

Um ambiente para a comunicação de agentes na Internet

Roberto Duarte Fontes
Luís Otávio Alvares

Universidade Federal do Rio Grande do Sul
Centro de Pós-Graduação em Ciência da Computação (UFRGS)
Campus do Vale - Bloco IV
Av. Bento Gonçalves, 9500 - Bairro Agronomia
Caixa Postal 15064
Cep 91501-970, Porto Alegre, RS
e-mail: [fontes,alvares]@inf.ufrgs.br

Resumo

O grande salto tecnológico ocorrido nos últimos decênios, em áreas como a informática e as telecomunicações, aliado à heterogeneidade de arquiteturas de *hardware* e sistemas operacionais, faz da Internet um ambiente propício para a utilização de agentes. Este trabalho tem por objetivo principal apresentar a especificação de um modelo destinado à interação de agentes autônomos no âmbito da Internet, de forma a possibilitar, entre outros, a localização de agentes e a conexão entre os agentes, permitindo a troca de mensagens entre eles. Para tanto, é apresentado um modelo de ambiente para facilitar a interação de agentes na Internet. Este modelo é depois especificado detalhadamente e materializado na forma de um protótipo implementado e que provê o serviço proposto.

Palavras-Chave: Sistemas Multiagentes, Internet, Redes de Computadores.

1. INTRODUÇÃO

Muito do tempo útil de trabalho humano que acaba sendo perdido com tarefas simples, poderia ser realizado por assistentes inteligentes junto a Internet, assistentes que teriam a finalidade de auxiliar o usuário da rede. Cada assistente inteligente constitui-se em um agente modular e autônomo, e encontra seu embasamento teórico na Inteligência Artificial e na Inteligência Artificial Distribuída respectivamente.

Os sistemas de Inteligência Artificial (IA) tradicionais concentravam sua atenção em um único agente, o qual seria provido de alguma espécie de inteligência, e sozinho seria especialista na realização de uma tarefa específica.

Atualmente o enfoque tende a ser outro, fala-se em Inteligência Artificial Distribuída. Nesta subárea da IA o enfoque recai sobre o aspecto de integração de agentes providos de alguma inteligência. São especialistas na atividade que desempenham, não trabalham mais isolados, mas de forma cooperativa tentam resolver um problema da melhor forma possível, caracterizando uma área da Inteligência Artificial Distribuída, a área dos Sistemas Multiagentes (SMA) [DEM 93].

Tendo em vista a crescente utilização das redes de computadores e a necessidade de agentes independentes e autônomos que possam realizar tarefas em grupo, através da interação e colaboração de outros agentes, os SMA estão-se proliferando rapidamente. A problemática sempre presente nas pesquisas era como fazer com que programas baseados no conhecimento pudessem comunicar-se em uma rede para solucionar problemas complexos.

A solução, então, foi a utilização de agentes. Agente é uma entidade real ou virtual que emerge num ambiente onde pode tomar algumas ações, que é capaz de perceber e representar parcialmente esse ambiente, de comunicar-se com outros agentes e que possui um comportamento autônomo - uma consequência de sua observação, seu conhecimento e suas interações com outros agentes [SIC 92].

A Internet é um ambiente privilegiado para a utilização de agentes em virtude de suas características. Dentre as principais, podem-se citar a arquitetura, a topologia, a heterogeneidade de *hardware* e *software*, além da abrangência.

A motivação maior deste trabalho foi a de facilitar a interação de agentes autônomos em um ambiente altamente dinâmico e heterogêneo, como a Internet.

Outro aspecto motivador, foi a possibilidade de fundir várias tecnologias em um único projeto: agentes, SMA, arquitetura, protocolos e serviços Internet, programação para redes, orientação a objetos e multiprogramação.

Os objetivos deste trabalho são o refinamento e implementação da arquitetura para a coordenação de agentes na Internet, proposta por Cazella [CAZ 97], a definição e especificação de um protocolo que torne a comunicação entre os agentes transparente ao usuário e a agregação deste protocolo à arquitetura.

Em outras palavras, o objetivo é a criação de um ambiente que facilita a implementação de sistemas multiagentes na Internet.

Este artigo está estruturado em 5 seções. Na seção 2, é mostrada a arquitetura geral proposta para facilitar a interação de agentes na Internet. A seção 3 detalha o modelo introduzido e a seção 4, mostra a estrutura da implementação realizada. Finalmente a seção 5 apresenta as conclusões do trabalho em termos de contribuições e limitações.

2. ARQUITETURA PARA COMUNICAÇÃO DE AGENTES

Várias arquiteturas para a coordenação da colaboração de agentes tem sido propostas. A visão geral da arquitetura de coordenação de agentes para a Internet, proposta por Cazella [CAZ 97a], é apresentada na figura 1. Nesta figura, podem-se identificar dois domínios de rede (com suas máquinas locais designadas por *cl..cn*), conectadas às respectivas máquinas “agent”, comunicando-se através da Internet. Cada máquina pode ter um ou mais agentes, independente da função ou origem desses (representados pelos hexágonos 1..n).

Esta arquitetura torna a comunicação entre agentes transparente para o usuário. Um agente deve conhecer apenas a identificação do agente com o qual quer interagir, não importando a sua localização. Os agentes podem estar localizados na mesma máquina, no mesmo domínio ou em domínios distintos.

