UNIVERSIDADE FEDERAL DE ITAJUBÁ

PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Nome: Murilo Fujita

Integração das funções de supervisório e gerenciamento de rede via SNMP

Orientador: Prof. Dr. Luiz Edival de Souza

Itajubá, Maio/2010

Dedicatória

Ao meu pai Mario Mitiyoshi Fujita que lutou persistentemente pela vida mostrando-me que se deve acreditar com afinco no que se deseja. A força de vontade dele é um valioso exemplo para realizar esta conquista e outras futuras.

Agradecimentos

Realizar este trabalho exigiu um somatório de contribuições muito importantes. Cito cada parcela a seguir:

Agradeço aos meus pais por investirem na minha educação e me ensinarem a valorizar o conhecimento.

Agradeço à minha irmã pelo apoio e paciência quando os momentos eram muito difíceis.

A esta Universidade pela oportunidade de realizar as pesquisas.

Ao meu orientador pelas ideias, as quais me conduziram ao longo dos estudos.

Aos amigos, acadêmicos ou não, pelas sugestões que colaboraram de diversas formas.

A minha gratidão a todos.

Sumário

Índice de Figuras

Índice de Tabelas

Lista de abreviações

Abreviação	Significado
API	Application Programming Interface
ASN	Abstract Syntax Notation
ATX	Advanced Technology Extended
BASH	Bourne Again Shell
BER	Basic Encoding Rules
CCITT	Consulative Committe for International Telegraph and Telephone
CGI	Common Gateway Interface
CSS	Cascading Style Sheets
DLL	Dynamically Linked Library
EGR	Estações de Gerenciamento de Rede
GPL	GNU Project License
HTTP	Hypertext Transfer Protocol
IAB	Internet Activities Board
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISO	International Organization for Standartization
ITU-U	International Telecommucations Union
LAN	Local Area Network
MG	Minas Gerais
MIB	Management Information Base
MIME	Multipurpose Internet Mail Extensions
OID	Object Identification
PC	Personal Computer
PEN	Private Enterprise Number

PID	Process Identification
POP	Post Office Protocol
RAM	Random Access Memory
RFC	Request For Comments
RNP	Rede Nacional de Pesquisas
SGMP	Simple Gate Management Protocol
SMI	Structure of Management Information
SMS	Short Message Service
SMTP	Simple Mail Transport Protocol
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator
UTP	Unshield Twist Pair
WAN	Wide Area Network

Resumo

Esta dissertação tem como objetivo utilizar o protocolo SNMP para exercer as atividades de Supervisão de Processos e Gerenciamento de forma remota. Como a utilização do protocolo é feita através de linhas de comando, uma arquitetura é desenvolvida para facilitar a interação. Esta aplicação amigável realiza a comunicação com os dispositivos monitorados bem como o estado dos computadores em rede.

Nessas redes há funções distintas a serem desempenhadas. O gerente concentra as informações de todas as máquinas que consulta através de um processo chamado *poll*. O agente responde às solicitações e podem ter dispositivos conectados caso estes não tenham o protocolo SNMP em seu firmware. Como exemplo, estes dispositivos podem ser placas de aquisição de dados, Controlador Lógico Programável ou *switches* gerenciáveis. Outra tarefa é chamada de *trap*, notificações enviadas pelos agentes sem que haja o pedido do gerente.

Os serviços do protocolo SNMP estão disponíveis através de comandos. Compreendendo o propósito dos comandos bem como suas opções, a sequência destes comandos forma o *script*. Na arquitetura proposta foram desenvolvidos vários scripts que são organizados em *scripts* para gerenciamento e para supervisão de processos.

Foram desenvolvidos oito *scripts* que fornecem as informações mais relevantes do agente como a capacidade de memória RAM, o desempenho da rede de comunicação, estatísticas, tempo que a máquina está ligada entre outros.

Para realizar a função de supervisão de processos foram desenvolvidos quatro *scripts* para obter os valores das variáveis de processo. O propósito é capturar as grandezas físicas em uma rede que podem ser do tipo digital ou analógico.

Palavras-chave: Protocolo SNMP, supervisório, gerenciamento de rede.

Abstract

The goal of this research is employ the SNMP protocol to accomplish the supervisory and management activies through remote access. The protocol is used through command line, a software was developed to handle easier. This friendly application executes a communication with managed devices and the computer healthy.

There are different functions in this networking to perform. The Network Management Station (NMS) concentrates the data of all machines through a process called poll. The agent replies the queries and devices can be connected if they have SNMP protocol in their firmware. For instance, these devices can be data acquisition board, Programmable Logic Controller or switches. Other task is called trap, warnings sent by agents without a request of NMS.

The SNMP protocol is available through the commands. Understanding the purpose of commands and their options, the sequence of commands became the script. This software has many scripts divided in scripts for management and for supervisory.

There are eight scripts that provide the most important data of agent such as RAM memory amount, performance of network, statistics, time up among other resources.

To perform the supervisory functions was developed four scripts to get values of variables in the process. The purpose is capture the physical quantities in a network that be digital or analogic kind.

Key words: SNMP protocol, supervisory, networking.

1. Introdução

Atualmente os parques computacionais interligam uma grande quantidade de dispositivos com a finalidade de compartilhar recursos. Independente do sistema operacional, computadores e diversos elementos de rede trocam informações através de protocolos de comunicação, sendo o mais utilizado o protocolo TCP/IP. Além dos aspectos de utilização de recursos e das características dos protocolos de comunicação utilizados, é preciso administrar as condições de funcionamento de cada equipamento interconectado, tarefa realizada pelo SNMP – *Simple Network Management Protocol*. Este é um protocolo de gerenciamento de rede que tem as seguintes características:

- Acesso remoto. Os comandos deste protocolo alcançam máquinas que estejam em outras redes para obter um valor de um objeto ou armazenar um novo;
- Informar sobre a ocorrência de erro. O protocolo pode ser configurado para monitorar um processo ou estado de um dispositivo e avisar o responsável sobre um determinado problema. Desta forma, é possível ter a localização precisa de onde e quando ameaças à integridade de redes ocorrem;
- Extensão das funcionalidades. Para obter respostas do sistema remoto, é possível personalizar o sistema criando seus próprios programas. Esta prática é recomendada, pois o protocolo realiza apenas tarefas padrões necessitando aumentar seu potencial.

Por outro lado, a automação industrial necessita de instrumentos que tenham funções de medir, comparar, transmitir e atuar nos processos para obter um determinado produto com a mínima intervenção humana. Independente da quantidade de dispositivos para administrar, é preciso de uma rigorosa fiscalização da condição do sistema. Para auxiliar nesta tarefa emprega-se o sistema supervisório que ao invés de utilizar painéis operados por botões,

chaves liga e desliga, um software mostra o ambiente com figuras, imagens, símbolos e ícones tornando mais amigável interagir com o conjunto. Um sistema supervisório tem que ter o cuidado de não poluir a tela com informações excessivas durante o funcionamento. Ao surgir emergências no sistema, deve facilitar ao operador a identificação da ocorrência no sistema.

Enfim, o supervisório tem o propósito de aquisitar dados, analisá-los e ser uma ferramenta que auxilie o operador a tomar decisões.

Para abordar estes assuntos, a dissertação é organizada da seguinte forma:

Capítulo 1: apresenta as atividades de gerenciamento de rede e supervisório e identifica diversos softwares que interpretam o protocolo SNMP.

Capítulo 2: aborda conceitos que envolvem o protocolo de gerenciamento, apresenta as funções das máquinas conhecidas como gerentes e agentes, bem como a instalação e configuração para tratar do seu funcionamento. Introduz as ferramentas e propósitos abordados na dissertação.

Capítulo 3: trata dos ajustes finos para estabelecer a comunicação e cuidar da segurança. Aborda, também, o conteúdo dos arquivos em relação ao comportamento do protocolo, à configuração dos roteadores e ao desenvolvimento de *scripts*, tanto para o gerenciamento de rede como para o supervisório.

Capítulo 4: reúne alguns resultados obtidos que são mostrados através de capturas de telas do software desenvolvido. Adicionalmente, mostra-se os procedimentos para criar um sistema que requer autenticação de acesso a certos conteúdos.

Capítulo 5: apresenta as conclusões da dissertação e sugestões de trabalhos futuros fazendo uso dos mesmos conceitos em outros dispositivos e empregando uma versão mais segura.

1.1. Objetivo

O objetivo desta dissertação é propor uma arquitetura de software capaz de realizar em um único ambiente, o gerenciamento operacional de dispositivos conectados em rede e a monitoração/controle de entradas/saídas conectadas aos dispositivos. Eventualmente, pode-se utilizar a arquitetura para controlar dispositivos, mas como os aspectos temporais não são tratados, o cumprimento de *deadlines,* isto é, o tempo máximo para execução de atividades de controle pode não ser garantido.

Assim os dados e informações necessárias são obtidos através dos serviços disponíveis no protocolo SNMP. Conforme será visto posteriormente, este protocolo possui serviços específicos para o gerenciamento de rede. Serão desenvolvidas extensões do protocolo para monitoração de processos. Este trabalho combina as atividades de gerenciamento com atividades de monitoração/controle, ou seja:

1. Gerenciamento de rede: utiliza serviços específicos do protocolo SNMP para obter informações relativas ao estado operacional e dos recursos computacionais da máquina remota. Exemplo, verifica se a máquina está ligada e quanto tempo em operação, verifica se não ocorreram mudanças de hardware da máquina, etc.
2. Monitoração/controle: também chamado Supervisório, responsável por monitorar valores de grandezas físicas obtidas por uma placa de aquisição de dados, ou qualquer outro dispositivo, que ofereça uma interface de entrada e saída entre a máquina e o processo monitorado. Eventualmente, exerce o controle em atuadores sem restrições rígidas de tempo.

1.2. Limitação da internet

Um fator relevante é o tempo que se leva para a informação percorrer sua trajetória, uma vez que os dados trafegam pela internet, e ao sair de uma arquitetura de rede e chegar a outra, valores pequenos de atrasos indicam um caminho não congestionado, enquanto que atrasos maiores refletem em congestionamento da rede. O tempo de atraso de uma transmissão deve ser aceitável ou uma conexão dedicada deve ser usada.

Cada situação deve ser estudada para avaliar o impacto do atraso. Neste trabalho não há casos extremos como missões críticas, portanto, determinados atrasos são aceitáveis sendo, o mais importante, a garantia da ocorrência do evento acontecer. Para que haja segurança na execução da tarefa, o protocolo conta com configurações que realizam relatórios, notificam por e-mail ou simplesmente exibem o comportamento determinado comando.

1.3. Opções de softwares que interpretam SNMP

Há uma boa variedade de softwares que interpretam SNMP, com características diversas desde gratuitos a comerciais. É importante antes da escolha do software, saber quais dispositivos estão conectados em rede para adequar o tráfego de informações, já que a quantidade de consultas também é relevante.

A seguir são listados alguns dos softwares que utilizam-se do SNMP [1]: HP Extensible SNMP Agent, Sun Microsytems, Concord SystemEDGE, Microsoft, Net-SNMP (formalmente chamado de projeto UCD-SNMP), SNMP research, HP OpenView NNM, HP OpenView ITO, Tivoli NetView, Castle Rock SNMPc, Computer Associates Unicenter TNG Framework, Veritas NerveCenter, OpenRiver, GxSNMP, Tkined e OpenNMS.

Adicionalmente, há programas que analisam o tráfego da rede como Concord eHealth, Trinagy, Cricket, InfoVista e MRTG. Devem ser usados para tratar das dificuldades encontradas na rede, pois guardam um histórico dos eventos ajudando a identificar o início de um problema.

O uso destes softwares não é muito amigável ao ambiente de trabalho de um profissional envolvido com automação. Neste sentido, esta dissertação propõe criar um ambiente no qual o profissional pode se abstrair de detalhes da utilização do protocolo e trabalhar com recursos disponibilizados em forma mais amigáveis e integrado à operação de supervisão.

1.4. Outras pesquisas envolvendo o protocolo SNMP

No texto a seguir, são apresentadas pesquisas que mostram a abrangência do emprego do protocolo SNMP.

Em [20] o protocolo SNMP foi empregado para criar uma rede de sensores inteligentes. A abordagem trata de uma alternativa de manter as informações confiáveis já que nas redes atuais é necessário monitoramento para detectar anomalias. Os sensores coletam informações de instrumentos de diversos fabricantes e, através do protocolo SNMP, armazenam individualmente nas OIDs das MIBS. Tais informações são entregues para os agentes que atualizam suas MIBs armazenando a informação mais recente. Os agentes, por sua vez, passam para o gerente que concentram todas as informações coletadas. No caso de situações como bateria fraca, baixa intensidade de sinal recebido entre outros casos que necessitem uma intervenção, traps são disparadas para notificar a ocorrência.

Em [21] a troca de redes que os usuários se conectam podem comprometer a integridade das suas informações. O propósito do sistema é monitorar e gerenciar dispositivos e modificar o estado do pacote para um usuário remoto. Este gerenciamento é controlado de acordo com as regras

estabelecidas para os usuários baseado no protocolo SNMP. Este protocolo fornece uma análise segura dos pacotes atualizados.

Em [22] apresenta um projeto sobre um sistema de comunicação pelas linhas energizadas. O sistema manipula sinais de alta frequência de 10 kHz a 30 MHz e um filtro separa o sinal e as informações recebidas. A MIB é necessária para definir todas as informações tais como: armazenar as configurações de cada equipamento como protocolo, segurança, método de conexão e emitir um sinal quando houver algum erro. Para este sistema foram desenvolvidas três ferramentas: uma para tratar se as informações chegaram ao destino, a segunda para atuar checando as entradas e saidas e a última para previnir a perda de informações buscando uma rota alternativa caso haja uma interrupção entre um gerente a agente, por exemplo.

Em [23] é proposto uma solução que permite a integração de equipamentos em um ambiente gerenciados pelo protocolo SNMP. Os autores adotaram o Net-SNMP que implementa o SNMPv1, SNMPv2c e SNMPv3. Para aumentar a possibilidade de recursos, podem ser usados SNMP multiplexing (SMUX) e Agent Extensibility (AgentX). Esses protocolos incluem uma biblioteca de desenvolvimento para novas aplicações SNMP em C e Perl APIs.

1.5. Proposta de utilização do protocolo SNMP

Como resultado desta pesquisa, elaborou-se uma arquitetura na qual o usuário pode consultar um computador que reúne informações de gerenciamento e controle com o uso do protocolo SNMP. A interface gráfica (GUI) é visualizada através de um navegador que exibe os recursos através de páginas. Há uma grande facilidade em utilizá-lo, pois é tão intuitivo quanto navegar pela internet. Os mesmo elementos encontrados em qualquer página como botões, caixas de textos e *checkboxes* são utilizados para interagir com o sistema.

A ideia é desenvolver uma aplicação que faz uso do protocolo SNMP. Em vez de digitar longas linhas de comando, que são susceptíveis a erros, a interface interpreta as solicitações do usuário e invoca o SNMP que interage com a máquina remota [4].

2. Visão Geral dos recursos para gerenciamento

2.1. Estrutura da informação para gerenciamento

O SMI é um conjunto de regras que define como os objetos podem ser acessados através de um protocolo de gerenciamento de rede [15]. É a estrutura que permite ao desenvolvedor entender mensagens dentro do contexto SNMP no ato de receber e enviar dados. Os objetos são armazenados em uma base de dados conhecida como MIB que será vista posteriormente.

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

i) OID: Cada objeto tem um nome ou OID que é representado de duas formas:
   - i.1) numérica: cada nó da MIB é identificado por um número e separada por ponto;
   - i.2) textual: cada nó da MIB tem um nome para facilitar a compreensão.

ii) Tipos e Sintaxes: ASN.1 define a notação que caracteriza os objetos dentro da MIB.

iii) Codificação: A forma como os objetos são codificados e decodificados para serem transmitidos por um meio como a Ethernet. A conversão é feita pelo BER, um identificador que analisa as sintaxes para interpretar os comandos SNMP através de bits e bytes.

Existem duas versões do SMI, sendo que a primeira versão é a adotada atualmente e a segunda versão é uma proposta experimental [19].

2.2. Sintaxe de Notação Abstrata 1

ASN.1 é um padrão de dados usado por diversos aplicativos e dispositivos para que as mensagens trocadas sejam compreendidas independentemente da plataforma. Por exemplo, é possível comunicar um PC com *Windows* instalado e uma máquina *Sun Sparc*. As aplicações que utilizam este padrão são numerosas e algumas são citadas a seguir: chamadas 0800, comércio através de internet, envio de e-mail seguro, sistema de tráfego aéreo, programas multimídia como *NetMeeting* da *Microsoft* entre outras [16].

O padrão define o formalismo para a especificação de tipos abstratos de informação que podem ser: inteiros, booleanos, strings, estruturas, listas entre outros. O ASN.1 tem uma característica semelhante às linguagens C/C++ e Java as quais têm instruções de como tratar os dados em tempo de execução, ou seja, como são definidos para serem examinados pelo analisador de sintaxe [14]. O padrão ASN.1 constrói os objetos que compõem a MIB definindo a organização dos nós de forma hierárquica. O exemplo a seguir mostra uma estrutura típica: declaração das definições a serem importadas, a forma como os nós são ligados aos outros e o tipo de acesso (somente-leitura ou com permissão de escrita).

```
1.  RFC1155-SMI DEFINITIONS ::= BEGIN
2.     EXPORTS -- EVERYTHING
3.  internet,    directory,    mgmt,    experimental,    private,
    enterprises,     OBJECT-TYPE,     ObjectName,     ObjectSyntax,
    SimpleSyntax,  ApplicationSyntax,  NetworkAddress,  IpAddress,
    Counter, Gauge, TimeTicks, Opaque;
4.
5.     -- the path to the root
6.
7.         internet OBJECT IDENTIFIER ::= { iso org(3) dod(6) 1 }
8.         directory OBJECT IDENTIFIER ::= { internet 1 }
9.         mgmt OBJECT IDENTIFIER ::= { internet 2 }
10.        experimental OBJECT IDENTIFIER ::= { internet 3 }
11.        private OBJECT IDENTIFIER ::= { internet 4 }
```

```
12.         enterprises OBJECT IDENTIFIER ::= { private 1 }
13.
14. -- definition of object types
15.
16. OBJECT-TYPE MACRO ::=
17. BEGIN
18.     TYPE NOTATION ::= "SYNTAX" type (TYPE ObjectSyntax)
19.         "ACCESS" Access
20.         "STATUS" Status
21.     VALUE NOTATION ::= value (VALUE ObjectName)
22.         Access ::= "read-only"
23.                  | "read-write"
24.                  | "write-only"
25.                  | "not-accessible"
26.         Status ::= "mandatory"
27.                  | "optional"
28.                  | "obsolete"
29. END
```

De acordo com a notação apresentada, o nó “internet” (linha 7), por exemplo, segue a sequência “iso”, “org” e ”dod” (ou 1.3.6 se for seguindo os números relacionados aos nós). A vantagem da notação numérica é a economia de espaço em pacotes, mais compacta. A ordem numérica torna eficiente a busca por um determinado objeto por causa da ordenação lexográfica que tem um comportamento igual a um dicionário, ou seja, organiza a seqüência.

A seguir é apresentada a aplicação prática deste padrão, tornando mais claro o assunto.

2.3. Base de informação para gerenciamento

A MIB tem um comportamento similar ao esquema de um sistema de banco de dados: armazena uma lista de objetos gerenciados e seus valores. Esta lista contém informações, por exemplo, se uma placa de rede está ligada, desligada ou em teste. Há também informações estatísticas como a quantidade

de mensagens enviadas, recebidas, ocorrências de erros e outros dados pertinentes à comunicação em rede.

Tais objetos são identificados através de OIDs que são uma sequência de inteiros não-negativos separados por ponto (.), organizados hierarquicamente como o UNIX ou DOS organizam os nomes de arquivos. Para facilidade de uso, um nome textual é associado com cada sequência do elemento de uma OID. A representação de um OID é feita através do padrão ASN.1 como mostrada abaixo:

"{" { {<*name*>["("<*number*>")"]} | <*number*>}... "}"
ou
<*number*> ["."<*number*>]...

Por exemplo:

{ iso org(3) dod(6) internet(1) } ou 1.3.6.1
{ internet 4 } ou 1.3.6.1.4
{ tcp 4 } ou 1.3.6.1.2.1.6.4

Uma vez apresentada a formação do OID, a Figura 1 mostra o arranjo hierárquico de como localizar um nó.

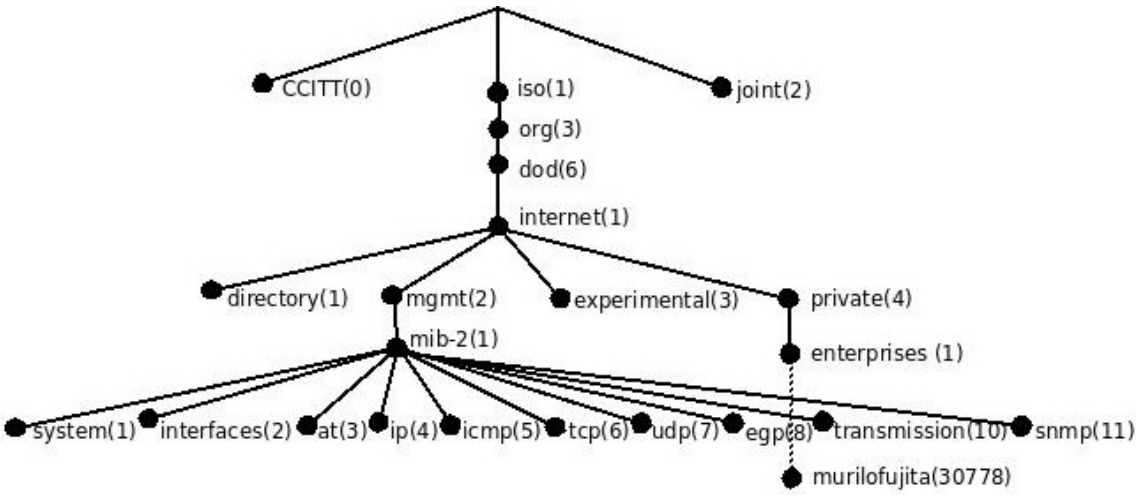

Figura 1 - Árvore MIB

O primeiro nó não tem nome e é dividido em três subníveis sendo:

1. CCITT (Consulative Committe for International Telegraph and Telephone), atual ITU-U (International Telecommunications Union).

2. ISO (International Organization for Standartization)

3. joint-iso-ccitt que é uma administração conjunta das duas organizações anteriores.

Somente a ramificação do nó ISO é usada, e nesta encontra-se o DOD, que é o Departamento de Defesa Americana. A IAB (*Internet Activities Board*) designou um sub-nó chamado internet que se divide em mais quatro, porém o nó “directory” não é usado.

As MIBs experimentais estão sendo desenvolvidas pelo IETF definindo sob o ramo “experimental”. A posição do OID definidos no módulo é determinada antes de escrever o módulo MIB. Para MIBs desenvolvidas pelo IETF, um ramo deve ser atribuído sob o ramo “internet experimental”, um local comum para criar os ramos “experimental” e cada módulo MIB lançado. Ramos podem ser criados dentro do ramo experimental para fazer teste e protótipo. Uma vez que a MIB publicada, os itens não podem ser trocados, eles podem apenas passar a obsoletos ou serem recriados. Desta forma, a prática padrão é

projetar sob um ramo “experimental”, testá-la, e então movê-la para o ramo “standard” quando o documento MIB é publicado [1].

Sob o nó mgmt está a MIB-2, que guarda as informações de gerenciamento. Inicialmente este nó teve o nome de MIB-1, mas a versão melhorada mudou para MIB-2. As melhorias da versão foram: abrangência para funcionar em mais dispositivos, melhoria com a compatibilidade SMI/MIB, aumento para entidades multi-protocolos e forma textual mais clara [1].

Para MIBs particulares existe um ramo específico que se encontra sob o ramo “enterprises”. Neste ramo, encontra-se o registro do autor desta dissertação, murilofujita. Para localizá-lo, portanto, deve-se seguir a hierarquia de nós representada por:

iso.org.dod.internet.private.enterprises.murilofujita ou .1.3.6.1.4.1.30778

O assunto seguinte trata do procedimento de como adquirí-lo.

Para facilitar a visualização da árvore MIB e o conteúdo de cada nó há uma grande variedade de programas gráficos como o da Figura 2, mostrando o iReasoning MIB browser. Muitos programas que fazem esta função são disponibilizados em versões de testes e funcionam durante certo tempo ou são limitados para determinadas operações. Para que funcione corretamente, os mesmos conceitos básicos devem ser configurados no programa: número da porta, nome do agente e da comunidade.

