

Plataforma SNMP para o Desenvolvimento de Objectos de Gestão

Abílio Pacheco de Carvalho, Nelson Pacheco da Rocha

Resumo - O trabalho aqui apresentado consiste numa plataforma SNMP para o desenvolvimento de objectos de gestão, constituída por um gestor SNMP e um agente SNMP com possibilidade de definição de novos objectos de gestão. Esta plataforma foi desenvolvida de forma genérica, podendo ser instalada em qualquer sistema de comunicações e/ou sistema operativo.

Abstract - The article presents a SNMP platform to develop managed objects. This platform consists of a SNMP manager and a SNMP agent with capabilities for the development of new managed objects. It is a general purpose platform able to work in different communication systems and/or operating systems.

I. INTRODUÇÃO

Os produtos de gestão de redes comercialmente disponíveis possuem particularidades próprias e operam com componentes de rede específicos, colocando dificuldades na sua utilização em cenários heterogéneos como aqueles das actuais redes de comunicação. Em redes de múltiplos segmentos é muito difícil obter informação acerca do nível de utilização da rede, bem como saber, em cada instante, quais os componentes que estão a funcionar com dificuldades. Assim, é necessária uma ferramenta capaz de monitorizar e controlar, de uma forma integrada, todos os elementos de rede, usando quer agentes de monitorização dedicados, quer as ferramentas de gestão que alguns desses elementos já possuem.

Neste contexto, a informação de gestão, é organizada num conjunto de objectos de gestão que constituem entidades capazes de representar os recursos da rede.

II. SISTEMA INTEGRADO DE GESTÃO

Em meados de 1990, começou-se a desenvolver no Departamento de Electrónica e Telecomunicações da Universidade de Aveiro (DETUA) um sistema integrado de gestão para redes locais, baseado em agentes de monitorização e numa Estação Central de Gestão (ECG) na qual se efectuam as operações de gestão [1,2] (Fig. 1). Os agentes de monitorização são ligados aos vários segmentos e permitem saber, em cada instante, o estado de funcionamento do canal de comunicações e dos

diversos elementos de rede, em especial daqueles que não possuem qualquer mecanismo de gestão associado. A rede a gerir por este sistema caracteriza-se por ser heterogénea e constituída por múltiplos segmentos, nos quais operam várias famílias protocolares como, por exemplo, TCP/IP e OSI.

A. Estação Central de Gestão

A Estação Central de Gestão (ECG) têm a seu cargo a integração das operações dos agentes de monitorização e das operações de outros agentes. A comunicação com os agentes de gestão, é feita usando diferentes protocolos de gestão que podem ser o Common Management Information Protocol (CMIP) [3], o Simple Management Information Protocol (SNMP), ou outros protocolos privados. É da responsabilidade da ECG integrar todos estes protocolos e toda a informação disponibilizada pelos diferentes agentes.

A ECG mantém uma base de dados de informação relevante sobre o funcionamento da rede coleccionada através dos agentes, e pode usar capacidades de sistemas periciais para tratamento e resolução de condições anómalas conhecidas. Possui também um sistema de visualização baseado no sistema X-Windows para a apresentação, ao gestor da rede, de forma amigável de um conjunto de mecanismos que lhe permite ter acesso a todos os aspectos, informações e detalhes da rede, e que suportam a representação da rede, a selecção de elementos específicos e a apresentação de resultados [4] (fig.2).

