



**Rui Manuel Duarte  
Marques**

**Avaliação de Desempenho de Protocolos de Gestão**



**Rui Manuel Duarte  
Marques**

**Avaliação de Desempenho de Protocolos de Gestão**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações realizada sob a orientação científica do Professor Doutor José Luís Guimarães Oliveira, Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e o Mestre Pedro Alexandre Sousa Gonçalves, equiparado a Assistente 2º triénio da Escola Superior de Tecnologia e Gestão de Águeda.

Dedico este trabalho à minha esposa e aos meus filhos pela paciência e apoio demonstrados, em especial, durante as longas horas que deixei de estar com eles.

## **o júri**

presidente

**Prof. Dr. Aníbal Manuel de Oliveira Duarte**  
professor catedrático do Departamento de Electrónica, Telecomunicações e Informática da  
Universidade de Aveiro

arguente

**Prof. Dr. Rui Pedro Sanches de Castro Lopes**  
prof. coordenador do Departamento de Informática e Comunicações da Escola Superior de  
Tecnologia e Gestão do Instituto Politécnico de Bragança

orientadores

**Prof. Dr. José Luís Guimarães Oliveira**  
professor associado do Departamento de Electrónica, Telecomunicações e Informática da  
Universidade de Aveiro

## **agradecimentos**

Os meus agradecimentos vão para todos aqueles que me auxiliaram no meu trabalho profissional ao longo destes meses, permitindo que pudesse dedicar tempo ao desenvolvimento desta dissertação.

Gostaria de realçar também a forma positiva e paciente com que o Professor Doutor José Luís Guimarães Oliveira me incentivou durante este processo.

Por fim, queria manifestar um agradecimento especial, ao Mestre Pedro Alexandre Sousa Gonçalves, pelo tempo que gastou comigo e apoio prestado, sempre disponível para esclarecer dúvidas e auxiliar nas tarefas mais complicadas.

**palavras-chave**

Gestão de Redes, Protocolos de Gestão, SNMP, WBEM, WS-Management,

**resumo**

Uma rede de computadores actual pode ser formada por milhares de equipamentos que apresentam características distintas e estão distanciados, quer entre si, quer em relação à entidade responsável por gerir a rede. Ao longo dos anos têm surgido soluções de gestão distintas, visando responder às necessidades impostas por este processo evolutivo e também pelo constante aumento dos requisitos computacionais. Perante estes factos é necessário fazer um balanço entre as vantagens e desvantagens desta evolução e assim poder escolher a solução que melhor responda aos desafios.

**keywords**

Network Management, Management Protocols, SNMP, WBEM, WS-Management

**abstract**

A computer network can now be formed by thousands of equipments with different characteristics usually scattered in a vast geographic area. Over the years they have been proposed different management solutions, aimed to meet the needs imposed by the evolutionary process and also by the constant increase in computational requirements. In this context it is necessary to make a balance between the advantages and disadvantages of these developments and thus to be able to choose the solution that best meets the challenges.

# Índice

<b>CAP. 1 – INTRODUÇÃO</b> .....	<b>3</b>
1.1 ENQUADRAMENTO .....	3
1.2 OBJECTIVOS .....	3
1.3 ESTRUTURA.....	4
<b>CAP. 2 - MODELOS DE GESTÃO DE SISTEMAS</b> .....	<b>5</b>
2.1 INTRODUÇÃO .....	5
2.2 SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP) .....	6
2.2.1 <i>Protocolo SNMP – comunicação entre Gestor e Agente</i> .....	8
2.2.2 <i>MIB, SMI e ASN.1</i> .....	10
2.2.3 <i>Monitorização remota e segurança</i> .....	13
2.2.4 <i>Limitações do protocolo SNMP</i> .....	14
2.3 WEB BASED ENTERPRISE MANAGEMENT (WBEM).....	15
2.3.1 <i>Common Information Model (CIM)</i> .....	17
2.3.2 <i>Codificação dos dados de gestão em CIM-XML</i> .....	19
2.3.3 <i>CIM-XML sobre HTTP</i> .....	20
2.3.4 <i>Arquitetura WBEM</i> .....	21
2.3.5 <i>Interface Cliente-Servidor (Gestor-Agente)</i> .....	23
2.3.6 <i>Implementações WBEM</i> .....	26
2.4 GESTÃO BASEADA EM WEB SERVICES (WS) .....	28
2.4.1 <i>Arquitetura WS</i> .....	29
2.4.2 <i>Simple Object Access Protocol (SOAP)</i> .....	30
2.4.3 <i>Web Services Description Language (WSDL)</i> .....	34
2.4.4 <i>Universal Description, Discovery and Integration (UDDI)</i> .....	36
2.4.5 <i>Normas de gestão baseada em WS</i> .....	38
2.5 ANÁLISE COMPARATIVA .....	44
2.6 SUMÁRIO.....	47
<b>CAP. 3 - EXPERIÊNCIAS REALIZADAS</b> .....	<b>49</b>
3.2 CENÁRIO DE TESTES .....	50
3.3 PROCEDIMENTOS E EXPERIÊNCIAS REALIZADAS.....	51
3.3.1 <i>Experiências realizadas com SNMP</i> .....	52
3.3.2 <i>Experiências realizadas com WBEM</i> .....	52
3.3.3 <i>Experiências realizadas com WS-Management</i> .....	54
3.4 RESULTADOS OBTIDOS .....	55
3.5 ESTUDO COMPARATIVO.....	56
3.6 SUMÁRIO.....	61
<b>CAP. 4 – CONCLUSÕES</b> .....	<b>63</b>
<b>ANEXOS</b> .....	<b>65</b>
ANEXO 1 – LISTA DE RFCs RELACIONADOS COM SNMP .....	65
ANEXO 2 – TROCA DE MENSAGENS WBEM.....	66
ANEXO 3 – TROCA DE MENSAGENS WS-MANAGEMENT .....	67
ANEXO 4 – PREFIXOS E OS NAMESPACES XML USADOS EM WS-MANAGEMENT .....	68
ANEXO 5 – URIS <i>wsa:ACTION</i> USADOS EM WS-MANAGEMENT.....	69
<b>REFERÊNCIAS</b> .....	<b>71</b>



# Cap. 1 – Introdução

---

## 1.1 Enquadramento

A rede de um operador de telecomunicações possui milhares de equipamentos de rede, bem como centenas de servidores que necessitam de ser configurados para que tudo funcione correctamente. Para além do imenso trabalho de configuração existem outros desafios que tornam a configuração de equipamentos um trabalho árduo e muito sujeito a erros: a imensa quantidade de equipamentos a configurar obriga a um trabalho repetitivo pouco adequado a um operador humano. Por outro lado, alguns dos equipamentos com diferentes funções na rede devem ser configurados para que cooperem entre si e assim permitam que o todo funcione de uma forma adequada.

A resposta a estas dificuldades é normalmente dada por um sistema de gestão de rede (Network Management System - NMS). O NMS fornece ou efectua a configuração de todo o equipamento de uma forma correcta sem os erros tipicamente associados à intervenção humana.

Algumas aproximações de gestão mais recentes propõem modelos de gestão baseados em políticas; existem um conjunto de regras definidas pelo operador humano junto do NMS que determinam o comportamento de todo o equipamento. Com o decorrer dos tempos a gestão de redes e sistemas tem sido feita com base em diferentes paradigmas e recorrendo a tecnologias que têm evoluído desde a gestão utilizando o protocolo SNMP até à gestão com base em Web Services. A flexibilidade e facilidade de implementação das soluções de gestão têm aumentado, mas os requisitos computacionais também.

Perante estes factos é necessário fazer um balanço entre as vantagens e desvantagens desta evolução e assim poder escolher a solução que melhor responda aos desafios.

## 1.2 Objectivos

O objectivo deste projecto é fazer um estudo comparativo das diferentes arquitecturas de gestão (SNMP, WBEM e Web Services) existentes. Para tal é necessário escolher uma solução de gestão de cada arquitectura, instalá-la e efectuar um conjunto de testes.

De acordo com os requisitos do projecto pretende-se realizar:

- Definição do conjunto de testes a efectuar às diferentes soluções de gestão;
- Instalação de uma solução de gestão baseada em cada arquitectura;
- Comparar o desempenho de cada uma das soluções em relação aos testes realizados.

## **1.3 Estrutura**

A estrutura desta dissertação baseia-se nos capítulos 2 e 3 onde são desenvolvidos os temas principais da mesma e descrito o trabalho experimental desenvolvido.

O capítulo 2 descreve com algum pormenor os modelos de gestão de redes, SNMP, WBEM e gestão baseada em Web Services, as principais tecnologias e princípios envolvidos nos mesmos e enumera algumas implementações correntes bem como suas principais características. É ainda feita uma análise comparativa entre os modelos que são objecto de estudo neste trabalho.

O capítulo 3 descreve os cenários de testes criados para os três modelos de gestão, as aplicações usadas nos mesmos, a natureza dos dados a obter e os objectivos a atingir com os testes realizados. São depois apresentados os resultados obtidos e um estudo comparativo entre os modelos testados.

No capítulo 4, final e conclusivo, são descritas e sustentadas as principais conclusões resultantes das experiências realizadas, explicados eventuais desvios relativamente a resultados esperados e apontadas algumas orientações respeitantes a trabalhos futuros a desenvolver sobre esta temática.

# Cap. 2 - Modelos de Gestão de Sistemas

---

## 2.1 Introdução

As redes de comunicação actuais são muito diferentes daquelas que deram origem à ARPANET – Advanced Research Projects Agency Network, precursora da Internet. A evolução do hardware e a melhoria da capacidade dos meios de transmissão, levaram ao surgimento de redes cada vez maiores, mais complexas e heterogéneas. Com o objectivo de dar resposta às necessidades de gestão de redes deste tipo, foram sendo desenvolvidas, ao longo dos anos, sucessivas ferramentas de gestão e monitorização mais adequadas às necessidades do momento. Neste contexto, desenvolveu-se um protocolo, que ainda hoje é um padrão na gestão das redes, o SNMP – Simple Network Management Protocol [1].

Aproveitando a evolução de alguns componentes de rede, como os comutadores e encaminhadores, cujo funcionamento e gestão têm vindo a subir de nível nas camadas do modelo OSI (Open Systems Interconnection), têm surgido, ao longo dos últimos anos, arquitecturas distintas, cada vez mais próximas das camadas aplicacionais. Por outro lado, o aumento considerável das taxas de transmissão das redes actuais tem possibilitado que tais modelos de gestão sejam cada vez mais complexos. Neste contexto surgiram modelos como WBEM (Web Based Enterprise Management) [2] e WS-Management (Web Services for Management) [3].

Entre os diversos modelos propostos para gerir uma rede actual, podemos distingui-los a dois níveis: Gestão de Sistemas e Gestão de Redes. Nos primeiros, estão incluídos os modelos WBEM e WS-Management, nos segundos, destaca-se o SNMP. A Figura 2.1 pretende ilustrar este ponto.

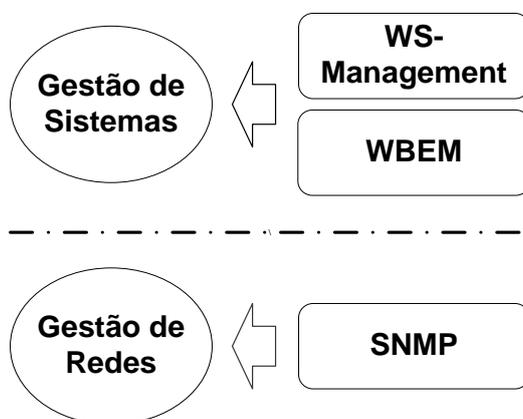


Figura 2.1 - Sistemas de Gestão

Este capítulo irá abordar, em primeiro lugar, o modelo de gestão baseado no protocolo SNMP (actualmente na terceira versão) e depois, os modelos WBEM e WS-Management. Serão feitas considerações comparativas entre as tecnologias em análise, e serão apontadas algumas das implementações actuais mais conhecidas.

## 2.2 Simple Network Management Protocol (SNMP)

Este protocolo, desenvolvido na década de 80 pelo IETF (Internet Engineering Task Force), surgiu com o objectivo de disponibilizar, face às necessidades da altura, uma forma simples e prática de realizar a monitorização e gestão de equipamentos de interligação numa rede TCP/IP.

O protocolo SNMP introduz um modelo de gestão que, por abuso de linguagem, recebe também o nome “Modelo de Gestão SNMP” ou, simplesmente, “Gestão SNMP”.

Este modelo de gestão assenta em 4 componentes [1]:

- nós de rede sujeitos a gestão (Agentes)
- estações de gestão (Gestor);
- informações de gestão (mensagens);
- protocolo de gestão – SNMP.

O diagrama da Figura 2.2 ilustra um sistema de gestão baseado em SNMP onde estão representados o Gestor e alguns tipos de Agentes comuns a este sistema.

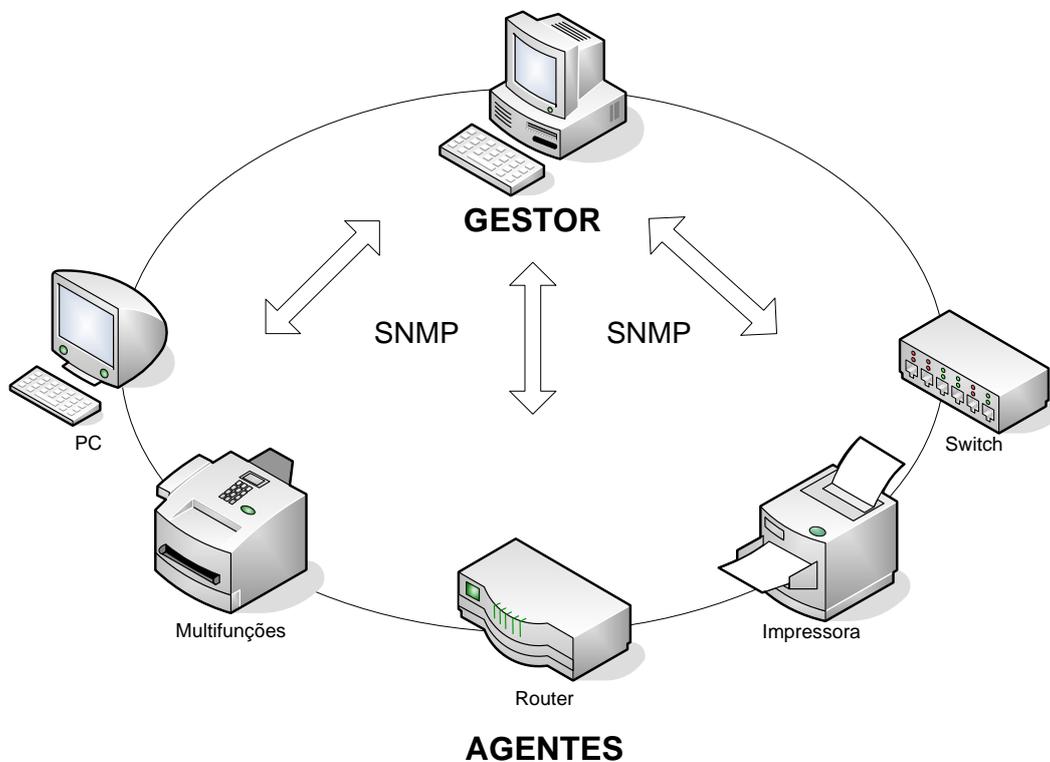


Figura 2.2 - Sistema de Gestão SNMP

Cada dispositivo de rede (Agente) é visto como um conjunto de objectos (variáveis) que representam informações referentes ao seu estado actual e são disponibilizadas para consulta e/ou alteração por parte do sistema gestor. O Gestor é responsável pela monitorização, relatórios e decisões, caso ocorram problemas, enquanto que o Agente tem ao seu cargo o envio e a alteração de informações bem como a notificação de ocorrência de eventos específicos ao Gestor (Traps).

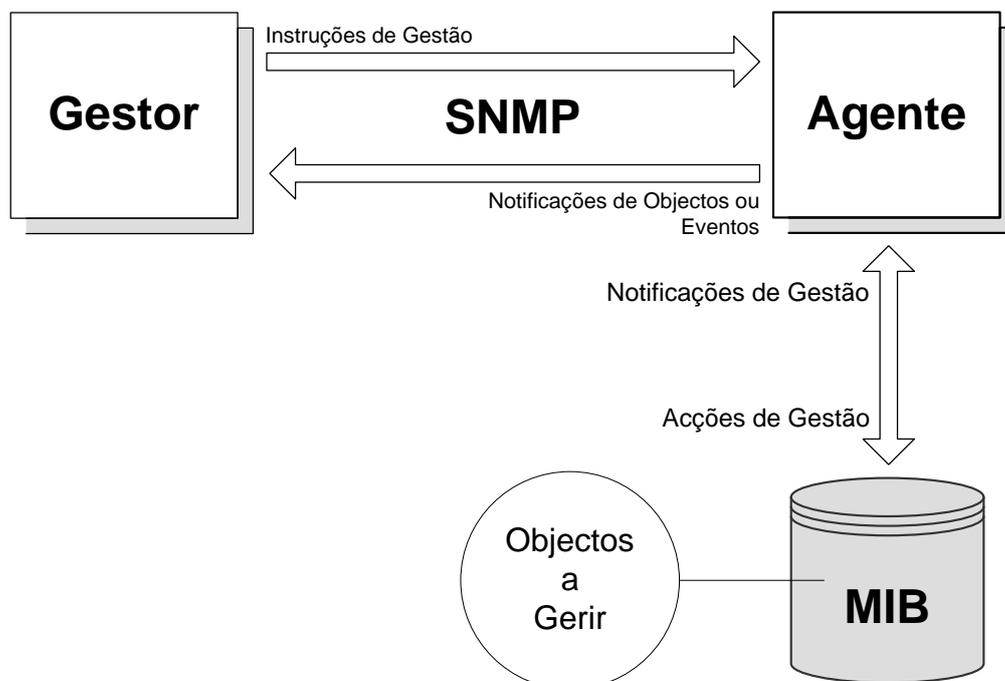
Para que este modelo funcione é necessário que,

1. a máquina gerida possua um Agente SNMP e uma base de informações de gestão – a MIB (Management Information Base) [4];
2. Agente e Gestor pertençam ao mesmo grupo, designado, comunidade.

O Agente corresponde ao processo executado em dispositivos geridos que estão espalhadas por uma rede TCP/IP que podem ser comutadores, encaminhadores, computadores, servidores de impressão, etc. – ou qualquer outro dispositivo com capacidade para comunicar informação de estado. O Agente constitui o servidor SNMP – responsável pela manutenção das informações de gestão da máquina. Cada Agente mantém uma base de dados local, composta por objectos (variáveis) que descrevem o seu estado e histórico de eventos. Mesmo assim, neste modelo, os Agentes são o mais simples possível, estando a maioria da complexidade do lado do Gestor.

O Gestor corresponde ao software de gestão SNMP executado num computador que permite a obtenção de informações de gestão dos dispositivos geridos, através da comunicação com um ou vários Agentes, enviando comandos e recebendo respostas. O Gestor constitui o cliente SNMP.

A Figura 2.3 representa os principais elementos constituintes da arquitectura de um sistema de gestão SNMP.



**Figura 2.3 – Arquitectura do sistema de gestão SNMP**

## 2.2.1 Protocolo SNMP – comunicação entre Gestor e Agente

A comunicação entre Gestores e Agentes é feita com base no protocolo SNMP. Este protocolo utiliza os serviços do protocolo UDP – User Datagram Protocol.

Na maior parte das situações, a troca de mensagens entre Gestor e Agente segue o seguinte modo: a estação gestora solicita informações ao Agente ou força-o, de alguma forma, a actualizar o seu estado; o Agente responde com as informações solicitadas ou confirma ao Gestor que actualizou o seu estado de acordo com o solicitado.

A seguir descrevem-se as mensagens usadas no protocolo SNMP e respectivas funções, agrupadas consoante os intermediários na comunicação:

### Gestor para Agente

- *Get-request* – pede o valor de uma ou mais variáveis, solicitando que os nomes das mesmas sejam explicitamente divulgados;
- *Get-next-request* – pede o valor da variável seguinte à corrente, por ordem alfabética, permitindo que o gerente percorra toda a MIB;
- *Get-bulk-request* – destina-se a transferências em alto volume, pedindo o valor de toda uma tabela de variáveis;
- *Set-request* - permite ao Gestor actualizar as variáveis de um Agente desde que a especificação do objecto o permita;

### Gestor para outro Gestor

- *Inform-request* – solicitação de carácter informativo que permite a um Gestor informar outro sobre quais as variáveis que está a gerir – descreve a MIB local.

### Agente para Gestor

- *Trap* – é usada pelo Agente para comunicar ao Gestor que um evento previamente determinado ocorreu.

A Figura 2.4 ilustra o sistema de gestão de redes baseado em SNMP, indicando quais os protocolos envolvidos na comunicação SNMP e o tipo de mensagens trocadas entre o Gestor e os objectos geridos (Agentes).

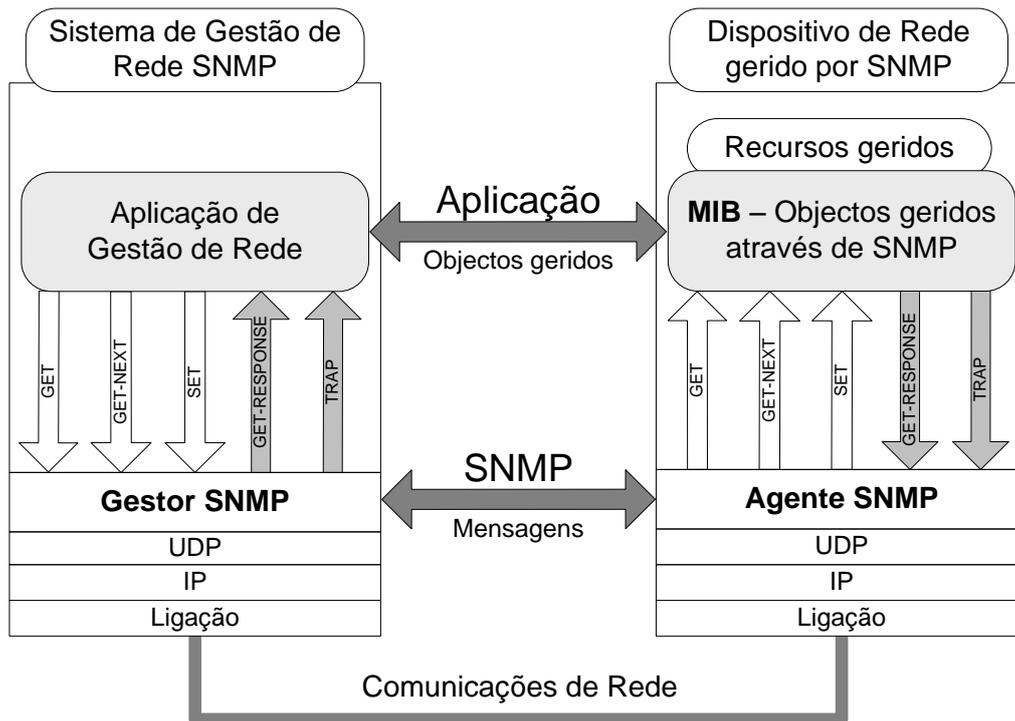
## Princípio de Funcionamento do SNMP (situação-tipo)

O princípio de funcionamento deste modelo de gestão pode ser explicado com base na seguinte situação-tipo (baseada em ambiente Windows) que descreve a forma como Gestor e Agente trocam instruções (mensagens) entre si.

1. O Gestor SNMP envia um pedido a um determinado Agente utilizando o seu endereço IP e o seu nome. O pedido é passado a um socket cujo porto é 161 (UDP). O nome é resolvido para endereço IP, utilizando os métodos de resolução disponíveis.
2. É criado um pacote SNMP que contém uma instrução (*Get*, *Get-Next* e *Set*) para um ou mais objectos, bem como o nome da comunidade e outra informação de validação. Este pacote é encaminhado para o porto 161 (UDP) do Agente.
3. O Agente SNMP recebe o pacote e verifica o nome da comunidade. Se for inválido, ou o pacote estiver corrompido, o pacote é descartado. Caso o nome da comunidade

seja válido, é verificado o nome ou endereço do Gestor que faz o pedido. De seguida o pedido é passado à respectiva DLL (Dynamic Link Library), sendo que o identificador do objecto pretendido é mapeado para a função API (Application Programming Interface) e a chamada API é executada. A DLL devolve a informação ao Agente. Tudo isto é válido, desde que o Agente esteja autorizado a aceitar pacotes do sistema de gestão, se tal não acontecer, os pacotes com os pedidos recebidos, serão descartados.

4. O pacote SNMP é devolvido pelo Agente ao sistema de gestão SNMP (Gestor) com a informação pedida pelo Gestor.



**Figura 2.4 – Sistema de Gestão SNMP**

### Comunidade SNMP

Associado ao Gestor SNMP está o conceito de Comunidade. Assim sendo, antes de se instalar o software de gestão SNMP é necessário definir uma comunidade SNMP. Uma comunidade é um conjunto de Agentes SNMP e é identificada pelo seu nome. A utilização de um nome de comunidade permite oferecer alguma segurança e limitação nos pedidos feitos a Agentes.

As comunicações entre Gestor e Agente são regidas pelos seguintes pressupostos:

1. um Agente não aceita pedidos feitos por comunidades para as quais não esteja configurado;
2. um Agente pode pertencer a várias comunidades em simultâneo, iniciar Traps e responder aos pedidos que lhe são feitos pelos Gestores das mesmas.

## 2.2.2 MIB, SMI e ASN.1

O Gestor SNMP interage com o Agente com o objectivo de ter acesso à informação de gestão contida na sua base de dados local do Agente, a MIB [4], seja para simples consulta ou para manipulação da mesma. A informação armazenada na MIB obedece a uma estrutura específica hierárquica, designada por SMI (Structure of Management Information) [5]. A troca da informação de gestão é feita com base numa notação sintáctica abstracta designada por ASN.1 (Abstract Syntax Notation One) [6].

De seguida é feita uma explanação mais detalhada destes três componentes do sistema de gestão SNMP cuja compreensão é determinante para entender a forma como o Gestor interage com os objectos a gerir no Agente.

