

EMPREGO DA API SNMP4J NA GERÊNCIA DE MODENS ADSL

Ranier William Lopes DELGADO

Centro de Ensino Superior de Juiz de Fora, Juiz de Fora, MG

Romualdo Monteiro de Resende COSTA

Resumo: Com o aumento do número de pessoas conectadas à Internet, as autoridades se preocupam com a qualidade do serviço oferecido. Assim, foi criado, em 2011, em atendimento as resoluções 574 e 575 da Agência Nacional de Telecomunicações (Anatel), a Entidade Aferidora da Qualidade de Banda Larga. O grande desafio desse processo é oferecer as condições necessárias à medição da qualidade pelos usuários. Considerando que uma parte considerável dos usuários utiliza conexões *Asymmetric Digital Subscriber Line* (ADSL) para acesso à Internet, este artigo apresenta um aplicativo para monitoramento de modems ADSL utilizando o protocolo *Simple Network Management Protocol* (SNMP). Ao contrário da solução da Entidade Aferidora, que é baseada em um hardware específico, o aplicativo apresentado pode ser utilizado por um grande número de usuários sem a necessidade de investimentos em equipamentos específicos.

Palavras-chave: SNMP; Gerência de Redes; Modem; ADSL; Java; SNMP4J.

1 INTRODUÇÃO

É cada vez maior o número de pessoas que se conectam diariamente à Internet, o mercado brasileiro no início de 2014 atingiu a marca de 22,73 milhões de acessos em banda larga com a tecnologia *Digital Subscriber Line* (DSL) (ANATEL, 2014). Entre as formas de conexão DSL, a tecnologia *Asymmetric Digital Subscriber Line* (ADSL) é uma tecnologia que permite transferência digital de dados em alta velocidade por meio de linha telefônica de cobre comum, com pares trançados. O ADSL divide o enlace de telefonia em dois canais lógicos, uma para voz, e outro para dados. O canal de dados, por sua vez, é redividido em dois outros canais, um maior para *download*, e outra menor, para *upload*, daí o nome assimétrico que surgiu da percepção que seria melhor reservar a maior parte da banda para *download*, uma parte menor para *upload* e por fim uma parte fixa para voz (TANENBAUM, 2011), caracterizando assim canais distintos como pode ser observado na Figura 1. Nessa figura, os espaços marcados em azul correspondem a faixas de frequência de proteção colocados entre os canais (banda de guarda):

FIGURA 1
Divisão conexão ADSL



Fonte: Tanenbaum (2011)

A base para qualquer sistema de gerenciamento da Internet é o SNMP (RFC 1157). Esse protocolo define um conjunto limitado de comandos e respostas possíveis, o que o torna bastante simples. O protocolo SNMP não é orientado à conexão sendo utilizado entre sistemas de gerenciamento (gerentes) e sistemas gerenciados (agentes), que podem ser, tipicamente, qualquer tipo de dispositivo de rede (MAURO; SCHMIDT, 2005). Para que um dispositivo ou recurso de rede possa ser gerenciado, basta existir um agente SNMP no dispositivo ou habilitá-lo, o qual, por sua vez, coleta informações numa base chamada **Management Information Base** (MIB) (RFC 1213), que pode ser consultada pelo sistema de gerenciamento usando o protocolo SNMP.

Atualmente, muitos equipamentos, incluindo os modems ADSL, possuem interfaces amigáveis, muitas vezes acessíveis através de um navegador Web utilizando, portanto outro protocolo, o **HyperText Transfer Protocol** (HTTP) (RFC 2616). Essa forma de utilização, no entanto, exige que o usuário realize as requisições de forma interativa com o equipamento, através de uma interface que, por mais amigável que seja, não é padronizada. Dessa forma, muitos usuários farão apenas a configuração inicial de seus dispositivos, não retornando, posteriormente, para verificar as condições de funcionamento ou mesmo falhas que não tenham sido percebidas na configuração inicial (ASSOLINI, 2012).

