

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO**

Paulo João Martins

**COMPARAÇÃO DOS PARADIGMAS
CLIENTE/SERVIDOR E AGENTES MÓVEIS:
UM ESTUDO EM GERÊNCIA DE REDES**

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação

Prof^a. Elizabeth Sueli Specialski, Dra.

Florianópolis, março / 2002

COMPARAÇÃO DOS PARADIGMAS CLIENTE/SERVIDOR E AGENTES MÓVEIS: UM ESTUDO EM GERÊNCIA DE REDES

Paulo João Martins

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação, Área de Concentração: Sistemas de Computação, e aprovada em sua forma final pelo Programa de Pós-graduação em Ciência da Computação.

Prof. Dr. Fernando A. O. Gauthier (Coordenador)

Banca Examinadora:

Prof.^a Dra. Elizabeth Sueli Specialski (Orientadora)

Prof. Dr. Paulo José de Freitas Filho

Prof. Dr. Luiz Fernando Jacintho Maia

“Não basta ensinar ao homem uma especialidade, porque se tornará assim uma máquina utilizável, mas não uma personalidade. É necessário que adquira um sentimento, um senso prático daquilo que vale a pena ser empreendido, daquilo que é belo, do que é moralmente correto. A não ser assim, ele se assemelhará, com seus conhecimentos profissionais, mais a um cão ensinado do que a uma criatura harmoniosamente desenvolvida. Deve aprender a compreender as motivações dos homens, suas quimeras e suas angústias, para determinar com exatidão seu lugar preciso em relação a seus próximos e à comunidade.”

Albert Einstein

À minha avó, Zenaide Anunciação Martins, *in memoriam*,

à minha mãe, Celina Martins, *in memoriam*, e

ao meu irmão, Miguel Cesar Martins.

AGRADECIMENTOS

Agradecimentos especiais à Profa. Elizabeth Sueli Specialski, pela sua orientação e apoio valiosos.

Aos professores Luis Felipe Dias Lopes e a Maria Emilia Camargo, pela confiança depositada na forma de cartas de recomendação.

Ao Prof. Paulo José de Freitas Filho, pelas sugestões feitas na apresentação do trabalho individual.

Agradeço à Daniela Barreiro Claro, pelo seu incentivo durante a realização deste trabalho e por seus comentários extremamente valiosos quando das discussões sobre Agentes Móveis; você mora em meu lado esquerdo do peito.

Aos meus amigos da pensão da Dona Dárcia (Rogério Azambuja, Ricardo Portes, Elemar Berbigier, Cristhian Flamarion Gomes de Carvalho, Carlos Pantaleão, Humberto, Marionei Zerbielli, Fernando e Roger), pela amizade e incentivo durante a escrita deste trabalho.

Às minhas amigas, em especial à Merisandra Cortês de Mattos e à Priscyla Waleska Targino de Azevedo Simões, pelo incentivo e apoio para a conclusão desta dissertação. Vocês ficarão em minhas lembranças.

Às funcionárias da secretaria do CPG-CC, Verinha e Valdete, que sempre estiveram prontas a nos atender e a passar a mão na nossa cabeça para nos confortar durante esta jornada; ficarei com saudades.

A toda a minha família, em especial à minha madrinha, Teresinha Adaci da Silva, e à minha tia, Valdeci Martins Rodrigues (*in memoriam*), e aos meus primos: Maria Angélica, João Rufino, Andréa, Joice, Luis Hipólito, Marcos e Flávio.

Agradeço a uma pessoa muito especial, Elisane Regina Cayser, que me incentivou a retornar ao mestrado e me deu todo o seu carinho, apoio e amor, que foram importantes para que este trabalho fosse realizado. Foi pena eu não ter sabido corresponder a isto tudo, mas não esquecerei você.

SUMÁRIO

LISTA DE FIGURAS.....	X
LISTA DE QUADROS.....	XI
LISTA DE TABELAS.....	XII
LISTA DE FÓRMULAS.....	XIII
LISTA DE GRÁFICOS	XIV
LISTA DE SIGLAS E ABREVIATURAS	XV
RESUMO.....	XIX
1. INTRODUÇÃO	21
1.1 JUSTIFICATIVA DA ESCOLHA DO TEMA.....	22
1.2 OBJETIVO GERAL	22
1.2.1 <i>Objetivos Específicos</i>	23
1.3. TRABALHOS RELACIONADOS AO ESTADO DA ARTE	23
1.4 ORGANIZAÇÃO DO TRABALHO	24
2. GERÊNCIA DE REDES.....	26
2.1 ÁREAS FUNCIONAIS DE GERENCIAMENTO DA ISO/OSI.....	26
2.1.1 <i>Gerenciamento de Falhas</i>	27
2.1.2 <i>Gerenciamento de Configurações</i>	27
2.1.3 <i>Gerenciamento de Desempenho</i>	27
2.1.4 <i>Gerenciamento de Contabilização</i>	27

2.1.5 Gerenciamento de Segurança.....	28
2.2 BASE DE GERENCIAMENTO DE INFORMAÇÕES (MIB).....	28
2.3 MODELO DE GERENCIAMENTO	30
2.4 PROTOCOLOS DE GERENCIAMENTO	33
2.4.1 Protocolo SNMP (Simple Network Management Protocol).....	33
2.4.2 Protocolo CMIP (Common Management Information Protocol).....	37
2.4.3 Comparações entre o SNMP e CMIP.....	37
2.5 GERENCIAMENTO CENTRALIZADO <i>VERSUS</i> DISTRIBUÍDO.....	39
2.6 CONCLUSÃO	40
3. AGENTES.....	42
3.1. TIPOS DE AGENTES.....	44
3.2. AGENTES MÓVEIS	46
3.3. VANTAGENS DO USO DE AGENTES MÓVEIS	47
3.4. LINGUAGENS DE PROGRAMAÇÃO E AGENTES MÓVEIS.....	49
3.5. SISTEMA DE AGENTES	50
3.6 PLATAFORMAS	52
3.6.1 Aglets	52
3.6.2 Voyager.....	53
3.6.3 Concordia	53
3.7 CONCLUSÃO	54
4. ESTUDO DE CASO - COMPARAÇÃO DO DESEMPENHO DO AGENTE MÓVEL E CLIENTE/SERVIDOR.....	55

4.1 AVALIAÇÃO DE DESEMPENHO	55
4.1.1 <i>Objetivos</i>	55
4.1.2 <i>Metodologia de Avaliação de Desempenho</i>	56
4.2. DEFINIÇÃO DO MODELO ANALÍTICO	64
4.2.1 <i>Modelo Analítico para o SNMP</i>	66
4.2.2 <i>Modelo Analítico para o paradigma Agente Móvel</i>	68
4.2.3 <i>Plataformas</i>	71
4.2.4 <i>Configuração da Máquina</i>	71
4.2.5 <i>Execução dos ambientes</i>	72
4.2.6 <i>Agente Móvel no Aglets</i>	73
4.2.7 <i>Agente Móvel no Voyager</i>	74
4.3 DEMONSTRAÇÃO DAS VANTAGENS DO AM EM ARQUITETURAS SNMP	75
4.4 ANÁLISE DO FATOR COMPRESSÃO	82
4.4.1 <i>Experimento</i>	84
5. CONCLUSÃO.....	85
5.1 <i>TRABALHOS FUTUROS</i>	86
6. REFERÊNCIAS BIBLIOGRÁFICAS	87

LISTA DE FIGURAS

FIGURA 1 - ESTRUTURA DA ÁRVORE DA MIB	29
FIGURA 2 - ELEMENTOS DE UM SISTEMA DE GERENCIAMENTO DE REDES	32
FIGURA 3 - CAMADAS DA ARQUITETURA DE REDES INTERNET	34
FIGURA 4 - MODELO DE GERÊNCIA SNMP	35
FIGURA 6- ESTRUTURA DO SERVIDOR DE AGENTES MÓVEIS	47
FIGURA 7 - MODELO DE GERÊNCIA CLIENTE/SERVIDOR	66
FIGURA 8 - MODELO DE GERÊNCIA UTILIZANDO AGENTES MÓVEIS.....	68
FIGURA 10 – MODELO SNMPv2 E v3 COM <i>MAX-REPETITIONS</i> = 10	76
FIGURA 11 – MODELO SNMPv2 E v3 COM <i>MAX-REPETITIONS</i> = 1000	76
FIGURA 12 – MIGRAÇÃO DO AM E O PEDIDO DE <i>GET-NEXT</i> LOCALMENTE	77

LISTA DE QUADROS

QUADRO 1 - GRUPOS DA MIB	30
QUADRO 2 - CRITÉRIOS PARA SELEÇÃO DE UMA TÉCNICA DE AVALIAÇÃO	60
QUADRO 3 – QUADRO DA DESCRIÇÃO DAS VARIÁVEIS NO MODELO SNMP	67
QUADRO 4 – QUADRO DA DESCRIÇÃO DAS VARIÁVEIS NO MODELO AGENTES MÓVEIS	70
QUADRO 5 – FATORES UTILIZADOS PARA A COMPRESSÃO E DESCOMPRESSÃO DO AGENTE MÓVEL.....	83

LISTA DE TABELAS

TABELA 1 – CRITÉRIOS PARA SELEÇÃO DE UMA TÉCNICA DE AVALIAÇÃO	61
TABELA 2 – TABELA PARA O CÁLCULO DOS TEMPOS TOTAIS DE GERENCIAMENTO NO MODELO SNMP	68
TABELA 3 – TABELA DOS VALORES EM UMA REDE LOCAL UTILIZANDO AGENTES MÓVEIS	71

LISTA DE FÓRMULAS

FÓRMULA 1 – FÓRMULA PARA CALCULAR A GERÊNCIA CLIENTE/SERVIDOR SNMP	67
FÓRMULA 2 – FÓRMULA PARA CALCULAR A GERÊNCIA TOTAL DA REDE USANDO SNMP	67
FÓRMULA 3 - MODELO ANALÍTICO PARA AGENTES MÓVEIS.....	70
FÓRMULA 4 - TESTE PARA VERIFICAR SE O DOCUMENTO DEVERÁ SER COMPACTADO	83

LISTA DE GRÁFICOS

GRÁFICO 1 - TEMPO DE RESPOSTA PARA O AGENTE MÓVEL E SNMP – LATÊNCIA 120 MS ...	78
GRÁFICO 2 - TEMPO DE RESPOSTA PARA O AGENTE MÓVEL E SNMP – LATÊNCIA 60MS	79
GRÁFICO 3 - TEMPO DE RESPOSTA PARA O AGENTE MÓVEL E SNMP – LATÊNCIA 1MS	79
GRÁFICO 4 - TEMPO DE RESPOSTA PARA O AGENTE MÓVEL E SNMP – LATÊNCIA 120MS	80
GRÁFICO 5 - TEMPO DE RESPOSTA PARA O AGENTE MÓVEL E SNMP – LATÊNCIA 60MS	81
GRÁFICO 6 - TEMPO DE RESPOSTA PARA O AGENTE MÓVEL E SNMP – LATÊNCIA 1MS	81
GRÁFICO 7 - COMPARAÇÃO ENTRE TEMPOS DE TRANSFERÊNCIA DE DOCUMENTOS SEM COMPRESSÃO E COM COMPRESSÃO COM AGENTES MÓVEIS	84

LISTA DE SIGLAS E ABREVIATURAS

AC	Ambiente Computacional
AGR	Aplicações de Gerenciamento de Redes
API	<i>Application Programming Interface</i>
ASN.1	<i>Abstract Syntax Notation Number One</i>
AT	<i>Address Translation</i>
ATM	<i>Asynchronous Transfer Mode</i>
ATP	<i>Agent Transfer Protocol</i>
Bps	<i>Bits por segundo</i>
BER	<i>Basic Encoding Rules</i>
CCITT	<i>Comité Consultatif International Télégraphique et Téléphonique</i>
CIM	<i>Common Information Model</i>
CMIP	<i>Common Management Information Protocol</i>
COD	<i>Code On Demand</i>
CORBA	<i>Common Object Request Broker Architecture</i>
CPU	<i>Central Processing Unit</i>
CS	Cliente/Servidor
DOS	<i>Disk Operating System</i>
DMTF	<i>Desktop Management Task Force</i>
DPI	<i>Distributed Protocol Interface</i>

EGP	<i>Exterior Gateway Protocol</i>
EGR	Entidade de Gerenciamento da Rede
GUI	<i>Graphical User Interface</i>
HMMP	<i>Hyper-Media Management Protocol</i>
HTTP	<i>HyperText Transfer Protocol</i>
IA	Inteligência Artificial
IAB	<i>Internet Activities Board</i>
ICMP	<i>Internet Control Message Protocol</i>
IETF	<i>Internet Engineering Task Force</i>
IP	<i>Internet Protocol</i>
ISO	<i>International Organization for Standardization</i>
ITU-T	<i>International Telecommunication Union - Sector T</i>
JDK	<i>Java Development Kit</i>
JVM	<i>Java Virtual Machine</i>
KB	<i>Kilobytes</i>
LAN	<i>Local Area Network</i>
MAG	<i>Mobile Agents Generator</i>
MAS	<i>Mobile Agent Server</i>
MCS	<i>Mobile Code Systems</i>
MIB	<i>Management Information Base</i>
MIT	<i>Management Information Tree</i>

MOF	<i>Managed Object Format</i>
MOS	<i>Messenger Operating System</i>
MTU	<i>Maximum Transmission Unit</i>
NMS	<i>Network Management Station</i>
ORB	<i>Object Request Broker</i>
OSI	<i>Open System Interconnection</i>
PDU	<i>Protocol Data Unit</i>
PPP	<i>Point to Point Protocol</i>
QoS	<i>Quality of Service</i>
RDPI	<i>Reverse Distributed Protocol Interface</i>
RAM	<i>Random Access Memory</i>
REV	<i>Remote Evaluation</i>
RFC	<i>Request For Comments</i>
RMI	<i>Remote Method Invocation</i>
RMON	<i>Remote Monitoring</i>
RPC	<i>Remote Procedure Call</i>
SDRAM	<i>Synchronous Dynamic RAM</i>
SNMP	<i>Simple Network Management Protocol</i>
SO	Sistema Operacional
SOAP	<i>Simple Object Access Protocol</i>
SOR	Sistema Operacional de Rede

SQL	<i>Structured Query Language</i>
TCP	<i>Transport Control Protocol</i>
TI	Tecnologia da Informação
UDP	<i>User Datagram Protocol</i>
XML	<i>Extensible Markup Language</i>
WAN	<i>Wide Area Network</i>

RESUMO

As redes de computadores têm aumentado o número de equipamentos, oriundos de diferentes fabricantes, os quais possuem, também, inúmeros protocolos para sua comunicação. Assim, os administradores têm enfrentado muitos problemas de gerenciamento dessas redes, pois o número de informações tem aumentado significativamente. Então, para gerenciá-las a contento, é preciso fazer uso dos recursos tecnológicos disponíveis. Hoje existe o paradigma Cliente/Servidor, o qual compete por banda na hora de gerenciar. Para isso, têm surgido algumas soluções a fim de minimizar a quantidade de tráfego nas redes, tanto na forma de executar serviços, como de gerenciá-la. Este trabalho se propõe em realizar uma modelagem analítica na comparação entre o paradigma Agente Móvel, que surge como uma alternativa ao paradigma Cliente/Servidor. Esse novo paradigma busca minimizar o problema de tráfego de rede, explorando algumas de suas funcionalidades, estimando métricas e definindo quando as mesmas devem ser utilizadas, fazendo, também, uma análise para estimar a viabilidade do agente trafegar compactado. As análises demonstram que ambos os paradigmas podem ser usados simultaneamente a fim de obter o melhor desempenho da rede, mas que o Agente Móvel pode ser superior em redes com maior latência.

Palavras Chaves: Gerência de Redes, SNMP, Agentes Móveis, Avaliação de Desempenho

ABSTRACT

Computer networks have increased the number of equipment from different makers and many protocols are used for communication among those devices. Managing such a heterogeneous network brought up more and more problems to administrators. Therefore, network administrators have had to make use of all available technological resources, since they deal with a significantly larger quantity of information. Today, we have the Client/Server paradigm, which (unfortunately) also competes for bandwidth. Therefore, some solutions have been created in order to minimize the amount of network traffic, both in the way services are executed as well as in the way they are managed. This work aims at accomplishing an analytic modeling to comparing the Mobile Agent (MA) paradigm, arising as an alternative, with the Client/Server paradigm, so as to minimize the problem of network traffic, explore some of the MA functionalities, assess metrics and define when they should be used. An analysis estimate the availability of compressing the Agent is also shown. Although the analyses show that both paradigms may be used simultaneously in order to achieve the best network performance, they also show that the Mobile Agent may be superior in networks with greater latency.

Keywords: Network Management, SNMP, Mobile Agents, Performance Evaluation

1. INTRODUÇÃO

Com o crescimento das redes de computadores, surgiu a necessidade de gerenciá-las constantemente, garantindo seu uso ininterrupto. Associado ao gerenciamento dessas redes está o planejamento do seu funcionamento. A gerência de redes consiste na monitoração dos estados e eventos relacionados aos recursos e na execução de ações de controle sobre os mesmos, a fim de garantir a disponibilidade da rede aos seus usuários. Para isso é necessário que se faça um gerenciamento de todos os elementos que a compõem.

Um dos mais importantes desafios de gerentes de instalações de processamento de dados é a gerência pró-ativa, i.e, aquela que se antecede ao surgimento dos problemas.

Para que isso seja possível, surge a necessidade de descentralização da gerência de redes, para a qual várias soluções vêm sendo propostas. O paradigma Cliente/Servidor vem sofrendo algumas modificações para tentar resolver o problema da gerência centralizada, e também para minimizar a quantidade de tráfego entre estação gerente-agente¹; entretanto, devido à competição por banda² junto das aplicações existentes, esta solução acaba alterando o desempenho da rede.

