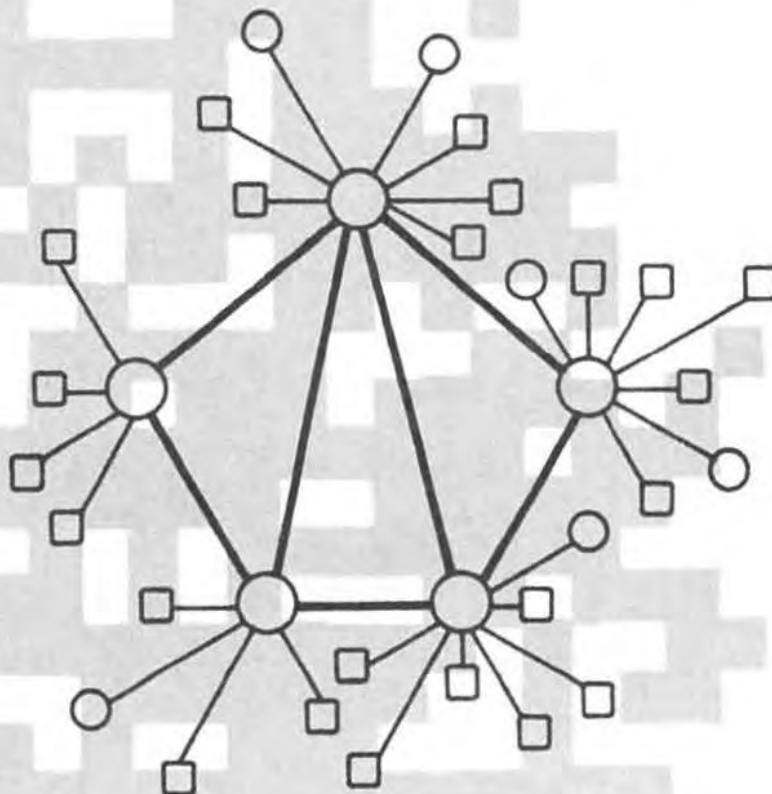


ANais

**1º SIMPÓSIO SOBRE
REDES DE
COMPUTADORES**



*Porto Alegre
Maio 25-27 de 1983*

SOCIEDADE BRASILEIRA DE COMPUTACAO - REGIONAL RS
GRUPO DE INTERESSE EM REDES DE COMPUTADORES DA SBC
PRO-REITORIA DE EXTENSAO - UFRGS
PRO-REITORIA DE PESQUISA E POS-GRADUACAO - UFRGS
CURSO DE POS-GRADUACAO EM CIENCIA DA COMPUTACAO - UFRGS
CENTRO DE PROCESSAMENTO DE DADOS - UFRGS

A N A I S

I SIMPOSIO SOBRE REDES
DE COMPUTADORES

PORTO ALEGRE, RS

MAIO 25-27, 1983



SABi



10º ANIVERSARIO DA POS-GRADUACAO EM CIENCIA DA COMPUTACAO-UFRGS
15º ANIVERSARIO DO CENTRO DE PROCESSAMENTO DE DADOS DA UFRGS

PORTO ALEGRE, CPGCC-UFRGS, 1983

U F R G S

COMISSÃO ORGANIZADORA

J. Rochol, Coordenador (CPGCC-UFRGS)

D. Menascê (PUC-RJ)

L. Moscato (NCE-UFRJ)

G. C. Rodrigues (NCE-UFRJ)

L. M. Tarouco (CPGCC, CPD-UFRGS)

Sociedade Brasileira de Computação - Regional RS

Ph. Navaux, Secretário (CPGCC-UFRGS)

COORDENAÇÃO GERAL

Curso de Pós-Graduação em Ciência da Computação - UFRGS

Editores: J. Rochol (CPGCC-UFRGS)

Z.C.P. Oliveira (CPGCC, CPD-UFRGS)

Capa: G.A.M. de Moraes

Redes: Comunicação de Dados

Protocolos

Redes e Sistemas de Comunicação

Sistemas Operacionais de Computadores

Videotexto

Automação Industrial

SUMÁRIO

	Pág.
PALESTRA DE ABERTURA <i>J. Rochol</i>	07
TRABALHOS	
POSCOT—Um Protocolo de Sessão para Redes de Computadores <i>S. Stiubiener e W. L. Zucchi</i>	10
Projeto e Implementação de um Equipamento PAD <i>E.G.L. de Campos e A.S. Barbosa Jr.</i>	44
Experiência no Projeto e Implementação da Rede Local REPUC <i>D.A. Menascê, L.F.G. Soares e D. Schwabe</i>	64
RELIME—Rede Local do IME <i>I.M. de Carvalho e E.L. Passos</i>	76
A Rede Local de Computadores do NCE/UFRJ <i>J.F.M. de Araujo, G.C. Rodrigues, D.F. Takano e M.H.C. de Azevedo</i>	104
Arquitetura de uma Rede Local com Integração de Serviços de Dados e Voz <i>E.G.L. de Campos</i>	123
Avaliação de Desempenho de Redes de Dados Locais <i>J.A.B. Moura e J.P. Sauvê</i>	125
A Note on Random Graphs, Shortest Edges and Distributed Computing <i>M.M. Krieger</i>	150

CONSTRUCTOR—Uma Proposta de uma Ferramenta para Projeto e Implementação de Sistemas Distribuídos	164
<i>P.F.W.K. de Buzin</i>	
Um Modelo para Sistemas de Mensagens por Computador	210
<i>L.M.R. Tarouco</i>	
Sistemas Operacionais para Redes de Computadores	230
<i>J.A.M. de Queiróz</i>	
Metodologia para Desenvolvimento de Software em Sistemas Distribuídos	253
<i>A.L.B. Fernandes, M. Moszkowicz e L. Della Valle</i>	
PALESTRAS	
STATMUX: Especificações, Desenvolvimento e Aplicações	279
<i>J.B. de Moraes - DIGITEL</i>	
Teletex em Rede de Pacotes	303
<i>R.G. Pinheiro, J.R.E. Leite e R.J.F. dos Santos - CPqD - Telebrás</i>	
Vídeotexto: Desenvolvimento de Terminais de Usuários	313
<i>R.M.B. Felizzola</i>	
Programa de Informática do Rio Grande do Sul	321
<i>A.E.M. Lima e N.B. Rosa - BADESUL</i>	

Os Sistemas de Informação e a Embratel	331
<i>J.L. Charbel - EMBRATEL</i>	
Algumas Idéias para Quando a Rede de Pacotes Chegar	353
<i>R. Brea - EMBRATEL</i>	
Rede Edisa para Automação de Bancos	377
<i>R.H. Hinrichs - EDISA</i>	

PALESTRA DE ABERTURA

Juergen Rochol

Coordenador do 19 Simpósio sobre
Redes de Computadores

Preside esta Sessão Solene de Abertura dos Trabalhos do 19 Simpósio Brasileiro de Redes de Computadores o Exmo. Sr. Vice Reitor da Universidade Federal do Rio Grande do Sul Prof. Sergio Lamb, contamos também com a honrosa presença do Prô-Reitor de Extensão Prof. Ludwig Buckup, do Coordenador do Curso de Pós-Graduação em Ciência da Computação Prof. Philippe Navaux, do Diretor do Centro de Processamento de Dados Prof. Edemundo da Rocha Vieira e do Representante da Divisão Acadêmica Prof. Paulo Alberto de Azeredo.

Senhoras e Senhores!

Em nome da Comissão Coordenadora deste 19 Simpósio sobre Redes de Computadores, gostaria de expressar as boas vindas, em nome da Comissão Organizadora deste Simpósio, a todos os participantes que aqui chegaram, dos mais diversos estados do Brasil.

Acreditamos desde já, que serão atingidos plenamente todos os objetivos deste Simpósio, tendo em vista a participação efetiva e maciça dos mais diversos setores, que atualmente realizam pesquisa ou desenvolvimento tecnológico na área de Redes de Computadores no Brasil.

Temos a satisfação de hoje vermos aqui reunidos, pesquisadores e cientistas das principais Universidades brasileiras e Centros de pesquisa, além de empresas da área de Informática.

Pelo nível e relevância dos trabalhos, que aqui serão apresentados durante estes três dias, tenho a certeza que este evento será um marco na pesquisa e desenvolvimento de Redes de Computadores no Brasil.

Esperamos também, que o evento seja ocasião de um intercâmbio frutífero entre os diversos grupos que atualmente realizam pesquisas, ou desenvolvem, Redes de Computadores no país.

O Grupo de Interesse em Redes de Computadores da Sociedade Brasileira de Computação bem como o Grupo de Pesquisas em Comunicação de Dados do Curso de Pós-Graduação em Ciência da Computação da UFRGS, sentem-se orgulhosos em terem podido organizar este evento, o que só foi possível graças ao apoio decisivo do Centro de Processamento de Dados e da Pró-Reitoria de Extensão desta Universidade, além do apoio financeiro da Digital Equipamentos Eletrônicos Ltda.

A importância que o Rio Grande do Sul assume dia a dia no cenário da Informática no Brasil, colocando-o atualmente entre um dos três grandes polos da indústria eletro-eletrônica e de informática do Brasil, se deve em grande parte ao pioneirismo, determinação e coragem dos pesquisadores, empresários e de setores governamentais do estado. Salientamos aqui o decisivo apoio que o Banco de Desenvolvimento do Sul (BADESUL), através do seu Programa de Informática vem dando às empresas que atuam neste setor.

A Universidade Federal do Rio Grande do Sul através do Curso de Pós-Graduação em Ciência da Computação, curso de Tecnólogos em Processamento de Dados e o Centro de Processamento de Dados, vem contribuindo com a formação de recursos humanos, altamente qualificados, viabilizando desta forma, a quase totalidade das empresas gauchas que se criaram nos últimos anos neste setor.

Ao observarmos esta realidade, preocupa-nos verificar o retraimento cada vez mais crescente dos financiamentos aos projetos de pesquisa das Universidades por Órgãos do governo como FINEP, CNPq e CAPES com prejuízos que se delineiam desastrosos para os programas de Pós-Graduação.

Os grupos de pesquisa das Universidades, além das dificuldades de recursos, se veem muitas vezes esvaziados devido a concorrência de salários bem mais compensadores oferecidos pelas indústrias, o que já levou ao desmantelamento de muitos e bons grupos de pesquisa. Acreditamos que o intercâmbio, Grupos de Pesquisa e Empresas, como o que se verifica neste evento, certamente propiciarão um melhor conhecimento mútuo e talvez represente o início de uma convivência mais benéfica para ambos.

Enfim, esperamos que o convívio durante os três dias do Simpósio, traga a concretização das esperanças que cada um trouxe consigo, acrescido de muitas e profundas amizades, para que todos possamos sair daqui na sexta-feira, melhores profissionais e enriquecidos como pessoas.

Muito Obrigado

Poscot - um protocolo de sessão para redes
de computadores

Stefania Stiubiener (Politécnica-USP)

Wagner Luiz Zucchi (Politécnica-USP)

RESUMO

O trabalho apresenta um protocolo de sessão, projetado de acordo com as recomendações da ISO (International Standard Organization), que pode ser implantado numa rede pública de computadores onde existe uma categoria de usuários que se liga à rede diretamente através de computadores e outra categoria, que se liga através de um equipamento PAD.

As características da rede de computadores, bem como do equipamento PAD, que afetam o protocolo de sessão são apresentados e discutidas nos Cap. 2 e 3.

O protocolo de sessão é apresentado no Cap. 5 como um conjunto de serviços oferecidos ao nível superior, através de primitivos define a interface entre o nível de sessão e o usuário de sessão.

Finalmente, no Cap. 5, discutimos a realização dos serviços de sessão dentro do nível, sem entrarmos em detalhes sobre a implementação.

POSCOT - um protocolo de sessão para redes públicas de computadores

1. INTRODUÇÃO

Observa-se ultimamente, nas atividades dos profissionais atuantes na área de redes de computadores, uma pronunciada tendência em adotar os princípios elaborados pela I.S.O. (International Standard Organization) em relação à arquitetura de sistemas abertos, princípios divulgados no documento conhecido como "Modelo I.S.O."

Neste sentido, o funcionamento de uma rede ou de um conjunto de redes interligadas é considerado como uma sequência de atividades agrupadas em sete níveis (físico, enlace, rede, transporte, sessão, apresentação e aplicação) e especificadas através de protocolos de nível.

Os órgãos internacionais de padronização na área elaboraram protocolos de acordo com o Modelo I.S.O. como também diretrizes para a execução de projetos de protocolos de acordo com as características específicas de cada rede.

Este trabalho tem o propósito de apresentar um protocolo de nível de sessão, projetado para atuar numa rede do tipo acima mencionado, onde existem terminais assíncronos que estão integrados através do equipamento PAD (Packet Assembly/Disassembly), protocolo este, denominado "Protocolo de Sessão para Computadores e Terminais-POSCOT".

Menciona-se que, numa rede semelhante, existem dois tipos de comunicações: ligações computador - computador e ligações computador-terminal, via PAD.

2. ESTRUTURA DE PROTOCOLOS DE UM SISTEMA PERTENCENTE A REDE PÚBLICA

Na fig. 1 é apresentada a estrutura de protocolos

(de acordo com o Modelo ISO), implementada num computador per_tencente a uma rede pública.

NÍVEL APLICAÇÃO	PROTOCOLO DE APLICAÇÃO	
NÍVEL APRESENTAÇÃO	PROTOCOLO DE APRESENTAÇÃO	
NÍVEL SESSÃO	PROTOCOLO DE SESSÃO	
NÍVEL TRANSPORTE	PROTOCOLO DE TRANSPORTE (A)	PROTOCOLO DE TRANSPORTE (B)
NÍVEL REDE	PROTOCOLO DE REDE(TIPO X25- NÍVEL 3)	
NÍVEL ENLACE	PROTOCOLO DE ENLACE(TIPO X25-NÍVEL 2)	
NÍVEL FÍSICO	PROTOCOLO DE NÍVEL FÍSICO	

Figura 1

De acordo com o Modelo I.S.O, em cada nível podem atuar um ou mais protocolos, tendo em vista que o agrupamento das tarefas de nível conduz a modularidade e consequentemente a protocolos de mais fácil implementação.

Observa-se na figura 1 a existência de dois protocolos em nível de transporte: o protocolo de transporte A que é utilizado para atender as exigências da comunicação entre computadores e o protocolo de transporte B, utilizado nas ligações PAD - computador, que possui os recursos necessários para comunicar-se, através de mensagens de nível trocados no ambiente da Arquitetura de Sistemas Abertos, com o protocolo CCITT X.29. Figura 2).

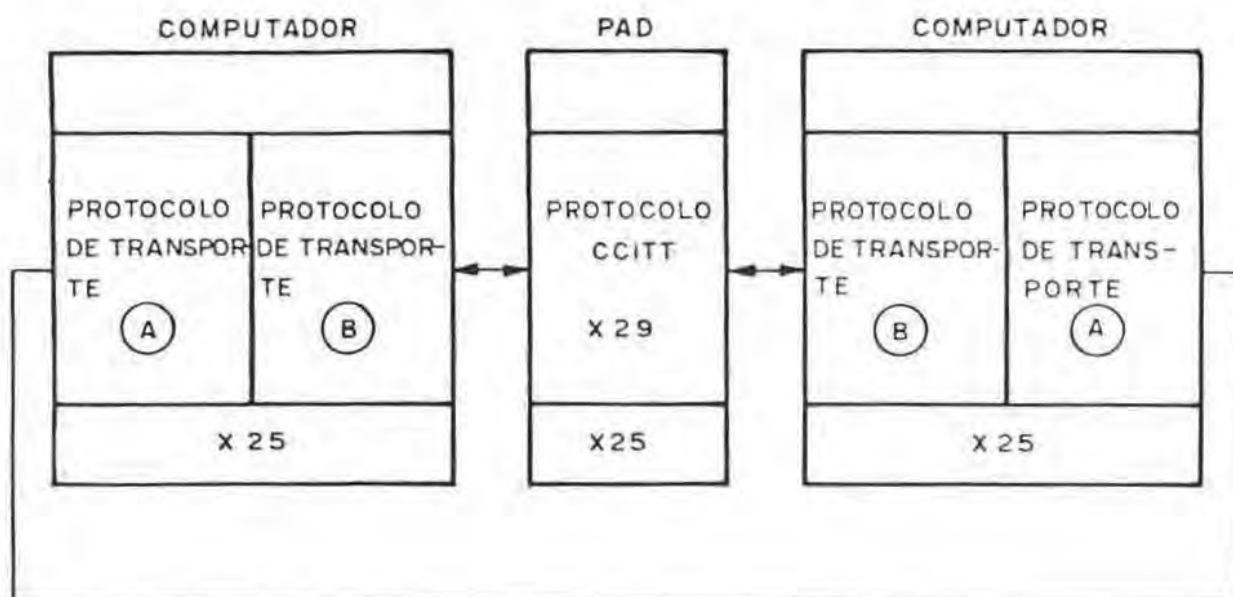


Figura 2

Na estrutura de protocolos apresentada na figura 1, cabe ao protocolo de nível de sessão a tarefa de criar "sessões" de comunicação entre aplicações, com base nos serviços fornecidos pelo(s) protocolo(s) de transporte.

O Modelo I.S.O. não menciona entre as tarefas do nível de sessão e conseqüentemente entre as funções do protocolo de sessão itens relacionados à comunicação via PAD. Porém, considerando a necessidade da existência de uma entidade de sessão que solicita serviços de transporte a serem atendidos via um protocolo tipo B, o protocolo POSCOT foi projetado para atuar em conjunto com o protocolo de transporte A e com o protocolo de transporte B.

3. DESCRIÇÃO SUSCINTA DOS PROTOCOLOS DO PAD

3.1. Introdução ao P.A.D.

O equipamento PAD é definido pela CCITT como uma interface entre os Equipamentos Terminais de Dados (ETD) e o

rando em modo START-STOP, de acordo com a Recomendação V₃, e uma rede pública, ambos situados no mesmo país.

Um terminal tipo start-stop pode, portanto, através do PAD, comunicar-se com qualquer computador ligado à rede pública operando por comutação de pacotes ou com um outro terminal start-stop, também conectado à rede através de um PAD.

As funções do PAD podem ser divididas em três grandes grupos:

a) funções de montagem de pacotes; responsáveis pela recepção dos caracteres enviados pelo terminal assíncrono e pelo agrupamento destes caracteres em pacotes, de acordo com os parâmetros que definem o funcionamento interno do PAD, como também com a Recomendações X.25 - Nível 3;

b) funções de desmontagem de pacotes: responsáveis pela separação dos caracteres de dados que chegam nos pacotes transmitidos pela rede e pelo envio destes caracteres ao terminal assíncrono, conforme os parâmetros internos do PAD.

c) funções de gerenciamento; responsáveis pelo controle de congestão no PAD, supervisão do estado das conexões, controle de temporização etc.

3.2. Protocolos do P.A.D.

As funções realizadas pelo PAD foram objeto de padronização pela CCITT, através de três recomendações publicadas em 1977 e revistas em 1980. Estas recomendações dividem as funções específicas do PAD em três protocolos, conhecidos como protocolo X₃, X₂₈ e X₂₉, respectivamente. O protocolo utilizado para acesso à rede é o protocolo CCITT X₂₅.

A Recomendação X₂₈ define o interface ETD/ECD que permite a um terminal start-stop acessar o PAD numa rede pública de dados.

A recomendação X₃ define os modos possíveis de funcionamento do PAD na rede pública, escolhidos conforme a conveniência dos usuários e a Recomendação X₂₉ define os proce-

dimentos que permitem, tanto a um computador da rede (ETD modo pacote) controlar o funcionamento do PAD, como também possibilita o inter-relacionamento entre dois PAD's, conectados através da rede pública.

Observa-se na figura 3 a estrutura de protocolos específica do PAD.

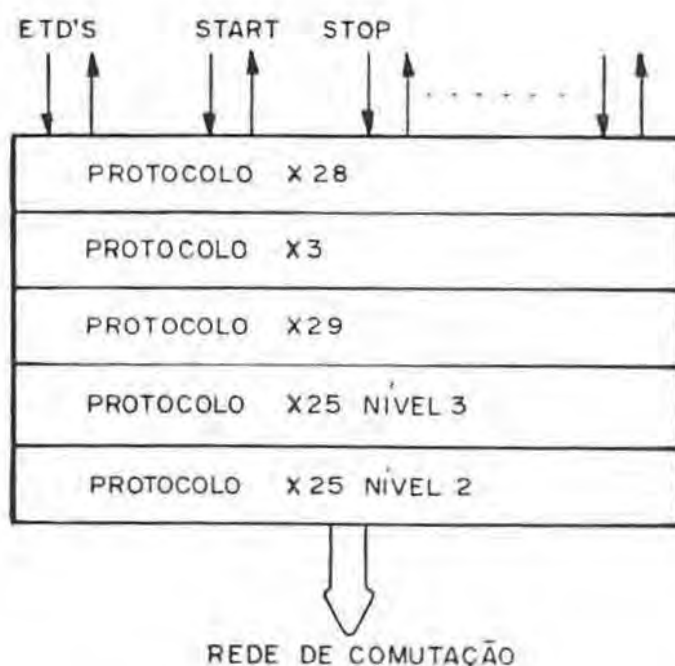


Figura 3

Do ponto de vista do protocolo de sessão, são relevantes as funções do PAD que podem ser alteradas pelo ETD modo pacote, de modo a adequar o PAD às aplicações específicas destes ETD's. Tais funções são definidas pelo protocolo X3 e são as seguintes:

- a) Seleção de sinais de envio de dados;
- b) Controle de fluxo do PAD pelo terminal assíncrono;

- c) Controle de eco dos caracteres originados pelo terminal assíncrono;
- d) Saída da fase de dados através de um caracter e passagem para a fase de espera de comandos;
- e) Seleção do máximo tempo aceito como intervalo de recepção entre caracteres consecutivos;
- f) Edição;
- g) Controle de sinais de serviço do PAD;
- h) Controle de sincronização mecânica do terminal:

O protocolo de sessão deve possibilitar às aplicações existentes na rede alterarem estes parâmetros da maneira mais conveniente. Os formatos das mensagens trocadas entre o PAD e o ETD modo pacote para atualização dos parâmetros do PAD é definido pela recomendação X₂₉.

Da mesma maneira, o usuário do ETD start-stop pode modificar os parâmetros do PAD conforme sua conveniência.

3.3. Endereçamento nas comunicações com o PAD.

O equipamento PAD é considerado pela rede pública, do ponto de vista de endereçamento, como um dos ETD's conectados à rede. O protocolo X25 Nível 3 estabelece as normas de endereçamento dos ETD's através da rede e, neste sentido, quando um terminal assíncrono comunica-se com um ETD modo pacote, é associado um canal lógico a esta conexão, estabelecendo-se um circuito virtual entre o terminal assíncrono o PAD e o ETD modo pacote.

Porém, quando um usuário da rede deseja ligar-se a um terminal conectado ao PAD, o endereço deste não é suficiente para estabelecer-se a ligação porque ao PAD estão ligados diversos terminais que o utilizam de forma multiplexada. Para identificar unicamente um determinado terminal é atribuído, a cada um deles, um "Identificador de Usuário na Rede - IUR". Através deste número, o PAD identifica os seus

terminais e os demais usuários da rede identificam cada terminal pela combinação do IUR e do endereço do PAD (Figura 4)

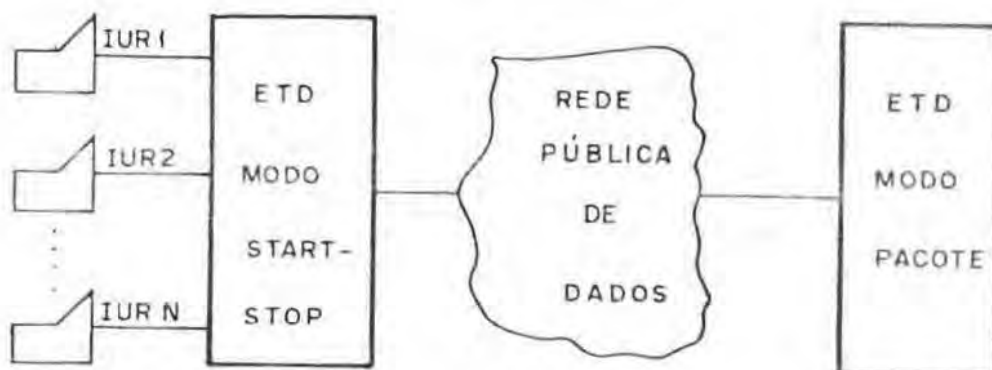


Figura 4

O endereçamento de um processo específico dentro do ETD modo pacote deverá ser feito pelo usuário situado no terminal assíncrono, utilizando os recursos de transporte de dados oferecidos pelo PAD, uma vez que os protocolos do PAD não oferecem a possibilidade de estabelecimento automático de conexões processo-a-processo.

4. SERVIÇOS OFERECIDOS PELO PROTOCOLO DE SESSÃO

4.1. Princípios Gerais

Os serviços de um nível são as capacidades que ele oferece para um usuário do nível imediatamente superior. A fim de fornecer os seus serviços, uma região (ou nível) realiza funções baseadas nos serviços oferecidos pelo nível que lhe é imediatamente inferior (Figura 5).

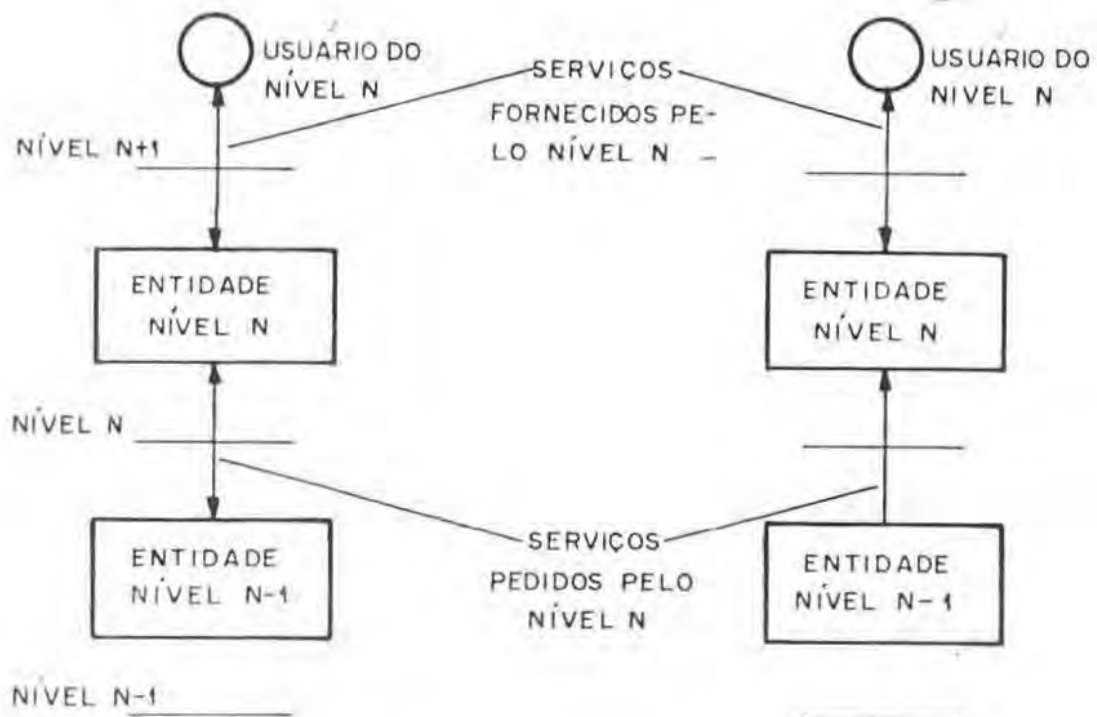


Figura 5

A especificação de um serviço é uma descrição da atividade entre o usuário do nível n e a região de nível n . Esta atividade é descrita especificando a funcionalidade dos serviços fornecidos e as primitivas utilizadas para requisição do serviço. A definição de um serviço é independente de uma interface específica entre o nível $n+1$ e o nível n .

4.2. Especificação dos Serviços

Todo serviço oferecido pelo nível n é especificado por um conjunto de primitivas e por uma série de parâmetros associados a cada uma das primitivas. Uma primitiva de serviço é definida pela sua funcionalidade, direção, e pelo conjunto de parâmetros que lhe podem ser associados.

A funcionalidade de uma primitiva é o aspecto específico do serviço que ela realiza. Naturalmente primitivas

pertencentes ao mesmo serviço possuem funcionalidades correlatas.

A direção indica a origem do relacionamento, entre o n-usuário e o n-nível, que a primitiva estabelece. Primitivas do usuário são originados pelo usuário no nível n+1. Primitivas para o usuário são as originadas pelo entidade do nível n que fornece o serviço.

Um modelo do relacionamento de interface entre os usuários do nível n e a região do nível n está representado na figura 6.

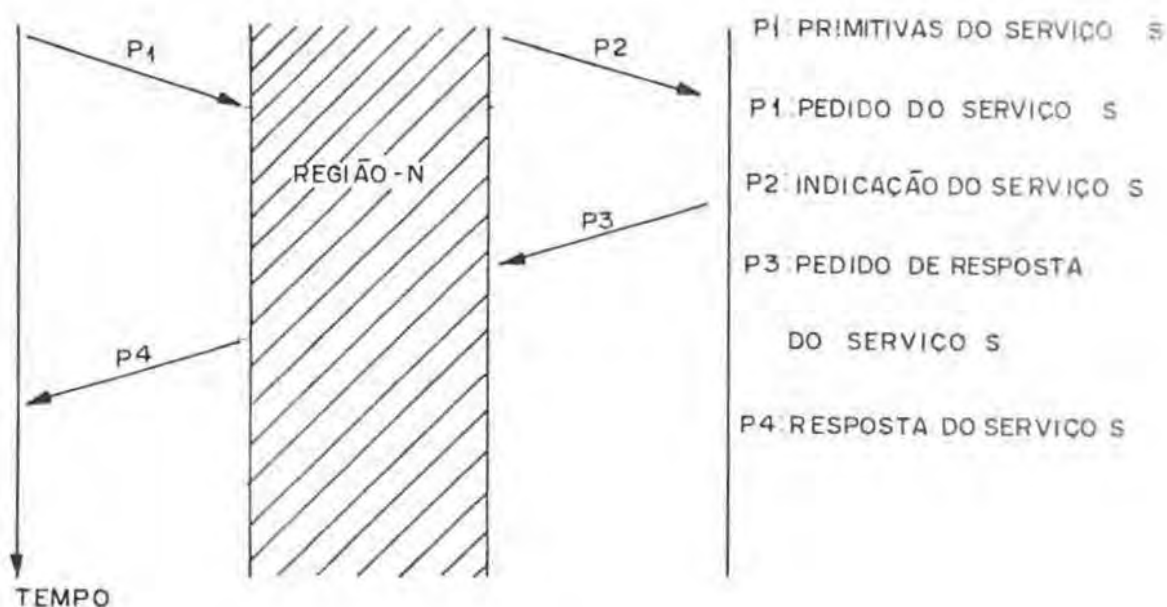


Figura 6

Os parâmetros associados com uma primitiva são especificados pela faixa de valores permissíveis

4.3. Visão Geral dos Serviços Oferecidos pelo Nível de Sessão

A finalidade básica do nível de sessão é estabelecer e manter um caminho lógico adequado para troca de dados

entre entidades de aplicação que desejam se comunicar. Estas entidades de aplicação são chamadas de usuários de sessão e o caminho lógico entre as entidades em comunicação é chamado de conexão de sessão.

Existe também um tipo especial de conexão de sessão que é utilizado nas comunicações entre um computador e um terminal assíncrono através do PAD. O formato dos dados transmitidos sobre esta conexão de sessão é diferente dos formatos utilizados nas conexões computador-computador dado as exigências peculiares do protocolo X29.

Para fornecer seus serviços, a região de sessão se apoia nos serviços fornecidos pelo nível de transporte. Tipicamente este será constituído por duas entidades de transporte: uma para conexões tipo canal virtual, e outra implementando o protocolo X29 para conexões computador-PAD.

A região de sessão fornece serviços de três tipos:

- a) Serviços de gerenciamento da conexão da sessão: permitem ao usuário de sessão criar, manter e terminar associações com um correspondente usuário de sessão
- b) Serviços de Transferência de Dados: fornecem os meios para troca de dados entre o par de usuários da sessão
- c) Serviços de Gerenciamento dos Dados Trocados: permitem aos usuários da sessão impor uma estrutura sobre suas interações

4.4. Serviços de Sessão

4.4.1. Serviços de Gerenciamento da Sessão

4.4.1.1 Serviço de Estabelecimento da Sessão

Este serviço permite estabelecer uma conexão de sessão entre dois usuários de sessão. Durante o estabelecimen

to os usuários podem trocar os respectivos endereços da sessão e uma limitada quantidade de dados em cada direção.

Primitivas de Interface: a) ABRA-SESSÃO

parâmetros: nome do usuário local,
identificador do processo remoto, dados de usuários, senha
sentido : do usuário de sessão para o nível de sessão

Obs: os identificadores dos usuários de sessão local é um nome local. É responsabilidade do nível de sessão mapear este nome local num endereço conhecido através da rede.

- dados de usuário constituem um conjunto limitado de dados que podem ser transferidos entre os usuários durante a fase de estabelecimento da conexão

- senha é uma palavra de código para acesso a aplicação restritas (no cap. V está explicado o que é uma aplicação com acesso restrito)

b) CONEXÃO-ACEITA

parâmetros: (nome do usuário local)
sentido: do nível de sessão para o usuário de sessão

Obs: informa que a conexão pedida pelo usuário local foi estabelecida

c) ESCUTE

parâmetros: (nome do usuário local, número de reentrância, senha)
sentido : do usuário de sessão para o nível de sessão

Obs: permite a um usuário colocar-se à disposição da rede para ser utilizado por usuários remotos.

O parâmetro "número de reentrância" indica o número máximo de conexões de sessões que o usuário está disposto a aceitar em qualquer instante.

O parâmetro senha é um código de acesso restrito opcional. Nos casos em que a senha é fornecida pelo usuário local somente serão estabelecidas conexões com usuários remotos que possuam a mesma senha

d) ABERTURA-REMOTA

parâmetros: (nome do usuário local, identificador do usuário remoto, tipo de ligação)

sentido: do nível de sessão para o usuário da sessão

Obs: Esta primitiva corresponde a um aviso de que uma conexão foi estabelecida por um usuário remoto com um usuário local que se tinha colocado à disposição da rede.

O parâmetro tipo de ligação informa se a ligação provém de outro computador ou de um PAD

A figura 7 apresenta as primitivas de interface associadas ao serviço de estabelecimento da sessão



Fig. 7 - Primitivas de Serviço para Abertura de Conexão



Fig. 8 - Primitivas de Serviço para Desconexão Normal

4.4.1.2. Serviço de Fechamento Normal da Sessão

Este serviço permite terminar uma conexão sem perda de dados que estejam sendo transferidos sobre a sessão. Quando um usuário solicita este serviço ele deve parar de enviar dados, porém ele continuará recebendo dados até que

o usuário remoto concorde com o fim da conexão de sessão

Primitivos de Interface:

a) FECHÉ-SESSÃO

parâmetros: nome do usuário local, causa

sentido: do usuário de sessão para o nível de sessão

Obs: no campo de causa, o usuário que inicia a desconexão pode colocar uma razão para a desconexão. Valores específicos deste campo dependem das entidades de aplicação envolvidas.

b) FIM-CONEXÃO

parâmetros: nome do usuário local, causa

sentido: da região de sessão para o usuário de sessão

Obs: esta primitiva informa ao usuário de sessão que:

a) o usuário remoto pediu uma desconexão, no caso em que o usuário local não pediu desconexão

b) o pedido do usuário de sessão local para uma desconexão foi atendido pelo usuário remoto, no caso em que o usuário local pediu desconexão

A figura 8 mostra os eventos de interface correspondente a este serviço

4.4.1.3. Serviço de Fechamento Forçado da Sessão

Neste serviço o usuário de sessão termina a conexão de sessão abruptamente, sem esperar pela confirmação do usuário remoto. O nível de sessão não garante a integridade

dos dados transferidos quando da utilização deste serviço.

Primitivos de Interface:

a) ABORTE-SESSÃO

parâmetros: nome do usuário local, causa
sentido : do usuário de sessão para a região de
sessão

b) SESSÃO-ABORTADA

parâmetros : nome do usuário local, causa
sentido : da região de sessão para o usuário da
sessão

A figura 9 mostra os eventos de interface corres-
pondente a este serviço

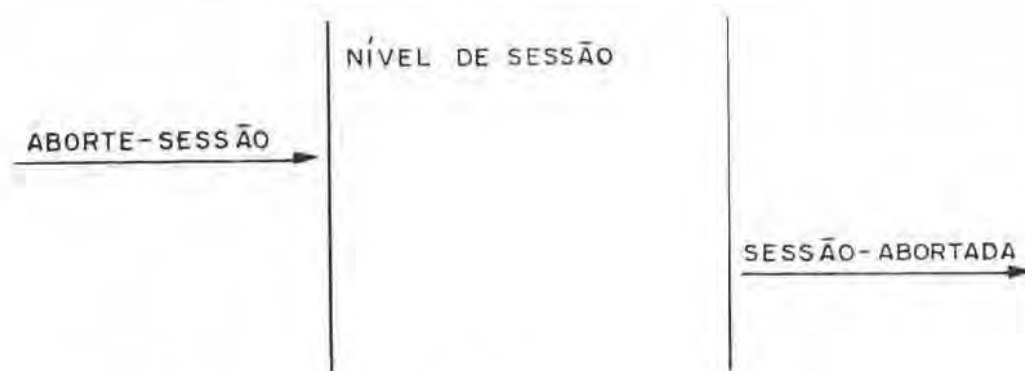


Figura 9

Obs: a região de sessão também pode abortar uma co-
nexão por sua própria iniciativa em decorrência
de falhas graves na comunicação. Neste caso
o primitivo SESSÃO-ABORTADA será enviado em
ambos os sentidos

4.4.2. Serviços de Gerenciamento dos Dados Trocados

4.4.2.1. Serviço de Negociação dos Parâmetros da Ses- são

É utilizado para caracterizar a conexão de sessão por um conjunto de parâmetros. Este conjunto de parâmetros é denominado perfil da conexão. Os parâmetros da sessão e seus possíveis valores estão descritos no Cap. V deste trabalho.

Primitivas do Serviço:

a) NEGOCIE-PARAMETROS

parâmetros: nome do usuário local, parâmetros, valores

sentido: do usuário de sessão para a região de sessão

Obs: "parâmetros" especifica os parâmetros que devem ser negociados entre os usuários de sessão e "valores" os respectivos valores de cada parâmetro

b) NEGOCIAÇÃO-COMPLETA

parâmetros : nome do usuário local, parâmetros, valores

sentido: da região de sessão para o usuário de sessão

Obs: este primitivo informa ao usuário de sessão sobre o resultado da negociação por ele pedida. Os valores dos parâmetros não precisam ser necessariamente iguais aos pedidos pelo usuário de sessão pois são resultado de uma negociação dentro da região da sessão.

A figura 10 mostra os eventos de interface correspondentes a este serviço



Figura 10 - Primitivas de Serviço para Negociação

4.4.2.2. Serviço de Leitura dos Parâmetros da Sessão

É utilizado pelo usuário de sessão para conhecer o perfil de uma sessão

primitivos do serviço

a) LEIA-PARAMETROS

parâmetros: nome do usuário local

sentido: do usuário de sessão para a região de sessão

b) INDICAÇÃO PARAMETROS

parâmetros: nome do usuário local, parâmetros, valores

sentido: da região de sessão para o usuário de sessão

A figura 11 mostra os eventos de interface associados a este serviço

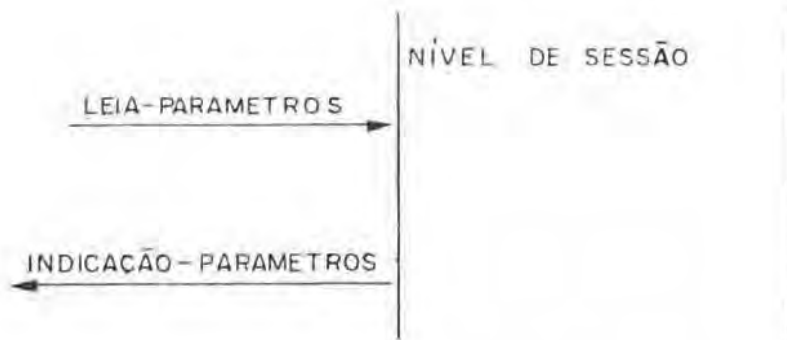


Figura 11 - Primitivas de Serviço para Leitura dos Parâmetros da Sessão

4.4.2.3. Serviço de Gerenciamento do Diálogo

Este serviço é utilizado pelo usuário de sessão para alterar o sentido da comunicação. Ele só deve ser utilizado nas conexões meio-duplex e pelo usuário que possui a vez de comunicação.

primitivas do serviço:

a) ALTERA-SENTIDO

parâmetros: nome do usuário local

sentido : do usuário de sessão para a região de sessão

Obs: através deste primitivo o usuário dá a vez de falar ao usuário remoto

b) SENTIDO-ALTERADO

parâmetros: nome do usuário local

sentido : da região de sessão para o usuário de sessão

Obs: este primitivo informa ao usuário de sessão que ele está recebendo a vez de falar

A figura 12 mostra os eventos de interface associados a este serviço



Figura 12 - Primitivas de Serviço para Gerenciamento do Diálogo

4.4.2.4. Serviço de Gerenciamento de Bloco Compactado (Quarentena)

Este serviço permite ao usuário de sessão apagar um bloco compactado de dados (veja o Cap. 5 parag. 5.3. para detalhes sobre blocos compactados) que está retido pela região de sessão.

primitivas

a) ANULA-BLOCO-COMPACTADO

parâmetros: nome do usuário local

sentido : do usuário de sessão para a região de sessão

Obs: quando este comando é utilizado os dados armazenados pela região de sessão são perdidos.