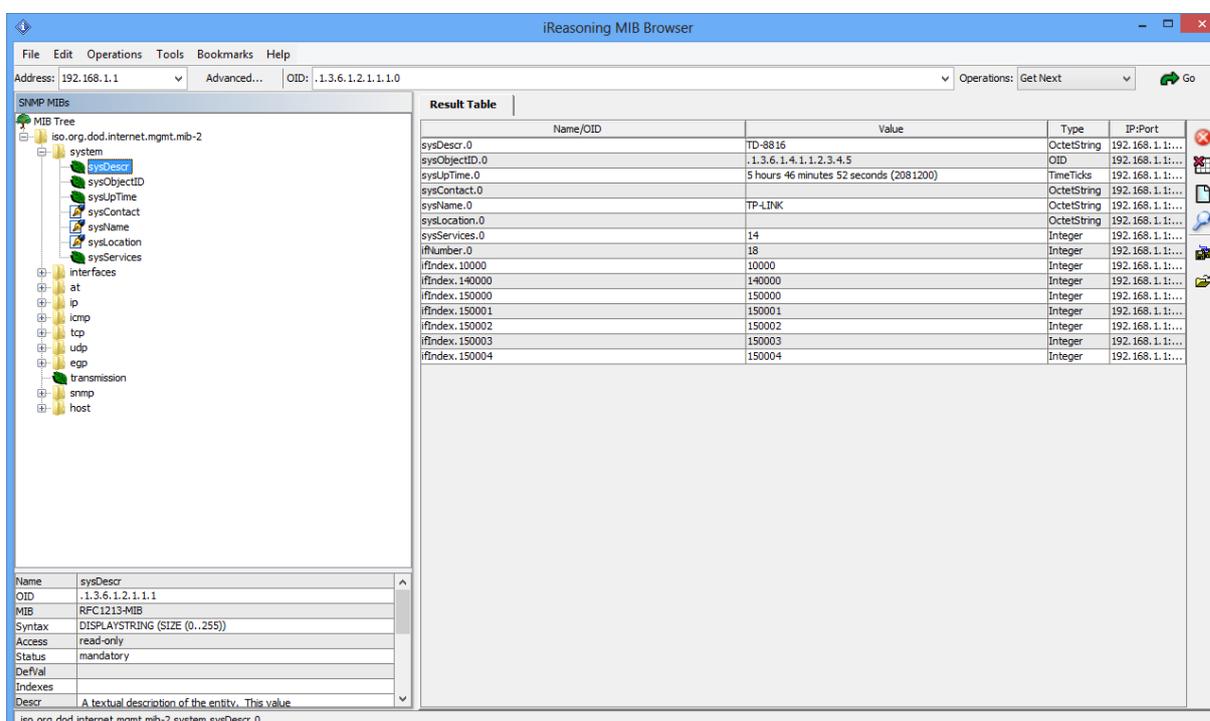
2 TRABALHOS RELACIONADOS

MIB Browser é uma importante ferramenta para gerenciar dispositivos de rede e aplicações compatíveis com SNMP. Essa ferramenta permite aos usuários carregar padrões, MIBs proprietárias, e até mesmo, identificar possíveis MIBs malformadas. Também permite emitir solicitações SNMP para recuperar dados do agente, ou fazer alterações nos valores dos objetos armazenados nesses agentes. A Fi-

gura 2 apresenta um esquema de funcionamento dessa ferramenta utilizada para trazer os objetos existentes em um modem ADSL.

Como pode ser observado na figura, o software MIB Browser permite trazer um grande conjunto de informações associadas a um dispositivo gerenciável. Essa característica, ao mesmo tempo em que permite uma ampla visão da situação do equipamento em questão, pode dificultar a interpretação das informações, justamente pela grande quantidade de informação. Dessa forma, sua utilização por usuários finais pode não ser adequada.

FIGURA 2
Funcionamento MIB Browser



Fonte: Software MIB browser (2014)

Outra ferramenta, o NET-SNMP (NET-SNMP, 2014), permite a implementação de um agente SNMP em equipamentos onde essa entidade não está disponível. Também é possível, através dessa ferramenta, implementar um gerente que envia os comandos com solicitações aos agentes. Essa ferramenta, no entanto, apesar das funcionalidades oferecidas é baseada em requisições de linha de comando, devendo o usuário conhecer não somente o protocolo SNMP, mas também a sintaxe dos comandos do programa, que devem ser interativamente executados.

Em relação às ferramentas com interfaces mais amigáveis, diversas ferramentas Web podem ser encontradas. A gerência através de browser oferece a faci-

lidade de acesso através de páginas interpretadas em navegadores, acessíveis em qualquer máquina na rede e com interface similar a dos mais populares sites. Entre essas ferramentas, podem ser citadas o *Multi Router Traffic Grapher* (MRTG) (MRTG, 2011), o Nagios (NAGIOS, 2014), o Zabbix (ZABBIX, 2014) entre outros. Essas ferramentas, no entanto, oferecem um nível de complexidade elevado em razão, principalmente, de serem bastante genéricas, permitindo a utilização de inúmeros parâmetros para monitoramento e controle.

Em relação à medição dos parâmetros de Internet banda larga poderia ser adequada a utilização de ferramentas que oferecessem uma abstração da comunicação SNMP e que, ao mesmo tempo, permitissem o desenvolvimento de uma interface mais amigável, voltada especificamente a essa questão. Nesse aspecto, a API SNMP4J (SNMP4J, 2014) merece ser destacada ao permitir a utilização de uma linguagem com alto nível de abstração (Java¹) para a construção da interface, ao mesmo tempo em que oferece acesso rápido aos objetos gerenciáveis utilizando o SNMP. Essa API será tratada em detalhes na Seção 4.