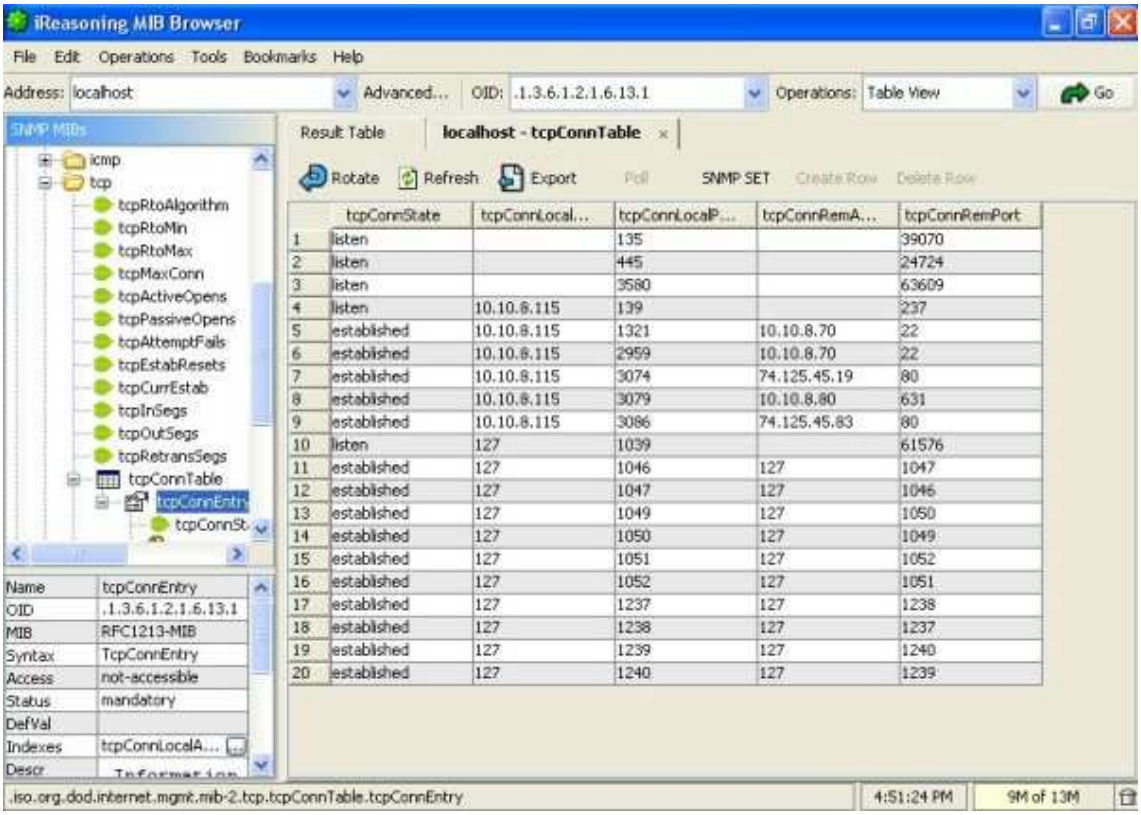

Figura 2 - Programa iReasoning MIB Browser

2.4. Registrando um nó particular

O IANA é o órgão responsável pela coordenação de vários recursos de internet como nomes de domínios, endereçamentos IP e protocolos. A MIB, fonte de consulta do protocolo SNMP, tem um ramo específico para fins particulares e que precisam ser registrados para evitar conflitos de registros no órgão. Cada proprietário desenvolve aplicações para seus equipamentos, sendo único sob o ramo “enterprises” da MIB, recebendo um *Private Enterprise Number*.

A solicitação é simples bastando preencher um formulário no endereço http://pen.iana.org/pen/PenApplication.page e respondendo aos e-mails mostrando interesse em adquirir o registro. As desistências são expressivas: ao fim do preenchimento do formulário, retorna-se uma previsão da PEN e o número efetivamente registrado é menor. Inicialmente meu PEN seria 31430 e

segundo o endereço http://www.iana.org/assignments/enterprise-numbers, foi estabelecido o número 30778. O tempo entre o envio dos dados pelo formulário e a confirmação do registro levou entre duas e três semanas.

2.5. Gerentes e Agentes

Como o SNMP é um protocolo de gerenciamento de redes, a comunicação pode abranger várias máquinas, conhecidas como gerentes e agentes. O gerente desempenha a tarefa chamada "*poll*" que é o ato de fazer consultas a todos os dispositivos que interpretam o SNMP, coletando informações da situação de funcionamento, estatísticas e informações relativas às redes. É possível ter mais de um gerente em uma rede, como ilustra a Figura 3. Caso a rede se expande por localizações geográficas distintas, cada LAN (*Local Area Network*) pode ter um gerente e as informações podem ser concentradas no gerente da WAN (*Wide Area Network*) [1].

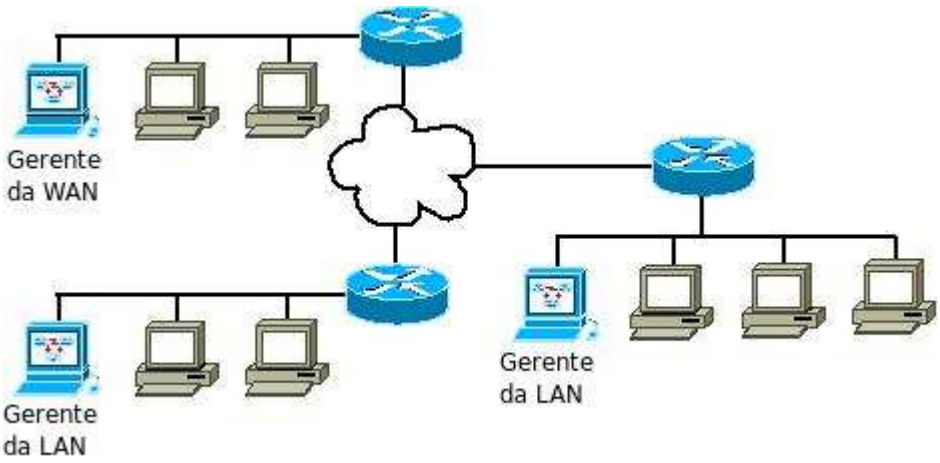

Figura 3 - Gerentes de WAN concentram informações de outras LANs

O agente pode ser um computador ou qualquer dispositivo de rede gerenciável que carrega uma ou mais MIBs buscando o objeto que é solicitado pelo gerente e devolve uma resposta. Outra capacidade é a de enviar "traps" que são mensagens assíncronas, ou seja, o agente envia dados sem esperar retorno

do gerente. Exemplificando, os "*traps*" avisam que uma placa de rede não está ativa ou que um novo equipamento foi ligado à rede. É possível configurar o agente para enviar múltiplas cópias de notificações para gerentes distintos. O gerente, por sua vez, concentra todas as mensagens oriundas dos agentes independentes estando na mesma rede ou não.

A rigor, chama-se de "*trap*" a mensagem que parte do agente quando utiliza-se a versão 1 (um). Seu propósito é relatar as condições de algo que esteja sendo monitorado, porém existe o problema de não se ter certeza se chegou ao destino. Buscando melhoramentos, as versões 2c e 3 substituíram o nome por "*notifications*" com o adicional de que o gerente confirma o recebimento para a máquina remetente.

2.6. Protocolos de Gerência de Rede

SGMP é o antecessor do SNMP e foi desenvolvido para gerenciar roteadores [18]. Ficou obsoleto sendo substituído pelo SNMP. É incompatível com SNMP, porém a notação ASN.1 foi herdada.

Nagios é um aplicativo com código-fonte aberto sob licença GPL (*GNU Public License*). Originalmente escrito para Linux e hoje existem versões para outros sistemas Unix e plugins para *Microsoft Windows*. É restrito a processos como serviços de SMTP (*Simple Mail Transport Protocol*), POP3 (*Post Office Protocol*), HTTP (*Hypertext Transfer Protocol*), porém não há documentação que trate de monitoramento de dispositivos.

2.7. O protocolo SNMP

Quando uma informação é armazenada em um computador remoto, o usuário abre uma conexão, faz uma autenticação geralmente através de uma combinação de nome e senha e, finalmente, o dado é obtido. A grande vantagem do protocolo SNMP é a simplicidade do processo para alcançar o mesmo resultado: um comando SNMP é enviado para consultar um agente remoto e imediatamente é devolvida a resposta.

Assim, uma das tarefas que o SNMP faz com êxito é de monitorar o estado da rede e seus equipamentos. Problemas podem acontecer fora do horário de trabalho e deve constantemente ser fácil avisar o administrador. Caso esteja indisponível, deve haver outra pessoa que saiba como reparar a falha do dispositivo: o responsável pode consultar remotamente o estado do equipamento. Adicionalmente, o protocolo pode detectar problemas no sistema e notificar o administrador através de e-mail ou SMS (*Short Message Service*) do celular. É possível obter a informação do estado do dispositivo como uma placa de rede de um computador. Em poucas palavras, o SNMP notifica sobre a ocorrência de algo errado do sistema.

O protocolo SNMP desempenha o gerenciamento de rede de forma otimizada através do endereçamento IP, permitindo gerenciar dispositivos remotamente. Para que este gerenciamento aconteça é preciso que o equipamento entenda o SNMP. Tal observação é pertinente, pois nem todos os fabricantes fazem uso deste protocolo em seus dispositivos.

Atualmente, pode ser usado para gerenciar sistemas Unix, Windows, impressoras, fornecedores de energia ininterrupta entre outros. São compostos por duas entidades: gerentes e agentes. Os gerentes são tecnicamente chamados de Estações de Gerenciamento de Rede (EGR) e são responsáveis por concentrar as informações oriundas dos dispositivos. O agente é um programa

que responde às solicitações vindas da EGR e notifica regularmente o estado de um objeto que se queira monitorar.

O software de gerência de rede permite que o gerente monitore e controle componentes de rede. É empregado para interrogar dispositivos como computadores, roteadores, comutadores, *bridges* entre outros determinando seu *status*, bem como obter estatísticas sobre das redes às quais fazem parte [3].

Consultar a documentação ou o representante do produto é uma forma de saber se o dispositivo é compatível com SNMP. Uma tentativa interessante é enviar um "*snmpwalk*" apontando para o dispositivo, esperando que devolva alguma informação. É provável que dispositivos domésticos não suportem SNMP. Hubs, comutadores e roteadores que interpretam o SNMP são identificados como "gerenciáveis" e custam significativamente mais caros.

O SNMP transporta mensagens entre gerentes e agentes através de pacotes UDP. Por padrão todos os dispositivos SNMP usam a porta 161 para enviar e receber informações e a porta 162 para receber avisos dos dispositivos gerenciados. É possível que as duas formas de comunicação aconteçam ao mesmo tempo e tal fato não interfere na integridade dos dados.

Existem três versões deste protocolo que é organizado pelo IETF:

- SNMP versão 1 (SNMPv1): É o padrão atual do SNMP. A segurança é baseada nos nomes de comunidades que funcionam como senhas. Os nomes são textos puros. Conforme a configuração, os nomes de comunidades podem ter acesso de somente-leitura ou de leitura e escrita.
- SNMP versão 2 (SNMPv2): Também chamada de SNMPv2c, é uma versão experimental e somente alguns fabricantes têm dispositivos que o suportam.
- SNMP versão 3 (SNMPv3): Suporta forte autenticação e comunicação privada entre as entidades gerenciadoras. Atualmente há uma RFC

(*Request for Comments*) proposta podendo chegar à condição de padrão segundo as normas do IETF.

Com relação à segurança, as documentações sobre a versão 1 do SNMP é dita insegura, porém, o real problema é a má administração. Para aumentar a segurança, alguns cuidados devem ser tomados como:

- Nome da comunidade. Por ser o equivalente a uma senha, o nome escolhido não pode ser elementar tanto para as comunidades de somente leitura como leitura e escrita. Por padrão, toda instalação tem a comunidade *public* que deve ser alterada antes de entrar em funcionamento. Uma leitura sobre escolha de senhas seguras é recomendada. Uma situação que não tem como evitar é a interceptação por *sniffers* – programas que capturam informações numa rede.
- Número da porta. Ao invés de trabalhar com o número da porta padrão, a 161 para *snmpd* e 162 para *snmptrapd*, recomenda-se qualquer outra porta que o sistema não esteja usando. Existe uma grande faixa de portas possíveis dificultando a descoberta da porta ativa.
- Nome do host ou IP. Para obter alguma resposta baseada no protocolo SNMP, o usuário mal-intencionado precisa saber quais máquinas são os agentes. Recomenda-se o entendimento e aplicações de regras de acesso de programas como *iptables* – software que possui um conjunto de regras para tratar entradas e saída de dados.
- Versão. Apesar de poucas opções, apenas 1, 2c e 3, o usuário só obtém a resposta se souber a versão do protocolo.

Para obter informações específicas além dos objetos localizados nas MIBs padrões, a tentativa requer conhecimento de OIDs privadas. Nesta dissertação, por exemplo, um usuário não autorizado precisa saber sobre o ramo murilofujita, cujo OID é 1.3.6.1.4.1.30778.

Com relação à carga de processamento, o protocolo foi testado em um Pentium I 133 MHz com 48 MB de memória RAM e, mesmo sendo uma

configuração modesta, obtiveram-se respostas dentro de três segundos. Como conseqüência, este protocolo pode ser aplicado em sistemas *embedded* (embarcados) diminuindo custos por não precisar de equipamentos de alto poder de processamento.

2.8. Comunicação do protocolo SNMP

A Figura 4 mostra a comunicação do protocolo SNMP através do modelo em camadas. Quando o gerente faz uma consulta, o SNMP na camada de aplicação cria um *socket* UDP na camada de transporte e especifica o destino através do IP do agente na camada de rede. A partir deste ponto, o gerente reuniu todas as informações para enviar o *poll* e, ao chegar ao destino, o agente desfaz os cabeçalhos adicionados em cada camada recebe a informação exatamente como foi enviada pelo computador remetente. O processo de *poll* utiliza a porta 161 para realizar a comunicação entre os processos agente/gerente. No *trap*, o agente utiliza a porta 162 [1].

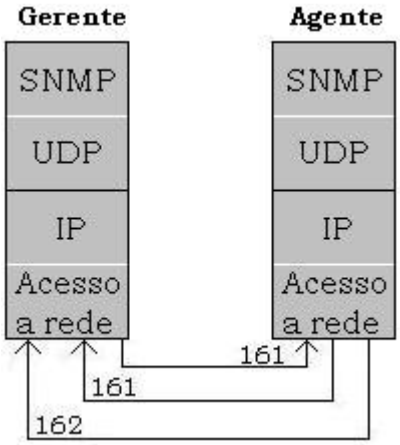

Figura 4 - Comunicação poll-trap

Um agente e/ou gerente pode abrir um *socket* UDP e enviar para destinos diferentes assim como um *socket* UDP criado pelo agente e/ou gerente pode receber dados de vários remetentes.

O IP desejado pode alcançar redes localizadas em pontos distantes. A Figura 5 mostra que o SNMP pode ser enviado através da WAN pelo uso de roteadores.

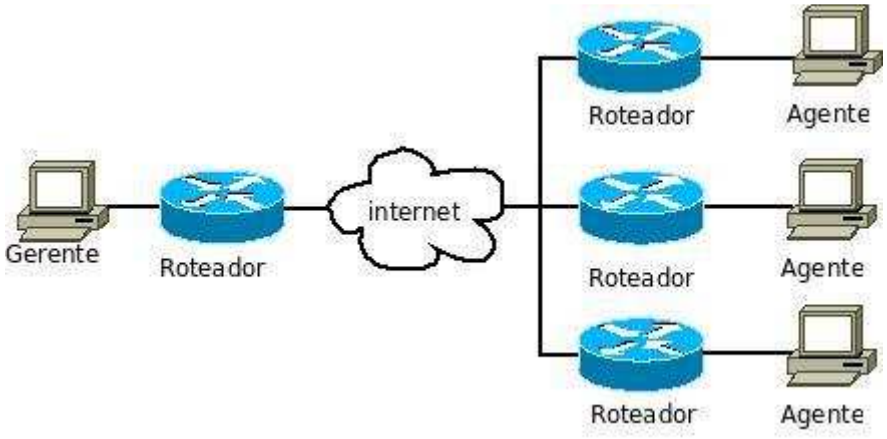

Figura 5 - Comunicação entre gerente e agentes

2.9. Instalação do SNMP no gerente

No gerente é instalado o sistema operacional Linux Ubuntu versão 8.10 com o kernel 2.6.24-21, um software livre que dá o direito ao usuário de copiar, modificar e distribuir conforme sua escolha. A forma de obter e instalar os softwares é igual para outras distribuições, com exceção do endereço dos repositórios.

A opção a seguir é feita para máquinas embarcadas, pois estas não têm recursos como tela gráfica, sendo contornadas com linhas de comando. Desta forma, a instalação pode atender desde máquinas de modesto desempenho até máquinas com maior poder de processamento.

Para que o protocolo SNMP seja instalado, o usuário que estiver autenticado deve ter permissão para instalar o programa, ou seja, ser o usuário administrador. Para o procedimento de instalação é preciso que o arquivo */etc/apt/source.list* aponte para os repositórios de programas e esteja configurado com as seguintes linhas:

```
deb http://br.archive.ubuntu.com/ubuntu/ hardy-updates main
restricted
deb-src http://br.archive.ubuntu.com/ubuntu/ hardy-updates
main restricted
deb http://br.archive.ubuntu.com/ubuntu/ hardy universe
deb http://br.archive.ubuntu.com/ubuntu/ hardy-backports main
restricted universe multiverse
```

Após salvar as modificações e sair, deve-se atualizar a listagem de pacotes com o comando:

```
murilo@fujita2:~$ apt-get update
```

O protocolo SNMP é instalado pelo software Net-SNMP através do comando:

```
murilo@fujita2:~$ sudo apt-get install snmp

Selecionando pacote previamente não selecionado snmp.
(Lendo banco de dados ... 115655 arquivos e diretórios
atualmente instalados.)
Descompactando snmp (de .../snmp_5.4.1~dfsg-4ubuntu4_i386.deb)
...
Instalando snmp (5.4.1~dfsg-4ubuntu4) ...
```

Para confirmar se o software está corretamente instalado o comando utilziado é:

```
murilo@fujita2:~$ dpkg –l snmp
```

como resposta:

```
Desired=Unknown/Install/Remove/Purge/Hold
|Status=Not/Installed/Config-files/Unpacked/Failed-
config/Half-installed
|/Err?=(none)/Hold/Reinst-required/X=both-problems(Status,Err:
uppercase=bad)
||/ Name   Version             Description
+++-==========================-================
```

```
ii  snmp  5.4.1~dfsg-4ubuntu4     NET  SNMP  (Simple  Network
Management Protocol) Apps
```

Indicando que o protocolo está presente no sistema. Já a resposta:

```
Desired=Unknown/Install/Remove/Purge/Hold
|Status=Not/Installed/Config-files/Unpacked/Failed-
config/Half-installed
|/Err?=(none)/Hold/Reinst-required/X=both-problems
(Status,Err: uppercase=bad)
||/ Name   Version             Description
+++-======-===========-=======================
pn  snmp   <none>     (no description available)
```

Indica que o SNMP não está instalado por causa da palavra “*none*”.

Para verificar se a tarefa que é executada no gerente foi inicializada, é necessário abrir um terminal e introduzir o comando *ps* que lista os processos. O sinal *pipe* (|) indica que a saída do comando `ps` é direcionada para a entrada do `grep`, um filtro que exibe a sequência de caracteres procurada (no caso “snmp”) e imprime as ocorrências encontradas:

```
murilo@fujita2:~$ ps -ef | grep snmp
```

Observa-se os processos ativos, como no exemplo a seguir:

```
 root  1599      1  0 Jul03 ?     00:00:00 /usr/sbin/snmptrapd
-Lf /var/log/snmptrapd.log
 root  4837  4829  0 11:40 pts/0  00:00:00 grep snmp
```

A resposta mostra o caminho completo do *daemon* (tarefas que executam em segundo plano) *snmptrapd* indicando que está ativo e direcionando os *logs* para o arquivo /var/log/snmptrapd.log. A inicialização deste *daemon* é mais demorada porque este checa algumas operações como listagem de interfaces, tornando-se ativo somente depois de completar algumas tarefas.

2.10. Instalação do SNMP no agente

Há muitos benefícios em executar o agente Net-SNMP em vez do da *Microsoft*. O agente *Microsoft* é uma aplicação estática com muitas limitações que não permite variedade de opções de configuração como estender o agente; recurso que amplia a capacidade de execução de tarefas. Por exemplo, não é possível implementar as funções de supervisório.

O software *net-snmp-5.4.1-3.win32.exe*, uma versão própria para o ambiente Windows foi escolhido para o agente, podendo ser obtido pelo endereço: h*ttp://sourceforge.net/project/showfiles.php?group_id=12694&package_id=162885* (consultado em 10 de março de 2009).

Iniciada a execução, a única modificação em relação às opções apresentadas na instalação é vista na Figura 6:

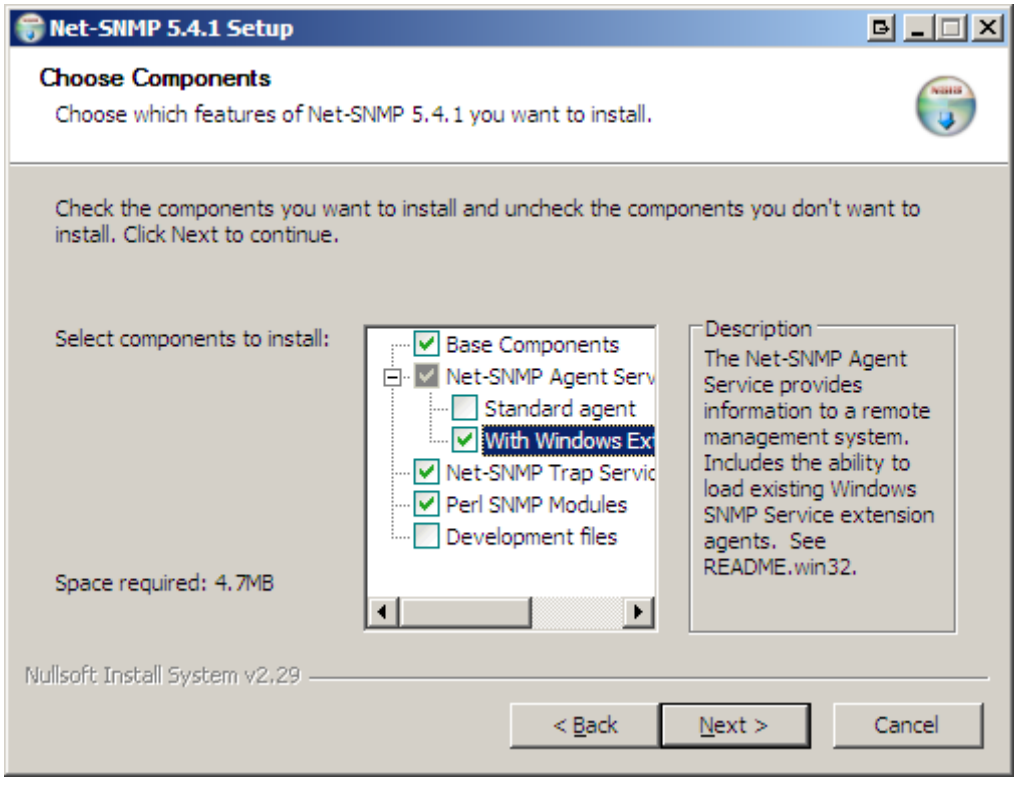

Figura 6 - Instalação do Net-SNMP para *Windows*

No ramo net-SNMP Agent Services, deve-se desmarcar a opção “Standard agent” que é sugerida na instalação e marcar a opção em destaque “With Windows Extension”. Esta opção torna possível a aplicação Net-SNMP e o agente *Microsoft* SNMP na mesma máquina, desde que os serviços sejam configurados para utilizar sockets distintos.

Terminadas a instalação, devem-se ativar os serviços de *poll* da seguinte forma:

Iniciar -> Net-SNMP -> Service -> Register Agent Service (ou digitando no DOS o comando “`C:\usr>registeragent.bat`”)

Este arquivo de lote (extensão bat) tem instruções para tornar o Net-SNMP um serviço (Figura 7) que pode ser controlado pelas ferramentas administrativas do *Windows*. Desta forma é possível controlar a inicialização do serviço, programar quais ações a serem tomadas no caso de falha, entre outras.

Para os serviços de *poll*, a instrução passada é:

```
snmpd.exe      -I-udp,udpTable,tcp,tcpTable,icmp,ip,interfaces,
system_mib,sysORTable,vacm_conf,proxy,pass,pass_persist     -Lf
C:\usr\log\snmpd.log
```

sendo que o parâmetro –I indica quais módulos das MIBs são carregadas e –Lf localiza o arquivo de registro dos eventos snmpd.

2.11. Configuração do agente

Toda vez que o *daemon* é iniciado, este procura os arquivos C:\usr\etc\snmp\ snmpd.conf. Neste é configurado os requisitos de segurança, extensão do agente, armazenagem de dados, tarefas como checagem de processos e carga da rede. Para iniciar o *daemon* Net-SNMP use o comando:

```
C:>net start “net-snmp agent”
```

e para pará-lo, use:

```
C:>net stop ”net-snmp agent”
```

Uma outra forma de controlar a atividade do *daemon* snmpd utilizando recursos gráficos é acessando o Painel de Controle, ferramentas administrativas, Serviços e é visualizada a tela como mostra a Figura 7.