A figura 13 mostra os eventos de interface associados a este serviço



Figura 13 - Primitivas do serviço para Gerenciamento de bloco compactado

4.4.2.5. Serviço de Recuperação de Erros das Conexões de Sessão

É utilizado sempre que o usuário origina uma primitiva incorreta ou incompatível com o estado da conexão correspondente

primitivas do serviço:

a) ERRO-LOCAL

parâmetros: nome do usuário local, causa do erro

sentido : da região de sessão para o usuário de sessão

A figura 14 mostra os primitivos de interface associados a este serviço

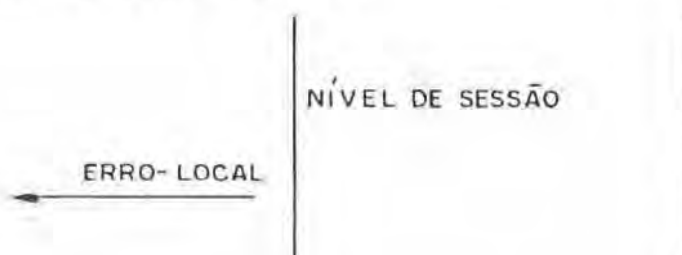


Fig. 14 - Primitivos do serviço de recuperação de erros das conexões de sessão

4.4.3 Serviços de Transferência de Dados

4.4.3.1. Serviço de Transferência de Dados sobre a Conexão de Sessão

Este serviço prove a transferência de um bloco de dados entre os usuários de sessão. Este bloco de dados é chamado de "unidade de dados do serviço de sessão". (UDSS). A integridade das UDSS é garantida pela região de sessão, e o seu tamanho máximo é um parâmetro negociável na região da sessão.

primitivas do serviço

a) ENVIA-BLOCO

parâmetros (nome do usuário local, dados, compactação)

sentido: do usuário de sessão para a região de sessão

Obs: o parâmetro "compactação" indica a localização do UDSS dentro de um bloco compactado. Se este parâmetro é igual a zero então existem outros UDSS depois deste no mesmo bloco compactado, caso contrário, este UDSS é o último do atual bloco compactado

b) RECEBE-BLOCO

parâmetros: nome do usuário local, dados

sentido : da região de sessão para o usuário de sessão

Obs: esta primitiva é utilizada para entrega de dados ao usuário destinatário

A figura 15 mostra os eventos de interface para este serviço

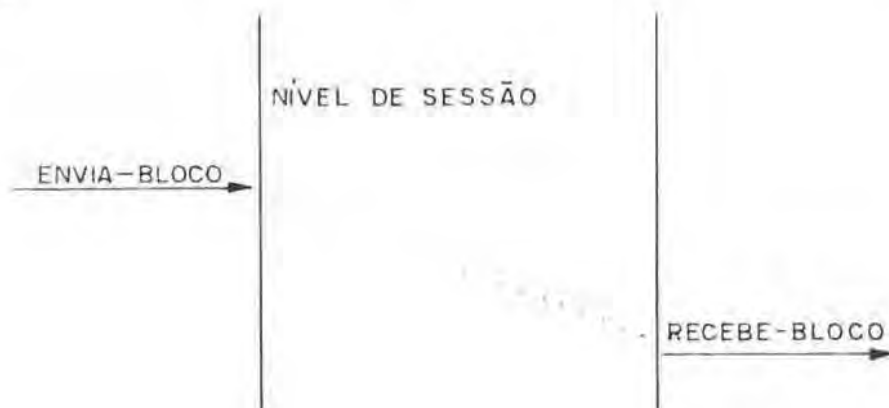


Figura 15 - Primitivas do serviço de transferência de dados

4.4.3.2. Serviço de Transferência de Telegramas sobre a Conexão de Sessão

Este serviço é utilizado pelo usuário de sessão para transmitir um bloco de dados curto sobre a conexão de sessão, independentemente dos mecanismos de controle de fluxo das regiões inferiores, do estado de recepção de blocos compactados, e de quem é a vez de transmitir nas conexões meio-duplex. Os dados transferidos pelo serviço de telegramas não são sincronizados com relação aos dados transferidos pelo serviço normal

Primitivas do serviço:

a) ENVIA-TELEGRAMA

parâmetros: nome do usuário local, dados

sentido : do usuário de sessão para a região da sessão

Obs: O conjunto de dados transferidos não deve ser maior do que 8 bits

b) RECEBE-TELEGRAMA

parâmetros: nome do usuário local, dados
sentido: da região de sessão para o usuário de
sessão

A figura 16 mostra os eventos de interface do serviço de transferência de telegramas

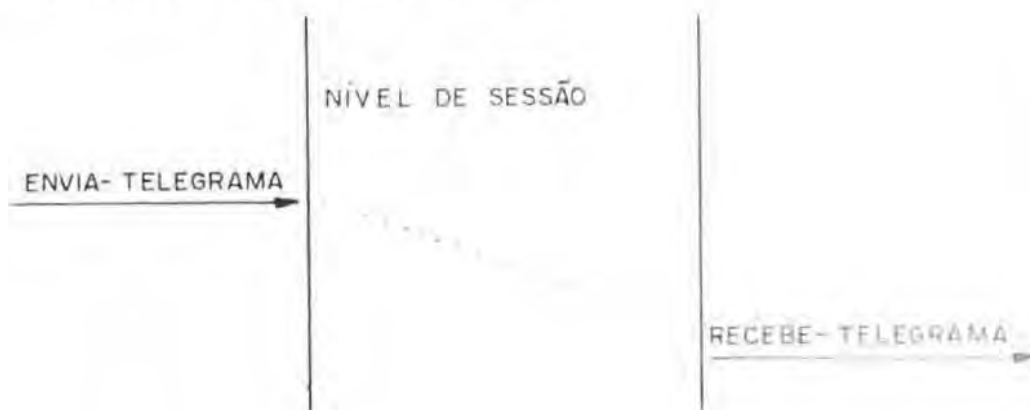


Figura 16 - Primitivas do Serviço de Transferência de Telegramas

5. PROTOCOLO DE SESSÃO-POSCOT

O protocolo de sessão é o conjunto de regras e de procedimento utilizados pelas entidades de sessão, nas suas comunicações visando o cumprimento dos serviços de sessão oferecidas aos usuários do nível de sessão.

Dado o carácter sintético deste trabalho descreveremos apenas os mecanismos utilizados pelo protocolo sem deter-nos no formato das mensagens trocadas pelos usuários

5.1. Estabelecimento e Fim de Sessões

O protocolo de sessão deve permitir aos seus usuários o estabelecimento de conexões de sessão, quer ativamente, isto é, chamando outro usuário através da rede, quer passivamente, isto é, sendo chamado por um usuário remoto.

As conexões de sessão são estabelecidas utilizan-

do-se dos serviços fornecidos pela região de transporte, razão pela qual, não há mensagens de nível para estabelecimento de sessões. As conexões de sessão são mapeadas uma a uma nas conexões de transporte e não há multiplexação na região de sessão.

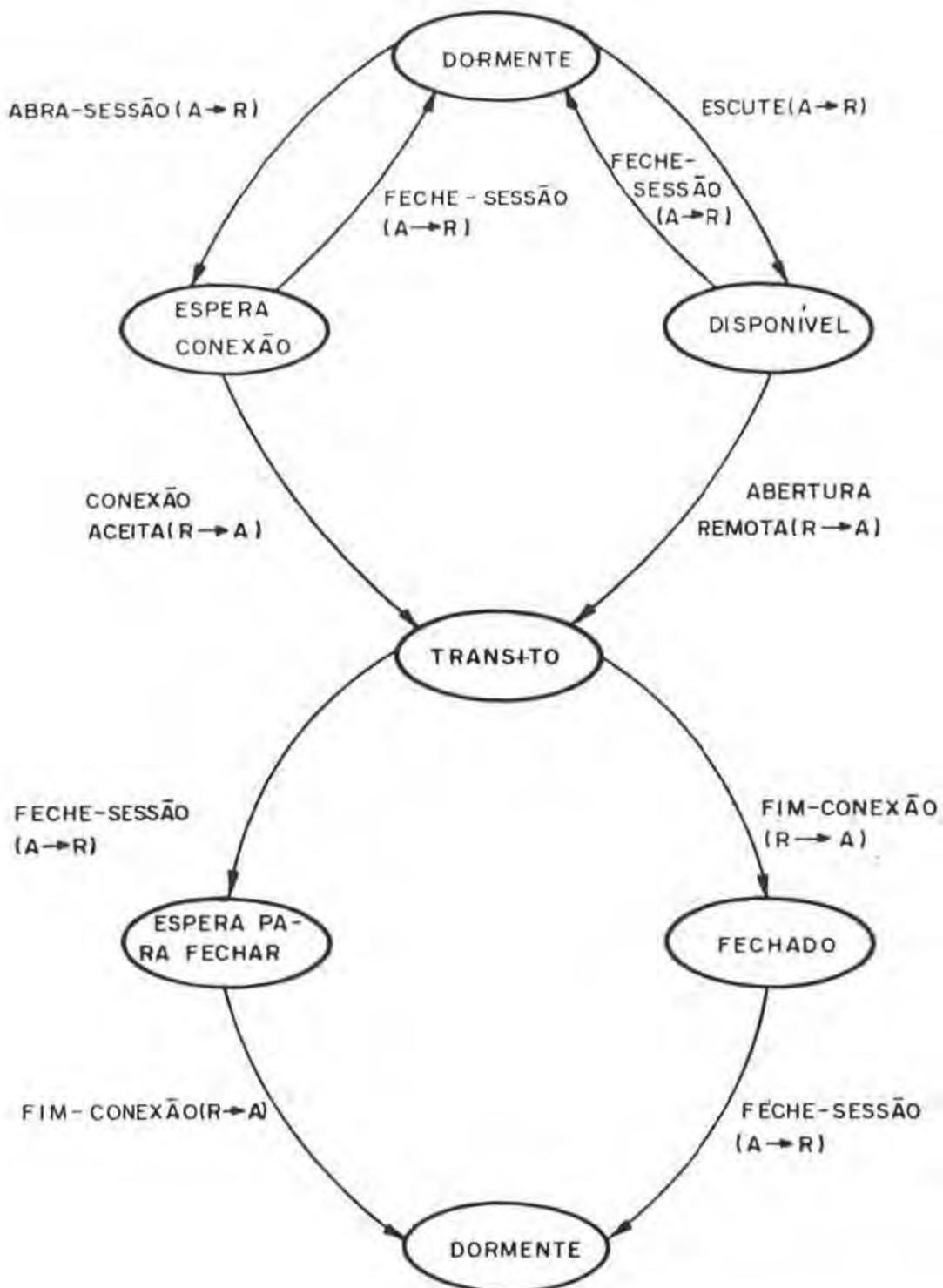


Figura 17 : diagrama de estados na interface entre região de sessão e o nível superior para estabelecimento e fim de sessões

O estabelecimento das conexões de sessão obedece a um esquema do tipo "two-way handshake", e o diagrama de estados da interface com o nível de aplicação está mostrado na figura 17.

Nesta fase da comunicação o protocolo de sessão deve, além de cuidar do estabelecimento das sessões, prever o controle de acessos a aplicações reentrantes, isto é, que devem ser acessíveis a vários usuários simultaneamente, e o controle de acesso a aplicações restritas, isto é, aplicações que são disponíveis apenas a um certo grupo de usuários.

O fim das conexões de sessão pode ser produzido por duas razões

- a. Desconexão local: pedida pela entidade de aplicação local
- b. Desconexão Remota: pedida pelo usuário remoto ou provocada por falhas na região de transporte

5.2. Negociação dos Parâmetros da Sessão

5.2.1. Parâmetros da Sessão

O formato das mensagens utilizados para negociação dos parâmetros da sessão deve ser compatível com o formato utilizado para negociação dos parâmetros do PAD (veja recomendação X29) de forma a permitir a utilização do mesmo protocolo para comunicações com esse tipo de equipamento. Toda sessão possui um conjunto de parâmetros a ela associada que estabelece o funcionamento da sessão durante a comunicação entre os processos de aplicação. A este conjunto de parâmetros damos o nome de perfil da conexão.

Uma vez que a sessão é estabelecida, de acordo com

os mecanismos descritos acima, é associado à sessão um perfil-padrão que permite a comunicação entre as entidades de aplicação ou a negociação de um novo perfil entre as entidades de sessão.

Esta negociação de perfil é feita pelas entidades de sessão a pedido das entidades de aplicação. Os seguintes parâmetros podem ser negociados:

Parâmetro 1

Controle da transferência de dados sobre a conexão de sessão.

Valores possíveis

- 0 - simplex
- 1 - meio duplex com troca de sentido automática após cada mensagem
- 2 - Meio duplex com troca de sentido determinada por mensagem de controle
- 3 - Duplex Completo
- 4 - Indiferente, isto é, a entidade de sessão concorda com qualquer mecanismo proposto pela entidade parceira.

Valor padrão

3

Parâmetro 2

Unidade de Quarentena; especifica quantas unidades de serviços de dados de sessão constituem uma unidade de quarentena (veja o item "serviços de controle de dados trocados na sessão", para maiores detalhes)-

Valores possíveis

1 a 255

Valor padrão

1

Parâmetro 3

Tamanho das unidades de dados de serviço de sessão (UDSS). É a unidade de comunicação entre o nível de sessão e o nível de transporte (Ref. 7)

Valores possíveis

1 a 255

Valor padrão

1

Parâmetro 4

Caracter de fim no formato linha.

Valores possíveis

2 - caracter CR

126 - todos os caracteres nas colunas 0 e 1 do alfabeto internacional número 5 e caracter DEL

6 - caracteres CR, ESC, BEL, ENQ, ACK.

18 - Caracteres CR, EOT, ETX

Valor padrão

2

Parâmetro 5

Entidade de transporte utilizada.

Valores possíveis

0 - Comunicação entre computadores da rede

1 - Comunicação entre computador e PAD

Valor padrão

Não há. O valor deste parâmetros é definido durante a fase de estabelecimento da sessão.

5.2.2. Mecanismo de Negociação dos Parâmetros da Sessão

A negociação dos parâmetros da região de sessão é feita pelas entidades de sessão, através de mensagens de formato padronizado trocados com as outras entidades de sessão que participam do sistema aberto de interconexão.

A região de sessão, a princípio, não conhece os parâmetros desejados para uma dada sessão que se estabelece, num dado momento, entre duas entidades de aplicação. O perfil das novas sessões será então o perfil-padrão até que a negociação de um novo perfil seja solicitada pela entidade de aplicação.

Quando uma nova negociação é pedida, a entidade de sessão deixa de enviar dados para sua entidade parceira e envia uma mensagem de negociação solicitando o novo perfil; a entidade parceira ao receber a mensagem de negociação atualiza suas variáveis de estado e envia uma resposta à entidade de sessão que iniciou a conexão.

Esta resposta pode ser de três tipos:

a. Aceitação da proposta:

A entidade de sessão que recebe o pedido de negociação concorda com ele e envia uma confirmação da sua ocupação. Doravante todos os dados serão transmitidos e recebidos, de acordo com o novo perfil. A outra entidade de sessão ao receber a confirmação da negociação também passa a enviar e receber dados de acordo com o novo perfil.

b. Envio de uma contra-proposta:

A entidade de sessão não pode aceitar a proposta oferecida por sua parceira e assim envia uma contra-proposta o mais próxima possível da proposta original, porém compatível com as suas possibilidades. A entidade de sessão sempre assume que sua parceira aceitará a contra-proposta e por isto já

passa a transmitir e receber dados de acordo com o perfil oferecido na primeira contra-proposta.

A entidade de sessão, que originou a negociação, ao receber a contra-proposta deve aceitá-la ou encerrar a sessão.

c. Fim da conexão

No caso em que a proposta é totalmente incompatível com os recursos da região de sessão esta pode encerrar a conexão de sessão.

5.3. Transferência de Dados de Usuário e Compactação

Dados podem ser transferidos entre entidades de aplicação usando uma sessão estabelecida. A unidade de dados entregues por um usuário à entidade de sessão cuja identidade é preservada ao longo da sessão é chamada de unidade de dados de serviço de sessão (UDSS). O tamanho de uma UDSS não é limitado e pode variar durante a sessão. A única limitação ao tamanho das UDSS é a memória disponível do sistema onde o protocolo de sessão é implantado.

O usuário pode desejar passar dados para a entidade de sessão porém impedindo a entrega dos dados para o correspondente usuário destinatário até que uma quantidade especificada de dados esteja disponível para entrega. Este serviço de reter porções de dados até totalizar uma certa quantidade é chamado de compactação (em inglês este serviço é chamado de "quarantining"). A quantidade máxima de dados a ser compactada num dado instante, define a unidade de compactação (u), e é um parâmetro estabelecido na fase de negociação. O usuário remoto pode concordar com a unidade de compactação proposta pelo usuário iniciador ou declarar um tamanho menor que se torna a unidade de compactação para ambos os usuários.

O esquema de funcionamento do serviço de compactação é o seguinte: o usuário A transfere n UDSS para o nível de sessão, indicando a primeira e a última UDSS que devem

ser compactadas. A entidade de sessão A transfere para sua entidade parceira as UDSS na medida em que as recebe e vai contando o número de UDSS transferidos. Se, em qualquer instante a unidade de sessão verifica que este número é maior do que a unidade de compactação ($n > u$) ela dá uma mensagem de erro ao nível de aplicação e envia para a entidade de sessão remota uma mensagem para descartar as UDSS ainda não entregues ao nível superior.

Caso contrário, ao receber a última UDSS do bloco a entidade de sessão a envia com uma marca especial indicando o fim da compactação.

A entidade de nível superior também pode requerer à entidade de sessão a anulação dos dados sendo compactados quando, a seu critério, detetar algum erro nos dados já transmitidos porém ainda não entregues à entidade de aplicação destinatária.

Se a entidade de aplicação deseja enviar dados para um PAD ela deve manter a unidade de compactação sempre igual a um e precaver-se de utilizar o serviço de anulação dos dados já transmitidos.

5.4. Controle do Diálogo

É um mecanismo utilizado nas comunicações meio-duplex para comutar o sentido de transmissão da linha entre as entidades de sessão parceiras. Este mecanismo pode ser realizado de duas maneiras de acordo com o valor do parâmetro 1.

- a. Se o parâmetro 1 é igual a 1 a troca de sentido é realizada toda vez que uma mensagem de dados indicando o fim de um bloco compactado é enviada.
- b. Se o parâmetro 1 é igual a 2 a troca de sentido é realizada por uma mensagem especial independente da transferência de dados. O envio desta mensagem é determinado pelos usuários de nível superior. So mente o usuário que está enviando dados pode soli-

citar o envio desta mensagem. Se o usuário envia a mensagem de troca de sentido durante a transmissão de um bloco compactado, o nível de sessão anulará as UDSS que estavam sendo compactadas.

Este serviço não é oferecido nas comunicações computador-PAD pois as comunicações com o PAD possuem sempre natureza duplex.

5.5. Transferência de Telegramas

Telegramas são conjuntos de dados de tamanho limitado que são transferidos entre os usuários de sessão, sem obedecer aos mecanismos de controle de fluxo, de compactação. O mecanismo de transferência de telegramas se apoiará nos recursos fornecidos pela região de transporte e não há garantia de sequencialização entre os telegramas e as mensagens do fluxo normal.

6. Conclusões

O protocolo de sessão Poscot apresentado neste trabalho, possui as características necessárias para atuar numa estrutura de protocolos, projetada de acordo com o Modelo I.S.O, para uma arquitetura de sistemas abertos. Especificamente, o Poscot considera uma estrutura onde atuam: o protocolo de transporte P.T.R, a Recomendação CCITT X25 e as Recomendações CCITT X3, X28 e X29.

O Poscot permite o estabelecimento, gerenciamento de transferência de dados e fechamento de sessões entre aplicações alojadas em computadores e também entre terminais assíncronos ligados a um PAD e aplicações em computadores; oferece primitivas de comunicação de interface com os níveis adjacentes, de transporte e de apresentação.

Bibliografia

1. International Organization for Standardization - ISO
"Reference Model of Open Systems Architecture"
ISO/TC 97/SC16/N, Nov. 1978
2. International Organization for Standardization - ISO
"Data Processing - Open Systems Interconnection -
Basic Reference Model"
ISO/TC 97/SC16/537 Mar. 1981
3. National Bureau of Standards - Institute of Computer
Sciences and Technology
"Specification of the Session Protocol"
Report nº ICST/HLNP-81-2; Mar. 1981
4. CCITT Provisional Recommendation X.3
"Packet Assembly/Disassembly Facility (PAD) in a Pu-
blic Data Network" Genebra, 1980.
5. CCITT Recommendation X.28 "DTE/DCE Interface for a Start-
-Stop Mode Data Terminal Equipment Accessing the Pa-
cket Assembly/Disassembly Facility (PAD) in a Public
Data Network situated in the Same Country", Genebra
1980
6. CCITT Provisional Recommendation X.29
"Procedures for the Exchange of Control Information
and User Data Between a Packet Mode DTE and a Packed
Assembly/Disassembly Facility (PAD)" Genebra 1980
7. Stiubiener, Stefania - Projeto de um Protocolo de trans-
porte segundo orientação da ISO. "Segundo Simpósio La-
tino-Americano sobre Redes de Computador", Junho, 1982.

PROJETO E IMPLEMENTAÇÃO DE UM EQUIPAMENTO
PAD ("PACKET ASSEMBLER/DISASSEMBLER")

Edit G. L. de Campos
Armando S. Barbosa Jr.
Escola Politécnica - Universidade de São Paulo

RESUMO

O objetivo deste artigo é apresentar o projeto de um montador/desmontador de pacotes (PAD) implementado no Laboratório de Sistemas Digitais da Escola Politécnica da USP. Trata-se de um concentrador inteligente de terminais a ele ligados via linhas privadas ou rede telefônica. Sua função básica é a de prover a comunicação entre esses terminais e uma rede pública de computadores.

I. INTRODUÇÃO

Durante os últimos anos, a equipe do Laboratório de Sistemas Digitais (LSD) da Escola Politécnica da USP esteve intensamente engajada no desenvolvimento de equipamentos de comunicação de dados, mais especificamente no projeto e implementação de sistemas voltados à comutação de pacotes, com o objetivo de criar um "know-how" e formar pessoal na área de redes de computadores.

No contexto de redes geograficamente distribuídas (públicas) foram desenvolvidos uma Central (nó) de comutação de pacotes [BARB 82] [RUGG 80] e um concentrador/conversor de

protocolos (PAD - "Packet Assembler/Disassembler") sob o patrocínio da Embratel-Telebrás. Vários outros trabalhos correlatos foram realizados, entre os quais destacam-se estudo, projeto e implementação de protocolos de alto nível (transporte) [STIU 81], sub-sistemas de testes de protocolos e sistema gerador automático de implementações de protocolos [MONT 82].

Este artigo apresenta a descrição da arquitetura de um dos equipamentos acima citados, o PAD, cujo desenvolvimento e implementação foram realizados, em um ano, pela equipe do LSD. A máquina foi construída de forma a atender as especificações da recomendação X.3 [CCIT 83] da CCITT, que determina as funções básicas de um PAD, a saber:

- . Recebimento de caracteres de terminais assíncronos e montagem destes dados em pacotes destinados a um computador remoto.

- . Desmontagem de pacotes de dados recebidos do computador remoto em caracteres a serem enviados aos terminais.

- . Estabelecimento e finalização de chamadas virtuais, assim como a execução dos procedimentos de interrupção e "reset".

- . Geração de sinais de serviço para o terminal.

- . Transmissão de pacotes, montados a partir de caracteres, de acordo com um mecanismo de temporização.

- . Reconhecimento e interpretação de sinais de controle provenientes dos terminais.

Tendo em vista o atendimento dos serviços acima descritos, montou-se uma infraestrutura de hardware e software, cujas características básicas serão descritas nos itens se-

guintes.

II. ESPECIFICAÇÃO DO PAD

A especificação do PAD foi feita com vistas a torná-lo um equipamento totalmente compatível com as recomendações do CCITT. Desta forma, fizeram parte da especificação de funções a serem implementadas no PAD todas aquelas apresentadas na recomendação X.3. A interface entre o equipamento e o terminal do usuário foi definido como sendo o descrito na recomendação X.28 [CCITX28], e o protocolo entre o PAD e o computador destino foi o X.29 [CCITX29]. O protocolo de acesso à rede de comutação de pacotes foi especificado como sendo o X.25 [CCITX25]. A figura 1 esquematiza a atuação destes protocolos na rede.

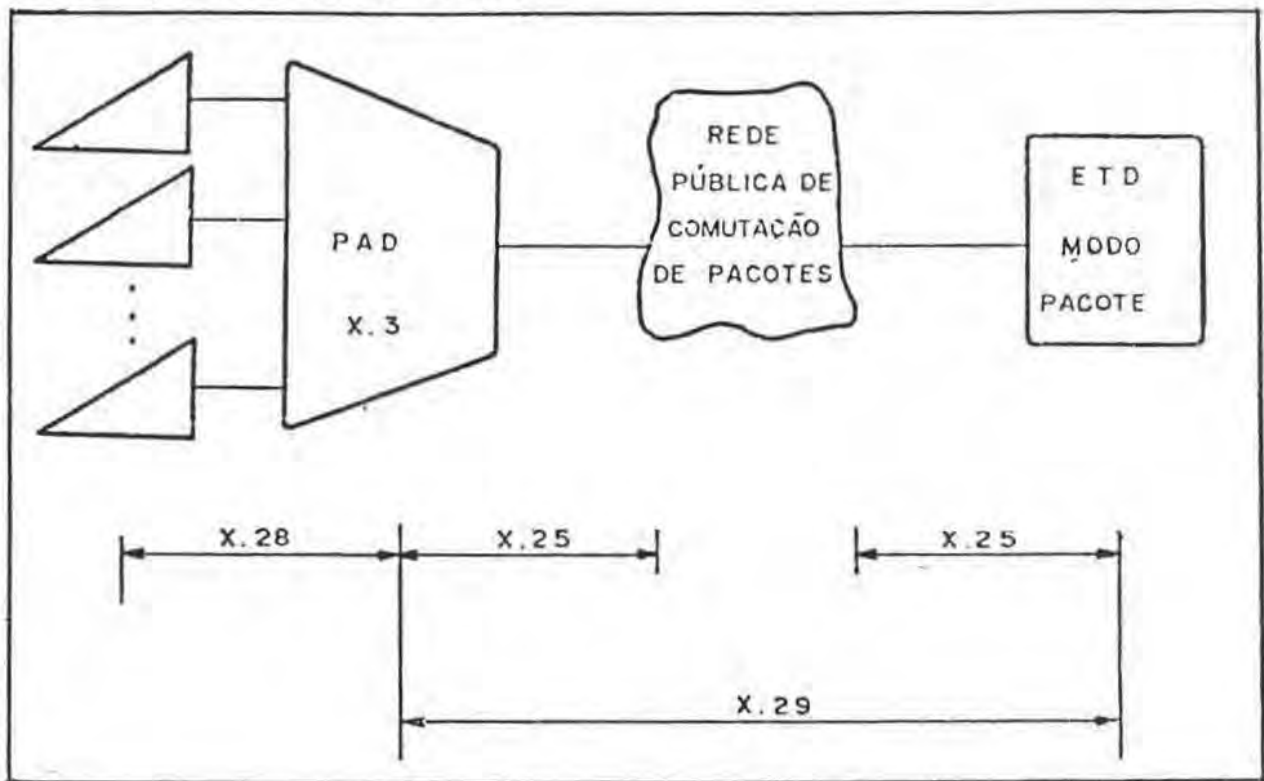


Fig. 1 - Protocolos na Comunicação com o PAD

Por outro lado, como o PAD é um equipamento destinado a processar tráfego proveniente de terminais assíncronos, e como o número de terminais que demandam este tipo de ligação pode variar drasticamente de uma localidade para outra,

ponderou-se que seria altamente desejável que o equipamento apresentasse facilidades de configuração quanto ao número máximo de terminais que pudesse aceitar. Como decorrência deste raciocínio, tornou-se um forte vínculo de projeto o desenvolvimento de uma máquina que apresentasse boas características de modularidade e expansibilidade. Definiu-se como sendo 120 o número máximo de terminais a serem aceitos pelo PAD, na sua configuração máxima, e como sendo 8 o número de interfaces com terminais a serem contidas em cada módulo.

Outro aspecto levado em consideração na fase de especificação foi o de elevar ao máximo possível a disponibilidade do equipamento, sem, no entanto aumentar demasiadamente a sua complexidade ou custo. Optou-se pelo desenvolvimento de uma máquina que permitisse redundância nos seus pontos vitais, reduzindo a probabilidade de falha total no equipamento. Esta configuração redundante está apresentada no item seguinte.

III. ARQUITETURA FÍSICA DO PAD

III.1 - ESTRUTURA DO PAD

Para construção do sistema físico destinado a suportar a implementação do PAD foi escolhida a família de componentes 8088/8089 da Intel, em virtude das suas boas características de desempenho e das facilidades disponíveis para construção de sistemas com múltiplos processadores. Foram definidos 2 tipos de módulos processadores, que funcionando cooperativamente possibilitam implementar as funções do PAD. (figura 2):

- . Módulo Processador de Linhas - MPL
- . Módulo Processador de Acesso à Rede - MPAR

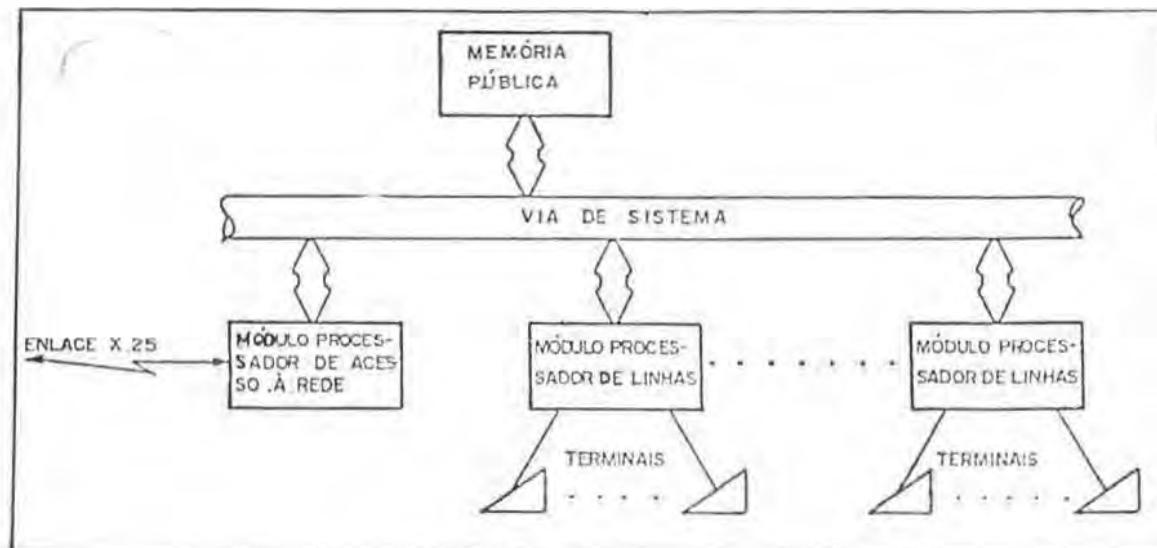


Fig. 2 - Módulos Processadores Componentes do PAD

III.2 - ESTRUTURA DOS MÓDULOS PROCESSADORES

O Módulo Processador de Linhas é aquele destinado a suportar a interface do PAD com os terminais assíncronos, oferecendo as facilidades especificadas na recomendação X.3. Neste módulo também residem os programas de implementação do protocolo X.28, de comunicação com o usuário. Cada MPL aceita conexões de até 8 terminais, cujas velocidades são detetadas automaticamente no início de cada sessão. O número de MPL's presentes no PAD pode ser configurado conforme necessidade. Pode-se ter acesso ao PAD também através da rede telefônica pública. Quando houver necessidade deste tipo de acesso, uma placa adicional deve ser acrescentada ao Módulo Processador de Linhas: a Interface com Linhas Discadas - ILD. A figura 3 apresenta as placas que compõem o MPL.

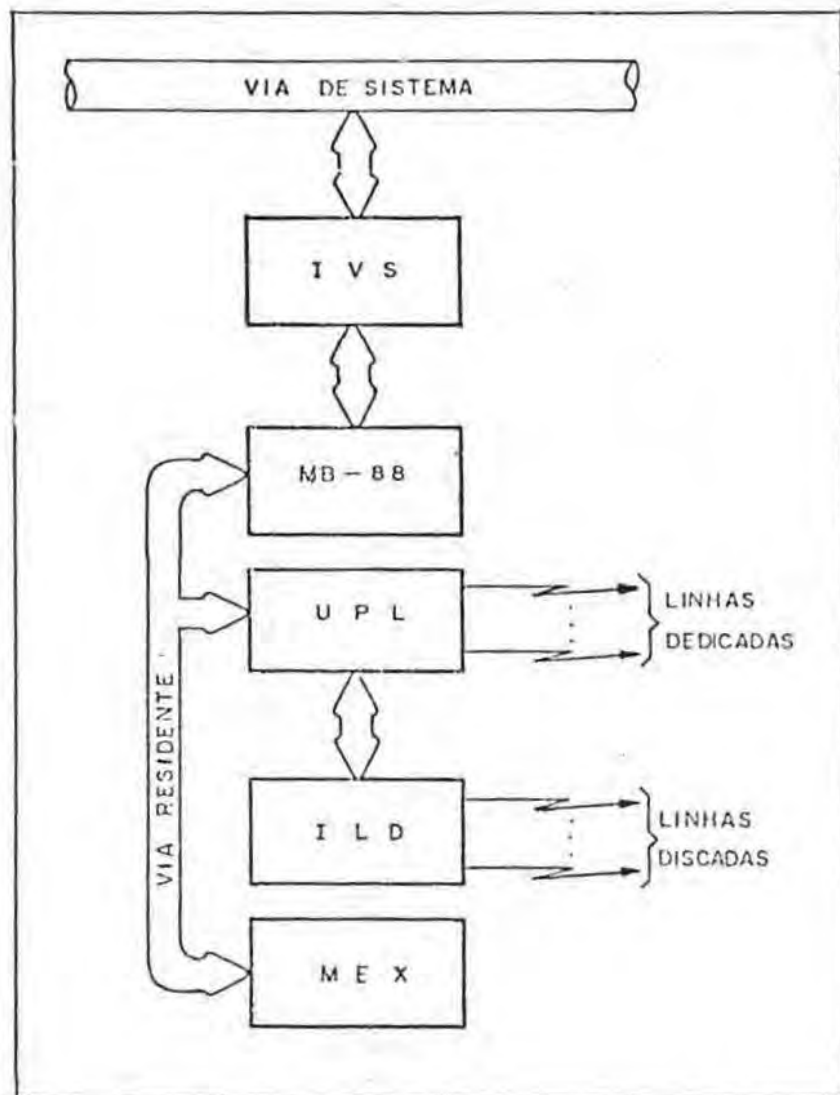


Fig. 3 - Placas do Módulo Processador de Linhas

O elemento principal constituinte dos MPL's, assim como do MPAR, é o Módulo Básico-88 (MB-88), que é uma placa que contém os processadores (8088 e 8089), controladores de interrupção, controladores de DMA, gerador e testador de paridade e demais circuitos para a geração dos sinais necessários à operação das demais placas que se acoplam a este módulo.

As demais placas componentes do MPL são: a Interface com a Via de Sistema (IVS), que contém as lógicas de con-

trole de acesso à via comum, ou via de sistema, a Unidade Processadora de Linhas (UPL), que possui as interfaces seriais para conexão aos terminais dos usuários e os geradores de "baud-rates" programáveis, a Interface com Linhas Discadas (ILD), que faz o tratamento da sinalização da rede pública discada, e a Memória de Expansão (MEX), que é uma placa de até 128 kbytes de memória RAM dinâmica para acomodar os programas de aplicação (protocolo X.28 e funções do X.3) e o software básico.

A configuração, em placas, do Módulo Processador de Acesso à Rede está esquematizada na figura 4.

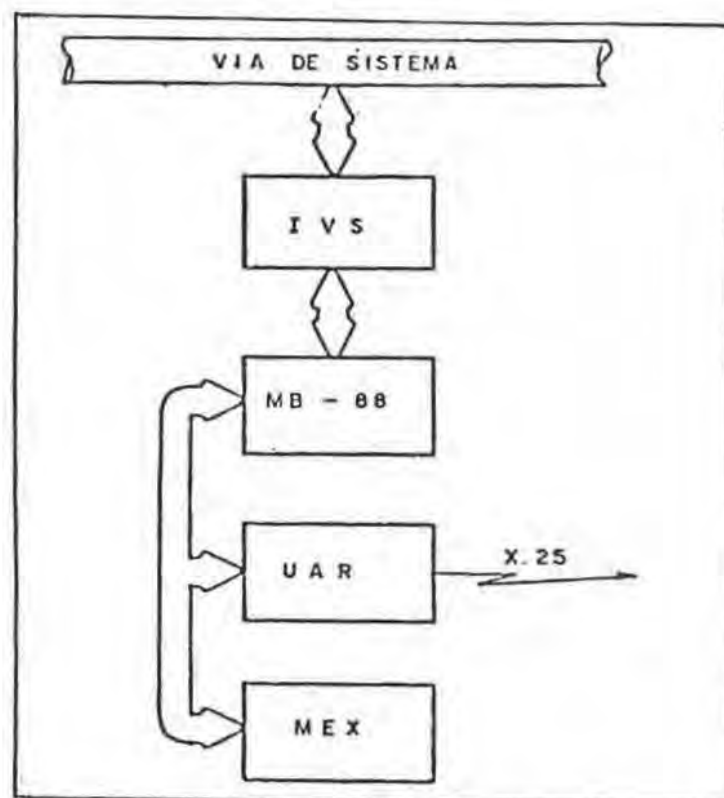


Fig. 4 - Placas do Módulo Processador de Acesso à Rede

Neste módulo o MB-88 e a IVS têm as mesmas funções descritas para o MPL. A placa Unidade de Acesso à Rede-UAR, por sua vez, possui o controlador de protocolo para implementação do enlace X.25 para comunicação com a rede de comutação de pacotes. A placa Memória de Expansão aloja os programas, neste módulo, dos protocolos X.25 e X.28 e o software básico.

A Memória de Sistema é constituída por uma placa Memória de Expansão - MEX de 128 kbytes. É destinada a suportar a troca de mensagens e dados entre os módulos MPL e MPAR, conforme será descrito no item seguinte.

III.3 - CONFIGURAÇÃO REDUNDANTE

Como se pode constatar na estrutura apresentada, este sistema físico apresenta alguns elementos centralizados vitais, cuja falha traz como consequência a perda total do sistema. Estes elementos são: a Memória Pública, o Módulo Processador de Acesso à Rede, a Via de Sistema e o enlace X.25. Desta forma, com o objetivo de aumentar a disponibilidade do sistema, todo o "hardware" foi projetado de forma a aceitar uma configuração redundante, permitindo que todos os elementos citados acima sejam duplicados. A figura 5 mostra um esquema do PAD em configuração redundante.

III.4 - O MECANISMO DE INTERRUPÇÕES ENTRE PROCESSADORES

Para que o mecanismo de comunicação entre módulos processadores opere de uma forma eficiente, foi elaborada uma infraestrutura de "hardware" capaz de dar suporte a estas comunicações, e de minimizar o "overhead" delas decorrente. Com este objetivo foi incorporado a cada módulo processador uma memória FIFO - "first-in-first-out", mapeada como dispositivo de E/S de sistema, sendo, portanto, acessível a todos os módulos processadores, mas somente para escrita. A leitura destas memórias FIFO pode ser feita somente pelo módulo processador ao qual ela pertence, e dela são recolhidos dados referentes à interrupções pedidas pelos demais módulos processadores. A leitura destas FIFO's é feita através da via residente do módulo processador, à qual está conectada como qualquer dispositivo de E/S, no que diz respeito à leitura. A figura 6 ilustra estas memórias FIFO e as suas cone-

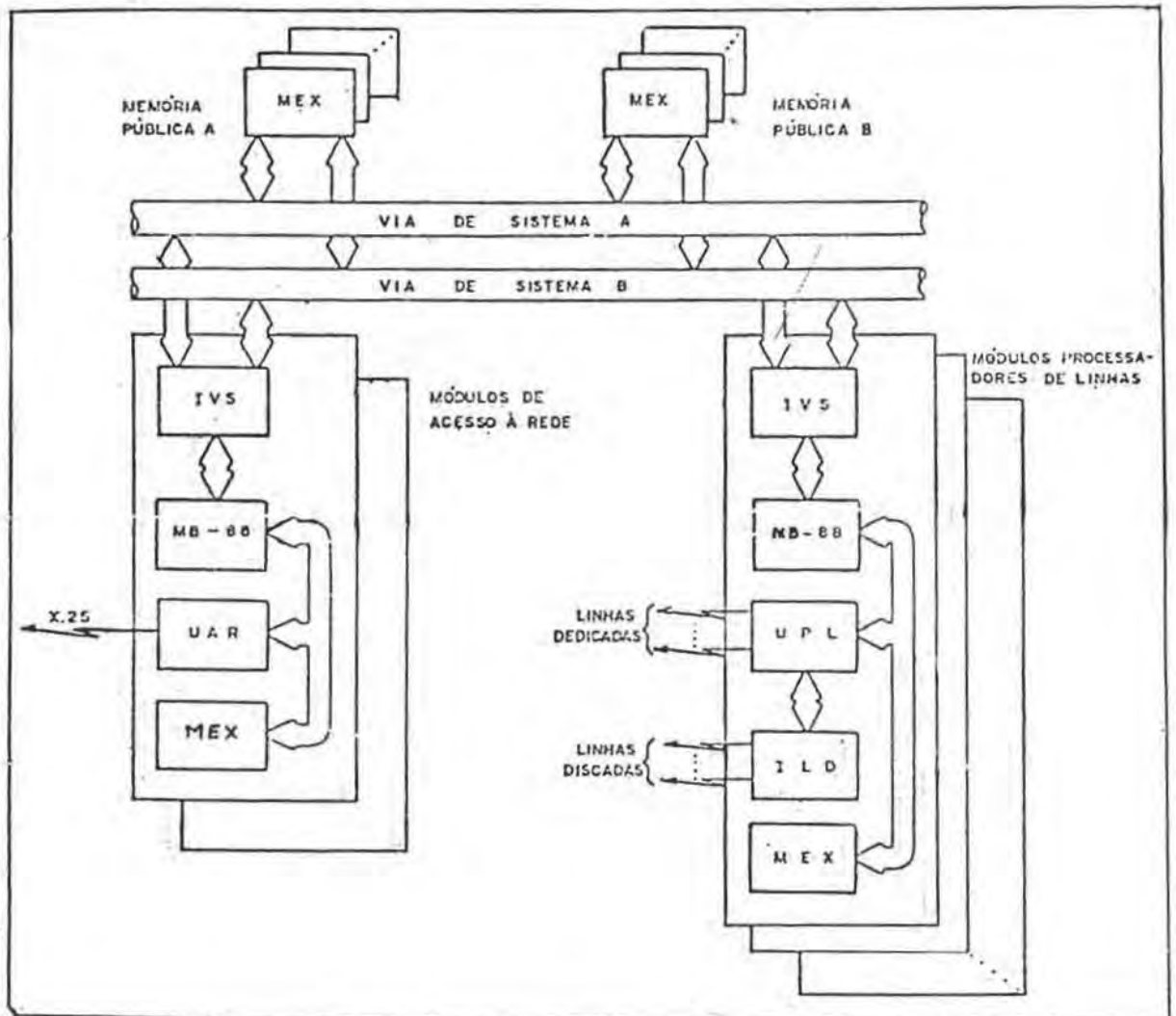


Fig. 5 - Estrutura do PAD em Placas de "Hardware"

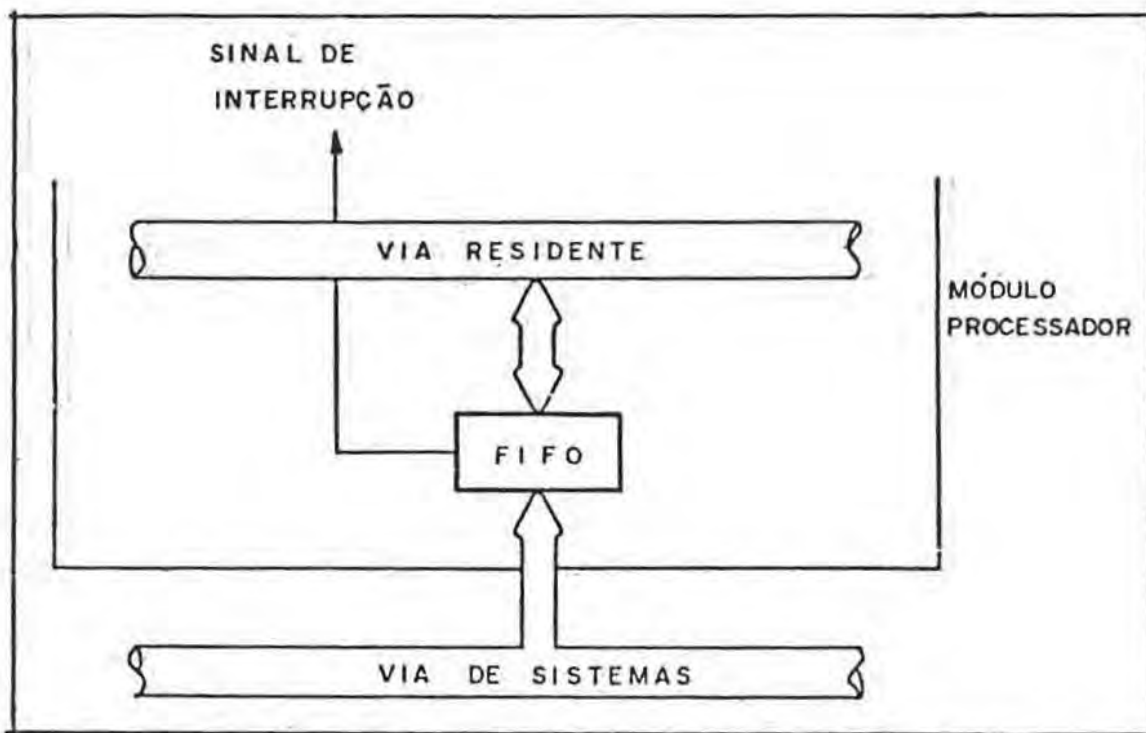


Fig. 6 - Conexão das FIFO's de Comunicação às Vias Residentes e de Sistema

xões às vias de sistema e residente.

