Universidade Federal de Itajubá
Programa de Pós-Graduação em Engenharia Elétrica

Proposta de Monitoramento Integrado para Ambientes Computacionais Utilizando Software Livre

CEZAR JOSÉ SANT'ANNA JUNIOR

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica como parte dos requisitos para obtenção do Título de Mestre em Ciências em Engenharia Elétrica

Março de 2007
Itajubá – MG

Universidade Federal de Itajubá
UNIFEI

CEZAR JOSÉ SANT'ANNA JUNIOR

Proposta de Monitoramento Integrado para Ambientes Computacionais Utilizando Software Livre

Dissertação apresentada à Universidade Federal de Itajubá para obtenção do título de Mestre em Engenharia Elétrica.

Área de Concentração:
Automação e Sistemas Elétricos Industriais
Orientador: Prof. Dr. Otavio Augusto Salgado Carpinteiro

Itajubá 2007

Dedico este trabalho a Deus, aos meus pais pelo amor incondicional e a minha esposa pela dedicação, carinho e amor, sem os quais este trabalho não seria possível.

Agradecimentos

Agradeço a todos que de alguma forma contribuíram para o desenvolvimento deste trabalho, em especial:

Em primeiro lugar, a Deus, por estar sempre presente na minha vida, em todos os momentos que eu pedi e, principalmente, naqueles em que não precisei fazê-lo.

Aos meus pais, Cezar e Linda, que garantiram minha educação, por acreditarem no meu sonho e por me ensinarem os valores que mais importam nesta vida.

À minha amada esposa, que ao meu lado, mesmo passando por momentos difíceis, sempre me apoiou na concretização deste sonho e lutou comigo, para ultrapassar todos os obstáculos que a Vida nos apresentou.

Aos meus sogros, Paulo e Teresinha, que também foram meus pais em todos os momentos em que estive com eles, tratando-me como a um filho.

Aos meus irmãos, Paulo, Lauro e Marcos, pelo amor incondicional e apoio.

Ao meu cunhado e minhas cunhadas pela paciência e carinho.

Aos meus sócios Caio, Roberto e Rodrigo, por aceitarem minhas limitações e compreenderem minhas faltas, sem nunca me faltarem com o apoio.

Ao amado mestre Maurílio que me mostrou a Luz, em meio a toda a escuridão.

Ao meu orientador Otavio, por acreditar em mim, nas mais difíceis circunstâncias e por me apoiar, mesmo quando nem sabia que este apoio me era vital.

Resumo

O monitoramento dos ambientes computacionais é um dos principais componentes, nos mais completos sistemas de segurança da informação. Tendo como objetivo maior monitorar a disponibilidade das informações aos atores que têm o direito de acesso a estas informações, [1] o propósito deste trabalho é propor um modelo de monitoramento integrado, visando identificar os pontos de falhas em uma conexão, reduzindo, assim, o tempo entre a queda da conexão e seu restabelecimento. Para isto é necessária a integração de diversas ferramentas *opensource*, onde, cada uma delas, monitora uma camada específica dos ambientes computacionais.

Abstract

The computational environments monitoring is one of the main components in the most complete security information systems. Having as objective to monitor the information availability to the actors whom the access right of access to these information has [1], this work intention is to consider a model of integrated monitoring; aiming at to identify the points of imperfections in a connection, thus reducing the time enters the fall of the connection and its reestablishment. For this the integration of diverse opensource tools was necessary, where each one of them monitorial a layer specifies of computational environments.

Lista de Abreviaturas e Siglas

API – APPLICATION PROGRAMMING INTERFACE

ARP – ADDRESS RESOLUTION PROTOCOL

CPU – CENTRAL PROCESS UNIT

HTML – HYPERTEXT MARKUP LANGUAGE

IP – INTERNET PROTOCOL

IPFIX – INTERNET PROTOCOL INFORMATION EXPORT

LAN – LOCAL AREA NETWORK

PDU – PACKET DATA UNIT

PF – PACKET FILTER

RRA – ROUND ROBIN ARCHIVES

RRD – ROUND ROBIN DATABASE

SNMP – SIMPLE NETWORK MANAGEMENT PROTOCOL

TCP – TRANSMISSION CONTROL PROTOCOL

UDP – USER DATAGRAMA PROTOCOL

XML – EXTENSIBLE MARKUP LANGUAGE

WAN – WIDE AREA NETWORK

Sumário

Lista de Figuras

Lista de Tabelas

1 INTRODUÇÃO

O monitoramento dos ambientes computacionais é um dos principais componentes nos mais completos sistemas de segurança da informação e tem, como objetivo maior, monitorar a disponibilidade das informações aos atores que tem o direito de acesso a estas informações [1]. Os modelos atuais de monitoramento de ambientes computacionais focam suas ações em dois pontos específicos: os servidores e na infra-estrutura de ligação da LAN. Esta abordagem não leva em conta o equipamento cliente, seus recursos e limitações. Desta forma, a correta detecção do ponto de falha no cliente é mais complexo e, muitas vezes, demorado.

1.1 Considerações Iniciais

A motivação deste trabalho foi a lacuna encontrada no monitoramento de ambientes computacionais heterogêneos. Esta lacuna é criada pela falta de inter-relações entre o que está sendo monitorado e os eventos que os usuários reportam. As atuais ferramentas *opensource* de monitoramento de redes concentram seus esforços em duas frentes: nos servidores ou na infra-estrutura de rede, esquecendo-se dos equipamentos que os usuários utilizam para acessar os recursos do ambiente.

Normalmente, quando um usuário faz uma reclamação sobre lentidão ou desempenho insuficiente de uma aplicação oferecida via rede, os administradores possuem ferramentas que lhes informam sobre o *link,* utilização de *CPU* do servidor e, algumas vezes, chegando ao status da interface de rede do computador cliente sem investigar outros fatores que poderiam

ser responsáveis pela falta de desempenho, tais como: *CPU* do cliente, utilização de memória, programas que estão em execução no computador do cliente e taxa de utilização de disco.

Um ambiente centralizado, onde a mesma equipe coordena todos os segmentos do ambiente, possuí uma tolerância maior a este cenário, já que a falha começa e termina dentro de um escopo definido. Quando ocorre um aumento na complexidade da organização este escopo passa a não ser bem delimitado, recaindo sobre várias equipes, com diferentes coordenadores, a correção do mesmo problema.

Possivelmente, cada equipe será responsável por um segmento específico dentro do ambiente (aplicativos, interconexão da *LAN*, interconexão da *WAN*, suporte físico dos computadores e equipamentos), tendo, cada qual, os programas de monitoramento focados em seu segmento. Assim, não existindo uma visão única da falha a sua detecção e provável correção é dificultada.

Neste cenário, onde as falhas se acumulam sem solução, a informação, tão vital para a empresa como é o sangue para o corpo humano [2], torna-se inacessível para aqueles que têm o direito de acessá-las. Os processos gerenciais e operacionais sofrem atrasos, gerando prejuízos.

A disponibilidade da informação é um dos pilares que sustentam a segurança da informação [2]. Sem ela, as empresas perdem diferenciais competitivos, tais como, agilidade e a capacidade de integração com fornecedores e clientes. Isto resulta numa crescente diminuição de sua participação no mercado.

1.2 Ambientes estudados para a elaboração do modelo

Para o desenvolvimento deste trabalho foram estudados três ambientes reais, em cada ambiente é observado um modelo de gestão e operação que se mostram falhos ou ineficientes.

1.2.1 Gestão Segmentada

Como primeiro ambiente de estudo tem-se uma grande empresa brasileira, dependente de sua infra-estrutura de tecnologia da informação e que optou pela terceirização da quase totalidade deste setor.

A terceirização economizou recursos de pessoal e econômico, mas acabou ocasionando uma série de problemas na gestão do setor de tecnologia da informação.

Com a divisão de sua infra-estrutura entre os diferentes prestadores de serviço esta empresa perdeu a capacidade de definir o ponto de falha em tempo hábil e muitas vezes nunca consegue encontra-lo.

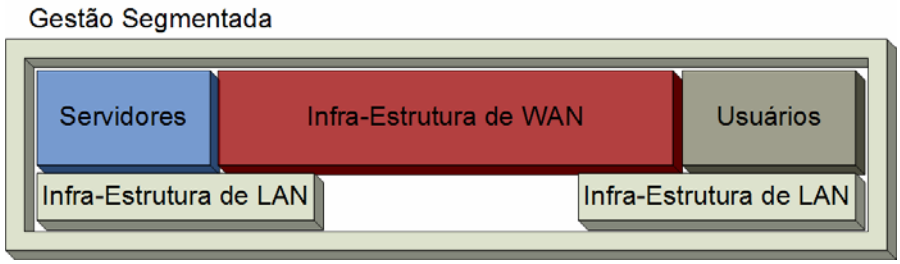

Figura 1 – Gestão Segmentada

Devido à gestão segmentada cada prestador de serviço informa através de relatórios que "todo funciona bem", mesmo quando é notada uma queda no desempenho das aplicações é quase impossível obter uma visão integrada da rede e definir onde deve-se investir para garantir o desempenho.

1.2.2 Servidor Remoto

Neste ambiente tem-se uma empresa que através de uma aplicação Java oferece alguns serviços de forma remota, mantendo controle somente da sua infra-estrutura, a empresa não consegue determinar quais as possíveis causas para o baixo desempenho de sua aplicação.

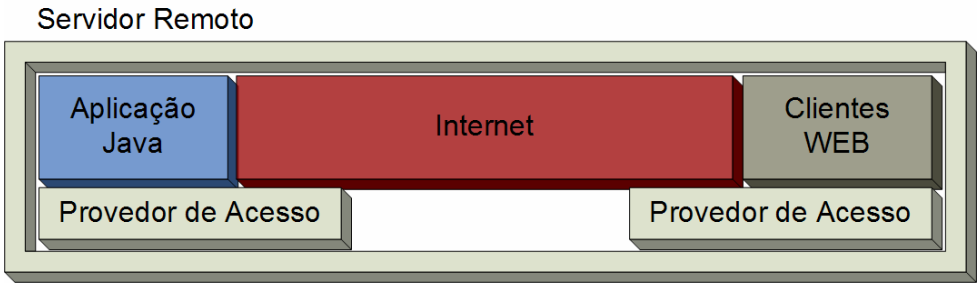

Figura 2 – Servidor Remoto

Cada cliente é responsável por sua própria infra-estrutura, impedindo a verificação dos requisitos necessários em tempo real, dificultando a correta detecção do ponto de falha.

A própria empresa não utiliza nenhum software de gestão para os ativos de sua área de tecnologia, nem sistemas de garantia de qualidade de serviço. Este quadro favorece os problemas de desempenho, pois não existe dados sobre a utilização e ou priorização no atendimento das requisições dos clientes.

1.2.3 Equipamentos Comprometidos

O último ambiente apresenta um cenário bem diferente dos anteriores, pois todos os equipamentos estão na mesma LAN, mas devido ao perfil dos usuários os computadores clientes geralmente se apresentam comprometidos, softwares instalados sem permissão, arquivos de configuração modificados, conexão física desfeita e roubo de peças internas principalmente pentes de memória.

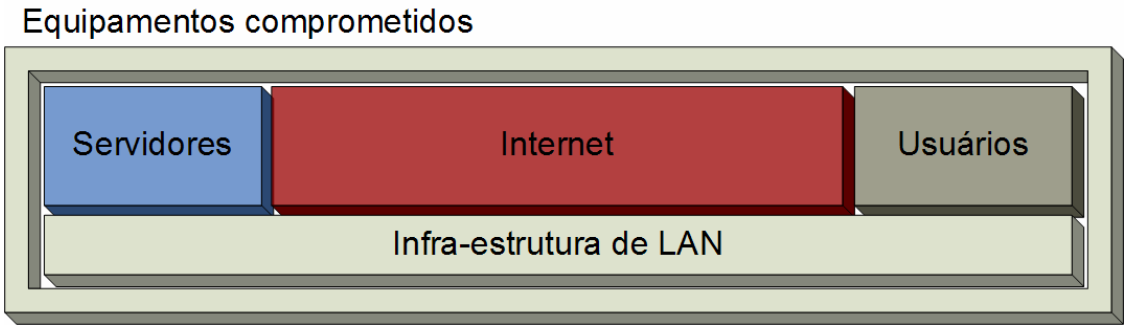

Figura 3 – Equipamentos Comprometidos

Sem um monitoramento integrado e com foco nos equipamentos dos usuários, a descoberta de uma alteração ou falha no acesso só será detectado e resolvido com intervenções no local. Tornando o processo lento e ineficaz.

1.3 Soluções existentes estudadas

Existem diversas ferramentas de monitoramento *opensource* no mercado, cada uma atuando em segmentos específicos dos ambientes computacionais monitorados. Algumas possuem flexibilidade suficiente para serem adaptadas para colher informações de outros segmentos, mas como exigem conhecimento avançado em linguagens de programação e das próprias ferramentas de monitoramento, têm essa personalização dificultada.

Com a segmentação do monitoramento a dificuldade em se obter uma visão integrada de uma falha ou queda no desempenho é muito grande, complicando o processo de correção do problema.

Este trabalho aborda as principais ferramentas *opensource* de monitoramento, com uma visão analítica, e estabelece a existência de pontos falhos em cada uma delas, apresentando a seguir estes resultados.

1.3.1 Nagios

O *nagios* é uma ferramenta de monitoramento de redes que foca suas ações no segmento de *hosts* e aplicativos, fornecendo informações completas sobre o estado de funcionamento dos servidores da rede e o estado dos aplicativos [3].

Os recursos do *nagios* estão diretamente ligados a dois fatores: o protocolo *SNMP* e os *scripts* criados pelos usuários. Através do *SNMP* é possível buscar informações sobre a utilização dos recursos físicos dos *hosts* monitorados, tais como utilização da *CPU*, memória *RAM*, memória *swap* e discos. Enquanto isso, através dos *scripts*, que são programas escritos em linguagens de programação interpretadas, são obtidos os dados sobre os aplicativos dos servidores, tais como: servidor de *web*, servidor de *DNS*, servidor de *e-mail* e qualquer outro aplicativo para o qual o usuário tenha a disposição de criar um *script,* seguindo os modelos do *nagios*. [3]

Esta capacidade de expansão torna o *nagios* um aplicativo bem flexível pois, dependendo do conhecimento e acesso do usuário, ele pode ser utilizado para monitorar qualquer equipamento em uma rede.

A interface com o usuário é simples e direta, facilitando a observação dos valores atuais dos pontos monitorados; mas, por ser toda baseada em valores pontuais, é difícil acompanhar o desenvolvimento ao longo do tempo de determinado recurso ou serviço.

O *nagios* não possui nenhum recurso gráfico de configuração, exigindo um vasto conhecimento por parte do usuário, tanto dos fundamentos teóricos de redes de computadores, como da própria ferramenta, pois cada arquivo de configuração possui um formato próprio, dificultando, assim, a implantação e aproveitamento máximo dos recursos que o *nagios* oferece.

A complexidade na configuração e instalação também está presente na correlação dos eventos, pois são apresentadas informações sobre o estado de serviços, independente da conexão de rede, sendo impossível verificar se um serviço está indisponível por um problema no link, por uma falha no software ou, ainda, uma falha na placa de rede do servidor que hospeda este serviço.

Por não atuar neste segmento o *nagios* também não apresenta dados sobre a utilização de *links* e quais protocolos estão trafegando pela rede, o que, novamente impede uma visão integrada de toda a infra-estrutura cliente-servidor.

1.3.2 Cacti

Utilizado para monitorar principalmente os recursos físicos de equipamentos de rede como *hosts*, roteadores e *switchs* o *cacti* trabalha principalmente com o SNMP. Este protocolo permite ao *cacti* acessar informações referentes ao hardware dos equipamentos de forma padronizada, possibilitando a criação de perfis para cada tipo de equipamento.

Um perfil engloba a principal função do equipamento e as principais informações para o tipo de equipamento.

Um exemplo é o perfil para roteador Cisco. Neste perfil encontram-se os valores de referência do SNMP para quantidade de octetos nos sentidos de entrada e saída das interfaces de um roteador, no uso do processador e na utilização da memória *RAM*. Dessa forma, em ambiente de produção, todos os roteadores fabricados pela Cisco utilizariam as mesmas configurações de coleta de dados [4].

O *cacti* utiliza *XML* como linguagem de formatação dos perfis que, uma vez criados, são, então, armazenados em um banco de dados, facilitando o seu acesso. A utilização do *XML* como padrão flexibiliza a utilização do *cacti*, pois permite que qualquer usuário crie um novo perfil para um equipamento específico, aumentando a capacidade de monitoramento do *cacti*.

Para coletar os dados o *cacti* utiliza dois processos. O primeiro é um *poller* escrito em *PHP* que solicita as informações de cada equipamento e as processa, somente após a coleta de todos os valores de todos os equipamentos. Este processo deve ocorrer a cada 300 segundos, pois este é o tempo fixado para o calculo das consolidações de dados. Quando o processo de coleta e processamento for superior aos 300 segundos é utilizado o segundo método: uma aplicação residente na memória, que lê e processa os dados de cada equipamento, independentemente. Esta solução é a mais indicada para grandes ambientes computacionais. [4]

No processo de armazenamento dos dados consolidados o *cacti* utiliza o formato *Round Robin Archives (RRA).* Este formato é ideal para armazenagem de dados de um determinado período de tempo, por exemplo, mil valores em cinco minutos ou mil valores em duas horas. A diferença irá acontecer na forma com que os dados são consolidados. Por ser um formato circular de escrita de dados, os novos dados irão subscrever os mais antigos e com menor significado, mantendo os arquivos de dados com tamanho fixo.

A interface com o usuário é amigável e organizada, podendo serem, ainda, criados perfis de usuários, definido o grau de acesso ao *software*.

Com uma atuação bem definida, mesmo com as possíveis customizações, o *cacti* não fornece uma visão integrada da conexão cliente servidor, pois não possuí uma forma prática de monitorar todos os clientes em um grande ambiente computacional, já que a inserção e remoção de equipamentos não acontecem automaticamente.

O monitoramento de *links* se limita às informações disponibilizadas pelo SNMP. Estas informações não incluem os protocolos, serviços ou *hosts* que estão utilizando o meio. Estas informações são de fundamental importância para um diagnóstico preciso do estado de um ambiente computacional e, qualquer modificação envolve a criação de *scripts* para coletar os dados.

1.3.3 Ntop

Idealizado com um sensor para coleta de dados da utilização da rede, o *ntop* utiliza a captura dos pacotes que estão passando pela interface de rede do *hos*t, onde ele está instalado, para produzir relatórios [6].

Para viabilizar a captura dos pacotes existem duas formas de integrar o *ntop* ao ambiente a ser monitorado [6].

A primeira é o posicionamento do *host* onde está instalado o *ntop*, que, a partir deste momento, fica conhecido como *ntop* no segmento da *LAN* que se deseja monitorar. Nesta solução a interface de rede do *ntop* deverá operar em modo promiscuo, ou seja, processar pacotes que não foram inicialmente encaminhados para ele. Esta abordagem possui uma desvantagem: a possibilidade da perda de pacotes em redes com alta taxa de pacotes Esta perda pode ocorrer devido a sobrecarga do hardware onde o *ntop* está instalado.

A segunda forma é o posicionamento do *ntop* entre os segmentos que se pretende monitorar. Esta topologia reduz a perda de pacotes pois coloca o *ntop* como meio de ligação, coletando toda a informação que ele conseguir processar, mas apresenta a desvantagem de criar um gargalo na rede, pois os pacotes que não forem processados pelo *ntop* serão descartados. Assim tem-se que definir qual é o melhor cenário: informações não tão precisas, sem comprometer o desempenho da rede; ou, informações exatas, mas com possibilidade de comprometer o desempenho da rede.

