Desenvolvimento de um Agente de Monitoração para Redes TCP/IP

Fernando Manuel Pipa, José Luís Oliveira, Joaquim Arnaldo Martins

***Resumo*- Neste trabalho é apresentado o desenvolvimento de um agente de monitoração. O agente é construído sobre uma plataforma SNMP que permite o desenvolvimento de aplicações de gestão que utilizam o *Simple Network Management Protocol* (SNMP) como protocolo de comunicação. No agente é implementada a *Remote Network Monitoring Management Information Base* (RMON MIB) que reúne um conjunto de objectos de gestão relacionados com a informação de monitoração.**

No intuito de permitir ao agente um modo de funcionamento local e remoto, este é dotado de algumas capacidades de visualização e manipulação da informação de gestão, de modo a tornar possível a monitoração quer localmente no agente, quer remotamente no gestor. Para isso é apresentado um módulo de visualização gráfica, no qual o acesso à informação é feita através de uma *interface* genérica por forma a permitir a sua portabilidade para qualquer aplicação de gestão.

***Abstract*- This work presents the development of a monitoring agent. This agent is built over a SNMP platform which allows the development of management applications that use the Simple Network Management Protocol (SNMP) as the communication protocol. The agent implements the Remote Network Monitoring Management Information Base (RMON MIB) which comprises a set of objects related with monitoring information.**

In order to allow local or remote monitoring operations, the agent is enhanced with some visualisation and operation capabilities. For that, it is supplied with a graphical module which collects object's values through an interface, in such a way that it can be portable to different management applications.

I. INTRODUÇÃO*

Actualmente, os utilizadores de serviços de informação necessitam de um acesso, cada vez mais rápido, a uma grande variedade de informação e recursos. Esses recursos podem estar distribuídos por diversas redes e utilizar diferentes sistemas de operação. No sentido de facilitar a partilha de informação e recursos através das redes, surgiram algumas comunidades que têm desenvolvido um conjunto de normas para os protocolos de comunicação.

Uma vez criadas as condições que tornaram possível as comunicações ao longo das diversas redes, o utilizador é confrontado com o problema de gestão desta colecção de redes e sistemas, como uma única rede de comunicações. O utilizador tem de ser capaz de monitorar a rede por forma a obter informação em tempo-real sobre as características de desempenho e tráfego, diagnosticar problemas de comunicação e reconfigurar a rede de acordo com as condições exigidas pela mesma. É neste sentido que surge aquilo que se designa por Gestão de Redes. Normalmente, a gestão abrange recursos que são constituídos por equipamentos de diversos fabricantes levando à necessidade de um sistema de gestão integrado.

Uma gestão integrada permite que um gestor desempenhe a sua função de um modo mais eficiente, através de um melhor acesso à informação (que pode ser obtida de uma maneira coerente a partir de todos os elementos do sistema), resultando num poder de decisão mais correcto, numa aplicação de controlos mais acertada e consequentemente num aumento da qualidade de serviço disponível ao utilizador. Para esta integração, contribui a produção de normas de gestão.

Assim, as normas representam a possibilidade de tornar real a integração de um sistema de gestão, permitindo que os fabricantes produzam equipamento de acordo com uma norma comum e que outras entidades desenvolvam serviços de gestão capazes de abranger os equipamentos dos diversos fabricantes.

As actividades de gestão requerem uma base de informação e um conjunto de ferramentas de manipulação de dados. A base de dados tem por finalidade armazenar a informação relativa ao estado da rede e à configuração do sistema, o desempenho actual e passado, registos de problemas, os parâmetros de segurança e a informação de contabilização.

A identificação dos elementos envolvidos num sistema distribuído, foi algo de importante na definição e normalização da informação de gestão, de modo a permitir que tais elementos possam ser geridos. Estes, tomam a designação de objectos de gestão, representando uma abstracção dos recursos existentes na rede. De um modo geral, os elementos de um sistema que necessitam de ser geridos, podem ser representados por objectos de gestão.

O objectivo deste trabalho consiste em definir e implementar objectos de gestão na arquitectura Internet (TCP/IP), com o desenvolvimento de um sistema integrado de monitoração de uma rede local. Este sistema

* A realização deste trabalho enquadrou-se no projecto PMCT/C/TIT/458.90 tendo sido financiado pela JNICT.

baseia-se num agente de monitoração remota que implementa a RMON MIB.

II. SNMP

O *Simple Network Management Protocol* (SNMP) é um protocolo de comunicação utilizado na troca de informação entre as entidades de gestão na arquitectura Internet. Este protocolo é constituído por três elementos que estão definidos nos *Request for Comments* (RFCs) emitidos pela *Internet Activity Board* (IAB). Assim, a estrutura da informação de gestão (SMI - *Structure of Management Information*) está especificada no RFC 1155 [1]. Este, apresenta o modelo administrativo e a organização relativa à definição dos objectos de gestão. A SMI define uma sintaxe a baixo nível para cada objecto de gestão, baseada no ASN.1[1] [2], bem como a informação relativa ao estado do objecto, os direitos de acesso e directivas para a construção do nome do objecto. A SMI assegura uma informação com um formato comum quando o gestor está a controlar equipamento de diferentes fabricantes, desde que os seus produtos implementem variáveis SNMP.

O segundo elemento refere-se à base de informação de gestão (MIB - *Management Information Base*) especificado no RFC 1213. Esta base dispõe de uma lista dos recursos da rede (objectos de gestão) que podem ser interrogados a qualquer agente, bem como define a informação que deve ser devolvida por este, em consequência de uma acção de pedido relativa a um objecto. A MIB é implementada no agente.

Os utilizadores do SNMP podem definir objectos que, por serem específicos, são referidos como variáveis estendidas. Estas, podem ser colocadas no domínio público, permitindo que alguém possa desenvolver um gestor capaz de as controlar.

A SMI define uma estrutura da MIB em árvore, tal como se demonstra na fig. 1.

A cada nó da árvore é atribuído um número e um nome simbólico. Deste modo, a representação de um objecto da MIB pode ser feito tanto na forma numérica, como na forma do nome. A representação numérica de um objecto é feita concatenando todos os números dos nós que se encontram no caminho, desde a raiz até ao objecto em causa.

O terceiro elemento do SNMP é definido pelo RFC 1157 que descreve o próprio SNMP. Enquanto que os documentos da SMI e da MIB definem os objectos de gestão, o SNMP define o modo como a estação de gestão adquire a informação dos objectos geridos. O SNMP utiliza como protocolo de transporte o *User Datagram Protocol over IP* (UDP/IP), o qual assume uma transmissão fiável não fazendo qualquer tipo de verificação de erros. As mensagens SNMP podem ser

[1]ASN.1 é uma linguagem OSI utilizada para descrever a sintaxe abstracta. Permite representar a informação num formato que é percetível para o utilizador.