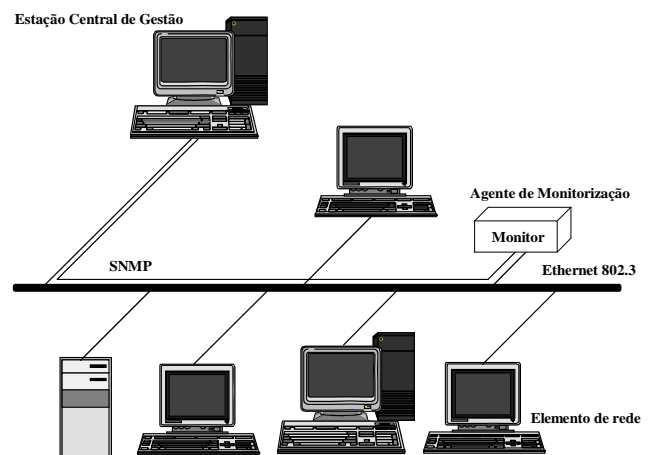


Figura 1. Sistema Integrado de gestão

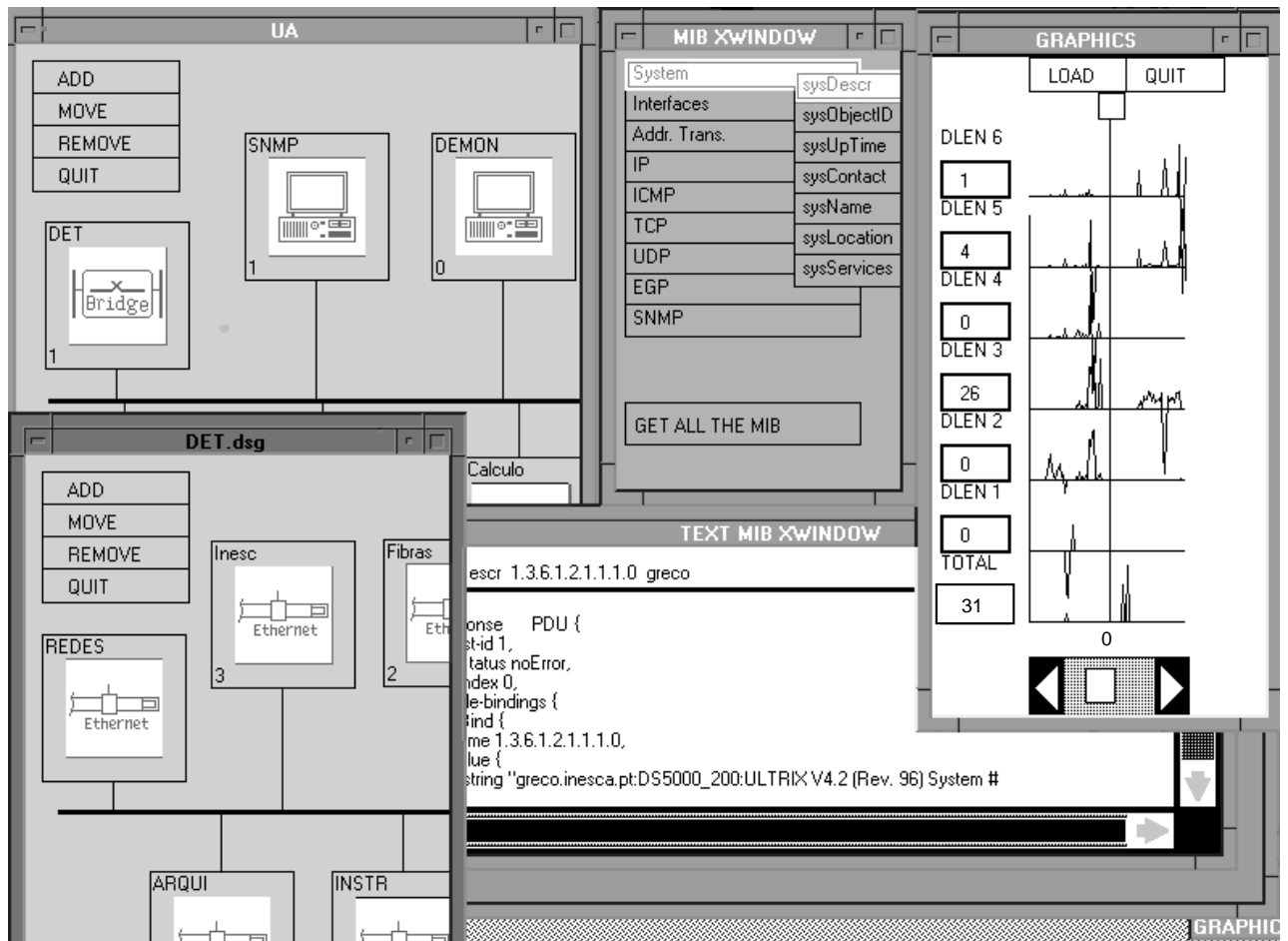


Figura 2. Interface gráfica do Sistema Integrado de Gestão

A ECG possui duas partes fundamentais: uma é o Processo de Comunicação (PC) que colecciona informação de gestão das Management Information Bases (MIBs) existentes nos agentes e envia comandos através do protocolo de gestão; a outra parte é a Aplicação de Gestão (AG), a qual tem a seu cargo o processamento da informação de gestão e a apresentação dos resultados. A interface entre estas duas partes é a Base de Dados de Informação de Rede (BDIR), que contém toda a informação coleccionada e permite a independência entre os dois módulos (Fig. 3).

A BDIR é uma representação abstracta da rede construída pela aplicação de gestão e é independente dos detalhes de comunicação. Por sua vez o PC permite à AG colocar na BDIR a informação obtida nos agentes e enviar comandos de gestão de acordo com as suas determinações.

B. Agentes de monitorização

Os agentes de monitorização são agentes especiais cuja finalidade é coleccionar informação relativa ao comportamento da rede e dos sistemas a ela ligados sem facilidades de gestão. Podem existir vários agentes de monitorização, desempenhando tipos diferentes de análise ou ligados a diferentes tipos de redes (Ethernet, Token-Ring, FDDI, ...). Estes agentes, têm somente funções de

gestão e são de grande importância, uma vez que sem eles não seria possível gerir num cenário transitório, sistemas com e sem facilidades de gestão.

As facilidades de gestão disponibilizadas por estes agentes prendem-se com os seguintes tópicos [1]:

- **Segurança** - Os agentes de monitorização podem analisar os pacotes em trânsito na rede e tirar conclusões acerca dos sistemas a elas ligados, sendo capazes de detectar sistemas novos, ou sistemas sem permissão de acesso a essa rede.
- **Optimização** - Os agentes de monitorização coleccionam dados directamente a partir do meio de transmissão podendo obter informação acerca das condições gerais de funcionamento da rede através de funções de processamento. Estas funções de processamento permitem libertar a ECG de tarefas como, por exemplo, o cálculo do tráfego total, poupando tempo e não sobrecarregando a rede com tráfego de gestão.
- **Informação de gestão em tempo real** - Os agentes de monitorização obtêm informação de gestão em tempo real, sendo possível a sua parametrização para fazer a aquisição com uma determinada cadência.
- **Funcionamento autónomo** - Para além das facilidades de gestão disponibilizadas à ECG, os agentes de monitorização podem também fazer a visualização de

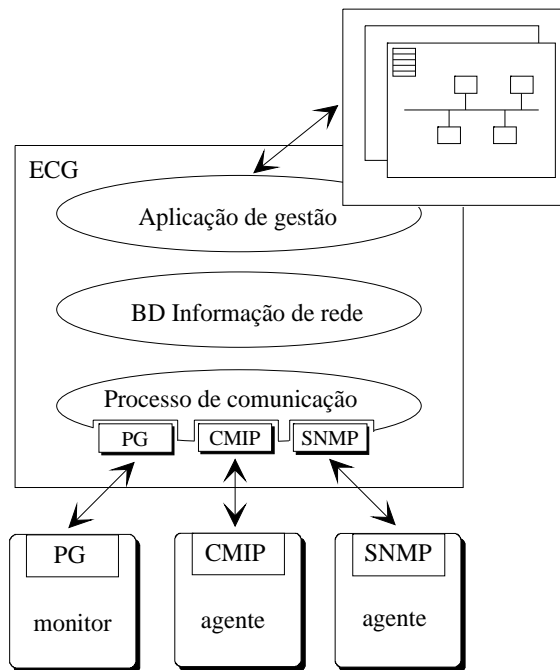


Figura 3. Arquitectura do Sistema Integrado de Gestão

aspectos de funcionamento da rede considerados relevantes.

- **Planeamento** - dado que o agente de monitorização colecciona informação acerca do meio de transmissão, ele é capaz de concluir sobre as condições de funcionamento da rede, nomeadamente saber quem são os maiores geradores de tráfego em determinado segmento.