Os campos dos bytes de interrupção entre processadores, através das FIFO's, são os mostrados na figura 7.



Fig. 7 - CAMPOS DO BYTE DE INTERRUPTÃO

Os 4 bits mais significativos identificam o processador que pediu a interrupção. Os demais bits identificam o tipo de interrupção. O uso dos campos será detalhado no item

a seguir.

IV - INFRA-ESTRUTURA DE COMUNICAÇÃO

De acordo com o mencionado no item anterior, existe um mecanismo de troca de informações entre os módulos processadores, cuja finalidade é prover uma estrutura capaz de dar suporte a estas comunicações.

Este mecanismo consiste no intercambio de mensagens entre o MPAR e os MPL's através das memórias FIFO e através da Memória de Sistema.

A comunicação via Memória de Sistema é feita utilizando-se o esquema de "mailboxes" (portas de comunicação), onde são armazenados endereços de buffers de informação na Memória de Sistema.

Por outro lado, o esquema de interrupções com o uso de FIFO's auxilia a troca de informações via portas de comunicação, no sentido de avisar aos respectivos módulos processadores, de forma rápida, a existência de mensagens a eles destinadas.

A seguir serão apresentados estes dois meios de comunicação.

IV.1 - COMUNICAÇÃO VIA MEMÓRIA DE SISTEMA

Esta memória é organizada em três segmentos principais, mostrados na figura 8, a saber:

- . Portas de comunicação do MPAR para os MPL's.
- . Portas de comunicação dos MPL's para o MPAR.
- . Área de Dados.

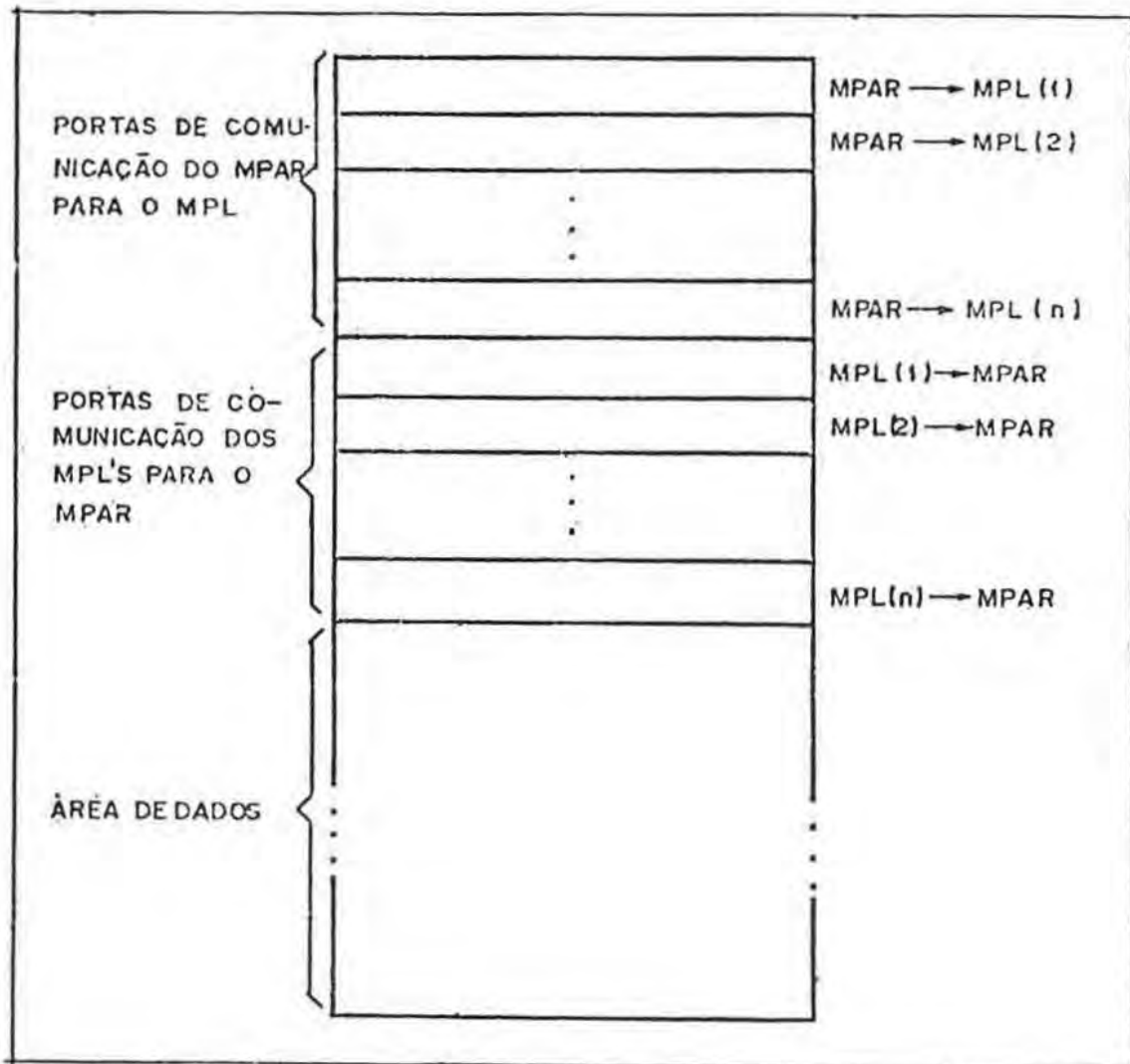


Fig. 8 - Organização da Memória de Sistema

As portas de comunicação são conjuntos de 3 bytes contendo o endereço de uma mensagem localizada na área de dados, como mostra a figura 9.

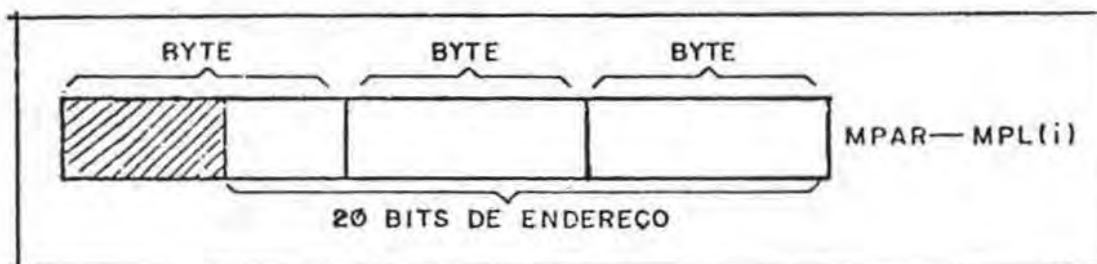


Fig. 9 - Portas de Comunicação MPAR + MPL(i) para Envio de Mensagens de Informação

Além de ponteiros para mensagens, cada uma dessas portas poderá conter, dependendo do caso, um ponteiro para uma área (blocos) livre na Área de Dados, requisitada por um MPL. O MPAR, sendo um elemento centralizador no que diz respeito à gerência da Memória Pública, recebe o pedido de concessão de um bloco livre de um MPL e atende-o a seguir, enviando-lhe, através da porta de comunicação, o endereço de um bloco livre.

A Área de Dados é subdividida em blocos, sendo que cada bloco contém um determinado número de "buffers" de 145 bytes cada. O número de "buffers" por bloco depende da taxa de utilização dos mesmos, pois, dependendo das circunstâncias, causa menos "overhead" alocar de uma vez um conjunto de "buffers" (bloco) maior a um MPL, ao invés de "buffers" isolados, tendo em vista que a cada pedido corresponde o envio de uma mensagem para a FIFO do MPAR, via o esquema de interrupções.

IV.2 - COMUNICAÇÃO VIA INTERRUPÇÕES

A comunicação entre os módulos processadores também é feito utilizando-se o mecanismo de interrupções e das FIFO's anteriormente descritas.

É por este mecanismo que são enviadas algumas mensagens necessárias para completar a comunicação via portas de comunicação; esta comunicação é feita, porém, de forma bastante rápida, aliviando o acesso à Memória de Sistema para a troca de mensagens de controle entre os processadores.

As mensagens de controle recebidas pelas FIFO's são as seguintes:

1 byte

MPAR	0011
------	------

AVISO DE MENSAGEM NA PORTA MPAR + MPL (i)
O MPAR avisa o MPL(i) que há uma mensagem para ele na área de dados cujo endereço está na porta MPAR + MPL(i).

MPL(i)	0000
--------	------

LIBERAÇÃO DE PORTA MPAR + MPL(i)
O MPL(i) informa no MPAR que já usou o conteúdo da porta, e portanto está liberada para uso posterior.

MPL(i)	0001
--------	------

LIBERAÇÃO DE BUFFER
O MPL(i) avisa ao MPAR que determinado "buffer" está livre.

MPAR	0000
------	------

LIBERAÇÃO DE PORTA MPL(i) + MPAR

MPL(i)	0011
--------	------

AVISO DE MENSAGEM NA PORTA MPL(i) + MPAR

MPL(i)	0010
--------	------

PEDIDO DE BLOCO
O MPL(i) pede um bloco livre na Memória de Sistema ao MPAR.

MPAR	0100
------	------

CONCEDE BLOCO
O MPAR cede um bloco livre ao MPL(i) que o pediu.

V. ARQUITETURA LÓGICA DO PAD

A arquitetura lógica do PAD consiste num conjunto de processos que implementam os protocolos X.3, X.28, X.29 e X.25 segundo as recomendações da CCITT.

Estes processos estão alojados nos módulos processadores, de acordo com a distribuição mostrada na figura 10. Esta distribuição física foi feita baseando-se em critérios que dizem respeito não apenas ao tamanho dos códigos comportados em cada módulo, mas sobretudo à minimização de troca de mensagens entre processos em módulos distintos, que normalmente envolvem o acesso à via de sistema, objetivando assim um desempenho global satisfatório, aliviando o uso desta via. Naturalmente, os processos fortemente acoplados foram alocados num só módulo.

Os processos implementados são os seguintes:

- PX3 - é responsável pela gerência de conjuntos de parâmetros associados a terminais, resumidos em tabelas de nominadas perfis, segundo os quais se caracteriza o funcionamento e a interação entre esses terminais e o PAD.
- PX28 - é responsável pelas seguintes funções:
- . Procedimentos de inicialização de uma ligação a nível terminal/PAD, a pedido do terminal.
 - . Procedimentos de intercâmbio de informações de controle entre terminal e PAD, tais como pedido de estabelecimento ou finalização de uma chamada virtual por parte do terminal, ou resposta do PAD quanto a esses pedidos e informes de erros de procedimento.
 - . Procedimentos de troca de dados entre terminal e PAD e controle de fluxo das informações.

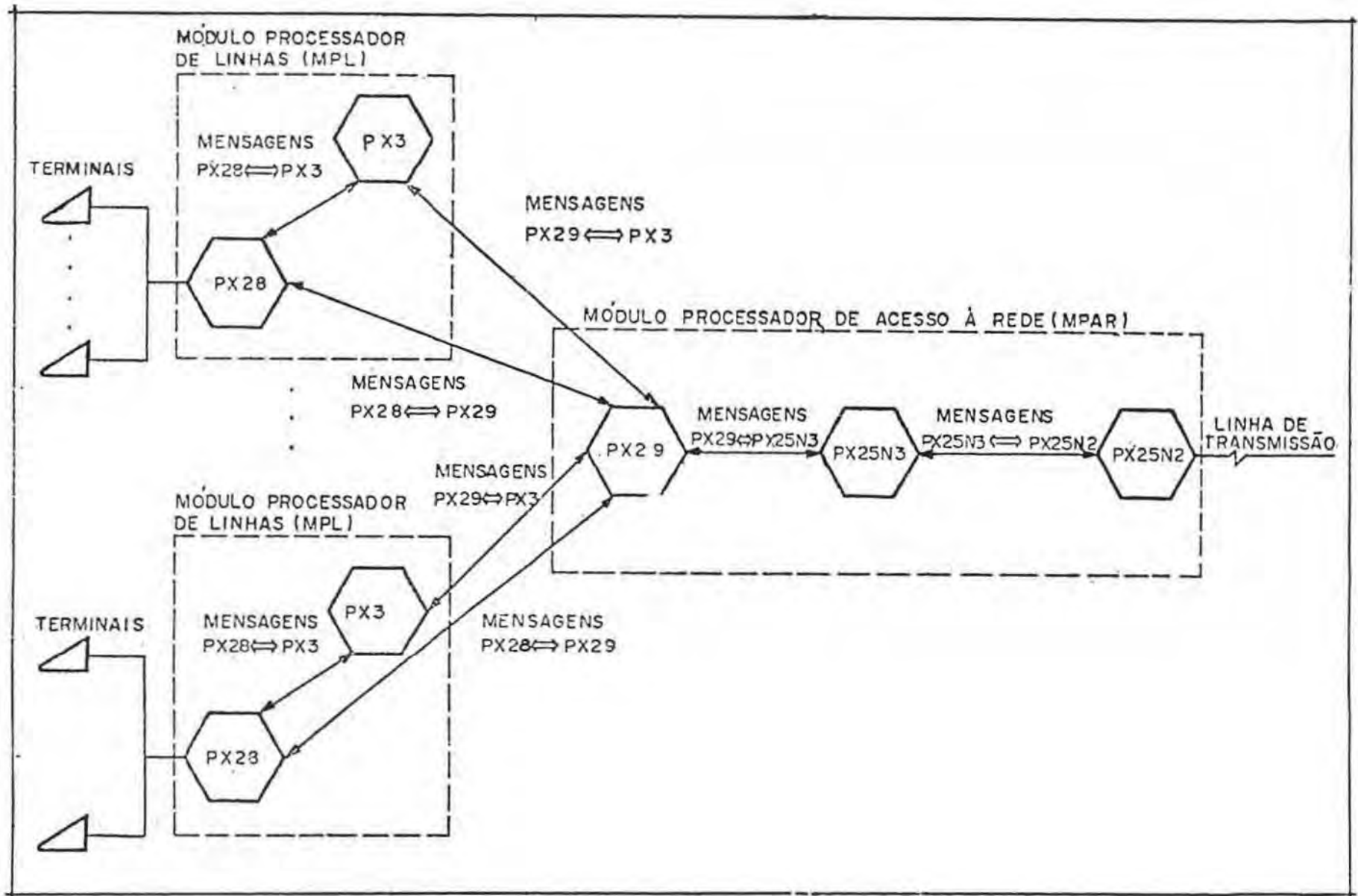


Fig. 10 - Arquitetura Lógica do PAD

- PX29 - é responsável pela interação entre o PAD e um computador remoto modo pacote, no sentido de explicitar a maneira de utilizar o protocolo X.25 para tornar viável o diálogo entre PAD e esse computador remoto, num nível superior ao do nível 3 do protocolo X.25. Suas funções básicas são:
- . Estabelecimento e término de uma chamada virtual a pedido do terminal, via protocolo X.28.
 - . Preparo dos pacotes no formato do nível 3 do X.25
 - . Intercâmbio de informações de controle entre o próprio PAD e o computador remoto.
- PX25N3 - é responsável pelos procedimentos de gerência de estabelecimento, vigência e término de chamadas virtuais (nível 3 do protocolo X.25)
- PX25N2 - é responsável pelos procedimentos de gerência das funções correspondentes ao nível de enlace (nível 2 do protocolo X.25)

Estes processos constituem a aplicação propriamente dita, implementada no PAD. Esta aplicação foi desenvolvida sobre uma infra-estrutura de suporte denominado Sistema de Comunicação [RUGG 80a] que, além de ser responsável pela escalação e ativação de tais processos, e gerenciamento de memória, provê ao usuário uma série de primitivas que viabilizam a troca de mensagens entre os processos acima descritos, como mostra a figura 10. Algumas destas primitivas são:

- . Envio de uma mensagem de um processo para outro.
- . Espera de chegada de uma mensagem para um processo.
- . Teste de existência de alguma mensagem para um processo.

- . Acionamento de uma temporização.
- . Suspensão de uma temporização.

O sistema de Comunicação permite e gerencia a troca de mensagens tanto entre processos residentes no mesmo módulo processador, como entre processos localizados em módulos distintos.

VI - CONCLUSÕES

O PAD foi inteiramente desenvolvido no Laboratório de Sistemas Digitais da Escola Politécnica, no prazo de um ano. Sendo um protótipo de laboratório, a tecnologia empregada na sua construção é a de "wire-wrap", num total de 10 placas.

O sistema, atualmente, encontra-se totalmente testado e em operação. A grande vantagem do equipamento é que ele permite que usuários possuidores de terminais assíncronos comuns possam ter acesso a recursos disponíveis numa rede pública de comutação de pacotes, sem a necessidade de ter implantados os protocolos do acesso à rede (X.25) nesses terminais.

REFERÊNCIAS BIBLIOGRÁFICAS

- [BARB 82] Barbosa, A.S., Ruggiero, W.W., Moscato, L.A., Campos, E.G.L., Stiubiener, S., "Rede Local no Laboratório de Sistemas Digitais da Escola Politécnica da Universidade de São Paulo", II SLARC-Simpósio Latino Americano sobre Redes de Computadores, São Paulo, junho de 1982.

- [CCIT X3] CCITT Provisional Recommendation X.3, "Packet Assembly/Disassembly Facility (PAD) in a Public Data Network", Genebra, 1980.
- [CCIT X25] CCITT Recommendation X.25, "Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment for Terminals Operating in the Packet Mode on Public Data Networks", 1977.
- [CCIT X28] CCITT Recommendation X.28, "DTE/DCE Interface for a Start-Stop Mode Data Terminal Equipment Accessing the Packet Assembly/Disassembly Facility (PAD) in a Public Data Network Situated in the Same Country", Genebra, 1980.
- [CCIT X29] CCITT Provisional Recommendation X.29, "Procedures for the Exchange of Control Information and User Data Between a Packet Mode DTE and a Packed Assembly/Disassembly Facility (PAD)", Genebra, 1980.
- [MONT 82] Monteiro, J.A.S., "Descrição, Validação e Geração Automática de Implementação de Protocolos"; Dissertação de Mestrado, Escola Politécnica da USP, São Paulo, 1982.
- [RUGG 80] Ruggiero, W.W., Barbosa, A.S., "Implementação de uma Central X.25 de Comutação de Pacotes através da Aplicação de uma Arquitetura Distribuída de Multi-microprocessadores", VII Seminário Integrado de Software e Hardware, 1980.

- [RUGG 80a] Ruggiero, W.W., Melnikoff, S.S.S., Lucena, C.J.,
"Implementação de um Sistema de Comunicação entre Processos através de uma Máquina de Arquitetura Distribuída", VII Seminário Integrado de Software e Hardware, 1980.
- [STIU 81] Stiubiener, S., "Protocolo de Transporte para Redes de Computadores", Tese de Doutorado, Escola Politécnica da USP, 1981.

AUTORES

EDIT G.L.DE CAMPOS

ARMANDO S. BARBOSA

Departamento de Engenharia de Eletricidade
Escola Politécnica da USP
A. Luciano Gualberto - Travessa 3, nº 158
Cidade Universitária "Armando de Salles Oliveira"
Caixa Postal 11.455 - CEP 01000
São Paulo, S.P.
Telefone: 211-2122, R. 392 ou R. 200

EXPERIÊNCIA NO PROJETO E IMPLEMENTAÇÃO
DA FEDE LOCAL REDPUC

Daniel A. Menascé
Departamento de Informática, PUC/RJ

Luiz Fernando Gomes Soares
Departamento de Engenharia Elétrica, PUC/RJ

Daniel Schwabe
Departamento de Informática, PUC/RJ
22453 Rio de Janeiro, RJ

1. Introdução

No início de 1981, teve início na PUC/RJ o projeto de uma rede local de microcomputadores. Esta rede, que seria confinada, isto é, localizada em um único "rack", serviria de base à implementação de uma central de comutação com arquitetura distribuída [1]. Após o término da especificação preliminar, chegou-se à conclusão que o projeto da rede local deveria ser transformado no objetivo principal, em detrimento do término da central de comutação. Esta mudança de ênfase pode ser justificada por três razões. Em primeiro lugar, o Brasil já estava importando tecnologia na área de comutação por pacotes e o CPqD da Telebrás já estava desenvolvendo uma central de comutação para redes públicas. Em segundo lugar, o projeto preliminar da rede local havia se mostrado bastante promissor e interessante. Finalmente, uma rede local seria um produto de maior utilidade e aplicabilidade; uma vez concluído, possibilitaria a implementação de uma rede local não confinada no campus da PUC/RJ.

A partir daquela época, os esforços voltaram-se para o projeto detalhado e implementação do hardware e software da rede local, que passou a chamar-se REDPUC.

Uma primeira decisão que foi adotada logo no início do projeto foi a de que toda a tecnologia do projeto seria desenvolvida na PUC/RJ, evitando-se assim copiar ou adaptarou

tros tipos de redes desenvolvidas no exterior. O esforço culminou no desenvolvimento de todos os protocolos da REDPUC, e no projeto do hardware e software que os implementam.

A REDPUC e seus protocolos já foram descritos em artigos recentes dos mesmos autores [2,3]. Este artigo tenta apresentar um relato de experiência obtida na sua implementação, ressaltando alguns aspectos que podem vir a servir de subsídios a outros grupos que atuam na mesma área.

A seção dois revê brevemente a arquitetura da REDPUC e seus protocolos. A seção 3.1 descreve a implementação do nível dois do protocolo de acesso ao embarramento, evidenciando a relação entre o hardware e o software. A próxima subseção apresenta o Núcleo de Sistema Operacional do processador de comunicações (PC) responsável pela implementação do Protocolo de Acesso ao Embarramento (PAE).

A implementação do nível três é ilustrada na seção 3.3 e o ambiente de implementação é descrito na seção quatro. Finalmente, o artigo conclui com uma descrição dos planos de pesquisa para os próximos anos.

2. Arquitetura da REDPUC

REDPUC é uma rede local utilizando como sistema físico de interconexão até dois embarramentos globais seriais compartilhados pelos diversos processadores de comunicação (PC). Estes podem ser desligados de um ou ambos os embarramentos em condições de falhas irrecuperáveis, neles mesmos, ou na própria estação (MB+ME - vide figura 1). A estrutura da rede é apresentada na figura 1.

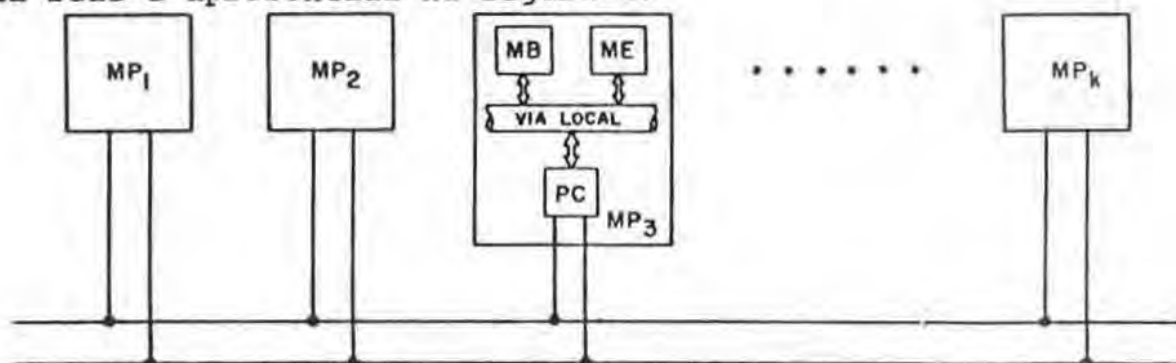


Figura 1 - A Estrutura da REDPUC

A comunicação entre os módulos é realizada por um protocolo de comunicações (Protocolo de Acesso ao Embarramento-PAE) que compreende três níveis independentes entre si.

O primeiro nível, nível físico, especifica as características físicas e elétricas do PAE. As características importantes deste nível, vista pelo nível imediatamente acima são: transmissão serial, síncrona, full-duplex em um sistema de comunicação do tipo difusão (broadcast) usando técnica de codificação Manchester.

O segundo nível, nível de embarramento, é o responsável pela ligação e desligamento de um módulo na rede, bem como do controle de acesso ao embarramento e detecção de erros. Para tanto utiliza um protocolo do tipo passagem de permissão ("token passing") com o controle realizado por uma lista denominada lista de controle de ciclos [2, 4]. Este nível implementa um serviço do tipo datagrama não confiável [5] , pois não realiza controle de erro, apenas existe uma grande probabilidade que a mensagem chegue a seu destino sem erro.

O terceiro nível, nível de transporte, converte a linha física propensa a erros, em uma ligação lógica, relativamente livre de erros, para a transferência de dados entre o transmissor e o receptor. Mais ainda, possibilita ao usuário, caso deseje, a transformação da ligação lógica em vários canais lógicos, realizando a função de um concentrador. Este nível realiza o controle de erro e de fluxo nos vários canais lógicos, bem como a sequenciação de mensagens, se desejada pelo usuário, nos mesmos. Este nível vai implementar desta forma dois tipos de serviço: datagrama confiável e circuito virtual.

Cabe aos processadores de comunicação (PC's) a realização do protocolo de comunicação PAE, bem como a interface com a estação. Cada PC possui capacidade de processamento tornando-o adequado não só para ligação com estações inteligentes, mas também para outras funções tais como serviço específico para rede (p.ex., estação de impressão) ou ligação com dispositivos pouco inteligentes (p.ex., ligação de ter-

minais à rede), entre outras.

3. Arquitetura de Software da REDPUC

3.1 - PAE2 - Interação hardware/software

A figura 2 apresenta um diagrama simplificado do Processador de Comunicação (PC), responsável pela interface com a estação usuária e pela realização do protocolo de acesso ao embarramento

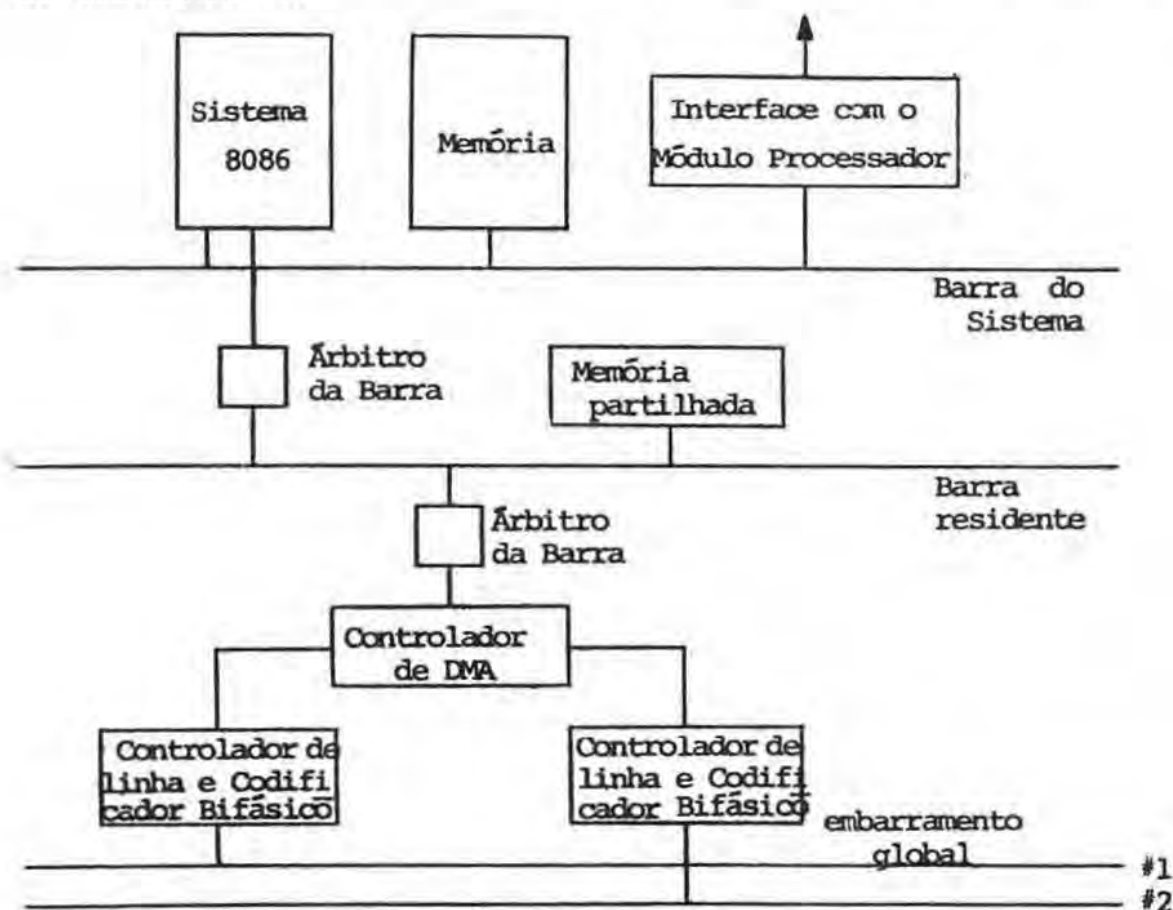


Figura 2 - Diagrama simplificado do Processador de Comunicação

A primeira dificuldade no projeto do PC foi a determinação de que tarefas seriam realizadas por software e quais seriam realizadas por hardware. Esta dificuldade resultou principalmente pela falta de circuitos integrados adequados ao projeto, e pelas altas taxas de transferência exigidas,

pois cada embaraamento deveria poder aceitar uma taxa de até 10 Mbps.

A ligação ao embaraamento foi realizada através do controlador de comunicação 2652-1 da Signetics e de circuitos para sincronismo, codificação e decodificação bifásica. A escolha deste controlador se deu pelo fato dele ser na época do projeto, o de maior velocidade encontrado no mercado (2 Mbps). Esta escolha trouxe como consequência um hardware adicional complexo e a introdução de um preâmbulo desnecessário ao quadro de nível 2 do PAE.

Um outro problema que se apresentou foi o da escolha do processador responsável, entre outras tarefas, pela realização do nível 3 e parte do nível 2 do PAE. A ligação do controlador de comunicação (2652-1) a este sistema poderia ser realizada de três modos: memória chaveada, por DMA e memória partilhada.

Na memória chaveada, um banco de memória ficou alocado a um só sistema por vez (2652-1 ou processador) podendo ser chaveada de um sistema para outro. Esta solução foi descartada pelo grande número de circuitos integrados que ela exigiria para sua realização.

A solução por DMA é inviável uma vez que a latência no atendimento ao pedido de um ciclo, bem como a realização deste ciclo de DMA, poderia gastar um tempo bem maior do que a chegada de dois caracteres a serem transferidos.

A solução de memória partilhada foi a adotada, uma vez que praticamente elimina o tempo de latência no atendimento de um DMA. Tal solução influenciou diretamente na escolha do processador. Sendo esta memória partilhada pelo 2652 através de DMA e pelo processador, ver figura 2, é necessário que o tempo de acesso à mesma seja o menor possível. A escolha de um processador de 16 bits teve aí seu fundamento e recaiu no 8086 da INTEL por três motivos: primeiro que o sistema 8086 foi desenvolvido para uma arquitetura "multibus" que facilita a realização da memória partilhada, segundo porque se sabia que em breve estaria no mercado o microprocessador 80186 que reduziria em muito o empacotamento do siste

ma; finalmente, porque vários pacotes de software desenvolvidos e que se utilizariam na implementação (p.ex., compilador C) existiam para o sistema 8086.

A realização de parte do nível 2 do PAE no processador 8086 trouxe implicações no software. As rotinas de atendimento das interrupções do PAE2 deveriam ser bastante eficientes e curtas, devido aos requisitos do protocolo. O problema surge do fato que nestas rotinas vários testes devem ser realizados para a escolha do procedimento adequado. O sistema 8086 só pode realizar estes testes sequencialmente o que ocasionaria um dispêndio de tempo precioso. A solução encontrada foi a de separar as rotinas de atendimento das interrupções em co-rotinas, selecionadas não pelo teste de variáveis, mas por modificações no vetor de interrupção.

Vários outros pequenos problemas, principalmente na geração das interrupções do nível 2 do PAE, foram resolvidos com introdução de hardware adicional, por exemplo, o fato do controlador de interrupção 8259 mascarar as interrupções e não inibi-las ocasionou mudanças na implementação do protocolo, bem como a utilização de circuitos integrados extras. Devido a todos os problemas encontrados a conclusão a que se chegou é que uma melhor solução seria a de se fazer uma interface com o embarramento microprogramada, sendo esta responsável por toda a realização do nível 2, liberando o processador 8086 para a realização do nível 3 e/ou aplicativos. Tal interface, em fase de projeto, vai atingir uma velocidade de de 10 Mbps por embarramento, e está sendo realizada de forma a poder ser facilmente integrada em um ou poucos circuitos integrados a curto prazo.

3.2 - O Núcleo de Sistema Operacional

Fornecendo a estrutura básica de software para a implementação dos diversos protocolos na REDPUC, encontra-se o núcleo do sistema operacional. O nível 2, como foi visto, é formado por um conjunto de co-rotinas que são ativadas essencialmente pelas interrupções de hardware.

O nível 3, que será descrito com mais detalhes na próxima seção, é formado por um conjunto de processos que executam em paralelo, que se comunicam entre si através de mensagens.

O núcleo do sistema operacional é responsável pelo escalonamento destes processos, além de oferecer as primitivas para a troca de mensagens e para gerência de memória. Além disso, o núcleo permite a multiplexação dos relógios de hardware em vários relógios lógicos que podem ser usados pelos vários processos.

As mensagens que podem ser trocadas pelos processos são de tamanho fixo (4 bytes), e podem ser de dois tipos: normais e do núcleo (urgentes). As mensagens do núcleo são utilizadas para comunicar aos processos eventos urgentes, tipicamente resultantes de interrupções de hardware.

Devido ao fato de que os processos que implementam o nível 3 serem fixos, pequenos e bem conhecidos, torna-se desnecessária a multiprogramação de processos através de "time-slices". Em vez disso, cada processo executa sem ser interrompido pelo núcleo (as interrupções do nível 2 são transparentes) e é responsável por devolver o controle ao núcleo em intervalos de tempo razoáveis.

O escalonamento de processos é feito de acordo com um esquema de prioridades. A prioridade de um processo é dada pela soma de uma prioridade estática e uma prioridade dinâmica¹. A prioridade estática tem três componentes: uma prioridade estática inicial, que reflete a importância relativa dos diversos processos; um modificador referente ao número de mensagens normais que o processo tem na sua fila de mensagens normais (quanto maior a fila, maior a prioridade); e um outro modificador, referente às mensagens urgentes.

A prioridade dinâmica de um processo pronto para executar é aumentada cada vez que um outro processo é escalonado. Desta forma, impede-se que um processo de baixa prioridade estática fique sem executar indefinidamente.

1 - O dinamismo da prioridade aqui é visto em relação ao processo de escalonamento.

Para a troca de mensagens o núcleo oferece primitivas que permitem o envio e recebimento de mensagens de forma bloquead e não-bloqueada. Quando um processo envia uma mensagem, esta é copiada para uma área interna do núcleo. Se não houver espaço no núcleo, o processo fica ou não bloqueado dependendo da primitiva executada.

Analogamente, quando um processo recebe uma mensagem, esta é copiada da área do núcleo para a área do processo. O processo pode ficar bloqueado se não houver mensagem para ele, dependendo da primitiva executada.

O núcleo oferece também primitivas para a alocação e liberação de buffers de tamanho fixo; no momento a gerência de memória é feita da forma mais simples possível.

O hardware do PC oferece um relógio para uso do software. Para permitir uma melhor utilização deste relógio, o núcleo permite que os processos programem vários relógios lógicos, que são multiplexados no relógio físico. Estes relógios são usados essencialmente como um serviço de "despertador".

3.3 - Arquitetura de Software do Nível 3 do PAE

O nível 3 do protocolo de acesso ao embarramento é implementado através dos seis processos ilustrados na figura, cuja descrição será feita a seguir..

O processo TX3 (Transmissor do Nível 3) ao ser ativado pelo Núcleo, varre a Fila de Transmissão do Nível 3 (FTX3) à procura da primeira mensagem a ser enviada através de um canal lógico, x cuja janela permita a transmissão (a janela de cada canal lógico tem tamanho igual a 1). Esta mensagem é então colocada na Fila de Transmissão do Nível 2 (FTX2). Ao mesmo tempo, é ligado o temporizador correspondente ao canal lógico x e um ponteiro para o "buffer" que contém a mensagem é guardado para uma eventual necessidade de retransmissão por esgotamento da temporização.

O processo RX3 (Receptor do Nível 3) retira a primeira mensagem da Fila de Recepção do Nível 2 (FRX2) e utiliza

o seu cabeçalho para atualizar as variáveis de controle de erro e controle de fluxo, e desligar a temporização dos canais lógicos que receberam confirmações de recebimento. Caso a mensagem recebida seja de informação ela é colocada na Fila de Recepção do Nível 3 (FRX3), sem o cabeçalho do Nível 3, porém precedida do número de canal lógico ao qual ela se destina. Caso a mensagem seja de supervisão, o processo SUP3 recebe uma mensagem de RX3 indicando o "buffer" onde se encontra a mensagem.

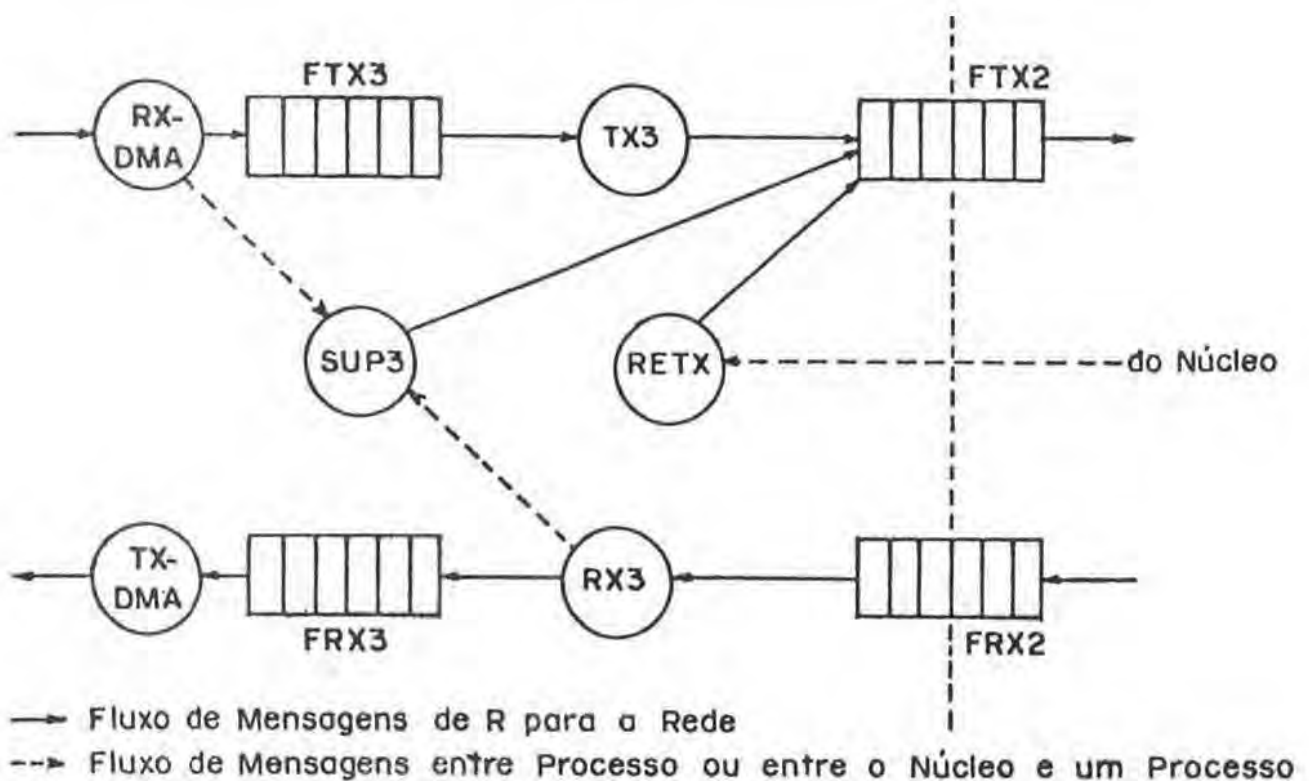


FIGURA 3 - PROCESSOS QUE IMPLMENTAM O NÍVEL 3

O processo SUP3 (Supervisor do Nível 3) se encarrega de realizar os procedimentos de estabelecimento e término

de conexões de transporte. As mensagens de supervisão recebidas da rede, via RX3, são analisadas por SUP3 e eventuais respostas são colocadas por SUP3 em FTX2. Os pedidos de estabelecimento ou término da conexão, localmente gerados, são entregues pelo processo RX-DMA a SUP3, que se encarrega de processá-los.