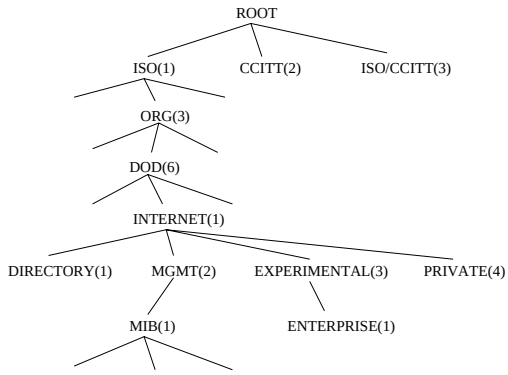


Fig. 1 - Estrutura da MIB em árvore.

transmitidas sobre qualquer mecanismo de transporte, desde que este permita uma comunicação bi-direccional feita com base num endereçamento. Na fig. 2 encontra-se representada a arquitectura SNMP.

As mensagens SNMP são representadas na notação ASN.1.

O SNMP utiliza um processo de *polling* no acesso à informação de gestão, não existindo uma ligação lógica entre o gestor e os objectos.

O SNMP é basicamente um protocolo do tipo pergunta-resposta, tendo 5 mensagens: GET-REQUEST, GET-NEXT-REQUEST, GET-RESPONSE, SET-REQUEST, e TRAP. A mensagem GET-REQUEST é utilizada pelo gestor para obter informação relativa a um ou mais objectos de uma MIB implementada num agente. O GET-NEXT-REQUEST tem uma função idêntica à do GET-REQUEST, permitindo percorrer a árvore da MIB pedindo o objecto hierarquicamente a seguir ao indicado no pedido. O GET-RESPONSE é usado pelo agente para retornar a informação de gestão pedida pelo gestor. O SET-REQUEST é utilizado pelo gestor para alterar o valor de um objecto da MIB. O TRAP desvia-se um pouco da filosofia das primitivas anteriores. Esta, é uma mensagem não solicitada, assíncrona, do agente para o gestor indicando que foi ultrapassado um limiar de decisão.

Rapidamente, o SNMP adquiriu uma grande popularidade, tendo sido implementado em diversos

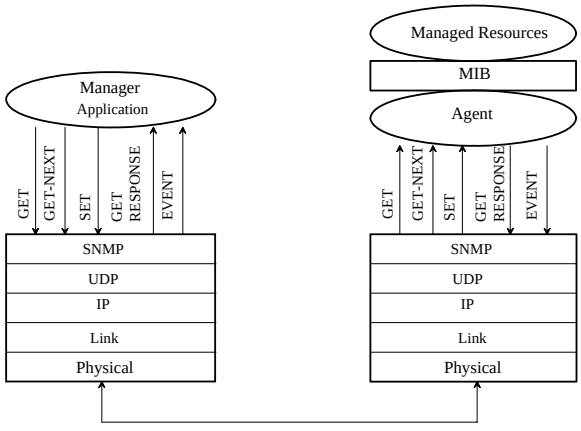


Fig. 2 - Comunicação SNMP.

dispositivos, tais como PCs, Macs, *bridges*, *routers*, *workstations*. Apesar disso, este possui algumas limitações e problemas. Uma das principais lacunas do SNMP refere-se à segurança, pelo que a *Internet Engineering Task Force* (IETF) aprovou em meados do ano de 1992 o *Secure SNMP* [3]. Esta foi a primeira alteração mais significativa que se efectuou no protocolo. No sentido de resolver outros problemas, foi proposta uma revisão ao SNMP, que se designou por *Simple Network Management Protocol version 2* (SNMPv2) [4].

O desenvolvimento do SNMPv2, tal como o SNMP, teve em conta a minimização do custo e da complexidade dos agentes, pondo o processamento de gestão nas estações centrais. Uma vez que, numa rede, existem mais agentes do que gestores, é fundamental que os primeiros consumam o mínimo dos recursos da rede. Um dos princípios do SNMP e do SNMPv2 é colocar a maior parte do processamento nas estações gestoras, ficando o agente com o menor processamento possível. Esta regra permite que a máquina que suporta o processo agente fique com tempo disponível para executar as suas próprias tarefas.

O SNMPv2 melhora o funcionamento das funções de controlo, através do comando SET. Por um lado, introduz um modo de *lock*, que permite à estação gestora configurar um recurso, sem que o seu acesso seja interrompido. Por outro, o SNMPv2 define os procedimentos para criar, modificar e remover linhas de tabelas.

O SNMPv2 foi dotado de uma nova capacidade no sentido de aumentar o seu desempenho relativamente ao SNMP. Com o auxílio do novo comando GETBULK, a estação gestora pode receber simultaneamente informação de várias variáveis. Este mecanismo produz uma baixa taxa de utilização dos recursos da rede e é fácil de implementar.

O SNMPv2 define novos tipos de dados incluindo contadores de 64 bits. Os contadores do SNMP são de 32 bits, mas com o crescimento das redes e o consequente aumento da complexidade, estes tornaram-se pequenos em determinados ambientes.

O SNMPv2 introduz um novo tipo de dados que se refere aos endereços OSI, permitindo assim um alargamento da gestão às redes OSI.

Se um agente SNMP não dispõe de informação relativa a uma parte do pedido feito pela estação gestora, porque a MIB que contém tais objectos não está implementada, esse pedido será completamente rejeitado. Este problema foi resolvido no SNMPv2 com a introdução de condições de excepção. Um agente SNMPv2 pode responder apenas a uma parte do pedido da estação de gestão, devido à falta de informação na MIB, pelo que o agente assinalará a falha com uma "excepção", retornando o resto da informação.

O SNMPv2 contém um vasto conjunto de códigos de erros, que podem ser utilizados pelo agente para justificar a sua incapacidade de resposta a certos pedidos da estação gestora. Tais mensagens de erro fornecem às aplicações de gestão mais informação relativa às falhas dos agentes, de modo que podem determinar se a falha é permanente ou temporária e tomar decisões para corrigir essas falhas.

Ao contrário do SNMP, o SNMPv2 permite a comunicação entre as estações gestoras. Em primeiro lugar, o SNMPv2 dispõe de transferência de informação confirmada entre os gestores, incluindo o envio de notificações. Isto é conseguido com o comando INFORM, no qual um gestor envia informação a outro, com pedido de confirmação da mensagem recebida. Por outro lado, foi definida uma SNMPv2 MIB [5] que controla a geração e a transmissão de eventos entre as estações gestoras.

A capacidade de comunicação entre gestores permite a construção de um sistema de gestão hierárquico. Num sistema deste tipo, um gestor de uma LAN, pertencente a um nível intermédio na estrutura hierárquica, é responsável pelo *polling* aos recursos pertencentes a essa LAN. Numa situação de detecção de uma falha, a estação gestora intermédia envia uma notificação à estação gestora do nível hierárquico superior. Este mecanismo reduz o tráfego de gestão global mantendo um bom nível de fiabilidade de gestão, que se baseia no *polling* dos agentes.

