AVALIAÇÃO DO DESEMPENHO DE AGENTES MÓVEIS NO GERENCIAMENTO DE REDES

Marcelo Gonçalves Rubinstein

PROPOSTA DE TEMA DE TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

Prof. Otto Carlos Muniz Bandeira Duarte, Dr.Ing.

Prof. Edmundo Madeira, D.Sc.

Prof. Carlos Becker Westphall, Dr.

Prof. José Ferreira de Rezende, Dr.

RIO DE JANEIRO, RJ - BRASIL

AGOSTO DE 1999

Resumo da Proposta de Tema de Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

AVALIAÇÃO DO DESEMPENHO DE AGENTES MÓVEIS NO GERENCIAMENTO DE REDES

Marcelo Gonçalves Rubinstein

Agosto/1999

Orientador: Otto Carlos Muniz Bandeira Duarte

Programa: Engenharia Elétrica

Este trabalho visa investigar o desempenho de agentes móveis no gerenciamento de redes, de modo a descobrir quando os agentes móveis podem aumentar a eficiência do gerenciamento. O desempenho dos agentes móveis é comparado com o do SNMP por meio de simulações de várias tarefas de gerenciamento. Os resultados indicam que os agentes móveis possuem um melhor desempenho quando o número de elementos gerenciados encontra-se entre dois limites: um inferior associado ao número de mensagens que passam por um enlace de gargalo que liga a estação de gerenciamento aos elementos gerenciados e um superior relacionado ao tamanho crescente do agente móvel. Uma melhora significativa do desempenho é obtida quando o agente móvel retorna à estação de gerenciamento após visitar um número fixo de nós. Além disto, os agentes móveis são menos sensíveis à latência e à banda passante do enlace de gargalo que o SNMP, porém sofrem maior influência das tarefas a serem realizadas.

Abstract of Thesis Proposal presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

PERFORMANCE EVALUATION OF MOBILE AGENTS IN NETWORK MANAGEMENT

Marcelo Gonçalves Rubinstein

August/1999

Advisor: Otto Carlos Muniz Bandeira Duarte

Department: Electrical Engineering

This work aims to investigate mobile agent performance in network management, in order to find when mobile agents can improve the management efficiency. Mobile agent performance is compared with the SNMP one in several management task simulations. Results show that the mobile agent performs better when the number of managed network elements ranges between two limits: an inferior bound associated with the number of messages that traverse a bottleneck link that connects the management station to the managed elements and a superior bound related to incremental size of mobile agent. A significant improvement is achieved when the mobile agent returns to the management station after visiting a fixed number of nodes. Moreover, mobile agents are less sensitive to latency and data rate of the bottleneck link than the SNMP, but the agents are more sensitive to the tasks to be performed.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos

AM : Agente Móvel;

ATM : Asynchronous Transfer Mode;

CMIP: *Common Management Information Protocol*;

CORBA: Common Object Request Broker Architecture;

EG : Estação de Gerenciamento;

FIPA: *Foundation for Intelligent Physical Agents*;

IETF: *Internet Engeneering Task Force*;

IP : *Internet Protocol*;

ISO : *International Standards Organization*;

MASIF : *Mobile Agent System Interoperability Facility*;

MbD : Gerenciamento por delegação (*Management by Delegation*);

MIB : *Management Information Base*;

MTU : Unidade máxima de transmissão (*Maximum Transfer Unit*);

ns : *network simulator*;

OMG : *Object Management Group*;

PDU : Unidade de dados do protocolo (*Protocol Data Unit*);

RMON: Monitoração remota (*Remote MONitoring*);

RPC : Chamada remota de procedimentos (*Remote Procedure Call)*;

SNMP : *Simple Network Management Protocol*;

TCP : *Transmission Control Protocol*;

UDP : *User Datagram Protocol*;

URL : *Universal Resource Locator.*

Capítulo 1

Introdução

COM o extraordinário desenvolvimento da rede mundial de computadores, a Internet, o volume de informações disponível pelo mundo, o custo de movimentação de dados e a pequena experiência em computação de muitos usuários têm sido motivo de preocupação para a comunidade de redes e para os desenvolvedores de aplicativos. O volume de informações disponível em diferentes sítios por todo o mundo não tem sido acompanhado pela capacidade de um usuário de procurar, selecionar e obter estas informações. Além disto, a movimentação de grandes volumes de dados através da rede possui um custo significativo e o número de usuários com pouco conhecimento em computação tem crescido e deverá continuar a aumentar rapidamente.

Uma possível solução para estes problemas consiste na utilização de agentes, que são programas que ajudam um usuário a realizar tarefas, agindo em interesse deste usuário. Caso os agentes possam executar suas tarefas em diferentes computadores da rede, estes são denominados agentes móveis. Estes agentes podem mover-se para o lugar onde os dados estão armazenados e, utilizando inteligência, selecionar as informações que o usuário necessita; economizando-se banda passante, tempo e dinheiro.

O modelo cliente-servidor tem sido utilizado há anos em sistemas distribuídos. Na maioria dos casos, o cliente e o servidor são executados em máquinas diferentes (figura 1.1), por conseqüência, aumentando os esforços e os custos de desenvolvimento e de manutenção de clientes em diversas plataformas. A chamada remota de procedimentos (*Remote Procedure Call* - RPC) é o mecanismo mais utilizado para a comunicação entre clientes e servidores. O modelo de programação remota é utilizado pelos agentes a fim de melhorar o desempenho através da eliminação do tráfego intermediário na rede. Neste modelo, um agente representando um cliente pode ser transportado para um hospedeiro junto com o procedimento a ser executado a fim de interagir com o servidor de agentes[1] deste hospedeiro, realizando suas tarefas localmente [1] (figura 1.2). Existem duas vantagens principais em se utilizar o modelo de programação remota no lugar da RPC. A primeira consiste no fato que somente resultados selecionados de pedidos serão enviados, diminuindo o tráfego na rede pois não haveria necessidade de transmitir comandos, todos os dados e resultados intermediários. Quanto à segunda, como um agente sabe como acessar os dados que necessita utilizar no hospedeiro, para realizar sua tarefa, o hospedeiro pode ser simplificado pois não precisa reconhecer uma linguagem complexa de pedidos ou saber como realizar cálculos complexos.

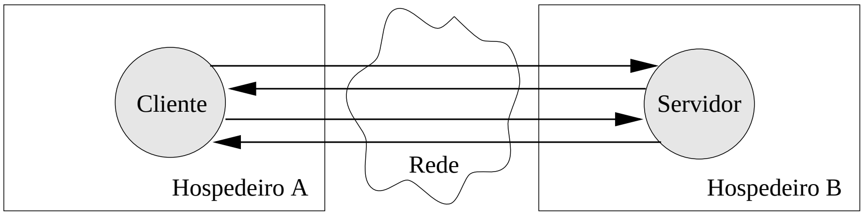

Figura 1.1: Modelo Cliente-Servidor.

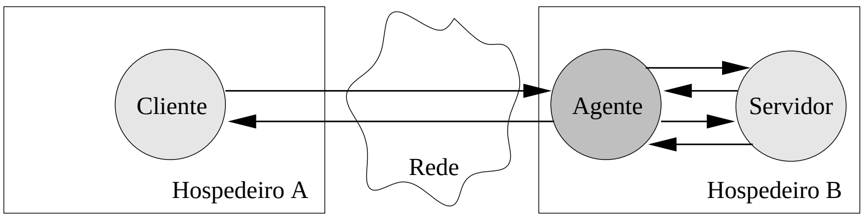

Figura 1.2: Modelo de Programação Remota.

[1]Não confundir com o servidor do modelo cliente-servidor.

1.1 Características de Agentes

Os agentes são encarregados de realizar tarefas e podem operar sobre requisitos fornecidos. Para executar estas tarefas, os agentes podem utilizar as seguintes capacidades: agência, inteligência e mobilidade [2].

A agência é o grau de autonomia e autoridade investida no agente, medida através de sua interatividade. Existem três níveis de agência: interatividade com o usuário, interatividade com a aplicação e interatividade com o agente, que indicam interações respectivamente com o usuário, com o usuário e as aplicações, e com o usuário, as aplicações e outros agentes.

A inteligência é o grau de raciocínio e comportamento aprendido: a habilidade do agente em aceitar as metas do usuário e realizar a tarefa a ele conferida. Existem quatro níveis de inteligência: preferência, raciocínio, planejamento e aprendizado. No mínimo, deve existir alguma declaração de preferências, talvez na forma de regras, com uma máquina de inferência ou algum outro mecanismo de raciocínio para agir sobre estas preferências. Níveis maiores de inteligência incluem um modelo do usuário ou alguma outra forma de entender e raciocinar a respeito do que o usuário quer que seja feito, podendo o agente planejar os meios de alcançar este objetivo. Níveis mais altos de inteligência podem ser obtidos através de sistemas que aprendem a se adaptam ao seu ambiente, em termos dos objetivos do usuário e dos recursos disponíveis ao agente.

A mobilidade corresponde ao grau de movimento dos agentes através da rede. Um agente pode ser fixo, residindo na máquina cliente (por exemplo, para gerenciar uma interface com o usuário) ou instanciado no servidor, ou móvel, criado em uma máquina e enviado para outra, por exemplo para a execução em um ambiente seguro. Um agente móvel pode ser transportado de uma máquina para outra no meio de sua execução e carregar com ele dados acumulados sobre o seu estado.

1.2 Estrutura dos Agentes Móveis

Os agentes móveis consistem de código, estado e atributos. O código de um agente móvel é um programa que define o comportamento do agente. Obviamente um código de um agente pode ser escrito em qualquer linguagem, porém um agente deve poder ser executado de uma mesma maneira em qualquer hospedeiro para o qual possa se mover [3]. Para isto, as linguagens que são interpretadas diretamente ou compiladas para uma linguagem intermediária baseada em interpretador que possa ser facilmente transportada e executada sem recompilação são as mais adequadas. Como exemplos, tem-se: Tcl, Perl e Java, sendo que o atual interesse em sistemas de agentes móveis é devido quase que inteiramente à grande adoção de Java [4].

Além do código, um agente carrega um estado que permite que suas ações subseqüentes sejam baseadas em suas ações passadas. Com isto, um agente pode retomar o trabalho de onde parou, após mudar para outro hospedeiro. O estado de um agente contém o ponto de execução no novo hospedeiro e as variáveis necessárias para a sua execução.

A terceira parte de um agente móvel consiste de atributos que descrevem o agente, seus requisitos e sua história para a infra-estrutura. Isto inclui dados como um identificador único de agente, o dono do agente como um endereço destinatário para resultados intermediários, mensagens de erro ou reclamações quanto a um mau comportamento de um agente, o tempo e o lugar de origem e a história do movimento.

1.3 Infra-estrutura de Agentes

Os agentes devem interagir com os hospedeiros para utilizar os serviços destes hospedeiros e negociar serviços com outros agentes. Os agentes móveis devem ainda poder se mover em redes de computadores. O ambiente que permite estas ações é denominado infra-estrutura de agentes e a entidade que provê suporte para os agentes em um hospedeiro particular é chamada servidor de agentes. Além destes serviços básicos, uma infra-estrutura de agentes deve também prover segurança a fim de proteger o hospedeiro (e os agentes) de acessos não autorizados e espionagem.