Tendo como principal característica o monitoramento da utilização da rede, o *ntop* não apresenta nenhum recurso para o monitoramento de *hosts* e serviços; somente a caracterização do tráfego é importante neste *software* [6].

O formato utilizado para armazenar os dados sofreu modificações nas diferentes versões Inicialmente, era utilizado um formato próprio de armazenagem, dificultando a integração do *ntop* com outras ferramentas; mas, atualmente, o formato foi modificado para utilizar os *RRAs*, possibilitando a criação de gráficos melhores e uma maior integração com outras ferramentas.

O *ntop* possuía uma grande limitação em suas versões iniciais: o monitoramento feito somente do segmento onde estava conectado. Esta limitação foi extinta com a implementação do protocolo *NetFlow* ao programa, permitindo ao *ntop* receber dados do tráfego de diferentes equipamentos. O *NetFlow* é um protocolo, proprietário Cisco que, em sua versão 9, foi recomendado como padrão de protocolo de monitoramento pelo *Internet Engineering Task Force (IETF)* [7].

A interface de monitoramento via *web* facilita o acesso ao *ntop*, sendo intuitiva e de fácil utilização. Através desta interface é possível a coleta de informações sobre o tráfego monitorado pelo *ntop,* possibilitando o tratamento destas informações por outros programas. Os formatos disponíveis atualmente são texto puro, *PHP* e *Perl*.

1.3.4 Análise das ferramentas

Na tabela a seguir são apresentados os critérios estudados e a classificação de cada ferramenta, onde "5" é excelente, "1" é muito ruim e "–" é funcionalidade não disponível.

Tabela 1: Critérios de comparação

Critério	***Nagios***	***Cacti***	***Ntop***
Monitorar Aplicativos no Servidor	4	2	–
Monitorar Aplicativos no Cliente	2	1	–
Monitorar Hardware do Servidor	4	4	–
Monitorar Hardware do Cliente	2	2	–
Monitorar Equipamentos de Rede	4	4	–
Monitorar Enlaces de Dados	–	–	5
Monitorar Atividades dos Usuários	–	–	4
Facilidade de Instalação	2	5	3
Facilidade de Configuração	1	5	4
Interface com o Usuário	3	4	4
Flexibilidade	2	3	1

Comparando os pontos obtidos por cada aplicativo é possível observar que nenhum deles atende a todos os critérios de forma satisfatória, o que motivou o desenvolvimento deste trabalho.

1.4 Solução Proposta

A proposta deste trabalho é um modelo de referência para o desenvolvimento de novos *softwares opensource* para monitoramento de redes, de forma que os mesmos atendam os critérios da tabela 1 e de forma satisfatória.

1.5 Descrição dos Capítulos

No Capítulo 2 serão abordados os conceitos teóricos necessários para o desenvolvimento do trabalho.

No Capítulo 3 serão apresentados à proposta desenvolvida neste trabalho, seus recursos e a infra-estrutura necessária para sua implantação.

O Capítulo 4 reúne os resultados obtidos a partir do desenvolvimento do modelo.

O Capítulo 5 traz as conclusões deste trabalho e propostas de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Redes de Comunicações

O surgimento das redes de comunicações data do século XIX, com as primeiras transmissões de telégrafo. Com a evolução das ciências ocorreram descobertas que modificaram toda a sociedade através de novas tecnologias de comunicação. Por volta das décadas de 40 e 50 o mundo assistia ao nascimento de um novo tipo de equipamento: eram os primeiros computadores eletrônicos [8].

Com o aumento da utilização dos computadores, por parte das instituições de ensino e empresas, começou o surgimento de novas demandas não existentes até o momento, tais como: necessidade de compartilhamento do tempo de processamento, das unidades de armazenagem e periféricos dos computadores da época [9].

Visando atender estas novas demandas houve a integração entre estas duas tecnologias, originando o que hoje é conhecido como Redes de Computadores.

2.1.1 Elementos das Redes de Computadores

No inicio as redes de computadores eram, normalmente, conexões ponto a ponto, que interligavam centros acadêmicos através de linhas telefônicas [8]. Com o passar do tempo o aumento da complexidade foi inevitável, pois cada empresa possuía um padrão próprio de

interligação dos seus equipamentos. Este cenário exigiu a criação de novos dispositivos para efetuar a interconexão entre as diversas plataformas existentes.

Com o problema da interconexão resolvido surgiram demandas pela melhora do desempenho o que forçou a utilização de novos dispositivos Este ciclo de novas demandas e novas soluções continua ainda hoje o que faz com que se tenha uma gama muito grande de equipamentos fazendo parte das redes de computadores. No entanto, atualmente, eles podem ser agrupados em [9]:

- *Hosts*: Qualquer dispositivo ligado a uma rede é chamado de *host*; mas, é comum associar este termo aos computadores e outros equipamentos que utilizam os recursos da rede.
- Roteadores: São os equipamentos responsáveis pela interconexão das diferentes redes, possuindo recursos de conectividade físicas e lógicas normalmente adaptáveis.
- *Switchs*: Equipamentos dedicados a conexão final entre os *hosts* de uma rede, separando- a em diferentes segmentos físicos.
- *Firewalls*: Equipamentos responsáveis pela filtragem dos dados que entram e saem por suas interfaces
- Meios: Equipamentos de interligação que podem ser cabos de cobre, fibras ópticas ou ainda o próprio ar, fornecem a conectividade entre os outros dispositivos.

Existem várias formas de se interconectar estes equipamentos. Estas formas dependem do propósito, desempenho, ambiente e recursos utilizados. Por este motivo têm-se as diferentes topologias de redes [10].

2.1.2 Topologias Físicas de Interconexão

Uma topologia é um mapa de uma inter-rede que indica os segmentos da rede, os pontos de interconexão e as comunidades de usuários [10]. Não é comum a representação de lugares geográficos em uma topologia, mas eles podem aparecer como num projeto arquitetônico de alto nível que mostra a localização e as dimensões de uma sala, mas não do que ela é feita [10].

As topologias podem ser classificadas como [10]:

- Plana: Onde todos os equipamentos de interconexão possuem as mesmas funções. É a topologia mais fácil de implementar e apresenta uma facilidade de gerenciamento

muito grande, desde que a rede não aumente; sendo então recomendada para pequenas redes.

- Hierárquica: Neste modelo a topologia é dividida em camadas discretas, sendo que cada camada é focada em um conjunto de funções específicas, permitindo a escolha correta dos equipamentos de cada camada. Uma topologia hierárquica típica é composta pelas camadas de núcleo, composta por equipamentos de alta tecnologia, otimizados para a disponibilidade e desempenho, distribuição, onde estão concentrados os equipamentos que controlam o fluxo de informações pela rede e a camada de acesso, formada pelos equipamentos que fornecem as conexões para os usuários da rede.
- Malha: Esta topologia é indicada para os casos onde a disponibilidade é o recurso mais importante de uma rede.

Devido a sua modularidade a topologia hierárquica é a mais indicada para os novos projetos.

2.1.3 Classificação das redes de computadores

Um dos critérios de classificação das redes de computadores é seu alcance geográfico. Dentro deste critério elas podem ser dividas como [11]:

- **Redes locais ou *LANs,*** as redes enquadradas nesta classificação são formadas por *hosts* interconectados respeitando o limite de alguns quilômetros entre as extremidades da rede.
- **Redes metropolitanas ou *MANs,*** são aquelas que cobrem uma área de médio porte como uma cidade ou um grande centro urbano.
- **Redes geograficamente distribuídas ou *WANs,*** são redes que possuem um alcance mundial, ligando países, continentes ou cidades. A maior *WAN* conhecida é a Internet.

Um critério mais abrangente de classificação, que utiliza, além do alcance geográfico, as tecnologias utilizadas para realizar a interconexão, também separa as redes de computadores em LANs e WANs. Sendo que as LANs utilizam tecnologias de altas velocidades e disponibilidade de conexão permanentes e as WANs se caracterizam pelas tecnologias de menores velocidades e conexões por demanda, muitas vezes apresentando um custo direto pelas informações trafegadas.

2.2 Modelos de Comunicação de Dados

Nas primeiras redes de computadores era comum a utilização de tecnologias proprietárias na interconexão de equipamentos de um mesmo fabricante, sendo estas tecnologias normalmente exclusivas de um modelo ou série do equipamento. Este quadro não facilitava em nada a evolução tecnológica, já que a cada nova descoberta era necessário reescrever todo o processo de comunicação. Foi nesta época que os primeiros modelos de comunicação em camadas foram criados para resolver este problema [12].

Estes modelos adotavam a postura de segmentar um grande problema, a conexão entre equipamentos, em pequenos problemas de menor complexidade, buscando facilitar o desenvolvimento de novas tecnologias de interconexão padronizadas. Além de facilitar a solução do problema da interconexão estes modelos forneciam independência entre as camadas, possibilitando a integração de diferentes plataformas em um único equipamento.

A independência das camadas foi possível graças ao conceito de comunicação ponto a ponto, onde cada camada do equipamento de origem se comunicava diretamente com a respectiva camada do equipamento de destino, como fica ilustrado na figura abaixo.

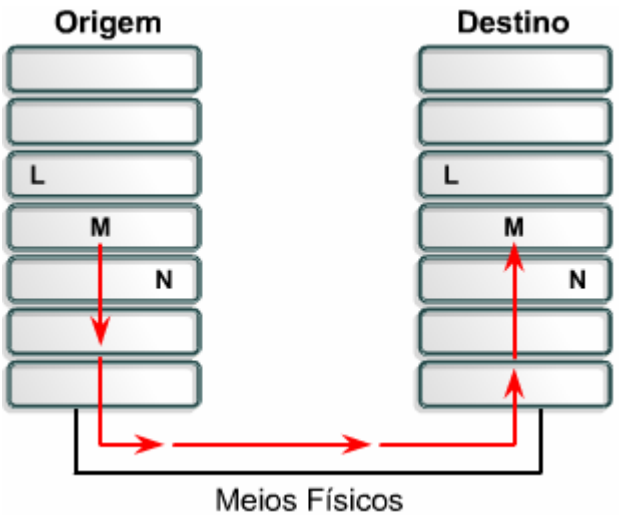

Figura 4 – Comunicação ponto a ponto

Desta forma as camadas L e N, na figura, desconhecem o funcionamento da camada M, tanto na origem como no destino, mas isto não impede que a camada L utilize os serviços da camada M e que a camada N forneça seus serviços para a camada M.

A seguir são apresentados os dois principais modelos de interconexão o modelo OSI, por sua importância acadêmica e o modelo TCP/IP por ser o padrão adotado pelo mercado.

2.2.1 Modelo OSI

Buscando a padronização dos termos e tecnologias a *ISO* estudou vários modelos de comunicação da época e propôs um padrão para o desenvolvimento dos novos equipamentos. Esta proposta foi chamada de *Open Systems Interconnect Reference Model* (OSI), ou Modelo de Referência Aberto de Interconexão de Sistemas.

O modelo OSI é composto por sete camadas, onde cada uma delas engloba os protocolos necessários para a realização da sua função [13]. Cada camada sendo uma peça completa e tendo suas interfaces definidas pelo modelo, os fabricantes agora possuíam um guia de interoperabilidade, permitindo que os avanços de uma determinada camada fossem automaticamente compartilhados com todo o sistema.

2.2.1.1 As camadas do Modelo OSI

O modelo OSI é composto pelas camadas, física, enlace de dados, rede, transporte, sessão, apresentação e aplicação. Cada camada é completa, ou seja, realiza o completo processamento das informações e todas as tarefas referentes ao seu encaminhamento entre suas interfaces. As camadas inferiores fornecem serviços para as camadas superiores no envio e recebimento das informações, utilizando os conceitos de comunicação ponto a ponto entre as camadas, o modelo OSI define as unidades básicas de cada camada, chamadas de *PDUs* ou *packet data units*, elas representam a forma como as informações são agrupadas em cada camada, a figura abaixo representa sua organização no modelo OSI.

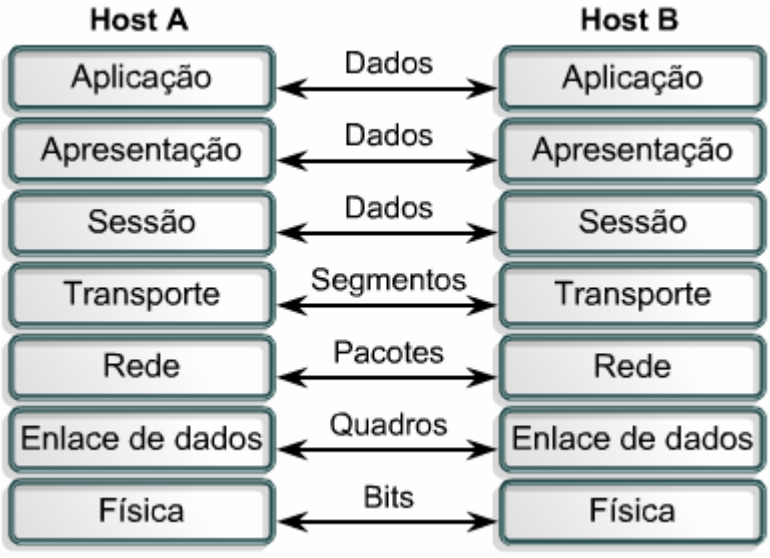

Figura 5 – PDUs das camadas do modelo OSI

Faz-se necessário uma descrição mais detalhada do funcionamento de uma rede baseada no modelo OSI.

Primeiramente, na camada de aplicação, são desenvolvidas as formas de interação que o usuário terá com os serviços oferecidos pela rede. Esta interação deve acontecer por meio de uma interface padronizada para cada serviço, fornecendo acesso aos recursos oferecidos pelo serviço, estas interfaces podem variar de uma simples linha de comando até uma interface gráfica com menus, apresentando como requisitos a facilidade de uso e padronização dos acessos. Na camada de apresentação são tratados dos formatos de armazenamento e transmissão da informação, tais como, codificação de vídeos, codificação de áudio, criptografia, compressão de dados e compressão de imagens. A camada de sessão estabelece as relações entre os aplicativos e as conexões da rede física, mantendo as informações necessárias para o correto encaminhamento dos dados.

As três camadas superiores atuam muito pouco na forma de transmissão dos dados, ficando mais focadas nas atividades relacionadas com a manipulação das informações, já as quatro camadas inferiores fornecem os meios efetivos para viabilizar o transporte da informação entre os equipamentos.

A camada de transporte cria canais virtuais no meio físico para garantir o envio e recebimento correto dos dados, devendo ser implementados controles para detecção e correção de erros, tais como, confirmação de entrega, retransmissão dos dados, segmentação da informação na origem, reconstrução das informações no destino e negociação das taxas de transmissão.

A camada de rede fornece os caminhos físicos necessários para a camada de transporte, podendo utilizar dois conceitos de comutação no momento da transmissão, a comutação por circuitos e a comutação por pacotes. A comutação por circuito mantém um meio exclusivo e único para a transmissão e recepção dos dados entre a origem e o destino, mantendo a ordem de cada unidade transmitida, ou seja, o primeiro pacote enviado será o primeiro pacote recebido, nesta tecnologia quando acontece um atraso, ele atinge todos os pacotes da mesma forma. A comutação por pacote utiliza algoritmos para definir qual é o melhor caminho para a transmissão de cada pacote em particular. Assim não há garantias de que o primeiro pacote enviado será o primeiro pacote recebido. Neste caso um atraso só afetará os pacotes que seguem pelo mesmo meio, na comutação por pacote o destino utiliza os serviços da camada de transporte para determinar a ordem correta para a remontagem dos dados. Para a definição da melhor rota a camada de rede deve utilizar um esquema de endereçamento lógico e hierárquico.

Na camada de enlace o foco está voltado para as topologias lógicas, definindo o controle de acesso ao meio, identificação física dos equipamentos e um conjunto muito básico de detecção e correção de erros, pois nesta camada ocorre a ligação entre as informações lógicas e o acesso ao meio físico real. A identificação física dos *hosts* é definida como forma de controla o acesso ao meio físico, determinando a origem e destino dos quadros.

As topologias lógicas são divididas em dois grupos, *broadcast* e passagem de *token*, na primeira a informação é distribuída em um meio compartilhado pelo primeiro equipamento que acessa o meio, sendo de maneira probabilística a determinação de quem irá utilizar o meio. Na passagem de *token* é utilizado um conceito determinístico na definição de qual *host* irá utilizar o meio, ou seja, somente o *host* que possuir o *token*, um quadro especial que o autoriza enviar informações pelo meio, reservando uma fatia de tempo fixo para cada *host* que compartilha o mesmo meio.

A camada física define a forma e os métodos relacionados com a conversão dos bits em sinais eletromagnéticos que possam ser transmitidos pelos diferentes meios de transmissão, como o ar, cabos de par metálico e fibras ópticas. A codificação dos dados representa um fator importante para o desempenho da transmissão, pois permite a utilização de freqüências maiores para o envio sem perdas das informações. Na camada física são parametrizados os níveis de potência para a transmissão do sinal, possibilitando a melhor utilização do meio de transmissão.

O modelo OSI é utilizado como referência no estudo de redes de computadores, mas não é o padrão adotado pelo mercado, pois sua tardia definição deu tempo para que um outro protocolo igualmente baseado em camadas e aberto se tornasse o padrão de fato no mercado [13].

2.2.2 Modelo TCP/IP

A maioria das redes utilizadas atualmente foi implantada nas décadas de 60 e 70, uma época em que o "estado da arte" era o *design* da rede. Mas, já no início dos anos 70, um novo aspecto das redes começou a ser estudado: as camadas de protocolos, surgindo várias arquiteturas e modelos que permitiam que as aplicações se interconectassem [14].

Um destes modelos foi utilizado pelo *DARPA* na criação do modelo da *ARPANET*. *O DARPA* começou em 1971 seus estudos partindo do *NCP*, um protocolo de comunicação entre *hosts*; até que em 1978 foi desenvolvido o modelo *TCP/IP*. *E*m 1980 o *DARPA* começou a migrar todos os equipamentos da *ARPANET* para a utilização do *TCP/IP* e em

1983, esta tarefa foi completada, nascendo assim o princípio da Internet como a conhecemos hoje [14].

Por ser aberto e integrado com o *UNIX* de *Berkeley* o *TCP/IP* foi rapidamente adotado pelas universidades, que nesta época eram os maiores centros de desenvolvimento de tecnologia. Sendo composto por quatro camadas que demonstra a pilha de protocolos que compõem o modelo *TCP/IP* [14].

2.2.2.1 As camadas do Modelo TCP/IP

A melhor forma de explicar o funcionamento das camadas do modelo TCP/IP é através da comparação com o modelo de referência OSI.

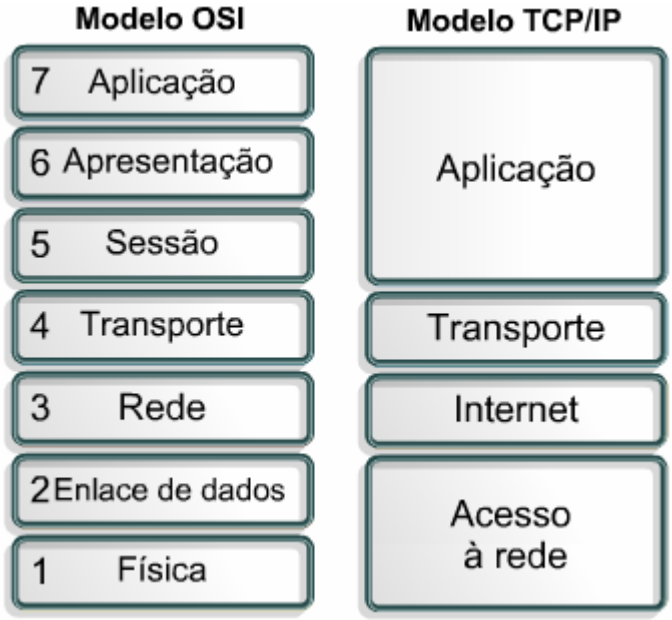

Figura 6 – Modelos de comunicação OSI e TCP/IP

A figura acima apresenta as associações entre cada uma das camadas do modelo TCP/IP com as camadas similares do modelo OSI.

