estruturas de Medição de Desempenho em Redes de Computadores**. 2007. Disponível em: <SPURGEON, Charles E. **Ethernet: The Definitive Guide**. 1. ed. USA: O'Reilly & Associates, 2000. 500 p.
- BLACK, Tomas Lovis. **Comparação de Ferramentas de Gerenciamento de Redes**. 2008. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/15986/000695315.pdf?sequence=1>>. Acesso em: 30 jan. 2011.
- BONOMO, Esley. **Gerenciamento e Monitoração de Redes de Computadores Utilizando Se Zabbix**. 2006. Disponível em: <<http://www.ginix.ufla.br/files/mono-EsleyBonomo.pdf>>. Acesso em: 15 nov. 2010.
- COSTA, Felipe. **Ambiente de Rede Monitorado com Nagios e Cacti**. Rio de Janeiro: Ciência Moderna Ltda, 2008.
- GALIANO FILHO, Adilson. **Avaliação da Ferramenta Zabbix**. 2010. Disponível em: <<http://www.ppgia.pucpr.br/~jamhour/RSS/TCCRSS08B/Adilson%20Galiano%20-%20Artigo.pdf>>. Acesso em: 15 dez. 2010.

FRANÇA, Renato Augusto Gomes Pina. **Uma Aplicação Telnet para Dispositivos Móveis**. 2005. Disponível em: <<http://dsc.upe.br/~tcc/20052/RenatoFranca.pdf>>. Acesso em: 16 jan. 2011.

FLAUSINO, Caio Marcus Dias. **Proposta de Reestruturação da Rede UNIFEI Utilizando os Atuais Paradigmas da Comunicação de Dados**. 2004. Disponível em: <<http://adm-net-a.unifei.edu.br/phl/pdf/0030990.pdf>>. Acesso em: 02 nov. 2010.

LOPES, Raquel V.; SAUVÉ, Jacques P.; NICOLLETTI, Pedro S. **Melhores Práticas para Gerência de Redes de Computadores**. Rio de Janeiro: Campus, 2003.

MAURO, Douglas R; SCHMIDT Kevin J. **SNMP Essencial**. Rio de Janeiro: Campus, 2001.

MELLO, Jorge Lucas De. **Protótipo De Um Agente SNMP para uma Rede Local Utilizando a Plataforma JDMK**. 2000. Disponível em: <<http://www.inf.furb.br/~pericas/orientacoes/JDMK2000.pdf>>. Acesso em: 22 jan. 2011.

MENEZES, Elionildo da Silva; SILVA, Pedro Luciano Leite. **Gerenciamento de Redes: Estudos de protocolos**. 1998. Disponível em: <<http://www.di.ufpe.br/~flash/ais98/gerrede/gerrede.html>>. Acesso em: 05 nov. 2010.

OETIKER, Tobias, **RRDtool Gallery**. 2011. Disponível em: <<http://oss.oetiker.ch/rrdtool/gallery/index.en.html>>. Acesso em: 24 nov. 2011.

OPPENHEIMER, Priscilla. **Projeto de Redes Top-Down: Um enfoque de análise de sistemas para o projeto de redes empresariais**. 1. ed. Rio de Janeiro: Campus, 1999. 492 p.

OLUPS, Rihards. **Zabbix 1.8 Network Monitoring**. 1. ed. Birmingham: Packt Publishing Ltd, 2010. 428 p.

RFC 792 - **Internet Control Message Protocol (ICMP)**. Disponível em: <
<http://www.faqs.org/rfcs/rfc792.html>>. Acesso em 15 jan. 2011.

RFC 1157 - **Simple Network Management Protocol (SNMP)**. Disponível em: <
<http://www.faqs.org/rfcs/rfc1157.html>>. Acesso em 25 jan. 2011.

SOLHA, Liliana Esther Velásquez Alegre. **Os Logs como Ferramenta de Detecção de Intrusão**. Boletim bimestral sobre tecnologia de redes, v. 3, n 3, 1999. Disponível em:
<<http://www.rnp.br/newsgen/9905/logs.html>>. Acesso em: 14 jan. 2011.