3 JUSTIFICATIVA

A necessidade do acesso a Internet definida como banda larga, tem levado os usuários, bem como os órgãos de regulamentação, a repensar seus meios de controle e gerenciamento de seus equipamentos, e é papel da aplicação dar esse controle a eles. Deve ser facultado ao usuário configurar o equipamento e gerenciá-lo, monitorando e controlando de forma rápida e intuitiva. Entre os equipamentos, o modem, em particular na tecnologia ADSL, atua como um elemento de ligação entre a rede local do usuário e a rede de acesso ao provedor de Internet contratado. Devido a essa importância, é necessário verificar, de forma constante os parâmetros de funcionamento do modem e, quando necessário, atuar para corrigir e configurar, apropriadamente, esses parâmetros.

Modems ADSL vulneráveis funcionando no Brasil são uma realidade. Como exemplo, em Assolini (2012) são reportadas vulnerabilidades de *firmware*, de scripts maliciosos e de servidores maliciosos de DNS que afetaram seis fabricantes de dispositivos de rede, resultando em usuários vítimas de ataques de *malwares* e visitas a páginas de *phishing*.

Ainda que os problemas de segurança não existissem, a própria qualidade do

¹ Java: Linguagem de programação (JAVA, 2014).

serviço da Internet banda larga muitas vezes não é garantida. Para tentar identificar problemas relativos ao serviço, a Entidade Aferidora da Qualidade foi criada, em atendimento à Resolução Anatel 574 e 575 de 28 de outubro de 2011 (ANATEL, 2014).

Para aferir a Internet banda larga, foram instalados dispositivos que verificam a taxa de *download* e *upload* oferecida aos usuários. Para isso, o dispositivo, periodicamente, faz o *download* de dados a partir de um ou mais servidores e mede parâmetros associados à percepção desses dados. A qualidade do *upload* também é medida de forma semelhante. Além da vazão, os testes realizados também incluem a latência, a perda de pacotes e o jitter², nesse caso sobre o protocolo da camada de transporte **User Datagram Protocol** (UDP) (COMMER, 1995).

As tarefas realizadas pelo dispositivo oferecido pela Entidade Aferidora da Qualidade fazem parte da gerência de redes. Mais especificamente, das áreas de configuração e falhas (SUBRAMANIAN, 2010).

A gerência de configuração qualifica, mede, analisa, informa diferentes componentes e dispositivos de rede. A gerência de falhas, por sua vez, trata as interrupções da comunicação, os serviços interrompidos em servidores e, eventualmente, taxa excessivas de erros em links de comunicação (SUBRAMANIAN, 2010).

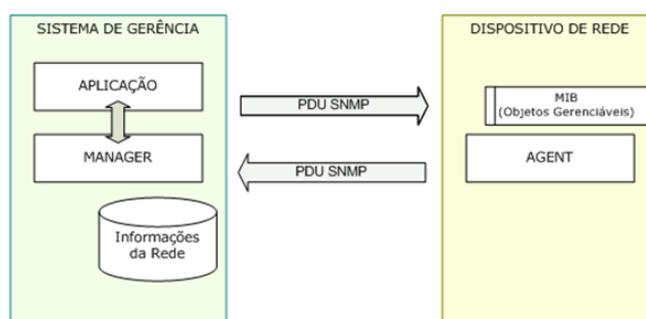
Por se tratar da arquitetura TCP/IP (COMMER, 1995), ao invés do emprego de equipamentos específicos, é possível utilizar o próprio modem ADSL, através do protocolo SNMP, para realizar a aferição de diversos parâmetros, em particular, daqueles relacionados à gerência de falhas e configuração. É justamente este o ponto de que trata este artigo, ao propor um protótipo que permita a aferição dos parâmetros disponíveis em um modem ADSL comum.

² Jitter: Variação da Latência da Rede (PETERSON; DAVIE, 2013).

4 A ARQUITETURA SNMP PARA GERENCIAMENTO

A Figura 3 ilustra os componentes arquiteturais propostos neste trabalho. Para efetuar a comunicação entre o gerente (manager) e o agente (agent) de rede é utilizado o protocolo SNMP. A requisição parte da aplicação, passa pelo gerente (sistema de gerência) vai para o agente (modem ADSL) através do protocolo SNMP e chega a MIB, deste agente, com a requisição. É na MIB que os parâmetros desejados encontram-se armazenados. Posteriormente, o agente retorna para o gerente a resposta à requisição, através do mesmo protocolo com a mensagem de resposta apropriada. A escolha do SNMP de objetos padronizados, encontrado na MIB de qualquer equipamento gerenciável dessa natureza, garante a proteção contra erros de interpretação ou falta de compatibilidade entre dados trocados.