A composição do modelo TCP/IP é de quatro camadas contra as sete do modelo OSI, o que inicialmente parece representar uma simplificação se torna um empecilho, ou fator complicador, para o desenvolvimento de novas tecnologias. Com a fusão de camadas como as camadas 5, 6 e 7 em uma única camada de aplicação o desenvolvimento de novos programas tornam freqüentes o re-trabalho e a incompatibilidade entre aplicativos, um exemplo a ser citado é o envio e recebimento de *e-mails*.

Quando foram desenvolvidos os primeiros aplicativos de trocas de e-mail os diferentes programadores utilizaram diferentes formas de codificação, tornando os textos enviados

muitas vezes ilegíveis no destino. Este cenário só foi modificado com a adoção de um padrão de codificação para a representação dos caracteres. Caso o modelo OSI tivesse sido utilizado os desenvolvedores utilizariam uma codificação padronizada pela camada 6, sem a necessidade de produzirem codificações próprias.

Com a fusão das camadas 1 e 2 do modelo OSI, obtém-se a camada de acesso à rede no modelo TCP/IP, esta camada define todos os padrões relacionados com o meio de transmissão a codificação do sinal, as forma de acesso ao meio e um conjunto completo de algoritmos de correção de erros básicos.

A camada de acesso à rede utiliza uma série de normas e protocolos para realizar suas tarefas, entre eles temos:

- IEEE 802.1, este conjunto de normas rege os padrões de gerenciamento e switch em rede de computadores.
- IEEE 802.2, esta norma define o funcionamento da subcamada de LLC *ou Logic Link Control*, estabelecendo os parâmetros da interface entre o elemento de transmissão e o sistema operacional do computador.
- IEEE 802.3, nesta norma são definidos os parâmetros de controle de acesso ao meio utilizando a tecnologia de CSMA/CD ou *Carrier Sense Multiple Access with Collision Detection*, viabilizando o compartilhamento do meio de transmissão entre diversos equipamentos com um desempenho adequado e um controle de erros eficiente.
- IEEE 802.11, a família de normas IEEE 802.11, definem as transmissões de dados em redes locais e geograficamente distribuídas utilizando o ar como meio de transmissão, o padrão 802.11g utiliza uma freqüência de 2,4GHz para atingir taxas de transmissão de até 54Mbps em cada canal.

A camada de Internet engloba as funções da camada 3 ou camada de rede do modelo OSI, utilizando principalmente o protocolo IP ou *Internet Protocol*, a camada de Internet determina qual é o melhor caminho para o encaminhamento dos pacotes.

Os algoritmos de escolha do melhor caminho são chamados de protocolos de roteamento, e são classificados em dois tipos, os IGP ou Protocolo de Roteamento Internos e os EGP ou Protocolo de Roteamento Externo.

Os IGPs são protocolos de roteamento projetados para serem utilizados dentro de um mesmo sistema autônomo, estabelecendo os melhores caminhos para o tráfego das

informações dentro de uma mesma rede lógica. Os EGPs são utilizados para interligar diferentes sistemas autônomos, permitindo que um pacote trafegue por diferentes redes até chegar ao seu destino. A associação destes dois tipos de protocolos é muito comum e fundamental para o funcionamento da Internet.

São considerados IGPs os seguintes protocolos, RIPv1, RIPv2, OSPF e os padrões proprietários IGRP e EIGRP, estes pertencem a empresa CISCO e o protocolo BGP é um EGP, sendo um padrão aberto, ele pode ser implementado por qualquer indivíduo, permitindo sua utilização em sistema *opensource*.

Um dos papeis da camada de Internet é a identificação exclusiva de cada equipamento conectado na mesma rede, para realizar esta tarefa o protocolo IP utiliza dois padrões o IPv4 e o IPv6, onde a principal diferença é a quantidade de bits utilizada para representar cada dispositivo. No IP versão 4 são utilizados 32 bits para endereçamento enquanto no IP versão 6 são utilizados 128 bits.

A necessidade do aumento do número de bits para representar os equipamentos ligados em rede surgiu do aumento da demanda por conectividade, onde além da mudança no padrão foram implementadas outras técnicas visando reduzir a necessidade por novos endereços IPs.

Algumas das técnicas envolvem o compartilhamento de um IP entre diversos *hosts*, a racionalização dos intervalos de IPs disponíveis e a possibilidade de estabelecer rotas entre sub-redes, redes originadas da divisão de redes hierarquicamente superiores.

A camada de transporte realiza as mesmas tarefas da camada de transporte do modelo OSI, para isso utiliza os protocolos TCP e UDP, sendo o TCP um protocolo orientado a conexão enquanto o UDP não é orientado a conexão.

A orientação à conexão permite um maior controle de todo o processo de troca de dados por uma rede, aumentando a confiabilidade da transmissão. Para prover esta confiabilidade o TCP inicia um processo de verificação de conectividade nos dois sentidos antes de começar o envio dos dados entre a origem e o destino. Além da checagem inicial o TCP fornece um mecanismo para confirmar o recebimento de todos os pacotes enviados, podendo reenviar os pacotes necessários, caso o destino não tenha recebido todos os pacotes corretamente.

A camada de aplicação do modelo TCP/IP engloba todas as funções das camadas 5, 6 e 7 do modelo OSI, sendo responsável pela interface com o usuário, o início e encerramento das sessões entre aplicativos e o formato de armazenamento e transmissão das informações. São exemplos de protocolos de aplicação o HTTP, DNS, SSH e FTP.

A figura abaixo representa a organização do modelo TCP/IP e seus principais protocolos.

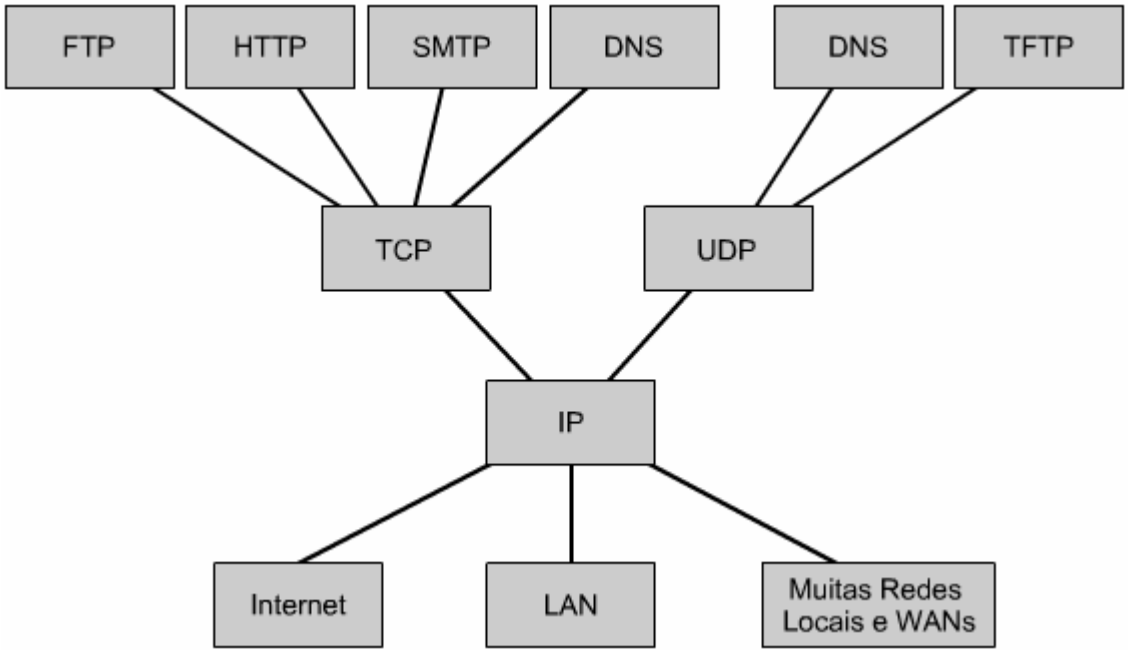

Figura 7 – Estrutura do modelo TCP/IP e seus protocolos

2.3 Desempenho em Redes de Computadores

O desempenho em ambientes computacionais é fator crítico neste trabalho, pois é o foco principal das ferramentas de monitoramento fornecer formas de analisar seu comportamento, nas diferentes situações de operação.

No monitoramento de ambientes computacionais é necessário quantificar o comportamento do tráfego de informações e as necessidades do ambiente para se estabelecer uma relação com o desempenho da rede, pois diferentes ambientes necessitam de desempenhos diferenciados de suas rede.

2.3.1 Variáveis de desempenho das redes

Para alguns, o desempenho pode ser definido como “A rede deve funcionar sem prejuízos para os usuários”, mas para outros o desempenho da rede está diretamente relacionado ao nível de serviço por eles contratados. Em ambos os casos fazem-se necessária a observação de itens específicos. A seguir, tem-se uma lista dos itens que devem ser observados na qualificação do desempenho [10]:

- **Capacidade (Largura de Banda):** Ela representa o valor nominal de um determinado circuito de transmissão, normalmente utiliza como escala o valor de bits por segundo (bps), devendo ser utilizada somente como referência, pois não traduz na realidade a velocidade máxima utilizável do canal. Todas as tecnologias são referenciadas aos seus valores nominais, por exemplo, o padrão FastEthernet é representado por 100Mbps.
- **Utilização:** Informa qual é a utilização do meio em relação ao valor nominal de sua capacidade, é expressa em porcentagem representando uma grandeza qualitativa e não quantitativa.
- **Utilização ótima:** Média máxima de utilização antes de a rede ser considerada saturada
- ***Throughput:*** Quantidade de dados livres de erros transferidos em uma unidade de tempo, normalmente um segundo.
- **Carga oferecida:** Somatória de toda a capacidade de envio de todos os nós da rede ao mesmo tempo.
- **Eficiência:** Relação entre o tráfego corretamente transmitido e o tráfego total
- **Retardo (latência):** Intervalo de tempo entre o momento em que uma estrutura está pronta para ser transmitida e a sua chegada em um ponto específico da rede.
- **Variação do Retardo:** Variação do tempo médio de retardo
- **Tempo de resposta:** O intervalo de tempo entre a solicitação de um serviço de algum serviço de rede e uma resposta ao pedido

2.4 Gerenciamento de redes

Gerenciamento de redes pode ser definido como um conceito geral onde são empregadas várias técnicas, ferramentas e sistemas para auxiliar um profissional no gerenciamento de diversos dispositivos, sistemas e redes [15]. Dessa forma, o gerenciamento pode ser dividido em cinco áreas principais: gerenciamento de falhas, gerenciamento de configurações, gerenciamento de contabilização, gerenciamento de desempenho e gerenciamento de segurança [15].

- **Gerenciamento de Falhas**
  - Funciona detectando, contabilizando e notificando os usuários de sistemas ou redes dos problemas. Em muitos ambientes, por menor que seja o *downtime* é algo inaceitável.
  - Define os seguintes passos para a resolução de uma falha
    1. Isolar o problema
    2. Resolver o problema
    3. Documentar os processos utilizados para isolar e resolver o problema, visando evitar o re-trabalho na busca da solução quando a mesma falha ocorrer novamente.

- **Gerenciamento de Configurações**
  - Monitora as configurações de sistemas e equipamentos. Dessa forma, interliga os efeitos que as diferentes versões de *hardware* e *software* provocam na operação da rede.
  - Qualquer sistema pode ter vários parâmetros de configuração que podem ser importantes para os engenheiros na busca de soluções, incluindo:
    - Versão do sistema operacional, *firmware*, etc;
    - Número e velocidades das interfaces de redes;
    - Número e capacidade dos discos rígidos;
    - Número e velocidade das *CPUs*;
    - Quantidade de memória *RAM*;
  - Estas informações, normalmente, são armazenadas em um banco de dados, que é atualizado em cada modificação do ambiente, possibilitando a inter-relação entre um problema e a modificação de algum parâmetro.
- **Gerenciamento de Contabilização**
  - A contabilização viabiliza que o processamento e os recursos da rede sejam distribuídos de maneira justa entre todos os grupos ou indivíduos que os utilizam.
  - Minimizando os problemas da rede através da divisão dos recursos baseado em suas capacidades.

- **Gerenciamento de Desempenho**
  - A gestão do desempenho visa mensurar e reportar os vários aspectos envolvidos na performance da rede ou sistema. Os passos a seguir são utilizados para se atingir estes objetivos:
  - Obtenção dos dados de desempenho;
  - Estabelecimento das linhas base para análise dos dados obtidos;
  - Definição dos valores de referências da performance. Quando estes valores são ultrapassados, isto indica um problema.
- **Gerenciamento de Segurança**
  - A gestão de segurança atua em duas frentes: primeiro controlando o acesso a certos recursos, tais como rede, computadores e equipamento; em segundo, auxiliando na detecção e prevenção de ataques que possam comprometer a rede ou os sistemas.

2.5 Protocolo *SNMP*

O *Simple Network Management Protocol (SNMP)* provê um conjunto "simples" de operações para facilitar o monitoramento e gerenciamento dos dispositivos de rede, tais como roteadores, *switchs*, servidores, impressoras entre outros. A informação que pode ser monitorada via *SNMP* é muito abrangente indo desde informações sobre a quantidade de tráfego fluindo de uma interface até itens mais diferenciados, como a temperatura no interior de um roteador [15].

2.5.1 Arquitetura

O funcionamento do *SNMP* é baseado nas relações entre duas entidades: o gerente e o agente. O gerente, normalmente, é o equipamento onde é executado o servidor da aplicação *SNMP*. Este servidor tem a função de solicitar as informações definidas nos equipamentos onde está sendo executada a versão cliente do *SNMP*, enquanto os agentes, além de responder as solicitações do gerente, podem enviar informações não solicitadas, chamadas de *traps*, como demonstrado na figura a seguir.

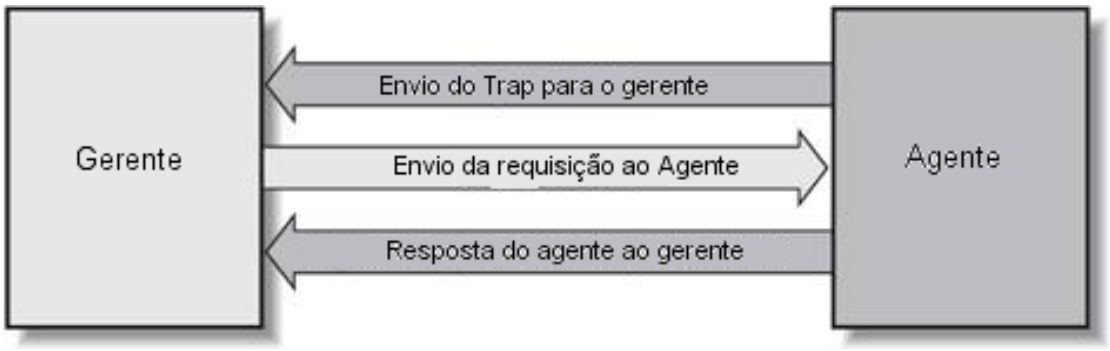

Figura 8 – Relações entre Gerente e Agente no SNMP

A implementação do SNMP em uma ambiente computacional apresenta algumas possibilidades de posicionamento do gerente e na quantidade dos gerentes. Pode-se ainda interagir na forma com que os equipamentos de gerência estão conectados. Temos, inicialmente, a possibilidade de um único gerente ou uma implementação de múltiplos gerentes, onde se tem as seguintes opções:

- Múltiplos gerentes: topologia plana
  - Cada gerente monitora um conjunto de agentes, mas não trocam informações com outros gerentes
- Múltiplos gerentes: topologia hierárquica
  - Cada gerente monitora um conjunto de agentes, enviando atualizações resumidas para um gerente geral que concentra informações de toda a rede.

Na forma de conexão é comum a existência de uma rede exclusiva para gerenciamento. Isto separa o tráfego gerencial do resto da rede, mantendo o monitoramento, mesmo no caso de uma falha na rede de dados que se está monitorando. Esta forma de conexão apresenta um maior custo fornecendo, em contrapartida, uma confiabilidade maior ao sistema.

2.5.2 Estrutura de Gerenciamento da Informação

A definição de uma estrutura padrão de representação das informações foi imprescindível para a adoção do SNMP como protocolo de gerenciamento, uma vez que cada fabricante deveria referenciar os elementos sempre da mesma forma.

A primeira versão da estrutura de gerenciamento da informação, do inglês *Structure of Management Information Version 1* (*SMIv1*) [16] formalizou o padrão de representação das informações para o *SNMP*. A segunda versão ou *SMIv2* [17] apresenta os aprimoramentos da segunda versão do *SNMP*.

A definição dos objetos gerenciados pode ser dividida em três atributos:

- **Nome**
  - O nome ou como é definido, identificador do objeto, do inglês *object identifier (OID)*, identifica exclusivamente um objeto gerenciado e pode ser apresentado em dois formatos: o numérico e o "em texto". Ambas representam a posição do objeto na árvore de objetos do *SNMP*.
- **Tipo e Sintaxe**
  - A formatação da informação de um objeto é definida por com subconjunto da ASN.1, *Abstract Syntax Notation One*. Este formato é utilizado na especificação de como os dados serão representados e transmitidos entre o gerente e o agente, no contexto do SNMP. A ASN.1 é independente de hardware ou software, possibilitando que um agente implementado em Windows 2000 possa enviar suas informações para um gerente implementado em Linux.
- **Codificação**
  - Utilizando o BER, *Basic Encoding Rules*, as informações dos objetos podem ser codificadas, enviadas por um meio de transporte, como a Ethernet e depois decodificado no destino, sem perder seu sentido.

A definição do OID é baseada no percurso que deve ser percorrido na árvore do SMI, partindo da raiz até a folha, onde se encontra o objeto desejado, como ilustra a próxima figura.

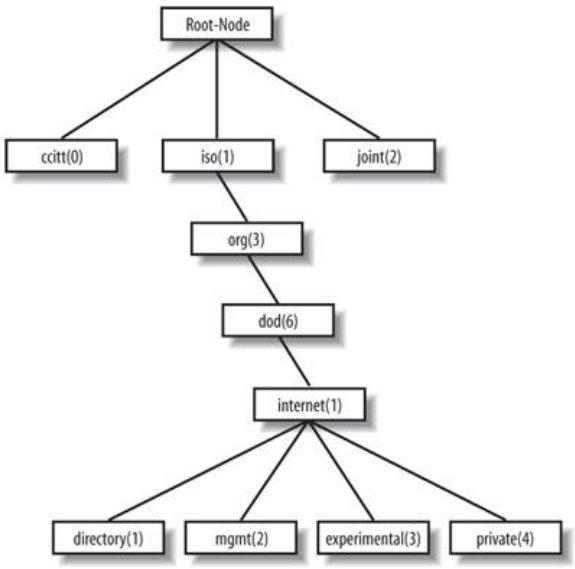

Figura 9 – Árvore de objetos da SMIv1

Dessa forma um objeto que se encontre abaixo do nó internet teria seu OID começado por iso.org.dod.internet na notação textual ou 1.3.6.1 na notação numérica.

2.5.3 Operação do SNMP

A operação do *SNMP* é baseada na troca de informações entre duas entidades. Estas informações devem ser transmitidas por elementos de redes de computadores a fim de fornecer a conectividade necessária entre o gerente e o agente.

