Gerenciador Simplificado de Redes Baseado no Protocolo SNMP

Eduardo Henrique Albuquerque da Silva*
Instituto Federal de Educação, Ciência e Tecnologia da Bahia.
Rua Emídio dos Santos, S/N.
Barbalho, Salvador, Bahia.
eduardohenrique@ifba.edu.br

Antonio Carlos dos Santos Souza†
Instituto Federal de Educação, Ciência e Tecnologia da Bahia.
Rua Emídio dos Santos, S/N.
Barbalho, Salvador, Bahia.
antoniocarlos@ifba.edu.br

RESUMO

Este artigo visa descrever a implementação e as funcionalidades de um Gerenciador de redes, que utiliza o protocolo SNMP como padrão. São abordados premissas do gerenciamento de redes, assim como diversas tecnologias. São citados sistemas que possuem o mesmo propósito, afim de justificar a presença ou a falta de elementos no Gerenciador proposto.

Palavras-Chave

Gerencia de Redes, SNMP, RMON, SDN, OpenFLow, Cacti, Nagios, Zabbix

1. INTRODUÇÃO

A evolução tecnológica sempre foi uma necessidade, ainda mais quando se trata de tecnologias envolvidas com computação. Durante a década de 1970, conceitos básicos de redes eram estabelecidos, havendo uma rápida evolução que levou a uma redução nos custos de recursos computacionais, engajando assim uma imersão de diversos segmentos da sociedade às redes de computadores. As redes de computadores deixaram de lado a configuração simples de redes pequenas e separadas para formarem grandes redes interconectadas. A dificuldade e complexidade que existia para gerenciar as redes era diretamente proporcional ao tamanho delas [1].

Mudanças continuaram a aparecer, como exemplo do surgimento de novos tipos de serviços oferecidos, variando entre correios eletrônicos, transferência de arquivos, aplicações multimídia e a própria Internet. Tais serviços aumentam consideravelmente a complexidade das redes, complexidade essa que é influenciada também pela existência de uma variedade de elementos como, padrões, sistemas operacionais e equipamentos.

Assim novas questões surgiram: qual seria a maneira mais adequada de se localizar e corrigir problemas de comunicação entre redes, além de administrar seus elementos e recursos de uma forma simples, flexível e extensível, potencializando sua eficiência e produtividade? Em resposta a essa demanda surgiram os protocolos de gerenciamento de redes que normalmente operam na camada de aplicação do TCP/IP.

Quanto aos protocolos de gerenciamento de redes, o Protocolo Simples de Gerenciamento de Rede (SNMP - do inglês *Simple Network Management Protocol*) certamente é o primeiro a ser lembrado. Foi criado por um grupo de engenheiros que almejavam contornar as arquiteturas proprietárias com uma solução aberta, simples e que veio a se tornar um padrão para o gerenciamento de redes [2].

O ambiente se tornou propício ao surgimento de softwares que conseguem se comunicar com os protocolos de gerenciamento. Esses softwares são os Gerenciadores, também chamados de NMSs, *Network Management Systems, Stations* ou *Softwares*. Extraem informações para que os interessados pela rede estejam cientes da situação e possam tomar alguma atitude, ou que o próprio software execute ações a depender das condições, algo que se tornou possível com a evolução das tecnologias de gerenciamento.

Um padrão foi estabelecido, atualmente muitos equipamentos que conseguem se conectar à Internet já possuem as configurações para o protocolo SNMP. Softwares como Zabbix, Cacti e Nagios [1], conseguem gerenciar os elementos inseridos na rede, extraindo bastante informações. Contudo, por serem softwares abrangentes acabam por precisar de mais recursos, seja para aprender a manusear as diversas funcionalidades do software, seja para deixar o ambiente configurado.

Um ambiente de gerenciamento de redes é preparado com algumas soluções tecnológicas de hardware e software, cujo objetivo é evitar a indisponibilidade de serviços computaci-

*Graduando em Análise e Desenvolvimento de Sistemas.
†Doutor em Ciência da Computação e Professor do Curso de Análise e Desenvolvimento de Sistemas.

[1]Sistemas para gerenciamento de redes muito utilizados. Serão discutidos na sessão de tecnologias correlatas

onais e a perda de dados importantes para o negócio. Essa indisponibilidade deve prever paradas por erros ou falhas, mas também paradas por sobrecarga nos equipamentos [3]. E o fator determinante do tipo de equipamento e de tecnologia a serem utilizados nesse ambiente é o nível de criticidade do negócio e da operação.

É imprescindível incluir nesse escopo os equipamentos elétricos, como nobreaks - equipamentos com uma bateria que permite o funcionamento do computador quando a fonte de energia principal é cortada, classificado como UPS (*Uninterruptible Power Supply*)[4] - e outros *hosts* (elementos monitorados na rede) de menor porte que, por serem configuráveis em rede e monitoráveis, possibilitam os serviços de gerenciamento também dessa categoria. Um equipamento crítico e imprescindível em muitos ambientes profissionais, ao qual geralmente os nobreaks são conectados é o servidor. E se o foco principal desse monitoramento fosse esses equipamentos imprescindíveis?

Com base nesse cenário, o presente trabalho visa descrever um software focado em gerenciamento de rede específico para nobreaks e servidores, representado através de uma interface web, elemento que está presente em diversos outros softwares, se tornando uma característica comum dos gerenciadores de rede. Diferente dos outros sistemas citados, que possuem maior complexidade para configurar e utilizar, a solução proposta traz simplicidade às questões de usabilidade e configuração. Apresenta também controle de acesso, onde o usuário deve fazer o login, sendo redirecionado para a próxima página com os elementos referentes ao seu nível de acesso. Apesar de ser voltado para equipamentos mais críticos, como nobreaks e servidores, o software reconhece normalmente outros tipos de *hosts*.

2. REFERENCIAL TEÓRICO

Neste tópico serão apresentadas tecnologias que somam ao Gerenciador proposto, e demonstram utilidade para o gerenciamento de redes em si. Além dos custos necessários para sua utilização.

2.1 RMON

O RMON (do inglês *Remote Monitoring*) [5] faz parte de um grupo aberto denominado IETF (*Internet Engineering Task Force*) [6], sendo assim, não é uma solução proprietária. O IETF identifica, propõe soluções para problemas e padronizam tecnologias e protocolos relacionados a utilização da internet. Contrariando a impressão natural sobre a tecnologia RMON, não veio para substituir o SNMP e sim para incrementá-lo, tornando NMSs que utilizam SNMP compatíveis com RMON também.

O RMON se apresenta como um complemento para o protocolo SNMP, surgindo como uma Base de Informação de Gerenciamento (MIB - do inglês *Management Information Base*), que atua como um índice de objetos, apontando os elementos que podem ser gerenciados. A MIB será melhor explicada posteriormente. O SNMP, por ser simples em sua forma de identificação e notificação de falhas, deu margem à necessidade da criação de meios para uma extração mais completa de informações da rede monitorada [7].

Tal necessidade foi suprida pelo RMON, que conta com: capacidade de identificar precisamente causas de falhas na rede monitorada, assim como a severidade dessa falha; interoperabilidade independentemente de fabricante (o que é importante, já que por não ser uma tecnologia proprietária, sua implementação varia de acordo com o fabricante); a então nova tecnologia também deveria oferecer ferramentas adequadas para diagnóstico da rede. Além de um meio para alertar o administrador dos eventuais problemas da rede, junto à métodos automáticos capazes de coletar dados a respeito desses problemas.

2.1.1 Objetivos do RMON

O RMON foi um dos primeiros, dentre as tecnologias de gerenciamento de redes, a ter como objetivo o gerenciamento proativo, tendo o trabalho de gerenciamento simplificado e a resolução dos problemas facilitados, aumentando assim a disponibilidade da rede e a queda dos custos de manutenção. Abaixo, segue com detalhes os objetivos almejados pelo RMON, que são citados segundo a própria documentação oficial [5].

1. Operações Offline
   Há algumas condições onde um software gerenciador pode vir a perder o contato com seu dispositivo de monitoramento remoto. Isso pode ser proveniente de uma tentativa de diminuir custos de comunicação, quando em uma rede em estado precário por exemplo, ou por acidente ocasionado por falha na rede, o que acaba por prejudicar a comunicação entre o software controlador e a sonda presente no dispositivo monitorado. Para contornar esse tipo de situação, as sondas RMON podem ser configuradas para realizar diagnósticos da rede e coletar estatísticas continuamente. A sonda realiza tentativas de notificar o software gerenciador quando oportuno, assim, mesmo se a comunicação com o software gerenciador estiver ruim ou inexistir, as informações de falha, desempenho e configuração podem ser continuamente acumuladas e comunicadas.

2. Monitoramento proativo
   Os recursos disponíveis nos dispositivos monitorados são potencialmente úteis para executar continuamente diagnósticos e armazenar logs de performance. Como o dispositivo está sempre disponíveis no surgimento de alguma falha, ele pode notificar falhas aos softwares gerenciadores e podem manter um histórico estatístico de informações sobre falhas. Tal histórico pode ser recuperado pelo software gerenciador para prover, adicionalmente, diagnósticos dos problemas.

3. Detecção e reportes de problemas
   A sonda pode ser configurada para reconhecer condições que levam a erros, checando continuamente se tais condições estão se formando e quando uma dessas condições ocorrer, o evento pode ser registrado em log e o software gerenciador notificado de formas diversas.

4. Dados de valor agregado
   Como um dispositivo de monitoramento remoto está localizado diretamente na parte monitorada da rede, ele tem a oportunidade realizar verificações, melhorando significativamente o trabalho do software de gerenciamento pois já recebe as informações analisadas.

Por exemplo, ao destacar os *hosts* na rede que geram mais tráfego ou erros, a sonda pode dar ao software de gerenciamento informações precisas de que necessita para resolver problemas.

5. Múltiplos Gerenciadores
   Possuir um ambiente com múltiplos gerenciadores é importante para prover melhor recuperação em caso de desastres. As sondas de monitoramento remoto de rede terão que interagir com uma variedade de softwares gerenciadores, provavelmente usando recursos de forma concorrente. Apesar das vantagens, o RMON disponibiliza acesso somente à algumas camadas de baixo nível na rede, atuando limitadamente até a subcamada de Controle de Acesso ao Meio (MAC - do inglês *Media Access Control*), localizada na camada de Enlace.

Com o RMON, o tráfego da LAN qual o dispositivo gerenciado está ligado é monitorado por uma sonda, que pode obter os dados transeuntes da subcamada MAC sem interferir no tráfego, assim acessando os endereços MAC de origem e de destino, fornecendo informações detalhadas sobre o tráfego de quadros enviados e recebidos por cada dispositivo na LAN [8]. Logo após a adoção do RMON, os usuários dessa tecnologia queriam uma gama de informações gerenciáveis maior do que as fornecidas pela camada de Enlace.

2.1.2 RMON2

Uma das principais limitações do RMON ocorre quando há um roteador ligado à rede, não é possível identificar a origem ou o destino dos dados, sendo caracterizado a ineficiência no que se trata de uma rede WAN, pois seria necessário trabalhar com a camada 3 (camada física) em diante do modelo OSI. Os gerentes de rede queriam rastrear protocolos pertencentes a camadas de mais alto nível e as sessões baseadas nesses protocolos para saber quais aplicações estavam usando quais protocolos e a que custo sobre a largura de banda rede disponível. Portanto, uma nova versão do RMON, o RMON2, foi criada para prover capacidades mais avançadas. Com o RMON2, as sondas utilizadas para monitoramento da rede conseguem operar com protocolos localizados acima do nível do enlace [8], permitindo a leitura do cabeçalho do protocolo da camada de rede, normalmente o protocolo IP. Com isso, é viabilizado a análise do tráfego que passa pelo roteador para determinar a fonte e destino reais dos quadros.

Com esta capacidade, o gerente de rede pode identificar soluções para questionamentos além da fronteira do roteador, questionamentos como: quais sub redes ou estações são responsáveis por uma eventual sobrecarga em um roteador, além de ser importante descobrir para onde esse tráfego está se dirigindo, caso seja de saída, ou de onde ele está partindo; caso essa grande quantidade de dados chegue por um roteador e deixe a rede através de outro roteador, é interessante saber quais estações ou redes são responsáveis pelo volume do tráfego. Com tais informações, o gerente de rede poderá fazer um melhor planejamento para solucionar o problema e consequentemente melhorar o desempenho da rede, tendo como exemplo, uma melhor distribuição de sistemas que estão demandando ou encaminhando muitos dados da rede ao acessar um servidor, visando otimizar o fluxo do tráfego.

Uma sonda RMON2 tem a capacidade de monitorar e decodificar protocolos do nível da camada de aplicação, levando em consideração que o RMON trata como aplicação tudo que vem acima da camada de transporte. Ao realizar averiguações além da camada de rede, consegue ler e decodificar protocolos de níveis superiores além de verificar os cabeçalhos dos protocolos do nível de aplicação [8]. Dessa forma, é permitido ao gerente de rede monitorar o tráfego com um alto grau de detalhamento. Utilizando o RMON2, um software de gerenciamento de rede pode gerar gráficos que apresentem informações do tráfego por protocolo, ou por aplicações, viabilizando totalmente a otimização da carga da rede e a manutenção do seu desempenho.

2.2 Protocolo SNMP

O SNMP, *Simple Network Management Protocol*, é o protocolo de gerenciamento de redes padrão na Internet, atualmente permitindo que uma ou mais máquinas sejam designadas como gerentes de rede. Estas máquinas recebem informações de todos os outros equipamentos conectados à rede, aos quais chamamos de *hosts*. As informações que são extraídas por meio dos agentes SNMP são processadas tornando possível a realização do gerenciamento e do monitoramento, por parte do sistema gerenciador, que tem como consequência a detecção de problemas ocorridos.