### MIB - Management Information Base

Num ambiente típico de rede, existe uma quantidade grande de dispositivos com funções, características e origens distintas. Esta heterogeneidade obriga a que a informação de gestão a circular na rede respeite regras bem definidas. Uma vez que cada dispositivo de rede a gerir mantém vários objectos (variáveis) que descrevem o seu estado, torna-se necessário que os mesmos estejam organizados numa estrutura de informação comum. Esta estrutura é designada por MIB – conjunto de objectos a gerir que procura abranger todas as informações necessárias a uma gestão eficaz da rede - a informação que um Gestor pode pedir a determinado Agente está contida na MIB que o mesmo possui [4, 7].

A Figura 2.5 apresenta o modelo hierárquico (em forma de árvore) definida pela ISO (International Standard Organization) para representar a estrutura da MIB, juntamente com a sub-árvore referente aos objectos usados numa segunda MIB, que surgiu posteriormente, a MIB II [8]. São apresentados os identificadores e nomes dos objectos geridos pelo SNMP, de acordo com a forma como são agrupados.

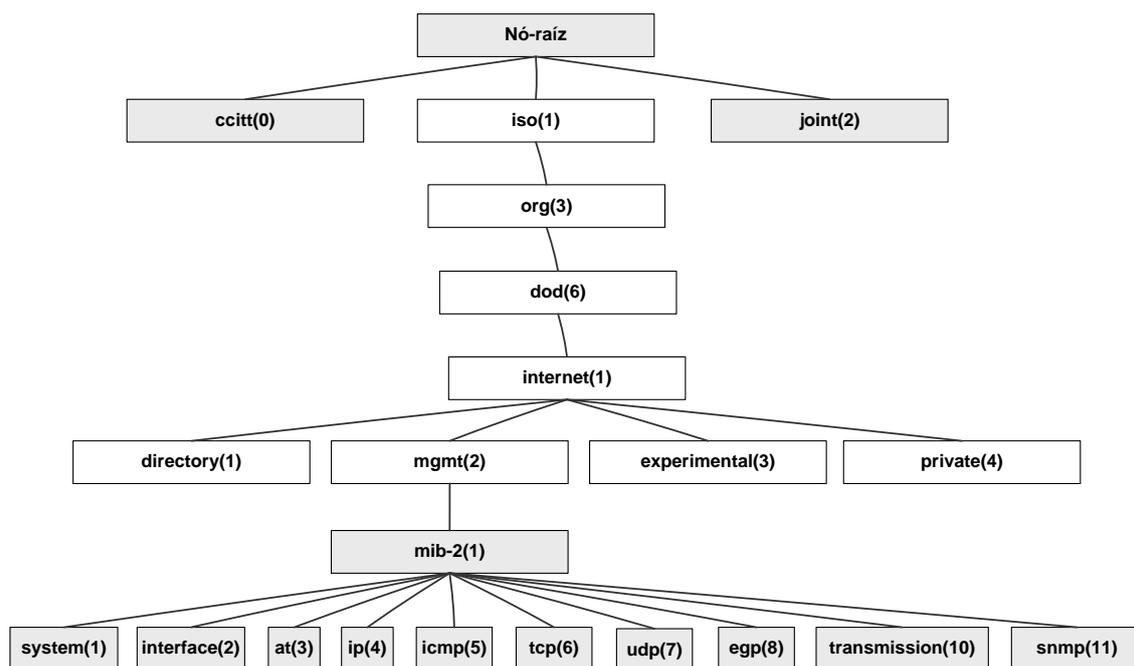
Podemos verificar que o nó da raiz não possui qualquer rótulo embora possua três sub-níveis (nós).

- *ccitt(0)* – administrado pela CCITT (Consultative Committee for International Telegraph and Telephone);
- *iso(1)* – administrado pela ISO;
- *joint-iso-ccitt(2)* – administrado em conjunto pela CCITT e pela ISO.

Sob o nó *iso(1)* fica o nó que pode ser utilizado por outras instituições: o *org(3)*, ficando abaixo deste, o *dod(6)* - pertence ao departamento de defesa dos EUA. O departamento de defesa dos EUA reservou o sub-nó *internet(1)* destinado à comunidade Internet, administrado pela IAB (International Activities Board).

Abaixo do nó *internet(1)* existem os nós: *directory(1)*, *mgmt(2)*, *experimental(3)* e *private(4)*. Destes importa destacar os seguintes:

- *mgmt(2)* – é neste nó que ficam as informações de gestão (conforme é possível observar, é sob este que está o nó da MIB II);
- *experimental(3)* – sob este nó estão as MIBs experimentais;
- *private(4)* – sob este fica o nó *enterprises(1)*, sob o qual, ficam os nós das indústrias de equipamentos.



**Figura 2.5 – MIB: modelo hierárquico**

Finalmente chegamos ao nó *mib-2(1)*. Os objectos que constituem a sub-árvore referente à MIB II, são usados para obter informações específicas dos dispositivos da rede. Esses objectos estão divididos em 10 grupos. Segue-se uma breve explicação de cada um destes grupos.

- *System* – permite que o Gestor saiba o nome do dispositivo, o fabricante, informações sobre o hardware e o software, a localização na rede e o que deverá fazer. Outras informações fornecidas são o horário da última vez que o dispositivo foi iniciado e o nome e endereço da pessoa de contacto. Assim, é possível que outra entidade, que não a empresa detentora dos dispositivos, possa gerir seus sistemas permitindo que a mesma compreenda facilmente a configuração a gerir e quem deverá ser contactado caso ocorram problemas com os diferentes dispositivos.
- *Interfaces* - lida com os adaptadores de rede. Controla o número de pacotes e bytes enviados, recebidos e descartados na rede, o número de broadcasts e o tamanho da fila de saída.
- *AT* - fornece informações sobre o mapeamento de endereços (por exemplo, Ethernet em endereços IP).
- *IP* - trata do tráfego IP recebido e emitido pelo nó. É particularmente sensível ao número de pacotes descartados (por qualquer que seja o motivo). Disponibiliza estatísticas sobre a fragmentação e a remontagem de dados. Estas informações são particularmente importantes quando os dispositivos a gerir são routers.
- *ICMP* – refere-se às mensagens de erro IP. Dispõe de um contador que regista o número de um tipo específico de mensagens de erro ICMP.

- *TCP* - regista a quantidade actual de ligações abertas, bem como o valor acumulado das mesmas; regista também o número de segmentos enviados e recebidos e estatísticas de erros.
- *UDP* - regista o número de pacotes UDP enviados e recebidos e quantos pacotes enviados não foram entregues devido a uma porta desconhecida ou a outro erro.
- *EGP* – utilizado em encaminhadores compatíveis com o protocolo EGP (Exterior Gateway Protocol) [9]. Controla quantos pacotes de um determinado tipo foram enviados, quantos foram recebidos e encaminhados correctamente e quantos foram recebidos e descartados.
- *Transmission* - marcador de lugar para MIBs de meios físicos específicos. Por exemplo, nesse grupo é possível manter estatísticas especificamente relacionadas à Ethernet.
- *SNMP* – destina-se ao cálculo de estatísticas sobre a operação do próprio SNMP. Quantas mensagens estão sendo enviadas, quais os tipos dessas mensagens, etc.

A descrição de um objecto da MIB é feita com base numa notação numérica extremamente condensada, baseada na estrutura SMI. A seguir apresenta-se um exemplo da forma como descrever um objecto na MIB II, na circunstância o objecto *iso.org.dod.internet.mgt.mib.system.sysdescr* que tem o seguinte OID (Object Identifier): 1.3.6.1.2.1.1.1. A Tabela 2.1 apresenta os 10 grupos da MIB II já descritos anteriormente, seus identificadores (ID) e o número de objectos de cada grupo.

Com a MIB II passou a ser possível obter informações gerais de gestão sobre o dispositivo a gerir, como por exemplo, o número de pacotes a transmitir e o estado da interface.

<b>Grupo</b>	<b>ID</b>	<b>N.º Objectos</b>
system(1)	1.3.6.1.2.1.1	7
Interface(2)	1.3.6.1.2.1.2	23
at(3)	1.3.6.1.2.1.3	3
ip(4)	1.3.6.1.2.1.4	42
icmp(5)	1.3.6.1.2.1.5	26
tcp(6)	1.3.6.1.2.1.6	19
udp(7)	1.3.6.1.2.1.7	6
egp(8)	1.3.6.1.2.1.8	20
Transmission(10)	1.3.6.1.2.1.10	0
snmp(11)	1.3.6.1.2.1.11	29

**Tabela 2.1 – Grupos da MIB II**

## **SMI - Structure of Management Information**

Dado que, um objecto a gerir, representa uma visão abstracta de um recurso real do dispositivo de rede, então, todos os recursos da rede a serem geridos, deverão ser modelados segundo uma estrutura de dados específica cujas regras de construção deverão estar de acordo com a SMI - a estrutura de informações de gestão [10].

A SMI é um conjunto de documentos que definem:

- a forma de distribuição e agrupamento das informações;
- as sintaxes permitidas;
- os tipos de dados permitidos.

### **ASN.1 - Abstract Syntax Notation One**

A linguagem formal usada para especificar os objectos da MIB designa-se por ASN.1 [6, 11]. Esta notação sintáctica abstracta (formal), retirada da OSI, é usada para descrever dados transmitidos por protocolos de comunicações (tal como o SNMP), de forma independente da linguagem usada, representação física dos dados, aplicação e complexidade – não são levadas em conta a estrutura dos dados e as restrições do equipamento onde vai ser implementada.

Para cada objecto a gerir são definidos, o nome, o identificador, a sintaxe, a definição e o acesso, sendo as instâncias do objecto, as variáveis. Passando a descrever,

- o nome do objecto é composto por uma pequena string de texto;
- o identificador do objecto (OID) é formado por um conjunto de números inteiros separados por pontos (conforme exemplificado na Tabela 2.1);
- a sintaxe do objecto descreve o formato ou valor da informação;
- a definição do objecto é uma descrição textual do mesmo;
- o acesso ao objecto representa o tipo de controlo que se pode ter sobre o objecto, podendo ser de leitura, leitura/escrita e não acessível.

### **2.2.3 Monitorização remota e segurança**

A versão inicial do protocolo SNMP [12] fornecia um pequeno, mas poderoso, conjunto de ferramentas que permitiam monitorizar e controlar dispositivos integrantes de uma rede. No entanto, esta primeira versão do SNMP apresentava duas limitações, embora permitisse a monitorização de componentes e nós de uma rede, não permitia a monitorização da própria rede, além disso, não possuía mecanismos de segurança, privacidade e autenticação.

No intuito de vencer as limitações relativas à monitorização da rede, desenvolveram-se vários objectos de gestão que passaram a ser integrados numa MIB específica. Estes objectos receberam a designação RMON (Remote Network Monitoring) [13].

Para tentar ultrapassar as questões de falta de segurança, foram sendo acrescentadas melhorias na versão inicial do protocolo. Surgiram várias versões sendo que as mais relevantes foram o SNMPv2 [14] e SNMPv3 [15].

O SNMPv2 definia uma série de características ao nível da segurança que, no entanto, acabaram por ser preteridas devido a dois factores: por um lado, a ausência de consenso relativamente à implementação dessas características e, por outro lado, reconhecidas falhas de definição das mesmas. Outros melhoramentos foram surgindo em novas versões do SNMPv2 mas, apesar dos esforços para aumentar a segurança, a versão final do SNMPv2 não acrescentava novos mecanismos de segurança ao protocolo SNMP. Essa lacuna só seria devidamente tratada na terceira versão do SNMP – SNMPv3.

O Anexo 1 apresenta uma tabela que sintetiza os RFCs mais recentes relacionados com SNMP e as suas principais versões.

## 2.2.4 Limitações do protocolo SNMP

As principais limitações do modelo SNMP dão-se ao nível da escalabilidade e eficiência. Tais limitações tornam-se problemáticas com o aumento das informações de gestão. Isso pode acontecer caso surja a necessidade de extensão da MIB (p. ex., inclusão de novos objectos) ou pelo crescimento da própria rede. Estes dois factores combinados acabam por aumentar consideravelmente o tamanho da MIB suportada pelos Agentes.

Os problemas de escalabilidade e eficiência tornam-se depois evidentes nos seguintes aspectos:

- aumento do overhead da rede – dado que o tráfego de gestão não está associado a um serviço disponível ao utilizador da rede, assim, todo o tráfego associado à gestão é considerado overhead – no SNMP, com o aumento de informações de gestão, aumenta, de forma proporcional o overhead da rede;
- aumento de atrasos – no SNMP, o tempo de consulta a um Agente aumenta proporcionalmente ao número de Agentes geridos, provocando atrasos na entrega ou no processamento de informações de gestão;
- capacidade de processamento do Gestor – o aumento da quantidade de Agentes provoca um aumento na informação de gestão a processar pelo Gestor; dado que o SNMP não possui uma estrutura de gestão distribuída nem um modelo de gestão hierárquico, aumentam as exigências nos recursos de hardware do lado do Gestor (CPU, memória, etc.); esses recursos não podem aumentar indefinidamente (problemas de custos e limitações com o próprio hardware existente);
- limite do segmento de rede do Gestor – dado que o SNMP utiliza um mecanismo centralizado de gestão, então os dados de todos os Agentes têm de passar pelo segmento de rede ao qual o Gestor está ligado, podendo causar sobrecarga naquele segmento.

A falta de mecanismos eficientes de transferência atómica de grandes quantidades de informações, a ausência de suporte a gestão distribuída, a característica verbosa dos objectos da MIB, causando OIDs longos, a ausência de compressão de dados de gestão e o uso de um protocolo de transporte não confiável (UDP), acabam por influenciar a baixa escalabilidade e eficiência deste modelo.

No campo da configuração de redes, surgem ainda outras limitações que explicam a razão pela qual o modelo SNMP se tem limitado às tarefas de monitorização, gestão de desempenho e falhas. A representação das informações de gestão através do SMI é limitada a tabelas de tipos escalares, não correspondendo à necessidade que os dados de

configuração de redes têm de modelos hierárquicos mais ricos; operações de alto nível (tais como, download, activação, desactivação, etc...) necessárias para tarefas de configuração de redes não estão disponíveis neste modelo (embora a operação *Set-request* possa ser usada de forma indirecta para esse fim, provoca aumento na complexidade das aplicações de gestão).

Razões semelhantes às anteriores explicam ainda porque o modelo SNMP não é extensível à gestão integrada de redes, sistemas, serviços e negócios.

As limitações do SNMP não se limitam a aspectos técnicos. Dada a especificidade desta tecnologia, o seu domínio exige especialização dos intervenientes. Além disso, o desenvolvimento de Agentes e Gestores é lento e dispendioso

A Tabela 2.2 sintetiza as principais limitações do modelo SNMP que foram apresentadas, agrupando-as em quatro grupos principais: escalabilidade e eficiência, gestão integrada e configuração de redes, segurança e, questões comerciais.

<b>Limitações do Modelo SNMP</b>	
<b>Escalabilidade e eficiência</b>	
Transferência ineficiente de grandes quantidades de informação	
Não suporta gestão distribuída e hierárquica	
Endereçamento verboso dos objectos da MIB (causa OIDs longos)	
Ausência de mecanismos de compressão de dados	
Protocolo de transporte não confiável	
<b>Gestão integrada e configuração de redes</b>	
Modelo de informação (SMI) limitado e não-hierárquico	
Interface de gestão limitada	
<b>Segurança</b>	
Mecanismo de comunidades limitado (SNMPv3 já inclui mecanismos de segurança)	
<b>Questões comerciais</b>	
Custo e tempo de desenvolvimento elevados; tecnologia de domínio específico	

**Tabela 2.2 – Resumo das limitações do modelo de gestão SNMP**

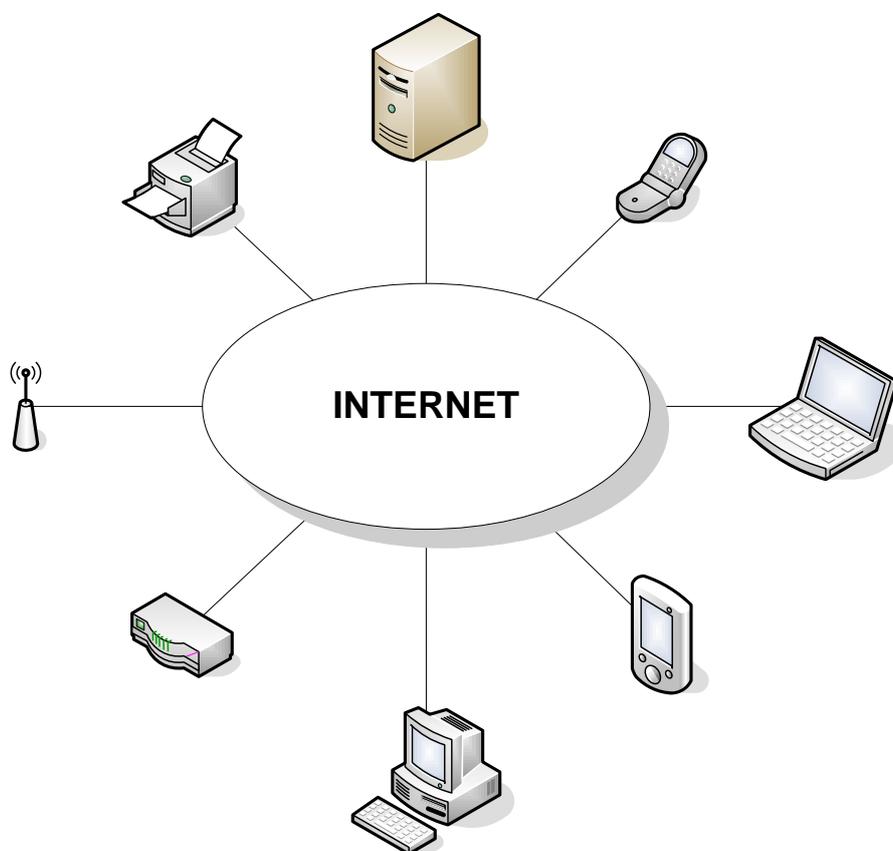
## **2.3 Web Based Enterprise Management (WBEM)**

O sistema de gestão de redes WBEM surgiu em 1996 em resultado de uma iniciativa conjunta de um consórcio de fabricantes em que se destacavam a BMC Software, Microsoft, Intel, Cisco e Compaq [16]. Visava disponibilizar um conjunto de

tecnologias-padrão de gestão e Internet com o objectivo de uniformizar a gestão de ambientes de sistemas computacionais distribuídos [2].

No que toca à gestão e manutenção, as redes actuais apresentam-se como extremamente complexas e difíceis. Existe uma enorme diversidade de dispositivos e tecnologias, cujo crescimento em tamanho e complexidade não pára. Por outro lado, o crescimento da Internet tem levado a que a gestão feita à distância usando uma plataforma de comunicação comum e de fácil utilização seja cada vez mais desejável. Associado a este último aspecto estão tecnologias de comunicação desenvolvidas para a Internet, como os protocolos HTTP (Hypertext Transfer Protocol) [17] e HTTPS (HyperText Transfer Protocol Secure) [18] e a linguagem XML (Extensible Markup Language) [19].

Na Figura 2.6 é possível observar a heterogeneidade dos vários elementos constituintes de uma rede actual típica com suas especificidades e exigências.



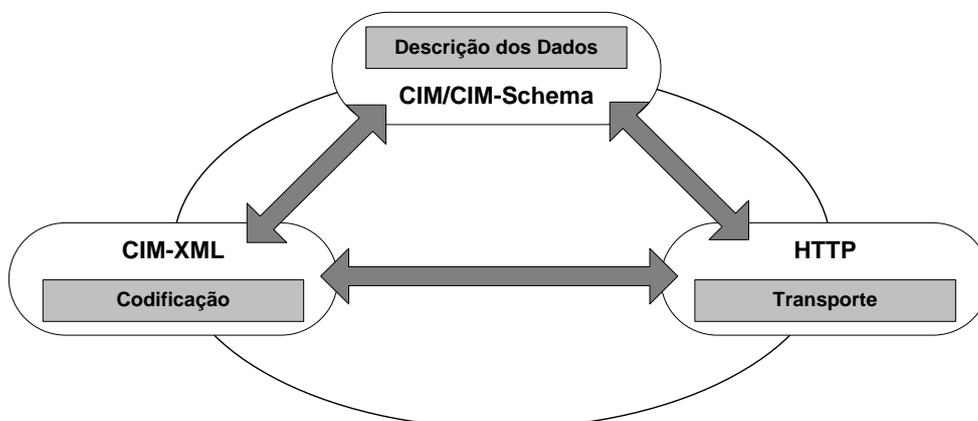
**Figura 2.6 – Rede típica actual**

Neste contexto, a DMTF (Distributed Management Task Force) propôs em 1998 um padrão de gestão de redes baseado na arquitectura WBEM. Este padrão pretende rentabilizar as tecnologias Web tirando partido dos seguintes elementos fundamentais a qualquer ambiente de gestão:

- um modelo de descrição de dados orientado a objectos – CIM (Common Information Model);
- um mecanismo de descrição/codificação dos dados de gestão (baseado em XML) – CIM-XML;

- um protocolo de transporte de informação via Web – HTTP.

A Figura 2.7 apresenta os três principais conceitos envolvidos no modelo de gestão WBEM e a interação existente entre eles.



**Figura 2.7 – Modelo WBEM simplificado**

### 2.3.1 Common Information Model (CIM)

O CIM é um modelo de dados que fornece um formato comum para definir/modelar as informações de gestão de sistemas de informação, redes, serviços e aplicações permitindo extensões adicionais de acordo com as especificações dos fabricantes. Para que tais informações possam ser lidas e compreendidas com facilidade e passíveis de ser analisadas por um compilador, utiliza-se um formato de texto para as representar, baseado na linguagem IDL (Interface Definition Language), designado por MOF (Managed Object Format) [20, 21]. Tal como a MIB usada para descrever recursos geridos por SNMP, a CIM permite descrever informação de gestão em formato de texto. Usando CIM e MOF, é possível modelar e visualizar os recursos a gerir seguindo uma abordagem orientada a objectos.

A sintaxe MOF é uma maneira de descrever definições de objectos em forma de texto e estabelece a sintaxe para escrever as definições. Os principais componentes de uma especificação MOF são descrições textuais de classes, associações, propriedades, referências, métodos e declarações de instâncias e seus qualificadores associados. Um ficheiro MOF é constituído, basicamente, por uma série de declarações de classes e instâncias de classes [20].

A Tabela 2.3 apresenta exemplos que ilustram a sintaxe MOF bem como respectiva descrição.

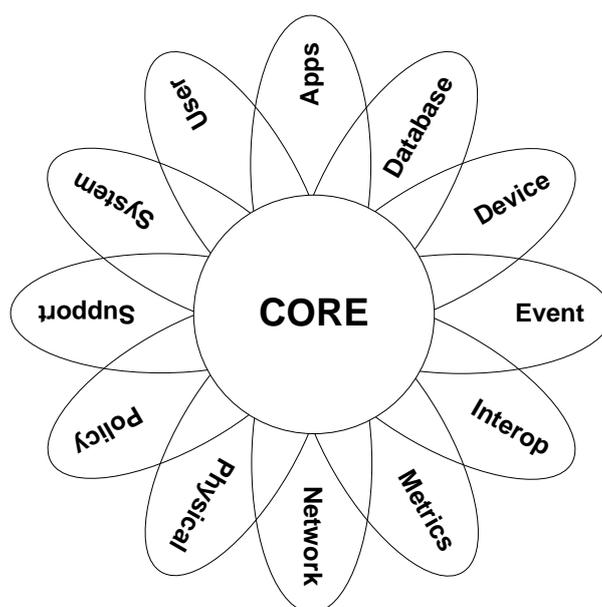
De referir ainda que não é possível implementar métodos em MOF. Essa tarefa está ao cargo dos Providers (ver 2.3.4 – Arquitectura WBEM).

O CIM é composto pela Especificação e pelo Esquema (Schema) [20]. A Especificação CIM define a linguagem (MOF) e a metodologia para descrever dados de gestão para integração com outros modelos de gestão (p. ex. MIBs do SNMP). A Especificação não descreve: implementações CIM específicas, APIs ou protocolos de comunicação. O CIM Schema fornece a descrição do modelo de gestão de acordo com áreas específicas

como sejam, por exemplo, armazenamento ou aplicações. Neste contexto, a DMTF propôs uma hierarquia de classes baseada em camadas que representa correctamente elementos a gerir das várias áreas distintas. Cada área de gestão é representada através de um CIM Schema específico conforme se pode observar num esquema conceptual de várias camadas apresentado na Figura 2.8.

Sintaxe MOF	Descrição
<pre>class ClassName {   data type PropertyName;   data type MethodName(); };</pre>	<p>Aqui temos:</p> <ul style="list-style-type: none"> <li>- uma classe com o nome ClassName,</li> <li>- uma propriedade, PropertyName, e</li> <li>- um método, MethodName.</li> </ul>
<pre>instance of ClassName {   PropertyName=value; };</pre>	<p>Aqui é possível observar a semelhança entre a declaração de uma instância e a declaração de uma classe.</p>
<pre>[Description("This class is an example")] class ClassName {   [Key, Minvalue(5), Maxvalue(10)] data type   PropertyName;   [Description("This method is an example")] data type   MethodName(); };</pre>	<p>Antes das declarações de propriedades e métodos, é possível colocar um qualificador entre parêntesis rectos ("[" e "]"). Cada classe, propriedade ou método pode ter diversos qualificadores separados por vírgulas (",").</p> <p>Neste caso temos quatro qualificadores diferentes: DESCRIPTION, KEY, MINVALUE e MAXVALUE. São qualificadores CIM padrão.</p> <p>Também é possível, caso seja necessário, declarar novos qualificadores.</p>

**Tabela 2.3 – Exemplos de sintaxe MOF**



**Figura 2.8 – CIM-Schema referente ao Core Model**

O CIM Schema é constituído por um núcleo (Core), um conjunto mais pequeno de classes, associações e propriedades que fornecem a base para a análise e descrição de sistemas geridos sobre o qual são depois construídos os vários Schemas para representar áreas específicas de gestão. Assim, diferentes áreas de gestão são trabalhadas pelos diferentes organismos-padrão da DMTF que se especializam nessas áreas.

Para além do Core, relacionado com informações que dizem respeito a todas as áreas de gestão, o CIM Schema tem ainda o Common e o Extension. O Common é um modelo de informações comuns a uma área específica (p. ex. Sistema, Aplicações, Rede, Dispositivo), porém, independente da plataforma. O Extension representa extensões específicas dos diversos modelos Common permitindo a gestão de ambientes e plataformas específicas. Pelo que foi exposto depreende-se que os modelos Core e Common não poderão fazer parte da Especificação do CIM uma vez que são produzidos de forma independente da plataforma.

