SNPUC - UM SERVIDOR DE NOMES PARA A REDPUC*

JOSÉ LAEDIO MEDEIROS** E DANIEL ALBERTO MENASCÉ***

SUMÁRIO

O problema enfocado neste trabalho é a denominação e localização de um ambiente de Redes Locais. Um objeto é qualquer entidade (lógica ou física) do ambiente. Em particular, este artigo apresenta uma especificação de um Servidor de Nomes (SN), desenvolvido para a rede local REDPUC. As principais funções do SN apresentado são: o estabelecimento do mapeamento entre nomes de processos e "portas" do nível de transporte e gerenciamento do espaço de nomes de processos.

ABSTRACT

This work addresses the problem of naming and placement of object in a Local Area Network environment. An object is any logical or physical entity of the environment. In particular, this paper contains a specification of a Name Server (NS) developped for the local network REDPUC. The main functions of this name server are: dynamic mapping between process names and transport level ports, as well as the management of the process name space.

* Trabalho parcialmente financiado pelo Banco do Nordeste do Brasil, pela FINEP e pela EMBRATEL S.A. (contrato C.DCD-002/82)

** M.Sc. (PUC/RJ, 1985); Banco do Nordeste do Brasil, Fortaleza - Ceará.

*** Professor Associado, Depto. de Informática, PUC/RJ; Rua Marques de São Vicente. , 225, CEP - 22453 - Rio de Janeiro - RJ

1. INTRODUÇÃO

Este trabalho enquadra-se no contexto de computação distribuída. O problema enfocado é a denominação e localização de objetos em um ambiente de Redes Locais. Um objeto é qualquer entidade (lógica ou física) do ambiente. Objetos podem ser individuais , tais como terminais, servidor de arquivos, pessoas, processos ou grupos de outros objetos, como por exemplo: uma lista de controle de acesso de usuários [9]. A seção dois apresenta conceitos básicos sobre servidores de nomes e descreve um sistema relacionado com o aqui descrito - o "Clearinghouse" [9]. A seção três descreve o servidor de nomes SNPUC. Nesta seção são apresentadas as primitivas oferecidas pelo SNPUC, uma descrição informal do seu funcionamento bem como o mecanismo de eleição utilizado para tornar a operação do SNPUC robusto em caso de falhas. O artigo conclui com algumas considerações finais sobre o ambiente de implementação.

2. CARACTERIZAÇÃO DE UM SERVIDOR DE NOMES

2.1 - Conceituação de Servidor de Nomes

A nomeação e localização de objetos em ambientes distribuídos deixa de ser trivial na medida em que esses ambientes tornam-se amplos, o que dificulta a identificação dos objetos e acesso a esses. Há portanto, a necessidade da introdução de um espaço de nomes que seja global à rede, ou interrede (no caso de interconexão de redes), e algo que transforme este espaço de nomes em endereços.

De uma maneira restrita, um Servidor de Nomes (SN) seria um agente capaz de oferecer uma forma padronizada de atribuição de nomes a objetos, bem como permitir que esses objetos sejam referenciados simplesmente pelos seus nomes, independente de suas localizações (endereços) no ambiente distribuído. De outra forma este conceito pode ser estendido no sentido de atribuir ao SN os serviços de mapear um espaço de nomes em um espaço de propriedades, dentre as quais uma seria o endereço. Ou seja, cada objeto poderia ter associado a seu nome, um conjunto de propriedades. Propriedades, estas caracterizadas como atributos referentes aos nomes.

Para efeito desse trabalho, definimos o SN como um agente responsável pelo fornecimento de primitivas que permitem, além da nomeação padronizada de nomes na interrede, e de sua principal função que é a associação dinâmica entre nomes de processos e "portas" no ambiente distirbuido, a administração tanto do espaço de nomes como do espaço de endereços, com o uso de um Banco de Dados distribuido.

Na seção seguinte, está descrito o "Clearinghouse" - um servidor de nomes desenvolvido pela Xerox [9].

O "Clearinghouse" é um agente descentralizado que tem como funções básicas nomear e localizar objetos em um ambiente distribuido. Utiliza-se de um Banco de Dados distribuido e replicado em alguns locais da interrede. A descentralização e réplica de informações, aumentam a eficiência quando de acessos a objetos, e a robustez do sistema no caso de falhas. Além dessas funções básicas, oferece facilidades para a administração do Banco de Dados e mecanismo de controle de acesso.

2.2.1 - Convenções de Nomes

Qualquer tipo de objeto é nomeado obedecendo a uma mesma convenção de nomes. Um nome é uma cadeia, não nula, de caracteres da forma <sub-cadeia1>@<sub-cadeia2>@<sub-cadeia3>, onde sub-cadeia1 é o nome local (L), sub-cadeia2 é o domínio (D) ao qual pertence o objeto denominado, e sub-cadeia3 identifica a organização (O). Desse modo temos L@D@O. Nenhuma das sub-cadeias pode conter ocorrências do caracter "@". A forma do nome com três cadeias, caracteriza uma hierarquia de três níveis, entretanto esta divisão é apenas lógica, pois os nomes são tratados com uma única cadeia. Os nomes são absolutos, ou seja, não existem dois objetos com o mesmo nome na interrede. Por exemplo, os nomes de objetos da rede PUC (organização), departamento de informática (domínio), seriam da forma <qualquer coisa>@info@puc. Um objeto, além do seu nome absoluto, pode ter um ou mais "aliases". "Aliases" são também absolutos. Um nome, ou é um nome distinto ou é um "alias", mas nunca ambos.

2.2.2 - Mapeamento

O "Clearinghouse" mapeia cada nome em um conjunto de propriedades associadas ao objeto que possui aquele nome. Uma propriedade é uma tupla ordenada, consistindo do nome da propriedade (nome-prop), um tipo de propriedade (tipo-prop) e o valor da propriedade (valor-prop). O "Clearinghouse" mantém mapeamentos da forma: {<nome-prop, tipo-prop1, valor-prop1>,...<nome-propk, tipo-propk, valor-propk>}, onde "k" pode ser qualquer.