Como explanado previamente, antes do surgimento do protocolo SNMP no cenário das redes de computadores soluções proprietárias eram majoritariamente utilizadas. Pela década de 1980 o protocolo TCP/IP já encontrava sua posição de destaque a medida em que cada vez mais redes de computadores o utilizava como base. Mesmo com a ascensão o TCP/IP tinha desvantagem, já que muitas das arquiteturas proprietárias possuíam soluções para gerenciamento da rede, sendo a partir desta necessidade que o SNMP surgiu [2].

Assim como o RMON, o protocolo SNMP também é mantido pelo grupo IETF, sendo um protocolo da camada de aplicação cuja função é facilitar a troca de informações de gerenciamento entre os dispositivos de rede, e tem sido escolhido para o monitoramento de redes, pois apesar de ser simples por conta da pouca exigência de recurso do software nos equipamentos monitorados, apresenta grande poder para resolução de problemas complexos ocorridos em ambientes de redes heterogêneas.

Exatamente por exigir menos dos agentes monitorados, as atividades mais complexas de processamento e as funções de armazenamento ficam sob a responsabilidade do sistema gerenciador.

Os principais objetivos do protocolo SNMP são [2]:

- Reduzir o custo da construção de um agente que suporte o protocolo;
- Reduzir o tráfego de mensagens de gerenciamento necessárias para gerenciar os recursos da rede;
- Reduzir o número de restrições impostas às ferramentas de gerenciamento da rede, devido ao uso de operações complexas e pouco flexíveis;

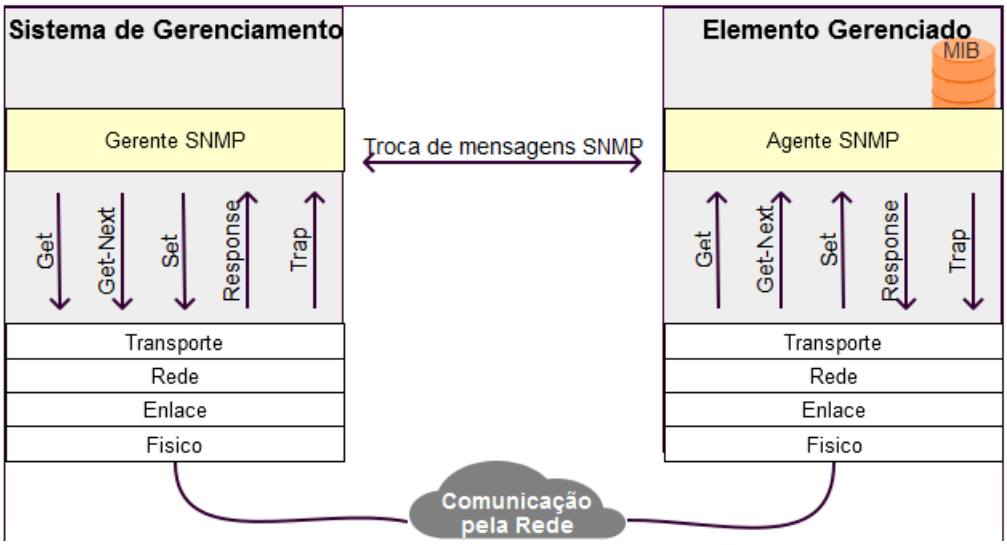

Figura 1: Visão geral do funcionamento da gerência com SNMP - Adaptada [9]

- Apresentar operações de simples entendimento, sendo facilmente usadas pelos desenvolvedores de ferramentas de gerenciamento;
- Permitir facilmente a introdução de novas características e novos objetos não previstos ao se definir o protocolo;
- Construir uma arquitetura que seja independente de detalhes, relevantes a somente a algumas implementações particulares.

A Figura 1 ilustra a simplicidade no gerenciamento pelo SNMP, tornando explicito a independência entre a aplicação gerenciadora e o objeto monitorado.

2.2.1 *Componentes do SNMP*

Para utilizar o protocolo SNMP corretamente e conseguir extrair seu máximo, é necessário conhecer como o mesmo funciona e seus componentes. Já foram citados os agentes, os *hosts* e os gerenciadores. Além destes serão detalhados outros componentes que completam a arquitetura SNMP.

Gerenciador: Comumente acessíveis através de um navegador e localizadas em máquinas principais da rede como servidores, sendo normalmente utilizados diretamente pelos administradores da rede. É a entidade responsável pela coleta e tratamento das informações dos elementos inseridos à rede, podendo também enviar informações e requisições para esses elementos.

O recebimento de informações por parte da entidade gerenciadora pode ser feito por requisições para os Agentes, por exemplo através de comandos aplicados pelo administrador da rede, onde os Agentes por sua vez se comunicam com os elementos monitorados. As informações também podem ser obtidas por uma comunicação previamente programadas, seja por parte do gerenciador, onde a informação é requisitada, ou por parte do Agente, onde a informação é simplesmente enviada. As entidades gerenciadoras são conhecidas como NMSs (*Network Management Systems, Network Management Stations* ou *Network Management Softwares*).

Agentes: Presente nos dispositivos alvo que são gerenciados pelo NMS, sendo responsáveis pela manutenção das informações de gerência da máquina. A comunicação entre o Agente e o Gerenciador ocorre por meio das operações SNMP, que serão discutidas posteriormente. Como discorrido antes, a comunicação pode ocorrer pela requisição de informações por parte do Gerenciador, criando assim um *polling* de comunicação (será detalhado posteriormente), ou pela interceptação das mensagens que foram enviadas pelo Agente quando alcançado o valor de determinado item monitorado, valor este que foi definido pelo NMS [2].

Os Agente são processos que exigem muito pouco do dispositivo que ele convive. Isso é essencial para que o desempenho do dispositivo seja minimamente afetado, deixando o processamento e tratamento das informações com o NMS. Para realizar o monitoramento das informações da máquina o Agente faz uso das chamadas de sistema, e para realizar o controle das informações da máquina utiliza-se de RPC (*Remote Procedure Call*) [10].

Dispositivo Gerenciado: A rede que está sendo gerenciada possui elementos (ou nós). Esses elementos tem suas informações extraídas ou alteradas pelo Agente SNMP quando houver uma solicitação para isso. É preciso que o Agente tenha permissão para acessas os dados que lhe foram requisitados. São os *hosts* monitorados.

***Structure of Management Information* (SMI):** É uma estrutura que demonstra como a MIB deve ser construída. Através do SMI, são especificados os identificadores, tipos de dados associados, a nomeação e representação (sintaxe mais definição) dos recursos da MIB [11].

O SMI tem o propósito de trazer a tona a simplicidade e a capacidade de extensão das MIBs. Por uma questão de padronização, não são suportados dados com estruturas

complexas, evitando inconsistências entre os tipos de dados de diferentes equipamentos, evitando também complicações com os provedores das MIBS - a empresa que quer que seu equipamento seja monitorado, por exemplo.

No conceito de monitoramento gerente/agente, as entidades monitoradas devem estar acessíveis fisicamente e logicamente. Ser acessível fisicamente remete a verificação que algum elemento da rede deve fazer, possivelmente o Agente, identificando o endereço e quantificando a informação de gestão de rede. Ser acessível logicamente remete ao armazenamento da informação de gestão de rede, sendo está informação recuperável e modificável(o SNMP executa a recuperação e modificação da informação) [9].

O SMI padroniza uma a definição da estrutura de uma MIB, de cada objeto dessa MIB - incluindo sintaxe e valor do objeto, além de fornecer uma técnica padrão para codificação desses valores do objeto. de modo que o acesso lógico a informação deste objeto seja possível.

***Management Information Base* (MIB):** De uma forma simplificada, é possivel dizer que a MIB é a definição dos objetos gerenciados que utiliza a sintaxe SMI. A MIB é estruturada como mostrado na Figura 2 É uma hierarquia estruturada em uma árvore. Cada elemento dessa árvore contém um identificador do objeto, o chamado OID (do inglês *Object Identifier*). Na prática, os OIDs representam as variáveis que podem ter seus valores lidos ou escritos.

Normalmente um Agente já vem com uma MIB especifica, a MIB-II. Com essa MIB pode-se obter as informações padrões do TCP/IP, como número de pacotes transmitidos ou o estado da interface [9].

Outros objetos podem ser definidos para uma MIB através de extensões privadas adicionadas no nó *private* - seguindo a Figura 2, é o OID 1.3.6.1.4. Também pode-se substituir o nó da árvore que possui os objetos da MIB existente ou um outro nó implementando as alterações desejadas (acrescentando ou removendo um objeto).

Os OIDs apresentados na árvore são únicos. Como mostrado com o exemplo do nó *private*, para se chegar ao objeto que deseja, deve-se percorrer a árvore toda [3]. Na prática pode-se utilizar o nome, o numero, ou até mesmo as duas formas misturadas, para percorrer a árvore. Utilizando o mesmo exemplo, poderia ser: iso.org.dod.internet.private.

2.2.2 Versões SNMP

A primeira versão do SNMP não se preocupou muito com segurança ou autenticação para acesso aos dados. Talvez porque tenha surgido como solução temporária para padronizar o gerenciamento de redes em arquiteturas baseadas em TCP/IP. A autenticação é constituída por comunidades, que não são nada mais que palavras chaves que funcionam como uma senha transmitida integralmente pela rede, sem nenhuma criptografia. Qualquer aplicação que utiliza SNMP que saiba a palavra chave pode ter acesso aos dados dos dispositivos monitorados, além do mais, basta interceptar alguma mensagem para saber qual a comunidade [9]. Com esta primeira versão não existe comunicação entre gerentes.

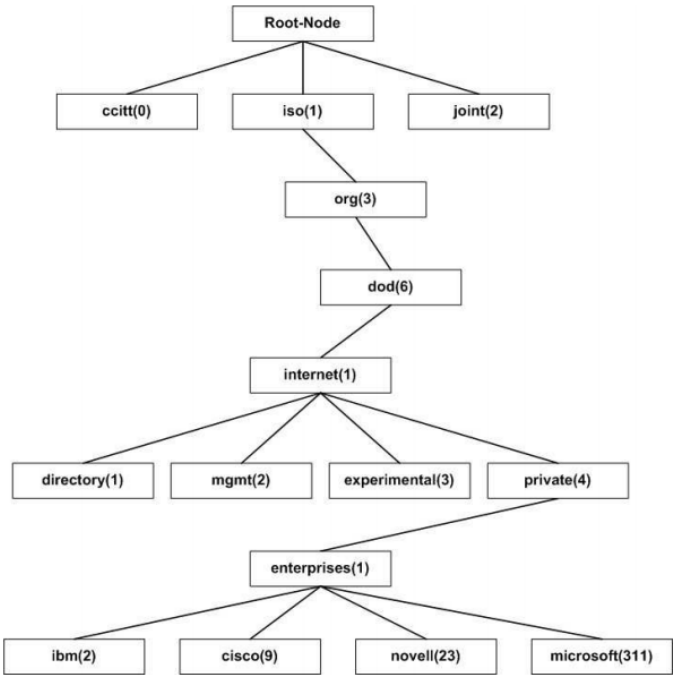

Figura 2: Estrutura de uma MIB [3].

O SNMP foi atualizado, ganhando novas versões, sempre com incremento de suas funcionalidades. Essas versões coexistem, ou seja, é possível enviar uma mensagem utilizando a segunda versão e momentos depois utilizando a terceira, por exemplo.

A segunda versão, tecnicamente identificada como SNMPv2c (*community string-based SNMPv2*), possui autenticação pelas comunidades ainda, além de manter o problema da transmissão dos dados descriptografados pela rede, contudo, adiciona algumas definições para incrementar a segurança, novas operações para melhoria de performance (as operações serão explanadas mais adiante), além de habilitar também a comunicação entre gerentes [3].

Com a terceira versão, SNMPv3, a melhoria crucial ficou por conta do problema que se estendia desde a primeira versão, a falha na segurança, atingindo então o nível esperado sem sem deixar de lado a simplicidade do protocolo. A arquitetura da nova versão proporciona uma maior extensibilidade, agilizando assim o desenvolvimento da rede sobre a supervisão do SNMP.

As novas características de segurança englobam autenticação de usuário, autorização e controle de acesso, nomenclatura de entidades, definição de pessoas e politicas - facilitando o agrupamento de usuários e a definição de permissões para o determinado grupo. É inserido também relacionamento através de proxy, onde um Agente pode se comunicar com *hosts* que originalmente não implementam SNMP. O Agente proxy recebe o pedido do NMS e então converte para um formato que o *host* entenda, possivelmente utilizando outro protocolo [13]. O mesmo ocorre com as respostas direcionadas ao NMS. O funcionamento do proxy é detalhado na

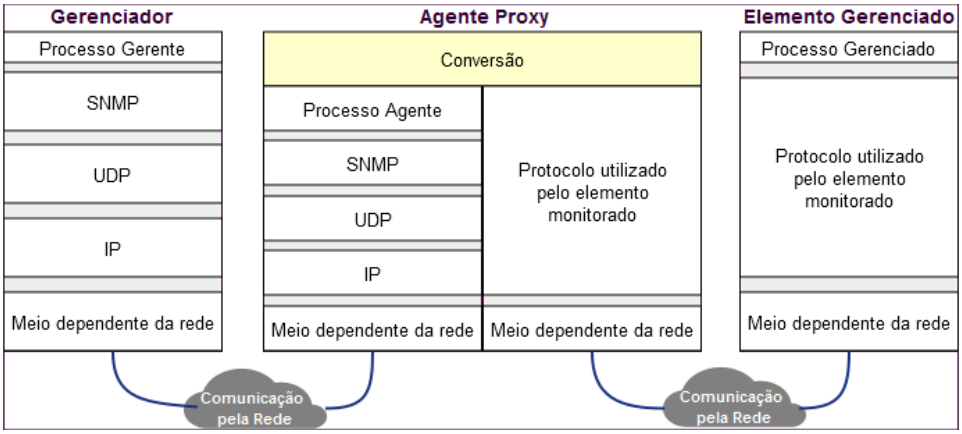

Figura 3: Funcionamento de um agente proxy - Adaptada [12]

Figura 3.