FIGURA 3
Comunicação SNMP



Fonte: Do Autor (2014)

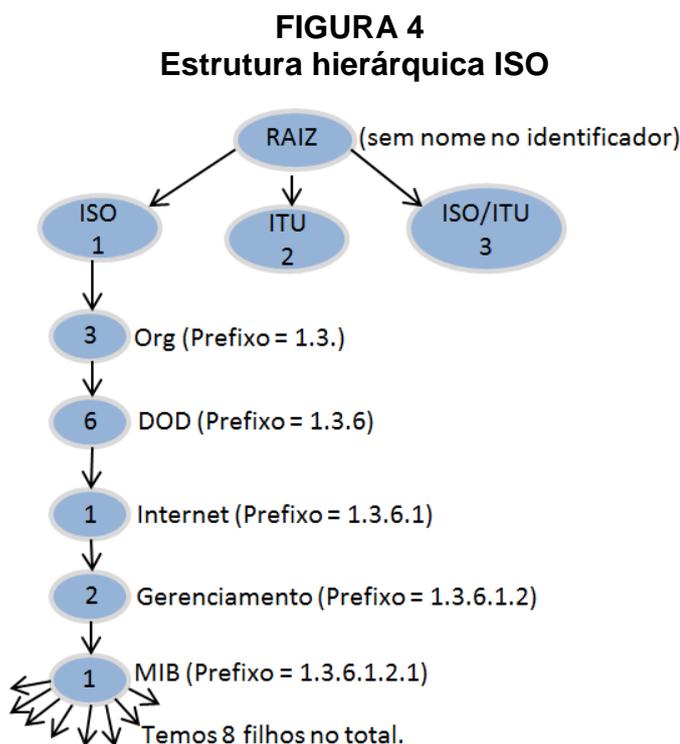
Na estrutura MIB, o campo nome ou identificador do **Object Identifier** (OID) (SUBRAMANIAN, 2010) define um nome único para um objeto gerenciado. O OID pode ser trabalhado em dois formatos: numérico e o simbólico (legível):

- Exemplo numérico: 1.3.6.1.2.1.1.1.0 (iremos adotar este formato para implementação do protótipo).

- Exemplo simbólico: iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0

Fazendo uma analogia com as definições do modelo entidade-relacionamento (ER) considerando, a MIB como uma tabela de um banco de dados, por exemplo, tabela de clientes, o objeto gerenciável (variável MIB) como um atributo da tabela, por exemplo, nome_cliente, a **Structure of Management Information** (SMI) como a especificação ou documentação do tipo de conteúdo a ser armazenado no atributo para representar o nome do cliente, por exemplo, varchar(50). Como um dicionário, que mostra como pronunciar uma palavra e, em seguida, apresenta o significado ou a definição dessa mesma palavra, uma MIB define um nome em texto de um objeto gerenciado e explica o seu significado (MAURO; SCHMIDT, 2005).

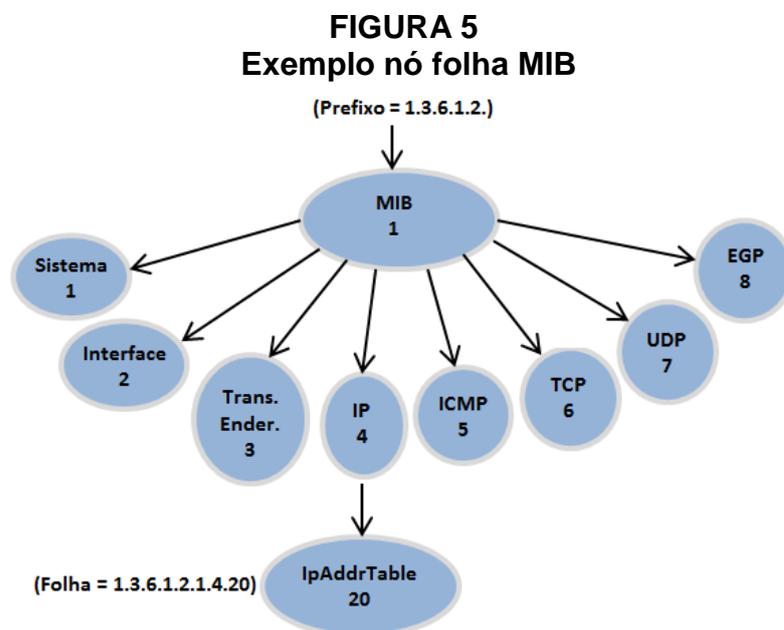
Os objetos gerenciáveis de uma MIB se relacionam através de uma estrutura hierárquica em árvore de objetos, onde o nó posicionado no início da árvore é denominado raiz. Todo nó intermediário que tiver filhos é conhecido como pai de uma subárvore e todo nó que não tiver filhos é chamado de nó folha. A Figura 4 apresenta a distribuição inicial dessa árvore. Conhecer essa estrutura é fundamental para escolher apropriadamente os objetos que terão seus valores avaliados.