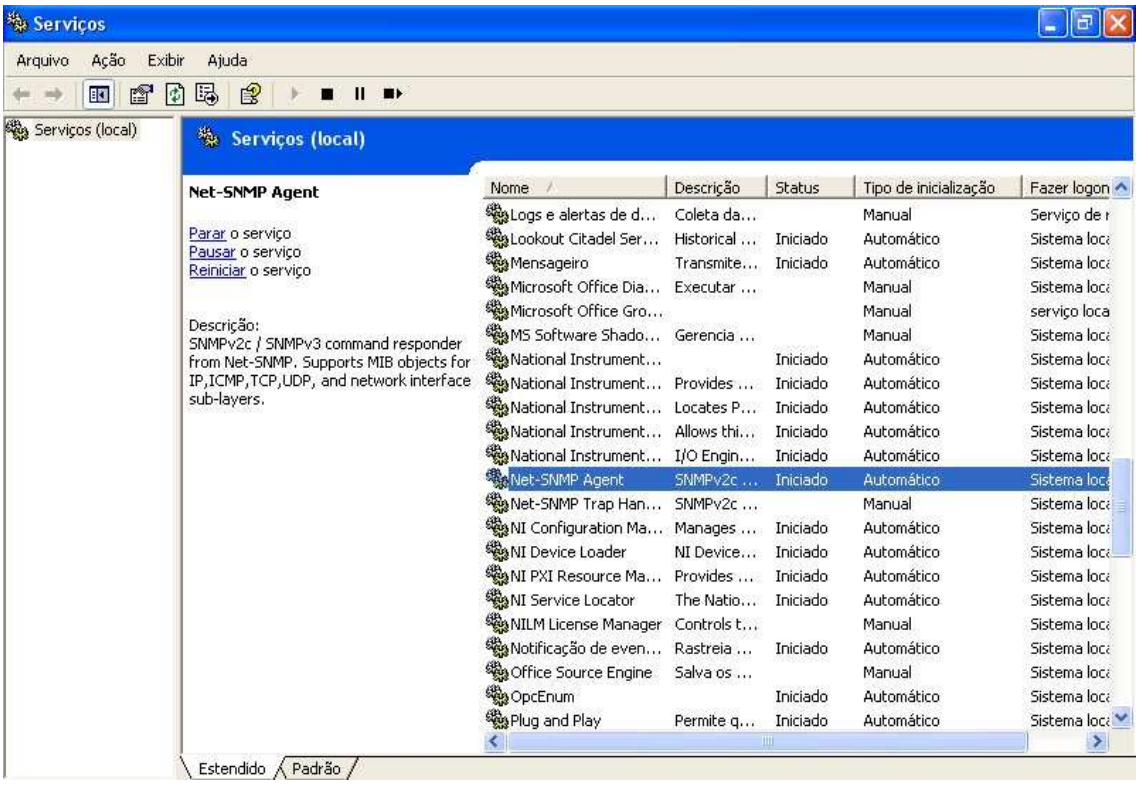

Figura 7 - Serviços Microsoft de controle de tarefas

Observa-se que a linha destacada indica “Net-SNMP Agent” que é encontrada quando instalado o Net-SNMP. Para que a máquina inicie automaticamente este serviço deve-se indicar “Automático” em “Tipo de

inicialização”. A cada modificação no arquivo que controla este *daemon*, este deve ser renicializado para que as mudanças entrem em vigor.

É possível que ambos os serviços coexistam e para isso deve evitar conflitos entre as atividades da *Microsoft* e do Net-SNMP fazendo com que os *daemons* escutem portas diferentes. Desta forma, estando ambos ativos, a forma de diferenciar qual serviço recebe a mensagem é através de uma porta distinta.

A porta Net-SNMP para *snmpd* pode ser modificada através do arquivo snmpd.conf ou usando uma opção na linha de comando que ignora a configuração do arquivo. O serviço da *Microsoft* usa a entrada *snmp* no arquivo SERVICES (%SystemRoot%\system32\drivers\etc\services) para determinar a porta que o serviço fica escutando. Simplesmente modifique as entradas e reinicie o serviço SNMP.

Depois de iniciados os serviços, é preciso conferir se estes estão ativos em cada máquina. O *daemon* snmpd do agente pode ser conferido pela Figura 7 através da coluna “status” que aparecerá a palavra “Iniciado” ou um vazio indicando, respectivamente, existência de atividade ou inatividade.

2.12. Executando o SNMP pela linha de comando

Cada comando SNMP tem uma sintaxe que necessita de certos parâmetros para funcionar. A *“man Page”*, documentação que acompanha os softwares de qualquer distribuição Linux, instrui a sintaxe do comando a seguir:

```
snmpget [COMMON OPTIONS] [-Cf] OID [OID]...
```

Exemplificando:

```
murilo@fujita2:~$ snmpget -v 1 -c fujitaro fujita
.1.3.6.1.2.1.1.1.0
```

```
SNMPv2-MIB::sysDescr.0 = STRING: Hardware: x86 Family 6 Model
8 Stepping 0 AT/AT COMPATIBLE - Software: Windows 2000 Version
5.1 (Build 2600 Uniprocessor Free)
```

Para este caso não foram usadas múltiplas OIDs, apenas a descrição do sistema indicada pela instância *sysDescr* como aparece na resposta. A opção "-v" indica a versão do; a opção "-c" é o nome da comunidade; a palavra "fujita" aponta a máquina que é consultada, podendo ser substituída pelo IP. Para que este comando recupere o valor procurado, um indexador 0 (zero) deve acompanhar a OID, porque se trata de uma resposta de uma única instância. A ausência do índice retorna uma mensagem de que não há resposta da máquina consultada.

Como existe uma forma mais agradável do que guardar sequências de números, há um comando que faz a tradução da forma numérica para textual, facilitando a compreensão. Sua sintaxe é:

```
snmptranslate [OPTIONS] OID [OID]...
```

Não é necessário informar a versão e nome da comunidade, pois o equivalente numérico e textual são os mesmos para qualquer versão. Exemplificando:

```
murilo@fujita2:~$ snmptranslate -Of .1.3.6.1.2.1.1.1
.iso.org.dod.internet.mgmt.mib-2.system.sysDescr
```

Ou na forma abreviada:

```
murilo@fujita2:~$ snmptranslate -Ofs  .1.3.6.1.2.1.1.1
sysDescr
```

O mesmo resultado é obtido usando-se a forma numérica ou textual para o comando *snmpget* e outras variações.

2.13. Traps

Traps são notificações que partem do agente avisando sobre o estado do equipamento ou do processo no caso de monitoramento constante. O *daemon snmptrapd* por padrão utiliza a porta 162 e deve ficar ativo no gerente armazenando as informações oriundas dos agentes. A Figura 4 mostra a situação descrita.

O arquivo /usr/local/share/snmp/snmptrapd.conf do gerente configura o *daemon* snmptrapd de acordo com as diretivas: controle de acesso, formato das mensagens e notificações de processamento.

O controle de acesso autoriza quais agentes podem enviar as traps. A seleção de agentes pode ser de acordo com o nome da comunidade, usuário, grupo ou até mesmo desabilitar a verificação de quais máquinas estão enviando as traps. Porém, segurança é sempre uma preocupação e não deve ser desativada, pois ao perceber mensagens que não foram reconhecidas de acordo com a configuração do sistema, possivelmente alguém não autorizado está tentando consultar dispositivos sem saber o nome da comunidade. Esta situação gera mensagens com avisos de falha de autenticação sendo possível identificar a origem.

Todos os eventos são registrados e configurados de maneira personalizada. O formato das mensagens diz respeito, por exemplo, à forma como as ocorrências são registradas e em qual arquivo devem ser armazenadas. A diretiva *format* funciona da mesma forma que os formatadores da linguagem C. Na Tabela 1 são encontrados alguns exemplos para configurar os logs:

Tabela 1 - Formatadores para personalizar as mensagens de traps

Formatador	Descrição
%A	Nome do *host*
%h	Hora da máquina local
%j	Minuto da máquina local
%P	Informa qual versão usada na consulta
%W	Descrição da *trap*

As notificações de processamento tratam da ação a ser tomada ao chegar uma trap. Uma diretiva chamada *traphandle* é ativada quando chega uma trap específica fazendo disparar um programa ajustado para realizar uma ação. Por exemplo, ao detectar uma determinada falha em um dispositivo, um trap é enviado para o gerente que faz o *traphandle* invocar um programa que envie e-mails para um certo endereço notificando sobre a situação. A demonstração da aplicação encontra-se no item 3.1.3. Habilitação da porta de comunicação 162.

A notificação feita por e-mail tem restrição de envio pois é checada de acordo com as regras do projeto Spamhaus. De acordo com a Figura 8, se a origem for um IP dinâmico, provavelmente residencial, os e-mails são descartados. Sendo de um endereço estático, os e-mails são considerados mais confiáveis e ao invés de serem separados como *spams*, são aceitos como *e-mails* válidos.

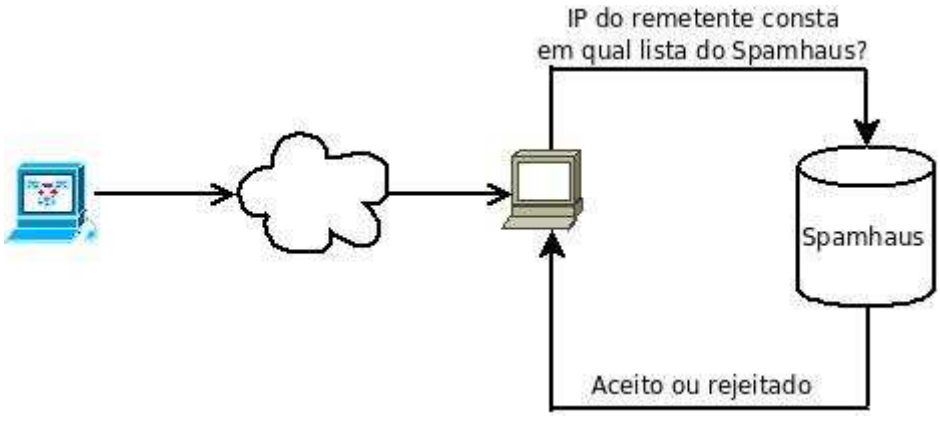

Figura 8 - Processo de verificação de e-mails para entrega

Para que as informações sejam coletadas, o gerente checa o nome da comunidade enviado pelo agente para autenticação. O item 3.1.3 exemplifica a instrução do arquivo para enviar e-mails e configurar outras funções através da instrução “*traphandle”*.

Outra forma de configurar o sistema é executar um aplicativo que cria arquivos de configuração SNMP de acordo com as respostas dadas. O questionário abrange opções de autenticação, formatação e opções de logs, comportamento do *daemon* e ação a ser tomada ao ocorrer um evento. Para execução do aplicativo deve estar instalado o pacote “Perl”, uma linguagem de programação portável para várias plataformas e seu código-fonte é aberto. A instalação é obtida através do comando:

```
murilo@fujita2:~$ apt-get install perl
```

Para iniciar a configuração, deve-se usar comando:

```
murilo@fujita2:~$ snmpconf –i
```

sendo que a opção “-i” informa que ao finalizar o aplicativo, os arquivos são copiados para o diretório em que o *daemon* busca os arquivos de configuração.

2.14. Sincronização

Uma preocupação que deve existir em qualquer parque computacional com troca de mensagens é a sincronização de relógio de cada elemento da rede. É importante para verificação dos eventos, a disposição em ordem cronológica para garantir a fidelidade da auditoria. Por exemplo, um problema pode ser precisamente determinado quando começou, mesmo se o dado trafegou por

várias máquinas intermediárias. Caso haja uma ação mal-intencionada, é possível saber há quanto tempo o invasor está tentando ganhar acesso.

Neste trabalho foi preciso lidar com a heterogeneidade de sistemas operacionais e buscar programas para este propósito atendendo cada um dos sistemas. Para o *Windows* é usado o *d4time50.msi* cuja interface é mostrada na Figura 9.

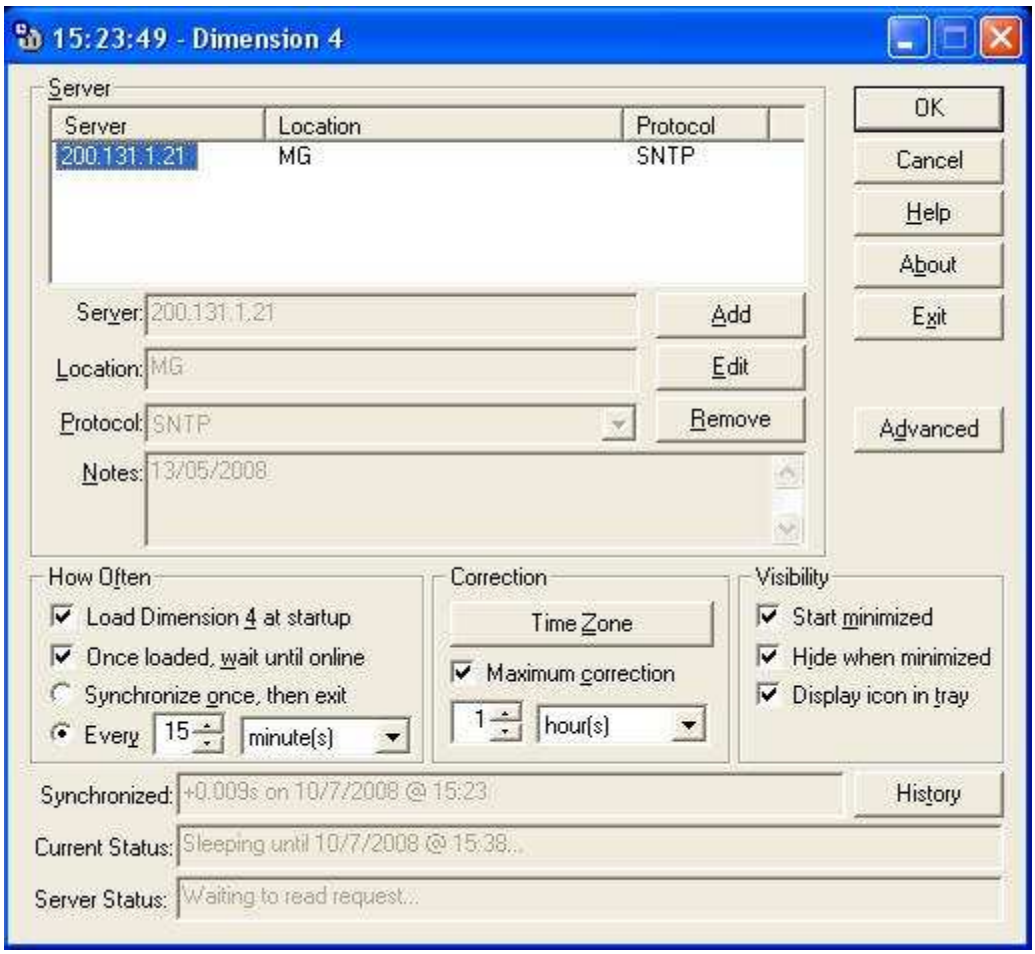

Figura 9 - Interface do programa d4time50.msi

É preciso remover os endereços dos servidores de hora sugeridos pelo programa e adicionar um que esteja no mesmo fuso horário. Como pode ser visto na Figura 9, o endereço do servidor é um IP (200.131.1.21) que pertence a RNP (Rede Nacional de Pesquisas), escolhido pela proximidade (MG).

Para o Linux, foi instalado o pacote ntpdate através do comando

```
murilo@fujita2:~$ apt-get install ntpdate
```

Tanto a instalação, como a execução do aplicativo é uma operação exclusiva do usuário administrador ou o sistema nega a instrução. Uma vez instalado o aplicativo, o comando que faz a sincronização entre o servidor da RNP e a máquina solicitante é:

```
murilo@fujita2:~$ ntpdate 200.131.1.21
```

Por exemplo, o crontab pode ser usado para realizar o ajuste temporal periodicamente. Ao entrar com uma sequência de números seguida da instrução a realizar em qualquer editor de texto para que este agendador de tarefas executa o evento.

2.15. Ferramentas necessárias

2.15.1. Cygwin

Cygwin é uma coleção de programas portada do padrão POSIX (*nix e *BSD) para serem executados no Windows através da biblioteca cygwin1.dll que atua como uma emulação API (*Application Programming Interface*) Linux fornecendo as funcionalidades do Linux. Esta biblioteca funciona com todas as versões comerciais recentes do *Windows* em 32 e 64 *bits* tendo como exceção o Windows CE. Está sob a licença GPL, sendo um software livre dando a liberdade de ser executado, copiado, distribuído, estudado e modificado para atender as necessidades do usuário.

As aplicações que são executadas em Linux não funcionam no ambiente cygwin e vice-versa. É necessário recompilar o código-fonte no ambiente que se deseja utilizar para que seja possível executá-lo.

O software pode ser obtido pelo endereço *http://cygwin.com/setup.exe*, [consultado em 20 de maio de 2008] e pode ser instalado em qualquer unidade, desde que o nome do diretório seja o sugerido na instalação que é cygwin.

O propósito deste software é aumentar a quantidade de recursos do sistema trazendo as funcionalidades do Linux. Desta forma é possível interagir com o net-snmp para criar pequenos programas e utilizar outros comandos no *Windows* para serem tratados pelo protocolo SNMP.

2.15.2. Visual Studio C++

Para que a placa de aquisição de dados da *National Instruments* capture valores analógicos ou digitais, é necessário um programa que realize a comunicação entre o computador e o dispositivo.

O fabricante disponibiliza vários códigos-fonte da linguagem C para diversas tarefas. O segundo arquivo é um cabeçalho que pode ser modificado. Este arquivo contém um conjunto de definições, comportamento e características das variáveis. O código-fonte contém as chamadas de funções para iniciar ou parar ou leitura de dados. Modificações foram feitas de forma que o binário direcione a saída para um arquivo com a frequência desejada atualizando a MIB. O terceiro arquivo necessário é uma biblioteca sem acesso ao código-fonte.

Reunido os três arquivos, é preciso um ambiente para compilação que faça a ligação entre eles e gere o executável para o modelo USB 6008. Esta finalidade é cumprida pelo Visual Studio C++.

2.15.3. Wake-On-LAN

Atualmente os computadores possuem fontes ATX (*Advanced Technology Extended*), ou seja, para ligá-los, um contato do tipo “normalmente

aberto” ao ser pressionado fecha o circuito ligando a máquina. Quando acontece o restabelecimento de energia, o computador mantém-se desligado.

Como este trabalho trata de gerenciamento remoto, o Wake-On-LAN é uma ferramenta com o propósito de resolver a situação descrita. Este pacote é instalado no gerente dando a capacidade de ligar o agente e continuar com as tarefas que estavam sendo realizadas antes da queda de energia.

O gerente envia um comando que é recebido pela placa de rede do agente, conectada à placa-mãe e encarregada de fechar o circuito. Nem todas as placas de rede suportam este recurso, tendo um custo adicional em relação às comuns. No presente momento a grande dos computadores vem equipados com ATX, salvo alguns modelos portáteis. A ferramenta desempenha a mesma função ao apertar o botão, portanto, é necessário que o agente esteja ligado à rede elétrica. Alguns problemas podem acontecer e para tal é recomendado checar as seguintes possibilidades: dependendo do tipo ou configuração do *switch*, pode acontecer o descarte desse de tipo de dado ou não estar instalada uma linguagem que interprete o acionamento como *perl* ou *python*.

Existem outros pacotes da plataforma Linux, porém a vantagem deste é a não exigência de privilégios de administrador para executar a ferramenta. Para direcionar o comando para uma máquina específica, deve-se ter de antemão o endereço físico ou “MAC address”. O Wake-On-LAN envia pacotes que consistem de um cabeçalho ethernet, um cabeçalho IP, um cabeçalho UDP, uma sequência com conteúdo FF, seguido do endereço físico e este processo é repetido dezesseis vezes.

2.15.4. Apache

Toda vez que uma máquina cliente insere um endereço no seu navegador, o servidor correspondente responde enviando páginas escritas em linguagem

HTML (*Hypertext Markup Language*) e suas variantes, além de objetos multimídia através do protocolo HTTP. Os documentos feitos nesta linguagem são exibidos obedecendo a uma formatação conforme as sintaxes para tamanhos e cores das fontes, alinhamentos, *frames* e outras marcações. Existe uma grande variedade de softwares que lidam com o protocolo, desde os de código aberto sob diversas licenças até os proprietários com diferentes faixas de preços.

O mais conhecido e usado é o Apache, um projeto feito com o esforço de voluntários tanto na parte de software como documentação. Como todos os demais softwares livres, pode ser instalado e modificado da melhor forma que atenda uma exigência particular.

Há uma diferença considerável sobre a forma como cada sistema operacional executa o Apache. O Unix e derivados criam uma cópia exata do processo-pai para atender todas as solicitações. Como a plataforma *Windows* não funciona desta forma, a solução foi reescrever o código para usar *threads* que compartilham recursos. A prática torna possível executar com uma quantia menor de memória, porém não tão eficiente como o sistemas Unix [7].

O Apache também é um *daemon* e por padrão, a comunicação usa a porta 80 através de TCP. Entre suas várias características, pode-se citar controle de acesso para determinados conteúdos, tráfego de dados encriptados, possuir vários domínios mesmo tendo um único IP e funcionalidades extras, assunto abordado no próximo item.

2.15.5. CSS

CSS (*Cascading Style Sheets* ou Folha de Estilos em Cascata) é uma ferramenta que complementa a linguagem HTML ajustando fontes, cores, margens, inserindo imagens de fundo e posicionando figuras. O CSS delimita blocos de instruções da mesma forma que a linguagem HTML e novas palavras-

chaves foram surgindo para formatar os elementos da página buscando atender as necessidades dos projetistas.

O cabeçalho do documento HTML deve conter informações como a localização do arquivo CSS, para que ao longo do texto as palavras-chaves instruam a forma como CSS deve estabelecer a posição dos elementos da página.

2.15.6. CGI-bin

Alguns programadores chamam de CGI-bin por causa do projeto original o qual colocou todos os programas CGI (*Common Gateway Interface*) em um diretório chamado bin. A função do CGI é executar pequenos programas feitos em linguagens como Perl, PHP, C entre outras produzindo páginas dinâmicas.

Para abstrair estes conceitos, a Figura 10 divide o sistema operacional Unix em camadas na qual pode ser visto que a camada mais interna é o *hardware* composto por dispositivos e *drivers*. Ao seu redor encontra-se o kernel que gerencia e controla a camada inferior, fazendo comunicação com a camada de programas e comandos, que realizam as tarefas. Finalmente, na camada mais externa encontra-se o *Shell*, o responsável pela interação entre usuário e o sistema operacional.

Figura 10 - Camadas do sistema operacional Unix

Sempre que o Shell executa uma linha de comando, a tarefa recebe um PID (*Process Identification*), ou seja, um número que identifica o processo. O *shell* entrega para o kernel realizá-la, torna-se inativo e no momento que o *prompt* reaparece, a tarefa foi cumprida indicando estar pronto para as próximas execuções [5].

Para que *scripts* funcionem, é necessário um cabeçalho para que o interpretador reconheça em qual linguagem foi codificado. A Tabela 2 mostra alguns exemplos de cabeçalhos de arquivos para executar *scripts*.

Tabela 2- Cabeçalhos para invocar interpretadores de comandos

Cabeçalho	Linguagem
#!/bin/bash	Bash
#!/usr/bin/Perl	Perl
#!/usr/bin/python	Python

Nesta dissertação, o Apache é responsável pela interface que é visualizada por um navegador. O objetivo é manipular linhas de comando através dos eventos exibidos pela interface tornando transparente para quem for operá-lo. Para que o usuário interaja com o sistema, o Apache precisa ativar um dos seus módulos, o CGI, que executa *scripts* no gerente interrogando o agente que responde consultando os dados da MIB. A resposta é formatada pelo Shell e entrega a página atualizada (Figura 11). Os *scripts* são menos robustos do que um programa compilado porque é uma linguagem interpretada e realizam instruções em série. São executados por vários interpretadores como Bash (*Bourne Again Shell*), Korn Shell (versão melhorada do Bash usada em Unixes comerciais), pdksh (Korn Shell de domínio público), C Shell entre outros. A origem destes interpretadores veio da necessidade dos projetistas precisarem de portabilidade e velocidade adicionando funcionalidades a cada versão, como decisões simples, tarefas que executam *loops*, operações matemáticas,

manipulação de texto, formatação de uma resposta de um comando e modificação de imagens. Outra capacidade é manipular metacaracteres (*, ? , []) que substitui um ou mais caracteres por possíveis valores, recurso conhecido como Expressões Regulares [5].

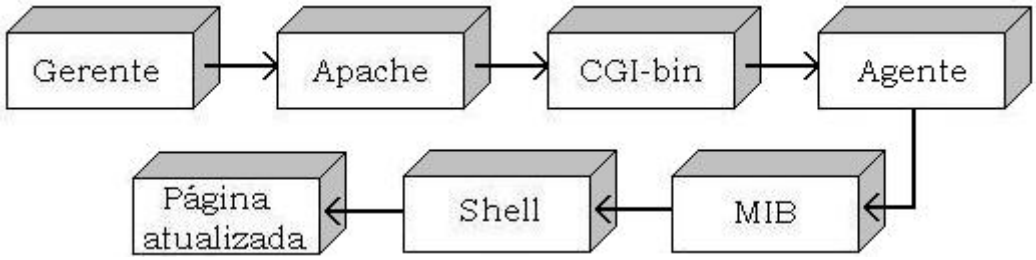

Figura 11 - Processo de comunicação através da interface gráfica

O padrão CGI foi feito originalmente para funcionar com sistemas operacionais Unix e Windows sendo que as informações são colocadas em variáveis que seguem o padrão Unix, como mostra a Tabela 3.

