Agentes Inteligentes para o Gerenciamento de Redes Locais ou Remotas

Rosfran Lins Borges Evandro de Barros Costa

Departamento de Tecnologia da Informação (TCI/CCEN)
Universidade Federal de Alagoas (UFAL)
{rlb, evandro}@dcc.ufal.br ou ebc@fapeal.br

Resumo

É conhecida a importância das redes de computadores na vida das pessoas nos dias de hoje. As possibilidades de aplicações dessa tecnologia são infindáveis, mas tudo isso é possível porque existe um conjunto de regras que regem tais comunicações. E legislando e mantendo essas regras está o gerente de redes. Trata-se de um profissional que lida no seu cotidiano com tarefas altamente estafantes. Com a intenção de aliviar a alta carga de trabalho do gerente de redes é que são desenvolvidas ferramentas que visam apoiar a sua tarefa. No presente trabalho propõe-se uma ferramenta de gerência de redes, aplicando conceitos modernos para tomadas automáticas de decisões, fazendo uso de Agentes Inteligentes e Sistemas Multi-agentes. Com essa abordagem, o papel do gerente de redes é bem mais humano: ele fica encarregado de submeter ordens e regras pré-definidas a um agente, que as utiliza no momento oportuno, evitando os erros comuns a essa área e a atuação do gerente em tarefas repetitivas.

Palavras-chave: Inteligência Artificial; Redes de Computadores; Tomada Automática de Decisões

1. Introdução

O gerente de redes de computadores precisa, não raramente, estar executando procedimentos simples e rotineiros, para solucionar problemas que ocorram em uma rede. Além da necessidade de que o gerente esteja quase sempre presente para resolver problemas simples, e que poderiam ser resolvidos de uma maneira previsível. Nessas situações, a utilização de um sistema para tomada automática de decisões, através da aplicação de alguns conceitos e técnicas presentes em IA, é uma boa alternativa e que vai, com o passar dos anos, se tornando mais próxima da realidade. Para o auxílio nessas soluções existem tecnologias eficientes, como a abordagem baseada em agentes inteligentes.

O presente trabalho tem por objetivo apresentar uma ferramenta de gerência de redes de computadores, usando diversos conceitos de Inteligência Artificial e Sistemas Distribuídos. As técnicas usadas foram agentes inteligentes, sistemas especialistas multi-agentes, bases de conhecimento compartilhado e tecnologias de sistemas distribuídos. Não existe a pretensão de tornar o sistema competitivo com as outras ferramentas produzidas comercialmente. O desenvolvimento do projeto visa apenas demonstrar a aplicação da Inteligência Artificial na solução de problemas comuns ao ambiente de gerência de redes.

2. Problematização da Tarefa do Gerente de Redes

A tarefa do gerente de redes é freqüentemente composta por atividades repetitivas e rotineiras. Tais tarefas são, por esse motivo, sujeitas a erros e a soluções pouco otimizadas. Um

exemplo de tarefa repetitiva é a manutenção das permissões em diretórios, no espaço do sistema de arquivos destinado aos usuários. Entre as pessoas que já tiveram experiência na área de administração ou gerência de sistemas, sabe-se que esse é um problema simples e bastante comum, mas que exige uma atenção constante.

Há inclusive a necessidade de que o gerente esteja quase sempre presente para resolver problemas simples, e que poderiam ser resolvidos de uma maneira previsível. É comum, por exemplo, um gerente de uma grande rede ser interrompido durante um final de semana, por exemplo, para remediar a dificuldade que determinado grupo de usuários vem tendo em consultar páginas de hipertexto (HTML), através de seus *browsers*, conectados remotamente à rede local onde o gerente é responsável. Tal problema com certeza exigirá que o gerente se desloque para o ambiente onde se localiza a rede local, para permitir que os usuários do sistema possam dar continuidade ao seu trabalho, não importando a que horas os funcionários da empresa necessitem desse serviço.

E se fosse permitido ao gerente de redes comandar remotamente uma espécie de robô, que perceba e resolva automaticamente os problemas que ocorrem na rede, baseado num conjunto de ordens preestabelecidas pelo próprio gerente? Ele poderia colocar um "robô" desse em cada uma das máquinas que ele quisesse gerenciar, de forma a automatizar grande parte das suas tarefas. Com base nesse anseio, a utilização de um sistema para tomada automática de decisões, através da aplicação de alguns conceitos e técnicas presentes em IA, é uma alternativa viável, que vai, com o passar dos anos, se tornando mais próxima da realidade. Para o auxílio nessas soluções existem tecnologias eficientes, como a abordagem baseada em multi-agentes inteligentes, utilizadas com intensidade no sistema desenvolvido.

3. Apresentação do Sistema Proposto

O sistema objetiva dar ao gerente um modelo de gerência unificado numa máquina apenas, chamada estação de gerência. Nessa estação de gerência são executadas as funções utilizadas no dia-a-dia do gerente de sistemas, como: o agente responsável pela apresentação mais conveniente dos dados coletados, módulos para configuração do sistema, editor de regras de inferência (para o sistema especialista), "ping", etc. Além desses recursos, foram incluídos outros não pertencentes ao domínio de gerência, como funções de integração, remoção e edição de usuários em um sistema remoto.