A mobilidade de código está sendo considerada como uma solução para descentralizar e, conseqüentemente, otimizar a gerência, uma vez que ela oferece um nível de flexibilidade necessário para lidar com os problemas. Para realizar o gerenciamento, existem inúmeras ferramentas, mas todas utilizam o paradigma Cliente/Servidor e, dessa forma, causam um aumento no *throughput*³ das redes. Para minimizar esse problema, o presente trabalho será desenvolvido seguindo o paradigma

¹ Gerente: é o módulo de software no monitor (estação de monitoramento) que executa as funções básicas de gerenciamento, como a recuperação de informações em outros componentes da configuração (TEIXEIRA JUNIOR, 1999).

Agente: é o módulo de software que obtém e armazena a informação de gerenciamento para um ou mais elementos da rede e passa essas informações para o monitor (TEIXEIRA JUNIOR, 1999).

² Banda: quantidade de tráfego que pode ser transportado por tempo da unidade.

³ *Throughput*: O parâmetro desempenho (*throughput*) calcula o número de *bytes* de dados do usuário transmitidos por segundo durante um determinado intervalo de tempo.

de agentes, buscando, através dele, verificar a viabilidade de minimizar o tráfego entre os equipamentos da rede.

1.1 JUSTIFICATIVA DA ESCOLHA DO TEMA

A escolha do tema gerência de redes, com o uso de agentes utilizando mobilidade, surgiu da necessidade imposta por algumas ferramentas: o tráfego intenso que as duas formas de comunicação entre as entidades (*polling*⁴ e registro de eventos) (STALLINGS, 1999) geram sobre a rede. Para realizar o gerenciamento, nem sempre a prioridade no tráfego é dada aos pacotes de informações de controle sobre os pacotes de dados do usuário; contudo, as informações de controle podem ser responsáveis pelo funcionamento da rede.

Inúmeros estudos têm sido realizados, com o intuito de solucionar ou amenizar o problema da gerência, tanto a sua centralização como a quantidade de informações para o monitoramento e controle. Este trabalho visa apresentar um estudo envolvendo uma análise analítica do paradigma Agente Móvel e SNMP (*Simple Network Management Protocol*) para indicar uma possível solução para tais problemas e, inclusive, oferecer este mecanismo como tendência para a construção de ferramentas que utilizem cada vez mais o paradigma dos agentes.

1.2 OBJETIVO GERAL

O objetivo geral deste trabalho é analisar a métrica⁵ de tempo de resposta para comparar os paradigmas Cliente/Servidor e Agentes Móveis. Esta análise servirá de base para decidir quando utilizar algumas de suas funcionalidades, tanto para decidir o uso do paradigma como para estimar o seu melhor desempenho, e, dessa maneira, chegar a uma melhor performance na gerência de redes de computadores, independentemente da topologia e das plataformas utilizadas.

⁴ *Polling*: é uma interação do tipo solicita/recebe resposta entre um gerente e um agente.

⁵ Métrica: ato de determinação de uma medida.

1.2.1 OBJETIVOS ESPECÍFICOS

- Minimizar o tráfego entre objetos-gerenciados e objetos-gerentes.
- Comparar a avaliação de desempenho entre os paradigmas Agentes Móveis e SNMP para o gerenciamento de redes usando a métrica tempo de resposta.
- Verificar e analisar em quais situações os Agentes Móveis podem melhorar o desempenho real da rede.
- Verificar em quais situações cada um dos dois paradigmas deve ser utilizado.

1.3. TRABALHOS RELACIONADOS AO ESTADO DA ARTE

O paradigma de Agentes Móveis vem sendo amplamente discutido, e inúmeros trabalhos apontam as vantagens advindas da sua utilização em diversos tipos de aplicações, principalmente na gerência de redes e telecomunicações.

Conforme Chess (1995), vários autores sugerem que os Agentes Móveis oferecem um método importante para executar transações e resgatar as informações em redes. Lange (1999) aponta os principais benefícios dos Agentes Móveis em ambientes distribuídos. Dentre os benefícios descritos, verificam-se a redução da carga na rede e a superação da latência. Esse autor enumera, também, várias aplicações que se beneficiam deste paradigma, tais como o resgate de informações distribuídas, a monitoração e a notificação, que são típicas da gerência e utilizadas pelo planejamento de capacidade.

A utilização de Agentes Móveis na descentralização da atividade de gerência vem sendo discutida em diversas pesquisas. Em Baldi et al. (1997;1998) encontra-se uma avaliação de vários paradigmas de código móvel em comparação com a arquitetura Cliente/Servidor no domínio da gerência de redes. Esses autores concluem que, através de uma avaliação quantitativa, a melhor solução a ser empregada na gerência de uma rede depende das características desta rede e das tarefas de gerência a serem executadas.

Bieszczad et al. (1998) também mostram várias aplicações de Agentes Móveis nas cinco áreas funcionais de gerenciamento definidas pela OSI (*Open System Interconnection*). Gray et al. (2000) descrevem as motivações e razões para a utilização de Agentes Móveis na gerência de redes e o estado da arte dos sistemas baseados em AM. Gavalas et al (2000) também propuseram uma infra-estrutura para utilização de Agentes Móveis na gerência de redes, e constatou que esse paradigma resulta em um ganho no tempo de resposta, e na utilização da largura de banda, em comparação com o SNMP. Puliafito et al. (1999) propuseram uma plataforma MAP (*Mobile Agents Platform*) que foi utilizada para criar agentes que desempenham funcionalidades simples de gerenciamento, como consulta a MIB (*Management Information Base*) dos elementos de rede. Esse mesmo autor desenvolveu uma API (*Application Programming Interface*) para que os agentes coletem dados das MIBs dos elementos de rede e uma interface que permite que os agentes sejam gerenciados remotamente por um gerente. Utilizando essa interface, estão as funcionalidades dos agentes que podem ser estendidas sem a necessidade de recompilação do código.

Alguns dos pesquisadores procuram mostrar a necessidade da integração de Agentes Móveis com SNMP para dar suporte às aplicações de legado e também por questões de migração e eficiência. Conforme Simões et al. (1999), esta integração é importante pelo fato do SNMP ser a única interface disponível para acessar as informações de gerenciamento. Zapf et al. (1999) apresentam uma pequena análise quantitativa dos benefícios da solução baseada em AM na redução da carga da rede.

1.4 ORGANIZAÇÃO DO TRABALHO

O presente trabalho está dividido em cinco capítulos. O primeiro capítulo apresenta a introdução, fornecendo uma visão geral do trabalho, justificativa, incluindo os objetivos gerais e específicos, e trabalhos relacionados ao estado da arte.

O segundo capítulo apresenta a área de gerência de redes, a necessidade de gerenciamento, o modelo de gerenciamento, as áreas funcionais do gerenciamento

definidas pela ISO, a MIB, os protocolos de gerenciamento e, por último, uma comparação entre SNMP e CMIP (*Common Management Information Protocol*).

O terceiro capítulo apresenta a definição e caracterização dos agentes, benefícios da mobilidade, a utilização dos agentes e das plataformas utilizadas, uma abordagem sobre as linguagens de desenvolvimento e alguns produtos.

O quarto capítulo apresenta uma descrição da metodologia utilizada para a avaliação de desempenho, juntamente com uma definição dos valores e métricas utilizados. Possui também o desenvolvimento parcial do ambiente proposto, com as plataformas utilizadas, configuração da máquina, execução dos ambientes, Agentes Móveis no *Aglets* e *Voyager* e a proposta de trabalho bem como as definições de modelo analítico tanto de modelo SNMP quanto do modelo Agente Móvel; também possui uma análise do fator de compressão do agente a ser transferido pela rede.

O quinto capítulo apresenta alguns aspectos sobre a conclusão deste trabalho e também algumas indicações sobre a possibilidade de trabalhos futuros relacionados ao tema.

2. GERÊNCIA DE REDES

Uma rede de computadores, se não for devidamente gerenciada, pode apresentar grandes problemas, em longo prazo (IBM, 1999). É fácil imaginar a dificuldade de se interconectar equipamentos tão diferentes tais como: computadores, comutadores, roteadores, dentre outros, se as convenções de gerenciamento (como uso de alarmes, indicadores de performance, estatísticas de tráfego e contabilização) forem muito diferentes entre si, com características próprias de cada equipamento ou fabricante. A dificuldade aumenta conforme a quantidade de componentes forem adicionados a elas, agregando mais funções e mais usuários.

Essas entidades podem existir sem mecanismos de gerenciamento; todavia, seu uso pode encontrar dificuldades com congestionamento, segurança, roteamento e desempenho. A atividade de gerenciamento está associada ao controle das atividades e do monitoramento do uso dos recursos. O trabalho básico engloba métodos para planejar, configurar, controlar, monitorar, corrigir falhas e administrar redes de computadores e telecomunicações. Para cumprir esses objetivos, as suas funções devem ser embutidas nos diversos componentes da rede, possibilitando descobrir, prever e reagir a problemas. Além disso, destaque-se a existência de gerência pró-ativa, que é aquela capaz de detectar os problemas antes de eles acontecerem e, dessa forma, permitir resolvê-los para que não influenciem no ambiente do sistema.

2.1 ÁREAS FUNCIONAIS DE GERENCIAMENTO DA ISO/OSI

O modelo funcional tem por objetivo descrever um conjunto de funções aos usuários, para administrarem as suas redes de modo a permitir que as aplicações realizem os serviços necessários ao controle e gerenciamento.

2.1.1 GERENCIAMENTO DE FALHAS

Tem como objetivo garantir o funcionamento contínuo da rede e de seus serviços. Possui um conjunto de funções destinadas à detecção, diagnóstico e à correção ou ao isolamento de anomalias no funcionamento do ambiente, sejam estes provocados por um problema material ou pelo mau funcionamento de um componente de software. Para isso, é realizada uma seqüência de testes de diagnóstico.

2.1.2 GERENCIAMENTO DE CONFIGURAÇÕES

Gerencia os serviços de interconexão de sistemas. Corresponde ao conjunto de funções relativas ao controle, à identificação, à supervisão e à coleta de informações sobre os objetos gerenciados.

2.1.3 GERENCIAMENTO DE DESEMPENHO

Possibilita a avaliação do comportamento dos objetos gerenciados numa tentativa de garantir a QoS (Qualidade de Serviço) acordada com os usuários. Fornece funções que permitem coletar dados estatísticos, manter e examinar o histórico do estado dos objetos, para um planejamento e análise do sistema.

2.1.4 GERENCIAMENTO DE CONTABILIZAÇÃO

Determina o custo de utilização da rede e dos recursos nela disponíveis. Essa função define, para quais recursos podem ser faturados os custos relativos a sua utilização, possibilitando a combinação desses custos distribuídos, no caso de vários recursos serem solicitados na realização das necessidades dos usuários.

2.1.5 GERENCIAMENTO DE SEGURANÇA

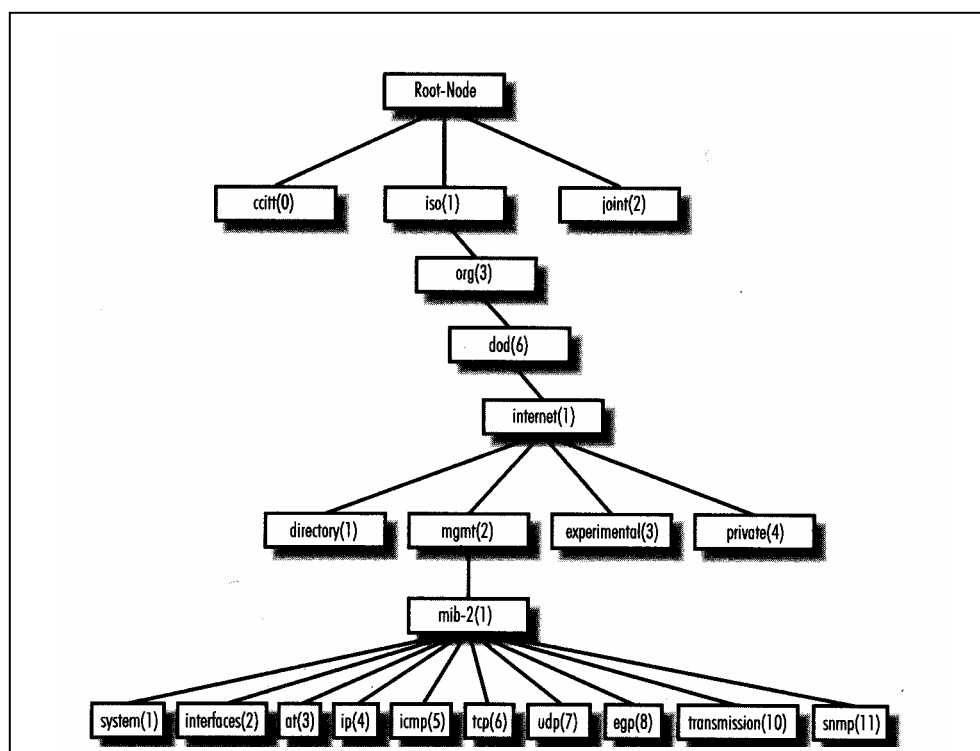
Corresponde ao conjunto de funções necessárias à criação, à supressão e ao controle dos mecanismos e serviços de segurança e do tráfego de informações sigilosas na rede. Assim, esse gerenciamento suporta a política OSI, relacionada ao bom funcionamento do gerenciamento OSI, bem como a proteção dos recursos suportados por esse gerenciamento.

2.2 BASE DE GERENCIAMENTO DE INFORMAÇÕES (MIB)

A MIB (*Management Information Base*) é o conjunto dos objetos gerenciados, que procura abranger todas as informações necessárias para a gerência da rede, possibilitando, assim, a automatização de grande parte das tarefas.

Basicamente, são definidos quatro tipos de MIBs: MIB I, MIB II, MIB experimental e MIB privada. As MIBs do tipo I e II fornecem informações gerais sobre o equipamento gerenciado, sem levar em conta as suas características.

Os objetos gerenciáveis definidos na MIB II estão divididos em grupos, sendo esta uma ampliação das informações contidas na MIB I. Cada grupo é composto por uma tabela com objetos gerenciados. Sua hierarquia dá-se da seguinte forma:



Fonte: (MAURO; SCHIMIDT, 2001)

Figura 1 - Estrutura da Árvore da MIB

Observa-se que o nó da árvore não tem nome ou número, mas tem três sub-árvores:

- 1) *ccitt(0)*, administrada pelo CCITT (ITU-T);
- 2) *iso(1)*, administrada pela ISO;
- 3) *joint-iso-ccitt(2)*, administrada pela ISO juntamente com o CCITT.

Sob o nó *iso(1)*, estão outras sub-árvores, como é o caso da *org(3)*, definida pela ISO para conter outras organizações. Uma das organizações que está sob a sub-árvore *org(3)* é o Departamento de Defesa dos EUA (DOD), no nó *dod(6)*. A *Internet(1)* está sob o *dod(6)* e possui quatro sub-árvores:

- *directory(1)*: contém informações sobre o serviço de diretórios OSI (X.500);

- *mgmt(2)*: contém informações de gerenciamento; é sob esta sub-árvores que está o nó da MIB II, com o identificador de objeto 1.3.6.2.1;
- *experimental(3)*: contém os objetos que ainda estão sendo pesquisados pela IAB (*Internet Activities Board*);
- *private(4)*: contém objetos definidos por outras organizações.

Abaixo da sub-árvore MIB II estão os objetos usados para obter informações específicas dos dispositivos da rede. Esses objetos são divididos em 11 grupos.

Os grupos da MIB II são:

Quadro 1 - Grupos da MIB

Grupos	Informações
<i>System(1)</i>	Sistema de operação dos dispositivos da rede
<i>*interfaces(2)</i>	Interface da rede com o meio físico
<i>Address translation(3)</i>	Mapeamento de endereços IP em endereços físicos
<i>Ip(4)</i>	Protocolo IP
<i>Icmp(5)</i>	Protocolo ICMP
<i>Tcp(6)</i>	Protocolo TCP
<i>Udp(7)</i>	Protocolo UDP
<i>Egp(8)</i>	Protocolo EGP
<i>Transmission(10)</i>	Meios de Transmissão
<i>Snmp(11)</i>	Protocolo SNMP

Fonte: (MAURO; SCHIMIDT, 2001)

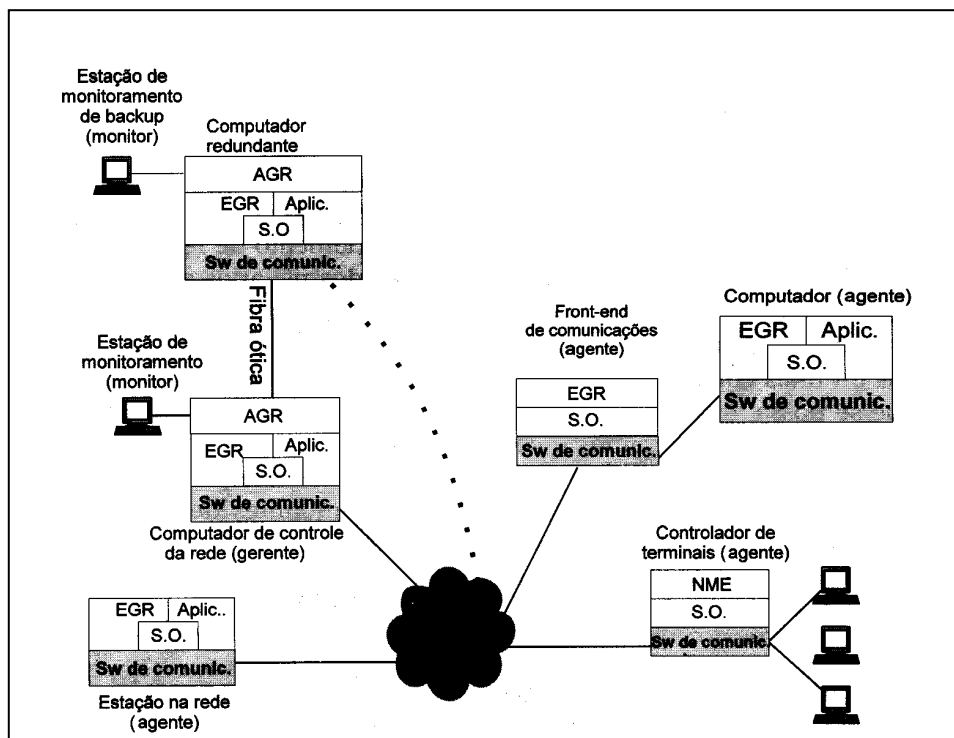
2.3 MODELO DE GERENCIAMENTO

O gerenciamento de uma rede deve utilizar um computador que interaja com os vários componentes dessa rede, de forma a extrair informações necessárias para a sua administração. O computador considerado gerente da rede contém uma base de dados com informações adequadas/necessárias para diagnosticar e buscar soluções para os problemas existentes.