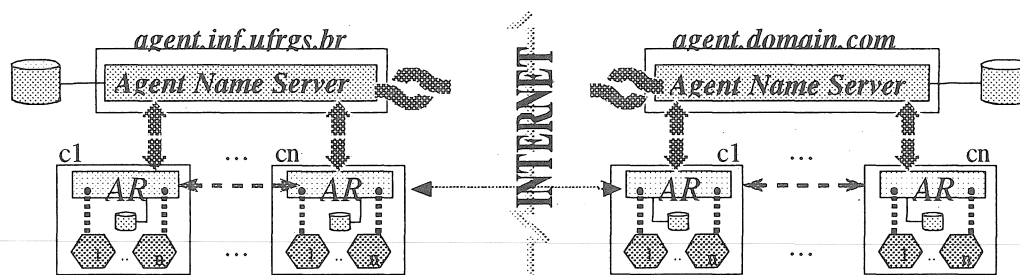


Figura 1. - Visão geral da arquitetura de interação de agentes para a Internet

As principais características desta arquitetura são:

- a existência de um servidor central – existe um servidor de nomes de agentes (ANS - *Agent Name Server*) por domínio, a qual desempenha a função de armazenar (em um arquivo local) e fornecer quando solicitado) os endereços dos agentes localizados no domínio e suas descrições;
- roteadores de agentes - os processos de comunicação entre os agentes são feitos por intermédio dos roteadores de agentes (AR - *Agent Router*) que desempenham os papéis de intermediadores na comunicação entre os agentes;

A arquitetura mostrada acima é independente do tipo de sociedade em questão, tendo um funcionamento que supre as necessidades de integração entre os agentes, independente do fato de a sociedade ser aberta ou fechada. O suprimento de tais necessidades é feito através do ANS, que é responsável por manter uma lista contendo as identificações de todos os agentes cadastrados no domínio referido.

Para haver uma maior agilidade no serviço prestado pelo ANS, ele é instalado numa máquina que possui, como endereço o nome “agent”, podendo este ser virtual. Como ilustração, tem-se a máquina “*agent.inf.ufrgs.br*”. Para que ocorra a resolução de nomes de agentes, basta uma requisição a esse servidor.

Cada máquina, em uma rede, pode conter um ou mais agentes específicos, independente de função e origem.

Nas máquinas de uma rede, onde os agentes estão localizados, além desses, ter-se-á um roteador denominado AR. O AR é um processo que está ininterruptamente em execução, aguardando por conexões, a fim de executar, principalmente, três funções de fundamental importância para a comunicação:

- distinguir, através de mecanismos internos, de qual e para qual agente na máquina onde está em execução, é destinada a informação que está chegando;

- intermediar a comunicação dos agentes. Em outras palavras, desempenhar o papel de meio de comunicação entre os agentes;

- gerenciar um arquivo de *cache*, onde são armazenadas as últimas requisições de identificação feitas pelos agentes ao ANS, por intermédio do AR.

A utilização de um arquivo de *cache*, local a cada AR, proporciona um melhor desempenho e uma menor centralização em relação ao ANS.

A centralização da comunicação dos agentes nos ARs não tira a flexibilidade da arquitetura, sendo esta necessária devido aos seguintes aspectos:

- integridade das máquinas – algumas verificações nas informações são feitas nos ARs para averiguação da segurança;

- tempo de CPU - seria inviável uma arquitetura em que todos os agentes ficassem em execução, no aguardo do momento de agirem, pois, dessa forma, o desempenho da CPU estaria altamente comprometido. Apenas os ARs ficam em execução e, no momento em que for necessária a ativação de um determinado agente, este será ativado pelo AR, a fim de que possa agir.

A seguir é mostrado um exemplo de interação entre agentes de domínios distintos, utilizando-se, como ilustração, o esquema apresentado na figura 2.

1) Numa primeira etapa do exemplo, supõe-se que o agente2 e o agente3 desempenham a função de agentes agenda. O dono do agente2 solicitou a marcação de uma reunião com o proprietário do agente3.

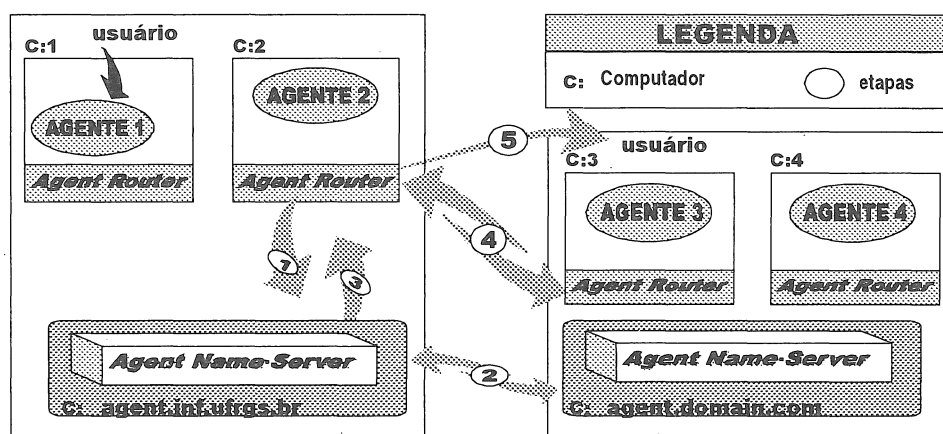


Figura 2. - Exemplo de uma interação na Arquitetura

Para poder marcar a reunião com o agente desejado, são dados vários passos.

