FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

FEUP

Monitorização de Processos Multimédia

Sandro David Ribeiro Fraga

Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Orientador: Prof. Dr. Maria Teresa Magalhães da Silva Pinto de Andrade

Co-orientador: Eng. Vítor Manuel Lago Teixeira

Junho de 2009

U.PORTO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

MIEEC - MESTRADO INTEGRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES 2008/2009

A Dissertação intitulada

"MONITORIZAÇÃO DE PROCESSOS MULTIMÉDIA"

foi aprovada em provas realizadas em 23/Julho/2009

o júri

Presidente Professor Doutor Eurico Manuel Elias Morais Carrapatoso
Professor Associado do Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Professor Doutor Joaquim Melo Henriques Macedo
Professor Auxiliar do Departamento de Informática da Escola de Engenharia da Universidade do Minho

Professora Doutora Maria Teresa Magalhães da Silva Pinto de Andrade
Professora Auxiliar do Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projecto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extractos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são correctamente citados.

Autor - **SANDRO DAVID RIBEIRO FRAGA**

Faculdade de Engenharia da Universidade do Porto

Resumo

A crescente evolução das tecnologias na área de ambientes de produção digital de televisão tem levado a um aumento do nível de complexidade destes sistemas. Por esse motivo, tornou-se importante desenvolver ferramentas para a monitorização e gestão desses sistemas, permitindo a visualização da ocorrência de problemas em qualquer ponto da cadeia de produção. Dessa forma será possível actuar de imediato minimizando o impacto negativo no processo de produção.

O protocolo SNMP (Simple Network Management Protocol) tem sido o principal protocolo utilizado para gestão e monitorização de redes. No entanto, nos últimos anos, as tecnologias de Serviços Web tornaram-se muito populares para a disponibilização e acesso a serviços em rede. O facto da sua implementação se basear em troca de mensagens XML (através do protocolo SOAP) e de ser possível aceder a esses serviços através de um simples browser web, torna muita atractiva a sua utilização em qualquer tipo de aplicação, nomeadamente na monitorização de sistemas que operam em rede. Uma vez que os principais fabricantes de sistemas de produção de TV oferecem actualmente funcionalidades de monitorização através do protocolo SNMP torna-se importante garantir um caminho de migração que use ambas as tecnologias (SNMP e Servicos Web).

Nesta dissertação o principal objectivo é a implementação de um agente SNMP que possibilite a monitorização de processos multimédia que são disponibilizados com interface SOAP. Estes processos multimédia serão acedidos a partir de uma comunicação feita entre o agente SNMP que os traduzirão para uma interface SOAP, criando um ponte entre as duas tecnologias. Esta ponte permitirá a utilização dos dois protocolos possibilitando que seja possível adicionar novos módulos de processamento. A utilização destas interfaces SOAP é cada vez mais comum nas aplicações desenvolvidas nesta industria, pelo qual se irá procurar uma forma de possibilitar que este tipo de monitorizações seja possível, procurando estudar a melhor forma de o fazer, e estudar problemas que poderão estar relacionados.

Abstract

The growing trend of technology in the manufacturing environments of digital television has led to an increased level of complexity of these systems. Therefore, it became important to develop tools for monitoring and management of these systems, allowing the visualization of the occurrence of problems at any point in the supply chain. That way it will be possible to act immediately and minimize the negative impact in the production process

The SNMP (Simple Network Management Protocol) has been the main protocol used for management and monitoring of networks. However, in recent years, the technologies of Web services have become very popular for the provision and access to network services. The fact that the implementation is based on exchange of XML messages (via SOAP protocol) and be able to access these services through a simple web browser, makes it very attractive to use in any type of application, particularly in monitoring systems operating network. Since the leading manufacturers of production systems now offer features TV monitoring through SNMP becomes important to ensure a path of migration that use both technologies (SNMP and Web Services).

In this dissertation the main objective is the implementation of an SNMP agent that allows the monitoring of processes that are provided with SOAP interface. These multimedia files are accessed from a communication made between the SNMP agent that lead to a SOAP interface, creating a bridge between the two technologies. This bridge will allow the use of this two protocols enabling possible new modules to be added for processing. The use of SOAP interfaces are increasingly common in applications developed in this industry, so its important to find a way to enable this type of monitoring, looking how to do so, and examine issues that may be related.

Agradecimentos

Gostaria de agradecer à empresa MOG Solutions, por me ter dado esta oportunidade de desenvolver esta tese de mestrado convosco.

Quero também agradecer a todos os colaboradores da empresa, pelo convívio e pela ajuda que me prestaram sempre que necessitei.

Ao Eng. Vitor Teixeira pelo seu apoio durante o desenvolvimento da tese e pela suas sugestões para a revisão do documento.

Ao Eng. Paulo Costa por me ter ajudado com os seus conselhos e sugestões para o desenvolvimento desta tese.

Gostaria de agradecer à Professora Maria Teresa Andrade pela sua ajuda e sugestões para a revisão do documento.

Ao meu grande amigo Sergio Sá, pelo seu apoio, disponibilidade e amizade, durante esta longa etapa como estudantes.

À minha querida namorada, Maria Isabel Martins da Rocha, pelo seu apoio, e paciência durante este ano complicado, com muito trabalho e muita pouca disponibilidade.

A todos os meus amigos, colegas e familiares que me ajudaram a chegar aqui, o meu profundo agradecimento.

Quero dedicar esta tese aos meus pais, por me terem ensinado a nunca desistir, a lutar sempre por aquilo que queremos. Nunca teria chegado aqui sem o vosso apoio e conselhos, Obrigado!

Sandro Fraga

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas e Símbolos

API	Application Programming Interface
ASN.1	Abstract Syntax Notation One
CORBA	Common Object Request Broker Architecture
CPU	Central Processing Unit
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
MIB	Management Information Base
MXF	Material Exchange Format
NMS	Network Management System
OSI	Open Systems Interconnection
RMON	Remote Network Monitoring
RPC	Remote Procedure Call
SGMP	Simple Gateway Monitoring Protocol
SMI	Structure of Management Information
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
TCP	Transmission Control Protocol
UDDI	Universal Description, Discovery and Integration
UDP	User Datagram Protocol
W3C	World Wide Web Consortium
WSDL	Web Service Definition Language
XML	Extensible Markup Language

Capítulo 1

Introdução

Perante os avanços tecnológicos nos ambientes de produção de televisão, com a entrada no mercado de televisão de novos equipamentos e tecnologias que vieram melhorar o processos de trabalho dos distribuidores e produtores, começaram a surgir novos problemas inerentes a crescente complexidade nas redes, bem como complexidade na monitorização destes sistemas baseados nestas novas tecnologias.

Existe então uma necessidade de monitorizar tanto os equipamentos, como a informação que é transmitida. Um exemplo simples para descrever este problema será, na necessidade que uma cadeia de televisão, que esteja a fazer captação de dados (áudio e vídeo) num local e a transferir esses mesmos dados com informação adicional (metadados - que é a informação adicional sobre os outros dados) para um outro local em tempo real (onde por exemplo será efectuado um tratamento dos dados para fazer a inserção de efeitos visuais, e a criação da própria cena que será exibida.) garante que não existam problemas nas transmissões.

O conceito de monitorização não é uma novidade quando estamos a falar de monitorizar e gerir dispositivos que estão ligados numa rede TCP/IP. Faz mais de 20 anos que surgiu um protocolo de gestão e monitorização que acabou por revolucionar esta necessidade que existiu e existe, a que denominaram de SNMP (Simple Network Management Protocol). Este protocolo sofreu modificações durante os anos e estamos nos dias de hoje na terceira versão deste protocolo, sendo a grande razão do seu sucesso o facto de ser muito simples, tanto de utilizar, como de implementar.

Existem várias soluções para a gestão de redes, sendo o SNMP a principal solução para tal, contudo há uns anos o conjunto de tecnologias normalizadas pelo W3C (World Wide Web Consortium), conhecido como Web Service levantou a atenção dos investigadores da área, pela possibilidade que este conjunto de tecnologias teria para uma possível implementação a serem utilizadas para gestão e monitorização de redes. Os Serviços Web são constituídos por um conjunto de normas e recomendações, baseados em XML (eXtensible Markup Language) para construção e integração de aplicações distribuídas na Web, focando a interoperabilidade e a independência de plataforma.

A opção de utilizar tecnologia baseada em Serviços Web para substituir o protocolo SNMP na monitorização e gestão de redes, só será verdadeiramente uma opção quando os principais intervenientes neste meio decidirem tomar esse passo. Neste momento existem vários estudos relativamente a uma possível migração, ou de uma possível adaptação entre as duas tecnologias.

1.1 Objectivo

Esta dissertação tem por objectivo especificar e desenvolver uma ferramenta de monitorização de processos multimédia em ambientes de produção de televisão. Para atingir esse objectivo, será necessário desenvolver uma série de estudos e análises dos processos actuais de produção de TV em rede, assim como dos mecanismos de monitorização a operar actualmente nestes ambientes. Para além disso, é necessário estudar as próprias tecnologias que se pretende utilizar no desenvolvimento desta ferramenta, nomeadamente o protocolo SNMP e o conjunto de especificações associados a Serviços Web, e formas de promover a interoperação entre as mesmas.

No final desta tese é esperado ter um agente SNMP totalmente funcional, capaz de comunicar a partir de qualquer uma das versões do protocolo e respondendo a pedidos de um gestor os dados relativos à monitorização do software em causa, que irá ser utilizado neste estudo. Para além disso, procurando também estudar que problemas estarão inerentes à implementação desta solução.

1.2 Descrição do Problema

O problema que se propõem resolver nesta tese de mestrado, estará na pesquisa de uma solução relativa à interligação entre os sistemas SNMP e SOAP. Para esse efeito, foi usado um software da empresa Media, Objects and Gadgets S.A.(MOG Solutions), o Toboggan Media Transfer Direct Write, que será abordado mais à frente nesta tese, mas que servirá como software base para o desenvolvimento deste trabalho uma vez que tem um interface SOAP responsável por manter as informações relativas aos processos que ocorrem neste software.

1.3 Estrutura da Dissertação

Para além da introdução, esta dissertação contém mais 4 capítulos.
No capítulo 2 é feita uma descrição do estado da arte onde são apresentadas as tecnologias e trabalhos relacionados com o trabalho desta tese.
No capítulo 3 é feita uma descrição sobre a implementação desenvolvida, procurando explicar o processo de desenvolvimento, a sua estrutura e como é que as tecnologias funcionam entre elas.
No capítulo 4 está uma breve síntese dos testes funcionais feitos a partir dos vários clientes SNMP, neste capítulo apenas se procurou estudar a funcionalidade da aplicação desenvolvida, procurando validar o seu funcionamento.
No capítulo 5 No último capítulo temos a conclusão e trabalho futuro.

Capítulo 2

Estado da Arte

Neste capítulo será apresentado o estado das tecnologias com que o trabalho desta tese aborda. Numa primeira fase, uma visão sobre o protocolo SNMP, a sua evolução e seu estado actual, e numa segunda fase a possibilidade levantada por vários investigadores, de olhar para as tecnologias baseadas em XML como uma hipótese clara para modificar a forma monitorizamos os dispositivos a partir de SNMP.

2.1 Protocolo SNMP

O SNMP (Simple Network management Protocol) é parte do conjunto de especificações desenvolvidas pelo IETF (Internet Engineering Task Force). Foi inicialmente introduzido em 1988 para responder à crescente necessidade de uma norma para a gestão de dispositivos de rede IP (Internet Protocol). O SNMP fornece aos seus utilizadores um "simples"conjunto de operações que permite que esses dispositivos sejam geridos remotamente.

Enquanto que o seu antecessor, o SGMP (Simple Gateway Management Protocol) foi desenvolvido para gerir routers de internet, o SNMP pode ser usado para gerir vários sistemas e dispositivos. Qualquer dispositivo que possa correr um software que permita a troca de informações SNMP, pode ser gerido. Não estamos exclusivamente a falar de dispositivos físicos, mas também, de software, como servidores Web e base de dados.

Outro aspecto da gestão de redes é a sua monitorização, ou seja, mas de monitorizar dispositivos individualmente, podemos gerir redes inteiras. A RMON (Remote Network Monitoring) foi desenvolvida para ajudar a perceber como é que a rede funcionava, assim como é que alguns dispositivos afectavam o bom funcionamento da rede como um todo. Na sua especificação base é incluída a definição da arquitectura do modelo de gestão, a definição das estruturas de dados e o protocolo de comunicação (Mauro e Schmidt, 2005).