As arquiteturas possuem um modelo clássico com os seguintes componentes: gerente, agente, base de informação de gerenciamento (MIB) e o protocolo de gerência. Os modelos de gerência mais adotados atualmente são baseados na arquitetura Cliente/Servidor ou gerente/agente.

O gerente e o agente interagem de forma a consultar a base de informações. Essa comunicação se dá pelo uso de um protocolo de gerência, responsável por definir as operações executadas por ambos.

Um sistema de gerenciamento de redes é composto de ferramentas para o monitoramento e o controle das atividades realizadas por essa rede. Ele consiste em módulos de software e também de hardware, implementados em seus componentes. O conjunto de softwares que implementa essas tarefas é denominado, quando instalado em um dos nós da rede, de EGR (entidade de gerenciamento da rede). Pelo menos um dos computadores na rede será designado para executar o papel de gerente. Além de suas EGRs, o gerente possui também uma série de programas chamados AGRs (aplicações de gerenciamento de redes). Esses programas tratam do monitoramento de desempenho da rede, do monitoramento de falhas e do controle de configuração. A AGR atende aos comandos dos operadores mostrando resposta na tela ou emitindo comandos para as EGRs espalhadas pela rede. Essa comunicação é efetuada por meio de um protocolo padrão de gerenciamento de redes na camada de aplicação (por exemplo, SNMP ou CMIP).



Fonte: (TEIXEIRA JUNIOR, 1999)

Figura 2 - Elementos de um Sistema de Gerenciamento de Redes

As tarefas de uma EGR (Entidade de Gerenciamento de Rede) são as seguintes:

- Coletar estatísticas sobre as atividades da rede.
- Armazenar estas estatísticas localmente.
- Atender aos comandos do centro de controle, incluindo:
 - transmitir as estatísticas coletadas para o centro de controle da rede;
 - modificar parâmetros;
 - fornecer informações de estado;
 - gerar tráfego artificial para teste.

2.4 PROTOCOLOS DE GERENCIAMENTO

Pode-se citar dois protocolos de gerência mais adotados: o CMIP (*Common Management Information Protocol*) para o modelo OSI, e o SNMP (*Simple Network Management Protocol*) para o modelo Internet, o mais amplamente utilizado atualmente.

Atualmente, com o advento da *Web*⁶, novas alternativas têm sido propostas. Uma delas é o protocolo HMMP (*Hyper-Media Management Protocol*), sob comando da *Desktop Management Task Force* (DMTF). A DMTF está trabalhando para a definição de padrões para a indústria através de uma nova sintaxe, a *Managed Object Format* (MOF), baseada no *Common Information Model* (CIM). Com uma abrangência maior em termos de usuários, o próprio protocolo padrão da *Web*, o HTTP (*HyperText Transfer Protocol*), tem sido utilizado para operações de gerência (MULLANEY, 1996). Outros padrões, como, por exemplo, o CORBA (*Common Object Request Broker Architecture*)⁷ estão sendo indicados para gerência na *Web* (EVANS; ROGERS, 1995).

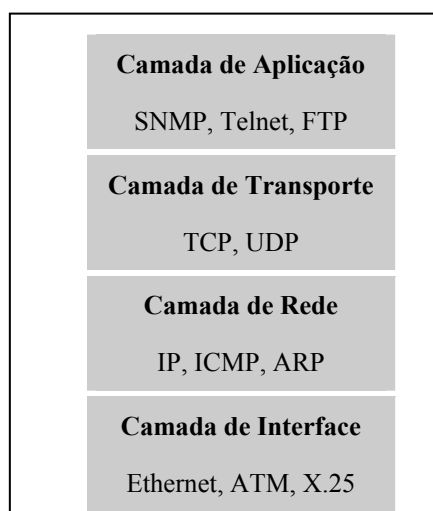
2.4.1 PROTOCOLO SNMP (*SIMPLE NETWORK MANAGEMENT PROTOCOL*)

O SNMP, desenvolvido pelo IETF (*Internet Engineering Task Force*) é usado amplamente para administrar redes TCP/IP. É um protocolo sem conexão. Os agentes SNMP residem nos dispositivos gerenciados e são projetados para operar usando o mínimo de recursos do sistema, não reduzindo a velocidade de operação do dispositivo para prover serviços de gerência de rede. O agente coleta dados sobre o número e tipo de mensagens de erro recebidas, o número de *bytes* e pacotes processados pelo dispositivo, o comprimento máximo da fila, e assim sucessivamente, e os armazena na MIB que reside naquele dispositivo. O processamento dos dados coletados é encerrado

⁶ *Web*: Rede Mundial de Computadores.

⁷ CORBA - *Common Object Request Broker Architecture* - é um padrão definido pelo OMG - *Object Management Group* - que permite a comunicação entre aplicações localizadas em máquinas diferentes e implementadas utilizando diferentes linguagens de programação.

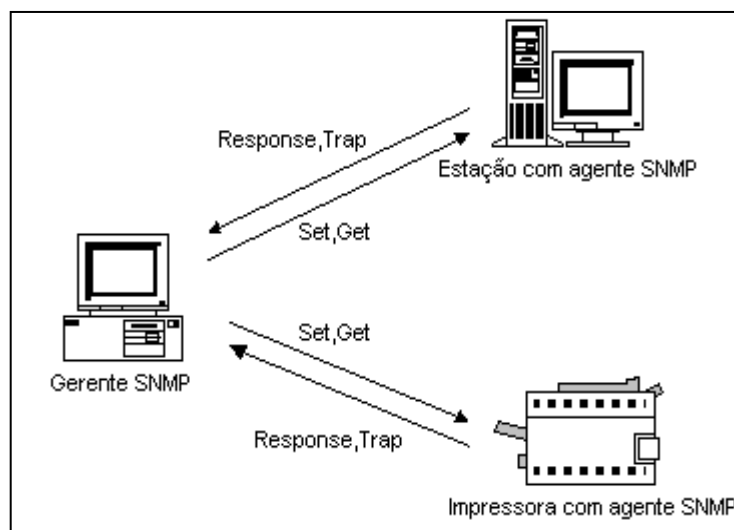
pela aplicação de gerenciamento. A aplicação de gerenciamento se comunica com os dispositivos gerenciados através do SNMP.



Fonte: Adaptado de (OLIVEIRA, 2000)

Figura 3 - Camadas da Arquitetura de Redes Internet

O SNMP é um protocolo de gerenciamento padrão para redes IP. O modelo de gerente/agente realiza o gerenciamento pela comunicação entre o gerente e vários agentes. O gerente controla de ponta a ponta a rede, enquanto o agente ajusta e controla os objetos gerenciáveis sob sua responsabilidade da forma como foi determinado pelo gerente, relatando os resultados para o gerente. Atualmente existem três versões do SNMP.



Fonte: (OLIVEIRA; 2000)

Figura 4 - Modelo de Gerência SNMP

O protocolo SNMPv1 ainda é a versão mais utilizada para gerenciamento de redes. Nessa versão foram definidas as principais funcionalidades do protocolo e suas unidades de dados. O SNMPv2 é apenas uma atualização da anterior, expandindo as funcionalidades do SNMP e aperfeiçoando a capacidade de gerenciamento dos agentes com o suporte a *proxies*⁸ e uma maior integração à plataforma de gerenciamento da OSI. O SNMPv3 trata sobretudo das questões referentes à segurança, acrescentando mecanismos de autenticação, privacidade, autorização. Também trata da administração, acrescentando mecanismos de nomeação de entidades, gerência de políticas de usuários, relações com *proxies* e configuração remota através de operações próprias.

O gerente pode solicitar a um agente (para o qual ele tenha autorização) o envio de valores de diversos elementos de informação. O agente responde com os valores que estão contidos em sua MIB.

O SNMP permite que a estação gerente manipule os agentes gerenciáveis através das seguintes primitivas:

⁸ *Proxy*: controla o fluxo de pacotes entre uma rede local (privada) e a rede pública (*Internet*).

- *get*: indica uma operação de consulta dos valores da MIB;
- *get-next*: permite emitir uma seqüência de comandos para recuperar um grupo de valores de uma MIB;
- *get-bulk*: permite que um aplicativo de gerenciamento recupere uma grande seção de uma tabela, de uma só vez;
- *get-response*: informação com resultado da operação;
- *set*: indica uma operação de modificação de valores da MIB;
- *trap*: informação de notificação enviada pelo agente ao gerente;
- *notification*: (SNMPv2 e SNMPv3);
- *information*: (SNMPv2 e SNMPv3);
- *report*: (SNMPv2 e SNMPv3);

O SNMP apresenta duas formas de comunicação: *polling* e registro de eventos.

O *polling* é uma interação freqüente entre o gerente e o agente, em que o gerente verifica se o agente está ativo e pode, opcionalmente, requisitar informações de sua MIB.

No registro de eventos, a iniciativa da comunicação parte do agente. O agente é configurado anteriormente para enviar informações periódicas sobre o seu estado de funcionamento para o gerente. As informações enviadas referem-se à ocorrência de algum evento significativo como troca de estado, falhas ou mesmo dados da MIB para atualizar algum documento dinâmico.

2.4.2 PROTOCOLO CMIP (*COMMON MANAGEMENT INFORMATION PROTOCOL*)

O CMIP comporta vários tipos de PDUs (*Protocol Data Unit*) que são mapeadas em operações equivalentes sobre os objetos gerenciados, os quais representam os recursos gerenciados. Essas PDUs são basicamente as seguintes:

- **M-GET**: executa a leitura dos atributos de objetos gerenciados;
- **M-SET**: executa a modificação dos atributos de objetos gerenciados;
- **M-ACTION**: executa uma ação qualquer sobre um objeto gerenciado;
- **M-CREATE**: cria uma instância de um objeto gerenciado;
- **M-DELETE**: remove uma instância de um objeto gerenciado;
- **M-EVENT-REPORT**: emite uma notificação sobre a ocorrência de um evento associado a um objeto gerenciado.

Além dessas mensagens de protocolos, são definidas facilidades adicionais que permitem selecionar o grupo de objetos sobre o qual é aplicável uma dada operação. A facilidade denominada escopo nos permite selecionar um grupo de instâncias de objetos sobre os quais é realizada uma única operação. A facilidade de filtro, por sua vez, permite definir um conjunto de testes aplicáveis a um grupo de instâncias de objeto, anteriormente selecionado através da facilidade de escopo, de modo a extrair um subgrupo ainda menor sobre o qual deve ser efetuada uma operação de gerenciamento. Por último, há a facilidade de sincronização, que permite sincronizar várias operações de gerenciamento a serem realizadas sobre instâncias de objetos selecionadas através das facilidades de escopo e filtro.

2.4.3 COMPARAÇÕES ENTRE O SNMP E CMIP

As principais similaridades entre os dois protocolos são:

- Ambos os protocolos usam o conceito de MIB.
- São protocolos de arquiteturas abertas (não proprietárias).

As diferenças entre o SNMP e o CMIP podem ser resumidas nos seguintes tópicos:

- Acesso a dados: SNMP é orientado mais à recuperação individual de itens de uma informação, enquanto o CMIP é mais orientado à recuperação de informações agregadas.
- *Polling x Reporting*: SNMP trabalha por *polling* (o gerente regularmente pergunta ao agente sobre seu *status*), enquanto o CMIP usa *reporting* (o agente informa ao gerente quando o *status* do objeto gerenciado mudou). Esta filosofia do CMIP é vantajosa sobre o SNMP quando se tem um número considerável de agentes a serem consultados.
- Tamanho e desempenho: SNMP é menor e mais rápido. CMIP requer maior capacidade de processamento e mais memória. Essa característica é associada à filosofia *polling*, que requer menos inteligência dos dispositivos sendo gerenciados.
- Funcionalidade: CMIP tem mais características e capacidades específicas.
- Nível de transporte: SNMP requer somente datagramas⁹ não confiáveis, o que implica que ele pode ser implementado em várias redes. O CMIP exige um nível de transporte orientado à conexão.
- Padronização: CMIP, a exemplo dos demais protocolos OSI, é um padrão internacional sujeito, portanto, a testes de conformidade. SNMP, por sua vez, somente pode dispor de testes de interoperabilidade.

⁹ Datagrama: Os pacotes independentes da organização, sem conexão, são chamados de datagramas, em analogia aos telegramas.

2.5 GERENCIAMENTO CENTRALIZADO *VERSUS* DISTRIBUÍDO

Com a gerência centralizada, há sérios problemas com o tráfego excessivo gerado entre a estação gerente e os dispositivos monitorados, podendo gerar limitações de flexibilidade, escalabilidade¹⁰, desempenho e eficiência, podendo, desta forma, contribuir para o congestionamento nas redes.

Como alternativa ao gerenciamento centralizado, alguns autores apresentam outros paradigmas: gerenciamento hierárquico fortemente distribuído e fracamente distribuído e o gerenciamento cooperativo.

Quanto às propostas e aos modelos relacionados ao gerenciamento distribuído citam-se: gerência por delegação (SILVA et al., 2000), agentes móveis (BRENNER et al., 1998) e redes ativas (TENNENHOUSE; WETHERALL, 1996).

A gerência por delegação (*Management by Delegation*) foi especificada em 1995. Ela possibilita a construção de sistemas distribuídos em gerência de redes. Para isso, ao invés de codificar todo o conhecimento de gerência dentro de uma única entidade, como o gerente, nesse modelo de gerência, certas responsabilidades existentes passam a ser delegadas para outros processos distribuídos na rede. Algumas das tarefas podem ainda ser controladas pela autoridade central, mas os gerentes locais podem executar várias tarefas de forma autônoma.

Na gerência por delegação, o objetivo é a descentralização do gerente através da delegação das tarefas de gerenciamento para processos localizados no recurso gerenciado ou localizados em máquinas que desempenham o papel de gerentes intermediários. A arquitetura de gerenciamento deste modelo inclui um protocolo de gerenciamento, agentes e ambientes de suporte para processos que devem ser executados e que devem estar presentes em cada dispositivo da rede.

Para diferenciar do modelo de gerenciamento tradicional, a estação de gerência faz o empacotamento de uma tarefa (código e dados) e o envia, como agentes, para ser

executada nos dispositivos da rede, delegando para eles a execução das tarefas. As execuções podem ser assíncronas, liberando a estação central para executar outras tarefas, gerando conseqüentemente um grau de paralelismo na arquitetura de gerenciamento.

O modelo de gerência por delegação em contraste com a gerência centralizada proporciona, as seguintes vantagens:

- automação das funções de gerenciamento através das aplicações delegadas que reduzem a carga da estação central de gerência;
- redução da necessidade de constantes *pollings* na rede entre o gerente e os agentes, aumentando, assim, o desempenho da rede.

Deve-se ter cuidado com a questão de autoridade entre os gerentes delegados, pois cada gerente deve saber o que pode controlar e a quem se reportar, de forma que o sistema fique coerente e não haja conflito de instruções entre gerentes diferentes.

Segundo Simões et al. (1999), a descentralização da gerência resolve potencialmente a maioria dos problemas existentes nas soluções Cliente/Servidor centralizadas, tornando as redes mais escaláveis, robustas, facilmente atualizáveis e personalizáveis, além de reduzir o tráfego na rede.

2.6 CONCLUSÃO

Quando ocorre a implantação de uma rede, em uma organização, é de fundamental importância que se introduza uma gerência, tanto para controlar e monitorar essa rede e garantir a manutenção de sua funcionalidade como para projeções futuras, estabelecimento de métricas e necessidades.

¹⁰ Escalabilidade: é a capacidade de ampliar o poder computacional da rede adicionando novos computadores ou servidores.

O modelo de gerenciamento SNMP é amplamente utilizado. As MIBs proprietárias são desenvolvidas por muitos fabricantes, e os agentes instalados nos dispositivos de rede aumentam o custo do valor final do equipamento. Em função disso, as ferramentas de gerenciamento procuram garantir a interoperabilidade entre os dispositivos gerenciáveis de forma que eles possam se comunicar entre si.

O modelo de gerência por delegação, em contraste com a gerência centralizada, chega a proporcionar vantagens significativas que solucionam ou amenizam o problema do tráfego intenso que as informações de gerência geram sobre a rede. Mas essa solução depende da estrutura e da forma de implantação na rede.

De acordo com Zapf (1999), o grau de descentralização não é igualmente aplicável a todas as áreas de gerência e a todas as redes. A situação deve ser cuidadosamente analisada com o intuito de se atingir os melhores resultados quanto à descentralização desejada, a utilização da largura de banda, a tolerância a falhas, e assim por diante.

A avaliação do desempenho dessas diferentes soluções é muito importante, para que se possa ter uma alternativa de escolha de um modelo que melhor se adapte à realização do gerenciamento capaz de identificar os possíveis problemas, de preferência antes que eles venham a ocorrer, procurando, assim, evitá-los.

3. AGENTES

Atualmente, o tráfego das informações na *Internet* e nos meios de comunicação em geral tem crescido bastante. Isso ocorre devido à grande melhoria na infra-estrutura de comunicação, em relação aos equipamentos e à estruturação, além do aumento do tráfego das informações que estão circulando pela grande rede. O acesso a essas informações também gerou um crescimento significativo – em quantidade – ao acesso à rede *Internet*, pois muitas pessoas possuem computadores pessoais e realiza os seus acessos tanto do seu escritório quanto da sua residência. Esse fato foi marcante após o barateamento dos custos relativos aos acessos a redes e devido ao surgimento da comunicação sem fio, evidenciando o uso dos portáteis.

O surgimento da *Internet* permitiu o aparecimento de novos meios de comunicação, cooperação e negócios entre usuários. Atualmente a tecnologia de *Web Services*¹¹ enfatiza a globalização e a utilização dessa grande rede de forma bastante disseminada. Há, porém, alguns desafios a serem ponderados, tais como a escalabilidade, visto que as soluções propostas devem estar aptas a interagir com redes de grandes escalas; a mobilidade física, fato este evidenciado pelas desconexões freqüentes, além das limitações de memória e largura de banda; o dinamismo das tendências tecnológicas, devido à necessidade de flexibilidade das aplicações.