Uma vez concluídos o modelo de dados CIM e as definições MOF referentes aos recursos a gerir, o pacote resultante é importado para um gestor de objectos CIM, o CIMOM (CIM Object Manager) – ver Figura 2.11. O CIMOM constitui um repositório central onde os clientes numa rede (as aplicações de gestão) podem obter informação sobre os recursos a gerir. Os ficheiros de texto MOF importados para o CIMOM contêm as definições dos CIM Schema implementados, possibilitando, através do respectivo compilador MOF, a definição de classes, propriedades e qualificadores relacionados com determinado recurso a gerir e a criação de instâncias dessas mesmas classes no próprio CIMOM.

A Tabela 2.4 apresenta algumas das operações CIM predefinidas [22].

### **2.3.2 Codificação dos dados de gestão em CIM-XML**

Embora a linguagem MOF forneça uma representação em forma de texto de informações de gestão modeladas usando o CIM, tal representação, por si só, não permite que aplicações distintas possam transferir informações de gestão entre si. Para tal é necessário representar informação de gestão num formato padrão, passível de ser mapeado para um protocolo de comunicação.

A escolha recaiu na linguagem XML [23], uma linguagem de marcadores conhecida como meta-linguagem uma vez que é usada para criar outras linguagens de marcadores. Constitui-se como um formato de texto simples e muito flexível derivado da linguagem SGML (Standard Generalized Markup Language) – (ISO 8879) [19]. A principal vantagem do XML é ser aberto e auto-descritivo - as tags acrescentam semântica aos dados. Isso possibilita a modelação de diferentes tipos de informação fornecendo a estrutura e regras necessárias para a sua descrição numa forma com significado para envio através de redes heterogéneas. O XML tem vindo a desempenhar um papel cada vez mais importante na troca de uma grande variedade de dados na Web ou em outras plataformas.

Um documento XML pode ter em anexo a descrição da sua ‘gramática’ – o XSD (XML Schema Definition) [24-26], que fornece mecanismos para definir e descrever a estrutura, conteúdo e, em certa medida, a sua semântica. Essa descrição é feita usando um mecanismo conhecido como DTD (Document Type Definition).

No sistema WBEM usa-se um XML Schema para descrever informação de gestão CIM e enviá-la através de redes heterogéneas. Classes e instâncias CIM são documentos XML que respeitam as regras definidas por esse XML Schema.

<b>Grupo Funcional</b>	<b>Dependência</b>	<b>Métodos</b>
Leitura Básica	Nenhuma	GetClass EnumerateClasses EnumerateClassName GetInstance EnumerateInstances EnumerateInstanceName GetProperty
Escrita Básica	Leitura Básica	SetProperty
Manipulação de Instâncias	Escrita Básica	CreateInstance ModifyInstance DeleteInstance
Manipulação de Schemas	Manipulação de Instâncias	CreateClass ModifyClass DeleteClass
Associação Transversal	Leitura Básica	Associators AssociatorNames References ReferenceNames
Execução de Consultas	Leitura Básica	ExecQuery
Declaração de Qualificadores	Manipulação de Schemas	GetQualifier SetQualifier DeleteQualifier GetInstance EnumerateQualifiers

**Tabela 2.4 – Operações CIM predefinidas (intrínsecas)**

### 2.3.3 CIM-XML sobre HTTP

Com o intuito de permitir que o WBEM possa operar de forma aberta e padronizada, a DMTF definiu um mapeamento das operações CIM para o HTTP. Este protocolo, usado para transporte de informação via Web, está generalizado, é aberto e existe em todos os sistemas operativos e arquitecturas. O facto de ser executado sobre TCP/IP, permite que o mesmo seja usado para inúmeras tarefas através de extensões dos seus métodos de solicitação, códigos de erro e cabeçalhos [17].

Definindo um número de cabeçalhos de extensão para CIM, aplicações cliente e servidor tomam ‘conhecimento’ que as informações CIM estão sendo enviados. A codificação CIM para XML favorece a abrangência em detrimento da especificidade trazendo compromentimentos ao desempenho. Assim, uma possível conclusão a retirar é que, o carácter mais generalista do sistema WBEM opõe-se à sua própria eficiência,

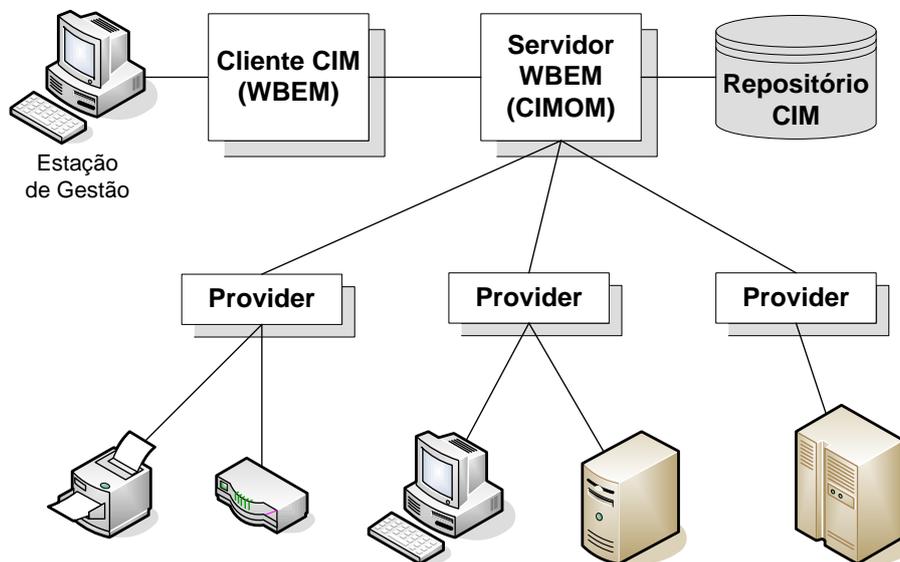
constituindo assim uma lacuna desta tecnologia. O Apêndice 2 mostra uma captura de mensagens WBEM em CIM-XML sobre HTTP trocadas entre Gestor e Agente referentes ao pedido e à respectiva resposta.

### 2.3.4 Arquitectura WBEM

A arquitectura WBEM segue o modelo cliente-servidor e o seu funcionamento assenta, essencialmente, em três entidades:

- um gestor de objectos, o CIMOM, é o elemento central, responsável por receber, fornecer e modificar a informação de gestão armazenada no repositório de dados de acordo com as instruções CIM específicas, verifica a sintaxe e semântica das mensagens e fornece segurança;
- os Providers, um conjunto de processos que actuam como intermediários entre o CIMOM e os recursos remotos geridos, permitindo a gestão uniforme de domínios de gestão heterogéneos;
- as aplicações de gestão que comunicam com o CIMOM via browser através de uma API.

O servidor WBEM (Agente) gere os diversos dispositivos de rede ligados a si através de Providers, responsáveis por interligar os diversos dispositivos a gerir ao CIMOM do Agente. O CIMOM é depois o responsável por interagir com o repositório de dados CIM. O cliente WBEM (Gestor) comunica via HTTP com o servidor WBEM. A Figura 2.9 apresenta um sistema heterogéneo típico gerido através de WBEM.

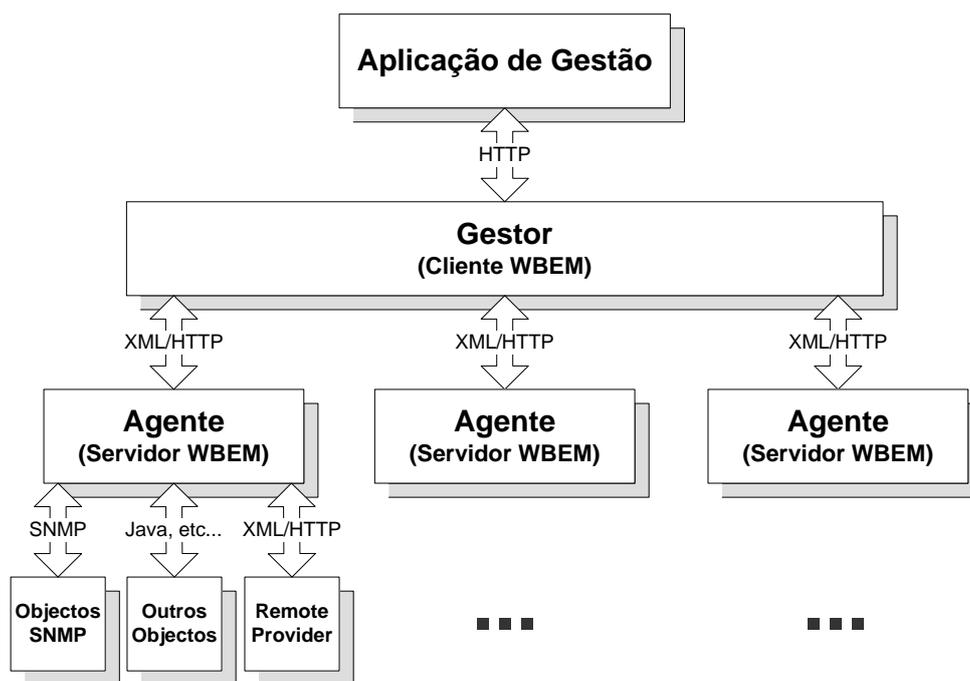


**Figura 2.9 – Exemplo de um sistema heterogéneo gerido através de WBEM**

Numa perspectiva de arquitectura de gestão, no modelo WBEM, o cliente corresponde ao Gestor e os servidores, correspondem aos Agentes. Um utilizador pode, via browser, através de uma aplicação de gestão (API), aceder ao Gestor (cliente WBEM). O Gestor comunica com os diversos Agentes (servidores WBEM) através de CIM-XML sobre

HTTP. Desta forma, o Agente pode obter informação relativa a objectos SNMP, Java ou quaisquer outros.

Caso o Gestor pretenda obter informação de recursos remotos, terá que fazer através do Agente e não acedendo directamente aos Providers desses recursos. É o respectivo Provider que envia para o Agente a informação necessária num formato-padrão passível de ser compreendido pelo CIMOM. Para tal, a comunicação entre Agente e Provider é feita com base em CIM-XML sobre HTTP. A Figura 2.10 ilustra esta explicação, apresentando os principais elementos envolvidos na arquitectura WBEM.

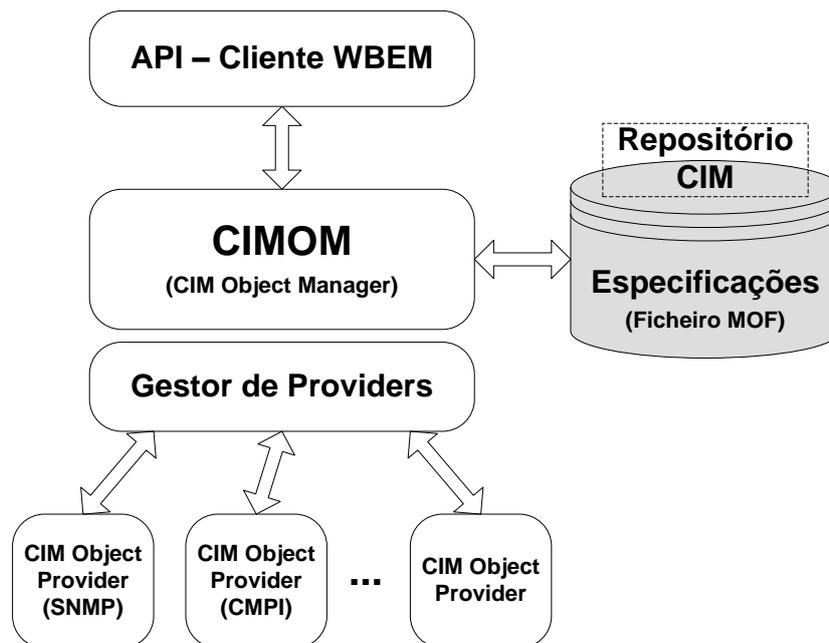


**Figura 2.10 – Arquitectura global do Sistema de Gestão WBEM**

Quando uma aplicação de gestão WBEM é executada, esta acede ao CIMOM que interpreta o pedido relativo a determinado objecto gerido e consulta/actualiza essa informação no repositório de dados. Por outro lado, quando um valor é alterado no objecto gerido, a informação é entregue pelos Providers ao CIMOM que depois se encarrega de actualizar o repositório CIM.

No que diz respeito aos tipos de dados trocados entre os objectos geridos e o CIMOM, apenas os dados estáticos são armazenados no repositório, enquanto que os dinâmicos são consultados aos Providers directamente pelo CIMOM.

Na Figura 2.11 é possível observar os diversos intervenientes neste processo bem como a forma como ocorre a comunicação entre os objectos a gerir, os Providers, o CIMOM e o repositório CIM. Neste contexto observa-se a existência de um Gestor de Providers, responsável por interligar os diversos Providers dos objectos a gerir ao CIMOM.



**Figura 2.11 – Arquitectura WBEM**

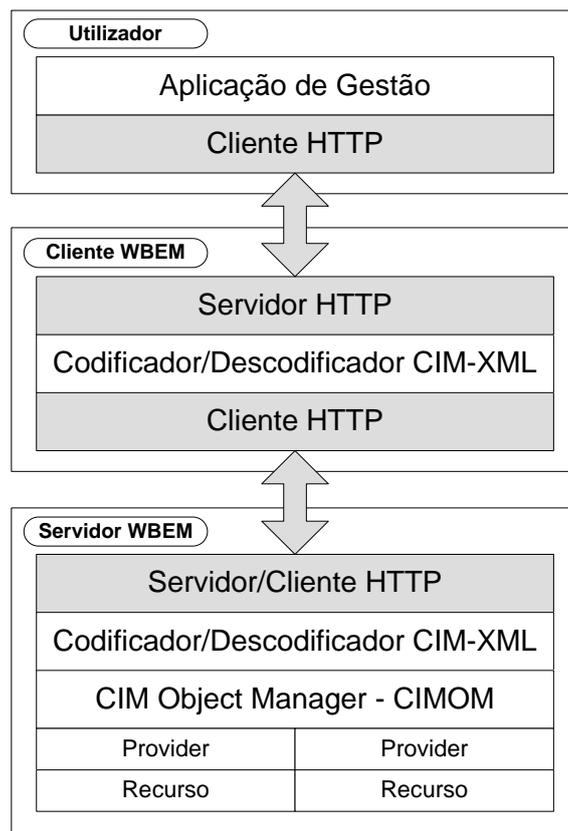
### 2.3.5 Interface Cliente-Servidor (Gestor-Agente)

De acordo com a Figura 2.10, a aplicação de gestão executada do lado do operador, não constitui, por si só, o cliente (Gestor) WBEM. O utilizador comunica com o cliente WBEM via HTTP. A Figura 2.12 esquematiza os diversos intervenientes na arquitectura WBEM bem como os seus limites de actuação.

O cliente WBEM possui, do lado da aplicação de gestão (utilizador), um servidor HTTP que fornece uma interface de utilizador baseada em Web. Este recebe pedidos da aplicação de gestão e fornece informações de gestão exportadas do Agente remoto (servidor WBEM). Neste processo, o Codificador/Descodificador CIM-XML, desempenha a função de adaptar essas trocas de informação. O cliente tem total desconhecimento da forma como os seus pedidos estão a ser tratados do lado do servidor, nem tampouco sabe da existência dos Providers.

O servidor WBEM (Agente) possui um servidor e um cliente HTTP. O servidor HTTP é necessário para que o Gestor se ligue ao Agente, quanto ao cliente, é o que permite ao Agente enviar mensagens ou notificações ao Gestor. As mensagens recebidas do lado do cliente WBEM são convertidas pelo Descodificador CIM-XML e entregues depois ao CIMOM. Quando se pretende enviar informação no sentido contrário, do servidor para o cliente WBEM, o Codificador CIM-XML é o responsável por adaptar especificações CIM para CIM-XML passíveis de serem enviadas por HTTP.

O papel do CIMOM é obter dados requisitados pelo Gestor. Estes podem estar ou não no repositório CIM. Caso não se encontrem no repositório, o CIMOM procura-os no Provider correspondente à operação CIM a ser executada. Os Providers comunicam com objectos geridos para aceder a dados e notificações de eventos provenientes de variadas fontes, como sejam, o registo de sistema ou um dispositivo SNMP. Os Providers conduzem depois essa informação ao CIMOM para ser integrada e interpretada [27].



**Figura 2.12 – Utilizador, cliente e servidor WBEM**

A Figura 2.13 ilustra com algum detalhe uma arquitectura típica de um Agente (servidor) WBEM onde são esquematizados os diversos conceitos que acabamos de expor.

Um dos elementos representados é o Gestor de Providers, responsável por administrar os diversos tipos de Providers que suportam diferentes tipos de linguagens. Estes Providers podem ser geridos por estarem registados no Repositório de Registo de Providers. Na maior parte dos sistemas WBEM estão presentes os seguintes tipos de Providers: Instance Provider, Method Provider, Association Provider e Indication Provider [28].

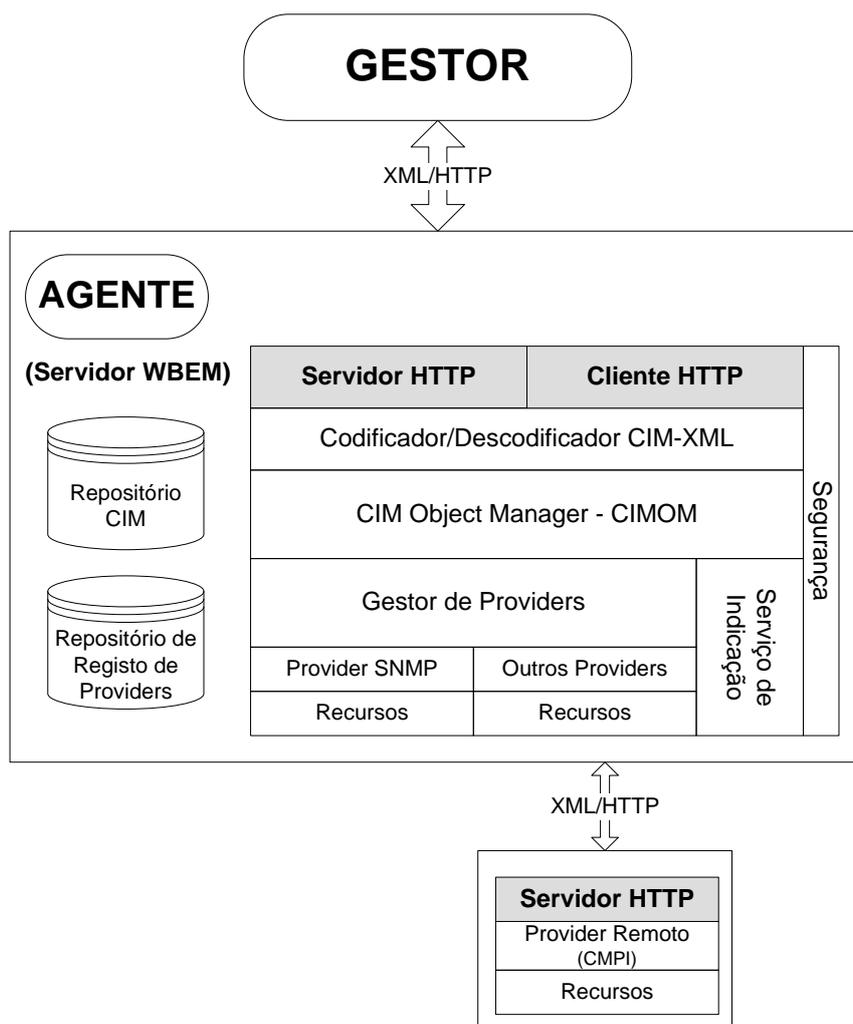
O Instance Provider modifica atributos no Agente, criando, modificando, eliminando e listando, instâncias de determinada classe ou obtendo/alterando, o valor de alguma propriedade. Tipicamente, este tipo de Provider mantêm uma lista de instâncias, quer por responder aos pedidos do operador (aplicação de gestão do lado do cliente WBEM), quer por efectuar um varrimento ao equipamento e verificar o que foi ligado. Por exemplo, após a instalação de um router, não existe qualquer forma do servidor WBEM saber a quantidade ou o tipo de portas que o mesmo contém. Nessa ocasião, um Instance Provider irá comunicar directamente com os drivers de baixo-nível do router para determinar tais características. Quando essa informação for pedida por um cliente WBEM, então o servidor invoca o Instance Provider para recuperar essa informação.

O Method Provider invoca métodos (funções definidas para determinada classe) no Agente com o objectivo de executar operações em objectos geridos. Quando o cliente

WBEM invoca um Method Provider, está na realidade a chamar um RPC (Remote Procedure Call) esperando receber depois o seu valor de retorno. O cliente WBEM especifica a instância (classe) onde o procedimento será invocado, o nome do método e os seus parâmetros. O servidor WBEM invoca o Method Provider referente ao método e classe do pedido e devolve os valores gerados. Como exemplo, podemos considerar um Method Provider que permite parar um serviço: *'void shutdown(boolean immediately);'*.

O Association Provider trata da criação, eliminação, listagem e outras operações entre classes e instâncias de forma dinâmica. Associações criadas dinamicamente, são suportadas por um Association Provider que permite seguir o caminho das instâncias criadas dinamicamente.

O Indication Provider efectua a monitorização através de eventos. Traduz a ocorrência de um evento para uma indicação CIM enviando-a para o CIMOM para posterior processamento e entrega. Diferente dos outros Providers cujo princípio de funcionamento se baseia em respostas a pedidos por parte do servidor WBEM, este tipo de Provider é activado por um acontecimento externo iniciando então a comunicação com o servidor. O Indication Provider irá requerer um mecanismo que lhe permita informar o servidor WBEM da existência de uma indicação. Este processo é conhecido por publicação em CIM-speak.



**Figura 2.13 – Arquitectura do Agente WBEM**

Observando, quer a Figura 2.11, quer a Figura 2.13, verificamos a existência de um outro Provider, também controlado pelo Gestor de Providers, o Provider SNMP. O seu objectivo é poder integrar dispositivos SNMP existentes. Funciona como um gateway WBEM/SNMP. Permite obter especificações MIB para MOF ou MIB para XML e converter operações CIM WBEM em operações SNMP (*Get*, *Set* e *Trap*).

Ainda em relação a diagrama apresentado na Figura 2.13, é possível observar um elemento denominado Serviço de Indicação, responsável por processar as operações de criação, modificação e eliminação da informação de indicação subscrita, onde está incluída a lista de clientes que subscreveram esse serviço.

No que respeita aos Providers remotos, a sua gestão é feita através do CMPI (Common Manageability Programming Interface) [29] – Figura 2.11 e Figura 2.13. O CMPI permite que os Providers sejam executados em sistemas remotos sem a necessidade de um CIMOM extra. Basta que o lado remoto inicie um processo em background que aceita pedidos remotos passando-os para o Provider local. Esta comunicação visa apenas transferir informação relativa a resultados finais.

A Figura 2.13 faz ainda referência à segurança. Um sistema WBEM típico permite a inclusão de três níveis de segurança: comunicações, autenticação e autorização de utilizadores. No primeiro nível, pode ser usado HTTP sobre o protocolo Secure Sockets Layer (SSL) [18]. O SSL é uma tecnologia padrão de segurança para o estabelecimento de uma ligação codificada entre um servidor Web e um browser – HyperText Transfer Protocol Secure (HTTPS). No segundo nível, a especificação CIM-XML pode fornecer autenticação básica através da configuração do cliente WBEM para que o mesmo forneça um identificador e uma palavra-passe do utilizador. No terceiro nível, o controle e autorização no acesso de utilizadores baseia-se no conceito de namespace (espaço de nomes). Uma aplicação de gestão deve ligar-se a determinado espaço de nomes para obter acesso às propriedades dos objectos. Se a autorização por espaço de nomes estiver activa, é possível controlar o acesso a essas propriedades de acordo com os privilégios (leitura/escrita) do utilizador.

### 2.3.6 Implementações WBEM

Cada empresa, consórcio ou grupo, desenvolve a sua própria implementação WBEM de acordo com as especificações padrão estabelecidas, garantindo também que todos os CIMOMs associados estejam acessíveis via operações HTTP padrão.

Existem no mercado diversas plataformas e produtos que implementam WBEM. As primeiras constituem plataformas de desenvolvimento disponíveis para quem queira implementar soluções WBEM e os segundos dizem respeito às soluções específicas de gestão desenvolvidas.

A seguir apresentam-se algumas das principais empresas e respectivas soluções empresariais que implementam WBEM.

- IBM –Tivoli Management Framework (TMF): plataforma de gestão de sistemas IBM (anteriormente Tivoli Systems, Inc. - empresa adquirida pela IBM em 1996 e transferida para a divisão de Software do grupo IBM). Baseia-se em CORBA (Common Object Request Broker Architecture) [30], uma arquitectura que permite à plataforma de gestão gerir um grande número de locais ou dispositivos remotos. Baseado nesta plataforma foram desenvolvidos uma quantidade considerável de implementações comerciais do WBEM [31].

- Cisco – CiscoWorks LAN Management Solution (LMS): solução industrial padrão da Cisco Systems, Inc. para gestão de redes. Ao utilizar o modelo CIM e XML, a Cisco oferece uma forma de partilhar dados com outros produtos de gestão de redes mais usados e populares [32].
- Sun – Solaris WBEM Services e Sun WBEM SDK (Software Development Kit) – constituem a implementação do padrão WBEM da Sun Microsystems, Inc. para o ambiente Solaris. Com estas duas ferramentas, programadores e gestores podem criar e modificar informação de sistema armazenada no formato CIM [33].
- HP (Hewlett-Packard) – as soluções WBEM disponibilizadas pela HP incluem WBEM Services, WBEM Providers, WBEM Clients e WBEM SDK [34].

Um dos produtos WBEM da HP mais conhecidos é o HP Open View – uma ampla gama de produtos de gestão de redes e sistemas. A designação do HP Open View foi alterada em 2007 passando a fazer parte de uma designação mais abrangente - HP Software.

- Microsoft – disponibiliza WBEM através da plataforma Windows Management Instrumentation (WMI). O WMI é a infra-estrutura de gestão de dados e operações para sistemas operativos baseados em Windows. Fornece a possibilidade de escrever scripts WMI ou desenvolver aplicações para automatizar tarefas administrativas em computadores remotos. Também fornece dados de gestão para outras partes do sistema operativo e para aplicações como System Center Operations Manager, Microsoft Operations Manager (MOM), ou Windows Remote Management (WinRM) [35].