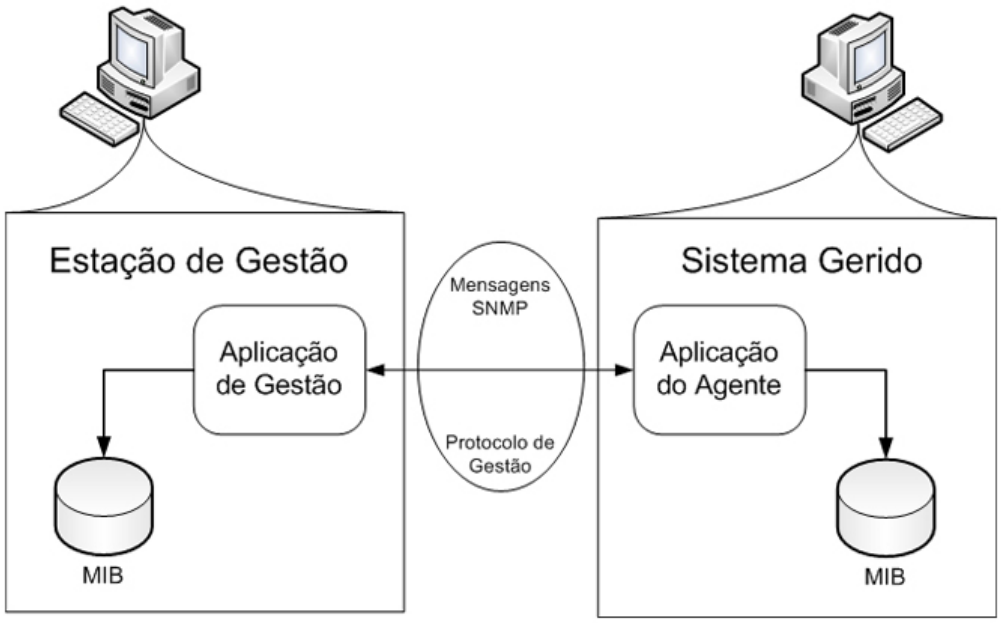

Figura 2.1: Arquitectura do protocolo SNMP

2.1.1 Arquitectura

O modelo de gestão de redes SNMP consiste nos seguintes componentes chave:

- A **estação de gestão** é responsável por fazer o interface entre o sistema de gestão da rede e o gestor da rede. Para além disso tem de incluir as seguintes características:
  - Tem de ter um interface com o gestor de rede capaz de monitorizar e controlar os elementos da rede.
  - Uma base de dados, capaz de armazenar a informação de gestão de todas as entidades da rede.
  - Um conjunto de aplicações de gestão para análise de dados e recuperação de falhas.
- Os **agentes SNMP** são entidades que residem nos dispositivos a serem geridos ou monitorizados. Os agentes são as componentes mais importantes dos sistemas de monitorização e fornecem as seguintes funcionalidades:
  - Implementação e manutenção dos objectos da Management Information Base.
  - Responde a pedidos de operação de gestão.
  - Gera notificações, como traps (unachnowlegded), e informações (acknowlegded).
  - Implementa segurança - SNMPv1 e SNMPv2c baseada em chaves de comunidade. A segurança mais elevada (autenticação e encriptação) está disponível na versão mais recente, o SNMPv3.
  - Define as políticas de acesso a gestores externos.

- **Informação de gestão (MIB)** - pode ser descrito como uma base de dados dos objectos que estão a ser geridos ou monitorizados. Qualquer tipo de informação de status e estatística que pode ser acedida pela por um sistema SNMP tem que estar definida na MIB. A SMI fornece uma forma de definir os objectos geridos enquanto que a MIB é a própria definição dos objecto. Podemos interpretar as MIBs da mesma forma que um dicionário, a MIB define o nome do objecto a ser gerido e explica o seu significado.

- **Protocolo de gestão** - protocolo utilizado na comunicação entre a estação de gestão e o agente (SNMP), utiliza o protocolo UDP (User Datagram Protocol) para transporte e está localizado ao nível da aplicação. As principais funcionalidades genéricas suportadas são as seguintes:

  - **Get** - Permite a estação de gestão obter um certo valor de um objecto de um agente.
  - **Set** - Permite a estação de gestão definir um certo valor ao objecto de um agente.
  - **Trap** - Permite o agente notificar a estação de gestão de alguma ocorrência importante.

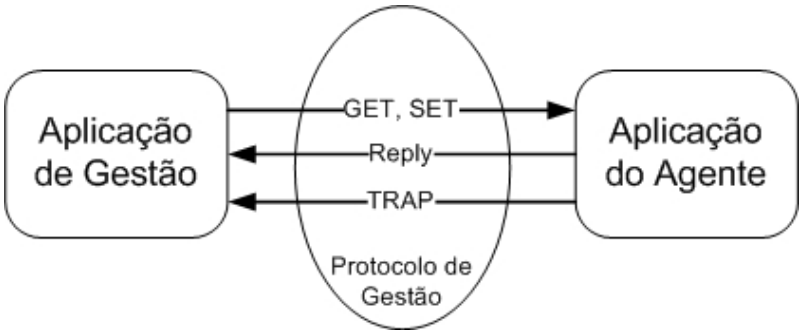

Figura 2.2: Operações básicas do Protocolo SNMP.

2.1.2 Management Information Base

A MIB(Management Information Base) poderá ser vista como uma base de dados dos objectos passíveis de serem geridos. Ela é composta por um conjunto de objectos relativos aos recursos que se pretende gerir.

2.1.2.1 Definição de um Objecto

A definição dos objectos de gestão são definidos pela SMI(Structure of Management Information) (McCloghrie e Rose, 1990) e a sua definição base tem as seguintes componentes:

- **OBJECT-TYPE** - Descrição do objecto na MIB e o seu identificador, OID(Object Identifier)

- **SYNTAX** - Define qual o tipo de dados será adicionado ao objecto MIB.

- **ACCESS** - O tipo de acesso que o objecto tem. READ-ONLY, permite apenas que sejam aplicadas sobre ele pedidos de leitura como Get, Get-Next e o READ-WRITE permite todas as operações de READ-ONLY, mais as operações de escrita, como o Set.
- **STATUS** - Este campo está relacionado com o estado deste objecto em relação à comunidade SNMP.
- **DESCRIPTION** - Neste campo é descrito a razão de existência deste objecto.

Para se definir a estrutura de cada objecto é utilizado a linguagem ASN.1 (Abstract Syntax Notation), e segue um exemplo da sintaxe que deverá ser utilizada:

```
<Nome do Objecto> OBJECT-TYPE
SYNTAX <Tipo de dados>
ACCESS <read-only, read-write, write-only, not-accessible>
STATUS <mandatory, optional, obsolete>
DESCRIPTION
"Descrição textual do objecto em causa."
::= { <OID Identificador do Objecto> }
```

2.1.2.2 Structure of Management Information

Para cada objecto existe os seguintes tipos primitivos para definir o tipo de dados utilizados no protocolo SNMP, que são definidos pela SMI(Structure of Management Information):

- **Integer**
- **Octetstring**
- **OBJECT IDENTIFIER**
- **NULL**.

Existe também tipos de dados mais complexos denominados "*Constructor Types*":

- **SEQUENCE** - Utilizado para tabelas.
- **SEQUENCE OF** - Utilizado para listas.

Os objectos de gestão são organizados hierarquicamente na MIB, numa árvore de objectos. Cada objecto é definido pelo seu OID (Object Identifier) que é um identificador constituído por um conjunto de números separados por pontos, ou poderá ser definido por um conjunto de palavras separadas por pontos transformando o processo de identificação mais legível. O OID é lido da root da árvore até ao nó que se pretende, criando assim o identificador pretendido. Na figura 2.3 temos a árvore de objectos definidos na SMIv1.

O nó mgmt(2) foi criado no intuito de incluir as informações relativas aos objectos de gestão

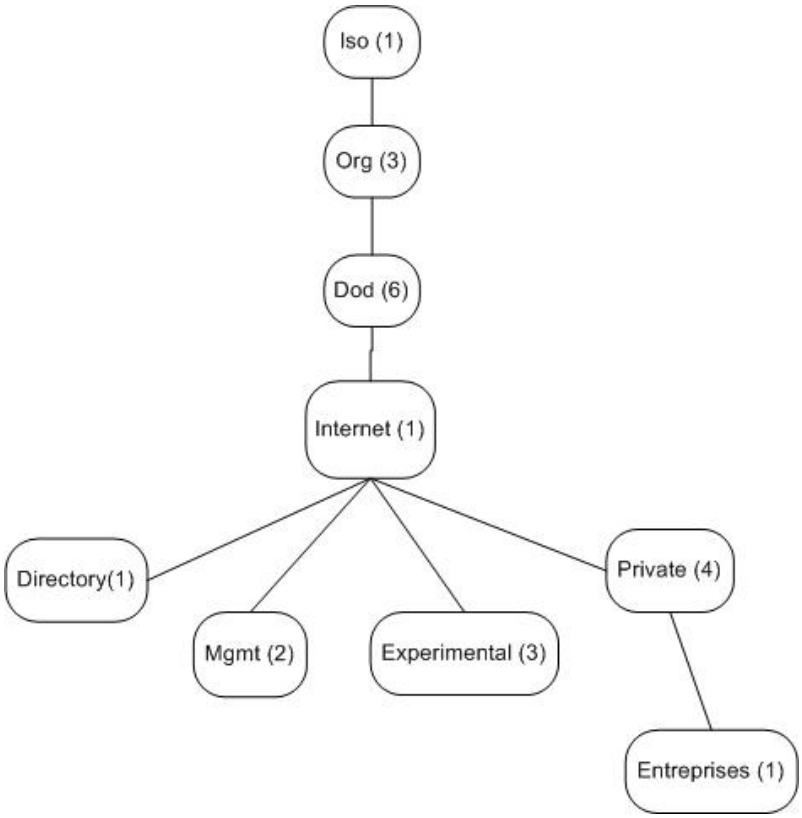

Figura 2.3: SMIv1

de internet. Abaixo deste nó temos a MIB standard da internet.

O nó experimental(3), como o próprio nome sugere foi criado para testes e desenvolvimento.

O nó private(4) que contém o nó Entreprises(1) abaixo dele foi criado para que os fabricantes de software e hardware pudessem definir os seus próprios objectos. Este nó foi o que impulsionou parte do sucesso do protocolo SNMP, permitindo o crescimento dos objectos de uma forma organizada, dando a cada empresa a liberdade de organizar os seus objectos da forma que entendesse. A atribuição e gestão dos nós, abaixo do nó entreprises(1) é feita pela IANA, podendo ser atribuídos a empresas, instituições, organizações, e mesmo a indivíduos.

2.1.2.3 SMIv2

O surgimento da segunda versão do protocolo SNMP (SNMPv2), levou ao aparecimento da segunda versão da SMI (SMIv2), que não teve o papel de substituir a versão anterior, apenas de expandir as funcionalidades existentes.

Com a SMIv2 (K. McCloghrie e Schoenwaelder, 1999) surgiram também novos tipos de dados, como por exemplo:

- **Integer32**
- **Counter32**

- **Gauge32**
- **Unsigned32**
- **Counter64**
- **BITS**

Foram também introduzidos convenções textuais, que permitiram a criação de objectos de uma forma mais abstracta, alguns desses objectos estão ilustrados a seguir:

- **Display String** - Informação textual do conjunto de caracteres ASCII.
- **MacAddress**- Endereço MAC, representado por seis octetos.
- **PhysAddress** - Endereço físico representado com uma OCTET STRING.
- **TimeStamp** - Valor do objecto sysUpTime numa ocorrência.