❶ O agente2 procura o endereço do agente destinatário no arquivo de agentes local (função realizada na verdade pelo AR). Não irá encontrar, pois, neste exemplo, o agente está localizado em um outro domínio.

❷ Uma pesquisa ao arquivo de *cache*, mantido pelo AR, é feita pelo agente. Supondo-se que esta é a primeira vez que o agente destinatário é requisitado por algum agente cadastrados nesta máquina (C2), não irá encontrar;

❸ Após as tentativas frustradas de localizar o endereço do agente desejado, tanto no arquivo local de agentes quanto no arquivo de cache, o agente entra em contato com o ANS da sua rede e solicita o endereço do agente desejado (item 1 do esquema).

Neste exemplo, apresenta-se a comunicação entre os agentes como se fosse direta, uma vez que fica transparente a participação dos AR nessa atividade.

Como, neste exemplo, o agente desejado não faz parte da rede onde está o ANS responsável pelo agente2, este terá de estabelecer uma comunicação com a máquina que possui o domínio referenciado no nome do agente, de entrar em contato com o ANS desta outra rede (item 2 do esquema) e, finalmente, solicitar o endereço requerido. De posse deste endereço, o AR repassa-o para o agente2 (item 3), e este agora pode entrar em contato com o agente3.

Deve observar-se que o acesso ao outro ANS ocorreu pelo fato de o agente em questão não estar na mesma rede do agente2.

2) Na segunda etapa (item 4), o agente2 estabelece uma comunicação com o agente3 (no esquema, a conexão é apresentada de maneira direta $ag2 \leftrightarrow ag3$, os detalhes são abordados no capítulo 6), sendo feita a partir deste momento, toda a interação exigida por um agente agenda. Após o estabelecimento da conexão, depois de toda a negociação ter terminado, a conexão é desfeita, e o endereço do agente3 é armazenado no arquivo de *cache* do AR, na máquina do agente2.

Supondo que, mesmo de posse do endereço completo do agente3, o agente2 não tenha conseguido estabelecer comunicação depois de algumas tentativas frustradas, o agente2 envia um *e-mail* informando o ocorrido para o usuário responsável pelo agente requerido (item 5).

3. ESPECIFICAÇÃO DO MODELO

A especificação terá fundamentação nos três componentes principais mencionados anteriormente:

- Agente – é a função específica do agente acrescida de um módulo responsável pelas funções de comunicação do protocolo;
- AR – sua principal função consiste em intermediar a comunicação entre os agentes;
- ANS – é um servidor responsável pela tarefa de localizar os agentes na Internet.

Com o objetivo de especificar em detalhes esses módulos são apresentados os respectivos diagramas de transição de estados: figura 4 (módulo Agente), figura 5 (módulo AR) e figura 6 (módulo ANS). Os códigos de resposta do protocolo estão descritos na tabela 1 (seção 3.4). Em [FON 99] encontra-se a definição completa do protocolo.

3.1 Agente

O Agente é formado por um módulo de aplicação que executa as funções da aplicação específica, e por um módulo responsável por toda a parte da comunicação, denominado cliente de comunicação. A figura 3 ilustra esses módulos.

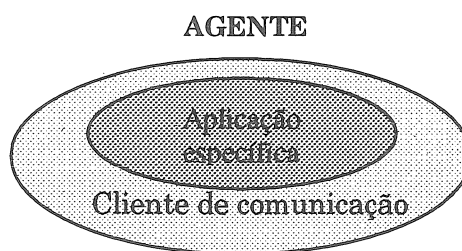


Figura 3. - Módulos do agente

A identificação do agente, é formada a partir do *e-mail* do seu proprietário no domínio onde está cadastrado e seu nome:

username @ domain /nome_agente*e-mail*

Pode-se exemplificar a identificação de um agente, com um agente de nome *agenda*, pertencente ao domínio "inf.ufrgs.br". Neste caso, a identificação do agente seria: fontes@inf.ufrgs.br/agenda.

Essa identificação de agentes proporciona à arquitetura:

- facilidade na identificação de agentes - se houver mais de um agente de funções semelhantes em atividade na mesma máquina (por exemplo, várias agendas do mesmo usuário ou de diferentes usuários), será possível identificar para qual desses agentes a informação é destinada;