É concedido enorme flexibilidade ao administrador da rede, incluindo a possibilidade de configuração remota dos parâmetros, de forma distribuída, mas é preciso ter cautela e configurar corretamente os controles de acesso para manter a segurança.

2.2.3 SNMP e UDP

O protocolo SNMP, como todo protocolo da camada de aplicação, precisa utilizar algum protocolo da camada de transporte para acessar as camadas de nível baixo, vide modelo OSI [14]. O SNMP utiliza o UDP (*User Data Protocol*), escolha que trouxe muitas vantagens para o funcionamento do protocolo, mas também desfavorecendo no quesito garantia de entrega dos dados.

São utilizadas duas diferentes portas para efetivar a comunicação. Pela porta 161 é estabelecido o *polling* de comunicação do Gerenciador para o Agente. Como o conceito de *polling* sugere, existe a vantagem de manter o Gerenciador como elemento principal da comunicação, servindo como canal para as requisições feitas aos Agentes. Essas requisições podem ocorrer em intervalos regulares de forma automatizada. Ainda por esse *polling*, transita a resposta do Agente à requisição feita, ainda utilizando a mesma porta. Há também a porta 162, para quando a comunicação for iniciada pelo Agente, normalmente enviando uma *trap* para alertar o Gerenciador de alterações e eventualidades em itens do *hosts* monitorado.

A Figura 4 detalha a comunicação, deixando mais claro a vantagem da utilização de duas portas, onde requisições e *traps* podem ser feitas simultaneamente e sem maiores problemas de bloqueio.

A principal razão pelo uso do protocolo UDP é por causa do seu menor impacto na rede, possuindo menos sobrecarga se comparado ao TCP. Como o SNMP é utilizado para descobrir problemas numa rede, se funcionasse sobre o TCP haveria maiores complicações, visto que o TCP inundaria a rede com retransmissões em busca do que procura. É característica do TCP, sendo orientado a conexão, garantir a entrega dos dados ao seu destino, se tornando necessário es-

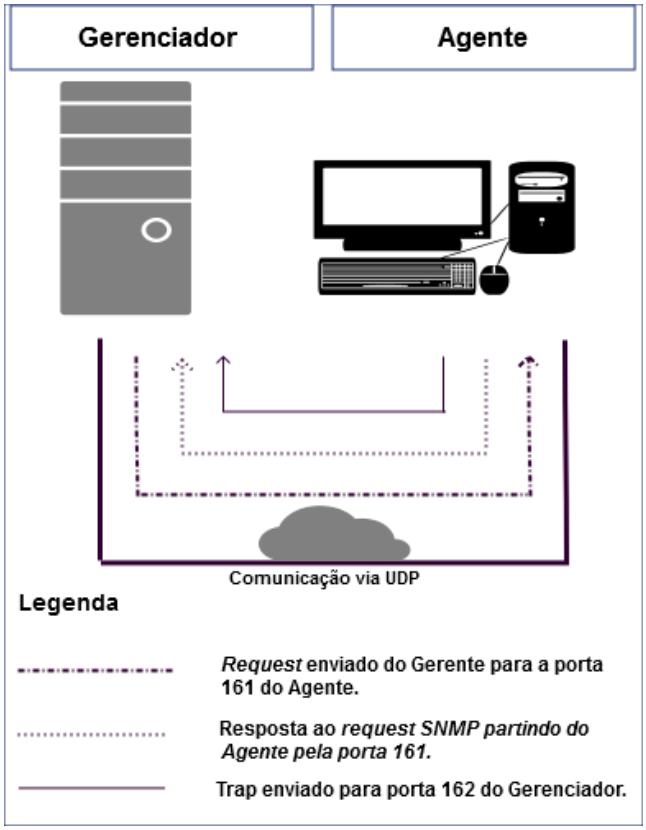

Figura 4: Comunicação SNMP detalhada - Adaptada [3]

tabelecer, manter e cancelar conexões entre as respectivas partes - Agente e Gerente.

Para amenizar o problema da confirmação da entrega dos dados na própria camada de aplicação pode-se usar o *time-out* como referência. Durante algum tempo o NMS espera a resposta do Agente ao ser feito uma requisição, se atingir o limite do tempo significa que houve perda, sendo a mensagem possivelmente comprometida e reenviada pelo NMS.

Normalmente a quantidade de tempo que o NMS espera e a quantidade de vezes que os dados são retransmitidos podem ser configuráveis [9].

Quando a mensagem parte do gerenciador para o Agente, não há tanto problema, pois o gerenciador pode enviar a mensagem novamente, contudo, se a mensagem partir do Agente para o NMS e houver perda de pacotes então o cenário é propício para maiores complicações, já que não há como o gerente saber que iria receber algum dado.

2.2.4 Operações SNMP

O SNMP inclui um conjunto de comandos de gestão e respostas que permitem efetuar operações de gerenciamento. Maior parte das operações SNMP são executadas pela entidade gerenciadora. As operações executadas pelos Agentess são apenas operações de notificação. Uma operação SNMP compreende a execução de um ou mais comandos SNMP que podem ou não ter respostas associadas [9].

Uma mensagem que representa a operação SNMP deve definir o servidor do qual obtemos ou alteramos os atributos dos objetos, e que será responsável por converter as operações requisitadas em operações sobre as estruturas de dados locais. Essa mensagem pode ter sido enviada pelo gerente da rede, por exemplo. Após verificar os campos da mensagem, o servidor deve usar as estruturas internas disponíveis para interpretá-la e enviar a resposta da operação de volta para quem requisitou o pedido, ou seja o cliente.

Uma mensagem é constituída por três partes principais [2]: A versão do protocolo; A identificação da comunidade, usada para permitir que um cliente acesse os objetos gerenciados através de um servidor SNMP; E por fim a área de dados, que é dividida em unidades de dados de protocolo (Protocol Data Units - PDUs). Cada PDU é constituída ou por um pedido do cliente, ou por uma resposta de um pedido (enviada pelo servidor). Ao utilizarmos a suíte de ferramentas para SNMP, o Net-SNMP [15] uma mensagem para resgatar uma informação ficaria com a seguinte estrutura:

```
snmpget -v1 -cpublic localhost iso.3.6.1.2.1.1.4.0
```

O SNMP, pela sua característica fundamental, é um protocolo de requisição/resposta simples. Seguindo o padrão da baixa complexidade, inicialmente as seguinte operações são definidas no SNMP:

- GetRequest - Mensagem que é enviada da entidade gerenciadora ao agente, permitindo recuperar o valor de objetos MIB do *host* em questão.
- GetNextRequest - Também direcionada do gerenciador para o agente, a mensagem recupera a próxima informação disponível depois de uma ou mais informações solicitadas provenientes dos objetos da MIB.
- SetRequest - Solicitação da entidade gerenciadora ao agente, para que seja permitido a modificação dos valores de objetos MIB em um *host*.
- GetResponse - Mensagem enviada pelo agente ao gerente, informando o valor de uma variável que lhe foi solicitado, também sendo utilizado para responder solicitações SetRequest.
- Trap - Usado pelo agente para informar à entidade gerenciadora algum evento importante sobre o *host*.
- GetBulkRequest - Introduzida com o SNMPv2, o GetBulkRequest é uma versão melhorada do GetNextRequest, onde em um pedido é possível realizar varias solicitações. Retorna apenas um Response com os valores das variáveis solicitadas.
- InformRequest - Também introduzida com o SNMPv2, possui o mesmo formato que a Trap do SNMPv2. Foi criado para solucionar o problema da confirmação, por parte do Gerenciador, do recebimento da *trap*. Através desse response, um Agente vai saber se deve reenviar a operação para o Gerente.

3. TECNOLOGIAS CORRELATAS

Nesta parte do trabalho serão descritos algumas características de softwares que também possuem a função de gerenciamento de redes. Alguns serviram de base para criação da aplicação proposta. Também será discorrido sobre tecnologias alternativas para o gerenciamento de redes.

3.1 SDN e OpenFlow

Redes Definidas por Software (SDN - do inglês *Software Defined Networking* [16]) é um paradigma que está cada vez mais popular. A ideia é desassociar o plano do controle do plano dos dados, mandando o primeiro para o software controlador/gerenciador. Antes de ser associado ao SDN, o OpenFlow [17] foi proposto como uma ferramenta para permitir inovação na rede. Os benefícios de usar OpenFlow permitiriam aos pesquisadores rodarem experimentos em hardware real sem interferir no tráfego.

3.1.1 Motivação para o SDN

Segundo Adrian Lara [18], após a sua utilização para o SDN, o OpenFlow acaba por ter um incremento em seu propósito, servindo para padronizar a comunicação entre switches e o software controlador, contemplando a arquitetura SDN. O OpenFlow é mantido e gerenciado pelo *Open Networking Foundation* (ONF), que identificou a dificuldade da comunidade de pesquisa em rede para realizar testes de novas ideias com os hardwares atuais. Tal dificuldade provém principalmente da forma como o código que roda dentro dos switches é disposto, não podendo ser modificado sem autorização do proprietário, além da própria disposição da infraestrutura da rede, estando "ossificada", tornando complicado testes de novas ideias de rede com configurações de tráfego realistas.

Ocasionalmente, a manutenção ou alteração em equipamentos proprietários requer um profissional especializado, que é um fator que muitas vezes inviabiliza tal iniciativa. Com um rede definida por software, é possível moldar o tráfego alterando as regras de redirecionamento, sem entrar em contato direto com o equipamento.

O destaque do SDN é a possibilidade da reconfiguração de toda a rede, acompanhando a demanda de serviços e utilização da rede. A motivação do SDN é retirar a complexidade do hardware e permitir flexibilidade e inovação no software.

Em resultante, o hardware é favorecido, tendendo a se tornar mais simplificado, focando somente na realização do encaminhamento do tráfego. Como contraposição, o software se torna responsável por gerenciar a rede e a forma como os dispositivos encaminham o tráfego, sendo capaz de alterar as regras de encaminhamento que antes eram fixas nos dispositivos. SDN trouxe a tona capacidades únicas que permitem diversas formas para o gerenciamento de redes.

3.1.2 Composição SDN-OpenFlow

Algumas características podem ser citadas para esclarecer a relação entre SDN e OpenFlow. O SDN consiste em desacoplar o controle dos dados em si, enquanto o OpenFlow descreve como um software controlador e um switch devem estabelecer comunicação para atender ao formato do SDN. O SDN dá ao usuário uma forma de abstração do estado de toda a rede e o OpenFlow abstrai os componentes da rede. O OpenFlow pode ser encarado como uma especificação quando se trata de padronizar switches. É encarado como uma arquitetura se visto sob o contexto de uma rede inteira.

Uma arquitetura SDN-OpenFlow consiste de três componentes principais: um switch compatível com OpenFlow e um software controlador que são intermediados pelo terceiro elemento, um canal seguro. Esse é o ponto onde o protocolo OpenFlow entra, lidando com o formato do encaminhamento das mensagens passadas entre o plano de controle e o switch OpenFlow. Uma tendência comum para as diversificadas formas de implantação do OpenFlow é explorar a forma de atualizar dinamicamente as regras de encaminhamento dos pacotes. Possuir um gerenciador sensível à rede possibilita ao software controlador encaminhar de forma dinâmica o tráfego, ajustando-se às necessidades [18].

3.1.3 Funcionamento do paradigma SDN

Como citado anteriormente, a composição de uma rede SDN consiste de uma coleção de switches gerenciados por um programa executado em um controlador. Cada switch tem uma tabela de fluxo que armazena uma lista de regras para processamento de pacotes. Cada regra consiste de um padrão (associado ao cabeçalho do pacote) e ações sobre os pacotes (tais como *forwarding, dropping, flooding, modifying,* ou envio do pacote para o controlador). Para cada regra, o switch mantém contadores do tráfego que medem os bytes e pacotes tratados até o momento.

Quando o pacote chega, um switch seleciona a regra de prioridade máxima, atualiza os contadores, e executa a(s) ação(ões) específica(s). Se não houverem regras correspondentes, o switch envia o cabeçalho do pacote para o controlador e espera uma resposta sobre quais ações tomar. Se toda vez que receber um pacote, o switch precisar se comunicar com o controlador para esperar uma decisão, então certamente haverá problemas. Switches também enviam mensagens de eventos, como *joins* para se juntar a uma rede ou *port change* quando links saem ou entram no ar [19].