Do ponto de vista da construção e desenvolvimento de MIBs, o SNMPv2 permite adicionar anotações às MIBs, sob a forma de *macros*, de modo a serem interpretadas pela máquina.

O SNMPv2 pode utilizar diversos protocolos de transporte, tais como, o TCP/IP, o IPX, o AppleTalk e o CLNP (*Connection-less Network Protocol*) do OSI. Para tal, foram estabelecidas especificações que descrevem o modo de colocar mensagens SNMPv2 nessas pilhas [6].

As estações de gestão podem, com o auxílio de agentes *proxy*, dialogar com outros protocolos de rede que não sejam TCP/IP.

Um dos principais objectivos do desenvolvimento do SNMPv2 foi a compatibilidade com o SNMP, sempre que possível. Relativamente à transição do SNMP para o SNMPv2 são sugeridos dois caminhos: a utilização de gestores com duas linguagens e o uso de agentes *proxy*. Uma estação gestora com dupla linguagem implementa o SNMP e o SNMPv2, sendo a opção do protocolo feita de acordo com o tipo de agente (SNMP ou SNMPv2) envolvido no processo de gestão. Por outro lado, pode-se utilizar um agente *proxy* para converter o formato de uma mensagem noutro formato. Por exemplo, um gestor SNMP pode converter mensagens SNMPv2 através de um agente *proxy*.

Além disto, o SNMPv2 utiliza o conceito de *Party MIB*, introduzido pelo *Secure* SNMP, que suporta múltiplos agentes numa única plataforma (esta foi uma das principais inovações do SNMPv2) [7].

Uma estação gestora, com uma *Party* MIB, pode comunicar com vários agentes através de um endereço IP, desde que esses agentes estejam registados como *parties* na estação gestora. Por outras palavras, o SNMPv2 permite mais do que um agente numa plataforma computacional. Resumindo, a *Party* MIB permite que

todos os elementos constituintes de um sistema computacional (sistema operativo, dispositivos de armazenamento, ou mesmo aplicações) estejam equipados com os seus próprios agentes.

III. OBJECTOS DE GESTÃO

Esta secção pretende dar uma introdução teórica sobre os objectos de gestão na arquitectura TCP/IP. Começa-se por apresentar a sintaxe abstracta que é utilizada na descrição dos dados de gestão, passando depois à definição da estrutura da informação.

A. Sintaxe Abstracta

Ao nível da camada de aplicação, as estruturas de dados que são trocadas entre as entidades protocolares são bastante complexas. Por essa razão, é necessário introduzir um formalismo na descrição dessas estruturas. Este formalismo é designado por sintaxe abstracta, permitindo definir dados sem atender às estruturas e restrições que são impostas pela máquina, na qual se pretende efectuar a implementação. Deste modo, é possível definir informação de uma forma aberta, isto é, independentemente da arquitectura a que se destina. Para este efeito, a Comunidade Internet optou por uma linguagem da ISO, nomeadamente a *Abstract Syntax Notation One* (ASN.1).

O emprego da ASN.1 tem dois objectivos distintos, um consiste na definição dos formatos dos PDUs (*Protocol Data Units*) trocados pelo protocolo de gestão, o outro na definição dos objectos geridos. Isto significa que a descrição das estruturas de dados (utilizadas na comunicação ao nível dos protocolos) e da informação de gestão nelas contidas, é feita com base na sintaxe abstracta.

A codificação das estruturas de dados especificadas pela ASN.1, é estabelecida pelas *Basic Encoding Rules* (BER) [8].

A implementação da ASN.1 na sua totalidade resulta numa complexidade tal, que a Internet optou por adoptar apenas um subconjunto das suas capacidades.

B. Módulos

Um módulo é um conjunto de descrições ASN.1 relativas a um tema comum. A sua sintaxe é a seguinte:

```
<<module>> DEFINITIONS ::= BEGIN
<<linkage>>
<<declarations>>
END
```

O termo <<module>> indica o nome do módulo. O <<linkage>> permite relacionar este módulo com outros, fazendo o EXPORT/IMPORT de definições para/de outros módulos. As definições ASN.1 encontram-se nas <<declarations>>.

A ASN.1 define três tipos de objectos:

- *tipos*, que definem novas estruturas de dados (o nome começa com letra maiúscula, p. e. Gauge);
- *valores*, que representam as instâncias de um tipo (o nome começa com letra minúscula, p. e. internet);
- *macros*, que são utilizadas para alterar a actual gramática da linguagem ASN.1 (o nome consiste em letras maiúsculas, p. e. OBJECT-TYPE).

C. Estrutura da Informação de Gestão

A estrutura da informação de gestão (SMI - *Structure of Management Information*) [1] descreve as estruturas e o esquema de identificação para a definição da informação de gestão. Dela, fazem parte as descrições do modelo de informação de um objecto com os tipos que lhe estão associados. As descrições formais da estrutura da informação são feitas com base linguagem ASN.1.

Os objectos de gestão são acessíveis através de uma base de informação virtual, designada por *Management Information Base* ou MIB, sendo definidos através da *Abstract Sintax Notation One* (ASN.1).

Cada tipo de objecto tem um nome, uma sintaxe e uma codificação. O nome é representado unicamente como sendo um *OBJECT IDENTIFIER*, o qual é atribuído de uma forma administrativa. A sintaxe define a estrutura de dados abstracta, relativa ao tipo de objecto em causa. A codificação indica como são representadas as instâncias daquele tipo de objecto, aplicando as *Basic Encoding Rules* à respectiva sintaxe.

Cada objecto de gestão é descrito através de uma macro ASN.1 definida na SMI. A macro é identificada como OBJECT-TYPE . Esta permite representar os principais aspectos do tipo de objecto, de uma maneira formal.

No sentido de tornar as descrições da MIB mais concisas e informativas, foi feito um melhoramento à SMI com a edição do RFC1212 [9].

D. Objectos Tabelados

O SNMP permite executar operações sobre objectos escalares. No entanto, para os programadores de aplicações de gestão é conveniente ter uma visão, em termos conceptuais, dos objectos da MIB estruturados sob a forma de tabelas. Cada uma dessas tabelas conceptuais, possui zero ou mais linhas, tendo cada linha um ou mais objectos que se designam por objectos tabelados. Esta conceptualização é formalizada pela macro OBJECT-TYPE utilizada na definição de objectos que correspondem a uma tabela.