A implementação destes agentes de monitorização baseia-se num computador do tipo IBM compatível com dois controladores Ethernet. Este sistema pode possuir vários módulos de monitorização, um módulo de protocolos, e para o funcionamento em modo autónomo possui também uma interface gráfica [5] (Fig. 4). O módulo de monitorização interage com o controlador Ethernet e colecciona todas as tramas do segmento a que está ligado. A informação coleccionada é processada e organizada em formatos adequados, que constituem as sintaxes locais, e que permitem facilmente a conversão para as sintaxes dos objectos de gestão definidos para suportarem essa informação. O módulo de protocolos desempenha as tarefas de interligação com a estação central de gestão para a transferência da informação de gestão. Por último, a interface gráfica permite fazer a visualização local da informação adquirida.

III. GESTÃO SNMP

O SNMP foi desenvolvido no sentido de constituir um protocolo de gestão para a internet, sendo o seu principal objectivo ser um protocolo simples e de fácil implementação[7]. Hoje em dia, ele é um padrão para

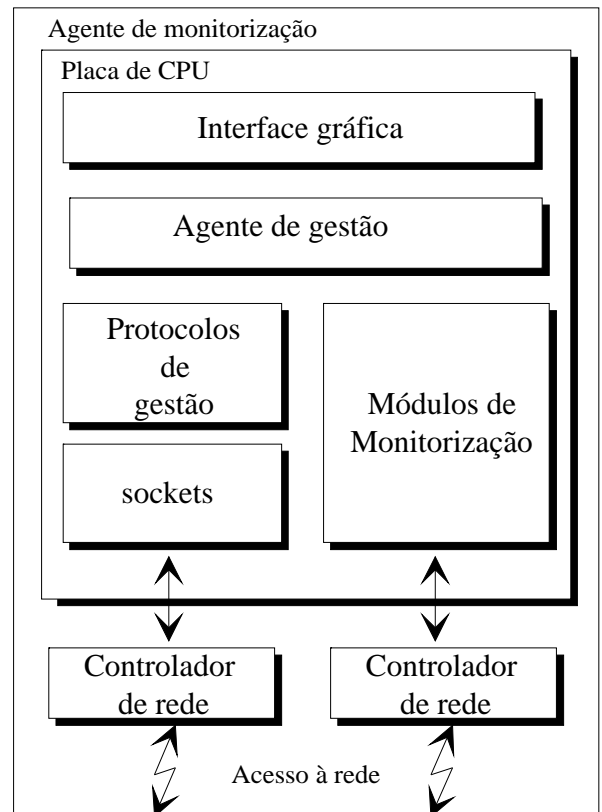


Figura 4. Agente de monitorização

gestão de redes locais, tendo sido adoptado por um número bastante significativo de fabricantes, sendo utilizado em muitos equipamentos, tais como: portões, encaminhadores, estações de trabalho, servidores de terminais, etc[7].

O protocolo SNMP é um protocolo comando-resposta assíncrono (Fig. 5), e suporta vários Protocol Data Units (PDUs), que permitem a execução de diferentes operações (Fig. 6):

- O **get-request**, é usado para requerer aos agentes informação de gestão específica, isto é, o objecto é exactamente especificado;
- O **get-next-request**, é usado para requerer informação de gestão, sem que o objecto pretendido seja especificado exactamente. O objecto retornado é aquele que se encontra lexicograficamente a seguir ao objecto especificado na operação, sendo o resultado da operação dependente do agente. É portanto uma operação de varrimento.
- O **set-request**, é usado para modificar o valor dos objectos com acesso de escrita;
- O **get-response**, é usado pelos agentes para responder às operações de get e set;
- O **trap**, é usado pelos agentes para assinalarem a ocorrência de situações anómalas através de mensagens não solicitadas.

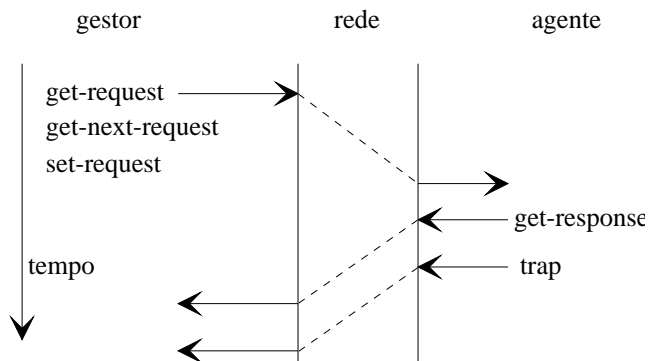


Figura 5. Protocolo comando-resposta

As funções de gestão, são especialmente necessárias quando existem problemas no funcionamento da rede [6], pelo que deverá ser a aplicação de gestão a assegurar o nível de segurança desejado para o tráfego de gestão. O serviço de transporte datagrama é o mais adequado para dar essa liberdade à aplicação de gestão, e também o mais rápido, uma vez que não é necessário o estabelecimento de ligação.

Por outro lado, a transmissão de mensagens SNMP pela rede, é feita através de um processo de serialização, do qual resulta uma sequência de octetos. Esta sequência de octetos pode ser transmitida por qualquer serviço de transporte, sendo apenas necessário que a entidade SNMP receptora seja capaz de aceitar mensagens que quando serializadas ocupam até 484 octetos[8].

Assim, o SNMP não é muito exigente a nível do protocolo de transporte requerido, sendo o protocolo de transporte UDP o preferido.