O controlador OpenFlow pode instalar ou desinstalar regras nos switches, ler estatísticas de tráfego e responder eventos. Para cada evento, o programa controlador define um manipulador que pode instalar regras ou pedidos de emissão de estatísticas do tráfego. O canal OpenFlow é o meio que

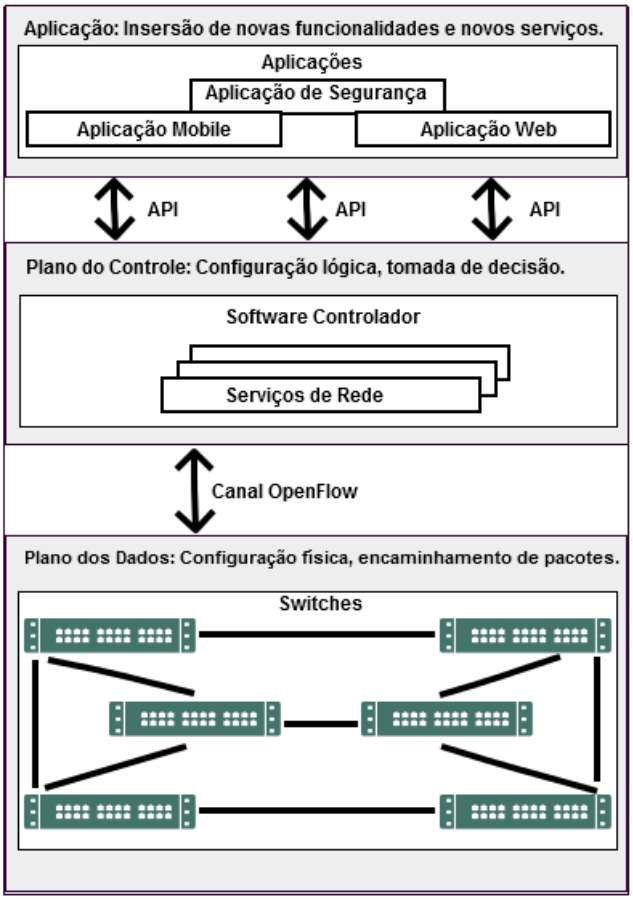

Figura 5: Arquitetura SDN-OpenFlow - Adaptada [16]

conecta cada switch OpenFlow a um controlador. Através dessa interface, o controlador configura e gerencia o switch, recebe eventos, e envia pacotes para o switch. Entre o caminho de dados (*datapath*) e o canal OpenFlow a interface é específica da implementação, contudo, todo canal de mensagens OpenFlow deve ser ajustado de acordo com o protocolo OpenFlow. O canal OpenFlow é normalmente encriptado com o TLS (*Transport Layer Security*), mas pode ser executado diretamente sobre o TCP.

O protocolo OpenFlow suporta três tipos de mensagens, sendo elas Controlador/Switch, Assíncronas e Simétricas, possuindo cada uma sub tipos de mensagens. Mensagens do tipo Controlador para Switch são iniciadas pelo controlador e usadas para gerenciar ou inspecionar o estado do switch. Mensagens Assíncronas são iniciadas pelos switches e usadas para atualizar o programa controlador sobre eventos da rede e mudanças do estado do switch. Já as mensagens Simétricas podem ser iniciadas tanto pelo switch ou pelo controlador e o fazem sem solicitação [20].

A expectativa sobre essa nova tecnologia é alta, havendo muito potencial para pesquisa e para aplicação no mercado de trabalho, contudo, é necessário muito esforço para por em prática, tornando inviável em inúmeros cenários.

3.2 Nagios

Nagios é sem dúvida uma das ferramentas NMS mais populares do mercado. Distribuído sob a licença GNU/GPL (Licença Pública Geral) [21], é um monitor de redes cujo foco encontra-se na construção de um núcleo com capacidade de monitoração de alta performance, flexível e escalável. Com a utilização de plugins o mesmo pode ser estendido a um gerenciador de redes [2]. Por ser um projeto open source permitiu o surgimento de outras ferramentas baseadas na sua arquitetura. As empresas também podem criar versões customizadas do Nagios com o objetivo de satisfazer as suas necessidades específicas.

Com Nagios é possível gerir e analisar cada um dos recursos disponibilizados por um *host*, como tempo de resposta, carga de processamento, uso de memória, entre outros. Seu ambiente web permite a geração tanto de relatórios e gráficos online quanto permite criar mapas de redes. Além disso existe a possibilidade de elaborar um grupo de usuários configurados para receber mensagens e alertas do sistema referente a distúrbios na rede e aos *hosts* monitorados. Possui como principais características: capacidade de monitorar serviços de redes (SMTP, POP3, HTTP, NNTP, entre outros); capacidade de monitorar recursos de *hosts*; poder enviar alertas por diversos meios, tais como e-mail, SMS; possibilita gerar relatórios e integração com SGBD's; Possui uma poderosa interface web e um singular sistema de notificações [22].

Infelizmente a solução não é multiplataforma tendo como requisito básico uma máquina com S.O. GNU/Linux ou Unix-like instalado; provavelmente o código precisará de várias modificações para rodar em outros sistemas. Vale lembrar que a restrição é apenas para instalar o Nagios e isso não impede o monitoramento de máquinas com Windows [22]. A sua instalação não apresenta grande complexidade para quem já possui alguma familiaridade com os comandos do terminal linux; seria interessante se o mesmo disponibilizasse um instalador simplificado baseado numa interface gráfica. Após a instalação faz-se necessário instalar os plugins, setar as configurações do servidor web e as do Nagios propriamente dito.

Diferentemente de outras ferramentas de monitoramento o Nagios não possui nenhum mecanismo interno de checagem de status dos *hosts* ou dos serviços disponíveis na rede. Entretanto o Nagios utiliza-se de plugins que realizam todo este trabalho pesado [22], os quais podem ser executáveis compilados ou scripts (Pearl, shell, etc). Ser "scriptável" também pode ser considerada uma qualidade do Nagios, pois através dos mesmos podemos expandir suas funcionalidades sem a necessidade de recompilar o código fonte, podendo efetuar a mudança em run time.

Há a facilidade de parametrização pelo próprio usuário dos serviços de rede que se deseja monitorar. É importante relatar que para qualquer incidente observado existe a possibilidade, desde que devidamente configurado, do envio de alarmes e notificações para tomada de ações corretivas e suporte. Uma das principais funcionalidades que fazem o Nagios tão flexível é a habilidade de utilizar macros na sua definição de comandos [22]. Contatos são definidos para que, em caso de evento, estes sejam avisados via e-mail ou sms, previamente cadastrados, sobre a situação/status de algum(uns) *host(s)* em geral ou específicos. As macros permitem referenciar informações de *hosts*, serviços e outros fontes em seus comandos. A documentação é bastante completa abrangendo vários aspectos da arquitetura do software, abrangendo instalação, configuração, segurança, tópicos básicos e avançados.

3.2.1 Configuração

Assim como a instalação, as configurações também requerem algum nível de aptidão com o shell.

1. CGI.CFG - Configura os parâmetros de autorização e utilização da interface web;
2. CHECKCOMMANDS.CFG - Arquivo onde é possível configurar os plugins;
3. MISCCOMMANDS.CFG - Específica algumas features, a exemplo envio de emails;
4. NAGIOS.CFG - Configurações principais;
5. RESOURCE.CFG - Define alguns parâmetro tais como o class path dos plugins, entre outros;
6. TIMEPERIODS.CFG - Agendamento de configurável para checagem de serviços e/ou servidores.

3.2.2 Considerações de Segurança

A caixa de monitoramento pode ser considerada uma *backdoor* nos outros sistemas monitorados da rede. Na maioria dos casos isso se justifica pela necessidade de obtenção de informações, o Nagios deve ter acesso através do firewall de modo a monitorar servidores remotos. Efetuar este tipo de monitoramento sempre requer um certo nível de confiança, no entanto isto apresenta a um potencial atacante uma *backdoor* atrativa para uma potencial invasão. Um invasor pode ter acesso rápido ao sistema caso ele comprometa o servidor de monitoramento primeiro. Isto mostra-se particularmente verdadeiro se o sistema em questão estiver utilizando chaves de segurança SSH compartilhadas para monitoramento.

Caso um intruso possua a habilidade de enviar resultados de consultas ou comandos externos ao serviço do Nagios ele poderá enviar dados de monitoramento falsos que pode criar eventos para execução de scripts. Outro ponto de vulnerabilidade é a possibilidade do tráfego de monitoramento de dados ser interceptado (sniff). Havendo ausência de criptografia um atacante pode ter acesso a informações valiosas do seu sistema. Um atacante pode capturar o tráfego de informações e determinar o melhor momento de comprometer o sistema sem ser detectado.

3.3 Cacti

É uma solução completa de gráficos de redes cuja arquitetura foi projetada de modo a aproveitar-se do poder das ferramentas de visualização e armazenamento de dados da RRDTool [23], a qual armazena toda a informação necessária para plotar os gráficos e os popula com os dados oriundos de um SGBD MySQL, assim como prover flexibilidade, adaptando-se a diversas situações, robustez e usabilidade [2].

A RRDTool foi criado por Tobias Oetiker, constituindo-se em um sistema de base de dados fazendo uso do algoritmo

de escalonamento Round Robin. Primitivamente seu desenvolvimento possuía por propósito armazenar uma série de dados numéricos correspondente aos estados de uma rede, entretanto pode ser utilizado para guardar quaisquer séries numéricas tais como carga de cpu, temperatura, uso de memória, entre outros[2]. Tal característica possibilita ao Cacti gerar uma série de gráficos referente a uso de memória, quantidade de processos, espaço em disco, entre outros.

Ao contrário do Nagios, o Cacti é multiplataforma podendo ser instalado em computadores com SO Windows. Ele pode ter acesso a gráficos de qualquer elemento da rede que seja compatível com o protocolo SNMP, porém possui suporte a configuração de outros protocolos além do SNMP. Analogamente ao Nagios, a arquitetura da ferramenta também possui suporte ao uso de plugins. Em comparação ao Nagios, falta um tutorial de instalação mais completo, bem como um tópico específico relacionado a segurança, o pode ser considerado pontos passíveis de melhoria. É distribuído como software open source, licenciado sob a GNU/GPL.

3.3.1 *Princípio de funcionamento*

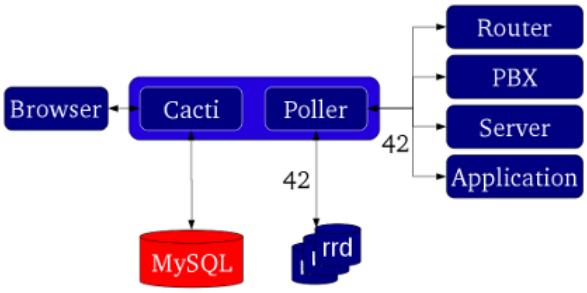

Figura 6: Disposição da infraestrutura do Cacti. [24]

Tanto administração quanto uso são feitos através do browser. A definição de novos gráficos e templates são registrados no Cacti Web Pages. Todos os dados administrativos serão armazenados em tabelas do MySQL. No mesmo servidor encontra-se o *poller*, o qual possui por função consultar todos os sistemas alvos, aqueles que suportam SNMP (router, PBX, aplicações). Os resultados das consultas aos alvos são estocados em arquivos RRD, os quais são utilizados pelo Cacti para gerar os gráficos [24]. A Figura 6 mostra como o Cacti é estruturado.

A operação do Cacti pode ser dividida em três tarefas:

1. Recuperação de dados - É feito por meio do *poller*, o qual é executado por meio do agendador do sistema operacional - um exemplo seria o crontab dos sistemas Unix-like;
2. Armazenamento de dados - Utiliza o RRDToll para armazenar dados. RRD é uma sigla para Round Robin Database. Este, foi idealizado para exibir e armazenar dados de séries temporais. Seu sistema de armazenagem é extremamente compacto e executa algumas tarefas específicas como consolidar dados brutos com dados já consolidados;
3. Apresentação de dados - Uma das features mais apreciadas do RRDToll é a função embutida de geração de gráficos.

3.3.2 *Templates*

Um template é um modelo, que por meio de uma estrutura pré-definida agiliza a criação de novos modelos. São similares as macros, utilizadas pelo Nagios, e as sub rotinas utilizadas em linguagem de programação, porém diferentemente delas não podem ser executados em tempo de execução. São definidos os seguintes templates :

1. Para gráficos - Caso possua diversos gráficos com características similares usar um template faz completo sentido. Como desvantagem desta abordagem qualquer modificação feita no template será propagada a todos os gráficos que o utilizam;
2. De dados - Análogo ao de gŕaficos com uma diferença marcante: as mudanças não são propagados aos arquivos RRD pré-existentes.
3. Hosts Templates - Possui um propósito diferente dos anteriores. Ao invés de abstrair os campos de um *host*, pode-se associar modelos de gráficos e consulta de dados a um *host* específico. Desta forma ao atribuir um template a um *host* todos os tipos de gráficos relevantes para este tipo de *host* estão a apenas um clique do usuário (*user friendly*). Ao criar um novo modelo as modificações são tidas em conta apenas para novos *hosts*, porém é possível aplicá-las aos dispositivos pré existentes.

O Cacti apresenta como ponto negativo o fato de não possuir um agente de descoberta automática, isto é, a rede precisa ser adicionada manualmente, tornando o trabalho do administrador demasiadamente mais complicado quando a rede for grande. Apesar da existência de plugins de terceiros que auxiliam nesse processo, o Cacti ainda não apresenta essa funcionalidade de forma nativa. Apesar desta dificuldade, o sistema é considerado poderoso e altamente escalável à medida em que possibilita o monitoramento de qualquer parâmetro mensurável em hardware [2]. O Cacti também conta com uma comunidade bastante ativa de usuários

3.4 Zabbix

Software criado por Alexei Vladishev e é atualmente desenvolvido e mantido pela Zabbix SIA, sendo Open Source GPLv2, apenas com uma versão que é considerada Enterprise. Foi projetado para monitorar em tempo real a disponibilidade e desempenho de componentes de infraestrutura de TI, através de uma interface web, de onde é possível visualizar todos dispositivos e seus respectivos status. Com Zabbix é possível reunir uma quantidade enorme de tipos de dados da rede, junto ao monitoramento em tempo real de alto desempenho onde diversos computadores, servidores, máquinas virtuais e variados dispositivos de rede podem ser monitorados simultaneamente [25].