Um "nome-prop" identifica uma propriedade especifica associada a um dado nome. Um "nome-prop" pertencente ao conjunto de propriedades associadas a um objeto é único no conjunto, isto é, nome-prop(i) $\neq$ nome-prop(j), se i $\neq$ j.

Um "valor-prop" é o valor atribuido a uma determinada propriedade. Há dois tipos de "tipo-prop". Tipo individual ou $\emptyset$, nesse caso "valor-prop" é tratado como uma cadeia de "bits" sem qualquer significação; e o tipo grupo ou 1, onde o "valor-prop" é tratado como um conjunto de nomes. Essa interpretação dada pelo "Clearinghouse" se refere à propriedade do nome, e não ao objeto que a possue. A seguir são mostrados alguns e

xemplos de mapeamento:

. tipo individual
Laedio@info@puc --> {<"usuário", Ø, qualquer coisa>,
<"senha", Ø, de autenticação>,
<"arquivos", 1, conjunto de arquivos do usuário>}

. tipo grupo
Alunos@info@puc --> {<"lista", 1,{"laedio@info@puc",
"nelson@info@puc",
... "ricardo"@info@puc}>}

O mapeamento de um nome em um conjunto de propriedade permite que diferentes usuários atuem sobre o mesmo "nome" simultaneamente, porém em propriedades diferentes. Da mesma forma, as propriedades do tipo grupo, que são associadas a um conjunto de nomes em "valor-prop", permite que vários usuários atuem sobre a mesma propriedade do mesmo "nome", em valores de propriedade distintos.

2.2.3 - Primitivas principais oferecidas aos usuários

2.2.3.1 - Sobre "nomes" e "aliases"

. adição de "nomes" ou "aliases";
. retirada de "nomes" ou "aliases";
. verificação da existência de "nomes" ou "aliases";
. troca de "nomes".

2.2.3.2 - Relativas a propriedades dos nomes

. verificação da existência de "propriedades" de "nomes";
. adição de "propriedades" a "nomes";
. retirada de "propriedades" de "nomes";
. troca "propriedades" de "nomes".

2.2.3.3 - Relativas a elementos de grupos de uma propriedade

. verificação da existência de "elementos" em grupos de propriedades;
. adição de "elementos" a grupos de propriedades;
. retirada de "elementos" de grupos de propriedades.

2.2.3.4 - Outras primitivas

. verificação genérica de "nomes";
. enumeração dos objetos de um domínio dado;
. enumeração dos domínios de uma organização dada;
. enumeração das organizações existentes na interrede;
. enumeração das propriedades de um "nome" dado.

2.2.4 - Estrutura do "Clearinghouse":

O "Clearinghouse" é composto pelo conjunto dos servidores distribuídos na interrede, os quais executam os serviços oferecidos aos usuários, ou pertinentes às tarefas in

rentes ao próprio "Clearinghouse". Além disso, oferece um conjunto básico de operações, que devem ser incorporados ao "software" do usuário, a fim de padronizar a comunicação entre o usuário e o "Clearinghouse".

Correspondendo a cada domínio (D), em cada organização (O), existe um ou mais servidores, cada um contendo uma cópia de todos os mapeamentos da forma <qualquer nome>@D@O. Estes servidores são chamados de Servidor Domínio (SD). Há pelo menos um SD para cada domínio no ambiente distribuido. Cada SD em uma organização (O), tem o nome da forma <qualquer nome>@O@servidor, mapeando o endereço de rede do servidor : <qualquer nome>@O@servidor --> {... <"endereço", Ø, endereço de rede> ...}.

Os mapeamentos de todos os SD's (D), de uma organização (O) (<qualquer nome>@O@servidor), estão contidos na porção do BD mantido por um servidor chamado de Servidor Organização (SO). Há no mínimo um servidor deste tipo para cada organização (O). Cada SO tem o nome da forma <qualquer nome>@servidor@servidor mapeando-o no seu endereço de rede: <qualquer nome>@servidor@servidor --> {...<"endereço", Ø, endereço de rede>...}.

2.2.4.1 - Interconexão entre os componentes do "Clearinghouse"

O conjunto de primitivas incorporadas ao "software" do usuário, contêm o endereço de no mínimo um servidor "Clearinghouse", ou sempre pode localizar um localmente ou em outro local da interrede, através de uma mensagem do tipo difusão.

Cada servidor "Clearinghouse" (SC) conhece o nome e endereço de cada SC na interrede. Desse modo, qualquer SC pode comunicar-se com qualquer SO da interrede diretamente.

Em qualquer solicitação do usuário ao "Clearinghouse", mesmo o pior caso, o número de SC's contactados não excede a três: servidor cujo endereço é conhecido, um SO que contêm a organização do "nome", e um SD dentro da organização.

Esta estrutura permite o uso de um algoritmo relativamente simples para o gerenciamento do "Clearinghouse". Entretanto requer que cópias de mapeamentos de todos os "nomes" da forma <qualquer coisa>@servidor@servidor (domínio organização) sejam armazenados em todos os servidores.

2.2.5 - Administração do "Clearinghouse"

As tarefas administrativas do "Clearinghouse" incluem os serviços de gerenciamento do espaço de nomes e de nomes de propriedades; inclusão de novas redes; decisão sobre como distribuir uma organização e objetos dentro dos domínios; adição, troca e retirada de serviços (tais como serviços de correio, servidores de arquivos, e os serviços próprios do "Clearinghouse"); adição e retirada de usuários; manutenção de

"senhas" dos usuários, etc); e a manutenção de uma lista de controle de acesso e outras características de segurança da rede.

