UNIVERSIDADE ESTADUAL DO RIO GRANDE DO SUL

UNIDADE EM GUAÍBA

CURSO DE ENGENHARIA EM SISTEMAS DIGITAIS

DIONATA DA SILVA NUNES

SISTEMA DE GERENCIAMENTO DE SWITCHES E ROTEADORES HETEROGÊNEOS

GUAÍBA

2007

DIONATA DA SILVA NUNES

SISTEMA DE GERENCIAMENTO DE SWITCHES E ROTEADORES HETEROGÊNEOS

Sistema de gerenciamento que centraliza o controle de múltiplos switches e roteadores. Gerencia o acesso dos conteúdos da rede dos usuários ou por grupo de usuários. Através do roteador, ele poderá determinar se a sala será isolada da rede ou não, poderá liberar e proteger a comunicação de grupos de usuários distintos (Ex.: sala dos professores), cuja interface será prática para o usuário que o utilizar, sendo necessário somente a configuração na interface gráfica, que será fornecida via navegador, para atuar no dispositivo de rede.

Dr. Rafael Ávila
Orientador

GUAÍBA

2007

DIONATA DA SILVA NUNES

SISTEMA DE GERENCIAMENTO DE SWITCHES E ROTEADORES HETEROGÊNEOS

Sistema de gerenciamento que centraliza o controle de múltiplos switches e roteadores. Gerencia o acesso dos conteúdos da rede dos usuários ou por grupo de usuários. Através do roteador, ele poderá determinar se a sala será isolada da rede ou não, poderá liberar e proteger a comunicação de grupos de usuários distintos (Ex.: sala dos professores), cuja interface será prática para o usuário que o utilizar, sendo necessário somente a configuração na interface gráfica, que será fornecida via navegador, para atuar no dispositivo de rede.

Aprovada em ...../..... /.....

GUAÍBA

2007

Dedico este trabalho a minha futura
esposa Elisa Dias Massena

Agradecimentos

Agradeço a todas as pessoas que de alguma forma contribuíram para que eu pudesse realizar este trabalho, agradeço em especial a minha família, aos meus amigos, minha amada e meu professor orientador Rafael Ávila.

“Jesus Cristo ontem, hoje e sempre...”

RESUMO

O sistema de gerenciamento de switches e roteadores que este trabalho se propõe a realizar, conta com a implementação de uma interface gráfica prática e objetiva para servir de apoio na gerência dos dispositivos de rede, a qual utilizará em sua implantação o protocolo de gerência SNMP para controlar switches e roteadores. Contará também com uma API comum aos dispositivos gerenciados, para que o sistema não se preocupe com modelos dos dispositivos diferentes, deste modo à interface será disponibilizada via navegador por ser uma arquitetura distribuída.

Palavras chaves: Sistema de gerência, switches, roteadores, SNMP.

ABSTRACT

The management system for switches and routers proposed in this work counts on the implementation of a practical graphical interface to support managing such devices, making use of the SNMP protocol to control them. It also counts on a common API so that the system does not depend on different models of routers and switches. In this way, the interface can be acessed through the browser, in a distributed architecture.

Key words: management system, switches, routers, SNMP

SUMÁRIO

1 INTRODUÇÃO

Nos dias atuais presenciamos o crescimento acelerado das redes internas de computadores que se consolidam nas organizações e se tornam indispensáveis para comunicação, troca de informações e compartilhamento de recursos, mas juntamente com o crescimento destas redes surge a necessidade da criação de sistemas para gerenciá-las. E é com este propósito que este trabalho se apresenta, para suprir a necessidade da gerência das redes internas de computadores em busca de melhorar sua administração, controle e segurança dos dispositivos de rede.

Este trabalho reuniu esforços para desenvolver um sistema de fácil gerência de switches e roteadores, cujo usuário que usufruir deste recurso de gerência não terá dificuldades em utilizá-la. O sistema tem o objetivo de proporcionar ao usuário uma maneira prática para executar as configurações da rede, tais configurações são feitas em uma interface gráfica, disponibilizada na rede via navegador.

O sistema não teve pretensão de criar um controle geral da rede como, por exemplo, a implementação de acesso de portas em determinado switches ou controle de fluxo de transmissão de pacotes de dados, mas um controle nos dispositivos que distribuem e direcionam a comunicação. Para que o sistema pudesse ser projetado, foi realizado um levantamento teórico em conceitos de rede e um estudo em tecnologias novas em questão de gerência da rede. Nesta proposta está incluído também o objetivo de fornecer um sistema de gerência distribuída, cuja interface será fornecida via WEB e todas as informações transmitidas por ela. Isto proporcionará um sistema mais dinâmico e de melhor disponibilidade, pois poderá ser acessada de qualquer local.

Sobre as configurações que o usuário poderá fazer na rede através do sistema desenvolvido, podemos dar como exemplo a seguinte situação, um usuário quer que em determinada sala de aula não tenha rede ou que a sala dos professores seja isolada do resto da rede de determinada entidade, o usuário por sua vez simplesmente fará as configurações sugeridas na interface gráfica e o sistema atuará na rede realizando-a. Esta é uma das possibilidades que este sistema de gerência tem o poder de realizar. Outro desafio que este trabalho se propôs a realizar, é no que diz respeito de sua atuação em diversos tipos de switches e roteadores, que para tanto foi implementada uma API comum para que seja possível o sistema atuar sem fazer distinção dos dispositivos de rede.

2 REFERENCIAL TEÓRICO

2.1 MODELO ISO E TCP/IP

A ISO[1] (*Internatinal Standards Organization*) foi a organização que determinou e modelou os protocolos de comunicação empregados em camadas chamado modelo OSI (*Open Systems Interconnection*) mostrado na figura 1. Este modelo está dividido em sete camadas cada uma tratando um nível diferente de abstração, na figura 1 podemos ver que o modelo trata de sistemas que estão abertos à comunicação com outros sistemas. O modelo OSI tem o objetivo de fornecer informações sobre o que cada camada deve fazer. [4] [3] [13].

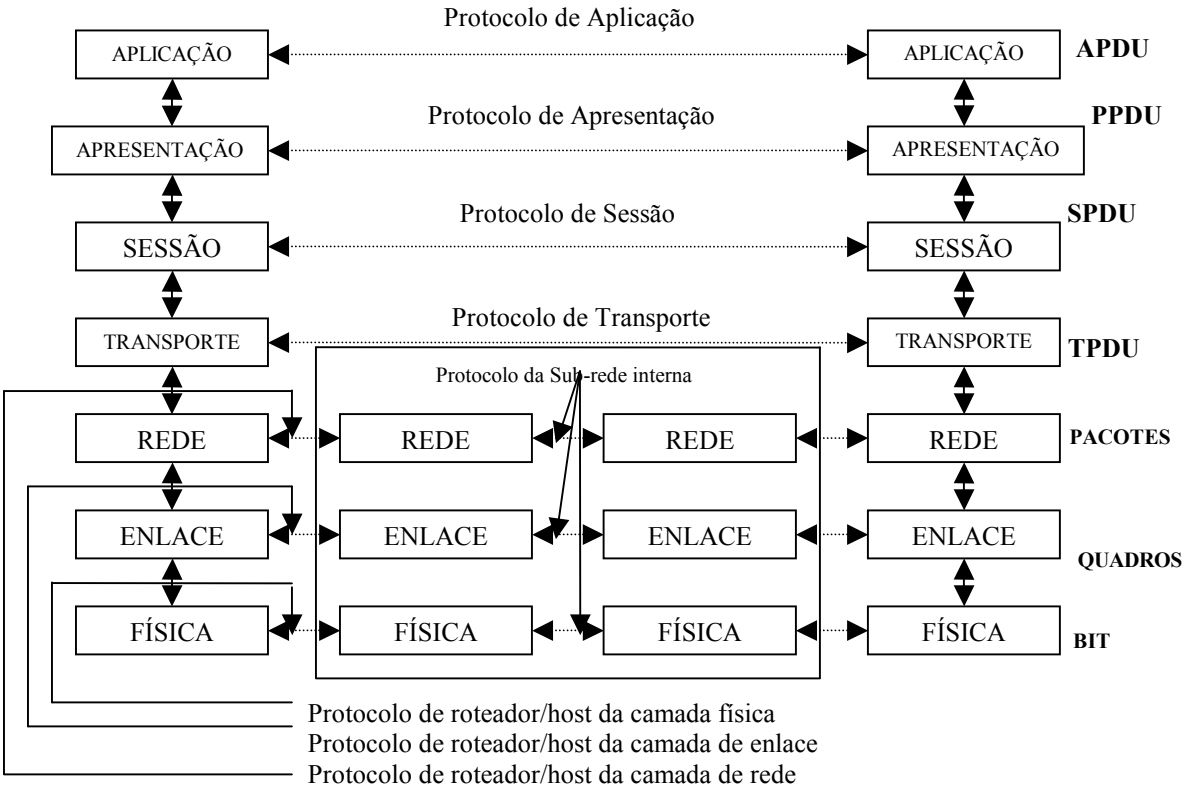

Figura 1 – Modelo de Referência OSI [3]

A camada física do modelo OSI mostrado na figura 1, está encarregada de realizar a transmissão de bit de dados através de um meio, ele deve garantir a transmissão dos bits

[1] Organização internacional fundada em 1946 tem por objetivo a elaboração de padrões internacionais.

sem erros, ou seja, se um bit sai com o valor um do transmissor e deve chegar ao receptor com o mesmo valor.

Já na camada de enlace de dados é inserida a definição de quadro de dados, que significa dividir o fluxo de transmissão de bits da camada física em pequenos pedaços de dados chamados de quadros. Desta forma o papel exercido é de garantir a transmissão dos dados sem erros, caso houver erro na transmissão do quadro, ele será reenviado. Existem mecanismos no nível de enlace que garantem a transmissão de dados correta.

O nível de rede tem como objetivo principal determinar a maneira como os pacotes de dados são roteados desde sua origem até seu destino, para isso existem protocolos incumbidos desta missão. Este nível será detalhado nas sessões seguintes por fazer parte relevante deste trabalho.

As demais camadas não serão detalhadas por não serem relevantes a este trabalho, mas suas definições serão citadas. O nível de transporte tem papel de aceitar dados da camada acima dela e dividi-los em unidades menores, caso necessário repassar a camada de rede e assegurar que todos os fragmentos chegarão corretamente à outra extremidade. A camada de sessão tem como objetivo permitir que os usuários de diferentes máquinas estabeleçam sessões entre elas. A camada de apresentação tem objetivo focado na sintaxe e na semântica das informações transmitidas. Na última camada a de aplicação, nela está contido uma série de protocolos necessários para os usuários como, por exemplo, o protocolo http. [3][4][12][11].

Mas não é só o modelo OSI que é utilizado para projeto de redes, um outro modelo bastante difundido é o modelo de referência TCP/IP mostrado na figura 2. Nesta ilustração

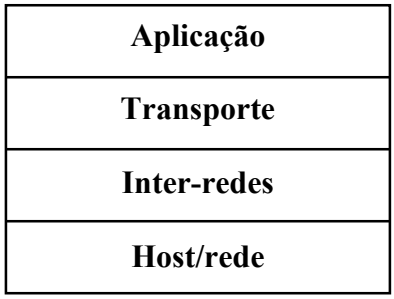

Figura 2 – Modelo de referência TCP/IP

do modelo TCP/IP, notamos que não estão presentes as camadas: física, enlace de dados, sessão e apresentação, pois o modelo não contempla estas camadas ou estão inseridas dentro de outra camada.

Na camada de host/rede do modelo TCP/IP que é mostrado na figura 2, temos pouca descrição sobre esta camada de rede como descreve a citação abaixo.

> Abaixo da camada inter-redes, encontra-se um grande vácuo. O modelo de referência TCP/IP não especifica muito bem o que acontece ali, exceto o fato de que o host tem de se conectar a rede utilizando algum protocolo para que seja possível enviar pacotes IP. Esse protocolo não é definido e varia de host para host e de rede para rede. Os livros e a documentação que tratam do modelo TCP/IP raramente descrevem esse modelo [...] (Tanenbaum, 2003, p. 47).

Na camada acima de host/rede mostrada na figura 2, chamada de inter-redes, exerce o papel de condutor dos pacotes de dados, de forma independente dos hosts até seu destino. Isso pode acarretar até mesmo em entregar os pacotes de dados em ordem diferente, cuja camada superior é quem terá a tarefa de reorganizar os pacotes, enviados e transportados pela camada de inter-redes. Outro detalhe desta camada é que ela foi baseada em uma camada de rede sem conexão para a comunicação dos pacotes de dados.

Na camada de transporte localizada acima da camada de inter-redes mostrada na figura 2, tem seu objetivo focado na comunicação entre *hosts* de origem e destino, para isso acontecer essa camada precisa usufruir os protocolos TCP (*Transmission Control Protocol*) e o protocolo UDP (*User Datagram Protocol*), o TCP tem a função de fragmentar os dados em mensagens discretas e entregar a camada de inter-redes para transportar ao seu destino, e o TCP receptor volta a montar as mensagens. Outras características que podemos ressaltar é o controle de fluxo e conexão confiável. Já o UDP é utilizado para aplicações que não querem controle de fluxo e nem que organizem as mensagens enviadas, focando somente a entrega imediata dos pacotes de dados.

