

Universidade Federal do Paraná

Departamento de Informática

Luis Carlos Erpen De Bona
Elias Procópio Duarte Jr.

O Delegado SNMP: Uma Ferramenta para Monitoração Distribuída

Relatório Técnico
RT-DINF 001/2001

Curitiba, PR
2001

O Delegado SNMP: Uma Ferramenta para Monitoração Distribuída

Luis Carlos Erpen De Bona
Elias Procópio Duarte Jr.

Universidade Federal do Paraná, Dept. de Informática
Cx. Postal 19081 – Curitiba – 81531-990 PR – Brasil
Telefone: 41-267-5244 Fax: 41-267-6874
e-mail: bona@inf.ufpr.br, elias@inf.ufpr.br

Resumo

Um sistema integrado de gerência de redes permite a monitoração e o controle dos diversos componentes de uma rede. A maioria dos sistemas de gerência segue o paradigma Gerente-Agente, no qual uma estação de gerência se comunica com agentes distribuídos pela rede usando um protocolo de gerência como o SNMP (Simple Network Management Protocol). Neste artigo apresentamos o Delegado SNMP que consiste de uma entidade intermediária entre gerente e agentes. Esta entidade permite descentralizar o sistema de gerência distribuindo a tarefa de monitoração da rede e seus componentes. O Delegado SNMP foi implementado ¹ como uma MIB (Management Information Base) que contém uma tabela dinâmica onde cada linha representa uma *thread* cuja tarefa corresponde à monitoração de um determinado objeto de um determinado agente. Com base em expressões lógicas dos valores dos objetos monitorados o Delegado SNMP emite alarmes. A distribuição das tarefas de monitoração é configurada pelos gerentes através da manipulação das linhas da tabela de delegação via aplicações SNMP. Como exemplo de aplicação, apresentamos uma implementação do algoritmo de diagnóstico distribuído Adaptive-DSD baseada no Delegado SNMP.

Abstract

An integrated network management system is a tool for controlling and monitoring of large computer networks. Most management systems follow the Manager-Agent paradigm, in which a central network management station uses a management protocol like SNMP (Simple Network Management Protocol) to communicate with a set of agents, which are distributed through the network. In this work we present the Delegado SNMP which is an intermediary management entity located between manager and an agent. This entity allows the construction of distributed network management systems which employ distributed monitoring. The Delegado SNMP has been implemented as a SNMP MIB (Management Information Base) in which a dynamic table has each entry corresponding to a *thread* whose task is to monitor a given object of a given agent. Based on logical expressions of the values of the monitored objects the Delegado SNMP may emit alarms. A manager distributes the monitoring tasks by configuring through SNMP applications a proper entry on the Delegado MIB table. We present an implementation of the Adaptive-DSD algorithm based on the Delegado SNMP as an example application.

Palavras-Chave: Gerência Distribuída, Gerência de Redes, SNMP, Diagnóstico, Algoritmos Distribuídos

¹Disponível via ftp no endereço <ftp://ucd-snmp.ucdavis.edu/contrib/Delegado.tar.gz>

1 Introdução

Com o aumento da complexidade e popularidade das redes de computadores os sistemas integrados de gerência vêm tornando-se cada vez mais necessários. Um sistema de gerenciamento em geral contém quatro componentes: nodos gerenciados, contendo cada qual um agente, uma ou mais estações de gerência de rede, um protocolo de comunicação e informações de gerência [1]. A maioria dos sistemas de gerência baseados em SNMP (Simple Management Network Protocol) [2, 3, 4] segue o paradigma Gerente-Agente no qual a tarefa de gerência é centralizada nas estações de gerência, como mostrado na figura 1.

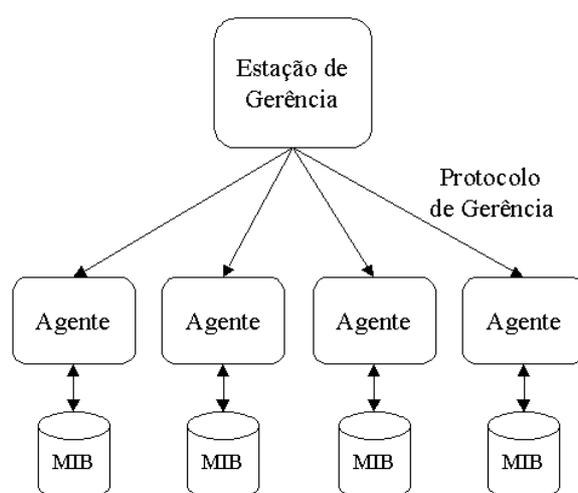


Figura 1: Sistema Centralizado de Gerenciamento.