Para que os objetos gerenciáveis da rede (objetos, ou nós, gerenciáveis são as máquinas de uma rede que são passíveis de serem monitoradas ou controladas por um conjunto de uma ou mais máquinas, denominado estação de gerenciamento) possam ser de fato gerenciados, é necessário que eles carreguem o módulo remoto. Esse módulo é um conjunto de funções e propriedades para sincronização de máquinas, controle de concorrência, funções de monitoramento, métodos para controle do sistema, verificação de recursos de máquina, além do sistema especialista multi-agente. Tudo isso é usado para o monitoramento e a aplicação de ações de gerência.

Através da estação de gerência o gerente da rede tem o controle sobre todas as máquinas com o módulo remoto carregado. A estação de gerência pode inclusive não estar necessariamente localizada na mesma rede local na qual as demais estações gerenciadas fazem parte. O gerente pode carregar o seus componentes de gerência a partir de um browser, compatível com Java™ 2, da sua própria casa, ou de qualquer lugar que tenha um computador conectado à Internet, eliminando assim a necessidade do gerente ter que se deslocar para a estação onde ocorreu o problema.

4. Descrição do Sistema

O sistema é composto por três módulos: o módulo local, o módulo remoto e o social. A razão da divisão do sistema nesses módulos é baseada na arquitetura de agentes desenvolvida para o projeto, que é composta por um agente local, um agente remoto e o outro um agente social. Um esquema simplificado dessa arquitetura multi-agente, para apenas um objeto gerenciado, pode ser visualizada na figura 1.1 abaixo:

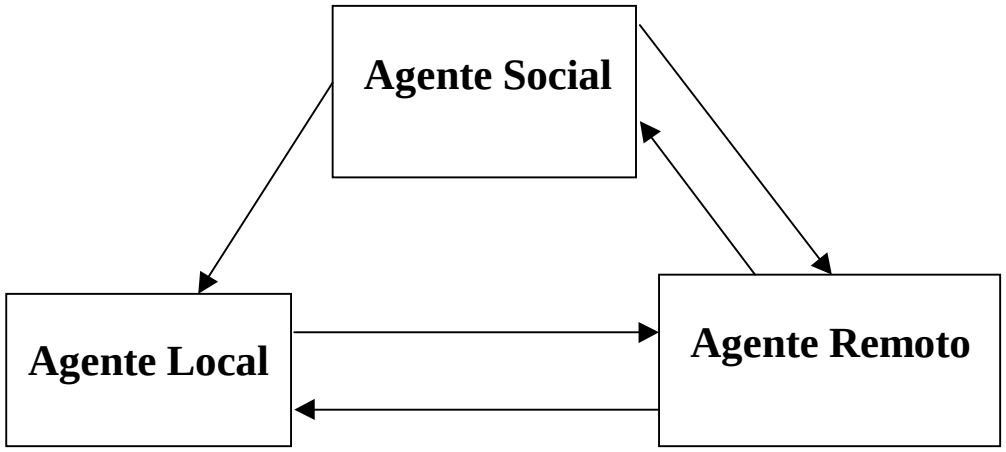

Figura 1.1 Arquitetura do Sistema

Cada um desses agentes tem um objetivo específico. O módulo local é o encarregado de produzir resultados satisfatórios para o gerente. Por isso, ele é executado na estação de gerência. O componente remoto é carregado na estação remota, para que ela seja vista como um objeto gerenciável pela estação de gerência. A função de tal componente é fornecer funções de monitoramento e controle, além de tratar da inferência das regras enviadas pelo gerente através do editor de regras. A parte do agente remoto responsável por esse tratamento e inferência de regras é o sistema especialista. A capacidade para tomada automática de decisões pertence a essa estrutura, e é devido à atuação dela no agente remoto que torna essa arquitetura uma plataforma para agentes inteligentes. O módulo social é o que serve como ponte de comunicação entre os agentes local e remoto, aplicando um conjunto de leis sobre as políticas de transmissão de dados. Ele também controla os critérios para distribuição de conhecimentos. Abaixo segue uma descrição mais sucinta das funções e propriedades de cada um dos módulos.

4.1 Agente Local

A função principal do agente local é apresentar os dados de gerência ao usuário. Tais dados são atualizados automaticamente numa freqüência definida pelo gerente de redes. Está disponível também no módulo gerente um editor de regras. Quando o usuário edita uma regra, ela é enviada, através da rede, para o sistema especialista, localizado no agente remoto. Da próxima vez que o agente local atualizar as informações de gerenciamento, o agente remoto realiza a inferência, baseado nas regras armazenadas no objeto gerenciado, e então retorna um conjunto de informações. Esse resultado é então apresentado ao usuário, tarefa essa exercida pelo agente local. Abaixo segue um esquema (figura 1.2) que ilustra a troca de mensagens:

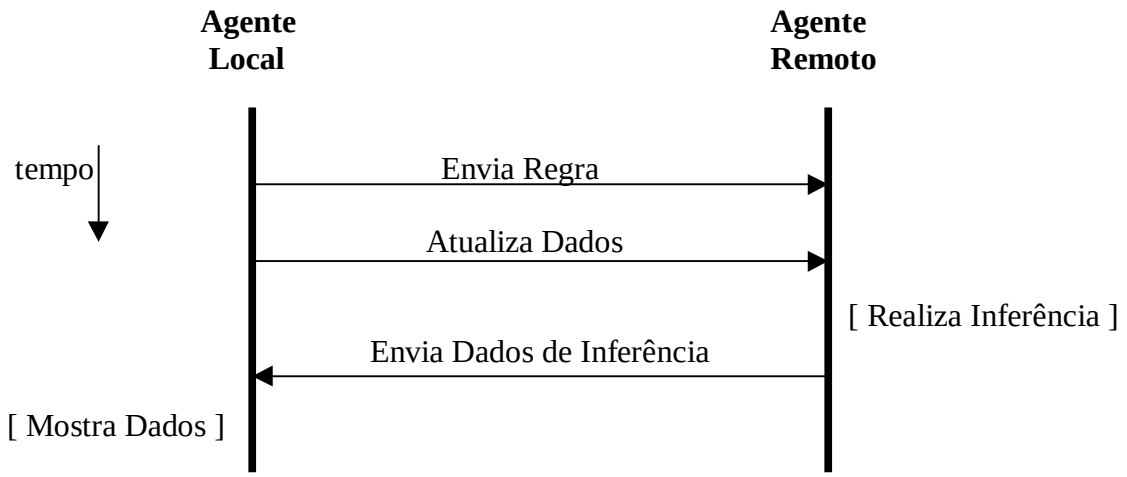

Figura 1.2 Esquematização da troca de mensagens

Observe que o agente social não participa desse tipo de comunicação entre os agentes remoto e local. Os dados apresentados ao gerente de rede podem estar numa apresentação diferenciada do usual, caso pelo menos uma das regras definidas seja inferida. Um dos possíveis resultados da prova de uma regra é uma advertência. Uma advertência pode ser um resultado pintado de vermelho ou com o fundo azul, por exemplo.

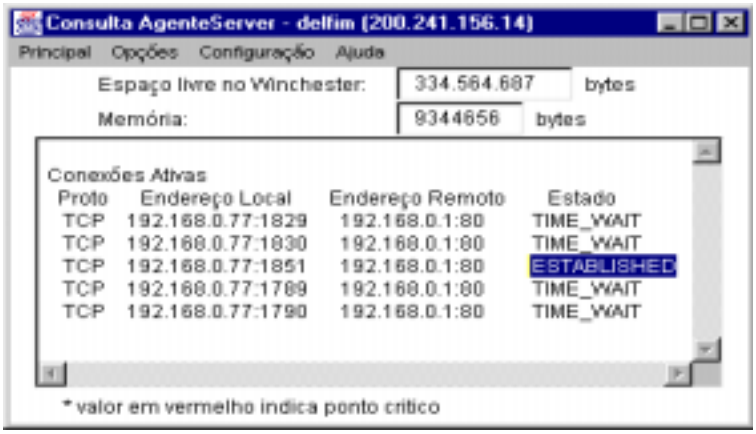

Figura 1.3

A tela mostrada acima é atualizada a cada 1 (um) segundo, que pode ser configurada de acordo com as necessidades do gerente. O nome na barra de título da janela é o nome do módulo local ("Consulta AgenteServer"), separado por um sinal de hífen ("-") do nome da estação remota (no exemplo, "delfim"). Entre parênteses está o seu endereço IP. Cada tela como essa, ilustrada na figura 1.3, é aberta na estação de gerência, para cada objeto gerenciado.

Além do estado das conexões remotas, são apresentadas estatísticas do uso dos protocolos IP, TCP, UDP e ICMP na estação gerenciada **[STEV97]**. Os estados dessas conexões são representados por um conjunto de variáveis, como, por exemplo, o número de pacotes recebidos, número de pacotes enviados, taxa de erros de saída, etc. Para o protocolo IP, são apresentadas 17 (dezessete) características:

- Pacotes recebidos

- Erros de cabeçalho recebidos
- Erros de endereço recebidos
- Datagramas encaminhados
- Protocolos desconhecidos recebidos (erros de protocolo)
- Pacotes recebidos descartados
- Pacotes recebidos entregues
- Solicitações de saída (de pacotes)
- Descartes de roteamento
- Pacotes de saída descartados
- Pacotes de saída sem rota
- Reagrupamento requerido (requisições para reagrupar fragmentos)
- Reagrupamento bem-sucedido
- Falhas de reagrupamento (erros ao tentar reagrupar os fragmentos recebidos)
- Datagramas fragmentados com êxito
- Falhas na fragmentação de datagramas
- Fragmentos criados (quantidade de fragmentos)

São mostradas 8 (oito) variáveis referentes ao protocolo TCP:

- Abertos ativos (conexões abertas para envio de dados)
- Abertos passivos (conexões abertas para recebimento de dados)
- Falhas em tentativas de conexão
- Conexões redefinidas
- Conexões atuais
- Segmentos recebidos
- Segmentos enviados
- Segmentos retransmitidos

Algumas estatísticas do protocolo UDP são apresentadas:

- Datagramas recebidos
- Datagramas enviados
- Nenhuma porta (não encontrada porta disponível)
- Erros de recebimento

E mais 26 (vinte e seis) variáveis para o protocolo ICMP:

- Mensagens (recebidas/enviadas)
- Erros (recebidos/enviado)
- Destino inatingível (recebidos/enviados)
- Tempo excedido (recebidos/enviados)
- Problemas de parâmetro (recebidos/enviados)
- Retardamentos de origem (recebidos/enviados)
- Redirecionamentos (recebidos/enviados)
- Ecos (recebidos/enviados)
- Respostas de eco (recebidas/enviadas)
- Datadores (recebidos/enviados)
- Respostas de datadores (recebidas/enviadas)
- Máscaras de endereço (recebidas/enviadas)

- Respostas nas máscaras de endereço (recebidas/enviadas)

Ao todo são 55 (cinqüenta e cinco) variáveis. A cada uma dessas variáveis é associado um valor inteiro maior ou igual a 0 (zero). Elas são apresentadas ao usuário seguidas por um sinal de igual e logo após o valor a ela atribuído pelo agente remoto. Todas as variáveis são enviadas pelo objeto gerenciado, mas antes de elas chegarem ao módulo gerente passam antes pelo sistema especialista, que realiza as inferências necessárias e transmite os dados, inferidos ou não, para o módulo local.

Existe um comando para testar a conexão de uma estação qualquer na rede (local ou Internet), semelhante ao utilitário "Ping", muito popular em sistemas UNIX. As diferenças principais estão na apresentação dos dados, que são bem mais detalhados.

Junto com essas funcionalidades, foram incluídas outras que não pertencem ao campo da gerência de redes. Nessa categoria se enquadra um utilitário para integrar, remover e modificar um usuário a uma rede Windows NT™. Dessa forma o gerente não precisa abrir uma sessão no servidor de arquivos dos usuários para alterar a senha de alguém, ou acrescentar um novo usuário ao sistema, o que aumenta a segurança, já que uma conta com acesso de Administrador não vai precisar estar aberta numa máquina rodando o Windows NT Server™, viabilizando a criação de uma estação só para administração de usuários.

Todas esses recursos estão disponíveis ao gerente de redes mesmo que ele não esteja numa máquina integrada à rede local sendo gerenciada. É suficiente que ele tenha uma conexão à Internet e um browser compatível com Java™ 2 (ou um browser não compatível e um Java Plug-In™), tal como o Internet Explorer™ 5 da Microsoft™. Ele pode assim gerenciar o objeto remoto e até alterar a senha de um usuário na rede local da empresa, tudo isso de sua própria máquina.

4.2 Agente Remoto

O agente remoto é o responsável por coletar informações do objeto gerenciado, realizar inferências baseadas nos dados coletados e exercer algum tipo de ação sobre a estação. Da arquitetura multi-agente desenvolvida, esse agente é o mais complexo. Ele envolve a utilização de alguns scripts (desenvolvidos em batch-files do Windows), consultas a bases de dados (locais ou remotas), execução de programas desenvolvidos em C++ e Assembly, etc. A necessidade da implementação de algumas rotinas em outros compiladores deve-se às facilidades de acesso a instruções de baixo nível presentes em algumas linguagens, como o Assembly e o C++. Tais programas executam ações como reinicialização remota, verificação de espaço livre em disco rígido, monitoramento da quantidade de memória RAM livre e fechamento de um processo em execução na máquina remota.

Abaixo segue-se a relação completa das ações que o agente remoto pode executar:

- Retornar espaço livre no winchester;
- Retorna quantidade de memória de acesso aleatório (RAM) livre;
- Retorna o nome e o endereço IP da máquina remota;
- Reinicializa o objeto gerenciado (reboot);
- Fecha o processo da Java Virtual Machine™ (JVM) que carregou o agente remoto;
- Retorna estado das conexões;
- Adiciona nova regra definida pelo gerente;
- Remove uma regra existente;
- Edita uma regra existente;
- Executa ações de gerência;

- Realiza inferência sobre as regras e os dados coletados;
- Envia para o agente local todos os dados coletados e inferidos.

Como as maiorias dessas ações já foram explicadas anteriormente, das funções definidas acima, apenas as duas últimas necessitam de um detalhamento especial. Elas são realizadas pelo sistema especialista localizado no agente remoto. Cada um dos agentes remotos vai ter um sistema especialista idêntico, porém muito possivelmente com bases de conhecimento diferentes. Detalhamos, na seção abaixo, o módulo sistema especialista.

4.2.1 Sistema Especialista

4.2.1.1 Definição

Um sistema especialista pode ser definido como um programa usado com a finalidade de tentar simular a maneira de tomar decisões de um especialista de uma área qualquer do conhecimento, como a Engenharia, Medicina, Geologia, etc. **[RUSS95]**. A parte do sistema especialista encarregada de realizar a inferência das regras chama-se "motor de inferência". Inferência é a tarefa de provar a conclusão de uma regra. O motor de inferência é a parte principal do SE. As regras utilizadas pelo motor são guardadas numa estrutura de dados denominada "base de regras". Um exemplo de regra numa base de regras de um sistema especialista no domínio da Segurança em Redes pode ser "SE PERMISSAO_DIRETORIO_GRUPO=W ENTAO DIRETORIO_GRUPO=DESPROTEGIDO", indicando que se o campo GRUPO do diretório analisado estiver com *flag* para escrita setado, isso carateriza que o diretório está desprotegido (conforme a base de regras configurada para o sistema especialista em questão). Para ajudar na inferência, existe a memória do especialista, denominada "base de conhecimento" ou "memória de trabalho". Esta base de dados serve para armazenar os fatos obtidos previamente ou coletados no decorrer da inferência. Um fato armazenado nessa base de conhecimento pode ser: "PERMISSAO_ARQUIVO_USER = RX", que mostra que o campo USER do arquivo sendo avaliado está com os *flags* para leitura e execução setados. Na figura 1.5 abaixo, é mostrada a janela principal do sistema especialista, no âmbito do domínio de gerência de redes.