A seguir estão descritas, sucintamente, algumas das tarefas do "Clearinghouse". É assumido que o "Clearinghouse" já está instalado na interrede.

2.2.5.1 - Gerenciamento do Espaço de Nomes

A alocação de nomes é gerenciada por uma autoridade de nomeação, a qual é responsável por garantir a existência de nomes absolutos. Há autoridades de nomeação para nomes de objetos, nomes de domínio, nomes de organização e nomes de propriedades.

2.2.5.2 - Adição de um novo usuário

O administrador do sistema, primeiro tenta incluir o novo usuário, através da chamada da rotina ADICIONA NOME (se não conseguir devido ao "nome" já existir, possibilita ao usuário substituir o "nome" dado). Em seguida, de acordo com as informações prestadas pelo novo usuário, registra sua "senha", seus privilégios, etc.

2.2.5.3 - Adição de um novo servidor (do usuário)

O administrador do sistema solicita ao componente do "Clearinghouse" existente no "software" do usuário, que este registre o "servidor" no SD apropriado. O registro , envolve no mínimo o nome do servidor, sua "senha" e seu endereço de rede. (São usadas para isso as funções ADICIONA NOME e ADICIONA NOME de PROPRIEDADE).

2.2.5.4 - Adição de novo servidor do "Clearinghouse"

O servidor a ser adicionado se enquadra em um dos seguintes casos: 1) é um SD para um novo domínio; 2) é um SD para um domínio existente; e 3) é um SD para uma nova organização. Assumindo que não existe conflito na nomeação, e que o " solicitante" tem permissão para requisitar tal serviço, teriamos de forma simplista os seguintes passos: 1) é atribuido um nome ao novo servidor; 2) se o novo servidor é de uma nova organização, o nome da nova organização é registrado, através da autoridade central de nomeação; 3) se o servidor é um novo domínio, o nome do domínio é registrado adequadamente; 4) o administrador decide que porção do BD do "Clearinghouse deve o novo servidor manter; 5) registra os dados que compõem o novo domínio (nomes, endereços, "senhas", etc. através das primitivas tais como ADICIONA NOME, ADICIONA PROPRIEDADES de NOMES, etc.); 6) solicita a todos os SC's, a inclusão do novo "servidor".

O "Clearinghouse" pode ser visto com mais detalhes na referência [9].

3.1 - Descrição do Servidor de Nomes (SNPUC)

O SNPUC situa-se no nível imediatamente acima do Protocolo de Transporte (nível 4)

O SNPUC é um agente descentralizado que tem como função básica localizar "processos", através de seus nomes no ambiente distribuido, e transformar esse espaço de nomes em um espaço de endereços. O espaço de nomes é composto por todos os nomes de processos conhecidos na interrede (que estão habilitados a uma conexão do nível de transporte ou já estão ativos em conexões já estabelecidas). Dessa forma as solicitações dos usuários não se restringem a informações disponíveis localmente, e sim, são estendidas ao ambiente das redes interligadas. Independente de sua localização, o mapeamento de nomes em endereços é transparente para os solicitantes dos serviços oferecidos pelo SNPUC.

O SNPUC considera objetos (nomes de processos) pertencentes a dois tipos de domínios:

tipo usuário (rede) - que engloba os processos pertencentes às redes locais

tipo servidor - que compõem os processos executores das funções do SNPUC.

3.1.1 - Convenções de Nomes

Todos os processos são nomeados obedecendo a uma única convenção. Um nome é uma cadeia, não nula, de caracteres da forma: sub-cadeia1/sub-cadeia2, onde sub-cadeia1 identifica o domínio (rede) ao qual pertence o processo denominado e sub-cadeia2 é um nome local (dentro da rede).

Os nomes são absolutos, ou seja, não existem dois processos com o mesmo nome dentro da interrede. Por exemplo, os nomes de processos da REDPUC são da forma: PUC/qualquer-coisa.

Nenhuma das duas sub-cadeias pode conter ocorrências do caracter "/".

A forma do nome com duas sub-cadeias caracteriza uma hierarquia de dois níveis, entretanto esta divisão é apenas lógica pois os nomes são tratados como uma única cadeia.

Para nomeação de servidores e interfaces de servidores é utilizada a mesma filosofia explicitada acima, sendo observado o seguinte: sub-cadeia1 = identificação de redes e sub-cadeia2 = identificação do servidor. Por exemplo, o servidor de nomes local da REDPUC é da forma: PUC/SNL.

3.1.2 - Descrição da arquitetura do SNPUC

O SNPUC é composto pelo conjunto dos Servidores de Nomes Locais - SNL's (um para cada rede), e de todas as Interfaces dos SNL's - ISNL's (uma para cada estação de cada rede).

Cada SNL tem acesso ao Banco de Dados da Rede (BDR) que contém:

i) mapeamento para o subconjunto de nomes que foram o domínio de rede local:

Ri/qq.nome -> { endereço do processo na rede local }

ii) mapeamento para o domínio servidores da interrede e interfaces da rede local:

Ri/Si -> { endereço do servidor na interrede }
Ri/ISi -> { endereço da interface na rede }

Cada ISNL tem acesso ao Banco de Dados da Estação (BDE) que contém:

i) mapeamento para o subconjunto dos usuários da estação do domínio rede local:

Ri/qq.nome -> { endereço da estação }

ii) mapeamento para o domínio Servidores do SNPUC e interfaces da rede local:

Ri/Si -> { endereço do servidor na interrede }
Ri/ISi -> { endereço da interface na rede }

A seguir (na fig. 3.1) é mostrada graficamente a arquitetura do SNPUC.