Na camada mais alta do modelo de referência TCP/IP mostrado na figura 2, temos a camada de aplicação contendo todos os protocolos de alto nível como: TELNET (protocolo de terminal virtual), FTP (protocolo de transferência de arquivos), SMTP (correio eletrônico) etc. [3][4][15[11].

2.1.1 Nível de rede

O nível de rede está acima do enlace de dados e tem como objetivo prover a excelência na transferência de pacotes de dados desde sua origem até o seu destino. Cabe ao nível de rede escolher o melhor caminho para um pacote chegar ao seu destino, contudo também está encarregado de evitar roteadores subcarregados e caminhos mais longos. Ele fornece ao nível de transporte transparência de suas operações de chaveamento e roteamento, desta maneira oferece ao nível transporte uma independência nestes aspectos citados e todos os recursos dos níveis inferiores como podemos citar: a camada física, a camada de enlace de dados e as tecnologias de transmissão.

O nível de rede implementa em seu algoritmo diversas funções, como roteamento de pacotes de dados (será detalhado mais à frente), multiplicação, endereçamento, sequenciamento, ligação entre os endereços do nível de rede e do nível de enlace de dados, ligação e estado de conexões de rede, detecção e recuperação de erros, implementa também um controle de congestionamento de pacotes, sequenciamento de pacotes e emissão de unidades de dados dos serviços de rede. Nesta seção serão analisadas as principais funções da camada de rede. [3] [4] [10] [14] [13].

2.1.1.1 Endereçamento

No nível de rede é oferecido o que chamamos de endereçamento SAP (Pontos de Acessos a Serviços), para que as estações possam ter acesso aos serviços de rede, podem ser oferecidos mais SAP por estação, mas a principal característica de que devemos destacar é o endereçamento das SAPs, independente dos demais endereçamentos das camadas do modelo OSI. Na camada de rede podem-se definir dois tipos de endereçamentos, horizontais e o hierárquico.

O endereçamento horizontal pode ser caracterizado por não dar importância na localização das estações dentro de uma rede. No padrão IEEE 802 utiliza-se este tipo de endereçamento nos endereços, globalmente administrados, constituindo pela assinatura dos usuários. Este tipo de endereçamento facilita na hora de reconfigurar as estações, por não se importar com a localização dos mesmos, mas traz desvantagens para os roteamentos por não conter informações de localização das estações.

Outra técnica empregada é o endereçamento hierárquico, neste caso a localização da estação é relevante para o endereçamento especificado, diferente do endereçamento horizontal onde o local era irrelevante. Para o endereçamento hierárquico a estação receberá o seu endereço de acordo com o nível hierarquia que ele se encontra, como por exemplo, o endereçamento SAP que é formado por três parâmetros: rede a que pertence, pela estação dentro da rede e o número da porta associada. [3] [4] [10] [9] [6] [5].

2.1.1.2 Roteamento

A mais importante função da camada de rede com certeza é o de roteamento de pacotes de dados, desde sua trajetória de origem até seu destino, podendo necessitar de vários *loops* até ser cumprido. Para tanto existem vários tipos de algoritmos que implementam políticas de escolha de rotas adequadas, a estrutura de dados e manutenção dos mesmos. Os algoritmos de roteamento podem ser classificados da seguinte maneira: algoritmos não-adaptativos e adaptativos. Os algoritmos não-adaptativos ao estabelecerem suas rotas para os pacotes de dados, não utilizam qualquer tipo de estimativas ou medidas para escolher a melhor rota, tudo é decidido previamente. Este tipo de algoritmo é conhecido como encaminhamentos por rota fixa, caminhos alternativos são tomados somente em caso de falhas. Este método traz desvantagem por não tirar o máximo proveito da utilização do meio, mas ela leva vantagem pela sua simplicidade. O outro grupo de algoritmos de roteamento citado é o adaptativo, com características contrárias aos algoritmos do grupo não-adaptativos. Para os adaptativos, as questões de tráfego, mudanças na topologia são levadas em consideração. Para um melhor caminho pode modificar rotas dos pacotes de dados. Ao realizar mudanças de rotas e escolhas de caminhos mais curtos é necessário realizar atualizações das tabelas de rotas periodicamente. Para tal procedimento existem várias maneiras para sua realização o modo centralizado, modo isolado e modo distribuído.

O primeiro modo de atualização das tabelas de rotas a ser analisado é o modo centralizado. Para este modo os nós da rede enviam informações sobre a carga da rede local para um ponto central analisar estas informações. Para este método se faz necessário um centro de controle de roteamento este que, por sua vez, está responsável pelos cálculos das tabelas de rotas, tomando as decisões dos melhores caminhos. Contudo, este tipo de modo

de atualização sempre toma decisões dos melhores caminhos de rotas, porque o centro de controle de roteamento contém todas as informações da rede. Para a realização das atualizações das tabelas de rotas são realizados cálculos freqüentes, os quais podem variar de acordo com as modificações na topologia da rede. Quanto mais modificações existirem na rede, mais cálculos são necessários para manter atualizadas as rotas, proporcionando um aumento no processamento, fazendo que este método tenha baixo rendimento. Outro problema que podemos analisar para o modo de atualização de tabelas de rotas e o tráfego elevado, que pode se obter para as linhas que levam até o centro de controle centralizado de atualização. Contudo se o tráfego for muito elevado, o desempenho deste modo de atualização fica a desejar. Existem outros problemas a enfrentar por este método, como o fato de o centro de controle estar com problemas físicos, isto resultará em problemas em toda rede. Este modo só é vantajoso se a rede for estável em mudanças na topologia, deste modo podemos ter um desempenho razoavelmente satisfatório.

O segundo modo de atualização de tabelas de rotas a ser analisado é o modo isolado, para este método é analisada a fila de mensagem que aproveita estas informações para atualização das tabelas de rotas. Este modo de atualização tem um melhor desempenho combinado com as rotas fixas, ou seja, ao analisar as mensagens que contém as informações é levado em consideração o tamanho destas mensagens, e a partir daí analisado da seguinte maneira: no que diz respeito ao tamanho da mensagem, elas são encaminhadas para filas de menor tamanho e as mensagens que passarem deste tamanho são enviadas as rotas fixas. Outro algoritmo de roteamento isolado que pode ser citado tem a seguinte descrição: quando um pacote chegar a um determinado receptor, ele é encaminhado para todos os nós e aos enlaces de saída, exceto ao que ele chegou. Existem vários algoritmos de roteamento isolado, mas não serão analisados neste trabalho por não se fazer necessário.

O último modo de atualização ser analisado é o distribuído, no qual as informações de carga de cada nó são enviadas para os outros nós e roteadores, para que possam ser calculadas as novas tabelas de rotas. Visando uma melhor compreensão, será analisado um algoritmo de rota distribuída. Quando um nó “quer saber” o caminho mais curto para determinado destino, ele envia um pacote de dados contendo um campo do destinatário e um contendo um caminho, inicialmente seu endereço, por meio de difusão e todos os enlaces de saída. Os nós que receberam estes pacotes de requisição analisam o seu

conteúdo e se não for o destinatário adicionará seu endereço e reenviará o pacote a todos seus enlaces de saída, ao destinatário chegará diversos pacotes, o primeiro pacote a chegar será o de menor tempo. E desta forma considerada o de caminho mais curto, os demais serão excluídos, um pacote resposta será enviado com esta rota. [3] [4] [6] [5] [10] [9] [14].

2.1.1.3 Controle de congestionamento

O congestionamento na sub-rede pode ser constatado quando há excessivos pacotes presentes, causando a diminuição no desempenho da rede como um todo. A queda de desempenho pode ser vista na figura 3.

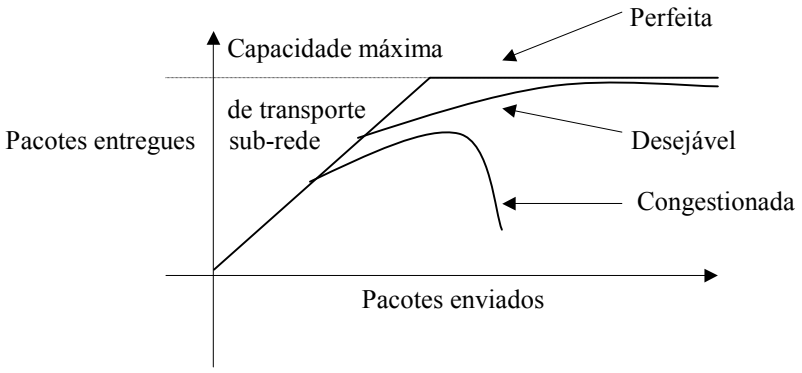

Figura 3 – Queda de desempenho devido ao tráfego na rede [3]

Os motivos que levam a um congestionamento são inúmeros. Podemos citar, por exemplo, se o tráfego de entrada de dados exceder as capacidades das linhas de saída, se os nós de uma determinada rede são muito lentos etc. Existem métodos de controle de congestionamentos para lidar com esta situação. Podemos salientar os métodos por controle de tráfego no enlace, pré-alocação de *buffers*, descarte de pacotes entre outros.

O método de controle de tráfego no enlace, o primeiro a ser analisado, funciona da seguinte maneira: todos os nós de uma rede têm a função de monitorar cada linha de saída. Desta forma, é determinado um limite de tráfego nas suas saídas. Quando exceder este limite, sua linha de saída entra em estado de alerta. Quando um nó qualquer envia um pacote de dados ao outro nó que está em estado de alerta, o pacote enviado é despachado por uma linha de alerta e um pacote de alerta é enviado ao transmissor do pacote recebido. O transmissor do pacote receberá o pacote de alerta fazendo com que ele diminua o envio ao nó que está em estado de alerta. Contudo, é possível que outros pacotes tenham sido

enviados para o nó congestionado, assim fazendo que seja gerado mais pacote de alertas. Mas quando um nó recebe um pacote de alerta de tráfego congestionado, um tempo determinado é estabelecido ao transmissor para não considerar pacotes de alertas. Após passar este tempo, outro tempo é destinado para verificar se não chegará nenhum outro pacote de alerta. Passando estes dois tempos o nó transmissor volta ao envia de pacotes normal.

Outro método de controle de tráfego a ser analisado é de pré-locação de *buffers*. Este método é usado quando forem utilizadas conexões de circuito virtuais. Para este método, quando acontece um congestionamento, ele pode ser resolvido por pré-alocação de *buffers*, ou seja, se um pacote chegar ao destino e estiver congestionado ele armazena em *buffers*, e se todos estiverem carregados à conexão não é concretizada. Este método enfrenta problemas no que diz respeito à alocação dos *buffers* que não podem compartilhar, mesmo quando não estão sendo utilizados. Uma solução para isso é a implementação de temporizadores nos buffers.

O último método a ser analisado é o controle de congestionamento de descarte de pacotes. Este método é bastante simples se um pacote não chegar a um nó, ele não fornece espaço para armazená-lo, e então ele é jogado fora. [3] [4] [6] [9] [10].

2.2 DISPOSITIVOS DE REDE

2.2.1 Bridge

As *bridges* podem ser definidas como equipamentos que tem objetivo de fracionar uma determinada rede local (LAN) em várias sub-redes, proporcionando uma diminuição no tráfego da rede, pois com a divisão somente as estações pertencentes à mesma sub-rede recebem quadros *broadcast.* Desta maneira, as *bridges* atuam na rede como filtros de pacotes de dados.

Inserindo as *bridges* no contexto da camada de rede, ela pode atuar tanto na camada física, quanto na camada de enlace de dados. Na camada física, as *bridges* exercem o papel de regenerador de sinal, ou seja, quando recebe um sinal na entrada ele é capaz de regenerar o sinal na saída. A *bridge* que atua na camada de enlace de dados tem o papel de verificar os endereços físicos dos quadros (MAC), tanto o de destino quanto o de origem.

A filtragem exercida das *bridges* sobre a rede se dá na seguinte forma: ela verifica os endereços de origem e destino de um determinado quadro. Baseado nesta informação ela toma decisões de encaminhamento dos quadros. Mas para isso ser possível, a *bridge* contém uma tabela com os endereços físicos relacionadas com suas portas.

As *bridges* mais utilizadas são as *bridges* transparentes, cujas estações conectadas a ela não percebem sua presença na LAN, ou seja, se uma *bridge* é colocada ou retirada da rede não é necessário mudar as configurações das estações, por elas não "sentirem" sua presença. Todos os parâmetros das *bridges* transparentes são determinados pela IEEE 802.1.d, que dentre as tarefas realizadas pode se ressaltar a filtragem, encaminhamento e bloqueio de quadros.