Uma infra-estrutura de agentes visa prover serviços genéricos, tais como: migração, comunicação, execução, segurança e tolerância a falhas, permitindo a um desenvolvedor de aplicações baseadas em agentes focar-se nos detalhes das aplicações, sem se preocupar com a infra-estrutura. De acordo com Lingnau et al. [3], uma infra-estrutura geral para agentes móveis deve prover mecanismos, não policiamento, para ser aceita de um modo geral. Além disto, funcionalidades específicas de cada sistema podem ser adicionadas de acordo com a necessidade [5].

Um extraordinário aumento nas atividades de pesquisa relacionadas a agentes móveis surgiu no início dos anos 90. Um grande número de fabricantes estavam envolvidos com o desenvolvimento de diversas plataformas, construídas sobre diferentes sistemas operacionais e baseadas em diferentes linguagens e tecnologias [6], tais como: o TACOMA (*Tromsø And COrnell Moving Agents*) [7], o Agent Tcl [8], o Odyssey [9] e o Aglets [10]. Nos últimos anos, esforços de padronização começaram a ser realizados pela FIPA (*Foundation for Intelligent Physical Agents*) [11], pela Agent Society [12] e pela OMG (*Object Management Group*) [13]. A FIPA foi criada para produzir especificações de tecnologias genéricas de agentes e a Agent Society visa facilitar a colaboração e a transferência de tecnologia entre desenvolvedores de sistemas de agentes, assim como incentiva a interoperabilidade entre padrões abertos de agentes. A OMG está trabalhando de modo a estabelecer padrões industriais para a tecnologia de agentes móveis e possibilitar a interoperabilidade entre sistemas de agentes como o Odyssey e o Aglets. Magedanz et al. [5] também trabalham na interoperabilidade de sistemas de agentes móveis (*Mobile Agent System Interoperability Facility* - MASIF). A especificação MASIF visa um ambiente de objetos móveis distribuídos e unificados que permita interações transparentes em relação à tecnologia e à localização entre objetos estáticos e móveis. Embora os escopos dos padrões da FIPA e da OMG sejam bem diferentes, uma cooperação e até mesmo uma integração destes padrões pode vir a ocorrer no futuro [6].

A maioria das infra-estruturas existentes, enquanto diferindo consideravelmente na realização, utilizam a mesma solução básica para portabilidade e segurança: não permitem que os agentes sejam executados na máquina real de processador, memória e sistema operacional. Ao invés disto, os agentes rodam em uma máquina virtual, usualmente um interpretador e um sistema em tempo de execução, os quais escondem os detalhes da arquitetura do sistema hospedeiro e confinam as ações dos agentes ao ambiente restrito [14].

A base do modelo de arquitetura utilizado em uma infra-estrutura de agentes (figura 1.3) é o servidor de agentes. Este servidor está encarregado de prover diversas funções como transporte de agentes, segurança, comunicações de agentes com o hospedeiro, com outros agentes e com seus donos, e armazenamento persistente. Um ambiente apropriado para a execução de um agente, denominado ambiente em tempo de execução, é a interface entre um agente e o seu hospedeiro. Suas principais funções são proteger o hospedeiro de agentes, capturar o estado de um agente e interagir com o servidor. Para cada agente sendo executado em um servidor, existe um ambiente em tempo de execução dedicado, criado pelo servidor. Com isto, um agente não tem acesso direto a outros agentes que estão sendo executados no mesmo hospedeiro. O ambiente é a interface entre o agente e seu hospedeiro e disponibiliza recursos ao agente de modo controlado. O usuário (dono) de um agente interage com a infra-estrutura através de um cliente.

O servidor de agentes é um programa (como um servidor de correio eletrônico ou um servidor de FTP) que é executado em cada computador que será acessível ao agentes e está encarregado dos agentes que estão sendo executados neste computador. Suas tarefas incluem aceitar agentes, criar o ambiente de execução apropriado, supervisionar as execuções dos agentes (no meio tempo respondendo pedidos sobre seus estados) e finalizá-las quando necessário. O servidor de agentes também deve organizar a transferência de agentes para outros hospedeiros, gerenciar comunicações entre agentes e entre agentes e seus donos, autenticar e controlar o acesso para todas as operações de agentes e recuperar agentes e as informações carregadas por eles no caso de falhas em computadores e redes.

As funções do servidor de agentes podem ser divididas em diversos módulos, tais

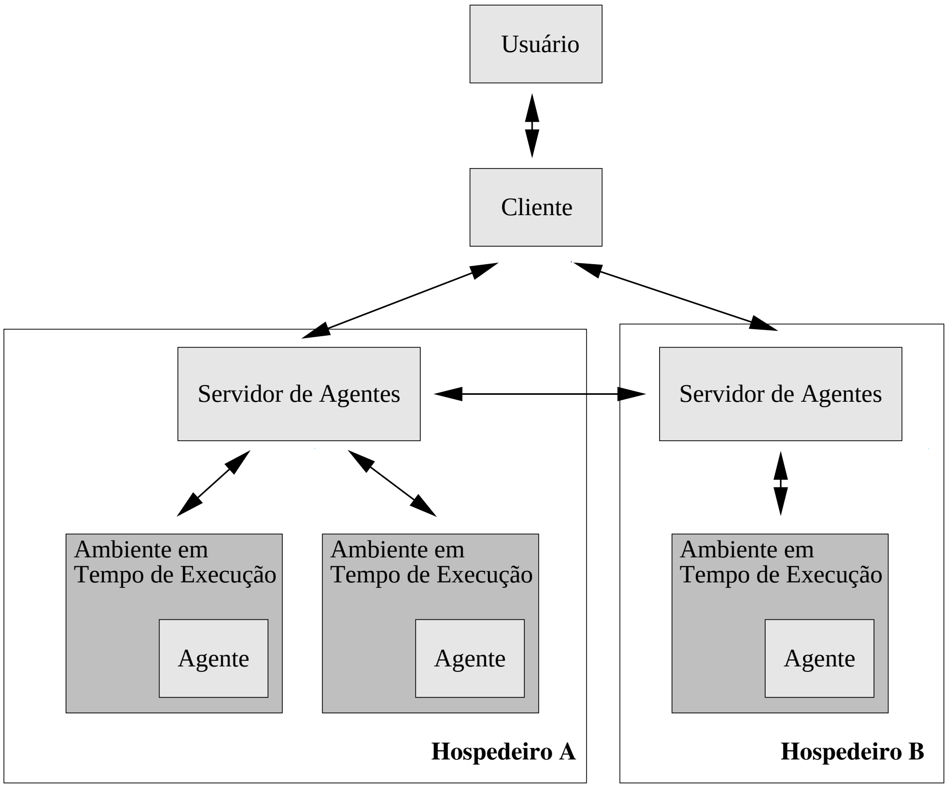

Figura 1.3: Modelo de Arquitetura.

como: Módulo de Segurança, Módulo de Tolerância a Falhas, Módulo de Comunicação, Módulo de Transferência e Módulo de Localização de Serviço.

Talvez o tópico mais importante para a completa aceitação de sistemas baseados em agentes seja a segurança. Em um sistema de redes de computadores, nem o agente nem as máquinas são "confiáveis" [15]. Muitas pessoas associam agentes móveis às ameaças impostas por programas maliciosos como o vírus e os cavalos de Tróia, pois um agente pode atacar uma máquina e ganhar acesso a recursos locais [15]. Deve-se evitar que um servidor adultere os agentes a seu cargo ou mesmo olhe o código de um agente que implementa, por exemplo, uma estratégia de negociação de um competidor.

Mecanismos de recuperação de agentes e das informações que carregam são utilizados no caso de falhas em computadores e redes, de modo que a execução possa voltar ao normal o mais rápido possível [16]. Persistência é utilizada, armazenando de tempos em tempos em memórias não voláteis os agentes, seus dados e seus estados. Este armazenamento baseia-se em pontos de checagem, que normalmente são pontos críticos na execução de um agente. Quando um destes pontos é alcançado, as informações descritas anteriormente são salvas na memória não volátil.

Existem muitas situações onde é mais eficiente estabelecer a comunicação entre um lugar e outro do que migrar o "agente chamador" para o lugar onde o "agente chamado" está [17]. Para realizar trabalhos que tenham alguma utilidade, os agentes devem poder se comunicar com o sistema hospedeiro, com outros agentes e com seus donos.

Uma das características principais de agentes móveis é seu poder de se mover de um hospedeiro para outro. A diferença entre este tipo de mobilidade e a encontrada em, por exemplo, sistemas de balanceamento de carga que permitem a realocação de processos é que o movimento acontece de acordo com um pedido explícito do agente ao invés de ocorrer por decisão do sistema operacional (sem o agente notar) [3]. Isto é, a linguagem de implementação do agente inclui uma sentença ou chamada de biblioteca da forma `MOVA-PARA hospedeiro`. Quando esta chamada é executada, o agente é suspenso, codificado para transmissão, transmitido para o novo hospedeiro, decodificado e instalado para retomar a execução neste novo local, preservando o estado atual e a integridade do agente.

O Módulo de Localização de Serviço ajuda um agente a realizar a sua tarefa, apresentando para ele algum hospedeiro que provê o serviço especificado [18]. Com isto, não existe necessidade do usuário saber o nome ou a URL (*Universal Resource Locator*) do hospedeiro. Em cada hospedeiro, existe uma lista dos hospedeiros localizadores de serviço. Um agente móvel migra para um destes hospedeiros específicos e se comunica com um agente estacionário, o agente descobridor, que provê o serviço de localização de serviço. O agente usa palavras-chave e parâmetros específicos para descrever o serviço especificado. Serviços são divulgados através do registro junto a um agente, incluindo no registro uma lista de todas as palavras-chave e parâmetros específicos que descrevem os serviços.

1.4 Direcionamento do Trabalho

O gerenciamento de redes é uma das aplicações que pode se beneficiar do uso de agentes móveis, por estar baseado de um modo geral em um paradigma centralizado. Os protocolos de gerenciamento SNMP (*Simple Network Management Protocol*) [19] e CMIP (*Common Management Information Protocol*) [20] baseiam-se no modelo cliente-servidor, no qual o gerente centraliza as informações e fornece a ordem para a execução de ações corretivas, e o agente[2] coleta as informações, armazena-as na MIB (*Management Information Base*) e executa as ordens do gerente. Devido à centralização, o gerenciamento não é escalável quando o tamanho ou a complexidade da rede cresce. Além disto, uma falha em um ponto da rede ou no gerente pode fazer com que parte da rede ou a rede toda fique sem gerenciamento. Os agentes móveis descentralizam o processamento e o controle da estação de gerenciamento, reduzindo o tráfego da estação de gerenciamento, tornando assíncrona a comunicação com a estação de gerenciamento, permitindo um balanceamento de carga de processamento e aumentando a flexibilidade do comportamento dos agentes de gerenciamento.

O objetivo deste trabalho é avaliar o comportamento dos agentes móveis em diversas tarefas de gerenciamento. Esta avaliação é feita através de simulações que comparam o desempenho do SNMP com o dos agentes móveis. Várias análises são feitas de modo a avaliar os efeitos da latência e da banda passante de um enlace que liga a estação de gerenciamento a uma rede, da tarefa a ser realizada e do retorno à estação de gerenciamento após visitar um número fixo de nós.

Este trabalho está organizado da seguinte maneira. No capítulo 2 são descritos os principais sistemas de gerenciamento de redes utilizados atualmente, assim como características do gerenciamento de redes com agentes móveis. No capítulo 3 são apresentados os resultados das simulações. Por último, no capítulo 4 são apresentadas as conclusões e as futuras direções que podem ser tomadas na continuação deste trabalho.