O SNMP tem-se manifestado como sendo o protocolo dominante para a troca de informação de gestão entre

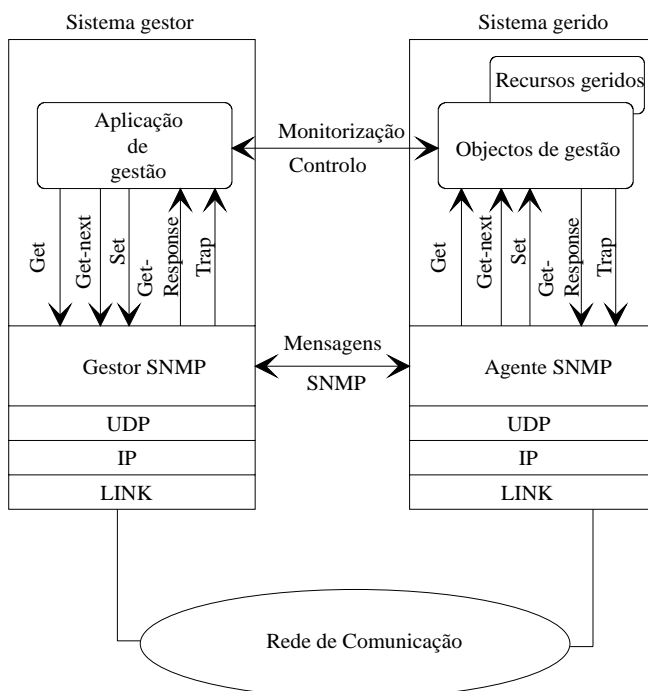


Figura 6. Arquitectura SNMP

estações de gestão e os elementos de rede. O SNMP é constituído por três componentes: o protocolo [8], a Management Information Base (MIB) [9,10], e a estrutura de informação de gestão (Structure of Management Information - SMI) [11] através da qual os objectos de gestão são definidos como variáveis escalares[12].

Os objectos de gestão são definidos usando a macro OBJECT-TYPE, através da qual é estabelecida uma relação de nomes entre os objectos de gestão, assim como a sua sintaxe e o nível de acesso[11]. Neste sentido é usado um subconjunto da linguagem Abstract Syntax Notation One (ASN.1) [11,13], como uma notação que fornece informação concisa para a implementação. O exemplo seguinte é uma definição ASN.1 de um objecto de gestão:

```
sysDescr OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS mandatory
::= {system 1}
```

A. Agente de monitorização SNMP

A comunidade Internet normalizou à partida uma colecção de objectos que constitui a MIB II [9,10] e que serve de base à gestão SNMP. No entanto, este conjunto de objectos, abrange somente uma área muito restrita dos aspectos de gestão a ter em consideração. Para superar esta limitação inicial, foram criados na comunidade Internet vários grupos de trabalho, com a tarefa de especificação de novas MIBs, visando cobrir novas áreas da gestão. Assim para a informação de monitorização foi definida e normalizada pela comunidade Internet a Remote Monitoring MIB (RMON-MIB) [14,15].

A RMON MIB endereça várias áreas da gestão como diagnóstico de falhas, análise de desempenho e planeamento. Para atingir estes propósitos, a RMON MIB possui algumas características, das quais, as mais relevantes são [15]:

- Operações "offline": A ECG pode não estar em permanente contacto com os dispositivos de monitorização, por razões de economia de comunicações ou por avaria na rede. Assim, a RMON MIB permite configurar os dispositivos de monitorização para fazerem continuamente diagnósticos e estatísticas mesmo quando não se processa qualquer comunicação.
- Historial da informação de monitorização: A RMON MIB define objectos que permitem ao dispositivo de monitorização reservar recursos e armazenar informação de monitorização correspondente a um determinado período. Em caso de avaria, a estação central de gestão pode proceder à análise deste historial e diagnosticar as causas do problema.
- Detecção e notificação de problemas: O monitor pode ser configurado para detectar condições de erro e notificar a estação central de gestão.

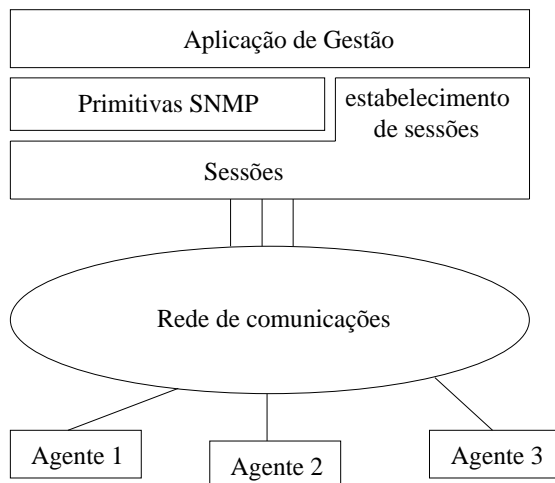


Figura 7. Gestor SNMP

- Informação acrescentada: Os dispositivos de monitorização são exclusivos para as funções de gestão. Assim, estes dispositivos têm oportunidade para fazer algum processamento na informação adquirida. Por exemplo assinalar quem são os elementos da rede com maior geração de tráfego ou erros.
- Aquisição em tempo real: Os dispositivos onde a RMON MIB é instalada possuem hardware dedicado que permite a aquisição e a disponibilização de informação de monitorização em tempo real.
- Extensibilidade a outras redes: Embora grande parte dos objectos definidos na RMON MIB tenha interesse na gestão de qualquer tipo de rede, existem alguns que são específicos para a rede Ethernet. No entanto, o desenho da MIB permite a definição de objectos com características similares para outras redes, por exemplo, as redes FDDI e Token Ring.

IV. GESTOR SNMP

O gestor SNMP é constituído pelo conjunto de primitivas do protocolo SNMP que são utilizadas pela aplicação de gestão, e por um mecanismo de estabelecimento de ligações de gestão baseado no conceito de sessão. O conceito de sessão permite a inicialização e manutenção de todos os parâmetros necessários para as operações SNMP entre o gestor e cada agente de gestão, como, por exemplo, o endereço do agente e o descritor da socket para o envio dos datagramas (fig. 7).

As primitivas disponíveis no gestor são cinco e são implementadas pelas seguintes funções:

- send_get_request
- send_get_next_request
- send_set_request
- rcv_get_response
- rcv_trap

As primitivas do SNMP permitem à aplicação de gestão a manipulação de variáveis de gestão, as quais, do ponto de vista do protocolo SNMP, são pares (nome, valor) que podem ser transportados em listas. Assim, os parâmetros

passados a cada primitiva são listas de pares (nome, valor) e também o descritor de sessão para identificar o destinatário da operação SNMP. Com estes parâmetros, cada primitiva controla uma mensagem SNMP, faz a sua codificação usando o código gerado pelos compiladores ASN.1 e em seguida procede ao seu envio para a rede.