Anteriormente as implementações das tabelas das *bridges* eram estáticas, onde as atualizações eram realizadas manualmente pelo administrador, através da operação de *setup* da *bridge.* Mas devido este trabalho de atualização das tabelas não ser prático, foram elaboradas as tabelas dinâmicas para *bridge*, que atualizam as tabelas automaticamente. Para ser possível estabelecer as tabelas dinâmicas, é necessário atualizar os endereços dos quadros de origem e de destino, é utilizado pela *bridge* para adicionar entradas nas tabelas e também para atualização da mesma da *bridge.*

Os problemas enfrentados pela *bridge* transparente estão relacionados com *loops* entre *bridge* redundantes na rede, mas para que usar mais de uma *bridge* em uma rede? É utilizado para tornar o sistema mais confiável, pois se uma *bridge* der problema, outra *bridge* assume. Para resolver os problemas de *loop* é necessário agregar à *bridge* o protocolo *spanning tree,* que pode ser definido como um grafo sem *loops*, ela está especificada pela IEEE para desenvolver topologias imunes aos *loops*. Outro método para solucionar os *loops* é utilizar SRB (*source routing bridges)*.

O SRB soluciona o problema das *loops* da seguinte maneira: dentre as tarefas realizadas como filtragem, encaminhamento e bloqueio de quadros são para a estação de origem, tornado-a responsável por elas, ou seja, o transmissor define não somente o endereço de destino, mas também qual *bridge* deve ir, através da especificação do endereço da *bridge* adicionada no quadro transmitido pela estação. O método SRB e definido pela IEEE para redes *Token Ring*. [10].

2.2.2 Switches

As redes podem ter vários tipos de topologias como estrela, anel, em barra etc. A topologia empregada pela rede pode exercer um papel fundamental na escolha no método de acesso aos meios físicos e compartilhados. Para exemplificar o que foi dito anteriormente, podemos citar a topologia em estrela, bastante usada atualmente, que determina um componente central para atuar como comutador, usualmente utilizado-se o Switch para essa tarefa. Desta forma, pode-se obter uma melhor gerência e manutenção das redes atuais, mas as redes de computadores que utilizam a topologia em estrela, antigamente enfrentavam problemas de segurança e baixa confiabilidade por ter um elemento central no qual seus problemas físicos e lógicos poderiam ocasionar a parada total da rede. Nos dias atuais com o avanço na construção destes equipamentos, se reflete um baixo índice de problemas físicos e lógicos resultando em equipamentos estáveis e alta confiabilidade na rede que eles são inseridos.

Os switches surgiram para aprimorar o desempenho e a funcionalidade dos equipamentos concentradores. Eles partem do princípio dos HUBs, mas agregam uma melhor tecnologia. Os switches proporcionam às estações não somente um meio de compartilhamento, mas também um meio de comunicação simultânea entre elas. Assim os switches têm a capacidade de atuar como um roteador ou como funções de pontes. A principal diferença dos switches sobre os HUBs são seus barramentos internos comutáveis que permitem chavear conexões, proporcionando, se assim desejar, a comunicação entre dois pontos da rede dedicados exclusivamente para elas, aumentando seu desempenho, e suas multiportas, são capazes de alocar uma estação em cada porta.

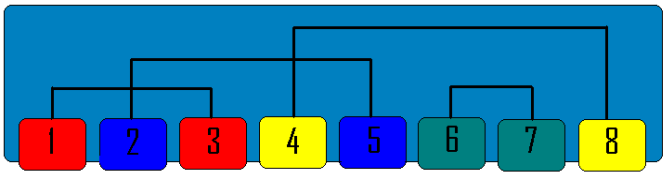

Figura 4 – Ligações interna de um Switch.

As estações atuam como entidades independentes. Deste modo as estações não ficam disputando meios de transmissão nas redes ethernet, que utilizam o protocolo CSMA/CD. Pode-se dizer que a transmissão é realizada sem colisões. Na figura 4 podemos analisar como funcionam as ligações internas de um switch. Na figura, a estação que está

na porta 1 envia dados para a estação da porta 3, a estação da porta 2 envia dados para a estação da porta 5, a estação da porta 4 envia dados para estação da porta 8 e estação da porta 6 envia dados para estação da porta 7. Podemos verificar que para cada conexão existente, como podemos ver na figura 4, a rede está todo tempo disponível, tudo isso graças ao chaveamento inteligente que é implementado pelo switch.

Os switches podem ser divididos em dois tipos básicos: os switches de camada 2, ou seja, dispositivos que operam na camada física e de enlace, e switches de camada 3, que atuam na camada de rede. [10].

2.2.3 Roteadores

Os roteadores podem ser definidos como sendo equipamentos que interligam redes locais e redes remotas em tempo integral. Desta forma, os roteadores proporcionam a comunicação entre estações de diferentes redes locais. Para tanto, ele utiliza um protocolo de comunicação, usualmente o protocolo TCP/IP. A figura 5 mostra a comunicação de duas redes locais utilizando roteadores. Os roteadores podem agregar diversas funcionalidades, como ser um elemento de decisão no que se trata na hora de escolher o melhor caminho de um determinado pacote para seu destino, permitir o tráfego de diferentes protocolos, atuar como filtro de pacotes e gerador de estatísticas.

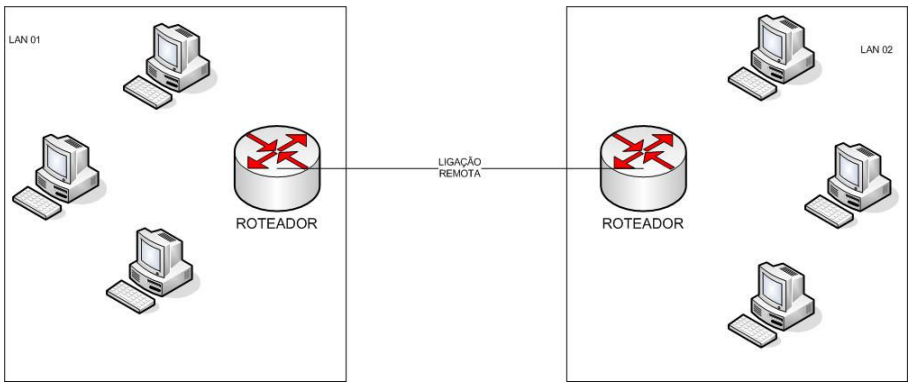

Figura 5 – Comunicação entre roteadores

Para haver comunicação entre redes locais é preciso definir as portas usadas para comunicação dos roteadores que são as portas UTP, FDDI ou AUI, utilizando para tanto, os protocolos como RS 232, RS 449 etc.

Os roteadores atuam nas três primeiras camadas do modelo de referência OSI, a camada física, de enlace de dados e de rede, todos já apresentados neste trabalho, implementando algoritmos de roteamento e uma série de regras como rotas estatísticas, rotas dinâmicas etc. No encaminhamento de pacotes, o roteador pode ser muito poderoso no tratamento de decisões de rotas. Com relação aos pacotes de dados, ele pode compreender informações complexas de endereços, contudo decidir o melhor caminho para seu destino, podendo complementar com mais informações no envio de determinado pacote de dados. As informações adicionais podem ser para um motivo específico ou somente informativo. Os roteadores, nas suas decisões de rotas, podem optar por esquemas de composições de pacotes e de acesso aos meios físicos completamente diferentes, mas para isso ele necessita ler as informações contidas nos pacotes, utiliza algoritmos de endereçamento para determinar a rota adequada, o encapsula novamente e o retransmite. [10].

2.3 VLANS

As redes de computadores nos dias atuais vêm crescendo e se tornando indispensáveis para a difusão da informação. Nestes mesmos parâmetros as redes locais, chamadas LANs *(Local Area Networks),* têm papel fundamental em empresas, universidades etc., para comunicação entre as estações, compartilhamento de recursos, organização e controle.

Dentro deste contexto surgem as redes locais virtuais, chamadas de VLANs para aprimorar e solucionar dificuldades enfrentadas pelas LANs. As VLANs podem ter várias definições, uma delas é considerar as VLANs como uma configuração lógica da rede, mas não através de fios, mas através de software. Uma das inúmeras vantagens de se utilizar as VLAN é poder configurar uma rede de modo a não mexer na estrutura física da rede, ou seja, acrescentar cabos à rede, mudança física da infra-estrutura da rede. Outra vantagem que redes locais virtuais oferecem é conter o tráfego broadcast, pois com as VLANS e possível criar domínios broadcast. Desta forma, somente as estações pertencentes a este domínio recebem quadros de broadcast, podendo não ser necessariamente do mesmo segmento de rede o que proporciona uma vantajosa independência topológica da rede.

Portanto, as vantagens que as redes locais virtuais agregam, e sua importância de implementação nos dias atuais, provêm da visão de segurança da informação e o aumento do desempenho proporcionado por ela. [10] [3]

2.3.1 Tipos de implementações de VLANs

As VLANs podem ser implementadas de diversas formas como número portas, endereço MAC, endereço IP, endereço IP multicast, e combinações destas. Isso depende em que o contexto está inserido a implementação das LANs. Umas das possíveis implementações é através de número das portas de um comutador relacionado a determinadas VLANs. Por exemplo, em um *switch* estão conectadas nas portas 1, 3, 4 e 8 determinadas estações, pode-se então definir que estas portas pertencerão à VLAN00 e enquanto as portas 2, 5, 6 e 7 estão ligadas outras estações à VLAN01. Na figura 6 é mostrada esta configuração.

Portas	1	2	3	4	5	6	7	8
VLAN	**00**	01	**00**	**00**	01	01	01	**00**

Figura 6 – Relação das portas físicas suas respectivas VLANs

Se for conectado um comutador em uma das portas mostrada na figura 6, ele respeitará as configurações relacionadas à VLAN que está definida para tal porta.

Em outro método, muito utilizado em implementações de VLANs é através de endereços MAC (*media access control),* funcionando da mesma maneira que as VLANs baseadas em portas, são definidos quais endereços MAC de uma rede pertencerão uma determinada VLAN. Este método traz uma vantagem sobre as VLANs implementadas por portas, se a estação for retirada do comutador não será preciso reconfigurar a VLAN novamente, pois o endereço físico da estação seguirá com ela.

Já o método por endereços IP, segue o mesmo padrão dos outros métodos de implementação já citados, a única diferença é que o agrupamento é feito pelo número IP de determinadas estações a uma VLAN. Por exemplo, algumas determinadas estações que contêm os seguintes números IPs 192.168.32.5, 192.168.32.9 e 192.168.32.38 podem ser agrupadas em um domínio broadcast da VLAN03. [3][10][16][15].

2.3.2 Tipo de configurações de VLANs

Para estabelecer uma rede local virtual em estações de trabalho, primeiramente é necessário definir qual tipo de configuração será utilizada. Para tanto são classificadas as configurações de VLANs em três tipos: manual, automático e semi-automático.

Na configuração manual as VLANs são configuradas através de um software de criação de VLAN, cujo administrador criará manualmente as VLANs definirá os parâmetros necessários para seu funcionamento como número de portas, endereços IPs etc. Também caberá a ele ter que reconfigurar, quando houver qualquer alteração na rede.

Na configuração automática as estações são conectadas e desconectadas automaticamente de uma VLAN, isto pode ser alcançado devido a critérios pré-estabelecidos pelo administrador da rede.

Já na configuração semi-automática agrega definições tanto da configuração automática, quanto da configuração manual. Neste contexto é definido geralmente da seguinte forma: o ponto inicial, como criar as VLANs e definir parâmetros é feito através de configuração manual, mas as migrações das estações é feito automático. [3][10][16][15].

2.3.3 Tipo de identificação de VLANs

Quando a estrutura da rede está montada através de switch interconectados, as VLANs podem formar grupos de estações de distintos segmentos de rede, por exemplo, a estação01 está conectado no switch01 e a estação02 está conectado ao switch02, mas tanto a estação01 quanto a estação02 pertencem a VLAN05, portanto o switch01 deve conter informações de status da estação do switch02 e vice-versa. Para esta troca de informações foram desenvolvidos três métodos: manutenção da tabela, identificação de quadros *(frame tagging)* e multiplicação por divisão de tempo (TDM).

O método por manutenção da tabela procede da seguinte forma, quando um membro de uma determinada VLAN envia um quadro de broadcast para outro membro da VLAN o switch cria uma entrada em um tabela e grava o endereço da estação. As tabelas criadas para os switches são constantemente trocadas entre elas para manterem atualizadas.

No método *frame tagging* funciona da seguinte maneira: quando a estação de um determinado grupo de uma VLAN envia quadro de broadcast para determinar o destino deste quadro, é adicionado ao MACframe um cabeçalho extra assim definindo a VLAN

destino. A informação adicionada chamada de *frame tagging* é utilizada pelos switch para determinar qual VLAN deve receber os quadros de broadcast.

O último método a ser esclarecido é a multiplicação por divisão do tempo (TDM), para este método uma conexão é estabelecida entre switches, denominado *trunk,* que é dividida em canais e compartilhada no tempo. *Trunk* é dividido de acordo com o número de VLAN existentes na rede, ou seja, se existem 8 VLANs então o *trunk* será dividido em 8, onde irá tráfegar a comunicação das VLANs, neste método é o *switch* receptor que determina a VLAN de destino. [3][10][16][15].