Tabela 3 - Variáveis de ambiente ao executar o CGI

Nome da variável	Significado
SERVER_NAME	O nome de domínio do computador que tem o papel de servidor.
REMOTE_ADDR	O endereço IP do computador solicitante
SCRIPT_NAME	O nome e o caminho do *script* executado
QUERY_STRING	Armazena informações do formulário após o sinal “?” na URL.

Todo formulário enviado é transformado em uma URL (*Uniform Resource Locator*), um endereço que especifica onde buscar o documento solicitado. Ao receber a requisição, o gerente divide a URL em duas partes que são separadas por um ponto de interrogação (?): um prefixo que especifica um

documento particular e um sufixo que contém informações adicionais armazenadas na variável QUERY_STRING. Os caracteres não imprimíveis são exibidos pelos sinais de porcentagem (%) seguidos de dois dígitos hexadecimais equivalentes e, em alguns casos, o caractere espaço é substituído por sinal de adição (+) [7].

2.15.7. Ipcheck

Para estabelecer o enlace de comunicação entre gerente e agente, é necessário saber o endereço da máquina remota. A questão é que muitas vezes os serviços contratados atribuem um IP.

Para lidar com esta situação, um programa adequado é o *ipcheck*. Escrito em *Python*, é uma linguagem de código aberto própria para trabalhar com domínios, é gratuita e portável para outras linguagens como C e C++.

Dispositivos distribuidores de internet como alguns modelos de roteadores domésticos possuem o recurso de atualizar o IP do local. Outra solução é instalar o programa em outro computador que permaneça ligado para capturar o novo endereço. A freqüência que o comando é invocado para atualizar pode ter como critério o intervalo de tempo que o IP é modificado. O novo IP é remetido para um serviço conhecido como DynDNS que é o responsável por associar o nome de domínio com o IP atualizado (representado na Figura 12). Nesta figura: (1) Roteador informa para o programa ipcheck o novo IP. (2) A informação é enviada para o DynDNS.

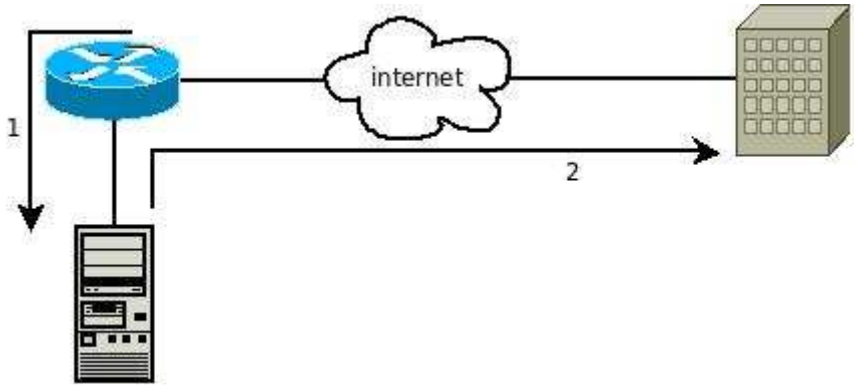

Figura 12 - Processo de localização do IP

3. Arquitetura de Integração do Gerenciamento de Rede e Supervisão de Processos

A Figura 13 ilustra uma aplicação prática do protocolo empregando as ferramentas de gerenciamento de rede e supervisório. O gerente estabelece uma comunicação com a agente para obter uma informação do equipamento. O agente por sua vez comunica com os equipamentos. Os equipamentos podem ser os mais diversos possíveis. Pode ser uma placa de aquisição de dados e neste caso existe um aplicativo específico para comunicação com o agente. A proposta é que esta implementação seja modular e independente da arquitetura gerente/agente, obtendo-se assim uma flexibilidade de uso do protocolo em qualquer equipamento. Em algumas aplicações de supervisório pode ser que o equipamento conectado, por exemplo, um Controlador Lógico Programável tenha placas de rede Ethernet. Neste caso o mesmo poderia ser conectado diretamente no roteador.

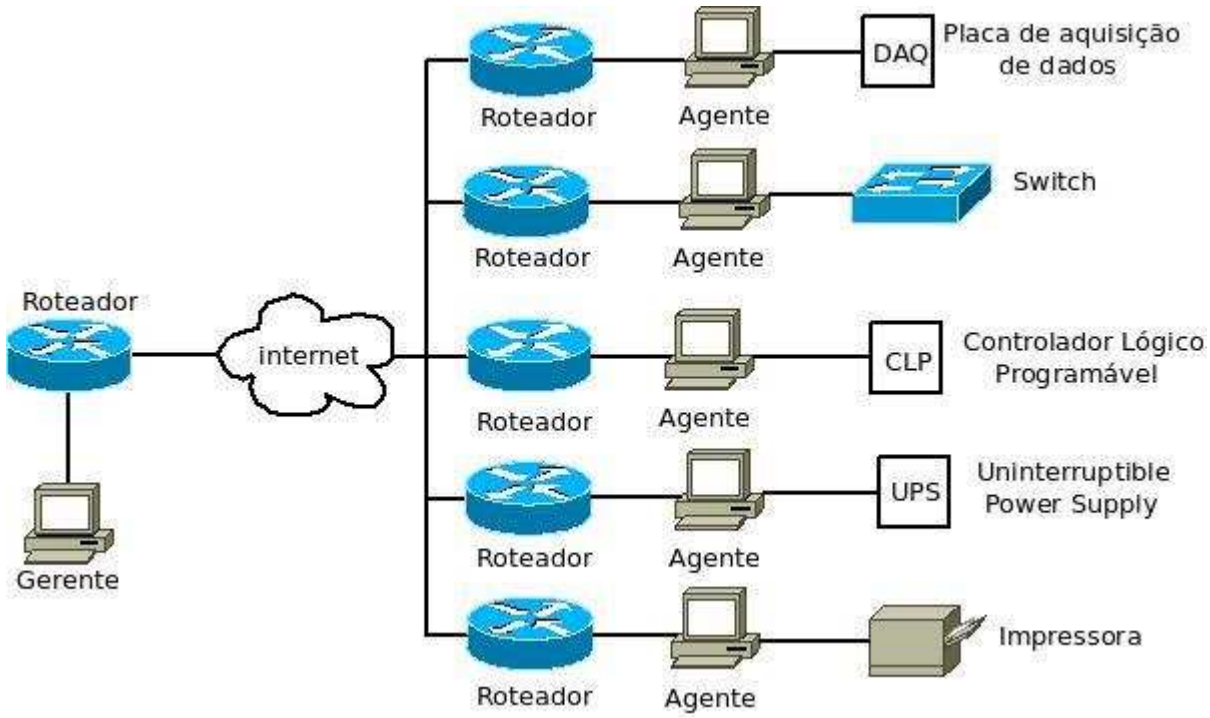

Figura 13 - Gerente aquisitando informação de um dispositivo através do agente

3.1. Configuração do gerente e agente

Na descrição a seguir encontram-se configurações da máquina local e remotas.

3.1.1. Habilitação da porta de comunicação 161

Manter a segurança das máquinas da rede é uma preocupação constante. Cada sistema operacional tem regras que tratam de cada situação, seja para lidar com as solicitações que chegam ou executar um programa localmente.

Por padrão, a instalação do Windows nega acesso a todas as conexões direcionadas à máquina, incluindo os pacotes SNMP. Uma forma de estabelecer qualquer tipo de comunicação é desativar o *firewall*, porém é uma medida desaconselhável, pois o torna vulnerável a ataques. Para que a segurança seja garantida, o melhor a ser feito é criar regras com permissão específica para conexões com propósito conhecido.

A Figura 14 mostra a caixa de diálogo pronta para permitir o tráfego de informações SNMP. Esta é encontrada em Painel de Controle, *Windows Firewall*, aba Exceção. Para adicionar o SNMP, deve-se clicar em “Adicionar Porta” surgindo outra caixa de diálogo como mostra a Figura 15.

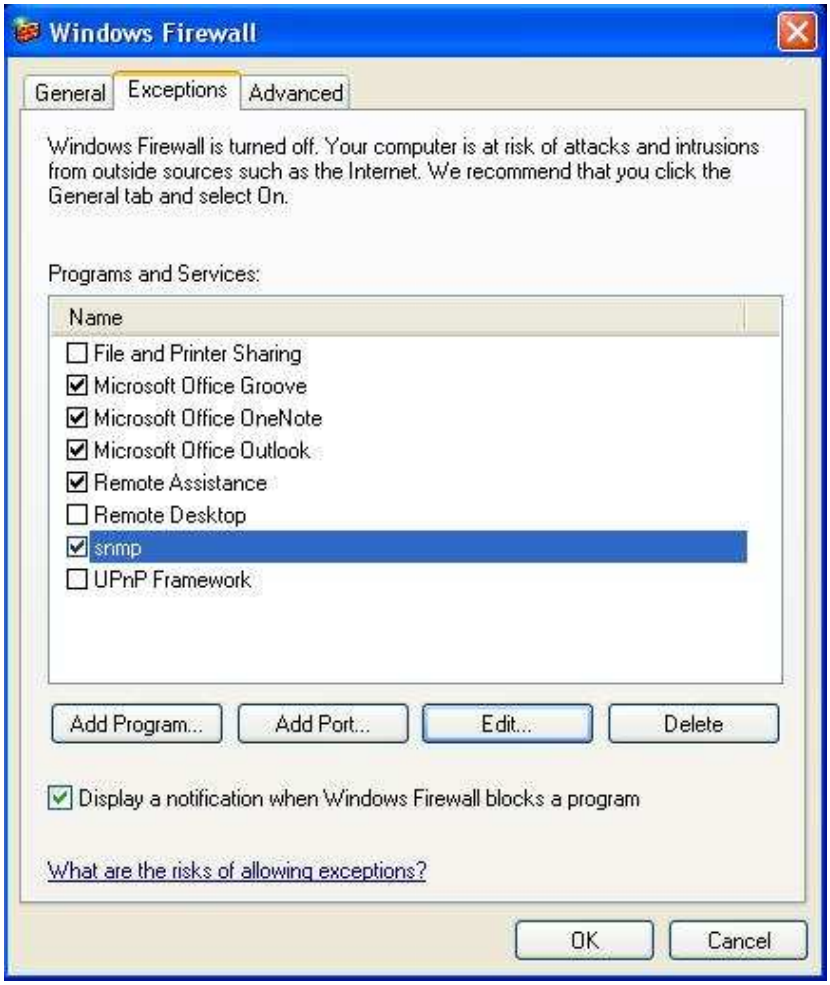

Figura 14 - Caixa de diálogo para adicionar regra de exceção para o SNMP

Um nome deve ser atrubuído para aparecer na lista de programas e serviços (Figura 15) com o número da porta usado. Deve ficar claro que este procedimento não abre a porta 161, apenas permite esta máquina receber informações SNMP.

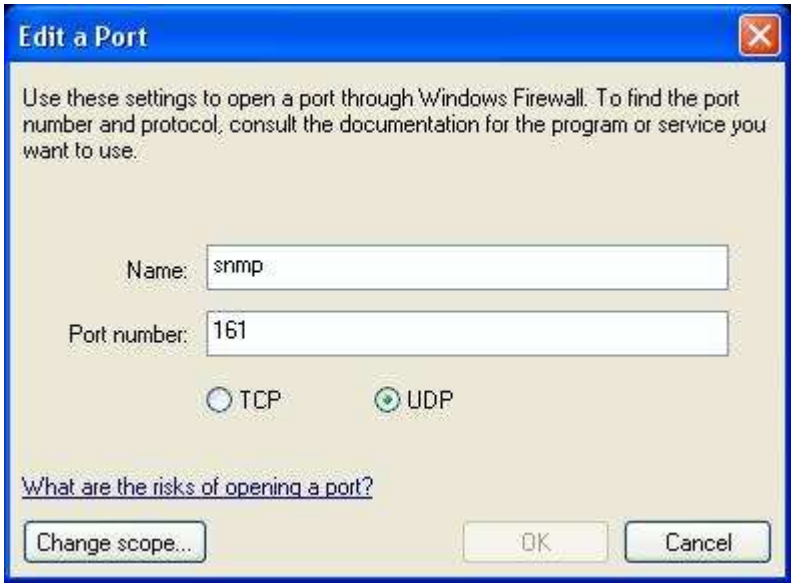

Figura 15 - Especificando a permissão de conexões SNMP

A máquina gerente, com a plataforma Linux, busca informações do agente com o sistema operacional Windows XP. Como o *Windows* tem duas opções que interpreta os comandos SNMP, deve-se diferenciar entre o serviço SNMP da *Microsoft* e o Net-SNMP. A Figura 7 mostra a situação de todos os serviços disponíveis, ativos ou inativos.

Deve-se instalar o agente Microsoft, porém, mantê-lo desabilitado para que as DLL (*Dynamically Linked Library* ou bibliotecas dinâmicas) estejam disponíveis. Os passos para instalar o agente Microsoft são: “painel de controle”, “Adicionar ou remover programas”, “Adicionar/remover componentes do Windows” e como mostra a linha destacada na Figura 16, confirme o componente no *checkbox* e clique em Avançar. É necessário indicar a localização dos arquivos de instalação do *Microsoft Windows* para o *Microsoft* SNMP.

Figura 16 - Instalação do SNMP da *Microsoft*

Com o serviço da *Microsoft* desativado, todo o funcionamento do protocolo SNMP é dado pelo Net-SNMP. Na Tabela 4 encontram-se os principais diretórios do software Net-SNMP, o qual efetivamente trata dos pacotes SNMP.

Tabela 4 - Estrutura dos principais diretório do Net-SNMP

Localização	Propósito
c:/usr/share/snmp/mibs	Agrupar as MIBs que são consultadas pelo gerente
c:/usr/lib	Biblioteca do protocolo
c:/usr/snmp/persist	Configurações de persistência
c:/usr/bin	Binários do protocolo
c:/usr/etc/snmp	Configurações do comportamento do *daemon*

c:/usr/log	Arquivos de registros

Após a instalação do Net-SNMP, é preciso configurar o arquivo C:\usr\etc\snmp\snmpd.conf que controla o *daemon snmpd* do agente. Uma forma básica do arquivo é exibida a seguir:

```
1  rocommunity fujitaro
2  rwcommunity fujitarw
3  monitor    -r  60  -e  linkUpTrap      "Generate  linkUp"
   ifOperStatus != 2
4  monitor    -r  60  -e  linkDownTrap  "Generate  linkDown"
   ifOperStatus == 2
```

As duas primeiras linhas mostram o nome da comunidade, sendo que a linha 1 que tem permissão de apenas leitura e a linha seguinte escrever em relação aos objetos das MIBs. As duas últimas linhas monitoram a comunicação entre gerente e agente. A cada sessenta segundos é impresso a palavra "*Cold Start*", caso a comunicação esteja ativa e "*netSnmpNotificationPrefix*", caso esteja inativo. Detalhes sobre a interpretação das mensagens pelo gerente encontra-se no item "3.1.3. Habilitação da porta de comunicação 162".

3.1.2. Instalação do Cygwin

Para instalar o *Cygwin* na plataforma *Windows*, deve-se obter o arquivo de instalação no endereço http://www.cygwin.com/setup.exe. Ao executar o arquivo, escolha "*Install from Internet*" para que os arquivos sejam baixados da internet e na caixa de diálogo seguinte, confirme as sugestões dadas clicando em "Avançar". A primeira vez que é executado não instala pacotes adicionais, deve conter apenas os que compõem a instalação básica [11]. Terminada a instalação e confirmado o funcionamento do cygwin, deve-se executar o mesmo arquivo para que outros pacotes sejam adicionados ao ambiente *Cygwin*. Não é

necessário reiniciar. Os principais diretórios do ambiente Cygwin são apresentados na Tabela 5.

Tabela 5 - Estrutura dos principais diretório do Cygwin

Localização	Propósito
C:/cygwin/bin	Binários do ambiente
C:/cygwin/cygdrive	Ponto de montagem dos dispositivos
C:/cygwin/etc	Configurações do ambiente
C:/cygwin/home	Arquivos pessoais
C:/cygwin/lib	Bibliotecas
C:/cygwin/var	Registros do eventos

A utilidade do software é criar *scripts* que desempenhem as mesmas funcionalidades do Linux. No entanto, é o *Windows* que gerencia algumas propriedades como inserção de caracteres não-imprimíveis. Por exemplo, ao escrever um *script* e teclar “enter”, o Linux sinaliza com o caractere de escape ‘\n’, enquanto o Windows insere ‘\n\r’, podendo ocorrer que o mesmo código funcione em um ambiente e não no outro. Um cuidado a ser tomado quando o *script* é desenvolvido no Windows é filtrar tais caracteres para que o interpretador de comandos execute obedecendo aos padrões do Linux.

3.1.3. Habilitação da porta de comunicação 162

Por não ser viável que o gerente busque informações de todos os objetos em todas as máquinas, a solução é receber notificações não solicitadas de eventos que ocorram. O gerente pode ser programado para ativar uma determinada tarefa quando ocorrer uma condição avisada através do *trap*. O

daemon snmptrapd por padrão abre a comunicação com a porta 162 e deve ficar ativo no gerente armazenando as informações oriundas dos agentes.

A partir da versão 5.3 e superiores do Net-SNMP, o sistema registra as ocorrências dos *traps* no arquivo snmptrapd.log como a seguir [12]:

```
1   MIB::netSnmpAgentOIDs.13 0 Cold Start TRAP, SNMP v1,
    community fujitaro
2   23/7/2008     10:54:6     10.10.8.115     NET     SNMP
    MIB::netSnmpAgentOIDs.13 3 Link Up TRAP, SNMP v1,
    community fujitaro
3   23/7/2008    10:54:6    10.10.8.115    DISMAN    EVENT
    MIB::dismanEventMIBNotificationPrefix   6   Enterprise
    Specific TRAP, SNMP v1, community fujitaro
```

A linha 1 indica que a comunicação foi iniciada. A linha 2 indica que a comunicação está ativa. A última linha avisa que o *daemon snmpd* da máquina agente não está ativo.

O arquivo /usr/local/share/snmp/snmptrapd.conf do gerente configura o *daemon snmptrapd* de acordo com as diretivas: controle de acesso, formato das mensagens e notificações de processamento.

O exemplo abaixo mostra a configuração do arquivo citado [12].

```
1  authCommunity   log,execute,net    fujitaro
2  pidfile  /var/run/snmp.pid
3  ignoreauthfailure  0
4  disableAuthorization yes
5  traphandle             NET-SNMP-AGENT-MIB::nsNotifyShutdown
   /home/murilo/down.sh desligado
6  traphandle SNMPv2-MIB::coldStart         /home/murilo/up.sh
   ligado
7  traphandle                              SNMPv2-MIB::coldStart
   /home/murilo/verifica_memoria.sh  memoria
```

Esta configuração é válida para versões 5.3 e superiores do Net-SNMP. A seguir encontram-se os significados das instruções:

- ***authCommunity*** diz que a comunidade fujitaro pode gerar logs, executar um programa quando acontecer um determinado evento e encaminhar a mensagem para um outro destino.
- ***pidfile*** diz em qual arquivo deve ser armazenado o número do processo SNMP.
- ***Ignoreauthfailure*** indicado por 0 (zero) significa que as mensagens dizendo que tiveram falhas de autenticação não são ignoradas.
- ***disableAuthorization*** não checa o controle de acesso aceitando todas as mensagens *traps* que chegam.

As instruções *traphandle* das linhas 5 e 6 invocam *scripts* pra situações distintas: quando o *daemon snmpd* do agente pára, é enviado um e-mail avisando a data e hora sobre o ocorrido. O mesmo acontece quando a comunicação é restabelecida. A instrução da linha 7 faz uma checagem na quantidade de memória RAM (Random Access Memory) da máquina assim que for ligada.

Em cada situação o campo “assunto” é personalizável. Como os e-mails são checados para serem identificados como *spams* ou não, para garantir a entrega, nesta dissertação o endereço usado é o domínio unifei.edu.br [10].

O *script snmpd* localizado em /etc/init.d/snmpd contêm as instruções de controle tanto para o *daemon snmpd* como para o *snmptrapd*. Para o gerente não se ativa o *snmpd*, apenas o *daemon* que recebe as notificações, o *snmptrapd*. Essa configuração é possível através do arquivo /etc/default/snmpd que contém as seguintes instruções:

```
1   SNMPDRUN=no
2   TRAPDRUN=yes
```

A atribuição das palavras “yes” e “no” são interpretadas pelo *script* inicializando ou não os serviços sendo “snmpdrun” para controlar o *snmpd* e “*trapdrun*” refere-se a *snmptrapd*.

3.1.4. A configuração do apache

O Apache é o responsável pela interface gráfica tornando amigável lidar com os comandos SNMP. O arquivo principal é /etc/apache/apache2.conf que é lido ao iniciar o serviço para determinar seu comportamento. O conteúdo informa quais módulos são carregados caso necessite comunicação com outras linguagens, número máximo de conexões simultâneas para evitar sobrecarga, localização dos arquivos que registram os eventos, página a ser carregada quando digitado um endereço errado que pertença àquele domínio entre outras.

O arquivo /etc/apache2/sites-available/default instrui o Apache a buscar os diretórios de hipertextos e os executáveis. O conteúdo é exibido abaixo [7]:

```
DocumentRoot /var/www/
   <Directory />
       Options FollowSymLinks
       AllowOverride None
   </Directory>

ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
   <Directory "/usr/lib/cgi-bin">
       AllowOverride None
       Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
       Order allow,deny
       Allow from all
   </Directory>
```

No primeiro bloco indicado por *DocumentRoot* encontra-se a localização de todos os arquivos com extensão *.htm e *.html que são os hipertextos e interpretados quando houver solicitação.

O segundo bloco designado por *ScriptAlias* contém a instrução onde se situam os arquivos executáveis. Os arquivos com extensão *.cgi tornam as páginas dinâmicas, ou seja, alteram o conteúdo da página a cada solicitação. O CGI executa binários de diversas linguagens como Perl, PHP, C entre outras e neste trabalho foi utilizado *Shell Script*, que são instruções realizadas pelo ambiente Bash, um interpretador de comandos de qualquer distribuição Linux.

3.2. Configuração do roteador

A configuração dos roteadores varia de acordo com as marcas e modelos existentes no mercado. No entanto, o princípio é o mesmo: configurar o dispositivo de forma que um pacote com determinadas especificações seja encaminhado para outra máquina. Um recurso normalmente conhecido como "*Port Forwarding*" (redirecionamento de portas) tem a função de analisar as solicitações que chegam e redirecionar para a máquina a qual executa um determinado serviço [13]. Na Figura 17 é mostrado o caso de redirecionamento de pacotes SNMP do seguinte modo: protocolos do tipo UDP que chegarem na porta 161 do roteador devem ser redirecionados para a máquina 192.168.254.98.

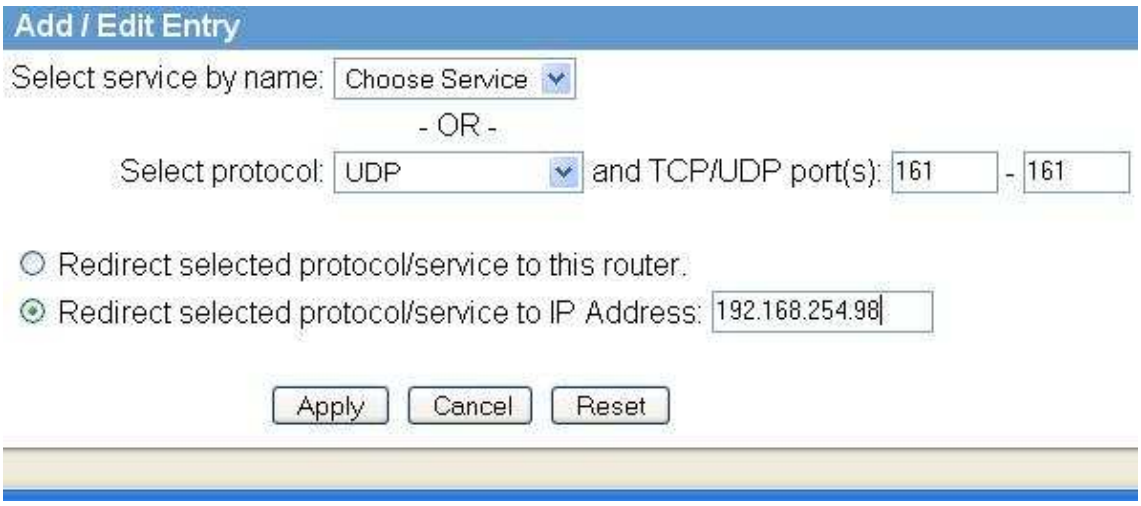

Figura 17 - Configuração de redirecionamento de pacotes

Ao aplicar as configurações, um quadro mostra a tabela de roteamento do roteador. A Figura 18 exibe os dados os quais foram usados para redirecionamento de portas.