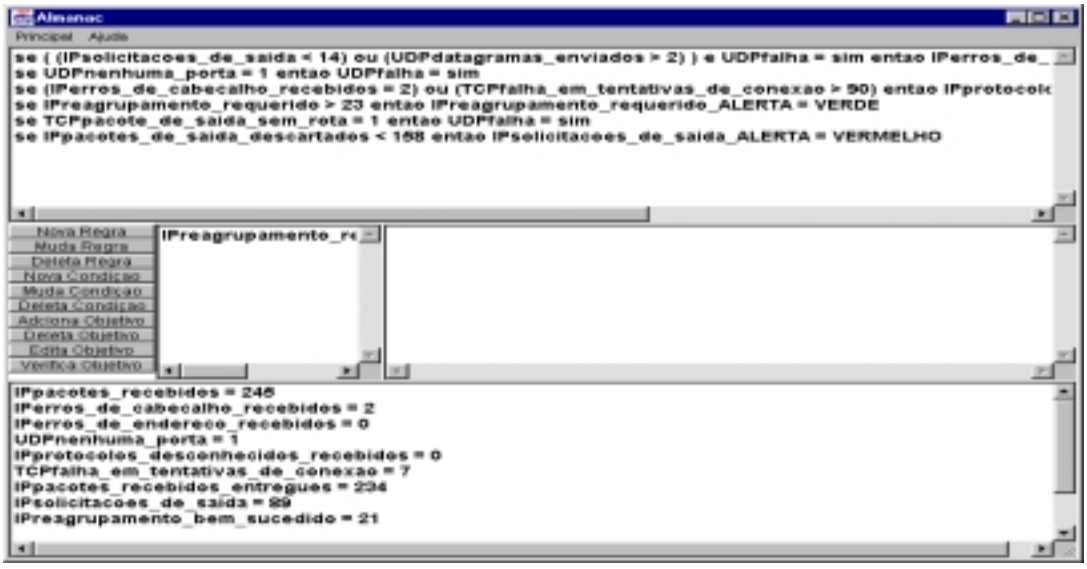

Figura 1.5

A parte superior da janela é a base de regras. A parte inferior é a base de conhecimento. O componente central esquerdo, com uma variável apenas, é a lista de conclusões a serem provadas. A área grande, na parte esquerda da janela, mostra possíveis mensagens de erro. A série de botões são o conjunto de operações disponíveis. Cada um desses componentes é explicado mais detalhadamente nas seções a seguir.

4.2.1.2 Base de Regras

A base de regras é uma tabela com um conjunto de regras necessárias para a inferência, baseada no conhecimento do especialista. A sintaxe das regras nessa base obedecem à seguinte gramática, definida na notação BNF (Backus-Naur Form):

REGRA → **se** PREMISSA **entao** CONCLUSÃO
PREMISSA → **(** PREMISSA **)** | TERMO **ou** PREMISSA | TERMO **e** PREMISSA | TERMO
CONCLUSÃO → IDENTIFICADOR **=** VALOR
TERMO → IDENTIFICADOR OP VALOR
ALFA → **a** | **b** | ... | **z** | **A** | **B** | ... | **Z** | _
NUM → **0** | **1** | ... | **9**
OP → **=** | **>** | **<** | **>=** | **<=** | **<>**
ALFANUM → ALFA | NUM
IDENTIFICADOR → ALFA ALFANUM | IDENTIFICADOR ALFANUM
VALOR → ALFANUM VALOR | ALFANUM

Os elementos em negrito são os símbolos terminais. Os que estão escritos inteiramente em maiúsculas são as variáveis.

A conclusão de uma regra pode executar uma determinada ação. Para que isso ocorra, a sintaxe da conclusão das regras tem uma estrutura um pouco diferente. No final do identificador da conclusão, concatenamos um caractere subscrito ("_") com uma cadeia de caracteres que indica a ação que a validação da premissa dessa regra deve executar. No caso da regra abaixo:

SE CAPACwinch < 10000 ENTAO CAPACwinch_ALERTA = AZUL

Caso a regra acima seja validada, isto é, se o espaço livre no disco rígido for menor que 10 Mb (dez megabytes), o sistema vai colocar o valor do espaço livre numa cor diferenciada, no caso, azul. Existem atualmente três funções de ação disponíveis. São elas: ALERTA, que pinta o valor da variável de uma determinada cor; JANELA, que abre uma janela no espaço de trabalho do gerente de rede; e ACAO, que executa uma determinada ação, representada por um *script*, ou um programa compilado numa linguagem qualquer.