> Em 1996, o subcomitê IEEE 802.1 especificou um padrão, determinado 802.1Q, que define o formato do *frame tagging*. O padrão também define o formato a ser utilizado nos *backbones multswitchs* e regulamento o uso de equipamentos de fabricantes diferente. O padrão IEEE 802.1Q promoveu a padronização de outras questões relacionadas as VLANs.Muitos fabricantes aceitaram esse padrão. [...] (FOROUZAN, 2006,p.364)

2.4 PROTOCOLOS DE GERENCIAMENTO

2.4.1 SNMP

O Protocolo SNMP (*Simple Network Management Protocol*), tem por finalidade o gerenciamento da interligação em redes, muito utilizado em software que empregam serviços de gerenciamento como controlar roteadores, detectar problemas e localizar computadores que estejam violando os padrões do protocolo. Atualmente é o protocolo padrão de gerenciamento de redes TCP/IP, mas já existe uma segunda versão, o SNMPv2, com mais recursos e segurança. [1][2].

2.4.1.1 MIB

Para definirmos uma MIB *(Managent Information Base)* antes temos que definir o conceito de objeto gerenciado. Objeto gerenciado pode ser definido como a visão abstrata de um recurso real do sistema. Desta forma todos os recursos da rede que devem ser gerenciados são modelados, e as estruturas dos dados resultantes são os objetos gerenciados. Em respeito às permissões dos objetos gerenciados eles podem ser lidos ou modificados, cujo cada leitura representará o estado real do recurso e cada alteração também será refletida no próprio recurso.

Portanto, MIB é o conjunto dos objetos gerenciados, que procura abranger todas as informações necessárias para gerência da rede. A primeira versão da MIB (MIB I) foi apresentada na RFC (*Request For Comment*) 1066, cuja versão definiu a base de informações necessárias para monitorar e controlar redes baseadas na pilha de protocolos TCP/IP. Posteriormente foi lançada uma nova versão da MIB a MIB II, pela RFC 1213, para uso baseado na pilha de protocolos TCP/IP. [1] [2] [17].

2.4.1.1.1 Estrutura da MIB

A Figura 7 mostra a árvore Hierárquica definida pela ISO, representando a estrutura lógica da MIB, mostra o identificador e o nome de cada objeto. O nó raiz da árvore hierárquica (fig.7) não possui rótulo, mas possui pelo menos três sub-níveis, sendo eles: o nó 2 administrado pela ITU; o nó 3 que é administrado pela ISO e pela ITU; abaixo do nó 1 que é administrado pela ISO, fica o nó org (3) que pode ser utilizado por outras instituições, em seguida fica o *dod* (6) que pertence ao departamento de defesa dos EUA. O departamento de defesa do EUA alocou um sub-nó para a comunidade da internet, que é administrada pela IAB (*International Activities Board*) e abaixo destes nós temos, entre outros, os nós: diretório, gerenciamento, experimental e privativo, podemos ver esta estrutura na figura 7.

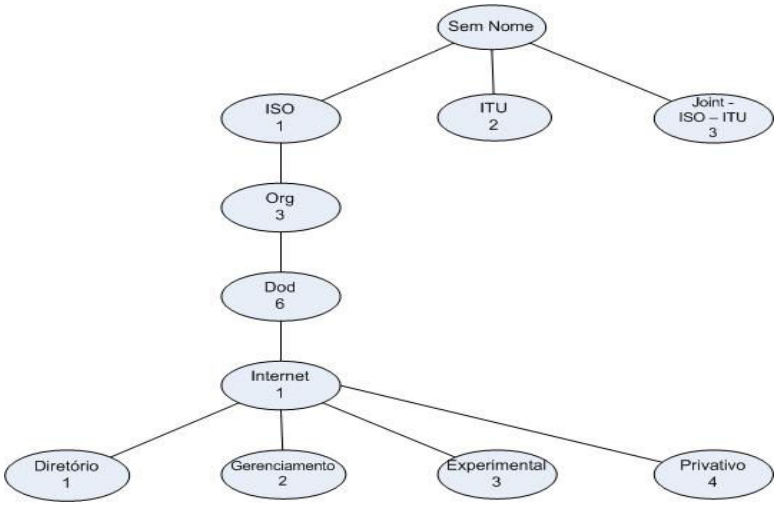

Figura 7 – Árvore Hierárquica definida pela ISO [1].

O MIB classifica as informações pra TCP/IP de gerenciamento em oito categorias mostrada na tabela 1. Estas categorias são muito importantes, porque os identificadores usados para especificar itens incluem um código por categoria. [1][2][8].

Tabela 1 – Categorias das informações de gerenciamento [1].

Categoria da MIB	Inclui Informações sobre
Sistema	**O sistema operacional do roteador ou host.**
Interfaces	**Interfaces de rede específicas**
Conv. End.	**Conversão de endereço (ex. mapeamento ARP).**
IP	**Software do Protocolo Internet**
ICMP	**Software do Protocolo de controle de mensagens da Internet.**
TCP	**Software do Protocolo de controle de transmissão.**
UDP	**Software do Protocolo de Datagrama do Usuário.**
EGP	**Software do Protocolo de Gateway Externo.**
Transmissão	**Meio de transmissão**
SNMP	**Protocolo SNMP**

2.4.1.2 Características do SNMP

O SNMP é um protocolo de gerenciamento de redes que tem por premissas definir a forma e o significado das mensagens trocadas entre o programa cliente de gerenciamento de rede chamado pelo administrador. E o programa do servidor de gerenciamento de rede que está sendo executado em um *host* ou roteador. Além disso, pode definir a forma de representação de nomes e valores de mensagens trocadas. O SNMP é definido no nível de aplicação do modelo OSI. Ele coleta informações dos agentes espalhados em uma rede baseada na pilha de protocolos TCP/IP, cujos dados são obtidos através de requisições de um gerente a um ou mais agentes utilizando os serviços de protocolos de transporte UDP (*User Datagram Protocol*) para enviar e receber suas mensagens através da rede, onde as variáveis que podem ser utilizadas, serão requisitadas pelas MIBs que podem ou não fazer parte da MIB II, privada ou experimental.

O protocolo de gerenciamento SNMP apresenta uma alternativa para gerência da rede, ou seja, ela não define um grande conjunto de comandos, o SNMP lança todas as

operações em um paradigma de busca e armazenamento (*fetch-store paradigm*). Os comandos são baseados em mecanismos de busca/alteração. Neste mecanismo de busca/alteração estão disponíveis as operações de alteração de um valor de um objeto, de obtenção dos valores de um objeto e suas variações.

A utilização de paradigma de "busca e armazenamento" traz uma série de vantagens como maior estabilidade, simplicidade e flexibilidade. Como conseqüência reduz o tráfego de mensagens, realiza o gerenciamento através da rede e permite a introdução de novas características. [1] [2] [7].

2.4.1.3 Operações do protocolo SNMP

Vamos agora descrever as operações existentes do protocolo SNMP, como mostrado na tabela 2.

Tabela 2 - Conjunto de possíveis operações do SNMP [1].

Comando	Significado
get-request	**Busca o valor de uma variável especifica**
get-next-request	**Busca o valor sem saber o nome exato**
4get-response	**Responde a uma operação de busca**
Set-request	**Armazena um valor em uma variável especifica**
Trap	**Resposta acionada por um evento**

Nas operações de busca de variáveis descritas na tabela 2 como *set-request* pode ter papel de alterar do valor da variável, cujo gerente tenha solicita que o agente faça uma alteração no valor da variável. Na operação *get-request* é utilizado para ler o valor de uma determinada variável, cujo gerente tenha solicitado que o agente obtenha o valor da variável. Na variação da operação *get-request* temos a operação *get-next-request* que obtém valores e nomes de variáveis de uma tabela de tamanho desconhecido, também é usado para ler o valor da próxima variável, cujo gerente fornece o nome de uma variável e o cliente obtém o valor e o nome da próxima variável.

> A operação get-first acomoda solicitações get-next referentes a uma ordenação léxica arbitrária de itens da MIB. Parta tanto, encontra, na ordem léxica, a próxima variável a que a operação get se aplica. Depois, aplica solicitação get a esse item [...] (COMER, 1999, p.452).

A operação *get-first* é a variação da operação *get*, que quando aplicada a uma variável simples, *get-first* é igual à *get*. Quando aplicada a um grupo, no entanto, *get-first* é igual a *get-next*. Na operação *trap* tem por finalidade comunicar um evento, cujo agente comunica ao gerente o acontecimento de um evento, previamente determinado. Existem sete tipos básicos de *trap* determinados conforme mostrado na tabela 3. [1][2][7][8].

Tabela 3 – Tipos básicos de Trap [1].

Comandos	Significado
CondStart	**A entidade que a envia foi reinicializada, indicando que a configuração do agente ou a implementação pode ter sido alterada;**
WarmStart	**A entidade que a envia foi reinicializada, porém a configuração do agente e a implementação não foram alteradas;**
LinkDown	**O enlace de comunicação foi interrompido;**
LinkUp	**O enlace de comunicação foi estabelecido;**
authenticationFailure	**O agente recebeu uma mensagem SNMP do gerente que não foi autenticada;**
EgpNeighborLoss	**Um par EGP parou;**
EnterpriseSpecific	**Indica a ocorrência de uma operação TRAP não básica.**

2.4.1.4 Mensagens do protocolo SNMP

Nas mensagens SNMP existe um diferencial da maioria dos protocolos TCP/IP, ele não possui campos fixos. Portanto a forma que ele usa para definir a notação formal é o padrão de codificação ASN.1 *(Abstract Syntax Notation)*. As mensagens SNMP consistem em três partes principais uma versão do protocolo, um identificador de comunidade do SNMP, e uma área de dados, cuja esta área de dados está dividida em unidade de dados do

protocolo (PDUs). A figura 8 mostra os campos e seus respectivos componentes do protocolo SNMP.

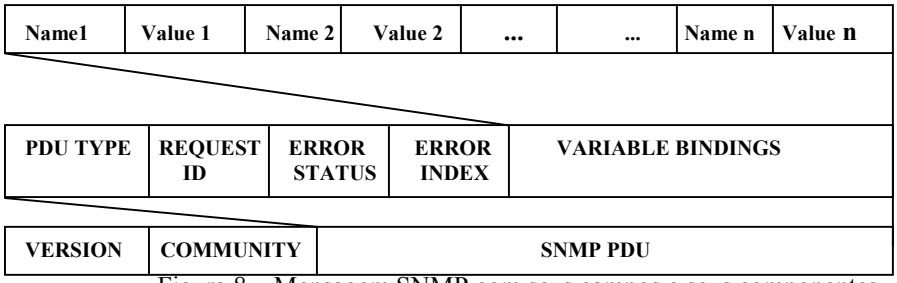

Figura 8 – Mensagem SNMP com seus campos e seus componentes.

No campo *version* está contida a versão SNMP, onde tanto o gerente quanto o agente deve utilizar a mesma versão, pois as mensagens que contêm versões diferentes são descartadas. Já no campo *community* identifica a comunidade, ou seja, é utilizado para permitir acesso do gerente as MIBs. No campo SNMP PDU(*Protocol Data Units*) contém a parte dos dados.

Após o servidor ter feito uma análise nos campos de uma mensagem SNMP, ele deve utilizar as estruturas internas disponíveis para interpretar a mensagem e enviar a resposta da operação ao cliente que o solicitou. [1] [2] [7].

2.4.2 ICMP

O ICMP *(Internet Control Message Protocol)* é uma ferramenta que fornece relatórios de erros, tratando diversos tipos de condições de erros, parte integrante do protocolo IP. O ICMP permite a que roteadores enviem mensagens de erro ou de controle a outros roteadores ou a outros hosts. O ICMP surgiu para servir de ferramenta de comunicação dos roteadores no que diz respeito a erros e informações sobre ocorrências inesperadas. Ele foi implementado para ser um relator de erros. Contudo, para corrigir os erros apontados pelo ICMP, o transmissor deve tomar outros procedimentos para sanar o problema. [1].

2.4.2.1 Mensagens ICMP

As mensagens ICMP são encapsuladas na parte de dados de um datagrama IP, que por sua vez é encapsulado na parte dos dados dos quadros da camada física, como podemos ver na figura 9.

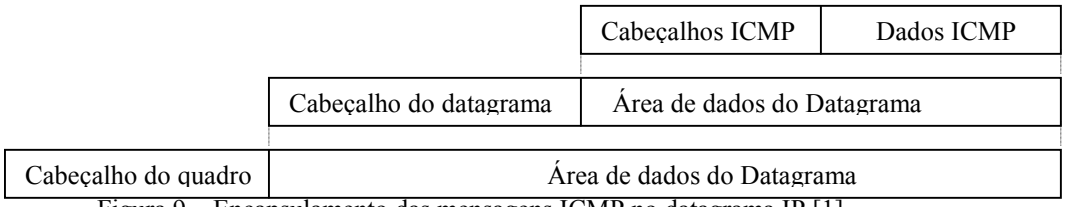

Figura 9 – Encapsulamento das mensagens ICMP no datagrama IP [1].