O WMI inclui um repositório de dados compatível com CIM e o gestor de objectos CIMOM. No entanto, a comunicação entre a aplicação gestora e CIMOM e entre Providers e CIMOM realiza-se segundo uma arquitectura de objectos distribuídos da Microsoft chamada DCOM (Distributed Component Object Model). Como tal, não satisfaz o padrão WBEM no sentido de usar browsers para comunicar entre Agentes e Gestores através de operações CIM sobre HTTP.

MOM – fornece monitorização operacional de ambientes servidor que suportam WMI [36, 37].

Para além dos produtos comerciais, existem actualmente várias implementações WBEM em código aberto (open-source) com alguma representatividade. Tais implementações apresentam algumas diferenças entre si no que toca às linguagens de implementação utilizadas e sistemas operativos a que se destinam. Algumas, mais populares e conhecidas, que suportam operações CIM sobre HTTP, são descritas a seguir.

- Pegasus – desenvolvido pelo The Open Group, é uma implementação open-source do modelo CIM e dos padrões WBEM da DMTF. Foi projectado para ser portátil e altamente modular. Sendo codificado em C++, traduz efectivamente os conceitos dos objectos CIM para um modelo de programação, mantendo a velocidade e eficiência de uma linguagem compilada. Devido à sua portabilidade pode ser executado na maior parte das versões de UNIX(R), Linux, OpenVMS, e Microsoft Windows [38].
- WBEM Services – este projecto desenvolvido pela Sun Microsystems, Inc. constitui um esforço para desenvolver uma implementação WBEM open-source em Java, adequada para aplicações comerciais (como o caso do Solaris) e não-comerciais. Consiste num conjunto de APIs, aplicações cliente e servidor e algumas ferramentas.

As APIs são baseadas no Java Specification Request (JSR) submetidas para aprovação pelo Java Community Process (JCP) 2.0 [39].

- OpenWBEM – implementação WBEM open-source de grau empresarial desenvolvida em C++, adequada para aplicações comerciais e não-comerciais. Fornece ferramentas para desenvolvimento de estruturas de gestão independentes da plataforma em causa, ajudando a ultrapassar barreiras relacionadas com o tipo de implementação fornecendo assim verdadeira interoperabilidade. Os programadores podem usar OpenWBEM como Agente de gestão e a plataforma WBEM para fornecer aplicações para configuração e mudança de gestão, monitorização da saúde do sistema e funcionalidades de gestão empresarial alargadas [40].
- SBLIM - Standards Based Linux Instrumentation for Manageability – projecto open-source destinado a melhorar a gestão de sistemas GNU/Linux. O objectivo deste projecto é fornecer uma implementação open-source, completa, de uma solução de gestão para Linux baseada em WBEM. A DMTF definiu um conjunto de normas e a WBEMsource ajudou a fomentar essa iniciativa. SBLIM foi licenciado sobre Common Public License (CPL) [41].

Com o objectivo de construir uma comunidade activa em volta do SBLIM [42], a IBM lançou em 2001 um projecto próprio baseado num sistema de colaboração modificado da SourceForge debaixo de uma licença CPL aprovada pela Open Source Initiative [43].

## 2.4 Gestão baseada em Web Services (WS)

A designação Web Services (WS) surgiu no ano de 1999 quando um consórcio de empresas liderado pela Microsoft e IBM concordaram em apoiar um conjunto de padrões que definem a forma de sistemas diferentes comunicarem entre si. Hoje, os Web Services são um padrão do W3C (World Wide Web Consortium) [44] – um consórcio destinado a desenvolver tecnologias de domínio público para a World Wide Web.

O principal objectivo do WS é proporcionar a interoperabilidade entre sistemas distribuídos, independente da plataforma e da linguagem de programação utilizada, disponibilizando uma melhor interligação entre aplicações. A ideia é que uma aplicação possa usar operações que não estejam implementadas no sistema em que está a correr.

A necessidade de integrar aplicações que se encontram em ambientes heterogéneos distintos, bem como possibilitar a comunicação entre as mesmas em tempo real, fez com que a computação distribuída viesse a assumir um papel cada vez mais importante nas comunicações actuais.

Os WS são uma tecnologia orientada para Internet com grandes analogias ao CORBA [45]. Constituem uma das tecnologias emergentes para desenvolvimento de aplicações distribuídas para a Internet que permite a comunicação entre aplicações diferentes e a integração de sistemas distintos de forma simples e directa (mesmo que desenvolvidos em linguagens e plataformas diferentes) [46]. Esta integração é possível uma vez que a linguagem usada na comunicação entre as diversas aplicações é o XML – linguagem genérica, aberta e universal. Estas características facilitam o desenvolvimento de aplicações através de WS, tornando esta tecnologia muito atractiva para quem

desenvolve software, sendo por isso, utilizada em inúmeras áreas incluindo, a gestão de redes.

No campo da gestão, os WS fornecem uma forma-padrão de definir e aceder a informação de gestão facilitando o desenvolvimento de aplicações de gestão. Além disso, o tráfego WS sobre HTTP atravessa as firewalls de numa rede (geralmente configuradas para impedir outros protocolos que não HTTP/HTTPS) mais facilmente que outros sistemas de gestão, como o SNMP [47].

A estrutura do WS baseia-se em conteúdos Web e serviços associados que depois ficam acessíveis a aplicações distribuídas. Uma plataforma WS divide-se em três áreas distintas com especificações para cada [46]. Apresentam-se a seguir cada uma destas áreas, bem como a especificação corrente mais relevante e estável.

- Protocolos de Comunicação – SOAP (Simple Object Access Protocol) [48] – permite a comunicação através dos vários WS.
- Descrição de Serviços – WSDL (Web Services Description Language) [49] – fornece uma descrição formal dos WS por forma a ser entendida pelos computadores onde correm os serviços.
- Serviços de Descoberta – UDDI (Universal Description, Discovery and Integration) [50] – um registo das descrições dos WS.

Sendo uma tecnologia emergente, investigadores continuam a desenvolver peças importantes como, QoS (Quality of Service) e modelos de descrição e interação, tirando partido das especificações disponíveis e incorporando mais módulos à medida que aumenta a maturação desta tecnologia [46].

## 2.4.1 Arquitectura WS

A arquitectura WS é composta por três entidades distintas:

- Service Provider – fornecedor (provedor) de serviços - entidade que disponibiliza serviços na Internet;
- Service Client/Requester – cliente (requisitante) de serviços - entidade que procura (requisita) e acede a serviços disponíveis na Internet através de fornecedores de serviços;
- Service Broker – intermediário de serviços - entidade que regista os serviços prestados por vários fornecedores e que disponibiliza um serviço de pesquisa de serviços para os clientes.

Na arquitectura WS agentes de software interagem entre si através da troca de mensagens entre serviços requisitados e fornecedores de serviços. Os requisitantes (clientes) são agentes de software que requisitam a execução do serviço e os fornecedores são agentes de software que fornecem o serviço. Assim, os Agentes podem ser requisitantes e fornecedores. Os fornecedores de serviços são responsáveis por publicar a descrição do serviço que disponibilizam. Os requisitantes devem ter a capacidade de encontrar as descrições dos serviços.

Os clientes (requisitantes) WS podem ser, browsers Web, aplicações finais de utilizador ou até outros serviços WS. Neste caso, um serviço WS pode requisitar operações de outro serviço WS permitindo deste modo uma hierarquia nos pedidos WS. Além disso,

permite a criação de serviços complexos com base na invocação de serviços mais simples [47].

A Figura 2.14 apresenta um diagrama de blocos com os principais elementos da arquitetura WS, bem como os protocolos envolvidos na comunicação entre os mesmos [46, 47, 51]. Num cenário típico um fornecedor de serviço (Provider) define a descrição do serviço WS e publica isso para um requisitante (Client/Requester) ou para um Agente de publicação (Broker). Por sua vez, o cliente/requisitante do serviço, usa uma função de procura para descobrir a localização do mesmo localmente ou para descobrir uma agência de publicação (p. ex.: um registo ou repositório). A descrição do serviço é usada pelo cliente/requisitante para ligar o serviço com o fornecedor e invocar ou interagir com a sua implementação. O serviço WS deve exibir características do fornecedor de serviço e da função requisitante.

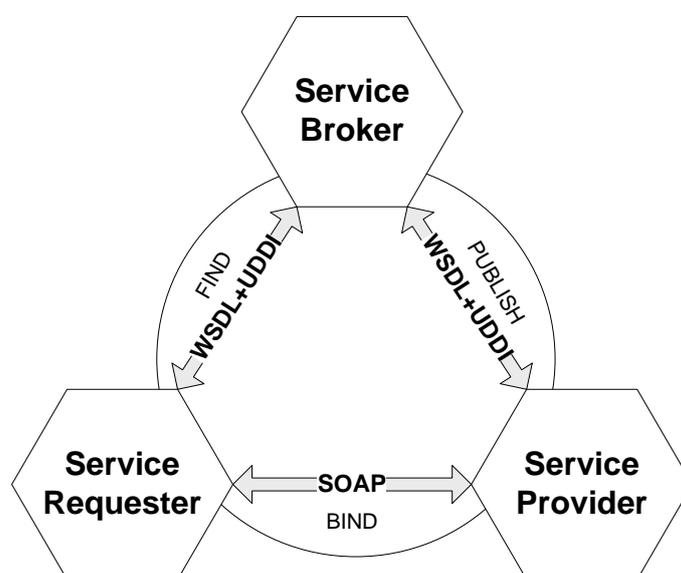
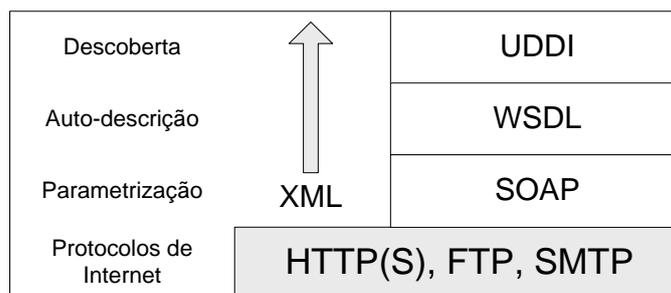


Figura 2.14 – Arquitectura WS

## 2.4.2 Simple Object Access Protocol (SOAP)

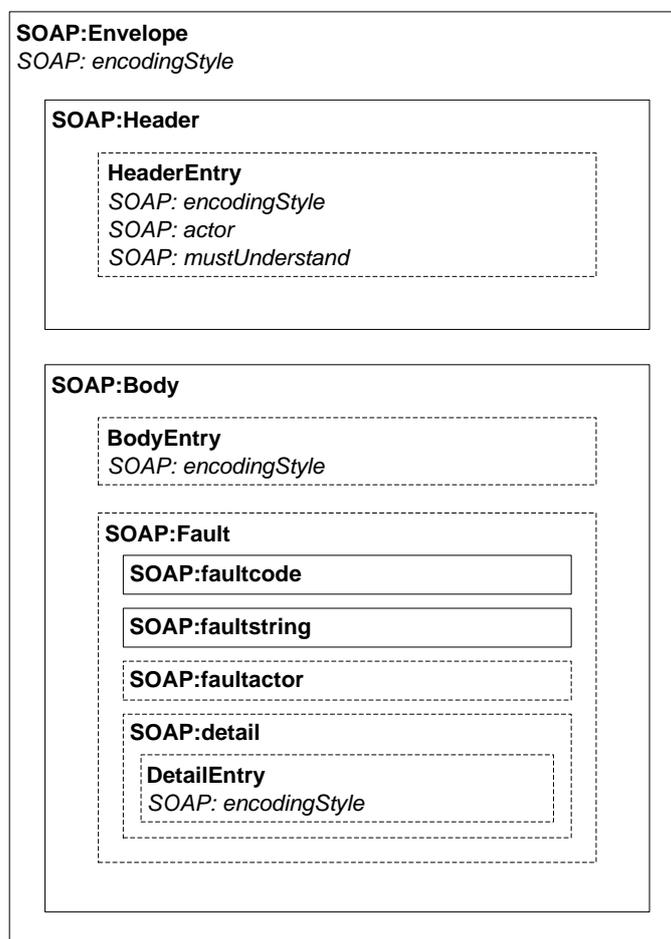
O SOAP (Simple Object Access Protocol) [48] é uma norma criada pela Microsoft e depois desenvolvida em colaboração com a Developer, IBM, Lotus e UserLand, que permite a troca numa rede de computadores de mensagens baseadas em XML, usando protocolos como HTTP, SMTP (Simple Mail Transfer Protocol) e MQSeries [46]. Estão assim garantidos alguns pressupostos necessários para mecanismos de comunicação para Web, como sejam, ser leve e independente da plataforma e linguagem. O SOAP pretende ainda ser simples e extensível.

A Figura 2.15 apresenta a pilha de protocolos do WS. É possível verificar que o SOAP está na base da pilha de protocolos, constituindo uma ferramenta de comunicações que outras aplicações podem utilizar. As questões de segurança não foram excluídas uma vez que o SOAP pode correr por cima de serviços de transporte seguros como o HTTPS, SMTP codificado e FTP (File Transfer Protocol) seguro [47].



**Figura 2.15 – Pilha de Protocolos WS**

Uma mensagem SOAP é um documento XML com uma estrutura simples. A Figura 2.16 destaca os seus elementos principais que são: um elemento Envelope e dois elementos child, o Header (opcional) e o Body [48].



**Figura 2.16 - Elementos principais de uma mensagem SOAP**

O elemento Envelope identifica o documento XML como um SOAP e é o elemento mais importante neste tipo de mensagens.

O elemento Header (cabeçalho) é opcional e contém informação que afecta o modo como certo conteúdo específico da aplicação é processado na mensagem SOAP. Um

exemplo disso é o caso de informação de autenticação que pode ser especificada como conteúdo de um objecto *SOAP:Header*. No caso de o elemento Header estar presente, deverá ser o primeiro elemento child do Envelope.

O elemento Body (corpo) contém a mensagem SOAP a ser enviada. Dentro do corpo, existe ainda o elemento child Fault (opcional) que fornece informações sobre erros que ocorrem quando a mensagem é processada. O elemento Fault que tem a particularidade de aparecer apenas uma única vez numa mensagem SOAP, apresenta os seguintes sub-elementos: *faultcode* (código que identifica o erro), *faultstring* (comentário explicativo do erro), *faultactor* (informação sobre a origem do erro, 'quem' causou) e *detail* (guarda a informação de erro da aplicação específica relacionada com o elemento Body).

Uma mensagem SOAP, cuja estrutura-tipo é apresentada na Figura 2.17 inclui ainda alguns atributos que importa referir: *encodingStyle*, *actor* e *mustUnderstand*. O *encodingStyle* é usado para definir tipos de dados usados no documento. Este atributo deve aparecer em qualquer elemento SOAP e irá aplicar o conteúdo desse elemento a todos os respectivos elementos child. O atributo *actor* indica quem deve processar a mensagem. Uma mensagem consegue identificar *actors* que indicam uma série de intermediários que processam as respectivas partes da mensagem a que se destinam transmitindo as restantes. O atributo *mustUnderstand* pode ser usado para indicar se um cabeçalho de entrada é obrigatório ou opcional. Se um elemento child contiver a instrução *soap:mustUnderstand="1"*, caso o receptor da mensagem não identifique o elemento em causa, este deverá falhar quando o cabeçalho for processado.