Fonte: Do Autor (2014)

Abaixo do prefixo 1.3.6.1.2.1 da estrutura da MIB podem ser encontrados oito identificadores: Sistema, Interface, Transender, IP, ICMP, TCP, UDP, EGP como ilustrado na Figura 5. Cada um desses identificadores conduz a um conjunto de ob-

jetos com determinada função. O identificador 1.3.6.1.2.1.4.20, por exemplo, que traz o nó folha de nome ipAddrTable (RFC 1213), retornará a informações relevantes para endereços IP da entidade gerenciada.



Fonte: Do Autor (2014)

Através dos identificadores dos objetos, um método `getOidString` pode ser implementado no gerente para prover a captura do valor de um nó folha qualquer, bastando receber como parâmetro o identificador do objeto gerenciado como mostra a ilustração da Figura 6. No exemplo apresentado nessa figura foi escolhido como objeto o endereço físico do equipamento MAC Address. Posteriormente esse valor é apresentado ao usuário, através do lançamento no campo do protótipo com o comando `jtMac.setText(MacAddress)` Figura 6.

FIGURA 6
Utilização da MIB no código fonte.

```
public String getOidString(OID oid) throws IOException {
    ResponseEvent event = get(new OID[]{oid});
    return event.getResponse().get(0).getVariable().toString();
}

String MacAddress = modem.getOidString(new OID(".1.3.6.1.2.1.2.2.1.6.10000"));
jtMac.setText(MacAddress);
```

Fonte: Do Autor (2014)

O exemplo apresentado na Figura 6 ilustra a utilização do protocolo SNMP, através do recebimento de uma resposta a uma solicitação de um objeto, em conjunto com os objetos da MIB.

Nesse caso, o SNMP é utilizado para enviar e receber dados dos objetos desejados existentes nos agentes. No contexto de um modem ADSL, por exemplo, esse protocolo poderia ser utilizado para, periodicamente, monitorar o modem e notificar o usuário caso alguns dos parâmetros mensurados não estejam dentro de valores considerados aceitáveis. Essa forma de utilização estaria relacionada à gerência de falhas, onde falhas não são o mesmo que erros. Uma falha é uma condição anormal cuja recuperação exige ação de gerenciamento e normalmente é causada por operações incorretas ou um número excessivo de erros (SUBRAMANIAN, 2010).

Outra utilização, mais genérica, seria a possibilidade de configurar remotamente o modem sem que fosse necessário vasculhar de forma interativa a sua interface Web, viabilizando o processo de configuração remota e caracterizando outra área da gerência de redes, a gerência de configuração. Essa gerência está relacionada à inicialização da rede e com uma eventual desabilitação de parte ou de toda a rede. Também está relacionado às tarefas de manutenção, adição e atualização de relacionamentos entre os componentes e da situação dos componentes durante a operação da rede (SUBRAMANIAN, 2010).

Para realizar a especificação de um programa de configuração e detecção de falhas em modems ADSLs, bibliotecas e *Application Programming Interface* (API) específicas podem ser utilizadas. Em particular, a linguagem de programação Java oferece um amplo conjunto de APIs com esse objetivo (COSTA, 2008). Java é a base de uma ampla gama de aplicativos em rede, e é um padrão global para desenvolvimento e fornecimento de aplicativos para celular, jogos, conteúdo on-line e software corporativo. Com milhões de desenvolvedores em todo o mundo, o Java permite desenvolver e implementar aplicativos e serviços diversos. Com ferramentas abrangentes, um ecossistema sólido e um desempenho eficiente, o Java oferece a portabilidade de aplicativos mesmo entre os ambientes computacionais mais diferentes.

5 DESENVOLVIMENTO DO PROTÓTIPO

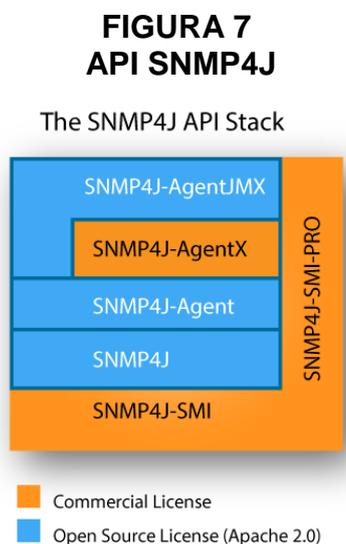
O objetivo do trabalho, como já mencionado, consiste em implementar um protótipo na linguagem Java que consiga estabelecer a comunicação SNMP com modems ADSL visando à gerência de falhas e de configuração desses equipamen-

tos. Para isso, também é objetivo deste trabalho desenvolver interface amigável e de fácil manuseio onde serão apresentados os parâmetros disponíveis nos principais modems ADSL para configuração e monitoramento de falhas e, avaliar outras ferramentas que possam ser comparadas com o protótipo implementado também fazem parte dos objetivos deste trabalho.