Os dados são extraídos dos seus agentes, presentes nos *hosts* monitorados, contudo, assim como o Nagios, o Zabbix também pode ser "scriptável", possibilitando a extração de informações customizadas que seu agente não é capaz de obter

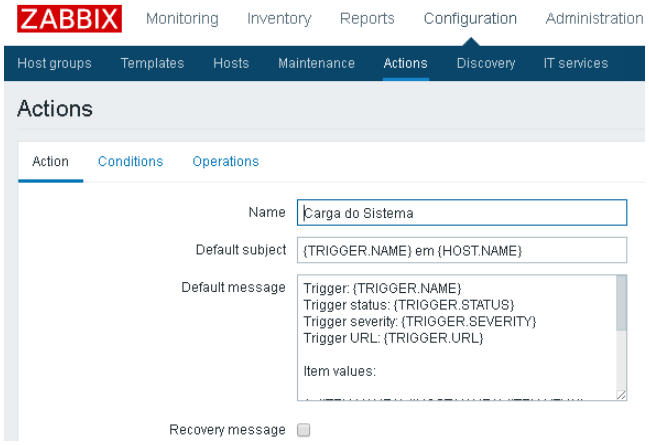

Figura 7: Utilização de macros pelo Zabbix.

[26]. É possível cadastrar regras de varreduras periódicas, permitindo, ao descobrir novos elementos, inserir ao monitoramento e notificar aos administradores. Com a utilização dos macros, as notificações se tornam altamente personalizáveis como mostra a Figura 7.

Pode ser integrado com diversos serviços para incrementar o mecanismo de notificação, podendo haver alertas por e-mail, utilizando SMTP; utilização do sistema de mensagens instantâneas, Jabber, integrado com o Gtalk, por exemplo; SMS. Também através de scripts, é possível que haja integração com o Google Calendar, para envio de SMS por exemplo. Essas funcionalidades alcançadas através de scripts podem ser alcançadas com o Nagios, que também aceita scripts. Outra integração possível é com o Telegram [27].

Assim como o Cacti, o Zabbix também permite a visualização de gráficos para acompanhamento e métricas das informações extraídas dos *hosts*. Permite a automatização de ações para mitigar incidentes, por exemplo, se houver um problema com o servidor Apache, o administrador da rede pode configurar o Zabbix para que o fluxo de ações seja o seguinte: Notificar o administrador, depois de 2 minutos mandar mensagem para outro grupo responsável, depois de 3 minutos reiniciar o Apache.

Como outras características do Zabbix pode-se citar: Os servidores rodam em sistemas Unix-Like, incluindo Linux, AIX, FreeBSD, OpenBSD, HP-UX e Solaris; há agentes nativos para sistemas Unix-like e versões do Microsoft Windows; a administração e monitoramento são via interface web; escalabilidade; flexibilidade; monitoramento agregado, distribuído e em tempo real; monitoramento proativo; autenticação segura de usuários; permissões de usuários e grupos; visualização de relatórios, gráficos, telas e mapas; monitoramento de acordo com nível de serviço; auto-descoberta de dispositivos de rede; suporte a protocolos como SNMP (versões 1, 2 e 3), IPMI, SSH, Telnet [26].

3.4.1 Arquitetura do Zabbix

A arquitetura do Zabbix encontra-se disponível dentro do modelo de três camadas, de acordo com o contexto dos serviços de rede. Essas camadas são: a aplicação, o banco de dados e a interface web [26].

Pode-se encontrar a camada de aplicação sendo representada pelo back-end, tendo como responsabilidade a coleta dos dados nos *hosts* monitorados. A camada de banco de dados é representada pela própria base de dados - MySQL/Mariadb, PostgreSQL, SQLite, Oracle e IBM DB2 são suportados - e fica responsável por armazenar as informações coletadas pelo back-end, apresentando-as ao front-end. A última camada, a interface web é representado pelo front-end, por onde os administradores da rede e aplicações que utilizam a API do Zabbix extraem informações do monitoramento.

O Zabbix Server é o componente principal do sistema, capaz de verificar remotamente os serviços de rede, bem como é o componente central para que os agentes enviem informações e estatísticas a cerca da disponibilidade e integridade dos equipamentos monitorados. O módulo recebe as informações, as processa, exibe relatórios, envia alertas e realiza ações pré-configuradas. É configurável somente em plataformas Unix-like, e como dito anteriormente, acessível através da interface web.

O Zabbix Proxy é um *host* responsável por coletar dados de desempenho e disponibilidade de equipamentos gerenciados remotos, porém com a responsabilidade de repassar as informações ao Zabbix Server, sendo um agregador de dados fazendo o papel de servidor para os *hosts* que ele monitora. Ele coleta informações e as consolida, diminuindo assim a carga de processamento do Server. Outra vantagem é que o Zabbix Proxy consegue rodar em equipamentos com menor desempenho do que o necessário para executar o Server.

O módulo Zabbix Agent é a aplicação encarregada de coletar as informações dos dispositivos gerenciados e enviar ao Zabbix Server ou Zabbix Proxy. Foi desenvolvido de uma forma a não impactar o ambiente monitorado. Também é capaz de acompanhar efetivamente o uso dos recursos e aplicações nos *hosts* gerenciados, tais como: processos, serviços, aplicativos em execução, disco rígido, entre outros [25]. Todas essas vantagens que o Zabbix trás acaba por deixá-lo confuso à vista de um usuário casual. A interface, por ser muito rica, torna a curva de aprendizado muito alta, acompanhado de uma navegação confusa. Utilizar uma funcionalidade útil como um alerta se torna de uma tarefa com complexidade elevada, uma vez que se deve configurar *hosts*, itens, *triggers* e por fim *actions*.

3.5 Impressões pessoais

Todos os softwares de gerenciamento de redes apresentados compartilham de características fundamentais para um software desse domínio, como exemplo a coleta de dados essenciais para dar poder ao administrador de planejar ações, poupar recursos e outras atividades cruciais para o bom funcionamento da rede. Outra característica indispensável é a disponibilidade do software, o que submete a integridade das informações caso venha a ficar indisponível por um determinado tempo.

4. GERENCIADOR SNMP

Para o começo da solução proposta, foi percebido a importância da prevenção de problemas que podem potencialmente prejudicar as atividades. Seja problemas no de-

sempenho de serviços, na rotina de backup, no servidor de e-mails, ou quedas de energia. Faz-se necessário saber em quais condições o problema está propício a aparecer. Dessa forma, uma automatização de quais ações a tomar se torna possível.

A melhor forma de se prevenir contra problemas no ambiente de trabalho é antecipando-se a eles. É sempre bom possuir o conhecimento de como esses problemas surgem. Em contrapartida, há problemas que simplesmente ocorrem, não havendo muitos meios de preveni-los e sim de contorná-los. Tem-se como exemplo, falta de energia, queda de sinal de internet, ou até mesmo problemas com dispositivos de hardware.

O Gerenciador é ideal para empresas que desejam monitorar seus elementos da rede utilizando uma ferramenta que possui uma curva de aprendizado menor, se comparado há soluções existentes e citadas na sessão de Tecnologias Correlatas. O Gerenciador atende também cenários de multiempresa, onde existem filiais ou departamentos.

Nesta sessão serão mostrados mais detalhes do trabalho proposto. Na Figura 8 pode-se ver a inicial com os *dashboards* exibindo estatísticas de elementos e dispositivos monitorados.

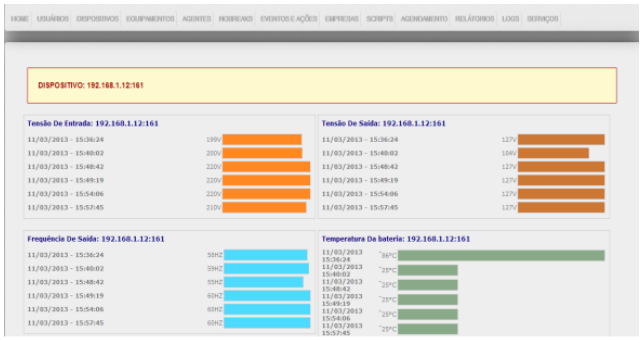

Figura 8: Tela principal do Gerenciador.

4.1 Requisitos

O processo de levantamento de requisitos é essencial para o sucesso de sistemas como este apresentado, detalhando e explicando mais sobre suas etapas. Serão apresentados requisitos funcionais na Tabela 2, que descrevem o que o sistema deve fazer, e na Tabela 1, requisitos não funcionais, explanando permissões de acesso, internacionalização, padronização e ambientação do sistema.

4.2 Arquitetura

Esta seção descreve a arquitetura proposta para o sistema. O modelo sugerido foi elaborado com base nos requisitos e no estudo das soluções de mercado citadas em trabalhos relacionado. A arquitetura contém três nós: AdmNode (Gerenciador), HostNode (Agente) e o ClientNode (Browser).

A Figura 9 facilita o entendimento da arquitetura. A seguir, uma breve descrição de cada nó e suas interdependências:

- Gerenciador: Este é o software proposto para o trabalho. Gerenciando a aquisição de dados e a atuando sobre o equipamento monitorado. Suas principais funções são o armazenamento em log das informações, notificação aos usuários/administradores sobre eventos ocorridos. Além disso, possibilita a configuração dos alarmes, notificações e por armazenar dados históricos condensados dos equipamentos. As informações deste módulo ficam armazenadas em banco de dados.
- Agente: Este módulo tem como objetivo disponibilizar informações sobre o equipamento conectado (UPS, Host) e envio de notificações para o Gerenciador, como por exemplo Bateria Baixa ou Desconexão da energia.
- Browser: Navegador para acesso ao gerenciador.

Código	Descrição	Categoria
[RNF01]	**Persistência** - As informações serão persistidas em banco de dados relacional. Além do funcionamento da transferência de informações entre gerenciador e agentes, o software deve possibilitar a utilização das informações salvas.	Padrão
[RNF02]	**Permissões de usuários do banco de dados** - Para utilização do banco de dados, deve ser admitido que o usuário configurado para integração com SGBD e solução de registro em LOG possui os privilégios necessários e suficientes para exercer as seguintes operações: 1. Criação/inserção (banco e log); 2. Remoção (banco); 3. Atualização (banco); 4. Visualização (banco e log); 5. Execução de stored procedures, function e views (banco); 6. Criação de tabelas temporárias (banco).	Segurança
[RNF03]	**Idioma do software** - O software deve ser facilmente estendível para admitir outros idiomas.	Internacionalização
[RNF04]	**Browsers compatíveis** - O software deverá ser compatível com as versões mais recentes dos navegadores: 1. Internet Explorer; 2. Mozilla Firefox; 3. Chrome; 4. Safari. Estes devem apresentar estabilidade e compatibilidade total com o gerenciador, não comprometendo a troca de dados ou a segurança do sistema.	Compatibilidade

Tabela 1: Requisitos Não Funcionais

Código	Nome	Descrição
[RF01]	Encriptação de senhas dos usuários	Caso a autenticação seja realizada através de tabelas do banco nativo da solução, o software deve garantir a encriptação das senhas de seus usuários, bem como a utilização de diferentes tipos de caracteres e quantidade mínima destes (8 no mínimo).
[RF02]	Mecanismo para recuperação de senha	Deve ser disponibilizado um mecanismo de recuperação e modificação de senha, caso o usuário esqueça a mesma. Deve ser criada uma nova senha sem informar a anterior.
[RF03]	Histórico de alterações realizadas pelos usuários	O software deverá registrar em banco de dados todas as alterações realizadas por usuários de todos os perfis, para posteriormente serem exibidas na interface.
[RF04]	Gerenciamento de equipamentos	O software deve permitir o gerenciamento de equipamentos, disponibilizando uma tela para cadastro acessível para usuários Administradores.
[RF05]	Busca automática de equipamentos	O sistema deve tornar possível encontrar novos equipamentos na rede automaticamente usando SNMP unicast [28], uma vez que tenha obtido a faixa de endereço IP e a porta de comunicação.
[RF06]	Visualização em tempo real dos dados dos equipamentos	O software deve permitir a visualização em tempo real das informações obtidas dos equipamentos, via SNMP. Estes valores devem ser obtidos dos caches mantidos localmente sem que tenha impacto sobre a realização de novas leituras.
[RF07]	Relação de últimos eventos	O software deve permitir a exibição de informações sobre os últimos eventos ocorridos (Log).
[RF08]	Agendamento de tarefas	Deve ser disponibilizada a execução de tarefas de forma imediata ou agendada. Cada tarefa agendada pode estar sujeita a uma frequência de execução (uma vez, diariamente, semanalmente e mensalmente).
[RF09]	Associação de notificação, equipamentos e usuários	O software deve permitir a associação de tipos de notificação por equipamento e usuários.
[RF10]	Personalização das mensagens de notificação	O software deve permitir a personalização das mensagens enviadas para usuários durante notificações, devendo ser realizado através de uma tela de cadastro disponível para Administradores.
[RF11]	Relatório dos problemas ocorridos nos equipamentos	O software deve possibilitar a impressão de um relatório que informe os problemas ocorridos em relação a um equipamento. Este relatório deve estar acessível para usuários de todos os níveis e deve informar também quais são o serviços afetados pelos referidos problemas.
[RF12]	Relatório dos eventos ocorridos	O software deve oferecer um relatório acessível a todos os usuários que informe sobre todos os eventos ocorridos agrupados por equipamento e tipo de evento.
[RF13]	Relatório de notificações enviadas	A solução deve disponibilizar um relatório que relacione envio de notificações com os usuários, agrupando as notificações por tipos de evento e usuários. Este relatório deve ser acessível para usuários com permissão de Administrador.
[RF14]	Exportação dos relatórios	O software deve permitir a exportações de dados de relatórios para os formatos CSV e XLS.
[RF15]	Armazenação das informações em Log	Deve ser acionada a persistência em Log das operações executadas. Esta rotina visa registrar as seguintes informações: 1. Tempo de execução de rotinas críticas (leituras e atuações); 2. Erros ocorridos; 3. Mensagens de debug ou marcos de execução dentro do código.
[RF16]	Limpeza de Log	O software deve disponibilizar uma tela para limpeza dos logs gerados, visando melhorar a manutenção no sistema, caso necessário.