E. Definição de Objectos

A definição da MIB consiste em duas partes: uma parte textual, na qual os objectos estão divididos por grupos e um módulo da MIB, no qual os objectos são descritos somente em termos da macro ASN.1 designada por OBJECT-TYPE, que está definida na SMI:

```
IMPORTS
  ObjectName
    FROM RFC1155-SMI
  DisplayString
    FROM RFC1158-MIB;

OBJECT-TYPE MACRO::=
BEGIN
  TYPE NOTATION ::=
    "SYNTAX" type(ObjectSyntax)
    "ACESS" Access
    "STATUS" Status
    DescrPart
    ReferPart
    IndexPart
    DefValPart
  VALUE NOTATION ::= value (VALUE
ObjectName)

Access ::= "read-only"
    | "read-write"
    | "write-only"
    | "not-accessible"
Status ::= "mandatory"
    | "optional"
    | "obsolete"
    | "deprecated"
DescrPart ::= "DESCRIPTION" value
(description Display String)
    | empty
ReferPart ::= "REFERENCE" value
(reference DisplayString)
    | empty
IndexPart ::= "INDEX" "{" IndexTypes "}"
    | empty
IndexTypes ::= IndexType
    | IndexType "," IndexType
IndexType ::= value (indexobject
ObjectName)
    | type (indextype)
DefValPart ::= "DEFVAL" "{" value
(defvalue ObjectSyntax) "}"
    | empty
END

IndexSyntax ::=
  CHOICE {
    number
      INTEGER (0 .. MAX),
    string
      OCTET STRING,
    object
      OBJECT IDENTIFIER,
    address
      NetworkAddress,
    ipAddress
      IpAddress
  }
```

O significado de cada clausula é apresentado em seguida.

SYNTAX: a sintaxe do objecto define o tipo de dados que modela o objecto.

ACCESS: o nível de acesso a um objecto é um dos seguintes:

- read-only, as instâncias do objecto podem ser lidas, mas não podem ser alteradas;
- read-write, as instâncias do objecto podem ser lidas e alteradas;
- write-only, as instâncias do objecto podem ser alteradas, mas não se podem ler;
- not-accessible, as instâncias do objecto não podem ser lidas nem alteradas.

STATUS: os requisitos de implementação de um objecto podem ser um dos seguintes:

- mandatory, os agentes de gestão devem implementar o objecto;
- optional, os agentes podem implementar o objecto;
- obsolete, os agentes não necessitam de implementar o objecto.

DESCRIPTION: é opcional, contendo uma definição textual do tipo de objecto.

REFERENCE: é opcional e contém uma referência textual para um objecto definido noutro módulo da MIB.

INDEX: pode ou não estar presente. Tem como finalidade definir a informação de identificação da instância, para um tipo de objecto. Se a cláusula INDEX não está presente e o tipo de objecto corresponde a um objecto não tabelado, então as instâncias desse objecto são identificadas adicionando um sub-identificador de zero ao nome desse objecto. No caso dos objectos tabelados, esta cláusula tem de estar presente, no sentido de indicar o modo como deve ser feita a identificação das instâncias desses objectos.

DEFVAL: O emprego desta cláusula é opcional. Esta define um valor que é atribuído por defeito quando é criada uma instância de um objecto.

Nome: o valor (nome) do objecto é, segundo a macro, do tipo ObjectName. Deste modo, os nomes dos objectos são atribuídos de acordo com os OBJECT IDENTIFIERs

IV. MIB PARA MONITORAÇÃO REMOTA

Em finais dos anos oitenta, foi introduzida uma nova classe de recursos, no âmbito da gestão de redes, identificada como monitores distribuídos. Estes consistem numa plataforma de *hardware* dedicada, que permite uma monitoração da rede sem a necessidade de apoio de mais nenhum recurso. Os monitores distribuídos fornecem uma variedade de ferramentas para detecção de falhas e aumento do desempenho, através da "escuta" de todos os pacotes que circulam na rede, contabilizando as estatísticas das actividades de envio e recepção, efectuadas sobre a rede.

A *Remote Network Monitoring MIB* (RMON MIB) [10] é constituída por um conjunto de objectos que representam os diversos aspectos da informação de monitoração.

A RMON MIB dispõe de uma grande quantidade de objectos que podem ser aplicados à gestão de qualquer tipo de rede, havendo contudo, alguns que são específicos à gestão de redes Ethernet. No entanto, a estrutura desta MIB permite também definir objectos específicos a outros tipos de rede, nomeadamente Token Ring [11], prevendo-se que em futuras versões da RMON estejam definidas extensões para outras redes, tais como a FDDI.

Com a monitoração remota pretende-se suportar os seguintes casos:

- Esta MIB permite configurar o monitor de modo a realizar diagnósticos e recolher estatísticas, mesmo em situações em que a comunicação com o gestor não é possível ou não está eficiente. O agente de monitoração pode notificar a estação gestora da ocorrência de condições excepcionais. Assim, as falhas, o desempenho e a informação de configuração podem ser continuamente armazenadas e enviadas ao gestor de uma forma conveniente e eficiente.
- Com base nos recursos disponíveis, o monitor pode diagnosticar continuamente a rede e armazenar a informação de monitoração, de modo a permitir ao gestor a identificação das causas de falhas que tenham ocorrido.
- O monitor pode ser configurado para reconhecer certas condições, normalmente condições de erro, analisando-as continuamente. Quando uma dessas condições é verificada, o evento pode ser registado e as estações gestoras são notificadas.
- O dispositivo de monitoração remota pode valorizar a informação recolhida. Por exemplo, o monitor pode dar ênfase às máquinas que geram mais tráfego ou erros.

A. A Estrutura da MIB

Os objectos da RMON MIB estão distribuídos pelos seguintes grupos:

statistics: contém as estatísticas recolhidas pelo monitor, para cada tipo de *interface*. As estatísticas são armazenadas numa tabela de contadores com informação sobre o número de pacotes, bytes, *broadcasts*, erros, *multicast*s e colisões num segmento da rede.

history: consiste no armazenamento de um número limitado de amostras estatísticas, recolhidas periodicamente da rede.

alarm: recolhe periodicamente amostras estatísticas das variáveis do monitor e compara-as com limiares de decisão previamente configurados. Se o valor de uma variável monitorada ultrapassar o limiar, então é gerado um evento. Para limitar a geração repetida de alarmes, este grupo dispõe de um mecanismo de histerese.

host: contém estatísticas associadas a cada uma das máquinas detectadas na rede. A detecção das máquinas na rede, consiste na análise de uma lista de endereços físicos (fonte e destino) construída a partir de pacotes sem erros.

hostTopN: é utilizado para preparar relatórios que descrevem as máquinas que constam no topo de uma lista, ordenada por uma das estatísticas. Estas são amostras das estatísticas base, durante um intervalo especificado pela estação gestora. O gestor selecciona também o número de máquinas que irão constar no relatório.

matrix: armazena as estatísticas relativas ao diálogo entre duas máquinas. Sempre que é detectada uma nova conversação, o monitor cria uma nova entrada nas suas tabelas.

filter: permite capturar pacotes segundo uma expressão de filtragem arbitrária. Os pacotes filtrados formam uma sequência de informação, também designada por "canal", que pode ser posteriormente capturada ou utilizada para gerar eventos.

capture: permite capturar pacotes que tenham passado através do "canal", isto é, filtrados.

event: controla a geração e a notificação de eventos a partir do monitor.