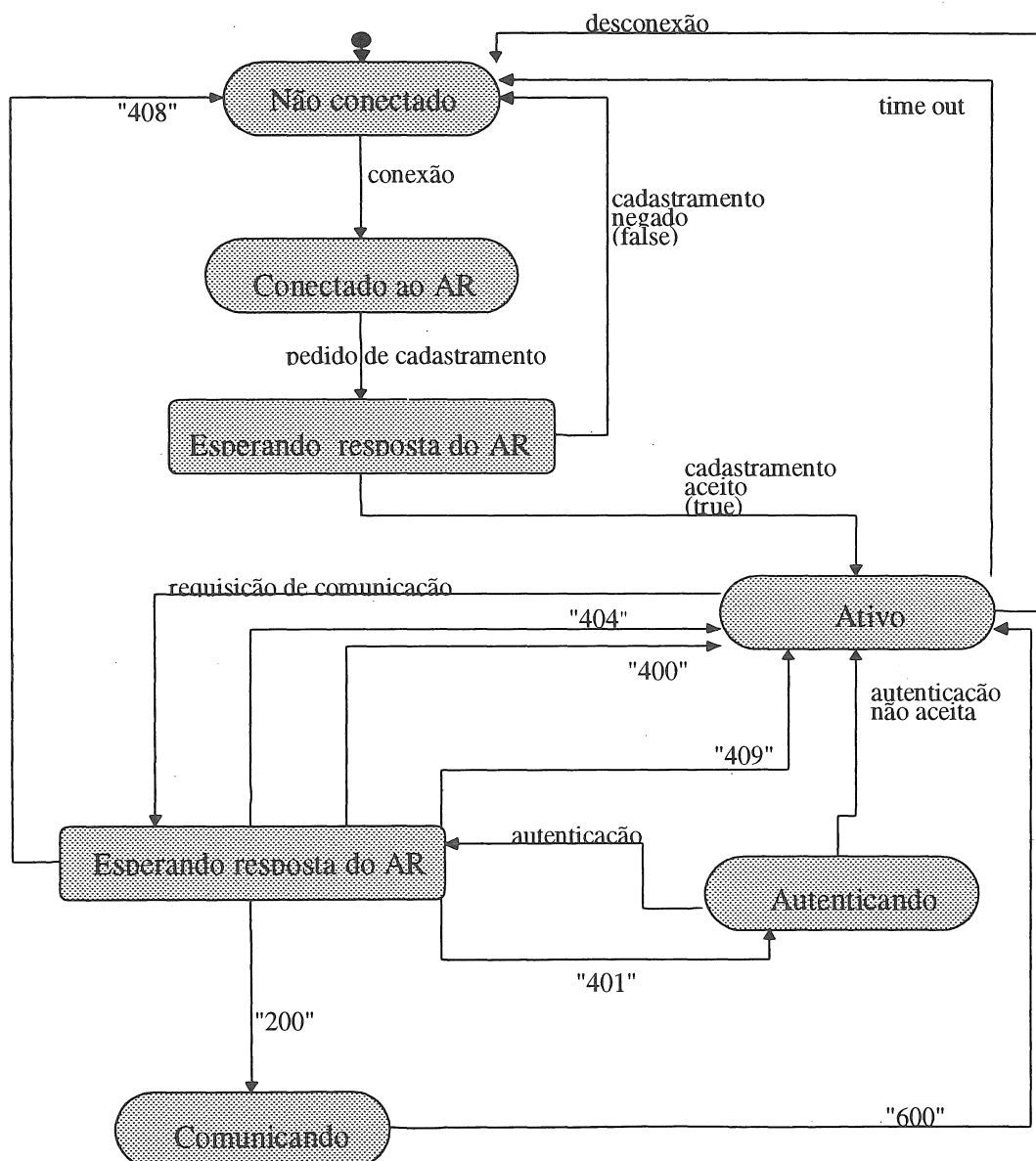


Figura 4: Diagrama de estados do Agente

- facilidade de roteamento - esta identificação do agente propicia uma facilidade para a determinação do domínio de rede em que se encontra o agente;
- identificação do proprietário do agente - devido à utilização do *e-mail* na composição da identificação do agente, fica fácil a determinação do proprietário deste. Em caso de um determinado agente1 tentar estabelecer conexão com um outro agente2 e, após um certo número de tentativas, essa conexão não for possível, define-se que o agente1 poderá enviar um *e-mail* informando ao responsável (proprietário), pelo agente2, que algum tipo de irregularidade está ocorrendo com o agente2.

Existe um controle de cadastramento de agentes, mantido pelo ANS, para que não seja possível o cadastramento de agentes com identificações já existentes, fazendo com que estas sejam únicas no domínio Internet.

3.2 Agent Router - AR

O AR é, ao mesmo tempo, um cliente e um servidor, sendo responsável pelo tratamento de todas as conexões provenientes dos agentes, exercendo o papel de intermediário na comunicação entre eles.

Após a conexão de um agente à porta lógica onde o AR está ativo e depois do envio de sua identificação para que seja processado seu *login*, é retornado, como resposta, uma mensagem indicando a aceitação ou não. Depois da realização do *login* do agente junto ao AR, para iniciar-se o processo de requisição de comunicação, basta ao agente enviar a identificação do agente destinatário (requisição). Se, na localização deste agente, ocorrer alguma falha, é retornada uma mensagem do protocolo informando-a ao agente.

A identificação é avaliada tanto no processo de *login* quanto na requisição, a fim de verificar a não existência de erros sintáticos.

De posse da localização do agente destinatário, o AR procede, se na mesma máquina, à execução desse agente; se, em máquina ou domínio diferente, a execução é realizada pelo AR da máquina onde o agente está localizado.

Na execução de agentes, que não fazem parte da máquina onde está o agente requisitante da comunicação, é acionada uma conexão para o AR pertencente ao endereço retornado, da consulta ao ANS. Em tal conexão, pode haver erros, que são mapeados para o protocolo e, a seguir, enviados ao agente com a finalidade de que este os processe.