Os nodos gerenciados podem ser praticamente qualquer componente conectado à rede: computadores, roteadores, terminais, impressoras, modems, hubs ou qualquer dispositivo que possa disponibilizar alguma informação à rede. O agente é a entidade que permite que o nodo gerenciado esteja acessível às estações de gerência. A estação de gerência, por sua vez, é um sistema que executa o protocolo de gerência de rede e uma ou mais aplicações de gerência. Estas aplicações são também chamadas de gerentes.

O protocolo de gerência permite à estação de gerência monitorar e controlar os nodos gerenciados. Isto implica em se poder ler - monitorar - , e alterar - controlar - informações de gerência de cada nodo monitorado. Além disto, existe uma operação alarme que permite que um nodo monitorado avise a estação de gerência sobre um evento extraordinário.

As informações de gerência constituem o conjunto de dados pertinentes a um nodo gerenciado. Cada unidade de informação é representada por um objeto de gerência. A coleção destes objetos é chamada de Base de Informação de Gerência ou “Management

Information Base” (MIB). A apresentação dos dados colhidos do dispositivo gerenciado na forma da MIB é uma das atribuições do agente.

Este trabalho propõe uma entidade de gerência auxiliar, chamada *Delegado SNMP*. O Delegado é uma entidade intermediária entre gerentes e agentes permitindo a descentralização da tarefa de gerenciamento. O Delegado SNMP tem estrutura semelhante à da Routing Proxy (RProxy), proposta por Duarte e Mansfield [5]. A RProxy é uma entidade que permite ao gerente acessar um determinado agente através de uma rota alternativa à rota IP. Esta proxy é composta por agentes quaisquer da rede que contenham a MIB adequada. O objetivo da rota alternativa dada pela RProxy é permitir que elementos da rede se comuniquem quando a rota determinada pela camada de rede não estiver funcionando adequadamente.

O Delegado SNMP pode ser usado como ferramenta de monitoração distribuída. Os gerentes delegam a tarefa de monitoração a um ou mais agentes. Estes agentes passam a realizar a tarefa de monitoração automaticamente, sendo chamados então de Delegados SNMP. Eles ficam responsáveis pela observação de objetos pré-definidos, e caso uma condição pré-definida seja detectada o Delegado emite um alarme. Desta forma é possível distribuir a tarefa de monitoração como mostra a figura 2. Nesta figura o gerente distribuiu a tarefa de monitorar seis nodos para dois Delegados SNMP.

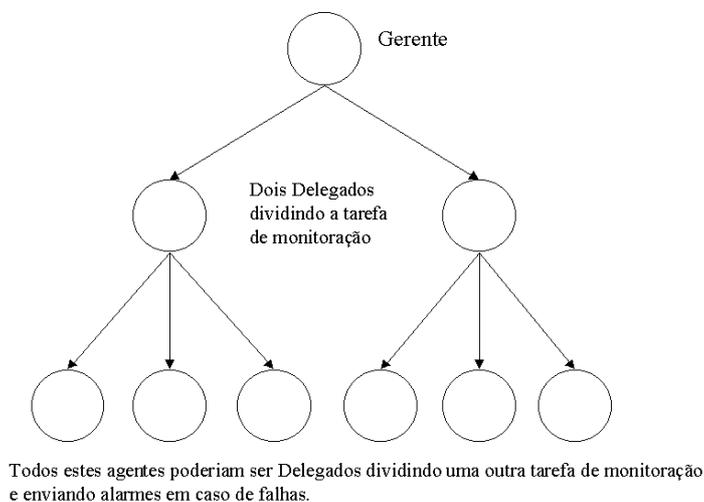


Figura 2: Distribuindo a tarefa de monitoração.

O Delegado SNMP também pode ser utilizado como uma ferramenta para monitoração tolerante a falhas. Para isto a distribuição da tarefa de monitoração deve ser feita de maneira que cada objeto seja monitorado por dois ou mais Delegados. A figura 3 mostra a utilização de dois Delegados para aumentar a tolerância a falhas.