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP:Header>
    <!-- Conteúdo do Header -->
  </SOAP:Header>
  <SOAP:Body>
    <!-- Conteúdo do Body -->
  </SOAP:Body>
</SOAP:Envelope>
```

**Figura 2.17 – Estrutura de uma mensagem SOAP**

Existem diferentes tipos de mensagens SOAP, sendo o mais utilizado o RPC em que o cliente pode invocar uma mensagem num servidor, e o servidor responde de imediato para o cliente com uma outra mensagem. Para utilizar SOAP com RPCs é necessário, para além de definir um protocolo RPC, os aspectos seguintes [46]:

- a forma como valores inseridos podem ser transportados entre o formato SOAP (em XML) e o formato da aplicação (como p. ex. uma classe em Java);
- onde é que são transportadas as várias partes da RPC (identificador do objecto, nome da operação e parâmetros).

A especificação XSD do W3C [26] fornece uma linguagem padrão para definir a estrutura do documento e os tipos de dados das estruturas XML. Assim, para permitir a transmissão dos valores e os respectivos tipos de dados, o SOAP assume um tipo

baseado na sua codificação em XML. Isto é feito através do atributo *encodingStyle* que produz uma codificação XML para cada tipo de estrutura de dados. Os argumentos e as respostas RPC também são representados usando esta codificação.

Existem implementações SOAP de RPCs em várias linguagens de programação, incluindo C, Java e Perl, as quais geram e processam automaticamente as mensagens SOAP. Caso as mensagens estejam conforme as especificações SOAP, podem depois ser trocadas por serviços implementados em diferentes linguagens.

A Figura 2.18 exemplifica a situação de uma chamada SOAP RPC e a respectiva resposta para uma situação-exemplo que poderia ser usada por companhias aéreas [46]. Pretende-se fornecer ao utente informação sobre o estado do seu voo. Na mensagem SOAP é enviada uma string com o nome da companhia aérea e um integer com o número do voo. A resposta é dada com um valor estruturado contendo sub-valores para o número da porta do aeroporto e o estado do voo.

(a) Chamada SOAP RPC	(b) Resposta SOAP RPC
<pre> POST /travelservice SOAPAction: "http://www.acme-travel.com/flightinfo" Content-Type: text/xml; charset="utf-8" Content-Length: nnnn  &lt;SOAP:Envelope xmlns:SOAP= "http://schemas.xmlsoap.org/soap/envelope/"&gt; &lt;SOAP:Body&gt; &lt;m:GetFlightInfo xmlns:m="http://www.acme-travel.com/flightinfo" SOAP:encodingStyle= "http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance"&gt; &lt;airlineName xsi:type="xsd:string"&gt;UL &lt;/airlineName&gt; &lt;flightNumber xsi:type="xsd:int"&gt;506 &lt;/flightNumber&gt; &lt;/m:GetFlightInfo&gt; &lt;/SOAP:Body&gt; &lt;/SOAP:Envelope&gt; </pre>	<pre> HTTP/1.1 200 OK Content-Type: text/xml; charset="utf-8" Content-Length: nnnn  &lt;SOAP:Envelope xmlns:SOAP= "http://schemas.xmlsoap.org/soap/envelope/"&gt; &lt;SOAP:Body&gt; &lt;m:GetFlightInfoResponse xmlns:m="http://www.acme-travel.com/flightinfo" SOAP:encodingStyle= "http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance"&gt; &lt;flightInfo&gt; &lt;gate xsi:type="xsd:int"&gt;10&lt;/gate&gt; &lt;status xsi:type="xsd:string"&gt;ON TIME&lt;/status&gt; &lt;/flightInfo&gt; &lt;/m:GetFlightInfoResponse&gt; &lt;/SOAP:Body&gt; &lt;/SOAP:Envelope&gt; </pre>

**Figura 2.18 – Exemplo de uma chamada (a) e a respectiva resposta (b) SOAP RPC**

### 2.4.3 Web Services Description Language (WSDL)

Embora o SOAP forneça comunicação básica para WS independente da plataforma e linguagem de programação usadas, não nos informa que mensagens devem ser trocadas para interagir com sucesso com um serviço. Esse papel é preenchido pela linguagem WSDL. Um formato XML desenvolvido pela IBM e Microsoft [46] para descrever WS como colecções de pontos finais de comunicação (endpoints) que podem trocar certas mensagens. Pode-se dizer que um documento WSDL descreve a interface WS e fornece utilizadores com um ponto de contacto.

WSDL fornece uma descrição da interacção entre clientes e serviços a dois níveis:

- abstracto – descrição do serviço ao nível da aplicação (interface abstracta);
- específico – detalhes de ligação dependentes do protocolo que os utilizadores devem seguir para aceder a serviços em pontos finais de comunicação de serviços concretos.

Esta separação justifica-se pelo facto de que, funcionalidades semelhantes de serviços ao nível da aplicação, são muitas vezes utilizadas em pontos de ligação distintos com ligeiras diferenças nos detalhes dos protocolos de acesso. A separação das descrições destes dois aspectos ajuda o WSDL a representar funcionalidades comuns entre pontos ligeiramente diferentes.

#### Descrição abstracta

WSDL define a descrição abstracta do serviço (ao nível da aplicação) em termos de mensagens trocadas na interacção entre serviços. Existem três componentes principais para a interface abstracta: vocabulário, mensagem e interacção.

No campo do vocabulário, o WSDL utiliza sistemas de tipos externos para fornecer definições de tipos de dados para troca de informações. Apesar do WSDL poder suportar qualquer tipo de sistema, a maioria dos serviços usa XSD. No exemplo de descrição abstracta WSDL apresentado na parte (a) da Figura 2.19, pudemos observar dois tipos de dados definidos em XSD (string e int) e dois tipos de dados definidos num schema externo (*FlightInfoType* e *Ticket*) [46]. Definições XSD externas podem ser importadas usando um elemento próprio para esse efeito que especifica a localização das mesmas.

As mensagens WSDL são descritas por tipos XSD ou elementos de um vocabulário pré-definido e fornecem dados enviados e recebidos entre serviços. No exemplo de descrição abstracta apresentado, podemos observar os elementos *operation* e *portType* que combinam mensagens para definir interacções.

Cada operação (descrita no elemento *operation*) representa uma troca de mensagens padrão suportada pelo serviço WS dando aos utilizadores acesso a determinada funcionalidade básica do serviço. O elemento *operation* é uma combinação simples de mensagens *input*, *output* ou *fault*. Quanto ao *portType*, representa uma colecção de operações que são suportadas colectivamente por um endereço de rede.

#### Informação específica de ligação

Além dos elementos envolvidos na descrição das funcionalidades do serviço ao nível da aplicação, é necessário encontrar respostas às questões que se seguem.

- Qual o protocolo de comunicação a usar (tal como SOAP sobre HTTP)?
- Como realizar interações de serviços individuais sobre esse protocolo?
- Onde termina a comunicação?

Os elementos *binding* WSDL fornecem a resposta às duas primeiras questões incluindo o protocolo de comunicação e o formato de dados específico para o *portType* completo. Em suma, explicam como uma determinada interação ocorre sobre um protocolo específico.

(a) Descrição abstracta	(b) Informação específica de ligação
<pre> &lt;message name="GetFlightInfoInput"&gt;   &lt;part name="airlineName" type="xsd:string"/&gt;   &lt;part name="flightNumber" type="xsd:int"/&gt; &lt;/message&gt; &lt;message name="GetFlightInfoOutput"&gt;   &lt;part name="flightInfo" type="fixsd:FlightInfoType"/&gt; &lt;/message&gt; &lt;message name="CheckInInput"&gt;   &lt;part name="body" element="eticketxsd:Ticket"/&gt; &lt;/message&gt; &lt;portType name="AirportServicePortType"&gt;   &lt;operation name="GetFlightInfo"&gt;     &lt;input message="tns:GetFlightInfoInput"/&gt;     &lt;output message="tns:GetFlightInfoOutput"/&gt;   &lt;/operation&gt;   &lt;operation name="CheckIn"&gt;     &lt;input message="tns:CheckInInput"/&gt;   &lt;/operation&gt; &lt;/portType&gt; </pre>	<pre> &lt;binding name="AirportServiceSoapBinding"   type="tns:AirportServicePortType"&gt;   &lt;soap:binding transport=     "http://schemas.xmlsoap.org/soap/http"/&gt;   &lt;operation name="GetFlightInfo"&gt;     &lt;soap:operation style="rpc"       soapAction="http://acme-travel/flightinfo"/&gt;     &lt;input&gt;       &lt;soap:body use="encoded"         namespace="http://acme-travel.com/flightinfo"         encodingStyle=           "http://schemas.xmlsoap.org/soap/encoding"/&gt;     &lt;/input&gt;     &lt;output&gt;       &lt;soap:body use="encoded"         namespace="http://acme-travel.com/flightinfo"         encodingStyle=           "http://schemas.xmlsoap.org/soap/encoding"/&gt;     &lt;/output&gt;   &lt;/operation&gt;   &lt;operation name="CheckIn"&gt;     &lt;soap:operation style="document"       soapAction="http://acme-travel.com/checkin"/&gt;     &lt;input&gt;       &lt;soap:body use="literal"/&gt;     &lt;/input&gt;   &lt;/operation&gt; &lt;/binding&gt; &lt;service name="travelservice"&gt;   &lt;port name="travelservicePort"     binding="tns:AirportServiceSoapBinding"&gt;     &lt;soap:address location=       "http://acmetravel.com/travelservice"/&gt;   &lt;/port&gt; &lt;/service&gt; </pre>

Figura 2.19 – Exemplos de descrição WSDL abstracta (a) e específica (b)

A parte (b) da Figura 2.19 exemplifica como o elemento *binding* descreve a forma para aceder a um determinado serviço através de SOAP. Ali podemos observar duas operações distintas que utilizam estilos diferentes de interacção. *GetFlightInfo* usa um estilo de interacção baseado em SOAP-RPC, no qual, todas as mensagens trocadas utilizam codificação (*encoding*) SOAP padrão. *CheckIn* usa um estilo de interacção baseado em mensagens cujo corpo (*body*) não contém tipos de codificações adicionais e utiliza XSD para descrever literalmente XML transmitido.

Para ajudar a responder à terceira questão, onde aceder a esta combinação de interface abstracta e detalhes de protocolos e vínculos, temos dois elementos WSDL:

- *port* – descreve um ponto de comunicação como uma combinação de *binding* e endereço de rede;
- *service* – agrupa um conjunto de *ports* relacionados.

No exemplo da parte (b) da figura Figura 2.19 um único *port* descreve um ponto final de comunicação que processa pedidos SOAP para o serviço *travelservice*.

### Como é utilizado o WSDL

Para utilizadores e programadores, a WSDL fornece uma descrição formal de interacção cliente-serviço. Durante o processo de desenvolvimento, programadores usam WSDL como entrada dum gerador que produz código cliente de acordo com os requisitos do serviço. Além disso, WSDL também pode ser usado como entrada para uma invocação dinâmica, a qual pode gerar pedidos de serviço correctos em tempo real. Em ambas as situações, utilizador e programador não precisam se preocupar com os detalhes no acesso ao serviço, basta obter a descrição WSDL e usá-la como entrada da infraestrutura de desenvolvimento e execução para trocar o tipo correcto de mensagens SOAP com o serviço em causa.

### 2.4.4 Universal Description, Discovery and Integration (UDDI)

O UDDI surgiu em 1999/2000 com o objectivo de proporcionar acesso a documentos WSDL através de mensagens SOAP. Era intenção dos seus autores que consumidores de serviços WS estivessem ligados a fornecedores através de um sistema de negociação dinâmico. Pretendia-se um mecanismo único e aberto para descrição e localização de negócios e seus serviços.

Segundo o OASIS UDDI Technical Comittee [52], as especificações UDDI foram desenvolvidas com o objectivo de formar fundamentações técnicas necessárias para a publicação e pesquisa de implementações WS, quer entre as empresas, quer dentro destas.

#### Especificações dos registos UDDI

O UDDI utiliza XML para descrever negócios, seus serviços e os detalhes de acesso a cada serviço. O modelo de informações de um registo UDDI é padronizado através do XML Schema.

As especificações UDDI oferecem aos utilizadores um meio unificado e sistemático de encontrar fornecedores de serviços através de um registo centralizado. Para especificar a descrição, publicação e pesquisa de um WS é necessário especificar:

- a estrutura de dados a ser utilizada na representação de negócios e seus serviços (definir a informação a fornecer acerca de cada serviço e a forma de o codificar);
- a API de consulta e actualização do registo (que descreve como informação do serviço pode ser acedida e actualizada).

O XML Schema subjacente aos registos UDDI é bastante simples, de facto, as especificações UDDI seguem um modelo semelhante ao de uma lista telefónica on-line de WS que agrupa a informação em [46, 53]:

- páginas brancas – inclui nomes e detalhes de contacto;
- páginas amarelas – fornece uma categorização baseada em negócios e tipos de serviços;
- páginas verdes – inclui dados técnicos acerca dos serviços.

O registo UDDI está organizado em torno de duas entidades fundamentais, *businessEntity* e *businessService* responsáveis pela descrição, dos negócios e respectivos serviços. Para além destes existe um terceiro elemento o *bindingTemplate*. O papel de cada um é descrito a seguir.

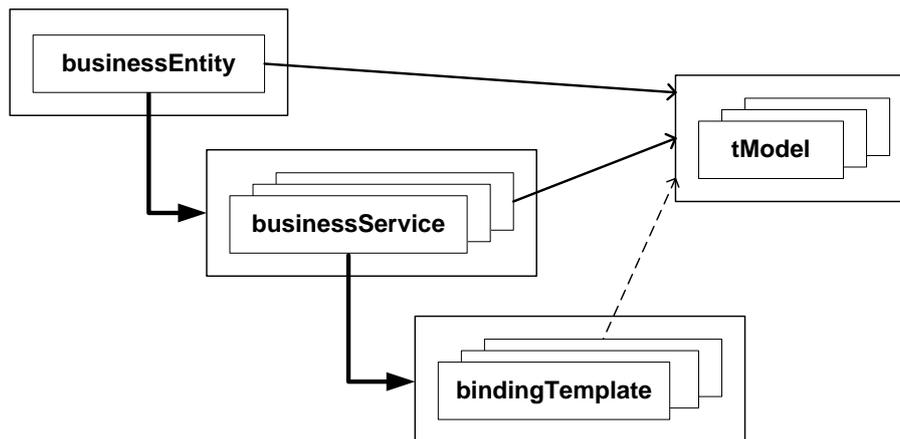
- *businessEntity* – fornece informação padrão referente às páginas brancas; estão incluídas informações básicas sobre o negócio como o nome, descrição, informações de contacto e identificadores; cada entidade de negócio pode conter um ou vários serviços (*businessService*);
- *businessService* – fornece informações acerca de um serviço como o nome, descrição e categoria; cada serviço contém uma lista de modelos de ligação (*bindingTemplate*) que fornecem informações técnicas de acesso ao serviço;
- *bindingTemplate* – representa um ponto de acesso ao serviço; cada serviço pode ser provido por diferentes pontos de acesso, cada um dos quais com características técnicas diferentes.

Os elementos *businessService* e *bindingTemplate* definem as informações referentes às páginas verdes.

A Figura 2.20 apresenta a estrutura de dados do UDDI. É possível verificar que cada um dos três elementos descritos atrás organizados hierarquicamente, *businessEntity*, *businessService* e *bindingTemplate*, podem referenciar um quarto elemento designado por *tModel*.

- *tModel* – é um modelo técnico que contém endereços e metadata de documentos técnicos a usar por quem pretenda desenvolver WS; a informação fornecida por tais documentos, permite encontrar compatibilidade entre fornecedores e consumidores de WS através de referências a namespaces-chave.

O *tModel* é referenciado pelos elementos *businessEntity* e *businessService* para classificação dos negócios e serviços por eles definidos, fornecendo assim informações referentes às páginas amarelas. Quando é referenciado por um modelo de ligação (*bindingTemplate*) é com o propósito de acrescentar informações técnicas de acesso ao serviço. Por exemplo, o documento WSDL dum WS pode ser registado como um modelo técnico que será referenciado no modelo de ligação do serviço oferecido por esse WS [46]. Um único modelo técnico pode ser referenciado por vários modelos de ligação.



**Figura 2.20 – Modelo de Informação do registro UDDI**

### **Servidor de registos UDDI (UDDI Register)**

Um servidor de registos UDDI é constituído por implementações de repositórios de informações compatíveis com as especificações UDDI. Tais repositórios podem ser considerados como WS que fornecem serviços de publicação e pesquisa de outros WS.

Um servidor de registos UDDI oferece, normalmente, dois modos de acesso:

- via interface Web disponibilizada por um browser – constitui um método simples para os fornecedores de serviços publicarem as suas informações de negócios e serviços e os requisitantes acederem a essas informações;
- via uma API UDDI – permite que fornecedores e requisitantes efectuem operações de publicação e pesquisa de negócios e serviços de forma dinâmica, em tempo real, sendo que, a comunicação com o servidor de registos é feita através de mensagens SOAP.

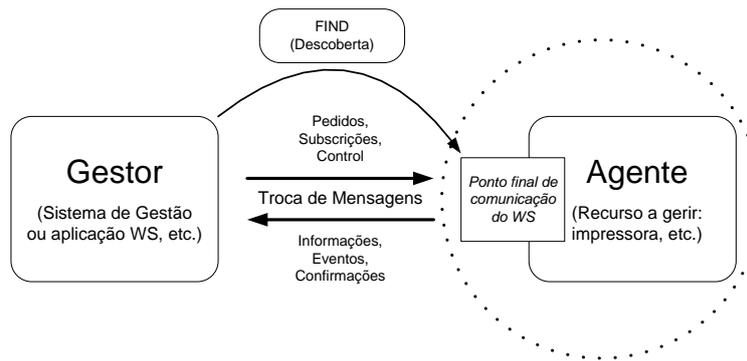
### **2.4.5 Normas de gestão baseada em WS**

Os WS são uma arquitectura de processamento distribuído baseada em XML que cumpre os requisitos de uma arquitectura orientada a serviços - SOA (Service Oriented Architecture). Uma aplicação de gestão de redes pode ser vista como mais um serviço disponibilizado pela rede, podendo ser implementada e disponibilizada como um WS.

É possível gerir recursos de rede através de WS, quer local, quer remotamente, usando mensagens que obedecem a interfaces bem definidas. O WSDL permite criar interfaces bem flexíveis, abrangendo operações simples como ler uma variável, até operações mais complexas necessárias à configuração de redes ou gestão de serviços e negócios.

Num sistema de gestão baseado em WS, o ponto final de comunicação do WS fornece o acesso ao recurso a gerir (Agente). O Gestor descobre esse ponto e troca mensagens com ele. É possível que o Gestor possa reter informação de gestão sobre o recurso ou afectar o seu estado alterando essa informação. Por sua vez, o Agente pode informar ou notificar o Gestor da ocorrência de eventos significativos.

A Figura 2.21 ilustra os papéis dos intervenientes no modelo de gestão baseado em WS, Gestor e recurso a gerir (Agente).

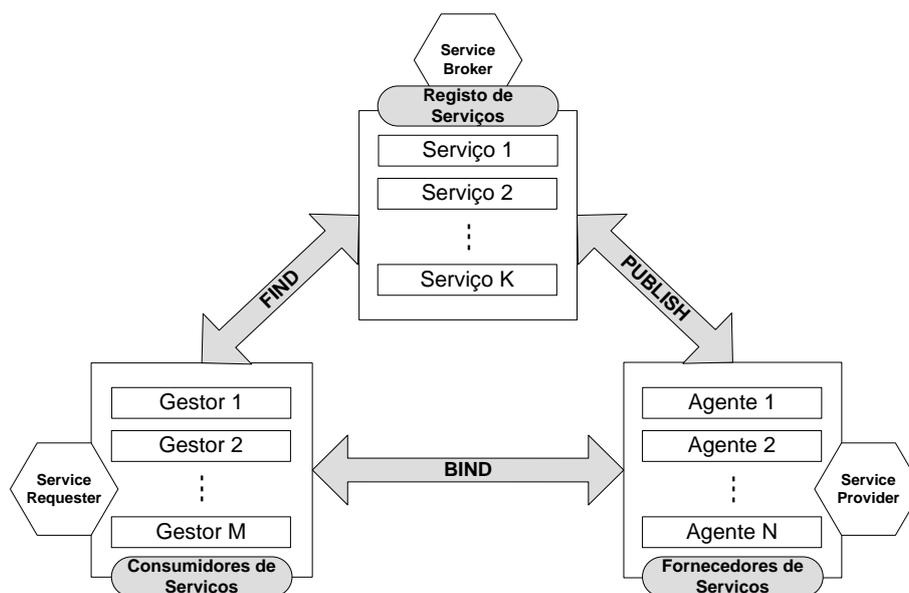


**Figura 2.21 – Conceito de gestão baseada em WS**

O Agente de software é implementado através de um fornecedor de serviços que oferece operações de gestão sobre o elemento de rede específico que representa. No registo central de serviços, o UDDI, estão publicadas as informações de acesso ao serviço dos diversos Agentes, bem como os detalhes das interfaces de gestão.

O Gestor actua como um requisitante de serviços que identifica no registo os fornecedores de serviços de gestão associados aos elementos de rede que pretende gerir. Para além de obter o endereço de acesso ao recurso a gerir (ponto final de comunicação do WS) obtém ainda quais as operações suportadas pelo Agente em causa. Assim o Gestor obtém todos os detalhes necessários à comunicação com o Agente específico.

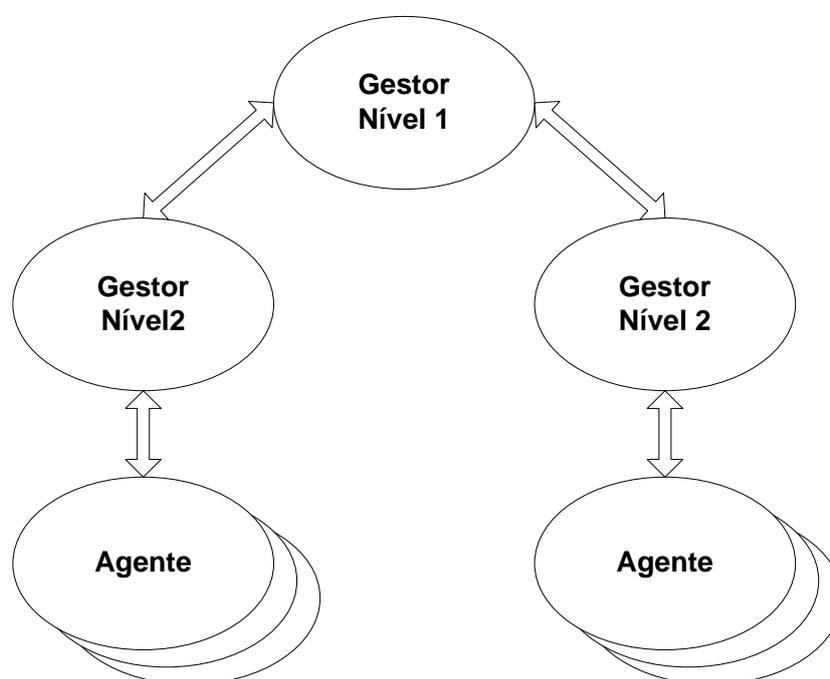
Na Figura 2.22 podemos verificar que um mesmo serviço de gestão pode ter múltiplas implementações. É ilustrada uma situação com K serviços publicados no registo central, N Agentes fornecendo serviços de gestão e M Gestores pedindo serviços de gestão. Numa situação extremo, pudemos ter todos os Agentes a suportarem operações comuns e padronizadas existindo uma única descrição de serviço apontando os diversos Agentes que a implementam.



**Figura 2.22 – Gestão de redes baseada na arquitectura WS**

Os diversos Agentes representados na Figura 2.22, podem estar organizados hierarquicamente e distribuídos pelos diferentes domínios de gestão de uma organização (através de critérios geográficos, área de gestão – redes, sistemas, serviços, etc. – centro de custo da empresa, clientes, etc.). Na Figura 2.23, é apresentado um exemplo de um sistema de gestão de uma organização estruturado a três níveis onde um Gestor de alto nível coordena dois Gestores de nível intermédio ligados a domínios distintos sob os quais se encontram vários Agentes.

Os WS também podem suportar comunicação entre Gestores organizados hierarquicamente. O facto dos WS permitirem definição de interfaces complexas e flexíveis potencia a sua aplicação na comunicação entre Gestores que passam, também eles, a implementar fornecedores de serviços de gestão.



**Figura 2.23 – Gestão distribuída e hierárquica**

Actualmente, existem dois padrões (especificações) na área da gestão de redes baseada em serviços através de WS [54]:

- Web Services Distributed Management (WSDM) – desenvolvido em Março de 2005 pelo OASIS (Organization for the Advancement of Structured Information Standards) [55-57];
- Web Services for Management (WS-Management) – desenvolvido em Maio de 2006 pelo DMTF [3].

O facto destas especificações terem vindo a ser apoiados por importantes intervenientes na área da gestão de redes baseada em serviços, leva a crer que ambos têm potencial para se tornarem, de facto, padrões nesta área [54].

## Web Services Distributed Management (WSDM)

O WSDM é uma especificação desenvolvida e padronizada pelo OASIS Web Services Distributed Management Comittee e está organizada segundo duas especificações:

- **MUWS** (Management Using Web Services) [55, 56] – define a forma de representar e aceder às interfaces de gestão de recursos distribuídos através de WS;
- **MOWS** (Management Of Web Services) [57] – define a forma de gerir WS como um recurso e define a forma de descrever e aceder a essa capacidade de gestão usando MUWS (fornece mecanismos e metodologias que permite que, aplicações a gerir através de WS, possam interagir entre si dentro dos limites da empresa ou organização).

A especificação MUWS fornece funcionalidades de gestão comparáveis ao SNMP e WS-Management [54]. O objectivo do MUWS é tornar um recurso arbitrário a gerir via WS. Para definir a forma de o conseguir, o MUWS assenta em várias especificações WS, tais como: XML Schema, WSDL, WS-Addressing, WS-Resource, Web Services Notification (WS-BaseNotification) e WSSecurity.

## Web Services for Management (WS-Management)

O WS-Management é uma especificação desenvolvida por um consórcio formado pela Sun, Microsoft, NEC, Intel, Dell e outras companhias. Esta especificação foi submetida ao DMTF em Setembro de 2005, tendo sido padronizada em Abril de 2006 [3].

Para promover a interoperabilidade entre aplicações e gestão de recursos geridos, esta especificação identifica um conjunto básico de especificações WS e requisitos de utilização que expõem um conjunto comum de operações central a todos os sistemas de gestão. Inclui a capacidade para fazer o seguinte [58]:

- descobrir (*Discover*) a presença de recursos de gestão e navegar entre eles;
- obter (*Get*), colocar/manipular (*Put*), criar (*Create*), e apagar (*Delete*) recursos de gestão individuais, tais como configurações e valores dinâmicos;
- enumerar (*Enumerate*) os conteúdo dos contentores e das colecções, tais como grandes tabelas e registos de eventos (logs);
- subscrever (*Subscribe*) eventos emitidos por recursos a gerir;
- executar (*Execute*) métodos de gestão específicos com parâmetros de entrada/saída bem definidos.

A Tabela 2.5 enumera algumas das operações referentes ao WS-Management que são relevantes para sistemas de gestão. Para cada uma destas áreas de abrangência, a especificação WS-Management define requisitos mínimos de implementação adequados às implementações de WS.

Diferente do WSDM (MUWS e MOWS), o WS-Management é organizado num único documento cobrindo apenas parte do que o WSDM abrange no que toca ao endereçamento: como identificar um recurso a gerir e como comunicar com ele.

Seguindo a natureza dos WS, o WS-Management, à semelhança do MUWS, depende também de uma série de especificações: XML Schema, WSDL, WS-Addressing, Web Services Eventing (WS-Eventing), Web Services Enumeration (WS-Enumeration), Web Services Transfer (WS-Transfer) e WS-Security [54]. O Anexo 4 lista os prefixos e os

namespaces XML usados na versão 1.0.0 da especificação WS-Management de 2008 [59].

O WS-Addressing define referências a pontos finais de ligação (endpoints) – também conhecidas por EPRs – que são utilizadas pelo WS-Management como modelo de endereçamento para recursos individuais [59, 60]. O EPR relativo ao recurso pretendido é mencionado no cabeçalho da mensagem SOAP. O EPR de gestão usa uma representação fixa, a qual é um registo dos seguintes cabeçalhos SOAP:

- *wsa:To* (obrigatório) – endereço de transporte do serviço;
- *wsman:ResourceURI* (obrigatório se for usado o modelo de endereçamento por omissão) – URI (Uniform Resource Identifier) do recurso de representação da classe ou instância de representação;
- *wsman:SelectorSet* (opcional) – identifica ou selecciona a instância do recurso a aceder se existir mais que uma instância de uma classe de recursos.

A operação a ser invocada sobre o recurso de gestão é indicada por uma URI através da mensagem *wsa:Action*. O Anexo 5 lista as URIs referentes à versão 1.0.0 da especificação WS-Management de 2008 [59].

<b>Acesso a Recursos</b>	
Get	Obtém representações de recursos
Delete	Elimina instâncias de recursos
Create	Cria recursos e modela “construtores” lógicos
Put	Actualiza integralmente um recurso específico
<b>Eventing</b>	
Subscribe	Pede que sejam enviados eventos para um cliente
Unsubscribe	Cancela a subscrição
Renew	Aumenta a duração da subscrição
SubscriptionEnd	Avisa um subscritor que terminou a subscrição
Acknowledgment of Delivery	Subscritor acusa fisicamente eventos de entrega (nível aplicacional)
Refusal of Delivery	Subscritor responde com um <i>fault</i> em vez de um <i>acknowledge</i> , seja por razões de segurança. Seja por alterações de política
<b>WS-Enumeration</b>	
Enumerate	Estabelece um contexto de enumeração
Pull	Interacção de operações de <i>pull</i> sobre o conjunto resultado
Release	Liberta o enumerador e os recursos associados

**Tabela 2.5 – Operações WS-Management [3, 59]**

As mensagens e paradigmas WS-Eventing são usadas para publicar eventos emitidos pelos serviços. A mensagem *wse:Subscribe* é usada para permitir aos clientes expressarem interesse em receberem eventos. A identidade da origem do evento é baseada no EPR do WS-Addressing. A mensagem *wse:Unsubscribe* cancela a subscrição de recepção de eventos.

O WS-Enumeration é o responsável pela repetição resultante da enumeração ou consulta de um conjunto de instâncias associadas a um determinado recurso que fornece tal mecanismo. A especificação WS-Enumeration indica essa enumeração como uma operação em três partes:

- *wsen:Enumerate* – mensagem inicial enviada para estabelecer o contexto de enumeração;
- *wsen:Pull* – operações de repetição usadas no conjunto resultado;
- *wsen:Release* – quando o repetidor da enumeração já não é necessário, é enviada esta mensagem libertando o enumerador e outros recursos associados.

A especificação WS-Transfer é usada como base para os recursos unários de acesso: *Get*, *Put*, *Create*, e *Delete* através das mensagens *wxf:Get*, *wxf:Put*, *wxf>Create* e *wxf>Delete*.

No campo da segurança, é desejável que operações e respostas de gestão estejam protegidas de ataques como acesso não autorizado a dados, interceptação, réplicas e modificação durante a transmissão. Além disso, a autenticação do utilizador através do envio de um pedido, permite aplicar regras no controle de acessos por forma a determinar como atender a um pedido de gestão. A especificação WS-Security prevê a utilização de uma chave de acesso associada ao perfil do utilizador onde o elemento *wsse:UsernameToken*, definido no WS-Security Username Token Profile 1.0, oferece a possibilidade de definir nome de utilizador e palavra passe. A Figura 2.24 exemplifica esta situação. O pedido da chave de segurança (*wst:RequestSecurityToken*) é feito com base nas especificações WS-Trust.