A comunicação entre ARs é quase idêntica à realizada entre agente e AR, incluindo todo o processo de *login*. A única diferença nesta conexão é o envio do código "200" do protocolo, ao agente requisitante da comunicação, pelo AR destinatário, após estar o agente devidamente conectado e registrado ao AR. A mensagem do protocolo, informa que a requisição foi aceita e que o agente está autorizado a iniciar a troca de mensagens com o agente destinatário. O envio do código não é feito diretamente pelo AR destinatário ao agente, mas sim roteado via AR remetente para o agente.

Quando do encerramento de qualquer conexão ao AR, seja esse término intencional ou por falha, é feito o tratamento das referências mantidas para a conexão, ou seja, o *logout*.

3.3 Agent Name Server - ANS

O ANS é o servidor de endereços de agentes, o qual fica sempre aguardando conexões, para então, proceder à resolução de nomes.

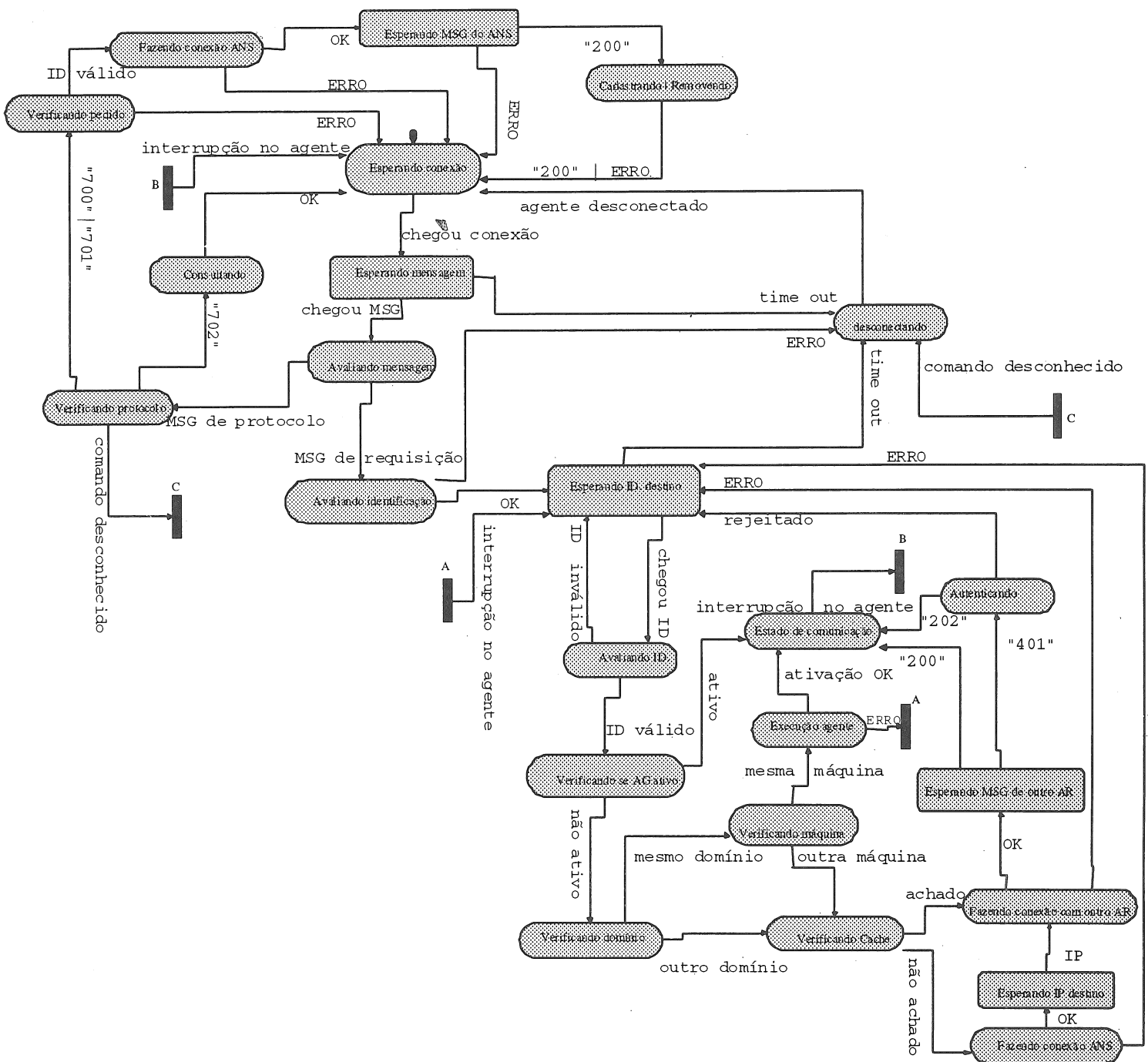


Figura 5. – Diagrama de estados do AR

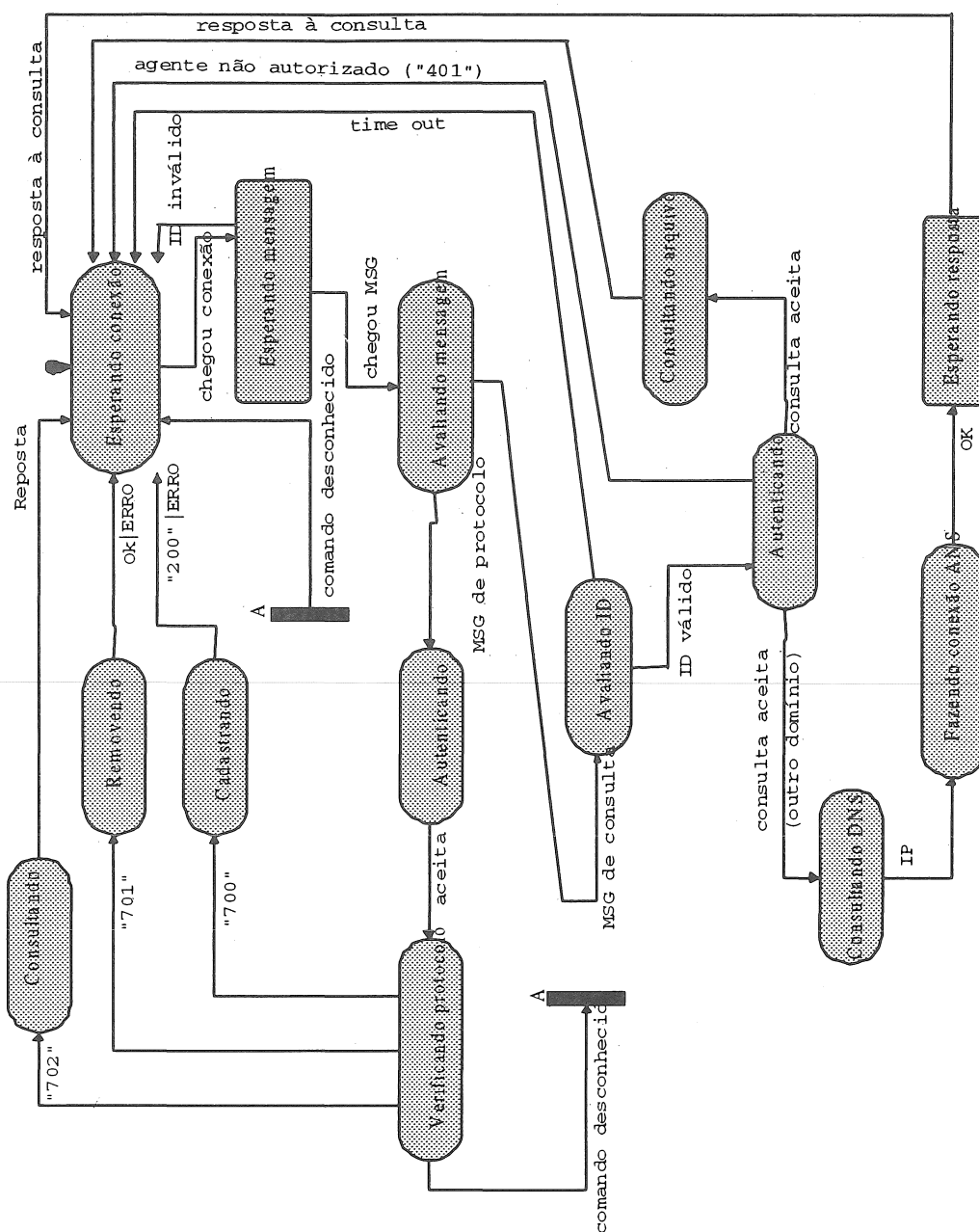


Figura 6. – Diagrama de estados do ANS

Na resolução de nomes, primeiramente é feita uma verificação para determinar a não existência de erros sintáticos na requisição de comunicação do agente. Não havendo erro sintático na requisição do agente, inicia-se o processo de resolução de nomes com a verificação, em nível de domínio, para qual destino é a requisição. Esta verificação, determina se o agente requisitado, faz parte do próprio domínio ou pertence a algum outro na Internet.

É feita uma consulta ao arquivo global de agentes, quando a requisição é para o próprio domínio, resultando numa mensagem com a informação que representa o endereço IP da máquina onde este agente está localizado, conforme exemplo: "143.54.8.170"; ou vazia "" significando que tal agente não existe. O número IP é enviado ao AR local, ou, se o agente não existir, um código do protocolo é repassado ao agente.

Uma consulta ao DNS é acionada, se a requisição é para outro domínio. Essa consulta deve retornar, uma mensagem com o endereço IP da máquina "agent" do domínio requisitado. Após ser

retornado o endereço, faz-se uma conexão ao ANS deste endereço, e envia-se a identificação do agente destinatário. O ANS do destinatário, faz todo o tratamento da requisição, conforme foi mencionado, e retorna o endereço IP da máquina, onde está localizado o agente destinatário, ou mensagem informando algum erro, segundo o protocolo.

Após a resolução de nomes os clientes (ARs ou ANSs), são automaticamente desconectados.