A implementação destes grupos num agente, é opcional, devendo, no entanto, a implementação de todos os objectos de um determinado grupo ser obrigatória. Pode-se ter vários agentes RMON no mesmo segmento da rede, implementando cada qual uma parte da RMON, no sentido de executarem funções de gestão específicas com melhor desempenho.

V. COMPILADORES DE MIBS

Um compilador de MIBs SNMP é uma ferramenta extremamente útil não só para os autores das MIBs, mas também para os programadores de agentes SNMP e utilizadores de aplicações de gestão SNMP. Para além de verificar a sintaxe da MIB, um compilador pode gerar automaticamente estruturas de dados e código necessário ao agente para implementar uma determinada MIB.

Um compilador é uma ferramenta que traduz um programa escrito numa linguagem (linguagem fonte) para outro equivalente, noutra linguagem (linguagem destino). Tipicamente a linguagem fonte é uma linguagem de alto nível, tal como o Pascal ou C, enquanto que a linguagem destino é uma linguagem de programação a baixo nível, como por exemplo linguagem máquina.

A linguagem fonte para um compilador de MIBs é a linguagem ASN.1 (*Abstract Syntax Notation One*). A ASN.1 não é uma linguagem de programação, mas sim uma linguagem para descrever informação estruturada. A ASN.1 assemelha-se à declaração dos dados de uma linguagem de alto nível. A entrada de um compilador de MIBs é uma colecção de módulos MIB escritos com base num subconjunto da linguagem ASN.1. Estes módulos da MIB contêm definições dos objectos de gestão, que

correspondem à informação disponível nos recursos da rede que pode ser manipulada através do SNMP.

Os compiladores de MIBs geram vários tipos de representação dos objectos de gestão. Algumas destas representações são declarações de estruturas de dados numa linguagem de programação de alto nível, tal como o C, e que podem ser compiladas e ligadas a uma aplicação de gestão (gestor ou agente). Outras, são ficheiros de dados que contêm definições de objectos de gestão, que são lidos para a memória por uma aplicação de gestão, durante a execução do programa. Nalguns casos, os compiladores geram código para auxiliar a implementação das MIBs nos agentes.

Algumas implementações de agentes não possuem informação relativa à árvore da MIB, durante a fase de compilação do agente. Em vez disso, estes agentes constroem a estrutura da árvore, durante a execução, a partir de um ficheiro de dados que possui a definição da MIB.

Como já foi referido, os compiladores de MIBs geram ficheiros de saída com vários formatos. Um formato comum a muitos compiladores, é o formato *mosy*, cujo nome deriva de um compilador muito popular, o MOSY [12]. Um ficheiro com este formato possui uma representação da árvore da MIB, que é utilizada pelas aplicações de gestão para fazer uma correspondência entre os descritores e os respectivos OBJECT IDENTIFIERS. Este ficheiro, especifica também a sintaxe, o acesso e o estado de cada objecto. O exemplo de um extracto de um ficheiro (gerado a partir do RFC1213) com o formato *mosy* pode ser o seguinte:

```
mib-2  mgmt.1
system  mib-2.1
. . .
sysDescr system.1 OctetString read-only
mandatory
sysObjectID system.2 ObjectID read-only
mandatory
. . .
```

Alguns compiladores produzem ficheiros com formatos de versões estendidas do *mosy*.

O MOSY é um compilador de MIBs, do domínio público e do tipo *standalone* (o compilador é um único programa, apenas com essa função), cujo nome corresponde à designação de *Managed Object Syntax-compiler (Yacc-based)*. Este compilador faz parte do pacote de *software* do ISODE (*ISO Development Environment*) e é utilizado na definição de novos objectos. Basicamente, o MOSY lê a descrição de um módulo de objectos de gestão, verificando a sua sintaxe, e produz um ficheiro ASCII contendo as definições equivalentes. Posteriormente, o ficheiro é lido como informação de configuração pelo programa de gestão.

VI. PLATAFORMA SNMP DE DESENVOLVIMENTO E IMPLEMENTAÇÃO DE OBJECTOS DE GESTÃO.

Realizou-se, na Universidade de Aveiro, o estudo e projecto de uma plataforma SNMP de desenvolvimento e implementação de objectos de gestão [13]. A plataforma define um ambiente de gestão SNMPv1 numa rede TCP/IP, disponibilizando primitivas SNMP para as aplicações de gestão (estação gestora e agente). Esta plataforma possibilita a extensão da MIB no agente, isto é, permite a definição e implementação de novos objectos, de modo a serem enquadrados na MIB que está associada ao agente. Os sistemas operativos suportados são o DOS e o UNIX, sendo a linguagem de desenvolvimento o C.

Em termos gerais, o ambiente definido pela plataforma SNMP engloba uma rede de comunicações à qual estão ligadas diversas entidades de gestão (gestores e agentes). Cada uma destas entidades dispõe de um controlo das sessões que lhes permite estabelecer comunicações com aplicações remotas. A aplicação tem igualmente acesso a um conjunto de primitivas SNMP, através das quais pode exercer a sua função de gestão.

Um possível cenário para a plataforma SNMP seria o representado na fig. 3, no qual a estação gestora efectua operações de gestão sobre cada um dos agentes, utilizando primitivas de gestão SNMP.

Encontra-se representada na fig. 4, a arquitectura de um agente de gestão. Neste está implementada uma estrutura de dados que representa a organização da MIB. O agente recorre a esta estrutura sempre que tem necessidade de obter ou alterar o valor de um dado objecto de gestão. Deste modo, a MIB funciona como uma *interface* entre o agente e os recursos, dando ao agente e ao gestor uma visão organizada da informação de gestão.

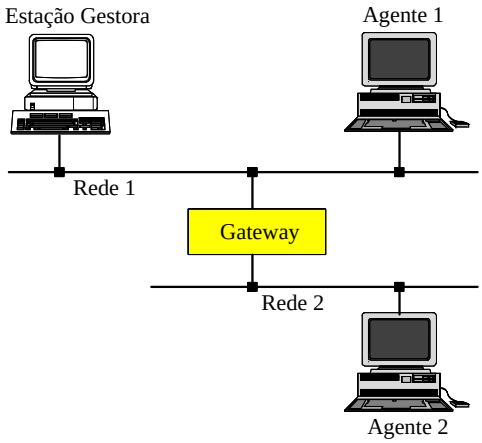


Fig. 3 - Ambiente de Gestão SNMP.

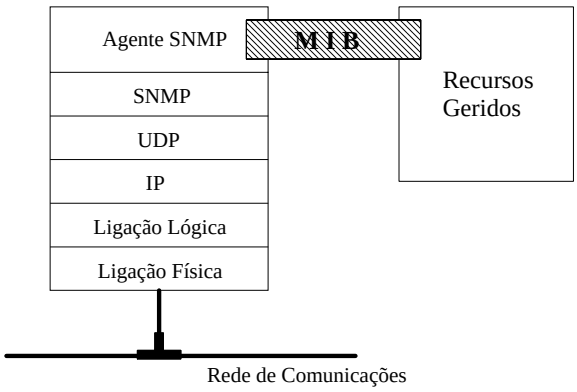


Fig. 4 - O agente SNMP, a MIB e os recursos geridos.