Uma das primeiras definições foi a escolha do protocolo da camada de transporte, que deveria ser utilizado. Existiam duas opções: o *TCP* e o *UDP*. A escolha foi baseada na simplicidade, ocupação do canal de comunicação e requisitos de processamento no agente. Por isto a escolha recaiu sobre o *UDP*.

A escolha do *UDP* apresenta alguns problemas, pois o gerente pode fazer uma solicitação e nunca receber a resposta. Para o agente a situação é um pouco mais grave, pois na necessidade de enviar um *trap* o mesmo pode nunca chegar ao gerente e como o gerente não necessita enviar confirmação do recebimento o agente nunca vai saber se o *trap* foi ou não recebido.

O *SNMP* é um protocolo da camada de aplicação do modelo *TCP/IP* ; assim ele é independente da tecnologia das camadas inferiores, atuando em diversas configurações de rede. A figura abaixo representa a integração entre o agente e o gerente no modelo *TCP/IP*.

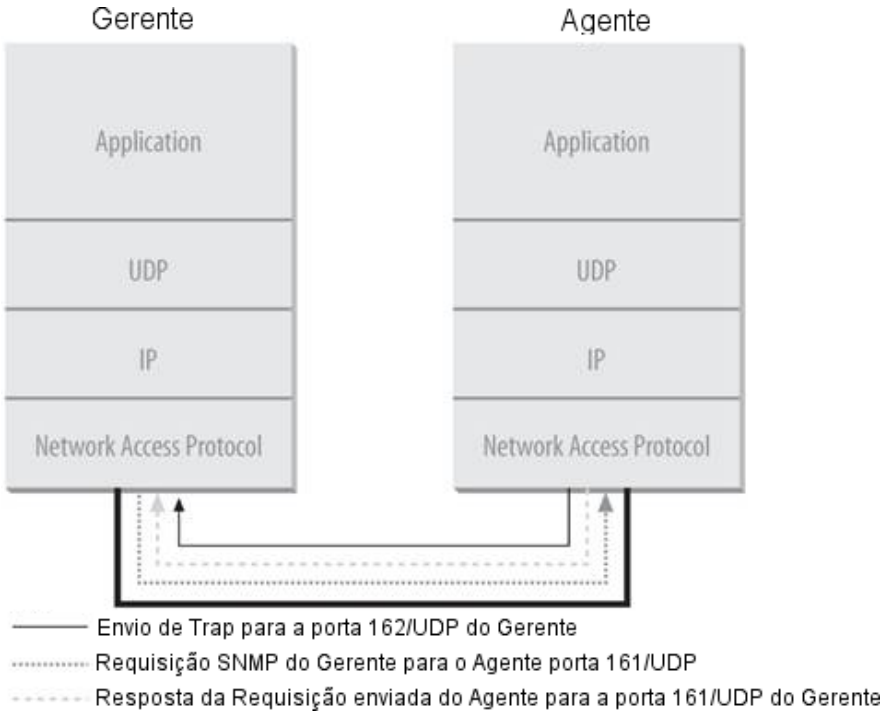

Figura 10 – Interconexão entre o Gerente e Agente SNMP

Nos processos que envolvem o gerente e o agente é possível realizar as seguintes operações [16] [18] [19]:

- ***get* e *getresponse***
    - A operação de *get* parte do gerente, enviando a requisição para o agente. Com o recebimento, o agente tentará processar o pedido, mas caso esteja sob grande utilização, pode ser que não seja possível processar o pedido. Em operação normal o agente terá sucesso no processamento da requisição e irá retornar ao gerente com a informação requisitada através de um *getreponse*. Assim que o gerente receber a informação ele irá processá-la.
- ***getnext* e *getresponse***
    - A operação de getnext permite ao gerente executar requisições sucessivas, dentro de um mesmo segmento da árvore SNMP do agente. Para cada getnext que o agente recebe, ele retorna com um getresponse. Assim que o gerente recebe o getresponse ele envia um novo getnext para o próximo valor na árvore, utilizando a busca em profundidade para encontrar o OID solicitado. A figura a seguir ilustra uma operação de getnext.

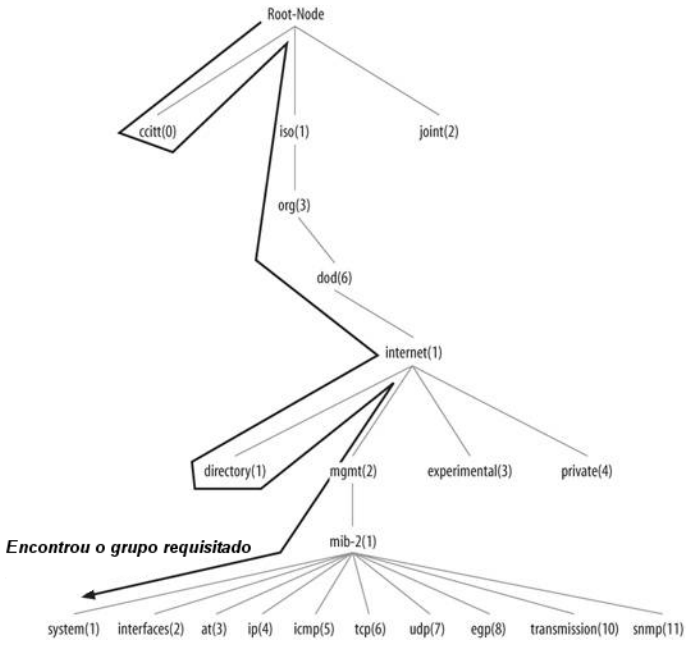

Figura 11 – Busca em profundidade no getnext

    - Após percorrer todo o grupo solicitado o agente retorna um erro como resposta ao último *getnext*. Desta forma, o gerente sabe que não existem mais objetos para receber do agente.
- ***getbulk*** (*SNMPv*2 e *SNMPv*3)
    - Definido a partir da versão 2 do *SNMP* a operação de *getbulk* permite ao aplicativo que atua como gerente receber um conjunto maior de informações, de uma única

vez. A operação *get*, como foi definida, comporta um número reduzido de informações, devido ao tamanho das mensagens ser limitado pela capacidade do agente. No caso de uma tentativa de ler vários objetos falhar, o retorno é uma mensagem de erro vazia; mas, utilizando o *getbulk* o agente envia o máximo que ele conseguir processar.

- ***set***
  - Além das requisições o *SNMP* também oferece um recurso para que os objetos do agente sejam modificados pelo gerente. Esta operação é possível através do comando *set*. Seu funcionamento é parecido com a operação *get*, mas ao invés de solicitar o valor armazenado no agente ele informa o novo valor para o objeto, recebendo uma mensagem de *noError* do agente.
- ***trap***
  - Um *trap* é originado no agente e encaminhado para o gerente. O endereço do gerente deve estar previamente configurado no agente. Uma vez que o gerente recebe um *trap* ele o processa, mas não envia uma confirmação para o agente. O *SNMP* utiliza o *UDP* para realizar o envio de *traps o que* possibilita que um *trap* nunca chegue até o gerente.
  - Alguns das situações que podem gerar um *trap*:
    - Parada de um aplicativo servidor, como o apache (servidor *web opensource*)
    - Mudança do estado de operação de uma interface de rede.
    - Um pico de mais de 5 minutos na utilização do processador de um servidor;
- ***notification*** (*SNMPv2* e *SNMPv3*)
  - A operação de *notification* foi uma tentativa de padronizar o formato do *PDU (Packet Data Unit)* dos *traps* do *SNMPv1*, introduzido nas versões 2 e 3 do *SNMP*
- ***inform*** (*SNMPv2* e *SNMPv3*)
  - Implementado na versão 2 do *SNMP* a operação *inform* permite a confirmação do recebimento de um *trap*.
- ***report*** (*SNMPv2* e *SNMPv3*)
  - Apresentada em uma proposta da versão 2 do *SNMP* , a operação *report,* foi implementada somente na versão 3, permitindo a comunicação de erros no processamento de mensagens de *SNMP* entre os dispositivos.

2.6 Protocolo *NetFlow* [20]

2.6.1 Introdução

O protocolo *NetFlow* foi criado pela empresa Cisco com a finalidade de atender as seguintes características:

- ***Monitoramento de Rede***
  - O *NetFlow* fornece monitoramento de rede intensivo, quase em tempo real. Técnicas de análise baseadas em *Flows* permitem a visualização de padrões de tráfego associados a um único roteador ou a toda uma rede distribuída, possibilitando uma atuação pró-ativa na solução de problemas.
- ***Monitoramento de Aplicativos***
  - O *NetFlow* fornece aos gerentes de rede um relatório detalhado, time-based, da utilização dos recursos de rede pelos aplicativos.
- ***Monitoramento de usuários***
  - O *NetFlow* fornece aos engenheiros de rede a possibilidade de entender, detalhadamente, a alocação dos recursos de rede, para cada um dos usuários, bem como montar perfis de utilização.
- ***Planejamento de Rede***
  - O *NetFlow* pode ser utilizado na coleta de dados por um longo período de tempo, criando a oportunidade de mapear e até antecipar o crescimento da rede, planejando o aumento de roteadores, largura de banda e interfaces mais rápidas. Dessa forma, otimizando os investimentos e reduzindo os custos operacionais da rede.
- ***Análise de Segurança***
  - O *NetFlow* identifica e classifica os ataques de DOS, DDOS, vírus e worms em tempo real, pois modificações nos padrões de rede indicam a existências de anomalias que são claramente representadas no dados do *NetFlow*.

- ***Contabilidade e Cobrança***
    - O *NetFlow* fornece uma forma refinada de medição de utilização de recursos, garantindo uma bem flexível e detalhada forma de contabilização. Provedores de serviços podem utilizar estas informações para cobrar seus clientes.
- ***Armazenamento para análises futuras***
    - Os dados do *NetFlow* podem ser armazenados e posteriormente utilizados por ferramentas de terceiros, para gerar relatórios que irão responder perguntas como: "quem", "o que", "onde" e "por quanto tempo" de forma definitiva.

2.6.2 NetFlow Flows

Cada pacote que é encaminhado, por um roteador, é examinado quanto a um conjunto de atributos comuns aos pacotes do protocolo IP. Estes atributos formam a identidade do pacote ou sua identificação única, definindo sua unicidade ou semelhança com outros pacotes.

Normalmente um fluxo de pacotes IP é baseado em cinco- ou no máximo, sete -atributos que os pacotes tenham em comum.

Os atributos utilizados pelo *NetFlow* são:

- Endereço IP de origem
- Endereço IP de destino
- Porta de Origem
- Porta de Destino
- Tipo de protocolo definido na camada 3 do modelo OSI
- Classe de Serviço
- Interface do roteador

Todos os pacotes com o mesmo endereço de origem/destino IP, porta de origem/destino, protocolo, interface e classe de serviço são agrupados dentro de um único *flow*, tendo seus pacotes e bytes contabilizados. Esta metodologia de determinação de *flow* é escalar, pois uma grande quantidade de informação sobre a rede que pode ser condensada em uma base de dados do *NetFlow*, chamada *NetFlow cache*.

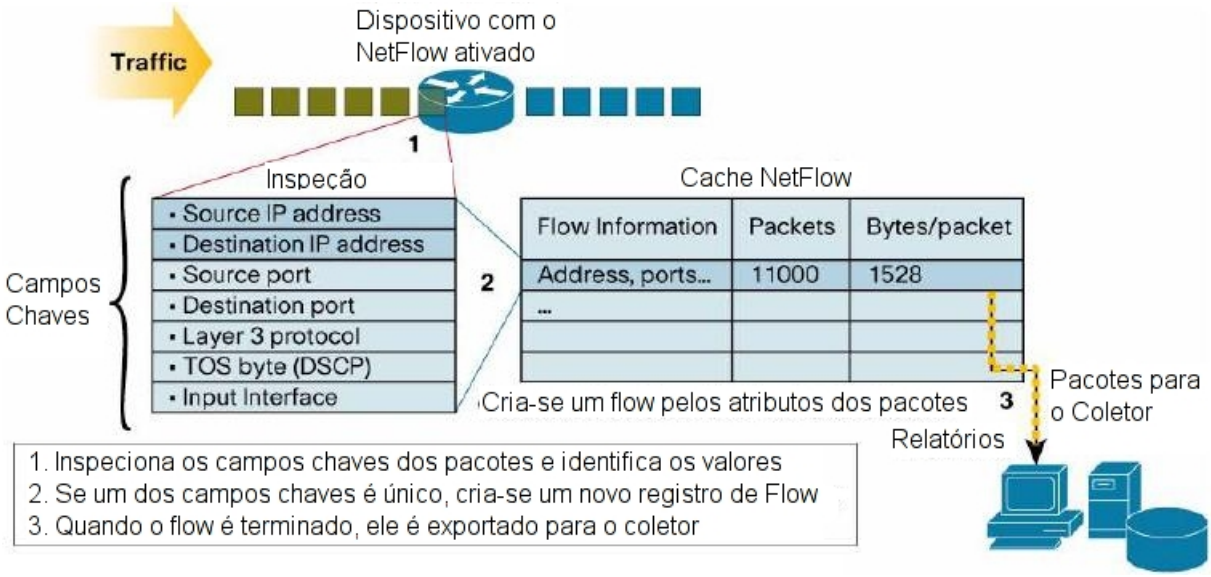

Figura 12 - Atributos chaves do *NetFlow*

Este fluxo de informações, demonstrado na figura acima, é extremamente importante para o entendimento do comportamento de um ambiente computacional.

- Determinação de quem origina o tráfego.
- Determinação de quem recebe o tráfego.
- Mapeamento dos aplicativos que estão utilizando a rede.
- Determinação das prioridades dos tráfegos.
- Determinação da utilização dos equipamentos de rede.
- Quantificação do tráfego.

Informações adicionais que um *flow* pode conter:

- Marcação de tempo, determinação do tempo de vida do *flow*, útil para o calculo de bytes/segundo ou pacotes/segundo.
- O próximo salto do roteamento do pacote, incluindo roteamento entre AS (Sistemas Autônomos) do protocolo BGP.
- As máscaras de sub-rede da origem e do destino, utilizados para o cálculo do prefixo.
- As *flags* TCP para o exame dos *handshakes*.

2.6.3 Operação do cache principal do NetFlow

O principal componente da estrutura do *NetFlow* é o cachê. Ele armazena as informações dos fluxos IP, permitindo que o exportador de *NetFlow* ou outro mecanismo de transporte

envie as informações do *NetFlow* para um equipamento coletor de *NetFlow*. O *NetFlow* opera criando uma entrada no cachê, um registro de fluxo, para cada fluxo ativo. No cachê é mantido um registro de todos os *flows* ativos, estes registros contêm campos que podem ser exportados para os coletores.

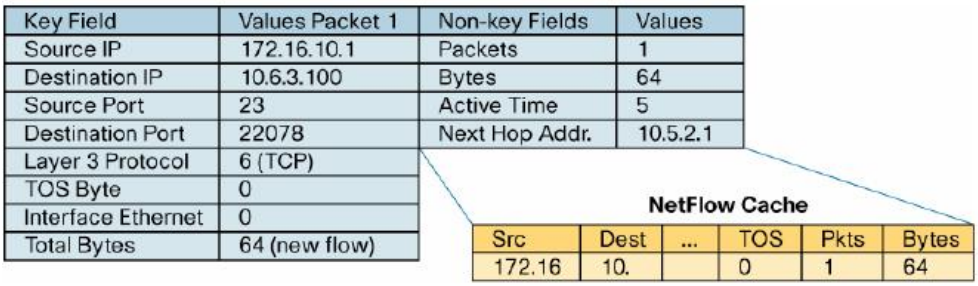

Figura 13 – Criação do registro no cachê

Através da observação dos atributos principais o *NetFlow* determina que o pacote capturado, Figura 13, é um novo *flow*; assim, uma nova entrada é criada no cache.

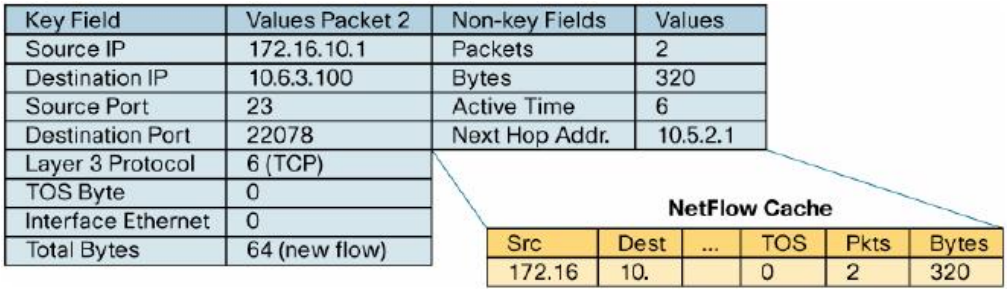

Figura 14 – Atualização do registro no cache

O segundo pacote capturado, Figura 14, pertence ao *flow* já registrado, sendo então somado ao registro já criado.

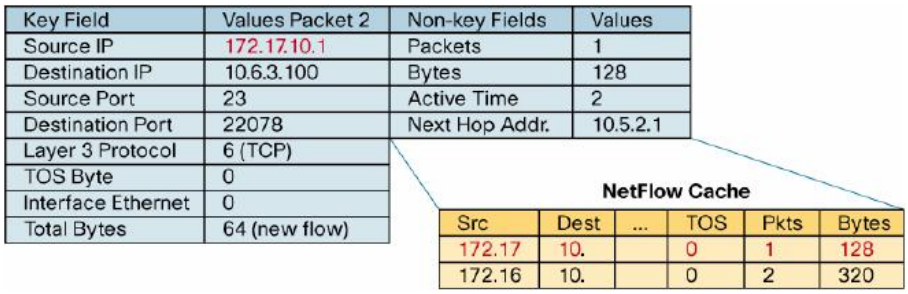

Figura 15 - Criação de novo registro

Analisando os atributos chaves do pacote IP da figura 15 é possível determinar que ele não pertence ao *flow* anterior. Dessa forma, uma nova entrada no cachê é criada para o novo *flow*.

2.6.4 Captura de dados no NetFlow

O *NetFlow* captura as informações de pacotes entrantes e pacotes que estão saindo do roteador.

O *NetFlow* coleta informação dos seguintes pacotes IP entrantes:

- Pacotes *IP-to-IP* packets
- Pacotes *IP-to-Multiprotocol Label Switching* (MPLS)
- Pacotes *Frame Relay-terminated*
- Pacotes *ATM-terminated*

O *NetFlow* coleta dados de todos os pacotes que saem, utilizando um destes recursos:

- Contabilidade *NetFlow* dos pacotes de saída
    - Coleta de dados somente do tráfego IP
    - Coleta de dados de todos os pacotes originados de MPLS e direcionados para redes IP.

2.6.5 Formatos de Exportação do NetFlow

Para o envio de informações o *NetFlow* utiliza o protocolo UDP, pois é mais rápido, possuí um cabeçalho menor, reduzindo a utilização de recursos no envio de informações do *NetFlow*.

Uma vez com uma entrada no cachê o *NetFlow* possui as seguintes regras para remover uma entrada do cachê (entrada expira).

- *Flows* que estão sem atividade por um tempo determinado
- *Flows* que possuem mais de 30 minutos de duração, isso não afeta a comunicação entre os hosts
- Com o cachê cheio, um conjunto heuristico de medidas é aplicado para remover de forma agressiva grupos de *Flows*
- Conexões TCP com flags FIN ou RST

Os *Flows* que expiraram são enviados agrupados e exportadas para os coletores de *NetFlow*, utilizando uma das versões do *NetFlow* disponível.