3.4 Mensagens

Existem três diferentes tipos de mensagens utilizadas no protocolo:

1. *de requisição* - a requisição, é a solicitação para que se ative o agente destinatário. Em todo o processo de requisição, é utilizado, para transmissão e recepção o tipo *string*;
2. *de comunicação* - na comunicação entre os agentes com o conceito de encapsulamento, cada mensagem pode ter, tamanho e formato arbitrários. O formato pode ser de qualquer tipo, proporcionando, assim, um elevado nível à comunicação. Juntamente à mensagem, são encapsulados *headers*, destinados ao controle do fluxo, da integridade e da segurança. Estes *headers* são:
 - assinatura – serve para impedir que agentes não autorizados, intervenham por engano ou maliciosamente na comunicação dos agentes;
 - destinatários – a comunicação de um agente pode ocorrer sob forma de *multicast* ou de *broadcast*. Esse *header* é composto por um número inteiro, maior que zero, que indica o número de destinatários, sucedido pelo mesmo número de *strings* representando a identificação desses destinatários. Para a transmissão em *broadcast*, emprega-se, como *header*, somente o número -1;
 - tamanho da mensagem – a fim de manter-se um controle da consistência da mensagem, usam-se números inteiros. Estes números referem-se ao tamanho da mensagem;
 - identificação do remetente – a identificação do remetente da mensagem serve para que o agente destinatário possa encaminhar ao remetente da comunicação uma resposta;
3. *de protocolo* - as mensagens de protocolo são tratadas independentemente das mensagens de comunicação, isso, é uma característica da utilização do conceito de empacotamento de mensagens.

Os códigos de *status* são formados por três dígitos inteiros. O primeiro dígito do código de *status* define a classe de resposta. Os últimos dois dígitos não são categorizados.

Na tabela 1, são apresentados os códigos de resposta do protocolo de agentes.

Os códigos de *status* são extensíveis, mas somente os da classe 6xx não precisam de nenhum tratamento nos ARs e ANS, pois são de escopo, apenas, do módulo de comunicação dos agentes.

4. IMPLEMENTAÇÃO DO AMBIENTE

São apresentadas a seguir os principais componentes do protótipo implementado para concretizar o modelo proposto.

A composição básica do AR é ilustrada na figura 7. Os módulos componentes são:

- gerenciador de conexões – tem, como função geral, o tratamento das conexões provenientes dos agentes;
- roteador - faz o roteamento das mensagens entre os agentes. Este módulo também é encarregado da decodificação dos *headers* das mensagens provenientes dos agentes, além de adicionar os

headers – o de identificação do remetente e o de controle de consistência – às mensagens destinadas aos agentes;

Tabela 1 – Códigos de resposta do protocolo

Código	Significado
000	<i>Unknown error</i> – Erro desconhecido.
200	<i>OK</i> - Sucesso.
400	<i>Bad Request</i> - Erro de sintático na requisição.
401	<i>Unauthorized</i> - Não autorizado (requer autenticação).
404	<i>Not Found</i> – Agente não achado.
408	<i>Request Timeout</i> – Tempo de requisição esgotado .
409	<i>Rejected requisition</i> – Requisição rejeitada.
500	<i>Internal Server Error</i> - Erro interno no servidor.
503	<i>Service Unavailable</i> - Servidor temporariamente sobrecarregado ou em
505	<i>Service Doesn't Exist</i> - Este serviço não existe no domínio requerido.
507	<i>Off Line</i> – Desligado ou fora de linha .
600	<i>Disconnect</i> – Desconectar.
700	<i>Recorder</i> – Cadastrar agente.
701	<i>Remove</i> – Remover agente.
702	<i>Consult</i> – Consultar dados dos agentes.

- *parser* - é responsável pelo *parser* da identificação dos agentes em todo o processo de comunicação (*login*, requisição e comunicação) e também pela verificação de erros sintáticos;
- *gerBD* - faz o gerenciamento do arquivo local dos agentes, onde encontram-se armazenadas as localizações (PATH) dos agentes no disco local, e também do arquivo de *cache*, onde são armazenados os últimos números IPs das máquinas, onde encontram-se os ARs acessados;
- *cliente* – trata-se do módulo responsável pelas conexões *socket* entre ARs, destinadas à troca de mensagens entre os agentes localizados em máquinas diferentes, também pelas conexões com o ANS;
- *segurança* – forma uma espécie de *firewall*, que faz a verificação do endereço IP da máquina de onde se origina a conexão, para determinar, segundo uma lista de acessos, a aceitação ou não desta para a realização do *login*;
- *login* - é destinado ao *login* das conexões dos agentes, a partir da identificação desses, a fim de ser utilizado como informação pelo módulo de roteamento;
- *protocolo* - o tratamento em nível de protocolo é realizado neste módulo;
- *ativador* – encarregado pela ativação do agente;
- *servidor* - é o servidor de conexões responsável pelas conexões *sockets* provenientes dos agentes e pelas transferências para o gerenciador de conexões, com a finalidade de que sejam tratadas.

A constituição interna básica do ANS é, como ilustrada na figura 8, formada pelos seguintes módulos:

- protocolo – módulo responsável pelo tratamento em nível de protocolo;
- gerBD – encarrega-se do gerenciamento do arquivo global de agentes por domínio. Nesse arquivo encontram-se armazenados os endereços IPs das máquinas onde estão localizados os agentes que pertencem ao domínio e suas descrições;
- servidor – é o servidor de conexões *socket*, ativo em uma determinada porta lógica, diferente da dos ARs, aguardando novas conexões;
- cliente – é o módulo responsável pelas consultas ao DNS e também pelas conexões, para requisições a outros ANSs;
- segurança – é responsável pela verificação das permissões dos agentes. O objetivo dessa verificação é a manutenção de uma hierarquia de acessos à resolução de nomes;
- gerenciador – o módulo gerenciador é responsável pela administração da distribuição das tarefas entre os módulos internos.