[2] Quando não especificado, o termo agente irá se referir neste trabalho aos agentes "inteligentes/móveis". Para marcar a diferença, quando necessário, o agente de um sistema de gerenciamento será doravante chamado agente de gerenciamento.

Capítulo 2

Sistemas de Gerenciamento de Redes

BASICAMENTE , um sistema de gerenciamento de redes contém quatro tipos de componentes: estações de gerenciamento, agentes de gerenciamento sendo executados em nós gerenciados, protocolos de gerenciamento e informações de gerenciamento [21]. Os agentes de gerenciamento são entidades cujo propósito é prover uma interface padronizada para o acesso a informações sobre o equipamento de rede no qual estes agentes residem [22]. Uma estação de gerenciamento utiliza o protocolo de gerenciamento para comunicar-se com os agentes de gerenciamento. As informações trocadas entre a estação de gerenciamento e os agentes de gerenciamento são definidas através de uma base de informações de gerenciamento, a MIB.

Os sistemas de gerenciamento de redes são classificados em centralizados e descentralizados, de acordo com a localização do processamento e do controle do gerenciamento. No gerenciamento centralizado, o processamento das informações obtidas e o controle das ações de gerenciamento estão concentrados na estação de gerenciamento. Atualmente, a maioria dos sistemas de gerenciamento é centralizada, mas alguns passos visando a descentralização já foram tomados e novos sistemas descentralizados também estão sendo desenvolvidos [23].

2.1 Gerenciamento Centralizado

Os protocolos de gerenciamento SNMP [24, 19, 25] e CMIP [20, 26], propostos respectivamente pelo IETF (*Internet Engineering Task Force*) e pela ISO (*International Standards Organization*), baseiam-se no modelo cliente-servidor. Nestes protocolos, a estação de gerenciamento age como um cliente que provê uma interface para o gerente da rede e interage com os agentes de gerenciamento, que são servidores que gerenciam o acesso remoto às suas informações locais (figura 2.1). Estes dados estão armazenados na MIB, uma base hierárquica de informações gerenciadas por cada agente de gerenciamento. As operações disponíveis à estação de gerenciamento para o acesso à MIB são de baixo nível. Esta interação de glanuralidade fina, chamada microgerenciamento, não possui uma boa escalabilidade devido ao tráfego intenso e à sobrecarga computacional na estação de gerenciamento [27].

Alguns passos visando a descentralização já foram tomados pelo IETF e pela ISO [23].

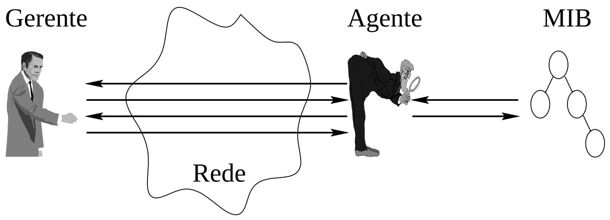

Figura 2.1: Componentes do gerenciamento via SNMP.

2.2 Modificações Visando a Descentralização

Na notificação de eventos, agentes SNMP notificam a estação de gerenciamento quando da ocorrência de alguns eventos significativos. Os agentes de gerenciamento utilizam mensagens chamadas *traps* enviadas sem um pedido explícito da estação de gerenciamento para diminuir o uso intensivo da varredura. A proposta da ISO utiliza agentes de gerenciamento mais complexos com maior capacidade de processamento. Em ambos os casos, o agente de gerenciamento somente é responsável pela notificação do evento.

Um gerenciamento de redes mais descentralizado é adotado no SNMPv2 [19, 28] (SNMP versão 2), no qual podem existir múltiplas estações de gerenciamento no mais alto nível de hierarquia, chamadas servidores de gerenciamento. Cada servidor é responsável por gerenciar agentes, porém responsabilidades podem ser delegadas a um gerente intermediário. Este gerente, também chamado agente procurador, tem funcionalidades de gerente para monitorar e controlar os agentes de gerenciamento sob a sua responsabilidade, e também age como um agente de gerenciamento a fim de prover informações e ser controlado por um servidor de gerenciamento (figura 2.2). A nova versão do SNMP, o SNMPv3, incorpora um novo esquema de segurança [19, 29].

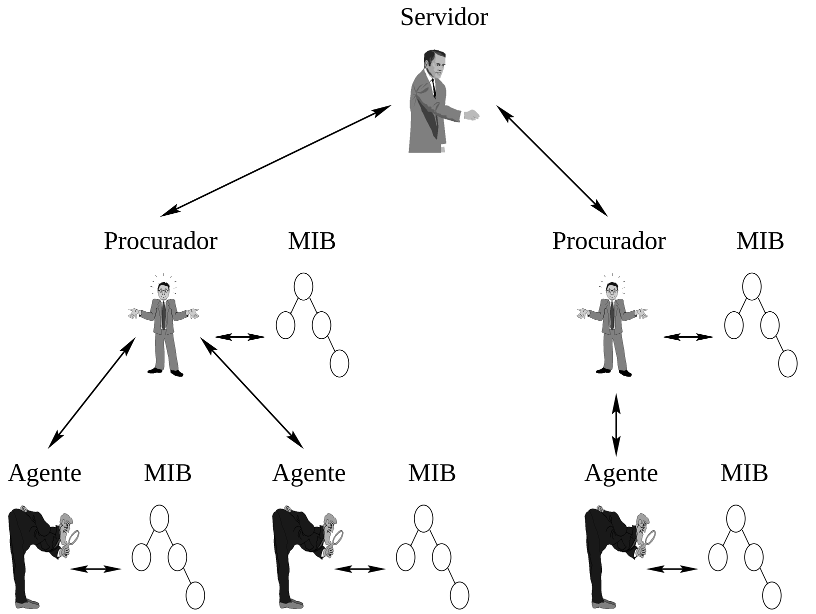

Figura 2.2: Componentes do gerenciamento via SNMPv2.

O IETF também propôs o RMON (*Remote MONitoring*) [30] que pode utilizar equipamentos de monitoração da rede chamados monitores para realizar o monitoramento pró-ativo de redes locais em segmentos locais ou remotos [31]. Estes monitores provêm informações sobre enlaces, conexões entre estações, modelos de tráfego e status dos nós da rede (figura 2.3). Estes monitores também detectam falhas e comportamentos errôneos, geram alarmes quando um limiar predefinido é atingido e identificam eventos complexos, mesmo não estando em contato com a estação de gerenciamento.

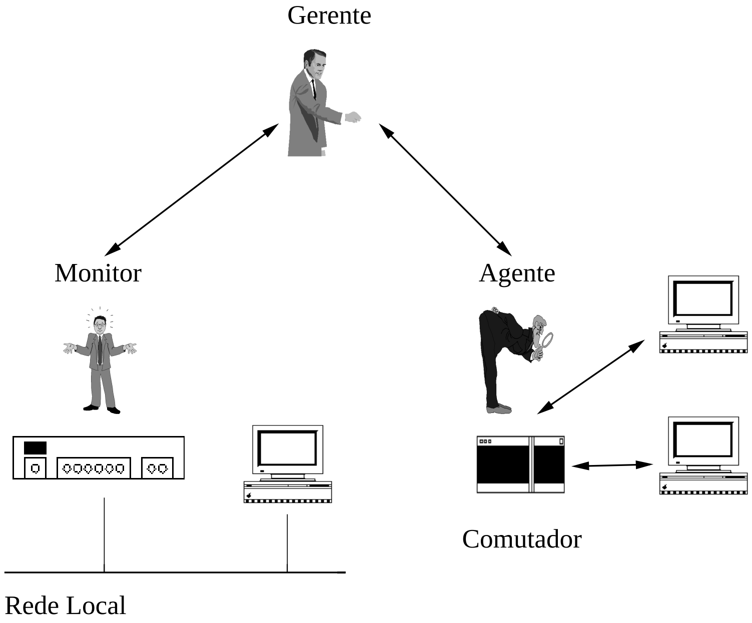

Figura 2.3: Componentes do gerenciamento via RMON.

Estas propostas ainda não provêm o grau de descentralização necessário à gerência de grandes redes [27]. Além disto, o poder de cálculo dos nós da rede também tem aumentado, tornando possível delegar funções de gerenciamento ainda mais complexas aos nós [32]. De modo a satisfazer as diversas necessidades das redes atuais, novos sistemas de gerenciamento de redes que podem analisar dados, tomar decisões e obter medidas pró-ativas para manter a qualidade de serviço da rede devem ser desenvolvidos [33].

Os agentes móveis podem ser utilizados para descentralizar o processamento e o controle da estação de gerenciamento [34].

2.3 Gerenciamento Descentralizado com Agentes Móveis

Várias vantagens [35] podem justificar a utilização de agentes móveis no gerenciamento de redes, tais como:

- redução do custo: caso o gerenciamento exija um grande volume de informações a ser veiculado pela rede, pode ser mais econômico enviar um agente que execute a tarefa diretamente nos agentes de gerenciamento, onde estão localizadas estas informações. Desta forma, o agente pode filtrar e selecionar somente as informações relevantes e transferi-las para a estação de gerenciamento;
- processamento assíncrono: um agente pode ser enviado através da rede, realizando suas tarefas em outros nós. Enquanto o agente se encontra fora de seu nó de origem, pode não ser necessário que este permaneça em operação;
- divisão de processamento: computadores com pequena capacidade de processamento podem ser utilizados de maneira eficiente para realizar tarefas simples de gerenciamento, distribuindo o processamento antes concentrado na estação de gerenciamento;
- flexibilidade: um novo comportamento dos agentes de gerenciamento pode ser introduzido pela estação de gerenciamento através do envio de um agente móvel com um código de execução diferente, substituindo em tempo real o código anterior.

Conforme apresentado anteriormente, um agente móvel pode ser transportado de uma máquina para outra no meio de sua execução e carregar com ele dados acumulados sobre o seu estado que incluem as informações obtidas na execução das tarefas. Conforme o número de nós visitados aumenta, o tamanho do agente também cresce, fazendo com que a migração para outro nó seja realizada com maior dificuldade. Uma possível solução para este problema é percorrer um número fixo de nós e, após isto, retornar ao nó de origem do agente ou enviá-lo os dados (reduzindo o tamanho do agente) e recomeçar a tarefa nos nós restantes (seção 3.2.4). O tamanho inicial de um agente é outra variável que muito influencia o desempenho do agente móvel. Este tamanho depende da tarefa a ser realizada e da linguagem utilizada na implementação, e quanto maior o tamanho, maior a dificuldade de migração.

As pesquisas na área de códigos móveis para o gerenciamento de redes são recentes [36]. Diversos trabalhos têm sido publicados, sendo que a maioria trata da implementação de arquiteturas de gerenciamento utilizando agentes móveis.

2.4 Trabalhos Relacionados

Magedanz et al. [37, 38] foram os primeiros a propor a utilização de agentes móveis no gerenciamento de redes de computadores e de telecomunicação. Estes pesquisadores aumentam a funcionalidade do agente e do gerente através de um ambiente de execução de agentes de modo a habilitar a realização de "aplicações móveis de gerenciamento", implementadas por *scripts* de agentes móveis.