4.2.1.3 Base de Conhecimento (Memória de Trabalho)

A base de conhecimento, ou memória de trabalho, é composta por uma lista de termos. Os termos obedecem à sintaxe definida anteriormente:

TERMO → IDENTIFICADOR OP VALOR

Ou seja, o termo é uma variável (identificador), seguida por um operador e um valor. O identificador, tal como na maior parte das linguagens de programação, é formado por um símbolo

alfabético, seguido por uma seqüência de símbolos alfanuméricos. O operador é qualquer um dos operadores relacionais válidos: =, >, <, >=, <= e <>. E o valor é definido como uma série de caracteres alfanuméricos.

Cada agente remoto tem a sua própria base de conhecimento diferenciada.

4.2.1.4 Motor de Inferência

O motor de inferência é o componente central do sistema especialista. É através dele que são efetuadas as tomadas de decisões do agente remoto. Ele é implementado através de uma função recursiva, que recebe apenas um argumento: o fato que se quer provar. O motor de inferência funciona da seguinte forma:

1. Ele verifica se a variável que se quer provar está na base de conhecimento;
2. Se estiver, a função termina, concluindo o fato;
3. Caso contrário, procura uma regra que possa concluir o fato;
4. Se encontrar, executa recursivamente essa mesma função sobre cada um dos termos da premissa, retornando ao passo 1;
5. Se não conseguir encontrar uma regra, a função não retorna nada, o que significa que a conclusão que foi objetivo da inferência não pôde ser concluída. A função recursiva termina.

Observando-se em maiores detalhes o algoritmo, vê-se que a técnica de busca usada na árvore de soluções do problema foi a “backward chaining”, ou encadeamento regressivo. Ou seja, busca-se logo a regra que prove o fato em análise. Depois, tenta-se avaliar toda a premissa da regra, para procurar concluí-la. Observe a seqüência [Conclusão → Premissa]. Por isso o nome “regressivo”. Veja também que o algoritmo acima é válido para regras simples, com premissas usando apenas um operador: o E lógico (AND). Por isso a necessidade do pré-compilador descrito anteriormente.

Dependendo da variável, ela pode ser verificada não só na base de conhecimento do objeto local, como também nos outros objetos gerenciados. E o papel principal do agente social é justamente escolher a máquina que pode ter essa informação. No algoritmo acima, o último passo deve então ser substituído pelos dois seguintes:

5. Se não encontrou uma regra que conclua o termo, consulte o agente social e veja se ele consegue encontrar um termo que prove aquela variável.
6. Se encontrou a condição em algum outro agente remoto, conclua condição; caso contrário, a função não retorna nada.

4.3 Agente Social

Apesar de implementar outras funcionalidades, a função principal do agente social é implementar a capacidade de conhecimento distribuído. As outras funções do agente social são de sincronização e controle de concorrência na atualização dos dados no agente local. Esse controle é necessário, porque só o Agente Remoto sabe o momento exato em que os dados monitorados estão prontos. Ele então usa o canal com o Agente Social para o sincronismo dos dados. Conforme descrito anteriormente, quando um fato a ser provado não é encontrado na base de conhecimento, nem existe qualquer regra que eventualmente possa provar esse fato, o motor de inferência submete o termo ao agente social, que então verifica a existência dele em outro objeto gerenciado. A figura 1.6 abaixo esquematiza esse processo, generalizado para *n* objetos gerenciados. Os fluxos de dados

em realce demonstram como se dá à troca de conhecimentos entre agentes remotos. Os números indicam a seqüência das instruções:

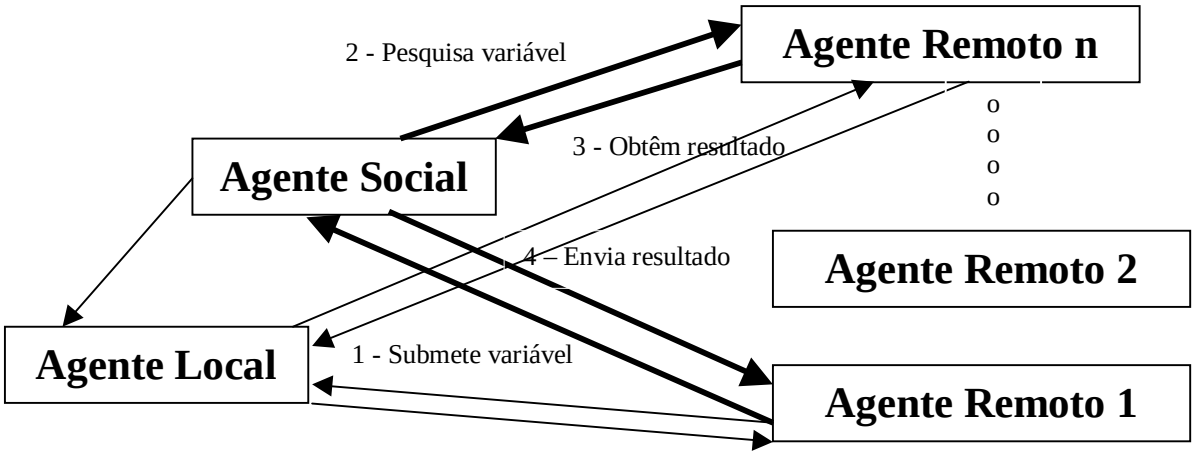

Figura 1.6

5. Agentes para a Segurança de Sistemas

Segurança de Sistemas é a área envolvida com o estudo e aplicação de técnicas e conceitos visando garantir a integridade, a confidencialidade, autenticidade e a disponibilidade das informações que integram algum sistema de computação. Atualmente, ela está tendo maior atuação na detecção de intrusão em sistemas conectados em rede, assim como pesquisa maneiras de evitar que tais ataques se efetivem e causem danos.