Não existe nem tipo de prioridade para ICMP, é tratado de forma igualitária às demais informações, contidas no protocolo IP. Ou seja, se tivéssemos em uma situação em que a rede se encontra congestionada, o ICMP só vai ajudar a aumentar o congestionamento. Não obtendo sucesso na busca de relatar erro que estejam acontecendo.

O ICMP dá autonomia no formato de mensagem, mas existem três campos que são igual para todas as mensagens: o campo tipo, código e soma de verificação. No campo tipo contém o tipo de mensagem a ser informada com bits para representação, que se pode definir como mostra a tabela 4.

Tabela 4 – Campo tipo [1].

Campo tipo	Tipo de mensagem ICMP
0	Resposta ao Eco (Echo Reply)
3	Destino Não-acessivel
4	Dissipação de Origem
5	Redirecionamento (mudar a rota)
8	Solicitação de Eco (Eco request)
11	Tempo Excedido para um Datagrama
12	Problema de parâmetro num Datagrama
13	Solicitação de indicação de hora
14	Resposta de indicação de hora
15	Solicitação de informação (obsoleto)
16	Resposta de informação (obsoleto)
17	Solicitação de mascara de endereço
18	Resposta de Mascara de endereço

No campo código, que é representado com oito bits, tem a função de servir como informação adicional às mensagens ICMP. Podemos analisar os tipos de informações que podem ser incluídas as mensagens de hosts inacessíveis na tabela 5 baixo.

Tabela 5 – Possíveis informações adicionas do campo código para hosts inacessíveis [2].

Valor de Código	Significado
0	Rede inacessível
1	Host inacessível
2	Protocolo inacessível
3	Porta inacessível
4	Fragmentação necessária e DF
5	Rota de origem em falha
6	Rede de destino desconhecida
7	Host de destino desconhecido
8	Host de origem isolado
9	Comunicação com redes destinatária administrativamente proibida
10	Comunicação com host destinatário administrativamente proibida
11	Rede inacessível para tipo de serviço
12	Host inacessível para tipo de serviço

E o campo soma de verificação, que é representado com 16 bits para verificação de erros, utiliza o mesmo algoritmo usado no protocolo IP.

O mais usual serviço oferecido do ICMP são solicitações de eco, cujos usuários podem fazer suas requisições através do comando *ping*, podemos observar seu formato na figura 9 logo abaixo. [1][2][7][8].

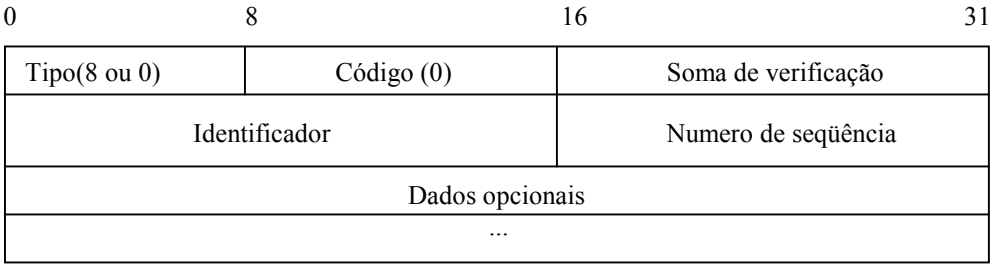

Figura 9 – Formato de mensagem ICMP para solicitação de ECO [2].

2.4.2.2 Mensagem *souce quench*

As mensagens ICMP podem exercer um papel importante no que diz respeito a congestionamento. As mensagens ICMP podem ser de alertas para os transmissores para diminuir a carga de envio de mensagem, já visto neste trabalho na seção da camada de rede. Nas mensagens de alertas chamadas de *Source quench* de congestionamento e adicionado um campo extra para o prefixo do datagrama na figura 10 mostra o formato desta mensagem. [1][2].

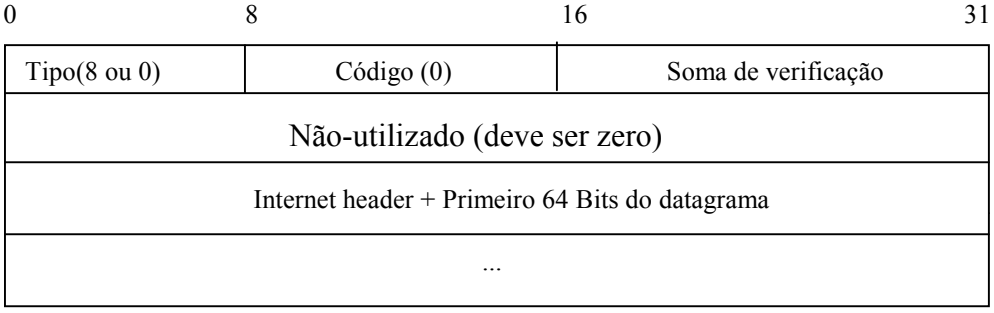

Figura 10 – Formato de mensagem ICMP *Source quench* [2].

2.4.2.3 Mudanças de rotas por roteador

Quando informações de rotas dos roteadores sofrem alterações, devidas a mudanças da topologia da rede ao por qualquer outro motivo que seja, a fim de evitar duplicação de informação nos roteadores nos arquivos de configuração dos *hosts*, necessitam de um mínimo de informação dos roteadores para efetuar a comunicação essa informações fornecidas através das mensagens ICMP. Para *hosts* que tem uma rota que não seja a melhor prevista pelo roteador, é enviada uma mensagem ICMP pedindo sua rota. Traz vantagens devido poder redirecionar um *host* apenas com um endereço do roteador central. O redirecionamento com ICMP tem suas limitações claras no que diz respeito a problemas de difusão de rotas por se limitar apenas a um roteador e um *host*. A figura 11 ilustra o formato de mensagens de redirecionamento ICMP. [1][2][7][8].

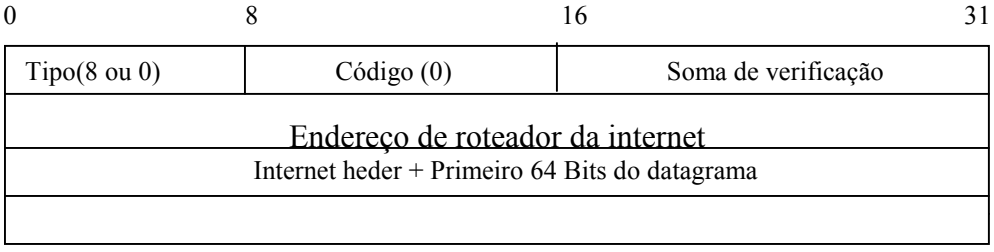

Figura 11– Formato de mensagem ICMP Source quench [2].

3 MODELAGEM DO SISTEMA DE GERENCIAMENTO

O projeto de desenvolvimento do sistema de switches e roteadores heterogêneas apresentado neste trabalho desenvolve um controle de dispositivos de rede sem se importar com as características de um determinado fabricante e sua marca, ou seja, um sistema que poderá atuar para qualquer tipo de roteador ou switch que esteja em uma rede local. O sistema de gerenciamento almejou buscar a facilidade na configuração destes equipamentos utilizando-se das tecnologias descritas no TCC I e TCC II. Os usuários da ferramenta podem se deparar com uma interface gráfica de fácil utilização para a administração dos dispositivos de rede podendo realizar diversas configurações de maneira fácil e prática.

Para a realização do sistema de gerenciamento foi preciso desenvolver primeiramente uma API[2] *(Application Programming Interface)* comum aos dispositivos como podemos ver na ilustração da figura 12. A API é, basicamente, um conjunto de rotinas e padrões estabelecidos por um software para a utilização de suas funcionalidades por programas aplicativos, sendo que os programas não estão interessados na implementação do software, mas apenas na utilização dos seus serviços.

O presente trabalho também contará com o desenvolvido de um *driver* para um roteador linux. O *driver*, no trabalho, é uma implementação específica, para cada tipo e modelo de dispositivo, e que não deve ser confundido com *drivers* no contexto de um Sistema Operacional.

Para efeito de modularidade, este trabalho utilizará a linguagem de programação PHP em seu desenvolvimento.

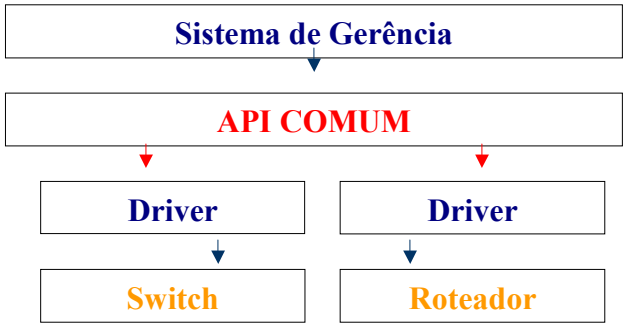

Figura 12 – Estrutura do Sistema gerenciamento.

[2] Mais recentemente o uso de APIs tem se generalizado nos chamados *plugins*, acessórios que complementam a funcionalidade de um programa.

Um dos objetivos da implementação do sistema é utilizá-lo na rede da UERGS na unidade de Guaíba, para servir de ferramenta de apoio na administração da sua infra-estrutura de rede, assim melhorando e facilitando na administração das configurações dos dispositivos de redes empregados. A aplicação desse sistema terá uma melhor disponibilidade de acesso a seus usuários, pois a ferramenta poderá ser acessada através do navegador de qualquer estação conectada na rede, precisando apenas de um cadastro para ser acessada. O sistema poderá ser de grande utilidade para os professores da unidade Guaíba, pois através de um micro conectado na rede da instituição, eles poderão criar uma interface lógica para se isolar da rede, bloquear o roteamento para a sub-rede da sala de aula onde está trabalhando etc. Enfim, muitas são as utilidades que podemos ter com o sistema desenvolvido e apresentado neste trabalho.

O funcionamento da ferramenta gerência pode ser vista na figura 13, que mostra um exemplo de atuação do sistema. Na figura 13, é representado o esquema de um professor em sala de aula executando ações no roteador através do sistema de gerenciamento. Uma das possibilidades que poderia ser realizado é o isolamento da sub-rede que se encontra a sala de aula, das demais sub-redes da LAN.

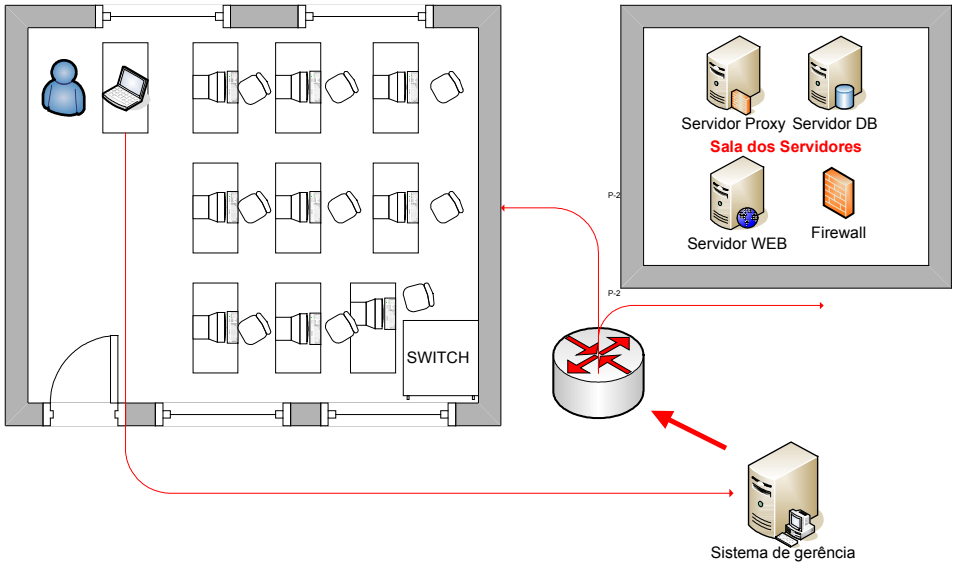

Figura 13 – Visão geral de funcionamento

4 DESENVOLVIMENTO PRÁTICO

Para a validação do projeto apresentado no TCC I sobre sistemas de switches e roteadores heterogêneos, é desenvolvido no TCC II uma parte do projeto, referente ao controle do dispositivo chamado roteador. Para tanto, foi construído um cenário de várias sub-redes, onde cada sub-rede se destina a um tipo de classe de usuário. Para se ter um controle independente por grupo de usuário, um roteador foi implementado para controlar a comunicação entre as sub-redes, como podemos ver na figura 14. Nesta figura podemos notar os diversos grupos de usuários encapsulados em suas sub-redes que são: servidores, acadêmica, professores, administrativa. As sub-redes são independentes uma da outra, sendo que cada ação feita em uma das sub-redes não afetará na outra.