3.1.3 - Serviços Oferecidos pelo SNPUC aos usuários

1. ADIciona NOME	(ADINOME)
2. RETira NOME	(RETNOME)
3. MUDa NOME	(MUDNOME)
4. VERifica NOME	(VERNOME)
5. ENUmera NOMEs	(ENUNOME)
6. MUDa ENDEreço do nome	(MUDENDE)
7. CONEXAO dinâmica entre nomes e portas	(CONEXAO)
8. ENUmera SERVidores	(ENUSERV)

3.1.3.1 - Especificação das Primitivas

1. ADINOME
ENTRADAS : nome (de acordo com a convenção de nomes)
SAIDAS : cret (código de retorno)
DESCRIÇÃO : Caso ainda não exista o nome na rede local, referido nome (se BDR e BDE tiver espaço) é adicionado ao domínio de rede correspondente, tanto no BDR como no BDE (da estação do processo solicitante), com mapeamento para o endereço de origem do pedido (endereço do ISNL de origem)
Ri/nome dado -> { endereço do ISNL de origem }.

CÓDIGOS DE RETORNO :
Ø - INSERÇÃO OK;
1 - "nome" JÁ EXISTE;
2 - BD SEM ESPAÇO;
3 - REDE DO "nome" NÃO É LOCAL.

2. RETNOME
ENTRADAS : nome
SAIDAS : cret (código de retorno)
DESCRIÇÃO : caso "nome" dado exista na estação local, ele é retirado dos BDR e BDE nos quais está armazenado.

CÓDIGO DE RETORNO :
Ø - RETIRADA OK;
1 - "nome" NÃO EXISTE.

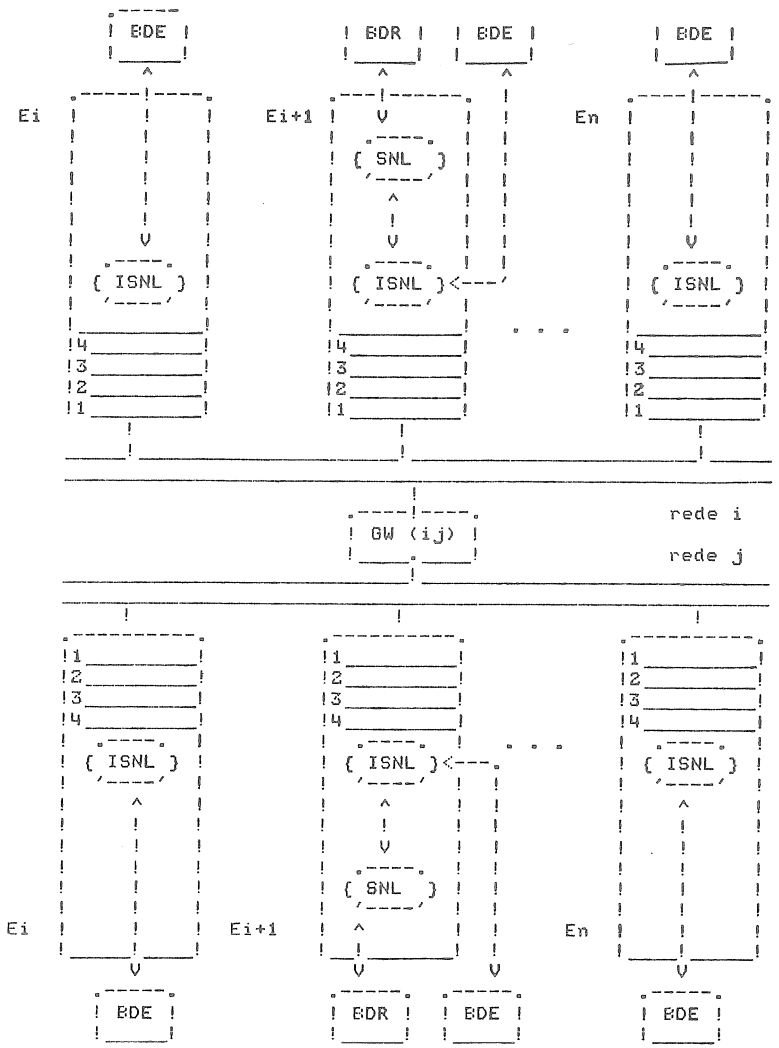

fig. 3.1 - arquitetura do SNPUC na interrede

LEGENDA: BDE- banco de dados da estação
BDR- banco de dados da rede (cada
SNL- servidor de nomes local (cada rede)
ISNL- interface do SNL (cada estação)
Ei - estação(i)
GW(i,j) - gateway de ligação da rede(i) com a rede(j)
1 - nível físico
2 - nível de acesso a barra
3 - nível interredes
4 - nível de transporte

3. MUDNOME

ENTRADAS : nome-novo
nome-velho

SAÍDAS : cret (código de retorno)

DESCRIÇÃO : caso "nome-velho exista na estação de origem da solicitação, "nome-velho e ainda não exista, a troca é realizada (no BDR e BDE)

CÓDIGO DE RETORNO : Ø - TROCA DE "nome" OK;
1 - "nome-novo" NÃO ACEITO;
2 - "nome-velho" NÃO EXISTE;
3 - "nome-novo" JÁ EXISTE.

4. VERNOME

ENTRADAS : nome

SAÍDAS : ender-nome (rede, módulo)
cret (código de retorno)

DESCRIÇÃO : caso "nome" exista na interrede (no BDE, ou no BDR local, ou no BDR de outra rede) é retornado o endereço correspondente, no qual o "nome" está mapeado (rede, módulo).

CÓDIGO DE RETORNO : Ø - "nome" EXISTE;
1 - "nome" NÃO EXISTE.

5. ENUNOME

ENTRADAS : tipo-enumeração
1 - para enumeração de todos os nomes cadastrados no módulo de origem do pedido;
2 - para enumeração de todos os nomes cadastrados na rede local.

SAÍDAS : nome(i), ender-nome(i) (relação dos nomes e respectivos endereços de nome(n), ender-nome(n) acordo com o tipo 1 ou 2)
...