O restante deste artigo está organizado da seguinte maneira. Na seção 2 apresentamos

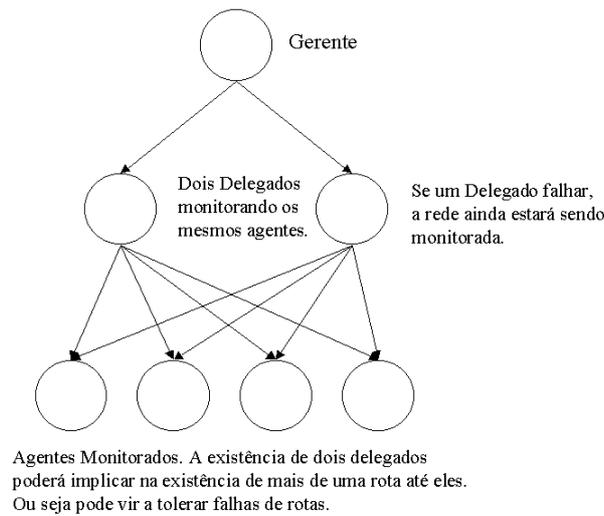


Figura 3: Utilizando o Delegado SNMP para aumentar a tolerância a falhas.

outros trabalhos relacionados à gerência distribuída. Na seção 3 passamos a apresentação mais detalhada do Delegado SNMP. Na seção 4 propomos um modelo para implementação do algoritmo Adaptive-DSD usando o Delegado SNMP. Já na seção 5 apresentamos conclusões e trabalhos futuros.

2 Gerência Distribuída

O Delegado SNMP descrito neste trabalho pode ser usado como uma ferramenta de gerência distribuída. Outras ferramentas, padrões e trabalhos relacionados serão apresentados nesta seção.

A maior parte dos sistemas de gerência baseados em SNMP segue o modelo Gerente-Agente. Neste modelo existem uma ou poucas estações de gerência para um grande número de agentes. A tarefa de gerência é centralizada nas estações de gerência. Os agentes, em geral, limitam-se a responder mensagens e eventualmente enviar alarmes aos gerentes comunicando a ocorrência de um determinado evento [6, 7].

Este modelo centralizado ocasiona uma saturação de tarefas para as estações de gerência que serão responsáveis por monitorar e controlar toda rede. A monitoração consiste em operações de leitura de determinado objeto que são repetidas a cada intervalo de tempo. Já o controle, consiste usualmente em avaliar os dados da monitoração e alterar valores dos objetos nos agentes que provocarão mudanças de estados nos nodos controlados. A centralização pode levar a uma má distribuição das tarefas necessárias para gerência efetiva da rede.

Uma das primeiras ferramentas práticas para gerência distribuída foi a *Routing Proxy*

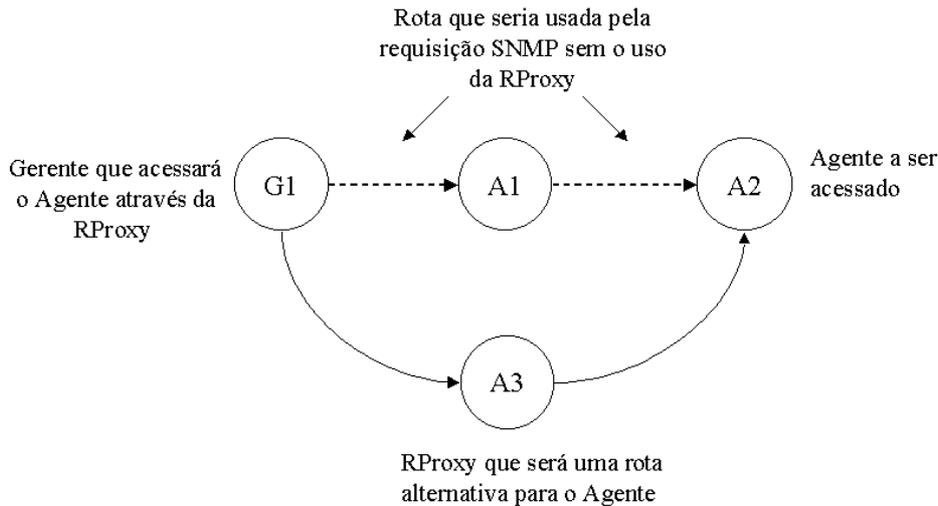


Figura 4: Utilização da Rproxy para obter rotas alternativas

(RProxy) [5, 8]. As mensagens trocadas entre o gerente e os agentes acessados em geral seguem a rota determinada pela camada de rede. Com isto, um gerente não pode colher informações por uma rota alternativa de um agente que esteja inacessível pela rota determinada pela camada de rede. A RProxy permite que um gerente acesse um agente por uma rota alternativa [9, 10]. Na figura 4 o gerente G1 acessa o agente A2 por uma rota alternativa através da RProxy no agente A3.