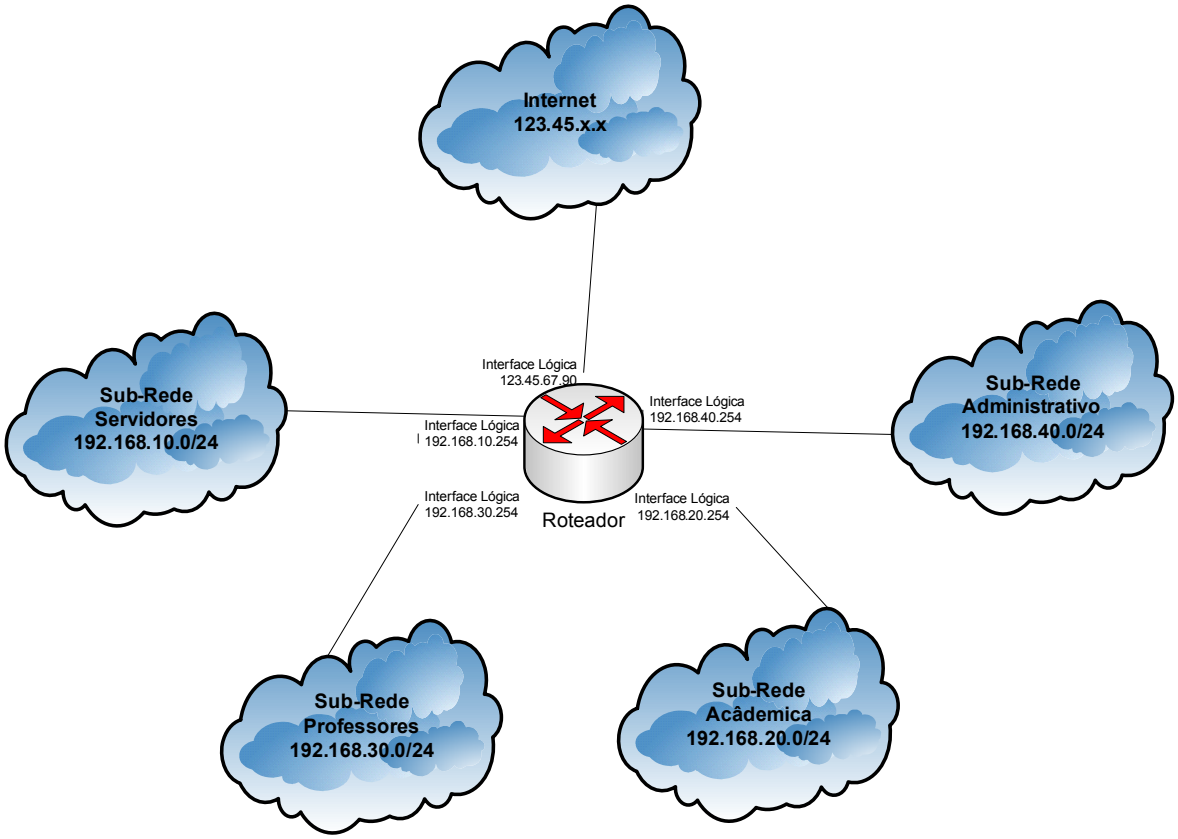

Figura 14 – Sub-Redes

Para a criação do cenário descrito para auxiliar na validação do Sistema de gerenciamento de switches e roteadores heterogêneos, foi necessário utilizar máquinas

virtuais (VMs) para o desenvolvimento prático. Criou-se três máquinas virtuais (VMs) com sistemas operacionais linux em cada uma delas, como podemos ver na figura 15. A primeira máquina virtual (VM) será desenvolvida o sistema de gerenciamento dos dispositivos totalmente modularizado na linguagem PHP. Nesta máquina virtual será criada também uma API comum que implicará em 4 funções básicas: criação de interfaces lógicas *(cria_il)*, remoção de interfaces lógicas *(remove_il)*, permitir roteamento *(permiti_rot)* e bloquear roteamento *(bloq_rot)*. O banco de dados (postgreSQL) utilizado pelo sistema de gerenciamento também estará nesta máquina virtual (VM).

A segunda máquina virtual (VM) estará o roteador, que foi implementado em um *linux ubuntu server*. Para criação das sub-redes, foi utilizada a tecnologia de VLANs. Como podemos ver na figura 15, foram criadas quatro placas de redes virtuais, sendo que cada placa virtual está destinada a uma sub-rede. A comunicação entre o sistema e o roteador foi feita através de SSH e não por SNMP. Por não se tratar de um dispositivo físico de roteamento, a melhor escolha foi utilizar o SSH para fins de validação do projeto. A última máquina virtual (VM) criada está destinada para testes do sistema de gerenciamento, onde ela representa uma estação de uma das sub-redes criadas. Podemos ver toda o esquemático de validação do TCC II na figura 15.

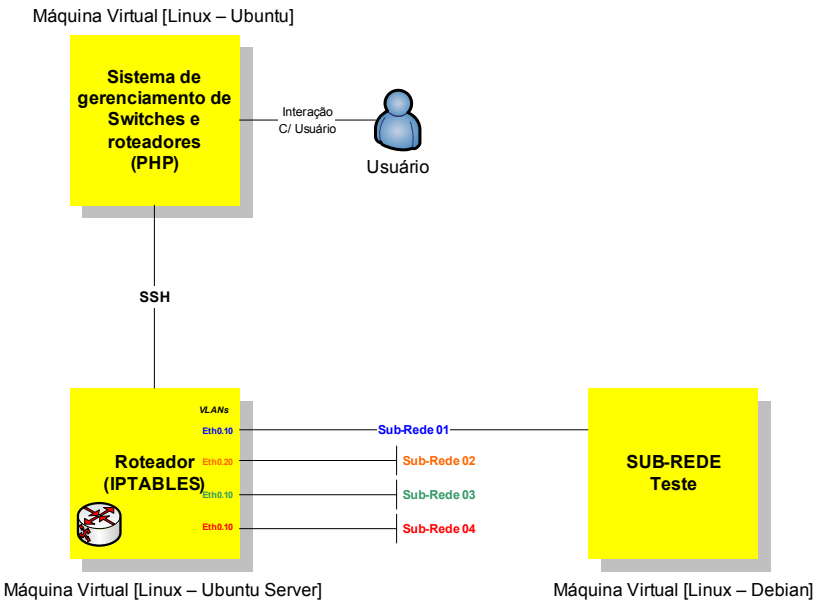

Figura 15 – Esquemático de validação do TCC II.

O trabalho desenvolvido preocupou-se em validar o sistema de gerenciamento proposto. O sistema implementado ficou dividido em várias etapas. A primeira etapa que nos deparamos é a tela de login para acessar o sistema, ver figura 16, onde usuários já foram previamente cadastrados no banco de dados (postgreSQL). Este por sua vez contém todos os usuários que poderão ter acesso ao sistema.

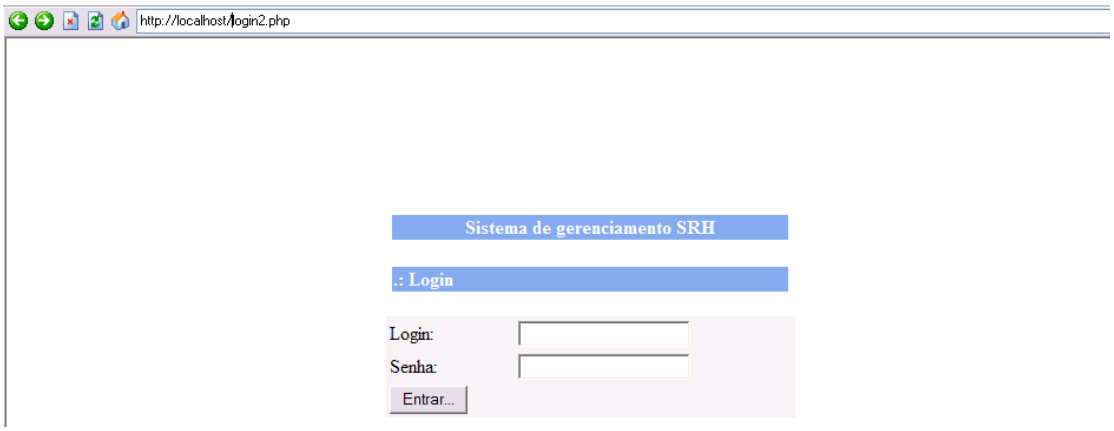

Figura 16 – Login de acesso ao sistema.

O banco de dados (postgreSQL) criado para ser utilizado pelo sistema de gerenciamento foi chamado de *Sistema,* onde foram criadas várias tabelas de dados para armazenamento e consulta**.** A tabela chamada *User* foi criada para armazenar nomes de usuários e senhas (*hash*), ver tabela 6.

Tabela 6 – Armazena nome de usuário e senha (*Tabela users*).

Usuário	**Senha (*hash MD5*)**
Dionata	*hash* do usuário
Eder	*hash* do usuário
Teste	*hash* do usuário

Para adição de novos usuários e suas respectivas senhas, se utiliza o seguinte comando: *INSERT INTO users VALUES ('nome_usuário', 'hash MD5');*, deste modo pode-se acrescentar mais usuário para terem acesso ao sistema.

Quando o usuário cadastrado digitar seu *login* de acesso ao sistema, haverá uma consulta ao banco de dados para verificação de usuário e senha. Se a consulta for bem

sucedida o usuário terá acesso ao sistema, se não for bem sucedida aparecerá uma tela de erro. O código de consulta de usuário e senha no banco de dados pode ser vista na figura 17.

```
<?php
session_start();
ob_start();
include('conexao.php');  // Conectando ao banco de dados Sistema
$login = $_POST["login"]; // login fornecido pelo usuário
$senha = $_POST["senha"]; // senha fornecido pelo usuário
$senha2=hash_md5($senha);
$sql="select * from users where usuario='$login' and senha='$senha2'";
$query=pg_query($conexao, $sql);
$total=pg_num_rows($query);
echo $logar;
  if($total>0){
   $usuario=pg_fetch_result($query,0,$name);
   session_register("usuario");
   session_destroy();
   header("location: menu2.php");
}else{
     session_destroy();
     header("location: erro2.php");
}
?>
```

Figura 17 – Tela de conferência de usuário cadastrados no BD

Quando o usuário efetuar o login com sucesso, ele se deparará com o menu principal, onde o usuário poderá escolher o tipo de ação a ser realizada. As ações já foram previamente apresentadas aqui: criação de interfaces lógicas, remoção de interfaces lógicas, bloquear roteamento e permitir roteamento. Esta tela pode ser vista na figura 18.

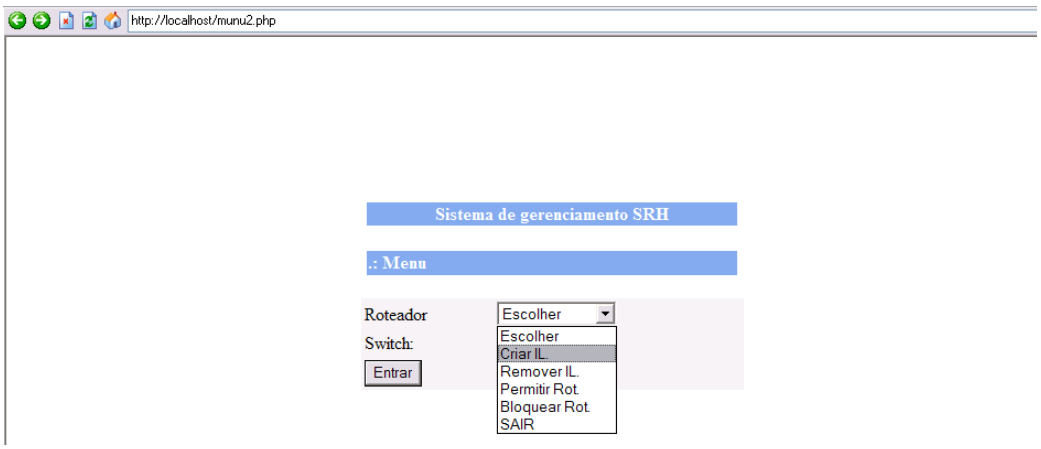

Figura 18 – Menu principal

4.1 Criar interfaces lógicas

O usuário quando escolher a opção no menu principal de *Criar IL* (interface lógicas), aparecerá uma tela (ver figura 19) para que ele determine os parâmetros necessários para sua criação. Para criar a nova interface lógica o usuário deverá fornecer os seguintes parâmetros: o IP da nova interface lógica, a máscara de rede que deseja atribuir para configuração da nova interface, a interface física que deseja ser manipulada e o número para a criação da nova VLAN.