```
<wst:RequestedSecurityToken>
  <wsse:UsernameToken>
    <wsse:Username>user-name/wsse:Username>
    <wsse:Password>password</wsse:Password>
  </wsse:UsernameToken>
  <!-- Conteúdo do Body -->
</wst:RequestedSecurityToken>
```

**Figura 2.24 - Pedido da chave de segurança em WS-Management**

## MUWS, WS-Management e a gestão de redes

Analisando as especificações MUWS e WS-Management conclui-se o seguinte [54]:

- MUWS é uma solução mais completa e exigente relacionada com gestão de sistemas distribuídos e composta por recursos de tipos bem distintos;

- WS-Management é uma solução mais leve que exige uma menor utilização da rede, menor capacidade de processamento e apresenta tempos de resposta menores.

Um dos objectivos recentes tem sido fazer convergir WSDM com WS-Management desenvolvendo um conjunto comum de especificações para gestão de recursos que possa ser suportado através de múltiplas plataformas.

Embora, inicialmente, MUWS e WS-Management tenham sido desenvolvidas para gestão de serviços de propósito geral, ambos podem ser aplicados à situação específica da gestão de redes. Nesse contexto, as especificações WS podem ser comparadas ao padrão de gestão de redes TCP/IP, o SNMP.

## 2.5 Análise comparativa

Desde a década de 80 que têm sido feitos esforços consideráveis na pesquisa e uniformização de protocolos, linguagens e outras tecnologias relacionadas com a gestão de redes e sistemas.

### SNMP

O primeiro esforço feito nesse sentido, cuja relevância não se limita apenas a aspectos de carácter histórico, foi o protocolo de gestão SNMP desenvolvido para redes baseadas em IP, cuja comunicação se baseia em pacotes UDP. Hoje é ainda muito utilizado devido, essencialmente, às características que apresenta, simples e leve, fazendo jus ao seu nome – Simple Network Management Protocol.

O funcionamento do SNMP assenta na troca de mensagens de gestão entre uma entidade gestora (Gestor) e uma entidade a gerir (Agente). O objectivo é o Gestor, por um lado, obter informação de gestão relevante contida na base de dados local do Agente, a MIB e, por outro lado, registar a ocorrência de eventos que ocorram também do lado do Agente (Traps). A informação armazenada obedece a uma estrutura específica, SMI, que descreve os objectos de gestão hierarquicamente representando-os de forma extremamente condensada através de uma notação numérica. A informação de gestão é transmitida usando uma notação sintáctica abstracta, ASN.1, retirada da OSI, que descreve dados transmitidos por protocolos de comunicação (como por exemplo, o SNMP) de forma independente da linguagem usada, representação física dos dados, aplicação e complexidade. Toda a troca de informação ocorre em formato binário. Os Agentes SNMP estão organizados por comunidades e só aceitam pedidos de gestores que estejam autorizados nessa comunidade. Este mecanismo fornece alguma segurança limitando os pedidos feitos pelos gestores.

Após a versão inicial do protocolo SNMP foram surgindo novas versões com o objectivo de introduzir alguns melhoramentos, quer ao nível da monitorização da própria rede, quer ao nível da privacidade e autenticação. Destacaram-se as versões SNMPv2 e SNMPv3 – esta constitui a última versão deste protocolo.

Quando comparado com outras tecnologias, nomeadamente, WBEM e WS-Management, este protocolo revela-se extremamente eficiente no que respeita ao tráfego gerado pelas suas mensagens – recorde-se que as suas mensagens usam uma notação escalar simples e extremamente condensada sendo ainda codificadas para código binário. No entanto, aquela que é a sua principal virtude, a simplicidade, redonda numa

das suas principais limitações, um modelo de informação e interface de gestão limitados não permitindo configurar Agentes, sendo por isso, mais usado para monitorização que para gestão intrusiva. Além disso, são-lhe imputadas ainda outras limitações como: a baixa escalabilidade (transferência de informação ineficiente com o aumento da quantidade de informação), não suportar gestão distribuída e utilizar uma tecnologia de domínio específico, o que gera elevados encargos com o desenvolvimento de produtos comerciais.

## **WBEM**

A Internet trouxe novas realidades no capítulo da gestão. A necessidade de gerir grandes quantidades de dispositivos distintos remotamente e em simultâneo e o surgimento de tecnologias direccionadas para Internet, como a linguagem XML, levaram ao aparecimento de novos paradigmas de gestão, como WBEM e WS.

O padrão de gestão WBEM baseia-se na troca de informação entre Gestor e Agente através do protocolo HTTP ou HTTPS (caso se pretenda garantir segurança) assente em pacotes TCP. Os objectos de gestão são descritos segundo o modelo CIM e representados através da linguagem MOF sendo depois codificados para CIM-XML para serem transportados via HTTP/HTTPS. A gestão pode ser feita via browser que comunica com o Gestor através de HTTP. O Gestor comunica depois com o Agente através de XML sobre HTTP. CIM e MOF estão para WBEM como a MIB e SMI estão para o SNMP.

Do lado do Agente existe o elemento central em toda a gestão WBEM, o CIMOM, responsável por interligar os recursos geridos, a base de dados CIM e a aplicação de gestão. Graças ao CIMOM, é possível que uma aplicação de gestão envie, receba e modifique informações de gestão referentes aos dispositivos a gerir. O CIMOM comunica com os recursos geridos, através de um conjunto de processos que actuam como intermediários, os Providers. Estes últimos podem depois comunicar com objectos SNMP dos recursos a gerir, quer sejam locais, quer sejam remotos – as MIBs destes objectos são mapeadas para CIM segundo normas definidas pelo DMTF.

Em comparação com a gestão SNMP, a gestão WBEM, apresenta algumas vantagens como o facto de usar, na comunicação entre Gestor e Agente, um protocolo aberto, conhecido e de ampla divulgação, HTTP, uma linguagem auto-descritiva, simples, aberta e flexível, XML, o que facilita o acesso a esta tecnologia e o desenvolvimento de Agentes e Gestores. No entanto, há um preço a pagar pela escolha da generalidade e aumento do nível de abstracção em comparação com o carácter mais específico e condensado do SNMP, a quantidade de tráfego trocado entre Gestores e Agentes aumenta consideravelmente. Mas como tudo na vida, é preciso pesar os pós e os contras ao fazer uma escolha do modelo de gestão.

Outra vantagem que o WBEM apresenta face ao simples e limitado SNMP, é o facto de permitir uma gestão interventiva, não se limitando à simples monitorização. Permite efectuar alterações nas configurações de um Agente, seja um encaminhador de pacotes, um computador, uma aplicação de software, etc.

## Gestão baseada em WS

No final da década de 90, surgiram intenções claras de uniformizar a comunicação entre sistemas diferentes espalhados ao longo da Internet proporcionando a interoperabilidade entre sistemas distribuídos. Inicialmente aplicados à gestão de negócios, os Web Services (WS) ganharam enorme popularidade por serem orientados para a Web e de carácter aberto.

O funcionamento dos WS assenta em serviços. Para tal, existem fornecedores, requisitantes e registos de serviços. Os fornecedores publicam nos registos os serviços por si disponibilizados. Quando um requisitante precisa de um serviço que não está disponível localmente, procura a sua localização no registo de serviços. Com a localização do ponto final de comunicação do serviço fornecida pelo registo, o requisitante solicita-o ao respectivo fornecedor. Os serviços são descritos formalmente usando a linguagem WSDL. Por sua vez os serviços são identificados e localizados através de uma especificação designada por UDDI. A troca de mensagens entre os diversos intervenientes é feita através do protocolo SOAP assente em HTTP, HTTPS, FTP ou SMTP. Um documento SOAP tem uma estrutura simples e, à semelhança do WSDL e UDDI é baseado em XML.

Relativamente à gestão baseada em WS, fornecedor e requisitante dos serviços actuam, respectivamente, como Agente e Gestor. O Agente, fornece operações de gestão sobre o elemento de rede específico que representa. O registo central de serviços, UDDI, contém as informações de acesso aos Agentes, incluindo detalhes relativos à interface de gestão. O Gestor, identifica no registo de serviços os fornecedores relativos aos elementos de rede que pretende gerir, obtendo os detalhes necessários à comunicação. Existem actualmente dois padrões de gestão de redes baseados em serviços, WSDM e WS-Management. O primeiro constitui uma solução mais completa relacionada com gestão de sistemas distribuídos, o segundo, uma solução mais leve. Desenvolvidos para gestão de serviços de propósito geral, ambos podem ser aplicados à situação específica da gestão de redes.

Sendo uma tecnologia baseada em XML, a gestão baseada em WS, à semelhança do WBEM, oferece interoperabilidade. Este aspecto diferencia positivamente estas duas tecnologias face ao SNMP, no entanto constitui também a principal limitação das mesmas, uma vez que contribui para o carácter verboso de ambas, condicionando o seu desempenho. Há que realçar que a tecnologia WS oferece uma gestão mais ampla que WBEM possibilitando a gestão integrada de redes, sistemas, serviços e negócios.

A Tabela 2.6 apresenta uma tabela de equivalência dos principais conceitos envolvidos nas três tecnologias de gestão em estudo.

Embora WS seja uma tecnologia extremamente popular, o seu uso na gestão de redes muito dependerá das iniciativas do IETF no que respeita à próxima geração de protocolos de gestão para a Internet baseados em WS. Por outro lado, muitas das tecnologias usadas actualmente, algumas já com alguns anos, como o caso do SNMP, continuarão a ser usadas por diversas razões: até ao momento, não surgiu, uma tecnologia consensual que preencha todos os requisitos dos diferentes domínios de gestão, por outro lado, o investimento feito no desenvolvimento de padrões e tecnologias de gestão, desde as mais antigas até às actuais, não pode ser desprezado e, por fim, a constante evolução na pesquisa e desenvolvimento desta área traz novas e melhores tecnologias. Assim, é de esperar uma coexistência entre diversas tecnologias de gestão no futuro [45].

<b>CONCEITO</b>	<b>SNMP</b>	<b>WBEM</b>	<b>WS</b>
<b>Consórcio</b>	IETF	DMTF	DMTF/W3C
<b>Modelo de Dados</b>	MIB	CIM	CIM
<b>Estrutura de Dados</b>	SMI	MOF	MOF
<b>Codificação</b>	ASN.1	CIM/XML	CIM/XML
<b>Transporte</b>	UDP	TCP	TCP
<b>Protocolo de Comunicação</b>	SNMP	HTTP(S)	HTTP(S)/FTP/SMTP SOAP

**Tabela 2.6 – Equivalência de Conceitos**

## **2.6 Sumário**

Este capítulo fez uma análise detalhada de três modelos de gestão distintos, SNMP, WBEM e gestão baseada em WS. Foram explanados com algum detalhe os principais conceitos, tecnologias e modo de funcionamento de cada modelo. Apontaram-se os pontos fortes e menos atraentes de cada sendo, no final, apresentada uma análise comparativa entre os mesmos.



# Cap. 3 - Experiências realizadas

---

## 3.1 Introdução

Ao longo das últimas duas décadas o interesse pela gestão de redes tem vindo a aumentar. Como resultado de iniciativas de pesquisa e esforços de padronização feitos nesta área, surgiu uma grande variedade de tecnologias e normas de gestão, visando atingir objectivos e propósitos distintos, de tal forma que existe hoje uma vasta oferta de modelos de gestão de redes. Por outro lado, muito do que foi desenvolvido quando a gestão de redes estava dando os primeiros passos é hoje ainda amplamente usado. Um caso típico disso é o protocolo SNMP, cuja substituição por tecnologias mais recentes tem vindo a ser adiada, de tal forma que muitos investigadores na área da gestão de redes têm até proposto, para alguns ambientes de gestão particulares, uma coexistência entre algumas destas tecnologias, como por exemplo, SNMP e WS [61].

O aumento da complexidade das redes de computadores, quer em quantidade e variedade de dispositivos quer em abrangência geográfica, bem como, o aumento da capacidade de processamento e armazenamento de informação por parte dos elementos constituintes de uma rede, o surgimento da Internet, a computação distribuída e diversos protocolos e padrões associados têm trazido novos desafios à gestão das redes, como por exemplo, questões de segurança, consumo de largura de banda e a necessidade de uniformizar as diversas tecnologias de gestão. Estes aspectos foram contribuindo para o surgimento de tecnologias e normas de gestão orientados para a Internet, sendo o WBEM e WS-Management exemplos disso.

Perante tanta e diversificada oferta, importa fazer uma avaliação de algumas das mais representativas e distintas tecnologias e normas de gestão: por um lado o padrão SNMP que tem vindo a servir de base à maioria das iniciativas nesta área, por outro lado algumas tecnologias e normas mais direccionadas para a realidade das redes actuais e para a utilização da Internet, como sejam o WBEM e a gestão baseada em WS.

Com o conjunto de testes realizados, pretendeu-se avaliar o mérito das tecnologias de gestão SNMP, WBEM e WS-Management (esta última em representação das tecnologias de gestão baseadas em WS) no que respeita à eficiência, capacidade de gestão e escalabilidade. Para tal procedeu-se à avaliação da sinalização e tempo de resposta para cada uma destas tecnologias, efectuando pedidos distintos de objectos, por parte de um Gestor a um Agente, medindo depois os parâmetros pretendidos com um analisador de protocolos.

Tendo sido verificada uma grande discrepância das quantidades de sinalização foi realizado um estudo comparativo sobre o comportamento do SNMP, WBEM e WS-Management com compressão das mensagens trocadas entre Gestor e Agente com o objectivo de verificar se a compressão seria uma solução viável para minimizar a

quantidade de informação de sinalização. Verificou-se existirem alterações significativas após a compressão devido ao carácter verboso das tecnologias baseadas em XML (WBEM e WS-Management) em relação à de carácter binário (SNMP).

## 3.2 Cenário de Testes

Pretendeu-se que o cenário escolhido para a execução dos testes fosse minimalista embora comportasse um conjunto mínimo de entidades do cenário de gestão real. A simplificação do cenário, além de simplificar o trabalho de instalação, reduziu também os erros de medição devido à intervenção de factores externos à configuração de teste. A Figura 3.1 ilustra a forma como foi feita a montagem dos testes.

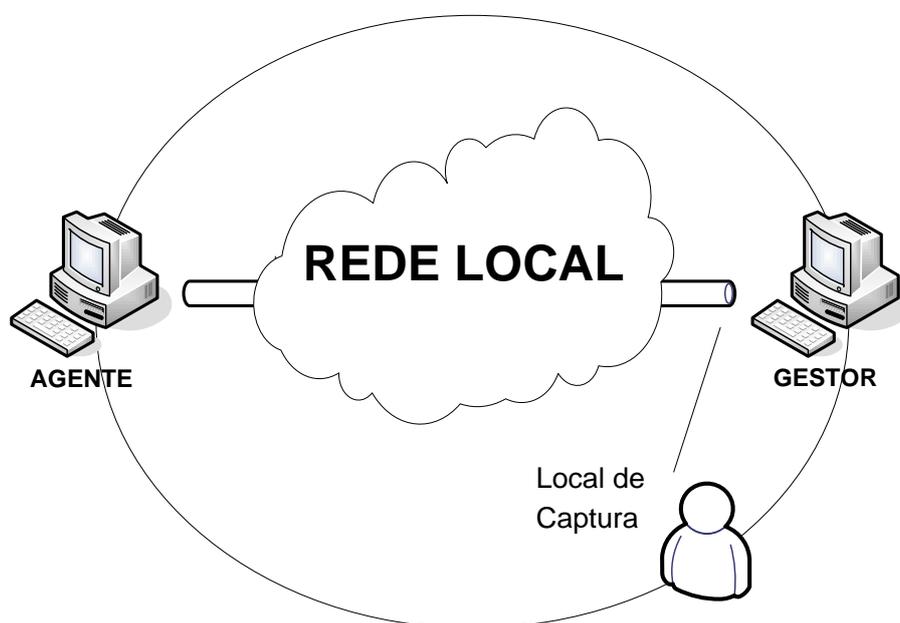


Figura 3.1 – Montagem de Testes

Os elementos do cenário de teste (Gestor e Agente de gestão) estavam em duas máquinas interligadas através de um troço de rede Ethernet (para tal utilizou-se um switch). Para medir os valores utilizou-se um analisador de pacotes (*Wireshark*) de forma a fazer a captura das mensagens trocadas para posterior análise.

Durante os testes realizados às diversas soluções de gestão, foram usadas sempre as mesmas máquinas de forma a garantir igualdade de condições entre testes - *Centrino 1.7 GHz 1 GB RAM* (servidor/Agente) e *Dual Core 1.8 GHz 512 MB RAM* (cliente/Gestor). Para garantir que as máquinas apresentavam a mesma capacidade de resposta para as várias soluções testadas, foi usado sempre o mesmo sistema operativo – *Linux Ubuntu 7.10*.

Os testes que foram realizados pretenderam medir, para as diversas soluções em teste, os seguintes valores:

- tempo médio de resposta (round-trip delay) entre pedido e resposta em operações simples de leitura;
- memória utilizada (número de bytes) pelas aplicações (Gestor e Agente);
- número de pacotes para uma operação de leitura.

Os testes foram realizados para as situações em que foram lidos, 1, 10, 100, 1.000 e 10.000 objectos simples. Utilizou-se um objecto relativo à interface de rede dos routers para gestão dos mesmos. Na troca de objectos entre Gestor e Agente, utilizou-se sempre o mesmo objecto duplicado na base de dados (BD) o número vezes necessário às experiências. A Tabela 3.1 apresenta uma representação deste objecto em termos de elementos/campos constituintes e número de bytes.

<b>Elemento</b>	<b>N. Bytes</b>
InterfaceID	2
NetworkID	2
RouterID	2
Total_BW	2
Address	4
Scope	2
Prefix	2
PrefixAddress	4
IfIndex	2
TOTAL	22

**Tabela 3.1 – Objecto Interface**

De forma a conseguir eliminar/minorar o efeito de erros de medição que pudessem surgir obtiveram-se 10 medidas para cada situação dos testes.

No que diz respeito aos testes realizados com compressão de dados utilizou-se o método Deflate [62].

### **3.3 Procedimentos e experiências realizadas**

Os procedimentos envolvidos nos testes incluem um conjunto de pedidos e respostas entre cliente e servidor e a interacção entre o servidor e uma BD. A BD contém a totalidade dos objectos envolvidos no teste (1, 10, 100, 1.000 ou 10.000). O cliente (Gestor) pede ao servidor (Agente) a totalidade dos objectos e este, por sua vez, após consultar a BD, fornece ao cliente a totalidade dos dados na BD. Com o analisador de

pacotes do lado do cliente, procede-se à análise detalhada das mensagens trocadas entre cliente e servidor. A Figura 3.2 esquematiza estes procedimentos.

A BD foi criada usando um Sistema de Gestão de Bases de Dados (SGBD) open source que utiliza a linguagem SQL (Structured Query Language), o MySQL. Para cada teste, usaram-se scripts SQL, que inseriram, respectivamente, 1, 10, 100, 1.000 e 10.000 objectos na BD.

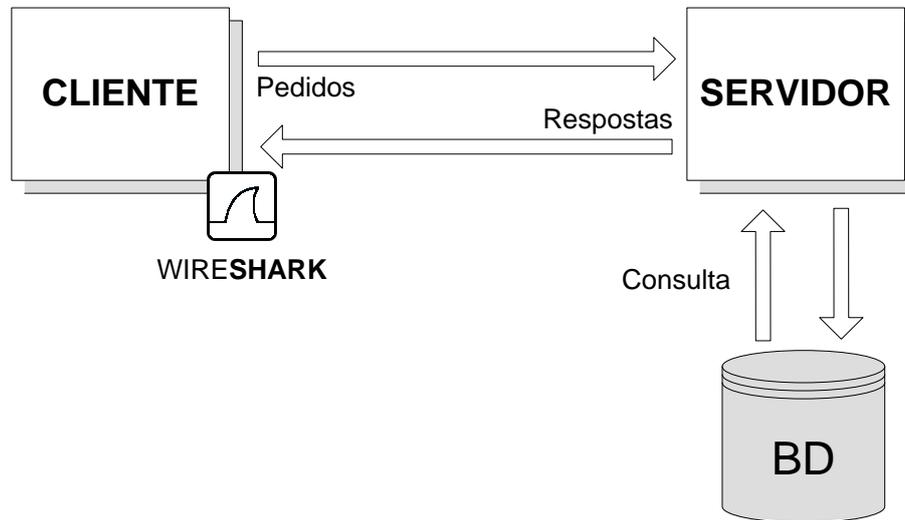


Figura 3.2 – Procedimentos nos Testes

### 3.3.1 Experiências realizadas com SNMP

Na realização dos testes com SNMP foi usado como cliente (Gestor) a aplicação *Snmplibwalk* [63] e como servidor (Agente) uma versão adaptada ao *Snmplibwalk* da API *Agent++* [64].

O *Snmplibwalk* utiliza pedidos *Getbulk* para realizar uma busca eficiente a uma entidade de rede para obter uma árvore de informações.

A API *Agent++* é um pacote de software open source para desenvolver Agentes SNMP em C++ capazes de responder aos vários pedidos SNMP.

O Agente consulta a base de dados procurando os valores pretendidos, codifica-os segundo o protocolo SNMP e envia-os para o Gestor, dando resposta ao pedido *Getbulk* da parte deste.

### 3.3.2 Experiências realizadas com WBEM

A abordagem aos testes com WBEM começou pela instalação, configuração e teste da plataforma WMI (Windows Management Instrumentation) da Microsoft, o MOM (Microsoft Operations Manager) cujo funcionamento é ilustrado na Figura 3.3 que mostra uma captura de uma janela de gestão do MOM.

Apesar de algumas dificuldades inerentes ao processo de instalação, dado que foi necessário efectuar várias actualizações, quer ao sistema operativo, quer ao SQL Server,

constatou-se existirem enormes potencialidades desta plataforma na gestão de computadores inseridos numa rede. Verificou-se também que a plataforma é simples de utilizar e permite uma gestão fácil dos clientes (neste caso, PCs) a correr o cliente MOM. Este aspecto é facilmente observável na Figura 3.3.

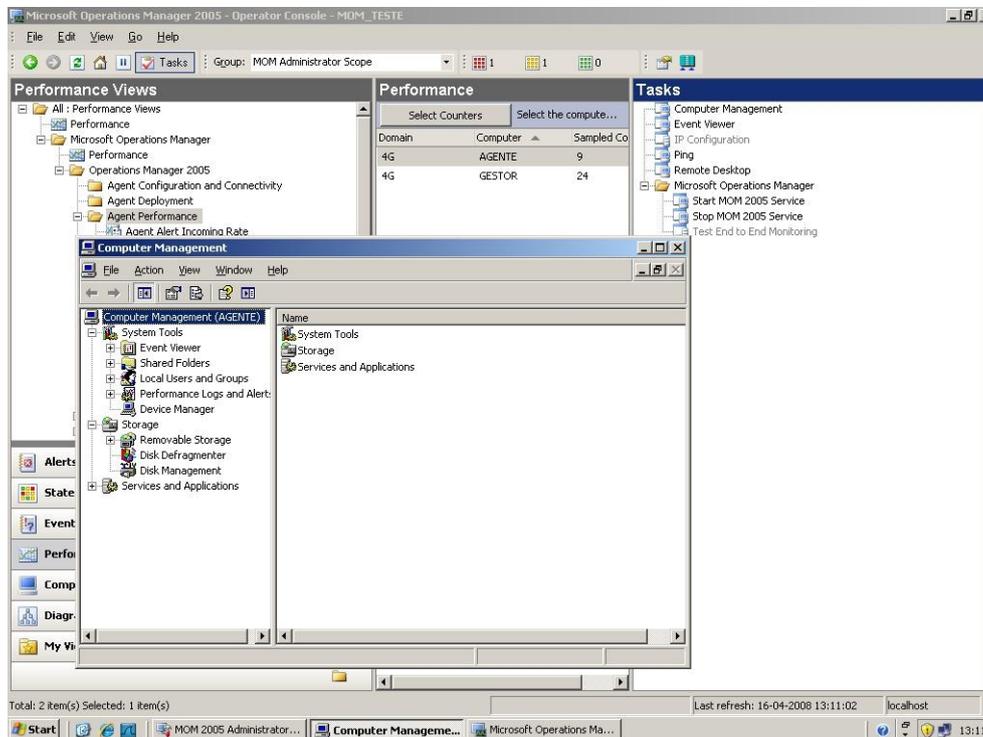


Figura 3.3 – Testes WBEM: MOM

No que respeita à realização dos testes WBEM, a solução MOM acabou por ser abandonada em face do surgimento de alguns constrangimentos que se descrevem a seguir:

- só funciona em ambiente Windows – dado que se pretendia uniformizar os sistemas operativos dos testes, seria mais conveniente usar o Linux;
- tráfego codificado – este facto foi verificado após observar o tráfego captado pelo *Wireshark* do lado do Gestor, não sendo totalmente impeditivo, dificultaria a análise dos resultados e a obtenção dos valores pretendidos;
- não utiliza XML – a Microsoft utiliza (no MOM) um protocolo binário para transporte da informação, tendo como consequência as mensagens serem bem mais pequenas que em WBEM;
- é um produto comercial – não permite acesso ao código fonte, aspecto essencial para forçar o transporte da quantidade de objectos pretendidos – este terá sido o aspecto que mais condicionou a utilização do MOM nos testes.

Os testes WBEM foram realizados utilizando uma aplicação baseada na plataforma open source de desenvolvimento WBEM, o *OpenWBEM* [40] (ver 2.3.6), recorrendo a um Provider especialmente desenvolvido para a extensão que foi feita ao modelo CIM. Como Gestor utilizou-se o *Owexecwql* e como Agente o *Owcimomd*.

### 3.3.3 Experiências realizadas com WS-Management

Nos testes WS-Management foi utilizado *Openwsman* – um projecto que fornece uma implementação open source de gestão WS e uma forma de apresentar informação de gestão em sistemas operativos Linux através do protocolo WS-Management [65].

O *Openwsman* suporta o modelo de dados CIM através de um plugin especial que trata as classes CIM de forma genérica. O mapeamento de dados CIM em recursos WS-Management é feito segundo as orientações do DMTF [66, 67].

A equipa do projecto *Openwsman* não desenvolveu um motor CIM, reutilizou um já existente. Isso pode ser observado na Figura 3.4 onde se verifica que é o Agente WBEM (na circunstância, o *Owcmomd*, referente ao *OpenWBEM*) a interagir com a BD (CIMOM). Ainda na figura é possível observar que, entre as aplicações *Wsmn* (cliente WS) e *Openwsman* (servidor WS), a comunicação é feita com base em SOAP sobre HTTP e entre *Openwsman* e *Owcmomd* (Agente WBEM) é feita com base em CIM-XML sobre HTTP.

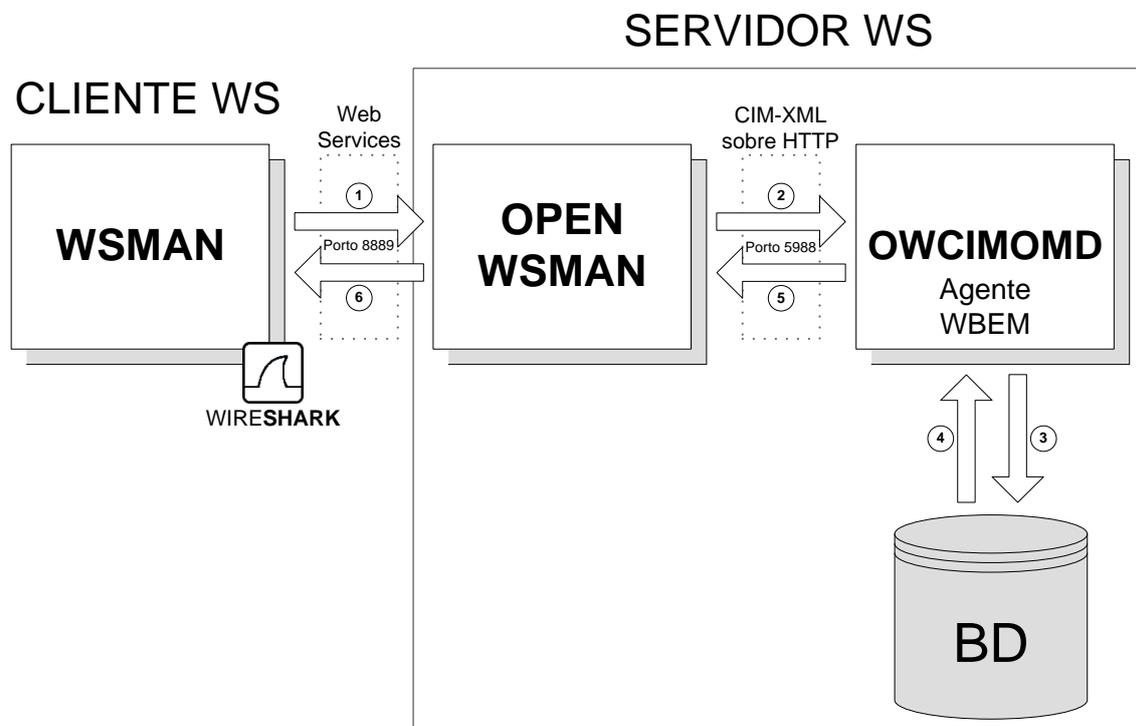


Figura 3.4 – Testes WS-Management

O funcionamento da implementação WS-Management ilustrada na Figura 3.4 assenta no seguinte: (1) o *Wsmn* contacta o *Openwsman* e pede o envio dos dados através de uma consulta – na circunstância, pede o envio da totalidade dos dados existentes da BD – usando SOAP sobre HTTP através do porto 8889; (2) o *Openwsman* reencaminha o pedido para o *Owcmomd* através do porto 5988 usando CIM-XML sobre HTTP; (3,4) o *Owcmomd* procura os objectos pretendidos da BD e codifica-os em CIM-XML; (5) o *Owcmomd* envia os dados codificados em CIM-XML para o *Openwsman* através de uma ligação HTTP usando o porto 5988; (6) o *Openwsman* recebe os dados em HTTP, abre-os e formata-os de novo em CIM-XML, com novas tags, e envia-os ao *Wsmn*