O processo RFTX (Retransmissor do Nível 3), ao receber o controle do Núcleo, examina se a temporização para algum canal lógico já expirou. Em caso afirmativo, ele recoloca a mensagem adequada em FTX2 para retransmissão e religa o temporizador.

Finalmente, os processos RX-DMA (Receptor de DMA) e TX-DMA (Transmissor de DMA) são responsáveis pela programação dos canais de DMA para recepção e envio de dados entre o MP e o PC.

4. Ambiente de Desenvolvimento

O software para a REDPUC está sendo desenvolvido em um módulo básico configurado com 128 k de memória, dois disquetes, um terminal de vídeo e uma impressora. No momento, outros sistemas de desenvolvimento com configuração semelhante estão sendo construídos.

A linguagem utilizada na codificação de 99% dos programas é C [6]. Apenas pequenos trechos de programa, no núcleo do nível 3 e partes do nível 2 foram escritos em linguagem assembler do 8086.

Em termos de espaço de memória consumido, o código implementando o nível 2 ocupa aproximadamente 2k bytes, o código do núcleo ocupa em torno de 3k bytes (mais 3 a 4k bytes de dados), e o código do nível 3 ocupa aproximadamente 7k bytes. Devemos ressaltar o fato de que estes números são obtidos com um compilador que produz código pouco otimizado; acreditamos que estes números possam ser reduzidos em torno de 20% por uma otimização mais cuidadosa.

5. Pesquisas Futuras

Como consequência do trabalho descrito neste artigo, foram iniciados diversos projetos na área de redes locais e automação de escritórios. Estes projetos visam consolidar a rede local desenvolvida na PUC/RJ, incluindo o projeto de interfaces para diversos microcomputadores brasileiros.

Além disso, estão sendo projetados diversos servidores especializados, tais como servidores de disco, servidor de impressora, e uma comporta X.25.

Com base nesta infraestrutura serão desenvolvidas aplicações do tipo transferência de arquivos, correio eletrônico, pastas eletrônicas, agendas e lembretes.

Finalmente, o grupo de redes da PUC/RJ está trabalhando na especialização de uma proposta de protocolo padrão de acesso a redes locais no nível imediatamente abaixo do nível de transporte.

Bibliografia

- [1] GOMES SOARES, L.F.; MENASCÉ, D.A. "Sobre a Evolução da Arquitetura de Centrais de Comutação de Pacotes". Anais do XV Congresso Nacional de Informática, outubro de 1982.
- [2] GOMES SOARES, L.F.; MENASCÉ, D.A. "Um Protocolo para Redes Locais do Tipo Difusão". Anales de la IX Conferencia Latinoamericana de Informatica, vol.2, PANEL INFO'82, Lima-Peru, agosto de 1982.
- [3] MENASCÉ, D.A.; GOMES SOARES, L.F.; JEFFERSON, F., CORREA C.H., "Descrição de Hardware e Software da Rede Local, REDPUC". Anais do XV Congresso Nacional de Informática, outubro de 1982.
- [4] MENASCÉ, D.A.; GOMES SOARES, L.F., "Arquitetura de uma Máquina Distribuída". Relatório Técnico, PUC/TELEBRÁS, Documento p 2, Laboratório de Engenharia e Sistemas de Computação, outubro de 1981.

- [5] MENASCÊ, D.A.; SCHWABE, D. "Redes de Computadores - Aspectos Técnicos e Operacionais". Terceira Escola de Computação, PUC/RJ, 1982.
- [6] KERNIJHAN, B.W.; RITCHIE, D. "The C Programming Language". Prentice Hall, 1978.

RELIME - Rede local do IME*

Ivan Melo de Carvalho**

Emmanuel Lopes Passos***

Sumário

Este trabalho mostra a evolução do projeto RELIME* com relação ao desenvolvimento de seus níveis 1 (físico) e 2 (enlace). Especifica a rede quanto à topologia, ao método de controle de acesso e à estrutura dos nós no tocante ao controle de enlace, ao processamento do método de acesso e à forma de interface com o hospedeiro.

Apresenta também a forma como foi implementada uma versão simplificada da rede, utilizando-se como hospedeiros 2 microcomputadores Schumec da Seção de Informática do IME.

* Trabalho financiado pela Finep - projeto 83

** Engenheiro Eletrônico, o aluno de mestrado em Informática do IME

*** Professor Adjunto do Programa de Pós-Graduação em Informática do IME.

INTRODUÇÃO

Rede local caso particular das grandes Redes, numa área limitada a 2,5 km e com recursos computacionais de uma única empresa.

Este trabalho tem como objetivo especificar uma rede local para microcomputadores e realizar uma implementação simplificada em "hardware" desta rede, desenvolvendo até o nível 2. A topologia é tipo "barra" e o método de acesso CSMA/CD¹.

A estrutura dos nós é Hospedeiro, Interface, Controlador e Transceptor, com protocolos Hospedeiro-nó e nó-nó. Nó-nó é basicamente uma parte do HDLC em conjunto com o mecanismo de controle de acesso CSMA/CD. Isto quer dizer que durante a transmissão de quadros em que não hajam colisões prevalece essencialmente o estabelecido no protocolo HDLC.

Desenvolvimento baseado nos princípios da rede Ethernet, cujas características conhecidas são: 10 Mbps, cabo coaxial, até nível 2 (tipo HDLC), CSMA/CD, arquitetura em camadas separando níveis 1 e 2, simples, barato, flexibilidade de do endereçamento etc.

2 - Desenvolvimento

Com relação a constituição das redes observa-se

que os fabricantes^{2,3} desenvolvem controladores (nível 2) do tipo padrão ou universais ("universal") que de um lado são ligados a um transceptor ("transceiver") que por sua vez liga-se ao meio físico da rede e por outro lado são ligados de forma paralela às interfaces com diversos tipos de "buses" e existentes (QBUS, UNIBUS, MULTIBUS, S100, etc).

Este tipo de configuração é exemplificada na figura 1 onde se observa a aplicação do conceito de modularidade.

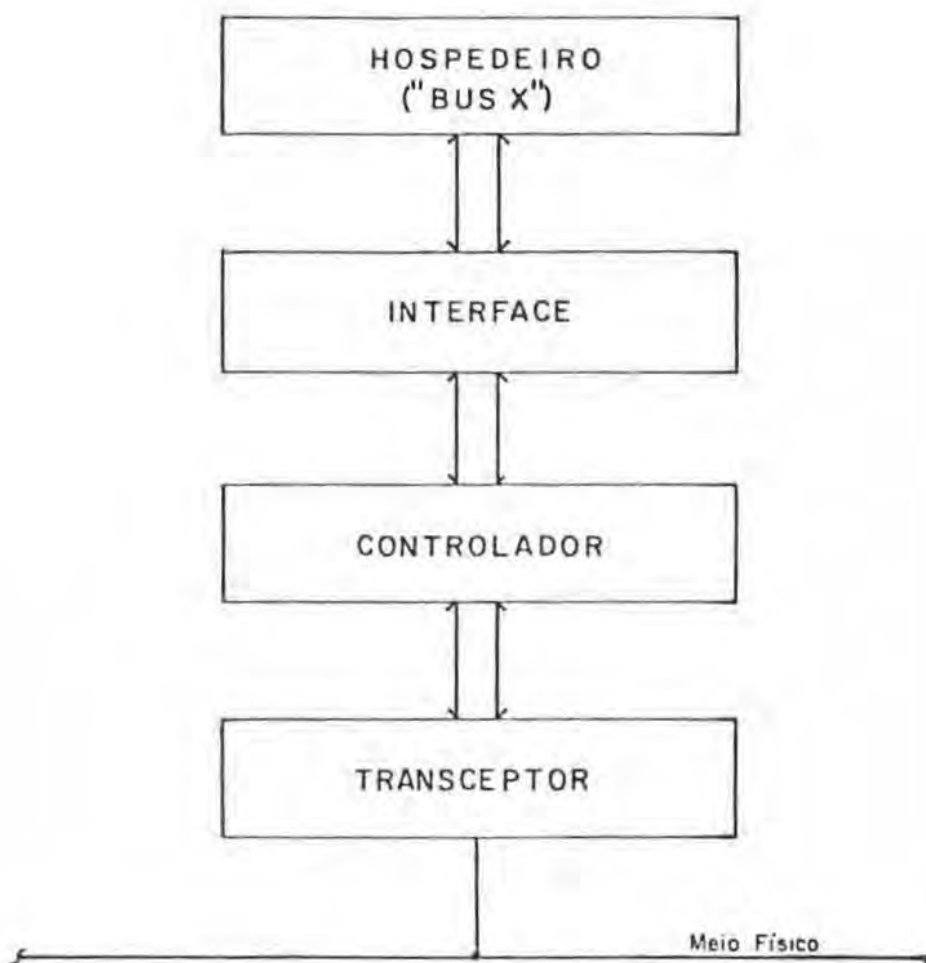


Figura 1 : Constituição de uma Rede Local.

A construção física de interface e do controlador é feita, na maioria dos casos, numa única placa com dimensões compatíveis com as placas do computador hospedeiro em questão, de forma que a mesma possa ser instalada dentro do computador, como mostrado na figura 2. Fica, assim, o nó da rede parcialmente embutido no hospedeiro permanecendo fora apenas o transceptor e os cabos de conexão do mesmo.

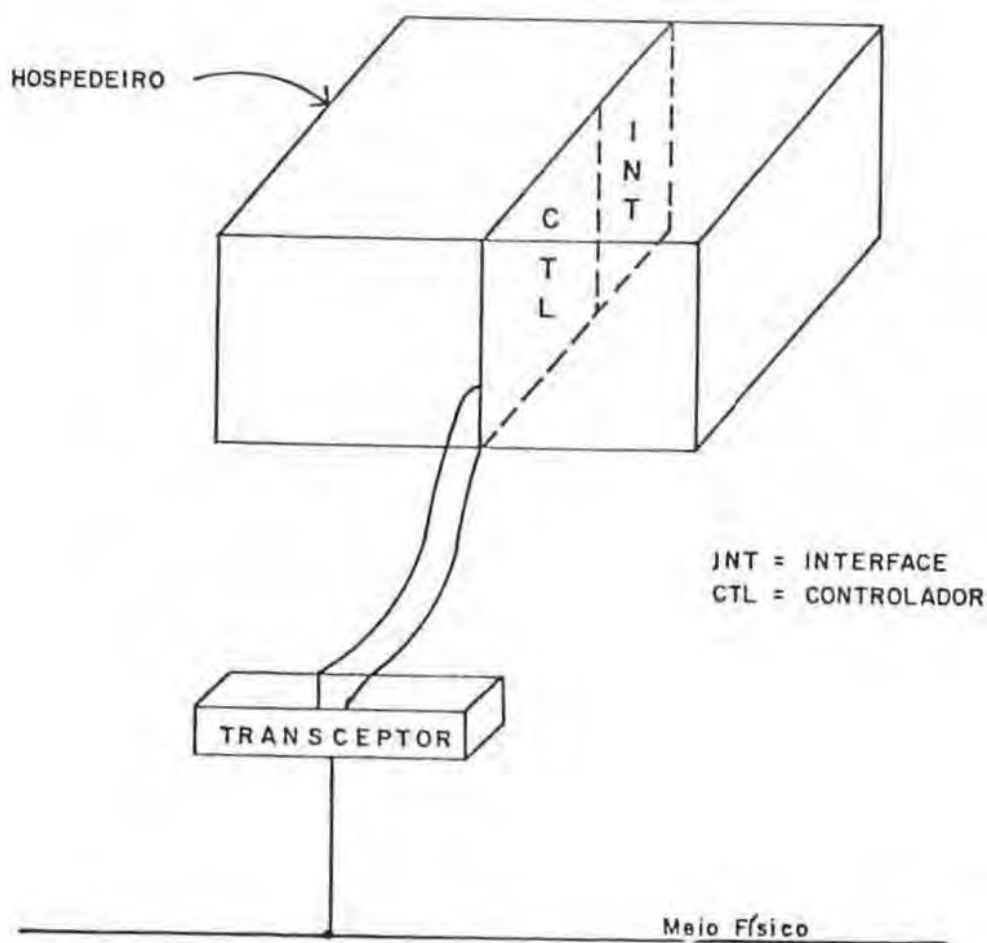


Figura 2 : Construção Física da Interface e do Controlador.

Aos controladores anteriormente citados são atribuídas principalmente funções do tipo.

a) Transmissão e recepção de quadros (nível de enlace), caracterizados por:

i) Permitirem diversas formas de endereçamento: único, em grupos ("multicast") e a todos (difusão ou "broadcast").

ii) Terem alguma capacidade de armazenamento temporário ("Buffers") tanto para recepção como para transmissão de dados, de forma a aliviarem o hospedeiro e o próprio meio físico em tráfego normal ou sob tráfego mais intenso.

b) Monitoração de ocorrência de erros.

Não há preocupação com recuperação de erros de transmissão/recepção até o nível 2, somente detecção, o que pode ser atribuído à projeção da rede Ethernet¹.

Admite-se que protocolos de níveis superiores no hospedeiro, aplicando políticas de reconhecimento ("acknowledgment") e outras técnicas de controle farão a recuperação de erros ocorridos com pedidos de retransmissão, etc.

3 - Especificação da Rede

Topologia

Nossa proposição é a de uma rede que utilize topologia do tipo de barra ("bus") pois acreditamos que a mesma possui uma confiabilidade intrínseca maior do que as demais aplicáveis a redes locais.

Método de Acesso

CSMA/CD - acreditamos que este método é o que permite a realização da gerência do contrôle de acesso, ao meio físico da rede, de forma mais distribuída.

Estrutura dos Nós

A figura 3 apresenta a estrutura do sistema HOSPEDEIRO, INTERFACE, CONTROLADOR e TRANSEPTOR que compõem um dos sistemas que estarão conectados à rede.

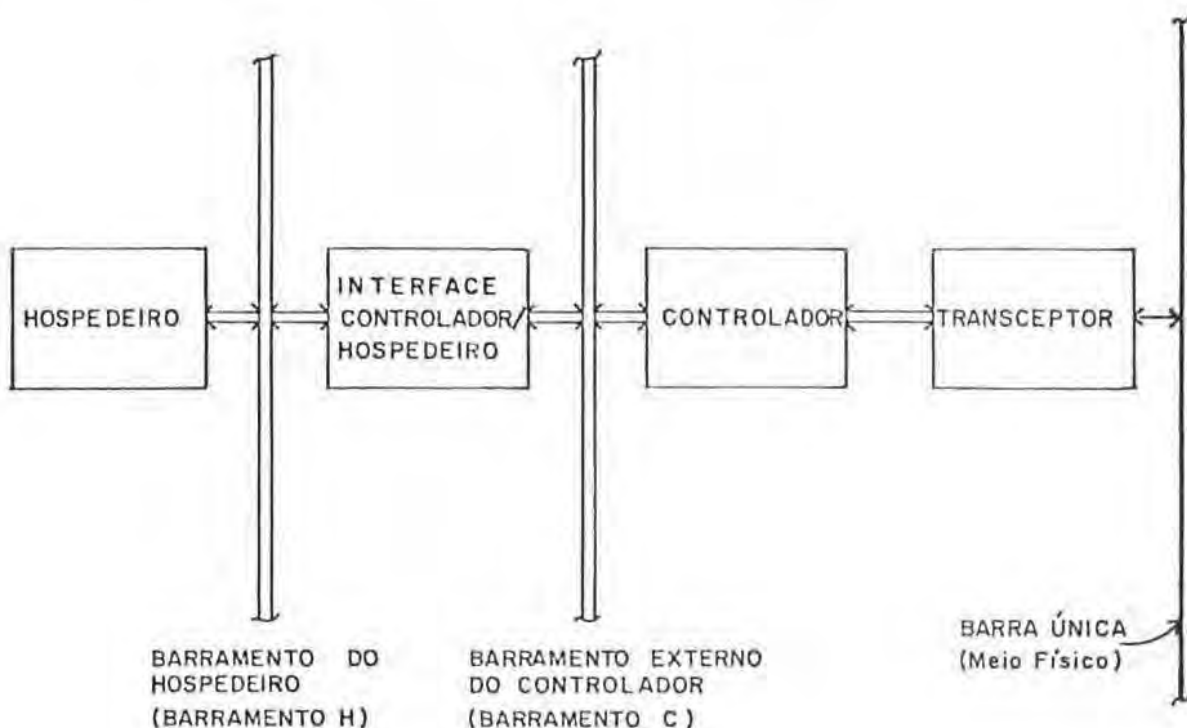


Figura 3 : A Estrutura da Rede Proposta.

Nos ateremos aqui a detalhar a estrutura e a operação do controlador.

a) Funções do Controlador

- i) Receber quadros colocados na barra única que se destinem a ele de acordo com as modalidades de endereçamento possíveis.
- ii) Informar ao Hospedeiro a recepção de quadros livres de erros.
- iii) Ter capacidade de armazenamento para recepção de até 2 quadros de tamanho máximo.
- iv) Transmitir quadros, a partir de dados enviados pelo Hospedeiro, fazendo a gerência necessária para evitar colisões e retransmiti-los caso estas ocorram.
- v) Ter capacidade para armazenar dados correspondendo a 2 quadros de tamanho máximo, para transmissão.
- vi) O CTL (Controlador) deve se apresentar ao hospedeiro como um dispositivo de E/S (I/O) que recebe comandos, fornece status, recebe e transmite dados de ou para o meio externo. A transferência de dados Hospedeiro/CTL poderá ser feita via instruções de E/S ou preferencialmente via DMA.

vii) O CTL deve se apresentar ao Hospedeiro como um canal de comunicação contendo dois subcanais (transmissão e recepção) independentes sob o ponto de vista lógico (*) embora interdependentes fisicamente em função da topologia BARRA da Rede.

viii) Gerência de filas na memória interna para grupos de dados correspondentes a quadros recebidos e/ou quadros a serem transmitidos ao meio externo.

(*) Isto permite que o controlador receba quadros endereçados a ele mesmo.

b) Detalhamento do Controlador

i) Diagrama em blocos

A estrutura interna do controlador foi concebida de forma que a mesma possuísse um conjunto de blocos tal, que permitisse, no caso de uma implementação, a utilização de pastilhas poderosas existentes no mercado, normalmente associados a uma determinada família de microprocessadores(*).

O diagrama em blocos do controlador proposto é apresentado a seguir (figura 4) onde procuramos simplificar as ligações, mantendo apenas os sinais que possuem uma maior importância para o entendimento lógico do mesmo.

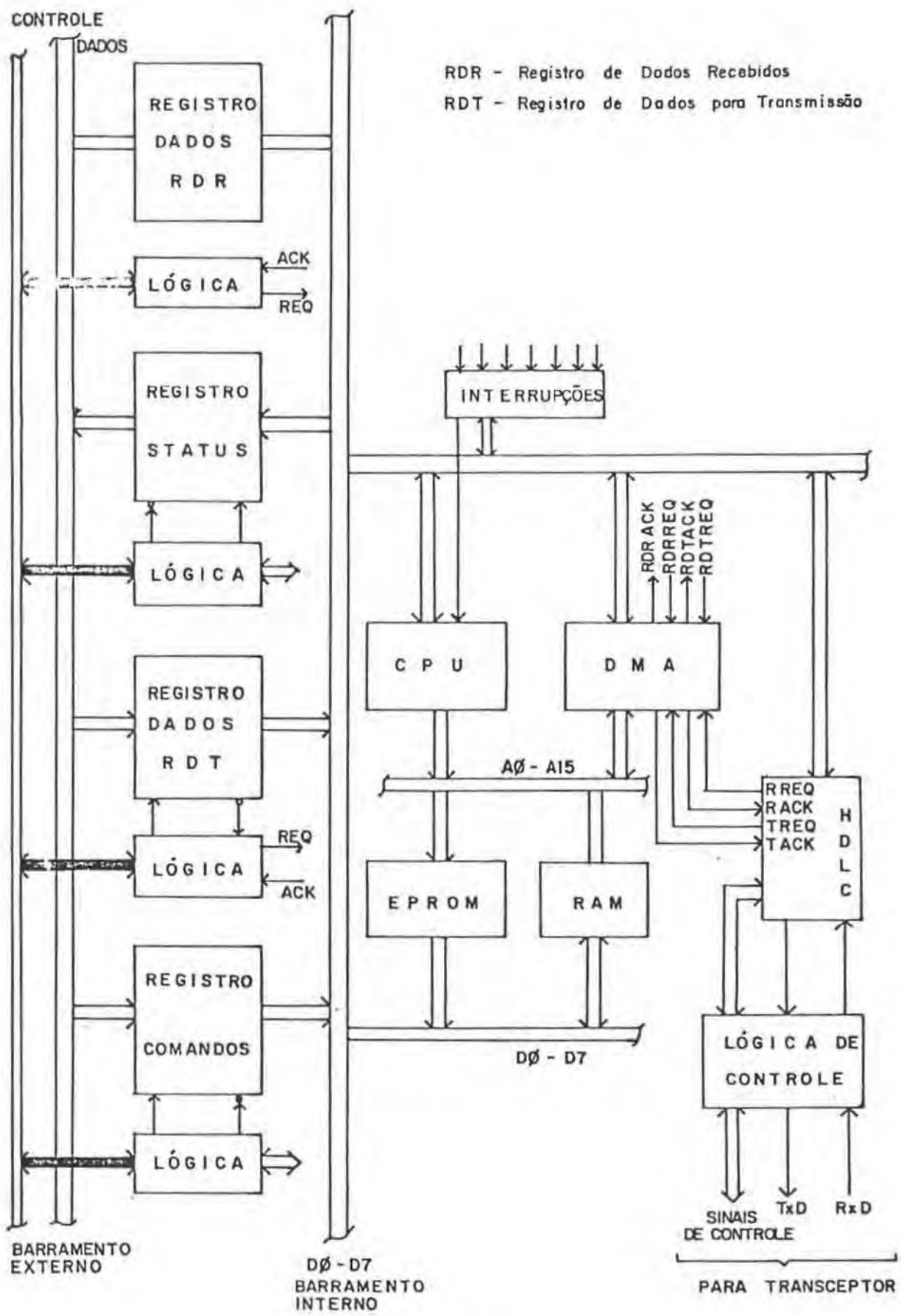


Figura 4 : Diagrama em Blocos do Controlador Proposto.

(*) Os fabricantes de microprocessadores procuram fornecer aos usuários pastilhas chamadas microperiféricos que suportem uma série de aplicações (comunicação série, comunicação paralela, DMA, etc).

ii) Descrição dos blocos principais

CPU - Gerencia a operação dos demais blocos do controlador. Executa um programa residente no módulo EPROM que permite:

- Inicializar os módulos DMA e HDLC.
- Interpretar e executar comandos enviados pelo Hospedeiro.
- Fornecer status ao Hospedeiro.
- Cumprir o protocolo de diálogo com o Hospedeiro para recebimento de comandos/dados e envio de dados/status.
- Preparar a operação do módulo DMA fornecendo ao mesmo os parâmetros necessários para que se realizem as transferências de dados via DMA (DADOS HOSPEDEIROS/MEM.CTL e MEM.CTL/HDLC).
- Preparar o módulo HDLC para envio/recepção de quadros, fornecimento de status e geração de sinais de controle.
- Gerencia da área de RAM, constituindo filas para recepção e transmissão de dados.

HDLC - Executa o protocolo HDLC.

- Endereço programável com 8 bits de comprimento.
- Recebe quadros de acordo com as diversas formas de endereçamento (único, "broadcast", etc).
- Transmite quadros no formato HDLC, inserindo seu endereço e todos os demais campos estabelecidos.
- Gera os sinais de controle necessários para o diálogo com a CPU e para a transferência de dados via DMA.
- Possui linhas de controle à disposição da CPU para uso em conjunto com o bloco LÓGICA DE CONTRÔLE.

DMA - Executa operações de DMA em 4 canais independentes.

Canal HDLC/MEM - transfere dados recebidos pelo módulo HDLC, vindos do meio externo, para a memória RAM.

Canal MEM/HDLC - Transfere dados da memória para o módulo HDLC que os transmitirá serialmente ao meio externo.

Canal RDT/MEM - Transfere dados enviados pelo hospedeiro registra

dos no RDT para a memó-
ria interna do CTL.

Canal MEM/RDR - Transfere dados da memó-
ria interna para o regis-
tro RDR que será poste-
riormente acessado pelo
hospedeiro.

Todas as transferências são coordenadas pelo CPU
que fornece ao módulo DMA os endereços de memória onde ini-
ciam os "buffers" de dados, o número de bytes a serem trans-
feridos, etc.

LÓGICA DE CONTROLE - Auxilia no contróle de acesso
à barra única.
- Detecta presença de sinais de
transmissores na barra.
- Acusa a ocorrência de colisões.
- Auxilia a CPU no atendimento
às interrupções possíveis den-
tro do controlador.

INTERRUPÇÕES

x Recebimento de comando do Hos-
pedeiro
x Leitura de status pelo Hospe-
deiro
x Término de transmissão para o
meio externo

- x Ocorrência de colisões
- x Término de transferência de dados nos canais de DMA
- x Relógio ("TIME-OUTS").

PROTOSCOLOS

a) N \bar{o} /N \bar{o} - O protocolo de conversação N \bar{o} /N \bar{o} é basicamente uma parte do HDLC (*) em conjunto com o mecanismo de controle de acesso CSMA/CD.

Isto significa que durante a transmissão de quadros em que não hajam colisões prevalece essencialmente o estabelecido no protocolo HDLC.

Adicionalmente, o mecanismo CSMA/CD estabelece que o início de uma transmissão seja vinculado a não existência de outro transmissor na barra (CD-"carrier detect").

Em caso de ocorrência de colisão, a mesma deve ser "reforçada" pelos nós que primeiro a perceberam, de forma que esta seja detectada também por todos os outros que porventura tenham participado desta transmissão simultânea.

(*) Referente ao formato dos quadros, à verificação de endereço por parte do receptor e verificação de erros.

Novas tentativas de transmissão após a ocorrência de colisões devem aguardar um período de tempo gerado de forma pseudo-aleatória por parte de cada nó envolvido na colisão.

ão. Isto diminuirá a probabilidade de ocorrência de novas colisões.

Protocolo Hospedeiro/Nô

Hospedeiro (Host) deseja transmitir comando para controlador (CTL).

- HOST escreve comando no registro de comandos do controlador, gerando uma interrupção no CTL.
- O recebimento do comando por parte do CTL é percebido (INTERRUPÇÃO) pelo HOST.
- CTL fornece status referente à aceitação do comando, interrompendo o HOST.

Em caso de desejo de transmissão de dados para o meio externo por parte do HOST, este escreve no registro de dados RDT, em dois ciclos consecutivos, o nº de BYTES que vão ser transferidos.

- CTL informa ao HOST a possibilidade ou não de receber estes dados (análise da área de RAM disponível).
- Em caso positivo, HOST transfere dados para memória interna do controlador (via RDT).
- CTL informa ao HOST o sucesso da transferência de dados para o meio externo referente ao primeiro quadro da fila de espera (caso vários quadros estejam no controlador aguardando transmissão para o meio externo).

- O formato dos dados passados do HOST ao CTL é mostrado a seguir (figura 5).

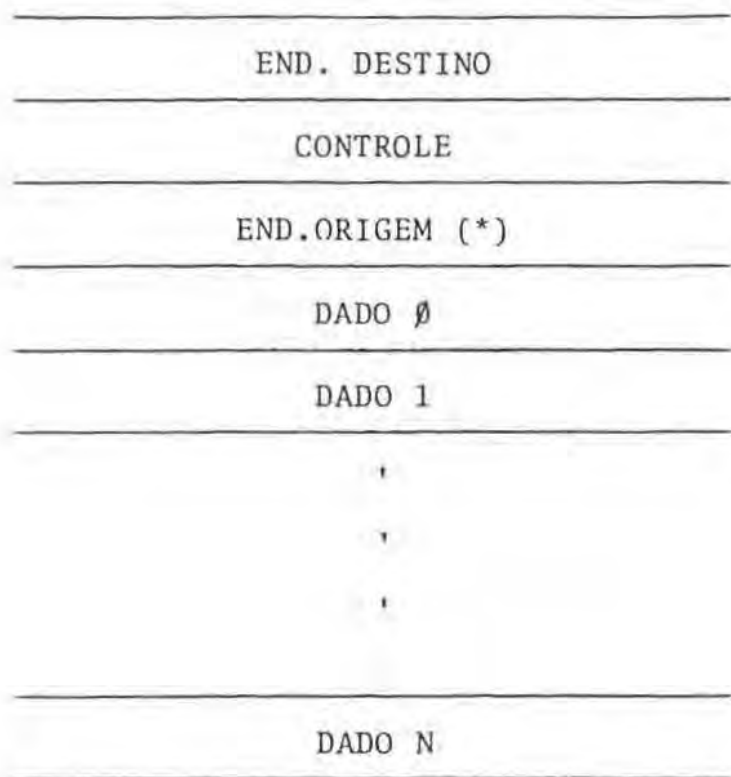


Figura 5

(*) Inserido pelo CTL que possui um único endereço que o identifica.

RECEPÇÃO DE DADOS

- HOST, através de um comando, habilitou o CTL à recepção de dados do meio externo.
- CTL captura quadros a ele endereçado e após verificar a origem e o comprimento, informa ao HOST.
- HOST informa ao CTL o desejo de receber os dados aguardando nos 2 próximos ciclos de escrita no registro de dados (RDR) o nº de BYTES deste quadro.

- HOST informa o desejo ou não de continuar a operação.
- Os dados passados do CTL ao HOST terão o formato apresentado na figura 6.

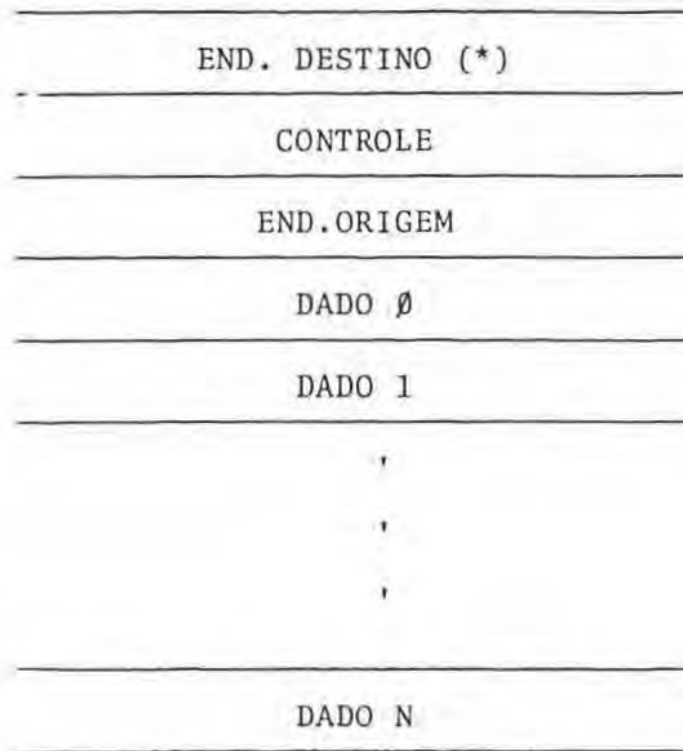


Figura 6

4 - Implementação e Testes Realizados

Simplificando-se a estrutura de "hardware" proposta foram montados os circuitos necessários à realização da interconexão de dois micros shumec da Seção de Informática do IME. Os circuitos foram montados em duas placas s-100 com ligações em "wire-wrap".

As simplificações feitas em relação ao proposto

consistiram principalmente de:

- a) Um único canal de DMA (uma pastilha Z80-DMA) ao invés dos 4 propostos. Este canal é então partilhado entre a recepção e a transmissão serial de dados. A transferência de dados entre os registros de dados a transmitir e recebidos e a memória é feita sob a supervisão da CPU, fazendo uso de interrupções.
- b) Não foi implementado um mecanismo de relógio para geração de "time-outs".

Estrutura do controlador

A figura 7 apresenta um diagrama simplificado do controlador (CTL) implementado.

O controlador utiliza um microprocessador Z80 e alguns microperiféricos da família Z80 para suporte das operações de transmissão/recepção síncrona (SIO), DMA e interfacçamento paralelo (PIO).

A descrição de cada bloco assim como a operação do controlador se aproxima do que foi apresentado aqui. Devemos ressaltar o partilhamento de um único canal de DMA entre transmissão e recepção serial de dados e o fato de que o Z80 e seus microperiféricos trabalham com sistema de interrupção vetorizado. As prioridades das interrupções são determinadas intrínsecamente em função de interligação dos microperiféricos ("Daisy-chain"). Isto simplifica o "hardware"

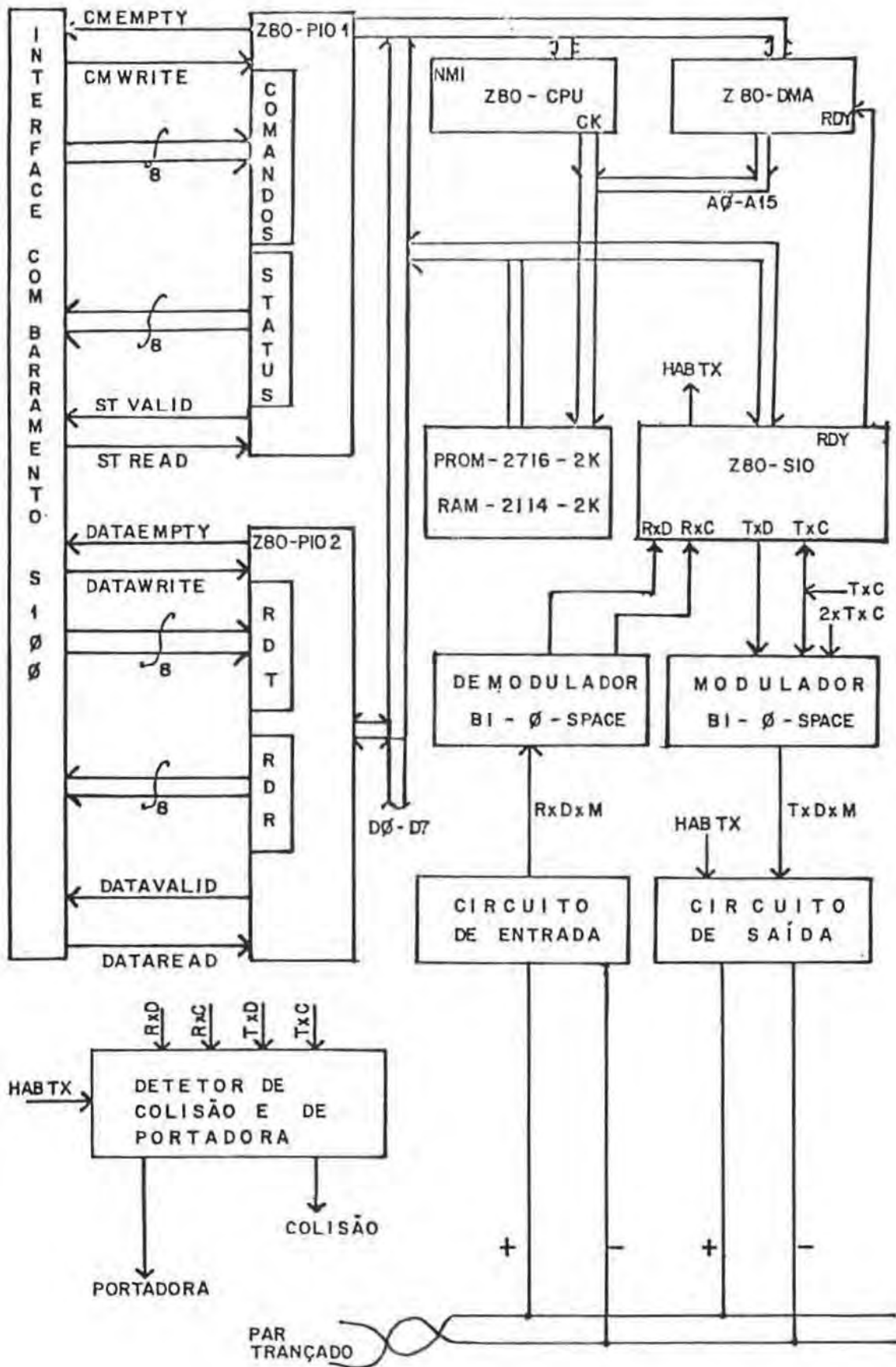


Figura 7 : Diagrama em Blocos do Controlador Implementado.

pois dispensa o uso de um controlador de interrupções.

Interface com hospedeiro

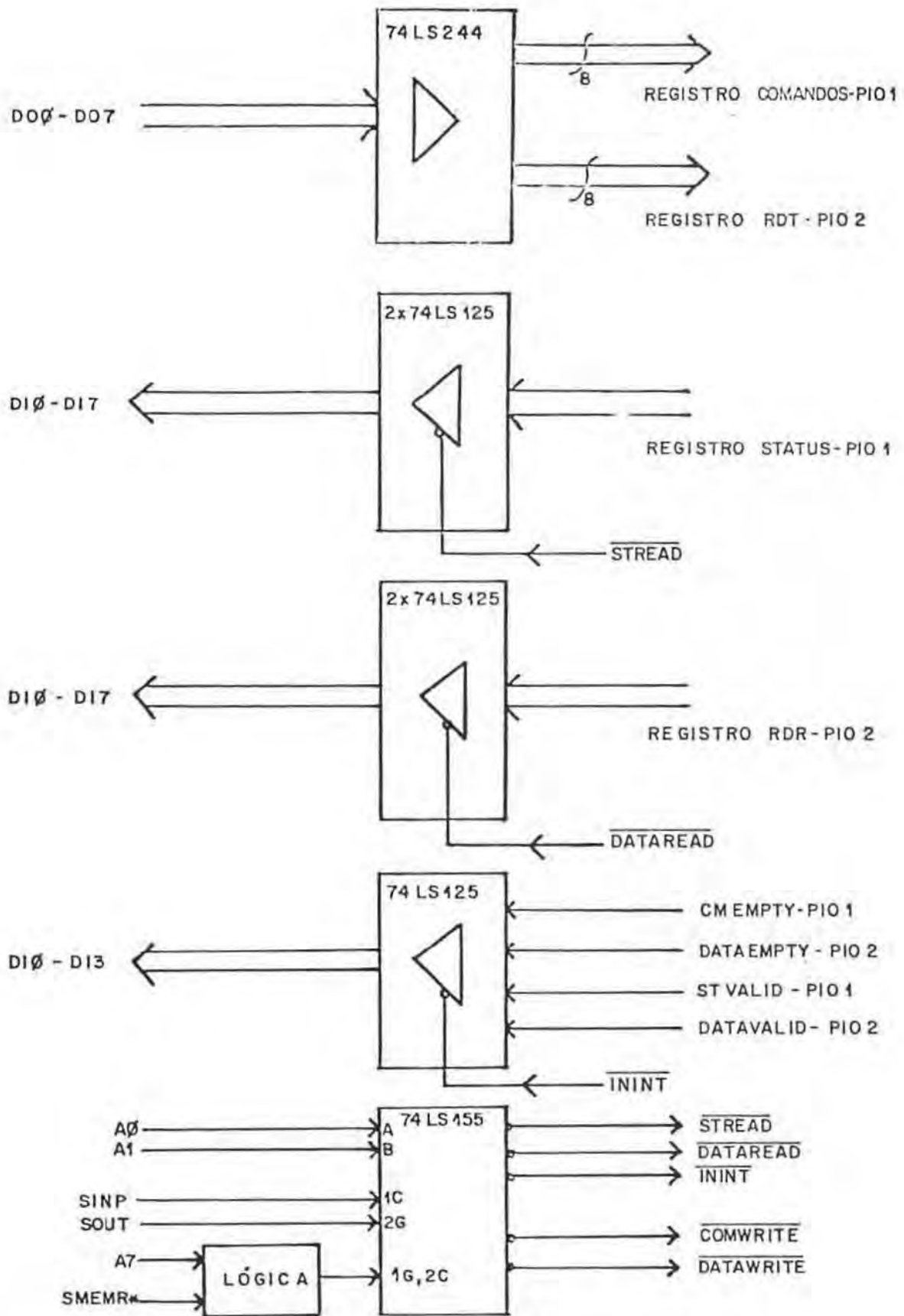
Os micros shumec possuem um barramento tipo S-100 (IEEE 696).

A interface implementada é mostrada na figura 8.

a) Descrição e Operação

Em função da interface, o controlador passa a ser visto pelo "host" (shumec) como um dispositivo de E/S contendo 3 canais de entrada e 2 canais de saída como descritos a seguir.

- Canal COMWRITE (8 bits) - destinado a escrita de comandos no controlador, é acionado através de uma instrução do tipo "OUT 00".
- Canal DATAWRITE(8 bits) - destinado a escrita de dados no controlador, é acionado através de uma instrução do tipo "OUT 0 1".
- Canal STREAD (8 bits) - através de uma instrução do tipo "IN 00" o host pode ler o registro do controlador que contém as informações de status do mesmo.
- Canal DATAREAD (8 bits) - através de uma instrução do tipo "IN 0 1" o host acessa o registro de dados do controlador.
- Canal ININT (4 bits) - através deste canal o



BARRAMENTO S100

Figura 8 : Diagrama em Blocos da Interface Implementada.

host toma conhecimento do status dos demais canais no tocante a presença de informações ainda não lidas nos canais de dados e/ou status por parte do host ou de informações referentes à ocorrência de leitura de um comando e/ou de um dado por parte do controlador. O significado dos bits constantes neste canal é mostrado a seguir. Este registrador ININT é lido através da instrução "IN 02".