SOUZA, Renata Santos de. **Desenvolvimento de uma aplicação para prover o uso de ferramentas de medição ativa**. 2005 Disponível em: <
http://projetos.inf.ufsc.br/arquivos_projetos/projeto_387/Projeto-1.pdf>. Acesso em: 25 abr. 2011.

STANGE, Rodrigo. **Ferramenta para Gerenciamento de Falhas em Rede Ethernet Baseada em Protocolo SNMP**. 2008. Disponível em: <
<http://www.inf.furb.br/~pericas/orientacoes/GerenciaSNMP2008.pdf>>. Acesso em: 26 jan. 2011.

TANENBAUM, Andrew S. **Redes de Computadores**. 4. ed. Rio de Janeiro: Elsevier Editora Ltda, 2003. 945p.

KUROSE, James F; ROSS, Keith W. **Redes de Computadores e a Internet: Uma abordagem top-down**. 3. ed. São Paulo: Pearson Addison Wesley, 2006. 634p.

ANEXO A - CONFIGURAÇÃO DO SERVIDOR ZABBIX

```

#!/bin/bash
### BEGIN INIT INFO
# Provides:          zabbix-server
# Required-Start:    $all
# Required-Stop:
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: 0 servidor Zabbix.
# Description:       0 servidor Zabbix.
### END INIT INFO
#
# Zabbix daemon start/stop script.
#
# Written by Alexei Vladishev <alexei.vladishev@zabbix.com>.

NAME=zabbix_server
PATH=/bin:/usr/bin:/sbin:/usr/sbin:/usr/local/zabbix/bin:/usr/local/zabbix/
sbin
DAEMON=/usr/local/zabbix/sbin/${NAME}
CONFIGFILE=/usr/local/zabbix/etc/${NAME}.conf
DESC="Zabbix server daemon"
PID=/var/tmp/${NAME}.pid

test -f $DAEMON || exit 0

set -e

case "$1" in
    start)
        echo "Starting $DESC: $NAME"
        start-stop-daemon --oknodo --exec $DAEMON --pidfile $PID --start --
-c $CONFIGFILE
        ;;
    stop)
        echo "Stopping $DESC: $NAME"
        start-stop-daemon --oknodo --exec $DAEMON --pidfile $PID --stop --
-c $CONFIGFILE
        ;;
    restart|force-reload)
        #
        # If the "reload" option is implemented, move the "force-reload"
        # option to the "reload" entry above. If not, "force-reload" is
        # just the same as "restart".
        #
        echo -n "Restarting $DESC: zabbix_server"
        $0 stop
        $0 start
        #
        start-stop-daemon --stop --quiet --pidfile \
            /tmp/${NAME}.pid --user zabbix --exec $DAEMON
        #
        sleep 1
        #
        start-stop-daemon --start --quiet --pidfile \
            /tmp/${NAME}.pid --user zabbix --exec $DAEMON
        #
        echo "$NAME."
        ;;
    *)

```

```
    N=/etc/init.d/$NAME
    # echo "Usage: $N {start|stop|restart|force-reload}" >&2
echo "Usage: $N {start|stop|restart|force-reload}" >&2
    exit 1
;;
esac

exit 0
```