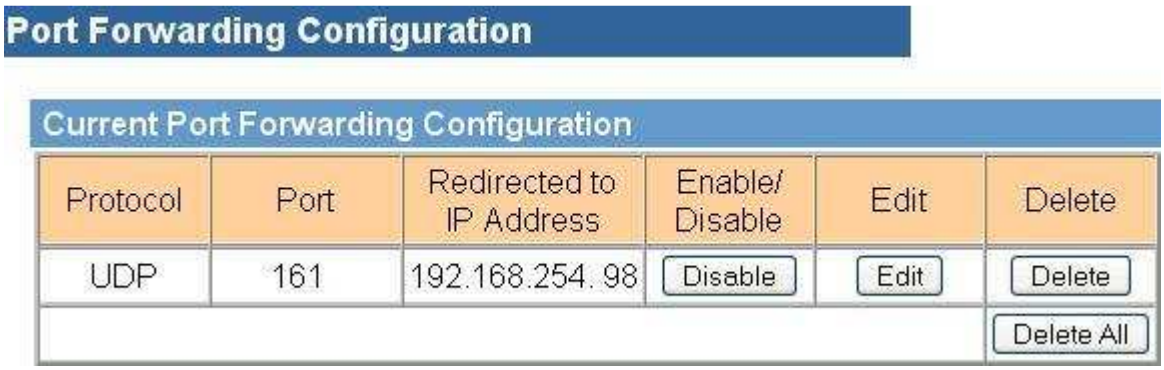

Figura 18 - Programação de redirecionamentos de portas do roteador

3.3. O desenvolvimento dos scripts

A máquina gerente deve ter um servidor de páginas para que a interface seja visualizada. Foi escolhido o Apache web Server pela sua versatilidade, pois atende as necessidades como interpretação de HTML e execução de binários.

Deve-se preparar o ambiente para que bibliotecas compartilhadas sejam carregadas para o funcionamento do protocolo. O sistema consulta uma variável chamada LD_LIBRARY_PATH para saber onde buscar as bibliotecas. É uma variável de ambiente útil para depurar novas bibliotecas ou usar as que não sejam padrão para propósitos especiais. O comando *export* torna-a global não necessitando informar o caminho completo da localização das bibliotecas. Na prática, a instrução que deve ser informada é:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
```

Este comando atribui a LD_LIBRARY_PATH manter o conteúdo anteriormente armazenado ($LD_LIBRARY_PATH) e acrescenta o diretório

/usr/local/lib. Caso contrário, o valor é sobrescrito. Outros diretórios podem ser informados separados por dois pontos (:). Lendo a variável informada, o que o sistema busca é as bibliotecas do protocolo SNMP listadas a seguir:

libnetsnmp.a	libnetsnmpagent.so.15.1.1	libnetsnmpmibs.so.15
libnetsnmp.la	libnetsnmphelpers.a	libnetsnmpmibs.so.15.1.1
libnetsnmp.so	libnetsnmphelpers.la	libnetsnmptrapd.a
libnetsnmp.so.15	libnetsnmphelpers.so	libnetsnmptrapd.la
libnetsnmp.so.15.1.1	libnetsnmphelpers.so.15	libnetsnmptrapd.so
libnetsnmpagent.a	libnetsnmphelpers.so.15.1.1	libnetsnmptrapd.so.15
libnetsnmpagent.la	libnetsnmpmibs.a	libnetsnmptrapd.so.15.1.1
libnetsnmpagent.so	libnetsnmpmibs.la	
libnetsnmpagent.so.15	libnetsnmpmibs.so	

O comando que exporta o valor da variável retém o conteúdo temporariamente na memória. Para evitar digitá-lo sempre que for ligar a máquina gerente, o usuário pode guardar em um arquivo que é lido assim que se autenticar no sistema. Através de qualquer editor de texto, deve-se inserir exatamente a mesma instrução no arquivo ${USER}/.bashrc [6].

Além da comunicação entre máquinas e dispositivos que é estabelecida através do protocolo, é necessária uma forma organizada de realizar tarefas por parte da máquina agente. Antes de iniciar o desenvolvimento dos *scripts*, é essencial conhecer as ferramentas do ambiente Bash. Na Tabela 6 encontram-se alguns comandos:

Tabela 6 - Alguns comandos do Linux e suas utilizações

Comando	Opção	Ação
grep	(sem opção)	Seleciona as linhas que aparecem a palavra pesquisada.

grep	-v	Seleciona as linhas que não aparecem a palavra pesquisada.
sed	/s/velho/novo	Substitui o texto “velho” pelo “novo”.
tr	-d	Apaga a ocorrência solicitada.
cut	-c I-	Imprime o texto a partir do caractere número I.
awk	(sem opção)	Recebe uma massa de dados na entrada e imprime um texto formatado.

Existem muitos outros comandos capazes de manipular arquivos, formatar textos, trabalhar com variáveis, armazenar valores, realizar cálculos e demais conveniências. Compreendendo o propósito dos comandos bem como suas opções, a pessoa está apta a sequenciar uma variedade de comando que formam o *script*. Ao analisar a saída de cada comando, existem várias lógicas para adequar às necessidades do sistema dependendo da criatividade do programador.

Após a realização de instalação e configuração do ambiente, a próxima etapa é o desenvolvimento de *scripts* para os fins de gerenciamento de rede e supervisório.

3.3.1. Desenvolvimento de scripts para gerenciamento

Uma fábrica ou qualquer outro local que empregue uma quantidade expressiva de dispositivos precisa ter controle do estado de cada um e este propósito é coberto através de recursos do protocolo TCP/IP. Por mais que a rede se expanda tornando-se complexa, todos os equipamentos podem ser checados através do gerenciamento de rede.

Para obter a informação da máquina remota, o usuário clica no botão fazendo o CGI invocar um comando SNMP buscando um valor armazenado no objeto da MIB. Até este ponto a informação está aquisitada, porém é necessário

selecionar somente a informação relevante e esta finalidade é desempenhada pelo Shell, o qual emprega filtros e extrai o conteúdo conveniente para o usuário. Algumas respostas estão além do que a MIB armazena, sendo necessário empregar um recurso conhecido como “extensão do agente” (arquivo snmpd.conf) [12]. Ao invés de consultar os objetos padrões, é possível programar o agente para realizar tarefas quando o gerente consultar uma determinada OID.

Um problema enfrentado no presente trabalho foi a forma como acentos são interpretados. Quando as saídas do agente não têm acentos, a mensagem é interpretada corretamente pelo gerente. O mesmo não acontece quando há caracteres acentuados trafegando de uma máquina a outra, aparecendo uma tradução para numeração hexadecimal, tornando a resposta incompreensível. A solução encontrada foi acrescentar a opção –Oa que substitui os acentos por um ponto final (.) .

3.3.1.1. Script para Gerenciamento Em Funcionamento – GEF

Ao clicar no botão “Em funcionamento”, o gerente pede que o agente execute o comando *uptime* e este informa quanto tempo a máquina está ligada desde a última inicialização. Respostas desta instrução são exemplificadas abaixo:

```
1  15:30:07 up 19 min, 0 users, load average: 0.00, 0.00, 0.00
2  17:38:15 up 2:10, 0 users, load average: 0.00, 0.00, 0.00
3  15:39:15 up 11 days, 4:12, 7 users, load average: 0.08,
   0.14, 0.07
```

Como pode ser notado na primeira linha, apenas o terceiro e quarto campos (19 min) são relevantes para saber o tempo de atividade da máquina. Já na próxima linha, é preciso apenas o terceiro campo (2:10), enquanto que na última linha são aproveitadas do terceiro ao quinto campos (11 days, 4:12) [6]. Este *script* armazena os valores dos cinco primeiros campos e os filtros

manipulam o conteúdo de cada variável de forma que a tela mostre os campos dia(s), horas e minutos da máquina consultada. O código-fonte encontra-se no anexo B.1.

3.3.1.2. Script para Gerenciamento de Log – GLog

Necessitando ter o conhecimento sobre o estado de comunicação entre as máquinas, o CGI-bin que é invocado através deste botão converte o arquivo de registros snmptrapd.log em uma página com formatação HTML. Um cuidado ao elaborar este *script* foi imprimir na tela as dez últimas linhas e atualizar a página a cada cinco segundos [7]. O código-fonte encontra-se no anexo B.2.

3.3.1.3. Script para Gerenciamento de Portas Ativas – GPA

O *script* invoca o comando *netstat* que devolve uma tabela com as conexões ativas (TCP e UDP) entre a máquina consultada e endereços externos bem como a situação da conexão (estabelecida, ouvindo, *socket* em espera, etc). Exemplificando, a resposta é da forma:

```
TCP    0.0.0.0:37     0.0.0.0:0        LISTENING
TCP    0.0.0.0:135    0.0.0.0:0        LISTENING
TCP    0.0.0.0:445    0.0.0.0:0        LISTENING
TCP    0.0.0.0:3580   0.0.0.0:0        LISTENING
UDP    0.0.0.0:37             *:*
UDP    0.0.0.0:123            *:*
UDP    0.0.0.0:161            *:*
```

A informação relevante está na segunda coluna após o sinal de dois pontos (:). Os filtros retiram a pontuação separando os campos para exibir apenas o número das portas ativas [6]. O código-fonte encontra-se no anexo B.3.

3.3.1.4. Script para de Gerenciamento Sistema – GS

A elaboração desta tarefa trata de mostrar as informações mais relevantes do ramo System da MIB sendo:

i. Descrição do sistema. Informa a arquitetura e o sistema operacional da máquina consultada.

ii. Contato. O endereço de e-mail do responsável para comunicar a ocorrência de um problema.

iii. Nome do agente. Nome da máquina consultada.

iv. Localização. Indica onde está situada a máquina consultada.

Dois campos estão logo abaixo para modificar o contato e a localização e ambos funcionam sob a mesma lógica: a informação é enviada para outro *script* que separa os campos, pois sinais são adicionados quando formulários são enviados (descrito no item 2.15.6). Em seguida a informação relevante é passada como parâmetro pela função *snmpset* para modificar a MIB, como mostrado abaixo:

```
murilo@fujita2:~$ snmpset -v 1 -c fujitarw fujita sysContact.0
s "`echo $QUERY_STRING | cut -c 9- | sed 's/%40/@/'`"
SNMPv2-MIB::sysContact.0 = STRING: murilo@unifei.edu.br
```

Para alterar o conteúdo do objeto é necessário que o nome da comunidade tenha permissão de escrita, como exemplificado acima. O código-fonte para visualizar as informações do sistema encontra-se no anexo B.4. Para alterar a informação do contato (responsável pelo sistema) veja anexo B.5. Para alterar a informação da localização do agente, veja anexo B.6.

3.3.1.5. Script para Gerenciamento de Conectividade – GC

Quando o gerente interroga sobre informações relativas à rede como IP, máscara de rede, gateway, endereço físico e os servidores de DNS, o agente responde com o comando *ipconfig*. Comandos do Linux selecionam somente a informação importante para serem exibidas e por fim, instruções em HTML estruturam os dados em forma de tabela sem bordas, para organizar a página. O código-fonte encontra-se no anexo B.7.

3.3.1.6. Script para Gerenciamento de Volume – GV

Para obter informações sobre as partições do agente é preciso invocar o comando *df* do software Cygwin.

```
murilo@fujita2:~$ snmpwalk -v 1 -c fujitaro fujita
1.3.6.1.4.1.30778.7.3.1.2.7.47.98.105.110.47.115.104
SNMPv2-
SMI::enterprises.30778.7.3.1.2.7.47.98.105.110.47.115.104    =
STRING: "Filesystem    Type    Size  Used Avail Use% Mounted
on
C:\\cygwin\\bin
      system,fixed     20G  7.6G   12G  40% /usr/bin
C:\\cygwin\\lib
      system,fixed     20G  7.6G   12G  40% /usr/lib
C:\\cygwin
      system,fixed     20G  7.6G   12G  40% /
c:    system,fixed     20G  7.6G   12G  40% /cygdrive/c
d:    system,fixed     19G   12G  6.3G  66% /cygdrive/d
f:
  system,removable    3.9G  351M  3.5G   9% /cygdrive/f"
```

Para selecionar a informação relevante neste caso, filtros precisam diferenciar o que pertence ao sistema das partições. A saída do comando, como mostrado a seguir, executa os trabalhos desta separação criteriosa, elimina sinais de pontuação e abaixo do cabeçalho, imprime as informações solicitadas.

```
murilo@fujita2:~$ snmpwalk -v 1 -c fujitaro fujita
1.3.6.1.4.1.30778.7.3.1.2.7.47.98.105.110.47.115.104 | grep
cygdrive | cut -c 3- | sed 's/"//g' | sed 's/\/cygdrive\///g'
| sed 's/system,//g' | awk 'BEGIN {print"Unid Tipo
Total Ocupado Livre %Ocupado\n"} {printf"%3s: %10s %7s
%9s %7s %9s\n",$6,$1,$2,$3,$4,$5}'