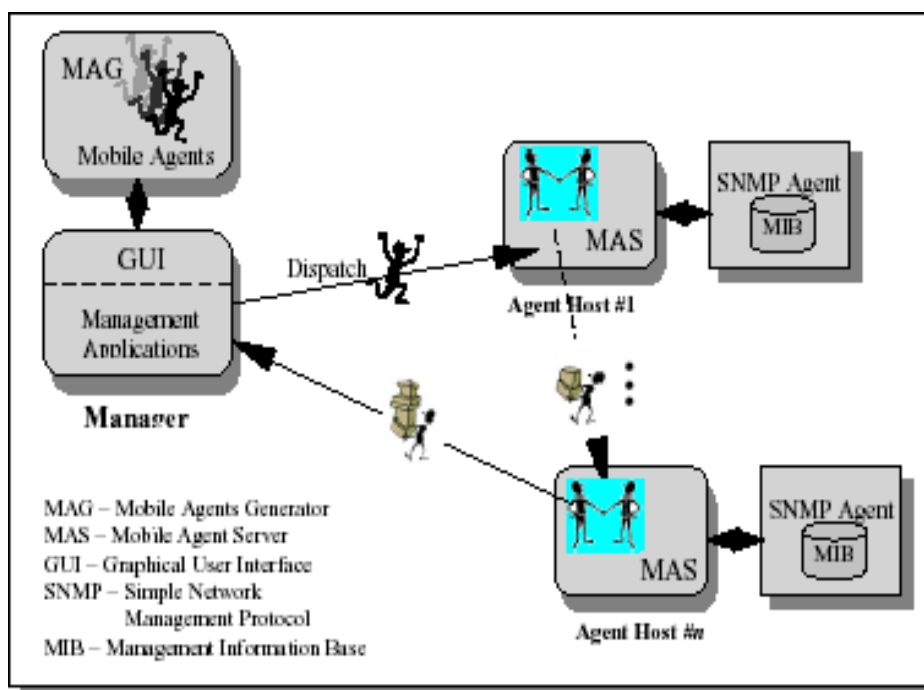
Há várias tentativas de aplicações moldarem-se aos novos paradigmas que surgem efemeramente, tais como CORBA, RMI (*Remote Method Invocation*), SOAP¹² (*Simple Object Access Protocol*); entretanto tais paradigmas não garantem a escalabilidade, a flexibilidade e as reconfigurações altamente exigidas pelas aplicações modernas. Uma das alternativas é explorar a mobilidade de código que permite “alterar dinamicamente as ligações entre os componentes”, permitindo maior flexibilidade, melhor utilização da infra-estrutura de comunicação e uma provisão de serviços de maneira autônoma, sem se preocupar com as conexões permanentes das redes. A utilização da mobilidade fez

¹¹*Web Services*: promete fazer com que clientes e servidores heterogêneos possam compartilhar aplicações usando módulos que são descritos, publicados, localizados e invocados usando uma rede de forma transparente.

¹²*SOAP*: é um protocolo de comunicação baseado em XML.

ressurgir a noção dos agentes de software permitindo uma interoperabilidade de diversas áreas.

O termo agente já era muito utilizado na área de Inteligência Artificial (IA), devido à necessidade de explorar as interações complexas com o mundo físico. Paralelamente à IA, surgia o termo Agente de Software, permitindo o desenvolvimento de partes do código, menores e mais confiáveis, garantindo assim a sua robustez. Várias definições de agentes foram incorporadas e, segundo elas, “agente são programas de computadores que empregam técnicas de IA para fornecer assistência ativa para usuários em tarefas baseadas em computador” (MÃES, 1994).



Fonte: (IBM, 1999)

Figura 5 - Infra-estrutura de uma Gestão de Redes baseada em AM

Há determinadas características que diferenciam agentes de programas comuns, ou de simples objetos desenvolvidos com códigos robustos, dentre as quais destaca-se a autonomia, por garantir ao agente a operabilidade sem a intervenção do usuário direto, pois os agentes possuem algum tipo de controle sobre de suas ações e dos seus estados internos. Além disso, os agentes devem ter a capacidade de interagir com outros agentes ou seres humanos, utilizando para isso alguma linguagem de comunicação, garantida pela habilidade social que o agente possui. Em se tratando de ambientes, os agentes

devem ser capazes de reagir aos ambientes em que estão inseridos e de tomar iniciativas em relação aos seus comportamentos. A orientação a objetivos é um fator preponderante nos agentes, pois uma vez que recebam uma determinada tarefa dos seus proprietários, esses agentes devem, se necessário, mover-se na rede para desempenhar tal função. Outrossim, a racionalidade, a adaptabilidade e a colaboração são características que os agentes de um modo geral devem possuir. A racionalidade se caracteriza pelo fato de os agentes irem ao encontro dos seus objetivos e não de encontro a eles. Adaptabilidade significa que uma vez que esses agentes cheguem em determinado ambiente, eles devem se adaptar a este e, muitas vezes, colaborar entre si para desempenharem um objetivo em comum.

3.1. TIPOS DE AGENTES

Os agentes podem ser classificados em vários tipos tais como:

- **Agentes Reativos:** Agem somente em resposta aos estímulos externos quando eles acontecem; não há, pois, nenhuma previsão de ações futuras, eles reagem momentaneamente.
- **Agentes Deliberativos:** São os que se revestem de características principalmente colaborativas, visto que se baseiam nas interações com outros agentes, negociando o cumprimento das tarefas que a eles foram designadas
- **Agentes Autônomos:** Executam as tarefas sem a intervenção humana, situam-se internamente em um ambiente de autodomínio para que possam executar suas tarefas com uma determinada autonomia. Um exemplo é o SNMP.

- **Agentes Inteligentes:** Estes tipos de agentes são oriundos principalmente da Inteligência Artificial, são específicos para a realização de um determinado problema, detêm uma base de conhecimento que lhes permite ir aprendendo com as suas próprias ações.
- **Agentes Estáticos:** São também chamados de agentes fixos ou estacionários; mantêm-se em uma mesma máquina, restringindo-se, assim, aos conhecimentos e recursos da própria máquina em que está hospedado.
- **Agentes Coordenados:** São capazes de gerenciar interdependências entre atividades. Os agentes podem ser coordenados com o intuito de alcançar um objetivo maior, ou ainda completar suas atividades.
- **Agentes Aprendizes e Adaptáveis:** São agentes que observam as ações dos usuários em *background*¹³, analisam o comportamento dos usuários e de outros programas, encontram padrões repetitivos, otimizando-os, e, por fim, automatizando-os da melhor forma possível.
- **Agentes Cognitivos:** São baseados em modelos de organizações sociais, tais como grupos, hierarquias e comércios.
- **Agentes Móveis:** Estes agentes detêm um diferencial em relação aos outros, visto que, para realizarem os seus objetivos, são capazes de migrar na rede em busca da sua meta.

¹³ *Background*: são processos ou programas que executam em segundo plano, e não existe comunicação direta do usuário com o processo durante o seu processamento.

3.2. AGENTES MÓVEIS

Os agentes com características de mobilidade são objetos ou programas que permitem transitar entre nodos¹⁴ de uma rede, executando o seu código e desempenhando a tarefa que foi a ele designada. Podem se mover em uma rede de computadores heterogêneos.

A idéia de execução remota dos agentes é antiga, visto que a submissão de esquemas, os *batch jobs*¹⁵, permite o processamento em lote enviado aos computadores de grande porte. O código se move para a máquina, e a execução passa a ser local.

A definição de agentes móveis segue as mesmas problemáticas referentes aos agentes, visto que ainda não há uma definição precisa para eles. Assim, seguem, abaixo, alguns conceitos:

“Agentes móveis são objetos compostos por código, dados e estado de execução que trafegam entre domínios protegidos” (KATO et al., 2000).

“Um agente móvel é um componente contendo pelo menos uma *thread*¹⁶ de execução, que é habilitado para autonomamente migrar para uma máquina diferente” (PICCO, 2001).

Enquanto os *applets*¹⁷ do Java realizam um *download*¹⁸ do Servidor para o Cliente, e enquanto os *Servlets*¹⁹ realizam um *upload*²⁰ do Cliente para o Servidor, os

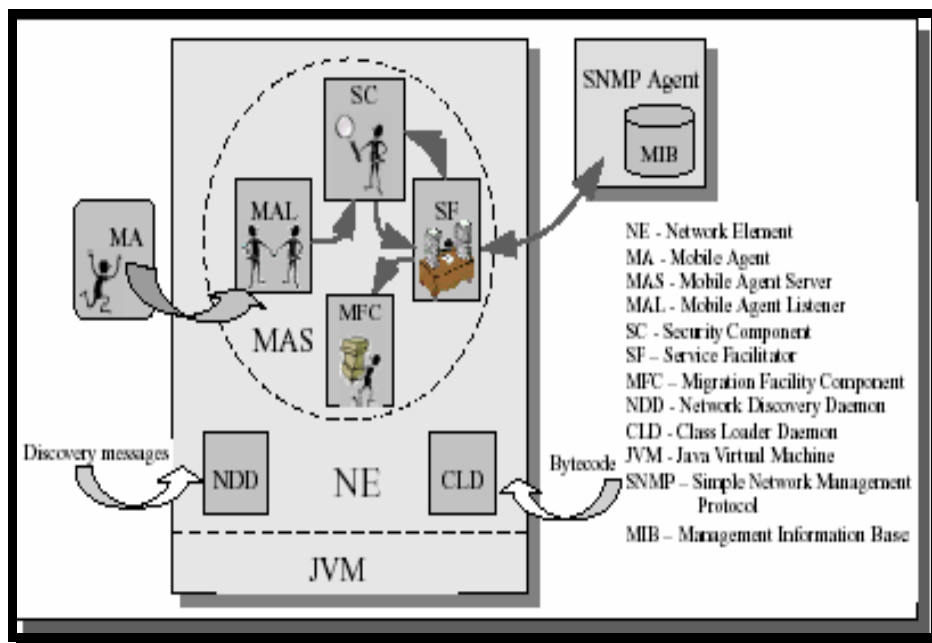
¹⁴ Nodos: São computadores interconectados em uma rede.

¹⁵ *Batch jobs*: São tarefas que executam sem a interação com o usuário.

¹⁶ *Thread*: A unidade básica usada para alocar tempo do processador de um nodo (computador). É um caminho de execução (fluxo de controle) através de um processo.

¹⁷ *Applet*: pequenos programas independentes que são enviados pela *internet* para funcionar em conjunto com os navegadores.

agentes móveis carregam o seu código consigo, com o intuito de interagir com a máquina local, sem ter nenhum envolvimento com a máquina de origem.



Fonte: (OSHIMA, 1999)

Figura 6- Estrutura do Servidor de Agentes Móveis

3.3.VANTAGENS DO USO DE AGENTES MÓVEIS

A tecnologia dos Agentes Móveis permite a obtenção de algumas vantagens em relação às tradicionais abordagens atualmente em voga: a redução do tráfego da rede devido aos agentes despacharem as tarefas para serem executadas nas máquinas de destino; a movimentação do processamento para o local onde os dados se encontram, o que permite uma redução em relação aos dados que transitam na rede. Além disso, há

¹⁸ *Download*: receber e armazenar um programa ou arquivo de dados proveniente de um computador distante, através de canais de comunicações de dados.

¹⁹ *Servlets*: São classes Java, que têm a propriedade especial de serem executadas em um servidor.

²⁰ *Upload*: Transmitir um arquivo usando a rede de computadores para o computador de um outro usuário.

uma diminuição no tempo de atraso da rede, facilitando a manipulação de agentes em relação aos sistemas críticos.

Uma vantagem citada em relação aos agentes é a execução assíncrona e autônoma. Esta vantagem se manifesta principalmente devido à fragilidade das conexões das redes. Se o agente é despachado para iniciar uma tarefa, e logo em seguida é realizada uma desconexão com a rede, o agente continua a sua execução com o intuito de realizar a sua tarefa. Uma vez que o usuário se reconecte, o agente já poderá estar com a resposta pronta para ser enviada ao usuário final. Dessa forma o usuário recebe a resposta das suas requisições, sem ter que recomeçar o processo.

Outra vantagem em relação aos agentes móveis é a sua adaptação dinâmica que lhe permite uma percepção em relação à mudança do ambiente e pode até se multiplicar para realizar uma determinada tarefa.

O encapsulamento²¹ de protocolos é outra vantagem, pois cada máquina, em um sistema distribuído, possui seu próprio código, que é necessário para implementar a transferência de dados. Assim, requisitos de segurança e eficiência demandam mudanças no protocolo que podem ocasionar problemas na manutenção do código. Os agentes móveis podem mover-se para máquinas remotas a fim de estabelecer canais de comunicação baseados em protocolos proprietários.

As redes de computadores são geralmente heterogêneas, tanto na perspectiva de hardware como de software. Os agentes móveis são independentes da máquina e da

²¹ Encapsulamento: Um objeto normalmente contém algumas variáveis de estado internas (ou seja, escondidas), juntamente com alguns procedimentos públicos, denominados métodos, para acessá-las. Os programas que utilizam o objeto são levados (e podem ser forçados) a chamar os métodos para lidar com o estado do objeto. Dessa forma, a pessoa que cria o objeto pode controlar o modo como os programas utilizam as informações dentro do objeto.

rede, sendo dependentes somente de seu ambiente de execução, facilitando a integração de sistemas.

Por outro lado, a habilidade dos agentes de reagirem dinamicamente a situações e eventos desfavoráveis, torna fácil a construção de sistemas distribuídos robustos e tolerantes a falhas. Se uma determinada estação for desligada, todos os agentes em execução na máquina podem ser advertidos para que possam continuar suas tarefas em outra estação da rede.

3.4. LINGUAGENS DE PROGRAMAÇÃO E AGENTES MÓVEIS

A definição da escolha da linguagem para desenvolver um agente terá efeito na arquitetura do agente produzido. As linguagens fornecem um conjunto de funcionalidades para a implementação dos agentes. As características que devem ser observadas na escolha são (BRENNER, 1998):

- 1) Orientação a objetos: os agentes são objetos e a comunicação de agentes ocorre através da invocação de métodos que constituem sua interface pública.
- 2) Independência de plataforma: são usados em computadores heterogêneos e em diferentes sistemas de agentes distribuídos. A linguagem deve oferecer facilidades que permitam esta interação.
- 3) Capacidade de comunicação: para isso, a linguagem deve possibilitar que se implementem componentes orientados à comunicação; através deles é que os agentes comunicam-se entre si e com recursos internos e externos em um ambiente de rede.

- 4) Segurança: deve ser oferecido pela linguagem um alto grau de funcionalidade, a partir dos modelos de segurança específicos da linguagem ou através da integração de modelos externos (protocolos de criptografia, *firewalls*²²).
- 5) Manipulação de código: as aplicações podem requerer que o código do programa seja manipulado em tempo de execução. Para isso, são necessários mecanismos de identificação do código do agente.

Mesmo tendo todas estas características, esta linguagem deve ser reativa, multitarefa e permitir armazenamento de dados persistentes.

Assim, um das linguagens bastante popular na construção de agentes é o Java, que é desenvolvida pela *Sun Microsystems* (SUN, 2000) e é orientada-a-objetos e baseada em redes.

3.5. SISTEMA DE AGENTES

Os sistemas de agentes são compostos por uma plataforma que pode criar, interpretar, executar, transferir ou extinguir agentes. Um sistema de agente permite que se identifique uma autoridade, ou seja, uma pessoa ou organização para a qual o agente foi determinado para executar a sua tarefa. Com a utilização deste sistema de agente, o usuário pode identificar o perfil do agente que o mesmo incorpora. Por exemplo, em se

²² *Firewalls*: São dispositivos constituídos por componentes de *hardware* (roteador capaz de filtrar pacotes) e *software* (*proxy server*), que controlam o fluxo de pacotes entre uma rede local (privada) e a rede pública (*Internet*).

tratando de um sistema de agentes como o *Agllets*²³, imediatamente pode-se abstrair que o mesmo foi implementado pela IBM, suporta o Java como uma linguagem de agentes, utiliza-se de uma classe para representar o itinerário que o agente móvel irá percorrer, além de utilizar a serialização²⁴ através de objetos Java.

A principal função de um sistema de agente é a transferência de um agente de um local para outro. Quando se vai proceder à transferência de um agente móvel de uma máquina para outra, há uma solicitação do local de origem para o local de destino. Antes de a máquina destino receber o agente, ela é responsável por certificar se o agente pode ser por ela interpretado. Em caso positivo, o agente transfere juntamente as classes necessárias, o estado, a autoridade e a segurança inerente ao agente.

Na máquina de origem, ao iniciar a transferência, o ambiente suspende a execução do agente e identifica suas partes a transferir. Em seguida há um processo de serialização do agente e do seu estado, ao mesmo tempo em que o protocolo de transporte é detectado. Há uma autenticação do cliente e sucede-se sua transferência para o destino. Na máquina de destino, para que seja efetuado o recebimento do agente, autentica-se o cliente e inicia-se a desserialização²⁵ do agente. Uma vez desserializado, há uma decodificação da classe e do estado do agente, o qual passa a ser executado e com isso é criada uma instância onde é restaurado o estado do agente, e se processa, então, sua efetiva execução.

²³ *Agllets*: Ambiente de trabalho para a construção de sistemas de agentes móveis desenvolvido pela IBM do Japão. Site: <http://www.aglets.org>

²⁴ Serialização é o armazenamento das informações do processo de execução de um agente, de forma que o mesmo possa ser reconstruído em outro meio.

3.6 PLATAFORMAS

Ultimamente, em se tratando de plataformas para agentes móveis, há uma tendência para a utilização daquelas baseadas em Java devido, principalmente, à sua independência, inerente à linguagem Java. Dentre todas as plataformas em voga no momento, o *Voyager*²⁶ da *ObjectSpace*, o *Aglets* da IBM e o *Concordia*²⁷ da Mitsubishi são as mais utilizadas para o desenvolvimento e implementação dessa nova tecnologia.