Pelo facto da estrutura das mensagens SNMP ser igual para todas as operações (com excepção dos traps) e também para a reutilização do código, foi construída uma primitiva genérica à qual é passado um parâmetro adicional, o parâmetro tag, para a distinção entre primitivas.

```
primitive(session, names, values, nelem, tag)
{
    msg = message_construct(session, names, values,
                           nelem, tag);
    encode(msg, buffer);
    sendto(session->sd, buffer, len, session->remote, ...);
}
```

Esta primitiva genérica permite o envio de primitivas get_request, get_next_request e set_request. Para a recepção de primitivas get_response e de traps o gestor SNMP contém a função rcv_primitive que devolve à aplicação de gestão a mensagem SNMP recebida:

```
Message_SNMP *rcv_primitive(session)
{
    rcvfrom(session->sd, data, ..., &remote,...);
    decode(&msg, ..., &data,...);
    return(msg);
}
```

A aplicação de gestão pode usar estas primitivas, competindo-lhe decidir quando as utilizar e quais os objectos a gerir. O código a seguir ilustra a simplicidade dos procedimentos necessários para uma interacção gestor-agente.

```
session = init_session(hostname, comunidade);
primitiva(session, names, valueToSend, 1, get_request);
msg=rcv_primitive(session); /*recebe um get_response */
extract_variables(msg);
```

V. AGENTE SNMP

Em virtude de um agente SNMP poder conter diferentes conjuntos de objectos de gestão conforme a informação de gestão que dele se pretenda extrair, a presente implementação do agente constitui uma plataforma de desenvolvimento de objectos de gestão, à qual podem ser facilmente adicionados novos objectos de gestão (definição de novas MIBs) sendo apenas necessária a construção das respectivas rotinas de acesso e de um procedimento simples de inicialização.

O agente SNMP é mais complexo do que o gestor, e é composto por três módulos: a unidade de controlo SNMP, a representação da MIB e a unidade de comunicação com os recursos geridos (Fig. 8, Fig. 10).

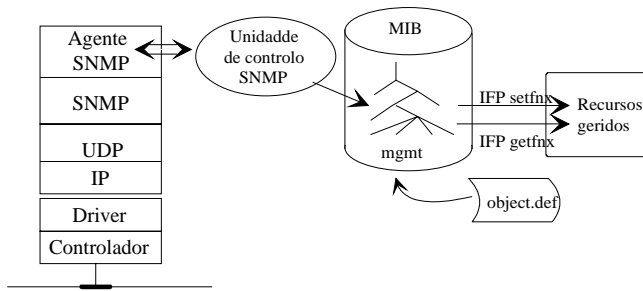


Figura 8. Agente SNMP

Para o caso particular de implementação da RMON-MIB a informação dos recursos geridos é fornecida pelo monitor de rede que é constituído pela interface de rede e pelos módulos de monitorização (Fig. 9).

A interface de rede do monitor é constituída por uma placa controladora Western Digital e por um driver Western Digital, que permite aos módulos de monitorização ler toda a informação que circula na rede. Esta escolha deve-se ao facto de possuímos já desenvolvido um driver de software para esta placa podendo, no entanto, ser utilizada qualquer outra interface desde que ela permita aceder a toda a informação da rede.

Os módulos de monitorização têm a seu cargo a colecção de informação de monitorização e a sua organização de forma adequada aos requisitos de informação da RMON-MIB.

Por razões de modularidade, a interface e os diversos módulos de monitorização estão integrados num único programa designado por monitor, que pelo facto de constituir um programa distinto do agente, necessita do estabelecimento de um mecanismo de comunicação.

No sistema operativo DOS para simular a existência desses dois processos, o monitor após a reserva de recursos é deixado residente em memória sendo colocado a correr através das interrupções do driver Western Digital, e o programa agente é colocado a correr

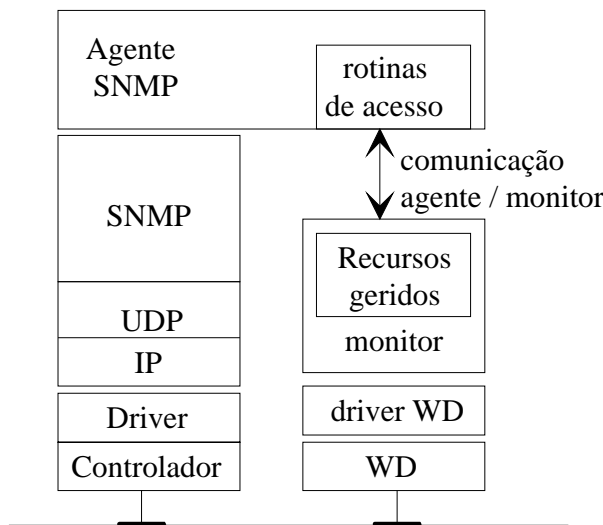


Figura 9. Agente SNMP com um monitor de rede associado

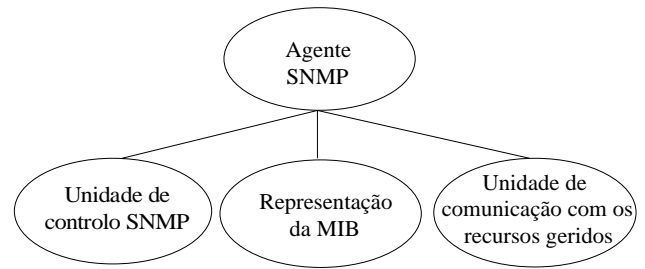


Figura 10. Diagrama de estrutura do agente SNMP

normalmente.

Como mecanismo de comunicação, é utilizada a partilha da área de memória onde é armazenada a informação de monitorização coleccionada pelo monitor. Para esta partilha é necessário que o agente tenha conhecimento da localização da área de memória a partilhar. Assim é-lhe passado através da tabela de interrupções do DOS o endereço inicial dessa área de memória. A escolha da tabela de interrupções prende-se com o facto dela conter um conjunto de posições de memória que não são alteradas quer pelo sistema operativo quer por outros programas. Para a igual interpretação dos dados são usadas as mesmas estruturas quer pelo monitor quer pelo agente.