DESCRIÇÃO : caso tipo-enumeração seja:
1 - são retornados ao processo solicitante todos os nomes e respectivos endereços (do domínio usuário) cadastrados no BDE local da ISNL de origem;
2 - são retornados ao processo solicitante todos os nomes e respectivos endereços (do domínio usuário) cadastrados no BDR da rede local.

6. MUDENDE

ENTRADAS : nome (a ser transporado)
ender-nome (novo módulo do nome)

SAÍDAS : cret (código de retorno)

DESCRIÇÃO : caso "nome" exista na rede local e "novo-módulo" exista e esteja ativo, o mapeamento no BDR é alterado para o novo endereço, sendo o "nome" retirado do BDE do módulo de origem e adicionado no BDE no novo módulo.

CÓDIGO DE RETRONO : Ø - MUDANÇA DE ENDEREÇO OK;
1 - "nome" NÃO EXISTE;
2 - MÓDULO DO "ender-nome" NÃO EXISTE ou INATIVO;
3 - BDE DO "novo-módulo" SEM ESPAÇO.

7. CONEXÃO

ENTRADAS nome1 (nome do processo que solicita a conexão)
nome2 (nome do processo com o qual o nome1 deseja conectar-se)

SAÍDAS portal (porta local do processo solicitante)

porta2 (porta local do outro processo)
cret (código de retorno)

DESCRIÇÃO : caso exista porta disponível tanto no módulo ISNL de origem, como na estação da interface onde "nome" está armazenado (se "nome2" existir) e "nome1" já esteja cadastrado no BDE local e BDR da rede, ou possa ser cadastrado com sucesso nos BD's, são alocadas duas portas, uma no módulo local da ISNL de origem (associada ao processo de "nome1"), e outra na estação do BDE do "nome" (associada ao processo de "nome2"). Os números das portas alocadas são retornadas ao processo solicitante.

CÓDIGO DE RETORNO : 0 - "porta1" e "porta2" ALOCADAS OK;
1 - NÃO EXISTE "PORTA" DISPONÍVEL;
2 - "nome1" NÃO PODE SER CADASTRADO;
3 - "nome2" NÃO EXISTE NA INTERREDE;
4 - MÓDULO DO "nome2" NÃO EXISTE ou NÃO ESTA ATIVO.

8. ENUSERV

ENTRADAS : nenhuma
: nome(i), ender(i) (relação dos servidores e respectivos endereços
... na interrede)
nome(n), ender(n)

DESCRIÇÃO : É feita uma solicitação ao BDE da ISNL ao qual foi feito o pedido e devolvidos todos os nomes e endereços de servidores cadastrados no domínio servi - dor.

3.1.4 - Serviços necessários para funcionamento do BDR de cada Rede Local

1 - CRIação do BDR (CRIABDR)
2 - RECuperação do BDR (RECBDR)

3.1.4.1 - Especificação das Primitivas

1. Criação do BDR

ENTRADAS : nenhuma
SAÍDAS : cret (código de retorno)
DESCRIÇÃO : Caso ainda não exista o arquivo (ou arquivos) que compõem o BDR da rede local no disco onde o SNL foi ativado e haja espaço em disco, o SNL cria um BDR local (BDRrede.SNL) com o domínio servidor (servidores da interrede), envia o domínio servidor às ISNL's da rede local e solicita aos SNL's das redes interligadas a inclusão do servidor atiado;
Caso já exista o arquivo BDrede.SNL na estação do SNL, o SNL considera o arquivo como válido, atualiza o domínio servidor (servidores da interrede) e complementa seus serviços da mesma forma do primeiro caso.

CÓDIGO DE RETORNO : 0 - "BDR" CRIADO COM SUCESSO;
1 - DISCO SEM ESPAÇO PARA O "BDR".

2. Recuperação do Servidor

ENTRADAS : nenhuma
SAÍDAS : cret (código de retorno)
DESCRIÇÃO : Caso haja espaço no disco onde o SNL foi ativado, o SNL cria um BDR (BDRrede.SNL), recupera o domínio servidor (servidores da interrede) e domínio usuário (processo da rede local), informa aos SNL's das redes interligadas o novo endereço do SNL recuperado.

CÓDIGO DE RETORNO : 0 - "BDR" RECUPERADO COM SUCESSO;

1 - DISCO SEM ESPEAÇO RARA O "BDR".

3.2. Descrição Informal do Funcionamento do SNPUC

Para que os nomes de processos constem do espaço de nomes do SNPUC (pertençam a um BDR e a um BDE) existem apenas duas formas:

1) através da solicitação explícita de inclusão de nomes de processos nos BD's do SNPUC com o uso da primitiva ADINOME;
2) através da utilização da primitiva CONEXÃO (usada para a associação dinâmica entre nomes de processos e portas, a qual possibilita a comunicação entre dois processos na interrede), nesse caso o processo solicitante do serviço, se ainda não constar do espaço de nomes do SNPUC é devidamente cadastrado pela SNL de origem.

Todas as solicitações ao SNPUC (interrede) devem ser endereçadas à ISNL da estação de origem do usuário (endereço bem conhecido), que devolve: ou o resultado do pedido ou o fracasso da operação solicitada.

As interfaces (ISNL's) são ativadas quando da entrada em operação das estações de cada rede local, começando assim os serviços do SNPUC.

A primeira tarefa de cada ISNL, após criar ou ativar o BDE, é o envio de uma mensagem (do tipo difusão) para o SNL da rede local, solicitando-lhe seu endereço. Caso o SNL esteja ativo e devolva seu endereço, a ISNL solicitante insere, no BDE local, o endereço do SNL ativo e considera-se apta a oferecer as primitivas do SNPUC. Caso não receba nenhuma resposta em um tempo "T" segundos (TIMEOUT), assume que o SNL local ainda não está ativo e aciona o algoritmo de eleição da estação que deve ativar o SNL da rede (vide 3.3.1). Caso a estação seja a sua, a interface deve ativar o SNL solicitando-lhe que crie o BDR (CRIABDR). Do mesmo modo procede quando é reativada devido à recuperação de falha de "hardware" do módulo onde ela estava em operação.