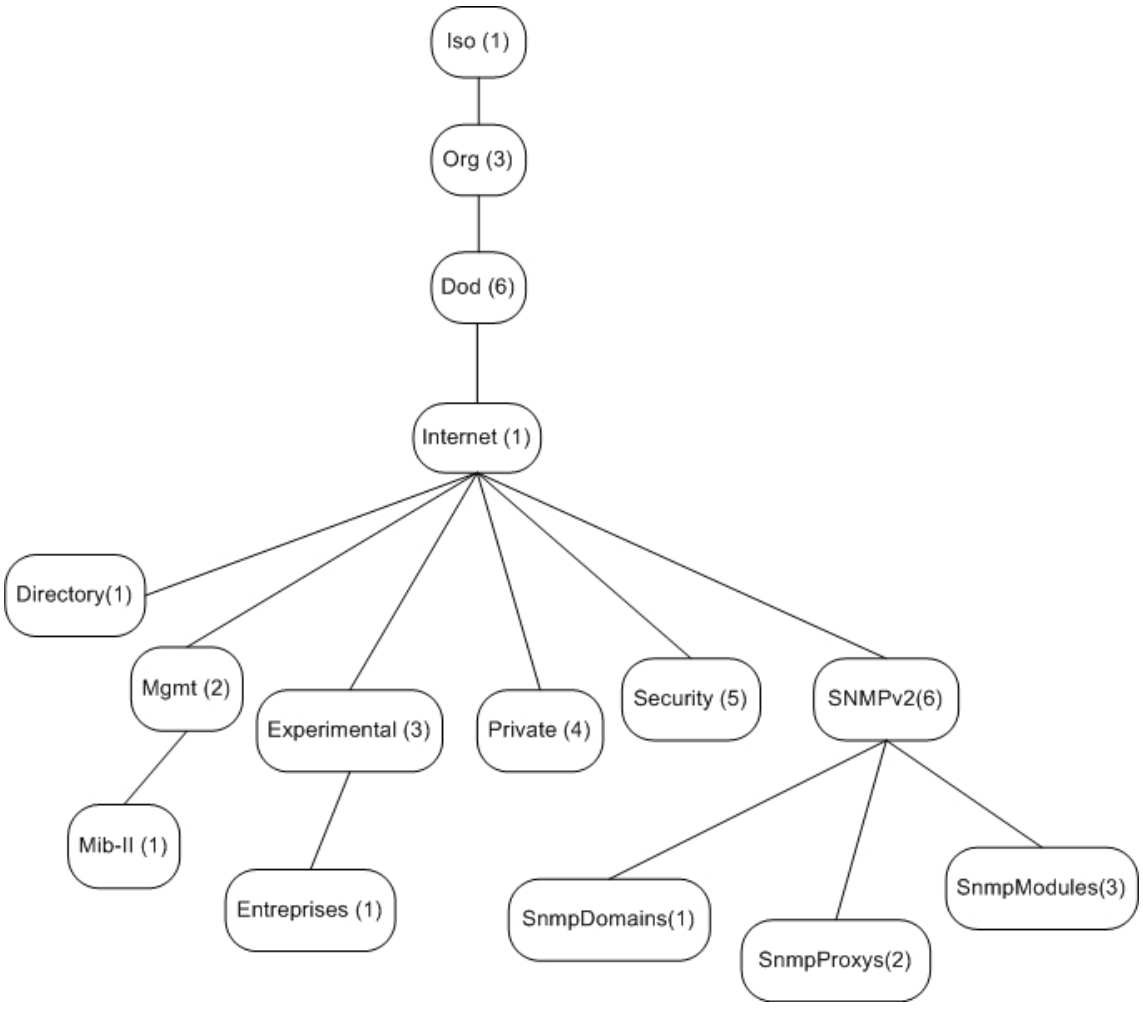

Figura 2.4: SMIv2

Uma outra modificação foi a estrutura de definição de um objecto. Esta nova estrutura permite um melhor controlo no acesso ao objecto. Para declarar um objecto é utilizada a seguinte sintaxe:

```
<Nome do Objecto> OBJECT-TYPE
SYNTAX <Tipo de dados>
UnitsParts <Descrição textual das unidades usadas>
MAX-ACCESS <read-only, read-write, read-create, not-accessible,
 accessible-for-notify>
STATUS <current, obsolete, deprecated>
DESCRIPTION
"Descrição textual do objecto em causa."
AUGMENTS { <Nome da Tabela> }
::= { <OID Identificador do Objecto> }
```

2.1.2.4 Management Information Base - II

Com a evolução do protocolo SNMP e da SMI, surgiu também a evolução da MIB, a MIB-II que é actualmente a mais importante do protocolo SNMP, na figura 2.5 temos os vários ramos que foram acrescentados com esta nova MIB, e que possibilitam uma mais ampla monitorização de parâmetros.

De um modo geral, o agente estrutura a informação que poderá vir a ser gerida ou monitori-

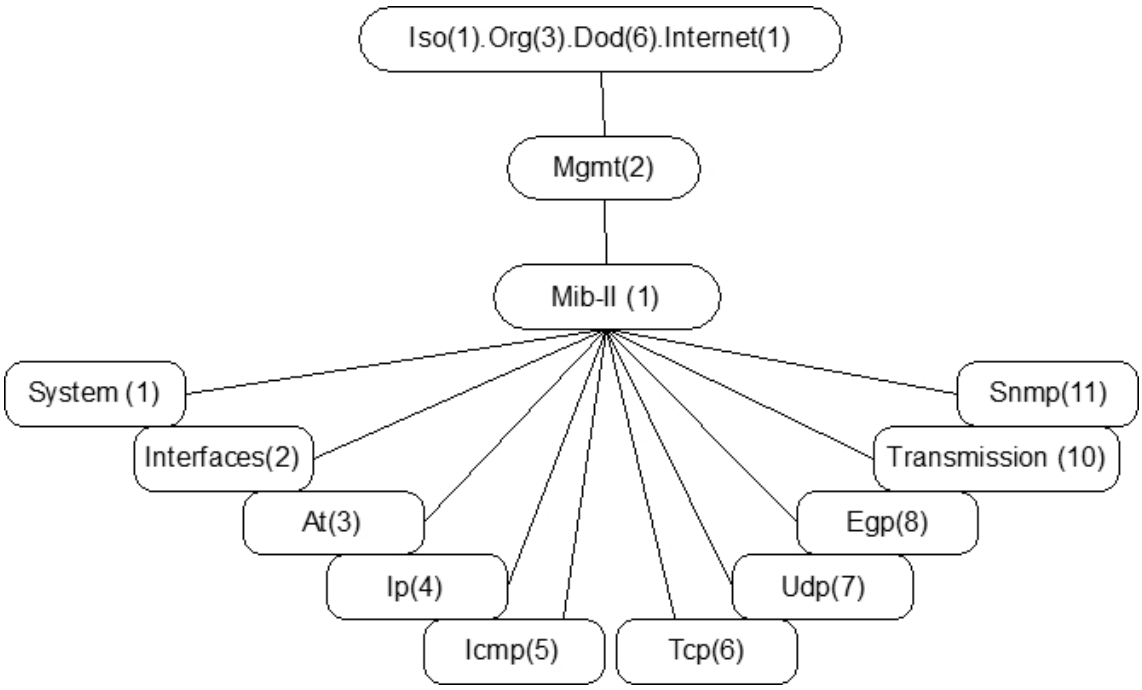

Figura 2.5: Mib-2

zada numa MIB (em ambas as versões da MIB). Essa estrutura terá de ser conhecida pelo gestor

que pretender monitorizar os dados que este agente estará programado a fornecer. Uma vez que um gestor está habilitado a gerir vários dispositivos, terá de ter configurado as várias informações das várias MIB dos dispositivos do qual pretende monitorizar.

2.1.3 SNMPv2

A primeira versão do protocolo tinha uma segurança muito limitada, e o mecanismo de comunidades não garantia uma transmissão segura de mensagens SNMP, sendo portanto falível a ataques de espionagem. Assim, qualquer intruso poderia observar uma mensagem, retirando a informação necessária para poder atacar uma rede, podendo modificar as suas configurações. De modo a evitar esses ataques, os fabricantes restringiram a utilização de alguns comandos, o que acabou por limitar as funções de monitorização. Com o objectivo de superar estas falhas foram versões 2 e 3 do protocolo SNMP.

A SNMPv2 oferecia novas funcionalidades e uma maior eficiência que a versão original. Como referimos anteriormente introduziu novas operações, às quatro operações básicas. No sentido de permitir a transferência de grandes quantidades de informação, comunicação entre estações de gestão e normalização de mensagens de notificação foram criadas as seguintes novas operações:

- **GetBulkRequest** - Sendo o objectivo a transferência de grandes quantidades de informação, como a transferência de uma tabela.

- **InfomRequest** - Introduzida no sentido de possibilitar a comunicação entre estações de gestão.

- **SNMPv2-Trap** - Semelhante ao Trap do SNMPv1, mas com algumas pequenas diferenças.

Apesar de os esforços terem sido feitos no sentido de melhorar a segurança, nada de novo foi acrescentado com a SNMPv2, continuando esta versão com deficiências de segurança, sendo esta a maior fraqueza do protocolo SNMP desde a sua criação.

2.1.4 SNMPv3

O SNMPv3 surgiu como forma de combater este problema de segurança que existia nas versões anteriores. Esta versão não tem novas operações, e suporta todas as operações definidas para a SNMPv1 e SNMPv2. Nesta versão foi abandonado a ideia de estação de gestão e agente, para passar a haver entidades SNMP, e temos um exemplo da nova estrutura na figura 2.6. Cada entidade é composta por um motor SNMP e por uma ou mais aplicações SNMP.

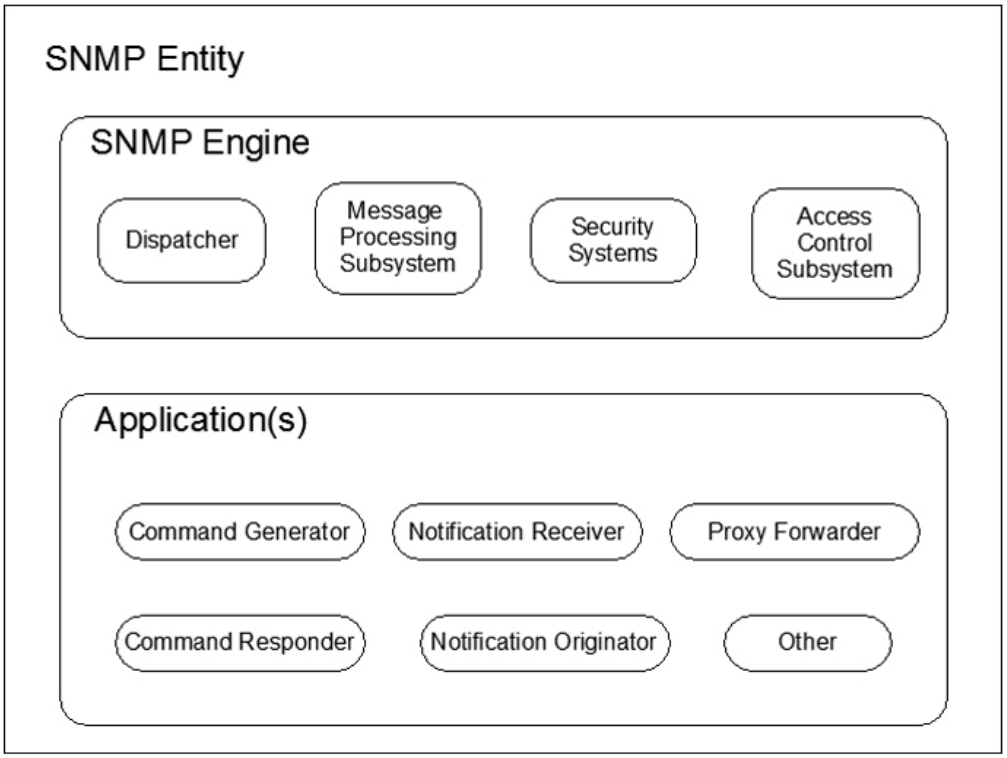

Figura 2.6: Entidade SNMPv3

2.1.4.1 Motor SNMPv3

O motor do SNMPv3 envia e recebe mensagens, autentica e encripta mensagens, controla o acesso a objectos geridos e é composto por quatro partes:

- **Dispatcher** - A função do dispatcher é a de enviar e receber mensagens. Determina qual a versão da mensagem recebida, e caso seja suportada, envia para o subsistema de processamento de mensagens. O dispatcher também envia mensagens para outras entidades SNMP.

- **Subsistema de processamento de mensagens** - Prepara as mensagens a serem enviadas e extrai os dados das mensagens recebidas. Este subsistemas pode ter múltiplos módulos de processamento de mensagens. Por exemplo, um subsistema pode ter módulos para processar pedidos de SNMPv1, SNMPv2, e SNMPv3.

- **Subsistema de segurança** - O subsistema de segurança fornece a autenticação e serviços de privacidade. A autenticação usa tanto a chave da comunidade (SNMPv1 e v2) ou autenticação de utilizadores SNMPv3. Esta autenticação para SNMPv3 usa algoritmos de codificação MD5 ou SHA para autenticar os utilizadores. O serviço de privacidade usa o algoritmo DES para encriptar e desencriptar mensagens SNMP.

- **Subsistema de controlo de acesso** - O subsistema de controlo de acesso é responsável por controlar os acessos a objectos da MIB. Podemos controlar que objectos um utilizador pode

aceder, bem como definir que operações podem ser executadas nesse mesmo objecto. Por exemplo, podemos querer limitar as permissões de read-write a uma certa parte da mib-II, e permitir um acesso read-only para a árvore toda.

2.1.4.2 Aplicações SNMPv3

Quanto as aplicações SNMP, estas utilizam serviços fornecidos pelo motor SNMP para realizarem as suas operações, dentro dos quais existem as seguintes aplicações:

- **Command Generator** - Monitorizam e manipulam dados de gestão (gerar pedidos e processar respostas).
- **Command Responder** - Facilitam o acesso à informação de gestão.
- **Notification Originator** - Gera as mensagens de notificação.
- **Notification Receivers** - Recebem as mensagens de notificação.
- **Proxy Forwarders** - Encaminham mensagens entre entidades.