A RProxy foi implementada como uma MIB [11]. A MIB da RProxy tem quatro objetos. Três deste objetos determinam a requisição feita pela RProxy. Este objetos indicam o agente e o objeto acessado e a community, que é uma chave de acesso para o objeto. O quarto objeto indica o resultado da requisição. Quando o gerente lê este objeto a RProxy executa um *snmpget* no agente e no objeto indicados na RProxy.

O DisMan (*Distributed Management Working Group*) [12] é um grupo do IETF que está trabalhando em problemas relacionados a gerência distribuída baseada em SNMP. Uma de suas atividades é a definição de um conjunto de MIB's que visam permitir sistemas distribuídos de gerência. Atualmente o DisMan já publicou seis documentos, quatro Internet Drafts e dois RFC's. Alguns de seus principais componentes são descritos a seguir.

A *ExpressionMIB* [13] é uma MIB que permite criar novos objetos a partir de expressões lógicas sobre outros objetos existentes. Estes novos objetos comportam-se exatamente como um objeto de qualquer MIB. Isto permite que o gerente no lugar de monitorar um conjunto de objetos possa monitorar uma expressão resultantes deste conjunto de objetos.

A *EventMIB* [14] serve para monitorar objetos locais ou remotos tomando algumas

ações quando uma determinada condição é atingida. O Delegado SNMP possui funcionalidade semelhante a da EventMIB, entretanto o Delegado SNMP é muito mais enxuto. A Event MIB possui quatro tabelas diferentes: triggers (gatilhos), objects (objetos), events (eventos) e notifications (notificações). Na tabela de gatilhos são definidos os objetos monitorados e como estes se relacionam com os eventos. A tabela de objetos é um repositório de objetos que podem ser anexados a uma notificação. A tabela de eventos descreve as ações a serem tomadas quando um gatilho for disparado, as ações podem ser o envio de notificações e/ou a alteração do valor de objetos de gerência. A tabela de notificação define algumas notificações padrão que podem ser utilizadas para responder aos eventos. Algumas destas tabelas ainda contêm várias outras tabelas.

A *ScriptMIB* [15] permite que funções de gerência sejam delegadas para agentes distribuídos. Neste caso delegar significa transferir e executar estas funções de gerência. As funções são *scripts* de gerência escritos em qualquer linguagem podendo até mesmo serem código objeto. A implementação ideal permitiria a delegação de qualquer função de gerência independentemente da linguagem usada [16]. A ScriptMIB ainda permite controlar e manipular os scripts. Os parâmetros e os resultados dos scripts também são acessíveis na ScriptMIB. Outras MIBs definidas pelo DisMan [12] são a *NotificationMIB*, a *Scheduling MIB* e a *MIB para Ping, Traceroute e Lookup Remotos*.

3 O Delegado SNMP

O Delegado SNMP, proposto neste trabalho, é uma entidade intermediária entre o gerente e os agentes. A principal motivação é atenuar a centralização da tarefa de monitoração no modelo Gerente-Agente. O Delegado SNMP permite que a tarefa de monitoração seja distribuída. Para que a descentralização não seja feita sob pena de uma diminuição no desempenho da rede, o Delegado não pode representar um custo adicional relevante. Para atender a esse requisito o Delegado foi proposto como uma MIB bastante simples que pode ser incluída nos agentes. Assim que a MIB do Delegado é incluída a um agente qualquer, ele passa imediatamente a ter a funcionalidade de um Delegado SNMP.

A MIB do Delegado SNMP possui duas propriedades primordiais: capacidade de realizar tarefas de monitoração e capacidade de informar os gerentes de um evento anormal. Por tratar-se de uma MIB, o Delegado pode ser configurado por um gerente qualquer. Esta configuração será feita através das próprias aplicações SNMP. Uma vez “configurada”, a MIB do Delegado terá a função de monitorar um conjunto qualquer de agentes. Depois de configurado, o Delegado passará a funcionar de maneira autônoma, ou seja, ele fica monitorando constantemente e automaticamente os objetos determinados. A comunicação do delegado com o gerente apenas voltará a ocorrer caso o Delegado verifique

uma condição pré-determinada em algum dos objetos monitorados. A comunicação do Delegado com o gerente nestes casos de aviso é através de alarmes *trap SNMP*, como apresentado na figura 5.