A utilização de agentes inteligentes para a solução de problemas de segurança é uma abordagem relativamente nova, e que está sendo continuamente defendida por grandes empresas de desenvolvimento de software em todo o mundo, como a IBM, que produziu o Tivoli; e a Hewlett-Packard, que comercializa o HP-OpenView. Ambas são ferramentas de apoio à gerência de redes que utilizam, com maior ou menor intensidade, o conceito de agentes computacionais inteligentes **[BUFF99]**.

5.1. Tipos de Ataques Mais Comuns a Máquinas Conectadas em Rede

Discutiremos aqui sobre a classificação normalmente usada para categorizar a enorme variedade de formas de ataque à máquinas conectadas em rede, em especial à Internet. A divisão é bastante ampla, mas classifica suficientemente bem, baseado nos causadores e nos meios sobre os quais o ataque é efetuado. Existem três classificações preliminares **[ASSA99]**:

- Vulnerabilidades introduzidas por usuários do sistema;
- Vulnerabilidades existentes em aplicações para o SO, e na programação da pilha TCP/IP;
- Vulnerabilidades causadas por falhas na configuração do SO.

O sistema proposto realiza atualmente 2 (dois) tipos de verificações: uma, baseada na primeira categoria exposta acima, que verifica a permissão em diretórios na área do sistema de arquivos que armazena as contas dos usuários; outra, classificada como pertencente à terceira categoria mostrada anteriormente, que tenta prevenir tentativas de intrusão conhecidas como Port Scan.

5.2. Aplicação Prática do Sistema: Controle de Permissões a Arquivos e Detecção de Port Scan

São descritos abaixo dois exemplos de aplicação prática do sistema de apoio à gerência de redes implementado, visando a prevenção e a detecção de intrusão a uma determinada máquina executando o Agente Remoto discutido anteriormente. Apesar do programa ter a capacidade de atuar em vários aspectos da gerência de redes (gerência de contabilidade, falha, manutenção, desempenho e segurança), o campo de atuação que foi testado foi o de Gerenciamento de Segurança.

5.2.1. Controle de Permissões a Arquivos

Algumas vezes o gerente da rede (ou o administrador de sistema), descobre que alguns dos diretórios na área dos usuários estão com as permissões de acesso (flags de arquivo) incorretas. Ele sempre tenta descobrir algumas dessas brechas de segurança, porque algum intruso pode se aproveitar do acesso a uma determinada conta para roubar dados valiosos do sistema, ou até mesmo danificá-lo. Até aí tudo bem, é uma tarefa simples, basta executar um *chmod* sobre o diretório e pronto, certo? Errado. Nem sempre o administrador (gerente) pode estar pronto, prestes a sanar em tempo hábil esse tipo de problema, que pode posteriormente fazê-lo perder tempo em tentar sanar as conseqüências de uma possível invasão ao sistema. E se vários diretórios aparecem abertos, o grau de dificuldade fica ainda maior **[FRIS96]**. É aí que entra o agente inteligente, que com a ajuda do seu módulo de inferência (Sistema Especialista), toma a decisão por si só, sem precisar incomodar o gerente da rede com essa espécie de problema de segurança. Para que a isso seja feito, o Agente Remoto (descrito anteriormente) realiza a seguinte seqüência de operações:

1. O Agente monitora constantemente as permissões nos diretórios, para tentar descobrir alguma permissão de leitura, execução ou escrita inválida;
2. Ele envia os dados coletados para o Sistema Especialista, para que ele tome a decisão apropriada, baseada na base de conhecimento recém atualizada e nas regras enviadas pelo gerente;
3. Caso ele consiga inferir alguma regra (tal como *Se permissao_diretorio_grupo=rw entao altera_diretorio_grupo=x*, significando que caso ele encontre um diretório com permissão de grupo para leitura ou escrita, substitua o flag de grupo para *somente execute* (x)), ele toma a decisão apropriada, baseada da conclusão da regra recém inferida.