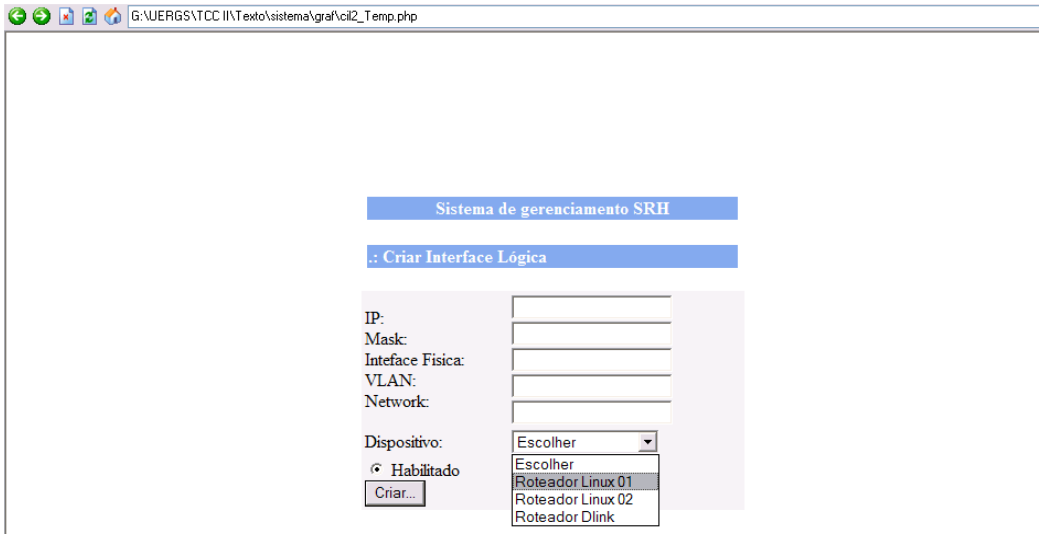

Figura 19 – Criar Interface lógica

O usuário ainda terá que escolher o dispositivo que vai "sofrer" as configurações feitas. Para isso, são listados para o usuário, os dispositivos cadastrados no banco de dados. Em nosso casso, temos somente o *roteador linux 01*, *roteador linux 02* e *dlink*. Mas os *roteadores linux 02* e *dlink* são somente para exemplo. A tabela 7chamada de *dispositivos*, foi criada para o armazenamento e consulta dos dispositivos de rede, nela contém os dispositivos cadastrados e seu respectivos IPs.

Tabela 7 – Armazena os tipos de dispositivos de rede (*Tabela dispositivo*).

ROTEADOR	IP
LINUX01	192.168.0.3
LINUX02	192.168.0.4
DLINK	192.168.0.5

Quando as informações requisitadas já tiverem sido fornecidas pelo usuário, o próximo passo, será clicar no botão "Criar" para que seja efetuada no roteador escolhida a configuração feita. Quando o usuário "mandar" executar as configurações, entrará em ação o algoritmo da API. A API coletará todos os dados fornecidos pelo usuário e colocará no banco de dados, na tabela chamada *Rede***,** através do comando *INSERT INTO rede VALUES ('dado01','dado02'...);*. Para ser estabelecida a conexão com o banco de dados, a linguagem PHP, utiliza função: *$conexao=pg_connect("host=localhost dbname=**sistema** user=**sistema** password=**sistema**"),* desta maneira, o sistema de gerenciamento tem acesso ao banco de dados criado e as tabelas. A API realizará também uma consulta sobre o dispositivo escolhido no banco de dados (figura 20 mostra o código da consulta) e chamará o *driver* correto para efetuar a ação de criação da nova interface lógica.

```
$roteador = $_POST["select"]; //dado fornecido pelo usuário
$sql="select * from dispositivo where roteador='$roteador'";//verificando no banco-tabela 7: dispositivo.
$query=pg_query($sql);
$total=pg_fetch_array($query,0,PGSQL_NUM);
```

Figura 20 – Trecho do Código da API para consulta do dispositivo.

Na figura 21 podemos ver o fluxo de ações do algoritmo da API.

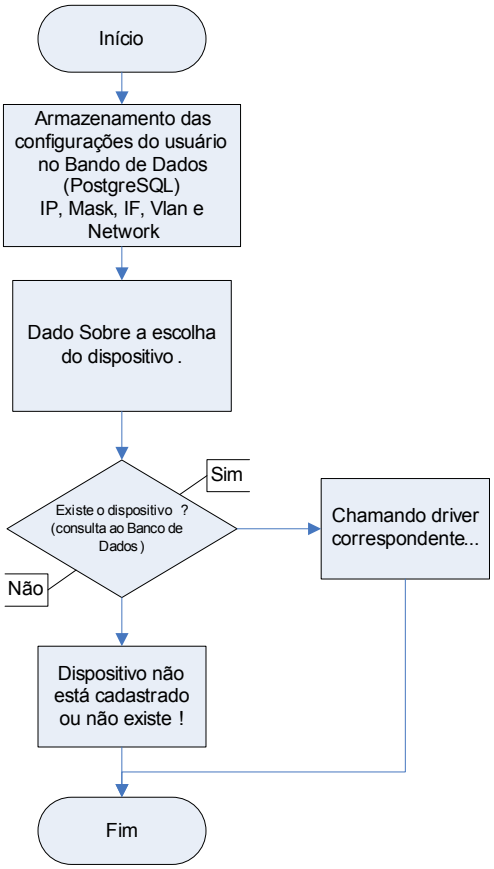

Figura 21 – Diagrama de blocos do algoritmo API.

A função especificada pelo usuário, criar interfaces lógicas, é executada no *driver* da seguinte maneira: Os dados necessários para configurar a nova interface são consultados no banco de dados na tabela *Redes,* ver tabela 8. Através da função *exec* do PHP, são executados comandos no terminal do linux, no qual é criado um canal de comunicação via SSH, com o comando: *ssh root@IP_server* na máquina roteadora. Neste canal são executados comandos para a criação e configuração para a nova interface de rede especificada pelo usuário. Os comandos executados para a criação da nova interface são: *vconfig add interface_física número_da_vlan* (este comando cria a placa virtual), *ifconfig vlan_criada IP netmask máscara_de_rede up* (configurando a nova interface lógica de rede), *vconfig set_flag nova_vlan* (seta a nova vlan criada). A figura 22 mostra o trecho do código do *driver* implementado que executa a criação da nova interface lógica.

```
//Trecho do código do driver implementado
function cria_il($ip2, $mask2, $if2, $vlan2, $ip_server) //Informações tiradas do BD - tabela 7: redes.
{

$placa_logica = $vlan2;
$vlan = "$if2.$vlan2";
$ip = $ip2;
$mascara = $mask2;

/*********************/
/* Criando vlan...*/
exec("sudo ssh root@$ip_server vconfig add $if2 $placa_logica");
exec("sudo ssh root@$ip_server ifconfig $vlan $ip netmask $mascara up");
exec("sudo ssh root@$ip_server vconfig set_flag $vlan 1");
exec("sudo ssh root@$ip_server ifconfig $vlan");
}
```

Figura 22 – Trecho do código da *driver* implementado para criação da nova IL

A possibilidade de criar interfaces lógicas traz uma série de vantagens para o usuário, pois o sistema de gerenciamento poderá criar sub-redes com extrema facilidade e rapidez. Utilizando o sistema de gerência poder-se criar um cenário de diversas sub-redes como mostrado na figura 13.

4.2 Remover interface lógica

No sistema de gerenciamento não existe apenas a opção de criar interfaces lógicas, existe também a possibilidade de remover interfaces lógicas criadas no roteador. Estas interfaces lógicas podem ser removidas quando o usuário escolher no menu principal *Remover IL* (interface lógicas). Neste momento, aparecerá na tela, a opção para que o usuário determine o dispositivo alvo, mostrada na figura 23.

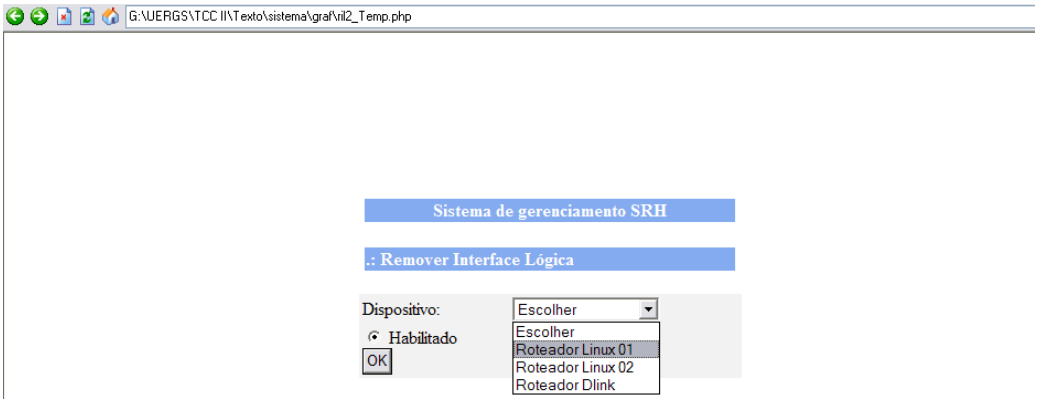

Figura 23 – Dispositivo a ser escolhido.

Após o usuário ter escolhido o dispositivo, aparecerá uma próxima tela onde o usuário poderá remover as interfaces já cadastradas no sistema. A API lista as informações que foram armazenadas na tabela *redes***,** do BD *Sistema*, ver tabela 8. A figura 24 mostra dois trechos de código da API que realiza esta ação.

```
//Numero de linhas do banco de dados
$num="select * from rede";
$query=pg_query($num);
$num2=pg_num_rows($query);

//Listando as informações da Tabela Redes do banco de dados
for($j=0; $j<=$num2; $j=$j+1)
  {
    $sql="select * from rede"; // selecionando a tabela 8: redes.
    $query=pg_query($sql);
    $rede=pg_fetch_array($query,$j,PGSQL_NUM);
    echo"<option value='$rede[3]'>$rede[0] - $rede[1] - $rede[2] - $rede[3]</option>"; // Mostra na tela lista
  }
```

Figura 24 – Trechos do código da API que lista as informações da tabela 8: *Redes.*

Tabela 8 - Armazena as informações fornecidas pelo usuário (*Tabela Redes*).

IP	**MASK**	**IF**	**VLAN**
192.168.10.254	255.255.255.0	eth0	10
192.168.20.254	255.255.255.0	eth0	20
192.168.30.254	255.255.255.0	eth0	30
192.168.40.254	255.255.255.0	eth0	10

Nesta lista o usuário deverá escolher qual a interface ele deseja remover. A figura 25 mostra a tela do sistema, onde o usuário fará esta remoção.

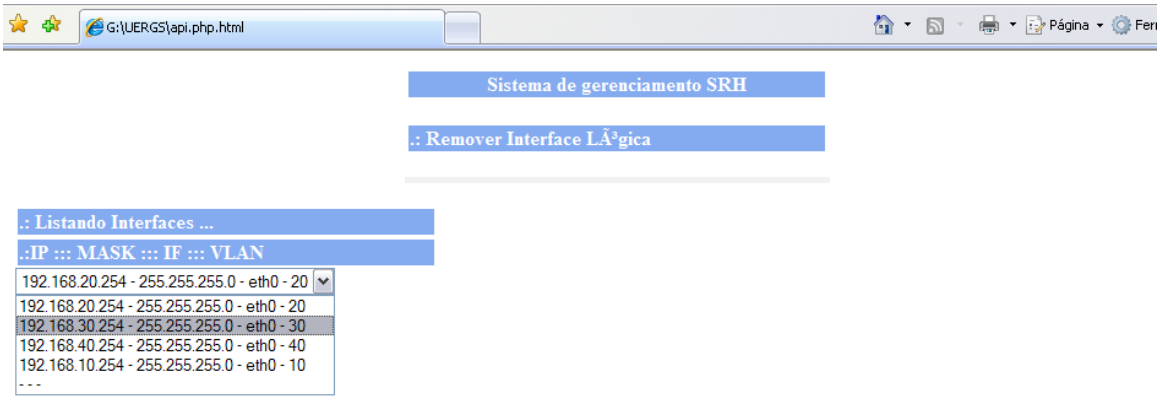

Figura 25 – Remover interfaces lógicas.

A remoção da interface lógica no roteador é feita da seguinte forma: no *driver* implentado com função *exec* do PHP, como foi descrito anteriormente, é possível executar os comandos no terminal do linux e com o protocolo SSH é criado o canal de comunicação com roteador. Assim aplicando os seguintes comandos para remoção da interface lógica: *vconfig rem interface_para_remoção.* A figura 26 mostra o trecho do código do *driver* implentado que realiza a remoção das interfaces lógicas.

```
/* Remove interfaces logicas*/
function remove_il($if2, $vlanrem, $ip_server) //inforamações retiradas do banco de dados
{
$vlan_r = "'$if2.$vlanrem";
exec("sudo ssh root@$ip_server vconfig rem $vlan_r"); //removendo interfaces lógicas
}
```

Figura 26 – Trecho do código do *driver* implementado de remoção de IL

A possibilidade de o usuário poder criar e remover interface lógica transforma o sistema de gerenciamento em uma ótima ferramenta de apoio para as redes. Tanto na remoção de interfaces quanto na sua criação, é feito de forma prática pela o usuário, ganhando tempo na configuração e re-configuração de estruturas de rede.