O protótipo deste trabalho deverá servir para auxiliar os usuários no monitoramento de suas bandas de internet contratadas, além de verificar a versão do *firmware*, a fim de evitarem os problemas como o de uma versão defasada e vulnerável. O modelo utilizado para testes foi o TPLINK - TP-8816 sem, no entanto, estar o protótipo atrelado a um tipo específico de modems. O protótipo foi elaborado e implementado na plataforma Java no IDE Netbeans 7.4.

Um aplicativo de gerenciamento de modems ADSL de fácil configuração e gerenciamento é de grande importância para qualquer usuário de internet, mas é importante frisar que um conhecimento mínimo é necessário. No caso do protótipo em questão o usuário deve conhecer o endereço IP do modem que ele deseja monitorar.

O protótipo, como já mencionado, foi desenvolvido em Java, utilizando a API SNMP4J cuja arquitetura é apresentada na Figura 7.



Fonte: SNMP4J (2014)

A API SNMP4J, disponível na biblioteca Java SNMP4J, fornece classes e interfaces para a criação, envio e recebimento de mensagens via SNMP (COSTA, 2008). A comunicação principal foi implementada através do método `get(OID oids[])`, como ilustrado na Figura 8. Dentro desse método um objeto da classe PDU, que cor-

responde a mensagem a ser enviada é construído. Esse método oferece um método add para definir quais nó folha (objetos existentes nos agentes) devem ser requisitados (pdu.add(new VariableBinding(oid))). Antes do envio da mensagem também deve ser escolhido o tipo de requisição, que neste é um GET e o alvo ou agente (getTarget()) que para qual será enviada a requisição através do método snmp.send.

FIGURA 8
Instanciação de uma nova PDU, envio do PDU e alvo.

```
533 public ResponseEvent get(OID oids[]) throws IOException {
534     PDU pdu = new PDU();
535     for (OID oid : oids) {
536         pdu.add(new VariableBinding(oid));
537     }
538     pdu.setType(PDU.GET);
539     ResponseEvent event = snmp.send(pdu, getTarget(), null);
540     if (event != null) {
541         return event;
542     }
543     throw new RuntimeException("GET timed out");
544 }
```

Fonte: Do Autor (2014)

Para a seleção individual dos dados recebidos via GET, foi implementado o método *GetOidString* que faz com que o OID seja selecionado retornando um valor da MIB como pode ser observado na Figura 9. A classe OID permite a definição e manipulação dos identificadores solicitados.

FIGURA 9
Captura de OID

```
526 public String getOidString(OID oid) throws IOException {
527     ResponseEvent event = get(new OID[]{oid});
528     return event.getResponse().get(0).getVariable().toString();
529 }
```

Fonte: Do Autor (2014)

6 APLICATIVO

O aplicativo tem uma interface criada com campos de texto e botões, utilizando a classe Swing Java de forma com que o usuário consiga identificar cada dado de forma simples e fácil com as rotulações em cada item (Figura 10).

FIGURA 10
Tela de visualização dos campos do protótipo

Comunicação Modens ADSL

DIGITE O IP: IP MODEM:

Fabricante:

Máximo de Conexões:

Marca: Máscara de Rede:

Modelo: Bytes Recebidos: Bytes Enviados:

Versão do Firmware: Taxa de Tranferência: Kbps Erros:

Versão do Software:

Versão Harware:

Tempo Ligado(UP TIME):

Mac Adress:

BUSCAR DADOS LIMPAR FECHAR

Fonte: Do Autor (2014)

Na tela são trazidas as seguintes informações: um campo texto editável, para que o usuário possa digitar o ip do modem que deseja gerenciar, um campo IP do modem não editável somente para exibição, fabricante não editável, marca não editável, modelo não editável, versão do firmware não editável, versão do hardware não editável, versão do software instalados no equipamento não editável, máscara de rede não editável, máximo de conexões não editável, bytes enviados não editável, bytes recebidos não editável, taxa de transferência não editável, erros não editável,

tempo ligado (UP TIME) não editável, e o mac address não editável, ainda existe um botão buscar dados, para fazer uma busca dos dados dos campos descritos cima de acordo com o número IP digitado pelo usuário, um botão limpar para limpar todos os campos do programa, e fechar que finaliza a aplicação.