Unid       Tipo   Total   Ocupado   Livre  %Ocupado

  c:      fixed     20G      7.6G     12G       40%
  d:      fixed     19G       12G    6.3G       66%
  f:  removable    3.9G      351M    3.5G        9%
```

O código-fonte encontra-se no anexo B.8.

3.3.1.7. Script para Gerenciamento de Dados Agente – GDA

Nesta situação, é retornada uma listagem detalhada através do comando *systeminfo*, que é intrínseco do *Windows* sobre o agente consultado contendo dados do sistema operacional, fuso horário, informações sobre memória, atualizações de software entre outras.

Uma observação com relação ao campo “System Up Time”/”Tempo de ativação do sistema” o qual informa o tempo de atividade desde a inicialização da máquina: quando o *Windows* entra no estado de hibernar, o contador de tempo não é reinicializado. Para saber efetivamente o tempo decorrido deve-se utilizar a tarefa “Em Funcionamento” descrita no item 3.3.1.1. O código-fonte encontra-se no anexo B.9.

3.3.1.8. Script para Gerenciamento Ligar Agente - GLA

Este *script* tem a tarefa de ligar o computador remotamente através do software do Linux wake-on-lan (descrito no item 2.15.3). Porém, antes de

executar o *script* surge uma tela pedindo autenticação. Assim que o sistema confirma que os dados estão corretos, o *script* GLA.cgi é acionado e uma tela informando a situação da comunicação entre gerente e agente é atualizada a cada dois segundos. O código-fonte encontra-se no anexo B.10.

3.3.2. Desenvolvimento de scripts para supervisório

A Figura 19 ilustra os processos de comunicação para controlar um dispositivo dependendo da combinação se for entrada ou saída, bem como se as grandezas são analógicas ou digitais. Independente das etapas que podem suceder segundo o esquemático abaixo, o agente sempre consulta a MIB, executa um *script* e aciona um binário compilado na linguagem C (não necessariamente nesta ordem). O código-fonte pode ser recompilado conforme a necessidade de interface com os dispositivos.

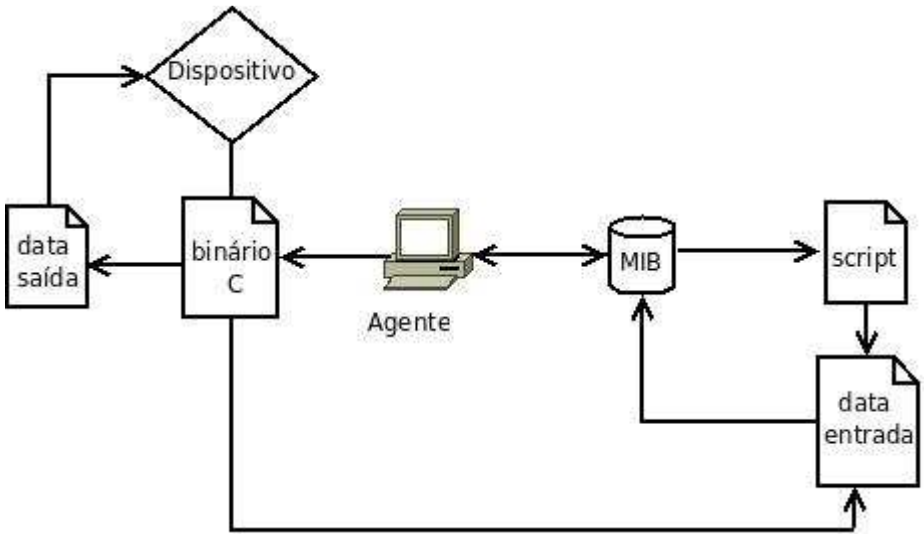

Figura 19 - Esquemático de entradas/saídas de hardware controlado por software

Para executar a função de supervisório será necessário capturar valores de tensão ou corrente e o estado de chaves liga/desliga conectados através de uma interface apropriada ao microcomputador. Caso seja necessário realizar controle, enviam-se comandos por meio de interfaces acionando dispositivos através de sinais analógicos e/ou digitais. Considerando as tarefas realizadas, o protocolo pode ser adaptável para outras grandezas ou mesmo, outros dispositivos.

Para realizar a parte de supervisório, o equipamento usado foi uma placa de aquisição de dados da *National Instruments* modelo USB 6008 conectada ao agente através da porta USB. A execução conjugada dos binários e *scripts* aquisita dados combinando entradas/saídas com sinais analógicos/digitais. A Figura 20 esquematiza o objeto de estudo para realizar as funções de supervisório (dispositivo da Figura 19).

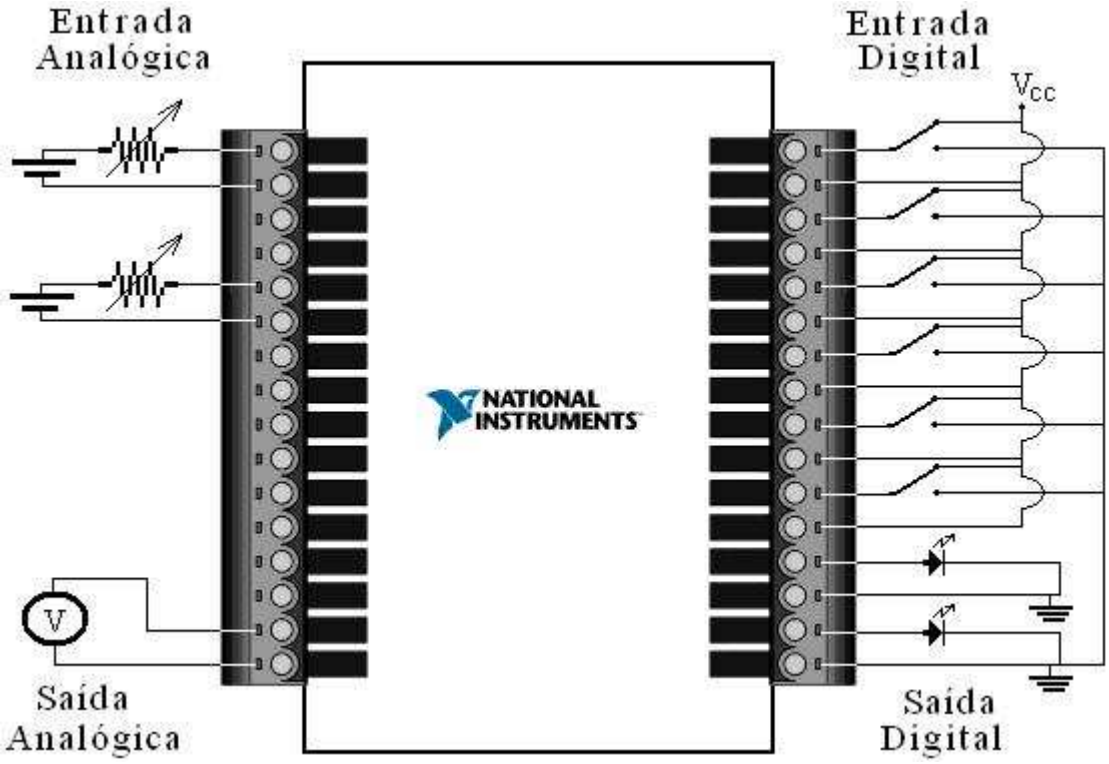

Figura 20 - Conexões da placa de aquisição de dados USB 6008

3.3.2.1. Script para Supervisório Entrada Analógica – SEA

O circuito conectado a placa de aquisição de dados contém fontes de alimentação de 12 V (corrente contínua) e um potenciômetro para causar variações na amplitude.

O binário em C permanece aquisitando dados constantemente de um canal e armazena o valor em cada linha do arquivo texto. Conhecimentos de funções em C foram aplicados para gerenciar as chamadas uma de cada vez, pois ocorre o estrangulamento do processador no caso de aquisições simultâneas. Cada canal executa as mesmas funções variando a porta de onde aquisita o valor, chama um ponteiro para abrir o arquivo com permissão de escrita, inicia a leitura e fecha o ponteiro. O código fonte do binário desenvolvido para a placa da National USB 6008 encontra-se no anexo A.1.

Já os *scripts* são executados paralelamente no caso de uma requisição do gerente. Quando acontece esta solicitação, o agente consulta a MIB e busca o *OID murilofujita* para ativar o *script* específico, o qual tem a instrução de buscar os valores das entradas analógicas. Estes poderão ser tratados de acordo com uma aplicação específica do usuário, ou seja, o mesmo poderá simplesmente disponibilizar estes valores em uma página HTML ou realizar um processamento para uma eventual tomada de decisão. O código fonte deste *script* encontra-se no anexo C.1.

3.3.2.2. Script para Supervisório Entrada Digital – SED

Na placa de aquisição de dados estão conectadas chaves do tipo liga/desliga. O binário compilado em C executa constantemente a leitura do estado de cada chave e armazena em um arquivo de texto. As informações são indicadas pelo número 1 (um) quando a chave encontra-se ligada e 0 (zero)

quando desligada. A sequência como é organizada a conexão dos fios também é relevante, pois cada linha do arquivo de texto armazena na ordem do bit mais significativo para o menos significativo. O código fonte do binário desenvolvido para a placa da National USB 6008 encontra-se no anexo A.2.

Quando o gerente solicita a informação sobre a entrada digital, o agente consulta a MIB para buscar a *OID murilofujita* e executa o *script* que busca o dado mais atual, elimina os dados que não sejam os valores das chaves e atualiza a página periodicamente ou realizar um processamento para uma eventual tomada de decisão. O código fonte deste *script* encontra-se no anexo C.2.

3.3.2.3. Script para Supervisório Saída Analógica – SSA

Para gerar uma saída analógica, o valor solicitado pelo usuário é tratado no gerente. Inicialmente é checado se o valor está dentro da faixa (0 a 5 V). Se estiver fora, o valor é fixado em 5 V caso ultrapasse o limite superior ou 0 V se for menor que o limite inferior. Após a verificação, um *script* com a instrução de transferir o arquivo de lote é enviado para o agente através do serviço de compartilhamento do *Windows*, já que o SNMP não transfere arquivos. O conteúdo do arquivo de lote é composto por um cabeçalho e o valor solicitado.

Para executar tarefas específicas como acionar o *script* descrito, o agente consulta o *OID murilofujita* contida na MIB. O código fonte deste *script* encontra-se no anexo C.3.

Em seguida, o binário compilado em C é executado tendo como parâmetro o valor contido no arquivo de lote. O código-fonte C utiliza as instruções *argc* e *argv* para saber quantos parâmetros são usados e emprega os parâmetros como entrada de dados. Para cada saída analógica deve existir um arquivo contendo o valor solicitado. O código fonte do binário desenvolvido para a placa da National USB 6008 encontra-se no anexo A.3.

3.3.2.4. Script para Supervisório Saída Digital – SSD

Para controlar as saídas digitais, o valor solicitado pelo usuário é tratado no gerente. Um arquivo de lote é montado contendo um cabeçalho e um número na base decimal dependendo de quais saídas o usuário solicita. Este arquivo é transferido do gerente para o agente pelo serviço de compartilhamento do *Windows* pois o SNMP não transfere arquivo. De maneira similar ao *script* anterior, para realizar esta instrução específica, o agente consulta o *OID murilofujita* contida na MIB. O código fonte deste *script* encontra-se no anexo C.4.

Em seguida, o binário compilado em C é executado tendo como parâmetro o valor contido no arquivo de lote. O código-fonte C utiliza as instruções *argc* e *argv* para saber quantos parâmetros são usados e emprega os parâmetros como entrada de dados. Com um único arquivo é possível controlar todas as saídas digitais, uma vez que o número decimal é convertido em binário. Feita a conversão, o binário em C deixa a porta em estado alto quando o valor é 1 (um) e baixo quando 0 (zero). O código fonte do binário desenvolvido para a placa da National USB 6008 encontra-se no anexo A.4.

3.3.3. Scripts acionados pelos traps

Quando um evento *trap* acontece é registrado nos relatórios do gerente. Desta forma, o gerente executa um *script* com o propósito de realizar uma determinada ação.

De acordo com a configuração do arquivo /usr/local/share/snmp/ snmtrapd.conf, a diretriz *traphandle* invoca os *scripts down.sh* e *up.sh* ao

receberem notificações *nsNotifityShutdown* e *ColdStart*, respectivamente, para enviar um e-mail avisando se o *daemon* SNMP está ativo ou não. O código fonte do script down.sh encontra-se no anexo D.2 e up.sh no anexo D.1.

O *script verifica_memoria.sh* checa a quantidade de memória RAM do agente assim que for ligado. A quantidade de memória RAM de cada máquina está catalogada em um arquivo e é comparada com o valor obtido quando o gerente consultar o agente. Assim, qualquer divergência numérica é relatada através de um e-mail avisando sobre a remoção ou adição na quantidade de memória RAM. Caso os números coincidam, nenhuma ação é tomada. O coódigo fonte do script *verifica_memoria.sh* encontra-se no anexo D.3.

4. Desenvolvimento de uma aplicação

Os ensaios da arquitetura desenvolvida foram realizados em uma sala com uma rede de computadores. Neste ambiente os usuários utilizam as máquinas para pesquisas e consultas.

Adicionalmente, outro resultado realizado com êxito foi a comunicação do protocolo em redes distintas. O gerente permaneceu no ambiente de desenvolvimento enquanto o agente foi testado em um domicílio.

Para acessar a arquitetura digita-se o endereço da máquina gerente em qualquer navegador que tenha comunicação com o gerente. A tela que aparece é a planta baixa da sala onde se encontra a rede de computadores como mostra a Figura 21. Etiquetas contendo o nome do ocupante contribuem para facilitar a localização no ambiente. Para o caso de alterar uma ou mais etiquetas, deve-se editar o arquivo /var/www/labels e executar o *script* labels.sh contido no mesmo diretório.

Para ter acesso aos dados de uma máquina, um clique no quadrado que envolve a figura faz aparecer a tela do aplicativo desenvolvido de acordo com a Figura 22. Apenas uma máquina foi utilizada para conectar a placa de aquisição de dados para realizar tanto as funções de gerenciamento de rede e de supervisão de processos. Esta máquina tem a etiqueta “Murilo”. As demais têm apenas os recursos de gerenciamento de rede.

Ao iniciar o aplicativo, é exibida uma tela de verificação e se há resposta do agente. Caso não haja, uma mensagem de “Sem comunicação com o agente” é impressa a cada dois segundos até que o enlace da conexão seja estabelecido. Em caso de sucesso de comunicação é impresso “Agente respondendo”, sinalizando a existência de comunicação entre as máquinas.

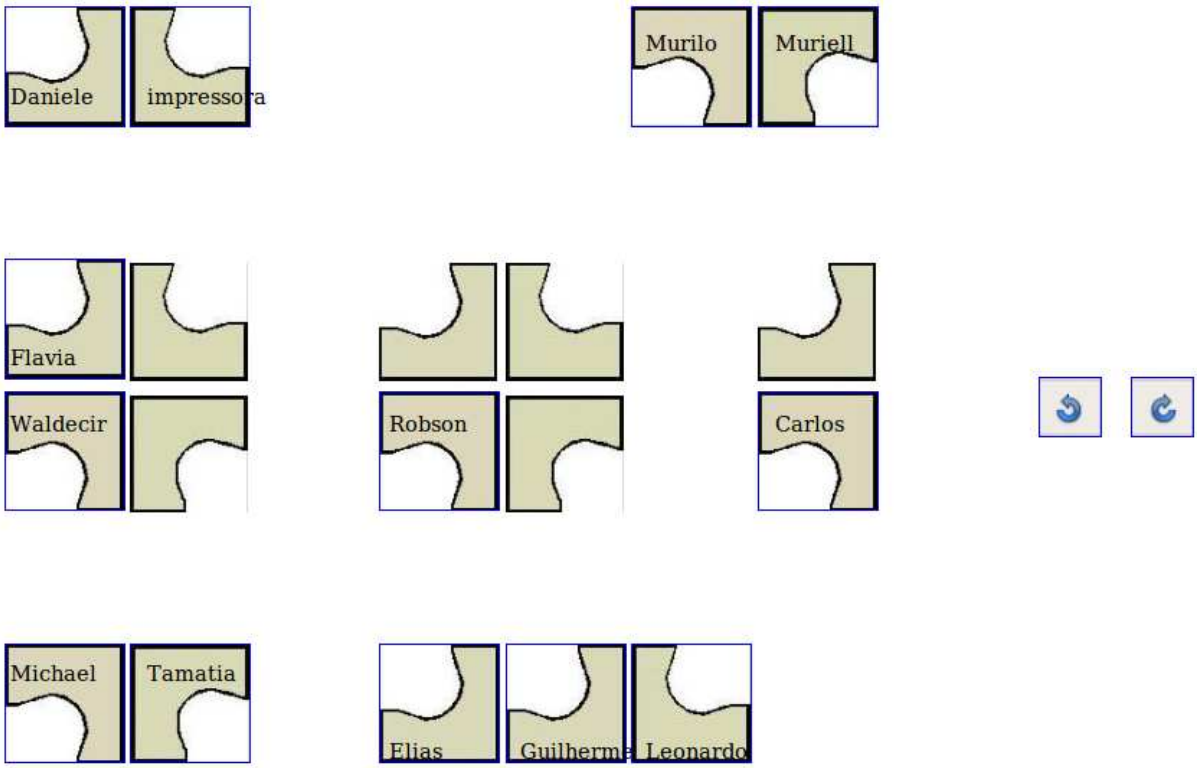

Figura 21 - Tela de entrada da arquitetura proposta

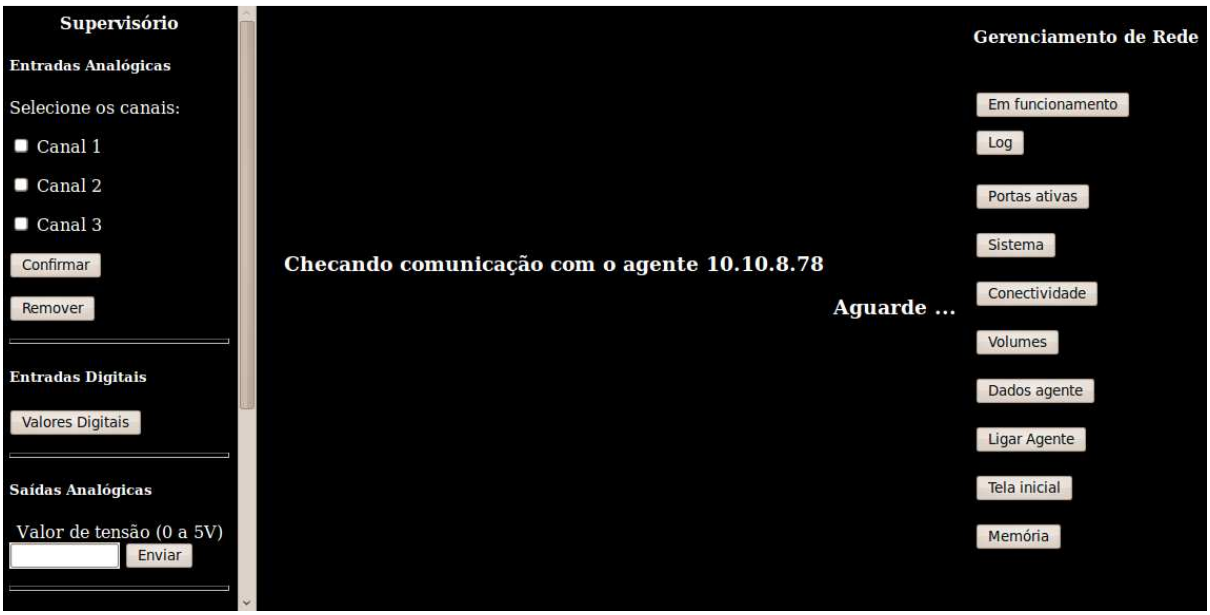

Figura 22 - Tela inicial do aplicativo para a máquina supervisório

Do lado esquerdo da Figura 22 localizam-se os recursos da ferramenta supervisório. Neste caso foram programadas somente três entradas analógicas (canal 1 a 3), seis entradas digitais, uma saída analógica e três saídas digitais.

Do lado direito estão as funcionalidades do gerenciamento de rede. Na parte central da tela é visualizada a resposta do solicitante.

4.1. Adequação do software para outras aplicações

A primeira forma abordada é utilizar um equipamento com o protocolo SNMP, pois o acesso é facilitado pela documentação do fabricante. Um documento satisfatório deve abranger informações como o número da porta utilizado, versão do protocolo, nome da comunidade, o ramo que está localizado o PEN e quais as instâncias em funcionamento.

Caso o equipamento não tenha o protocolo SNMP, deve-se instalar o software Net-SNMP em uma máquina que é denominada agente e configurar o arquivo *snmpd.conf* para adequar-se aos demais equipamentos da rede. Tanto o software como a documentação para configuração podem ser conseguidos no mesmo endereço: *http://www.net-snmp.org*.

Uma vez escolhido o software SNMP deve ser instalado entre as várias opções, o aplicativo desenvolvido pode ser modificado para atender necessidades como mudança de interface, formatação do texto, adição de outros equipamentos entre outras necessidades [17]. A funcionalidade do aplicativo desenvolvido parte do princípio que os botões e formulários são enviados para um aplicativo padrão CGI que realiza o evento solicitado pelo usuário. Desta forma, a máquina gerente faz um processo de busca e/ou armazenamento de uma ou mais variáveis e devolve a informação atualizada por meio de uma página HTML. Este processo é aplicado para qualquer equipamento que se deseja monitorar como um *switch*, um CLP (Controlador Lógico Programável), uma UPS (*Uninterruptible Power Supply*) entre outros.

4.2. Excutando o software

Ao digitar o endereço da máquina gerente na barra de endereços do navegador, é executado o software chamando automaticamente o arquivo que contém as instruções como a divisão da tela e elementos gráficos. Estas instruções buscam os arquivos que tenham o nome *índex*, variando a extensão *.html ou *.htm. A tela é dividida em quadros, separando a parte que o usuário consulta o sistema da que se exibe resposta.

Como há uma diferença na forma como os sistemas operacionais apresentam os caracteres, podem surgir palavras mal-formadas por causa dos acentos. A correção é feita configurando o navegador para a codificação correta através dos menus localizados na parte superior: Ver, Codificação, UTF-8.

Para interagir com o sistema, o operador clica no botão rotulado com um nome que realiza uma tarefa determinada. Cada botão é associado a um evento que é ativado pelo CGI para executar o *script* associado.

4.2.1. A ferramenta de Gerenciamento de rede

Como exemplo da utilização do gerenciamento de rede, a tela da Figura 23 mostra, as informações na parte central, após o botão “sistema” ser acionado pelo usuário. O *script* a ser executado é o GS (3.3.1.4 da página 58). Esta é uma situação específica na qual o usuário pode modificar os valores dos objetos que armazenam o contato do responsável pela rede ou a localização do agente consultado. O *script* lê o conteúdo preenchido nos campos que são armazenados na variável QUERY_STRING, filtra os dados relevantes eliminando os dígitos hexadecimais, e a cadeia resultante torna-se um parâmetro do comando *snmpset*.

De maneira similar, os outros botões acionam os *scripts* correspondentes apresentados no item 3.3.1.

O desenvolvimento de gerenciamento de rede exige forte familiaridade com os serviços do protocolo TCP/IP e *scripts* para obter informações específicas das máquinas conectadas em rede e filtrar as respostas; exibindo o que for relevante. De acordo com os resultados obtidos, o usuário dos recursos de gerenciamento de rede poderá usufruir de informações de uma maneira mais amigável, uma vez que todo o tratamento dos dados é processado por *scripts*.

Figura 23 - Detalhe da página mostrando os campos para modificar objetos

4.2.2. A ferramenta Supervisório

A Figura 24 mostra um exemplo da utilização dos recursos do supervisório para controlar as saídas digitais. O *script* captura quais portas foram selecionadas através da URL enviada pelo formulário e armazena na variável QUERY_STRING. Cada porta selecionada tem um valor que segue o sistema de numeração binária para montar o arquivo de lote, transferido e executado no agente no caso do evento receber um clique no botão

“Confirmar”. Como resposta, a parte central da tela exibe o estado de cada porta. Ao clicar em “Remover”, as caixas selecionadas são desmarcadas.

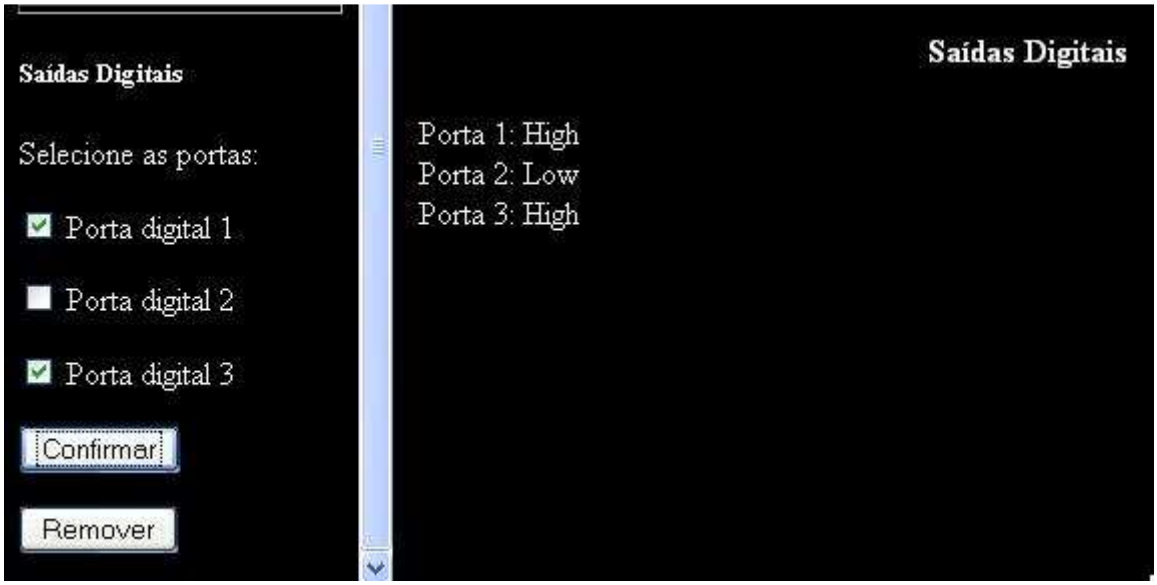

Figura 24 - Detalhe da página sobre as saídas digitais

Adicionalmente, a Figura 25 mostra a tela executando uma tarefa de aquisitar valores de tensão através das entradas analógicas. O usuário seleciona os canais clicando nos elementos da linguagem HTML conhecido como *checkboxes*. Ao executar o clique sobre o botão “Confirmar”, os valores do formulário são enviados através da URL e o *script* verifica quais portas foram solicitadas buscando os valores armazenados na variável QUERY_STRING. A instrução seguinte é buscar o valor de tensão mais recente de acordo com as portas solicitadas. De posse do dado, esta informação é inserida na montagem do HTML através do CGi-bin para atualizar a página.

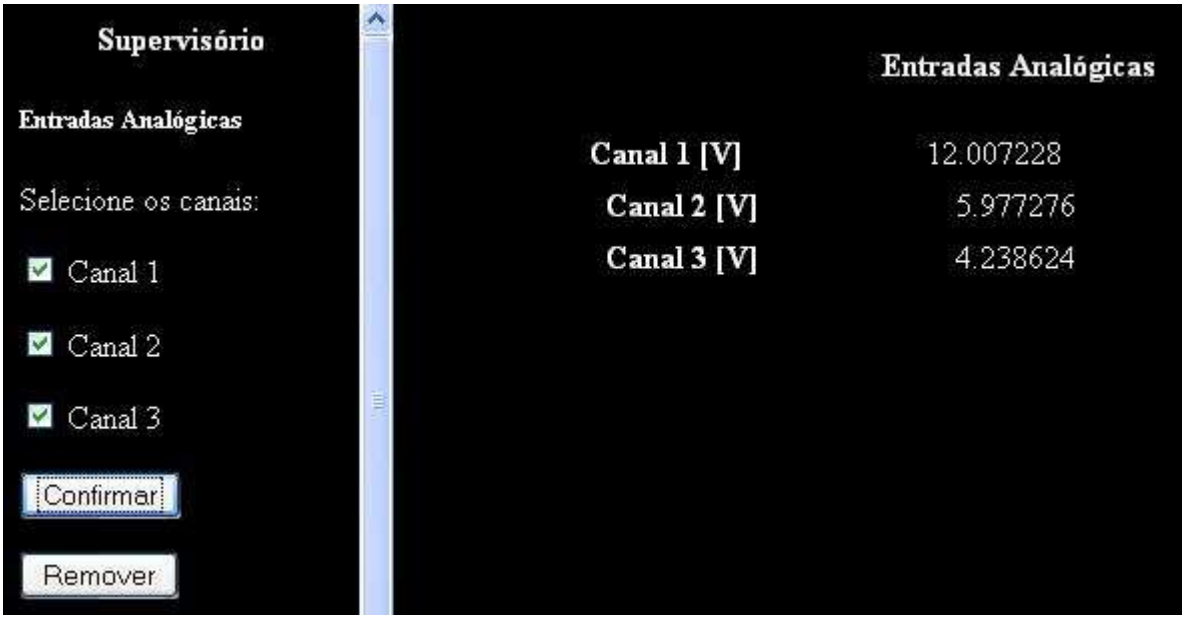

Figura 25 - Detalhe da página sobre as entradas analógicas

4.3. Protegendo através de autenticação

Por tratar de um sistema remoto usando endereços que são alcançáveis por qualquer pessoa, é preciso proteger o acesso através de mecanismos de autenticação do Apache. Primeiro é necessário dizer qual é o arquivo que contém as regras de acesso. Em qualquer ponto do arquivo /etc/apache/apache2.conf deve ter a linha [5]:

```
AccessFileName .htaccess
```

O `.htaccess` pode desempenhar diversas tarefas tais como: proteger diretórios com senha, redirecionamento automático, alteração de extensões de arquivos, bloqueio de usuários com determinados endereços IP, autorização de IPs específicos, impedimento de listagem de diretórios, ativação SSI, usar páginas índice diferentes, adicionar tipos MIME (*Multipurpose Internet Mail Extensions*) entre outras.

No mesmo arquivo há um controle de acesso para que os arquivos não serem baixados e os códigos-fontes explorados. A configuração abaixo impede esta ação:

```
<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>

<Directory /usr/lib/cgi-bin/78/restrito/>
    AllowOverride AuthConfig
</Directory>
```

Uma vez informado que as configurações seguem o arquivo .htaccess, é preciso editar as regras de acesso com o seguinte conteúdo:

```
AuthName "Acesso restrito"
AuthType Basic
AuthUserFile /usr/lib/cgi-bin/78/restrito/.htpasswd
require valid-user
```

De acordo com as linhas acima, pode-se descrever as instruções, Tabela 7:

Tabela 7 - Significado das instruções do arquivo `.htaccess`

Instrução	Descrição
AuthName	Mensagem que aparece na caixa de diálogo solicitando nome de usuário e senha.
AuthType	É tipo de autenticação e neste caso é um HTTP básico.
AuthUserFile	O *path* (caminho) que contém a lista de usuários e senha válidos.
require	Determina que somente usuários válidos tenham acesso ao conteúdo.

O último passo é cadastrar aqueles que têm permissão de acesso. A inclusão destes usuários não tem relação com os que têm uma conta na máquina Linux definidos pelo arquivo /etc/passwd. A finalidade é exclusivamente ter acesso pelo *web Server*. O comando deve ser executado no diretório que se queira proteger e, no caso, os executáveis localizados em /usr/lib/cgi-bin/78/restrito.

```
murilo@fujita2:~$ htpasswd –m –c .htpasswd user
```

A Tabela 8 apresenta a descrição das opções do comando:

Tabela 8 - Significado das opções para criar o arquivo `.htaccess`

Opção	Descrição
-m	Indica que a senha dos usuários é encriptada pelo algoritmo MD5.
-c	Cria o arquivo de senhas. Deve-se observar que esta opção é usada somente no primeiro cadastramento para que todos sejam armazenados em um único arquivo, caso contrário as senhas ficam em arquivos diferentes. Caso necessite modificar a senha de um usuário existente, esta opção deve ser omitida.
user	Deve ser substituído pelo nome autorizado pelo sistema.

O resultado da configuração é mostrado na Figura 26, que exibe uma caixa de diálogo exigindo a autenticação:

Figura 26 - Tela requerendo autenticação para acessar o sistema

5. Conclusões

A proposta do trabalho de dissertação tem como objetivo juntar as ferramentas de supervisório e gerenciamento de rede, com o propósito de monitorar equipamentos conectados em redes. Uma vez estabelecida a meta utilizou-se o protocolo SNMP, o qual tem o potencial para estabelecer comunicações remotas e atender as tarefas de monitoramento.

A arquitetura mostra uma interface amigável que ao disparar os eventos de Supervisão de Processos, executa um programa que faz o agente comunicar com o dispositivo e armazenar o resultado em um arquivo. Desta forma, os *scripts* buscam a informação mais recente e a exibem na tela. Para o caso de Gerenciamento de rede, as informações são desorganizadas e é função dos *scripts* selecionar o que é relevante.

Outra característica relevante do protocolo é verificar que a troca de mensagens é considerada satisfatória. Ao longo do período de desenvolvimento e testes, percebe-se que os eventos atendem à expectativa.

Além disto, um ponto forte da implementação deste protocolo é a facilidade em adquirir uma determinada informação de uma máquina remota. Com um único comando processa-se a autenticação e quais informações deseja-se obter.

Outra vantagem é o custo, pois o meio de comunicação, a internet, é aproveitado para interligar dispositivos. Além disso, a maioria das ferramentas é código-aberto podendo ser adquiridas em sites específicos na rede mundial de computadores.

Também foi verificado que máquinas com configurações modestas têm condições de comunicar-se através do protocolo SNMP. Computadores com essas características abrangem máquinas antigas e embarcadas.

Apesar da forma simples de autenticação, procurou-se cuidados para proteger o sistema. Assim, o invasor tem que conseguir a combinação do nome

da comunidade, IPs das máquinas que têm o protocolo instalado, número da porta o qual o serviço está ativo e a versão do protocolo.

Como comentado, desde que as informações não sejam capturadas através de *sniffers* dentro, a combinação de todos esses parâmetros fornece um ambiente seguro. Para evitar que usuários dentro da rede executem este tipo de programa, regras que negam a utilização de certos softwares podem ser aplicadas.

A desvantagem é que novas funcionalidades requerem a elaboração de um *script* para que desempenhe cada nova tarefa de monitoramento. No entanto, requer uma lógica simples, pois a linguagem *Script Shell* é interpretada e não compilada.

5.1. Trabalhos futuros

A aplicação do protocolo é extensa podendo ser empregada para os mais variados fins de monitoramento. Melhorias significativas são obtidas com o lançamento dos mesmos equipamentos, porém com aplicativos gerenciáveis embutidos facilitando o acompanhamento de dados consultados.

Atualmente a computação abrange uma grande variedade de dispositivos, e um tipo é bem interessante é o embarcado (aparelhos dedicados a realizar uma tarefa específica). Devido ao tamanho reduzido, têm um processador, memória e capacidade de armazenamento (geralmente uma memória *flash*) proporcionais à sua dimensão. Dentro da *flash* encontra-se o *firmware*, o programa que torna o dispositivo embarcado funcional. Os fabricantes lançam constantemente novas versões dos seus produtos e acompanhando a tendência, um recurso adicional é o gerenciamento. Uma sugestão é substituir o firmware de determinados equipamentos não-gerenciáveis por um que tenha embutido o protocolo SNMP.

Além de manter os recursos existentes, é possível recuperar informações como tempo que o equipamento esteve em funcionamento, por exemplo.

No caso de tráfego de dados confidenciais, outro trabalho possível aplicando o protocolo SNMP é empregar a versão 3 (três), que dispõe de mais recursos como autenticação de usuário e criptografia, para consultar dispositivos tornando a rede de computadores consideravelmente mais segura. A versão 3 possui um controle mais granular, ou seja, um critério de controle mais refinado para permitir ou negar acesso de determinados usuários, grupos ou máquinas. Para que o dispositivo consultado responda, um conjunto de parâmetros deve estar de acordo com uma configuração:

- nome do usuário: a configuração de cada usuário deve coincidir com as demais informações a seguir.
- algoritmo de encriptação: MD5 ou SHA, programas que fazem a encriptação ao enviar e desencriptação ao chegar no destino.
- chave pública e privada: AE5 ou DES, programas que conferem se as chaves do remetente e destinatário combinam.
- nível de segurança: as mensagens podem ser encontradas nas seguintes situações: não-autenticadas e não-encriptadas, autenticadas e não-encriptadas ou autenticadas e encriptadas.

Referências Bibliográficas

[1] Mauro, Douglas and Schmidt, Kevin, “The Essential SNMP”, O´Reilly, 2001. ISBN 0-596-00020-0.

[2] Comer, Douglas, “Redes de Computadores e Internet”, Bookman, Volume 1, 2ª edição, 2001. ISBN 0-13-083617-6.

[3] Comer, Douglas, “Interligando Redes TCP/IP”, Volume 2, Bookman. ISBN 85-352-0395-8.

[4] Sommerville, Ian, “Engenharia de Software”, Addison Wesley, 6ª edição, 2003. ISBN 85-88639-07-6.

[5] Neves, Julio Cezar, “Programação Shell Linux”, Brasport, 4ª Edição, 2004. ISBN 85-7452-164-7.

[6] Burtch, Ken O., “Scripts de Shell Linux com Bash”, Ciência Moderna, 2005. ISBN 85-7393-405-0.

[7] Wainwright, Peter, “Professional Apache”, Wrox, 2000. ISBN 1861003021.

[8] Schildt, Herbet, “C Completo e Total”, McGrawHill, 1990. ISBN 853460595-5.

[9] Proença, Adriano, “Manufatura Integrada por Computador: Sistemas Integrados de Produção”, Campus, 1995. ISBN 8570019629.

[10] Disponível em <http://www.spamhaus.org>, Acesso em 16 junho 2008.

[11] Disponível em <http://aurelio.net/cygwin>, Acesso em 20 maio 2008.

[12] Disponível em <http://www.net-snmp.net>, Acesso em 30 novembro 2007.

[13] Cisco Systems, “Cisco Networking Academies”, Semestre 1.

[14] Wenhui, Sun; Feng, Liu; Honghui, Li; Xudong, Chen; “Implementing a CORBA/SNMP gateway with design patterns”, 2003.

[15] Teegan , Hugh A., “Distributed performance monitoring using SNMP V2”. IEEE (*Institute of Electrical and Electronics Engineers*), 1996.

[16] http://asn1.elibel.tm.fr/en/introduction/index.htm, 17 fevereiro 2008.

[17] Drake , Peter. “Using SNMP to manage networks”. IEEE (*Institute of Electrical and Electronics Engineers*),1991.

[18] Case, Jeffrey D.; Davin , James R.; Fedor , Mark S.; Schoffstall , Martin L; ”Internet network management using the simple network management protocol”. IEEE (*Institute of Electrical and Electronics Engineers*), 1989.

[19] RFC 1157 – Simple Network Management Protocol (SNMP).

[20] Gumudavelli, S., Gurkan, D., Hussain, A., Wang, R. “A Network Management Approach for Implementing the Smart Sensor Plug and Play”, IEEE 2009.

[21] Jung, S., Lee, J., Han, Y., Kim, J., Na, J., Chung, T. “SNMP-based Integrated Wire/wireless Device Management System”, IEEE, 2007.

[22] Lee, J., Cho, W., An, J., Yoo, D., Lee, K. "A Study on the Advanced PLC System using the MIB and SNMP", IEEE, 2006.

[23] Janeiro, J., Liebing, C. “Developing Management Applications based on SNMP”, IEEE, 2010.

Anexos

A. Código-fonte dos arquivos binários

Encontram-se a seguir os códigos-fonte desenvolvidos em linguagem C [8] específicos para a placa de aquisição de dados e os *Shell scripts* que atualizam dados na interface.

A.1. Binário da placa de aquisição de dados USB 6008 para entradas analógicas