Uma entidade SNMP que contem uma ou mais, aplicações Command Generator e/ou Notification Receiver chama-se gestor SNMP. E uma entidade que contenha uma ou mais aplicações Command Responder e/ou Notification Originator chama-se um Agente SNMP.

2.1.4.3 Segurança

No sentido de melhorar a segurança quase inexistente nas versões anteriores foram inseridas as seguintes modificações. Foram criadas quatro áreas de segurança:

- Autenticação - identificação da origem, integridade da mensagem e alguns aspectos de segurança na resposta.
- Privacidade - confidencialidade.
- Autorização e controlo de acesso.
- Capacidade de configuração e administração remota para os três aspectos anteriores.

O subsistema de segurança prevê a existência de três modelos, O modelo de segurança "User-Based", modelo de segurança "Community-Based", e outro modelo de segurança. Para além do modelo são definidos três níveis de segurança:

- Sem autenticação e sem privacidade (noAuthNoPriv).
- Com autenticação e sem privacidade (authNoPriv).
- Com autenticação e privacidade (authPriv).

2.1.5 Limitações do SNMP

O protocolo SNMP continua ainda a ser o protocolo dominante na gestão e monitorização de redes IP, contudo os gestores das redes começaram a reparar nalgumas limitações (Soldatos e Alexopoulos, 2007). Limitações no que respeita a configuração de gestão, desenvolvimento de aplicações, e quanto a descentralização das tarefas de gestão, bem como em questões de "escalabilidade e eficiência" (Yoon et al., 2003).

A escalabilidade refere-se ao número de agentes que podem ser geridos/monitorizados num único sistema de gestão.

A eficiência refere-se a velocidade e eficiência que um sistema consegue operar em acções de entrega, como por exemplo em processamento de dados. Ambos estes problemas foram constatados com o aumento da informação trocada em operações de gestão.

No passado, quando a troca de informação era muito inferior à de os dias de hoje, a simplicidade do SNMP era uma grande vantagem para implementar e projectar uma NMS (Network Management Systems). Contudo as redes evoluíram imenso, assim como a informação de gestão trocada na rede e processada em gestores e agentes.

2.2 Web Services na Gestão de Redes IP

Nesta secção será feita uma revisão sobre a arquitectura básica dos Serviços Web e as suas principais normas. Para além disso serão expostas duas soluções possíveis que foram apresentadas pela industria no sentido de utilizar Serviços Web na gestão de redes, bem como uma outra solução, baseada na utilização de gateways, para ser possível gerir dispositivos SNMP num ambiente de gestão baseado em Serviços Web.

Os Serviços Web tem captado o interesse da comunidade cientifica e da industria de gestão de redes IP, como sendo uma potencial alternativa para o protocolo SNMP, dada a sua pertinência tanto para paradigmas de gestão descentralizada (por exemplo, CORBA) e sistemas de gestão baseados em XML, que proporcionam eficiência na gestão e configuração de operações (Soldatos e Alexopoulos, 2007). O problema é que os sistemas de gestão de redes baseados em XML não podem gerir agentes SNMP directamente (Yoon et al., 2003). Havendo trabalho a ser feito nesse sentido e que iremos abordar esse ponto um pouco mais em pormenor. Como resultado desse interesse tem sido levado a cabo uma serie de estudos para avaliar o impacto que a utilização de Serviços Web teria na gestão de redes IP, procurando estudar a performance, utilização da largura de banda, comparação com outros sistemas de gestão/monitorização (como o OSI e o CORBA) (Pavlou et al., 2004) (Lemos Vianna et al., 2006) (de Lima et al., 2006).

2.2.1 Descrição de Web Services

O Serviço Web pode ser simplesmente descrito como uma arquitectura para distribuição de serviços, sendo os componentes da arquitectura independentes da plataforma permitindo intero-

perabilidade entre aplicações. A normalização dos Serviços Web e das tecnologias relacionadas, está a ser feito pelo W3C (Pras et al., 2004), que definiu estas três normas, o SOAP, o WSDL e UDDI. Isso permite que qualquer aplicação capaz de lidar com dados XML e comunicar sobre um protocolo Web (por exemplo: HTTP) pode ser um cliente de um Web Service, independente da plataforma e linguagem de desenvolvimento utilizada.

O SOAP é usado como mecanismo de comunicação entre os Serviços Web e as aplicações dos clientes. A WSDL é uma linguagem usada para descrever as interfaces dos Serviços Web e a UDDI permite aos Serviços Web registarem as suas características num esquema de registo.

A arquitectura básica dos Serviços Web é composta por três elementos: o fornecedor, o consumidor e o registo. O fornecedor está responsável por disponibilizar os serviços que podem ser acedidos pelos clientes (consumidores). Para além disso, o fornecedor pode optar por fazer a publicação de um serviço num esquema de registos, possibilitando assim a um consumidor a pesquisa de serviços mais apropriados às suas necessidades. Na figura 2.7 representa a interacção entre os elementos básicos dessa arquitectura e as normas que estão envolvidas.

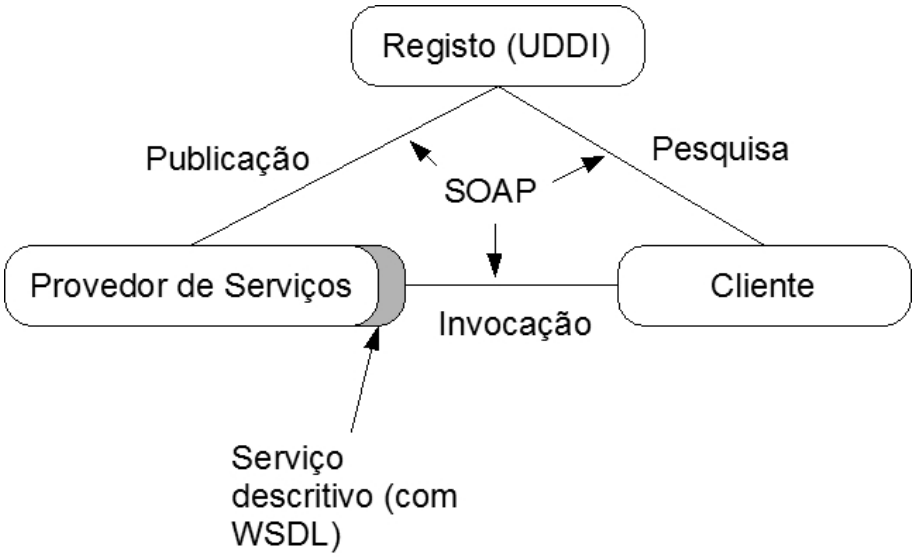

Figura 2.7: Arquitectura dos Servicos Web

2.2.2 SOAP

O SOAP é um protocolo cujo objectivo é o de transferir estruturas de informação num ambiente distribuído e descentralizado. Usa tecnologias XML para definir um formato de mensagens a ser utilizado nas trocas de informação a partir de uma variedade de protocolos. A Framework foi desenvolvida para ser independente de qualquer linguagem de programação e outras semânticas de implementação. Os dois principais objectivos da implementação do SOAP são a simplicidade e a extensibilidade. O SOAP tenta atingir estes objectivos omitindo da Framework de mensagens, informação que é normalmente encontrada em sistemas distribuídos, como por exemplo, a

segurança, correlação, routing, e fiabilidade e message exchange patterns (MEPs) (W3C, 2007b). SOAP define quarto pedaços básicos de informação que veremos de seguida:

- A forma como o XML é estruturada.
- As convenções representando os procedimentos de uma chamada numa mensagem XML.
- Um ligação ao http, para assegurar que as mensagens xml são transportadas correctamente.
- As convenções para representar um mensagem de erro de volta para o emissor.

O SOAP não define um modelo de objecto, simplesmente fornece um Framework adequado para a comunicação de mensagens XML entre emissor e receptor, geralmente chamadas como nós de processamento SOAP. Um nó pode ser emissor ou receptor, ou ambos ao mesmo tempo. Por exemplo, um cliente pode enviar uma mensagem SOAP dentro de um pedido HTTP. O servidor processa a mensagem e retorna a resposta SOAP na mensagem HTTP de resposta.

2.2.2.1 Estrutura

Uma mensagem SOAP é um documento XML bem estruturado que pode conter três elementos. Um envelope, um cabeçalho opcional e um corpo.

O envelope é um elemento que vai identificar a mensagem como sendo uma mensagem do tipo SOAP, e terá no seu conteúdo o Body e o header. A principal função do envelope é indicar o inicio e fim da mensagem ao receptor. Assim que o receptor chegue ao identificador </Envelope>, irá saber que a mensagem chegou ao fim e poderá começar a processa-la. O envelope é simplesmente uma estrutura de empacotamento. O header (ou cabeçalho) é um elemento opcional, mas se estiver presente tem se ser o primeiro elemento do envelope. Contem um bloco de informação relevante de como a mensagem tem de ser processada, isso inclui especificações de entrega, autenticação e transições de contextos. O Body (ou corpo), contem a mensagem actual que se quer enviar ao receptor. Pode ser um elemento XML que descreve a invocação de um procedimento - por exemplo, descrevendo paramentos e argumentos com o nome de um procedimento - ou outros conteúdos XML, com uma lista completa de informações. Estas duas técnicas são geralmente chamadas RPC-style e mensagens SOAP, respectivamente.

2.2.2.2 Transporte SOAP

O SOAP é bastante flexível na forma como é usado, e onde é usado. Uma ilustração dessa flexibilidade é a capacidade de podermos transferir mensagens do tipo SOAP sobre HTTP, FTP, SMTP (W3C, 2007b). Vamos aprofundar mais um pouco a capacidade de transferir mensagens SOAP sobre HTTP. O protocolo HTTP é de longe o protocolo mais utilizado na internet, para troca de mensagens SOAP. As próprias especificações do SOAP dão uma certa atenção especial a este tipo de transporte. Uma mensagem de pedido SOAP é enviado junto com o pedido HTTP e o servidor retorna a resposta da mensagem SOAP na resposta ao http, e isso é feito da seguinte forma:

A seguir temos um exemplo da estrutura de um pedido HTTP contendo a mensagem SOAP (W3C, 2000):

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI"
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <SOAP-ENV:Body>
       <m:GetLastTradePrice xmlns:m="Some-URI">
           <symbol>DIS</symbol>
       </m:GetLastTradePrice>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Eis o exemplo de uma mensagem SOAP embebida na resposta HTTP:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
   <SOAP-ENV:Body>
       <m:GetLastTradePriceResponse xmlns:m="Some-URI">
           <Price>34.5</Price>
       </m:GetLastTradePriceResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.2.3 WSDL

WSDL é linguagem baseada em XML para descrição de serviços de redes como um conjunto de parâmetros operacionais sobre mensagens que sejam tanto orientadas a documentos ou informação orientada a procedimentos. As operações e mensagens são descritas abstractamente, e são depois vinculadas a um protocolo de rede e um formato de mensagem para definir um parâmetro. Parâmetros concretamente relacionados são combinados em parâmetros abstractos (serviços). WSDL é extensível para permitir a descrição dos parâmetros e as suas mensagens, independentemente do formato da mensagem, ou do protocolo de rede a ser usado para a comunicação (W3C,

2007a).

Com a utilização do WSDL, a tecnologia de Serviços Web pode descrever praticamente tudo, descrevendo o que faz, como faz e como os utilizadores dessa Web Service podem utiliza-la. Das várias vantagens do WSDL temos como principais as seguintes descritas a seguir:

1. Permite uma escrita fácil, bem como a manutenção dos serviços, fornecendo uma aproximação mais estruturada para definir dispositivos com tecnologia de Serviços Web.

2. Facilita o consumo de Serviços Web, pois reduz a quantidade de código (e potenciais erros) que as aplicações dos clientes têm de implementar.

3. Facilita a implementação de modificações que serão menos susceptíveis de "quebrar "aplicações de clientes SOAP. A descoberta dinâmica de descrições WSDL que essas modificações sejam implementadas automaticamente nos clientes usando WSDL, assim potenciais implementações mais caras para o cliente não sejam feitas sempre que ocorre uma modificação.