Tabela 2: Requisitos Funcionais

4.3 Principais Funcionalidades

O controle de acesso ao gerenciador é feito a partir de logon na ferramenta, pela interface web do gerenciador. Assim como no Cacti e no Zabbix, esta interface irá organizar e direcionar as ações do administrador da ferramenta de forma que não seja preciso configurar usuários, logons, senhas e permissões de acesso a partir de linhas de comando como no Nagios. O controle de acesso será definido por criação de senhas fortes, alinhadas ao cadastro do usuário, e configurada de acordo com o nível deste na rede monitorada.

É possível criar usuários e perfis de usuários com diferentes permissões, o que soma com a possibilidade de configuração de requisitos feitos diretamente pelo usuário em um ambiente amigável. Os Agentes monitoráveis podem ser integrados aos indicadores desejados, de forma que ações sejam previamente definidas e configuradas para execução. A ferramenta tem o intuito de ser acessível por qualquer plataforma

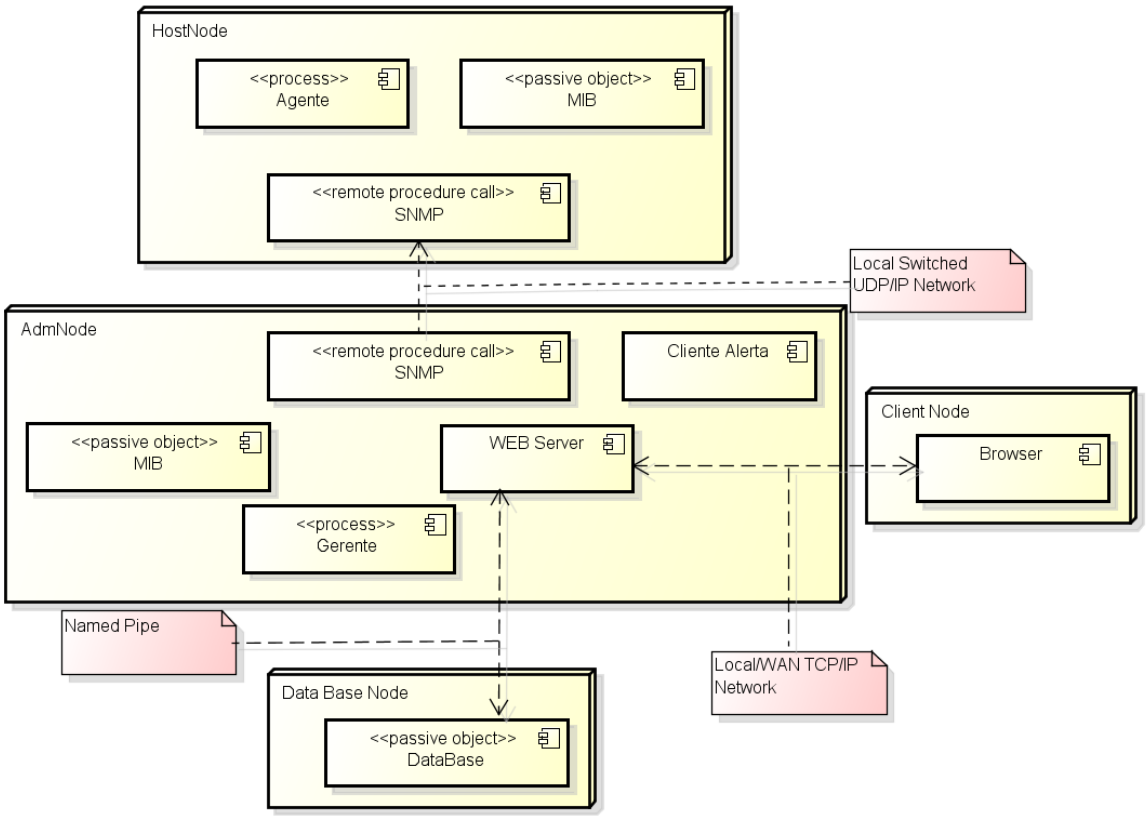

Figura 9: Visão de implantação da solução.

com acesso a um navegador e rede (internet ou localmente onde se encontra o Gerenciador). Para que esteja compatível e portável em vários ambientes, o Gerenciador consegue trabalhar com as três versões do SNMP.

A presença de uma interface web foi uma característica observada nas ferramentas apresentadas na sessão de trabalhos correlatos, chegando-se a conclusão que é um modulo que não poderia faltar para um gerenciador de redes. A interface web proposta visa o gerenciamento de agentes a partir de um ambiente customizável de acordo com o nível de permissão de cada usuário, determinado pelo administrador da ferramenta.

O cadastramento de avisos, alertas e notificações se fará a partir de níveis de permissão definidos previamente, de acordo com a função desempenhada pelo usuário na rede monitorada. Tanto para o administrador quanto para o usuário menos técnico, todos os campos serão configuráveis de acordo com o tipo de permissão.

Antes de continuar com as funcionalidades, é importante relembrar algumas definições:

UPS - Sigla para *Uninterruptible Power Supply*, que corresponde a diferentes aparatos que provêm alimentação de energia para equipamentos quando fontes primárias falham [4]. No contexto deste trabalho são os nobreaks.

Host - Um dispositivo de rede, em geral, é um computador desktop, notebook ou servidor, que possui IP.

No contexto do Gerenciador proposto, existem dois tipos de Agentes. O Agente Host, módulo da solução para monitoramento e automação de *shutdown* de equipamentos, implantado em servidores e estações eletricamente suportadas por um equipamento UPS.

Existem também o Agente UPS, módulo da solução responsável pela comunicação com o equipamento UPS através de comunicação serial. Traduz protocolos proprietários para o SNMP provendo a MIB para consultas. Acumula funções do Agente Host no equipamento onde está instalado

Evento - Um acontecimento percebido por sensores ou pelo próprio software mediante execução de lógica previamente programada e integrada ao software.

Ação - Uma tarefa ocasionada ou em resposta a um evento ou acionada mediante agendamento ou solicitação direta do usuário.

Há uma funcionalidade essencial e de destaque no Gerenciador que é a descoberta de agentes. Pode ser realizado a busca pelo IP direto ou por várias faixas de IP e porta. Nesse processo, será considerado as versões do protocolo de autenticação SNMPv3. Mais detalhes estão presentes na Figura 10

Para descrever o processo de listagem de Agentes, é impor-

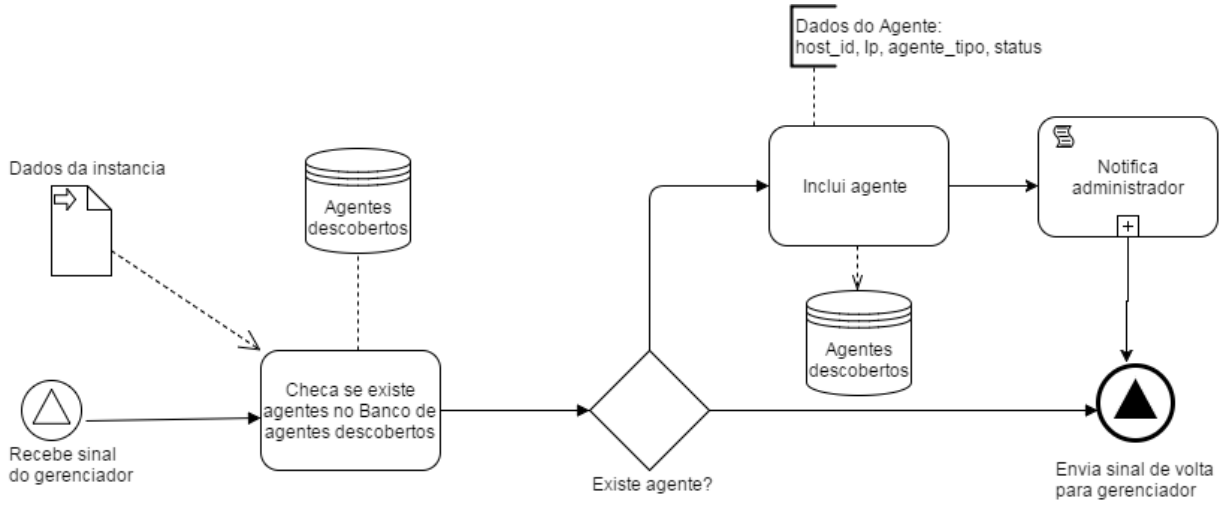

Figura 10: Processo de descoberta de Agentes.

tante falar sobre o processo de cadastro. Há uma tela para cadastro, contendo IP e nome do dispositivo da rede, comunidade, porta principal SNMP, porta de notificações (*trap*), tipo de equipamento (nobreak, *host*). Nesta tela também é necessário carregar a MIB do equipamento ou dispositivo. A Figura 11 representa a tela de cadastro.

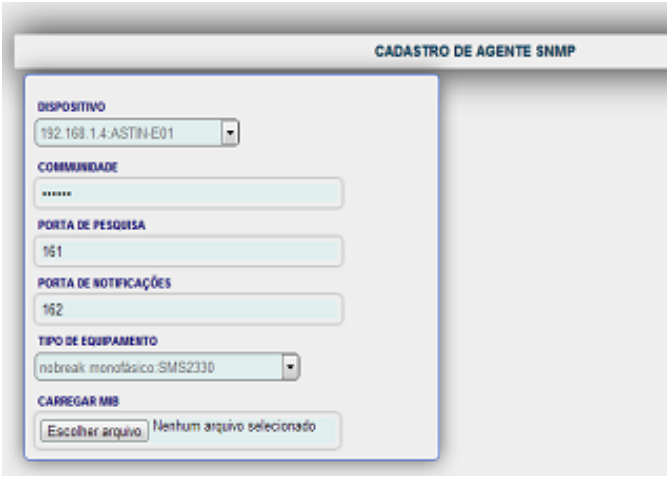

Figura 11: Tela de cadastro de Agentes.

O processo de descoberta de Agentes inclui a listagem de agentes disponíveis. Esse processo torna viável a inclusão dos agentes no gerenciamento através de uma consulta a base durante o processo de descoberta. Ao contrário de outras soluções, o Gerenciador proposto não realiza varredura na rede para localização dos agentes, visto que o agente é proativo em publicar sua existência ao gestor, o que torna o descobrimento mais rápido e eficiente. O próximo passo é incluir esses agentes detectados no gerenciamento.

Como diferencial de inovação a ferramenta proposta faz, não só a leitura de arquivos MIB no padrão .XML, como também a leitura de arquivos de extensão .MIB, garantindo maior aceitação aos ambientes de TI que deverão ser monitorados por esta solução.

Podem ocorrer eventos no sistema, que são percebidos via *trap* ou pelo monitoramento da OID de acordo com a escolha do administrador. Os eventos que já são conhecidos e suas condições aceitáveis podem ser definidos no gerenciador, junto às ações a serem tomadas, que são atribuídas pelo administrador.

O sistema permite a configuração dos eventos ocorridos, das ações a serem tomadas, incluindo a possibilidade de notificações aos usuários. Dessa forma, após o recebimento de uma *trap*, pode ser tomada uma ação em um ou mais Agentes e enviar notificações aos usuários. O software, ao realizar a gravação em banco de dados, mantém um mecanismo de relacionamento entre um evento ocorrido/registrado, as atuações e notificações realizadas.

Assim, a persistência visa manter em banco de dados, as seguintes informações:

- Eventos ocorridos;
- Atuações e notificações ocorridas;
- Tarefas acionadas;

O próximo passo é definir o tipo de evento e a variável a ser monitorada, indicando um valor para a variável, que é a condição do evento.

Na página de listagem dos eventos cadastrados, é possível escolher:

- Os agentes para o qual o evento cadastro será monitorado;
- As ações SNMP;
- As notificações a usuários.

Nome do evento	Trap monitorada	Variável monitorada	Valor indicador	Prioridade	Status	Monitorar Agente	Ações SNMP	Notificar Usuário	Editar	Excluir
Segundos Que As cargas Estão Em Modo Bateria	upsTrapOnBattery	upsSecondsOnBattery	(Maior ou igual) { >= 60 }	Alta	Habilitado					

Figura 12: Tabela com listagem de Eventos.

A tela de listagem de eventos é representada na Figura 12 Após o evento, notificações podem ser disparadas, junto a outras ações cabíveis. A integração dessas funcionalidades é essencial para melhor experiencia do usuário.

Ao notificar usuários, deve ser escolhido os perfis a serem notificados e a forma de notificação - até então SMS ou e-mail. Assim, o software permite a associação de tipos de notificação por equipamento e usuários. Além disso, é possível configurar a forma como cada usuário deve ser notificado.

Ainda sobre o acompanhamento de Agentes, existe forma de utilização da funcionalidade de comunicação entre Gerenciador e Agentes que pode ser muito interessante, o *shutdown* sob demanda. É um processo que permite o desligamento ou hibernação do *host* que hospeda o agente, uma vez que o administrador solicite o procedimento via aplicação.