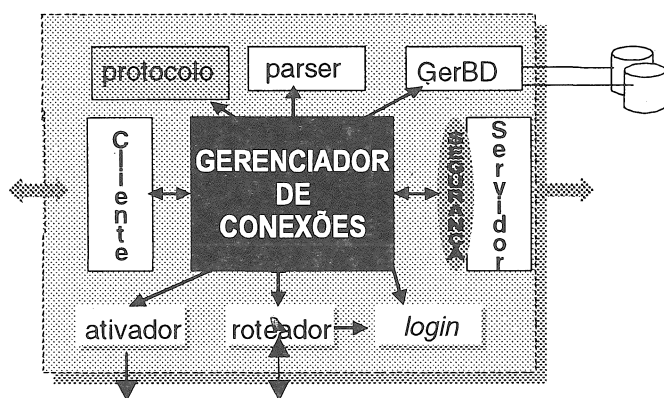


Figura 7 – Componentes do AR

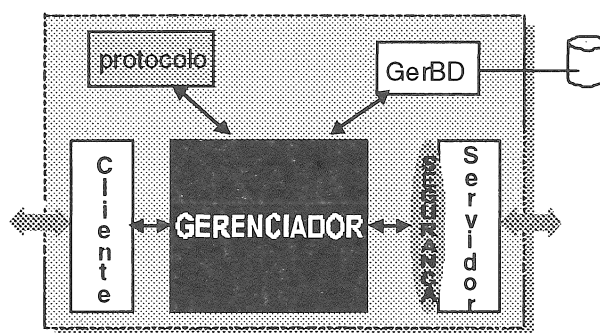


Figura 8 – Componentes do ANS

5. CONCLUSÃO

O resultado deste trabalho, foi a obtenção de um ambiente, robusto no aspecto de suporte ao agente, oferecendo toda uma infra-estrutura de comunicação, permitindo, dessa forma o estabelecimento de comunicação direta entre agentes autônomos, construídos com diferentes linguagens de programação e executados em plataformas heterogêneas.

O ambiente propicia a implementação de sistemas multiagentes no âmbito da Internet, sem que os desenvolvedores de agentes tenham de se preocupar com uma parte dispendiosa: a conexão entre os agentes para a troca de mensagens.

Este ambiente (arquitetura/protocolo) certamente não é único e, possivelmente, não seja o melhor para algumas classes de aplicações, mas reúne pontos que são de grande peso a seu favor, que são o de facilitar a interação de agentes autônomos em um ambiente altamente dinâmico e heterogêneo, como a Internet, e o de possibilitar a utilização de toda a estrutura já definida para a Internet, como é o caso do DNS e dos domínios já existentes.

O ambiente implementado, apresenta significativas similaridades com a especificação proposta posteriormente pela Foundation for Intelligent Physical Agents (FIPA) [FIP 97].

A maior contribuição obtida com este ambiente para a interação de agentes na Internet, acredita-se ter sido, principalmente, na área de Sistemas Multiagentes (SMA), pois, com este

ambiente, torna-se mais fácil a implementação de futuros sistemas multiagentes, sem a preocupação da parte de estabelecimento de conexão entre os agentes.

Uma limitação do modelo proposto é a dependência em relação ao ANS. Essa dependência faz com que haja, em caso de pane na máquina *agent*, onde está instalado o ANS, uma não comunicação entre agentes localizados em máquinas diferentes (desde que o AR local não tenha o endereço de destino em *cache*). Essa limitação pode ser minorada com a utilização de tolerância a falhas a partir da replicação do ANS, por exemplo [FON 99].

Na atual versão do protótipo, implementado em Java devido às características favoráveis como portabilidade, independência de arquitetura, suporte a concorrência (*multithread*), suporte a rede, entre outras, existe a restrição do limite de 50 conexões por porta *socket*, imposta pela versão de Java utilizada.

BIBLIOGRAFIA

- [CAZ 97] CAZELLA, S. C. **Uma arquitetura para coordenar a interação de agentes na Internet**. Porto Alegre: CPGCC da UFRGS, 1997. Dissertação de mestrado.
- [CAZ 97a] CAZELLA S., ALVARES L. **Uma Arquitetura para integrar sociedades de agentes na Internet**. Anais XXIII Conferência Latino-Americana de Informática - CLEI'97, Valparaíso, Chile, novembro 1997.
- [DEM 93] DEMAZEAU, Yves. **Distributed Artificial Intelligence & Multi-Agent Systems**. In: **SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICIAL**, 10., Porto Alegre. Anais... Porto Alegre : SBC, 1993.
- [FIP 97] FIPA - Foundation for Intelligent Physical Agents. **Agent Management**. Disponível por WWW em <http://www.fipa.org/spec/f8a21.doc>.
- [FON 99] FONTES, Roberto D. **Um ambiente para a interação de agentes na Internet**. Porto Alegre: CPGCC da UFRGS, 1999. Dissertação de mestrado.
- [SIC 92] SICHMAN, Jaime Simão; DEMAZEAU, Y.; BOISSIER, Oliver. **When can knowledge-based Systems be Called Agents?**. XII Congresso da Sociedade Brasileira de Computação – IX Simpósio Brasileiro de Inteligência Artificial. Pp 172-185, Rio de Janeiro, 1992.