através de SOAP sobre HTTP, usando o porto 8889; finalmente, o cliente WS recebe os dados pedidos, lê-os e imprime-os no ecrã.

### 3.4 Resultados obtidos

Foram realizados diversos testes, para 1, 10, 100, 1.000 e 10.000 objectos, sendo depois calculada as médias referentes ao tempo de resposta, número de bytes e número de pacotes para cada conjunto de testes, de acordo com a quantidade de objectos pedidos e solução de gestão testada.

No que diz respeito ao estudo efectuado com compressão de dados, registaram-se os valores referentes ao número de bytes verificados para cada uma das soluções de gestão e quantidade de objectos pedidos.

Todos os resultados estão registados na tabela da Tabela 3.2.

OBJECTOS	T (mseg)	N. PACOTES	N. BYTES	N. BYTES (c/ compressão)
<b>SNMP</b>				
1	0,002	2	437	325
10	0,022	20	4.332	1.190
100	0,170	182	38.604	10.204
1.000	1,009	1.802	391.558	97.506
10.000	9,602	18.002	3.933.958	987.863
<b>WBEM</b>				
1	2,787	12	3.539	1.920
10	2,792	32	15.221	3.154
100	3,095	146	125.050	12.492
1.000	3,443	1.129	1.210.153	97.671
10.000	11,066	11.122	12.075.185	917.270
<b>WS-Management</b>				
1	0,019	3	3.652	1.932
10	0,059	5	8.226	2.250
100	0,314	34	55.710	6.185
1.000	2,607	323	520.316	43.490
10.000	26,113	3.207	5.052.661	401.573

**Tabela 3.2 – Resultados SNMP, WBEM e WS-Management**

Numa primeira análise, verifica-se que o número de bytes cresce com o número de objectos pedidos para os três modelos de gestão analisados. O modelo SNMP é o que exige menos quantidade de tráfego. Este facto não constitui grande surpresa face às características das mensagens trocadas em SNMP. A tecnologia WS-Management apresenta quantidades de tráfego próximas da tecnologia SNMP.

É de realçar também um valor elevado para o tempo médio de resposta ao pedido de 1 único objecto na tecnologia WBEM e na tecnologia WS-Management quando o número de objectos pedidos atinge os 10.000.

A generalidade dos valores medidos está de acordo com as previsões teóricas, havendo que registar apenas ligeiras diferenças no que toca à tecnologia SNMP. Ao observar as capturas SNMP, verificou-se que esta implementação envia apenas 10 elementos por pacote, não ultrapassando um total de 350 bytes, ficando muito aquém dos 1514 bytes que as outras tecnologias conseguem.

Por fim há que registar uma considerável diminuição na quantidade de tráfego, após utilização de compressão de informação. Este aspecto é mais evidente para as tecnologias baseadas em XML, WBEM e WS-Management.

## 3.5 Estudo comparativo

Com base Tabela 3.2 relativa aos resultados obtidos nas experiências, construíram-se os respectivos gráficos (Figura 3.5 a Figura 3.8) que relacionam os três modelos de gestão no que respeita à análise do tempo, número de bytes, número de pacotes e número de bytes com compressão.

### Análise do tempo

Começando pela análise do tempo, Figura 3.5, podemos verificar que ambas as soluções apresentam uma variação linear – o tempo decorrido entre o pedido do Gestor e a resposta do Agente aumenta proporcionalmente com a quantidade de objectos pedidos.

A solução de gestão baseada em SNMP é a que apresenta melhores resultados, seguindo-se, respectivamente, as soluções WBEM e WS-Management. Estes resultados justificam-se pelo tipo de mensagens em causa, as mensagens SNMP têm um carácter binário e, em comparação com as mensagens CIM-XML e SOAP sobre HTTP das tecnologias WBEM e WS-Management, necessitam de menos bytes, logo, menos tempo também. É de notar uma inversão nesta lógica, quando comparamos a quantidade de informação trocada nas mensagens WBEM e WS-Management e também quando comparamos a quantidade de pacotes face ao número de bytes na tecnologia SNMP – ver Figura 3.6 e Figura 3.7 referentes ao número de bytes e número de pacotes, respectivamente.

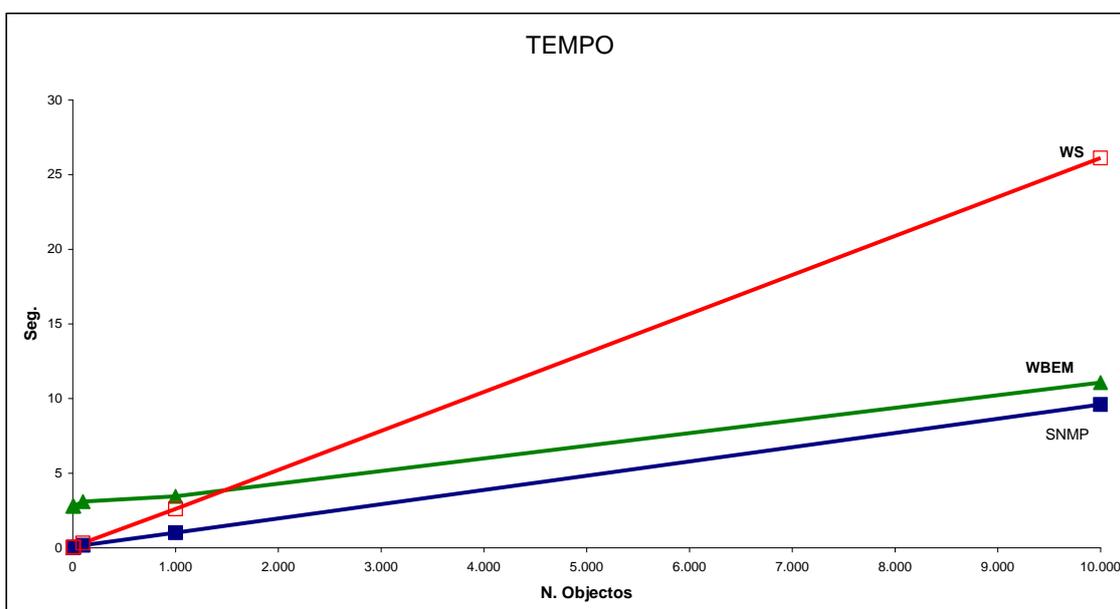
Um outro aspecto a realçar no gráfico do tempo é o facto da tecnologia WBEM apresentar um valor de tempo inicial – referente à troca de um objecto – consideravelmente superior ao das outras tecnologias, cujo valor é muito próximo de zero. Este aspecto, já mencionado na análise da Tabela 3.2, é observável no gráfico por a recta apresentar ordenada na origem. Esta situação justifica-se pelo facto do servidor WBEM exigir ao cliente uma autenticação válida e, como consequência, cliente e servidor necessitem de trocar informação necessária para essa autenticação. Este processo não foi realizado de forma automática, por programação, incluindo a palavra-passe na consulta feita à base de dados, antes, foi realizado a dois tempos, dado que a autenticação foi introduzida manualmente pelo operador humano.

Apesar da tecnologia WBEM apresentar um valor de tempo inicial bastante elevado, o seu comportamento é muito razoável em comparação com o SNMP, cuja recta apresenta um declive até mesmo ligeiramente inferior.

Do gráfico da Figura 3.5, é também possível observar que os piores resultados em termos de tempo de resposta registaram-se na tecnologia WS-Management, existindo

até uma considerável diferença entre esta última e as outras duas, WBEM e SNMP. A explicação do pior resultado do WS-Management face ao SNMP é óbvia face às diferenças na natureza destas duas tecnologias, a primeira com carácter verboso e por isso mais pesada, a segunda com carácter binário e como tal mais leve. Já a comparação entre os resultados obtidos para WS-Management e WBEM, exige outro tipo de explicação. O pior resultado da solução WS-Management justifica-se pelo facto desta funcionar como um adaptador para a tecnologia WBEM (ver a Figura 3.4) e, como consequência, os tempos de resposta de cada um dos servidores se somarem. Dado vez que as mensagens WBEM são consideravelmente maiores que as WS-Management e existe a necessidade de efectuar operações de descodificação/codificação na adaptação das duas tecnologias, compreende-se facilmente o mau desempenho do WS-Management neste capítulo.

Pela observação do gráfico é também possível verificar que o declive da recta referente ao WS-Management é francamente superior ao das outras duas tecnologias, indiciando que o desempenho desta solução se degrada mais rapidamente que as outras duas quando aumenta a quantidade de objectos trocados.



**Figura 3.5 – Tempo médio de resposta: estudo comparativo**

### **Análise do número de bytes**

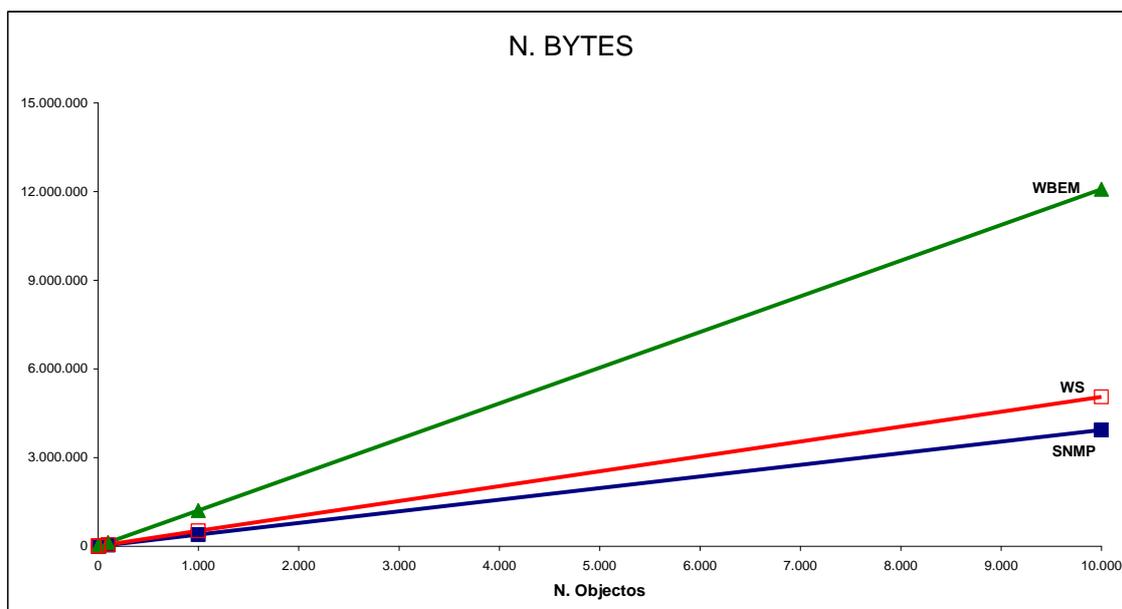
Observando o gráfico da Figura 3.6 verifica-se que a quantidade de bytes trocada entre Gestor e Agente em função da quantidade de objectos pedidos aumenta de forma linear para as três soluções em análise. Esta constatação era de esperar, uma vez que, mais informação requisitada, implica maior tráfego entre Gestor e Agente.

A solução que exige menor troca de informação entre Gestor e Agente é a baseada em SNMP. A explicação para este resultado é a mesma que justifica um menor tempo médio de resposta da solução SNMP. O seu carácter binário torna-a mais eficiente quando comparada com as tecnologias de carácter verboso – WBEM e WS-Management. Além disso é notório, pela observação das mensagens relativas a estas tecnologias baseadas em XML (ver Anexo 2 e 3) que, ambas, apresentam uma escolha

pouco cuidada no que respeita ao tamanho de identificadores e outros elementos afectos à sintaxe usada trazendo consequências à eficiência destas tecnologias.

A solução que gera mais tráfego é a WBEM – apresentando valores significativamente mais elevados que as outras duas tecnologias. Este facto indicia uma aparente discrepância com as conclusões relativas à análise do tempo, em que o WBEM apresenta resultados francamente melhores que o WS-Management. A justificação para este facto prende-se com um aspecto observável directamente nas mensagens WBEM trocadas entre Agente e Gestor, em particular na resposta do primeiro a pedidos do segundo – Anexo 2. Verifica-se que o servidor WBEM repete a descrição de informação para cada objecto trocado – como o namespace do objecto. À medida que o número de objectos aumenta (neste caso a partir de 10), esta repetição causa um considerável aumento de tráfego comprometendo a prestação desta tecnologia em comparação com as restantes – o maior declive da recta referente ao WBEM no gráfico da Figura 3.6 bem como os valores registados na Tabela 3.2 ajudam na observação deste facto.

É de destacar o bom comportamento do WS-Management que, tal como o WBEM, tem um carácter verboso, mas apresenta resultados francamente melhores, aproximando-se da solução SNMP.



**Figura 3.6 – Número de Bytes: estudo comparativo.**

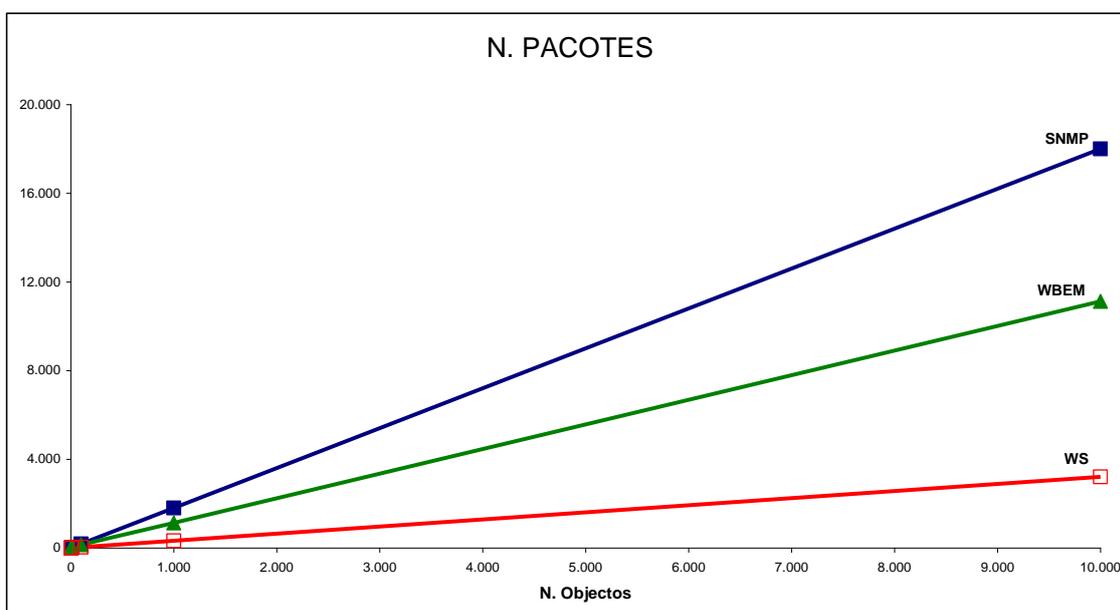
### **Análise do número de pacotes**

À semelhança do que aconteceu com as duas análises anteriores, também no que respeita ao número de pacotes envolvidos nas mensagens de gestão, estes apresentam um crescimento linear com o número de objectos trocados para as três tecnologias em estudo. O gráfico da Figura 3.7 ilustra bem este ponto.

A solução menos eficiente no que toca ao número de pacotes trocados revelou ser a solução SNMP. Embora esta seja a solução que menos tráfego gerou, no entanto, verificou-se pela análise dos pacotes SNMP capturados, que o tamanho médio dos

pacotes é francamente menor em comparação com as outras duas tecnologias em análise (não ultrapassando os 350 bytes), cujo tamanho médio dos pacotes aumenta com o número de objectos (atingindo os 1514 bytes). Constatou-se que WBEM e WS-Management utilizam sempre a capacidade máxima de transporte dos pacotes TCP. Por outro lado, a solução SNMP utilizada cria pacotes pequenos não aproveitando totalmente a capacidade de transporte UDP o que faz aumentar a quantidade de pacotes necessários e o overhead causado pelo protocolo de transporte.

A solução mais eficiente no que respeita ao número de pacotes é a solução baseada em WS-Management apresentando valores francamente melhores que as outras duas tecnologias. Este facto é uma consequência directa do bom desempenho desta solução relativamente ao número de bytes – ver Figura 3.6.



**Figura 3.7 – Número de Pacotes: estudo comparativo**

### **Análise do número de bytes com compressão das mensagens**

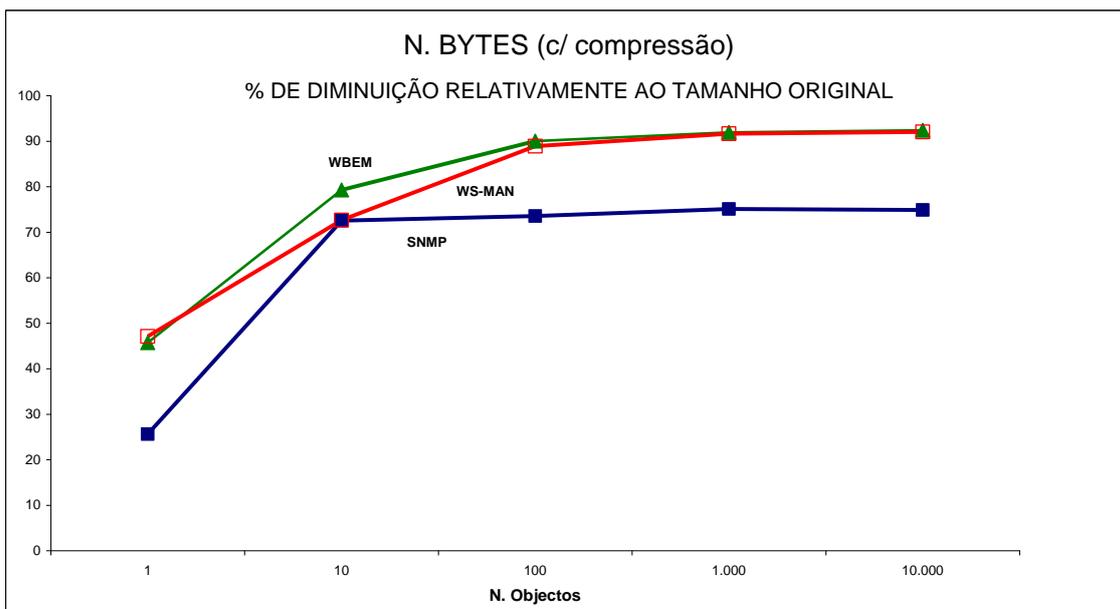
A análise feita às mensagens trocadas revelou que, nas tecnologias baseadas em XML, WBEM e WS-Management, havia, muita repetição de informação e pouco cuidado na escolha de identificadores e outros elementos referentes à sintaxe usada. Estes aspectos terão contribuído para o aumento da quantidade de informação trocada nas mensagens entre Gestor e Agente. Perante isso, entendeu-se que um estudo do comportamento destas tecnologias em face da compressão de dados seria importante. Além disso, saber como se comportaria o SNMP, cujas mensagens têm carácter binário, perante a compressão dos dados, seria também relevante.

Os resultados obtidos para a quantidade de informação trocada (número de bytes) estão registados na Tabela 3.2. A análise dos valores da tabela permite-nos concluir que, à semelhança do que aconteceu sem a compressão, também neste caso, ambas as grandezas respeitantes a cada solução estudada, aumentaram linearmente com a quantidade de objectos pedidos.

Na construção do gráfico respeitante à análise dos valores da Tabela 3.2 para o número de bytes com compressão (ver Figura 3.8), optou-se por construir um gráfico cujas grandezas variam de forma logarítmica, com o propósito de realçar a factor compressão relativamente a cada uma das tecnologias em causa.

Pela observação do gráfico da Figura 3.8, é notória uma clara diminuição no número de bytes trocados entre Gestor e Agente. Essa diminuição fez-se sentir mais nas tecnologias WBEM e WS-Management, onde a percentagem de diminuição chegou perto dos 90% logo a partir de 100 objectos pedidos, mantendo-se depois nesse valor. Estes resultados vieram de encontro ao esperado e justificam-se por dois factores. O primeiro, a característica verbosa (e pouco ‘cuidada’) das mensagens baseadas em XML das tecnologias WBEM e WS-Management. O segundo, o facto da informação fornecida pela base de dados resumir-se à repetição em massa (100, 1.000 e 10.000) do mesmo objecto, o que faz aumentar a eficiência do algoritmo de compressão.

Já em relação ao SNMP, verifica-se existir uma diminuição do número de bytes, embora, como seria de esperar, não tão expressiva como nas outras duas tecnologias, dado o seu carácter binário. No entanto é de realçar que, a partir dos 10 objectos, a percentagem de diminuição aproxima-se dos 70% e fixando-se perto dos 75% a partir dos 100 objectos.



**Figura 3.8 – Número de Bytes com compressão**

A compressão dos dados trocados entre Gestor e Agente nas três soluções de gestão em causa permite constatar que o desempenho da tecnologia SNMP relativamente ao número de bytes, passa de melhor (sem compressão) para a pior (com compressão). Este resultado seria de esperar uma vez que esta tecnologia utiliza mensagens codificadas em binário, ao contrário das outras que se baseiam em XML. De facto, o que influencia estes resultados é o desempenho do próprio algoritmo de compressão que, no caso do SNMP (informação binária) se revela inferior ao das tecnologias baseadas em XML.

## **3.6 Sumário**

Neste capítulo foi feita uma descrição detalhada das experiências realizadas para avaliar as três soluções de gestão, SNMP, WBEM e WS-Management, indicando, o que se pretendeu avaliar, como foi feita essa avaliação, quais os resultados esperados e quais os resultados obtidos, apresentando depois as devidas justificações quer, quando os valores se revelavam discrepantes à previsão inicial, quer quando vinham ao encontro do esperado. A avaliação em causa incidiu sobre quatro aspectos: o tempo médio de resposta, o número de bytes, o número de pacotes e o número de bytes com compressão. Para cada um destes factores foram apresentados os valores medidos e gráficos que relacionam as três tecnologias em avaliação.



# Cap. 4 – Conclusões

---

Com este trabalho pretendeu-se realizar um estudo comparativo entre diversas tecnologias de gestão de redes começando pelo SNMP, depois WBEM e, finalmente o WS-Management. Uma rede de computadores actual é composta por inúmeros e diversificados componentes cuja gestão é extremamente complexa e suscita questões pertinentes sobre qual a melhor solução (Network Management System – NMS) a aplicar em determinada situação.

No estudo realizado pretendeu-se efectuar uma análise comparativa acerca da eficiência das diversas tecnologias de gestão, quer em termos da quantidade de sinalização trocadas, o número de mensagens trocadas durante as operações de gestão bem como o tempo de resposta. Foram efectuados testes recorrendo a pares cliente-servidor de gestão baseados em cada uma das tecnologias e foram trocados quantidades predefinidas de objectos de um tamanho conhecido. O tráfego criado no processo foi capturado e analisado recorrendo a um interpretador de protocolos. Foram recolhidos os valores de sucessivos testes com as diversas soluções de gestão e foram construídos gráficos de evolução das várias grandezas com o crescimento da quantidade de objectos trocados.

De uma forma geral podemos verificar que todas as grandezas medidas crescem linearmente com o aumento do número de objectos trocados. Os declives de cada uma das rectas dependem da performance particular de cada uma das tecnologias,

No que respeita ao **tempo de resposta** podemos verificar que a melhor performance foi atingida pela solução baseada em SNMP e que a pior das performances foi medida à solução baseada em WS-Management. Podemos ainda verificar que a solução WBEM apresenta um tempo mínimo de resposta elevado que se deve ao facto de o servidor exigir ao cliente uma autenticação válida e de cliente e servidor necessitarem de trocar informação para essa autenticação. A diferença de tempo de resposta entre WBEM e WS-Management deve-se ao facto de a solução WS-Management funcionar como um adaptador para a tecnologia WBEM e por isso os tempos de resposta de cada um dos servidores se somarem.

A análise da **quantidade de informação** trocada entre o cliente e o servidor mostrou que a solução que menos tráfego gerou foi a solução SNMP e que a solução que mais tráfego gerou foi a solução WBEM. A melhor prestação da solução SNMP deve-se ao carácter binário dos pacotes do protocolo o que evita a transferência de grandes quantidades de texto das tags XML. A diferença de prestação entre as soluções WBEM e WS-Management deve-se ao facto de o servidor WBEM repetir a descrição de informação como o namespace do objecto dentro de cada um dos objectos trocados. Essa repetição provoca uma considerável quantidade de tráfego que compromete a prestação do WBEM.

De realçar a grande melhoria obtida por ambas as tecnologias quanto se comparou o número de bytes em face da **compressão das mensagens**. Nesse caso, a solução SNMP passou a apresentar a pior prestação das três tecnologias sendo a melhor o WS-Management. Este resultado não constitui grande surpresa uma vez que a solução

SNMP utiliza mensagens codificadas em binário sendo que as outras duas se baseiam em XML – texto. Assim verifica-se um desempenho inferior do algoritmo de compressão com informação binária. O facto do WS-Management apresentar melhores resultados que o WBEM não está relacionado com a compressão, apenas se manteve a tendência desta tecnologia gerar menos informação que WBEM. O facto de ambas apresentarem a mesma percentagem de diminuição a partir de 100 objectos, confirma a afirmação anterior.

Em termos do **número de pacotes** trocados entre cliente e servidor verificou-se que a solução mais eficiente foi a solução baseada em WS e a menos eficiente foi a solução baseada em SNMP. O facto de a solução WS-Management se ter mostrado mais eficiente do que a solução WBEM está relacionado com a menor quantidade de informação trocada entre o cliente e o servidor. Menor quantidade de informação obriga a um menor número de pacotes para transportar a informação. O facto de a solução SNMP ter apresentado um maior número de pacotes do que a solução WS-Management já não é confirmado pela quantidade de informação trocada entre as duas entidades. Após a análise dos pacotes capturados nas experiências SNMP verificamos que o tamanho médio dos pacotes SNMP é significativamente mais pequeno do que os pacotes WBEM e WS-Management que utilizam sempre a capacidade máxima de transporte dos pacotes TCP. Trata-se de uma limitação da solução SNMP utilizada que cria pacotes pequenos não aproveitando totalmente a capacidade de transporte do protocolo UDP e assim aumenta a quantidade de pacotes trocados, bem como o overhead causado pelo protocolo de transporte.

Como **trabalho futuro** pensamos que seria relevante estender este estudo a outros protocolos de gestão como COPS (Common Open Policy Service) [68], NETCONF (Network Configuration Protocol) [69] e/ou Diameter [70].

Por outro lado, um estudo sobre o comportamento de protocolos de gestão numa perspectiva operacional, quando a dimensão da rede exige milhões de pequenas mensagens de autenticação em simultâneo.

## Anexo 1 – Lista de RFCs relacionados com SNMP

RFC	Ano	Status	Título / Autores
<b>Referentes à primeira versão do SNMP</b>			
1155	1990	STANDARD	Structure and identification of management information for TCP/IP-based internets. M.T. Rose, K. McCloghrie.
1156	1990	HISTORIC	Management Information Base for network management of TCP/IP-based internets. K. McCloghrie, M.T. Rose.
1157	1990	HISTORIC	Simple Network Management Protocol (SNMP). J.D. Case, M. Fedor, M.L. Schoffstall, J. Davin
1213	1991	STANDARD	Management Information Base for Network Management of TCP/IP-based internets:MIB-II. K. McCloghrie, M. Rose.
4022	2005	PROPOSED STANDARD	Management Information Base for the Transmission Control Protocol (TCP). R. Raghunarayan
4113	2005	PROPOSED STANDARD	Management Information Base for the User Datagram Protocol (UDP). B. Fenner, J. Flick
4293	2006	PROPOSED STANDARD	Management Information Base for the Internet Protocol (IP). S. Routhier
<b>Referentes ao SNMPv2</b>			
1901	1996	HISTORIC	Introduction to Community-based SNMPv2. J. Case, K. McCloghrie, M. Rose, S. Waldbusser
2578	1999	STANDARD	Structure of Management Information Version 2 (SMIv2). K. McCloghrie, D. Perkins, J. Schoenwaelder
2579	1999	STANDARD	Textual Conventions for SMIv2. K. McCloghrie, D. Perkins, J. Schoenwaelder
2580	1999	STANDARD	Conformance Statements for SMIv2. K. McCloghrie, D. Perkins, J. Schoenwaelder.
3416	2002	STANDARD	Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP). R. Presuhn
3417	2002	STANDARD	Transport Mappings for the Simple Network Management Protocol (SNMP). R. Presuhn
3418	2002	STANDARD	Management Information Base (MIB) for the Simple Network Management Protocol (SNMP). R. Presuhn
3584	2003	BEST CURRENT PRACTICE	Coexistence between Version 1, 2, and 3 of the Internet-standard Network Management Framework. R. Frye, D. Levi, S. Routhier, B. Wijnen.
<b>Referentes ao SNMPv3</b>			
3411	2002	STANDARD	An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks. D. Harrington, R. Presuhn, B. Wijnen
3412	2002	STANDARD	Message Processing and Dispatching for the Simple Network Management Protocol (SNMP). J. Case, D. Harrington, R. Presuhn, B. Wijnen.
3413	2002	STANDARD	Simple Network Management Protocol (SNMP) Applications. D. Levi, P. Meyer, B. Stewart
3414	2002	STANDARD	User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3). U. Blumenthal, B. Wijnen
3415	2002	STANDARD	View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP). B. Wijnen, R. Presuhn, K. McCloghrie

## Anexo 2 – Troca de Mensagens WBEM

PEDIDO de 1 Objecto