3.6.1 AGLETS

A plataforma *Aglets* foi criada em 1996, através de um kit ASDK. Possui uma interface de usuário chamada Tahiti, através da qual pode criar, despachar, monitorar e extinguir agentes. Para a instalação do kit, é necessária a instalação da máquina virtual Java, a princípio na versão 1.1.X.

Esta plataforma contém seis elementos básicos, o *Aglet*, *Proxy*, Contexto, Mensagem, Respostas Futuras e Identificador. O *Aglet* é um objeto Java e também um agente móvel com a capacidade de mover-se de um modo autônomo. O *Proxy* atua como um representante do *Aglet*, protege o agente do acesso direto, os métodos públicos e sua real implementação. A mensagem é um objeto que permite haver uma troca de mensagens que podem ser realizadas em modo síncrono e assíncrono. As respostas

²⁵ Desserialização é o processo inverso da serialização, ou seja, a partir das informações armazenadas, reconstruir o processo de execução de um agente.

²⁶ *Voyager*: <http://www.ncsa.uiuc.edu/STI/ALG/Tools/agent.html>

²⁷ *Concordia*: <http://www.meitca.com/HSL/Projects/Concordia/Welcome.html>

futuras são utilizadas para o envio de mensagens assíncronas. E o identificador permite que cada *Aglet* seja identificado como único, em toda a sua existência.

3.6.2 VOYAGER

A plataforma *Voyager* foi elaborada pela ObjectSpace para o desenvolvimento de aplicações distribuídas. É baseada em Java e possui recursos para agentes móveis, mantendo a capacidade de criar e mover objetos remotamente. Esta plataforma é composta por um ORB²⁸.

Embora essa plataforma possua vários elementos básicos, o principal é o *Voyager* ORB, que tem a capacidade de migrar objetos, garantir a persistência²⁹ dos objetos e sua segurança, fornecendo o ciclo de vida dos agentes. Esta plataforma já está sendo utilizada para ambiente comercial, tendo aplicações em desenvolvimento baseadas na mobilidade do código.

3.6.3 CONCORDIA

A plataforma Concordia permite o desenvolvimento e o gerenciamento de agentes móveis. Foi escrita em Java e desenvolvida pela Mitsubishi Electric com o objetivo de transferir agentes através da mobilidade. Essa plataforma é constituída de pelo menos três partes, Máquina Virtual Java (JVM), Servidor Concordia e pelo menos um agente.

²⁸ ORB: Caminho de ligação dos objetos.

²⁹ Persistência: objeto persistente é aquele que existe após o processo ou *thread*, que lhe criou ter parado de existir.

Há um gerenciador de agente que provê um ambiente para a criação, execução e destruição do agente. Esse gerenciador é responsável também pelo ciclo de vida do agente e possui uma interface para o gerenciador dos agentes. A principal função desse gerenciador é a transmissão do agente, realizada através do Java RMI. Quando da transferência, a integridade do agente é verificada realizando-se uma imagem dele antes de ser copiado efetivamente para a máquina destino. Com isso, garante-se que caso haja algum problema na transmissão, o agente de origem não seja danificado e nem seja perdida a sua referência.

Diante do exposto, vê-se que a tecnologia de agentes móveis está sendo aplicada a vários ambientes e áreas da informática, conseguindo minimizar e aprimorar a utilização de mais recursos disponíveis na rede de computadores.

3.7 CONCLUSÃO

As informações apresentadas também servem como justificativa das escolhas relacionadas à utilização da plataforma e da linguagem Java. Essas descrições referentes às características diferenciam um software normal de um agente, assim abrangendo o limite funcional que um software pode ter em uma rede, já que o agente móvel possui maior flexibilidade e interoperabilidade. Essas mesmas descrições são também importantes para definir os requisitos que devem existir no dispositivo que hospedará um agente móvel. A utilização de uma plataforma como *Aglets* constitui uma forma bastante prática e direta para criação e manipulação de agentes móveis.

4. ESTUDO DE CASO - COMPARAÇÃO DO DESEMPENHO DO AGENTE MÓVEL E CLIENTE/SERVIDOR

4.1 AVALIAÇÃO DE DESEMPENHO

Usa-se a avaliação de desempenho para comparar diferentes alternativas de configurações de sistemas com o objetivo de encontrar aquela que melhor satisfaça as exigências da relação custo/benefício. O objetivo principal na avaliação de um sistema computacional é obter o maior desempenho com um determinado custo.

Avaliar um sistema computacional é considerado uma “arte”, pois avaliações bem sucedidas não podem ser realizadas mecanicamente. Para cada uma delas, requer-se um conhecimento profundo do sistema que está sendo observado e uma escolha minuciosa da técnica de avaliação, carga de trabalho e ferramentas que serão utilizadas.

4.1.1 OBJETIVOS

Para realizar uma avaliação de desempenho deve-se utilizar técnicas e metodologias, que auxiliarão na resolução de problemas, tais como:

- Especificação dos requisitos de desempenho.
- Avaliação de projetos alternativos.
- Comparação entre dois ou mais sistemas.
- Definição de valores de parâmetros ideais.
- Procura de gargalos³⁰ de desempenho.
- Caracterização da carga de um sistema.

³⁰ Gargalo: um fator limitante para um determinado sistema.

4.1.2 METODOLOGIA DE AVALIAÇÃO DE DESEMPENHO

Na escolha de uma metodologia de avaliação de desempenho, deve-se tomar muito cuidado para não incorrer em erros comuns, como: falta de objetivos, objetivos tendenciosos, técnica errada de avaliação e detalhamento demasiado ou inexistente. Jain (1991) propõe uma abordagem sistemática, descrita a seguir.

4.1.2.1 Definição dos Objetivos e do Sistema

Em um projeto de avaliação de desempenho deve-se determinar os objetivos a serem estudados e a definição do que consiste o sistema, assim delineando os seus limites. Esses objetivos podem ser difíceis de precisar, mas são essenciais para a resolução do problema, assim como a definição exata das "fronteiras" do sistema. Isso poderá afetar as métricas de desempenho, bem como as cargas usadas para a comparação.

Para isso, será analisada a comparação entre os modelos Cliente/Servidor e Agentes Móveis utilizados para o gerenciamento de redes. Comparam-se os objetos-gerenciados e objetos-gerente, a fim de determinar, dentre os dois modelos, aquele mais adequado para minimizar o tráfego na rede.

4.1.2.2 Elaboração da Lista de Serviços e Resultados Esperados

Em cada sistema deve-se prover uma lista de serviços, e para cada um deles existe um conjunto de possíveis resultados, desejados, ou não. Como exemplo, há um sistema de base de dados que pode responder a uma consulta correta, incorretamente ou não responder. Uma lista dos serviços e possíveis conseqüências são importantes para selecionar as métricas corretas e a carga de trabalho. Para esse estudo, o resultado

esperado é obter o tempo de latência³¹ para o tráfego de objetos de gerência nos segmentos de redes, para onde será retornado uma variável lida, ou um conjunto de variáveis da MIB, utilizando o protocolo SNMP e também os agentes móveis.

4.1.2.3 Seleção das Métricas

As métricas são critérios para a comparação do desempenho. Não há uma definição-padrão das métricas inseridas no contexto de avaliação de desempenho. Dessa forma, elas dependem do comportamento dos componentes do sistema a ser estudado. São associadas aos três tipos de resultados possíveis de uma solicitação de serviço:

- Solicitação atendida corretamente.
- Solicitação atendida incorretamente.
- Solicitação não atendida.

As métricas mais comuns são:

- Tempo de resposta: é definido como o intervalo de tempo entre a requisição e a resposta fornecida pelo sistema. Esta é a métrica definida no presente estudo de caso.
- *Throughput*: é definido como a taxa (requisições por unidade de tempo) de requisições que podem ser executadas pelo sistema. Cresce à medida que a carga de trabalho aumenta até que se atinja um limite, o qual é chamado de capacidade nominal do sistema.
- Utilização de um recurso: é medida como a fração de tempo em que o recurso esteve ocupado resolvendo o serviço requisitado.

³¹ Latência ou *network transit delay* é o tempo – medido geralmente por milissegundos (ms) necessário para realizar a transferência de um *bit* de informação de um ponto a outro de uma rede. A latência é determinada pela largura de banda da conexão, pelo tempo de propagação da rede e também pela velocidade dos equipamentos envolvidos na transmissão. Normalmente, a latência de uma rede *Ethernet* é inferior a 10 milissegundos, enquanto que a latência de uma conexão através de um modem ultrapassa a 100 milissegundos (SINGHAL; ZYDA, 1999).

- Confiabilidade: é medida pela probabilidade de ocorrência de erros ou pelo tempo entre ocorrências de erros.
- Disponibilidade: é o tempo em que o sistema esteve ou fica disponível para atender às requisições de serviços.
- Eficiência.
- Produtividade.
- Relação custo/desempenho.

No momento em que se está avaliando as métricas, deve-se verificar se elas são globais ou individuais. Quando individuais refletem a utilização do sistema por um único usuário; quando globais refletem a utilização do sistema como um todo. A utilização, a confiabilidade e a disponibilidade são métricas globais, enquanto o tempo de resposta e o *throughput* são métricas que tanto podem ser individuais como globais.

Depois de definidas as métricas, é importante considerar os itens a seguir:

- Baixa variabilidade: reduzir o número de repetições para se conseguir o nível de confiança estatístico desejado.
- Não-redundância: se existir duas métricas que possuem a mesma informação, deve-se, então, estudar somente uma delas. Cabe observar que isso não é sempre tão óbvio.
- Conjunto completo de métricas: todos os resultados possíveis devem ser incluídos no conjunto de métricas de desempenho.

Para este estudo de caso, a métrica que será estudada é o tempo de resposta entre as máquinas gerente e cliente, tanto no modelo Cliente/Servidor quanto no de Agentes Móveis.

4.1.2.4 Elaboração da Lista de Parâmetros

Deve-se fazer uma lista de todos os parâmetros que afetam o desempenho. Essa lista pode ser dividida em dois tipos: parâmetros de sistema (que geralmente não variam de uma instância do sistema para outra) e de carga (que são características das solicitações dos usuários e, portanto bastante variáveis).

Nos parâmetros de sistema, as características são relacionadas ao desempenho. Vejam-se exemplos de parâmetros relacionados ao desempenho de sistemas Cliente/Servidor: protocolos de rede, número máximo de conexões suportadas por um servidor ou número máximo de consultas suportado por um sistema gerenciador de base de dados, dentre outros.

Os parâmetros de carga, aqueles derivados da caracterização da carga de trabalho submetida ao sistema, subdividem-se em dois segmentos:

- Parâmetros de intensidade de carga: é a medida de carga submetida ao sistema, indicada pelo número de trabalhos (requisições, comandos, transações) que disputam os recursos do sistema. Exemplo: número de buscas/dia no servidor *proxy*, número de requisições submetidas ao servidor de arquivos.
- Parâmetros de demanda de carga: são os valores que especificam as necessidades de serviços por cada componente básico sobre cada recurso. Exemplo: tempo de CPU necessário a uma transação em um servidor de base de dados, tempo de transmissão sobre uma LAN de respostas emitidas por um servidor.

Quadro 2 - Critérios para Seleção de uma Técnica de Avaliação

Parâmetros que serão utilizados neste trabalho	
Tamanho Inicial do Agente Móvel (TIAM)	1500 octetos
Tamanho da PDU <i>GetRequest</i> da variável <i>ifInErrors</i> – tamanho da solicitação inicial da requisição da NMS para a máquina que esta sendo gerenciada	42 octetos
Tamanho da PDU <i>GetResponse</i> da variável <i>ifInErrors</i> – tamanho da resposta da máquina que está sendo gerenciada para a NMS	51 octetos
Tempo médio de acesso a MIB para o SNMP por equipamento de rede (TMIBSNMP)	65 ms
Tempo médio de acesso a MIB para o agente móvel por equipamento de rede (TMIBAM)	78 ms

Fonte: (ARANTES, 2001)

4.1.2.5 Seleção dos Fatores para Desempenho

A alteração dos parâmetros é o que pode influenciar com mais intensidade o desempenho do sistema. Os valores que eles podem assumir são chamados de níveis. Para facilitar, é melhor começar com uma quantidade menor de fatores e poucos níveis em cada um deles, e ir aumentando a lista baseada na necessidade. Uma boa maneira de escolher os fatores é usar os parâmetros que mais influenciam no desempenho.

Para a escolha desses parâmetros é importante considerar a economia, as políticas e as limitações tecnológicas que possam existir, bem como as limitações impostas pelo responsável pelas decisões e o tempo disponível para a tomada das mesmas. Dessa forma, aumentam-se as chances de encontrar uma solução aceitável e implementável.

O parâmetro que pode influenciar na comparação entre os dois modelos é a quantidade de variáveis analisadas principalmente entre o SNMP e o Agente Móvel.

4.1.2.6 Seleção da Técnica de Avaliação

Usam-se, como técnicas de avaliação, a simulação, a modelagem analítica e a medição. Há várias considerações que ajudam a decidir qual a técnica a ser usada e estas estão ordenadas na Tabela 3 em ordem decrescente de importância.

Tabela 1 – Critérios para Seleção de uma Técnica de Avaliação

Critério	Mod. Analítica	Simulação	Medição
1. Etapa	Qualquer	Qualquer	“Protótipo Final”
2. Tempo Disponível	Pequeno	Médio	Variável
3. Ferramentas	Analistas	Linguagens Comp.	Instrumentação
4. Precisão	Baixa	Moderada	Variável
5. Equilíbrio de parâmetros	Fácil	Moderado	Difícil
6. Custo	Pequeno	Médio	Alto
7. Aceitabilidade	Baixa	Média	Alta

Fonte: Adaptado de (JAIN, 1991)

Deve-se levar em consideração a fase do ciclo de vida em que o sistema se encontra. As medições somente são possíveis se algo similar ao sistema proposto existir. Para projetos novos, pode-se escolher por modelagem analítica ou simulação.

Outra consideração é o tempo disponível para se fazer a avaliação. Em quase todos os casos os resultados são requeridos para “ontem”. Se isso acontecer, a modelagem analítica é provavelmente a única escolha. Pois simulações tomam bastante tempo.

Há, ainda, o nível de precisão desejado que é outra consideração importante. Vê-se que a modelagem analítica requer muitas simplificações e suposições. As simulações podem ter mais detalhes e menos suposições e ainda estar mais frequentemente próximas da realidade. Já as medições, apesar de soar como mais próximas da realidade, não podem gerar resultados precisos, simplesmente porque parâmetros, tais como configurações do sistema, tipos de cargas de trabalho e tempo de medição, podem ser únicos para o experimento.

O principal objetivo de todo estudo de desempenho é também comparar diferentes alternativas para encontrar um valor ótimo. Em modelos analíticos, tem-se, geralmente, a melhor visão sobre o efeito da interação entre os parâmetros. Utilizando a simulação, é possível buscar a melhor combinação de valores para esses parâmetros. Frequentemente não fica clara a relação de compensação existente entre os parâmetros. A medição é uma técnica menos desejável nesse sentido.

E o custo destinado ao projeto é bastante importante. Sabe-se que a medição exige instrumentos e tempos reais, além de ser a mais cara das três técnicas. Simulação é uma boa alternativa pela facilidade de alteração de configurações.

Segundo Jain (1991), deve-se escolher dentre as três técnicas a ser usadas, dependendo do tempo e dos recursos disponíveis para a resolução do problema e do nível de acuidade desejado. Mas o ideal é que, independentemente da técnica escolhida, possa-se usar as outras para corroborar o resultado.

A técnica utilizada para este estudo de caso é a modelagem analítica.

4.1.2.7 Seleção da Carga

Na carga, considera-se uma lista de solicitações de serviço ao sistema, sobre cujos usos reais deve-se refletir.. A seleção da carga deve levar em consideração o seguinte:

- Todos os serviços que afetem o desempenho devem ser solicitados.
- O nível e o detalhe devem refletir a realidade, ou seja, se o sistema recebe uma grande variedade de solicitações, o uso de apenas uma como carga não é representativa.
- Os parâmetros como taxa de chegada de solicitações, uso de recursos, seqüência e quantidade de uso devem também estar o mais próximo possível do uso real.
- A carga deve representar a utilização atual do sistema.

Deve-se analisar os valores máximos e médios de carga se for considerado todo o período de tempo que são amostrados. Assim, pode-se ter a noção da influência de períodos críticos para um sistema *on-line*³² se considerados os horários de pico.

Para este estudo será utilizada carga diferente, nas solicitações dos gerentes aos clientes, ou seja, várias solicitações diferentes.

4.1.2.8 Planejamento dos Experimentos

Quando de posse da lista de fatores e seus níveis, deve-se decidir uma seqüência de experimentos de modo a obter o máximo de informações com o mínimo de esforço. A prática recomenda é dividir os experimentos em duas fases:

- Número de fatores alto e o número de níveis mais baixos.
- Número de fatores reduzido e número dos níveis mais significativos aumentados.

4.1.2.9 Análise e Interpretação dos Dados

Os resultados das avaliações e simulações são quantidades sem método e podem ser diferentes a cada experimento repetido. Para a comparação de alternativas é necessário ter que considerar a variabilidade dos resultados. Uma simples comparação de médias pode levar a resultados insatisfatórios. É importante sabermos que a interpretação de resultados de uma análise é a chave da arte de analisar. Devemos entender que análises somente produzem resultados, e não conclusões. Tais resultados servem de base para que os analistas ou pessoas ligadas à tomada de decisão possam extrair conclusões.

Cabe, aqui, que salientar que este modelo analítico assume que os enlaces e os nós da rede não possuem carga e não possuem perda. Assim, o tempo de processamento na

³² *On-Line*: Uma estação de trabalho que está conectada ao servidor em uma rede local, devolvendo a resposta de um serviço após a solicitação do mesmo.

camada de aplicação e os detalhes do protocolo SNMP, como segmentação de pacotes e protocolo de transporte, aqui não são considerados. Da mesma forma, o modelo não computa o tempo referente à plataforma do agente móvel.