Labetoulle et al. [31] apresentam um estado da arte de agentes inteligentes no gerenciamento de redes, do paradigma gerenciamento por delegação [39] ao modelo de agentes móveis. Karmouch e Pham [36] descrevem o uso atual de agentes móveis em telecomunicações e gerenciamento de redes.

Várias arquiteturas para o gerenciamento de redes utilizando agentes móveis e inteligentes foram propostas na literatura [39, 22, 40, 41, 42, 43, 44].

O gerenciamento por delegação (*Management by Delegation* - MbD) [39, 32, 45] foi o primeiro paradigma a visar a descentralização e a automação de tarefas de gerenciamento através da delegação dinâmica de funções de gerenciamento aos agentes. Kooijman [46] estende o MbD através do uso de agentes responsáveis pelo gerenciamento de uma área, sendo que a execução destes agentes está baseada na ocorrência de eventos específicos. Neuman et al. [47] implementam um protótipo de gerenciamento por delegação utilizando agentes móveis e CORBA (*Common Object Request Broker Architecture*).

Alguns autores [22, 40, 48, 49, 41, 50] desenvolvem arquiteturas que utilizam ao mesmo tempo agentes móveis e SNMP, provendo o acesso a agentes legados. Baldi et al. [22] desenvolvem um protótipo de uma infra-estrutura de gerenciamento com agentes móveis e SNMP. A interação do agentes móveis com os agentes SNMP ocorre porque nem todas as estações possuem a infra-estrutura necessária para a execução dos agentes móveis; porém o melhor seria utilizar os agentes móveis diretamente [22]. Morin et al. [48] definem uma arquitetura de gerenciamento na qual um servidor e vários gerentes se comunicam com agentes SNMP localizados nos elementos gerenciados e analisam o desempenho desta arquitetura através de medidas em uma rede Ethernet comparando os agentes móveis com o SNMP. Pagurek et al. [41, 51] implementam uma infra-estrutura de mobilidade de código e um conjunto de tarefas simples que interagem com agentes localizados em componentes da rede. Eles também propõem um modelo de rede inteligente no qual o comportamento e o estado fazem parte do modelo, podendo ser atualizados de forma dinâmica [52, 53].

Vários pesquisadores [42, 54, 55, 56, 57, 43, 41, 33, 58] aplicam agentes inteligentes nas arquiteturas de gerenciamento. Westphall et al. [54] implementam agentes autônomos integrados ao SNMP, com características de flexibilidade e adaptabilidade para um ambiente de gerenciamento de redes heterogêneas. Protótipos de gerência pró-ativa utilizando sistemas especialistas são implementados por Westphall et al. [59] e Tarouco et al. [55], enquanto Oliveira et al. [56] usam redes neurais para gerenciar de modo pró-ativo redes ATM e Neuman et al. [57] utilizam lógica difusa para realizar um gerenciamento pró-ativo distribuído. A arquitetura de Ku et al. [43] também suporta agentes móveis e o padrão MIB-II [60], enquanto Ji et al. [58] descrevem um agente inteligente que processa as informações coletadas por agentes SNMP e utiliza estas informações para detectar anomalias que tipicamente precedem uma falha na rede.

Alguns autores [43, 44, 61, 62, 41] utilizam Java [63] como a linguagem de implementação das infra-estruturas de agentes móveis.

CORBA também é utilizada no gerenciamento de redes por diversos pesquisadores [64, 65, 47]. Madeira et al. [65] implementam um modelo de monitoração assíncrono, flexível, genérico e configurável, enquanto que Noemi et al. [66] descrevem uma linguagem interpretada de implementação de aplicações de gerência [67] que junto com CORBA flexibiliza o papel do agente de gerenciamento e torna o seu papel mais dinâmico.

Este trabalho visa avaliar o desempenho dos agentes móveis no gerenciamento de redes. Algumas análises semelhantes estão sendo realizadas [27, 40]. Baldi et al. [27, 68] avaliam paradigmas de códigos móveis em aplicações de gerenciamento de redes através de um modelo quantitativo. Sahai e Morin [40, 69] realizam medidas de tempos de resposta e de utilização de banda passante de agentes móveis e aplicações cliente-servidor em uma rede Ethernet. Neste trabalho, como em outros [70, 34, 71, 72], compara-se o desempenho dos agentes móveis versus agentes SNMP através de simulações.

Devido à centralização do processamento e do controle, o gerenciamento de redes no SNMP não é escalável quando o tamanho ou a complexidade da rede cresce. Para solucionar este problema, os agentes móveis são utilizados, sendo importante descobrir quando estes aumentam a eficiência do gerenciamento.

Capítulo 3

Simulações

A ANÁLISE do uso de agentes móveis no gerenciamento de redes é realizada através da comparação do desempenho dos agentes móveis e do SNMP em diversas tarefas de gerenciamento.

3.1 Modelo das Simulações

A topologia de rede utilizada nas simulações (figura 3.1) consiste de elementos dispostos em uma rede local de menor custo (maior banda passante e menor atraso) conectada à Estação de Gerenciamento (EG) através de um enlace de gargalo de maior custo (maior latência e menor banda). A rede local é uma Ethernet com 250 nós, banda passante igual a 10 Mbps e latência de 10 μs. Os valores para o enlace de gargalo variam com o experimento realizado. Os parâmetros de desempenho considerados foram o número de octetos transmitidos e recebidos pela EG e o tempo de resposta na obtenção de variáveis SNMP (objetos da MIB-II [60]).

O simulador de rede *ns* (*Network Simulator*) é utilizado neste trabalho [73]. Este simulador de eventos discretos oferece uma infra-estrutura para simulações em redes de computadores com abstrações para nós e enlaces e possui diversos protocolos já implementados; sendo que nas simulações deste trabalho, foram utilizados as funções de rede local Ethernet e do UDP. Alguns módulos do UDP foram alterados para

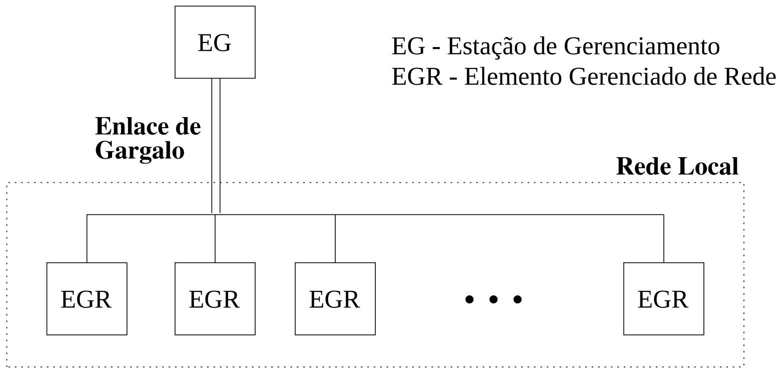

Figura 3.1: Topologia utilizada nas simulações.

que um pacote fosse efetivamente recebido.

O modelo das simulações assume que enlaces e nós não possuem carga, enlaces não têm perdas, e não considera o tempo de processamento na camada aplicação (ex.: tempo de consulta à MIB). O protocolo de transporte UDP foi utilizado em todas as simulações e os tamanhos de pacotes utilizados são valores obtidos através de testes realizados com várias aplicações SNMP disponíveis no Linux. Os cabeçalhos das camadas UDP e IP também foram considerados (figura 3.2). A unidade máxima de transmissão (MTU) usada em todos os enlaces é 1500 octetos; com isto, não há fragmentação de mensagens SNMP por estas serem pequenas. Cada pedido de uma variável é enviado em uma mensagem diferente. Em todas as simulações, o agente móvel segue um itinerário preestabelecido de lugares [18] - em relação à figura 3.1, o agente móvel parte da EG, segue estação-a-estação de uma ponta à outra da rede e retorna à EG. Mais detalhes a respeito da simulação estão descritos no apêndice A.

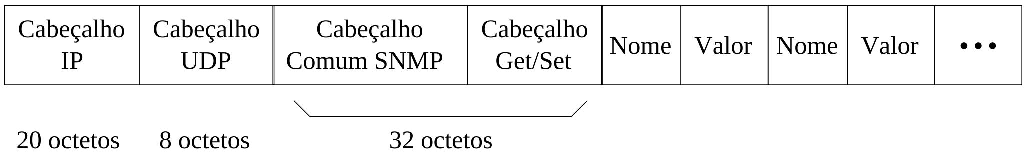

Figura 3.2: Encapsulamento de pedidos e respostas de variáveis SNMP.

3.2 Resultados das Simulações

Vários experimentos foram realizados de modo a avaliar os efeitos da latência e da banda passante do enlace que liga a EG à rede local, da tarefa a ser realizada e do retorno à EG após visitar um número fixo de nós no desempenho do gerenciamento.

3.2.1 Efeito da Latência e da Banda Passante do Enlace

O efeito da latência e da banda passante do enlace que liga a EG à rede local é avaliado através da obtenção da variável SNMP chamada *ifInErrors*, a qual representa o número de pacotes recebidos descartados por causa de erros, da interface de cada estação da rede local.

Três diferentes valores para a latência são utilizados, como na tabela 3.1. A banda passante do enlace é 2 Mbps e a tamanho inicial do agente móvel é 5 k-octetos.

Tabela 3.1: Valores para diferentes latências do enlace de gargalo.

Valores para o enlace	Latência (ms)	Banda passante (Mbps)
v_1	10	2
v_2	4	2
v_3	0.1	2

O número de elementos gerenciados de rede é n. O SNMP envia n pedidos (um após receber o outro) enquanto que o agente móvel é enviado para uma estação da rede local, obtém a variável e percorre todas as outras estações buscando os valores da variável. Ao terminar, o agente móvel retorna à EG.

A figura 3.3 mostra que o comportamento do agente móvel não muda com a latência (todas as três curvas estão muito próximas umas das outras). Porém para o SNMP, a diferença de desempenho entre os tempos de resposta para os três valores de latências aumenta de modo mais rápido quando n é grande, pois os pacotes SNMP passam várias vezes pelo enlace de gargalo. O tempo de resposta para o

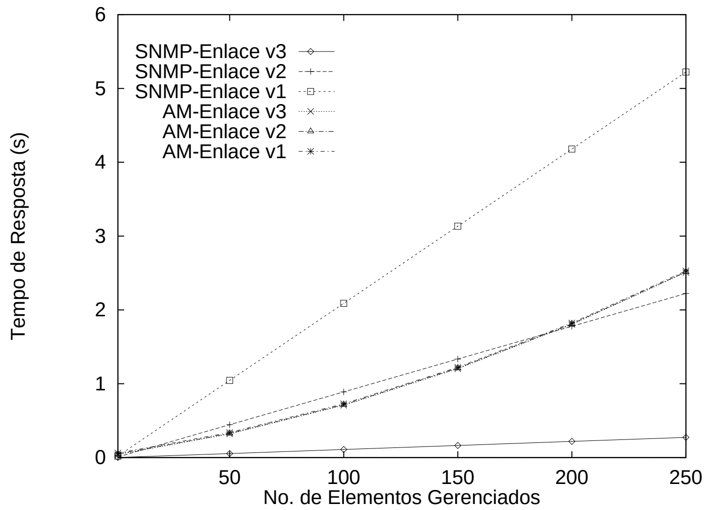

Figura 3.3: Tempo de resposta para diferentes latências do enlace.