A. Unidade de controlo SNMP

Na unidade de controlo do protocolo SNMP, o agente procede à recepção das mensagens, ao seu processamento e ao conseqüente envio da resposta, conforme se ilustra no seguinte fragmento de código:

```
while(1) {
    msg = recv_request(sessiond);
    process(&msg);
    send_get_response(sessiond, msg);
}
```

O agente lê o datagrama UDP recebido e guarda o endereço do gestor, que é depois utilizado para o envio da resposta. É feita também a descodificação do datagrama para uma estrutura interna, a estrutura mensagem, sendo em seguida passada à função de processamento.

No processamento, para cada variável na lista de variáveis, é pesquisado o respectivo protótipo sendo em seguida invocada a correspondente rotina de acesso ao recurso.

Após o processamento de todas as variáveis da lista, o agente envia de volta a mensagem, que foi devidamente modificada pela função de processamento e pelas rotinas de acesso.

Durante o processamento, é na própria mensagem recebida que são actualizados os respectivos campos, isto é, o campo valor é preenchido com a informação requerida ou em caso de erro é identificado o respectivo erro.

A utilização da mesma mensagem na recepção e na emissão, é importante porque permite que um único bloco de memória seja reservado quando a mensagem é recebida, e seja libertado logo após o seu processamento e envio. Este procedimento permite utilizar o mínimo de memória necessária e também uma gestão simples dessa memória.

B. Representação da MIB

A representação da MIB é construída a partir de um ficheiro em formato mosy (Managed Object Syntax Compiler), resultado da compilação da MIB, cuja informação é utilizada para preencher uma estrutura, a estrutura `object_type`, que implementa a árvore de objectos (OIT).

Os objectos de gestão são colocados numa tabela "hash", em que cada elemento da tabela é uma lista ligada com múltiplas ligações entre estruturas `object_type` (`ot`), representando cada `ot` um objecto de gestão, tal como ilustra a Fig. 11.

```
typedef struct object_type {
    char *ot_text; /* descritor */
    char *ot_id; /*identificador - string */
    OID ot_name; /* identificador */
    OS ot_syntax; /* sintaxe SMI do objecto */
    int ot_access; /* acesso */
    int ot_status; /* status */
    caddr_t ot_info; /* informação do objecto */
    IFP ot_getfnx; /* metodo get / get_next */
    IFP ot_setfnx; /* metodo set */
    struct object_type *ot_chain;
    struct object_type *ot_sibling;
    struct object_type *ot_children;
    struct object_type *ot_next;
} object_type, *OT;
```

Por razões de busca mais rápida, os objectos são colocados em diversas listas ligadas, cada uma com origem numa posição da tabela "hash". O seu posicionamento é calculado pela respectiva função de "hashing", a qual foi desenvolvida de forma a gerar um valor entre 0 e o máximo da tabela, utilizando como chave o nome dos objectos. Assim, para procurar um objecto na MIB, é necessário calcular em primeiro lugar em que lista ligada é que o objecto se encontra, sendo este um processo simples e rápido, uma vez que consiste somente na execução da função de "hashing".

Depois de localizada a lista ligada, ela deve ser percorrida até se encontrar o objecto pretendido. Por exemplo, para procurar um objecto de nome "nome_objecto" pode usar-se a seguinte linha de código:

```
OT ot;
for(ot = Hash[ FHASH( nome_objecto)];
    ot && strcmp(ot->ot_text, nome_objecto); ot = ot->ot_chain)
    continue;
/* neste ponto, ot é o protótipo do objecto procurado */
```

Os restantes ponteiros existentes permitem outras alternativas de pesquisa conhecendo por exemplo o objecto através do seu OBJECT IDENTIFIER, ou para localizar o objecto lexicograficamente a seguir para as operações de `get-next-request`.

Como se pode observar, a procura dos objectos é mais demorada ao percorrer-se a lista ligada, sendo portanto conveniente que as listas sejam pequenas. Assim o tamanho da tabela "hash" deve ter um comprimento adequado ao número de objectos que a MIB contém.

Depois de encontrado o objecto, pode-se aceder ao recurso associado através dos ponteiros para função `ot_getfnx` e `ot_setfnx`.

Para o processamento das mensagens SNMP, e porque os objectos são identificados pelo seu OBJECT IDENTIFIER, usa-se também a estrutura `object-instance` que combina o nome do objecto e o seu protótipo:

```
typedef struct object_instance {
    OID oi_name; /* nome da instância */
    OT oi_type; /* protótipo do objecto */
} object_instance, *OI;
```

A representação da MIB, que se baseia nestas estruturas, assume um elevado grau de complexidade, sendo portanto necessário dotar o programador de mecanismos que auxiliem a sua manipulação. Neste sentido, existe um conjunto de funções que permitem o acesso à MIB, quer para a pesquisa de objectos quer para a conversão de nomes. Desta forma é feito o encapsulamento dos mecanismos internos à representação da MIB, sendo fornecido ao programador o seguinte conjunto de funções que constituem os métodos de acesso à MIB:

- OID_text2oid(name) - converte o descritor de um objecto no respectivo OBJECT IDENTIFIER.
- OT_name2obj(oid) - procura um objecto na árvore através do OBJECT IDENTIFIER e retorna o seu protótipo.
- OT_text2obj(text) - procura um objecto na árvore através do descritor e retorna o seu protótipo.
- OI_name2inst(oid) - procura um objecto na árvore através do OBJECT IDENTIFIER e retorna a estrutura

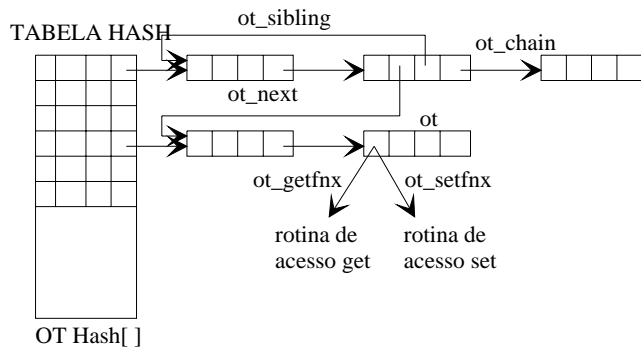


Figura 11. Disposição dos objectos na representação da MIB

object_instance correspondente.