4.3 Permitir roteamento

Outra função que se pode realizar no sistema de gerenciamento é permitir roteamento. Neste caso, o sistema de gerenciamento fornece a possibilidade de reabilitar o roteamento de uma sub-rede previamente bloqueada. Esta função pode ajudar o administrador de rede, que necessita ter o controle independente das sub-redes existentes

em uma determinada infra-estrutura. Um exemplo de infra-estrutura de rede pode ser vista na figura 13. Nesta opção o administrador de rede poderá liberar o fluxo de dados para uma determinada rede bloqueada.

Para escolher esta opção o usuário do sistema de gerenciamento selecionará no menu principal a função *Remover rot*. (Remover roteamento). Após a seleção, surgirá a tela onde o usuário terá que fornecer os parâmetros da sub-rede que habilitará novamente o seu roteamento, esta tela pode ser vista na figura 27. Os parâmetros requisitados são: a interface física e o número da VLAN que está a sub-rede para bloqueio.

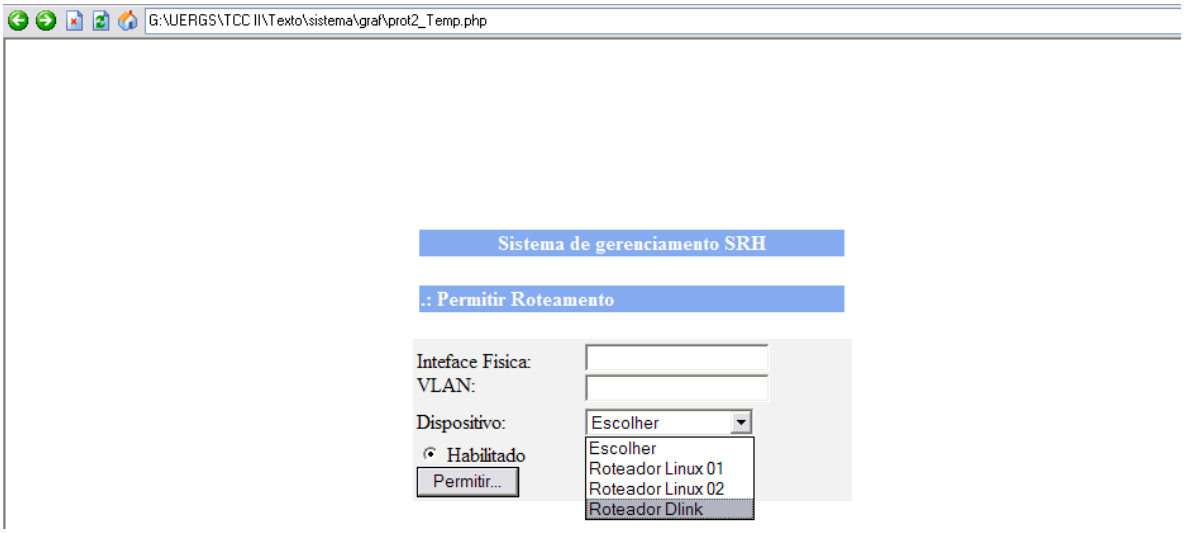

Figura 27 – Permitir roteamento

Para a realização da ação de permitir o roteamento, o *driver* implementado ativa a vlan, que estava desativada, através do comando: *ifconfig vlan_desativada up*. A figura 28 mostra o trecho do código que realiza está ação.

```
function permite_rot($if2, $vlan2, $ip_server) //Informações retiradas do BD
{
//echo "Permitindo roteamento...";
$vlan = "$if2.$vlan2";
$seroute= $ip2;
exec("sudo ssh root@$ip_server ifconfig $vlan up"); //permitindo roteamento
}
```

Figura 28 – Trecho do código do *driver* implementado de permissão para roteamento

4.4 Bloquear roteamento

A última função implementada neste trabalho, trata-se do bloqueio do roteamento das sub-redes. Esta opção traz uma série de vantagens para o administrador de rede, porque possibilita chavear o fluxo de dados de uma determinada sub-rede. Um exemplo prático, onde esta função pode ser muito útil, é o caso de um professor que pretende aplicar uma prova em uma determinada sala de aula (isolada em uma sub-rede), mas não quer que seus alunos tenham acesso a internet. Bloqueando o roteamento na sub-rede que a sala de aula se encontra, o professor estará automaticamente bloqueando a internet.

Para bloquear o roteamento pelo sistema primeiramente você deve escolher no menu principal a opção *bloquear rot.*(bloquear roteamento), neste instante aparecerá uma tela, que pode ser vista na figura 29. Nesta tela serão pedidos às configurações para o bloqueio da sub-rede que se deseja restringir, os parâmetros pedidos são: a interface física e o número da VLAN que está a sub-rede.

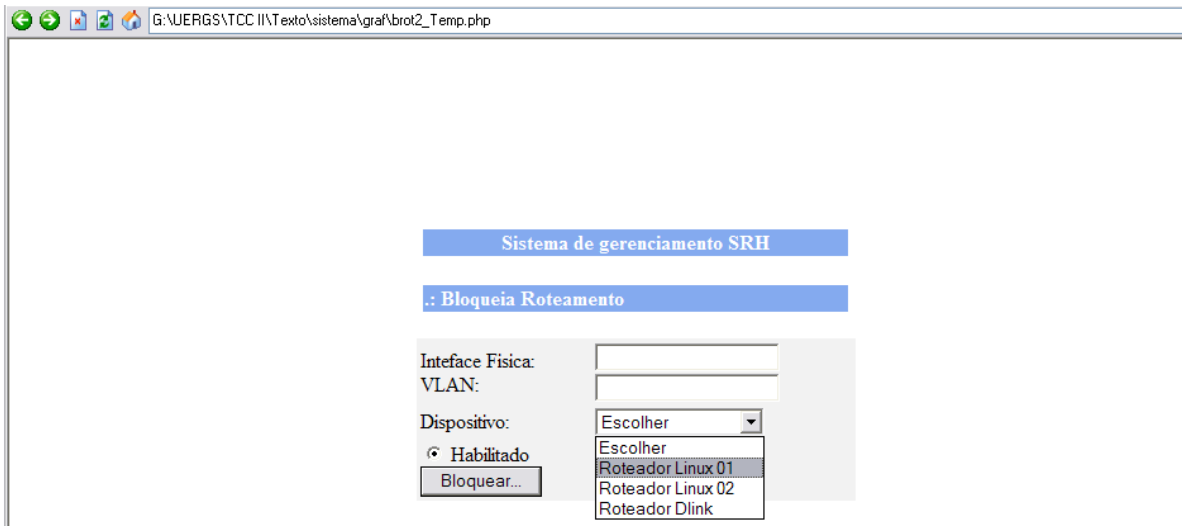

Figura 29 – Bloquear roteamento

Para a realização da ação, bloquear roteamento, o *driver* implementado realiza a desativação da vlan, através do comando: *ifconfig vlan_desativada down.* A figura 30 mostra o trecho do código que realiza a ação.

```
//função bloquear roteamento
function bloq_rot($if2, $vlan2, $ip_server)//informação retirada do BD tabela Redes
{
$vlan = "$if2.$vlan2";
exec("sudo ssh root@$ip_server ifconfig $vlan down"); //Desativação da vlan
}
```

Figura 30 – Trecho do código do *driver* implementado de bloquear roteamento

Após a escolha ter sido realizada, entra em ação a API, que já foi mostrada na figura 21. Todas as funções passam pela API, que chama e direciona para o *driver* correto**.** Para este projeto foi implementado um *driver* para um roteador linux, o fluxo das informações processadas por ele é mostrado no diagrama da figura 31. Neste diagrama são mostrados os passos das informações processadas pelo *driver linux*.

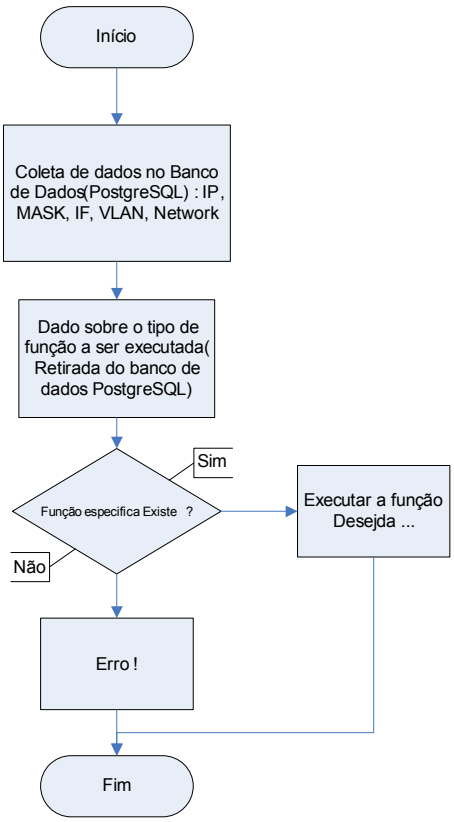

Figura 31 – Diagrama de blocos do algoritmo do *driver*_linux implementado.

5 CONCLUSÃO

Nos dias atuais, devido ao crescimento das redes locais, vêm se tornando indispensáveis para qualquer instituição a criação de sistemas que as gerencie. Desta forma se faz necessário o desenvolvimento de trabalhos acadêmicos desta natureza, por tornar mais inteligente e dinâmico o controle das infra-estruturas empregadas. A utilização de novas tecnologias proporciona as infra-estruturas existentes um melhor desempenho, podemos citar as tecnologias de VLANs e máquinas virtuais. Estas tecnologias aproveitam o que já existe de infra-estrutura e extraem melhorias, tanto em termos de seu desempenho a sua administração dos recursos existentes. O sistema proposto neste trabalho buscou integrar tecnologias a fim de um bem comum tanto na questão de dinamismo, quanto na questão de inteligência e facilidades no que diz respeito ao controle das redes locais. Ao concentrar tecnologias, que antes trabalhavam de forma independente, em um único sistema agregou-se um melhor potencial para a ferramenta.

Os potenciais deste projeto provêm da reunião de tecnologias e serviços, que na maioria dos casos trabalham de forma não integrada. O presente trabalho trouxe a proposta de criar uma ferramenta de apoio às redes locais e compreende o grande campo de inovações que possa se fazer em termos de infra-estruturas de rede. Pode-se dizer, que as inovações tecnológicas ajudam cada vez mais a criar sistemas de gerenciamento de redes, como no caso desta ferramenta de apoio as redes locais. A grande vantagem desse sistema proposto é a mobilidade do gerenciamento, onde pode ser acessado por qualquer tipo de navegador.

6 REFERÊNCIAS BIBLIOGRÁFICAS

[1] COMER E. Douglas;Interligação em redes TCP/IP volume I; Rio de Janeiro; Ed: Campus, 1999.

[2] COMER E. Douglas;Interligação em redes TCP/IP volume II; Rio de Janeiro; Ed: Campus, 1998

[3] TANENBAUM S. Andrew; Redes de Computadores; Rio de Janeiro; Ed:ElsevierEditora Ltda.

[4] SOARES Gomes Fernando Luiz, LEMOS Guido, COLCHER Sérgio; Redes de computadores; Rio de janeiro; Ed: Campus, 1995.

[5] TAROUCO M. R. Liane; Redes de computadores locais e de longa distância; São Paulo; Ed: Mcgraw-hill, 1986.

[6] GIOZZA William Ferreira; Redes locais de computadores protocolo de alto nível e avaliação de desempenho; São Paulo; Ed: Mcgraw-hill, 1986.

[7] COMER E. Douglas; Redes de computadores e internet; Porto Alegre; Ed: Bookman, 2001.

[8] FLINT Amettimatthew; Internetworking with TCP/IP Volume I; Rio de janeiro; Ed: Campus, 1995.

[9] HELD Gilbert; Comunicação de dados; Rio de Janeiro; Ed: Campus, 1999.

[10] FOROUZAM A. Behrouz; Comunicação de dados e redes de computadores.

[11] CIRNE Walfredo; curso redes; Disponível em: <http://walfredo.dsc.ufcg.edu.br/cursos/2003/redes20031/p2a.pdf>.Acesso em: 12 abril. 2007.

[12] PICCININ; UFSM Informática; Disponível em: <http://www-usr.inf.ufsm.br/~piccinin/osi>.Acesso em: 12 abril. 2007.

[13] Redes de computadores e suas aplicações na educação; Disponível em: <http://penta.ufrgs.br/homeosi.html>Acesso em: 13 abril. 2007.

[14] PICCININ; UFSM Informática; redes; Disponível em: <http://wwwusr.inf.ufsm.br/~piccinin/osi/rede.html> Acesso em: 13 abril. 2007.

[15] FAQ Tech the; O que é VLAN; Disponivel em
<http://www.tech faq.com/lang/pt/vlan.shtml> Acesso em: 13 maio. 2007.

[16] WASHINGTON University in ST. Luiz; conceitos sobre VLANs; Disponivel em <http://www.cs.wustl.edu/~jain/cis788-97/ftp/virtual_lans/index.htm>. Acesso em: 14 maio. 2007

[17] CISCO systen, inc ; Disponivel em <http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/snmp.htm>. Acesso em: 15 junho. 2007