SNMP aumenta linearmente com n, pois este tempo é igual para qualquer elemento gerenciado da rede local. Para o agente móvel, o tempo de resposta aumenta mais rapidamente quando n é grande porque o agente móvel possui um tamanho que torna a migração mais difícil. Para o enlace v_3, o SNMP possui um desempenho melhor que o do agente móvel por causa da pequena latência, porém para o enlace v_1, o agente móvel tem um menor tempo de resposta pelo fato da latência ser grande. Se a latência é média (enlace v_2), no início, o SNMP possui um menor tempo de resposta. Conforme o número de elementos gerenciados aumenta, o agente móvel tem um desempenho melhor até que este número de nós visitados alcance um determinado valor (seção 3.2.2).

O efeito da banda passante do enlace de gargalo também é avaliado. Três diferentes valores para a banda passante são utilizados (tabela 3.2). A latência do enlace é 4 ms e o tamanho inicial do agente móvel é 5 k-octetos.

Na figura 3.4, para a menor banda passante (enlace v_1 da tabela 3.2), o agente móvel e o SNMP apresentam tempos de resposta maiores pois a pequena banda faz com que a transferência de agentes móveis e de pacotes SNMP seja dificultada. Para os outros valores de banda passante, o agente móvel e o SNMP se comportam de maneira semelhante à descrição apresentada anteriormente.

Tabela 3.2: Valores para diferentes bandas passantes do enlace de gargalo.

Valores para o enlace	Latência (ms)	Banda passante (Mbps)
v_1	4	0.05
v_2	4	2
v_3	4	5

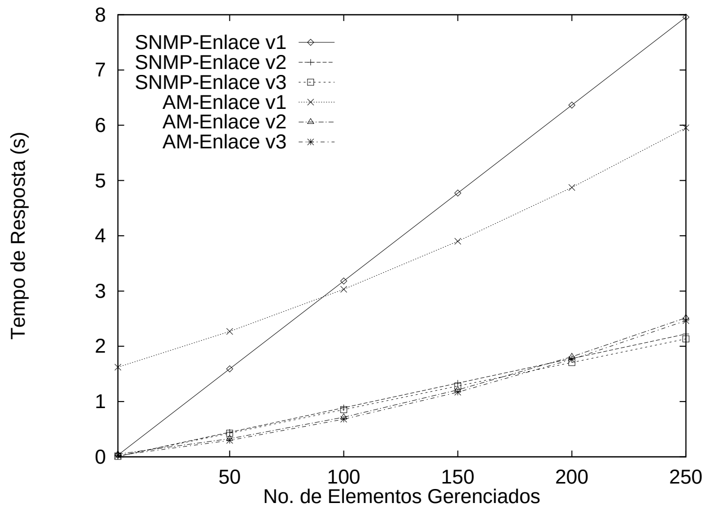

Figura 3.4: Tempo de resposta para diferentes bandas passantes do enlace.

Um experimento mantendo constante o produto tempo de resposta x banda passante também é realizado (tabela 3.3). O tamanho inicial do agente móvel é 5 k-octetos.

Analisando a figura 3.5, pode-se observar que o comportamento do agente móvel é praticamente o mesmo quando a banda passante é média ou grande (enlaces v_2 e v_3 da tabela 3.3), porém uma banda passante pequena (enlace v_1) influencia muito o desempenho do agente móvel. O SNMP é afetado principalmente pela pequena banda passante mas também é influenciado por latências pequena e média (enlaces v_2 e v_3).

Tabela 3.3: Valores para a latência e a banda passante do enlace de gargalo.

Valores para o enlace	Latência (ms)	Banda passante (Mbps)
v_1	10	0.1
v_2	1	1
v_3	0.1	10

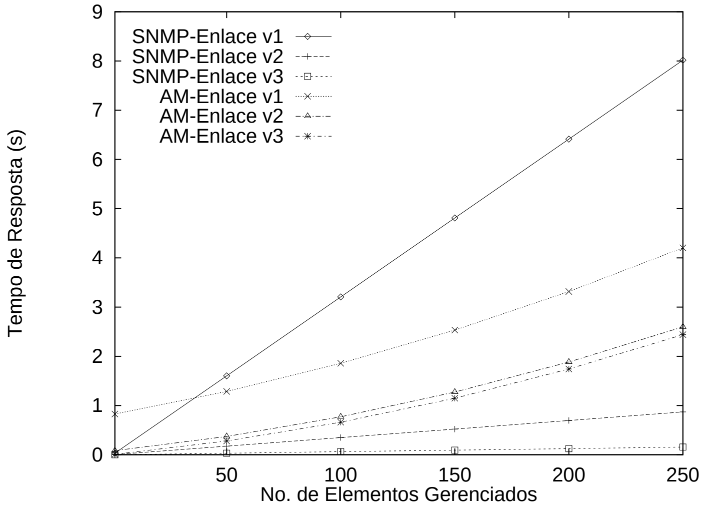

Figura 3.5: Tempos de resposta para diferentes latências e bandas passantes do enlace.

3.2.2 Efeito do Tamanho Inicial do Agente

O efeito do tamanho inicial do agente móvel também é analisado na topologia apresentada anteriormente, com os valores da banda passante e da latência iguais a 4 ms e 2 Mbps, respectivamente. Cinco diferentes tamanhos iniciais de agentes móveis são utilizados (1, 3, 5, 7 e 9 k-octetos). A tarefa é a mesma do experimento anterior (busca da variável *ifInErrors*).

Para um pequeno número de elementos a serem gerenciados, o SNMP utiliza um número menor de octetos do que o agente móvel para realizar a tarefa, porém com o aumento do número de elementos a serem gerenciados, a sobrecarga (cabeçalhos da PDU GetRequest, do UDP e do IP) associada ao número de respostas do SNMP torna viável a utilização do agente móvel (figura 3.6). Com o aumento do tamanho

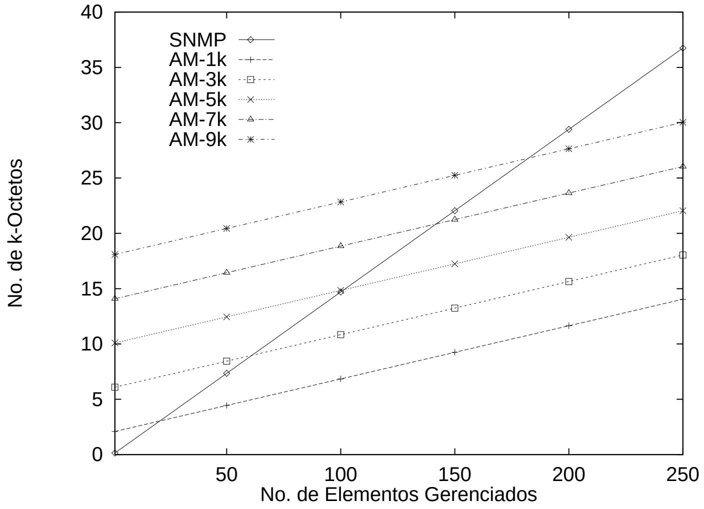

Figura 3.6: Número de octetos para o SNMP e para diferentes tamanhos de agentes móveis.

do agente móvel, o número de octetos que passam pelo enlace de gargalo cresce proporcionalmente.

Para o tempo de resposta, quando o número de elementos a serem gerenciados aumenta, a diferença de desempenho entre os agentes com diferentes tamanhos cresce de modo mais rápido (figura 3.7).

Uma análise do comportamento dos agentes móveis e do SNMP em função do número de elementos gerenciados também pode ser realizada através da figura 3.7. Observando-se as curvas AM-5k e SNMP, no início, o SNMP possui um desempenho melhor pelo fato do tamanho de uma mensagem do SNMP ser menor que o tamanho inicial do agente móvel. Conforme o número de elementos gerenciados aumenta, o tempo de resposta para o SNMP cresce proporcionalmente, fazendo com que o agente móvel obtenha um melhor resultado até um ponto, na figura 3.7 quando o número de elementos a serem gerenciados é 200, a partir do qual o desempenho do SNMP passa a ser melhor devido ao tamanho significativo do agente móvel.

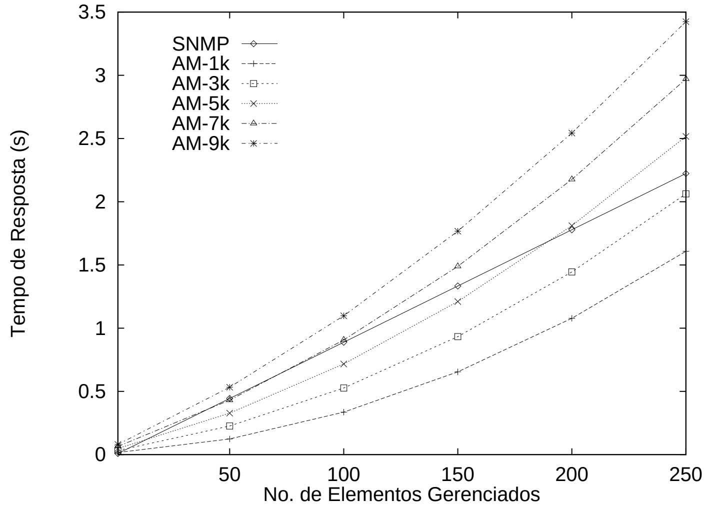

Figura 3.7: Tempo de resposta para o SNMP para diferentes tamanhos de agentes móveis.

3.2.3 Efeito da Tarefa a Ser Realizada

Uma análise quanto ao comportamento do agente móvel e do SNMP em diferentes tarefas também é realizada. Três tarefas são utilizadas: a tarefa t_1 é a anterior, a tarefa t_2 está relacionada ao pedido da variável SNMP chamada *sysContact*, que é o nome da pessoa de contato responsável pela máquina e como contactá-la, e a tarefa t_3 está relacionada à variável SNMP chamada *sysDescr*, que é uma descrição textual do sistema. Os tamanhos dos pedidos e das respostas para as tarefas, incluindo todas as sobrecargas são apresentados na tabela 3.4. Os valores para o enlace de gargalo são os mesmos do experimento anterior e o tamanho inicial do agente móvel é 5 k-octetos.

Tabela 3.4: Tamanho dos pacotes de pedido e resposta para as tarefas.

Tarefa	Pedido (octetos)	Resposta (octetos)
t_1	71	76
t_2	69	91
t_3	66	173

Na figura 3.8, o número de octetos varia com a tarefa a ser realizada pois quanto

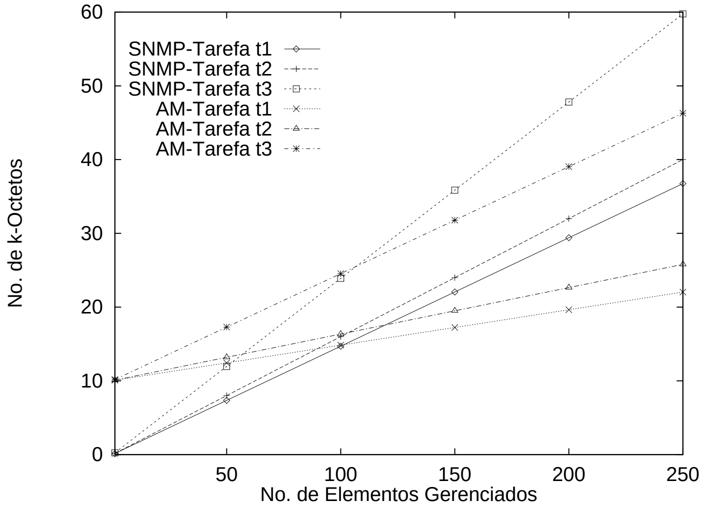

Figura 3.8: Número de octetos para diferentes tarefas.