5.2.2. Detecção de Intrusão com Port Scan

Uma técnica bastante usada por *hackers* é a varredura das portas disponíveis num determinado servidor, para descobrir uma que tenha alguma facilidade de acesso, por meio da qual ele possa se infiltrar e assim ter acesso, mesmo que restrito, à máquina **[GRAH99]**. Normalmente isso acontece devido a administradores pouco atenciosos, que deixam serviços sem utilidade para o sistema executando no servidor, serviços esses que, por serem inúteis ao sistema, são ignorados completamente pelo gerente. Ao ignorar esses serviços, ele se esquece de cuidar dos detalhes de controle de acesso, e de efetuar periodicamente vistorias nos *logs* do serviço, para verificar se algo errado está ocorrendo. A seguinte *signature* (assinatura), foi implementada na aplicação em questão:

1. O Agente Remoto realiza uma monitoração nas conexões do objeto gerenciável da rede;
2. O módulo monitor do Agente Remoto envia os dados para o módulo de inferência (Sistema Especialista).
3. Baseado nas regras armazenadas na Base de Regras pelo gerente de redes, ele (o agente) tenta encontrar 10 (dez) tentativas de conexão mal-sucedidas a, respectivamente, 10 (dez) portas consecutivas (por exemplo, tentativas mal-sucedidas de acesso às portas 2001, 2002, 2003,..., 2010, na máquina executando o Agente Remoto). Isso é um indicativo, não definitivo, mas bastante convincente, de tentativa de intrusão por análise prévia de indefesas através do mecanismo *port scan*.

6. Conclusão

Depois do que foi exposto, percebe-se a vantagem da utilização de uma arquitetura de agentes na solução de problemas de gerência. O gerente não precisa estar sempre manipulando a estação gerenciada para resolver ou prever falhas. Também elimina-se a necessidade de que o gerente de redes esteja se locomovendo para o objeto gerenciado para eliminar problemas ou executar otimizações, já que o gerente pode efetuar suas tarefas cotidianas de qualquer máquina com uma conexão à Internet. Há inclusive a vantagem da utilização de uma arquitetura de gerenciamento com uma estação local de gerência.

Outro aspecto importante é a distribuição de tarefas entre os objetos gerenciados, o que diminui a sobrecarga de operações na estação de gerência. Como cada agente remoto tem o seu próprio sistema especialista, ele envia os dados inferidos prontos para o agente local. O agente local só precisa receber esses dados e apresentá-los de forma conveniente ao gerente de redes. Esse esquema de distribuição não é aplicado no modelo SNMP de gerência. Toda a inteligência do sistema está concentrada na estação de gerência, ao contrário do que acontece no sistema em discussão. O motivo normalmente atribuído para essa escolha se deve ao fato de não se querer sobrecarregar os nós gerenciados com mais software, e tornar os agentes o mais simples possível **[TANE96]**.

Todos esses aspectos tornam a utilização da Inteligência Artificial para o auxílio ao gerente de redes uma solução viável e bastante útil. Cada vez mais as empresas de desenvolvimento de software para Gerência de Redes percebem a utilidade dessa tecnologia, e investem em pesquisa nessa área, que tende a crescer cada vez mais.

Aprimoramentos estão sendo adicionados ao sistema, de forma a cobrir algumas funcionalidades normalmente requeridas por gerentes de redes, tais como aspectos mais avançados de gerenciamento de segurança (detecção de intrusão) e desempenho. Há também a intenção de, num projeto futuro, oferecer recursos para o gerenciamento das aplicações que executem nas máquinas gerenciadas (por exemplo, servidor WWW, sistema de correio eletrônico, etc.).

BIBLIOGRAFIA

[ASSA99] – Assad, Rodrigo Elia – Agentes Inteligentes x Segurança – DI/UFPE - 12/1999

[BUFF99] – Buffalo University – Network Management Resources
fonte: http://netman.cit.buffalo.edu/

[CAMP98] - Campione, Mary, et al. - The Java Tutorial - A Pratical Guide to Programmers
fonte - http://java.sun.com/docs/books/tutorial/index.html

[CEST96] - Cesta, André Augusto - A Linguagem de Programação Java
fonte: http://www.dcc.unicamp.br/~cmrubira/aacesta/java/javatut.html

[CHAN98] - Chan, Patrick - The Java Developers Almanac 1998 (Java Series) - (July 1998) Addison-Wesley Pub Co.

[FRIS96] - Frisch, Aaleen - Essential System Administration : Help for Unix System Administrators (Nutshell Handbook) / O'Reilly & Associates / December 1996

[GRAH99] - Graham, Robert - Network Intrusion Detection Systems (NIDS) FAQ
fonte: http://www.robertgraham.com/pubs/network-intrusion-detection.html

[HUNT98] - Hunt, Craig, Estabrook, Gigi (Editor) - TCP/IP Network Administration - O'Reilly & Associates / January 1998

[MURC99] - Murch, Richard, Johnson, Tony - Intelligent Software Agents, (January 1999) Prentice Hall

[RUSS95] - Russell, Stuart T., Norvig, Peter - Artificial Intelligence : A Modern Approach (Prentice Hall Series in Artificial Intelligence) Prentice Hall / February 1995

[STAL99] - Stallings, William - Snmp, Snmpv2, Snmpv3, and Rmon 1 and 2 - 3rd edition (January 1999) Addison-Wesley Pub Co

[STEV97] – Stevens, W. Richard – TCP/IP Illustrated, Volume 1: The Protocols - (Julho 1997) Addison-Wesley Pub Co

[TANE96] - Tanenbaum, Andrew S. – Computer Networks - Third Edition - 1996 Prentice-Hall, Inc.

[WALL96] - Wall, Larry, et al Programming Perl (2nd Edition) / O'Reilly & Associates / October 1996

[WEIS99] - Weiss, Gerhard (Preface) Multiagent Systems : A Modern Approach to Distributed Artificial Intelligence (June 1999) MIT Press