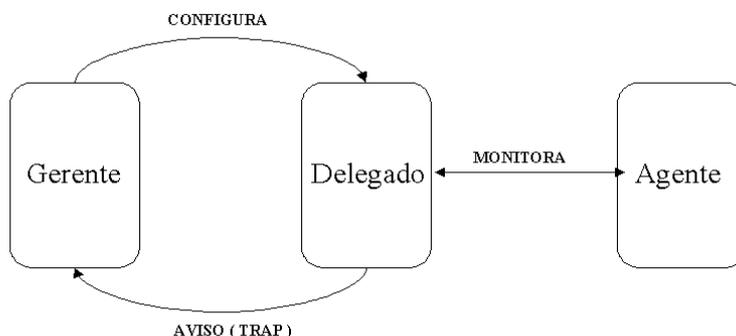


Figura 5: O Delegado SNMP, intermediário entre o gerente e o agente.

O Delegado SNMP é uma MIB composta por objetos que permitem a especificação de uma ou mais tarefas de monitoração, e objetos que disponibilizam informações sobre o estado do próprio Delegado. O Delegado SNMP possui duas partes. Uma parte contém o *status* (OID - Object Identifier - 1) onde estão os objetos de informações e controles gerais do Delegado SNMP; a outra parte contém a *table* (OID 2) onde está a tabela dinâmica na qual cada linha representa uma tarefa de monitoração.

Os objetos da árvore status são apenas dois. O primeiro, bastante simples, indica o número de linhas da tabela de tarefas de monitoração. O segundo objeto é usado para controle da tabela, sendo também um método alternativo para a criação de novas linhas na tabela. Ao ler este objeto uma nova linha é criada na tabela de delegação, sendo o índice desta linha o valor de retorno para o gerente.

As linhas da tabela indicam as tarefas de monitoração. Cada linha é composta por objetos para controlar a linha e para determinação do objeto a ser monitorado, configuração do agente acessado, destino e condição para emissão do alarme e intervalo de teste.

O objeto de controle da linha, *dgStatus*, é usado para verificar se determinada linha está ativa, ou para excluir ou adicionar uma linha. Os objetos necessários para especificar a monitoração correspondem às informações necessárias para execução de um comando *snmpget*. São definidos os seguintes objetos: *dgGetParam*, *dgIpAddress*, *dgObjectId*, *dgCommuntiy* e *dgGetParam*. Respectivamente estes objetos indicam o agente acessado (endereço IP), o identificador do objeto (OID) a ser monitorado, chave de segurança (COMMUNITY) e os parâmetros de configuração do agente.

O objeto *dgIpAddress* recebe o endereço IP do agente a ser acessado na forma numérica ou pelo nome. O *dgObjectId* é o OID de um objeto qualquer desde que o valor deste objeto

<i>IpAddress</i>	<i>Community</i>	<i>ObjectId</i>	<i>Result</i>	<i>LCondition</i>	<i>Manager</i>	<i>IntervalTest</i>
arara	private	Icmp.icmpEchoReps.0	15	>1000	pardal	30
sabia	private	Tcp.tcpInErros.0	05	>= 20	pardal	15
azulao	private	Udp.udpNoPorts.0	0	!=0	pardal	20

Tabela 1: Exemplo de algumas tarefas de monitoração na tabela do Delegado SNMP

seja um inteiro ou contador. No objeto *dgGetParam* estão os valores de configuração do *snmpget*, cuja sintaxe é a mesma do comando *snmpget* to pacote UCD-SNMP [14]. A validação dos valores destes objetos não é feita no momento da atribuição. A validação é feita pela constatação de erros na montagem do pacote de requisição SNMP ou na recepção do pacote de resposta SNMP proveniente do agente.

Existe também um objeto, apenas para leitura, *dgResult* que armazena o resultado obtido na última iteração do Delegado. O *dgResult* ao ser lido faz com que o Delegado SNMP comece a realizar a tarefa de monitoração delegada, antes de um *snmpget* neste objeto a linha não estará ativa.

Para configuração do alarme o Delegado SNMP usa dois objetos: *dgLCondition* e *dgManager*. O objeto *dgLCondition* indica uma condição lógica de teste aplicada ao objeto *dgResult* que quando falha força o envio do alarme. O objeto *dgManager* indica o IP do gerente alvo do alarme.