Caso o módulo onde o SNL de uma determinada rede pare de funcionar, o SNL deve ser recuperado em outra estação da rede, sendo usado para tal o mesmo algoritmo de eleição (vide 3.2.1). Nesse caso, a ISNL do módulo eleito ativa o SNL e solicita-lhe a recuperação do BDR (RECBDR).

Para a execução das primitivas oferecidas aos usuários e dos serviços inerentes ao protocolo do SNPUC é usada de uma forma geral o esquema de comunicação entre os componentes do SN, mostrado na fig. 3.2.

AÇÕES DE CADA ENTIDADE :

1. USUÁRIOS
   . Solicitam a execução de primitivas à ISNL da estação local;
   . Recebem respostas das solicitações.

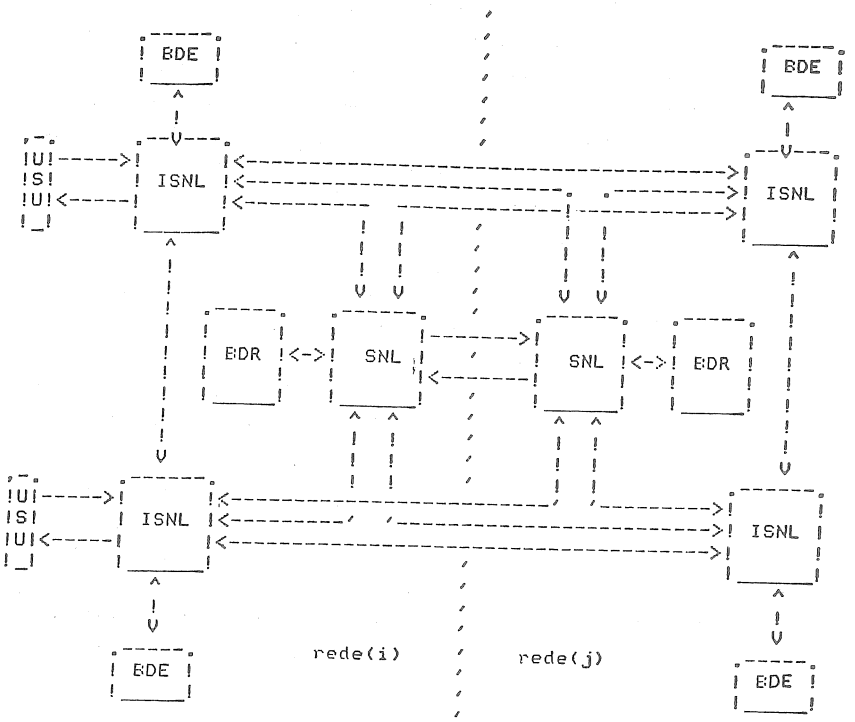

fig. 3.2 - esquema geral de comunicação do SNPUC

2. ISNL's
   . Recebem solicitações, ou dos usuários ou de outro componente do SNPUCC;
   . Consultam o BDE e executam um dos passos seguintes:
   1) caso atenda à solicitação localmente, ou detete qualquer situação de insucesso, devolve o solicitante a resposta adequada; fim;
   2) caso contrário, solicita ao SNL (da rede local ou de outra rede) o serviço; recebe a resposta da sua solicitação, ou do SNL ao qual enviou a requisição ou da ISNL do módulo endereçado pelo SNL requisitado; devolve a resposta adequada ao solicitante; fim;

3. SNL's
   . Recebem solicitações das ISNL's locais ou das SNL's (outra rede) e dependendo da solicitação, executa um dos passos seguintes:
   1) consulta o BDR (local) e devolve resposta ao INSL/SNL; fim.
   2) consulta o BDR e envia solicitação ao ISNL (de destino da rede local); fim.
   3) consulta o BDR e envia solicitação ao SNL de outra rede; fim.

A fim de um melhor entendimento, suponhamos que o processo "X", do módoulo "Mi" da rede "I", deseja se comunicar com o processo "Y", do módulo "Mj" da rede "J", e que o processo "X" já está cadastrado no BDE e BDR da rede "I" e o processo "Y" já se encontra no BDE de "Mj" e BDR da rede "j", além de existirem portas disponíveis em ambos os módulos, P1 e P2 respectivamente. O processo "X" se utiliza da primitiva "CONEXAO" passando os parâmetros <nome1> = X, <nome2> = Y e recebe de volta <cret = Ok> , <portal> = P1, e <porta2> = P2. Teríamos então o quadro mostrado na fig. 3.3

Na execução de suas primitivas o SNPUC utiliza-se do protocolo interredes (PI) e as ações realizadas são as seguintes:

1 - a interface do módulo de origem do processo "X" ao receber a solicitação CONEXAO (X,Y), verifica (através do PT) se existe porta disponível (hipótese=existe), consulta o BDE para ver se "X" já é cadastrado (hipótese = é cadastrado) ; em seguida envia uma mensagem ao SNL da rede "J" (endereço bem conhecido) <IP , os, id-ISNL, Y>;

2 - o SNL da rede "J" verifica-se "Y" existe na rede local (hipótese = existe) e envia ao ISNL do módulo onde "Y" consta do BDE local, uma mensagem solicitando alocação de porta para o processo "Y" <IP, os, id-ISNL, Y >;

3 - a ISNL (módulo de endereço de Y), verifica (através do PT) se existe porta disponível (hipótese = existe) e devolve a ISNL de origem de "X" a mensagem <IP,os,P2> informando a porta alocada, no caso P2;

4 - a ISNL de origem ao receber a mensagem da ISNL da outra rede, devolve ao usuário <OK,P1,P2> significando que houve sucesso na execução da primitiva e informando

quais portas foram alocadas.