REGISTRO ININT							
D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	DATA VALID	ST. VALID	DATA EMPTY	CM EMPTY

- CM EMPTY - Quando igual a "1", indica que o último comando escrito pelo host, já foi lido pelo controlador.
- DATA EMPTY - Quando igual a "1" indica que o último dado escrito pelo host, já foi lido pelo controlador.
- STVALID - Este bit, quando igual a "1" informa ao host que o último Status fornecido pelo controlador é válido ou seja ainda não foi lido.

DATAVALID - Este bit, quando igual a "1" informa ao host que o último dado fornecido pelo controlador ainda não foi lido.

Conceitualmente o canal ININT deveria, ao nível do host, estar veinculado a algum mecanismo de interrupção, o que possibilitaria que o diálogo host-CTL pudesse ser feito de forma assíncrona. Achamos, entretanto, que este mecanismo poderia ser implementado posteriormente, admitindo que nesta primeira fase este diálogo fosse feito através de um "polling" do registro ININT.

Transceptor

Não houve, em nossa implementação a preocupação de distinguirmos fisicamente os circuitos que em outras redes compõem os transceptores. O que fizemos foi incorporar estes circuitos à placa destinada ao controlador.

O processo de modulação que utilizamos para embutir o clock no sinal a ser transmitido denomina-se modulação "Bi- \emptyset -space".

Este processo consiste em criar transições de níveis no sinal a ser transmitido numa taxa correspondente à taxa de transmissão. Adicionalmente, criam-se transições num ponto correspondente no tempo à metade de um bit quando o mesmo for igual a " \emptyset ".

Na recepção deveremos distinguir as transições que correspondem a taxa de transmissão das inseridas no meio de cada bit "0". O diagrama de tempo destas duas operações é mostrado na figura 9 .

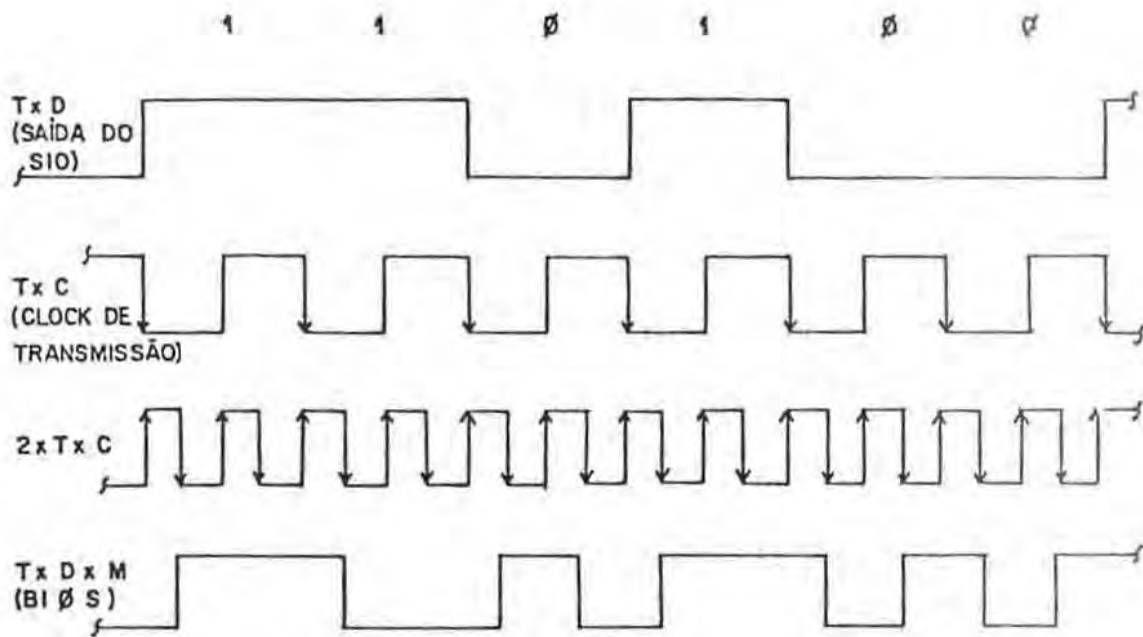
Detecção de transmissor presente e detecção de colisão.

A parte do protocolo CSMA/CD referente à detecção de portadora corresponde na verdade a detecção de um transmissor presente na barra. A realização deste mecanismo consistiu num circuito multivibrador monoestável com "re-trigger" ajustado para uma frequência tal que sempre que algum transmissor estivesse presente sua saída fosse igual a "1". Este circuito faz uso dos pulsos gerados a cada transição necessários ao circuito de recepção síncrona utilizado.

Sua saída é então conectada à entrada CTSB do circuito SIO de forma que a CPU lendo um dos registros de leitura do SIO (RR0) e testando a condição do bit correspondente à entrada CTSB tem meios de monitorar a situação da barra antes de habilitar uma transmissão.

O circuito utilizado para detecção de colisão tem como princípio de funcionamento, comparar os dados que estão sendo transmitidos com os que estariam sendo recebidos. A ocorrência de uma colisão determinará fatalmente uma degeneração nos dados transmitidos de forma que uma comparação simultânea entre o que está sendo recebido (vindo da barra) com

TRANSMISSÃO



RECEPÇÃO

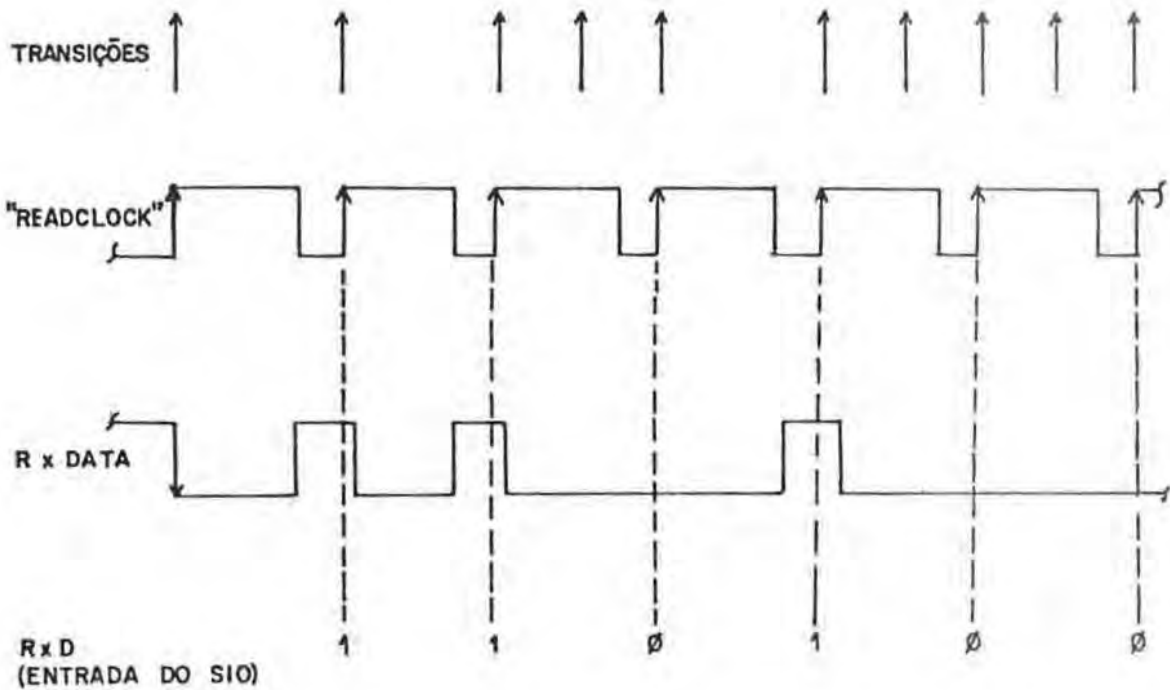


Figura 9 : Diagrama de Tempo do Processo de Modulação e De modulação.

o que está sendo transmitido (antes da barra) resultará numa desigualdade, indicativa de colisão. O resultado desta colisão é passado à CPU através de uma interrupção não mascarável para que devidas providências sejam tomadas.

Circuitos de entrada e saída e meio de transmissão

Decidimos utilizar como meio de transmissão um par trançado, sem blindagem, que atende às necessidades de taxa de transmissão desejada (400 kbps) e às possíveis distâncias que a rede venha a percorrer dentro do IME⁸.

Os circuitos de entrada ("line receivers") e saída ("line drivers") devem então possuir entrada diferencial e saída balanceada respectivamente. Estas características dão ao conjunto "driver/receiver" par trançado uma boa imunidade a ruído. Em nossa implementação optamos pelas pastilhas 75174 ("driver") e 75175 ("receiver").

Testes realizados

Os itens apresentados a seguir mostram a evolução dos testes, já realizados, em nossa implementação.

- i) Cada placa foi testada no sentido de verificar o bom funcionamento dos circuitos de CPU, EPROM, RAM, SIO, PIO e DMA.
- ii) Individualmente, foi testada cada interface com o hospedeiro. Foram feitas trocas de dados entre hospedeiro e controlador através dos canais COMWRITE, DATAWRITE, STREAD e DATAREAD.

iii) Cada conjunto hospedeiro/controlador foi testado no tocante à comunicação serial com o SIO programado na modalidade SDLC. Isto é, foi feita uma conexão tipo "LOOP" onde verificou-se se os dados transmitidos, após passarem pelos circuitos de modulação e demodulação, eram recebidos corretamente.

iv) Foram realizados testes de comunicação entre os dois sistemas "Hospedeiro/Controlador". Isto é, dados entrados pelo terminal de um hospedeiro eram enviados ao outro "Hospedeiro/Controlador" com posterior apresentação no terminal deste último.

O programa final do controlador deverá ser capaz de aceitar comandos dentro do protocolo estabelecido, enviar status, fazer transmissão e recepção de quadros nas modalidades única e por difusão, etc.

Conclusão:

Os objetivos estão sendo alcançados, pois já conseguiu-se transferência de dados de um terminal ao outro (nos dois sentidos).

As continuações:

a) Criação de protocolos de níveis superiores(3a7) até uma aplicação de automação de escritório.

- b) Desenvolvimento de outros controladores para os outros micros Shumeck existente no IME.
- c) Desenvolvimento de controladores específicos para suportarem o partilhamento de discos e impressoras.

BIBLIOGRAFIA (Rev. 11/80)

- 1 - Digital, Intel, Xerox, the Ethernet, A Local Area Network, Data link layer and Physical layer Specification, Version 1.0, set. 30, 1980.
- 2 - 3Com, 3C400 multibus Ethernet Controller, 3Com Corporation
- 3 - Interlan, NI2010 bus Ethernet Communications Controller, Interlan.
- 4 - MFE, Recording Techniques, MFE, Application Note NO.64B, out 1975.
- 5 - Kotelly, G., Local Area Networks, EDN, 109-150, fev. 1982.
- 6 - Zilog, Z80CPU, Z80DMA, Z80 SIO, Z80 PIO, Z80 CTC, Zilog, Product Specification Catalog, 1981.
- 7 - Osborne, McGraw-Hill, Interfacing to S.100/IEEE696 Micro computers, Osborne/McGraw-Hill, 1981.

A REDE LOCAL DE COMPUTADORES DO NCE/UFRJ

José Fábio Marinho de Araújo
Guilherme Chagas Rodrigues
Diogo Fujio Takano
Moacyr Henrique Cruz de Azevedo

Resumo:

Esse trabalho descreve a rede local em desenvolvimento no Núcleo de Computação Eletrônica da UFRJ. Essa rede é em anel e tem como finalidade conectar diversos computadores existentes no campus, de forma rápida e confiável. Uma primeira versão da rede já está em operação experimental.

Abstract:

This paper describes the local area computer network being developed at the Federal University of Rio de Janeiro. It is a ring network and its objective is to interconnect several computers existing in the campus providing a mean of communication fast and reliable between these machines. A first version of the network is already in experimental operation.

1 - INTRODUÇÃO:

Os avanços na tecnologia de microeletrônica, que levaram ao surgimento de mini e microcomputadores de baixo custo, estão causando profunda alteração na atividade de processamento de dados. Estimativas recentes {1} mostram que os microcomputadores que representavam 7.5% do mercado de computadores em 1975, em 1980 já representavam 23% e a previsão é de que em 1985 terão ocupado 43% do mercado.

A tendência é portanto, no sentido do usuário dispor de capacidade de processamento local para muitas de suas aplicações. Um outro aspecto dessa mudança é que o processamento da informação deixa de ser um recurso escasso nas organizações. Com isso toda uma gama de novas aplicações que não se justificavam do ponto de vista econômico, passam a ser viáveis.

Esse fenômeno já está ocorrendo na Universidade Federal do Rio de Janeiro - UFRJ. Os dois grandes grupos de usuários do Núcleo de Computação Eletrônica - NCE: a administração da Universidade e a comunidade acadêmica, estão começando a dispor de seus próprios minis e microcomputadores para aplicações específicas, alguns instalados pelo próprio NCE e outros adquiridos pelos departamentos.

Se de um lado a computação assim descentralizada tem uma série de atrativos, em contra-partida um mini e um microcomputador isoladamente nem sempre atende todas as necessidades dos usuários. No caso dos departamentos de ensino e pesquisa, por haver algumas vezes necessidade do poder computacional de uma máquina de grande porte, e no caso do processamento administrativo pela necessidade de se ter acesso a informações de sistemas diferentes.

Nesse contexto, uma rede local conectando essas máquinas permitirá que elas compartilhem recursos, sejam eles informações, poder de computação de uma máquina de maior porte ou um periférico mais dispendioso como uma unidade traçadora de gráficos, por exemplo.

Com essa motivação, o NCE iniciou um projeto de desenvolvimento de uma rede local, no âmbito do projeto REDLAC - Rede Latino Americana de Computadores, patrocinado pela O.E.A. Este projeto congrega quatro estabelecimentos de ensino superior: dois brasileiros e dois mexicanos. A finalidade do projeto REDLAC é o de se desenvolver em cada uma das entidades participantes, um grupo com conhecimentos e tecnologia próprias na área de redes de computadores, bem como trocarem entre si esta experiência adquirida. Dentro do projeto REDLAC fo

ram desenvolvidas quatro redes locais, cada uma com tecnologia diferente, visando particularidades próprias de cada instituição. Este artigo relata a experiência obtida pelo grupo da UFRJ no desenvolvimento de sua rede local.

2 - SITUAÇÃO DE COMPUTAÇÃO DA UFRJ

O Núcleo de Computação Eletrônica - NCE é o órgão responsável pelo processamento de dados na UFRJ. Suas responsabilidades incluem:

- a) Desenvolver sistemas para a administração da Universidade e para o Hospital Universitário;
- b) Processar estes sistemas regularmente;
- c) Colocar seus computadores à disposição da comunidade acadêmica para ensino e pesquisa.

Dentro da atividade de desenvolver sistemas administrativos para a UFRJ, vários sistemas foram desenvolvidos usando microcomputadores projetados e construídos no próprio NCE. Este é o caso de um sistema de controle de medicamentos para o Hospital Universitário e um sistema de acompanhamento orçamentário ligado à reitoria.

Quanto à área acadêmica, vários departamentos possuem mini e micro computadores para suas pesquisas próprias, já havendo cerca de duas dezenas deles espalhados pelo campus da UFRJ.

A motivação para se construir uma rede local na UFRJ surgiu da necessidade de interligar estes computadores dispersos no campus de modo a que seus usuários possam ter um melhor aproveitamento de seus equipamentos. A ligação destes pequenos computadores a um computador de grande porte, daria uma nova dimensão à utilização dos mesmos.

Como suporte a essas atividades o NCE conta com um computador de grande porte, Burroughs 6700, um computador de médio porte Digital PDP-11/70 e diversos mini e microcomputadores.

O NCE dispõe também de uma Divisão de Pesquisa e Desenvolvimento que tem como uma de suas diretrizes o desenvolvimento de projetos que sejam utilizados na Universidade e passíveis de virem a ser transferidos para a indústria

brasileira. Dentro dessa linha, diversos produtos já foram desenvolvidos e es
tão sendo industrializados (2).

Outra motivação para o desenvolvimento da rede local foi a de criar
possibilidade de se desenvolver sistemas administrativos distribuídos, usando
microcomputadores brasileiros, o que seria mais barato, eficiente e liberaria
as máquinas de grande porte importadas para o uso acadêmico.

3 - ARQUITETURA DE REDES LOCAIS

Ao contrário das redes de longa distância, desenvolvidas utilizan
do-se a infra-estrutura de telefonia já existente e sujeitas portanto às limita
ções de velocidade e taxa de erro dessas, as redes locais conectam um conjunto
de equipamentos em uma área geográfica limitada, em geral pertencentes a uma
mesma organização, o que permite projetar uma rede com características específi
cas para transmissão de dados e adaptada às necessidades da organização (3, 4).

As características de redes locais em geral incluem:

- * alta velocidade de transmissão; a faixa usual é 200 Kbits/s a 10 Mbits/s (o
que significa de 25.000 caracteres por segundo a 1.200.000 caracteres por se
gundo);
- * taxa de erros muito baixa, da ordem de 1 bit em cada 10^{12} transmitidos (cada
caracter é representado por 8 bits);
- * possibilidade de difusão de mensagens (broadcast) para todas as máquinas co
nectadas à rede.

Equipamentos em uma rede de computadores, podem ser conectados de di
versas formas (topologias). As formas mais utilizadas em redes locais são "bar
ra simples" e "anel".

Nas redes em barra simples, o meio de transmissão é aberto e apenas
uma estação (computador) pode estar transmitindo num determinado instante do
tempo; as outras verificam se a mensagem lhes é destinada e caso seja, a copi
am.

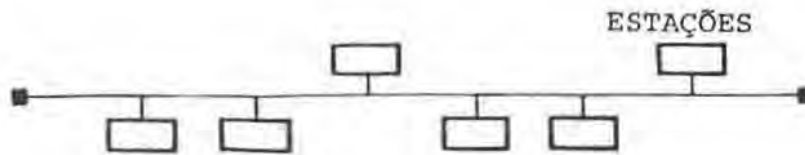


FIGURA 1: Rede em Barra Simples

Nas redes em anel, o tráfego de mensagens flui em um sentido no anel. As estações examinam a mensagem enquanto ela flui e a copiam caso lhe seja destinada.

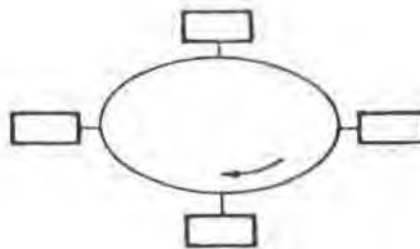
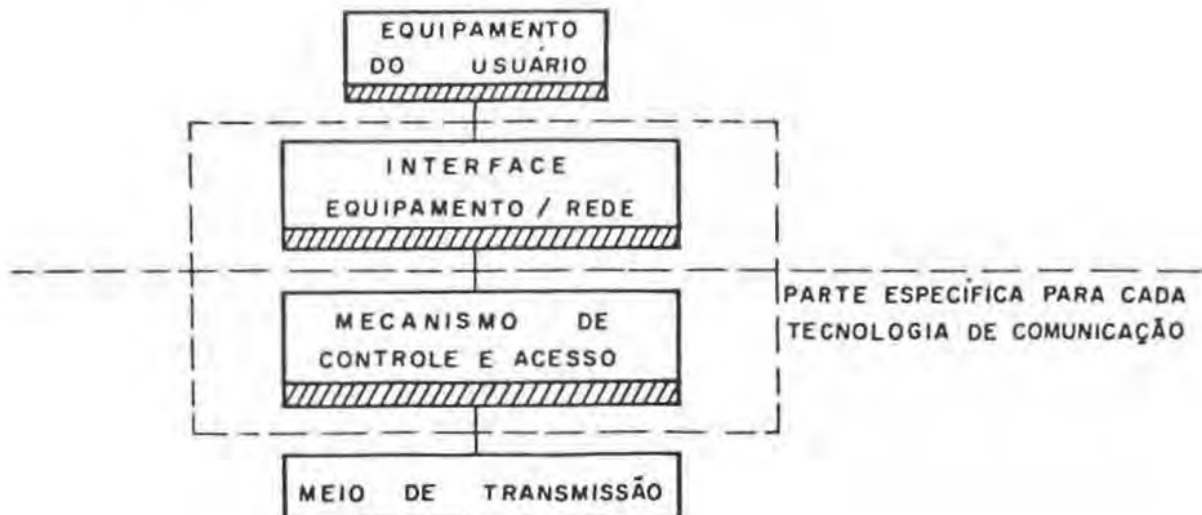


FIGURA 2: Rede em Anel

COMPONENTES DE UMA REDE LOCAL

Os elementos que compõem uma rede local são:

- * meio de transmissão;
- * mecanismo de controle e acesso ao meio;
- * interface equipamento usuário/rede;
- * protocolos.



Os meios de transmissão mais utilizados são fio comum trançado e cabo coaxial. Fibra ótica e micro ondas estão sendo utilizados por enquanto apenas em redes experimentais.

O mecanismo de controle e acesso ao meio, tem como finalidade resolver o problema de qual estação pode utilizar o meio de transmissão em um certo instante do tempo e de que forma.

As técnicas empregadas são dependentes da topologia escolhida e implementadas por hardware (eletrônica) ou hardware e software.

A interface equipamento usuário/rede consiste no hardware e software de comunicação entre o computador usuário e a rede.

O conjunto de regras de procedimento, formato de mensagens e comandos para troca de informações, recuperação de erros, etc., entre as máquinas é chamado de protocolo. Essas funções devem ser definidas e implementadas em cada computador conectado à rede.

4 - DESCRIÇÃO GERAL DA REDE LOCAL NCE/UFRJ

A topologia escolhida para implementar a rede local NCE/UFRJ foi a de anel. Mais especificamente, decidimos implementar uma rede em anel baseada no projeto desenvolvido na Universidade de Cambridge(5,6). Os motivos que nos levaram a tal escolha foram mais políticos do que técnicos, pois não havia na época do início do projeto, experiência em larga escala com nenhuma rede local que mostrasse a superioridade de uma topologia em detrimento de outra. O fator que mais pesou na escolha foi o de termos no NCE intercâmbio com a Universidade de Kent, que já estava implementando este tipo de rede e portanto tivemos acesso a esta experiência.

As características técnicas principais da rede são:

- 1) Velocidade do anel de 10 Megabits/segundo que é comparável à velocidade de leitura de um disco magnético;
- 2) O meio de transmissão é composto por dois pares de fios trançados, usados em comunicação telefônica;
- 3) A distância máxima entre as estações é de 100 metros;

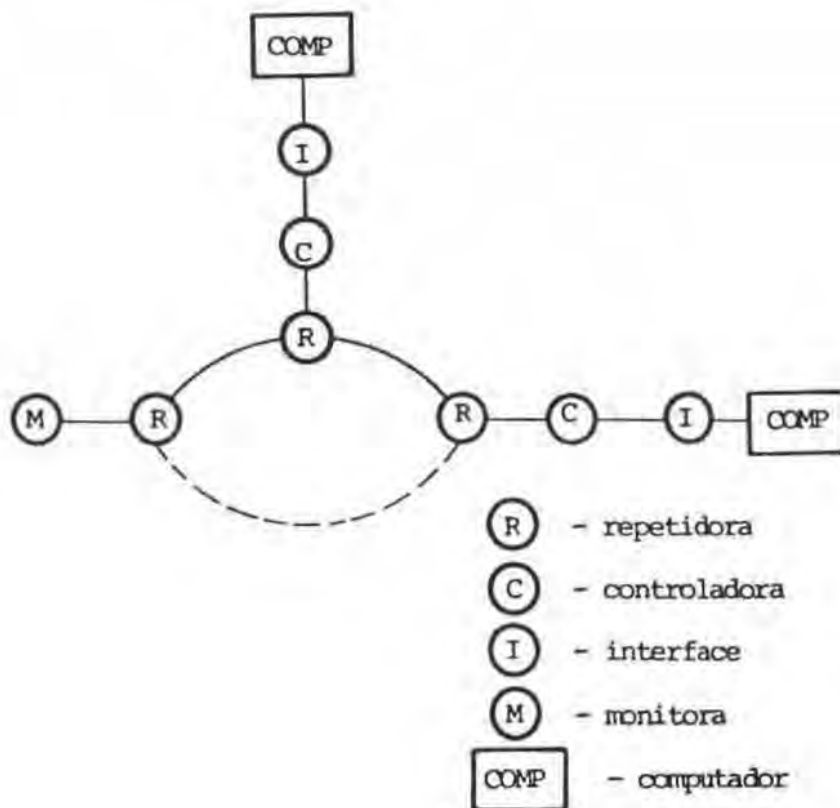
4) O mecanismo de acesso ao meio de transmissão é chamado de "empty slot". Consiste em fazer circular pelo anel, grupos de informações chamados pacotes que tem tamanho fixo, no caso de 38 bits. Quando uma estação deseja transmitir dados a outra, espera passar por ela um pacote vazio e coloca no mesmo a informação a ser transmitida. Mais adiante é descrito em detalhe o mecanismo de acesso ao meio de transmissão;

5) Há uma estação especial chamada monitora cuja função principal é inicializar a rede colocando pacote vazio no anel e controlar alguns tipos de erros como veremos em detalhes mais adiante.

Componentes da Estação

Uma estação é o dispositivo que permite a ligação de um computador à rede. Em uma rede em anel, a integridade de toda a rede é dependente do funcionamento de cada estação, pois basta uma estação parar para interromper o funcionamento da rede.

Tendo em vista este problema de segurança, a estação da rede local NCE/UFRJ está dividida em três partes: repetidora, controladora e interface de acesso.



Repetidora

A repetidora é responsável pela recuperação do sinal. Sua função é a de retransmitir para a próxima estação o que recebeu da estação anterior. Evidentemente, a informação pode ser modificada por controle explícito da controladora, como por exemplo para inserir dados em um pacote vazio. A repetidora funciona com alimentação elétrica e sinais de controle vindos do próprio meio de transmissão, para tornar seu funcionamento independente de qualquer componente local. O projeto desta parte foi realizado tendo em vista a maior confiabilidade possível, de modo que sinais da controladora que entram na repetidora são protegidos por relés. Em caso de falha ou falta de luz local, estes relés são abertos e a repetidora funciona sozinha mantendo a continuidade da rede.

Controladora

A controladora é a parte que controla toda a transmissão e recepção de dados. Recebe mensagens do computador através da interface e coloca estas na rede e vice-versa.

Interface

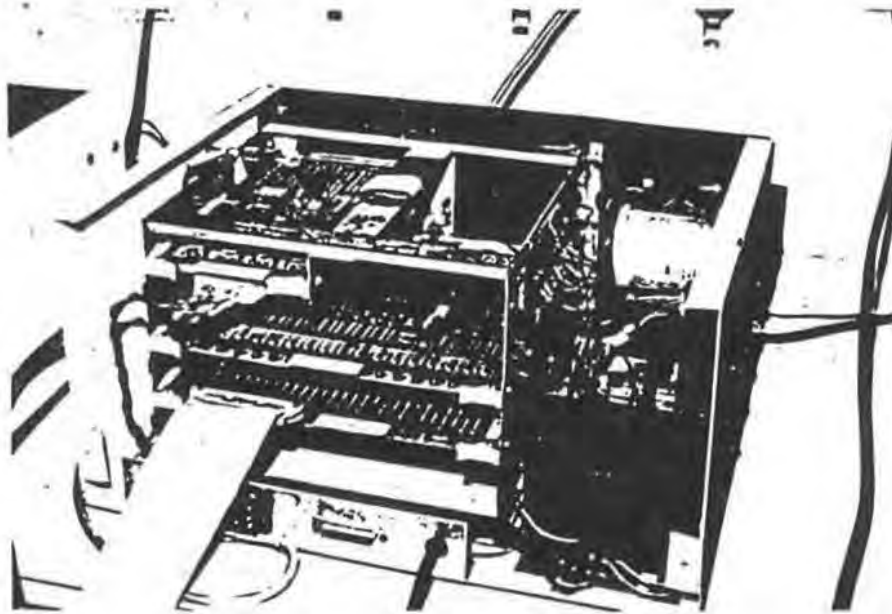
A interface é responsável em compatibilizar o mecanismo de entrada e saída do computador específico com a controladora. Esta interface se faz necessária devido a grande variedade de características dos computadores no mercado. Se existisse um só tipo de mecanismo de entrada e saída, a controladora seria projetada para trabalhar com este mecanismo e a interface não existiria, porém este não é o caso que ocorre na prática. A interface é então específica para cada computador que quisermos ligar à rede.

Monitora

Resta mencionar a monitora que é uma estação especial responsável pela inicialização da rede, controle de erros e que não está ligada a nenhum computador. A estação monitora está ligada também a uma repetidora e é responsável pela alimentação elétrica de todas as repetidoras. Vale à pena ressaltar que a monitora é uma peça fundamental da rede e sua falha causa a falha de toda a rede. A existência da monitora faz com que esta rede em anel não seja uma rede totalmente distribuída pois existe uma estação especial sem a qual a rede não

funciona.

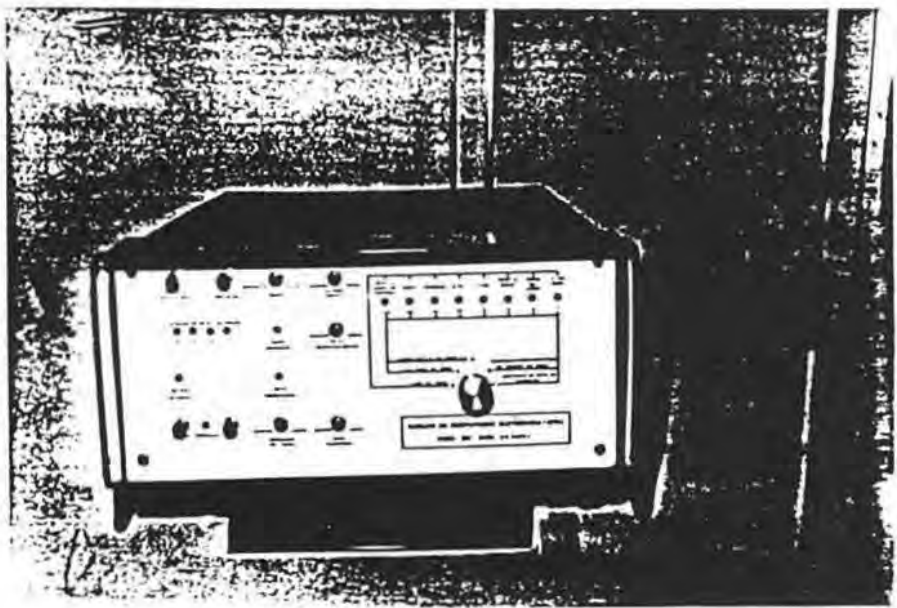
Na primeira fase do projeto estamos conectando os computadores B6700, PDP-11/70 e microcomputadores SDE-40 projetados no NCE.



Estação com as placas da repetidora, controladora e interface (de cima para baixo no rack).



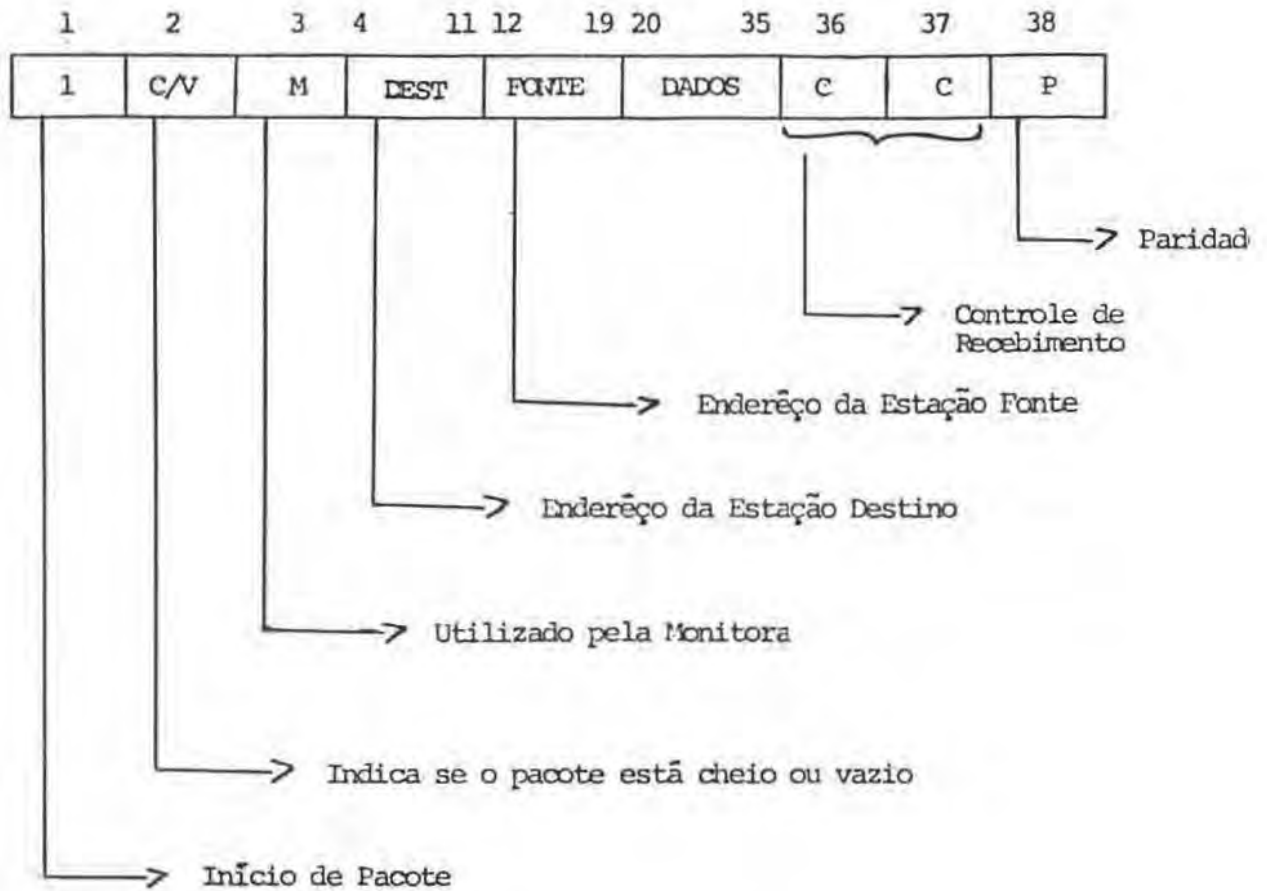
Microcomputador SDE-40 com sua estação (caixa sobre a mesa).



Estação Monitora.

5 - MECANISMO DE ACESSO

Os pacotes que circulam no anel tem o seguinte formato:



Quando uma estação precisa transmitir uma mensagem a outra, realiza o seguinte procedimento:

- 1) Parte a mensagem em pedaços de dois caracteres;
- 2) Espera a passagem de um pacote vazio. Quando este pacote passar, coloca no mesmo as seguintes informações:
 - endereço da estação do destino;
 - endereço da estação remetente;
 - dois caracteres da mensagem.
- 3) Este pacote é transmitido para a próxima estação e desta para a próxima e vai circulando pelo anel;

4) Quando este pacote passar pela estação de destino, ela reconhecerá seu endereço no pacote e copiará para si os dois caracteres da mensagem e o endereço da estação remetente. Ao final do pacote, a estação receptora coloca informações sobre o recebimento deste pacote que pode assumir quatro valores:

- pacote aceito (se o pacote foi lido normalmente);
- pacote ignorado (se o computador de destino estiver fora do ar);
- estação ocupada (se a estação ainda não teve tempo de processar o pacote anterior e portanto sem condições de aceitar este pacote);
- pacote rejeitado (se a estação estiver se comunicando com outra estação diferente e não desejar ser interrompida no meio desta comunicação);

5) Este pacote continua circulando pelo anel até a estação de origem (remetente) a qual esvazia o pacote, porém examinando a condição de recebimento do pacote. Se houve algum problema no recebimento, a estação deverá retransmitir o pacote;

6) Se houver mais caracteres a transmitir da mensagem, repete o procedimento de transmissão de um pacote (a partir do item 2).

Sobre este mecanismo de controle, há algumas considerações que vale à pena ressaltar. A primeira delas é que parte da capacidade de transmissão da rede é usada para transportar informação de controle da própria rede como endereços das estações de destino e origem, informações e recebimentos do pacote e outras informações de controle não mencionadas aqui. Na verdade usa-se apenas 42% da capacidade do pacote para transmitir informação útil. Este fato embora reduza a capacidade útil de transmissão da rede, ainda permite taxas de transferência efetivas da ordem de 4.2 Megabits/segundo.

Outro problema que pode ocorrer é o de uma estação após enviar um pacote, não esvazie o mesmo por defeito de funcionamento. Isto fará com que este pacote cheio fique circulando indefinidamente. Este tipo de erro é detectado pela monitora com o seguinte mecanismo: todo pacote cheio que passa pela monitora, recebe uma marca especial; se passar pela monitora um pacote com esta marca, significa que este pacote não foi esvaziado pela estação que o encheu e então a monitora o esvazia.

6 - CONFIABILIDADE

Em redes semelhantes, implantadas na Inglaterra, verificou-se que a taxa de erros é extremamente baixa. A taxa de erros da rede é cerca de um milhão de vezes menor que as redes digitais usando o sistema telefônico (10^{-11} contra 10^{-5}). Podemos então considerar a rede praticamente isenta de erros em termos do desempenho.

Embora a taxa de erros seja baixa, há a possibilidade de haver uma mensagem com erro. Este problema é solucionado através do protocolo de comunicação entre os computadores conectados.

7 - PROTÓCOLOS

As regras e procedimentos de comunicação entre as máquinas e processos, são modernamente definidos e implementados em níveis, onde cada nível se comunica com o nível equivalente da outra máquina, oferecendo ao nível superior um certo serviço, para tal utilizando os serviços oferecidos pelo nível inferior.

As funções de cada nível da Rede Local UFRJ são descritas a seguir (7,8).

APLICAÇÃO	
TRANSPORTE	Nível 3
LINHA	Nível 2
FÍSICO	Nível 1

NÍVEL 1: Nível de sinais elétricos; define que voltagem deve ser transmitida e a maneira de codificação dos bits. É implementado em hardware e corresponde à parte da repetidora da estação. É responsável em transformar um par de fios condutores em uma facilidade de comunicação.

NÍVEL 2: Este nível é responsável por transmitir uma mensagem completa entre duas estações. Tem uma parte que é implementada em hardware que corresponde ao mecanismo de transmitir um minipacote de dois bytes, do modo que foi visto anteriormente. Há também uma parte implementada em software que é responsável pela transmissão de vários minipacotes formando uma mensagem de até

2K bytes de comprimento. Neste nível, os pacotes são automaticamente repetidos em caso de não recebimento. É responsável em transformar uma facilidade de comunicação em um serviço de transmissão (e recebimento) de mensagens.

NÍVEL 3: Este nível é responsável pelas seguintes tarefas:

- a) Estabelecimento de ligação lógica entre dois processos de máquinas distintas;
- b) Controle de fluxo entre estes dois processos de modo que se um processo envia mais mensagens do que seu parceiro consegue aceitar, este nível equipara as velocidades dos dois processos;
- c) Ativação de processos remotamente;
- d) Controle da integridade das mensagens, evitando que mensagens sejam perdidas, duplicadas ou contenham informação com erro. É responsável em transformar um serviço de transmissão de mensagens entre computadores em um serviço de transmissão confiável entre processos de computadores diferentes.

O usuário se comunica com a rede através desse nível, que é responsável ainda por fragmentar as mensagens recebidas dos processos usuários em blocos de até 2K bytes.

O usuário utiliza apenas cinco primitivas para interagir com a rede que são chamadas como subrotinas:

1 - IDENT (NOME, TIPO).

Informa ao protocolo de transporte que um processo está ativo e pode portanto aceitar ligações com outros processos.

Parâmetros

NOME - nome do processo

TIPO - indica se esse processo aceita apenas uma vez várias ligações simultâneas.

2 - LIGA (NO, NOME, TIOUT, END, COD)

Pede ligação com o processo dado por NOME na estação NO. É encaminhado ao nível de transporte da estação desejada um pedido de ligação. Se o processo não estiver ativo, ele é ativado e seu endereço devolvido em END. Um código de retorno indica se o processo foi ativado ou não, se a estação desejada está desligada ou se passou TIOUT segundos e não houve resposta da outra estação.

Parâmetros:

- NO - endereço da estação onde está o processo desejado.
- NO ME - dá o nome do processo desejado.
- TIOUT - tempo em segundos pelo qual se deseja esperar pela resposta.
- END - parâmetro que retorna o endereço do processo desejado.
- COD - códigos de retorno, informando se a ligação foi completada com sucesso ou se não foi possível completar a ligação por algum motivo (estação desejada não estava ativa, processo desejado não estava presente, etc.)

3 - ESCR (END, TAM, VETOR)

Envia uma mensagem para um processo remoto.

Parâmetros:

- END - endereço do processo para o qual queremos enviar uma mensagem.
- TAM - tamanho da mensagem em caracteres.
- VETOR - vetor que contém a mensagem.

4 - LE (END, TIOUT, TAM, VETOR, COD)

Lê uma mensagem de determinado processo ou de qualquer processo.

Parâmetros:

- END - endereço do processo do qual queremos ler. Se END = zero, significa que queremos ler de qualquer processo e nesse caso END retorna o endereço do processo que enviou a mensagem.
- TIOUT - tempo máximo em segundo pelo qual desejamos esperar a mensagem.
- TAM - tamanho da mensagem lida.
- VETOR - vetor que contém a mensagem lida.
- COD - código de retorno:
 - 0 - leitura completada
 - 1 - passou-se TIOUT segundos e não chegou nenhuma mensagem para esse processo.

5 - FIM - Informa ao processo de transporte que esse processo deseja sair da rede.

8 - SERVIÇOS OFERECIDOS PELA REDE

Além do protocolo de transporte que permite a processos dos usuários uma maneira fácil de estabelecer comunicação com processos em máquinas remotas, uma série de outros serviços estão sendo implementados ou planejados para os usuários da rede. Esses serviços são:

- Protocolos de Alto Nível para Transferência de Arquivos.

Esse serviço tem a finalidade de oferecer aos usuários primitivas que facilitem a transferência de arquivos entre as diversas máquinas da rede.

Com isso, um usuário de micro que tem pouca capacidade de armazenamento poderá por exemplo, manter seus arquivos nos discos das máquinas maiores e copiá-las facilmente para seu próprio micro, total ou parcialmente quando for processá-los.

- Protocolos de Alto Nível para Submissão Remota de Programas.

Esse serviço permitirá ao usuário solicitar que um programa editado em uma máquina, seja incluído na fila de processamento de uma outra máquina. Com isso, um programa longo, pode ser completamente editado em um micro e posteriormente submetido para execução em uma máquina de maior porte.

- Concentradores de Terminais.