4.2. DEFINIÇÃO DO MODELO ANALÍTICO

A inteligência do sistema de gerenciamento concentra-se em uma ou mais estações de gerenciamento (*Network Management Station* - NMS) responsáveis por interagir com agentes que armazenam as informações dos diversos recursos da rede. Esses agentes de gerência são entidades computacionais com o propósito de prover uma interface padronizada para o acesso às informações dos recursos onde eles estão localizados. Cada agente armazena os dados do recurso correspondente em uma base de informação de estrutura hierárquica chamada de *Management Information Base* (MIB). A comunicação segue um protocolo de gerência, que especifica o formato do pacote para cada operação.

A mobilidade de código está sendo considerada como uma solução para otimizar a gerência de redes, uma vez que ela oferece um nível tal de flexibilidade, necessária para lidar com problemas de gerência centralizada, dentre outros. Com isso, as funções de gerenciamento devem se descolar para os dados, ao invés de mover os dados para as funções.

No paradigma de agentes móveis, a NMS inicializa apenas uma função e delega a esta um caminho a ser percorrido, de recurso em recurso. O agente móvel pode reter seu estado de execução, independentemente de quantos *hosts*³³ sejam utilizados na execução.

Os agentes, além de aumentar a flexibilidade e facilitar a distribuição da carga, ainda reduzem o tráfego entre a NMS e os recursos monitorados, já que as consultas

³³ *Host*: computador encarregado de funções centralizadas como a disponibilização de arquivos de dados às estações de trabalho que os solicitarem.

realizadas nos recursos são feitas localmente, sem a necessidade de se trocar mensagens pela rede. O agente realiza a compressão semântica de dados mas, apesar disso, não se pode afirmar que este fato implica a otimização do tráfego da rede, pois o tamanho do agente móvel a trafegar na rede é diretamente proporcional ao tamanho das informações coletadas e do número de recursos visitados. Mesmo que o tamanho do agente seja grande o suficiente para desconsiderar as vantagens com a compressão semântica dos dados, existem situações onde, ainda assim, o uso desse paradigma é recomendável.

Sabe-se que os métodos de gerência centralizados podem comprometer o desempenho de gerência da rede, principalmente com o seu crescimento e aumento de complexidade. Mas, sabe-se, também, que o paradigma dos agentes móveis possivelmente se torne uma alternativa para a gerência descentralizada. Assim, este trabalho apresenta uma avaliação entre os dois modelos a fim de verificar a melhor forma de utilizar os agentes para uma melhor performance na rede.

Para poder avaliar o desempenho de uma operação de gerência, a métrica utilizada será o tempo de resposta. Para realizar esta operação de gerência, deve-se obter o valor das variáveis do elemento gerenciado através do SNMP na MIB do recurso. Esta métrica será utilizada para manter uma homogeneidade com Rubinstein (2001) e Arantes (2001), para facilitar a comparação dos resultados obtidos.

Dessa forma, este modelo visa calcular o tempo necessário para a realização de uma determinada tarefa de gerência por cada uma das técnicas em uma topologia genérica de rede. Assim, pretende-se identificar a melhor forma de utilizar os agentes para obter o melhor desempenho na sua utilização. Para tanto, os modelos assumem que os enlaces e os nós da rede não possuem carga e não têm perda. Em relação ao SNMP, deve-se assumir que o tempo de processamento na camada de aplicação e os detalhes do protocolo, como segmentação de pacotes e protocolo de transporte, não serão considerados. No que concerne aos agentes, não serão computados os tempos referentes à plataforma.

Baseado nos estudos feito por Rubinstein (2001), sobre avaliação do desempenho de Agentes Móveis, o tempo de processamento em cada elemento de rede a ser

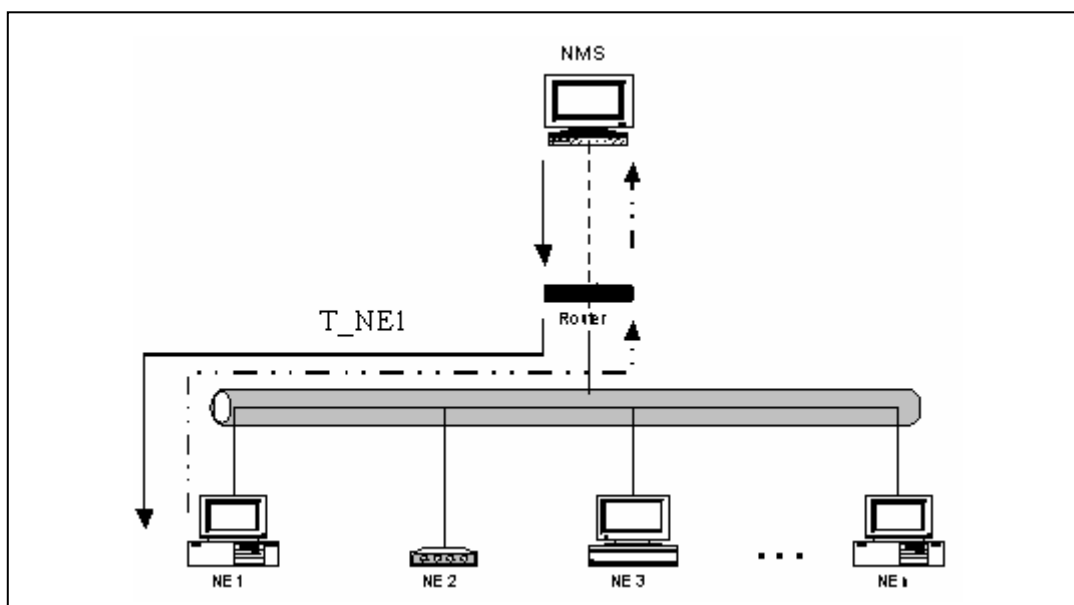
gerenciado, quando o agente SNMP local consulta MIB, não é significativo, e, portanto, o modelo apenas considera o tempo médio de acesso a MIB em ambos os paradigmas.

Tamanhos referentes a pedidos e respostas dizem respeito ao tamanho das PDUs (*GetRequest* e *GetResponse*) do SNMP. Através dessas PDUs, é indicada a tarefa de gerência a ser executada.

4.2.1 MODELO ANALÍTICO PARA O SNMP

O tempo de resposta resultante de uma operação de gerência em um dispositivo dá-se em função dos tempos necessários para a realização das seguintes tarefas:

- 1) trafegar a requisição até o dispositivo a ser monitorado;
- 2) realizar a requisição;
- 3) trafegar a requisição e a resposta até a estação gerenciadora.



Fonte: (ARANTES, 2001)

Figura 7 - Modelo de Gerência Cliente/Servidor

Para se obter o tempo que um pacote levará da máquina gerente – que é a **NMS** – até máquina gerenciada – a **NE₁** – e retornar, deverá ser utilizada a seguinte fórmula:

$$T_{NE_1}(\text{SNMP}) = L + [P / B] + \text{TSNMPMIB} + L + [R / B] \quad (1)$$

Fórmula 1 – Fórmula para calcular a gerência Cliente/Servidor SNMP

Onde se tem:

Quadro 3 – Quadro da descrição das variáveis no modelo SNMP

Descrição das variáveis utilizadas no modelo SNMP
$T_{NE_1}(\text{SNMP})$ = Tempo de resposta de um pacote para ir da NMS até o dispositivo NE_1 e retornar com a resposta.
L = Latência correspondente ao segmento de rede que o pacote está percorrendo.
P = Tamanho do Pedido (<i>GetRequest</i>).
B = Largura de Banda da Rede correspondente ao segmento que o pacote está percorrendo.
R = Tamanho da Resposta (<i>GetResponse</i>).
TSNMPMIB = Tempo de Comunicação entre o agente no equipamento gerenciável (SNMP) com a MIB.

O cálculo do tempo total de resposta de uma variável disparada de **NMS** (Fig. 7), passando por todos os dispositivos e retornando, é dado pela soma da latência de todos os segmentos por onde trafegarem as requisições do pacote SNMP, somando a divisão do tamanho da informação transitada pela largura de banda do segmento:

$$T_{NE_Total}(\text{SNMP}) = T_{NE_1}(\text{SNMP}) * QNE \quad (2)$$

Fórmula 2 – Fórmula para calcular a gerência total da rede usando SNMP

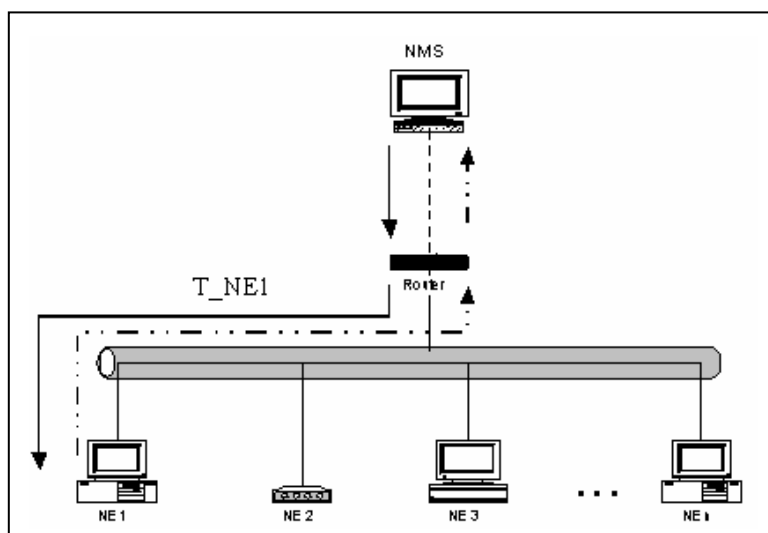
Onde se tem:

Tabela 2 – Tabela para o cálculo dos tempos totais de Gerenciamento no modelo SNMP

Tempos totais para obter todas as respostas das variáveis de gerenciamento
$T_{NE_1}(SNMP)$ = Tempo resultante do gerenciamento do equipamento de rede que está sendo gerenciado.
QNE = Quantidade de equipamentos de rede que estão sendo gerenciados.
$T_{NE_Total}(SNMP)$ = Tempo total para obter as respostas do gerenciamento de todos os equipamentos de rede.

4.2.2 MODELO ANALÍTICO PARA O PARADIGMA AGENTE MÓVEL

Para realizar a operação de gerência, a estação gerenciadora envia o agente para o primeiro recurso a ser gerenciado; este obtém a resposta e segue para o próximo recurso, até percorrer todos os recursos do segmento da rede, e finaliza retornando para a estação gerenciadora.



Fonte: (ARANTES, 2001)

Figura 8 - Modelo de Gerência utilizando Agentes Móveis

Um agente é composto de três partes:

- a parte do código do agente o qual corresponde a um certo algoritmo;
- a *thread* em execução (com uma pilha de execução);
- a parte de dados do agente a qual corresponde aos valores das variáveis globais do agente.

A migração de um agente para um servidor consiste dos seguintes passos:

- a) serialização da construção e dos dados de uma mensagem que inclui o código e os dados do agente;
- b) envio da mensagem para o agente servidor (usou-se uma conexão TCP);
- c) destruição do agente não servidor de origem do agente;
- d) recepção da mensagem no destinatário servidor do agente;
- e) criação de uma nova *thread* para a execução do agente;
- f) criação de uma classe de carregamento associado com o agente; esta classe é inicializada com o código do agente;
- g) desserialização dos dados do agente;
- h) reinício do agente (com o ponto de entrada principal).

A migração do agente é composta dos seguintes componentes:

- Serialização do agente: esta fase corresponde ao passo **a** na descrição da plataforma. A mensagem é construída, incluindo a parte do código e dos dados do agente.

- Transferência do agente: esta fase corresponde aos passos **b**, **d** e **e** na descrição da plataforma. A mensagem é enviada para o servidor destino do agente. Na destinação do *host*, a *thread* é criada e a mensagem é entregue para esta *thread*. O passo **c** não está incluído nesta fase. O agente é destruído somente depois da mensagem ter sido enviada.
- Instalação do agente: esta fase corresponde aos passos **f**, **g** e **h** na descrição da plataforma. Uma nova classe de carregamento é criada para a chegada do agente. O carregamento é inicializado com o código do agente e então os dados do agente são serializados.

O tempo total no paradigma agente móvel pode ser calculado por:

$$T_{AM} = L + (TAM / B) + (L + ((TAM + QNE - 1 * R) / B) + L + ((TAM (QNE * R / B) \quad (3)$$

Fórmula 3 - Modelo Analítico para Agentes Móveis

Onde se tem:

Quadro 4 – Quadro da descrição das variáveis no Modelo Agentes Móveis

Descrição das variáveis utilizadas no modelo Agentes Móveis

T_{AM} = Tempo total do gerenciamento dos equipamentos utilizando agentes móveis.

L = Latência correspondente ao segmento de rede que está sendo percorrido pelo AM.

TAM = Tamanho Inicial do Agente Móvel.

B = Largura de banda da rede correspondente ao segmento que está sendo percorrido pelo AM.

QNE = Quantidade de equipamentos que estão sendo gerenciados.

R = Resposta obtida pelo Agente Móvel junto ao equipamento gerenciado.

Em uma rede local composta por duas máquinas interconectadas a 10Mbps e executando Linux, os resultados obtidos com um agente com um tamanho de 1.475 *bytes*, foram os seguintes:

Tabela 3 – Tabela dos valores em uma rede local utilizando Agentes Móveis

Valores obtidos em uma rede local executando Agente Móvel	
Latência da rede local	0,6 ms
Banda na rede local	842 Kb/s
Invocação do método local do Java	1 μ s
Invocação do método remoto do Java em rede local	3,1 ms
Serialização do agente	3,2 ms
Transferência do agente em rede local	8ms
Instalação do agente com a classe de carregamento do Java	4,3 ms
Instalação do agente com a classe privada de carregamento do Java	23,8 ms

4.2.3 PLATAFORMAS

As plataformas utilizadas foram o *Voyager* e o *Agllets*, sendo para elas desenvolvidos agentes que contivessem as características de mobilidade e comunicação com o ambiente no qual se inseriam, com o intuito de trocar informações na rede. A seleção dessas duas plataformas se deteve por alguns fatores preponderantes em relação ao desenvolvimento do ambiente proposto, tais como a facilidade de documentação encontrada na Internet, a disponibilidade de *download* do *software*, enfim, a facilidade de obtenção de material. Além disso, foi possível o contato com listas de discussões, permitindo uma maior interoperabilidade com usuários ativos em programação. A versão anterior do *Voyager* não tinha tempo de expiração, mas atualmente essa duração limita-se a trinta dias, uma vez que se trata de um produto comercial.

4.2.4 CONFIGURAÇÃO DA MÁQUINA

A máquina onde foi instalado o *Voyager*, juntamente com o *Agllets*, contém duas versões do JDK (*Java Development Kit*) a 1.1.8 para trabalhar com o *Agllets*, e 1.2.2

para integrar com o *Voyager*. Em se tratando de mais uma máquina, é necessário que ambos os sistemas de agentes estejam presentes nas máquinas. Vale ressaltar que não há interoperabilidade entre essas plataformas, mesmo que elas estejam sendo executadas nas mesmas máquinas. Cada agente pertencente a uma plataforma será recebido por sua plataforma específica, além de também ser encaminhado pela sua própria plataforma.

Após a etapa de instalação das plataformas e de configuração das variáveis de ambiente presentes nos documentos fornecidos pelas plataformas, é necessário que se evidencie os locais onde ficarão os arquivos que serão compilados e interpretados pelo Java. No *Aglets*, os arquivos fontes devem ser colocados dentro de um diretório denominado */public*, enquanto que no *Voyager*, os arquivos podem ser colocados dentro de qualquer diretório, o qual será evidenciado no *Classpath* da máquina em questão. Caso alguma classe seja necessária em ambas as plataformas para a execução de um agente, é necessário que haja um *download* das mesmas, para que se possa realizar a instanciação dos agentes e sua posterior execução.

4.2.5 EXECUÇÃO DOS AMBIENTES

Em relação à execução do ambiente, os *Aglets* podem ser inicializados tanto em linha de comando do DOS (*Disk Operating System*), disparando um processo em *background* que fica escutando uma porta³⁴ *default*³⁵ 434 ou pré-estabelecida. O comando que permite a execução do ambiente é: *Agletsd-port 8000*; então, o processo em *background* fica escutando a porta 8000 e esperando a chegada dos agentes móveis, ou daqueles criados estaticamente nesta máquina para se comunicarem com esta porta. Além disso, os *Aglets* podem ser inicializados através de uma interface que gerencia todo o ciclo de vida dos agentes, desde a sua criação até a sua desativação.

³⁴ Porta: São endereços que representam processos em um *host*. Cada um dos processos em execução deve possuir um número único de porta no *host*, o que irá permitir que os dados sejam entregues diretamente ao processo apropriado.

³⁵ *Default*: quando alguma variável vem pré-estabelecida pelo fabricante.

O ambiente do *Voyager*, por sua vez, é inicializado através da linha de comando: *Voyager 8000*, que permite que seja aberta uma porta de comunicação para o recebimento e saída dos agentes para outras máquinas em outras portas.

Foram realizadas duas execuções dos agentes móveis nos dois ambientes propostos.

4.2.6 AGENTE MÓVEL NO AGLETS

O desenvolvimento dos Agentes Móveis com o ambiente *Aglets* ocorreu primeiramente através do uso de uma interface que permite que características sejam herdadas desses Agentes Móveis, trabalhando assim com métodos que fornecem informações sobre o ambiente ou mesmo sobre os agentes.

Com o intuito de utilizar as operações de mobilidade presentes nesta plataforma, é necessário que se importe outra interface relativa aos eventos e às mobilidades dos agentes. A herança dos agentes permite que se obtenham informações relativas aos Agentes Móveis, através de comandos tais como: *AgletInfo agInfo = p.getAgletInfo();*

De posse desse objeto, pode-se obter maiores informações quanto à origem e o endereço em que ele se encontra, além de informações como o horário de sua criação, dentre outras coisas.

```
agInfo.getOrigin();
```

```
agInfo.getAddress();
```

A mobilidade dos agentes é garantida pela própria classe estendida *Aglet*, que permite que um agente se desloque de um local para outro, ou seja, mova-se de uma máquina para outra, comunicando-se através de uma porta. Para que isso seja realizado, torna-se necessário que o seguinte comando seja executado:

```
dispatch(new URL (agInfo.getOrigin()));
```

Neste item, o agente está sendo deslocado para a máquina de origem.