- OI_next2inst(oid) - procura o objecto lexicograficamente a seguir ao objecto identificado pelo seu OBJECT IDENTIFIER e retorna a estrutura object_instance correspondente.
- OI_text2inst(text) - procura um objecto na árvore através do descritor e retorna a estrutura object_instance correspondente.

C. Inicialização do agente

O submódulo Inicialização procede à construção da representação da MIB recorrendo a uma função apropriada, e em seguida para cada objecto suportado, procura o correspondente protótipo object_type, e os respectivos campos ot_getfnx e ot_setfnx são inicializados com os endereços das correspondente rotinas de acesso. Normalmente estas rotinas desempenham as operações get-request e set-request para vários objectos, sendo para tal associada uma constante a cada objecto e colocado o seu valor no campo ot_info, permitindo assim às rotinas de acesso distinguir os vários objectos de que são responsáveis. Por exemplo, se as rotinas o_matrix e s_matrix forem as rotinas de acesso para os objectos de gestão do grupo matrix da RMON-MIB, a inicialização deste grupo de objectos de gestão é feita como ilustra o seguinte fragmento de código:

```
#define matrixControlDataSource2
...
OT ot;
if ( ot = text2obj ("matrixControlDataSource")) {
  ot -> ot_setfnx = s_matrix,
  ot -> ot_getfnx = o_matrix,
  ot -> ot_info = (caddr_t) matrixControlDataSource;
...

```

Assim, para a inclusão de novos objectos de gestão é necessário somente escrever as rotinas de acesso get e set, e efectuar as respectivas inicializações.

Para além destas inicializações, o agente lê algumas informações adicionais de configuração como:

- O nome das comunidades suportadas, e o endereço IP dos gestores;
- O nível de acesso para cada comunidade;
- A vista da MIB associadas a cada comunidade.

O nome da comunidade é uma string ASCII, e a ela devem ser associados os endereços IP dos gestores que usam essa comunidade. O endereço 0.0.0.0 permite que a comunidade seja usada por qualquer gestor.

O nível de acesso especifica para os gestores associados a comunidade se têm permissão de leitura ou escrita.

Por sua vez, a vista da MIB define a colecção de objectos de gestão que podem ser acedidos, sendo identificada por um nome, um OBJECT IDENTIFIER, e pelos nomes das subárvores que a compõem como, por exemplo, system, interfaces, matrix e statistics.

Depois de todas as inicializações o agente entra num ciclo de espera para o atendimento às mensagens SNMP que chegam ao endereço de transporte, que é normalmente a porta 161 UDP.

D. Unidade de comunicação com os recursos geridos

A unidade de comunicação com os recursos geridos é constituída pelas rotinas de acesso aos recursos. Cada objecto de gestão ou grupo de objectos de gestão possui duas rotinas de acesso: uma rotina de acesso para as operações get-request e get-next-request e outra para as operações set-request. Estas rotinas são apontadas pelos campos ot_getfnx e ot_setfnx, seguem um padrão no que se refere à sua estrutura e aos seus parâmetros, e fazem parte do trabalho de definição de novos objectos. A sua estrutura é constituída por um corpo comum que facilita a sua definição para os novos objectos a criar. Por forma a uniformizar a comunicação com os recursos geridos, as rotinas de acesso são invocadas sempre da mesma forma:

```
int result = metodo (oi, v, offset)
OI oi;
VarBind *v;
int offset;
```

Os parâmetros têm o seguinte significado:

- O parâmetro oi é do tipo object_instance, contém o nome da variável e um ponteiro para a estrutura object_type;
- O parâmetro v contém também o nome da variável, e um campo para armazenar o seu valor. A estrutura deste parâmetro é a estrutura VarBind usada na mensagem SNMP:


```
typedef struct {
  ObjectName name;
  ObjectSyntax value;
} VarBind ;
```
- O parâmetro offset, identifica a operação a ser desempenhada, pode ser get, get-next ou set.

Para as operações get-request e get-next-request o endereço de metodo é igualado ao endereço apontado por ot_getfnx, e para as operações set-request é igualado ao endereço apontado por ot_setfnx.

VI. CODIFICAÇÃO E DESCODIFICAÇÃO DE MENSAGENS SNMP

A codificação e descodificação de mensagens protocolares está directamente relacionada com a especificação dos protocolos e com os mecanismos associados ao método de especificação. No caso do protocolo SNMP, a especificação é feita em linguagem ASN.1, sendo portanto a codificação e descodificação dependente da linguagem ASN.1.

A codificação e descodificação fazem parte de um processo de transformação de sintaxes, no qual estão envolvidas por um lado a sintaxe abstracta usada na especificação e a correspondente sintaxe local e, por outro lado, a sintaxe de transferência usada para as mensagens enquanto em trânsito na rede de comunicações [16] (Fig. 12).

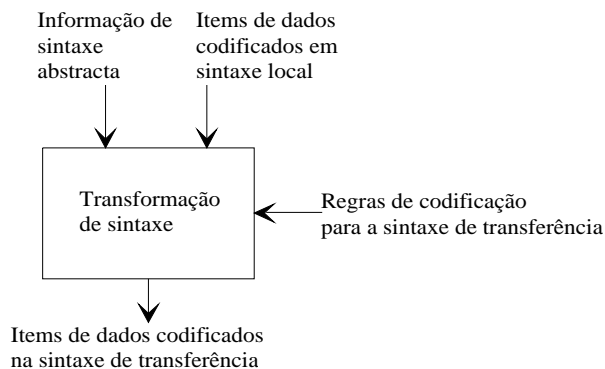


Figura 12. Transformação de sintaxe

Os procedimentos do protocolo são feitos com base na manipulação de estruturas C que constituem a sintaxe local. A partir destas estruturas a informação é codificada-descodificada utilizando-se as funções de codificação/descodificação geradas pelos compiladores de ASN.1. Estes compiladores permitem a geração automática das funções de codificação e descodificação as quais são posteriormente ligadas ao código do protocolo conforme ilustra a Fig. 13.