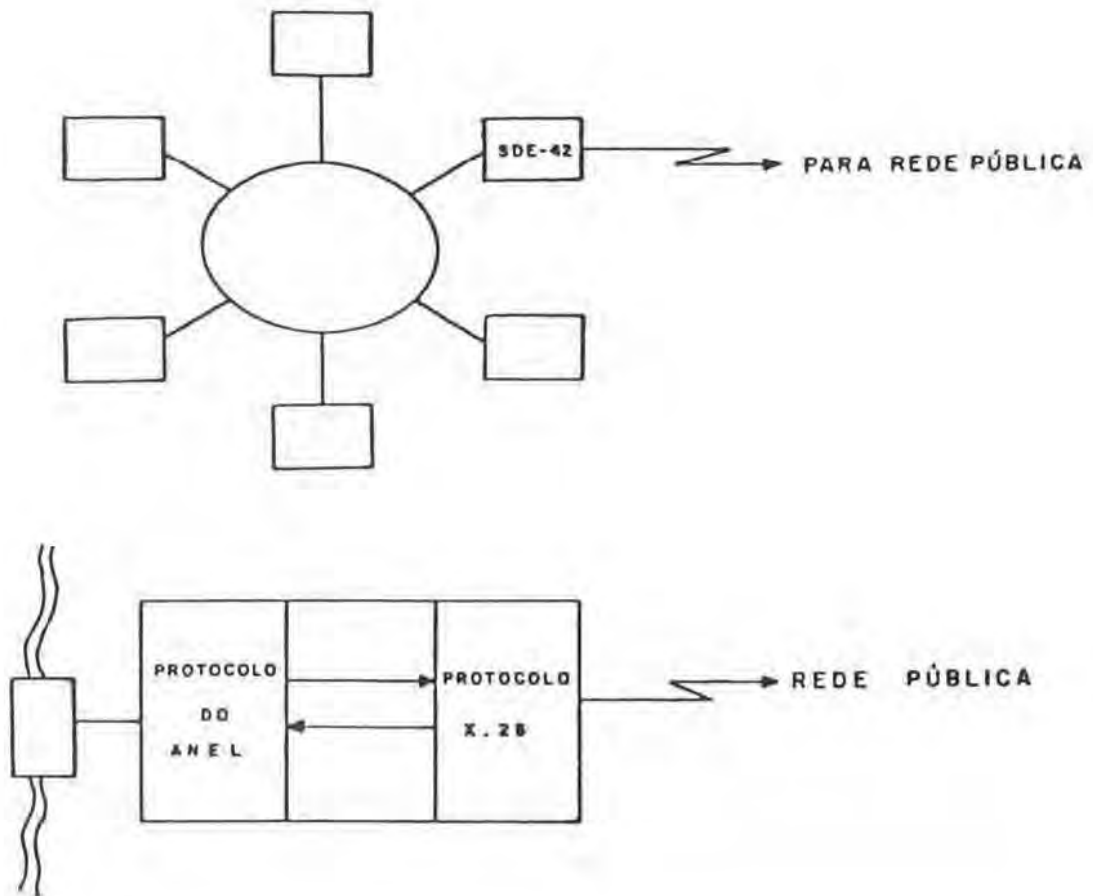
Esses concentradores farão a conexão de terminais assíncronos à rede, permitindo que um usuário em um desses terminais se conecte a qualquer um dos computadores ligados a rede como se estivesse diretamente conectado a ele.

O concentrador é baseado em um microprocessador Z-80 e pode controlar até 8 terminais. Eles poderão estar conectados diretamente a uma das estações do anel ou através de uma linha de velocidade mais baixa a uma estação especial, de modo a se poder instalar terminais em edifícios onde não circule o anel.

- Comporta para Rede Pública de Comutação de Pacotes.

A partir de 1983 estará disponível no Brasil uma rede pública de comutação de pacotes, que permitirá a troca de informações entre todos os computadores ligados a ela. A comporta (gateway) em desenvolvimento, será o caminho de saída para todos os computadores ligados ao anel e fará os seguintes serviços: conversão de protocolos do anel para o protocolo X-25 de acesso à rede pública, verificação de autorização para ligações externas, contabilidade

do uso da rede pública por parte de cada usuário.



A implementação da comporta, utiliza um microcomputador brasileiro EMBRACCI P SDE-42 com 48K bytes de memória e 2 unidades de diskette.

CONCLUSÃO

Foi apresentada uma descrição da rede local em anel desenvolvida pelo NCE/UF RJ no âmbito do projeto REDLAC, bem como dos serviços de alto nível necessários a sua efetiva utilização como suporte ao processamento de dados distribuído e ao compartilhamento de recursos computacionais.

- BIBLIOGRAFIA

- 1) "Moving away from main frames: The Large Computer Maker's Strategies for Survival", Business Week, February, 1982;
- 2) "Computação na UFRJ, Pesquisa Tecnológica, Avaliação e Perspectivas", Núcleo de Computação Eletrônica, 1981;
- 3) Clark D.D., Pogran K.T. and Reed D.P. "An Introduction to Local Area Networks", proc. of the IEEE, vol. 66, nº 11, November, 1978;
- 4) Araujo J.F. "Tecnologia de Redes Locais", Relatório Técnico NCE nº 0182, Fevereiro, 1982;
- 5) Hopper A, "Data Ring at Computer Laboratory, University of Cambridge", Computer Science Technology: Local Area Network, US National Bureau of Standards Special Publication, 1979, pp. 500-531;
- 6) Wilkes, M, V. and Wheeler, D.J., "The Cambridge Digital Communication Ring", Local Area Communications Network Symposium, Boston, May 1979, pp. 47-61;
- 7) Takano D.F. et. alu "Arquitetura e Implantação da Rede Local NCE UFRJ", Revista Brasileira de Computação - V.1 - nº 2, 1981, pp. 95 a 109;
- 8) Rodrigues G., "Descrição e Protocolo de Comunicação da Rede Local NCE/UFRJ", 8ª SEMISH, Florianópolis, Brasil, 1981;
- 9) Binns S, Dallas I. and Spratt E. "Further Developments on The Cambridge Ring Network at the University of Kent", Proceedings of IFIP TC 6 International In-Depth Symposium on Local Computer Networks - Florence, Italy, April 1982, pp. 183-204.

ARQUITETURA DE UMA REDE LOCAL COM
INTEGRAÇÃO DE SERVIÇOS DE DADOS E VOZ

Edit G. L. de Campos
Escola Politécnica
Universidade de São Paulo

RESUMO

Durante os últimos anos, o Laboratório de Sistemas Digitais (LSD) da Escola Politécnica investiu seus esforços no projeto e implementação de um nó chaveador de pacotes para redes públicas de comutação de pacotes. A arquitetura do nó é baseada na interligação de um conjunto de módulos processadores segundo uma estrutura de anel, com a utilização do método de acesso de inserção de atraso (ou de registrador), registrando taxas de transmissão de 1M bps.

Atualmente, tem-se investigado a possibilidade de integrar a comunicação de dados e de voz numa única rede, o que implica na necessidade de escolha de uma arquitetura que permita um alto grau de determinismo do atraso médio na transmissão de pacotes, principalmente em se tratando dos serviços de transmissão de voz. Este determinismo é importante na medida em que se deseja uma qualidade aceitável na comunicação vocal, que normalmente envolve taxas de transmissão de 64K bps para cada canal unidirecional, utilizando a técnica de codificação PCM ("Pulse Code Modulation").

Considerando este aspecto, a equipe do LSD optou por uma estrutura em anel para a rede local, baseando-se em estudos e trabalhos de simulação do comportamento de diversas arquiteturas (Ethernet-CSMA/CD, anel de "token", anel de inserção de atraso, "slotted ring") em condições variadas.

Tanto o anel de "token" como o de inserção de atraso apresentam resultados simulados satisfatórios para aplicações de

tempo real (voz); além disso, essas estruturas são relativamente simples de implementar, pois são caracterizados por ligações ponto-a-ponto com pouca utilização de circuitos analógicos (projeto basicamente digital), e pela flexibilidade de escolha dos meios de transmissão (par trançado, cabo coaxial, fibras óticas).

Cabe observar, também que além do "know how" adquirido com o desenvolvimento do nó chaveador de pacotes, foi realizado, no LSD, uma experiência interligando dois módulos processadores num esquema em anel, utilizando fibras óticas, com método de acesso "token pass".

Por motivos de facilidade de implementação e por questões de confiabilidade, pretende-se implementar, no LSD, uma interface para um anel de "token", com a meta de alcançar uma taxa de transmissão de 10M bps. Tendo em vista a integração de serviços, estão sendo estudados protocolos destinados a atender prioritariamente o serviço de voz, e ao mesmo tempo assegurar o grau de confiabilidade necessário à transmissão de dados (alta confiabilidade) e à transmissão de voz (baixa confiabilidade).

As estações conectadas à rede local serão os próprios equipamentos de computação, ou concentradores aos quais estarão ligados, através de interfaces apropriadas, terminais, aparelhos telefônicos ou qualquer outro equipamento síncrono ou assíncrono de baixa velocidade (até 9600 bps).

A rede local descrita deverá prever, também, dois servidores "gateway" (comportas), um para a rede pública de comutação de pacotes (acesso X.25) e outro para a rede pública de telefonia.

AVALIAÇÃO DE DESEMPENHO
DE REDES DE DADOS LOCAIS

J. Antão B. Moura e Jacques P. Sauvé

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS E TECNOLOGIA
GRUPO DE REDES DE COMPUTADORES
(DEE-DSC)
CAMPINA GRANDE-PB. - 58.100

RESUMO: Sub-redes em anel e em barramento para Redes de Dados Locais são estudadas e comparadas usando Atraso Médio de Mensagens e Vazão como medidas de desempenho. Simulação Digital é usada na investigação. O estudo considera tráfego homogêneo e não-homogêneo.

PALAVRAS-CHAVES: Redes de Dados Locais; Sub-Redes de Comunicação. Topologia em Anel; Topologia em Barramento; Protocolo de Acesso ao Meio de Transmissão; Avaliação de Desempenho; Simulação.

Março/1983

AVALIAÇÃO DE DESEMPENHO
DE SUB-REDES DE DADOS LOCAIS

J. Antônio B. Moura e Jacques P. Sauvé
GRC/UFPb

1. INTRODUÇÃO

Nos últimos anos, observou-se o desenvolvimento de um número de Redes de Dados Locais (RDLs) [FARB73, METC76, BIBA79]. A motivação primordial para tal desenvolvimento foi o compartilhamento de recursos: hardware, software e informação. As aplicações de RDLs incluem automação de escritórios, controle de processos e processamento distribuído. A organização básica de uma RDL pôde ser vista na Figura 1. Tal organização consiste de uma coleção de usuários que se ligam à sub-rede de comunicação via interfaces. A sub-rede de comunicação consiste das interfaces e de um meio de transmissão e transfere mensagens entre interfaces. O meio de transmissão pode ser um cabo coaxial, canal de rádio, fibra ótica, etc. No caso de um meio de transmissão usando cabo coaxial ou fibra ótica, a topologia é frequentemente estruturada em anel [FARB73] ou em barramento [METC76, BIBA79].

A escolha do protocolo de acesso afeta consideravelmente o desempenho da sub-rede de comunicação de uma RDL [LAM79, MOUR79, BUX81]. A avaliação de desempenho de protocolos de acesso ao meio de transmissão é portanto de interesse. No passado, a avaliação e comparação de desempenho de protocolos de acesso eram feitas à medida que se especificavam tais protocolos. Isto resultava frequentemente em estudos com suposições "ad hoc", o que dificultava uma comparação unificada de vários protocolos [TROP81]. Além do mais, ge

ralmente assumia-se que o tráfego gerado por cada interface tinha características estatísticas idênticas, i.e., os estudos supõem que o tráfego na sub-rede de comunicação era "homogêneo" [LAM79, MARK80, BUX81, TROP81]. O tráfego "não-homogêneo" é de interesse porque numa RDL com terminais e computador(es), o tráfego é comumente polarizado pelo(s) computador(es).

Recentemente, Bux [BUX81] considerou uma comparação unificada de 4 alternativas para a sub-rede de uma RDL. Sua investigação resumiu-se na obtenção das relações "tempo de resposta vs vazão" de cada uma das alternativas usando modelos analíticos. O tráfego na sub-rede foi ainda assumido ser homogêneo.

Este artigo contribui para a área de avaliação de desempenho de RDLs. Apresenta-se uma avaliação unificada e objetiva de 5 alternativas para a sub-rede de comunicação de uma RDL: um barramento com acesso aleatório (CSMA-CD*); um barramento com acesso aleatório (CSMA-CD) e com prioridade para o tráfego de reconhecimento [TOKO77]; um anel com ficha [NEWH69]; um anel com quadros [YU76]; e um anel com inserção de registro [LIU77]. No estudo desenvolvido, o desempenho destas 5 alternativas é avaliado para tráfego homogêneo e não-homogêneo.

* CSMA-CD: Carrier - Sense Multiple Access
With Collision Detection

O desempenho observado sob tráfego não-homogêneo é sensivelmente diferente daquele sob tráfego homogêneo. Isto indica que a consideração de tráfego não-homogêneo além de ser mais realista é também importante para uma comparação mais adequada das sub-redes de RDLs.

2. SUB-REDES DE INTERESSE

Nesta secção, apresenta-se a descrição das 5 alternativas para a sub-rede de comunicação de uma RDL que são de interesse neste estudo. O meio de transmissão das 2 primeiras alternativas é estruturado numa topologia em barramento como esboçado na Figura 2. As outras 3 sub-redes tem topologia em anel (Figura 3). O fluxo de informação no anel é assumido ser unidirecional. O tráfego na sub-rede é devido a pacotes de dados e pacotes de reconhecimento. Assume-se que para cada pacote de dados enviado e recebido corretamente, um pacote de reconhecimento é preparado e enviado de volta ao remetente do pacote de dados.

2.1. BARRAMENTO COM CSMA-CD

No protocolo CSMA-CD [METC76], cada interface monitora o meio de transmissão e transmite apenas quando o meio está desocupado. As interfaces monitoram as próprias transmissões e cessam de transmitir no instante em que detectam a presença de outras transmissões no meio. Neste caso, diz-se que ocorreu uma colisão. Afim de se resolverem colisões, as tentativas de retransmissão são marcadas para o futuro de acordo com uma função de distribuição de atraso chamada de

"função de retirada".

No estudo, serão consideradas duas funções de retirada: uma que dobra a média do atraso com cada colisão (Retirada Exponencial Binária) e outra que incrementa linearmente a média do atraso a cada colisão (Retirada Incremental Linear) [MOUR79].

2.2. BARRAMENTO COM CSMA-CD E COM PRIORIDADE PARA TRÁFEGO DE RECONHECIMENTO.

A sub-rede descrita em 2.1 trata os tráfegos de dados e de reconhecimento igualmente, i.e., o protocolo de acesso ao barramento é o mesmo indiferentemente do tipo de pacote sendo transmitido. O tráfego de reconhecimento contribui para a intensificação de colisões. Afim de se evitar esta intensificação, o protocolo CSMA-CD com prioridade para reconhecimento (CSMA-CD/PR) [TOKO77] força as interfaces esperarem 2τ segundos (a partir do instante em que o barramento é detectado desocupado) antes de iniciarem suas transmissões de pacotes de dados (segundo o protocolo CSMA-CD da secção 2.1). O parâmetro τ é o tempo de propagação na extensão total do barramento. As interfaces com pacotes de reconhecimento transmitem no instante em que o barramento for detectado desocupado (de acordo com o protocolo CSMA-CD da secção 2.1).

2.3. SUB-REDE EM ANEL COM FICHA

O protocolo de acesso ao anel é o mecanismo de controle a base de ficha de Newhall e Farmer [NEWH69]. Neste protocolo, uma ficha de controle (representada por uma sequência especial de bits) é passada sequencialmente ao redor do anel. Qualquer interface, ao receber a ficha, pode removê-la do anel, transmitir um paco

te e então passar a ficha para a próxima interface.

2.4. SUB-REDE EM ANEL COM QUADROS

O protocolo de acesso é o mecanismo de controle de Yu e Majithia [YU76]. Com este mecanismo, o anel é estruturado como um anel contínuo de quadros móveis de tipos alternantes: dados e reconhecimento. Pacotes de dados e reconhecimento são transmitidos em quadros de dados e reconhecimento respectivamente. Cada interface tem um nível de prioridade para uso dos quadros de dados. O "Supervisor do Anel*", se requerido, pode mudar o nível de prioridade de uma interface. Uma interface pode transmitir um pacote somente se o quadro que lhe chega corresponde ao tipo de pacote a ser transmitido e o quadro está desocupado ou endereçado a ela e se a interface tem nível de prioridade superior ou igual ao indicado no quadro (se o quadro for de dados).

2.5. SUB-REDE EM ANEL COM INSERÇÃO DE REGISTRO

O mecanismo de controle de acesso ao anel é o descrito em [LIU77]. Cada interface operando com este protocolo dispõe de dois registros (um Registro de Saída e um Registro de Atraso) e de uma chave 'S' para comutação entre esses dois registros (Figura 4). Enquanto a interface não quer transmitir, a chave 'S' coloca o Registro de Atraso em série com o anel (posição RA), i.e., os pacotes que chegam à interface do anel são colocados imediatamente no segmento seguinte do anel para a próxima interface. Uma vez cheio o Registro

* Um usuário da rede que se encarrega de supervisionar a operação da sub-rede de comunicação.

de Saída, a interface espera o fim da transmissão via Registro de Atraso (se for o caso de uma transmissão estar em progresso) e então comuta a chave para a posição RS, inserindo assim o Registro de Saída no anel. Caso cheguem novas transmissões à interface, elas são armazenadas no Registro de Atraso onde aguardam o esvaziamento do Registro de Saída. Neste instante, a chave é comutada de volta à posição RA.

3. AVALIAÇÃO DE DESEMPENHO

As medidas utilizadas na avaliação de desempenho das sub-redes são Atraso Médio de Mensagens e Vazão, $E[T]$ e V , respectivamente. As definições destas medidas são como seguem:

- $E[T]$: é a medida dos tempos de permanência das mensagens na sub-rede.
- V : é a taxa agregada máxima de tráfego que a sub-rede suporta com um atraso médio de mensagens finito.

Observa-se que o tempo de permanência de uma mensagem na sub-rede é igual ao intervalo de tempo entre a chegada da mensagem à interface remetente até a completa e perfeita recepção da mensagem pela interface destinatária.

Modelos para as sub-redes de interesse construídos através da teoria das filas são complexos e de difícil solução. Esta dificuldade deve-se principalmente aos seguintes fatores. Primeiro, a caracterização do comportamento das interfaces com relação a transmissões, tanto nas sub-redes em anel quanto nas sub-redes em barramento, tem que considerar a dependência probabilística entre novas chegadas de mensagens

às interfaces e o comprimento do intervalo entre duas transmissões consecutivas de cada interface (tempo efetivo de serviço). Segundo, o detalhamento dos modelos para sub-redes em barramento com protocolos CSMA-CD inclui a incorporação dos efeitos de colisões passadas (memória) e a reação dinâmica da Função de Retirada. A dependência estatística no primeiro fator é de relevância na avaliação do desempenho das sub-redes com tráfego não-homogêneo, pois neste caso, a análise tem que discriminar os tempos efetivos de serviço ao contrário do caso de tráfego homogêneo onde todas as interfaces tem tempos efetivos de serviço com as mesmas características estatísticas. A inclusão de memória nos modelos das sub-redes em barramento com protocolo CSMA-CD faz-se necessária pois o desempenho de tais sub-redes varia com a Função de Retirada [MOUR81].

Os modelos de simulação foram desenvolvidos e implementados nas linguagens C e GPSS360. Nos modelos, fazem-se as seguintes suposições:

1. O processo de chegada de mensagens para a interface número i ($i = 1, 2, \dots, 16$) é Poisson com taxa λ_i
2. O comprimento das mensagens (em bits) tem distribuição geométrica com média $1/\mu$
3. As mensagens são segmentadas em pacotes
4. A cada pacote é adicionado um cabeçalho contendo endereços das interfaces remetentes e destinatária, tipo de pacote, etc.
5. O meio de transmissão é livre de ruído
6. Para cada pacote de dados transmitido, um reconhecimento é preparado e enviado à interface remetente; não há re

conhecimento para pacotes de reconhecimentos e estes não são incluídos nos de dados.

7. Para o estudo de tráfego homogêneo, as interfaces do par remetente-destinatário são escolhidas uniformemente entre as 16 interfaces.

Para o caso de tráfego não-homogêneo, a interface número 1 é escolhida com probabilidade $\frac{1}{2}$ para ser remetente; a escolha das interfaces destinatárias é feita uniformemente entre as 15 interfaces restantes, as quais enviam todo o seu tráfego para a interface número 1.

8. Uma interface pode, em qualquer instante de tempo, estar transmitindo ou recebendo, mas não ambos simultaneamente. A mudança de um modo para outro é instantânea.

9. No caso de colisões nas sub-redes em barramento, o atraso de retransmissão é distribuído exponencialmente com média ' m ' ditada pela Função de Retirada da seguinte forma: para um pacote que sofreu ' c ' colisões, a Retirada Exponencial Binária (REB) atribui

$$m = 2^{c-1} (2\tau)$$

e com a Retirada Linear Incremental (RLI) [MOUR81],

$$m = \tau(3c - 1)$$

10. Transmissões em colisão são abortadas após 2τ segundos depois do início da primeira transmissão envolvida na colisão.

11. O atraso de propagação é idêntico para todos os pares remetente/destinatário e igual a τ segundos.

Os valores dos parâmetros usados na execução das simulações são dados na Tabela 1. Algumas observações são ainda necessárias. Primeiro, a capacidade dos Registros de Saída e de Atraso da sub-rede em Anel com Inserção de Registro corresponde ao comprimento do cabeçalho mais o tamanho de mensagens, ou seja, 8100 bits. Segundo, o comprimento de um quadro de dados na sub-rede em Anel com Quadros é 750 bits ($H + D$) e o comprimento de um quadro de reconhecimento, 100 bits. Finalmente, adotam-se 8 bits para o tamanho de ficha. Os resultados a serem apresentados tem intervalos de confiança (nível 90%) de $\pm 10\%$.

3.1. SUB-REDE EM BARRAMENTO

A figura 5 apresenta a curva $E[T]$ vs V (note que $V = \sum_k \lambda_k$) para as redes em barramento com os protocolos CSMA-CD e CSMA-CD/PR quando o tráfego é homogêneo. A Função de Retirada utilizada é a REB. Observa-se que até $V = 0.3 C$, as duas sub-redes apresentam essencialmente as mesmas características. Isto acontece porque a intensidade de tráfego é baixa, causando poucas colisões. À medida que o tráfego aumenta, aumenta também o número de colisões e as diferenças em desempenho entre as duas sub-redes se tornam perceptíveis. A sub-rede com o protocolo CSMA-CD sofre maior atraso que a sub-rede que dá prioridade de transmissão aos pacotes de reconhecimento. As vazões máximas observadas da Figura 5 são (com relação à capacidade do barramento) 43.5% e 42.2% para sub-redes CSMA-CD e CSMA-CD/PR respectivamente.

respectivamente.

O desempenho sob tráfego não homogêneo é mostrado na Figura 6, ainda para o caso da Função de Retirada REB. Consideram-se os casos de $\rho = 50\%$ e $\rho = 97\%$, i.e., a interface número 1 gera 50% e 97% do tráfego para a sub-rede respectivamente. Observa-se que à medida que ρ cresce, há uma melhora no desempenho. Isto ocorre porque as colisões agora diminuem, já que diminui o número de interfaces a gerar tráfego. A tabela 2 resume os valores da vazão das sub-redes em barramento operando com REB quando $\rho = 6.25\%$, 50% e 97%. Note-se que como se considera o caso de 16 interfaces, $\rho = 6.25\%$ corresponde a tráfego homogêneo. Como se desprende da Tabela 2 e da Figura 6, a sub-rede CSMA-CD/PR perde a vantagem de melhor desempenho sobre a sub-rede CSMA-CD (vide também a Figura 5). Isto ocorre porque, à medida que ρ aumenta, diminuem as colisões, e, portanto, há menos necessidade de dar prioridade ao tráfego de reconhecimento - esta prioridade, pelo contrário, é agora uma causa de desperdício na utilização da capacidade do barramento pelos pacotes de dados.

As curvas $E[T]$ vs V para o caso da Função Retirada RLI são apresentadas na Figura 7 quando $\rho = 6.25\%$ e $\rho = 97\%$. O desempenho de ambas as redes sob tráfego homogêneo é sensivelmente superior àquele propiciado com a Função de Retirada REB (vide Figura 5). A vazão da sub-rede CSMA-CD aumentou de 42.2% para 63% e da sub-rede CSMA-CD/PR, o aumento foi de 43,5% para 75%. Os desempenhos quando $\rho = 97\%$ são inalterados (compare com os da Figura 6): com $\rho = 97\%$, ocorrem poucas colisões diminuindo assim os efeitos da Função de Retirada no desempenho de ambas as sub-redes. A Função de Retirada RLI, entretanto, diminui os efeitos das característiti

cas do tráfego no desempenho das sub-redes (mais transparência às variações nas características do tráfego). Da figura 7, vê-se que a rede CSMA-CD/PR com RLI é menos sensível a variações em ρ .

3.2. SUB-REDES EM ANEL

As curvas $E[T]$ vs V para as três sub-redes em anel são apresentadas na Figura 8 (tráfego homogêneo). As vazões obtidas são $0.833C$, $0.930C$ e $1.67C$ para as redes com Ficha, Quadros e Inserção de Registro, respectivamente. O Anel com Inserção de Registro permite a utilização simultânea de diferentes trechos do anel (cada trecho com uma capacidade C), alcançando assim uma vazão agregada maior que C . O Anel com Quadros também permite transmissões simultâneas mas alguns quadros podem ser sub-utilizados devido à variação no tamanho dos pacotes. Isto faz com que tenha uma vazão menor que aquela do Anel com Inserção de Registro. Já o Anel com Ficha limita o número de transmissões, em qualquer instante, a no máximo uma. O atraso médio de mensagens para os Anéis com Ficha e com Inserção de Registros são aproximadamente iguais e menores que aquele do Anel com Quadros. $E[T]$ do Anel com Quadros é maior por causa do efeito de segmentação das mensagens em pacotes e por causa da espera por um quadro vazio. O Anel com Inserção de Registro apresenta as melhores características $E[T]$ vs V sob tráfego homogêneo.

As curvas $E[T]$ vs V quando $\rho=50\%$ e 97% são apresentadas na Figura 9. Da observação desta figura, conclui-se que a vazão do Anel com Ficha aumenta ligeiramente quando ρ cresce enquanto que ocorre o contrário para as outras duas sub-redes. Isto é por

que o paralelismo (transmissões simultâneas) dos Anéis com Quadros e Inserção de Registro gradualmente desaparece à medida que ρ se aproxima de 100% (quando $\rho = 100\%$, apenas a interface número 1 transmite mensagens). A vazão do Anel com Ficha aumenta com ρ porque diminui a interferência com tráfego de reconhecimento no segmento do anel saindo da interface número 1 (entre as interfaces 1 e 2, no sentido do fluxo de dados). O Anel com Quadros tem a menor vazão quando $\rho \rightarrow 100\%$ porque a interface número 1 não pode utilizar os quadros de reconhecimento. O comportamento relativo de $E[T]$ dos três anéis ainda é o mesmo observado sob tráfego homogêneo.

4. CONCLUSÕES

O desempenho de 5 alternativas para a sub-rede de comunicação de uma RDL foi avaliado através de simulação digital. As simulações produziram as curvas características de Tempo de Resposta Médio vs Vazão ($E[T]$ vs V) quando o tráfego na sub-rede era homogêneo e quando uma das interfaces polarizava a geração e recepção de mensagens (tráfego não-homogêneo). Observou-se que o desempenho das sub-redes variava com o tipo de tráfego. As redes em barramento e o Anel com Ficha apresentavam melhor desempenho para tráfego não-homogêneo. Isto devia-se à diminuição de interferência entre as interfaces à medida que o tráfego se torna mais polarizado. Por outro lado, o desempenho dos Anéis com Quadros e com Inserção de Registro era melhor com tráfego homogêneo, devido à capacidade de transmissão simultânea. Este paralelismo se desvanecia à medida que apenas uma interface passa a gerar o tráfego.

Em geral, o Anel com Inserção de Registro apresentou melhor desempenho. Esta sub-rede, porém, é muito susceptível às variações no tráfego da sub-rede, i.e., não há transparência ao tipo de tráfego. As redes em barramento com CSMA-CD/PR e o Anel com Ficha são

relativamente transparentes. A propósito, estas duas sub-redes apresentam essencialmente o mesmo desempenho (comparem-se as Figuras 7,8 e 9). O Anel com Ficha entretanto, tem um mecanismo de acesso centralizado enquanto que o das redes em barramento é distribuído. Isto pode constituir-se em vantagem para a sub-rede em barramento sob a ótica de confiabilidade. Neste aspecto, também sofre a rede de Inserção de Registro, pois as interfaces intervêm diretamente na operação do anel através do Registro de Atraso. Por sua vez, o Anel com Quadros apresenta o problema de sincronização das interfaces. Enfim, a seleção de uma determinada sub-rede discutida neste artigo, deve levar em consideração não apenas o nível de desempenho oferecido, mas também a complexidade do hardware em cada interface necessária para implementar o protocolo de acesso, e a implicação que este hardware tenha na confiabilidade de operação da RDL.

AGRADECIMENTOS

Os autores agradecem o suporte do CNPq a esta pesquisa.

REFERÊNCIAS

- BIBA79 Biba, K.J. e Yeh, J.W, "Fordnet: A Front-End Approach to Local Computer Networks", Anais do Local Area Computer Network Symposium, Boston, May 1979, pp. 199-215.
- BUX81 Bux, W., "Local Area Subnetworks: A Performance Comparison", Local Networks for Computer Communication, Ed. A. West e P. Janson, North-Holland, 1981, IFIP Series.
- FARB73 Farber, D.J. et al, "The Distributed Computer System", Anais do Congresso COMPCON 1973, fevereiro 1981, pp.31-34.

- LAM79 Lam, S.S., "Multiple Access Protocols", TR-86, Dept. of Computer Science, University of Texas at Austin, janeiro 1979.
- LIU77 Liu, M.T., e Reames, C.C., "A Loop Network for Simultaneous Transmission of Variable Length Messages", Distributed Processing, IEEE EH0127-1, setembro 1977, pp.3.31-3.36.
- MARK80 Mark, J.W., "Distributed Scheduling Conflict-Free Multiple Access for Local Area Communications Networks", IEEE Transactions on Communications, Vol. COM-28, No.12, dezembro/80
- METC76 Metcalfe, R.M e Boggs, D.R., "Ethernet: Distributed Packet Switching for Local Computer Networks", CACM Vol.19, No.7, julho 1976, pp.395-404.
- MOUR79 Moura, J.A.B., Field, J.A. e Wong, J.W., "Evaluation of Collision Control Algorithms in Ethernets", Anais do 6º Data Communication Symposium, Pacific Grove, California, dezembro 1979, pp.82-86.
- MOUR81 Moura, J.A.B. e Field, J.A., "Collision-Control Algorithms in Carrier-Sense Multiple Access (Collision Detection) Networks", IPC Press, Computer Communications, fevereiro 1981, pp.10-18.
- NEWH69 Newhall, E.E. e Farmer, W.D., "An Experimental Distributed Switching System to Handle Bursty Computer Traffic", Anais do Symposium on Optimization in Data Communications, outubro 1969, pp.1-32.
- TOBA75 Tobagi, F.A. e Kleinrock, L., "Packet Switching in Radio Channels: Part I - Carrier - Sense Multiple Access Modes and Their Throughput and Delay Characteristics", IEEE Transactions on Communications, Vol. COM-23, No.12, dezembro/75

- TOKO77 Tokoro, M. C Tamaru, K., "Acknowledging Ethernet", Anais do Congresso COMPCON 1977, pp. 320-325.
- TROP81 Tropper, C., "Local Computer Network Technologies", Academic Press, New York, 1981.
- YU76 Yu, W., "Design and Simulation of an Adaptive Loop-Type Computer Communication Network", Tese de M.A.Sc., Dept. of Electrical Engineering, University of Waterloo, 1976.
- MAJI74 Majithia, J.C. e Bhar, R., "Optimum Packet Size in a Packet - Switched Data Network", Electronic Letters, Vol.10, No.10, maio/1974, pp. 191-193.

PARÂMETRO	VALOR(ES)
- Capacidade do Anel/Barramento (C)	3 Mbit/s
- Comprimento do Cabeçalho (H)	100 bits
- Comprimento de Pacote de Reconhecimento (R)	100 bits
- Comprimento Médio de Mensagens ($1/\mu$)	1000 bits
- Comprimento Máximo de Mensagens	8000 bits
- Comprimento Mínimo de Mensagens	1 bit
- Comprimento de Pacotes de Dados (Porção do Dados) (D)	650 bits (Anel) *
	4000 bits (Barramento)**
- Atraso de Propagação do Meio de Transmissão (τ)	5 μ s