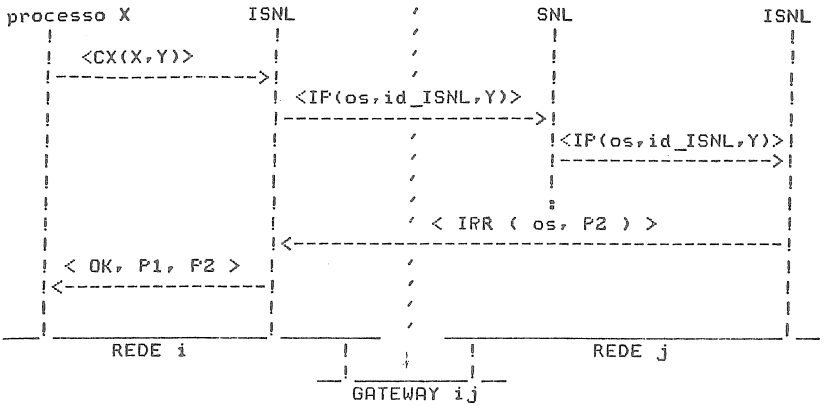

fig. 3.3. fluxo de mensagens trocadas na execução da primitiva CONEXAO

Legenda :

CX - CONEXÃO
IP - identificação da primitiva
IPR - identificação da primitiva resposta
os - ordem de solicitação
id-ISNL - identificação da ISNL requisitante
P1 - número da porta 1
P2 - número da porta 2

3.2.1 - Descrição informal do Algoritmo de Eleição (AE) da estação que deve ativar/recuperar o SNL em cada Rede Local

Os módulos elegíveis são aqueles ativos da Rede Local. São assumidas as seguintes hipóteses:

1. todas as estações executam o mesmo AE;
2. todas as estações tem implementados todos os níveis de comunicação: nível físico, nível de acesso a barra, protocolo interredes e protocolo de transportes;
3. nas transmissões de mensagens é usado o serviço de conexões confiáveis e com controle de erros fim-a-fim do PT [4].

Cada ISNL deve manter um vetor de estados que espelhe seu estado de operação:

E(i) = "reativação"
(quando da recuperação do módulo que estava falho).

E(i) = "eleição"
(quando estiver sendo executado o AE)

E(i) = "criação/recuperação"
(quando o BDR estiver sendo criado/recuperado)

E(i) = "normal"
(quando pronto para executar as primitivas oferecidas aos usuários)

Ender-SNL = endereço do SNL local (ativo ou falho)

ISNL-eleita = identificação da ISNL que deve recuperar o SNL/BDR.

Em qualquer momento as ISNL's, ou têm o endereço do SNL ativo, ou supostamente falho.

O SNL é considerado falho por uma ISNL quando:

1) a própria ISNL detetou a falha do SNL após ter enviado uma mensagem e não ter enviado uma mensagem e não ter conseguido resposta em um tempo determinado (TIMEOUT) ;
2) quando tiver recebido uma mensagem de um SNL ou ISNL de outra rede(1) avisando que o SNL local está inativo;
3) quando tiver recebido uma mensagem de uma das ISNL's da rede local tentando saber se as ISNL's de mais alta ordem estão ativas.
4) quando tiver recebido uma mensagem de uma SNL de maior prioridade informando que ela se considera eleita para recuperar o SNL/BDR rede local.

ASSERÇÃO 1

Num dado instante os vetores de estados das ISNL's podem estar configurados em uma das formas abaixo:

1. a) E(i) = "eleição" e Ender-SNL = "indefinido"
   b) outras ISNL's:
      E(j) = "normal" e
      Ender-SNL = "endereço do SNL falho"
      ou E(j) = "eleição" e
      Ender-SNL = "indefinido" ;

2. b) E(i) = "criação/recuperação" e
   Ender-SNL = "indefinido"
   outras ISNL's:
      E(j) = "eleição" e Ender-SNL = "indefinido",
      ou E(j) = "criação/recuperação" e
      Ender-SNL = "indefinido";

3. c) E(i) = "normal e Ender-SNL = "endereço SNL ativo"

outras SNL's:
E(j) = "criação/recuperação" e
Ender-SNL = "indefinido",
ou E(j) = "normal" e Ender-SNL = "endereço do SNL ativo";

Em qualquer dos casos em que o SNL for detetado como falho, a ISNL(i) muda seu estado de "normal" para "eleição" (E(i) = "eleição") e dispara a execução do AE.

Ao final da execução do AE somente uma ISNL será eleita para ativar/recuperar o SNL/BDR na rede local. Se por ventura houver falha da estação(2) do ISNL eleita, o AE deve ser executado novamente por todas as ISNL's locais ativas. Do mesmo modo faz-se necessário quando o BDR não consegue ser criado/recuperado por falta de espaço em disco na estação da ISNL escolhida(3).

ASSERÇÃO 2

Após o AE ter sido concluído com sucesso todas as ISNL's locais ativas estarão com seus estados em "normal" e com endereço da SNL igual ao endereço da ISNL responsável pela ativação/recuperação do SNL, ou seja:

a) E(i) = "normal" e Ender-SNL = "Ender-ISNL-eleita";
b) todas as outras ISNL's:
E(j) = "normal" e Ender-SNL = "Ender-SNL-eleita".

Vejamos agora a descrição estrita do AE que tem como base as referências [6,7].