```
#include "stdafx.h"
#include <stdio.h>
#include <iostream>
#include "D:\test\allchannels\NIDAQmx.h"

#define              DAQmxErrChk(functionCall)              if(
DAQmxFailed(error=(functionCall)) ) goto Error; else

void channel1 (int i, int32 error,TaskHandle taskHandle,int32
read, float64 data[], char errBuff[]);
void channel2 (int i, int32 error,TaskHandle taskHandle,int32
read, float64 data[], char errBuff[]);
void channel3 (int i, int32 error,TaskHandle taskHandle,int32
read, float64 data[], char errBuff[]);
int main(void)
{
inti=0;
int32       error=0;
TaskHandle  taskHandle=0;
int32       read=0;
float64     data[1000];
char        errBuff[2048]={'\0'};

   for (i=0; i>-1; i++)
   {
   channel1 (i,error,taskHandle,read, data, errBuff);
   channel2 (i,error,taskHandle,read, data, errBuff);
   channel3 (i,error,taskHandle,read, data, errBuff);
   }
}
```

```
void channel1 (int i, int32 error,TaskHandle taskHandle,int32
read, float64 data[], char errBuff[])
{
FILE*p1;

p1=fopen("voltage1.txt","a+");
    if (p1 == NULL)
    {
printf("Erro no arquivo voltage1.txt\n");
}

 /*********************************************/
// DAQmx Configure Code
/*********************************************/
DAQmxErrChk (DAQmxCreateTask("",&taskHandle));
DAQmxErrChk
(DAQmxCreateAIVoltageChan(taskHandle,"Dev1/ai0","",DAQmx_Val_C
fg_Default,-13.0,13.0,DAQmx_Val_Volts,NULL));
DAQmxErrChk
(DAQmxCfgSampClkTiming(taskHandle,"",10000.0,DAQmx_Val_Rising,
DAQmx_Val_FiniteSamps,1000));

/*********************************************/
// DAQmx Start Code
/*********************************************/

DAQmxErrChk (DAQmxStartTask(taskHandle));

/*********************************************/
// DAQmx Read Code
/*********************************************/
DAQmxErrChk
(DAQmxReadAnalogF64(taskHandle,100,10.0,DAQmx_Val_GroupByChann
el,data,300,&read,NULL));

for (i=1;i<2;i++)
{
printf("1   %f\n",data[99]);
fprintf(p1,"%f\n",data[99]);
_sleep(1000);
}

Error:
if( DAQmxFailed(error) )
DAQmxGetExtendedErrorInfo(errBuff,2048);
if( taskHandle!=0 )  {
/*********************************************/
// DAQmx Stop Code
/*********************************************/
DAQmxStopTask(taskHandle);
DAQmxClearTask(taskHandle);
```

```
}

fclose(p1);

if( DAQmxFailed(error) )
printf("DAQmx Error: %s\n",errBuff);
}

void channel2 (int i, int32 error,TaskHandle taskHandle,int32
read, float64 data[], char errBuff[])
{
FILE*p2;

p2=fopen("voltage2.txt","a+");
    if (p2 == NULL)
    {
printf("Erro no arquivo voltage1.txt\n");
}

 /*********************************************/
// DAQmx Configure Code
/*********************************************/
DAQmxErrChk (DAQmxCreateTask("",&taskHandle));
DAQmxErrChk
(DAQmxCreateAIVoltageChan(taskHandle,"Dev1/ai1","",DAQmx_Val_C
fg_Default,-13.0,13.0,DAQmx_Val_Volts,NULL));
DAQmxErrChk
(DAQmxCfgSampClkTiming(taskHandle,"",10000.0,DAQmx_Val_Rising,
DAQmx_Val_FiniteSamps,1000));

/*********************************************/
// DAQmx Start Code
/*********************************************/

DAQmxErrChk (DAQmxStartTask(taskHandle));

/*********************************************/
// DAQmx Read Code
/*********************************************/
DAQmxErrChk
(DAQmxReadAnalogF64(taskHandle,100,10.0,DAQmx_Val_GroupByChann
el,data,300,&read,NULL));

for (i=1;i<2;i++)
{
printf("2   %f\n",data[99]);
fprintf(p2,"%f\n",data[99]);
_sleep(1000);
}
```

```
Error:
if( DAQmxFailed(error) )
DAQmxGetExtendedErrorInfo(errBuff,2048);
if( taskHandle!=0 )  {
/*********************************************/
// DAQmx Stop Code
/*********************************************/
DAQmxStopTask(taskHandle);
DAQmxClearTask(taskHandle);
}

fclose(p2);

if( DAQmxFailed(error) )
printf("DAQmx Error: %s\n",errBuff);
}


void channel3 (int i, int32 error,TaskHandle taskHandle,int32
read, float64 data[], char errBuff[])
{
FILE*p3;

p3=fopen("voltage3.txt","a+");
    if (p3 == NULL)
    {
printf("Erro no arquivo voltage3.txt\n");
}

/*********************************************/
// DAQmx Configure Code
/*********************************************/
DAQmxErrChk (DAQmxCreateTask("",&taskHandle));
DAQmxErrChk
(DAQmxCreateAIVoltageChan(taskHandle,"Dev1/ai2","",DAQmx_Val_C
fg_Default,-13.0,13.0,DAQmx_Val_Volts,NULL));
DAQmxErrChk
(DAQmxCfgSampClkTiming(taskHandle,"",10000.0,DAQmx_Val_Rising,
DAQmx_Val_FiniteSamps,1000));

/*********************************************/
// DAQmx Start Code
/*********************************************/

DAQmxErrChk (DAQmxStartTask(taskHandle));

/*********************************************/
// DAQmx Read Code
/*********************************************/
```

```
DAQmxErrChk
(DAQmxReadAnalogF64(taskHandle,100,10.0,DAQmx_Val_GroupByChann
el,data,300,&read,NULL));

for (i=1;i<2;i++)
{
printf("3   %f\n",data[99]);
fprintf(p3,"%f\n",data[99]);
_sleep(1000);
}


Error:
if( DAQmxFailed(error) )
DAQmxGetExtendedErrorInfo(errBuff,2048);
if( taskHandle!=0 )  {
/*********************************************/
// DAQmx Stop Code
/*********************************************/
DAQmxStopTask(taskHandle);
DAQmxClearTask(taskHandle);
}

fclose(p3);

if( DAQmxFailed(error) )
printf("DAQmx Error: %s\n",errBuff);
}
```

A.2. Binário da placa de aquisição de dados USB 6008 para entradas digitais

```
#include "stdafx.h"
#include <iostream>
#include <stdio.h>
#include "D:\test\digital\NIDAQmx.h"

#define             DAQmxErrChk(functionCall)             if(
DAQmxFailed(error=(functionCall)) ) goto Error; else

int main(void)
{
int32error=0;
TaskHandletaskHandle=0;
uInt32data;
charerrBuff[2048]={'\0'};
int32read;
FILE*pw;
```

```
begin:

pw=fopen("digital.txt","a+");
if (!pw)
{
   printf(">>> digital.txt ERRO <<<\n");
   exit(1);
}

/*********************************************/
// DAQmx Configure Code
/*********************************************/
DAQmxErrChk (DAQmxCreateTask("",&taskHandle));
DAQmxErrChk
(DAQmxCreateDIChan(taskHandle,"Dev1/port0","",DAQmx_Val_ChanFo
rAllLines));

/*********************************************/
// DAQmx Start Code
/*********************************************/
DAQmxErrChk (DAQmxStartTask(taskHandle));

/*********************************************/
// DAQmx Read Code
/*********************************************/
DAQmxErrChk
(DAQmxReadDigitalU32(taskHandle,1,10.0,DAQmx_Val_GroupByChanne
l,&data,1,&read,NULL));
system("cls");
printf("Bit:   543210\n");
printf("Valor: ");

for(int i=5 ;i>=0 ;i--)
{
printf("%d",((data&(1<<i))!=0));
fprintf(pw,"%d",((data&(1<<i))!=0));

}
fprintf(pw,"\n");
_sleep(1000);

Error:
if( DAQmxFailed(error) )
DAQmxGetExtendedErrorInfo(errBuff,2048);
if( taskHandle!=0 ) {
/*********************************************/
// DAQmx Stop Code
/*********************************************/
DAQmxStopTask(taskHandle);
DAQmxClearTask(taskHandle);
```

```
}

fclose(pw);

goto begin;

if( DAQmxFailed(error) )
printf("DAQmx Error: %s\n",errBuff);
//printf("End of program, press Enter key to quit\n");
getchar();
return 0;
}
```

A.3. Binário da placa de aquisição de dados USB 6008 para saídas analógicas

```
#include "stdafx.h"
#include <stdio.h>
#include <math.h>
#include <iostream>
#include "D:\test\output\NIDAQmx.h"
#if defined(WIN32) || defined(_WIN32)

#endif

#define              DAQmxErrChk(functionCall)              if(
DAQmxFailed(error=(functionCall)) ) goto Error; else

#define PI3.1415926535

int main(int argv, char *argc[])
{
int         error=0;
TaskHandle  taskHandle=0;
float64     data[1000];
char        errBuff[2048]={'\0'};
uInt32      i=0;

if (argv != 2)
{
printf("Valor nao informado\n");
exit(1);
}

for(; i<10; i++)
data[i] = atof(argc[1])*sin(PI/2);
i=0;
```

```
/*********************************************/
// DAQmx Configure Code
/*********************************************/
DAQmxErrChk (DAQmxCreateTask("",&taskHandle));
DAQmxErrChk
(DAQmxCreateAOVoltageChan(taskHandle,"Dev1/ao1","",0.0,5.0,DAQ
mx_Val_Volts,NULL));

/*********************************************/
// DAQmx Start Code
/*********************************************/
DAQmxErrChk (DAQmxStartTask(taskHandle));

printf("Generating samples continuously. Press Ctrl+C to
interrupt\n");
while( 1 ) {
#if defined(WIN32) || defined(_WIN32)
_sleep(1000);
printf (" %4.2f\n", data[i]);
#else
#error - This example requires a platform specific sleep call.
/*********************************************/
// This example requires a platform specific sleep call.
// For example:
//
// #include <windows.h>
// Sleep(1);// For Windows platform
//
// #include <unistd.h>
// usleep(1000);// For Linux platform
/*********************************************/
#endif

/*********************************************/
// DAQmx Write Code
/*********************************************/
DAQmxErrChk
(DAQmxWriteAnalogScalarF64(taskHandle,1,10.0,data[i],NULL));
if( ++i>=500 )
i = 0;
}

Error:
if( DAQmxFailed(error) )
DAQmxGetExtendedErrorInfo(errBuff,2048);
if( taskHandle!=0 ) {
/*********************************************/
// DAQmx Stop Code
/*********************************************/
DAQmxStopTask(taskHandle);
DAQmxClearTask(taskHandle);
```

```
}
if( DAQmxFailed(error) )
printf("DAQmx Error: %s\n",errBuff);
printf("End of program, press Enter key to quit\n");
getchar();
return 0;
}
```

A.4. Binário da placa de aquisição de dados USB 6008 para saídas digitais

```
#include "stdafx.h"
#include <stdio.h>
#include <iostream>
#include "D:\test\digital_output\nidaqmx.h"

#define              DAQmxErrChk(functionCall)              if(
DAQmxFailed(error=(functionCall)) ) goto Error; else


int main(int argc, char *argv[])
{
int         error=0;
TaskHandletaskHandle=0;
uInt32      data=0;
char        errBuff[2048]={'\0'};
int32written;
intdadosSaida=0;

if (argc != 2)
{
printf ("Digite um valor entre 0 e 7");
exit(1);
}

if (argc == 2)
{
dadosSaida = (int) *argv[1];
printf("%d\n",dadosSaida);
dadosSaida -= 48;
printf("%d\n",dadosSaida);
data = (uInt32)dadosSaida;
}



/*********************************************/
// DAQmx Configure Code
```

```
/*********************************************/
DAQmxErrChk (DAQmxCreateTask("",&taskHandle));
DAQmxErrChk
(DAQmxCreateDOChan(taskHandle,"Dev1/port1","",DAQmx_Val_ChanFo
rAllLines));

/*********************************************/
// DAQmx Start Code
/*********************************************/
DAQmxErrChk (DAQmxStartTask(taskHandle));

/*********************************************/
// DAQmx Write Code
/*********************************************/
DAQmxErrChk
(DAQmxWriteDigitalU32(taskHandle,1,1,10.0,DAQmx_Val_GroupByCha
nnel,&data,&written,NULL));

Error:
if( DAQmxFailed(error) )
DAQmxGetExtendedErrorInfo(errBuff,2048);
if( taskHandle!=0 ) {
/*********************************************/
// DAQmx Stop Code
/*********************************************/
DAQmxStopTask(taskHandle);
DAQmxClearTask(taskHandle);
}
if( DAQmxFailed(error) )
printf("DAQmx Error: %s\n",errBuff);
return 0;
}
```

B.Scripts de gerenciamento de rede

B.1. Código-fonte do GEF

```
#!/bin/bash
#GEF = Gerenciamento Em Funcionamento

echo "content-type: text/html"
echo
echo
echo "
<HTML> <HEAD> <BODY BGCOLOR="black" TEXT="white"> </HEAD>
```

```
<H4><div  align="center">  Tempo  de  funcionamento  do  agente
10.10.8.78 </div></H4>"

n1=`snmpwalk -v 1 -c fujitaro 10.10.8.78 1.3.6.1.4.1.30778.6 |
tail -n 1 | cut -c 79-105 | awk '{printf "%s\n",$1}'`
n2=`snmpwalk -v 1 -c fujitaro 10.10.8.78 1.3.6.1.4.1.30778.6 |
tail -n 1 | cut -c 79-105 | awk '{printf "%s\n",$2}'`
n3=`snmpwalk -v 1 -c fujitaro 10.10.8.78 1.3.6.1.4.1.30778.6 |
tail  -n  1  |  cut  -c  79-105  |  awk  '{printf  "%s\n",$3}'  |  sed
's/,//'`
n4=`snmpwalk -v 1 -c fujitaro 10.10.8.78 1.3.6.1.4.1.30778.6 |
tail  -n  1  |  cut  -c  79-105  |  awk  '{printf  "%s\n",$4}'  |  sed
's/,//'`
n5=`snmpwalk -v 1 -c fujitaro 10.10.8.78 1.3.6.1.4.1.30778.6 |
tail  -n  1  |  cut  -c  79-105  |  awk  '{printf  "%s\n",$5}'  |  sed
's/,//'`
n6=`snmpwalk -v 1 -c fujitaro 10.10.8.78 1.3.6.1.4.1.30778.6 |
tail -n 1 | cut -c 79-105 | awk '{printf "%s\n",$6}'`
n7=`snmpwalk -v 1 -c fujitaro 10.10.8.78 1.3.6.1.4.1.30778.6 |
tail -n 1 | cut -c 79-105 | awk '{printf "%s\n",$5}' | cut -c
2`

if [ "$n4" = day ]
  then
    if [ "$n6" = min, ]
        then
           echo "<div align="center"><strong> $n3" dia 0:"$n5
</strong><div>"
    fi

    if [ "$n7" = : ]
        then
          echo  "<div  align="center"><strong>  $n3"  dia  "$n5
</strong></div>"
    fi
fi

if [ "$n4" = days ]
  then
    if [ "$n6" = min, ]
        then
          echo "<div align="center"><strong> $n3" dias 0:"$n5
</strong></div>"
    fi

    if [ "$n7" = : ]
        then
          echo  "<div  align="center"><strong>  $n3"  dias  "$n5
</strong></div>"
    fi
```

```
fi

if [ "$n5" = user -o users ]
  then

checa2pontos1=`echo "$n3" | cut -c 2`
checa2pontos2=`echo "$n3" | cut -c 3`

    if [ "$checa2pontos1" = :  ]
      then
        echo   "<div   align="center"><strong>   0   dia   "$n3"
</strong></div>"
    fi

    if [ "$checa2pontos2" = :  ]
      then
        echo   "<div   align="center"><strong>   0   dia   "$n3"
</strong></div>"
    fi

    if [ "$n4" = min ]
      then
        echo  "<div  align="center"><strong>  0  dia  0:"$n3"
</strong></div>"
    fi
fi

echo "</BODY>"
echo "</HTML>"
```

B.2. Código-fonte do GLog

```
#!/bin/bash
#GLog = Gerenciamento de Logs

echo "content-type: text/html"
echo
echo
echo "
<HTML> <HEAD> <BODY BGCOLOR="black" TEXT="white"> </HEAD>
"

total_linhas=`wc  -l  /var/log/snmptrapd.log  |  awk  '{printf
"%s\n",$1}'`
let as10ultimas_linhas=$total_linhas-10

for i in `seq $as10ultimas_linhas 1 $total_linhas`
  do
```

```
    head -n $i /var/log/snmptrapd.log | tail -n 1 | grep -v
netsnmp_assert
    echo "<br>"
  done

echo                 "<meta                 http-equiv="Refresh"
content="5;URL=http://10.10.8.70/cgi-bin/status.cgi">"

echo "</BODY>"
echo "</HTML>"
```

B.3. Código-fonte do GPA

```
#!/bin/bash
#GPA = Gerenciamento de Portas Ativas

echo "content-type: text/html"
echo
echo
echo "
<HTML> <HEAD> <BODY BGCOLOR="black" text="white"> </HEAD>

<H3><div align="center"> Portas ativas do agente 10.10.8.78
</div></align></H3>

<H4> Portas TCP </H4>

<pre>"

snmpwalk       -v       1       -c       fujitaro       fujita
.1.3.6.1.4.1.30778.5.3.1.2.7.47.98.105.110.47.115.104   |   awk
'{printf  "%5s\n",$2}'  |  tr  '[:punct:]'  '  '  |  awk  '{printf
"%s\n",$5}'

echo "</pre> <H4> Portas UDP </H4> <pre>"


snmpwalk       -v       1       -c       fujitaro       fujita
1.3.6.1.4.1.30778.12.3.1.2.7.47.98.105.110.47.115.104   |   sed
's/SNMPv2-
SMI::enterprises.30778.12.3.1.2.7.47.98.105.110.47.115.104    =
STRING: "//' | awk '{printf "%5s\n",$2}' | tr '[:punct:]' ' '
| awk '{printf "%s\n",$5}'

echo "</pre>"
echo "</BODY>"
echo "</HTML>"
```

B.4. Código-fonte do GS

```
#!/bin/bash
#Gerenciamento de Sistema

echo "content-type: text/html"
echo
echo "
<HTML> <HEAD> <BODY BGCOLOR="black" TEXT="white">
<H4><div  align="center">  Informações  do  sistema  -  Agente
10.10.8.78 </div></H4>
"

echo "<table>"
echo "<tr>"
echo "<td>Descrição</td>"
echo "<td>"
snmpwalk -v 1 -c fujitaro fujita  sysDescr | cut -c 43-
echo "</td>"
echo "</tr>"

echo "<tr>"
echo "<td>Contato</td>"
echo "<td>"
snmpwalk -v 1 -c fujitaro fujita  sysContact | cut -c 35-
echo "</td>"

echo "<tr>"
echo "<td>Nome do agente</td>"
echo "<td>"
snmpwalk -v 1 -c fujitaro fujita  sysName | cut -c 33-
echo "</td>"
echo "</td>"

echo "<tr>"
echo "<td>Localização</td>"
echo "<td>"
snmpwalk -v 1 -c fujitaro fujita  sysLocation | cut -c 36-
echo "</td>"
echo "</tr>"
echo "</table>"

echo "<pre>"
echo
echo
echo
echo
echo
echo "</pre>"
```

```
echo "
<form method="get" action="/cgi-bin/78/GS_contact.cgi">
<h4><div align="center"> Modificar Contato </div></h4>
<input type="text" NAME=contato SIZE="50">
<input type="submit" value="Modificar">
</form>
"

echo "
<form method="get" action="/cgi-bin/78/GS_location.cgi">
<h4><div align="center"> Modificar Local </div></h4>
<input type="text" NAME=local SIZE="50">
<input type="submit" value="Modificar">
</form>
"

echo "</BODY>"
echo "</HTML>"
```

B.5. Código-fonte do GS_contact

```
#!/bin/bash
#GS_contact = Gerenciamento de Sistema (mudar contato)

echo "content-type: text/html"
echo
echo "
<BODY> <HEAD> <BODY bgcolor="black" text="white"> </HEAD>
"

snmpset -v 1 -c fujitarw fujita sysContact.0 s "`echo
$QUERY_STRING | cut -c 9- | sed 's/%40/@/'`"

echo "<div align="center"><H3>Contato modificado com sucesso
</H3></div>"
echo "<meta http-equiv="refresh"
CONTENT="URL=http://10.10.8.70/cgi-bin/system.cgi">"

echo "</BODY>"
echo "</HTML>"
```

B.6. Código-fonte do GS_location

```
#!/bin/bash
#GS_location = Gerenciamento de Sistema (mudar localizacao)
```

```
echo "content-type: text/html"
echo
echo "
<BODY> <HEAD> <BODY bgcolor="black" text="white"> </HEAD>
"

snmpset -v 1 -c fujitarw fujita sysLocation.0 s "`echo
$QUERY_STRING | cut -c 7- `"
answer=`head -n 1 | awk '{printf "%s\n",$1}'`
#answer=`echo $QUERY_STRING`
echo "<br>"
echo "$answer"

#if [ "$answer" = "SNMPv2-MIB::sysLocation.0" ]
#
#  then
#       echo "<div align="center"><H3>Local modificado com
sucesso </H3></div>"
#  else
#    echo "<div align="center"><H3> Erro! Local nao modificado
</H3></div>"
#fi

#echo               "<meta               http-equiv="refresh"
CONTENT="URL=http://10.10.8.70/cgi-bin/system.cgi">"

echo "</BODY>"
echo "</HTML>"
```

B.7. Código-fonte do GC

```
#!/bin/bash
#GC = Gerenciamento de Conectividade

echo "content-type: text/html"
echo
echo
echo "
<HTML> <HEAD> <BODY BGCOLOR="black" text="white"> </HEAD>

<H4><div align="center"> Dados sobre a conectividade do agente
10.10.8.78 </div></H4>"

echo "<table>"
echo "<tr>"
echo "<td> IP </td>"
echo "<td>"
```

```
snmpwalk -v 1 -c fujitaro fujita 1.3.6.1.4.1.30778.8.3.1.2.7.47.98.105.110.47.115.104 | sed -n '/Ethernet adapter Local Area Connection/{x;p;x;N;N;N;N;N;p;q;};1!{H;g;};1,0!s/[^\n]*\n//;h;' | grep "IP Address" | cut -c 45-
echo "</td>"
echo "</tr>"

echo "<tr>"
echo "<td> Mascara de rede </td>"
echo "<td>"
snmpwalk -v 1 -c fujitaro fujita 1.3.6.1.4.1.30778.8.3.1.2.7.47.98.105.110.47.115.104 | sed -n '/Ethernet adapter Local Area Connection/{x;p;x;N;N;N;N;N;N;p;q;};1!{H;g;};1,0!s/[^\n]*\n//;h;' | grep "Subnet Mask" | cut -c 45-
echo "</td>"
echo "</tr>"

echo "<tr>"
echo "<td> Gateway </td>"
echo "<td>"
snmpwalk -v 1 -c fujitaro fujita 1.3.6.1.4.1.30778.8.3.1.2.7.47.98.105.110.47.115.104 | sed -n '/Ethernet adapter Local Area Connection/{x;p;x;N;N;N;N;N;N;N;p;q;};1!{H;g;};1,0!s/[^\n]*\n//;h;' | grep "Default Gateway" | cut -c 45-
echo "</td>"
echo "</tr>"

echo "<tr>"
echo "<td> MAC </td>"
echo "<td>"
snmpwalk -v 1 -c fujitaro fujita 1.3.6.1.4.1.30778.8.3.1.2.7.47.98.105.110.47.115.104 | sed -n '/Ethernet adapter Local Area Connection/{x;p;x;N;N;N;N;N;N;N;p;q;};1!{H;g;};1,0!s/[^\n]*\n//;h;' | grep "Physical Address" | cut -c 45-
echo "</td>"
echo "</tr>"

echo "<tr>"
echo "<td> Servidores DNS </td>"
echo "<td>"
snmpwalk -v 1 -c fujitaro fujita 1.3.6.1.4.1.30778.8.3.1.2.7.47.98.105.110.47.115.104 | sed -n '/Ethernet adapter Local Area Connection/{x;p;x;N;N;N;N;N;N;N;N;p;q;};1!{H;g;};1,0!s/[^\n]*\n//;h;' | sed -n '/DNS Servers/{N;p;}' | cut -c 45- | sed 's/"//g' | head -n 1
echo "<br>"
```