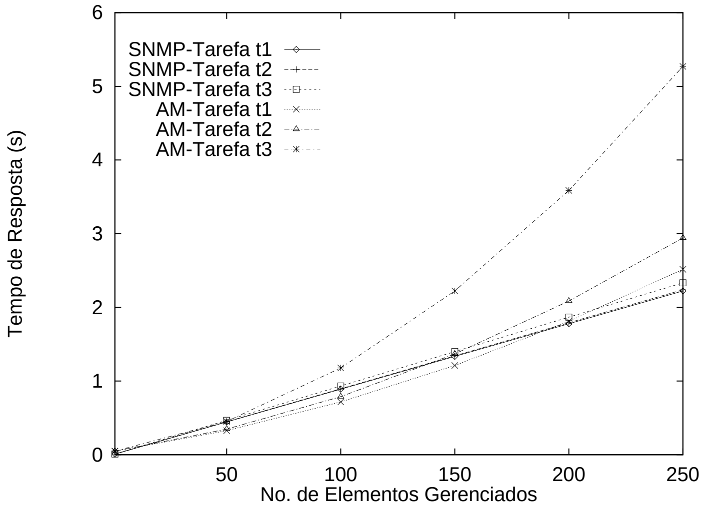

Figura 3.9: Tempo de resposta para diferentes tarefas.

maior o número de octetos de cada pedido/resposta, maior o número total de octetos transmitidos e recebidos pela EG. Todas as curvas relativas ao agente móvel começam com 10 k-octetos pois o agente passa duas vezes pelo enlace de gargalo e seus tamanho inicial é 5 k-octetos.

Na figura 3.9, o tempo de resposta varia de maneira diferente quando utilizando o SNMP ou o agente móvel. Para o SNMP, a tarefa não tem grande influência pois o número de octetos trocados entre a EG e o elemento gerenciado é pequeno, porém para o agente móvel, quando o número de octetos trocados aumenta (por exemplo da tarefa t_1 para a tarefa t_3), os tempos de resposta cresce bastante.

3.2.4 Efeito do Retorno à Estação de Gerenciamento

A política de fazer o agente móvel retornar à EG, de forma a reduzir o seu tamanho, é analisada nesta seção. Esta análise considera que o agente móvel visita um "número fixo de nós por viagem", ou seja, o número de elementos gerenciados antes de retornar à EG para "descarregar". As tarefas consideradas são as três apresentadas anteriormente, porém obtendo as variáveis de todos os 250 elementos gerenciados. Os valores do enlace de gargalo e do tamanho inicial do agente são os mesmos do experimento anterior e o número de estações visitadas por viagem varia de 1 a 250.

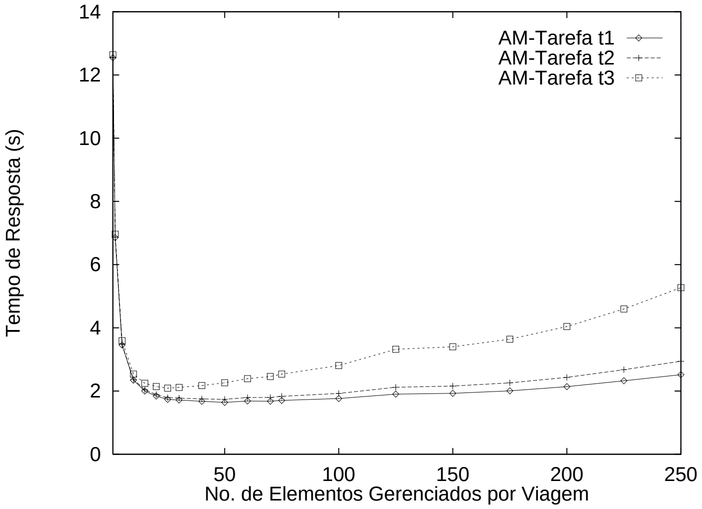

Figura 3.10: Tempo de resposta com o retorno do agente móvel.

Na figura 3.10, para um número pequeno de elementos gerenciados por viagem, o tempo de resposta diminui acentuadamente quando o número de elementos gerenciados aumenta pois em uma viagem o agente percorre poucos nós e retorna à EG. Conforme o número de nós percorridos continua aumentando, o tempo de resposta diminui até um ponto onde o tempo começa a crescer devido à dificuldade de migração do agente por causa do seu tamanho. O "ponto ótimo" varia com a tarefa; para as tarefas t_1 e t_2, percorrer 50 nós e retornar à EG proporcionaria o melhor

Tabela 3.5: Tempo de resposta com o retorno do agente móvel.

No. de Nós	Tarefa t_1 (s)	Tarefa t_2 (s)	Tarefa t_3 (s)
1	12,5447	12,5574	12,6357
2	6,85835	6,8736	6,96198
5	3,45717	3,47772	3,59178
10	2,34267	2,37095	2.53538
25	1,73637	1,79328	2,09114
50	1,64127	1,73873	2,26206
75	1,70217	1,83168	2,53606
100	1,76235	1,92463	2,8091
150	1,92801	2,15618	3,40155
200	2,13909	2,43337	4,03951
250	2,51619	2,942	5,27003

resultado, enquanto que para a tarefa t_3, 25 é o número de nós a serem visitados por viagem (tabela 3.5). De acordo com esta tabela, há uma diminuição do tempo de resposta em relação à obtenção das 250 variáveis em uma só viagem de 35, 41 e 60%, respectivamente, para as tarefas t_1, t_2 e t_3.

Quanto ao número de octetos, este diminui quando o número de nós percorridos de uma vez aumenta, conforme a figura 3.11 e a tabela 3.6, pois quanto maior o número de vezes que o agente móvel retorna à EG, maior o número de octetos transmitidos e recebidos por ela.

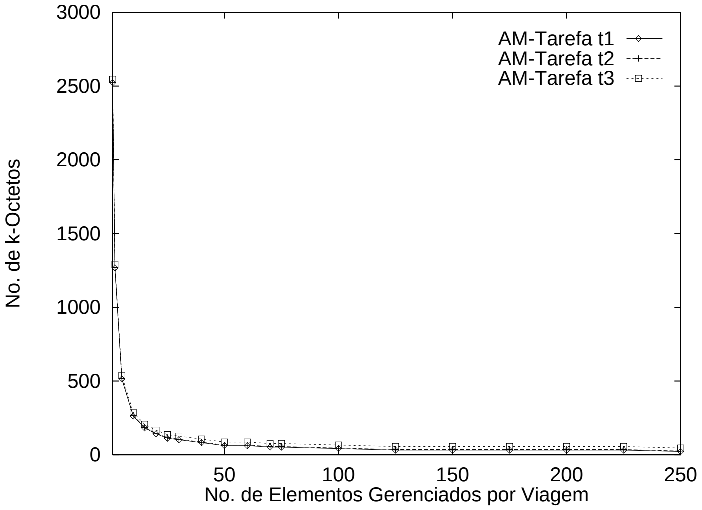

Figura 3.11: Número de octetos com o retorno do agente móvel.

Tabela 3.6: Número de octetos com o retorno do agente móvel.

No. de Nós	Tarefa t_1 (k-octetos)	Tarefa t_2 (k-octetos)	Tarefa t_3 (k-octetos)
1	2522,75	2526	2545,75
2	1267,38	1270,88	1291
5	514,15	517,8	538,15
10	263,075	266,775	287,2
25	112,43	116,16	136,63
50	62,215	65,955	86,44
75	52,172	55,914	76,402
100	42,129	45,873	66,364
150	32,086	35,832	56,326
200	32,086	35,832	56,326
250	22,043	25,791	46,288

Capítulo 4

Conclusões

ESTE trabalho analisou a utilização de agentes móveis em várias tarefas de gerenciamento. O desempenho dos agentes foi comparado com o do SNMP em diversas simulações e foi avaliado através do número de octetos transmitidos e recebidos pela estação de gerenciamento e do tempo de resposta na obtenção de variáveis SNMP.

Vários experimentos foram realizados de modo a avaliar os efeitos da latência e da banda passante do enlace que liga a estação de gerenciamento à rede local, da tarefa a ser realizada e do retorno à estação de gerenciamento após visitar um número fixo de nós no desempenho do gerenciamento.

Em relação ao tempo de resposta, os resultados das simulações indicam que os agentes móveis possuem um melhor desempenho que o SNMP quando o número de elementos gerenciados encontra-se entre dois limites: o inferior associado ao número de mensagens que passam pelo enlace de gargalo que conecta a estação de gerenciamento à rede local remota e o superior relacionado ao tamanho crescente do agente móvel que dificulta a migração.

Observa-se que se o agente voltar para a estação de gerenciamento após coletar um número fixo de variáveis, o desempenho melhora pois o tamanho do agente é reduzido de tempos em tempos. É obtida uma diminuição de até 60% no tempo de resposta quando da utilização desta política. Resultados ainda melhores podem ser obtidos se o agente enviar os dados à estação de gerenciamento ao invés de retornar à mesma.

Os resultados indicam ainda que os agentes móveis são menos sensíveis à latência e à banda passante do enlace de gargalo, pois os pacotes do SNMP passam várias vezes por este enlace.

O agentes móveis sofrem maior influência das tarefas a serem realizadas, pelo fato que, para o SNMP, o número de octetos trocados entre a estação de gerenciamento e a elemento gerenciado é pequeno.

Em relação ao número de octetos, o agente móvel possui um melhor desempenho que o SNMP quando o número de elementos gerenciados ultrapassa um valor relacionado à sobrecarga das várias respostas do SNMP.

Conclui-se que o paradigma de agentes móveis aumenta o desempenho do gerenciamento remoto de redes quando a maior parte do movimento dos agentes ocorre dentro de redes de menor custo (maior banda passante e menor latência); em outras palavras, quando somente uma pequena porcentagem da migração envolve enlaces de gargalo.

Diversas direções, complementares ou não, podem ser tomadas a partir dos resultados já obtidos:

- Colocação de carga e perda na rede e de processamento nos nós: para a avaliação do comportamento dos agentes móveis quando a rede estiver saturada. No modelo centralizado, isto causa uma piora do estado da rede pois as informações de gerenciamento trafegam pela mesma rede que já estava congestionada. O maior problema nesta análise através de simulações está na recuperação quando do não recebimento de uma variável no SNMP. Retransmissões são difíceis de ser realizadas por causa do número de parâmetros que envolvem o estabelecimento do tempo de retransmissão. Uma solução seria a utilização de um protocolo confiável abaixo do SNMP (ex.: TCP), embora o uso do SNMP com o TCP não seja comum [19];

- Novos comportamentos dos agentes móveis quando do retorno à Estação de Gerenciamento (EG): ao invés de retornar à EG, o agente móvel poderia enviá-la seus dados, prosseguindo do ponto onde está. Com isto, um menor tempo de resposta seria obtido e também o número de octetos transmitidos e recebidos pela EG seria menor;

- Novas topologias com gargalos: testes com topologias similares no formato às topologias da Internet podem ser feitos de modo a analisar o comportamento dos agentes móveis em uma topologia mais geral. Na figura 4.1 é apresentado um exemplo de topologia similar às da Internet. Os resultados obtidos em [34] podem ser estendidos, já que as análises englobaram somente o tamanho inicial dos agentes móveis. O gerador de topologias GT-ITM [74] pode ser utilizado para gerar topologias aleatórias, como em [34].