Temos de seguida um exemplo de um documento WSDL, para a descrição de um serviço:

```
<?xml version="1.0"?>
<definitions name="StockQuote"
targetNamespace="http://example.com/stockquote.wsdl"
          xmlns:tns="http://example.com/stockquote.wsdl"
          xmlns:xsd1="http://example.com/stockquote.xsd"
          xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
          xmlns="http://schemas.xmlsoap.org/wsdl/">
    <types>
       <schema targetNamespace="http://example.com/stockquote.xsd"
              xmlns="http://www.w3.org/2000/10/XMLSchema">
           <element name="TradePriceRequest">
              <complexType>
                  <all>
                      <element name="tickerSymbol" type="string"/>
                  </all>
              </complexType>
           </element>
           <element name="TradePrice">
              <complexType>
                  <all>
                      <element name="price" type="float"/>
                  </all>
              </complexType>
           </element>
       </schema>
```

```
    </types>
    <message name="GetLastTradePriceInput">
        <part name="body" element="xsd1:TradePriceRequest"/>
    </message>
    <message name="GetLastTradePriceOutput">
        <part name="body" element="xsd1:TradePrice"/>
    </message>
    <portType name="StockQuotePortType">
        <operation name="GetLastTradePrice">
           <input message="tns:GetLastTradePriceInput"/>
           <output message="tns:GetLastTradePriceOutput"/>
        </operation>
    </portType>
    <binding name="StockQuoteSoapBinding"
     type="tns:StockQuotePortType">
        <soap:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="GetLastTradePrice">
           <soap:operation
           soapAction="http://example.com/GetLastTradePrice"/>
           <input>
               <soap:body use="literal"/>
           </input>
           <output>
               <soap:body use="literal"/>
           </output>
        </operation>
    </binding>
    <service name="StockQuoteService">
        <documentation>My first service</documentation>
        <port name="StockQuotePort" binding="tns:StockQuoteBinding">
           <soap:address location="http://example.com/stockquote"/>
        </port>
    </service>
</definitions>
```

2.2.4 UDDI

A UDDI (Universal Description, Discovery and Integration) é uma tecnologia que permite às organizações uma rápida pesquisa e publicação de Serviços Web (OASIS, 2004b). A UDDI consiste numa especificação da estrutura de dados a ser utilizada na representação de negócios e seus

serviços e da especificação de uma API para o acesso e armazenamento de tais informações. Utiliza XML para descrever os seus negócios e serviços, bem como detalhes de acesso a cada serviço. Esta estrutura de dados, ou registo UDDI é definida de modo a facilitar a publicação e pesquisa de negócios e serviços. Esse registo que pode ser visto de forma análoga como uma lista telefónica, fornece três "listas", páginas amarelas (negócios e serviços organizados por categorias), páginas brancas (negócios e serviços organizados pelo nome), páginas verdes (informações técnicas sobre os serviços).

Um registo contém tipos de informações:

- *Business entity*, neste elemento encontram-se informações básicas sobre o negócio, como o nome, a descrição do negócio, uma lista de categorias que descreve e classifica o negócio.
- *Business service*, qualquer entidade business service contem uma descrição do negocio, incluído uma lista de categorias que o descreve e classifica este serviço.
- *Specification pointers*, Qualquer entidade business service tem uma lista de templates vinculados que apontam para informação adicional sobre o serviço.
- *Service Types*, Um modelo técnico (tModel) define tipos de serviços. Múltiplos negócios podem oferecer o mesmo tipo de serviço definido pelo tModel. Um tModel define a informação contida no service, como o nome, o nome da organização, o tipo de service, e apontadores para especificações sobre o tipo de service. Tipicamente, isto é o documento WSDL para o serviço.

2.2.5 Gestão de redes utilizando Serviços Web

Existem algumas iniciativas para a criação de normas de Serviços Web para a gestão de redes, e nesse sentido iremos abordar as duas mais importantes. Uma das especificações é a WS-Management , do grupo DMTF (Distributed Management Task Force) (DMTF, 2008) e a outra a MUWS(Management using Web Services) (OASIS, 2004a), resultado do trabalho do comité técnico Web Services Distributed Management(WSDM) do consorcio OASIS. Ambas propõem uma forma de se fazer a gestão de dispositivos de rede, a partir de operações Web services. (Vianna, 2007)

2.2.5.1 MUWS(Management using Web service)

A MUWS define como é que um recurso de tecnologias de informação conectado na rede pode fornecer interfaces de gestão, de forma que o recurso possa ser gerido localmente ou remotamente, usando tecnologias baseadas em Serviços Web. A especificação MOWS define como os Serviços Web podem ser geridos, utilizando conceitos e definições expressas na especificação MUWS. As funcionalidades de gestão disponibilizadas via MUWS são aquelas geralmente encontradas em sistemas que fazem gestão de recursos distribuídos de tecnologias de informação (Vianna,

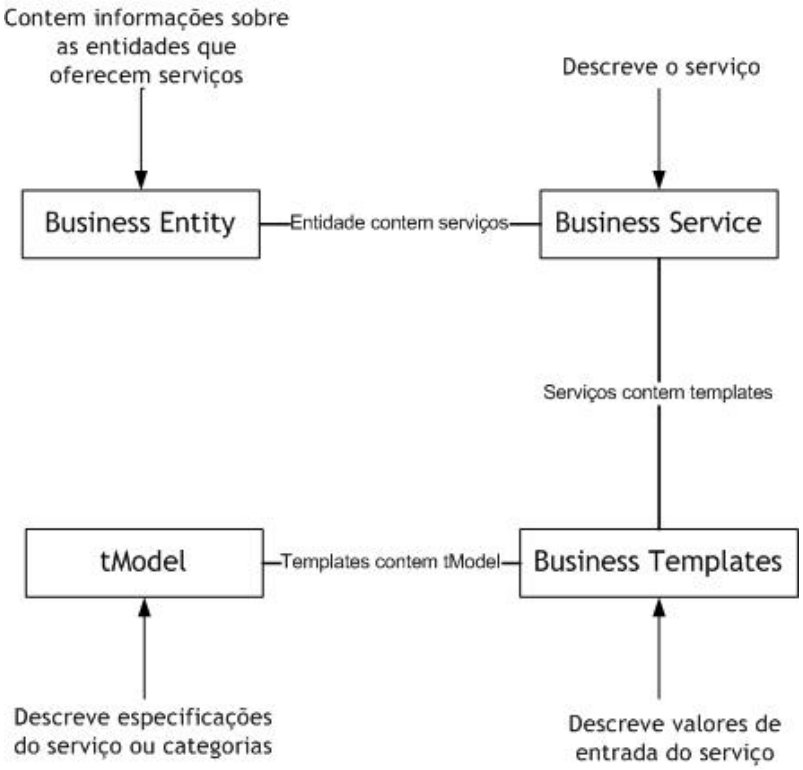

Figura 2.8: Uddi

2007) (OASIS, 2004a). Alguns dos exemplos de gestão que podem ser realizadas pela MUWS são:

- Monitorização da qualidade de serviço (QoS).
- Cumprimento de um SLA (Service Level Agreement).
- Controlo de uma tarefa.
- Gestão do ciclo de vido de um recurso.

2.2.5.2 Web Service for Management (WS-Management)

Define uma interoperabilidade entre aplicações de gestão e dispositivos geridos. Isto só é possível a partir da definição de um conjunto central de especificações de Serviços Web e requisitos de utilização para disponibilizar um conjunto básico de operações fundamentais para a gestão de sistemas (Vianna, 2007) (DMTF, 2008). Algumas dessas operações são:

- Discover - Para a descoberta de dispositivos de gestão.
- Get, Put, Create, Rename e Delete - São operações utilizadas na manipulação de recursos geridos, com valores dinâmicos e configurações.
- Enumerate - Para a recuperação de conteúdos de tabelas ou logs.

- Subscribe - para receber eventos emitidos por recursos geridos.
- Execute - Para a execução de métodos de gestão específicos, incluído parâmetros de entrada e saída.

2.2.5.3 Gateways SNMP/SOAP

A motivação para a utilização de Web services veio do simples facto que tecnologias como o SOAP responderem aos problemas que o protocolo SNMP tem e que vários investigadores procuram resolver à alguns anos. A sua flexibilidade e a disponibilidade de várias ferramentas permitem um rápido desenvolvimento e implantação de serviços Web nos sistemas.

Apesar das limitações do protocolo SNMP, a sua substituição por Serviços Web não parece um cenário possível de acontecer, alias este protocolo deverá manter-se por vários anos devido ao simples facto que substituir este protocolo significaria fazer uma actualização ou mesmo a substituição total de imensos dispositivos de rede que nos dias de hoje funcionam utilizando um agente SNMP para tratar da monitorização e gestão dos dispositivos, essa opção tornara-se ia num processo muito dispendioso e que não seria aceite de muito bom grado pelos operadores de redes que utilizam o serviço como está neste momento. A utilização de Serviços Web a partir de um SNMP gateway é uma opção bem mais viável, visto que se pode manter os agentes SNMP exactamente como estão, apenas se trabalharia com um sistema baseado em Serviços Web do lado do Manager (Lemos Vianna et al., 2006).

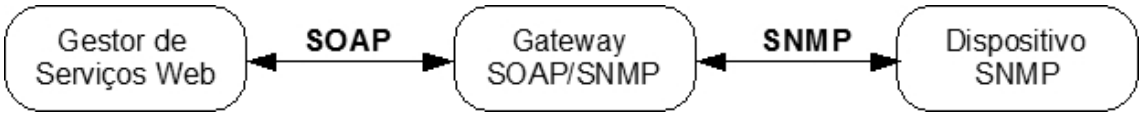

Figura 2.9: Visão geral de uma gateway SNMP/SOAP.

As gateways SNMP podem ser criadas a partir de diversas estratégias, definindo como é que as operações SNMP e a sua informação definidas nas MIB poderiam ser mapeadas em operações de Serviços Web.

- **Gateways ao nível do protocolo** - Operações SNMP são directamente mapeadas em operações dos serviços Web. Por exemplo: uma operação de Set do SNMP tem uma directa correspondência com uma operação de set de Serviços Web.
- **Gateway ao nível do Objecto** - A informação de gestão da MIB SNMP é mapeada em operações dos serviços Web directamente. Por exemplo, o objecto SNMP sysuptime que normalmente existe nos dispositivos, e que responde os tempos que um agente SNMP está a funcionar, poderá ser mapeada numa operação Web service correspondente.
- **Gateway ao nível do serviço** - Os serviços de gestão SNMP são mapeados em operações Web services. Por exemplo a operação de Set da gestão de objectos responsável por controlar a transferência e execução dentro de dispositivos de gestão de *scripts* de gestão, por

exemplo usando o IETF script MIB poderá ser transformada numa simples operação Web services. Gateways a nível do protocolo e do objecto são facilmente criadas pois o seu código é criado automaticamente pelos bem definidos elementos do SNMP. Operações SNMP no caso de ao nível do protocolo, e SNMP MIB no caso de ser ao nível do objecto

2.3 Vantagens e desvantagens

As tecnologias Web Service, que são um conjunto de normas e protocolos, baseados em XML, para construção e integração para aplicações distribuídas na Web, sendo das suas vantagens principais a interoperabilidade e independência de plataforma. As arquitecturas destas tecnologias permitem que sejam criados programas para a sua utilização em varias linguagem de programação, bem como para serem suportadas em plataformas diferentes, possibilitando assim, que vários computadores de uma rede possam comunicar utilizando essas normas - SOAP, WSDL e UDDI.

A utilização de XML para a construção de mensagens dos protocolos usados, permite a descrição de dados de uma maneira estruturada, sem haver uma ligação directa à plataforma ou a linguagem de programação. Esta independência conduz à interoperabilidade entre aplicações. Assim qualquer aplicação é capaz de compreender os dados XML e comunicar sobre um protocolo Web (por exemplo: HTTP). A utilização do protocolo HTTP permite as tecnologias Web Service que funcionem através de firewall, não havendo necessidade de fazer modificações para que tal seja possível.

Esta tecnologia permite reutilização de serviços ou componentes dentro de uma infra-estrutura, podendo ser um ponto forte por parte do Web Service, mas esta reutilização está depende de alguns factores tais como a interoperabilidade, um registo central, entre outros.

Dentro de outras vantagens ou benefícios que o Web Services apresenta temos a transparência na localização, a escalabilidade, a disponibilidade e a reduzida dependência nos fornecedores.

Um dos pontos que poderá diferenciar a utilização de Web Services, com o protocolo SNMP para monitorização, será pelo seu baixo custo e tempo de desenvolvimento reduzido a partir da utilização de normas e ferramentas utilizadas em diversas áreas e não só em gestão de redes, ao contrário do SNMP, que tem um custo mais elevado e tempo de desenvolvimento superior.

A utilização de XML, aliado ao resto das tecnologias Web, irá conseguir dar uma resposta as questões de escalabilidade e eficiência, que eram dois pontos considerados como limitações do protocolo SNMP. Estas informações no formato XML podem ser manipuladas fácil e eficientemente e transferidas através de protocolos largamente utilizados como o HTTP. Informação mais em pormenor em (Mcgovern, 2003)

Capítulo 3

Implementação

Neste capítulo é apresentado o trabalho de implementação desenvolvido para esta tese.

3.1 Introdução

O protótipo desenvolvido vai permitir a um gestor fazer a monitorização de processos via SNMP, em que os objectos geridos em questão serão acedidos a partir de uma comunicação SOAP. Para tal, foi utilizado um software específico da empresa MOG Solutions o Toboggan Media Transfer Direct Write, que serve de software exemplo para estudar esta tipo de monitorização.