A condição de teste é composta por um operador lógico (>, <, >=, <=, == e !=) seguido de um espaço e de um número. Um exemplo de condição seria “>= 5”. A verificação da sintaxe desta condição é feita na atribuição do objeto. Caso a sintaxe esteja incorreta o gerente receberá uma mensagem de erro. O outro objeto contém o endereço IP de um gerente qualquer que esteja preparado para receber os alarmes. Por fim, o objeto *dgIntervalTest* indicará o intervalo para a realização da requisição e teste do objeto monitorado. Este tempo é dado em segundos, sendo o intervalo mínimo de 1 segundo e o intervalo máximo é 2 elevado 32 segundos.

Um Delegado SNMP pode receber um número indeterminado de tarefas de monitoração. A tabela 1 mostra um exemplo do conteúdo da tabela de um Delegado SNMP.

Para que cada tarefa de monitoração do Delegado possa ser executada de forma independente é necessária a utilização de *threads*. Uma thread é uma unidade de trabalho independente. Ela é executada sequencialmente e é “interrompível” para que o processador possa alternar para outra thread. *Multi-threading* é uma técnica útil para aplicações que executam um número de tarefas independentes que não têm necessidade de serem serializadas [17]. Para cada linha da tabela uma thread é criada. Esta thread fica em *loop* infinito até que o valor do objeto *status* deixe de ser ativo. De forma simplificada a thread teria o seguinte funcionamento:

```

Repetir enquanto a linha for ativa
  Enviar requisicao do objeto monitorado
  Se resultado sem erro
    Armazena Resultado
    Aplica condicao logica sobre resultado, se falhar emite alarme
  Se resultado com erro
    Envia Alarme
  Dorme intervalo de tempo

```

Para fazer as requisições e enviar os alarmes foi preciso criar funções que fossem *multi-thread*. Para isto, baseamo-nos nas implementações do ucd-snmp e modificamos o código para torná-lo *thread-safe*.

A MIB do Delegado SNMP foi implementada integrada ao UCD-SNMP [18], um pacote SNMP de domínio público mantido pela Universidade da Califórnia em Davis (UCDavis). Este pacote foi desenvolvido em linguagem C, e permite a inclusão de novas MIB's, de forma modular, aos agentes SNMP. O sistema foi implementado utilizando computadores PC/Pentium executando sistema operacional Linux.

4 Uma Implementação do Algoritmo de Diagnóstico Adaptive-DSD Baseada no Delegado SNMP

Nesta seção apresentamos um exemplo de aplicação do Delegado SNMP na implementação do algoritmo de diagnóstico Adaptive-DSD (*Adaptive Distributed System-Level Diagnosis*)[19]. Trata-se um algoritmo distribuído tolerante a falhas útil na gerência de falhas de redes locais.

No algoritmo Adaptive-DSD identificadores sequenciais de 0 a $N - 1$ são atribuídos aos nodos. Os nodos executam testes sequencialmente até encontrarem um nodo sem-falha, ou seja o nodo i testa o nodo $i + 1$. Se o nodo $i + 1$ estiver falho, o nodo i testa o nodo $i + 2$ e assim por diante. Desta forma o grafo de testes do sistema é um anel. A figura 6 mostra nodos rodando o Adaptive-DSD. As arestas direcionadas representam testes, e os nodos marcados são os considerados falhos. Na figura o nodo 0 encontrou o nodo 1 falho e passou a testar do nodo 2.

Ao realizar um teste sem-falhas o nodo testador obtêm informações de diagnóstico do nodo testado. Estas informações incluem os resultados dos testes realizados pelo nodo testado, além das informações de diagnóstico por ele obtidas. A definição de latência deste algoritmo é em termos de *rodadas de testes*. Uma rodada de testes é definida como o período de tempo no qual todos os nodos sem-falha do sistema testam um nodo sem-falha.

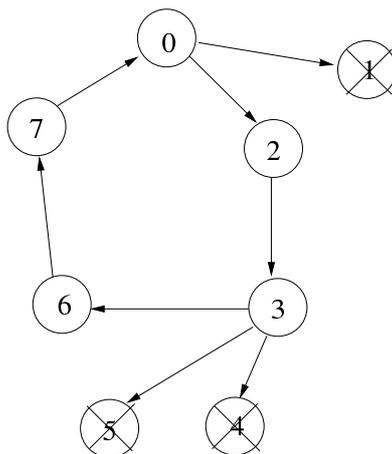


Figura 6: Exemplo de conjunto de testes executados por nodos rodando Adaptive-DSD.