A. Implementação de um Módulo da MIB no Agente SNMP

Nesta secção apresenta-se o procedimento para a implementação de um módulo da MIB na plataforma SNMP.

Em primeiro lugar é necessário definir os objectos de gestão que irão ser implementados. Isto consegue-se através da construção de um módulo da MIB. A sintaxe do módulo é definida no RFC 1212 (*Concise MIB Definitions*).

O próximo passo consiste em compilar o módulo da MIB para um formato que seja capaz de ser interpretado pelo programa que implementa o agente. Para tal, é utilizado o MOSY que gera um ficheiro com as especificações dos objectos num formato específico utilizado na construção das estruturas de dados que representam a MIB.

Seguidamente definem-se as funções que correspondem às primitivas de SET, GET e GET-NEXT para cada um dos objectos.

As estruturas de dados, que representam a organização da MIB são criadas durante a execução do agente (resultado da leitura do módulo da MIB compilado) e inicializadas através de uma função definida pelo autor do módulo. Nessas estruturas de dados existe, associado a cada objecto, um campo com a indicação das funções que correspondem ao GET/GET-NEXT e SET para esse objecto. É então necessário construir uma ou mais funções de inicialização, capazes de preencher essas estruturas de dados com informação relativa às funções que correspondem às primitivas SNMP, para cada um dos objectos definidos. No agente, estas inicializações são efectuadas logo após a criação das estruturas de dados internas que representam a MIB.

B. Implementação de um Agente de Monitoração Remota

A arquitectura do agente de monitoração está representada na fig. 5.

Esta arquitectura tem como suporte um PC equipado com duas placas de rede, uma para a comunicação SNMP e a

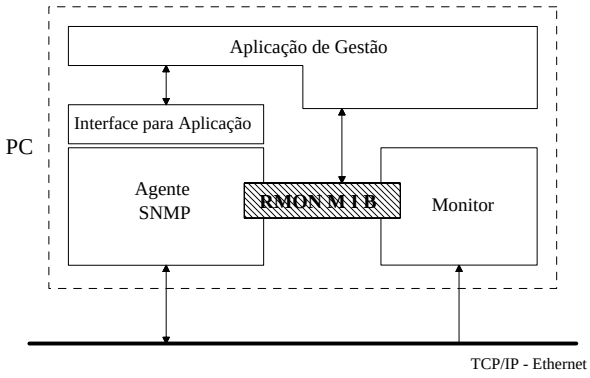


Fig. 5 - Arquitectura de um agente de monitoração remota.

outra para a monitoração da rede. Dela fazem parte um agente SNMP construído com base na plataforma SNMP e um monitor da rede. A informação de monitoração é representada pelos objectos definidos na RMON MIB (RFC 1271). Estas duas entidades, o agente e o monitor, utilizam o mesmo suporte computacional em ambiente DOS, funcionando este, na sua globalidade, como agente de monitoração.

O agente de monitoração realiza, essencialmente, três tarefas: uma consiste em fazer a monitoração da rede através de execução de um programa específico que recolhe toda a informação para tabelas que correspondem às definidas na RMON MIB; a outra consiste em responder aos pedidos da estação gestora, de acordo com o protocolo de comunicação SNMP; a terceira está associada às aplicações de gestão locais com visualização gráfica.

O agente SNMP, na sua fase de inicial, começa por executar uma função que faz a leitura das definições contidas no módulo da MIB compilado, tendo como resultado a criação das estruturas internas que representam a organização da MIB. Essas estruturas são posteriormente inicializadas por funções que indicam, para cada objecto, as funções de GET/GET-NEXT e SET a serem utilizadas pelo agente aquando dos pedidos efectuados pelo gestor. Realizadas estas operações, o agente aguarda pelos pedidos de gestão. Quando é recebida uma mensagem, esta é processada no sentido de identificar o tipo de operação a ser executada e os objectos a que se referem essa operação de gestão.

A informação de monitoração para a RMON MIB é fornecida pelo monitor que consiste basicamente em duas rotinas: uma faz o processamento do pacote que é capturado da rede, incrementando os contadores estatísticos relativos à informação de monitoração; a outra rotina tem como finalidade fazer uma amostragem aos contadores em intervalos de tempo definidos pelo utilizador. As amostras recolhidas são armazenadas em tabelas que correspondem às definidas para os grupos da RMON MIB.

No sentido de dotar o agente com capacidades de gestão locais, foi criada uma *interface* SNMP que permite às aplicações de gestão realizar, de uma forma normalizada,

operações sobre os objectos da MIB. Deste modo, uma aplicação construída sobre essa *interface* pode ser facilmente implementada em qualquer entidade que disponha dos serviços de gestão SNMP. Daqui resulta a possibilidade do desenvolvimento de aplicações de gestão de uma forma genérica, independentemente da entidade gestora a que se destinam (gestor ou agente).

Para aumentar o desempenho de uma aplicação, esta pode ter acesso directo às estruturas da MIB, permitindo assim que as operações de gestão sejam realizadas de uma maneira mais eficiente.

VII. NOTAS DE IMPLEMENTAÇÃO

Pretende-se, nesta secção, apresentar a forma como foram abordados certos pontos que não estão especificados no documento da RMON MIB ou são tratados de uma forma pouco clara.

etherHistoryUtilization: Este objecto pertence ao grupo history e representa uma estimativa da utilização do meio físico (rede Ethernet), em centenas de percentagem, durante o intervalo de tempo de amostragem. O cálculo do valor deste objecto teve por base o tamanho do pacote Ethernet, bem como o espaçamento entre pacotes que é designando por *inter-frame gap*.

A fig. 6 representa o formato de um pacote Ethernet [14]. Para além do número de bytes total, calculados com base no tamanho do pacote fornecido pela placa de rede (inclui os bytes do FCS - *Frame Check Sequence*), é preciso ter em conta o número de bytes relativos ao preâmbulo, ao SFD (*Start of Frame Delimiter*) e ao *inter-frame gap* para o cálculo da utilização de um segmento da rede Ethernet (em centenas de percentagem):

$$\text{Utilização} = \frac{\text{Octetos} + \text{Pacotes} * 20}{C_{\text{Ethernet}} * T_{\text{amostragem}}} * 10000 \qquad (1)$$

onde:

Octetos - nº de bytes recolhidos num intervalo de amostragem (inclui os bytes do FCS);

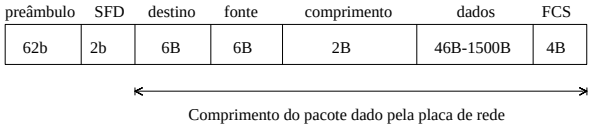
Pacotes - nº de pacotes capturados no intervalo de amostragem;

C_{Ethernet} - Capacidade da rede Ethernet = 10Mbits/s;

$T_{\text{amostragem}}$ - Período de amostragem em segundos.