Sendo o processamento protocolar efectuado com base nas estruturas C, é possível tornar a máquina protocolar independente do processo de codificação/descodificação, isto é, independente das particularidades dos compiladores de ASN.1. Assim, para a plataforma SNMP de desenvolvimento de objectos de gestão foram usados os compiladores de ASN.1 do ISODE [17] e o compilador de ASN.1 desenvolvido pelo INESC[18], permitindo a sua instalação em vários sistemas operativos.

VII. CONCLUSÃO

Este trabalho enquadra-se no vasto esforço de investigação e de desenvolvimento nas áreas tecnológicas associadas à comunicação de dados em particular nos sistemas de gestão de redes.

Foi apresentado um Sistema Integrado de Gestão que

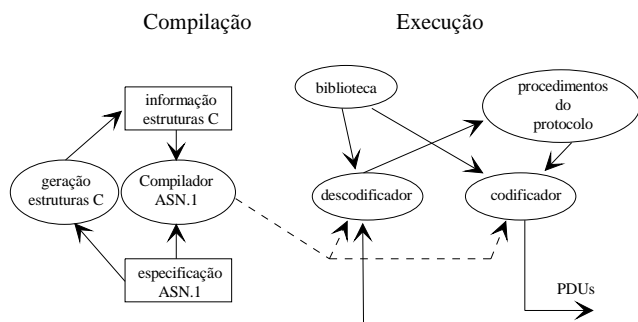


Figura 13. Relação compilação / execução ASN.1

integra facilidades de gestão já instaladas e as facilidades de gestão fornecidas pelos agentes de monitorização. Para este sistema foi desenvolvido uma plataforma SNMP para o desenvolvimento de objectos de gestão constituída por um gestor SNMP e por um agente de gestão SNMP. A plataforma SNMP é independente do sistema operativo e dos compiladores de ASN.1 e permite facilmente a sua evolução para a nova versão do SNMP (SNMPv2) [19].

O Sistema Integrado de Gestão e a plataforma SNMP permitem o estudo de ferramentas de gestão de redes, nomeadamente a correcta definição e implementação de objectos de gestão relevantes.

AGRADECIMENTOS

O trabalho de Mestrado que serviu de base para este artigo, foi o projecto Greco, suportado pela JNICT através do contrato PMCT/C/TIT/458/90 e por uma bolsa de Mestrado.

Por outro, o desenvolvimento do trabalho não seria possível sem o apoio quer material quer humano do INESC, particularmente do grupo Telemática do INESC Aveiro.

REFERÊNCIAS

- [1] J. Arnaldo Martins, Nelson Pacheco da Rocha, José Luis Oliveira, César Toscano "Integrated Management System for Interconnected Local Area Networks" - Mediterranean Electrotechnical Conference, Melecon'91, Ljubljana, Jugoslávia, Maio 1991
- [2] Abílio Pacheco de Carvalho, Nuno Eduardo da Silva, Nelson Pacheco da Rocha, Joaquim Sousa Pinto, "Sistema Distribuído de Monitorização de Redes Locais Ethernet", "Comunicações" (III Congresso Português das Comunicações), APDC, Fevereiro 1991.
- [3] "ISO IS9595-1: Information Processing - Open Systems Interconnection - Management Information Service Definition - Part 1: Overview", ISO, 1990.
- [4] Abílio Pacheco de Carvalho, Nelson Pacheco da Rocha, "Greco- Gestão de Redes de Comunicação, Sistema de Display Local e Remoto" INESC, Julho 1992.
- [5] J. Sousa Pinto, José L. Oliveira, J. Arnaldo Martins, "PC Workstation for Ethernet Monitoring and Management", SBT/IEEE, International Telecommunications Symposium, Brazil, 3-6 Sep. 1990.
- [6] Marshall T. Rose, "The Simple Book, An Introduction to Management of TCP/IP - based Internets", Prentice Hall, 1991.
- [7] Ben-Artzi Amatzia, Chandna A., Warriar U., "Network Management of TCP/IP Networks: Present and Future", IEEE Network Magazine, Julho de 1990.

- [8] Jeffrey D. Case, Mark S. Fedor, Martin L. Schoffstall, and James R. Davin, "A Simple Network Management Protocol". Request for Comments 1157, DDN Network Information Center, SRI International, Maio 1990.
- [9] Marshal T. Rose, Keith McCloghrie, "Concise MIB Definitions". Request for Comments 1212, Performance Systems International, Hughes LAN Systems, Março 1991.
- [10] Marshall T. Rose, "Management Information Base for Network Management of TCP/IP-based Internets: MIB-II, Request for Comments 1213, Performance Systems International, Março 1991.
- [11] Marshal T. Rose, Keith McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", Request for Comments 1155, DDN Network Information Center, SRI International, Maio 1990.
- [12] Nelson Pacheco da Rocha, Abílio Pacheco de Carvalho, José Luis Oliveira, "Defining Managed Objects for an Integrated Management System with Monitoring Agents", ERCIM Workshop on Network Management, Heraklion, Creta, Grécia, Outubro 1992.
- [13] Douglas Steedman, "ASN.1 - Abstract Syntax Notation One, The Tutorial and Reference", Technology Appraisals, 1990.
- [14] Mark Hoerth, "The RMON MIB - a New Standard for Remote Monitoring of Networks", Telecommunications, March 1992.
- [15] Steven Waldbusser, "Remote Network Monitoring MIB", Request for Comments 1271, Carnegie Mellon University, November 1991.
- [16] Marshall T. Rose, "The Open Book, A Practical Perspective on OSI", Prentice Hall, 1990.
- [17] Marshall T. Rose, "The ISO Development Environment - ISODE - User's Manual".
- [18] João Filipe Serra da Silva, Vitor Manuel Paulino Vargas "Compilador de ASN.1 baseado em informação em linguagem C", Manual do programador (versão 3.0), INESC, Março 1991.
- [19] Jeffrey D. Case, Keith McCloghrie, Marshall T. Rose and Steven Waldbusser, "Coming Soon to a Network Near You", Data Communication, Novembro 1992.