Na figura seguinte têm um pequeno diagrama do sistema para uma melhor percepção do problema em questão e de forma a termos noção das componentes com que se irá trabalhar

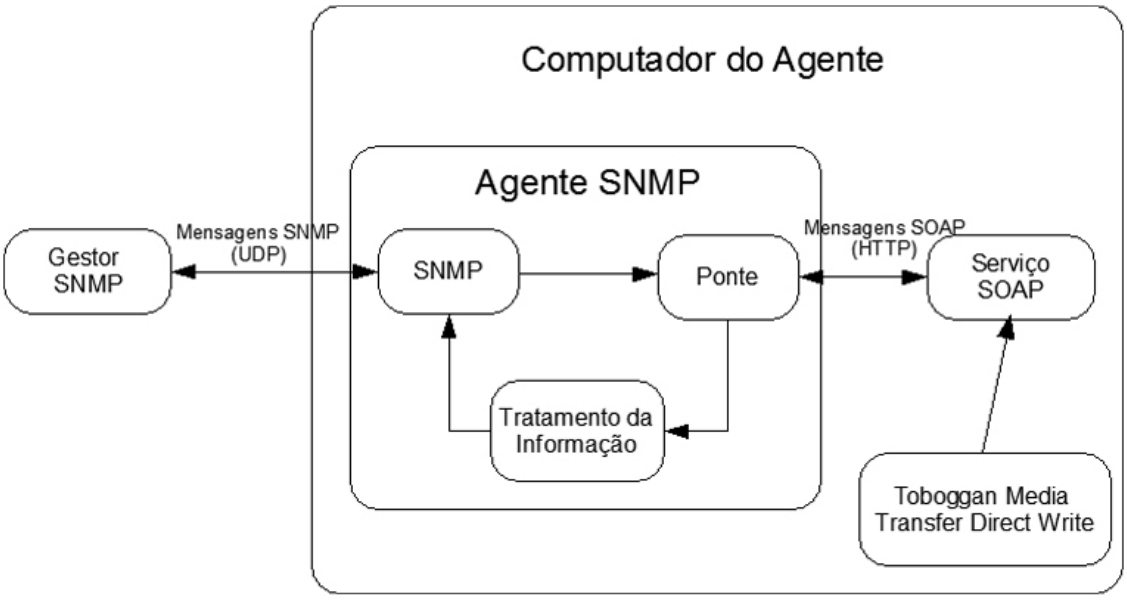

Figura 3.1: Visão geral da implementação.

3.2 Toboggan Media Transfer Direct Write

O Toboggan Media Transfer Direct Write (Solutions, 2009) é um software que simplifica o processo de fazer a transferência de conteúdos profissionais de vídeo baseado em sistemas de pós-produção Avid.

Através da automatização do processo de transferência, o Toboggan Media Transfer elimina os múltiplos passos que são exigidos no fluxo de trabalho de um editor de video, permitindo-lhe passar todos os seus arquivos de MXF para ambientes AVID e partilhar os seus conteúdos maximizando assim a produtividade.

Na figura seguinte temos o painel principal do Toboggan em que podemos ver a várias informações relativamente a cada ficheiro.

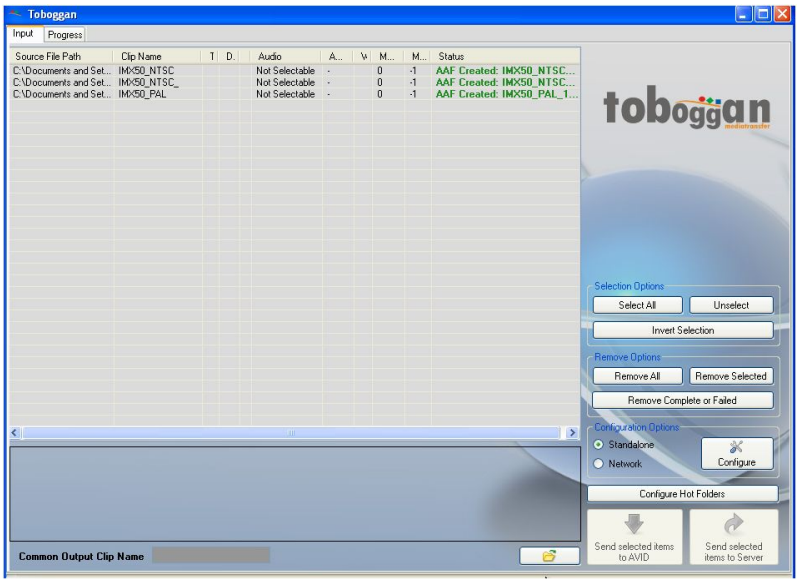

Figura 3.2: Painel principal do Toboggan.

Ao proceder ao envio dos conteúdos para os sistemas da Avid, podemos observar o estado dos processos de transferência no painel demonstrado na figura 3.3.

Foi com a informação deste painel que se iniciou a escolha da informação relevante a ser monitorizada.

Neste painel podemos ver o estado de conclusão, a percentagem de conclusão do processo a que o ficheiro se refere. Se escolhermos algum dos ficheiros temos uma série de informação relativamente ao ficheiro, onde está a informação de entrada, para onde vai a informação de saída, bem como outras informaçõs específicas

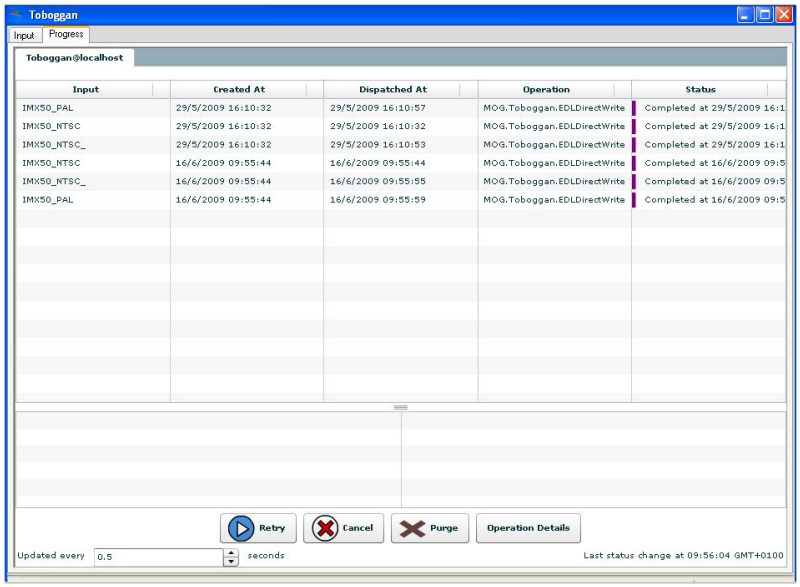

Figura 3.3: Progresso de transferência de ficheiro e informaçãos várias no Tobogan.

3.3 Tecnologias escolhidas

Para o desenvolvimento do projecto foi avaliado um conjunto de frameworks que se podia utilizar como base para a criação deste sistema. Foram encontradas várias soluções para as várias linguagens de programação existentes, e de todas foi escolhida uma solução em python.

A escolha da linguagem em python foi tomada devido aos seguintes factores:

- O facto de esta linguagem ter sido desenvolvida para facilitar a leitura e reutilização de código.
- Normalmente um código desenvolvido em python é mais reduzido que uma mesma implementação criada nas linguagens C, C++ ou Java, por exemplo.
- Para além disso, python tem uma vasta colecção de livrarias que facilitam a sua utlização para vários fins.

O projecto foi desenvolvido no sistema operativo XP, uma vez que é aconselhado a utilização do mesmo para usar o software da empresa MOG Solutions.

3.3.1 Bibliotecas

Para ser possível implementar o sistema SNMP utilizando python instalamos a versão 2.5 do python, a biblioteca de SNMP pysnmp-4.1.10 (Etingof, 2004), pyans1 (Etingof, 2008) e pycrypto (Litzenberger, 2008).

O pysnmp é uma Framework em python que permite a implementação de um sistema SNMP. Actualmente é possível implementar tanto agentes como gestores para todas as versões do protocolo SNMP a partir de métodos simples facilitando assim a criação de um sistema SNMP. Os

únicos requisitos que esta livraria tem é a dependência que existe em relação à instalação tanto do pyasn1 como do pycrypto, assim como a necessidade de utilizar uma ferramenta chamada "smi-dump"da libsmi (Strauss) e um outro pequeno script chamado libsmi2pysnmp para traduzir as MIBs da linguagem ASN.1 para python de forma a que seja possível trabalhar com MIBs que não se encontrem já no repositório do pysnmp.

O pyasn.1 é utilizado para manusear objectos das MIBs, e o pycrypto é usado na encriptação e autenticação de mensagens SNMP.

3.4 Arquitectura do Sistema

Inicialmente, para o desenvolvimento deste projecto, o objectivo será de avaliar as mensagens SNMP GET , GETNEXT e GETBULK. Uma vez que como só procuramos fazer monitorização, não haveria necessidade de procurar estudar mais nenhum tipo de mensagem nesta fase.
Na figura seguinte temos uma visão mais ao pormenor do lado do agente:

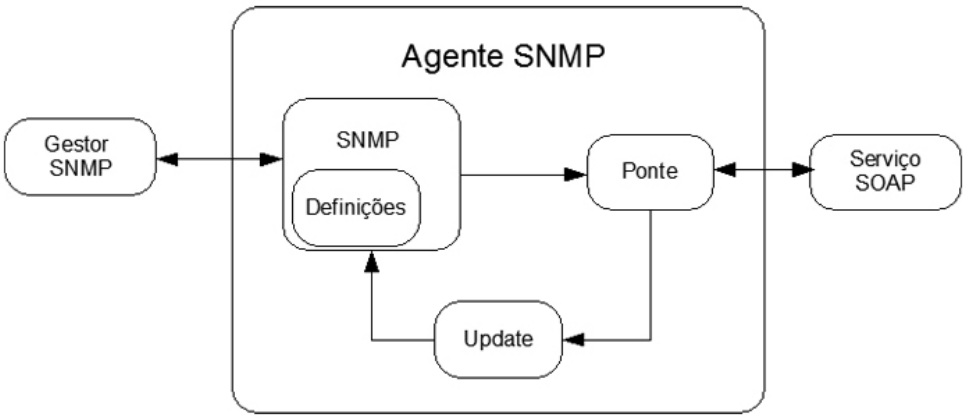

Figura 3.4: Arquitectura do agente SNMP.

O sistema é responsável por monitorizar os processos que estão a ocorrer no software da MOG, acedendo a esses dados a partir de chamadas SOAP.

Ao abrir o Toboggan este irá lançar um serviço SOAP que terá as informações que pretendemos monitorizar. Para aceder as mesmas, bastará fazer as chamadas SOAP seguindo as indicações da WSDL do serviço.

O módulo bridge, está encarregue de fazer as chamadas ao serviço SOAP e de seguida proceder ao primeiro tratamento dos dados provenientes dessa chamada. Esse tratamento resume-se na criação de funções específicas para cada tipo de dados que pretendemos obter, como podemos ver mais em pormenor em na secção 3.3.2 .

O bloco SNMP está responsável por tratar das definições dos objectos a serem monitorizados, bem como o tratamento relativo aos pedidos vindos de um gestor SNMP.

3.4.1 Definição da MIB

O primeiro passo a tomar para a criação do agente SNMP, foi fazer uma avaliação dos dados possíveis de serem retornados por parte da interface SOAP, e dessa forma acabamos por ficar com cinco parâmetros que serão mais importantes de monitorizar.

- **Status** - São os vários estados em que o processo pode estar. Existem cinco estados possíveis - Completed, Idle, Cancelled, Processing e Failed. Cada um destes estado corresponde a um valor inteiro de 0 a 5 que a interface SOAP utiliza para definir os estados, e que o Agente terá de fazer correspondência com a palavra correcta, de forma a responder correctamente a pedidos.

- **Percentage** - É um valor inteiro de 0 a 100 que corresponde a percentagem de conclusão do processo.

- **Id** - É o identificador do processo.

- **Config** - Numa fase inicial, este parâmetro será um XML com várias informações sobre o processo. Uma vez que são informações estáticas e com uma importância menor nesta fase do projecto. No futuro será possível criar objectos para monitorizar ou mesmo controlar estes parâmetros.

- **Index**

Havendo a possibilidade de existir vários processos, e uma vez que o número de processos vai mudando, a utilização de uma tabela para a organização da informação. Foi definida uma tabela na MIB, em que as colunas dessa tabela são os parâmetros acima referidos, mais uma coluna para o valor do index. Por uma questão de organização, foi renomeado cada parâmetro com o prefixo "tbg".