***Overflow* dos contadores**: Uma vez que os contadores são implementados em variáveis de 4 bytes, o *overflow* destes será inevitável ao fim de um determinado tempo. O contador mais crítico é o que contabiliza o número de bytes total, pelo que o tempo de ocorrência de *overflow* depende da taxa de utilização da rede. Para se ter uma ideia deste tempo, na fig. 7 está representado o tempo de ocorrência de *overflow* em função da taxa de utilização da rede.

Nesta representação, assumiu-se que a taxa de utilização da rede era devida apenas aos bytes contabilizados no



inter-frame gap = 9.6 us = 12 bytes

SFD - *Start of Frame Delimiter*

FCS - *Frame Check Sequence*

b - bits

B - bytes

Fig. 6 - Formato de um pacote Ethernet.

contador. Na realidade, para além destes bytes é necessário ter em conta o número de pacotes no cálculo da taxa de utilização (1), pelo que o gráfico apresenta uma estimativa do valor do tempo por defeito, contando que a taxa de utilização é real.

A RMON dispõe de meios para informar o gestor da ocorrência de um *overflow*. Isto consegue-se através da utilização do grupo alarm, vigiando periodicamente um objecto deste tipo e gerando um alarme aquando da detecção do *overflow*.

Variáveis com Valores por Defeito: Existem algumas variáveis (objectos) definidas no RFC1271 que não possuem valores por defeito. Perante uma situação destas, a questão que se põe é a seguinte: o que deve devolver o agente como resposta a uma operação de GET desses objectos? Existem duas possibilidades [15]:

- Definir um valor por defeito e retorná-lo até que

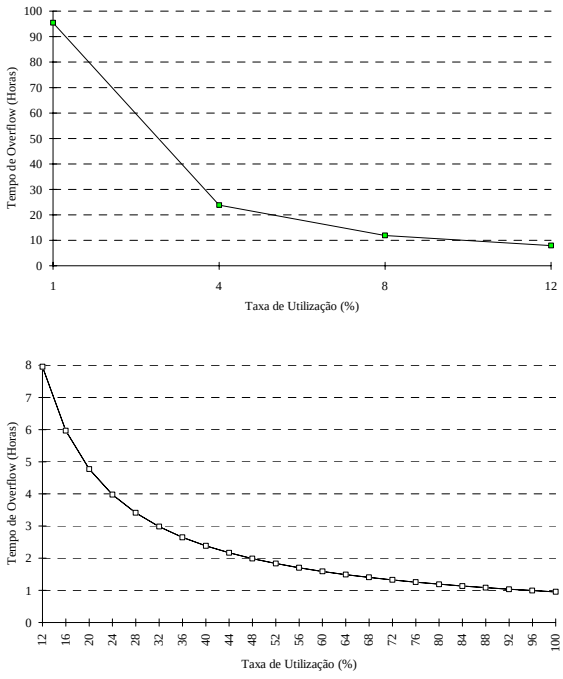


Fig. 7 - Representação do tempo de ocorrência de *overflow* em função da taxa de utilização da rede.

seja alterado.

- Devolver NOSUCHNAME até que a variável seja inicializada pelo gestor.

No caso de ser possível encontrar valores por defeito satisfatórios para as referidas variáveis, então a primeira escolha é preferível. Isto porque a maior parte dos inteiros podem tomar o valor zero, as variáveis lógicas podem ser verdadeiras (1) ou falsas (0), etc. Esta opção só faz sentido se não gerar situações ambíguas quanto ao valor do objecto. Se isso acontecer, a escolha recai na segunda hipótese. O retorno de NOSUCHNAME torna essas variáveis "invisíveis", eliminando a possibilidade de interpretações erradas quanto aos seus valores. No entanto, esta alternativa obriga a que o gestor tenha um pleno conhecimento da MIB implementada no agente, de modo a poder inicializar essas variáveis.

Implementação de Objectos: Viu-se, em secções anteriores, que a implementação dos grupos da RMON é opcional, pelo que um agente poderá implementar apenas alguns deles. Quanto aos objectos de um grupo, as suas implementações são obrigatórias. Pode acontecer um agente não possuir informação relativa a um determinado objecto, por exemplo, devido às capacidades limitadas dos recursos que utiliza. Numa situação destas, que valor deve devolver o agente face a uma operação de GET para esse objecto?

Existem duas hipóteses [16]: uma consiste em retornar um valor adequado (zero no caso de ser uma variável inteira), a outra é retornar NOSUCHNAME. A primeira situação pode induzir o gestor em erro, levando-o a pensar que aquele é o valor efectivo do objecto. O retorno de NOSUCHNAME é o mais correcto, tendo contudo o inconveniente de interromper os algoritmos de recolha de dados de uma tabela, com base na operação GET, uma vez que recebendo um valor deste tipo abortam o pedido de informação. Se a informação é adquirida com a primitiva GET-NEXT, ao chegar o objecto cujo valor de retorno é NOSUCHNAME, a função GET-NEXT avança para o próximo objecto (em termos da árvore da MIB), não havendo qualquer interrupção no processo de aquisição de informação. Deste modo, o gestor concluirá que tal objecto não está implementado.

Analisando as duas hipóteses em termos de conformidade com as normas das MIBs, verifica-se que no caso em que é retornado um valor para um objecto nessas condições, o grupo, ao qual pertence esse objecto, respeita a respectiva norma porque implementa esse objecto, mas não implementa a sua semântica. No caso de retorno de NOSUCHNAME, o grupo implementado não fica de acordo com a norma porque não implementa aquele objecto, sendo no entanto, em termos de valor, a situação mais correcta.

VIII. DEFINIÇÃO DE UM AMBIENTE DE GESTÃO COM MONITORAÇÃO REMOTA

Com base no agente de monitoração remota que foi desenvolvido, pode-se definir um ambiente de gestão de acordo com a fig. 8.

Este cenário é constituído por um agente de monitoração e dois gestores. Estas entidades de gestão foram construídas sobre a plataforma SNMP que permite a implementação dos gestores quer em PCs quer em máquinas UNIX.

Um cenário deste tipo permite obter uma perspectiva global da utilização da rede, através da monitoração do tráfego e da actividade das máquinas, utilizando para isso a RMON MIB. Um agente, para obter um melhor desempenho numa actividade específica de monitoração, pode implementar apenas subconjuntos da RMON MIB.

O agente de monitoração, para além de comunicar com os gestores SNMP, pode funcionar também como um sistema de gestão independente. Neste caso, o agente dispõe de algumas funcionalidades de gestão com capacidades de visualização gráfica, de modo a permitir uma representação local da informação de monitoração e notificar a ocorrência de alguma falha.