Caso um agente UPS seja selecionado para esse processo isso implicará no *shutdown* de todos os *host* que dependam unicamente desse UPS, podendo haver inclusive o desligamento do Gerenciador. Será exibida uma tela de confirmação sempre que o processo for solicitado pelo administrador. Deve ser exibido um alerta ao usuário sempre que for solicitado desligamento de um UPS além da tela de confirmação padrão.

Há também o processo inverso denominado *Wake on LAN* ou WOL [29]. O processo é responsável pela inicialização automática de equipamentos disponíveis na rede quando estes são hospedeiros de agentes monitorados pelo Gerenciador. Uma vez que o administrador solicite o procedimento via aplicação, os equipamentos devem ter suporte à tecnologia, estar configurados para Wake On LAN e o meio físico não ser impeditivo. A Figura 13 detalha o processo.

No quesito armazenamento de informações, são utilizados bancos de dados, logs e arquivos em diversos formatos. *Templates* podem tornar o armazenamento adequado à infraestrutura local. Para o tratamento de logs e registro de históricos, são armazenadas tudo sobre as monitorias executadas no período de 24 horas.

Logs diários serão descartados de forma a melhorar a performance e diminuir o grau de armazenamento de bancos de dados da ferramenta. Entretanto, os eventos que ocorrerem de forma anômala, sem pré-definição, deverão ser garantidos por outro período de tempo, com o objetivo de auxiliar na análise futura de problemas mediante o histórico de incidentes armazenados. Para estes casos, a data e hora do incidente serão guardados, e estes dados serão utilizados para

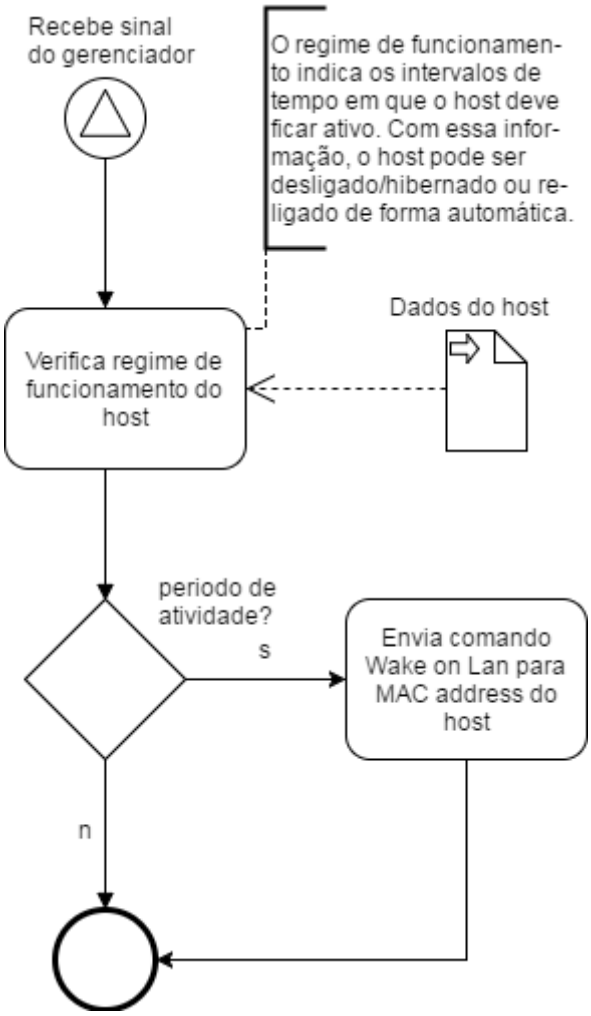

Figura 13: Processo referente ao Wake On LAN.

notificações e alertas quando oportuno.

5. MONITORAMENTO

O principal propósito de um gerenciador de redes é o monitoramento dos seus elementos. Com o Gerenciador SNMP proposto, temos o monitoramento em tempo real. O processo provê a visualização em tempo real de estatísticas relativas a um Agente selecionado, nesse processo as informações são atualizadas dinamicamente na tela do relatório logo que essas são coletadas. Não há registro das informações em banco ou em arquivos, para garantir fidedignidade das informações

exibidas, sem atrasos que pudessem incorrer devido a esse armazenamento.

Na Figura 14 são listados os Agentes que estão ligados a um *host* monitorado. Ao clicar em monitorar, as informações da tela são como as apresentadas na Figura 15.

O Gerenciador também exibe informações em tempo real sobre os estados dos equipamento. Estas são as informações a serem disponibilizadas para o usuário:

- Conectividade (se está conectado ou não);
- Equipamento com problemas.

Estes estados são obtidos em uma busca periódica a partir dos eventos ocorridos sobre o equipamento. Os eventos são definidos pela caracterização de uma OID específica para aquele estado.

6. RELATÓRIOS E LOGS

Serão mostrados processos fornecem acesso a visualização de logs, relatórios gráficos e estatísticos, com base em dados históricos. É utilizado consultas diretas a cada Agente utilizando protocolo SNMPv3 com autenticação e criptografia.

A consulta de informações é feita em tempo real, para exibição de informações nos relatórios, sem que as informações sejam armazenadas em disco, antes ou depois da exibição do relatório, refletindo com maior precisão possível a situação real do elemento em questão.

6.1 Relatórios

O Gerenciador permite a criação de alguns tipos de relatórios.

Na Figura 18 há uma amostra com informações sobre os últimos eventos ocorridos.

Há o relatório de problemas ocorridos, que informa os problemas ocorridos em relação a um equipamento sendo relatados:

- Falha de comunicação com o equipamento;
- Falha de comunicação com o software (plugin) que gerencia o equipamento;

Este relatório está acessível para usuários de todos os níveis e informa também quais são o serviços afetados pelos referidos problemas. Segue um exemplo do relatório na Figura 19.

O próximo relatório a ser abordado é o que apresenta todos as ações realizadas, sendo representado pela Figura 20

Por ultimo há o relatório com os envios de notificações com os usuários. Este relatório agrupa as notificações por tipos de evento e usuários, sendo representado na Figura 21

Para facilitar auditorias e analises por parte do usuário, o software permite exportar os dados de relatórios para os formatos PDF, CSV e XLS. Isto é disponibilizado através de uma tela que possibilita direcionar a saída de relatórios do sistema para cada um destes formatos.

6.2 Logs

O Gerenciador utiliza logs como uma importante forma de manter o controle das ações feitas durante a sessão dos usuário. Isto possibilita que o administrador tenha maior controle sobre o que ocorre no software.

São registrados em banco de dados todas as alterações realizadas por usuários de todos os perfis, para posteriormente serem exibidas na interface. São registrados os seguintes dados:

1. Nome do usuário;
2. Nível do usuário;
3. Data/Hora;
4. Alteração realizada.

Em logs, estão guardados os registros das *traps* que são enviadas ao gerenciador, gravando data, hora, Ip e porta. Na Figura 16 é mostrada a tela com a listagem de logs.

Os arquivos de log armazenam importantes acontecimentos, como erros ao tentar alguma alteração, ao acessar algum elemento, ao inserir senha errada, ao acessar o sistema em si, ao criar, remover ou atualizar itens.

Ao monitorar Agentes UPS, existe a possibilidade de criação de um gráfico, também com informações em tempo real. Através do gráfico é possível notar se há alguma variação nos seguintes atributos:

- Tensão de entrada(V);
- Tensão de saída(V);
- Frequência de saída(Hz);
- Carga da bateria(%);
- Temperatura da bateria(°C).

Há um exemplo de gráfico gerado na Figura 17.

7. TRABALHOS FUTUROS

É necessário sempre procurar por melhoras. Seria interessante o Gerenciador proposto trabalhar com outros meios de notificações, tendo como referência o Zabbix. É totalmente viável a integração como aplicativos de mensagem atuais. É totalmente viável o suporte a outras tecnologias para necessidades mais abrangentes, tendo o RMON como exemplo, se integrando ao SNMP através da MIB.

Uma possível melhora com a experiencia do usuário na tela de principal, melhorando os fluxos que preenchem o *dashboard*. Atualmente podem ficar informações de quatro Agentes na tela principal, contudo, se algum dos Agentes por algum motivo não estiver com os itens disponíveis, ainda assim um espaço dos quatro disponíveis é ocupado na *dashboard*, permanecendo um espaço vazio.

IP / PORTA	NOME	TIPO / MODELO	STATUS	GERENCIAR MIB	MONITORAR	EDITAR	EXCLUIR	DASHBOARD
200.150.128.14:161	Servidor 3	Nobreak Monovolt (Station II)	Ativo					
192.168.1.1:161	192.168.1.1	nobreak monofásico (SMS2330)	Ativo					
192.168.1.10:161	192.168.1.10	nobreak monofásico (SMS2330)	Inativo					
192.168.1.2:161	PRJ-01	nobreak monofásico (SMS2330)	Ativo					
192.168.1.4:163	E01	Agente Host (Agente Host)	Inativo					
192.168.1.12:163	S07	Agente Host (Agente Host)	Ativo					
192.168.1.4:161	E01	Agente Host (Agente Host)	Inativo					
192.168.1.3:161	14UD0	nobreak monofásico (SMS2330)	Ativo					
192.168.1.12:161	S07	Nobreak Trifásico (Sinus Triphases)	Inativo					

Figura 14: Lista de Agentes monitorados.

- Status Da bateria: 2
- Tempo Estimado De carga: :mib-2.33.1.2.3.0': No Such Object available on this agent at this OID
- Tensão Na Bateria: :mib-2.33.1.2.5.0': No Such Object available on this agent at this OID
- Temperatura Da Bateria: 50
- Tensão De Entrada: :mib-2.33.1.3.3.1.3.0': No Such Instance currently exists at this OID
- Tensão De Saída: :mib-2.33.1.4.4.1.2.0': No Such Instance currently exists at this OID
- upsOutputPower: :mib-2.33.1.4.4.1.4.0': No Such Instance currently exists at this OID
- upsOutputPercentLoad: :mib-2.33.1.4.4.1.5.0': No Such Instance currently exists at this OID
- upsBypassFrequency: :mib-2.33.1.5.1.0': No Such Object available on this agent at this OID
- upsBypassNumLines: 1

Figura 15: Visualização em tempo real dos Dados de um Equipamento.

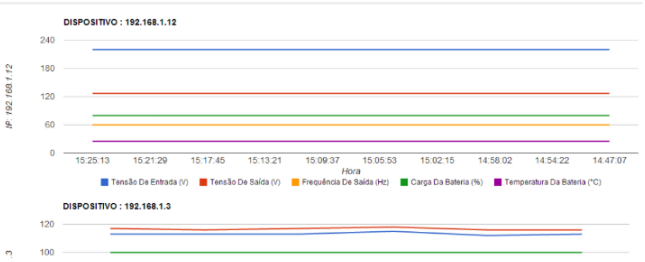

Figura 17: Gráfico com detalhes de Agente UPS.

EXPORTAR

Descrição Do Log	Nome Do Usuário	Email Do Usuário	Data De Criação
ATUALIZOU DISPOSITIVO	Leonardo Souza Da Silva	allezowagon@hotmail.com	04/04/2017 - 14:12:27
ATUALIZOU DISPOSITIVO	Leonardo Souza Da Silva	allezowagon@hotmail.com	04/04/2017 - 14:11:36
ATUALIZOU DISPOSITIVO	Leonardo Souza Da Silva	allezowagon@hotmail.com	04/04/2017 - 14:11:26
CADASTROU AGENTE MANUALMENTE	Leonardo Souza Da Silva	allezowagon@hotmail.com	04/04/2017 - 14:08:12
CADASTROU AGENTE MANUALMENTE	Leonardo Souza Da Silva	allezowagon@hotmail.com	04/04/2017 - 14:05:26
ATUALIZOU DISPOSITIVO	Leonardo Souza Da Silva	allezowagon@hotmail.com	04/04/2017 - 13:53:58
ATUALIZOU DISPOSITIVO	Leonardo Souza Da Silva	allezowagon@hotmail.com	04/04/2017 - 13:53:44
ATUALIZOU DISPOSITIVO	Leonardo Souza Da Silva	allezowagon@hotmail.com	04/04/2017 - 13:51:21
ATUALIZOU DISPOSITIVO	Leonardo Souza Da Silva	allezowagon@hotmail.com	04/04/2017 - 13:49:36
ATUALIZOU DISPOSITIVO	Leonardo Souza Da Silva	allezowagon@hotmail.com	04/04/2017 - 13:49:00
ATUALIZOU DISPOSITIVO	Leonardo Souza Da Silva	allezowagon@hotmail.com	04/04/2017 - 13:47:29
CADASTROU DISPOSITIVO	Leonardo Souza Da Silva	allezowagon@hotmail.com	04/04/2017 - 13:46:06
Usuário Acessou O Sistema	Leonardo Souza Da Silva	allezowagon@hotmail.com	04/04/2017 - 09:14:15

Figura 16: Tela com a listagem de logs.

8. CONCLUSÃO

Foram enfrentados desafios ao decorrer do desenvolvimento do sistema. A utilização de código em PHP facilita no fato da solução ser voltada inteiramente para o ambiente web. Problemas envolvendo o a comunicação pelo protocolo UDP foram pertinentes também.

Após as motivações apresentadas, entende-se a necessidade de organizações e empresas para monitorar a sua rede e os elementos nela inseridos. Devem ser levados em consideração o tamanho da empresa em relação a recursos tecnológicos ou humanos que influenciam diretamente no tamanho da rede utilizada por ela. Ao falar tamanho da rede, deve-se considerar a quantidade de transferência de dados diários, se existem filiais da empresa ou estações de clientes ligadas a ela.