A Cisco desenvolveu 9 versões do protocolo *NetFlow*, mas somente cinco foram apresentadas e aceitas pela industria. São elas as versões 1, 5, 7, 8 e 9. A versão 1 foi a primeira versão do *NetFlow;* a versão 5 manteve os mesmos recursos da versão 1, acrescentando um maior número de informações de contabilidade; a versão 7 é uma redução da versão 5, feita para uma plataforma específica de equipamentos Cisco; a versão 8 foi a primeira a incluir o envio de informações agregadas, reduzindo a utilização do meio de transmissão e melhorando a performance do protocolo;. e por último. a versão 9. a mais flexível e expansiva, podendo incorporar mais recursos. sem a necessidade de reformulação. As versões 2, 3, 4 e 6 não foram apresentadas ou não possuem mais suporte do fabricante.

2.6.5.1 Versão 1

Foi a versão inicialmente oferecida como formato de exportação, sendo atualmente raramente utilizada, somente em casos de compatibilidade com aplicativos ou equipamentos legados. Substituída pela versão 5.

2.6.5.2 Versão 5

Esta versão adicionou suporte ao BGP, sistemas autônomos (AS) e números seqüenciais de *flows*, seja pela facilidade de configuração ou pelas informações disponibilizadas. Esta versão é adotada como padrão de mercado para desenvolvimento de ferramentas de monitoramento de redes.

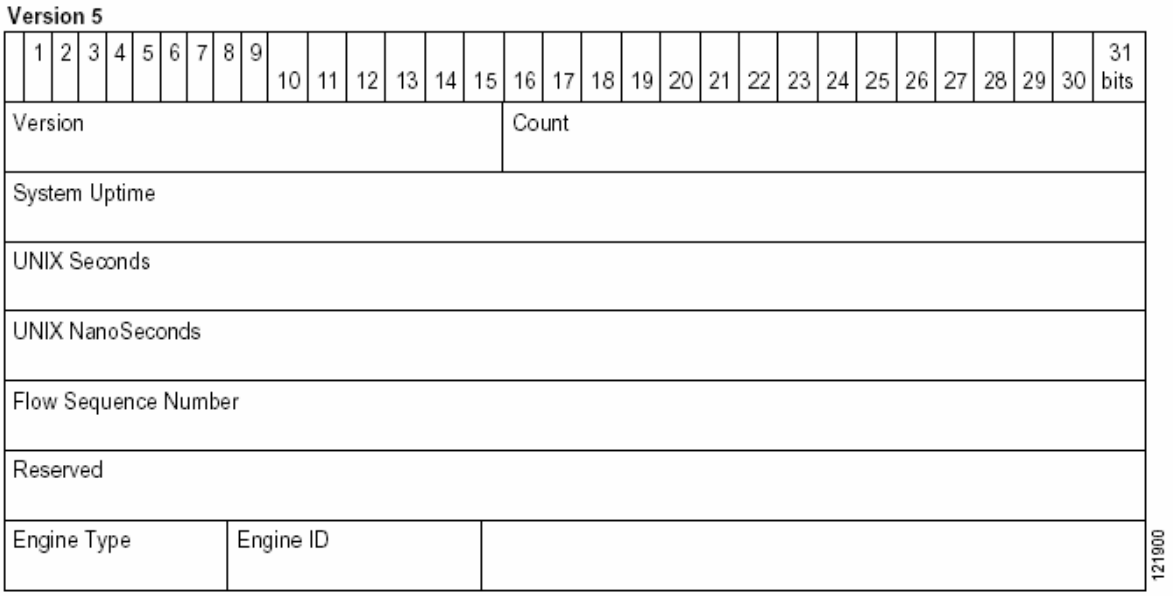

Figura 16 – Cabeçalho dos pacotes NetFlow versão 5

A figura 16 mostra a disposição das informações do cabeçalho dos pacotes *NetFlow* versão 5. Este cabeçalho é similar no formato e nas informações em todos os formatos, com poucas diferenças entre eles.

2.6.5.3 Versão 7

Esta versão é muito similar a versão 5, onde as diferenças estão em um conjunto de recursos retirados da versão 5 que não eram suportados pelas plataformas de hardware para o qual a versão 7 foi projetada. Sendo projetada para trabalhar em *switchs* da família *Catalyst*, séries 6000 com suporte a cartões de chaveamento multi-camadas, e não em roteadores, não havia sentido manter recursos de AS e BGP.

2.6.5.4 Versão 8

Este novo formato de exportação introduziu o conceito de agregar as informações no cachê antes de exportá-las para os coletores, permitindo um pré-processamento dos dados antes do envio. A tabela 2 lista os esquemas de agregação válidos na versão 8 do *NetFlow*.

Tabela 2 - Modos de agregação do *NetFlow* versão 8

	Agregação Router					Agregação TOS				
Atributos	AS	ProtoPort	DstPrefix	SrcPrefix	Prefix	AS	ProtoPort	SrcPrefix	DstPrefix	Prefix
Source Prefix				*S*	*S*			*S*		*S*
Source Prefix Mask				*S*	*S*			*S*		*S*
Destination Prefix			*S*		*S*				*S*	*S*
Destination Prefix Mask			*S*		*S*				*S*	*S*
Source App Port		*S*					*S*			
Destination App Port		*S*					*S*			
Input Interface	*S*			*S*	*S*	*S*		*S*		*S*
Output Interface	*S*		*S*		*S*	*S*			*S*	*S*
IP Protocol		*S*					*S*			
Source AS	*S*			*S*	*S*	*S*		*S*		*S*
Destination AS	*S*		*S*		*S*	*S*			*S*	*S*
First Timestamp	*S*	*S*	*S*	*S*	*S*	*S*	*S*	*S*	*S*	*S*
Last Timestamp	*S*	*S*	*S*	*S*	*S*	*S*	*S*	*S*	*S*	*S*
# of Flows	*S*	*S*	*S*	*S*	*S*	*S*	*S*	*S*	*S*	*S*
# of Packets	*S*	*S*	*S*	*S*	*S*	*S*	*S*	*S*	*S*	*S*
# of Bytes	*S*	*S*	*S*	*S*	*S*	*S*	*S*	*S*	*S*	*S*

O envio de dados, agregados por um parâmetro, facilita a análise e reduz a quantidade de informação transmitida. Exemplificando: para monitorar a divisão de aplicativos no tráfego de rede pode-se utilizar a agregação *RouterProtoPort*, onde somente os atributos porta do

aplicativo de origem, porta do aplicativo de destino e o protocolo *IP* (ex: *TCP*) serão enviados além das informações padrões de tempo e contabilidade.

2.6.5.5 Versão 9

A principal vantagem da nova versão do *NetFlow* é a sua capacidade de trabalhar baseado em modelos. Estes modelos fornecem uma flexibilidade no design do formato de gravação dos registros do *NetFlow*. Esta funcionalidade permite avanços ao *NetFlow,* sem a necessidade de mudanças na base do protocolo; diferente do que ocorre nas versões anteriores do protocolo, onde cada aprimoramento requisitava uma mudança da versão nos programas de exportação e coleta dos dados *NetFlow*.

O trabalho com modelos levou a uma mudança no layout do pacote *NetFlow*. O pacote, agora, é composto por um cabeçalho, seguido por, pelo menos, um ou mais modelos ou *Flowsets*. Um modelo *Flowset* provê uma descrição dos campos que estão presentes no próximo conjunto de dados *NetFlow*. Dessa forma, os campos que irão compor os dados do *NetFlow* podem variar, personalizando os relatórios e reduzindo a utilização de recursos. Este envio pode acontecer no mesmo pacote ou, em qualquer pacote subseqüente.

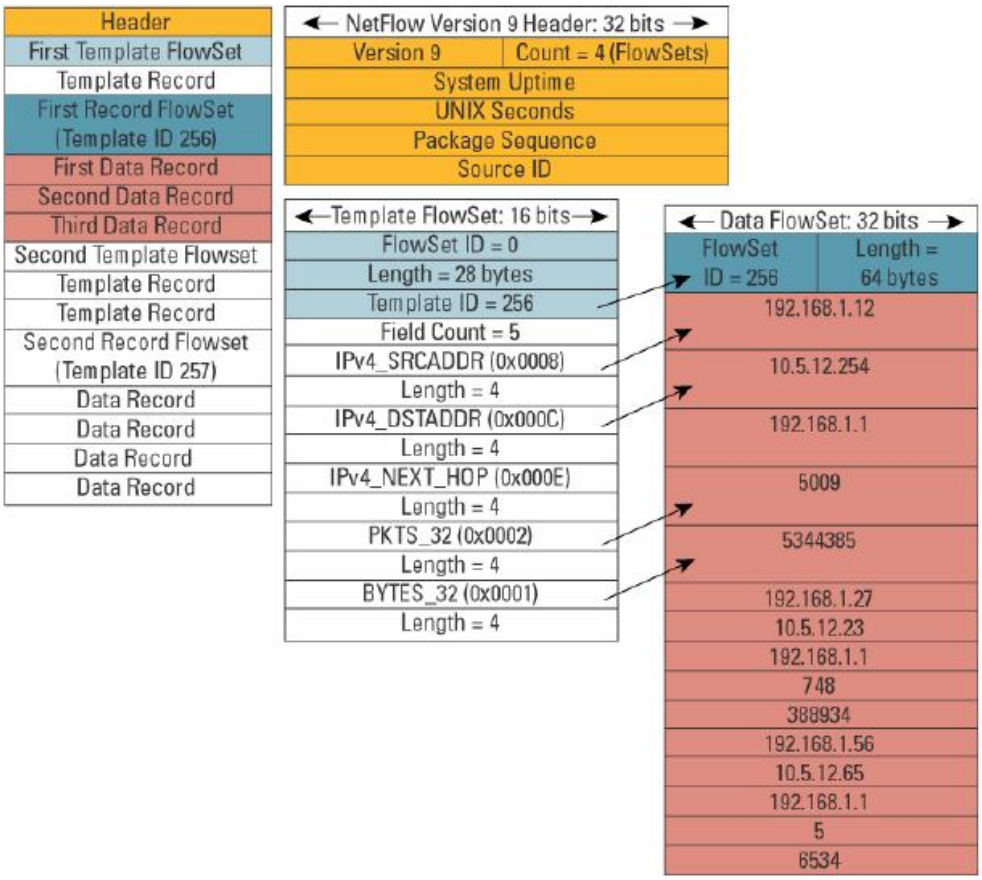

Figura 17 – Formato do datagrama UDP da versão 9 do *NetFlow*

A figura 17 ilustra a organização dos modelos e das informações dentro de um datagrama UDP desta versão do *NetFlow*. Pode-se observar que é possível o envio de mais de um modelo e de mais de um conjunto de informações, em um mesmo datagrama.

O *Internet Protocol Information Export* (IPFIX) é baseado neste formato de exportação de informações.

2.6.6 Infra-estrutura do NetFlow

Para o correto funcionamento do *NetFlow* são necessários, pelo menos, dois tipos de equipamentos:

- Exportadores de *NetFlow*
    - Normalmente representados por roteadores ou *switchs* Cisco são equipamento que possuem o software analisador de tráfego e o sistema de cachê *NetFlow* instalados, tornando possível a captura das informações dos *flows* que atravessam estes equipamentos. Atualmente existem aplicações que transformam um computador normal em um exportador *NetFlow*, aumentando ainda mais a gama de aplicação deste protocolo.
- Coletores de *NetFlow*
    - Normalmente é um computador normal com um software que receba e reconheça os datagramas UDP do *NetFlow* em alguma de suas versões. Vale lembrar que um exportador da versão 5 deve enviar para um coletor que entenda esta versão e vice e versa.
    - O coletor possui uma interface gráfica com o usuário, facilitando a visualização das informações recebidas e processadas pelo coletor.

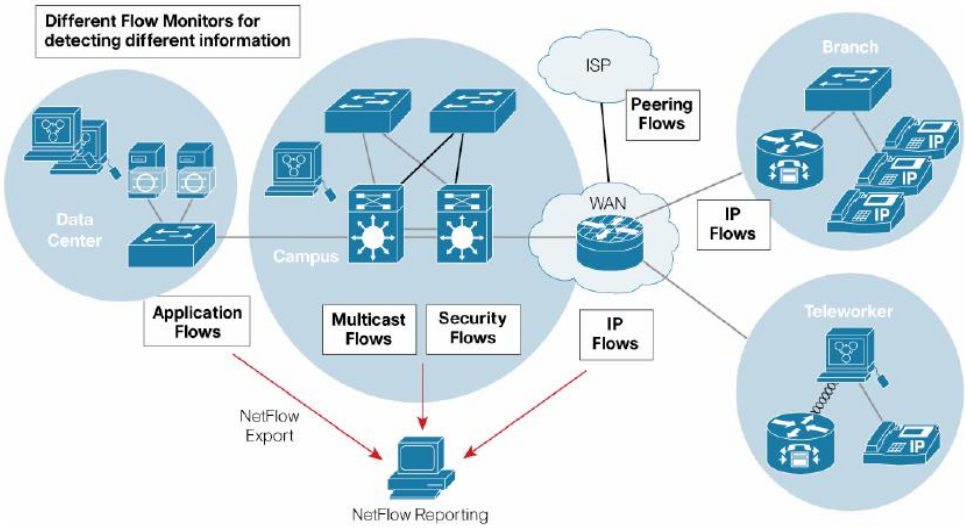

Figura 18 - Esquema básico de interligação entre exportador e coletor *NetFlow*

A figura acima ilustra o processo de interligação e troca de informações entre os diversos exportadores e o coletor central *NetFlow*.

2.7 Ferramentas de software

Para a integração de todas as técnicas de monitoramento requeridas por este trabalho, se faz necessário a utilização de algumas ferramentas de software pré-existentes.

2.7.1 PHP

O PHP é uma linguagem de programação cujo objetivo é a criação de scripts, que são executados do lado do servidor de *web*, sendo que o código é embutido dentro do *HTML*. É um software *opensource* e compatível com um dos mais importantes servidores de *web*, o apache.

Por possuir recursos avançados de lógica e ter fácil acesso à banco de dados esta linguagem de programação é bastante difundida e está disponível aos usuários, através do navegador de *web*, facilitando o acesso aos aplicativos, sem a necessidade de instalar nenhum software adicional no cliente.

O PHP apresenta suporte a XML e programação orientada a objeto, tendo uma sintaxe muito próxima da linguagem C.

Um exemplo de como embutir o PHP dentro do HTML pode ser o código abaixo:

<HTML>

```
<HEAD>
<TITLE>Exemplo de PHP em HTML</TITLE>
</HEAD>
<BODY>
<P>Olá!!!!
<?php
    $nome = 'Cezar';
    $idade = '30';
    echo “Seu nome é: $nome e sua idade é: $idade”;
?>
</BODY>
</HTML>
```

O resultado aparece no tela do navegador como:

Olá!!!!
Seu nome é: Cezar e sua idade é: 30

Por ser integrado ao *HTML* e interpretado pelo servidor de *web* o *PHP* é multiplataforma, estando disponível para Linux, Windows, Solaris e MAC OS.

No PHP não é necessária alocação ou declaração de variáveis, tornando o programa muito versátil e adaptável. Mas este fator necessita de um maior cuidado na escrita dos programas, visando evitar sobrescrever o conteúdo das variáveis.

Além de poder ser embutido no *HTML* o *PHP* pode ser interpretado a partir de um arquivo diferente, pois o servidor irá analisar o resultado do programa *PHP* e converter as saídas para o formato *HTML*. Isto não impede o desenvolvimento de aplicações *standalone* em PHP.

O *PHP* oferece uma *API* para a manipulação do protocolo *SNMP* diretamente, sem a necessidade de uma biblioteca externa no processamento deste protocolo.

A integração do *PHP* com a maioria dos sistemas *opensource* de gerenciamento de banco de dados, como o *MySQL* e o *PostgreSQL*, facilitam o desenvolvimento de sites dinâmicos e seguros, pois reforça os procedimentos de autenticação do PHP.

A oferta de recursos gráficos é muito boa no PHP, possibilitando a criação de interfaces intuitivas e funcionais para as aplicações. Estes fatores associados com a existência de ambientes de gerenciamento de conteúdo em PHP reduzem drasticamente os esforços de manutenção de um sistema baseado na *web*.

2.7.2 PFFlowd

Devido ao alto custo de aquisição de equipamentos Cisco, muitos programadores desenvolveram soluções alternativas para capturar e gerar informações compatíveis com o protocolo *NetFlow*. Uma destas soluções é o *pfflowd*, um aplicativo que atua em conjunto com o filtro de pacotes do sistema operacional *UNIX OpenBSD*, fornecendo informações similares as do *NetFlow* versão 5.

Seu funcionamento é baseado no conceito de redundância do filtro de pacotes do *OpenBSD:* o pf (*Packet Filter*). O pf possuí uma interface lógica no *kernel* do *OpenBSD* chamada *pfsync0.* Esta interface é responsável por encaminhar as tabelas de estado de conexões para o filtro redundante em outro computador. O *pfflowd* intercepta as mensagens de sincronismo de estado e recolhe as estatísticas sobre os *flows* ativos no pf.

Seu processo de configuração é simples e rápido. Toda a configuração acontece na linha de comando, passados como parâmetros para o programa de exportação.

Um exemplo de como executar o pf*flow*d:

#pf*flow*d -n IP_coletor:porta_coletor

IP_coletor: End. IP do computador que está executando o programa coletor de *NetFlow*

porta_coletor: Número da porta onde o programa coletor está associado.

Para realizar a contabilização dos pacotes é necessário que o filtro de pacotes utilize a opção *statefull*, ou seja, monitorando os *flows*, como um roteador. Esta opção é configurada através das configurações *keep state* do pf.

O *pfflowd* exporta as informações nos formato 1 e 5 do *NetFlow*, não apresentando recursos de agregação ou modelos, recursos encontrados somente nas versões 8 e 9 respectivamente.

2.7.3 Flow-tools

O *flow-tools* é uma biblioteca e um conjunto de programas utilizados para coletar, enviar e processar os dados *NetFlow*, gerando relatórios de utilização da rede. Os programas podem ser utilizados em conjunto ou individualmente dentro de um mesmo servidor ou em servidores distribuídos em grandes ambientes computacionais. A biblioteca disponibiliza uma API par o desenvolvimento de aplicativos customizados para exportar pacotes *NetFlow* nos formatos 1,5,6 e as catorze definições da versão 8 e suas subversões [21].

Os programas que utilizam a rede em seu funcionamento utilizam a representação *IP* Local/*IP* Remoto/Porta, para identificar os elementos que compõem o processo de troca de informações, sendo:

- IP Local
  - Endereço IP onde irá ser ligado o processo de captura de pacote *NetFlow*. Em computadores com mais de uma interface de rede é possível selecionar onde o *socket* estará funcionando.
  - Caso seja utilizado o valor "0", o *kernel* do sistema será forçado a escolher um IP para ser utilizado.
- IP Remoto
  - Identifica o emissor dos pacotes *NetFlow*. Caso utilizado o valor "0" serão aceitos pacotes de qualquer endereço IP.
- Porta
  - Define em qual porta o aplicativo será associado para o recebimento ou envio de pacotes *NetFlow*.

Os seguintes programas estão incluídos nos pacotes de distribuição do *flow-tools*:

- ***flow-capture***
  - Programa para coletar, comprimir e armazenar os *flows* exportados por um roteador ou outro dispositivo que possuí a capacidade de gerenciar a utilização de espaço em disco destinado para o armazenamento dos pacotes *NetFlow*.
- ***flow-cat***
  - Faz a concatenação dos arquivos de *flows*. Normalmente estes arquivos são divididos em janelas de tempo de 5 minutos e a utilização do *flow-cat* possibilita unir estes arquivos em intervalos maiores, como um dia, um mês ou qualquer outro intervalo que seja conveniente.
- ***flow-fanout***
  - O *flow-fanout* replica os datagramas *NetFlow* recebidos para outros endereços, facilitando a conexão de múltiplos coletores a um mesmo roteador.
- ***flow-report***
  - Gerador de relatórios dos dados *NetFlow*. Os relatórios incluem pares Origem/Destino IP, Origem/Destino AS e relatórios *top talkers.* Mais de 50 tipos de relatórios são atualmente suportados.
- ***flow-tag***
  - Marca *flows* baseados no endereço IP ou no número de AS. O *flow-tag* é utilizado para agrupar *flows* em redes. As marcações podem ser utilizadas em conjunto com

o *flow-fanout* ou *flow-report* para gerar relatórios de tráfego baseados em redes específicas.