Para implementar o algoritmo Adaptive-DSD usando o Delegado SNMP, deve-se disponibilizar ao Delegado uma tabela que mostra o mapeamento do endereço IP de cada estação para seu identificador sequencial dentro do algoritmo. Esta tabela contém, para cada entrada, o estado do agente correspondente: falho ou sem-falha. Cada Delegado está configurado para monitorar um objeto do agente cujo identificador é seguinte ao seu. Se tal agente estiver falho, o Delegado parte para a monitoração do objeto do agente seguinte, e assim por diante. Caso um novo evento seja detectado, isto é, um agente falho torna-se sem-falha, ou vice-versa, o Delegado emite uma *trap* para todos os agentes informando-os sobre o novo evento. Desta forma, a latência do algoritmo pode ser reduzida. Entretanto como o serviço de alarmes é não-confiável os nodos devem continuar obtendo informações de diagnóstico daqueles nodos testados sem falha.

5 Conclusão

Neste trabalho propusemos e implementamos uma entidade de gerência chamada *Delegado SNMP*. O Delegado SNMP funciona como um intermediário entre gerentes e agentes, num sistema integrado de gerência. O gerente delega a esta entidade a monitoração de um conjunto de agentes da rede, de forma que caso ocorra uma condição pré-determinada no agente monitorado, o Delegado envia um alarme ao gerente. Esta monitoração acontece em intervalos que são determinados pelo gerente. O Delegado SNMP tem o objetivo de servir de ferramenta para descentralizar o sistema de gerência distribuindo a tarefa de monitoração. A implementação desta entidade representa o resultado prático da gerência por delegação.

O Delegado SNMP foi implementado como uma MIB. Esta MIB foi codificada em linguagem C utilizando as bibliotecas padrão em conjunto com a biblioteca SNMP da

UCDavis e a biblioteca que permite o uso de *threads* . O código resultante é modular podendo ser utilizado em qualquer agente ucd-snmp versão 3.6.1.

Como trabalhos futuros propomos a criação de outros algoritmos distribuídos tolerantes a falhas usando o Delegado além de dotar o Delegado de inteligência para permitir um aumento na sua autonomia.

Referências

- [1] Rose, M. T. *The Simple Book - An introduction to Internet Management*, 2 ed., Prentice-Hall, Englewood Cliffs, NJ, 1994.
- [2] McCloghrie, K., and M. Rose, Editors, "Management Information Base for Network Management of TCP/IP based internet: MIB-II", *RFC 1213*, Hughes LAN Systems, Performance Systems International, 1991.
- [3] J. Case, M. Fedor, M. Schoffstall, e J.Davin, "Simple Network Management Protocol (SNMP)", *RFC 1157*, SNMP Research, Performance Systems International, Performance Systems International, MIT Laboratory for Computer Science, 1990.
- [4] M. Rose, and K. McCloghrie, "Structure of Management Information for TCP/IP based internets," *RFC 1155*, Performance International, Hughes Lan Systems, 1990.
- [5] E. P. Duarte Jr., G. Mansfield, S. Noguchi, and M. Miyazaki, "Fault-Tolerant Network Management," in *Proc. ISACC'94, Monterrey, Mexico, 1994*.
- [6] A. S. Tanenbaum, *Redes de Computadores*, Campus, Rio de Janeiro, 1997.
- [7] D. E. Comer and D. L. Stevens, *Internetworking with TCP/IP*, v. 2, 2 ed, Prentice-Hall, Englewood Cliffs, NJ, 1994.
- [8] E. P. Duarte Jr., and M. A. Musicante, "Formal Specification of SNMP MIB's Using Action Semantics: The Routing Proxy Case Study," *Proc. IEEE/IFIP International Symposium on Integrated Network Management IM'99*, Boston, May 1999.
- [9] E. P. Duarte Jr., G. Mansfield, T. Nanya, and S. Noguchi, "Improving the Dependability of Network Management Systems," in *International Journal of Network Management*, Vol. 8, No.4, pp. 244-253, July-August 1998.
- [10] E. P. Duarte Jr., "SNMP-Based Fault-Tolerant Network Monitoring" em *XXV Seminário Integrado de Hardware e Software (SEMISH'98)*, Anais do XVIII Congresso Nacional da Sociedade Brasileira de Computação, Belo Horizonte, 1998.
- [11] A. B. Silva, J. R. G. Rodrigues, e S.I. Yuassa, *Implementacao de uma Proxy de Roteamento SNMP Usando Tabelas Dinâmicas*, Trabalho de Conclusão de Curso, orientador: Elias P. Duarte Jr., Bacharelado em Informatica, UFPR, Curitiba, 1999.
- [12] *Distributed Management (DisMan) Charter* - <http://www.ietf.org/html.charters/disman-charter.html>.