Com as tecnologias relacionadas apresentadas, fica claro o custo de implantação e a potencialidade de cada uma delas. A empresa deve considerar suas estruturas e recursos ao escolher uma das tecnologias. Das apresentadas, o SDN-OpenFlow é a tecnologia que mais demandaria alterações na estrutura da rede.

O Gerenciador proposto se faz ideal para ambientes de rede de porte elevado, contudo, especificado nos elementos críticos da rede, que seriam os servidores e nobreaks. Se for desejado um gerenciamento completo, e se existir recursos disponíveis para a implantação de tal, então a melhor escolha pode ser uma das outras tecnologias discutidas neste trabalho.

NOME DO EVENTO	TRAP MONITORADA	VARIAVEL MONITORADA	VALOR DE PARÂMETRO	PRIORIDADE	DATA DE OCORRÊNCIA
Segundos Que As cargas Estão Em Modo Bateria	upsTrapOnBattery	upsSecondsOnBattery	(Maior ou igual) { >= 60 }	Média	03/04/2017 - 12:00:19
Segundos Que As cargas Estão Em Modo Bateria	upsTrapOnBattery	upsSecondsOnBattery	(Maior ou igual) { >= 60 }	Média	03/04/2017 - 12:25:55
Segundos Que As cargas Estão Em Modo Bateria	upsTrapOnBattery	upsSecondsOnBattery	(Maior ou igual) { >= 60 }	Média	03/04/2017 - 12:26:25
Segundos Que As cargas Estão Em Modo Bateria	upsTrapOnBattery	upsSecondsOnBattery	(Maior ou igual) { >= 60 }	Média	03/04/2017 - 12:39:25
Segundos Que As cargas Estão Em Modo Bateria	upsTrapOnBattery	upsSecondsOnBattery	(Maior ou igual) { >= 60 }	Média	03/04/2017 - 12:40:19
Segundos Que As cargas Estão Em Modo Bateria	upsTrapOnBattery	upsSecondsOnBattery	(Maior ou igual) { >= 60 }	Média	03/04/2017 - 12:51:21
Segundos Que As cargas Estão Em Modo Bateria	upsTrapOnBattery	upsSecondsOnBattery	(Maior ou igual) { >= 60 }	Média	03/04/2017 - 12:51:39
Segundos Que As cargas Estão Em Modo Bateria	upsTrapOnBattery	upsSecondsOnBattery	(Maior ou igual) { >= 60 }	Média	03/04/2017 - 11:44:28
Segundos Que As cargas Estão Em Modo Bateria	upsTrapOnBattery	upsSecondsOnBattery	(Maior ou igual) { >= 60 }	Média	03/04/2017 - 11:45:27
Segundos Que As cargas Estão Em Modo Bateria	upsTrapOnBattery	upsSecondsOnBattery	(Maior ou igual) { >= 60 }	Média	03/04/2017 - 17:59:28
Segundos Que As cargas Estão Em Modo Bateria	upsTrapOnBattery	upsSecondsOnBattery	(Maior ou igual) { >= 60 }	Média	03/04/2017 - 18:00:29

Figura 18: Relatório de Eventos ocorridos.

NOME DA AÇÃO	MENSAGEM	DATA DE OCORRÊNCIA
Executar Script (Criar Pasta Desktop)	Erro De Comando SNMP: No response from 192.168.1.12:163	03/04/2017 - 11:36:38
Executar Script (Criar Pasta Desktop)	Erro De Comando SNMP: No response from 192.168.1.12:163	03/04/2017 - 11:36:47
Executar Script (Criar Pasta Desktop)	Erro De Comando SNMP: No response from 192.168.1.12:163	03/04/2017 - 11:37:29
Executar Script (Criar Pasta Desktop)	Erro De Comando SNMP: No response from 192.168.1.12:163	03/04/2017 - 11:37:38
Executar Script (Criar Pasta Desktop)	Erro De Comando SNMP: No response from 192.168.1.3:163	03/04/2017 - 13:05:16
Executar Script (Criar Pasta Desktop)	Erro De Comando SNMP: No response from 192.168.1.6	03/04/2017 - 13:05:22
Executar Script (Criar Pasta Desktop)	Erro De Comando SNMP: No response from 192.168.1.12:163	03/04/2017 - 13:05:29
Executar Script (Criar Pasta Desktop)	Erro De Comando SNMP: No response from 192.168.1.3:163	03/04/2017 - 13:05:45
Executar Script (Criar Pasta Desktop)	Erro De Comando SNMP: No response from 192.168.1.6	03/04/2017 - 13:05:51
Executar Script (Criar Pasta Desktop)	Erro De Comando SNMP: No response from 192.168.1.12:163	03/04/2017 - 13:05:58
Executar Script (Criar Pasta Desktop)	Erro De Comando SNMP: No response from 192.168.1.3:163	03/04/2017 - 18:00:15
Executar Script (Criar Pasta Desktop)	Erro De Comando SNMP: No response from 192.168.1.6	03/04/2017 - 18:00:21
Executar Script (Criar Pasta Desktop)	Erro De Comando SNMP: No response from 192.168.1.12:163	03/04/2017 - 18:00:27
Executar Script (Criar Pasta Desktop)	Erro De Comando SNMP: No response from 192.168.1.3:163	03/04/2017 - 18:09:40
Executar Script (Criar Pasta Desktop)	Erro De Comando SNMP: No response from 192.168.1.6	03/04/2017 - 18:09:46

Figura 19: Relatório de problemas.

9. REFERÊNCIAS

[1] A história da internet: pré-década de 60 até anos 80 [infográfico]. https://www.tecmundo.com.br/infografico/9847-a-historia-da-internet-pre-decada-de-60-ate-anos-80-infografico.htm - Acessado em Novembro de 2016.

[2] Tomas Lovis Black. Comparação de ferramentas de gerenciamento de redes. 2008.

[3] Herbert Domingues Pires Fabiano Rocha Abreu. Gerência de redes. *Departamento de Engenharia de Telecomunicações, Universidade Federal Fluminense. Niterói, 07 de Março de 2017.*

[4] Barry M Epstein. General purpose uninterruptible power supply.

NOME DA AÇÃO	MENSAGEM	DATA DE OCORRÊNCIA
Executar Script (Criar Pasta Desktop)	Ip do dispositivo: 192.168.1.3:163 nome do dispositivo: CRIAMUNDO descrição da ação: Executar Script (Criar Pasta Desktop) servicos afetados: Servidor Proxy,Servidor Web,Servidor De E-Mail	03/04/2017 - 11:36:31
Executar Script (Criar Pasta Desktop)	Ip do dispositivo: 192.168.1.6:161 nome do dispositivo: ASTIN-E01 descrição da ação: Executar Script (Criar Pasta Desktop) servicos afetados: Serviço De Simulação De Nobreak	03/04/2017 - 11:36:32
Executar Script (Criar Pasta Desktop)	Ip do dispositivo: 192.168.1.3:163 nome do dispositivo: CRIAMUNDO descrição da ação: Executar Script (Criar Pasta Desktop) servicos afetados: Servidor Proxy,Servidor Web,Servidor De E-Mail	03/04/2017 - 11:36:40
Executar Script (Criar Pasta Desktop)	Ip do dispositivo: 192.168.1.6:161 nome do dispositivo: ASTIN-E01 descrição da ação: Executar Script (Criar Pasta Desktop) servicos afetados: Serviço De Simulação De Nobreak	03/04/2017 - 11:36:41
Executar Script (Criar Pasta Desktop)	Ip do dispositivo: 192.168.1.3:163 nome do dispositivo: CRIAMUNDO descrição da ação: Executar Script (Criar Pasta Desktop) servicos afetados: Servidor Proxy,Servidor Web,Servidor De E-Mail	03/04/2017 - 11:37:22
Executar Script (Criar Pasta Desktop)	Ip do dispositivo: 192.168.1.6:161 nome do dispositivo: ASTIN-E01 descrição da ação: Executar Script (Criar Pasta Desktop) servicos afetados: Serviço De Simulação De Nobreak	03/04/2017 - 11:37:23

Figura 20: Relatório de Ações SNMP realizadas.

NOME DO USUÁRIO	MENSAGEM	STATUS	DATA DE OCORRÊNCIA
Leonardo Souza Da Silva	Envio De Email Para: Leonardo Souza Da Silva < gmc.manager.sms@gmail.com > Data Programada Para Envio: 2013-03-27 17:03:44	Executado	03/04/2017 - 17:03:50
Leonardo Souza Da Silva	Envio De SMS Para: Leonardo Souza Da Silva < 7181421962 > Data Programada Para Envio: 2013-03-27 17:03:44	Executado	03/04/2017 - 17:03:50
Leonardo Souza Da Silva	Envio De Email Para: Antonio Carlos < gmc.manager.sms@gmail.com > Data Programada Para Envio: 2013-03-27 17:03:44	Executado	03/04/2017 - 17:03:53
Leonardo Souza Da Silva	Envio De SMS Para: Antonio Carlos < 7333333333 > Data Programada Para Envio: 2013-03-27 17:03:44	Executado	03/04/2017 - 17:03:53
Leonardo Souza Da Silva	Envio De Email Para: Leonardo Souza Da Silva < gmc.manager.sms@gmail.com > Data Programada Para Envio: 2013-03-27 17:04:17	Executado	03/04/2017 - 17:05:30
Leonardo Souza Da Silva	Envio De SMS Para: Leonardo Souza Da Silva < 7181421962 > Data Programada Para Envio: 2013-03-27 17:04:17	Executado	03/04/2017 - 17:05:30
Leonardo Souza Da Silva	Envio De Email Para: Antonio Carlos < gmc.manager.sms@gmail.com > Data Programada Para Envio: 2013-03-27 17:04:17	Executado	03/04/2017 - 17:05:34
Leonardo Souza Da Silva	Envio De SMS Para: Antonio Carlos < 7333333333 > Data Programada Para Envio: 2013-03-27 17:04:17	Executado	03/04/2017 - 17:05:34
Leonardo Souza Da Silva	Envio De Email Para: Leonardo Souza Da Silva < gmc.manager.sms@gmail.com > Data Programada Para Envio: 2013-03-27 17:57:57	Falha	03/04/2017 - 17:58:33
Leonardo Souza Da Silva	Envio De SMS Para: Leonardo Souza Da Silva < 7181421962 > Data Programada Para Envio: 2013-03-27 17:57:57	Executado	03/04/2017 - 17:58:33

Figura 21: Relatório de envio de notificações.

https://www.google.com/patents/US4675538 - Acessado em 06 de Abril de 2017.

[5] Steve Waldbusser. Remote network monitoring management information base version 2. 2006.

[6] Internet engineering task force. http://www.ietf.org - Acessado em 15 de Março de 2017.

[7] O protocolo de gerenciamento rmon. https://memoria.rnp.br/newsgen/9901/rmon.html - Boletim bimestral sobre tecnologia de redes produzido e publicado pela RNP - Acessado em 13 de Janeiro de

2017
.
[8] Luciano Paschoal Gaspary. Rmon e rmon2-remote network monitoring. Febuary 10, 2001.
[9] Francisco Oliveira Filipe Pedrosa, José Teixeira. S.n.m.p. departamento de ciência de computadores. FCUP. 17 de Maio de 2004.
[10] Robert Thurlow. Rpc: Remote procedure call protocol specification version 2. 2009.
[11] Anderson KARING. Protótipo de um sistema de monitoramento de desempenho de redes de computadores baseado no protocolo snmpv3. 2002.
[12] William Stallings. Snmp and snmpv2: the infrastructure for network management. 1998.
[13] DC Lynch. Gl obalization of t he internet, the internet system handbook, ed. daniel c. lynch-marshall t. *Rose, Boston: MA: Addison Wesley*, 1993.
[14] AS Tanenbaum. Computer networks 3rd edition prentice hall international inc. *ISBN: 0-13-394248-1*, 1996.
[15] Net-snmp. `http://net-snmp.sourceforge.net` - Acessado em 15 de Março de 2017.
[16] Software-defined networking (sdn) definition. `https://www.opennetworking.org/sdn-resources/sdn-definition`. Acessado em 10 de Fevereiro de 2017.
[17] Openflow. `https://www.opennetworking.org/sdn-resources/openflow`. Acessado em 10 de Fevereiro de 2017.
[18] Adrian Lara. Using software-defined networking to improve campus, transport and future internet architectures. 2015.
[19] Marco Canini, Daniele Venzano, Peter Peresını, Dejan Kostic, and Jennifer Rexford. A nice way to test openflow applications. Princeton University. 2012.
[20] Openflow switch specification, version 1.1.0 implemented (wire protocol 0x02). `http://archive.openflow.org`, February 28, 2011.
[21] Richard Stallman et al. The gnu project, 1998.
[22] Nagios core. `https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/` - Acessado em Novembro de 2016.
[23] Tobias Oetiker. Rrdtool, 2005.
[24] Cacti documentation. `http://docs.cacti.net/manual:087` - Acessado em Novembro de 2016.
[25] Washingtion Ernando Pereira Benício. Monitoramento e gerenciamento de redes utilizando zabbix. IFSP. 2015.
[26] Janssen dos Reis Lima. *Monitoramento de redes com Zabbix: monitore a saúde dos serviços e equipamentos de rede*. Rio de Janeiro, Brasport, 2014.
[27] Telegram messeger. `hhttps://telegram.org`.
[28] Network address translation (nat) behavioral requirements for unicast udp. `https://www.rfc-editor.org/rfc/rfc4787.txt` - Acessado em 05 de Abril de 2017.
[29] Philip Lieberman. Wake-on-lan technology, 2010.