A visualização da informação é suportada por um módulo constituído por rotinas gráficas. Esse módulo possui uma *interface* SNMP, que permite o acesso à informação de monitoração através das primitivas SNMP. Isto significa que o módulo opera sobre dos objectos utilizando as primitivas GET, GET-NEXT e SET, de modo a obter a informação necessária para a visualização gráfica e permitir que o utilizador possa também controlar os parâmetros de configuração do agente de monitoração. Com esta *interface*, o módulo gráfico é dotado de uma portabilidade que torna possível a sua implementação em diferentes gestores e agentes SNMP, sem necessidade de alteração do *software*.

Os gestores comunicam com os agentes de monitoração remota através do protocolo SNMP, realizando diversas

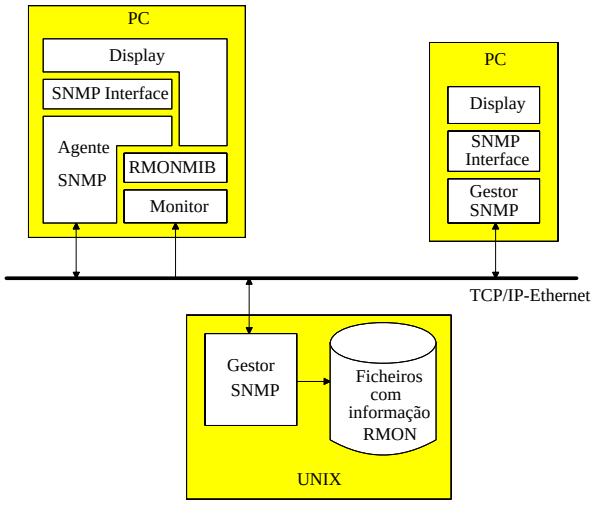


Fig. 8 - Exemplo de um ambiente de gestão baseado num agente de monitoração

operações de gestão tais como a aquisição da informação da MIB ou a configuração de alguns dos seus parâmetros. O gestor pode também armazenar a informação da MIB num ficheiro para uma análise posterior, quer para efeitos de estudos de tráfego na rede, quer para diagnosticar possíveis causas de falhas que tenham ocorrido.

IX. CONCLUSÕES

Actualmente, o processo de integração dos sistemas de gestão está a ser dificultado pelos fabricantes com a apresentação de *interfaces* gráficas sofisticadas, juntando características particulares aos seus próprios sistemas. Torna-se, por isso, necessário melhorar a gestão de redes do ponto de vista de funcionalidade, facilidade de utilização e integração, através da elaboração de normas que devem ser respeitadas pelos fabricantes no desenvolvimento dos sistemas de gestão.

De facto, o sucesso da arquitectura *Internet* (TCP/IP) implicou o desenvolvimento de uma estrutura de gestão para este tipo de redes, do qual surgiu o protocolo SNMP que teve como primeiro objectivo, a resolução do problema de gestão a curto prazo.

Os problemas resultantes das limitações do SNMP, levaram a uma revisão do próprio SNMP, que resultou num protocolo designado por SNMPv2.

Uma das principais actividades de gestão consiste na monitoração de redes, no sentido de diagnosticar falhas, planear e melhorar o desempenho destas. Para isto contribuiu a criação da RMON MIB, que representando uma nova geração de ferramentas de gestão complementa algumas das deficiências da gestão SNMP. Os agentes que implementam esta MIB designam-se normalmente por agentes de monitoração, ou simplesmente agentes RMON.

Para o desenvolvimento de um agente SNMP são necessárias ferramentas que permitam o diálogo com o gestor (implementação do protocolo SNMP) e que implementem a estrutura de dados que representa a organização da MIB, bem como a forma de aceder e processar (codificação e descodificação) essa informação. Essas ferramentas possibilitaram a definição e implementação de um sistema integrado de monitoração de redes locais, baseado num agente de monitoração que suporta a RMON MIB.

O agente de monitoração remota foi desenvolvido com base na plataforma SNMP, tendo-lhe sido adicionadas algumas capacidades de gestão por forma a permitir um modo de funcionamento local. Para tal foi implementado um módulo gráfico para a representação dos vários parâmetros relacionados com a rede (tráfego, erros, protocolos, etc). Este módulo assenta sobre uma *interface* SNMP que lhe permite manipular a informação de gestão de uma forma normalizada, através das primitivas GET, GET-NEXT e SET. O módulo assim desenvolvido pode ser implementado sobre qualquer entidade de gestão (gestor ou agente) que possua capacidades de manipulação dos valores dos objectos.

REFERÊNCIAS

[1] M. T. Rose, K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", RFC 1155, DDN Network Information Center, SRI International, May 1990.

[2] "Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)", International Organization for Standardization / International Electrotechnical Commission, 1987, International Standard 8824.

[3] J. Galvin, K. McCloghrie, J. Davin, "SNMP Security Protocols", RFC 1352, Trusted Information Systems, Inc., Hughes LAN Systems, Inc., MIT Laboratory for Computer Science, July 1992.

[4] J. Case, K. McCloghrie, M. T. Rose, S. Waldbusser, "Introduction to version 2 of the Internet-standard Network Management Framework", RFC 1441, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.

[5] J. Case, K. McCloghrie, M. T. Rose, S. Waldbusser, "Manager-to-Manager Management Information Base", RFC 1451, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.

[6] J. Case, K. McCloghrie, M. T. Rose, S. Waldbusser, "Transport Mappings for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1449, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.

[7] K. McCloghrie, J. Galvin, "Party MIB for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1447, Hughes LAN Systems, Inc., Trusted Information Systems, Inc., April 1993.

[8] "Information Processing Systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)", International Organization for Standardization, December 1987, International Standard 8825.

[9] M. T. Rose, K. McCloghrie, "Concise MIB Definitions", RFC 1212, DDN Network Information Center, SRI International, March 1991.

[10] S. Waldbusser, "Remote Network Monitoring Management Information Base", RFC 1271, DDN Network Information Center, SRI International, November 1991.

[11] S. Waldbusser, "Token Ring Extensions to the Remote Network Monitoring MIB", RFC1513, Carnegie Mellon University, September 1993.

[12] Marshall T. Rose, Julian P. Onions, Colin J. Robbins, "The ISO Development Environment: User's Manual", Performance Systems International, Inc., July 1991.

[13] Abílio Carvalho, Nelson Rocha, "Plataforma SNMP para o Desenvolvimento de Objectos de Gestão", Revista do DETUA, Aveiro, Dezembro 1993.

[14] ANSI/IEEE, "Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications", ANSI/IEEE Std. 802.3-1985, The Institute of Electrical and Electronics Engineers, 1986.

[15] Kooijman, D. M. Wisse, "The Annotated RMON Developer's Guide", June 1993.

[16] Robin Iddon, "RmonMIB Implementation Problems", RMON MIB Mailing List, AXON Networks Inc., April 1993.