O método void Start() inicia a porta de comunicação da aplicação com o equipamento criando uma nova instância de SNMP e executando a tarefa transport.listen(), que literalmente escuta o meio aguardando um comando para ser enviado para o agente SNMP (Figura 11).

FIGURA 11
Iniciação do processo de escuta da rede

```
public void start() throws IOException {  
    TransportMapping transport = new DefaultUdpTransportMapping();  
    snmp = new Snmp(transport);  
    transport.listen();  
}
```

Fonte: Do Autor (2014)

Ao se executar o protótipo todos os campos são preenchidos e, com o dado versão do *firmware*, por exemplo, o aplicativo consegue fazer o monitoramento desse quesito, a fim de suprir a falha na questão de um equipamento desatualizado, que pode ficar vulnerável a ataques conforme mencionado no artigo de Assolini (2012), visando à gerência de falhas.

O campo taxa de transferência pode verificar como o serviço de acesso a internet está se comportando, assim podendo monitorá-lo e verificar se está de acordo com o contratado, assim abrangendo a gerência de configuração que serve para medir, aferir e monitorar dispositivos de redes.

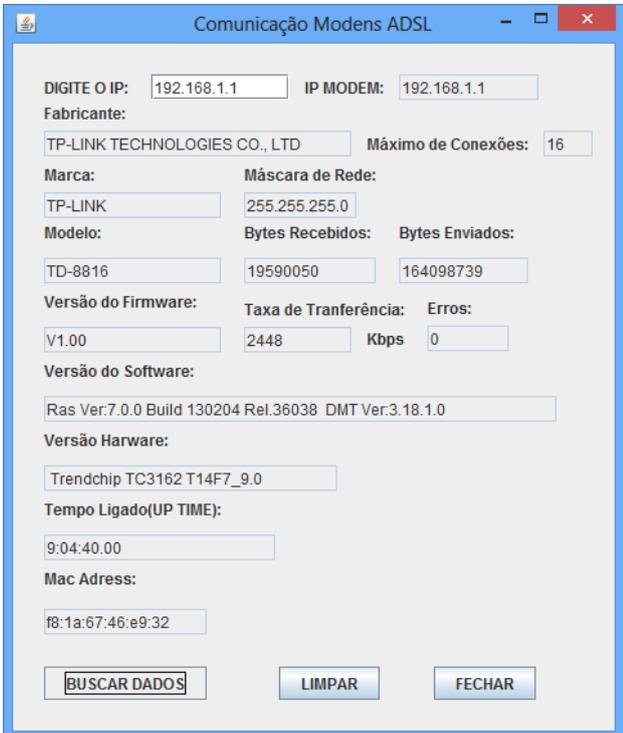
Pode se verificar que o software é eficaz no que se diz respeito ao monitoramento, basta o equipamento estar com o serviço de SNMP habilitado, com o nome da comunidade configurado (PDU) e conectado ao computador que tem o programa instalado que todos os dados são trazidos de forma instantânea de acordo com o endereço IP digitado no campo editável para esse fim. Não é necessário login no browser como se faz para se acessar comumente modems ADSL para sua configuração, assim facilitando ainda mais esse gerenciamento por qualquer indivíduo. Após várias pesquisas constatou-se que, não existe software que faça esse tipo de gerenciamento da mesma forma do que foi implementado.

7 ESTUDO DE CASO

Foi utilizado arbitrariamente para a configuração e testes o IP 192.168.1.1 pelo motivo de o modem em questão já estar com este IP, mas poderia ser usado qualquer outro endereço IP que será monitorado da mesma forma, o modem do modelo TP-8816 está com o SNMP ativado e com a comunidade (PDU) com o nome de PDU para facilitar o entendimento nos testes e conectado via cabo em uma interface ethernet. Dessa forma os testes ocorreram sem nenhum problema.

Verificou-se após a execução do protótipo que o equipamento está com a versão do firmware versão V1, e em contra partida verificou-se no site do fabricante que já existe atualização para este equipamento até a versão V8, dessa forma a correção poderá evitar várias vulnerabilidades e possíveis problemas no equipamento visando a gerência de configuração (Figura 12).