A tabela abaixo ilustra como estará a tabela da MIB definida:

tbgStatusIndex	tbgStatus	tbgId	tbgPercentage	tbgConfig
1	tbgStatus.1	tbgId.1	tbgPercentage.1	tbgConfig.1
2	tbgStatus.2	tbgId.2	tbgPercentage.2	tbgConfig.2
3	tbgStatus.3	tbgId.3	tbgPercentage.3	tbgConfig.3

Tabela 3.1: Exemplo da estrutura dos dados da MIB.

Acima vemos o exemplo de como é que a informação será inserida. Por exemplo, o OID de tbgStatus é 1.3.6.1.3.26.1.1.1.2, para associar o valor de Status do primeiro processo teremos de o associar ao OID de tbgstatus mais o índice respectivo, ou seja, 1.3.6.1.3.26.1.1.1.2**.1**.

3.4.2 Comunicação com interface SOAP

Uma vez que temos de comunicar com uma interface SOAP para aceder aos dados que pretendemos monitorizar, eis o exemplo de um pedido à interface SOAP em causa para pedir valores dos estados dos processos.

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001
/XMLSchema-instance>
    <SOAP-ENV:Header>
    <ns0:Toxmlns:ns0="http://schemas.xmlsoap.org/ws/2004/08/addressing">
    http://localhost:13405/Monitoring/</ns0:To>
    <ns0:Actionxmlns:ns0="http://schemas.xmlsoap.org/ws/2004/08/addressing">
    Monitoring/GetStatus</ns0:Action>
   </SOAP-ENV:Header>
      <SOAP-ENV:Body>
     <tns:GetStatus xmlns:tns="Monitoring"/>
     </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Deste pedido iremos receber a resposta a um pedido de GetStatus:

```
<soap:Envelope xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Header>
      <wsa:Action>Monitoring/GetStatusResponse</wsa:Action>
      <wsa:MessageID>urn:uuid:58f0f8a9-ecf8-4295-af30-0f9ef31fcd45</wsa:MessageID>
      <wsa:RelatesTo>urn:uuid:6ac500dd-01d1-4531-97eb-ff6d2a12ad9c</wsa:RelatesTo>
      <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
      </wsa:To>
      <wsse:Security>
         <wsu:Timestamp wsu:Id="Timestamp-7c279bd9-8aaa-4d23-9aae-5641b92de00f">
            <wsu:Created>2009-03-18T11:33:24Z</wsu:Created>
            <wsu:Expires>2009-03-18T11:38:24Z</wsu:Expires>
         </wsu:Timestamp>
      </wsse:Security>
   </soap:Header>
   <soap:Body>
```

```
      <GetStatusResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="Monitoring">
        <GetStatusResult><![CDATA[
        <StatusList><StatusItem><Status>2</Status>
        <Percentage>0.0</Percentage>
<ID>12373707071c927543149e9e483a42b9891147c837</ID></StatusItem>
<StatusItem><Status>2</Status>
<Percentage>0.0</Percentage>
<ID>12373758859aa85d206755549663c49db92a325a8a</ID></StatusItem>
<StatusItem><Status>2</Status>
<Percentage>0.0</Percentage>
<ID>12373758864f515b2aa0656ad98e68cebf4f675764</ID></StatusItem>
<StatusItem><Status>2</Status>
<Percentage>0.0</Percentage>
<ID>12373758869ac109bc09072716c492a19899d64cab</ID>
</StatusItem></StatusList>]]></GetStatusResult>
      </GetStatusResponse>
   </soap:Body>
</soap:Envelope>
```

Na corpo da mensagem SOAP da resposta a esse pedido de GetStatus, podemos encontrar a informação que se pretende, com o identificador <GetStatusResult>. As informações dos processos existentes no Toboggan estão organizadas numa lista de itens (<StatusList>). E cada item (<StatusItem>) contém três valores:

- O estado - <Status>2</Status>
- A percentagem - <Percentage>0.0</Percentage>
- e o identificador - <ID>12373707071c927543149e9e483a42b9891147c837</ID>

Depois do pedido executado, e de ser ter recebido a resposta, será necessário fazer um parser ao XML recebido de forma a ser possível separar a informação. E nesse sentido foram criados métodos para facilitar e organizar a informação que os métodos do SNMP irão aceder. Os métodos a seguir definidos fazem uma divisão da informação vinda do XML de resposta recebido:

- getStatusXML() - Função que esta encarregue de fazer a chamada ao serviço SOAP obtendo uma resposta ao pedido de GetStatus. Da resposta iremos obter os ID dos processos que estão a ser processados no Toboggan, e também temos a informação sobre o estado, a percentagem de conclusão do ficheiro.
- GetOptDes(ID) - Função que esta encarregue de fazer a chamada ao serviço SOAP obtendo uma resposta ao pedido do GetOperationDescription. O parâmetro de config é exactamente este XML.

- GetStatusID() - É uma função secundária que trata a resposta do GetStatus respondendo uma lista com os vários ID existentes.
- GetNumOpt() - É uma simples função encarregue de retornar o número de processos a decorrer.
- GetStatus() - É uma função secundária que trata a resposta do GetStatus respondendo uma lista com os vários status existentes.
- GetPercentage() - É uma função secundária que trata a resposta do GetStatus respondendo uma lista com os várias percentagens de conclusão dos ficheiros existentes.

3.4.3 Agente SNMP

Para a parte do SNMP e utilizando o pysnmp, foi desenvolvido o agente SNMP que permitisse suportar responder a pedidos SNMP das três versões existentes.

É apresentado de seguida, um diagrama que demonstra como é que o sistema SNMP está funcionar do lado do agente.

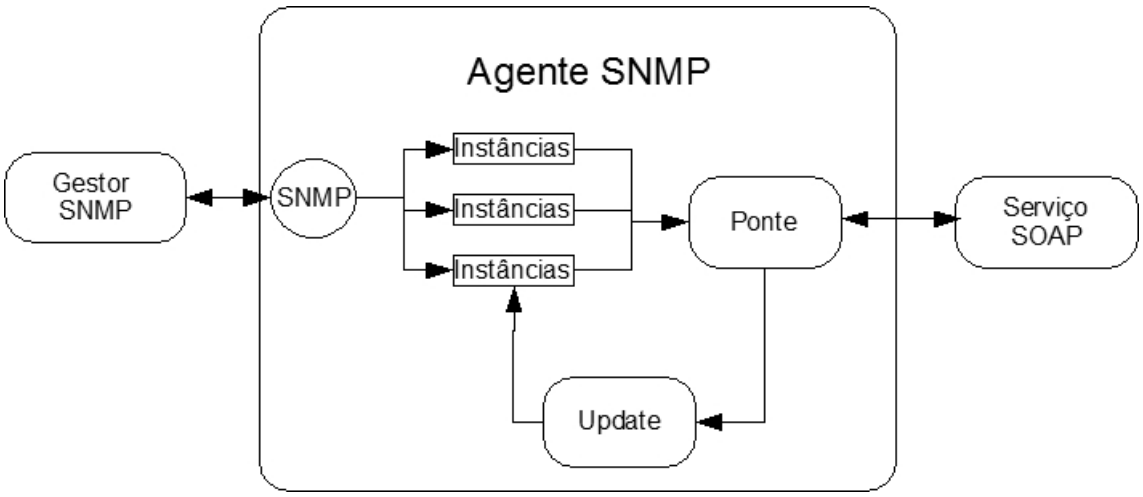

Figura 3.5: Agente SNMP

3.4.3.1 Definições

No módulo SNMP são feitas as várias definições que irão possibilitar a este agente poder funcionar. Das definições gerais de funcionamento, destacamos as seguintes:

- Definir a ligação (Por norma utiliza-se a porta 161 e utiliza-se o protocolo UDP).
- É necessário fazer um *load* da MIB onde se encontram os objectos que vamos utilizar (Neste caso fazer load da MIB criada para este trabalho a MOG-MIB), para que seja possível modificar os objectos em causa.

- Definir o modelo de segurança, definindo para o SNMPv3 a chave de autenticação, a privacidade e o nome de um utilizador. E para as outras duas versões definir a chave da comunidade.

Com esta biblioteca, para permitir que se possa monitorizar algum objecto, é necessário criar instâncias relativas a esse objecto.

O trabalho desenvolvido nesta parte do projecto está concentrado na definição de instâncias que serão monitorizadas, existindo um motor que irá criar as várias instâncias SNMP para cada objecto. Este motor irá atribuir para cada uma das instâncias o seu respectivo OID (sem esquecer o valor de índex no fim) e o valor nesse nó. O valor nesse nó será obtido a partir de métodos que irão chamar os métodos acima referidos que fazem a ligação ao SOAP, de seguida só terá de tratar esses dados de forma a só retornar o valor desejado.

Figura 3.6: Exemplo de definição de uma instância

Na figura 3.6 temos um exemplo de como fazemos essa definição. No valor dessa instância temos um método chamada Status(i) que irá retornar o valor correspondente ao processo pedido, neste caso temos Status(1), logo irá retornar o estado do primeiro processo.

Para cada objecto temos um método responsável por comunicar com os métodos que fazem comunicação com a interface SOAP e retornar o dados.

Sendo assim, a única preocupação que resta é a de garantir que seja possível fazer a actualização dos dados.

3.4.3.2 Actualização dos valores

Para cada objecto a ser monitorizado, existe uma função de actualização responsável por actualizar os dados relativos a sua coluna da tabela. O modo de funcionamento é simples, todas as instâncias são apagadas e de seguida é criada novamente a tabela com as novas instâncias e os seus respectivos novos valores.

O sistema de actualização funciona a cada pedido SNMP feito. Uma outra opção seria de ter um sistema de actualização periódico, e a razão pela qual não se seguiu por essa opção foi que após alguns testes iniciais se tenha chegado a alguns erros pontuais, onde se tentava fazer uma actualização ao mesmo tempo que existia um pedido SNMP. O erro surge aquando de um pedido a um nó possivelmente inexistente pela actualização naquele momento.

Depois de alguns testes de performance podemos concluir que apesar de não ser a melhor opção, o seu consumo de memória não é muito elevado e não irá influenciar muito o funcionamento do sistema.

3.4.3.3 Mensagem Trap

Uma outra operação também implementada foi a do Trap SNMP. Esta operação é uma mensagem enviada pelo agente SNMP ao gestor relatando a ocorrência de um erro. Neste caso, essa mensagem será enviada quando algum processo de transferência do toboggan falhar. O agente avalia o estado dos estados dos processos e quando encontrar um processo com o valor do Status igual a dois (que equivale ao estado de Failed) faz esse envio.

De forma a garantir que não se esteja constantemente a relatar o mesmo erro, o agente SNMP mantém uma lista com os identificadores (ID) dos processos garantido que não se esteja a sobrecarregar o gestor com as mesmas informações.

3.4.4 Problemas encontrados

Um dos problemas encontrados durante o desenvolvimento do projecto, foi o facto de a biblioteca escolhida, o pysnmp, ter um bug na função relativa à destruição de instâncias. A função unexportSymbol está encarregue dessa tarefa, e no inicio deste projecto não havia referencia de que esta função teria problemas. Depois de ter contactado a pessoa responsável pela livraria o Ilya Etingof, e ter relatado estes problemas, esse erro foi resolvido ao fim de uma semana. Sem que tenha havido grande prejuízo para o desenvolvimento do projecto.

Capítulo 4

Teste Funcionais

Neste capítulo é apresentado os testes feitos ao Agente SNMP desenvolvido, procurando avaliar o seu funcionamento. Estes testes foram feitos utilizando diversos clientes SNMP, no sentido de validar o trabalho desenvolvido.

4.1 Definição

Para se poder validar o agente SNMP criado, foi importante fazer uma avaliação de performance, e mesmo poder tirar alguma conclusão foi necessário escolher alguns clientes SNMP (Manager). Para esse efeito, eis os pontos que se teve em consideração:

- Ser possível monitorizar agentes SNMP, utilizando qualquer uma das versões.
- Ter um software que possua uma interface SOAP que possibilite o acesso a parâmetros relativos aos seus processos (Para o desenvolvimento desta tese utilizamos o software da MOG - o Toboggan Media Transfer Direct Write).

4.2 Clientes SNMP

De todos os softwares de teste utilizados destacamos a utilização dos seguintes clientes SNMP:

4.2.1 O MG-Soft Professional Edition

Este software tem uma ferramenta o MIB Browser Professional Edition, que permite fazer pedidos SNMP, mas a grande vantagem para este caso em concreto, é a possibilidade de se poder ver a tabela toda de uma só vez, monitorizando todos os parâmetros ao mesmo tempo (Corporation, 2009).

A única desvantagem é o facto de se ter utilizado uma versão de teste do software,e portanto

não se poder testar a terceira versão do protocolo SNMP, tendo sido utilizado a versão do protocolo SNMP.