- ***flow-filter***
  - Filtra os *flows* baseado em qualquer um dos atributos exportados pelo *NetFlow*.
- ***flow-import***
  - Utilizado para importar dados de outros formatos como ASCII ou c*flow*d
- ***flow-export***
  - Utilizado para exportar os dados do formato *flow-tools* para os formatos ASCII ou c*flow*d.
- ***flow-send***
  - Utilizado para enviar dados pela rede utilizando o protocolo *NetFlow*.
- ***flow-receive***
  - Recebimento de pacotes exportados pelo *NetFlow*, mas sem o armazenamento do *flow*-capture.
- ***flow-gen***
  - Gerados de dados para teste do coletor.
- ***flow-dscan***
  - Ferramenta para detecção de alguns tipos de ataque de negação de serviço e processos de descoberta de rede.
- ***flow-merge***
  - Une arquivos de fluxos em ordem cronológica.
- ***flow-xlate***
  - Faz a tradução de alguns campos do fluxo.
- ***flow-expire***
  - Expira fluxos usando a mesma política do *flow-capture*.
- ***flow-header***
  - Mostra meta informações em um arquivo de fluxo.
- ***flow-split***
  - Divide arquivos de fluxos para pequenos arquivos baseados em tamanho, tempo ou *tags*.

Exemplos de utilização [22]

Armazenar em /*NetFlow*/bb3 e expirar arquivos mais antigos quando atingir 3G de dados exportados por 192.168.0.1 para a porta 9800:

flow-capture -w /*NetFlow*/bb3/ -E3G? 0/192.168.0.1/9800

Imprimir todos os fluxos do dia 23/10/2003 das 07h00 às 07h30 da manhã:

flow-cat ft-2003-10-23.070000 ft-2003-10-23.071500 | *flow*-print

Imprimir todos os fluxos do dia 23/10/2003 das 07h00 às 07h30 da manhã que entraram pela interface 1 (ver o descritor da interface via snmp) e tiveram origem na porta 80.

flow-cat ft-2003-10-23.070000 ft-2003-10-23.071500 | *flow*-filter -i 1 -p 80 | *flow*-print

Imprimir todo os fluxos do dia 23/10/2003 das 07h00 às 07h30 da manhã que casam com a *access-list flow-acl*, sendo o arquivo *flow-acl: ip access-list standard* destino *permit* 200.201.0.0 0.0.255.255

flow-cat ft-2003-10-23.070000 ft-2003-10-23.071500 | *flow*-filter -f *flow*-acl -D destino | *flow*-print

Imprimir todo os fluxos do dia 23/10/2003 das 07h00 às 07h30 da manhã que casam com a *access-list flow-acl* e fazer um relatório de IP's de destino ordenados pelo campo 2 (bytes).

flow-cat ft-2003-10-23.070000 ft-2003-10-23.071500 | *flow*-filter -f *flow*-acl -D destino | *flow*-stat -S2 -f8

2.7.4 RRDTool [23]

RRDTool faz referência à *Round Robin Database tool*. *Round Robin* é uma técnica que trabalha com um conjunto fixo de dados e um ponteiro para o elemento atual. Pode-se fazer a analogia a um círculo, onde os dados estão posicionados na circunferência e o ponteiro é a linha que liga o dado até o centro do círculo. Quando o dado é gravado o ponteiro salta para uma nova posição na seqüência da anterior. Como se trata de um círculo, não existe final ou começo, podendo ser preenchido infinitamente. Uma vez que todas as posições livres foram utilizadas as primeiras posições preenchidas começam a ser reutilizadas. Dessa forma, o tamanho físico de armazenagem não aumenta com o tempo, necessitando de menor manutenção. O *RRDTool* trabalha com o *Round Robin Databases (RRDs).* Armazenando e acessando dados destas bases.

As informações que são armazenadas em uma RRDs devem estar relacionadas com o tempo entre as medições, por exemplo, km/hora, bits/segundo, Bytes/segundo ou pessoas em uma sala / por minuto, sendo que um programa cria a RRD e um outro faz a inserção dos valores, em intervalos fixos de tempo.

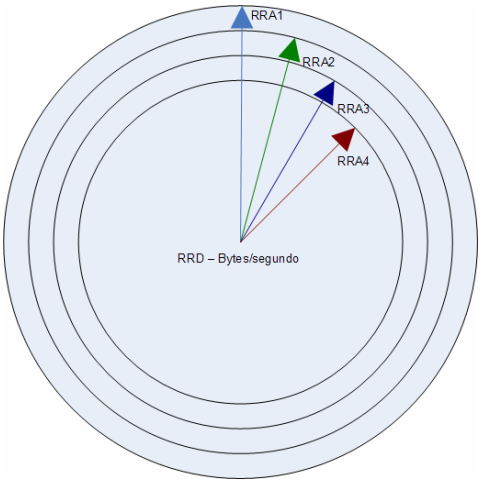

Figura 19 - Exemplo da composição de uma RRD

A figura acima ilustra o fato que dentro de uma única *RRD* é possível existirem vários *RRAs, Round Robin Archives*, onde cada arquivo armazena a mesma informação, mas com intervalos de amostragens diferentes.

No exemplo da figura 19, tem-se uma RRD para armazenar a taxa de transferência de uma interface de um roteador. Cada RRA possui um número de amostras e cada amostra representa a média da taxa no período. Exemplificando: a RRA1 tem como tempo de amostragem cinco minutos e armazena os valores das duas últimas horas, sendo então necessários 24 pontos de armazenagem. O RRA2 tem como tempo de amostragem 30 minutos e ele armazena as últimas doze horas de tráfego, sendo então necessários 24 pontos. Dessa forma é possível armazenar intervalos que vão de uma hora até um ano de duração, somente modificando a taxa de amostragem nos RRAs.

O pacote RRDTool é composto pelos seguintes aplicativos:

- ***rrdcgi***
  - *CGI* para criação dinâmica de paginas *HTML* que contém gráficos das *RRDs*, com a possibilidade de utilizar modelos.
- ***rrdcreate***
  - Programa para a criação de novas *RRDs*.
- ***rrddump***
  - Utilitário para extrair o conteúdo de uma *RRD* no formato *XML*.
- ***rrdfetch***
  - Faz a busca de dados dentro de uma *RRD*.
- ***rrdfirst***
  - Retorna a data da primeira amostra de dados de um *RRA* dentro de uma *RRD*.
- ***rrdgraph***
  - Funções de criação de gráficos da *Round Robin Database*.

- ***rrdinfo***
  - Extrai as informações do cabeçalho de uma *RRD*.
- ***rrdlast***
  - Retorna a data do último registro em uma *RRD*.
- ***rrdlastupdate***
  - Retorna a mais recente atualização em uma *RRD*.
- ***rrdresize***
  - Utilitário para alterar o tamanho físico de um *RRA* e criar um novo arquivo rrd.
- ***rrdrestore***
  - Restaura o conteúdo de um *RRD* utilizando um arquivo *XML*.
- ***rrdtune***
  - Utilitário para modificação dos parâmetros mais básicos de uma *RRD*.
- ***rrdupdate***
  - Comando utilizado para inserir um novo registro dentro de uma *RRD*.
- ***rrdxport***
  - Utilitário de exportação dos dados de uma ou mais *RRD* para o formato *XML*.

3 SOLUÇÃO PROPOSTA

Para o correto gerenciamento dos recursos computacionais em uma rede é necessário monitorar e analisar todo o ambiente. A proposta deste trabalho é um modelo de como o monitoramento e a análise podem ser executados, minimizando a utilização dos meios de transmissão da rede, mas sem perder a qualidade e a precisão nos valores coletados.

3.1 Descrição do Modelo

No monitoramento integrado de um ambiente computacional é necessário coletar dados das duas extremidades de uma conexão, do servidor e do cliente. O monitoramento de servidores e do meio de transmissão é comum nos sistemas atuais, mas monitorar o computador do usuário ainda é problemático em grandes redes, devido à quantidade e a localização dispersa destes equipamentos [24].

Atualmente, é comum o monitoramento distribuído em grandes redes de computadores. Esta solução estabelece que cada equipamento atue como um agente, enviando os dados de seu desempenho para um gerente, quando solicitado. Para melhorar o desempenho desta solução pode ser acrescentado um nível intermediário de gerentes, reduzindo o processamento no gerente central [15]. O modelo desenvolvido neste trabalho utiliza estes mesmos princípios, mas implementa diferentes modos de operação ao sistema. Os modos operacionais Monitor e de Análise, que modificam quais equipamentos são monitorados e a forma como as informações são processadas e apresentadas aos usuários do sistema.

3.1.1 Modo Monitor

A implementação do modo monitor deve oferecer monitoramento constante da infra-estrutura e das conexões entre os clientes e servidores conforme ilustrado na figura abaixo.

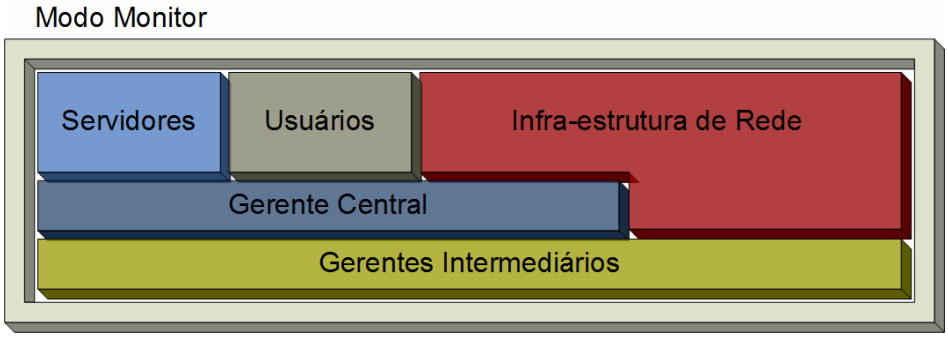

Figura 20 - Integração do modo monitor

Neste arranjo as conexões entre os servidores e clientes são monitoradas pelo gerente principal, que também monitora a infra-estrutura central, já os gerentes intermediários monitoram somente a infra-estrutura de acesso, como exemplificado a seguir.

Para monitorar uma grande rede, como ilustrado na figura 21, é necessário coletar dados de todos os elementos das redes WAN, Local 1 e Local 2, inclusive dos computadores dos usuários.

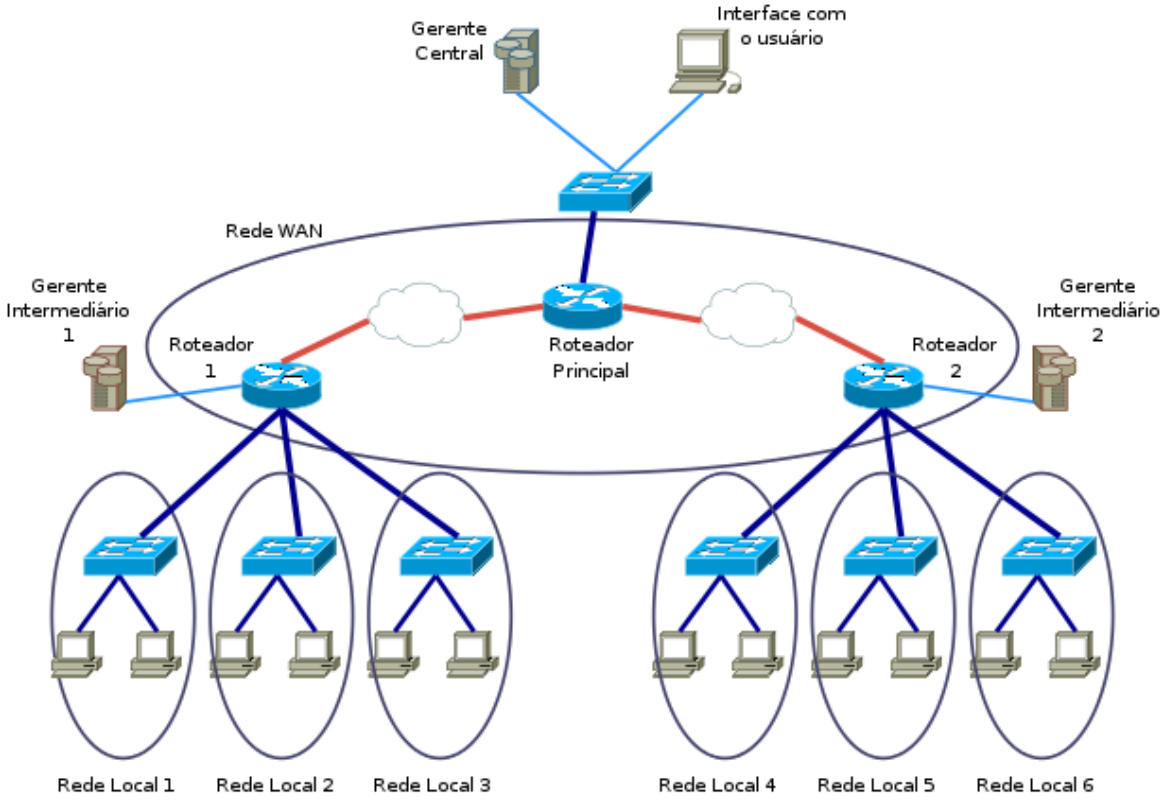

Figura 21 - Gerenciamento Distribuído

Mantendo o conceito original, o monitoramento é dividido entre os três gerentes: o central e os intermediários, de acordo com sua localização; portanto o central monitora a rede WAN, e os intermediários, as redes locais ligadas aos seus respectivos roteadores.

O gerente Central utiliza o protocolo *NetFlow* para monitorar os aplicativos que estão utilizando os *links* da *WAN* e também quais usuários estão acessando estes aplicativos. Através do *SNMP* o gerente central controla os estados e as configurações dos roteadores que fazem parte da *WAN*, como informações de *CPU*, memória e tabelas de roteamento.

Os gerentes intermediários utilizam o *SNMP* para coletar informações dos concentradores das redes locais, como informações da *CPU*, estado de interface, número de computadores por interface e os erros de transmissão detectados nos segmentos da rede local. Para monitorar as transmissões entre as redes locais os gerentes intermediários utilizam o *NetFlow*, coletando assim dados sobre protocolos e aplicativos utilizados nas comunicações entre usuários das diferentes redes locais reais e virtuais.

O usuário tem acesso a todos os dados coletados remotamente via a interface com o usuário, um sistema que fornece acesso a todos os gerentes durante o modo monitor e permite ao usuário controlar o funcionamento do sistema, dos gerentes e dos equipamentos que estão sendo monitorados.

3.1.2 Modo de Análise

A implementação do modo de análise ocorre de forma diferenciada do modo monitor, pois além das operações normais é necessário o monitoramento das atividades no computador cliente, além disso a troca de informações entre o gerente principal e os gerentes intermediários passa a ser constante durante a análise da rede, com demonstra a figura abaixo.

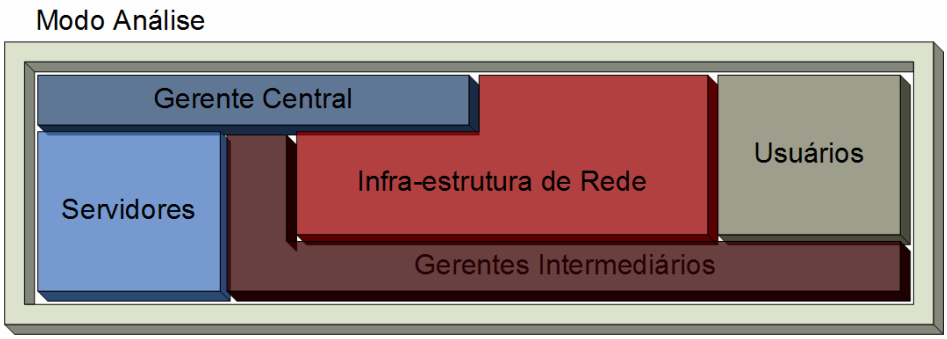

Figura 22 - Componentes do mode de análise

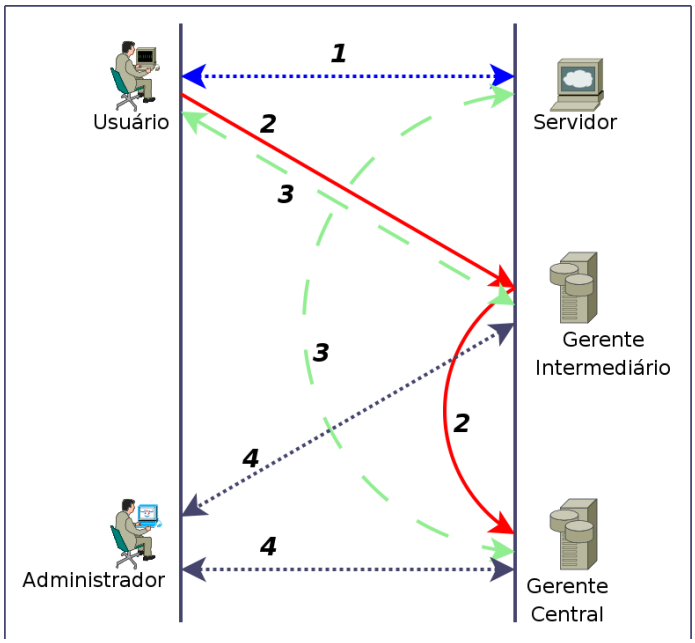

Figura 23 - Modo de Análise

No modo de análise o sistema modifica a forma de coleta de dados, de forma a atender uma demanda específica, além da operação normal da rede Esta demanda é gerada por um usuário que detectou uma falha em algum serviço que ele esta utilizando. A figura 23 demonstra todas as trocas de informações para o sistema sair do modo monitor e entrar no modo de análise. A conexão 1 apresenta uma falha, o usuário informa o gerente intermediário por 2, este encaminha para o gerente central, as conexões 3 representam as solicitações das informações dos respectivos gerentes para o computador do usuário e o servidor enviarem informações de hardware e software para a análise pelo administrador nas conexões 4.

Desta forma, o sistema, durante uma falha, pode acompanhar toda a extensão da conexão, fornecendo uma visão integrada e única do problema, sem a necessidade do monitoramento constante dos computadores dos usuários, ou seja, sem sobrecarga na rede e não importando o quão distribuído é o ambiente, pois todos os segmentos de redes devem ser monitorados.

3.2 Arquitetura do Modelo

Um dos critérios analisados é a flexibilidade do sistema, fundamental devido as constantes mudanças nas tecnologias de transmissão e armazenagem atuais. Para atingir este objetivo, a modularidade é um fator predominante no modelo, bem como a padronização das estruturas de configuração, facilitando a adição de recursos sem a necessidade de reestruturação do modelo. A figura a seguir ilustra esta organização.