```
POST http://192.168.2.4:5988/cimom HTTP/1.1
Authorization: Basic cm9vdDpxd2VydHk=
Host: 192.168.2.4:5988
Pragma: no-cache
Proxy-Connection: Keep-Alive
Content-type: application/xml; charset="utf-8"
CIMProtocolVersion: 1.0
CIMOperation: MethodCall
CIMMethod: ExecQuery
CIMObject: root%2Fcimv2
Content-Length: 476
```

```
<?xml version="1.0" encoding="utf-8" ?>
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
<MESSAGE ID="4711" PROTOCOLVERSION="1.0">
<SIMPLEREQ>
<IMETHODCALL NAME="ExecQuery"><LOCALNAMESPACEPATH>
<NAMESPACE NAME="root"></NAMESPACE>
<NAMESPACE NAME="cimv2"></NAMESPACE>
</LOCALNAMESPACEPATH>
<IPARAMVALUE NAME="QueryLanguage">
<VALUE>WQL</VALUE>
</IPARAMVALUE><IPARAMVALUE NAME="Query">
<VALUE>select * from D_Interface</VALUE>
</IPARAMVALUE></IMETHODCALL>
</SIMPLEREQ>
</MESSAGE>
</CIM>
```

RESPOSTA

```
HTTP/1.1 200 Ok
Date: Mon, 7 Apr 2008 15:52:49 GMT
Cache-Control: no-cache
Server: openwbem/3.1.0 (CIMOM)
Content-Type: application/xml; charset="utf-8"
Man: http://www.dmtf.org/cim/mapping/http/v1.0 ; ns=01
01-CIMOperation: MethodResponse
Content-Length: 1338
```

```
<?xml version="1.0" ?><CIM CIMVERSION="2.0" DTDVERSION="2.0"><MESSAGE ID="4711"
PROTOCOLVERSION="1.0">
<SIMPLERSP>
<IMETHODRESPONSE NAME="ExecQuery">
<IRETURNVALUE>
<VALUE.OBJECTWITHPATH><INSTANCEPATH><NAMESPACEPATH><HOST>127.0.0.1</HOST><L
OCALNAMESPACEPATH><NAMESPACE NAME="root"></NAMESPACE><NAMESPACE
NAME="cimv2"></NAMESPACE></LOCALNAMESPACEPATH></NAMESPACEPATH>
<INSTANCENAME CLASSNAME="D_Interface"><KEYBINDING NAME="InterfaceID"><KEYVALUE
VALUETYPE="numeric">39280</KEYVALUE></KEYBINDING></INSTANCENAME></INSTANCEPATH
><INSTANCE CLASSNAME="D_Interface"><PROPERTY NAME="InterfaceID" TYPE="uint16"
><VALUE>39280</VALUE></PROPERTY><PROPERTY NAME="ADDR" TYPE="string"
><VALUE>2001:690:2380:778f:250:daff:fed6:499c</VALUE></PROPERTY><PROPERTY
NAME="Bandwidth" TYPE="uint16" ><VALUE>10</VALUE></PROPERTY><PROPERTY
NAME="Uplink" TYPE="uint16" ><VALUE>1</VALUE></PROPERTY><PROPERTY NAME="Scope"
TYPE="uint16" ><VALUE>20064</VALUE></PROPERTY><PROPERTY NAME="Prefix" TYPE="uint16"
><VALUE>12</VALUE></PROPERTY><PROPERTY NAME="PrefADDR" TYPE="string"
><VALUE>214:748:364:217:222:2222:0001</VALUE></PROPERTY><PROPERTY
NAME="BandwidthUnit" TYPE="uint16" ><VALUE>1800</VALUE></PROPERTY><PROPERTY
NAME="IfIndex" TYPE="uint16">
<VALUE>11</VALUE></PROPERTY></INSTANCE></VALUE.OBJECTWITHPATH>
</IRETURNVALUE></IMETHODRESPONSE></SIMPLERSP></MESSAGE></CIM>
```

## Anexo 3 – Troca de Mensagens WS-Management

PEDIDO de 1 Objecto

```
POST /wsman HTTP/1.1
Authorization: Basic cm9vdDpxd2VydHk= Host: 192.168.2.2:8889 Accept: /* Content-Type:
application/soap+xml;charset=UTF-8 User-Agent: opensman 2.0.0b1 Content-Length: 1045 Expect:
100-continue

<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
xmlns:wsen="http://schemas.xmlsoap.org/ws/2004/09/enumeration"><s:Header><wsa:Action
s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate</wsa:Action>
<wsa:To s:mustUnderstand="true">http://192.168.2.2:8889/wsman</wsa:To><wsman:ResourceURI
s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/*</wsman:ResourceURI><wsa:MessageID
s:mustUnderstand="true">uuid:9e953b49-4a70-1a70-8002-
e675dd8f1300</wsa:MessageID><wsa:ReplyTo><wsa:Address>http://schemas.xmlsoap.org/ws/2004/08
/addressing/role/anonymous</wsa:Address></wsa:ReplyTo></s:Header><s:Body><wsen:Enumerate><ws
man:OptimizeEnumeration/><wsman:Filter
Dialect="http://schemas.microsoft.com/wbem/wsman/1/WQL">select * from
D_Interface</wsman:Filter><wsman:MaxElements>10000</wsman:MaxElements></wsen:Enumerate></
s:Body></s:Envelope>
```

RESPOSTA

```
HTTP/1.1 200 OK Server: opensman/2.0.0b1 Content-Length: 1241 Content-Type:
application/soap+xml;charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsen="http://schemas.xmlsoap.org/ws/2004/09/enumeration"
xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd">
<s:Header><wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To><ws
a:Action>http://schemas.xmlsoap.org/ws/2004/09/enumeration/EnumerateResponse</wsa:Action><wsa:
RelatesTo>uuid:9e953b49-4a70-1a70-8002-
e675dd8f1300</wsa:RelatesTo><wsa:MessageID>uuid:9ea4172a-4a70-1a70-8005-
b24b07361600</wsa:MessageID>
</s:Header>
<s:Body>
<wsen:EnumerateResponse>
<wsman:Items>
<p:D_Interface xmlns:p="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/D_Interface"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<p:ADDR>2001:690:2380:778f:250:daff:fed6:499c</p:ADDR><p:Bandwidth>10</p:Bandwidth><p:Bandw
idthUnit>1800</p:BandwidthUnit><p:IfIndex>11</p:IfIndex><p:InterfacelD>39280</p:InterfaceID><p:Pref
ADDR>214:748:364:217:222:2222:0001</p:PrefADDR><p:Prefix>12</p:Prefix><p:Scope>20064</p:SCO
pe><p:Uplink>1</p:Uplink></p:D_Interface>
</wsman:Items>
<wsen:EnumerationContext/>
<wsman:EndOfSequence/></wsen:EnumerateResponse>
</s:Body>
</s:Envelope>
```

## Anexo 4 – Prefixos e os Namespaces XML usados em WS-Management

Conforme a versão 1.0.0 da especificação de 2008 [59].

<b>Prefixo</b>	<b>Namespace XML</b>	<b>Especificação</b>
wsman	<a href="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd">http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd</a>	Esta especificação
wsmid	<a href="http://schemas.dmtf.org/wbem/wsman/identity/1/wsmanidentity.xsd">http://schemas.dmtf.org/wbem/wsman/identity/1/wsmanidentity.xsd</a>	Esta especificação – descoberta de versões de protocolos suportados
s	<a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a>	<i>SOAP 1.2</i>
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>	<i>XML Schema 1, XML Schema 2</i>
wsdl	<a href="http://schemas.xmlsoap.org/wsdl">http://schemas.xmlsoap.org/wsdl</a>	<i>WSDL/1.1</i>
wsa	<a href="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://schemas.xmlsoap.org/ws/2004/08/addressing</a>	<i>WS-Addressing</i>
wse	<a href="http://schemas.xmlsoap.org/ws/2004/08/eventing">http://schemas.xmlsoap.org/ws/2004/08/eventing</a>	<i>WS-Eventing</i>
wsen	<a href="http://schemas.xmlsoap.org/ws/2004/09/enumeration">http://schemas.xmlsoap.org/ws/2004/09/enumeration</a>	<i>WS-Enumeration</i>
wxf	<a href="http://schemas.xmlsoap.org/ws/2004/09/transfer">http://schemas.xmlsoap.org/ws/2004/09/transfer</a>	<i>WS-Transfer</i>
wsp	<a href="http://schemas.xmlsoap.org/ws/2004/09/policy">http://schemas.xmlsoap.org/ws/2004/09/policy</a>	<i>WS-Policy</i>
wst	<a href="http://schemas.xmlsoap.org/ws/2005/02/trust">http://schemas.xmlsoap.org/ws/2005/02/trust</a>	<i>WS-Trust</i>
wsse	<a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd</a>	<i>WS-Security</i>
wsu	<a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd</a>	<i>WS-Security</i>

## Anexo 5 – URIs *wsa:Action* usados em WS-Management

Conforme a versão 1.0.0 da especificação de 2008 [59].

<a href="http://schemas.xmlsoap.org/ws/2004/09/transfer/Get">http://schemas.xmlsoap.org/ws/2004/09/transfer/Get</a>
<a href="http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse">http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse</a>
<a href="http://schemas.xmlsoap.org/ws/2004/09/transfer/Put">http://schemas.xmlsoap.org/ws/2004/09/transfer/Put</a>
<a href="http://schemas.xmlsoap.org/ws/2004/09/transfer/PutResponse">http://schemas.xmlsoap.org/ws/2004/09/transfer/PutResponse</a>
<a href="http://schemas.xmlsoap.org/ws/2004/09/transfer/Create">http://schemas.xmlsoap.org/ws/2004/09/transfer/Create</a>
<a href="http://schemas.xmlsoap.org/ws/2004/09/transfer/CreateResponse">http://schemas.xmlsoap.org/ws/2004/09/transfer/CreateResponse</a>
<a href="http://schemas.xmlsoap.org/ws/2004/09/transfer/Delete">http://schemas.xmlsoap.org/ws/2004/09/transfer/Delete</a>
<a href="http://schemas.xmlsoap.org/ws/2004/09/transfer/DeleteResponse">http://schemas.xmlsoap.org/ws/2004/09/transfer/DeleteResponse</a>
<a href="http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate">http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate</a>
<a href="http://schemas.xmlsoap.org/ws/2004/09/enumeration/EnumerateResponse">http://schemas.xmlsoap.org/ws/2004/09/enumeration/EnumerateResponse</a>
<a href="http://schemas.xmlsoap.org/ws/2004/09/enumeration/Pull">http://schemas.xmlsoap.org/ws/2004/09/enumeration/Pull</a>
<a href="http://schemas.xmlsoap.org/ws/2004/09/enumeration/PullResponse">http://schemas.xmlsoap.org/ws/2004/09/enumeration/PullResponse</a>
<a href="http://schemas.xmlsoap.org/ws/2004/09/enumeration/Renew">http://schemas.xmlsoap.org/ws/2004/09/enumeration/Renew</a>
<a href="http://schemas.xmlsoap.org/ws/2004/09/enumeration/RenewResponse">http://schemas.xmlsoap.org/ws/2004/09/enumeration/RenewResponse</a>
<a href="http://schemas.xmlsoap.org/ws/2004/09/enumeration/GetStatus">http://schemas.xmlsoap.org/ws/2004/09/enumeration/GetStatus</a>
<a href="http://schemas.xmlsoap.org/ws/2004/09/enumeration/GetStatusResponse">http://schemas.xmlsoap.org/ws/2004/09/enumeration/GetStatusResponse</a>
<a href="http://schemas.xmlsoap.org/ws/2004/09/enumeration/Release">http://schemas.xmlsoap.org/ws/2004/09/enumeration/Release</a>
<a href="http://schemas.xmlsoap.org/ws/2004/09/enumeration/ReleaseResponse">http://schemas.xmlsoap.org/ws/2004/09/enumeration/ReleaseResponse</a>
<a href="http://schemas.xmlsoap.org/ws/2004/09/enumeration/EnumerationEnd">http://schemas.xmlsoap.org/ws/2004/09/enumeration/EnumerationEnd</a>
<a href="http://schemas.xmlsoap.org/ws/2004/08/eventing/Subscribe">http://schemas.xmlsoap.org/ws/2004/08/eventing/Subscribe</a>
<a href="http://schemas.xmlsoap.org/ws/2004/08/eventing/SubscribeResponse">http://schemas.xmlsoap.org/ws/2004/08/eventing/SubscribeResponse</a>
<a href="http://schemas.xmlsoap.org/ws/2004/08/eventing/Renew">http://schemas.xmlsoap.org/ws/2004/08/eventing/Renew</a>
<a href="http://schemas.xmlsoap.org/ws/2004/08/eventing/RenewResponse">http://schemas.xmlsoap.org/ws/2004/08/eventing/RenewResponse</a>
<a href="http://schemas.xmlsoap.org/ws/2004/08/eventing/GetStatus">http://schemas.xmlsoap.org/ws/2004/08/eventing/GetStatus</a>
<a href="http://schemas.xmlsoap.org/ws/2004/08/eventing/GetStatusResponse">http://schemas.xmlsoap.org/ws/2004/08/eventing/GetStatusResponse</a>
<a href="http://schemas.xmlsoap.org/ws/2004/08/eventing/Unsubscribe">http://schemas.xmlsoap.org/ws/2004/08/eventing/Unsubscribe</a>
<a href="http://schemas.xmlsoap.org/ws/2004/08/eventing/UnsubscribeResponse">http://schemas.xmlsoap.org/ws/2004/08/eventing/UnsubscribeResponse</a>
<a href="http://schemas.xmlsoap.org/ws/2004/08/eventing/SubscriptionEnd">http://schemas.xmlsoap.org/ws/2004/08/eventing/SubscriptionEnd</a>
<a href="http://schemas.dmtf.org/wbem/wsman/1/wsman/Events">http://schemas.dmtf.org/wbem/wsman/1/wsman/Events</a>
<a href="http://schemas.dmtf.org/wbem/wsman/1/wsman/Heartbeat">http://schemas.dmtf.org/wbem/wsman/1/wsman/Heartbeat</a>
<a href="http://schemas.dmtf.org/wbem/wsman/1/wsman/DroppedEvents">http://schemas.dmtf.org/wbem/wsman/1/wsman/DroppedEvents</a>
<a href="http://schemas.dmtf.org/wbem/wsman/1/wsman/Ack">http://schemas.dmtf.org/wbem/wsman/1/wsman/Ack</a>
<a href="http://schemas.dmtf.org/wbem/wsman/1/wsman/Event">http://schemas.dmtf.org/wbem/wsman/1/wsman/Event</a>



# Referências

---

1. Case, J., et al., *RFC1067 - Simple Network Management Protocol*. 1988.
2. DMTF. *WBEM*. 2008 [acedido 04/Junho/2008]; Disponível em: <http://www.dmtf.org/standards/wbem/>.
3. Allen, P.C., et al., *Web Services for Management (WS-Management) Version 1.0.0a*. 2006.
4. McCloghrie, K. and M. Rose, *RFC1066 - Management Information Base for network management of TCP/IP-based internets*. 1988.
5. Rose, M. and K. McCloghrie, *RFC1065 - Structure and Identification of Management Information for TCP/IP-based Internets*. 1988.
6. Interconnection, I.p.s.-O.S., *Specification of Abstract Syntax Notation One (ASN.1)*. International Organization for Standardization. International Standard 8824. 1987.
7. McCloghrie, K. and M. Rose, *RFC1156 - Management Information Base for Network Management of TCP/IP-based internets*. 1990.
8. McCloghrie, K. and M. Rose, *RFC1213 - Management Information Base for Network Management of TCP/IP-based internets: MIB-II*. 1991.
9. Mills, D.L., *RFC904 - Exterior Gateway Protocol formal specification*. 1984.
10. Rose, M. and K. McCloghrie, *RFC1155 - Structure and Identification of Management Information for TCP/IP-based Internets*. 1990.
11. The ASN.1 Consortium, I. 2004 [acedido 27/8/2008]; Disponível em: <http://www.asn1.org/index.htm>.
12. Case, J., et al., *RFC1157 - A Simple Network Management Protocol (SNMP)*. 1990.
13. Waldbusser, S., *RFC2819 - Remote Network Monitoring Management Information Base*. 2000.
14. Case, J., et al., *RFC 1901 - Introduction to Community-based SNMPv2*. 1996.
15. Harrington, D., R. Presuhn, and B. Wijnen, *RFC 3411 - An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*. 2002.
16. WINSITPRO. *What is WBEM?* 1998 [acedido 04/Junho/2008]; Disponível em: <http://www.windowsitpro.com/Articles/ArticleID/3568/3568.html>.
17. W3C, W.W.W.C.-. *HTTP - Hypertext Transfer Protocol*. 2003 [acedido 15/Abr/2008]; Disponível em: <http://www.w3.org/Protocols/>.
18. SSL. *What is SSL?* 2006 [acedido 25/Abr/2008]; Disponível em: <http://www.ssl.com/>.

19. W3C, W.W.W.C.-. *Extensible Markup Language (XML)*. 2003 [acedido 04/Mar/2008]; Disponível em: <http://www.w3.org/XML/>.
20. WBEMSOLUT. *CIM*. 2003 [acedido 04/Junho/2008]; Disponível em: <http://www.wbemsolutions.com/tutorials/CIM/cim.html>.
21. DMTF. *CIM*. 2008 [acedido 04/Junho/2008]; Disponível em: <http://www.dmtf.org/standards/cim/>.
22. WBEMSOLUT. *WBEM Operations*. 2003 [acedido 04/Junho/2008]; Disponível em: <http://www.wbemsolutions.com/tutorials/CIM/wbem-operations.html>.
23. Bray, T., et al., *Extensible Markup Language (XML) 1.0 (Fourth Edition)*. *W3C Recommendation 16 August 2006, edited in place 29 September 2006*. 2006.
24. Gao, S.S., C.M. Sperberg-McQueen, and H.S. Thompson, *W3C XML Schema Definition Language (XSDL) 1.1 Part 1: Structures*. *W3C Working Draft 30 August 2007*. 2007.
25. Peterson, D., et al., *XML Schema 1.1 Part 2: Datatypes*. *W3C Working Draft 17 February 2006*. 2006.
26. W3C, W.W.W.C.-. *XML Schema (XSD)*. 2008 [acedido 01/Junho/2008]; Disponível em: <http://www.w3.org/XML/Schema>.
27. WBEMSOLUT. *DMTF Glossary*. 2003 [acedido 04/Junho/2008]; Disponível em: <http://www.wbemsolutions.com/tutorials/CIM/glossary.html>.
28. Lee, S.-J., et al., *Design of a WBEM-based Management System for Ubiquitous Computing Servers* 2004.
29. Kirk, M. *Systems Management: Common Manageability Programming Interface (CMPI)*. 2003 [acedido 04/Junho/2008]; Disponível em: [http://www.openpegasus.org/uploads/40/4031/CMPI\\_Specification\\_13.pdf](http://www.openpegasus.org/uploads/40/4031/CMPI_Specification_13.pdf).
30. CORBA. *Common Object Request Broker Architecture*. [acedido 14/Maio/2008]; Disponível em: <http://www.corba.org>.
31. IBM. *Tivoli*. [acedido 04/Junho/2008]; Disponível em: <http://www-306.ibm.com/software/tivoli/>.
32. CISCO. *CiscoWorks LMS*. [acedido 04/Junho/2008]; Disponível em: [http://www.cisco.com/en/US/products/sw/cscowork/ps2425/products\\_data\\_sheet09186a00800a9e97.html](http://www.cisco.com/en/US/products/sw/cscowork/ps2425/products_data_sheet09186a00800a9e97.html).
33. SUN. *Solaris WBEM Services*. 2008 [acedido 04/Junho/2008]; Disponível em: <http://www.sun.com/software/solaris/wbem/>.
34. HP. *HP WBEM Solutions*. 2008 [acedido 04/Junho/2008]; Disponível em: <http://h71028.www7.hp.com/enterprise/cache/9920-0-0-225-121.html>.
35. MICROSOFT. *Windows Management Instrumentation*. 2008 [acedido 04/Junho/2008]; Disponível em: <http://msdn2.microsoft.com/en-us/library/aa394582.aspx>.
36. MICROSOFT. *Microsoft Operations Manager 2005 Product Overview*. 2004 [acedido 04/Junho/2008]; Disponível em: <http://technet.microsoft.com/en-us/opsmgr/bb498244.aspx>.

37. MICROSOFT. *Using Microsoft Operations Manager*. 2008 [acedido 04/Junho/2008]; Disponível em: <http://msdn2.microsoft.com/en-us/magazine/cc188704.aspx>.
38. PEGASUS. *The Open Pegasus*. [acedido 04/Junho/2008]; Disponível em: <http://www.openpegasus.org/>.
39. WBEMServices. *WBEM Services*. 2005 [acedido 04/Junho/2008]; Disponível em: <http://wbemservices.sourceforge.net/>.
40. *OpenWBEM*. [acedido 04/Junho/2008]; Disponível em: <http://www.openwbem.org/>.
41. SBLIM. *Standards Based Linux Instrumentation*. 2008 [acedido 04/Junho/2008]; Disponível em: <http://sblim.wiki.sourceforge.net/>.
42. SMITH, J.T. *IBM initiates Open Source project, seeks developers*. 2001 [acedido 04/Junho/2008]; Disponível em: [<http://www.linux.com/articles/14484>.
43. OPENSOURCE. *The Open Source Definition*. 2006 [acedido 04/Junho/2008]; Disponível em: <http://www.opensource.org/>.
44. W3C, W.W.W.C.-. *Web Services Activity*. 2002 [acedido 04/Junho/2008]; Disponível em: <http://www.w3.org/2002/ws/>.
45. Pavlou, G., *On the Evolution of Management Approaches, Frameworks and Protocols: A Historical Perspective*. 2007.
46. Curbera, F., et al., *Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI*. Internet Computing, IEEE, 2002. **6**(2): p. 86-93.
47. Neisse, R., et al. *Implementation and bandwidth consumption evaluation of SNMP to Web services gateways*. in *Network Operations and Management Symposium, 2004. NOMS 2004. IEEE/IFIP*. 2004.
48. Mitra, N. and Y. Lafon, *SOAP Version 1.2 Part 0: Primer (Second Edition)*. W3C Recommendation 27 April 2007. 2007.
49. Chinnici, R., et al., *Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language*. W3C Recommendation 26 June 2007. 2007.
50. Clement, L., et al., *UDDI Version 3.0.2. UDDI Spec Technical Committee Draft, Dated 20041019*. 2004.
51. Gottschalk, K., et al., *Introduction to Web services architecture*. IBM Systems Journal, 2002. **41**(2): p. 8p.
52. OASIS. *Universal Description, Discovery and Integration*. [acedido 03/Out/2008]; Disponível em: <http://www.uddi.org/>.
53. Martin-Flatin, J.P. and P.A. Doffoel, *Web Services for Integrated Management: a Case Study*. Technical Report DataTAG-2003-1, 2003.
54. Moura, G.C.M., et al., *On the Performance of Web Services Management Standards - An Evaluation of MUWS and WS-Management for Network Managemen*. Integrated Network Management, 2007. IM '07. 10th IFIP/IEEE International Symposium on, 2007: p. 459-468.

55. Vambenepe, W. *Web Services Distributed Management: Management Using Web Services (MUWS 1.0) Part 1*. 2005 [acedido 03/Out/2008]; Disponível em: <http://docs.oasis-open.org/wsdm/2004/12/wsdm-muws-part1-1.0.pdf>.
56. Vambenepe, W. *Web Services Distributed Management: Management Using Web Services (MUWS 1.0) Part 2*. 2005 [acedido 03/Out/2008]; Disponível em: <http://docs.oasis-open.org/wsdm/2004/12/wsdm-muws-part2-1.0.pdf>.
57. Sedukhin, I. *Web Services Distributed Management: Management of Web Services (WSDM-MOWS) 1.0*. 2005 Março 2005 [acedido 03/Out/2008]; Standard]. Disponível em: <http://docs.oasis-open.org/wsdm/2004/12/wsdm-mows-1.0.pdf>
58. DMTF, *DMTF Fact Sheet: WS-Management*. 2006.
59. Allen, P.C., et al., *Web Services for Management (WS-Management) Specification Version 1.0.0*. 2008.
60. Arora, A., et al., *Web Services for Management (WS-Management)* 2005.
61. Lima, W.Q.d., et al., *Evaluating the Performance of SNMP and Web Services Notifications*. 2006.
62. Hollenbeck, S., *RFC 3749 - Transport Layer Security Protocol Compression Methods (Deflate)*. 2004.
63. *SNMPBULKWALK*. [acedido 04/Jun/2008; Disponível em: <http://www.net-snmp.org/docs/man/snmpbulkwalk.html>].
64. *AGENT++*. [acedido 04/Jun/2008; Disponível em: <http://www.agentpp.com/>].
65. *Openwsman*. [acedido 04/Junho/2008]; Disponível em: <http://www.openwsman.org/project/openwsman>.
66. DMTF, *WS-Management — CIM Binding*. 2006.
67. DMTF, *WS-CIM Mapping Specification*. 2006.
68. D. Durham, E., et al., *RFC2748 - The COPS (Common Open Policy Service) Protocol*. 2000.
69. R. Enns, E., *RFC4741 - NETCONF Configuration Protocol*. 2006.
70. Calhoun, P., et al., *RFC3588 - Diameter Base Protocol*. 2003.