



UNIVERSIDADE DO MINHO
ESCOLA DE ENGENHARIA

MESTRADO INTEGRADO EM ENGENHARIA DE TELECOMUNICAÇÕES E INFORMÁTICA

GESTÃO DE REDES

Trabalho Prático Nº2

LUÍS FILIPE FREITAS FERREIRA A70016



EDUARDO MIGUEL MENDES DA SILVA A70216



31 de Janeiro de 2017

Índice

	Página
1 Introdução	5
2 Objetivos	6
3 Requisitos	7
4 Ferramenta SNMP para Monitorização IP	8
4.1 Arquitetura da Solução	8
4.1.1 Objetos escolhidos e intervalo de <i>polling</i>	9
4.2 Conceção	9
4.3 Resultados Obtidos	11
5 Conclusão	15

Lista de Figuras

	Página
1 Janela principal do programa.	11
2 Janela de visualização em forma de gráfico.	11
3 Janela de visualização em forma de lista de valores.	12
4 Janela principal em modo execução.	12
5 Janela principal após alteração do IP.	13
6 Janela principal após alteração do IP.	13
7 Janela principal após alteração do IP.	14

Lista de Tabelas

Página

Lista de exemplos

	Página
1 Método start	10
2 Método getAsString	10
3 Método getTarget	10

1. Introdução

No âmbito da unidade curricular de Gestão de Redes, do curso de Engenharia de Telecomunicações e Informática, foi proposto aos alunos a realização de três trabalhos práticos. Este relatório diz respeito ao trabalho prático número 2.

O principal objetivo deste trabalho prático, consiste em consolidar os conhecimentos adquiridos na unidade curricular, nomeadamente sobre o modelo de gestão preconizado pelo INMF (*Internet Standard Management Framework*) que engloba componentes como o protocolo SNMP (*Simple Network Management Protocol*) e as MIBs (*Management Information Bases*).

Foi pedido aos alunos a conceção de uma ferramenta SNMP para monitorização do número de octetos/bytes que entram e saem das interfaces de rede operacionais de um sistema na rede local.

"if you need motivation don't do it"

Elon Musk

2. Objetivos

Familiarização com a arquitetura e filosofias do modelo de gestão preconizado pelo *Internet-standard Network Management Framework* (INMF), dando especial relevo ao *Simple Network Management Protocol* (SNMP) e às *Management Information Bases* (MIBs).

Saber aplicar APIs SNMP para construção de ferramentas de monitorização IP, assim como, desenvolver experimentalmente essas ferramentas para a observação em tempo real de pacotes que entram e saem de um sistema ou *host*.

Desenvolver conhecimento de implementação em linguagem Java, utilizada para a realização do que é pedido no enunciado do projeto.

3. Requisitos

Sistema Unix com um agente SNMPv2c instalado (preferencialmente o NET-SNMP).

Pacote de desenvolvimento numa linguagem de programação que disponibilize APIs para construção dum gestor SNMPv2c, como por exemplo, linguagem Java.

Pacote de desenvolvimento numa linguagem de programação que disponibilize APIs para a construção de uma interface gráfica amigável para o utilizador de forma a promover uma visualização facilitada dos resultados obtidos.

4. Ferramenta SNMP para Monitorização IP

Neste segundo trabalho prático o objetivo é elaborar uma ferramenta capaz de monitorizar (o mais aproximadamente em tempo real) o número de octetos/bytes que entram e saem das interfaces de rede operacionais dum sistema/host na rede local (endereço IP e porta UDP do agente SNMP devem ser configuráveis pelo utilizador). A ferramenta deve disponibilizar uma qualquer forma de visualização (com atualização o mais rápida e efetiva/útil possível) para que seja fácil ao gestor de rede ver a evolução dos valores.

Nesta ferramenta foi utilizado o protocolo de gestão de redes SNMP. Este protocolo permite descobrir o estado da rede e proceder à sua monitorização.

Para a sua implementação foi usada a biblioteca SNMP4J na linguagem Java, obtendo-se assim todas as funcionalidades do protocolo SNMP.

4.1 Arquitetura da Solução

Na nossa arquitetura são utilizadas duas classes, a *MainProject* e a *GraphicInterface*. A classe *MainProject* apenas executa a classe *GraphicInterface* sendo esta considerada a principal com diversas implementações, tais como:

- Configuração do endereço IP e porta UDP do agente SNMP pelo utilizador;
- Utilização das variáveis/objetos da MIB-2;
- Definição do intervalo de *polling*;
- Interface gráfica.