- [13] B. Stewart, “Distributed Management Expression MIB”, *Internet Draft Expression MIB*, Cisco Systems Inc, 1999.
- [14] B. Stewart, R. Kavasseri, “Distributed Management Event MIB”, *Internet Draft Event MIB*, Cisco Systems Inc., Cisco Systems Inc, 1999.
- [15] Levi, D. and J. Schoenwaelder, “Definitions of Managed Objects for the Delegation of Management Scripts”, *RFC 2592*, Nortel Networks, TU Braunschweig, 1999.
- [16] Maurício J. Trezub, and Juliano F. Faoro, *Uma Implementação da ScriptMIB Usando TCL*, Trabalho de Conclusão de Curso, Orientador: Elias P. Duarte Jr., Bacharelado em Informática, UFPR, 1999.
- [17] Stallings, William. *Operating Systems: Internals and Design Principles*. 3 ed. New Jersey. Prentice-Hall, Englewood Cliffs, NJ, 1998.
- [18] *UCD-SNMP* <http://ucd-snmp.ucdavis.edu>
- [19] R.P. Bianchini, and R. Buskens, “Implementation of On-Line Distributed System-Level Diagnosis Theory,” *IEEE Transactions on Computers*, Vol. 41, pp. 616-626, 1992.

Apêndice 1 - Descrição ASN.1 da MIB do Delegado SNMP

```
DELEGATE-MIB DEFINITIONS ::= BEGIN
```

```
-- Delegate MIB.
--
-- Authors: Eduardo Kuminek
--          Luis Carlos Erpen De Bona
--
-- Federal University of Parana
-- Department of Informatics, Curitiba, Brazil
--
-- July, 1999
--
```

```
IMPORTS
```

```
    enterprises, OBJECT-TYPE
        FROM SNMPv2-SMI
    DisplayString
        FROM SNMPv2-TC;
```

```
ufpr          OBJECT IDENTIFIER ::= { enterprises 2022 }
delegate      OBJECT IDENTIFIER ::= { ufpr 13 }
status        OBJECT IDENTIFIER ::= { delegate 1 }
table         OBJECT IDENTIFIER ::= { delegate 2 }
```

```
dgTableCount OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of rows in the Delegation Table"
    ::= { status 1 }
```

```
dgCreateRow OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Creates a new row, and gives back its number"
    ::= { status 2 }
```

```
dgRowStatus OBJECT-TYPE
    SYNTAX INTEGER
```

ACCESS read-write
STATUS mandatory
DESCRIPTION

"The Delegate keeps a dynamic table of delegation.
Before a row is used, the manager should
choose an unused entry (e.g. walking through the
Delegate's MIB) and allocate that row, setting
the dgRowStatus to 4.
When the row is not needed anymore, it should be free-d.
This object may take the following values:

- 1 active: Delegation is running
- 2 notInService: not created row
- 3 notReady: Delegation isn't running
- 4 or 5 create: allocate row
- 6 destroy: free row"

::= { table 1 }

dgGetParam OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..255))
ACCESS read-write
STATUS mandatory
DESCRIPTION

"SNMPGET Parameter Line"

::= { table 2 }

dgIpAddress OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..255))
ACCESS read-write
STATUS mandatory
DESCRIPTION

"IP address of agent to be accessed by the delegate"

::= { table 3 }

dgCommunity OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..255))
ACCESS read-write
STATUS mandatory
DESCRIPTION

"Community to be used by the delegate"

::= { table 4 }

dgObjectId OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..255))
ACCESS read-write
STATUS mandatory

```

DESCRIPTION
    "Object ID to be accessed by the delegate"
 ::= { table 5 }

dgResult OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Reading this, the delegate will start running"
 ::= { table 6 }

dgLCondition OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Logical Condition to generate a Trap"
 ::= { table 7 }

dgIntervalTest OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Delegation Monitoring/Test Interval"
 ::= { table 8 }

dgManager OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Target Manager for the generated trap"
 ::= { table 9 }

END

```