O critério básico para que uma ISNL seja eleita será vinculado à sua prioridade em relação às outras ISNL's da rede na qual o SNL deve ser ativado/recuperado. Prioridade essa que é estabelecida em função da identificação das estações dentro da rede. Cada módulo no ambiente da rede local tem um número único que o distingue dos demais, variando de 1 a "n", onde "n" é o número máximo de nodos na rede. O de mais alta prioridade é exatamente o módulo "n". A ISNL eleita deve ser aquela do módulo de maior numeração dentre os que estiverem ativos no momento da execução do AE.

Quando uma ISNL(i) descobre falha do SNL (seja qual for a forma - vide considerações anteriores) ela aciona o protocolo de eleição objetivando eleger a si própria, ou outra ISNL, como componente do SN a ativar/recuperar o SNL/BDR na respectiva estação.

O protocolo de eleição executado por qualquer ISNL(i) é dividido em duas partes. Primeiro, a ISNL(i) tenta contatar com todas as outras ISNL's de maior prioridade (módulos de numeração maior que a sua). Se pelo menos uma das ISNL's recebe a mensagem en-

(1) sempre que um componente do SN (SNL ou ISNL), em qualquer rede deteta a falha do SNL de outra rede, este envia uma mensagem para o grupo das ISNL's da rede do SNL falho avisando a queda do mesmo.

(2) fato detetado por esgotamento de temporização.

(3) situação detetada pela ISNL que solicita a ativação/recuperação do SNL/BDR.

viada, atestando que está ativa, a ISNL(i) coloca-se na situação de espera até que uma ISNL de prioridade mais alta torne-se a eleita. Entretanto, passada em espera um certo limite de tempo pré-estabelecido (TIMEOUT), sem que receba alguma mensagem sobre a ISNL eleita, a ISNL(i) reaciona o protocolo de eleição desde o início. De outro modo, se nenhuma das ISNL's de maior prioridade responderem num limite de "T" segundos, a ISNL(i) assegura-se que todas elas estão falhas e considera-se como componente eleito.

Após o cumprimento da primeira etapa, a interface escolhida envia mensagem a cada uma das ISNL's de prioridade mais baixa que a sua, a fim de informar que ela foi a eleita. Ativa o SNL localmente e solicita-lhe, se for a primeira vez (através da primitiva CRIA-BDR) que crie o BDR da rede, ou se por motivo de falha (através da primitiva REC BDR) que recupere o BDR da rede. Dependendo do resultado da execução das primitivas solicitadas, a ISNL(i) envia novamente mensagem às ISNL's de menor prioridade informando se o SNL/BDR foi, ou não, criado/recuperado com sucesso.

As ISNL's de menor prioridade que a ISNL (eleita) ao receberem a mensagem de eleição da interface eleita, mudam seus "estados" de "eleição" para "criação/recuperação" (do SNL/BDR) e aguardam até o recebimento da mensagem informando que o SNL/BDR foi criado/recuperado normalmente, quando só então trocam seus "estados" para "normal" (E(j)= = "normal"). Se após um tempo determinado (TIMEOUT) não receberem tal mensagem, reiniciam o protocolo de eleição. Da mesma forma se receberem uma mensagem da ISNL eleita informando que não existe espaço em disco suficiente para recuperar o BDR, reativam o AE.

4. CONSIDERAÇÕES FINAIS

Maiores detalhes sobre o Servidor de Nomes aqui apresentado, podem ser encontrados na referência [10]. Além dos tópicos tratados neste artigo, estão especificados, em [10], a sintaxe e semântica do protocolo do SN, bem como os aspectos relativos à implementação.

O SN objeto deste trabalho foi desenvolvido na linguagem Pascal MT Plus, no ambiente da REDPUC [1 a 4], entretanto tem portabilidade para ser transportado para outras redes locais. Constitue-se como pioneiro aqui no Brasil e é flexível no sentido de suportar novos serviços para melhor atender às necessidades dos seus usuários.

BIBLIOGRAFIA:

1. GOMES SOARES, L.F.; MENASCÉ,D.A. e outros, "Descrição de Hardware e Software da Rede Local, REDPUC", Anais do XV Congresso Nacional de Informática, Rio de Janeiro, outubro 1982.

2. GOMES SOARES, L.F.; MENASCÉ,D.A., "Um protocolo para redes locais do tipo difusão", Anais do IX Seminário Integrado de Software e Hardware, julho 1982.

3. GOMES SOARES, IERUSALIMSCHY, R., "Especificação de Protocolos através de redes de Petri com temporização - O Protocolo de Acesso a Barra da Rede Local - REDPUC", RBC, Rio de Janeiro, Vol. 3, Nº 3, 1983/1984, pp. 171-193.

4. ZÚNICA DE SPIRITO,P.A., "Um ambiente para implementação de serviços em uma rede local de computadores", Dissertação de mestrado, Departamento de Informática, PUC/RJ, 1984.

5. MENASCÉ, D.A., "Especificação de um protocolo para interligação de Redes locais", Anais do IV Congresso da Sociedade Brasileira de Computação, julho 1984.

6. MENASCÉ, D.A., POPEK, G.J., and MUNTZ, R.R., "A Locking Protocol for Resource Coordination in Distributed Databases", ACM Transactions on Database Systems, Vol.5 Nº 2, pp. 1Ø3-138, junho 198Ø.

7. GARCIA-MOLINA, H., "Elections in a Distributed Computing System", Princeton University - Departament of EECS, dezembro 198Ø.

8. MENASCÉ, D.A. e SCHWABE D., "Redes de Computadores - Aspectos Técnicos e Operacionais", Editora Campus, 1984.

9. OPPEN, Derek C. and DALAL, Yogen K., "The Clearinghouse: A Decentralized Agent for Locating Named Objects in a Distribuited Envionment", Xerox Corporation, outubro 1981.

10. MEDEIROS, J. Laedio, "Especificação e Implementação de um Servidor de Nomes para uma Rede Local", Dissertação de Mestrado, Departamento de Informática, PUC/RJ, fevereiro 1985.