ANEXO B – COMPOSIÇÃO DE UM TEMPLATE ASSOCIADO A UM SWITCH

Ao analisar o código abaixo percebe-se como um *template* é composto, incluindo informações simples como o nome do *template*, qual grupo ele pertence, como também a definição de *triggers*, itens monitorados e gráficos gerados.

```
<?xml version="1.0" encoding="UTF-8"?>
<zabbix_export version="1.0" date="03.11.10" time="14.45">
  <hosts>
    <host name="Template_3COM_4500">
      <proxy_hostid>0</proxy_hostid>
      <useip>1</useip>
      <dns></dns>
      <ip>127.0.0.1</ip>
      <port>10050</port>
      <status>3</status>
      <useipmi>0</useipmi>
      <ipmi_ip>127.0.0.1</ipmi_ip>
      <ipmi_port>623</ipmi_port>
      <ipmi_authtype>0</ipmi_authtype>
      <ipmi_privilege>2</ipmi_privilege>
      <ipmi_username></ipmi_username>
      <ipmi_password></ipmi_password>
    </host>
  </hosts>
  <groups>
    <group>Templates</group>
  </groups>
  <triggers>
    <trigger>
      <description>Port 25 on {HOSTNAME} Status Change</description>
      <type>0</type>expression>{Template_3COM_4500:ifOperStatus.125.diff(0)}=1</e
xpression>
      <url></url>
      <status>1</status>
      <priority>3</priority>
      <comments></comments>
    </trigger>
  </triggers>
  <items>
    <item type="4" key="ifInOctets.113" value_type="3">
      <description>Port 13 Traffic IN</description>
```

```
<ipmi_sensor></ipmi_sensor>
<delay>30</delay>
<history>90</history>
<trends>365</trends>
<status>0</status>
<data_type>0</data_type>
<units>b</units>
<multiplier>0</multiplier>
<delta>2</delta>
<formula>1</formula>
<lastlogsize>0</lastlogsize>
<logtimefmt></logtimefmt>
<delay_flex></delay_flex>
<authtype>0</authtype>
<username></username>
<password></password>
<publickey></publickey>
<privatekey></privatekey>
<params></params>
<trapper_hosts>localhost</trapper_hosts>
<snmp_community>public</snmp_community>
<snmp_oid>ifInOctets.4227722</snmp_oid>
<snmp_port>161</snmp_port>
<snmpv3_securityname></snmpv3_securityname>
<snmpv3_securitylevel>0</snmpv3_securitylevel>
<snmpv3_authpassphrase></snmpv3_authpassphrase>
<snmpv3_privpassphrase></snmpv3_privpassphrase>
<applications/>
</item>
<graph name="Port 20 Traffic" width="900" height="200">
<ymin_type>0</ymin_type>
<ymin_type>0</ymin_type>
<ymin_item_key></ymin_item_key>
<ymin_item_key></ymin_item_key>
<show_work_period>1</show_work_period>
<show_triggers>1</show_triggers>
<graphtype>1</graphtype>
<yaxismin>0.0000</yaxismin>
<yaxismax>100.0000</yaxismax>
<show_legend>0</show_legend>
<show_3d>0</show_3d>
```

```
<percent_left>0.0000</percent_left>
<percent_right>0.0000</percent_right>
<graph_elements>
  <graph_element item="Template_3COM_4500:ifInOctets.120">
    <drawtype>1</drawtype>
    <sortorder>0</sortorder>
    <color>FFFF00</color>
    <yaxisside>1</yaxisside>
    <calc_fnc>2</calc_fnc>
    <type>0</type>
    <periods_cnt>5</periods_cnt>
  </graph_element>
  <graph_element item="Template_3COM_4500:ifInErrors.120">
    <drawtype>1</drawtype>
    <sortorder>1</sortorder>
    <color>FF0000</color>
    <yaxisside>1</yaxisside>
    <calc_fnc>2</calc_fnc>
    <type>0</type>
    <periods_cnt>5</periods_cnt>
  </graph_element>
  <graph_element item="Template_3COM_4500:ifOutOctets.120">
    <drawtype>1</drawtype>
    <sortorder>2</sortorder>
    <color>00FF00</color>
    <yaxisside>1</yaxisside>
    <calc_fnc>2</calc_fnc>
    <type>0</type>
    <periods_cnt>5</periods_cnt>
  </graph_element>
  <graph_element item="Template_3COM_4500:ifOutErrors.120">
    <drawtype>1</drawtype>
    <sortorder>3</sortorder>
    <color>000000</color>
    <yaxisside>1</yaxisside>
    <calc_fnc>2</calc_fnc>
    <type>0</type>
    <periods_cnt>5</periods_cnt>
  </graph_element>
</graph_elements>
</graph>
```

```
</graphs>  
<macros/>  
</host>  
</hosts>  
<dependencies/>  
</zabbix_export>
```