```
snmpwalk -v 1 -c fujitaro fujita 1.3.6.1.4.1.30778.8.3.1.2.7.47.98.105.110.47.115.104 | sed -n '/Ethernet adapter Local Area Connection/{x;p;x;N;N;N;N;N;N;N;N;p;q;};1!{H;g;};1,0!s/[^\n]*\n//;h;' | sed -n '/DNS Servers/{N;p;}' | cut -c 45- | sed 's/"//g' | head -n 2 | tail -n 1
echo "</td>"
echo "</tr>"


echo "</table>"

echo "</BODY>"
echo "</HTML>"
```

B.8. Código-fonte do GV

```
#!/bin/bash
#GV = Gerenciamento de Volumes

echo "content-type: text/html"
echo
echo
echo "
<HTML> <HEAD> <BODY BGCOLOR="black" TEXT="white"> </HEAD>
"

echo "<div align="center"><H3> Partições do agente 10.10.8.78 </H3></div>"

echo "<pre>"

snmpwalk -v 1 -c fujitaro fujita 1.3.6.1.4.1.30778.9.3.1.2.7.47.98.105.110.47.115.104 | grep cygdrive | cut -c 3- | sed 's/"//g' | sed 's/\/cygdrive\///g' | sed 's/system,//g' | awk 'BEGIN {print"Unid Tipo Total Ocupado Livre %Ocupado\n"} {printf"%3s: %10s %7s %9s %7s %9s\n",$6,$1,$2,$3,$4,$5}'

echo "</pre>"
echo "</BODY>"
echo "</HTML>"
```

B.9. Código-fonte do GDA

```
#!/bin/bash
```

```
#GDA = Gerenciamento de Dados do Agente

echo "content-type: text/html"
echo
echo "<HTML> <HEAD> <BODY bgcolor="black" text="white">
</HEAD>"
echo "<H3><div align="center"> Dados do agente 10.10.8.78
</div></H3>"
echo "<pre>"

snmpwalk -v 1 -c fujitaro fujita
1.3.6.1.4.1.30778.10.3.1.2.7.47.98.105.110.47.115.104 | grep -
v STRING

echo "</pre>"
echo "</BODY>"
echo "</HTML>"
```

B.10. Código-fonte do GLA

```
#!/bin/bash
#GLA = Gerenciamento de Ligar Agente

echo "content-type: text/html"
echo
echo "<BODY BGCOLOR="black" text="white">"
echo
echo "<div align="center"><H3>Ligando Agente</H3></div>"
echo "<pre>"
echo "<p>"
echo "</pre>"
echo "<div align="right"><H4> Aguarde ... </h4></div>"

/usr/bin/wakeonlan 00:0C:6E:FB:9F:AB > /dev/null

while true
do
answer=`ping -c 1 fujita | head -n 2 | tail -n 1 | awk
'{printf "%s\n",$3}'`
  if [ "$answer" = "from" ]
    then
        echo "<pre>"
        echo "<p>"
        echo "</pre>"
        echo  "<H3><div align="center"> Agente respondendo
</div></H3>"
        exit
  else
        echo "<H3> Sem comunicação com o agente </H3>"
```

```
        sleep 2
        continue
  fi
done

echo "</BODY>"
echo "</HTML>"
```

C.Scripts do supervisório

C.1. Código-fonte do SEA

```
#!/bin/bash
# SEA - Supervisorio Entradas Analogicas

echo "content-type: text/html"
echo
echo "
<HTML> <HEAD> <BODY BGCOLOR="black" text="white">
<div align="center"><H4> Entradas Analógicas </div></h4>"

seleciona_canais=`echo $QUERY_STRING | sed 's/canal=//g'`
separa_canais=`echo $seleciona_canais | tr "&" "\n"`
valor1=`echo $separa_canais | awk '{printf "%s\n",$1}'`
valor2=`echo $separa_canais | awk '{printf "%s\n",$2}'`
valor3=`echo $separa_canais | awk '{printf "%s\n",$3}'`

function funcao1
{
echo "<table border=0 width="80%">"
echo "<tr>"
echo "<th>Canal 1 [V]</th>"
echo "<td>"
snmpwalk -v 1 -c fujitarw fujita 1.3.6.1.4.1.30778.2 | tail -n
1 | awk '{printf "%9s\n",$4}' | sed 's/"//g'
echo "</td>"
echo "</tr>"
echo "</table>"
}

function funcao2
{
echo "<table border=0 width="80%">"
echo "<tr>"
echo "<th>Canal 2 [V]</th>"
echo "<td>"
```

```
snmpwalk -v 1 -c fujitarw fujita 1.3.6.1.4.1.30778.3 | tail -n
1 | awk '{printf "%9s\n",$4}' | sed 's/"//g'
echo "</td>"
echo "</tr>"
echo "</table>"
}

function funcao3
{
echo "<table border=0 width="80%">"
echo "<tr>"
echo "<th>Canal 3 [V]</th>"
echo "<td>"
snmpwalk -v 1 -c fujitarw fujita 1.3.6.1.4.1.30778.4 | tail -n
1 | awk '{printf "%9s\n",$4}' | sed 's/"//g'
echo "</td>"
echo "</tr>"
echo "</table>"
}

case $valor1 in
  1) funcao1;;
  2) funcao2;;
  3) funcao3;;
esac

case $valor2 in
  1) funcao1;;
  2) funcao2;;
  3) funcao3;;
esac

case $valor3 in
  1) funcao1;;
  2) funcao2;;
  3) funcao3;;
esac

echo                "<meta                http-equiv="refresh"
content="2;URL=http://10.10.8.70/cgi-bin/daq.cgi">"
echo "</BODY>"
echo "</HTML>"
```

C.2. Código-fonte do SED

```
#!/bin/bash
# SED - Supervisorio Entradas Digitais

echo "content-type: text/html"
```

```
echo
echo
echo "
<HTML> <HEAD> <BODY bgcolor="black" text="white"> </HEAD>

<H1><div   align="center">   Valores   Digitais   das   chaves
</div></H1>

<pre>



</pre>"

echo "<H1><div align="center">"
snmpwalk       -v       1       -c       fujitaro       fujita
1.3.6.1.4.1.30778.11.4.1.2.7.47.98.105.110.47.115.104.1  |  cut
-c 80- | sed 's/"//'
echo "</div></H1>"

echo                 "<meta                http-equiv="refresh"
content="3;URL=http://10.10.8.70/cgi-bin/SVD.cgi">"

echo "</BODY>"
echo "</HTML>"
```

C.3. Código-fonte do SSA

```
#!/bin/bash
# SSA - Supervisorio Saida Analógicas

echo "content-type: text/html"
echo
echo
echo "
<BODY> <HEAD> <BODY bgcolor="black" text="white"> </HEAD>
"

rm -fR /media/windows/output.bat
 voltage2=`echo $QUERY_STRING | cut -c 9- | sed 's/%2C/./'`

echo "<div align="center"> <H3> Saída Analógicas </div></H3>"

testa_numeros_fracionarios_positivo=`echo  "$voltage2  >  5"  |
bc`

if [ $testa_numeros_fracionarios_positivo -eq 1 ]
```

```
  then
        voltage2=5
        echo    "<div    align="center"><H3>    Valor    máximo
</H3></div>"
fi

testa_numeros_fracionarios_negativo=`echo  "$voltage2  <  0"  |
bc`

if [ $testa_numeros_fracionarios_negativo -eq 1 ]
  then
        voltage2=0
        echo    "<div    align="center"><H3>    Valor    mínimo
</H3></div>"
fi

echo "<div align="center"> <H3>"
echo "$voltage2"
echo "V </div></H3>"
printf    "@echo    off\nD:\log\output.exe    "$voltage2""    >
/media/windows/output.bat
snmpwalk  -v  1  -c  fujitaro  fujita  1.3.6.1.4.1.30778.14  >
/dev/null

echo "</BODY>"
echo "</HTML>"
```

C.4. Código-fonte do SSD

```
#!/bin/bash
# SSD - Supervisorio Saidas Digitais

declare -i valor1
declare -i valor2
declare -i valor3
declare -i argumentos=0

echo "content-type: text/html"
echo
echo "
<HTML> <HEAD> <BODY BGCOLOR="black" text="white">
<div align="center"><H4> Saídas Digitais </div></h4>"

seleciona_canais=`echo        $QUERY_STRING        |        sed
's/saida_digital=//g'`
separa_canais=`echo $seleciona_canais | tr "&" "\n"`
```

```
valor1=`echo $separa_canais  | awk '{printf "%s\n",$1}'`
valor2=`echo $separa_canais  | awk '{printf "%s\n",$2}'`
valor3=`echo $separa_canais  | awk '{printf "%s\n",$3}'`
  if [ $valor1 != 0 ]
    then
        argumentos+=1
  fi
  if [ $valor2 != 0 ]
    then
        argumentos+=1
  fi
  if [ $valor3 != 0 ]
    then
        argumentos+=1
  fi

function funcao3 {
                valor_final1=1
                valor_final2=1
                valor_final3=1
                echo "Porta 1: High<br>"
                echo "Porta 2: High<br>"
                echo "Porta 3: High<br>"
}

function funcao2 {
        if [ $valor1 -eq 1 -a $valor2 -eq 2 ]
           then
                valor_final1=1
                valor_final2=1
                valor_final3=0
                echo "Porta 1: High<br>"
                echo "Porta 2: High<br>"
                echo "Porta 3: Low<br>"
        fi
        if [ $valor1 -eq 1 -a $valor2 -eq 3 ]
           then
                valor_final1=1
                valor_final2=0
                valor_final3=1
                echo "Porta 1: High<br>"
                echo "Porta 2: Low<br>"
                echo "Porta 3: High<br>"
        fi
        if [ $valor1 -eq 2 -a $valor2 -eq 3 ]
           then
                valor_final1=0
                valor_final2=1
                valor_final3=1
                echo "Porta 1: Low<br>"
                echo "Porta 2: High<br>"
```

```
                echo "Porta 3: High<br>"

        fi
}

function funcao1 {
        if [ $valor1 -eq 1 ]
           then
                valor_final1=1
                valor_final2=0
                valor_final3=0
                echo "Porta 1: High<br>"
                echo "Porta 2: Low<br>"
                echo "Porta 3: Low<br>"
        fi
        if [ $valor1 -eq 2 ]
           then
                valor_final1=0
                valor_final2=1
                valor_final3=0
                echo "Porta 1: Low<br>"
                echo "Porta 2: High<br>"
                echo "Porta 3: Low<br>"

        fi
        if [ $valor1 -eq 3 ]
           then
                valor_final1=0
                valor_final2=0
                valor_final3=1
                echo "Porta 1: Low<br>"
                echo "Porta 2: Low<br>"
                echo "Porta 3: High<br>"
        fi
}

function funcao0 {
                valor_final1=0
                valor_final2=0
                valor_final3=0
                echo "Porta 1: Low<br>"
                echo "Porta 2: Low<br>"
                echo "Porta 3: Low<br>"
}

case $argumentos in
        0) funcao0;;
        1) funcao1;;
        2) funcao2;;
        3) funcao3;;
esac
```

```
total=`eval                                                    "echo
$valor_final3*4+$valor_final2*2+$valor_final1*1 | bc"`

rm -fR /media/windows/digital_output.bat
printf   "@echo   off\nD:\log\digital_output.exe   "$total""   >
/media/windows/digital_output.bat
snmpwalk  -v  1  -c  fujitaro  fujita  1.3.6.1.4.1.30778.15   >
/dev/null

echo "</BODY>"
echo "</HTML>"
```

D. Scripts dos traps

D.1. up.sh

```
 dia="`date +%d`"
 mes="`date +%m`"
 ano="`date +%y`"
 hora="`date +%H`"
 min="`date +%M`"
identifica_pc="`grep  "Cold  Start"  /var/log/snmptrapd.log  |
tail -n 1 | awk '{printf "%s\n",$3}'`"

echo "$identifica_pc - Deamon SNMPD ligado em $dia/$mes/$ano
as $hora:$min" > /home/murilo/mail_ligado.txt

/usr/bin/mail   murilo@unifei.edu.br   -s   "SNMPD   ligado"   <
/home/murilo/mail_ligado.txt
```

D.2. down.sh

```
 dia="`date +%d`"
 mes="`date +%m`"
 ano="`date +%y`"
 hora="`date +%H`"
 min="`date +%M`"
identifica_pc="`grep                  netSnmpNotificationPrefix
/var/log/snmptrapd.log   |   tail   -n   1   |   awk   '{printf
"%s\n",$3}'`"
```

```
echo "$identifica_pc - Deamon SNMPD desligado em $dia/$mes/$ano as $hora:$min" > /home/murilo/mail_desligado.txt

/usr/bin/mail murilo@unifei.edu.br -s "SNMPD desligado" < /home/murilo/mail_desligado.txt
```

D.3. verifica_memoria.sh

```
echo "content-type: text/html"
echo
echo
echo "
<HTML> <HEAD> <BODY BGCOLOR="black" TEXT="white"> </HEAD>"

quantidade_memoria ()
  {
   memoria_resposta=`snmpwalk -v 1 -c fujitaro $identifica_pc -Oa 1.3.6.1.4.1.30778.10.3.1.2.7.47.98.105.110.47.115.104 | egrep '(Total Physical Memory|Mem.ria f.sica total)' | sed 's/,//' | awk '{printf "%s\n",$4}'`
   if [ ! "$memoria_resposta" = "$memoria_da_maquina" ]
    then
        printf "alteração na quantidade de memória do agente $identifica_pc \nDe $memoria_da_maquina MB para $memoria_resposta MB" > /home/murilo/alteracao_memoria.txt
        /usr/bin/mail murilo@unifei.edu.br -s "Alerta memória" < /home/murilo/alteracao_memoria.txt
    else
        echo "Memória ok"
   fi
  }

identifica_pc=`grep "Cold Start" /var/log/snmptrapd.log | tail -n 1 | awk '{printf "%s\n",$3}'`
 if [ $identifica_pc = "10.10.8.65" ]
    then
memoria_da_maquina=
quantidade_memoria
 fi

 if [ $identifica_pc = 10.10.8.66 ]
    then
        memoria_da_maquina=495
quantidade_memoria
 fi
```

```
 if [ $identifica_pc = 10.10.8.67 ]
    then
        memoria_da_maquina=1280
quantidade_memoria
 fi

 if [ $identifica_pc = 10.10.8.68 ]
    then
        memoria_da_maquina=768
quantidade_memoria
 fi

 if [ $identifica_pc = 10.10.8.69 ]
    then
        memoria_da_maquina=896
quantidade_memoria
 fi

 if [ $identifica_pc = 10.10.8.71 ]
    then
        memoria_da_maquina=512
quantidade_memoria
 fi

 if [ $identifica_pc = 10.10.8.72 ]
    then
        memoria_da_maquina=1.536
quantidade_memoria
 fi

 if [ $identifica_pc = 10.10.8.74 ]
    then
        memoria_da_maquina=1.024
quantidade_memoria
 fi

 if [ $identifica_pc = 10.10.8.75 ]
    then
        memoria_da_maquina=1.024
quantidade_memoria
 fi

 if [ $identifica_pc = 10.10.8.76 ]
    then
        memoria_da_maquina=1.024
quantidade_memoria
 fi

 if [ $identifica_pc = 10.10.8.78 ]
    then
        memoria_da_maquina=1280
```

```
quantidade_memoria
 fi

 if [ $identifica_pc = 10.10.8.79 ]
    then
        memoria_da_maquina=3582
quantidade_memoria
 fi

 if [ $identifica_pc = 10.10.8.80 ]
    then
        memoria_da_maquina=1012
 memoria_resposta=`snmpwalk   -v   1   -c   fujitaro   10.10.8.80
1.3.6.1.4.1.30778.13.3.1.2.7.47.98.105.110.47.115.104   |   grep
Mem | awk '{printf "%s\n",$2}'`
    if [ $memoria_resposta -ne $memoria_da_maquina ]
then
          printf "alteração na quantidade de memória do agente
$identifica_pc     \nDe     $memoria_da_maquina     MB     para
$memoria_resposta MB" > /home/murilo/alteracao_memoria.txt
         /usr/bin/mail    murilo@unifei.edu.br    -s    "Alerta
memória"  < /home/murilo/alteracao_memoria.txt
    fi
 fi

echo "</BODY>"
echo "</HTML>"
```

E. Guia prático para instalar o protocolo SNMP

E.1. Instruções para a instalação do gerente

O SNMP é instalado através do apt-get, ferramenta que executa a ação de download e instala o pacote. O comando é:

```
$ sudo apt-get install snmp
```

Para fazer o ajuste do deamon snmptrapd, deve-se editar o aquivo /etc/snmp/snmptrapd.conf através do editor vi:

```
$ vi /etc/snmp/snmptrapd.conf
```

O arquivo tem o conteúdo abaixo.

```
--- início de /etc/snmp/snmptrapd.conf ---
traphandle                NET-SNMP-AGENT-MIB::nsNotifyShutdown
/home/murilo/down.sh  desligado
traphandle  SNMPv2-MIB::coldStart           /home/murilo/up.sh
ligado
traphandle                                  SNMPv2-MIB::coldStart
/home/murilo/verifica_memoria.sh  memoria
--- fim de /etc/snmp/snmptrapd.conf ---
```

Por fim, deve-se configurar quais os serviços funcionam na máquina gerente através do arquivo /etc/default/snmp. O arquivo deve ser configurado como abaixo:

```
--- início de /etc/default/snmp ---
SNMPDRUN=no
TRAPDRUN=no
--- fim de /etc/default/snmp ---
```

Estas instruções fazem que o daemon snmpd fique desabilitado e snmptrapd seja habilitado.

Por fim, a variável de ambiente deve ser configurada para localizar as bibliotecas necessárias para o SNMP através do comando:

```
$ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
```

Interface

Os arquivos de hipertexto devem estar localizados de acordo com o arquivo /etc/apache2/apache2.conf a qual, por padrão, é /var/www. Neste mesmo diretório estão os arquivos CSS os quais formatam os elementos da página.

Scripts

O mesmo arquivo de configuração do Apache citado no item anterior busca os scripts no diretório /usr/lib/CGI-bin. Desta forma, os eventos de interface executam os scripts sempre que solicitados.

E.2. Instruções para a instalação do agente

Porta de comunicação

Abrir a porta que faz a comunicação do protocolo SNMP. A configuração é feita através do menu Iniciar -> Configurações -> Painel de Controle -> Firewall do Windows -> aba Exceções. Ao clicar no botão "Adicionar Porta" é aberta outra caixa diálog a qual o nome pode ser preenchido com "snmp" e o número da porta é "161". Por fim, marcar a opção "UDP".

Instalação do protocolo SNMP

Obter o aplicativo no endereço http://www.net-snmp.org/download.html optando pela última versão disponível de acordo com o tipo de arquitura da máquina a ser instalada.

Iniciada a instalação, a única alteração a ser feita é alterar o checkbox de "Stardant agent" para "With Windows Extension" como mostra a Fig. O passo seguinte é clicar em Iniciar -> Service -> Register Agent Service.

É fundamental que seja instalada as bibliotecas do windows seguindo o procedimento a seguir: Iniciar -> Painel de Controle -> Adicionar ou remover programas -> Adicionar/Remover componentes do windows -> Ferramentas de gerenciamento e monitoramento. Será pedido para que insira o CD do Windows para realizar a instalação das bibliotecas.

Para configurar os serviços da máquina agente, os passos são: Iniciar -> Painel de Controle -> Ferramentas administrativas -> Serviços. O serviço Net-SNMP deve ser configurado para "Automático" enquanto "Serviço SNMP" deve aparecer como "Desativado".

Para fazer o ajuste do deamon snmpd, deve-se editar o aquivo /etc/snmp/snmpd.conf através do editor vi:

```
$ vi /etc/snmp/snmpd.conf

--- inicio de /etc/snmp/snmpd.conf ---
rocommunity fujitaro
rwcommunity fujitarw
extend .1.3.6.1.4.1.30778.1 /bin/sh "D:/log/usb6008.bat"
extend    .1.3.6.1.4.1.30778.2    /bin/sh    "tail    -n    1
D:/log/voltage1.txt"
extend    .1.3.6.1.4.1.30778.3    /bin/sh    "tail    -n    1
D:/log/voltage2.txt"
extend    .1.3.6.1.4.1.30778.4    /bin/sh    "tail    -n    1
D:/log/voltage3.txt"
extend .1.3.6.1.4.1.30778.5 /bin/sh "netstat -an | findstr TCP
| findstr 0.0.0.0"
extend .1.3.6.1.4.1.30778.6 /bin/sh "uptime"
extend .1.3.6.1.4.1.30778.7 /bin/sh "netstat -e -s"
extend .1.3.6.1.4.1.30778.8 /bin/sh "ipconfig /all"
extend .1.3.6.1.4.1.30778.9 /bin/sh "df -h -T"
extend .1.3.6.1.4.1.30778.10 /bin/sh "systeminfo"
extend    .1.3.6.1.4.1.30778.11    /bin/sh    "tail    -n    1
D:/log/digital.txt"
extend  .1.3.6.1.4.1.30778.12  /bin/sh  "netstat  -an  |  findstr
UDP | findstr 0.0.0.0"
extend  .1.3.6.1.4.1.30778.13  "c:\windows\system32\cmd.com  /C"
"d:/log/apagador.exe"
extend  .1.3.6.1.4.1.30778.14  "c:\windows\system32\cmd.com  /C"
"d:/log/output.bat"
extend  .1.3.6.1.4.1.30778.15  "c:\windows\system32\cmd.com  /C"
"d:/log/digital_output.bat"
--- fim de /etc/snmp/snmpd.conf ---
```

Instalação do Cygwin

Obter a última versão do binário através do endereço http://www.cygwin.com. Dois cliques e a instalação inicia não sendo necessário modificar as opções sugeridas.

Configuração do Apache

O conteúdo do arquivo /etc/apache2/sites-available abaixo instrui onde devem ficam os arquivos HTML e os executáveis cgi-bin.

```
--- inicio de /etc/apache2/sites-available/default ---
        DocumentRoot /var/www
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>

        ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
        <Directory "/usr/lib/cgi-bin">
                AllowOverride None
                Options          +ExecCGI          -MultiViews
+SymLinksIfOwnerMatch
                Order allow,deny
                Allow from all
        </Directory>
--- fim de /etc/apache2/sites-available/default ---
```

Configuração de roteador

A interface varia entre os fabricantes e modelos, porém o princípio deve empregado da mesma forma em todos eles: configurar o número da porta e direcionar o IP para a máquina que deve responder o comando quando solicitado de fora de rede.

F. Os comandos SNMP

A seguir encontram-se listados os principais comandos SNMP do software Net-SNMP para Windows e Linux versão 5.4.1.

1. ***snmpget***: busca valores de objeto formado por uma única instância. Para este comando deve ser informado o índice da instância. É possível recuperar múltiplas instâncias desde que cada índice seja explicitado.
2. ***snmpgetnext***: busca a partir do objeto procurado até o fim do ramo. Para cada *snmpget* enviado para a máquina consultada, um *snmpresponse* é devolvido para cada objeto que é encontrado e uma *flag* aponta a existência do próximo objeto. Este modo recursivo fica ativo até retornar um erro que é sinalizado quando não há objeto algum a ser recuperado. Adequado para buscar objeto na forma textual.
3. ***snmpset***: seu propósito é armazenar o valor de um objeto gerenciado ou criar uma nova linha na tabela. Este comando pode modificar mais um objeto por vez, mas no caso de um falhar, todos voltam ao estado original até o ponto da falha.
4. ***snmpwalk***: desempenha a mesma função do *snmpgetnext* capaz de recuperar uma subárvore.
5. ***snmpnetstat***: executa o mesmo comando *netstat* dos sistemas operacionais baseados em UNIX através do SNMP. É possível recuperar informações como portas ativas e informações relativas às redes.
6. ***snmpdf***: executa o mesmo comando *df* dos sistemas operacionais baseados em UNIX através do SNMP. Este comando retorna uma tabela com o tamanho das partições, a quantidade disponível e ocupada bem como o percentual ocupado da capacidade de armazenamento.
7. ***snmpd***: este comando é o responsável por responder a todas as requisições SNMP através de uma porta.
8. ***snmptrapd***: este comando é o responsável por capturar as informações de um evento e criar um relatório de condições do que se está monitorando.
9. ***snmptranslate***: a função mais básica é traduzir da notação textual para numérica ou vice-versa. Outra capacidade é a de exibir toda a estrutura do

objeto codificado na linguagem ASN.1 mostrando a localização do nó, descrição etc.