FIGURA 12
Funcionamento do Modem



The screenshot shows a software window titled "Comunicação Modems ADSL" with various fields for modem configuration. The fields are as follows:

| | | | |
|------------------------|---|------------------------|---------------|
| DIGITE O IP: | 192.168.1.1 | IP MODEM: | 192.168.1.1 |
| Fabricante: | TP-LINK TECHNOLOGIES CO., LTD | | |
| Máximo de Conexões: | 16 | | |
| Marca: | TP-LINK | Máscara de Rede: | 255.255.255.0 |
| Modelo: | TD-8816 | Bytes Recebidos: | 19590050 |
| | | Bytes Enviados: | 164098739 |
| Versão do Firmware: | V1.00 | Taxa de Transferência: | 2448 Kbps |
| | | Erros: | 0 |
| Versão do Software: | Ras Ver:7.0.0 Build 130204 Rel.36038 DMT Ver:3.18.1.0 | | |
| Versão Hardware: | Trendchip TC3162 T14F7_9.0 | | |
| Tempo Ligado(UP TIME): | 9:04:40.00 | | |
| Mac Address: | f8:1a:67:46:e9:32 | | |

Buttons at the bottom: BUSCAR DADOS, LIMPAR, FECHAR.

Software

- V8: Firmware Utilitário
- V7: Firmware Utilitário
- V6: Firmware Utilitário
- V5: Firmware Utilitário
- V4: Firmware
- V3: Utilitário
- V2: Firmware
- V1: Firmware

Fonte: Do Autor (2014)

8 CONCLUSÃO

O protótipo apresentado foi capaz de trazer as configurações necessárias para a gerência de falhas e de configuração, conforme descrito neste artigo e no que diz respeito à prevenção de falhas e a monitoração do equipamento, bem como a checagem do serviço de internet prestado visando melhorar disponibilidade. O protótipo traz as informações como versão de *firmware* instalado, assim mostrando ao usuário se seu equipamento está atualizado, facilitando a manutenção preventiva do mesmo, protegendo-o contra ataques, ameaça ou qualquer outro tipo de vulnerabilidade que possa existir. O protótipo traz também dados como a taxa de transferência, que verifica a disponibilidade e qualidade do sinal de internet, assegurando para o usuário o monitoramento proativo de sua internet, visando a gerência de configuração.

Como a configuração do software e sua utilização é feita de forma fácil pelos usuários, pois só existe um campo (IP) editável para se digitar e o botão buscar registra a requisição, gerando dados relevantes para os usuários, sem que eles necessitem de um conhecimento aprofundado na área de informática e assim gerando valor agregado.

Sendo feito na linguagem Java, o protótipo se torna de fácil manutenção e melhorias de parte dos analistas de sistemas, por ser uma linguagem amplamente difundida e bastante utilizada atualmente no mercado mundial.

Pode-se verificar então que o projeto atente as demandas impostas para suas funcionalidades iniciais e ainda existe a possibilidade de melhoria e crescimento do mesmo se necessário.

REFERÊNCIAS

AGÊNCIA NACIONAL DE TELECOMUNICAÇÕES. Divulgados resultados das medições da banda larga no primeiro trimestre de 2014. 2014. Disponível em: <<http://www.anatel.gov.br/>>. Acesso em: 24 jun. 2014.

ASSOLINI, Fabio. **O conto dos 1001 modems ADSL**. São Paulo, SP: KarsperiskyLab, 2012. Disponível em: <<http://brazil.kaspersky.com/sobre-a-kaspersky/centro-de-imprensa/blog-da-kaspersky/conto-modems-adsl>>. Acesso em: 7 maio 2014.

COMMER, Douglas E. **Interligação de rede com TCP/IP Volume 1**. 3. ed. Rio de Janeiro: Campus, 1998. 2 v.

COSTA, Daniel G. **Java em Redes**. 1. ed. Rio de Janeiro: Brasport, 2008.

FOROUZAN, Behrouz A. **Data communication and networking**. 4th. ed. New York: McGraw-Hill, 2007.

JAVA. Disponível em: <<https://www.java.com>>. Acesso em 24 jun. 2014.

MAURO, Douglas R; SCHMIDT, Kevin J. **Essential SNMP**. 2nd. ed. Sebastopol: O'Reilly, 2005.

MRTG. Disponível em: <<http://oss.oetiker.ch/mrtg/>>. Acesso em: 24 jun. 2014.

NAGIOS. Disponível em: <<http://www.nagios.org>>. Acesso em: 24 jun. 2014.

NET-SNMP. Disponível em: <<http://www.net-snmp.org/>>. Acesso em: 24 jun. 2014.

PETERSON, Larry L; DAVIE, Bruce S. **Redes de Computadores**. 5. ed. Rio de Janeiro: Elsevier, 2013.

SUBRAMANIAN, Mani. **Network Management**. 2nd. ed. New Jersey: Dorling Kindersley, 2010.

SNMP4J. Disponível em: <<http://www.snmp4j.org/index.html>>. Acesso em: 24 jun. 2014.

TANENBAUM, Andrew S; WETHERALL, David J. **Computer Networks**. 5th. ed. Boston: Pearson, 2011.

ZABBIX. Disponível em: <<http://www.zabbix.com/>>. Acesso em: 24 jun. 2014.