Em relação à MIB-2 foram utilizados os dois objetos, sendo eles:

- *ifInOctets*(.1.3.6.1.2.1.2.2.1.10.1) que representa o número total de octetos recebidos na interface.
- *ifOutOctets*(.1.3.6.1.2.1.2.2.1.16.1) que representa o número total de octetos transmitidos para fora da interface.

4.1.1 Objetos escolhidos e intervalo de *polling*

Os objetos escolhidos foram os que na nossa perspectiva mais se adequavam ao que era pedido uma vez fazendo *snmpget* de cada um dos OIDs obtinha-se o número de bytes que entravam ou saíam.

Quanto ao intervalo de *polling* foi decidido que seria de 1 segundo. Este valor foi assim definido porque como se trata de uma ferramenta de monitorização em tempo real, o tempo não deve exceder os 5 segundos e quanto maior fosse esse tempo a margem de erro aumentava e a probabilidade de se obter dados incorretos também. Assim, é permitido ao utilizador analisar os dados uma vez que estes se encontram em constante atualização.

4.2 Conceção

A biblioteca SNMP4J permitiu através do endereço IP e da porta UDP identificar o agente SNMP correspondente e a execução de comandos do protocolo. No nosso caso apenas usamos o comando *snmpget*.

Para iniciar uma sessão SNMP utilizamos o método *start*, ilustrada no Exemplo 1, que utiliza a interface *TransportMapping* para usar como recurso o protocolo de transporte. De seguida, é criada uma nova sessão SNMP e devido ao facto da comunicação ser assíncrona é necessário implementar o método *listen* para ficar à escuta de respostas.

Exemplo 1: Método start

```
1     private void start() throws IOException {
2         TransportMapping transport = new DefaultUdpTransportMapping();
3         snmp = new Snmp(transport);
4         transport.listen();
5     }
```

Para a execução do comando *snmpget* implementamos o método *getAsString*, como podemos ver no Exemplo 2 que através do OID devolve em formato *string* o resultado do comando.

Exemplo 2: Método getAsString

```
1     private String getAsString(OID oid) throws IOException {
2         PDU pdu = new PDU();
3         pdu.add(new VariableBinding(oid));
4         pdu.setType(PDU.GET);
5         ResponseEvent event = snmp.send(pdu, getTarget(), null);
6         if (event != null) {
7             return event.getResponse().get(0).getVariable().toString();
8         }
9         throw new RuntimeException("NULL");
10    }
```

Por fim foi implementado o método *getTarget* para definir todos os parâmetros necessários para execução do comando *snmpget*, como podemos observar no Exemplo 3.

Exemplo 3: Método getTarget

```
1     private Target getTarget() {
2         Address targetAddress = GenericAddress.parse(address);
3         CommunityTarget target = new CommunityTarget();
4         target.setCommunity(new OctetString("public"));
5         target.setAddress(targetAddress);
6         target.setRetries(2);
7         target.setTimeout(1500);
8         target.setVersion(SnmpConstants.version2c);
9         return target;
10    }
```

Estas operações ocorrem de acordo com o intervalo de *polling*.

Em relação à interface gráfica, foi utilizada a API Java Swing para a criação de janelas e para a apresentação de gráficos a API XChart.

4.3 Resultados Obtidos

Ao iniciar a aplicação o utilizador depara-se com a janela da Figura 1. Aqui o cliente deve, inicialmente, introduzir o endereço IP desejado, bem como a respetiva porta no formato, por exemplo, 127.0.0.1/5555. O utilizador, neste instante, tem também a possibilidade de terminar o programa carregando no botão *EXIT*.



Figura 1: Janela principal do programa.

Após a confirmação, através do clique no botão *OK*, é então aberto duas novas janelas como podemos observar nas Figura 2 e 3. Estas serão usadas para a exibição dos valores obtidos dos objetos *ifInOctets* e *ifOutOctets*, numa na forma de gráfico e noutra na forma de lista de valores. Esta implementação teve em vista uma melhor organização dos dados de forma a simplificar a interpretação ao utilizador.