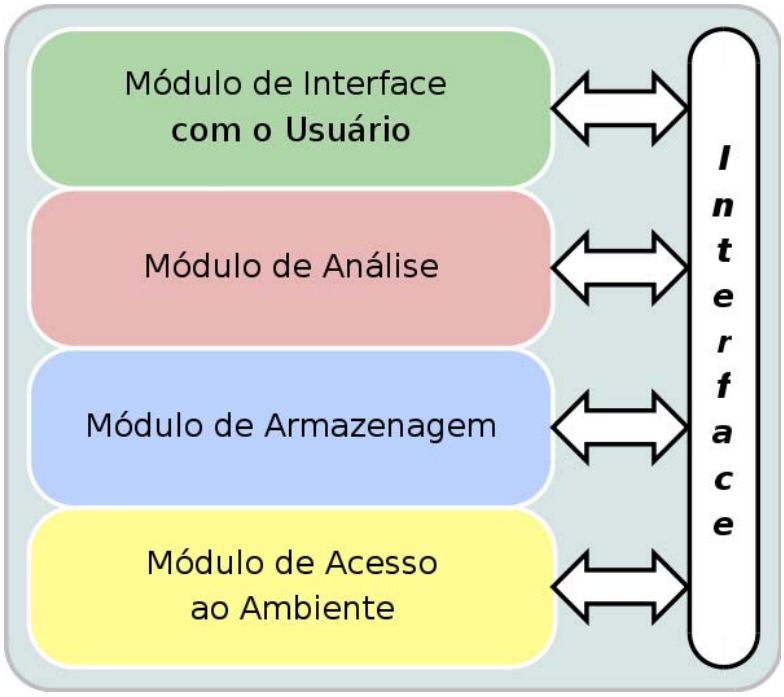

Figura 24 - Organização dos Módulos

3.2.1 Modulo de Acesso ao Ambiente

O acesso ao ambiente compreende todo o conjunto de aplicativos e ferramentas utilizadas como interface entre o sistema e os equipamentos monitorados.

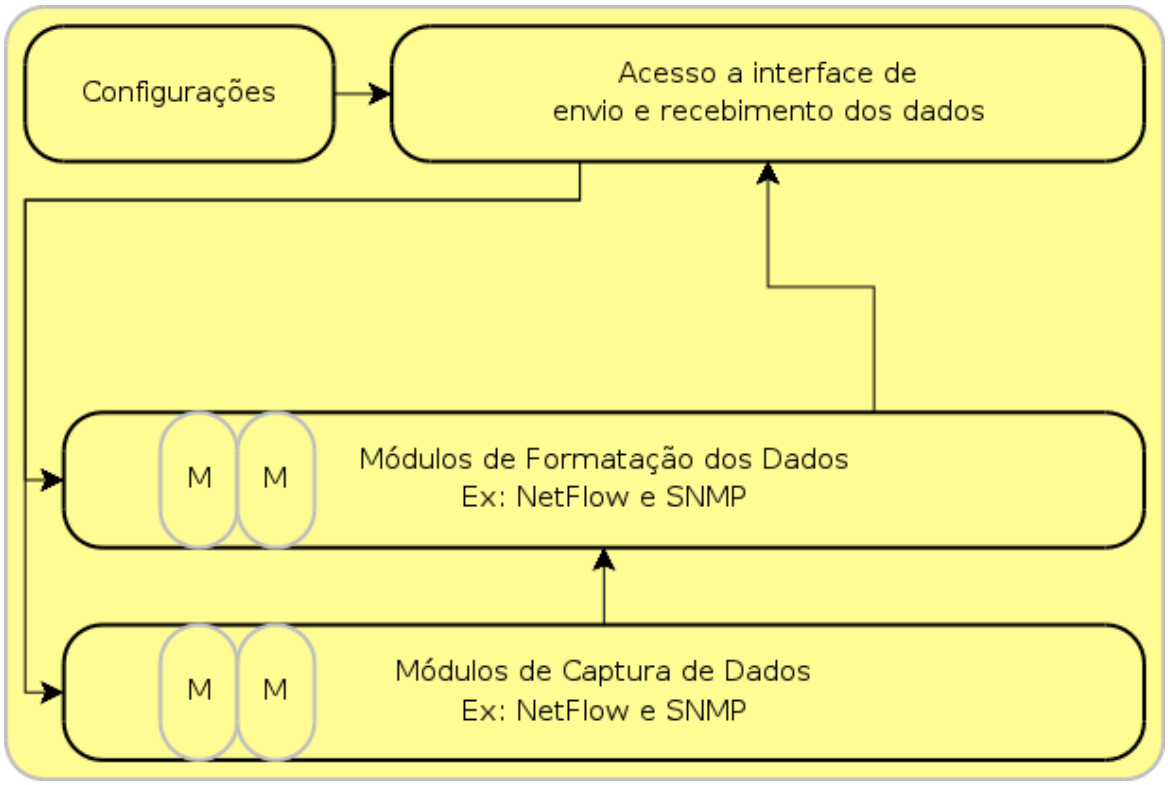

Figura 25 - Arquitetura Interna do Módulo de Acesso ao Ambiente

A figura 25 ilustra a organização dentro do módulo de acesso ao ambiente. As configurações são obtidas via uma interface com todas as camadas, implicando que o usuário pode fornecer as configurações por uma interface gráfica, arquivos armazenados ou ainda uma tabela em um banco de dados.

As ferramentas internas definem os formatos e protocolos de captura de informações, como por exemplo, os protocolos *SNMP* e *NetFlow*, onde existem ferramentas já desenvolvidas e funcionais. Mas, dependendo da situação, pode ser necessário uma solução personalizada, obrigando o desenvolvimento de um módulo interno adicional. Este módulo deve seguir os padrões pré-estabelecidos pelo modelo, visando garantir a flexibilidade e a facilidade originais.

3.2.2 Módulo de Armazenagem

Neste módulo são definidos os padrões de armazenagem dos dados, tornando independente a forma de coletar os dados e o formato de armazenagem, permitindo a utilização de interfaces para diferentes bancos de dados e formatos de armazenagem.

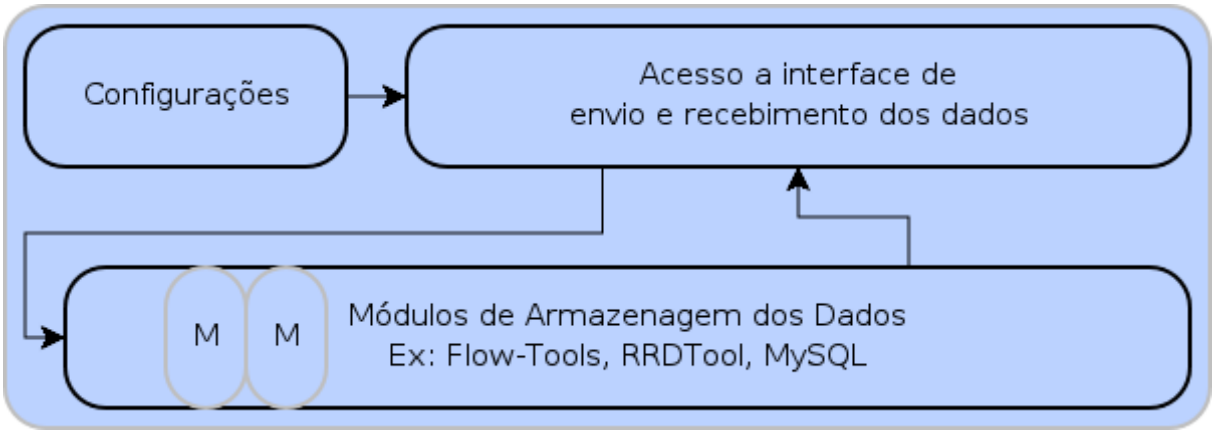

Figura 26 - Arquitetura Interna do Módulo de Armazenagem

Em muitos casos, principalmente em protocolos como o *NetFlow*, é comum a ferramenta de coleta de dados ser a responsável pelo armazenamento. Assim, no módulo de armazenagem pode-se utilizar mais de um formato de armazenagem, dependendo da informação que se quer armazenar. O principal objetivo da divisão em módulos é auxiliar no desenvolvimento de novas ferramentas ou facilitar a integração de ferramentas pré-existentes.

No protocolo *SNMP* as informações não possuem um formato pré-definido de armazenagem, mas utilizando o formato *RRA*, pode-se padronizar a forma como as informações obtidas pelo *SNMP* serão processadas pelos módulos de análise e interface com o usuário.

No módulo de armazenagem são definidos, ainda, o formato dos arquivos de configuração e a maneira de exportar e importar as configurações para outros gerentes.

Este módulo pode apresentar uma diversidade muito grande de soluções e, dependendo da quantidade de dados, podem-se ir de arquivos textos até banco de dados relacionais, sempre visando o aspecto de desempenho e facilidade de configuração.

3.2.3 Módulo de Análise

O processo de análise dos dados está diretamente ligado ao tipo de dados e seu significado. Com a modularidade do sistema é possível adaptá-lo para monitorar diferentes tipos de dados e expressar os resultados de forma padronizada, como por exemplo, a

velocidade de uma interface de transmissão e a quantidade de vírus reportado por um sistema de detecção de vírus em e-mails.

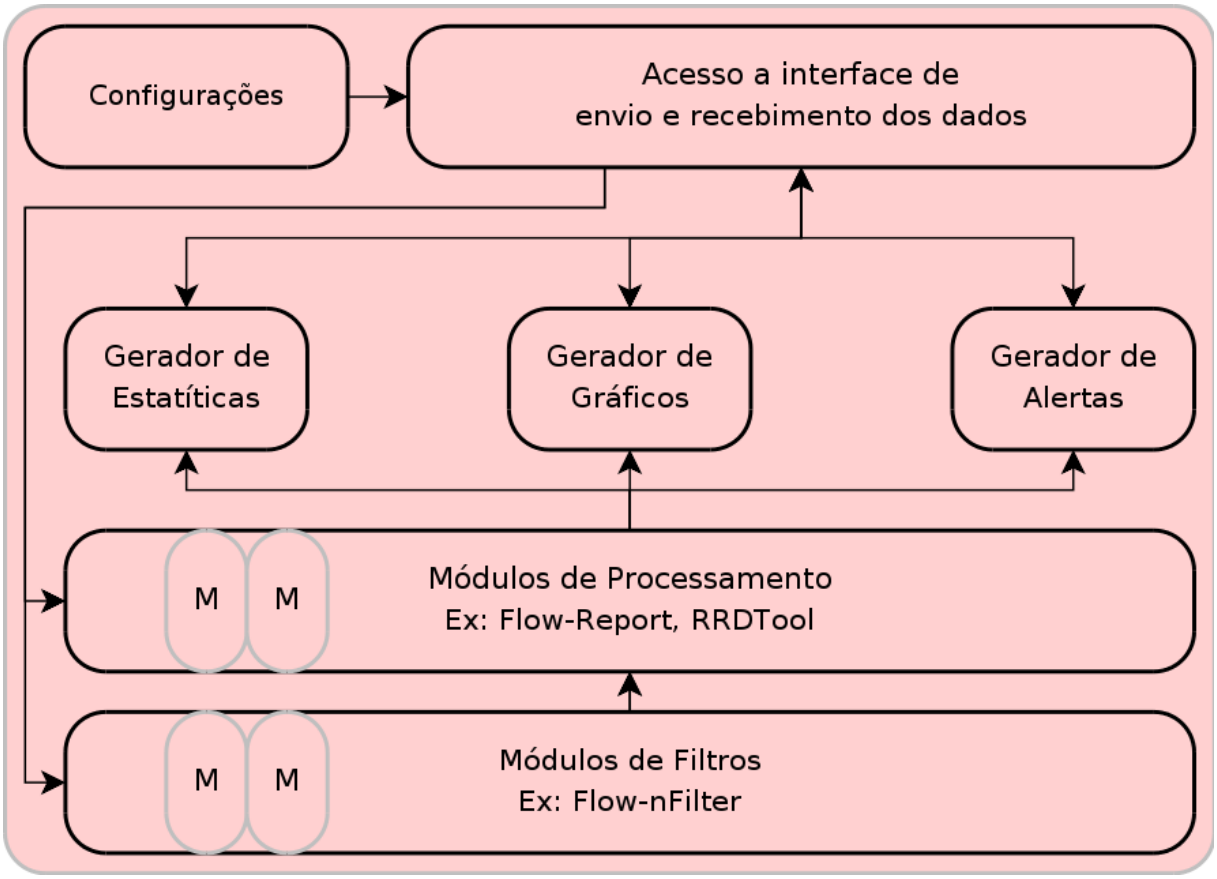

Figura 27 - Arquitetura Interna do Módulo de Análise

O módulo de análise irá gerar todas as informações relativas ao comportamento do ambiente computacional, sendo responsável pela filtragem dos dados brutos, facilitando a interpretação dos administradores do ambiente.

Os processos de geração de alarmes e relatórios são de vital importância para um sistema de monitoramento de rede, pois permite uma recuperação mais rápida da rede, quando são bem implementados. Neste módulo podem ser adicionadas técnicas de inteligência artificial, visando uma maior pró-atividade do sistema.

Os administradores do ambiente devem possuir um excelente conhecimento sobre o funcionamento da rede e seus pontos falhos. Através do modo monitor é possível obter esta visão geral do ambiente, com todas suas particularidades.

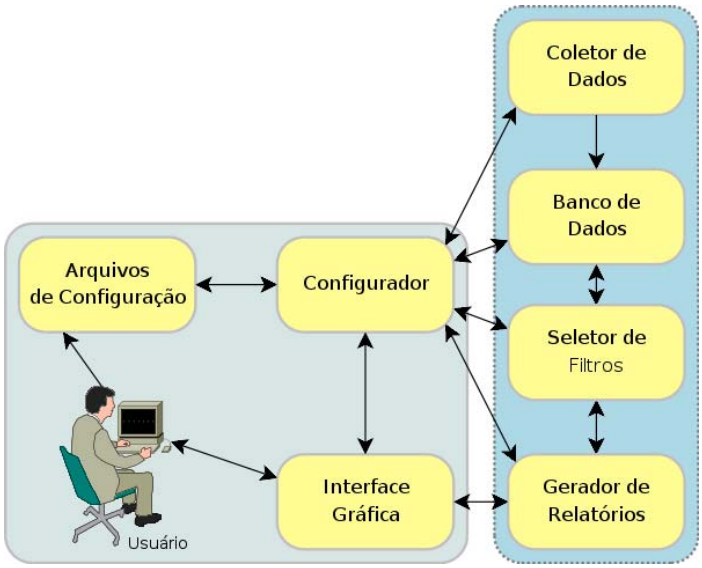

Figura 28 - Recursos do Usuário no Modo Monitor

A figura acima ilustra a relação do usuário do sistema durante a operação normal do ambiente. Existe a possibilidade de realizar consultas pré-configuradas ou criar novos filtros e modelo de relatórios, aprimorando os resultados de uma análise. Todas as ações do usuário são interpretadas por uma interface gráfica, que repassa os comandos para o módulo de análise, que solicita os dados relativos ao módulo de armazenagem, gerando o resultado e apresentando-os na interface com o usuário.

3.2.4 Módulo de Interface com o Usuário

Com uma interface modular, os novos recursos são introduzidos ao sistema, de forma integrada e, principalmente padronizada, reduzindo o tempo de aprendizagem dos novos recursos.

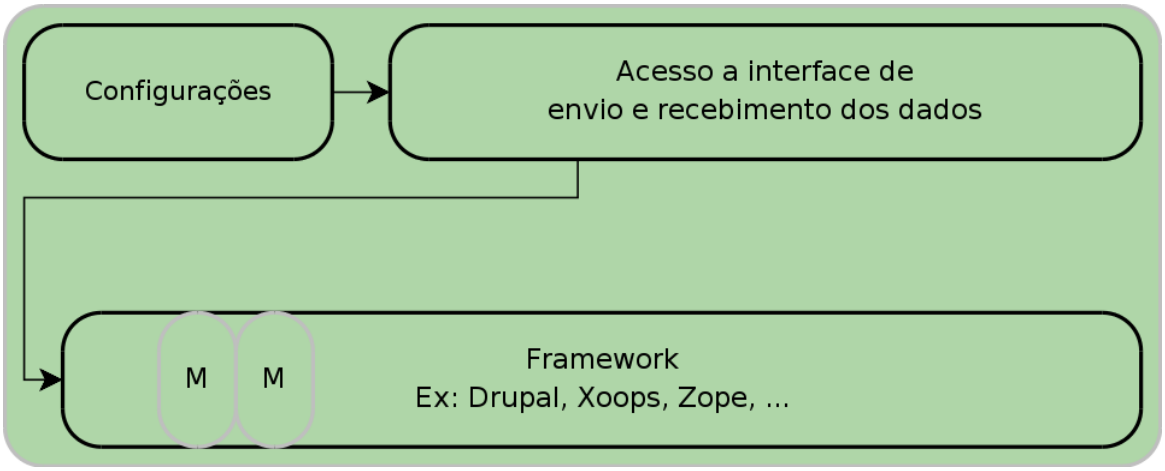

Figura 29 - Arquitetura Interna do Módulo de Interface com o usuário

A interface com o usuário deve ser padronizada e disponível pela rede, idealmente sendo possível monitorar e gerenciar o ambiente de qualquer localidade interna o externa, tomando-se as devidas medidas de segurança nos acessos. Por este motivo é imprescindível a existência de mecanismos de autenticação para o acesso ao sistema.

Criptografia deve ser uma parte importante da interface com o usuário, não somente no armazenamento de senhas, mas também, dependendo dos dados coletados, eles também devem ser protegidos.

Uma interface gráfica é fundamental para tornar o ambiente amigável para o usuário. É fundamental, também, a existência de uma interface auxiliar para a recuperação de desastres, como por exemplo, uma interface texto disponível por uma porta serial do equipamento gerente.

3.3 Ferramentas e Implementação no Ambiente de Testes

Para validar o modelo foi realizada sua implementação em um ambiente computacional real. Este ambiente é representado pela topologia na figura abaixo.

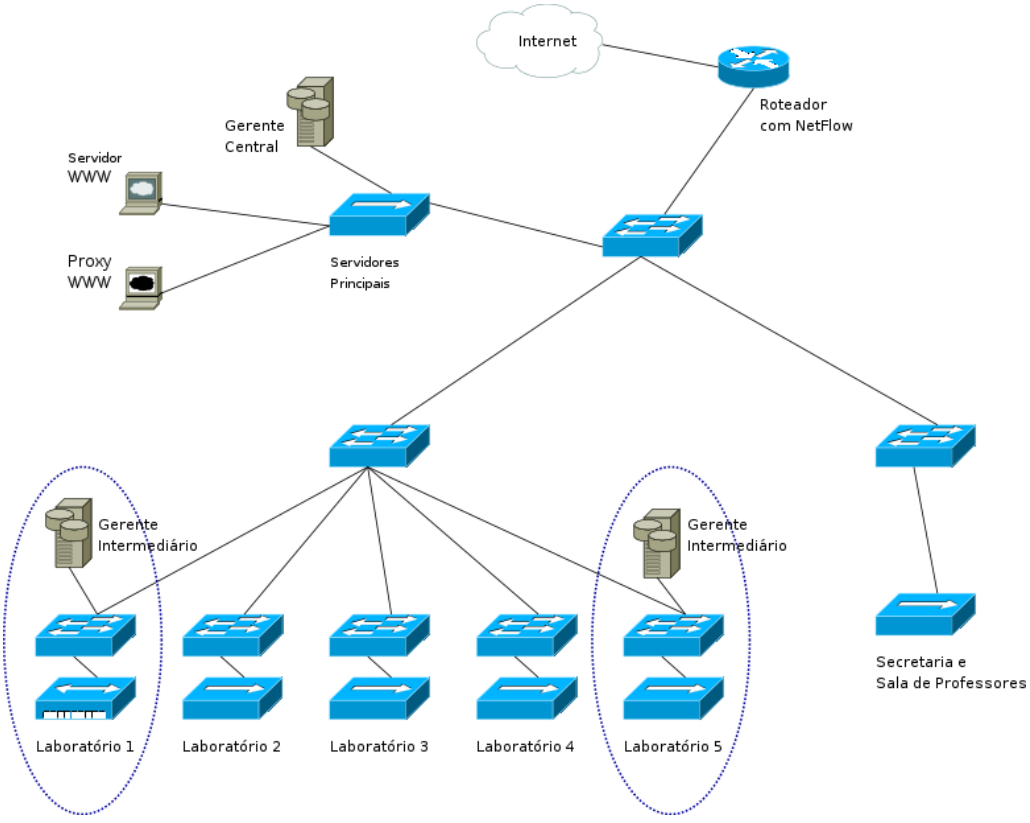

Figura 30 - Ambiente de Implementação

O ambiente de implementação foi a rede de computadores da faculdade Gennari & Peartree na cidade de Pederneiras no Estado de São Paulo. O ambiente utilizava o *Cacti* para o monitoramento, mas devido a suas deficiências, a proposta de implementar uma solução alternativa foi aceita.