TABELA 1

VALORES DO PARÂMETROS USADOS NAS SIMULAÇÕES

* Calculado através da fórmula [MAJI74]: $D = \sqrt{2(H + A)/\mu}$

** Adotado para evitar degradação de desempenho [MOUR79]

SUB-REDE	δ	VAZÃO (%C)
CSMA-CD	6.25%	42.2
	50.0 %	58.0
	97.0 %	78.0
CSMA-CD/PR	6.25%	43.5
	50.0 %	56.0
	97.0 %	76.0

TABELA 2

VAZÃO(%C) VS δ DAS SUB-REDES EM BARRAMENTO COM REB

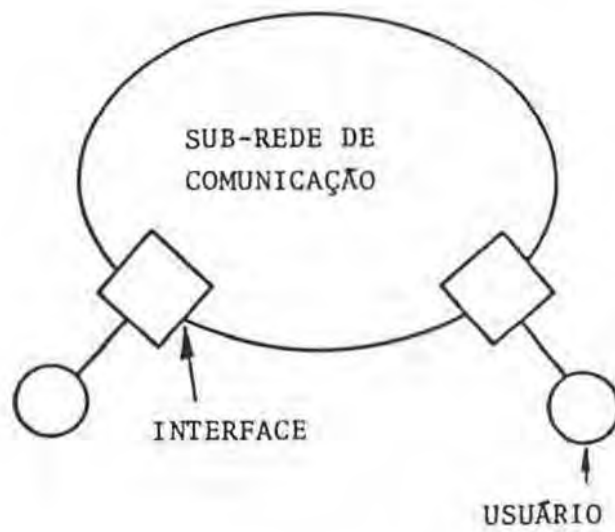


FIGURA 1: ORGANIZAÇÃO DA SUB-REDE DE UMA RDL

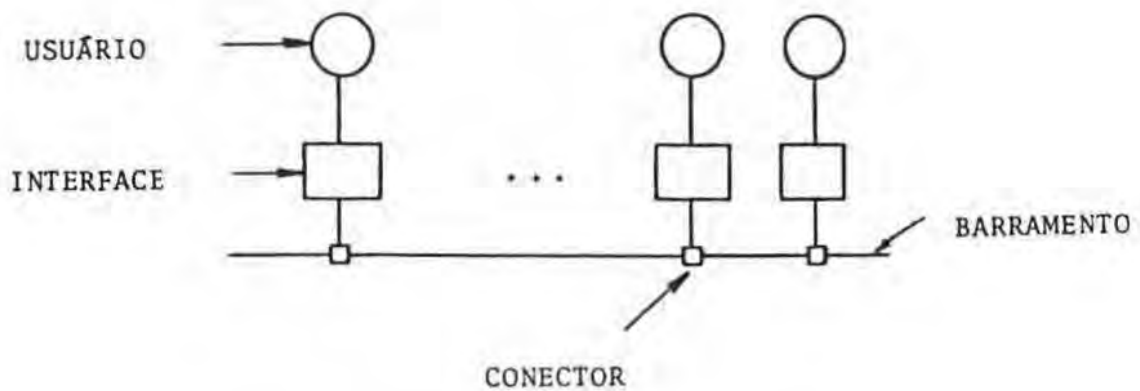


FIGURA 2: BARRAMENTO

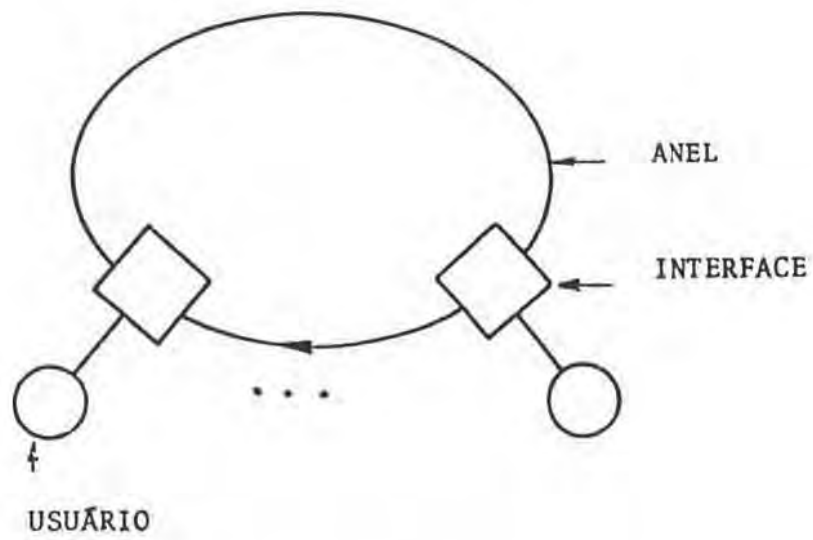


FIGURA 3: ANEL

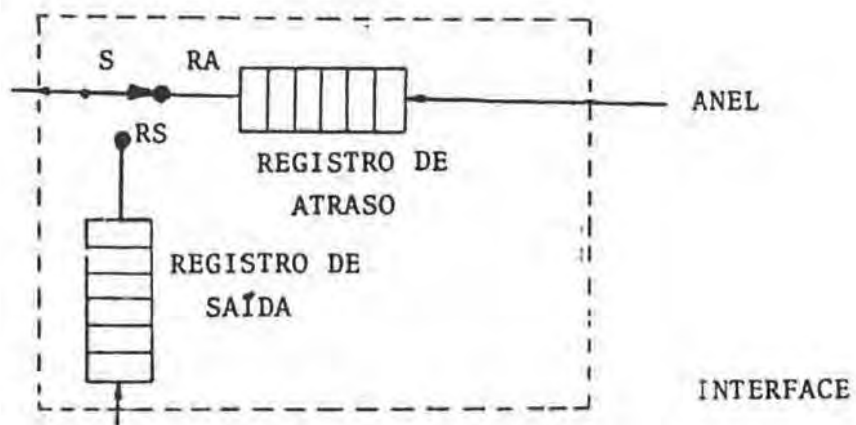


FIGURA 4: INSERÇÃO DE REGISTRO

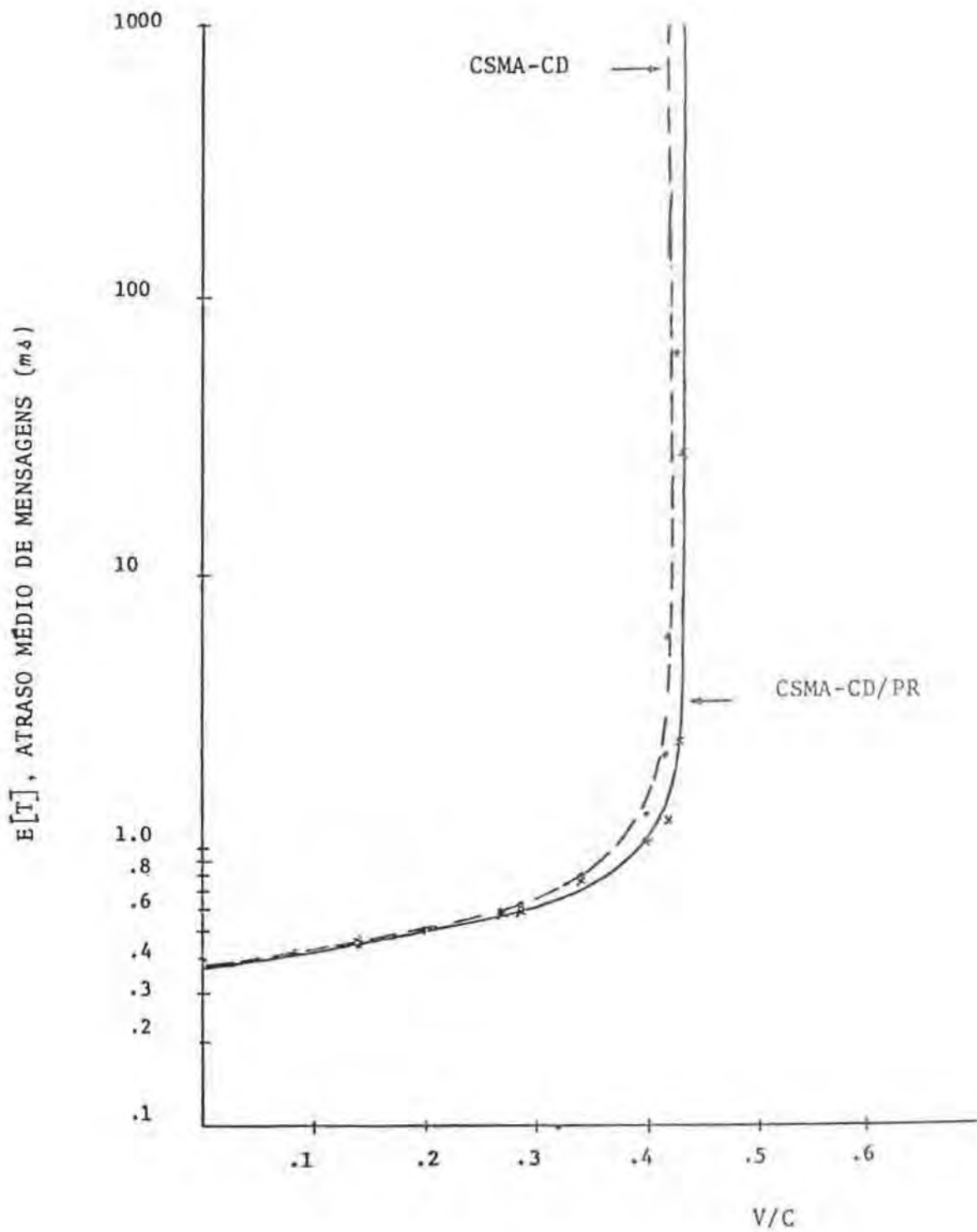


FIGURA 5: DESEMPENHO DAS SUB-REDES EM BARRAMENTO
COM REB E SOB TRÁFEGO HOMOGÊNIO

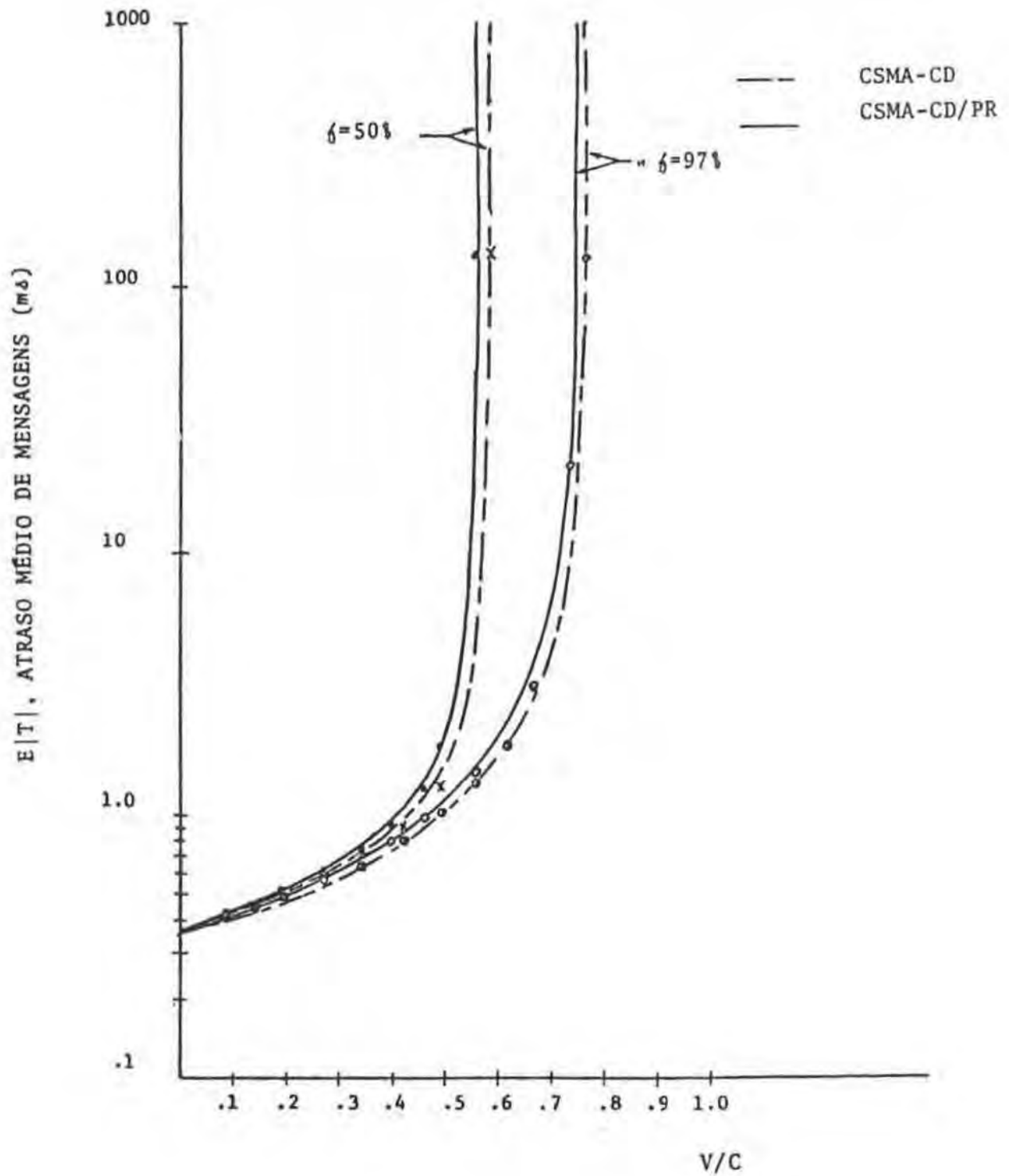


FIGURA 6: DESEMPENHO DAS SUB-REDES EM BARRAMENTO COM REB SOB TRÁFEGO NÃO-HOMOGÊNEO

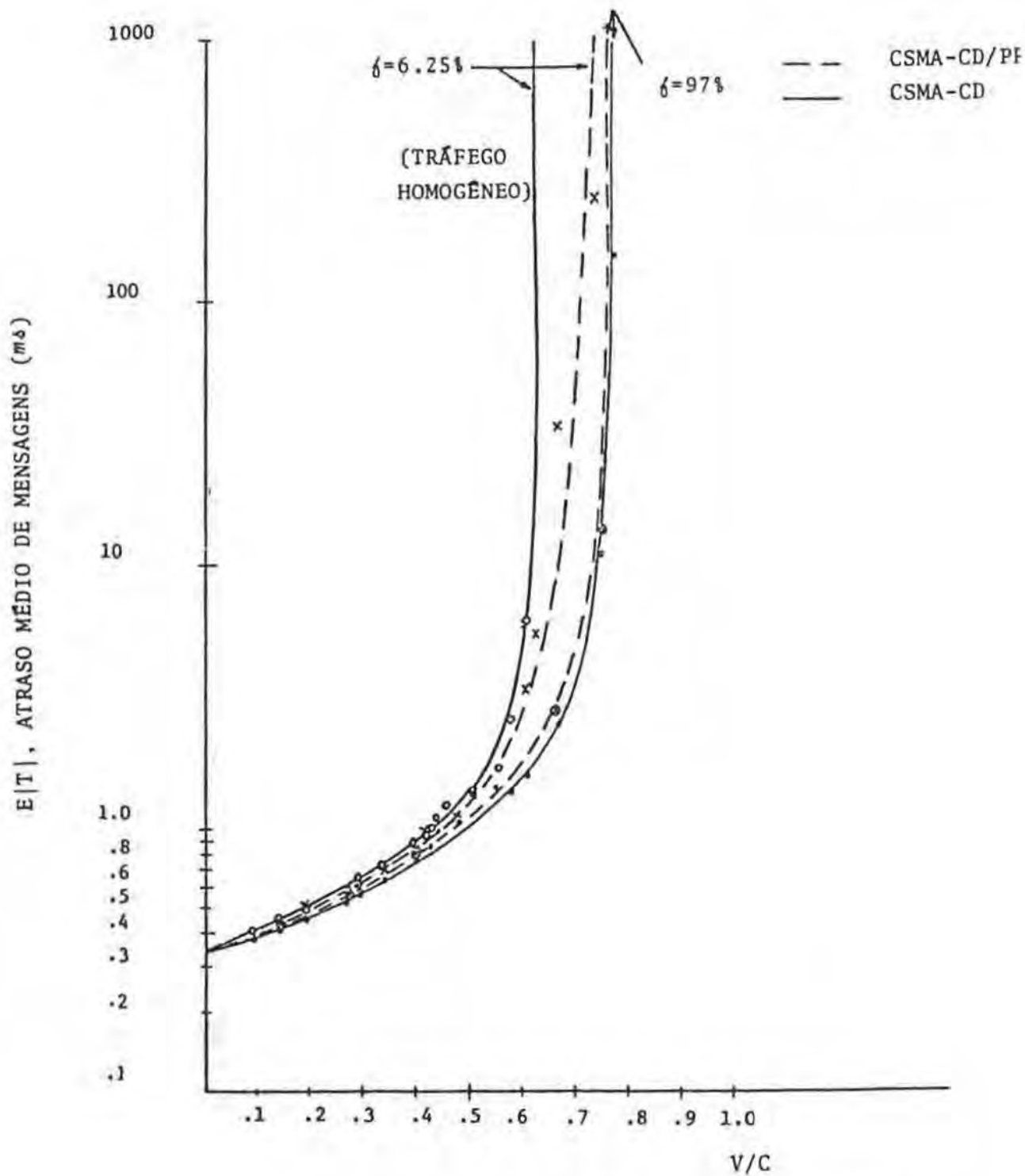


FIGURA 7: DESEMPENHO DAS SUB-REDES EM BARRAMENTO COM RLI SOB TRÁFEGO HOMOGÊNIO E NÃO-HOMOGÊNIO

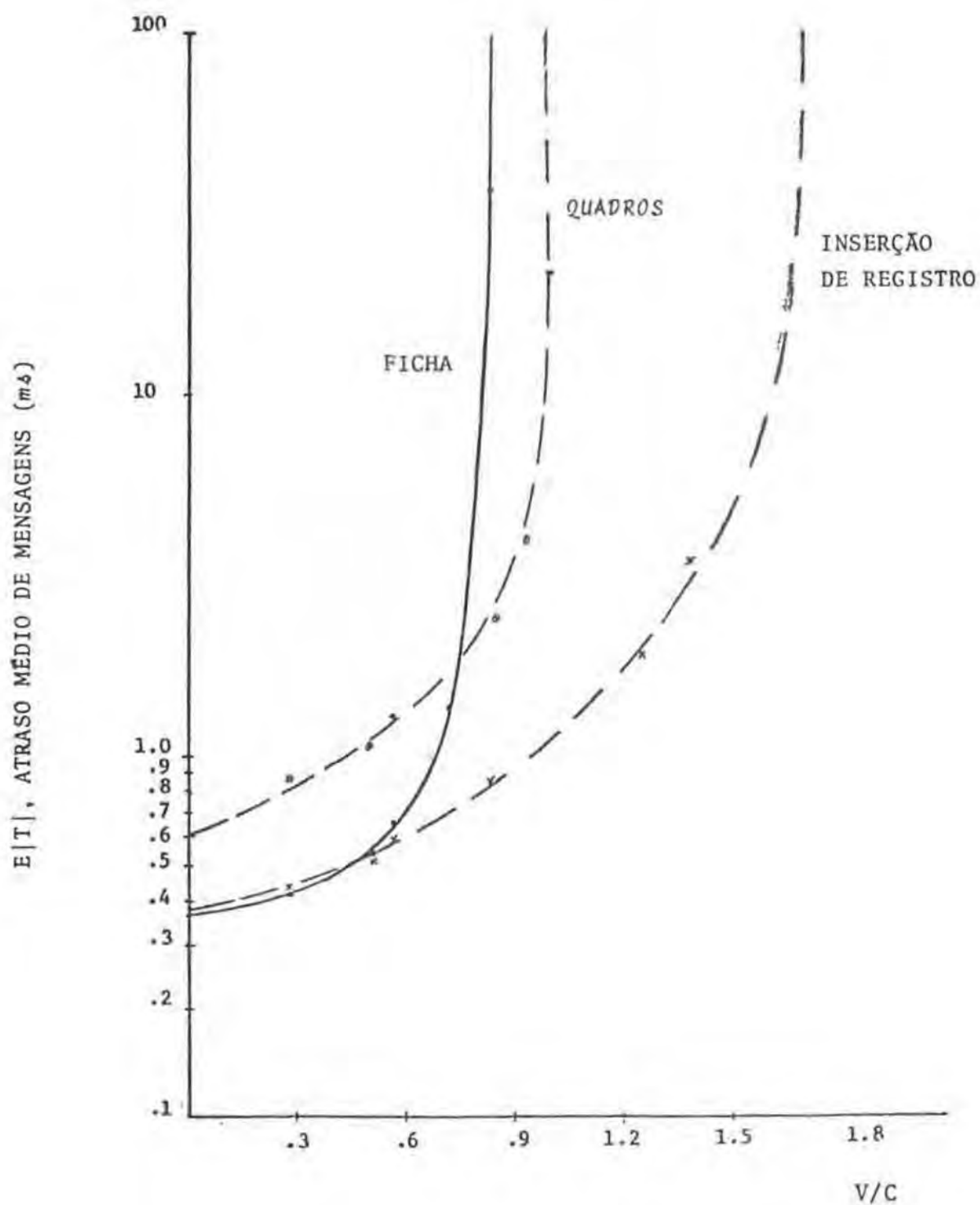


FIGURA 8: DESEMPENHO DAS SUB-REDES EM ANEL SOB TRÁFEGO HOMOGÊNIO

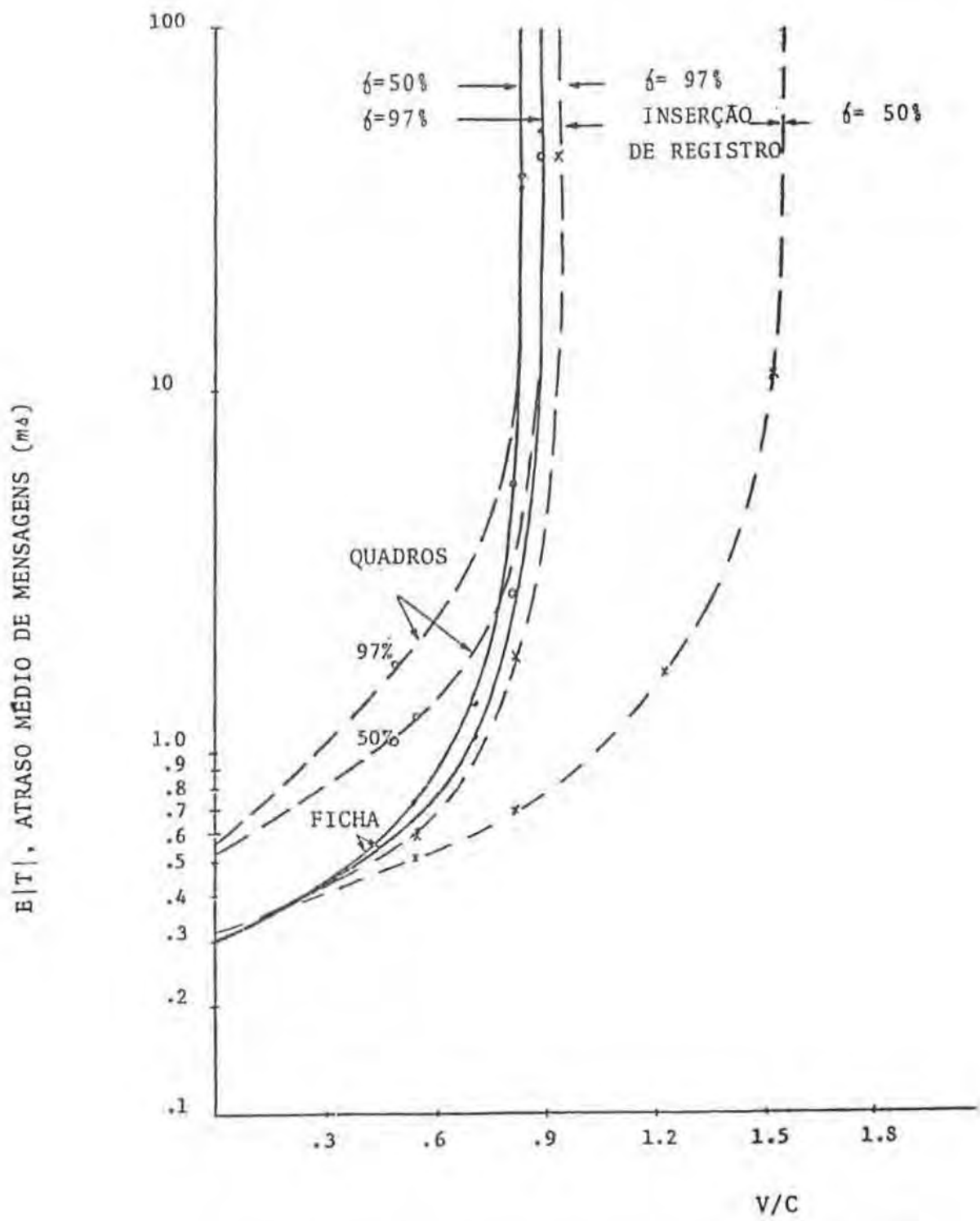


FIGURA 9: DESEMPENHO DAS SUB-REDES EM ANEL SOB TRÁFEGO NÃO-HOMOGÊNIO

A NOTE ON RANDOM GRAPHS, SHORTEST EDGES AND DISTRIBUTED COMPUTING

Michael M. Krieger *

March 26, 1983

Commemorating the 70th birthday of Paul Erdős

ABSTRACT

The average behavior of graph algorithms can often be analyzed by studying the expected structure of random graphs over a suitable sample space. Since the analysis considers "events" distributed throughout the graph, it can provide the basis for designing and analyzing algorithms which are similarly distributed. As such, results about random graphs have special relevance to modelling computation in distributed networks.

In this paper we prove some probabilistic results about shortest edges which are related to shortest paths and spanning trees. With this as a vehicle, we illustrate the connection between random graphs and distributed computing.

Computer Science Department

Universidade Federal de Minas Gerais

Engineering Systems Department

University of California, Los Angeles

This work was supported in part by the National Science Foundation Grants MCS 78-07468 and MCS 78-18924, the Advanced Research Projects Agency of the Department of Defense under Contract MDA 903-82-C-0064, and the Fulbright Commission (Brazil). For exceptional typing in difficult circumstances, thanks are due to Caruzo de Medeiros Araujo Filho.

INTRODUCTION

This paper is organized as follows: in section A we define the concept of random graphs and sketch the history of research about them; section B considers the application of results about random graphs to graph algorithms; sections C and D develop specific results concerning the distribution of shortest edges in a random graph and their relation to minimal spanning trees; and section E relates these results to algorithms and distributed computing and poses problems for future research.

Throughout our discussion we will consider undirected graphs with labeled vertices and without loops or multiple edges. Notation will follow standard practice, any exceptions being explicitly defined.

A. Random Graphs

We begin by introducing the concept of random graphs and their structure.

Let S be a collection of graphs and C some graph property of interest. If we can show that a randomly selected member G has property C with probability P , then we know something about the structure of a random graph (in S). In particular, if S is infinite and its members can be subdivided into classes S_i of graphs having i vertices, then we may be able to reveal the expected structure of sufficiently large graphs in S by proving results of the form

$$\lim_{n \rightarrow \infty} \text{Prob} \{ G \in S_n \text{ has property } C \} = P.$$

A classical example is: C is the property of connectivity and S_n consists of graphs with n vertices and $n/2(\log n + c)$ edges. In this case it is known that $P = \exp(-e)^{-c}$.

At this point we note in passing that this suggests a general approach to graph algorithms with improved average behavior: design algorithms to behave optimally for the most probable structure of the input and to work reasonably for the worst case. An early example of this approach is outlined in Krieger (13).

Clearly, then, to exploit the expected structure we must identify both the properties of interest and the appropriate classes of graphs over which to average.

The following three examples will illustrate some important models of random graphs and at the same time highlight developments in this area.

1. Classical Model

As an integrated body of results, the theory of random graphs begins with the seminal papers of Erdős and Renyi (4, 5 and 6). They considered the following model: let $S_{n,p}$ be the class of graphs with n vertices and $p = p(n)$ edges; then the random graph G_p is said to (not) have property C if

$$\lim_{n \rightarrow \infty} \text{Prob} \{ G \in S_{n,p} \text{ has property } C \} = 1 \text{ (= 0, resp) } \quad (1).$$

We will denote this limiting probability by $P(p, C)$.

Generally Erdős and Renyi were concerned with properties of connectivity and the existence of certain subgraphs (e.g., trees and cycles of specified size), for which they proved many results considerably more refined than (1).

In particular, they sought "threshold" functions for each property: $p(x)$ is a threshold for property C if whenever $\lim_{x \rightarrow \infty} (f(x)/p(x)) = \infty$ and $\lim_{x \rightarrow \infty} g(x)/p(x) = 0$, then $P(f, C) = 1$ and $P(g, C) = 0$.

Surprisingly, it was found that numerous graph properties indeed exhibit sharp thresholds.

It is convenient to think of these results as saying that almost all sufficiently large graphs with edge density greater than $p(n)/\binom{n}{2}$ have property C . Intuitively the class of graphs consists of those with average vertex degree $2p(n)/n$. Thus, if we think of the existence of each property as depending on the edge density, we can observe the "evolution of a random graph" by letting $p(n)$ become an ever faster growing function.

For many years after the initial works, relatively little appeared about the structure of random graphs. In the last few years, however, a host of new papers have appeared. This renewed activity almost surely derives from stimulus in the mid 70's of algorithmic applications of random graphs (see section B, below) and several new structural results, principally that of Posa (16). Leuker (15) includes a reasonably comprehensive bibliography.

2. Geometric Models

In networks and other applications of graphs, the relative position of nodes is relevant and so the distance (or cost) of an edge between two nodes plays a role. It may be the case, as in packet radio, that two nodes are connected if they are within a certain distance.

More specifically we define a sample space as follows: let R be some region (e.g. the unit square or the Euclidean plane) on which points (nodes) are distributed by a Poisson process of fixed density; two nodes v and w are connected by an edge whenever $d(v,w) < r$, a given radius of connection; we are interested in the expected structure of graphs formed this way. Since r parameterizes the edge density of the graphs, we can again ask about the probability of some property as a function of r for sufficiently large graphs.

In the most general setting, such questions fall within the difficult area of stochastic geometry. Specific questions involving connectivity and Euclidean distance were long ago considered as models for epidemics and other biological problems (see, for example, Gilbert (8)). Results paralleling the classical models have been elusive; a few are presented in De Witt and Krieger (2); De Witt (1) surveys the history in some detail and makes a significant start toward a Euclidean theory.

3. Fixed degree model

Let n labeled vertices and some number s be given. We form a (directed) graph by letting each vertex choose s neighbors at random. The space of all $\binom{n-1}{s}^n$ possible graphs thus formed is $D(n,s)$. A space $UD(n,s)$ of undirected graphs may be formed by ignoring direction and assigning to $G \in UD(n,s)$ the sum of the probabilities of graphs $G' \in D(n,s)$ which yielded G . Shamir and Upfal have recently proposed this model, considering asymptotically the relation between and the existence of 1-factors (17).

In many real network situations, this model may be more appropriate than that of Erdős and Renyi since it prevents the occurrence of isolated vertices. Note that $UD(n,s)$ is not a subspace of some $S_{n,p}$ since members of $UD(n,s)$ do not all have the same number of edges. For example, members of $UD(n,1)$ may have as few as $n/2$ or as many as n edges.

B. Algorithms and Random Graphs

As alluded to in the the previous section, by exploiting the most probable structure of graphs, algorithms can be expected to achieve much improved average performance. For this reason, one might have expected the results Erdős and Renyi (5) to have been seized upon by algorithmic analysts. In fact they were ignored for almost a decade and a half. This situation has a two-fold explanation.

First, algorithmic analysis - especially in probabilistic terms - only came into its own in the early seventies under the influence of Knuth(11) and the Cook/Karp results (9). Furthermore, the earlier approach to average analysis principally arose in the context of sorting where the analysis focused on familiar combinatorial properties of permutations and other linear objects.

Secondly, the work of Erdős and Renyi was little known beyond the circle of pure combinatorialists, most of whom had little interest in computing until recently. Moreover, the computer scientist who stumbled upon those early papers would almost surely have found them forboding, and turned away.

Perhaps the earliest computer reference to the possibility of using the ER work was that in Frank and Frisch (7). Actual algorithmic use of random graph results first appeared in Krieger(12) and Karp(10), the latter surveying a variety of problems. The application in(12) provides a convenient illustration. As noted earlier, a graph with average degree $\log n + c$ is connected with probability $\exp(-e^{-c})$. This implies that a random graph with $n \cdot \log n$ edges is almost surely connected. From this, it follows that the shortest $n \cdot \log n$ edges of a graph contain an MST. Thus, by modifying the standard MST algorithms to focus on the smallest $n \cdot \log n$ edges, fast expected time algorithms were achieved.

In a similar vein, improvements have been made for a variety of problems, including shortest paths, colorings, and Hamiltonian path (see references in(15)).

For shortest path and MST problems, short (least cost) edges play a critical algorithmic role. To that end, we analyze their structure in the following sections.

C. The Local Structure of Ordered Graphs

Let G be a complete graph with randomly assigned edge weights and distinguishable vertices $1, 2, \dots, n$. Since we are only concerned with relative order of the edges, we may take these weights to be the integers $1, 2, \dots, t$ where we define $t \equiv n(n-1)/2$ for convenience. The weight or cost of an edge $e = (v, w)$ will be denoted by $c(e)$ or $c(v, w)$. While our ultimate concern is with properties of undirected graphs, it is convenient to treat edges as directed as follows.

Def: An edge (v, w) of G is of type $(i:j)$ if $i \leq j$ and the edge is the i^{th} smallest from vertex v and j^{th} smallest from w . Such an edge is said to be on rank i with respect to v and rank j with respect to w . If $i < j$, then $e = (v, w)$ is directed from v (the head) to w (the tail) and vice versa if $i > j$; when $i = j$, e is bidirectional. By $NV(v)$ we denote the vertex nearest to v , i.e., that vertex w minimizing $c(v, w)$. $NE(v)$ denotes the edge $N(v, NV(w))$. Note that $NE(v)$ is of type $(1:j)$. The notations NV and NE obviously can be extended when v is replaced by a set of points to denote the vertex and edge closest to some member of the set. S_k consists of the set of all $(k:j)$ edges, $1 \leq j \leq n-1$.

Let F_k be the set of edges of type $(k:j)$ where k is fixed and $k \leq j \leq n-1$. An edge on F_k is called a k -edge. D_k denotes the set of $(k:k)$ edges.

We would like to compute $P_{i,j}$, the probability that (v, w) is of type $(i:j)$ as well as the expected size of F_k over all assignments of $1, 2, \dots, t$ to the edges of G . Thus we denote by σ_i the value of

$$\sum_{j=1}^{n-1} P_{i,j} \text{ and by } \tau_i \text{ that of } \sum_{j=i}^{n-1} P_{i,j} \text{ over all weight assignments.}$$

For clarity in writing proofs, we define $m \equiv n-2$. $(x)_r$ denotes the falling factorial $x(x-1)\dots(x-r+1)$. We make use of the following identities

$$(D1) \quad \sum (r-1)_x (t-r)_{y-x} = \frac{(t)_y}{y+1} \binom{y}{x}^{-1}$$

This follows readily from the binomial product identity (25) in Knuth (11).

$$(D2) \quad \sum_{k=0}^m \binom{m+k}{k} \cdot 2^{-k} = 2^m$$

Finally, we have

$$(D3) \quad \binom{m}{x} \binom{2m}{x+k}^{-1} = (x+k)_k \frac{m!}{(2m)_{m+k}} \binom{2m-x-k}{m-k}$$

Lemma 1:

$$P_{i,j} = \frac{2^{-\delta_{ij}}}{2^{n-3}} \binom{n-2}{i-1} \binom{n-2}{j-1} \binom{2n-4}{i+j-2}^{-1}, \quad \delta_{ij} = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{if } i \neq j \end{cases}$$

Proof: Let $e = (v,w)$ be an $(i:j)$ edge in G and assume e has length r .

Then there are

$$\binom{m}{i-1} \binom{m}{j-1} (r-1)_{i+j-2} (t-r)_{2m-(i+j-2)} \quad (1)$$

assignment of weights to the edges from v and w which make e an $(i:j)$ edge. The first two terms represent the choices of edges less than r , the third counts the ways to assign $1, 2, \dots, r-1$ to these edges, and the last counts the ways to assign weights greater than r to the remaining edges.

Using identity (D1) we sum (1) over r :

$$\sum_{r=1}^t (r-1)_{i+j-2} (t-r)_{2m-(i+j-2)} = \frac{(t)_{2m+1}}{2^{m+1}} / \binom{2m}{i+j-2}$$

In the case of $i = j$ we must multiply by 2 to account for the possibility that e is a j -edge for v and i -edge for w . Dividing by $(t)_{2m+1}$ for the 2^{m+1} edges considered here, yields the result.

Lemma 2:

$$\sigma_k = \frac{2}{n-1} - P_{k,k}$$

Proof: Each vertex has a k -edge. Thus there are n end points of k edges. Edges of type $(k:k)$ account for $2tP_{k,k}$ of them so there are $n - tP_{k,k}$ edges in S_k . The probability of such an edge is, thus

$$|S_k| / t = n/t - P_{k,k} = 2/(n-1) - P_{k,k}.$$

Lemma 3:

$$\tau_k = \frac{1}{n} \left[1 + \binom{2k-2}{k-1} / 2^{2k-2} \right]$$

Proof: $\tau_k = \sigma_k - \sum_{j=1}^{k-1} P_{j,k} = \frac{2}{n-1} + P_{k,k} - \frac{2}{2n-3} \sum_{j=1}^k \binom{m}{k-1} \binom{m}{j-1} / \binom{2m}{k-1+j-1}$ (2)

Now

$$\binom{m}{k-1} \binom{m}{j-1} / \binom{2m}{k-1+j-1} = \binom{k-1+j-1}{j-1} / 2^{k-1+j-1} = \frac{1}{2^{k-1}} \binom{k-1+j-1}{j-1} / 2^{j-1}$$

so using this and identity (D2) we find that the sum in (2) approaches

$$\frac{1}{2^{k-1}} \sum_{\ell=0}^{k-1} \binom{k-1+\ell}{\ell} / 2^{\ell} = \frac{1}{2^{k-1}} \cdot 2^{k-1} = 1$$

whence

$$\tau_k = \frac{2}{n-1} - \frac{2}{2n-3} + P_{k,k} = \frac{1}{n} + P_{k,k}$$

Similarly approximating $P_{k,k}$ yields the lemma.

Applying Stirling's approximation to $\binom{2k-2}{k-1}$ gives

Cor: For sufficiently large k and $n \gg k$

$$\tau_k = \frac{1}{n} \left[1 + \frac{1}{\sqrt{\pi (k-1)}} \right]$$

By way of illustration we note a few of these values

$$P_{1,1} = P_{1,2} = \frac{1}{2n-3}; P_{1,3} = \frac{n-3}{(2n-3)(2n-5)} = \frac{1}{4n}$$

$$P_{2,2} = \frac{n-2}{(2n-3)(2n-5)}; P_{2,3} = \frac{3}{2} \frac{n-2}{(2n-3)(2n-5)} = \frac{3}{2} P_{2,2}$$

$$t\tau_1 = \frac{3n^2-5n}{2(2n-3)} = \frac{3n}{4}$$

$$t\tau_2 = n - tP_{2,2} - tP_{1,2} = \frac{5n}{8}.$$

D. Minimal Spanning Trees

We first note the standard result about the structure of minimum spanning trees:

Lemma 4: An edge e is in the MST M of G if and only if there is a partition $L+R$ of the vertices of G for which $C(e)$ is least among edges $\{v,w\}$ satisfying $v \in L, w \in R$.

Remark 1: Clearly $NE(v) \in M$, since we only need take $L = \{v\}, R = V - \{v\}$, so M contains F_1 . Thus of the $n-1$ edges in M , by Lemma 2 we know that approximately 75% are 1-edges. This constitutes a combinatorial demonstration of a result first proved analytically by DeWitt (1).

The ease of finding these edges suggests the possibility of improved MST algorithms. Indeed the results in this paper grew out of attempts to exploit and generalize this observation.

If we consider F_1 as a subgraph, we have

Lemma 5:

F_1 is a forest with $|D_1|$ components none of which is an isolated point.

Proof: As a subset of h , F_1 can contain no cycles, i.e., F_1 is a forest. Let v_0 be some vertex in F_1 and $v_0, v_1, v_2, \dots, v_r$ be the path formed in F_1 by putting $v_i = NV(v_{i-1})$ until this procedure terminates. This only can happen when some v_r satisfies $NV(v_r) = v_{r-1}$ i.e., the path terminates in a (1:1) edge. Hence each component has a (1:1) edge.

If (v,w) and (x,y) are (1:1) edges, let $v = v_0, v_1, \dots, v_p = x$ be the unique path joining v and x (we can assume without loss of generality that $w \neq v_1$ and $y \neq v_{p-1}$.) But then we must have $v_p = NV(v_{p-1}), v_{p-1} = NV(v_{p-2}), \dots, v_2 = NV(v_1)$. This contradicts $v_0 = NV(v_1)$.

Corollary: Each component of F_1 consists of a (1:1) edge with a (possibly empty) tree rooted at each end. (see figure 1)

Note that the edges of the two trees are directed opposite to those in some definitions of rooted trees.

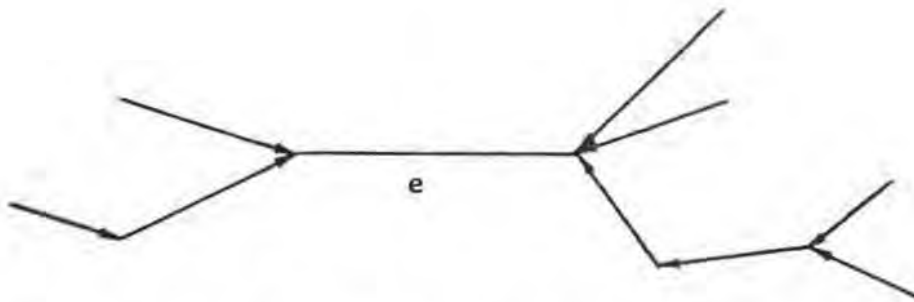


Figure 1: Typical Component of F_1 ; e is of type (1:1)

We know that the MST M contains F_1 and that the expected number of edges in F_1 is $.75n$. This is a good start at building M and leads us to ask if much more of M can be easily found in F_2, F_3 , etc? By way of answering this we have the following

Theorem: $E_x \{ |M \cap F_2| \} > .123n$

To prove this we need two lemmas.

Lemma 6: The probability that an edge e is of type (1:1) and $NE(e)$ is of type (2:k) is $O(2k/3^{k+1})$.

Proof: The number of ways the edge $f = NE(e)$ can be of type (2:k) and have $c(f) = r$ is given by

$$2(n-2) \binom{n-3}{k-1} (t-r)_{3(n-2)-1-k} (r-1)_k \quad (*)$$

In expression (*), $2(n-2)$ counts the number of ways to choose the endpoints of f and $\binom{n-3}{k-1}$ counts the choices of endpoints for the edges from z which are less than r (see Figure 2). Of the $3n-6$ edges from vertices x, y , and z , only these $k-1$ and e have lengths less than r and these can be assigned in $(r-1)_k$ ways. The remaining edges other than f can be given weights greater than r in $(t-r)_{3n-6-1-k}$ ways.

Summing over r using D_1 yields

$$2(n-2) \cdot \binom{n-3}{k-1} (t)_{3n-6} / (3n-6) \cdot \binom{3n-7}{k} = (t)_{3n-6} \frac{2k \binom{n-2}{k}}{(3n-6) \cdot \binom{3n-7}{k}}$$

Dividing by $(t)_{3n-6}$ --the total number of ways the lengths may be assigned-- and approximating the binomial coefficients completes the proof.

Lemma 7: Let e and e' be edges of type (1:1); let $f = NE(e)$ and $f = NE(e')$.
 Then, $\text{Prob}\{f \text{ is of type } (2:2) \text{ and } f = f'\} = 1/32n$.

Proof: Let $f = (v,w)$, $e = (x,v)$, $e' = (w,y)$ and $c(e) = r$, $c(f) = q$, and $c(e') = s$. Because of our assumptions on e, f , and e' , all edges in the coboundary of $\{x,y,v,w\}$ have a cost greater than q . There are $4n - 16$ such edges. Also the edges (x,y) , (x,w) , (y,v) have cost greater than q so we have to assign $4n - 13$ costs from the $t - q$ available. Similarly, we must assign the values f and s from $q - 1$. Finally, noting that x and y could have been chosen in $(n-2)(n-3)$ ways, we have the expression

$$\sum_q (n-2)(n-3)(q-1)_2 (t-q)_{4n-16} + 3$$

for the required probability which we can evaluate as

$$(n-2)(n-3)(t)_{4n-10} / (4n-10) \binom{4n-11}{2} = \frac{2(n-2)(n-3)}{(4n-10)_3 (t)_{4n-10}}$$

This approaches $1/32n$ as n becomes large.

We now can complete the proof of the Theorem, first noting that $P_{1,1} = \frac{1}{(2n-3)}$ implies that there are about $n/4$ edges of type (1:1). Each such edge has a nearest neighbor edge in the MST and of type (2:k). Of these, we know $n/9$ are of type (1:2) by Lemma 6. Using Lemma 7 we note that $n/64$ of the edges of type (2:2+) meet a type (1:1) edge at either end. These observations suffice to show that the number of members of F_2 in the MST is $n/4 - n/9 - n/64 = .123n$, proving the Theorem.

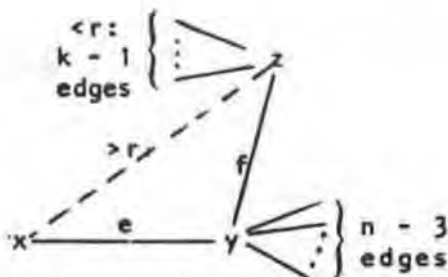


Figure 2: for Lemma 6

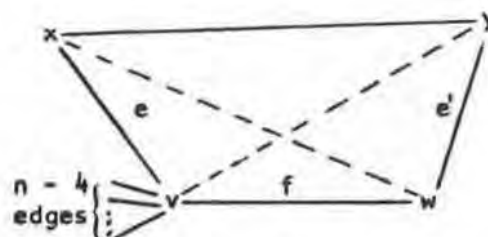


Figure 3: for Lemma 7

E. Shortest edges and algorithms

The forgoing results have implications for general as well as distributed algorithms.

We have seen that essentially $7/8$ of the MST can be found among the shortest 2 edges, i.e., in S_2 . This suggests that the entire MST will be found in S_k for a very small k . From an algorithmic point of view, these edges are easy to find. On the one hand, the shortest "few" from each vertex can be flagged when the data is input at virtually no extra computing cost. If the data are constantly being updated, then maintaining a heap of edge distances from each vertex provides an efficient way to access the short edges.

The actual integration of these edges into a complete, efficient MST algorithm is dependent on the choice of data structure for the graph representation. As such it merits a separate discussion as will be forthcoming in (14).

In distributed environments, nodes will generally 'know' the distance (cost, delay) to their nearest neighbors. Information about more distant nodes may be unavailable, costly, or stale. If this local information can be exploited for required computations, communications overhead can be decreased. Low-cost spanning trees are a commonly used structure supporting network activity and the previous sections suggest that such a tree can be built by each node making connection to its k nearest neighbors.

Of course, our analysis is not complete; namely, we need to know a value of k sufficient to assure connection. Putting $k = \log n$ is an attractive guess and raises:

Question 1: On a random weighted graph, what is the relation between E , the smallest $n \cdot \log n$ edges, and $S_{\log n}$.

In practice the absolute least cost tree is rarely required; hence we may only want to be sure that k is sufficient to guarantee connectivity at reasonable cost. Thus we ask:

Question 2: For what k is S_k connected? This is answered in (17) for the model of §A.3 above: $b > 2$ implies $\lim_{n \rightarrow \infty} \{\text{Prob}(G \in \text{UD}(n, b) \text{ is disconnected})\} = O(n^{-c/2})$.

Question 3: Suppose that m is the cost of the MST and s_k that of the cheapest tree in S_k ; what is the expected value of $s_k - m$?

Similar questions and analysis apply to shortest paths, which are perhaps even more fundamental to networks. For example, De Witt and Krieger (3) showed that generally shortest paths have relatively few edges. In the present context, we

would like to have answers to

Question 4: If E is the set of edges in the shortest path from v to w , what is the probability P_k that $E \subseteq S_k$?

Clearly answers to these and similar questions will provide tools for more efficient computation.

References

1. H.K. DeWitt, "The Theory of Random Graphs with Applications to the Probabilistic Analysis of Optimization Algorithms", UCLA, 1977 (dissertation).
2. H.K. DeWitt and M.M. Krieger, "Expected Structure of Euclidean Graphs" in Algorithms and Complexity: New Directions and Recent Results, ed. J.F. Traub Academic Press, New York, 1976, p.451.
3. _____, "Expected Structure of Shortest Paths and Algorithm Behavior", in Proc. 10 th S-E Conf. Combinatorics, Graph Theory, and Computing, pp 367-380.
4. P. Erdős and A. Renyi, "On Random Graphs, I", Publicationes Mathematicae Debrecen 6 (1959), pp. 290-297; reprinted in Paul Erdős: The Art of Counting J. Spencer, ed., MIT Press, Cambridge, Mass., 1973.
5. _____, "On the Evolution of Random Graphs", Magyar Tud. Akad. Mat. Kut. Int. Közl. 5 (1960), pp. 17-61; reprinted as above.
6. _____, "On the Strength of Connectedness in Random Graphs", Acta Math. Acad. Sci. Hung. 12 (1961) pp. 261-267.
7. H. Frank and I. Frisch, Communications, Transmission, and Transportation Networks, Addison-Wesley, (1971) p. 390.
8. E.N.Gilbert, "Random Plane Networks", J.SIAM, 9 (1961) pp. 533-543.
9. R.M.Karp, "Reducibility among combinatorial problems," in Complexity of Computer Computations, R.E.Miller and J.W.Thatcher, eds., Plenum Press, New York, 1972, pp. 85-104.
10. _____, "The Probabilistic Analysis of Some Combinatorial search Problems as in 2 above.
11. D.E.Knuth, The Art of Computer Programming, Vol. 1: Fundamental Algorithms, Addison-Wesley, Reading, Mass., 1973, p. 94 et seq.; see also analysis in Vol. 2 and 3.
12. M.M.Krieger, "Notes on Algorithmic Analysis", UCLA Computer Science Department, 1974-75.
13. _____, "Random Graphs and Minimal Spanning Tree Algorithms", in Algorithms and Complexity: New Directions and Recent Results, ed. J.F.Traub, Academic press, New York, 1976, p.471.
14. _____, "Graphs, Data Structures, and Spanning Trees: A Tutorial for Fast Implementations" (in prep.)
15. G.S.Leucker, "Optimization Problems on Graphs with Independent Random Edge Weights", SIAM J. Comp., pp. 338-351.
16. L.Posa, "Hamiltonian circuits in random graphs", Discrete Math., 14 (1976) pp. 359-364.
17. E.Shamir and E.Upfal, "One-Factor in Random Graphs Based on Vertex Choice", Discrete Math.

TÍTULO: CONSTRUCTOR - UMA PROPOSTA DE UMA FERRAMENTA PARA
PROJETO E IMPLEMENTAÇÃO DE SISTEMAS DISTRIBUÍDOS

Autor: Paulo F. W. Keglevich de Buzin

Instituição: Curso de Pós-Graduação em Ciência da Computação

RESUMO

No presente trabalho é apresentado o problema da programação distribuída e o relacionamento do ambiente de processamento distribuído com o ambiente de processamento paralelo, concorrente e seqüencial. Baseado no problema de processamento distribuído, é apresentado um modelo que distingue a parte física da parte lógica de um ambiente distribuído. É desenvolvido um modelo de arquitetura do sistema distribuído para a parte algorítmica, e baseado neste é definida a linguagem de alto nível. Constructor de programação distribuída com mecanismo de sincronização e de "scheduling" baseado em envio de mensagens.

A seguir, são apresentadas uma série de exemplos da eficácia da linguagem para resolver problemas importantes, que vão desde controle de periféricos em hardware até transações de Banco de Dados. Depois são apresentados aspectos de implementação. Finalmente, nas conclusões é constatada a introdução de um novo estilo de programação, inerente à programação em ambientes distribuídos.