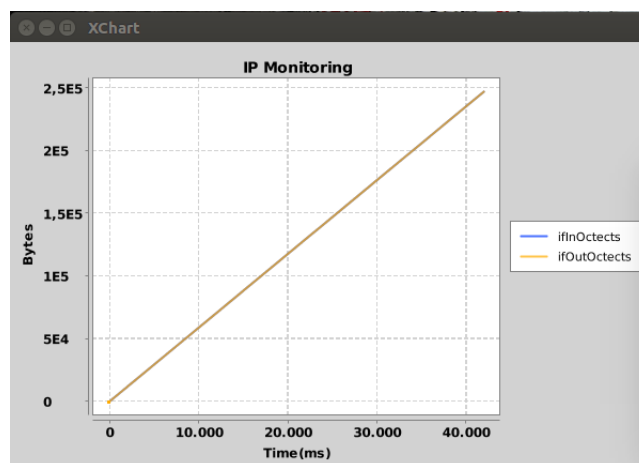


Figura 2: Janela de visualização em forma de gráfico.

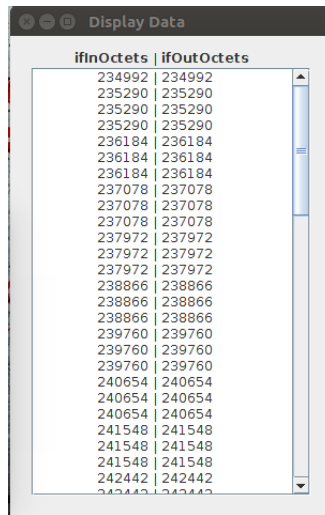


Figura 3: Janela de visualização em forma de lista de valores.

Neste momento a janela principal é modificada de forma a permitir ao utilizador pausar a execução do programa, através de um clique no botão *STOP*, permitindo assim não só a pausa na obtenção dos valores dos objetos como também a mudança do IP e/ou da porta a utilizar, como podemos observar na Figura 4.



Figura 4: Janela principal em modo execução.

Se o objetivo for alterar o IP e/ou a porta UDP do agente SNMP, após esta operação ser concretizada, os valores das janelas de exibição de dados, do gráfico e da lista de valores, desaparecem reiniciando os valores para os novos dados inseridos pelo utilizador. Na Figura 5 podemos observar a utilização de um "novo" IP, inserido após o utilizador ter feito *STOP* ao programa, alterado e feito novamente *OK*.



Figura 5: Janela principal após alteração do IP.

Por fim, nas Figuras 6 e 7 podemos observar que a exibição dos valores foi reiniciada, eliminando os anteriores e mostrando apenas os valores referentes ao novo IP e/ou nova porta.

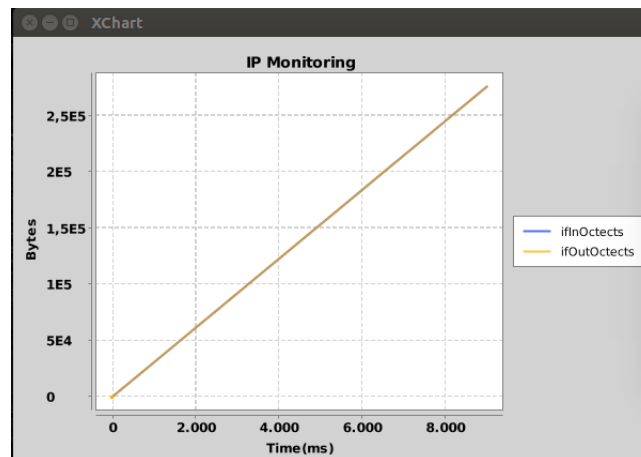


Figura 6: Janela principal após alteração do IP.

Display Data	
ifInOctets	ifOutOctets
267474	267474
267474	267474
267474	267474
268368	268368
268368	268368
268368	268368
269262	269262
269262	269262
269262	269262
270156	270156
270156	270156
270156	270156
271050	271050

Figura 7: Janela principal após alteração do IP.

5. Conclusão

Neste relatório é descrita a solução apresentada para resolução do segundo trabalho prático proposto para a unidade curricular de Gestão de Redes.

Podemos afirmar que todos os objetivos arquitetados para este trabalho foram alcançados, foi possível desenvolver a ferramenta de monitorização IP na totalidade e responder à questão colocada sem quaisquer problemas.

Este trabalho permitiu assim uma melhor consolidação dos conceitos, mecanismos e protocolos subjacentes ao modelo de gestão preconizado pelo INMF, realçando o protocolo SNMP e as MIBs. Permitiu também uma aprendizagem aprofundada sobre as APIs SNMP para a construção de ferramentas de monitorização assim como a sua utilização e aplicação prática.