4.2.7 AGENTE MÓVEL NO VOYAGER

O desenvolvimento do agente móvel para a plataforma *Voyager* exige a criação de uma interface de comunicação onde são definidos os métodos e as propriedades que o agente móvel irá executar. Para a execução do agente, é necessário que ele importe também alguns pacotes proprietários do *Voyager*. A criação remota de um agente ocorre através de um objeto *Factory*, como citado abaixo:

```
IAgente agente = (IAgente) Factory.create( "Agente" );
```

É preciso importar outro pacote quando, além de autonomia, o agente necessite mobilidade para migrar de uma máquina para outra. Há uma instanciação da classe *Mobility*, fornecendo estes subsídios ao agente móvel:

```
IMobility mobility = Mobility.of(agente);
```

Para que o agente realize efetivamente a migração, é necessário que seja efetivado o comando de mover-se.

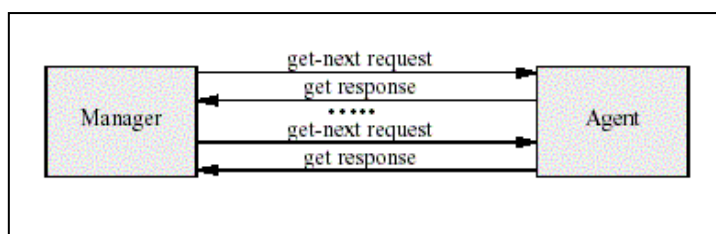
```
mobility.moveTo( "//yemanja:7000" );
```

Assim é que se fornece a mobilidade a determinados objetos caracterizados por autonomia e denominados agentes. Nesse sentido, pode haver uma migração de código para as plataformas dos seus respectivos agentes.

Baseado na literatura e no modelo matemático, foi realizada a comparação entre os dois paradigmas. Trabalhos como os de Rubinstein (2001), Costa (1999) e Arantes (2001) usam a comparação entre esses paradigmas utilizando a consulta da variável *ifInErrors* da MIB; então, para manter esse parâmetro de comparação, a mesma variável foi utilizada: *ifInErrors*. Em Rubinstein (2001) observou-se que o tempo de resposta para o Cliente/Servidor cresce proporcionalmente ao número de elementos gerenciados, pois o tempo para gerenciar cada componente da rede é praticamente o mesmo para todos os elementos. Para o agente móvel, o tempo de resposta cresce mais rapidamente, conforme o número de elementos gerenciados aumenta; isso ocorre devido ao crescimento do agente móvel. Na topologia utilizada, observou-se que o SNMP é bem melhor do que o do agente móvel.

4.3 DEMONSTRAÇÃO DAS VANTAGENS DO AM EM ARQUITETURAS SNMP

Considerando-se uma tabela SNMP que contém mil entradas, o uso da operação *get-next* (SNMPv1) implica que haverá pelo menos um *get-next* para cada linha da tabela.

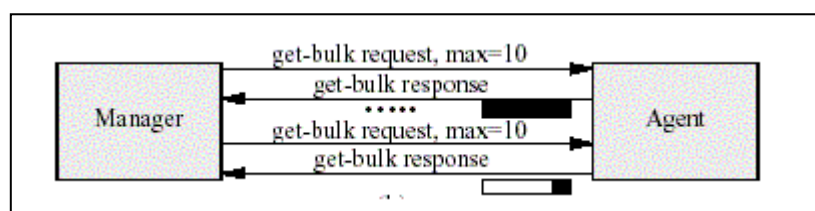


Fonte: (DAMIANOS et al., 2001)

Figura 9 – Modelo SNMPv1

Isso vai causar uma latência enorme, visto que cada operação de *get-next* deve ser terminada antes de outra ter início.

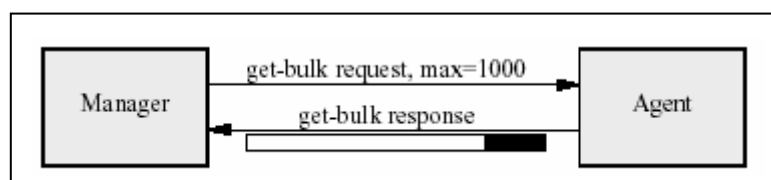
A situação melhora com a introdução da operação *get-bulk*, que permite ao SNMP uma transferência de blocos maiores de dados, especificando o número máximo de valores a serem retornados – *max-repetitions* (SNMPv2 e SNMPv3). O uso de um número pequeno de *max-repetitions* pode resultar numa grande troca de mensagens.



Fonte: (DAMIANOS et al., 2001)

Figura 10 – Modelo SNMPv2 e v3 com *max-repetitions* = 10

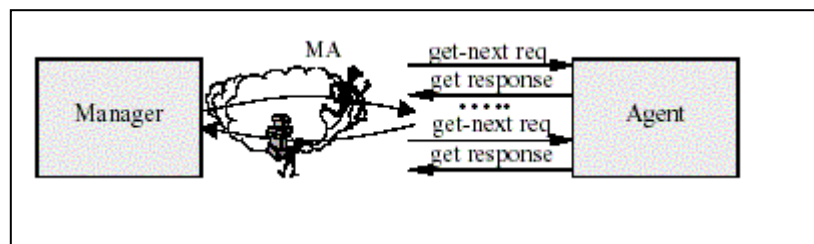
Com a utilização de números muito grandes, o agente pode retornar dados que não pertençam à tabela e, dessa forma, causar um grande congestionamento na rede.



Fonte: (DAMIANOS et al., 2001)

Figura 11 – Modelo SNMPv2 e v3 com *max-repetitions* = 1000

A melhor maneira de terminar com o congestionamento da rede e com a latência é a introdução de agentes móveis. Um objeto AM é despachado pela sua *thread* correspondente e visita um número predeterminado de *hosts*. Quando recebido pelo Servidor de AM, o AM adquire a tabela SNMP através de sucessivos *get-next*.



Fonte: (DAMIANOS et al., 2001)

Figura 12 – Migração do AM e o pedido de *get-next* localmente

O conteúdo da tabela é criptografada³⁶; se desejado, é encapsulada antes de retornar para a NMS.

Os três gráficos que seguem abaixo demonstram a utilização do SNMPv1, SNMPv2, SNMPv3 e o Agente Móvel para realizar a busca de uma ou cinco variáveis. (Uma variável – SNMPv1-1, SNMPv2, v3-1 e Agente-1), (cinco variáveis – SNMPv1-5, SNMPv2,v3-5 e Agente-5) para as seguintes variáveis da MIB: *ifInOctets*, *ifOutOctets*, *ifInError*, *ifInDiscards*, *ifOutDiscards*. A largura de banda é de 2 Mbps e a latência é de 120 ms.

³⁶ Criptografada: é a forma de codificar o código, a fim de obter segurança.

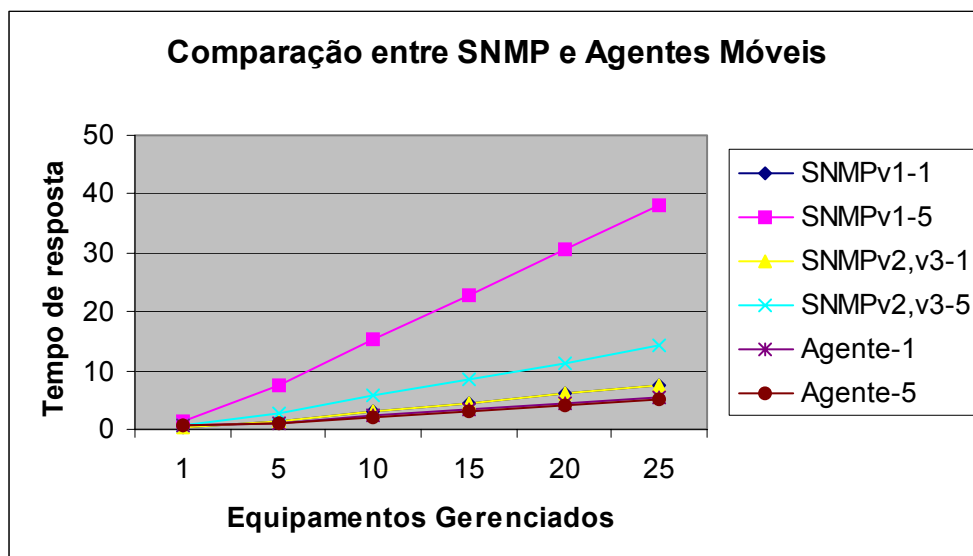


Gráfico 1 - Tempo de resposta para o Agente Móvel e SNMP – Latência 120 ms

Pode-se observar que tanto os agentes móveis quanto o SNMP estão muito próximos em termos de valores. Pode-se, assim, concluir que, para o gerenciamento de uma única variável, o agente tem um ganho pouco significativo em relação ao SNMP. Então, a escolha referente à ferramenta de gerenciamento não necessita estar necessariamente vinculada a um dos modelos; neste caso, a escolha depende muito dos objetivos no gerenciamento. Mas, por outro lado, quando a quantidade de variáveis aumenta, pode-se observar que existe uma diferença significativa entre os dois paradigmas.

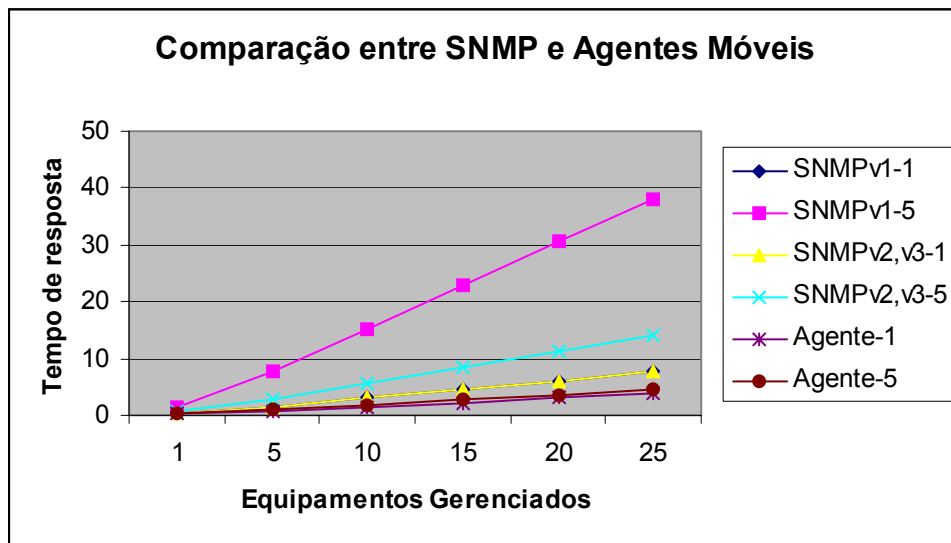


Gráfico 2 - Tempo de resposta para o Agente Móvel e SNMP – Latência 60ms

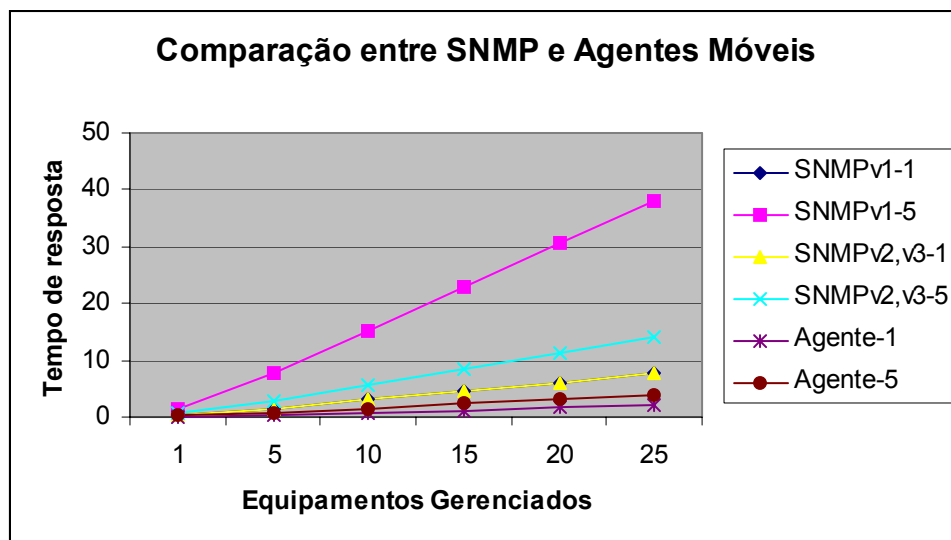


Gráfico 3 - Tempo de resposta para o Agente Móvel e SNMP – Latência 1ms

Pode-se observar que, nos Gráficos 2 e 3, quando a latência da rede é alterada, o desempenho do agente móvel em comparação ao agente SNMP passa a ter maiores ganhos: o desempenho do agente independe da latência da rede, tanto que seu desempenho permanece quase que inalterado; entretanto, para o SNMP, as diferentes latências criam interferências, o que demonstra que não possui ganhos significativos. As três figuras abaixo demonstram a utilização do SNMPv1, SNMPv2, SNMPv3 e o

Agente Móvel para realizar a busca de 5 variáveis tais como: *ifInOctets*, *ifOutOctets*, *ifInError*, *ifInDiscards*, *ifOutDiscards*. Todas foram utilizadas para realizar o gerenciamento da rede, cada uma com 32 bits. A largura de banda é de 2 Mbps.

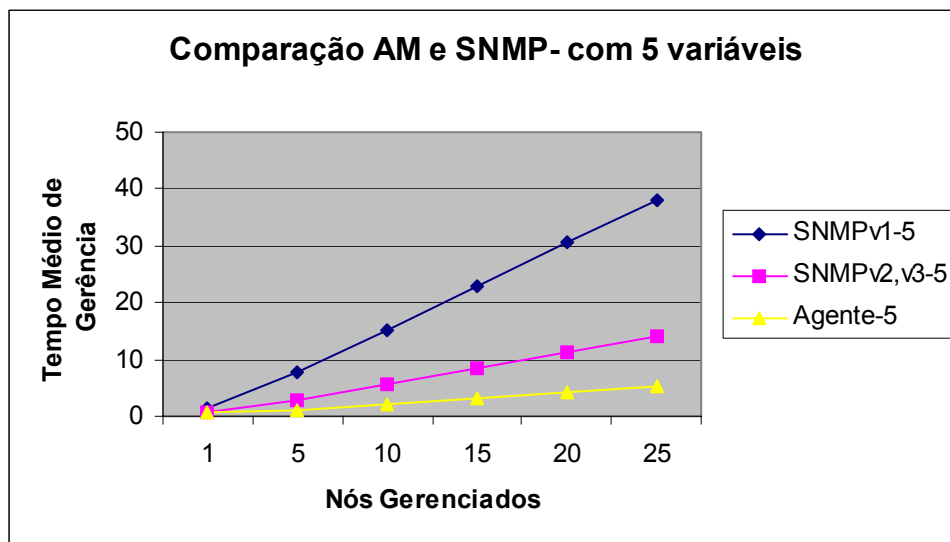


Gráfico 4 - Tempo de resposta para o Agente Móvel e SNMP – Latência 120ms

Pode-se observar que, quando utilizado um maior número de variáveis, o tempo de resposta para o SNMP cresce proporcionalmente, enquanto para o agente móvel permanece quase que constante. Assim, pode-se observar que o uso de agentes móveis pode trazer um aumento significativo na performance quando a quantidade de dados tender a aumentar. Dessa forma, fica evidente que o uso de agentes móveis permite minimizar as comunicações entre um cliente e o servidor.

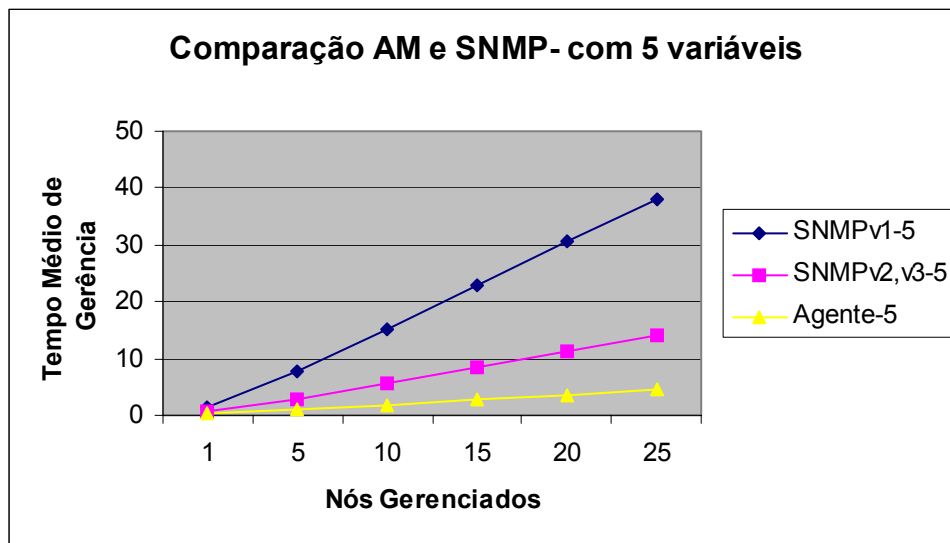


Gráfico 5 - Tempo de resposta para o Agente Móvel e SNMP – Latência 60ms

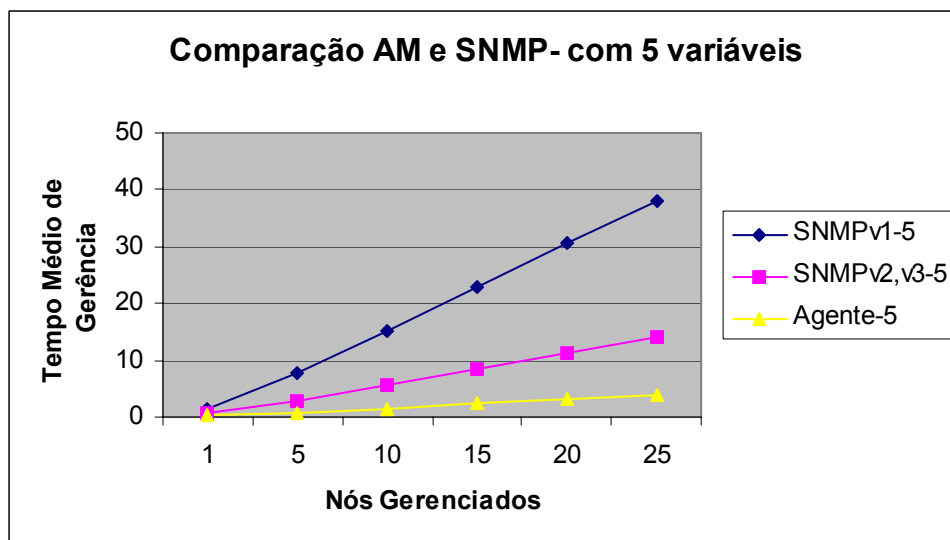


Gráfico 6 - Tempo de resposta para o Agente Móvel e SNMP – Latência 1ms

Comparando todas as figuras sobre a avaliação do desempenho, tanto do agente móvel quanto do SNMP, é possível concluir que a utilização do agente móvel em ferramentas de gerenciamento de redes terá um ganho significativo de desempenho. Para tanto, outros fatores devem ser levados em conta, como o impacto do crescimento do agente na rede, dentre outros.