A faculdade possui aproximadamente 300 computadores, sendo que, em sua maioria, são computadores de uso exclusivo dos alunos que possuem privilégios limitados, o que, no entanto, não os impede de modificar as configurações básicas dos computadores e nem a instalação de novos programas, gerando uma série de falhas durante as aulas.

Os funcionários do suporte sofrem com a falta de dados dos computadores dos alunos durante uma tentativa de correção de falha, tornando este ambiente propício para testar o modo de análise do modelo desenvolvido neste trabalho. Para fins de comparação o sistema foi implementado em dois dos cinco laboratórios disponíveis

3.3.1 Arquitetura Implementada

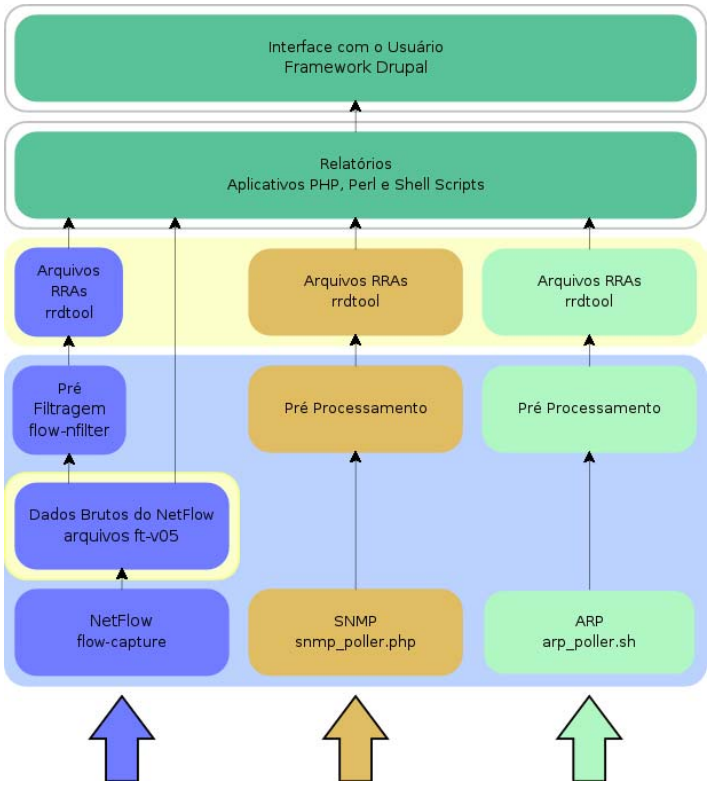

Figura 31 - Arquitetura implementada

A figura 31 apresenta o fluxo dos dados coletados desde o módulo de acesso ao ambiente até a interface com o usuário, citando as ferramentas utilizadas na implementação do sistema no ambiente de testes.

4 METODOLOGIA E RESULTADOS

4.1 Metodologia

Para validar o modelo de gerenciamento proposto, foi realizada a implementação do gerente central em um computador com o sistema operacional *Gentoo* Linux 2006.1 [25], a escolha do sistema deve-se pela versatilidade da distribuição e confiabilidade do *kernel,* Linux versão 2.6.18 [26].

Nos gerentes intermediários o sistema operacional escolhido foi o *OpenBSD* 4.0 [27], este sistema é derivado do BSD UNIX apresentando recursos avançados de segurança, outra vantagem do *OpenBSD* em relação ao *Gentoo* Linux na implementação dos gerentes intermediários é a possibilidade de gerar estatísticas no formato do *NetFlow*, possibilitando um controle maior das atividades dos usuários na rede local.

O processo de instalação do *Gentoo* Linux 2006.1 apresenta um grau de dificuldade moderado para os iniciantes em computação, exigindo conhecimentos sobre particionamento de discos rígidos e da configuração de interfaces de redes, bem como o processo de compilação do *kernel*.

O *Gentoo* Linux oferece uma quantidade significante de personalizações que podem ser aplicadas durante a compilação do sistema, melhorando a compatibilidade com o hardware onde o sistema está sendo instalado, isso compensa a dificuldade inicial de instalação [26].

Uma vez instalado, o *Gentoo* Linux utiliza uma base de dados que pode ser atualizada pela Internet para manter uma relação de todos os programas disponíveis e as dependências necessárias para a correta instalação dos aplicativos [25].

O computador utilizado como gerente central tem recursos modestos para os padrões atuais, sendo composto por um processador AMD modelo *Athlon* de 1,2GHz de freqüência de *clock* e 512MB de memória RAM, mais uma unidade de armazenamento de 40GB. Com esta configuração o desempenho da ferramenta, tanto no processo de coleta, como na geração dos gráficos, foi adequado para as necessidades dos usuários do sistema.

Os computadores que atuam como gerentes intermediários possuem uma composição ainda mais modesta que o gerente central, processador *Intel* modelo *Pentium* de 166Mhz de freqüência de *clock*, somente 32MB de memória RAM e discos rígidos de 2,1GB de espaço total. Estas especificações demonstram que dependendo do ambiente não são necessários grandes investimentos financeiros para implantar um sistema de gerenciamento de rede satisfatório.

No desenvolvimento da interface com o usuário foi adotado o CMP *Drupal* [28], pois este apresenta uma interface modular para o desenvolvimento de sites de *web*, facilitando a inclusão ou modificação dos recursos oferecidos pela ferramenta. Além de contar com diversos módulos externos desenvolvidos por uma comunidade ativa que conta com colaboradores nacionais trabalhando na sua tradução.

O sistema foi desenvolvido principalmente em PHP para manter a compatibilidade com qualquer sistema operacional que o usuário venha a utilizar, pois é somente necessário um navegador de Internet para acessar a ferramenta pela *web*. Para disponibilizar o sistema, foi necessária a instalação e configuração de um servidor de *web,* no computador onde o sistema está instalado.

A escolha do Apache [29] com servidor de *web*, é justificada por sua robustez e eficiência no trato com as necessidades do sistema e dos usuários, tais como, suporte a conexões criptografadas, domínios virtuais, autenticação integrada com o banco de dados *MySQL* e suporte as linguagens PHP e *Perl*. A instalação do apache foi realizada através do sistema de gerenciamento de pacotes do *Gentoo* Linux o *emerge*. O emerge faz uma varredura na base de dados de aplicativos e verifica as dependências referentes à instalação do apache e quais as opções em tempo de compilação estão disponíveis, em seguida realiza a instalação recursiva de todas as dependências até chegar à compilação e instalação do Apache.

O gerente principal e os gerentes intermediários foram habilitados como gerentes SNMP através da instalação do programa *net-snmp*, permitindo o envio de pacotes *snmp-get* e *snmp-get-next* aos computadores reservados para alunos, mantendo o monitoramento dos clientes disponível sempre que requisitado pelo usuário.

Todos os computadores de alunos possuem um sistema de dual-boot, ou seja, podem ser iniciados por dois sistemas alternadamente, Windows 2000 e o Ubuntu Linux, estes dois sistemas apresentam uma implementação das bibliotecas e um agente SNMP operacionais. Através da habilitação destes agentes durante o processo de boot dos dois sistemas foi possível acessar seus recursos de hardware e software.

No roteador central foi habilitado e configurado o protocolo NetFlow em sua versão 5, definindo o gerente central como coletor, assim todo o tráfego de rede é automaticamente reportado para o gerente central, que o processa através da ferramenta *flow-tools*.

O processo de instalação do *flow-tools* é semelhante ao das outras ferramentas, somente sua configuração exige a atenção em alguns pontos principais. A definição da porta onde o coletor estará associado deve ser a mesma escolhida na configuração do roteador, assim como a versão do *NetFlow* esperada.

A configuração e instalação do *rrdtool* devem ser executadas utilizando o *emerge*, de forma a disponibilizar as bibliotecas compartilhadas para os scripts escritos em perl e bash.

Após a instalação e configuração de todas as ferramentas, foram realizados testes no período de três semanas, compreendidas entre os dias 1 de dezembro de 2006 e 22 de dezembro de 2006, testes de desempenho nos processo de correções de falhas nos laboratórios de informática da FGP. Estes testes eram constituídos do monitoramento da atividade dos laboratórios e o tempo de resposta entre a abertura de um chamado de suporte e sua correção.

Dos cinco laboratórios da FGP somente os laboratórios 1 e 5 tiveram o monitoramento baseado no modelo. Os laboratórios 2, 3 e 4 continuaram com a solução anterior, mantidos como referência de desempenho.

Através de um conjunto de treinamento o grupo de suporte técnico da FGP foi preparado para utilizar a ferramenta nos processos de gerenciamento e monitoramento do ambiente de rede da faculdade. Formado por 3 colaboradores de nível técnico o grupo de suporte utilizou a ferramenta para monitorar os laboratórios de teste e quantificou os resultados obtidos.

Foram observados valores de tempo de resposta, tempo de descoberta de uma falha e utilização inadequada do ambiente de rede, foi ainda fornecido um questionário para os

usuários, a fim de verificar as diferenças entre os laboratórios. Ainda foram verificados os critérios contidos na tabela 1, visando à comparação com os sistemas já existentes.

4.2 Resultados

Durante os testes ficou claro que o sistema apresenta alguns fatores de complicação, principalmente quanto à configuração das ferramentas, muitas configurações devem ser realizadas em arquivos texto e não em uma interface gráfica como seria o ideal.

Depois de configurado o sistema se mostrou estável mesmo em um hardware de baixo custo e foi possível monitorar o uso das estações de trabalho com um maior número de detalhes que a solução anterior.

Os resultados mais importantes foram obtidos na redução do tempo entre a abertura de um chamado e a correção efetiva do problema, pois antes o usuário notificava pessoalmente o técnico e este deveria investigar localmente o computador com problemas. Depois da implantação do monitoramento remoto a verificação ocorria no momento do registro da reclamação e através do SMNP o técnico pode observar as mudanças de configuração, os aplicativos que estavam em execução, à utilização ou carga no processador e da memória. Isso permitiu uma melhora significativa nos diagnósticos dos problemas, facilitando a correção dos mesmos.

Em alguns casos o computador com falha não estava acessível pelo sistema, isso prejudicou o processo de correção, onde foi levantado o questionamento de sua eficácia. Nestes casos foram constatados problemas físicos de conectividade, tais como desconexão do cabo de rede, desligamento de um *switch* e problemas com as fontes de energia dos computadores. No caso do desligamento do *switch* os técnicos haviam sido informados pelo sistema da parada do dispositivo, mas devido à forma de alerta, um e-mail para o administrador, não foram tomadas as devidas providências para evitar o problema, sendo necessário à utilização de uma forma mais eficaz de alerta para o corpo técnico.

Os usuários do sistema foram convidados a preencher um formulário de avaliação de desempenho do sistema, onde foram feitas questões sobre o sistema como um todo e a melhora introduzida na administração da rede após sua implementação.

A seguir são apresentados os pontos observados pelos técnicos, bem como suas considerações sobre o sistema e suas opiniões acerca de seu funcionamento.

No quesito desempenho, as avaliações foram positivas, focando as observações na necessidade de baixo investimento em hardware e na estabilidade dos sistemas operacionais escolhidos, pois durante as três semanas de testes foi necessário reiniciar somente uma vez um dos gerentes intermediários, sendo que os outros dois equipamentos não apresentaram problemas.

No item de facilidade de operação, os usuários apontaram uma série de pontos que deveriam ser melhorados para aumentar a produtividade do trabalho de gerenciamento da rede através do sistema, alguns dos pontos onde houve uma convergência entre os técnicos foram:

- Falta de interface gráfica para configuração das diferentes ferramentas, isso tornou o processo de implementação confuso e difícil, visto que a padronização dos arquivos de configuração não foi completa nesta fase.
- Falta de criação de relatórios personalizados, pois os oferecidos ou eram muito completos dificultando sua visualização ou eram resumos, não fornecendo informações com a precisão necessária.
- Intervalo de amostragem padronizado para todos os períodos de tempo, tornando impossível a visualização mais precisa em intervalos passados.
- Falta da possibilidade de criação de um alarme para controlar os desligamentos não autorizados dos computadores.

No impacto causado pela implementação do sistema os técnicos concordaram na melhora da eficiência nas correções das falhas mais comuns e também no monitoramento dos servidores e do link, fornecendo uma visão integrada de toda a rede. Sendo possível a determinação dos pontos de estrangulamento de desempenho da rede.

Visando obter um valor quantitativo do impacto foram medidos os tempos de resposta na correção das falhas nos cinco laboratórios e através da média destes valores foram obtidos dados utilizados na tabela comparativa, onde fica evidente a melhora no desempenho das correções dos erros mais comuns do ambiente de teste.

Tabela 3 - Comparativo entre os tempos de resposta

	Tempo médio de Resposta (min)				
Reclamação	**Lab 1**	**Lab 2**	**Lab 3**	**Lab 4**	**Lab 5**
Lentidão no acesso	16	23	22	26	18
Falha no acesso	21	28	27	31	22
Rede não funciona	13	34	32	38	12
Problema na Impressão	26	42	38	37	24

A tabela 3 apresenta os resultados da implementação das ferramentas no ambiente de teste, onde os laboratórios 1 e 5 foram monitorados com o modelo proposto.

Para posicionar o sistema integrado de gerência (SIG) em relação às soluções existentes os usuários pontuaram o sistema, baseados na mesma tabela comparativa do início deste trabalho e os resultados são apresentados na tabela abaixo.

Tabela 4 – Pontuação do SIG em relação aos critérios iniciais

Critério	*SIG*	*Nagios*	*Cacti*	*Ntop*
Monitorar Aplicativos no Servidor	3	4	2	–
Monitorar Aplicativos no Cliente	3	2	1	–
Monitorar Hardware do Servidor	4	4	4	–
Monitorar Hardware do Cliente	2	2	2	–
Monitorar Equipamentos de Rede	4	4	4	–
Monitorar Enlaces de Dados	3	–	–	5
Monitorar Atividades dos Usuários	3	–	–	4
Facilidade de Instalação	2	2	5	3
Facilidade de Configuração	1	1	5	4
Interface com o Usuário	2	3	4	4
Flexibilidade	2	2	3	1

5 CONCLUSÕES

O monitoramento integrado de um ambiente computacional é necessário nas redes atuais, devido a falta de integração entre as ferramentas existentes e muitas vezes a dificuldade de instalação e configuração impede que os gerentes de redes atinjam o potencial máximo dessas ferramentas. Este quadro impossibilita o máximo retorno sobre o investimento feito na infra-estrutura e nos softwares necessários para o correto funcionamento da empresa.

As soluções existentes não cobrem todo o canal de comunicação, ou seja, cliente – meio – servidor, ficando focadas somente em um ou no máximo dois destes pontos e a integração de ferramentas de diversos fabricantes é extremamente difícil pela falta de padronização das interfaces de software.

Este trabalho propôs um modelo no qual os desenvolvedores podem se basear para a criação de novas ferramentas de gerencia de rede com um viés forte na unificação da visão da rede, para testar o modelo foram integradas algumas ferramentas utilizando os conceitos deste modelo e verificados os resultados da implementação em um ambiente real de computadores.

Os resultados obtidos comprovam a necessidade de um modelo de visão única, mas também expressa as dificuldades de se atingir tal objetivo, pois os fatores envolvidos são muitos e diversos, não garantindo a completa satisfação dos usuários.

Mesmo obtendo uma avaliação positiva do sistema ficam claros os pontos de falha tanto do sistema como no modelo, entre eles temos a dificuldade de padronizar os métodos de configuração das ferramentas já existentes e a necessidade de uma interface gráfica robusta e completa, sem dificultar sua visualização para os usuários.

5.1 Trabalhos Futuros

Dentro das possibilidades que o campo de gerenciamento integrado de redes estão disponíveis, a integração do modelo com elementos de inteligência artificial, utilização de linguagem mais robustas como o Java no desenvolvimento de uma solução cliente servidor, fornecendo maior segurança no acesso e nas trocas de informação, bem como uma interface gráfica aprimorada.

O modelo pode ser aprimorado com a utilização de conceitos de UML em sua modelagem e na abordagem do problema com uma visão orientada a objetos, facilitando a integração entre os dispositivos e a criação de interfaces entre as camadas.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] ISO/IEC FDIS 17799. "**Information techniques — Security techniques — Code of practice for information security management**" (2nd edition), 2005

[2] M. Semola, "Gestão da Segurança da Informação", 2003

[3] Site: http://www.nagios.org, visitado em 28/10/2006

[4] Site: http://www.cacti.net, visitado em 22/11/2006

[5] Site: http://oss.oetiker.ch/rrdtool/doc/rrdtool.en.html, visitado em 22/11/2006

[6] Site: http://www.ntop.org, visitado em 27/11/2006

[7] Site: http://www.ietf.org/rfc/rfc3955.txt, visitado em 28/11/2006

[8] TANENBAUM, Andrew S. "**Redes de Computadores**". Rio de Janeiro, Editora Elsevier, 2003

[9] SOARES, Luiz F.; COLCHER, Sérgio; LEMOS, Guido. "**Redes de Computadores: das LANs, MANs e WANs às redes ATM**". 2. ed. Rio de Janeiro: Editora Campus, 1995.

[10] OPPENHEIMER, Priscilla. "**Projeto de redes *Top-Down***". Rio de Janeiro, Editora Campus, 1999

[11] CHIOZZOTTO, Mauro; SILVA, Luis A. Pinto. **TCP/IP Tecnologia e Implantação**. 1. ed. São Paulo: Editora Érica, 1999.

[12] LAMMLE, Todd. "**CCNA: Cisco Certified Network Associate: Guia de Estudos**". Rio de Janeiro, Editora Campus, 2003

[13] HUNT, Craig; "**TCP/IP Network Administration**", 2nd Edition. Ed O'Reilly, 1997

[14] MURHMMER, Martin W.; ATAKAN, Orcun; BRETZ, Stefan; PUGH, Larry R.; SUZUKI, Kazunari; WOOD, David H., "**TCP/IP Tutorial and Technical Overview**", IBM, 1998, RedBook: GG24-3376-05

[15] MAURO, Douglas, SCHMIDT; "**Essential SNMP**", 2nd Edition. Ed O'Reilly, 2005

[16] RFC 1155

[17] RFC 2578

[18] RFC 3416

[19] RFC 2571

[20] Site http://www.cisco.com/*NetFlow* visitado em 12/01/2007

[21] Site http://www.splintered.net/sw/*flow*-tools/docs/, visitado em 12/01/2007

[22] Site http://www.splintered.net/sw/*flow*-tools/docs/*flow*-tools-examples.html, visitado em 12/01/2007

[23] Site http://oss.oetiker.ch/rrdtool/, visitado em 14/01/2007

[24] JERKINS, Judith L.; WANG, Jonathan L. From network measurement collection to traffic performance modeling: challenges and lessons learned. **J. Braz. Comp. Soc.**, Campinas, v. 5, n. 3, 1999.

[25] Site: http://www.gentoo.org, visitado em 1/12/2006

[26] Site: http://www.kernel.org, visitado em 1/12/2006

[27] Site: http://www.openbsd.org, visitado em 1/12/2006

[28] Site: http://drupal.org, visitado em 2/12/2006

[29] Site: http://www.apache.ogr , visitado em 2/12/2006