- Colocação de inteligência nos agentes: os agentes móveis podem realizar um gerenciamento pró-ativo, verificando quando determinados limiares são ultrapassados. Com isto, um problema que pode vir a acontecer pode ser diagnosticado ou na pior das hipóteses, uma anomalia pode ser cadastrada quando esta não puder ser relacionada a nenhum evento conhecido. Além disto, funções de gerenciamento podem ser enviadas a priori e utilizadas quando a rede estiver em situação de congestionamento. Os agentes inteligentes podem ainda avaliar e reagir a efeitos transientes.

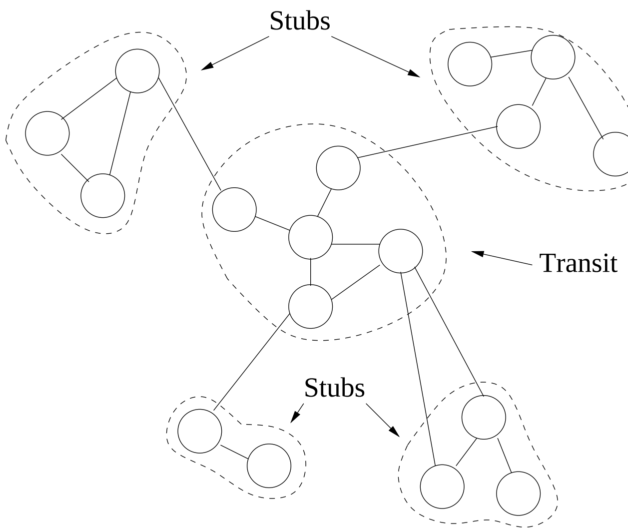

Figura 4.1: Um exemplo de topologia transit-stub.

Publicações

[1] Rubinstein, M. G., and Duarte, O. C. M. B. Service location for mobile agent systems. In *IEEE/SBT International Telecommunications Symposium (ITS'98)* (São Paulo, Brasil, agosto de 1998), pp. 623–626.

[2] Rubinstein, M. G., and Duarte, O. C. M. B. Análise da eficiência de agentes móveis no gerenciamento de redes. In *XIX Congresso Nacional da Sociedade Brasileira de Computação, XXVI Seminário Integrado de Software e Hardware - SEMISH'99* (Rio de Janeiro, Brasil, julho de 1999), vol. 1, pp. 167–178.

[3] Rubinstein, M. G., and Duarte, O. C. M. B. Evaluating the performance of mobile agents in network management. In *Proceedings of the IEEE Global Telecommunications Conference GLOBECOM'99 (a ser publicado)* (Rio de Janeiro, Brasil, dezembro de 1999).

[4] Rubinstein, M. G., and Duarte, O. C. M. B. Evaluating tradeoffs of mobile agents in network management. *Networking and Information Systems Journal 2*, 2 (julho de 1999).

Referências Bibliográficas

[1] Cockayne, W. R., and Zyda, M. *Mobile Agents.* Manning Publications Co., 1998.

[2] Gilbert, D., and Janca, P. IBM intelligent agents. IBM Corporation, EUA, http://www.networking.ibm.com/iag/iagwp1.html.

[3] Lingnau, A., and Drobnik, O. An infrastructure for mobile agents: requirements and architecture. In *Proceedings of the 13th DIS Workshop* (Orlando, Flórida, EUA, setembro de 1995).

[4] Kiniry, J., and Zimmerman, D. A hands-on look at Java mobile agents. *IEEE Internet Computing 1*, 4 (julho de 1997), 21–30.

[5] Breugst, M., and Magedanz, T. On the usage of standard mobile agent plataforms in telecommunication environments. In *Proceedings of the 5th International Conference on Intelligence in Services and Networks (IS&N'98)* (Antuérpia, Bélgica, maio de 1998), pp. 275–286.

[6] Breugst, M., and Magedanz, T. Mobile agents - enabling technology for active intelligent network implementation. *IEEE Network 12*, 3 (maio de 1998).

[7] Johansen, D., van Renesse, R., and Schneider, F. B. An introduction to the TACOMA distributed system. Relatório técnico, University of Tromsϕ/Cornell University, Tromsϕ/Ithaca, Noruega/EUA, junho de 1995.

[8] Gray, R. S. Agent Tcl: a transportable agent system. In *CIKM Workshop on Intelligent Information Agents, Fouth International Conference on Information and Knowledge Management (CIKM'95)* (Baltimore/Maryland, EUA, 1995).

[9] Odyssey. General Magic, http://www.genmagic.com/agents, 1997.

[10] Programming mobile agents in Java. IBM Corporation, IBM do Japão, http://www.trl.ibm.co.jp/aglets, 1996.

[11] Chiariglione, L. Helping agent technologies get across to the market place. Foundation for Intelligent Physical Agents (FIPA), http://drogo.cselt.stet.it/fipa/general_agents.htm.

[12] The Agent Society. http://www.agent.org, 1997.

[13] The Mobile Agent Facility Specification. http://www.genmagic.com/agents/MAF.

[14] Peine, H., and Stolpmann, T. The architecture of the Ara platform for mobile agents. In *1st International Workshop on Mobile Agents* (Berlim, Alemanha, abril de 1997), pp. 50–61.

[15] Abdalla, M., Cirne, W., Franklin, L., and Tabbara, A. Security issues in agent based computing. In *XV Simpósio Brasileiro de Redes de Computadores (SBRC'97)* (São Carlos, SP, Brasil, maio de 1997), pp. 231–244.

[16] de Oliveira, P. C., and Cardozo, E. Mobile agent-based systems: an alternative paradigm for distributed systems development. In *XV Simpósio Brasileiro de Redes de Computadores (SBRC 97)* (São Carlos, SP, Brasil, maio de 1997), pp. 508–524.

[17] Baumann, J., Hohl, F., Radouniklis, N., Rothermel, K., and Straβer, M. Communication concepts for mobile agent systems. In *1st International Workshop on Mobile Agents* (Berlim, Alemanha, abril de 1997), pp. 123–135.

[18] Rubinstein, M. G., and Duarte, O. C. M. B. Service location for mobile agent systems. In *IEEE/SBT International Telecommunications Symposium (ITS'98)* (São Paulo, Brasil, agosto de 1998), pp. 623–626.

[19] Stallings, W. SNMP and SNMPv2: The infrastructure for network management. *IEEE Communications Magazine 36*, 3 (março de 1998), 37–43.

[20] Yemini, Y. The OSI network management model. *IEEE Communications Magazine 31*, 5 (maio de 1993), 20–29.

[21] Meyer, K., Erlinger, M., Betser, J., Sunshine, C., Goldszmidt, G., and Yemini, Y. Decentralizing control and intelligence in network management. In *Proceedings of the 4th International Symposium on Integrated Network Management* (Santa Bárbara, EUA, maio de 1995).

[22] Baldi, M., Gai, S., and Picco, G. P. Exploiting code mobility in decentralized and flexible network management. In *First International Workshop on Mobile Agents* (Berlim, Alemanha, abril de 1997), pp. 13–26.

[23] Kahani, M., and Beadle, H. W. P. Decentralised approaches for network management. *Computer Communications Review 27*, 3 (julho de 1997), 36–47.

[24] Case, J., Fedor, M., Schoffstall, M., and Davin, J. A Simple Network Management Protocol (SNMP). RFC 1157, maio de 1990.

[25] Stallings, W. *SNMP, SNMPv2, and RMON: Pratical Network Management.* Addison-Wesley, 1996.

[26] Sluman, C. A tutorial on OSI management. *Computer Networks 17*, 4&5 (outubro de 1989), 270–278.

[27] Baldi, M., and Picco, G. P. Evaluating the tradeoffs of mobile code design paradigms in network management applications. In *Proceedings of the 20th International Conference on Software Engineering (ICSE'98)* (Kyoto, Japão, abril de 1998), pp. 146–155.

[28] Case, J., McCloghrie, K., Rose, M., and Waldbusser, S. Introduction to version 2 of the Internet-standard network management framework. RFC 1441, abril de 1993.

[29] Stallings, W. SNMPv3: A security enhancement for SNMP. *IEEE Communications Surveys 1*, 1 (setembro de 1998).

[30] Waldbusser, S. Remote network monitoring management information base. RFC 1757, fevereiro de 1995.

[31] Cheikhrouhou, M., Conti, P., and Labetoulle, J. Intelligent agents in network management: a state-of-the-art. *Networking and Information Systems Journal 1*, 1 (1998), 9–38.

[32] Goldszmidt, G., and Yemini, Y. Distributed management by delegation. In *The 15th International Conference on Distributed Computing Systems* (Vancouver, Canadá, junho de 1995).

[33] Gürer, D., Lakshminarayan, V., and Sastry, A. An intelligent-agent-based architecture for the management of heterogeneous networks. In *Proceedings of the 9th IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM'98)* (Delaware, EUA, outubro de 1998).

[34] Rubinstein, M. G., and Duarte, O. C. M. B. Evaluating the performance of mobile agents in network management. In *Proceedings of the IEEE Global Telecommunications Conference GLOBECOM'99 (a ser publicado)* (Rio de Janeiro, Brasil, dezembro de 1999).

[35] Nwana, H. S. Software agents: an overview. *Knowledge Engineering Review 11*, 3 (setembro de 1996), 1–40.

[36] Pham, V. A., and Karmouch, A. Mobile software agents: An overview. *IEEE Communications Magazine 36*, 7 (julho de 1998), 26–37.

[37] Magedanz, T., and Eckardt, T. Mobile software agents: A new paradigm for telecommunications management. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS'96)* (Kyoto, Japão, abril de 1996).

[38] Magedanz, T., Rothermel, K., and Krause, S. Intelligent agents: An emerging technology for next generation telecommunications? In *Proceedings of the IEEE INFOCOM'96* (São Francisco, EUA, março de 1996).

[39] Goldszmidt, G., and Yemini, Y. Delegated agents for network management. *IEEE Communications Magazine 36*, 3 (março de 1998), 66–70.

[40] SAHAI, A., AND MORIN, C. Enabling a Mobile Network Manager (MNM) through mobile agents. In *Second International Workshop on Mobile Agents* (Stuttgart, Alemanha, setembro de 1998).

[41] BIESZCZAD, A., PAGUREK, B., AND WHITE, T. Mobile agents for network management. *IEEE Communications Surveys 1*, 1 (setembro de 1998).

[42] CHEIKHROUHOU, M., CONTI, P., LABETOULLE, J., AND MARCUS, K. Intelligent agents for network management: Fault detection experiment. In *Sixth IFIP/IEEE International Symposium on Integrated Network Management (IM'99)* (Boston, EUA, maio de 1999).

[43] KU, H., LUDERER, G. W., AND SUBBIAH, B. An intelligent mobile agent framework for distributed network management. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'97)* (Phoenix, EUA, novembro de 1997).

[44] PULIAFITO, A., TOMARCHIO, O., AND VITA, L. MAP: Design and implementation of a Mobile Agent Plataform. *Journal of System Architecture (a ser publicado)*.

[45] SCHÖNWÄLDER, J. Network management by delegation - from research prototypes towards standards. *Computer Networks 29*, 15 (novembro de 1997), 1843–1852.