Sabe-se que há, praticamente, um consenso, entre muitos pesquisadores, de que o uso de AM em aplicações envolvendo a gerência de redes deva ser uma solução a complementar o paradigma Cliente/Servidor, e não uma alternativa ao modelo (BALDI, 1997). Todavia, entende-se que o uso de AM na redução do tráfego deve ser aplicado nas situações em que as tarefas podem ser executadas de forma autônoma, tais como tarefas de monitoração e ajustes de controle. Para fins de diferencial, este trabalho também analisou a questão que envolve a quantidade de variáveis a serem gerenciadas, e demonstrou que quanto maior o número de variáveis mais diretamente o uso de AM influencia o desempenho das ferramentas de gerenciamento.

Analisando-se trabalhos anteriores sobre o assunto (RUBINSTEIN, 2001; COSTA, 1999; ARANTES, 2001) encontraram-se algumas conclusões, tais como:

O efeito da latência demonstra que os AMs sofrem menor influência que o modelo SNMP; a banda passante do enlace, em que os resultados foram apurados, demonstra que os tempos de respostas crescem igualmente em cada um dos protótipos, o que reforça a informação de que os modelos sofrem influência da banda passante. Entretanto, analisando todos os gráficos aqui produzidos, pode-se observar que os AMs podem ter um desempenho melhor em relação ao modelo SNMP, visto que o aumento que envolve a manipulação de dados favorece o uso dos AMs, pois passam a ter melhores resultados comparativamente ao modelo SNMP. Contudo, o que se tem percebido é que os dois modelos podem ser utilizados em conjunto, explorando a melhor maneira de utilização e integração dos modelos a fim de obter melhores resultados no uso de ferramentas de gerenciamento de redes.

4.4 ANÁLISE DO FATOR COMPRESSÃO

Segundo Sahuguet (1997) e Ismail et al. (2001), como existe uma grande troca de documentos entre nós gerenciados, quando o agente estiver muito grande, para otimizar a transferência, pode-se usar a compressão de dados.

Existem muitos algoritmos diferentes de compressão que podem ser usados. O paradigma de agentes móveis oferece uma elegante resposta para esse problema de

compressão. O cliente pode enviar um agente para o servidor. Este agente realiza uma busca localmente e obtém o documento, comprime-o, e retorna para o cliente com o documento comprimido. No lado do cliente, o agente descomprime o documento e entrega-o para o cliente. A principal motivação para analisar este aspecto é a eficiência na transferência do documento.

Para realizar este experimento assumiu-se bw como banda entre duas máquinas. Supôs-se o documento comprimido num fator de fc (se S é o tamanho inicial do documento, então a compressão do documento tem um tamanho de $fc * S$), e o tempo de compressão e descompressão, respectivamente, tc e td . Então, foram identificadas as condições da rede para transferir um documento comprimido em uma aplicação.

O tempo de transferência de um documento comprimido é: $tc + fc*S/bw + td$. Conseqüentemente, o benefício da compressão é: $tc + fc*S/bw + td < S/bw$. Essa implicação é importante para redes com banda entre máquinas desta maneira:

$$bw < S*(1-fc) / (tc + td) \quad (4)$$

Fórmula 4 - Teste para verificar se o documento deverá ser compactado

Onde se tem:

Quadro 5 – Fatores utilizados para a compressão e descompressão do Agente Móvel

Fatores de compressão e descompressão do Agente Móvel	
Bw	Banda entre duas máquinas
Fc	Fator de compressão
S	Tamanho inicial do documento
$fc * S$	Compressão do documento
Tc	Tempo de compressão
Td	Tempo de descompressão

4.4.1 EXPERIMENTO

Foram utilizados documentos com tamanhos entre 100Kb e 2000Kb. A ferramenta para compactar foi o gzip que é acessível à classe Java. A largura de banda utilizada foi de 842 Kb/s. Para cada tamanho do documento, foram mensurados o fator de compressão e os tempos de compressão e descompressão. Substituindo tais valores, obteve-se um benefício da compressão igual a:

- $bw < 74$ Kb/s para um documento de tamanho de 100 Kb,
- $bw < 164$ Kb/s para um documento de tamanho de 500 Kb.

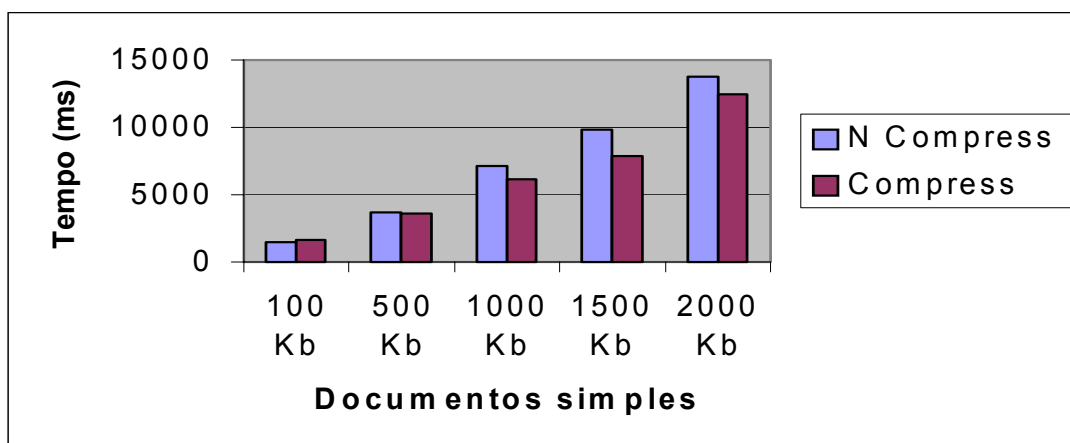


Gráfico 7 - Comparação entre tempos de transferência de documentos sem compressão e com compressão com Agentes Móveis

Pode-se observar, então, que uma das alternativas para melhorar o desempenho dos Agentes Móveis está associada ao fator de compressão que, a partir de uma determinada quantidade de dados, pode diminuir o tempo de transferência dos resultados obtidos; deve-se, entretanto, computar os tempos de compactação e descompactação nos modelos analíticos analisados.

5. CONCLUSÃO

Para analisar os paradigmas de Agentes Móveis e Cliente/Servidor, deve-se, antes, levar em conta as diferenças existentes entre cada um e, dessa forma, explorar as respectivas potencialidades.

A avaliação de desempenho de sistemas computacionais exige, do analista, conhecimentos na área específica de atuação além de várias técnicas de análise, as quais incluem, obrigatoriamente, alguns conhecimentos de estatística, probabilidade, projetos de experimentos, simulação e teoria das filas. Para tanto, o analista deve compreender a terminologia que afeta a análise de desempenho e, corretamente, deve saber projetar experimentos, usar modelos de filas simples ou simulação, além de saber interpretar seus resultados.

Tanto para gerência de redes como para planejamento de capacidade há a necessidade de ferramentas de coleta e monitoração do sistema.

Os agentes móveis fornecem melhores soluções no que diz respeito aos problemas de eficiência e tolerância a faltas, dada a enorme quantidade de dados que tem de ser transferida no âmbito das abordagens tradicionais. A natureza dos agentes móveis não induz um modelo de mobilidade geral para as aplicações; tal modelo deve ser escolhido em função da análise dos requisitos de cada aplicação específica. Para isso, deve-se observar a utilização dos agentes móveis com outros paradigmas, de forma que venham a cooperar entre si, obtendo-se, assim, um melhor aproveitamento das funcionalidades de cada um.

O desenvolvimento de agentes móveis utilizando a tecnologia Java é facilitado pela portabilidade da arquitetura de suporte, a *Java Virtual Machine*.

Os agentes móveis são atualmente considerados os sucessores da tecnologia de comunicação Cliente/Servidor (em particular o RPC), sendo uma eficiente proposta para o futuro.

Vê-se que a utilização da compressão traz benefícios para os agentes, para a troca de informação entre aplicações que utilizem pacotes enormes. Assim, deve-se analisar

quando utilizar esta técnica para se beneficiar de seu uso, e não apenas aumentando o tempo de resposta da aplicação.

Diante dessas análises, pode-se observar que o uso de agentes móveis pode trazer um aumento significativo na performance quando há migração e permite minimizar as comunicações entre um cliente e o servidor. Pode-se, então, dizer que um dos grandes benefícios dos agentes móveis é minimizar a iteração entre o cliente e o servidor na realização das aplicações.

5.1 TRABALHOS FUTUROS

Este trabalho demonstrou algumas das potencialidades do paradigma agentes móveis em relação ao paradigma Cliente/Servidor. Como proposta de outros trabalhos, sugere-se que outras métricas sejam analisadas criando-se, assim, melhor suporte à definição do uso de agentes móveis e SNMP. Outra pergunta que surge é quanto ao impacto que o aumento de tamanho do agente pode causar na rede. Uma forma de analisar os dois paradigmas está baseada na definição de cada tipo de aplicação. Assim, podem ser realizadas comparações e, conseqüentemente, decidir como utilizar o melhor de cada um deles, beneficiando as aplicações.

6. REFERÊNCIAS BIBLIOGRÁFICAS

AdventNetSNMPv3 API. Dezembro, 1999. Disponível em:
<<http://www.adventnet.com>>. Acesso em: 18 ago. 2001.

Aglets Plataforma. Dezembro, 1999. Disponível em: <<http://www.trl.ibm.co.jp/aglets/>>.
Acesso em: 18 ago. 2001.

ARANTES, J. A.. Modelo Analítico para Avaliar Plataformas Cliente/Servidor e Agentes Móveis Aplicado à Gerência de Redes. Florianópolis, 2001. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, 2001.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 6023: Informação e documentação: referências: elaboração. Rio de Janeiro, 2002.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 14724: Informação e documentação: trabalhos acadêmicos: apresentação. Rio de Janeiro, 2002.

BALDI, M.; GAI, S.; PICCO, G. P. Exploiting Code Mobility in Decentralized and Flexible Network Management. INTERNATIONAL WORKSHOP ON MOBILE AGENTS, THE FIRST. Proceedings... Berlin, Germany, p.13-26, 1997.

BALDI, M.; PICCO, G. P. Evaluating the Tradeoffs of Mobile Code Design Paradigm in Network Management Applications. INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING (ICSE), 20TH, Japan, p. 146-155, Apr 1998.

BIESZCZAD, A.; WHITE, T.; PARUEK, B. Mobile Agents for Network Management. IEEE Communications Surveys, Sep. 1998.

BRENNER, W.; ZARNEKOW, R.; WITTIG, H.. Intelligent Software Agents. Springer. Berlin, Germany, 1998.

CHESS, D.; HARRISON, C.; KERSHENBAUM, A. Mobile Agents: Are They a Good Idea? New York: IBM Research Division, T.J. Watson Research Center, Yorktown Heights, Mar. 1995.

- COSTA, T. F. S. Avaliação Analítica do Uso de Agentes Móveis na Gerência de Redes. Florianópolis, 1999. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, 1999.
- DAMIANOS, G.; DOMINIC, G.; MOHAMMED G., O'MAHONY M. Advanced Network Monitoring Applications Based on Mobile/Intelligent Agent Technology. Colchester, CO4 3SQ, U.K: University of Essex, 2001.
- EVANS, E; ROGERS, D. Using java applets and CORBA for multi-user distributed applications. IEEE Internet Computing, May/June. 1995. Disponível em: <<http://computer.org/internet/>>. Acesso em: 15 Jan. 2001.
- GAVALAS, D.; GREENWOOD, D.; GHANBARI, M.; O'MAHONY, M. Advanced Network Monitoring Applications Based on Mobile/Intelligent Agent Technology. Computer Communication Journal, v. 23, n. 8, p. 720-730. Apr. 2000.
- GRAY, R. S.; CYBEKO, G.; KOTZ, D.; RUS, D. Mobile agents: Motivations and State of the Art. Relatório Técnico (TR2000-365). Department of Computer Science, Dartmouth College. Hanover, NH, USA, 2000. Este relatório técnico é um capítulo do livro Handbook of Agent Technology, AAAI/MIT Press. 2000. Em edição.
- IBM Corp. ATM-SWITCHING-NEDE-MIB v.4.1.2. 1999.
- ISMAIL, L.; HAGIMONT, D.; MOSSIÈRE, E J. Evaluation of the Mobile Agents Technology: Comparison with Client/Server Paradigm. SIRAC Project (IMAG-INRIA), 2001.
- JAIN, R. The Art of Computer Systems Performance Analysis. New York: Ed. John Wiley & Sons, 1991.
- Java Documents and API. Disponível em: <<http://java.sun.com>>. Acesso em: 18 ago. 2001.
- KATO, K.; MATSUBARA, K; SOMEYA, Y.; ITABASHI, K.; MORIYAMA, Y.; YOSHIDA, M. Mobile Substrate: Experiences of Middleware-Layer Object Mobility. ECOOP - WORKSHOP ON MOBILE OBJECT SYSTEM: OPERATING

- SYSTEM SUPPORT, SECURITY AND PROGRAMMING LANGUAGES, 6th. Tsukuba, Japan. 2000.
- LANGE, D.; OSHIMA, M. Seven Good Reasons for Mobile Agent. *Communications of the ACM*, v.42, n.3, p. 88-89. Mar. 1999.
- MÃES, P. Agents that Reduce Work and Information Overload. *Communications of the ACM*. July. 1994.
- MAURO, D. R; SCHIMIDT, K. J. *SNMP Essencial*. Tradução de Teresa Cristina Feliz de Souza. Rio de Janeiro: Ed. Campus, 2001.
- MULLANEY, P. Overview of a Web-Based Agent - *The Simple Times*, v.4, n. 3, p. 11-16, July 1996.. Disponível em: <<http://www.simple-times.org/pub/simple-times/issues/4-3.html>>. Acesso em: 18 Ago. 2001.
- OLIVEIRA, D. V. *Mobilidade em Gerência de Redes SNMP*. Florianópolis, 2000. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação, Universidade Federal de Santa Catarina, 2000.
- OSHIMA, M.; LANGE, D. *Programming and Deploying Java Mobile Agents with Aglets*. Addison Wesley. 1999.
- PICCO, G. P. Mobile Agents: State of the Art and Research Opportunities. Cooperative Information Agents V. INTERNATIONAL WORKSHOP, 5th. Modena, Italy, v.2182, p. 247, Sep. 2001.
- PULIAFITO, A.; TOMARCHIO, O. Advanced Network Management Functionalities through the use of Mobile Software Agents. INTERNATIONAL WORKSHOP ON INTELLIGENT AGENTS FOR TELECOMMUNICATION APPLICATIONS (IATA'99), 3rd. Stockholm, Sweden, Ago.1999.
- RUBINSTEIN, M. G. *Avaliação de Desempenho de Agentes Móveis no Gerenciamento de Rede*. Rio de Janeiro, 2001. Tese (Doutorado em Engenharia Elétrica) – Programa de Pós-Graduação em Engenharia Elétrica (COPPE), Universidade Federal do Rio de Janeiro, 2001.

SAHUGUET, A. About Agents and Databases. May 1997.

SILVA, J. L. C.; SILVA, A. N.; SOUZA J. N. Modelos de Implementação de Gerência de Redes baseados na Arquitetura de Redes Ativas e no Conceito de Delegação de Autoridade. Universidade Federal de Pernambuco e Universidade Federal do Ceará. Artigo submetido ao Simpósio Brasileiro de Redes de Computadores, março. 2000.

SIMÕES, P.; SILVA, L.M.; FERNANDES, F.B. Integrating SNMP into a Mobile Agent Infrastructure. IFIP/IEEE INTERNATIONAL WORKSHOP ON DISTRIBUID SYSTEMS: OPERATIONS & MANAGEMENT (DSOM'99), 10th. Proceedings... Zurique, Suisse. Oct. 1999.

SINGHAL, S.; ZYDA, M. Networked virtual environments: design and implementation. Reading MA: Addison-Wesley, 1999. ISBN 0-201-32557-8.

STALLINGS, W. SNMP, SNMPv2, SNMPv3 and RMON 1 and 2. Addison-Wesley, 1999. ISBN 0201485346.

SUN Microsystems Inc. Java Dynamic Management Kit 4.2. – Tools Reference. Part-number 806-6633. Dezembro/2000. Disponível em: <http://docs.sun.com>.

TEIXEIRA JÚNIOR, J. H., et al. Redes de Computadores. Serviços, Administração e Segurança. São Paulo: Ed. Makron Books, 1999.

TENNENHOUSE, D. L.; WETHERALL, D. J. Towards an Active Network Architecture. Laboratório de Ciência da Computação, Instituto de Tecnologia de Massachusetts – MIT. 1996. Disponível em: <<http://www.sce.carleton.ca/netmanage/activeNetworks/mmcn96.html>>. Acesso em: 19 mar. 2001.

ZAPF, M.; HERRMANN, K.; GEIHS, K.. Decentralized SNMP Management with Mobile Agents. IFIP/IEEE IM CONFERENCE ON NETWORK MANAGEMENT, VI. Frankfurt, Germany, 1999.