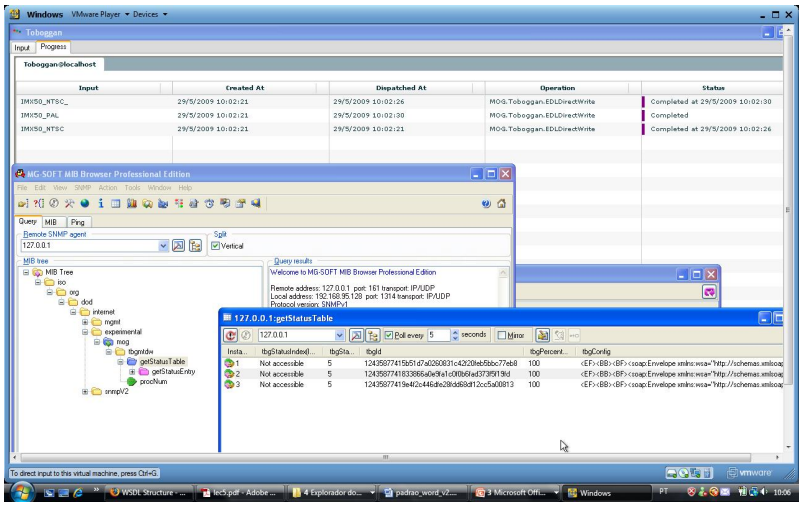

Figura 4.1: MG-Soft - MIB Browser Professional Edition

4.2.2 ManageEngine OpManager

É uma ferramenta para monitorização de redes e que permite uma visão completa do que ocorre na rede com base no protocolo SNMP (MOpManager, 2009).

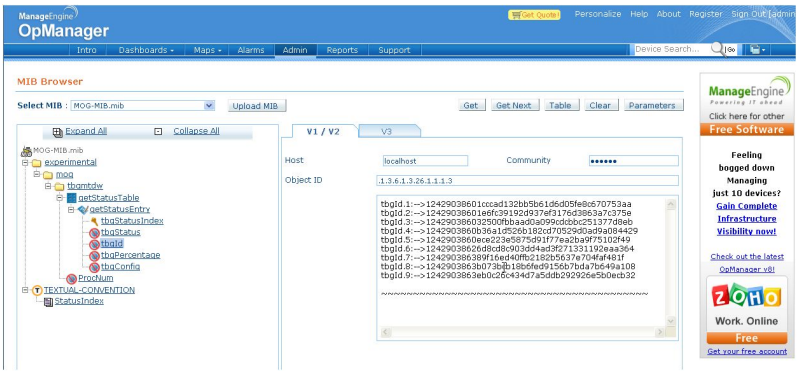

Figura 4.2: ManageEngine OpManager

- Permite monitorizar o CPU, Memória, HD, Processos, Serviços. Suporte a MIBs (Management Information Base).
- Criação de relatórios de utilização e disponibilidade de recursos.
- Software web-based multiplataforma.
- Suporte a SNMP, WMI, CLO, Telnet, SSH entre outros.

O ManageEngine OpManager é um software completo para monitorização de redes. Este oferece a integração entre help-desk, WAN, servidores, aplicações e análise de tráfego WAN. O OpManager automatiza várias tarefas de monitoramento e remove a complexidade associada à gestão da rede.

Neste caso utilizamos uma ferramenta que este software possui para monitorizar objectos de uma MIB. O único problema é o facto obrigar que se facam os pedidos, não existindo a possibilidade de se poder definir a monitorização de certos elementos, com a definição de um periodo de actualização fixo.

4.2.3 Análise de Resultados

Num ambiente de testes, o objectivo primordial foi analisar até que ponto pode ou não ser vantajoso monitorizar estes processos multimédia utilizando para esse efeito o protocolo SNMP. Utilizando os clientes acima referidos, foi possível proceder a vários testes de comunicação via SNMP, tendo encontrado erros numa fase inicial, que levou a um recuo na ordem de trabalhos.

Acabou por se perceber que o erro provinha de um problema com a livraria de SNMP, que tinha alguns bugs relativos a uma pequena acção que era necessário fazer (apagar instâncias SNMP).

Nos testes feitos com a primeira versão do protocolo, demonstraram as grandes limitações que esta versão apresenta, e uma vez que se está a usar tabelas, usar esta versão não tem qualquer benefício, a não ser que estejamos a monitorizar apenas dois ou três processos, em que quase não se nota a velocidade.

De uma maneira geral, os testes demonstraram que a opção de criar uma ponte entre um agente SNMP e uma interface SOAP para a monitorização de processos é possível e torna-se muito vantajosa se pensarmos na quantidade de dispositivos que poderão ser monitorizados desta forma.

Capítulo 5

Conclusões e Trabalho Futuro

5.1 Satisfação dos Objectivos

O trabalho desenvolvido culminou com a criação de agente SNMP capaz de monitorizar processos multimédia a partir de uma ponte com a interface SOAP. A primeira parte do trabalho concentrou-se no estudo inicial sobre o funcionamento do protocolo SNMP, tentando perceber as suas limitações e potencialidades. Na segunda parte do estudo, procurou-se estudar que opções estavam a ser estudadas para combater o SNMP, tendo o Web Services sido a tecnologia que foi proposta tanto pela comunidade científica, como pela indústria, para eventualmente no futuro poder vir a ser uma opção para a gestão e monitorização de redes. Desse estudo, como vimos anteriormente foram propostas duas padronizações para estas tarefas de gestão (o WSDM e o WS-Management), bem como uma outra perspectiva que junta a utilização de SNMP e tecnologias Web Services criando uma comunicação a partir de uma gateway SNMP/SOAP entre o agente e o gestor. De seguida foi feito o levantamento do problema proposto, procurando especificar o melhor possível uma abordagem capaz de conseguir resolver o problema em questão, uma vez que para o estudo deste problema foi utilizado um software da empresa MOG Solutions que funcionava com uma interface SOAP, houve a necessidade de haver uma familiarização com o software, tentando perceber a sua utilidade, de forma a ser mais fácil conseguir definir que parâmetros poderiam ser importantes monitorizar.

Após a definição dos parâmetros, definiu-se uma MIB para o software em questão com os respectivos parâmetros definidos como objectos dessa MIB.

Por último foi implementado um agente SNMP capaz de permitir a monitorização de processos multimédia a partir de uma comunicação via SOAP, tendo havido a necessidade de criar um módulo dentro do agente SNMP que possibilita-se essa mesma comunicação.

Depois de finalizado o trabalho e o estudo, concluímos que o protocolo SNMP é amplamente utilizado para a gestão e monitorização de redes de computadores, e que apresenta as suas limitações relativamente a problemas de escalabilidade e eficiência. A simplicidade e facilidade foram

as suas maiores aliadas para o sucesso que este protocolo teve, mas estas duas desvantagens fizeram com que fosse equacionado o estudo de alternativas ao protocolo SNMP para a gestão e monitorização de redes. O grande crescimento das tecnologias Web Services, e o facto de estas tecnologias baseadas na linguagem XML, poderem responder aos problemas que o protocolo SNMP apresenta, fizeram com que se começasse a estudar a sua utilização para funções de gestão e monitorização. Mas como o Vianna faz referência neste artigo (Lemos Vianna et al., 2006), em que diz que apesar das limitações do SNMP não é provável que as tecnologias Web Services venham a substituir completamente o SNMP. O dispositivos que funcionam com o protocolo SNMP irão se manter no mercado por vários anos, pois substituir o SNMP iria implicar a actualização de muitos dispositivos, ou mesmo a substituição completa, o que tornaria o processo muito dispendioso e que provavelmente não seria aceite pelos operadores de redes com muito grado. podemos considerar uma continuidade do protocolo SNMP, por alguns anos.

5.2 Trabalho Futuro

Para trabalho futuro nesta área, seria proposta a implementação de funções SNMP de 'Set' no agente, uma vez que na realização deste trabalho não se procurou estudar essa possibilidade, mas para este caso em específico seria necessário estar a trabalhar com uma interface SOAP que possibilita a alteração dos parâmetros.

É proposto que se faça um estudo mais aprofundade relativamente ao consumo do CPU e da memória, (por exemplo para casos em que se esteja a monitorizar muitos processos ao mesmo tempo) para se ter uma perspectiva melhor relativamente à performance que este tipo de agentes SNMP podem ter.

Referências

MG-Soft Corporation. Mg-soft corporation, Junho 2009. http://www.mg-soft.si/, último acesso dia 5 de junho.

W.Q. de Lima, R.S. Alves, R.L. Vianna, M.J.B. Almeida, L.M.R. Tarouco e L.Z. Granville. Evaluating the performance of snmp and web services notifications. pages 546–556, April 2006. doi: 10.1109/NOMS.2006.1687583.

DMTF. *Web Services for Management (WS-Management) Specification*, Fevereiro 2008. http://www.dmtf.org/standards/published_documents/DSP0226_1.0.0.pdf.

Ilya Etingof. Asn.1 library for python, Maio 2008. http://sourceforge.net/projects/pyasn1/.

Ilya Etingof. Python snmp framework, Dezembro 2004. http://pysnmp.sourceforge.net/.

D. Perkins K. McCloghrie e J. Schoenwaelder. Structure of management information version 2 (smiv2), Maio 1999. RFC2578 disponível em (http://tools.ietf.org/html/rfc2578).

R.L. Lemos Vianna, M.J.B. Almeida, L.M.R. Tarouco e L.Z. Granville. Investigating web services composition applied to network management. pages 531–540, Sept. 2006. doi: 10.1109/ICWS.2006.81.

Dwayne C. Litzenberger. Python cryptography toolkit, Maio 2008. http://www.amk.ca/python/code/crypto.

Douglas R. Mauro e Kevin J. Schmidt. *Essential SNMP, Second Edition*. O'Reilly Media, Inc., 2005. ISBN 0596008406.

K. McCloghrie e M. Rose. Structure and identification of management information for tcp/ip-based internets, Maio 1990. (RFC1155) disponível em (http://tools.ietf.org/html/rfc1155).

James Mcgovern. *Java Web Services Architecture*. Morgan Kaufmann, San Diego, 2003. ISBN 1558609008.

MOpManager. Opmanager ,network monitoring software., Junho 2009. http://www.manageengine.com/products/opmanager/download.html, último acesso dia 16 de junho.

OASIS. *Web Services Distributed Management: Management Using Web Services (MUWS)*, Dezembro 2004a. http://docs.oasis-open.org/wsdm/2004/12/muws/cd-wsdm-muws-part1-1.0.pdf.

OASIS. *The Universal Discovery Description and Integration Technical Committee Draft V 3.02 (UDDI)*, Outubro 2004b. Disponívek em http://uddi.org/pubs/uddi-v3.0.2-20041019.htm, acedido a última vez em 18 de Junho de 2009.

G. Pavlou, P. Flegkas, S. Gouveris e A. Liotta. On management technologies and the potential of web services. *Communications Magazine, IEEE*, 42(7):58–66, July 2004. ISSN 0163-6804.

Aiko Pras, Thomas Drevers, Remco van de Meent e Dick Quartel. Comparing the performance of snmp and web services-based management. *Network and Service Management, IEEE Transactions on*, 1(2):72–82, Dec. 2004. ISSN 1932-4537. doi: 10.1109/TNSM.2004.4798292.

John Soldatos e Dimitris Alexopoulos. Web services-based network management: approaches and the wsnet system. *Int. J. Netw. Manag.*, 17(1):33–50, 2007. ISSN 1099-1190. doi: http://dx.doi.org/10.1002/nem.612.

Mog Solutions. Mog solutions, Junho 2009. http://www.mog-solutions.com/.

Frank Strauss. libsmi - a library to access smi mib information. Disponivel em http://www.ibr.cs.tu-bs.de/projects/libsmi/, acedido em Junho 2009.

Ricardo Lemos Vianna. Uma solução para composição de serviços de gerenciamento de redes utilizando padrões web service. Master's thesis, Universidade Federal do Rio Grande do Sul, Maio 2007.

W3C. *Simple Object Access Protocol (SOAP) 1.*, Maio 2000. Disponível em http://www.w3.org/TR/2000/NOTE-SOAP-20000508/, acedido a última vez em 23 de Junho de 2009.

W3C. *Web Services Description Language (WSDL) Version 2.0 Part 0: Primer*, Junho 2007a. Disponível em http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626/, acedido a última vez em 18 de Junho de 2009.

W3C. *The Simple Object Access Protocol 1.2*, Abril 2007b. Disponível em http://www.w3.org/TR/soap12-part1/, acedido a última vez em 18 de Junho de 2009.

Jeong-Hyuk Yoon, Hong-Taek Ju e James W. Hong. Development of snmp-xml translator and gateway for xml-based integrated network management. *Int. J. Netw. Manag.*, 13(4):259–276, 2003. ISSN 1099-1190. doi: http://dx.doi.org/10.1002/nem.478.