[46] KOOIJMAN, R. Divide and conquer in network management using event-driven network area agents. Relatório técnico, Technical University of Delft, Holanda, maio de 1995.

[47] DE CASTRO E SILVA, J. L., DA SILVA, A. N., AND DE SOUZA, J. N. Modelos de implementação de gerência por delegação em sistemas de redes heterogêneas, baseados nos conceitos de agentes móveis e objetos remotos cooperantes. In *XVII Simpósio Brasileiro de Redes de Computadores (SBRC'99)* (Salvador, Brasil, maio de 1999), pp. 579–580.

[48] SAHAI, A., AND MORIN, C. Towards distributed and dynamic network management. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS'98)* (New Orleans, EUA, fevereiro de 1998).

[49] SAHAI, A., MORIN, C., AND BILLIART, S. Intelligent agents for a Mobile Network Manager (MNM). In *Proceedings of the IFIP/IEEE International Conference on Intelligent Networks and Intelligence in Networks (2IN'97)* (Paris, França, setembro de 1997).

[50] PAGUREK, B., WANG, Y., AND WHITE, T. Integration of mobile agent with SNMP: Why and how. Relatório técnico, Carleton University, Canadá, 1999.

[51] SCHRAMM, C., BIESZCZAD, A., AND PAGUREK, B. Application-oriented network modeling with mobile agents. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS'98)* (New Orleans, EUA, fevereiro de 1998).

[52] WHITE, T., PAGUREK, B., AND BIESZCZAD, A. Network modeling for management applications using intelligent mobile agents. *Journal of Network and Systems Management (a ser publicado)* (setembro de 1999).

[53] WHITE, T., PAGUREK, B., BIESZCZAD, A., SUGAR, G., AND TRAN, X. Intelligent network modeling using mobile agents. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'98)* (Sidnei, Australia, novembro de 1998), pp. 1082–1087.

[54] KOCH, F. L., AND WESTPHALL, C. B. Gerenciamento de redes de computadores utilizando inteligência artificial distribuída - uma abordagem operacional. In *XVI Simpósio Brasileiro de Redes de Computadores (SBRC'98)* (Rio de Janeiro, Brasil, maio de 1998), pp. 144–158.

[55] ARTOLA, E. S., AND TAROUCO, L. M. R. Um sistema especialista para gerência pró-ativa remota. In *XIV Simpósio Brasileiro de Redes de Computadores (SBRC'96)* (Fortaleza, Brasil, maio de 1996), pp. 118–139.

[56] Nascimento, A., Franklin, M., and Oliveira, M. Desenvolvendo agentes inteligentes para a gerência pró-ativa de redes ATM. In *XVII Simpósio Brasileiro de Redes de Computadores (SBRC'99)* (Salvador, Brasil, maio de 1999), pp. 681–696.

[57] Carvalho, E., Belchior, A. D., and de Souza, J. N. Gerenciamento pró-ativo distribuído baseado em lógica difusa. In *XVII Simpósio Brasileiro de Redes de Computadores (SBRC'99)* (Salvador, Brasil, maio de 1999), pp. 649–664.

[58] Hood, C. S., and Ji, C. Intelligent agentes for proactive fault sectection. *IEEE Internet Computing 2*, 2 (março de 1998), 65–72.

[59] da Rocha, M. A., Fernandez, L. F. N., and Westphall, C. B. Gerência pró-ativa de redes de computadores usando agentes e técnicas de inteligência artificial. In *XIV Simpósio Brasileiro de Redes de Computadores (SBRC'96)* (Fortaleza, Brasil, maio de 1996), pp. 97–117.

[60] McCloghrie, K., and Rose, M. Management information base for network management of TCP/IP-based internets: MIB-II. RFC 1213, março de 1991.

[61] Puliafito, A., Tomarchio, O., and Vita, L. A Java-based distributed network management architecture. In *Proceedings of the 3rd International Conference on Computer Science and Informatics (CSI'97)* (Durham, EUA, março de 1997).

[62] Puliafito, A., and Tomarchio, O. Advanced network management functionalities through the use of mobile software agents. In *Proceedings of the 3rd International Workshop on Intelligent Agents for Telecommunication Applications (IATA'99)* (Estocolmo, Suécia, agosto de 1999).

[63] Arnold, K., and Gosling, J. *The Java Programming Language.* Addison-Wesley, 1996.

[64] Haggerty, P., and Seetharaman, K. The benefits of CORBA-based network management. *Communications of the ACM 41*, 10 (outubro de 1998), 73–79.

[65] de Queiroz, J. A. G., and Madeira, E. R. M. Facilidade de monitorização assíncrona em um ambiente de gerência CORBA. In *II Seminário Franco-Brasileiro em Sistemas Informáticos Distribuídos* (Fortaleza, Brasil, novembro de 1997), pp. 135–146.

[66] Rodriguez, N., Catunda, M., and Ierusalimschy, R. Extensão dinâmica de agentes CORBA. In *XVI Simpósio Brasileiro de Redes de Computadores (SBRC'98)* (Rio de Janeiro, Brasil, maio de 1998), p. 768.

[67] de Moura, A. L., Ishikawa, E., Lima, M. E., and Rodriguez, N. Aplicações de gerência extensíveis. In *XVI Simpósio Brasileiro de Redes de Computadores (SBRC'98)* (Rio de Janeiro, Brasil, maio de 1998), pp. 125–143.

[68] Fuggetta, A., Picco, G. P., and Vigna, G. Understanding code mobility. *IEEE Transactions on Software Engineering 24*, 5 (maio de 1998), 342–361.

[69] Sahai, A., and Morin, C. Mobile agents enhanced thin client approach to network management. In *Proceedings of the IEEE Singapore International Conference on Networks (SICON'98)* (Kent Ridge, Singapura, junho de 1998).

[70] Rubinstein, M. G., and Duarte, O. C. M. B. Análise da eficiência de agentes móveis no gerenciamento de redes. In *XIX Congresso Nacional da Sociedade Brasileira de Computação, XXVI Seminário Integrado de Software e Hardware (SEMISH'99)* (Rio de Janeiro, Brasil, julho de 1999), vol. 1, pp. 167–178.

[71] Rubinstein, M. G., and Duarte, O. C. M. B. Analyzing mobile agent scalability in network management. Relatório técnico, Grupo de Teleinformática e Automação, COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil, maio de 1999.

[72] Rubinstein, M. G., and Duarte, O. C. M. B. Evaluating tradeoffs of mobile agents in network management. *Networking and Information Systems Journal 2*, 2 (julho de 1999).

[73] Fall, K., and Varadhan, K. NS Notes and Documentation. Relatório técnico, The VINT Project, janeiro de 1999.

[74] Zegura, E. W., Calvert, K. L., and Donahoo, M. J. A quantitative comparison of graph-based models for internet topology. *IEEE/ACM Transactions on Networking 5*, 6 (dezembro de 1997), 770–783.

Apêndice A

Simulador *ns*

O SIMULADOR *ns* [73] é um simulador de protocolos de rede em desenvolvimento no projeto *Virtual InterNet Testbed* (VINT), uma colaboração entre UC Berkeley, LBL, USC/ISI e Xerox PARC. O *ns* é um simulador orientado a objetos escrito em linguagem C++. Os usuários geram *scripts* na linguagem OTcl que são interpretados e executados pelo simulador. O simulador suporta uma hierarquia de classes em C++ e uma hierarquia de classes similar para o interpretador OTcl. Os usuários criam novos objetos do simulador através do interpretador que instancia estes objetos e gera um espelho em C++, criando um objeto correspondente.

O simulador, descrito por uma classe em Tcl *class Simulator*, oferece um conjunto de interfaces para a configuração de uma simulação e para a escolha do tipo de escalonador de eventos utilizado para conduzir a simulação. Um *script* de simulação, normalmente, inicia criando uma instância desta classe e chamando métodos para a criação de nós, topologias e para a configuração de outros aspectos da simulação.

As topologias são criadas no *ns* através de nós e da conexão destes nós por enlaces. O *ns* permite a utilização de dois tipos de nós, ponto-a-ponto e multiponto. A função de um nó, quando recebe um pacote, é examinar seus campos, normalmente o endereço destino, mapeando os valores em um objeto de interface de saída que é o próximo receptor deste pacote.

Os enlaces no *ns* são compostos por objetos chamados conectores. Os conectores apenas geram dados para um receptor. O pacote pode ser enviado para este receptor ou pode ser descartado. Um enlace é criado com características específicas de banda passante e atraso. Os enlaces possuem conectores do tipo *queue* que modelam as filas de saída associadas aos enlaces. O *ns* suporta diversos tipos de escalonadores de pacote, para as simulações foi usado o *drop-tail queueing* (FIFO). Nas simulações realizadas, foi utilizado o método *make-lan* para criar a topologia da rede local, enquanto que o enlace de gargalo foi criado através do procedimento *duplex-link*.

Os agentes do *ns* representam pontos onde os pacotes são gerados ou consumidos e são usados para a implementação de protocolos de várias camadas. Existem vários agentes já implementados no *ns*, sendo que neste trabalho, o *UDP* foi utilizado. Um agente do *ns* chamado *UDPSink* foi criado de modo que o UDP recebesse efetivamente um pacote.

As aplicações são executadas em cima dos agentes de transporte. Existem dois tipos de aplicações: fontes e geradores de tráfego. Neste trabalho os geradores de tráfego são utilizados por atuarem em conjunto com agentes UDP. A aplicação utilizada nas simulações do SNMP envia o pedido de uma variável para cada estação da rede a cada segundo, não sendo necessário o recebimento da resposta na camada aplicação já que o tempo de processamento não é computado no *ns*. A aplicação somente é responsável pelo envio do agente móvel ao primeiro nó.

A.1 Detalhes das Simulações

As simulações para as análises dos efeitos da latência e da banda passante, do tamanho inicial do agente e da tarefa a ser realizada estão baseadas em um único modelo. Para o SNMP, em relação aos agentes do *ns*, várias instâncias de um gerente são criadas, uma para cada agente de gerenciamento, pois há a necessidade da conexão de cada um destes agentes com o seu respectivo gerente. Esta conexão visa somente estabelecer os endereços fonte e destino para o envio e o recebimento dos pacotes, não significando que o protocolo de transporte seja orientado à conexão. Quanto à aplicação, várias instâncias foram criadas, cada uma responsável pelo criação de um pacote de pedido de variável. Cada instância foi ligada a um gerente, sendo executada a cada segundo. No final, somente os tempos decorridos entre o envio de um pacote e o recebimento do mesmo são computados.

Para o agente móvel, as conexões relativas ao gerente e aos agentes do *ns* foram feitas através do procedimento *simplex-connect* para o envio de pacotes em um só sentido. Seguindo o procedimento apresentado na seção 3.1, o gerente é conectado ao primeiro agente do *ns*, este agente é conectado ao próximo e assim sucessivamente, até o último agente do *ns* que é conectado ao gerente, de modo que o agente móvel retorne à estação de gerenciamento.

Para a análise do retorno à estação de gerenciamento, a simulação é mais complexa pois o número de agentes do *ns* e as conexões entre estes agentes e o gerente dependem do número de nós a serem visitados em uma viagem pelo agente móvel (número de passos). Além disto, após cada retorno à estação de gerenciamento, uma aplicação do *ns* era iniciada, sendo o instante de início calculado de acordo com o número de nós da rede local e com o número de passos.