SUMÁRIO

1. INTRODUÇÃO	166
2. O MODELO PARA O PROJETO DO CONSTRUCTOR	167
3. A PROPOSTA	173
3.1. Tipos e declarações	174
3.2. Mecanismo de sincronização	176
3.3. Comandos.....	177
3.4. Processos	179
3.5. Construção de ambientes	180
3.6. Servers	182
3.7. Clients.....	186
3.8. Interfaces	187
3.8.1. Interface Hardware	187
3.8.2. Interface Software	189
4. EXEMPLOS E APLICAÇÕES	192
4.1. Construindo mecanismos de sincronização e seqüencialização	192
4.2. "Rendezvous" Cliente/Servidor	194
4.3. "Readers" e "Writers"	197
4.4. Transações	198
4.5. Escalonamento para otimização de acesso a disco	200
4.6. "Driver" de console	201
4.7. Carga de código executável	204
5. ALGUMAS CONSIDERAÇÕES A RESPEITO DA IMPLEMENTAÇÃO ...	205
6. CONCLUSÃO	207
7. AGRADECIMENTOS	208
8. BIBLIOGRAFIA	209

1. INTRODUÇÃO

Dentro do contexto do projeto da rede local da UFRGS - REDURGS [1], especial ênfase é dada ao software. O objetivo básico na abordagem do software é obter um suporte adequado para as várias aplicações possíveis sobre uma rede de processamento distribuído. A tarefa mais importante no desenvolvimento deste suporte é o projeto de uma linguagem de alto nível de programação distribuída onde seja oferecido ao projetista de sistemas mecanismos para sincronização e envio de mensagens sob forma de primitivas simples e poderosas, além de outras vantagens como veremos a seguir.

Os mecanismos e primitivas do sistema são projetados de tal forma que seja possível construir com simplicidade; topologias de rede, mecanismos de sincronização de processos, controle de tempo real, escalonamento de processos, etc... O nome escolhido para o sistema é CONSTRUCTOR porque foi projetado de modo a poder construir qualquer ambiente de processamento de programa.

O sistema CONSTRUCTOR é constituído de vários níveis de protocolos, como sugerido em [1]. Procurou-se diminuir a complexidade do kernel distribuído de modo a facilitar a sua implementação, mas sem limitar o poder de resolução de problemas na linguagem CONSTRUCTOR de alto nível.

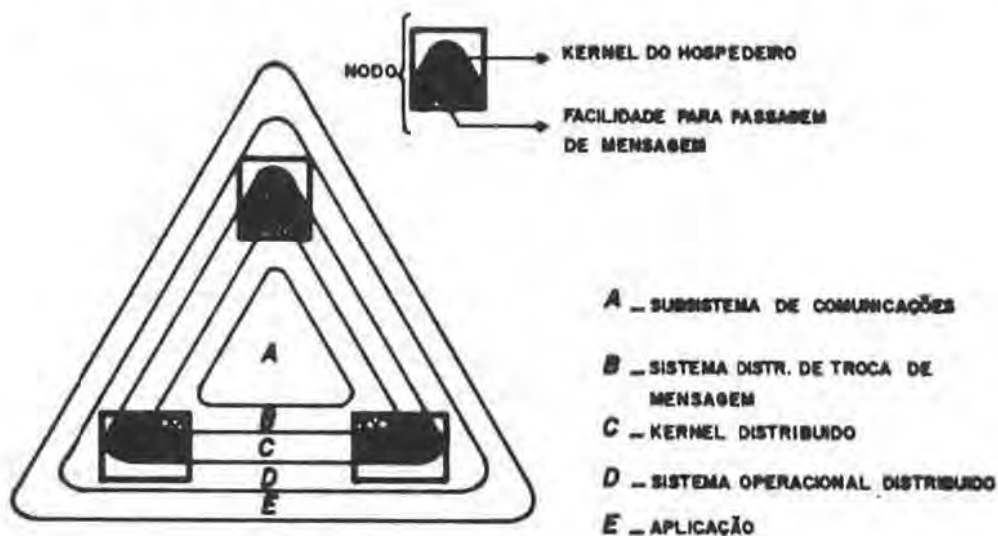


Fig. 1. Níveis de protocolo do ambiente distribuído CONSTRUCTOR.

2. O MODELO PARA O PROJETO DO CONSTRUCTOR

Ao resolver os problemas de processamento distribuído, também teremos soluções para problemas de ambientes de processamento paralelo, concorrente e seqüencial.

Neste trabalho entende-se como um ambiente de processamento paralelo aquele que possui vários processadores fortemente conectados para execução de processos, ao passo que no ambiente concorrente temos só um processador para a execução dos vários processos. Deste modo, um ambiente distribuído engloba um ambiente paralelo, que por sua vez engloba um ambiente concorrente, que ainda por sua vez engloba um ambiente seqüencial (fig. 2).

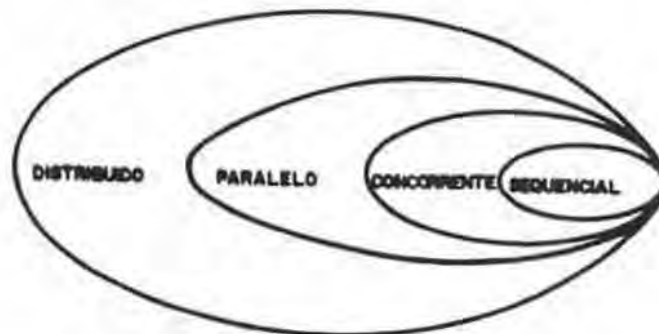


Fig. 2. Relacionamento entre os diferentes ambientes em computação.

Até ao nível de ambiente concorrente não existe preocupação com a localização de unidades de programa. Com o processamento paralelo existe a preocupação com a localização das unidades de programa ou com a associação algoritmo-processador para a computação de um programa, sem no entan

to preocupar-se com a característica do meio de comunicação. Já na construção de ambientes distribuídos existe a preocupação com o meio de comunicação, de modo que se procura alo_{car} na rede módulos auto-contidos que se intercomunicam de modo que seja possível controlar as situações de erro e de retardo de mensagens.

O processamento distribuído necessita de dois níveis de esquema de programação para obter uma abstração da distribuição [3], ou seja, a especificação da configuração deve ser independente da implementação das partes algoritmicas do programa, e vice-versa. Desta maneira o sistema apresentaria a característica de poder construir de uma maneira versátil um ambiente distribuído.

Dentro do modelo escolhido para o ambiente do sistema CONSTRUCTOR, é possível distinguir entre o sistema real (redes, nodos, hospedeiros, processadores e memórias) e o sistema lógico ou parte algoritmica (processos e módulos), os quais depois de definidos podem ser associadas para construir um sistema como um todo.

A concepção da parte algoritmica do constructor parte do modelo de arquitetura distribuído (Distributed System Architecture Model [2]). Dentro deste modelo, um dos maiores objetivos no projeto de um sistema operacional distribuído é fornecer aos usuários objetos abstratos implementados por SERVERs. Um objeto abstrato pode ser especificado por: um conjunto de estruturas de dados e um conjunto de operações ou funções. Um SERVER pode ser construído por outros SERVERs, de forma que é possível o estabelecimento de uma hierarquia.

Incluiu-se no modelo o conceito de CLIENT (baseado no conceito de procedimentos nos "Data Flow Description" da análise estruturada) que atuam sobre os SERVERs para, por meio de suas operações, implementarem aplicações ao usuário.

Os módulos podem ser CLIENTs, SERVERs e INTERFACES. Os únicos módulos que implementam operações são os SERVERs. Os processos dentro de um CLIENT só podem comunicar-se entre si por meio de operações implementadas em SERVERs. O módulo INTERFACE representa a interconexão entre dois ambientes diferentes, permitindo deste modo incluir na configuração de um ambiente elementos inteligentes tais como terminais, periféricos em geral, interfaces com outras redes de comunicações, transdutores ou coletores de processos de fabricação, etc... Também permite representar a interconexão entre ambientes diferentes de software (o que interessa para sistemas operacionais). Este tipo de módulo interage com o seu ambiente também por meio de operações em SERVERs, e somente os SERVERs podem atuar sobre as interfaces. Um interface é composto de um protocolo, um conjunto de dados em uma dada formatação e um conjunto de funções de atuação sobre o mesmo. Um SERVER acoplado a um interface pode funcionar como "driver" em software e implementar recursos para o ambiente.

O meio de comunicação entre os elementos do sistema real é assumido como fracamente conectado ("loosely connected") possuindo um retardo de modo que a produção de um evento e a sua materialização é distinguível. No caso do meio de comunicação ser o elemento passivo MEMÓRIA (compartilhado), então a comunicação é fortemente conectada e a produção de um evento e a sua materialização é indistinguível, pois não há retardo na comunicação. A comunicação sempre se dá entre elementos ativos, a maneira pela qual ocorre a comunicação é definida pelo protocolo, entre os mesmos [13].

De acordo com o modelo considerado, cada camada de protocolo da rede interage horizontalmente por meio de CLIENTs e SERVERs, e verticalmente por meio de INTERFACES (fig. 2).

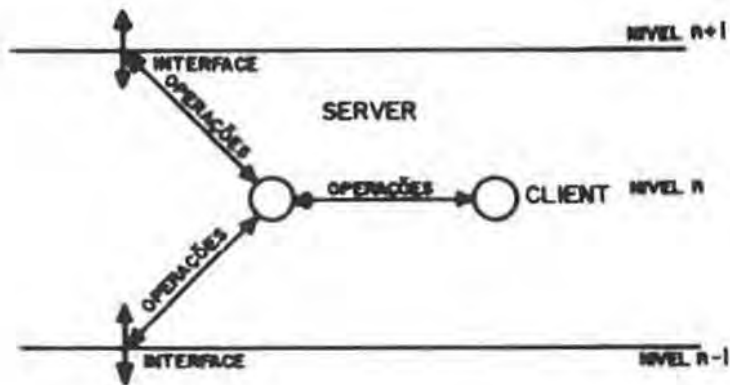


Fig. 3. Interação entre os componentes de um ambiente.

Desta forma a comunicação horizontal seria dentro do ambiente do sistema e a comunicação vertical seria para fora do ambiente do sistema. Assim, a interconexão entre dois ambientes se daria por meio do INTERFACE. Com este modelo é possível implementar a intercomunicação entre dois hospedeiros via SERVERs. (fig. 3).

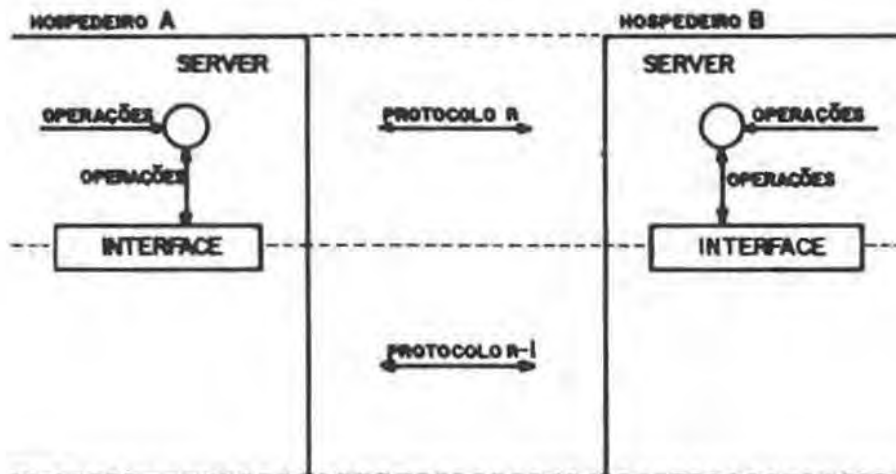


Fig. 4. Interconexão entre ambientes em dois níveis de protocolo.

Entende-se como ambiente o conjunto de recursos disponíveis e acessíveis de uma maneira uniforme dentro de um sistema único por meio de uma série de serviços fornecidos. Assim por exemplo, um nível de protocolo em um modo de rede pode ser considerado como ambiente. Da mesma forma, os elementos de um controlador de periféricos ou de processo podem ser considerados como um ambiente.

Um ambiente é construído definindo os elementos reais que o constituem e as conexões entre os mesmos, a seguir são definidos os elementos lógicos (CLIENTS, SERVERS e INTERFACES) que depois são localizados sobre os elementos reais.

O sistema CONSTRUCTOR é definido de forma que cada elemento ativo necessite apenas ter os endereços dos outros elementos do mesmo ambiente e suas condições de acesso. O restante das informações necessárias para verificação de disponibilidades de cada elemento é de responsabilidade única do próprio elemento ou do elemento ativo mais fortemente conectado ao mesmo. Isto facilita a manutenção da consistência em um sistema e reduz drasticamente o acúmulo de informações sobre o ambiente em cada elemento.

Por operações entende-se um conjunto de funções ou instruções executadas de modo exclusivo causando transformações sobre dados e/ou estados do sistema. As execuções de operação são mutuamente exclusivas e podem ser feitas por qualquer processo, seja CLIENT ou SERVER, desde que tenha acesso às mesmas. A execução de uma operação pode ser feita por envio de informação (comunicação por mensagens) ou por envio e recepção de informação (chamada remota de funções ou procedimentos). No último caso, o processo que ativou a operação fica suspenso até a completa execução da mesma.

A seguir é apresentada a proposta da linguagem CONSTRUCTOR, de maneira sucinta e informal. Esta primeira proposta ainda não é definitiva e está em processo de veri-

ficação de consistência e de potencialidades. As considerações de projeto que levaram a escolha desta alternativa como proposta de linguagem, assim como os detalhes de especificação, estão no trabalho de Tese presentemente em desenvolvimento.

3. A PROPOSTA

A linguagem CONSTRUCTOR segue o modelo previamente definido, possuindo, portanto, quatro tipos de módulos: CLIENTs, SERVERs, INTERFACE HARDWARE e INTERFACE SOFTWARE. As operações possíveis são definidas somente nos processos dos módulos tipo SERVER.

Processos podem se comunicar de duas maneiras diferentes: diretamente, por troca de mensagens ou indiretamente, lendo e escrevendo variáveis compartilhadas. Processos também podem se sincronizar de duas maneiras básicas: diretamente, por sinalização explícita, ou indiretamente, por teste e alteração de variáveis compartilhadas.

Os conceitos de SERVERs e operações permitem implementar com simplicidade qualquer dos modos de comunicação e sincronização acima descritos.

O conceito de módulos da linguagem CONSTRUCTOR é uma extensão da idéia de Wirth [4], onde os módulos isolavam partes do programa dependentes do tempo e de sincronização. No caso do CONSTRUCTOR, os módulos também isolam características funcionais e sintáticas diferentes. Isto concedeu maior clareza e concisão à linguagem, permitindo inclusive, a detecção de vários erros lógicos (inclusive de programação distribuída) em tempo de compilação.

Dentro dos módulos CLIENTs e SERVERs são definidos qualquer número de processos. Os módulos INTERFACE serão de dois tipos: Hardware e Software. O tipo Hardware é voltado principalmente para controle de hardware e de tempo real. O tipo SOFTWARE é voltado para a intercomunicação de programas em software.

Os módulos podem ser compilados separadamente e ligados durante a inicialização e carga. Neste caso apenas é necessário indicar quais as operações que fornecem para e

utilizam do ambiente em que farão parte, assim como os respectivos parâmetros.

3.1. Tipos e declarações

As declarações são similares as do Pascal [11] com algumas diferenças. Os tipos pré-definidos são booleano (BOLL), inteiro (INT), real (REAL), caracter (CHAR) e palavra (WORD). É possível ao usuário definir os seus próprios tipos de variável usando RANGE para definição de intervalos, ENUM para definição de escalares, REC para definição de registros, CAP para definição de "capability" e SET para definição de conjuntos. De uma forma geral tem-se:

```
TYPE <nome> = <def-tipo>;
onde <def-tipo> pode ser:
RANGE (<intervalo-de-escalar>)
ENUM <lista-de-escalar>
REC (<lista-dos-campos>)
CAP (<lista-de-operações>)
SET (<designação>), onde <designação> pode ser
      INT, CHAR, WORD ou ENUM.
```

Ao usuário é permitido declarar somente estes cinco tipos. Os tipos RANGE, ENUM e REC são idênticos aos do PASCAL [11] com a sintaxe do EDISON [9]. O que é diferente é o CAP, que é similar ao registro (REC) só que ao invés de campos, define uma lista de entradas as quais podem ser conectadas a operações em módulos por comandos de atribuição. Esta atribuição é válida quando os parâmetros são compatíveis. O tipo SET é o mesmo do PASCAL, porém com sintaxe diferente na declaração.

A declaração de variáveis introduz um ou mais identificadores e define os seus tipos, tal como o PASCAL. A diferença é a declaração de arranjos que é feita como no exemplo a seguir:

```
VAR linha [1..10]: CHAR  
declara um arranjo de 10 caracteres.
```

No caso de parâmetros de procedimentos e operações é possível deixar em aberto a especificação de intervalo do arranjo, especificando com um '*' como em arranjo `[*,*]:INT`. Internamente Arranjo é visto como Arranjo `[1:n, 1:m]:INT` com `n` e `m` fixados na chamada do procedimento.

A declaração de `PROCEDURE` e `FUNCTION` é idêntica ao `PASCAL` [11], com as mesmas regras, com a diferença de ser possível passar variáveis do tipo `RANGE` ou `INIT` como parâmetro e usá-las para declarar variáveis do corpo do procedimento ou função. Isto permite a expansão dinâmica de variáveis na pilha de um processo sem os problemas de administração de memória do `POINTER` em `PASCAL`.

As constantes são declaradas de maneira idêntica ao `PASCAL`. As declarações de tipos e constantes devem ocorrer no início do programa, de módulos ou de processos.

O tipo pré-definido `WORD` corresponde a uma palavra da máquina onde reside o sistema e é de tamanho equivalente a `INT` ou `BOOL`. Com este tipo é possível realizar diretamente operações lógicas ou de aritmética inteira sem sinal e executar código previamente carregado por meio de um interface.

As variáveis só podem ser declaradas dentro de `SERVERS` e/ou processos.

3.2. Mecanismo de sincronização

O conceito de SERVER é praticamente idêntico ao de RESOURCE [5,8] com algumas pequenas variantes nas construções sintáticas.

As operações e os comandos de ativação são primitivas e se assemelham ao conceito de primitivas de entrada e saída para sincronização de processos de Hoare [6], com a diferença que o processo receptor da mensagem não precisa saber a identidade de quem envia a mensagem. As operações são uma generalização das "procedures" e de passagem de mensagem.

Estas operações são definidas por comandos SELECT os quais são baseados nos comandos guardados de Dijkstra [7], e são ativados por comandos SEND, para envio de mensagens, ou por comandos CALL, para chamada "procedures" onde o processo chamador fica suspenso até que conclua a operação chamada. Estas primitivas são as mesmas apresentadas por Andrews [5].

Os comandos de entrada declaram, sincronizam e se quenciam operações. Cada comando de operação tem a forma:

```
SELECT <operação> {ALSO <operação>} [ELSE <comando>]
      ENDSEL;

<operação> ::= <nome-de-operação> [<parâmetros-formais>]
      [WHEN <expressão-booleana>] [BY <expressão aritmética>] DO <lista-de comandos>.
```

A guarda da operação consiste no nome da operação e na expressão booleana. O nome da operação habilita a guarda se existe pelo menos uma invocação pendente. Se existem mais de uma invocação pendente para a qual a expressão booleana seja verdadeira, as invocações serão ordenadas pela expressão aritmética, de acordo com o menor valor (se a

opção BY é omitida, a ordenação é a de tempo de ativação).

O comando `SELECT` tem a seguinte semântica [8]: se pelo menos uma guarda de operação é verdadeira, uma é selecionada não-deterministicamente; se existe mais de uma invocação da operação selecionada que satisfaça a expressão de sincronização, aquela que minimiza a expressão de seqüencialização é escolhida; se nenhuma guarda é verdadeira e `ELSE` é presente, então este é selecionado; se nenhum comando de operação pode ser selecionado, o comando `SELECT` é retardado até que uma seleção seja possível. (Um comando `SELECT` com uma guarda `ELSE` nunca retarda). Uma vez que a seleção é feita, os correspondentes comandos são executados com os parâmetros reais da invocação escolhida. O comando `SELECT` termina assim que terminam os comandos selecionados para execução.

3.3. Comandos

A linguagem `CONSTRUCTOR` contém quatro tipos de comandos: seqüencial (nulo, aborto de programa, atribuição e chamada de procedure); alternância, iteração e comandos para a invocação e execução de operações.

Os comandos seqüenciais são os seguintes:

nulo:	<code>SKIP</code>
aborta programa:	<code>ABORT</code>
atribuição:	<code><lista-de-variáveis>:=<lista-de-expressões></code>
alternância:	<code>IF <expressão-booleana> DO <comando></code> <code>{[ALSO <expressão-booleana> DO <comando>]}</code> <code>ENDIF</code>
iteração:	<code>WHILE <exp-bool> DO <comandos></code> <code>{[ALSO <exp-bool> DO <comandos>]}</code> <code>ENDWHILE</code>
saída de iteração:	<code>OUT</code>
chamada de procedure:	<code><nome-de-proc>[<parâmetros-formais>]</code>

Nos comandos de alternância e iteração é possível usar, no fim, um comando guardado [ELSE <comandos>] que é por definição a negação dos outros comandos guardados.

Para a concorrência a comunicação temos os seguintes comandos:

```
operação:    SELECT <corpo da operação> ENDSEL
chamada:     CALL <designação-de-operação>[(parâmetros-reais)]
envio:       SEND <designação-de-operação>[(parâmetros-reais)]
concorrente: CONC <lista-de-chamadas-ou-envio-de-operações>
              ENDCONC
```

A designação de operação é o nome de uma operação definida por um SERVER ou um campo de uma variável tipo CAP. Os nomes de tais designações são subscritos se o processo, SERVER ou variável tipo CAP, foram declarados como um arrays.

O comando SELECT já foi descrito na secção anterior. O comando de chamada termina quando a operação designada foi executada e os resultados, quaisquer que sejam, retornam. O comando envio termina tão logo os parâmetros reais foram salvos para transmissão; conseqüentemente nenhum resultado retornará se houverem parâmetros resultantes. Um comando concorrente CONC contém um ou mais comandos de chamada ou envio de mensagens que são invocados em paralelo, pelo menos conceitualmente. Este comando termina quando todos os comandos da lista terminarem. O critério de passagem de parâmetros pode ser por valor (by-value), por referência VAR (by-reference) e por resultado RES (by-result). Uma exceção ocorre quando o parâmetro for por referência (VAR) e o meio de intercomunicação entre o processo que ativa o comando de comunicação e o processo que implementa a operação for fracamente conectado (não há compartilhamento de memória). Neste caso, o efeito será como se o parâmetro fosse por resultado (RES).

3.4. Processos

Um processo contém um conjunto de variáveis, procedimentos, funções e comandos. Tem a forma:

```
PROCESS <nome> ["["intervalo"]"] [AREA = <tamanho-area>]:  
[declaração de variáveis]  
[declaração de procedimentos e funções]  
lista de comandos  
ENDPROS <nome>;
```

Um processo executa um comando por vez e termina quando a sua lista de comandos termina. As variáveis declaradas dentro de um processo são locais a ele. Quando estiverem em SERVERs os processos podem implementar operações. As operações são automaticamente disponíveis a outros processos em um mesmo SERVER. Os processos e os comandos SELECT permitem a programação de operações de exclusão mútua. Os processos são o único mecanismo para exclusão mútua.

A prioridade dos processos na criação é, por definição, mínima, podendo ser obtida ou alterada por meio da variável PRIORITY associada a cada processo.

Se um intervalo é especificado no cabeçalho de uma declaração de processo, uma família de processos idênticos é criada, uma para cada valor do intervalo. Dentro de uma instância, a variável especial MYPROCESS contém a identidade do processo.

As variáveis de um processo são temporárias e existem enquanto existir o processo. É admitida recursividade dentro dos processos.

O termo AREA define a área de trabalho do processo (pilha), onde o mesmo deve colocar as variáveis dos seus procedimentos e funções.

Adicionalmente, é possível declarar um procedimento tipo biblioteca dentro de um processo. Desta forma, quando o procedimento for chamado, o respectivo código será procurado na biblioteca, carregado na pilha do processo e posto em execução. Esta também é uma elegante maneira de obter "overlay" sobre a área de dados do processo. A forma geral da declaração é a seguinte [9,10]:

```
LIB PROCEDURE <nome>[(<parâmetros formais>)]
```

3.5. Construção de ambientes

Em programação distribuída a especificação e construção de ambientes adquire especial importância [3]. Depois de especificados os módulos e interfaces, surge o problema de onde localizá-los e de como fazer a comunicação entre os mesmos, se direta (memória compartilhada) ou indireta (envio de mensagens). Para resolver este problema a linguagem deve permitir a descrição dos elementos de um sistema físico onde irá operar o algoritmo programado, e a descrição do tipo de interconexão que existe entre os mesmos. Ao definir a interconexão fica definida a topologia e o modo de comunicação entre os elementos. Finalmente, sobre os elementos são localizados os módulos.

No sistema CONSTRUCTOR foram considerados quatro tipos de ambientes físicos. O primeiro é o de uma rede interconectando vários hospedeiros, o segundo é o de um hospedeiro contendo mais de um processador, que podem compartilhar uma memória, o terceiro é o de um hospedeiro com um processador, e o quarto é uma biblioteca de armazenamento de programas (módulos ou procedures), para futura e ligação e uso em outros ambientes.

A estrutura geral para construção dos ambientes é a seguinte:

A. Para redes (ambientes de processamento distribuído)

```
CONSTRUCT <nome-arquivo> = NET[(<endereço-de-nodo>)]  
ELEMENT {<nome-N> = NODE (<ender>);} }  
CONNECT {(<nome-N>, <nome-N>);} }  
LOCALIZE {(<nome-módulo>, <nome-N>);} }
```

B. Para Hospedeiros com mais de um processador (para ambientes de processamento paralelo):

```
CONSTRUCT <nome-arquivo> = HOST[(<endereço-de-nodo>)]  
ELEMENT {<nome-P> = PROCESSOR (<ender>);} }  
        {<nome-M> = SHARED MEM (<ender>);} }  
CONNECT {(<nome-P>, <nome-P>);} }  
        {(<nome-M>, <nome-P>);} }  
LOCALIZE {(<nome-modpross>, <nome-P>)} }  
        {(<nome-modpross>, <nome-M>)} }
```

C. Para hospedeiro com um processador (para ambientes de processamento concorrente).

```
CONSTRUCT <nome-arquivo> = HOST [, (<endereço-de-nodo>)]
```

D. Para biblioteca:
para procedimentos:

```
CONSTRUCT <nome-arquivo> = LIBP;  
<declaração-de-tipos-e-constantes>  
PROCEDURE <definição-formal>  
<corpo-do-procedimento>
```

ou para módulos:

```
CONSTRUCT <nome-arquivo> = LIBM;  
<declaração-de-tipos-e-constantes>  
<declaração-de-módulos>
```

Nos casos A e B todos os elementos do ambiente no nível da declaração CONSTRUCT devem estar localizados pela declaração LOCALIZE, não importando se foi declarado CONSTRUCT dentro de um módulo ou em todo o ambiente. Já o mesmo não ocorre com as opções C e D.

A unidade léxica CONSTRUCT é o símbolo inicial da gramática da linguagem. O termo <endereço-de-nodo> é o endereço de um nodo dentro de uma rede; quando é emitido, é assumido como sendo nodo onde é carregado o programa. No caso de uma rede, o termo <endereço-de-nodo> pode ser também o ponto de conexão entre duas redes.

O termo <ender> pode ser endereço de nodo, número de processador ou endereço de início de memória. O termo <nome-módulo> pode ser nome de INTERFACE, SERVER ou CLIENT. O termo <nome-modpross> pode ser nome de módulos ou de processos.

No caso de biblioteca, é construído um arquivo em condições de ser ligado em tempo de carga de programa (LIBM) ou durante a execução de programa (LIBP).

A declaração CONSTRUCT só pode ocorrer no início do programa ou no início de módulo CLIENT ou SERVER.

Deste modo, estes módulos podem conter expansões do ambiente em que estão contidos (por exemplo: sub-redes ou um multiprocessador).

3.6. Servers

Conforme o modelo, os SERVERS definem um conjunto de operações e encapsulam um conjunto de dados permanentes sobre os quais atuam as operações. A estrutura básica é similar e dos Recursos (RESOURCE [5,8]) com algumas variantes adicionais. Um SERVER contém no mínimo um processo e,

opcionalmente, variáveis permanentes, ou procedimentos compartilhados, ou outros SERVERs e CLIENTs. Os processos, as variáveis permanentes e os procedimentos compartilhados do SERVER são considerados fortemente conectados e as variáveis permanentes possuem uma granularidade de uma palavra, de modo que o acesso a um elemento é uma transação atômica, mutuamente exclusiva. Os processos de um SERVER só podem acessar as variáveis permanentes do seu próprio módulo. Os SERVERs também podem utilizar operações definidas por outros módulos no mesmo nível, ou do módulo imediatamente superior hierarquicamente, ou dos módulos imediatamente inferiores.

Os comandos de inicialização de um SERVER, se existirem, devem ser comandos seqüenciais colocados no início do módulo que somente acessem as variáveis permanentes do módulo.

De um modo geral a forma sintática é a seguinte:

```
SERVER <nome-sv>["["<intervalo>"]"];
<declaração-de-variáveis-e-tipos>
(OP<nome-operação>[(<parâmetros-formais>)]["["
    <modo-de-invocação>"]"]})
DEFINE {<nome-operação>["["<modo-de-invocação>"]"]})
UTILIZE {<nome-operação>[(<parâmetros-formais>)]
    [IN <nome-MOD>][WHEN <exceção>]}

[SHARED
    <declaração-de-procedimentos-compartilhados>
ENDSHARED;]
[INICIALIZE
    <comandos-de-inicialização>
ENDINIC;]
[<construção-de-ambiente>]
{<declaração-de-processos>}
ENDSERVER;
```

No caso de ser um SERVER já compilado em arquivo então tem-se:

```

SERVER <nome-su>["["intervalo "]""]
[<declaração-de-tipos>]
{OP <decl-de-operação>}
DEFINE <lista-de-operações>
UTILIZE <lista-de-operações>
EXTERNAL (<nome-de-arquivo>);
ENDSERVER <nome-su>;

```

O termo <nome-MOD> pode ser um INTERFACE ou outro SERVER. O termo <modo-de-invocação> indica de que modo pode ser chamada uma operação, se é por comando CALL ou SEND, ou via interrupção de interface (INTR-similar à opção SEND). Se não é especificado, fica implícito que é ambos (CALL e SEND). Por exceção entende-se erros que ocorrem durante o tempo de execução, estes são pré-definidos e podem ser:

- NUMERROR: Quando o resultado de uma operação numérica pré-definida não cai dentro do intervalo implementado, ou for indefinido. Por ex.: Overflow, underflow, operação inválida, divide por zero.
- RESERROR: Quando uma restrição de intervalo ou de índice é violada, ou é feita uma tentativa de acesso de uma capability sem operação associada.
- EXECERROR: Quando houver tentativa de executar código com operações incompatíveis em um INTERFACE SOFTWARE ou de executar um procedimento tipo LIB que não esteja na biblioteca ou possua parâmetros incompatíveis.
- SELECERROR: Quando todas as alternativas de um comando de seleção sem a opção ELSE estão fechadas, ou é invocada uma operação em um processo já extinto.

ESPERROR: Quando o espaço dinâmico de memória de um processo excede o seu limite.

COMERROR: Quando surge problemas nas comunicações entre processos (operações). Ex: Time-out, parity, etc...

HARDERROR: Qualquer outro erro devido ao equipamento como por exemplo "power failure".

Deste modo, com a opção WHEN, as operações podem ser usadas como rotinas de tratamento de exceções. Nesta opção só podem ser usadas operações que estejam dentro do próprio SERVER.

Com a especificação do intervalo irão ocorrer várias instâncias do mesmo server, assim é possível saber qual a instância por meio da variável MYSERVER.

O termo <nome-de-arquivo> é uma indicação do arquivo onde deve ser procurado o respectivo módulo.

Por <construção-de-ambiente> entende-se não só a construção de ambiente conforme descrito no ítem 3.6, mas também a definição de outros módulos. Nesta construção, quando a declaração CONSTRUCT é omitida, é assumido um ambiente corrente no hospedeiro ou nodo onde foi localizado este módulo SERVER.

Na declaração UTILIZE, quando é omitida a opção IN, a operação é assumida estar dentro do próprio módulo.

A declaração SHARED indica que a seguir vem um conjunto de procedimentos que podem ser chamados por qualquer processo do módulo. Estes procedimentos são auto-contidos e não podem utilizar variáveis globais, somente os parâmetros e as variáveis internas. Com esta opção é possível implementar o compartilhamento de código de procedimento, pois as

variáveis estarão na pilha de cada processo.

3.7. CLIENTs

O CLIENT é um módulo composto de um conjunto de processos que realizam tarefas. Estes processos se comunicam entre si somente por intermédio de processos em SERVERS, por meio dos quais também acessam recursos do ambiente que compõem. Os CLIENTs não possuem dados permanentes, ou compartilhados. Os CLIENTs também não podem definir operações.

A forma geral do CLIENT é a seguinte:

```
CLIENT <nome-CL>["["intervalo"]"]
[<declaração-de-tipos-e constantes>]
UTILIZE {<nome-operação>[(<parâmetros-formais>)]
        IN <nome-SY> [WHEN <exceção>]}
<declaração-de-processos>
ENDCLIENT;
```

ou no caso de estar em biblioteca:

```
CLIENT <nome-CL>["["intervalo"]"]
UTILIZE {<nome-operação>[(<parâmetros-formais>)]
        IN <nome-SV> [WHEN <exceção>]}
EXTERNAL;
ENDCLIENT;
```

A descrição dos itens é idêntica à dos SERVERS. Quando existem várias instâncias de um mesmo CLIENT, então é possível saber qual é a instância pela variável MYCLIENT.

3.8. Interfaces

Conforme o modelo, um Interface é um módulo por meio do qual dois ambientes diferentes podem interagir. É composto de um protocolo, implementado pelos ambientes interagentes, uma formatação de dados e um conjunto de operações.

Na interação entre dois ambientes, via interface, pode haver dois tipos de relacionamento: mestre-escravo e de igualdade. Isto leva, na prática, a três tipos de interface, quanto ao relacionamento: interface mestre, interface escravo e interface simples. Na definição, por exemplo, um interface será mestre quando o ambiente que o contém possui uma relação de escravo com o mesmo, e vice-versa.

O conceito de INTERFACE permite a implementação de comunicação convencional, por meio de protocolos conforme a recomendação ISO [13], entre nodos de uma rede com software completamente diferente.

O interfaceamento com ambientes de hardware apresenta características bem peculiares e diferentes do interfaceamento com ambientes de software. Assim, para efeitos de clareza e concisão de linguagens, considerou-se adequado definir dois tipos de interface: INTERFACE HARDWARE e INTERFACE SOFTWARE.

3.8.1. Interface Hardware

Uma das características peculiares deste tipo de interface é que o mesmo encapsula construções gramaticais especiais para associação de variáveis e endereços reais de hardware. A outra característica é a declaração de procedimentos como co-rotinas para definir a maneira com que as variáveis especiais são acessadas. Com isto é definido o tipo de operação sobre o interface. Finalmente, outra caracte-

rística é a condição de exceção expressa por expressão booleana de variáveis do interface.

As co-rotinas podem ser executadas à maneira das operações SELECT ou como procedimento.

A forma geral do INTERFACE HARDWARE é a seguinte:

```
INTERFACE HARDWARE <nome-int>;
COUPLE <nome-SERVER>;
[<declaração de tipos e constantes>]
VAR {<nome>["["intervalo ""]:<tipo> AT <endereço
                                     físico>;}
ACCESS {<operação com parâmetros reais> WHEN <ex-
                                     pressão booleana>
[OPER <nome>[(<parâmetros-formais>);
              <corpo-de-procedimento>]
ENDHARD;
```

Onde <endereço físico> também pode considerar arquiteturas que tenham dois espaços de endereçamento: um para memória, e outro para portas de entrada e saída ou de interfaces físicos. Desse modo <endereço físico> pode ser:

para memória: <endereço octal>
 /<endereço hexadecimal>

ou para partes:

```
PORT <endereço octal>
PORT /<endereço hexadecimal>
```

O comando ACCESS implementa a interrupção do interface de hardware com a flexibilidade adicional dada pela expressão booleana. Na chamada de operação do SERVER podem ser colocadas quaisquer das variáveis internas como parâmetro real.

Os procedimentos declarados pelo interface podem ser referenciados pelo SERVER concatenando o nome do interface com o nome do procedimento, da mesma forma como a referência de um campo de registro em PASCAL. Assim pode ser chamado como qualquer outro procedimento dentro de um processo.

Cada interface hardware é acoplado a um SERVER, e este SERVER deve estar fortemente conectado com o interface.

3.8.2. Interface Software

A primeira diferença com relação ao INTERFACE HARDWARE é que este não interage apenas com um server do ambiente, mas sim com todo o ambiente que o contém. O INTERFACE SOFTWARE é composto de duas partes: uma que pode interagir com todo o ambiente e outra que interage apenas com um SERVER específico (para fins de monitoração, controle e depuração de programas).

Um programa pode ser carregado para a memória, ou carregar outros códigos de programa, por meio do INTERFACE SOFTWARE. Quando o interface é aquele pelo qual o ambiente referido é carregado, então o mesmo é denominado MASTER, e não declara nenhuma variável e nem possui a segunda parte, que é a de acoplamento de SERVER.

A declaração de variáveis é similar ao do INTERFACE HARDWARE, exceto que não considera o endereçamento de portas, e permite a superposição de variáveis, dependendo do endereço e do tamanho das declarações (redefinição de campos).

De uma forma geral o Interface pode ser:

```

INTERFACE SOFTWARE <nome>["["intervalo"]"]:
MODE {<nome-operação>["<nodo-de-invocação>"]}
[UTILIZA {<nome-operação>[{<parâmetros-formais>}]
      IN <nome-MOD>[WHEN <exceção>]:}]
FURNISH {<nome-operação>[{<parâmetros-formais>}]}]
<segunda-parte>
ENDSOFT;

```

Onde <segunda-parte> pode ser:

```

MASTER

```

ou

```

COUPLE <nome-server>;
[<declaração de tipos e constantes>]
VAR {<nome>["["intervalo"]"]:<tipo> AT <endereço>;}
ACCESS {<operação com parâmetros reais> WHEN <exp.
                                             bool>;}

[OPER <nome>[{<parâmetros-formais>}];
  <corpo-de-procedimento>]

```

Na primeira parte, é gerada uma espécie de capability (V.3.1) com informações adicionais que indicam se a operação é fornecida ou utilizada, e qual o modo de invocação. A responsabilidade da configuração das operações e parâmetros é do código que é carregado logo a seguir da capability do interface, nas variáveis da segunda parte. Se não houver compatibilidade, então ocorre uma exceção de execução (EXECERRO). O código carregado não necessita necessariamente utilizar todas as operações descritas no interface.

São implicitamente definidas as operações SUSPEND, ATIVATE, STOP, EXECUTE e RESTART dentro do INTERFACE SOFTWARE.

No caso de haver declaração de variável para carga de código, é recomendável que toda a área seja primeiro declarada como tipo WORD e depois haja outras declarações. No INTERFACE SOFTWARE a área de memória é ocupada, de forma

contígua, conforme a ordem das declarações, de modo que o código carregado tem condições de saber onde e o que está acessando.

O item OPER é o mesmo das co-routines do interface Hardware e pode ser executado pelo mesmo processador em que executa o SERVER ou pelo processador do interface.

Quando o interface é MASTER, ele é associado com o interface servo do outro ambiente, conforme a conexão na topologia definida em ambos os ambientes. No caso de haver mais de um interface possível de conexão em um mesmo ponto da topologia, então a ordem de codificação é assumida.

Conforme a configuração acima, é possível um SERVER ler um código gerado pelo compilador e distribuir as suas partes conforme a tabela inicial de descrição de topologia, carregando a memória dos vários interfaces software distribuídos em uma rede. O acoplamento entre dois ambientes é de responsabilidade do ambiente mestre que inicializa o interface.

Também é possível carregar código gerado por compiladores de outras linguagens e por em execução, pois o uso das operações do interface não é obrigatório.

4. EXEMPLOS E APLICAÇÕES

A seguir é mostrada a utilização da linguagem na solução de alguns problemas típicos, para dar uma idéia da sua potencialidade. Deve ser observado que os problemas abordados estão longe de explorar todas as potencialidades da linguagem.

Não serão abordados exemplos de aplicação em redes e de tempo real devido a extensão da descrição do problema, necessária para o entendimento do exemplo.

4.1. Construindo mecanismos de sincronização e sequencialização

Antes de entrar nos exemplos é importante ressaltar que a linguagem permite a obtenção do número de invocações pendentes em uma operação por meio da instrução INVOC (<nome-operação>).

A. Semáforos:

```
SELECT P WHEN SEM > 0 DO SEM:=SEM-1
      ALSO V DO SEM:=SEM+1
ENDSEL;
```

Assumindo que P e V sejam invocados pelo comando CALL, a operação acima implementa as operações P e V sobre o semáforo SEM.

Se quizéssemos dar prioridade para as operações P então teríamos:

```
SELECT P WHEN SEM > 0 DO SEM:=SEM-1
      ALSO V WHEN (SEM=0 OR INVOC(P)=0) DO SEM:=SEM+1
ENDSEL;
```


A operação V só será executada se, havendo uma invocação pendente para ela, nenhuma operação P pode ser executada (sem=0) ou não tiver nenhuma invocação pendente em P.

B. Monitores:

Como exemplo é apresentado o monitor PAGEBUFFER do sistema SOLO de Brinch Hansen [12]:

```
SERVER Monitor;
OP Read (VAR text[1..512]: CHAR; VAR eof: BOOL):
  Write (VAR text [1..512]:CHAR; eof:BOOL):
DEFINE Read {CALL}, write {CALL}
PROCESS Pagebuffer;
VAR buffer [1..512]:CHAR;
    full,last: BOOL;
WHILE TRUE DO
SELECT Read(text,eof) WHEN full
  DO text:=buffer;
    full:=FALSE
    eof:=last;
ALSO Write (text,eof) WHEN NOT full
  DO buffer:=text;
    full:=TRUE
    last:=eof;
  ENDSEL;
ENDWHILE;
ENDPROS;
ENDSERVER;
```

O monitor pode ser usado como "buffer", interfaceando um Cliente com um "driver" de entrada e saída, por exemplo.

C. Outros

Quando é necessário atender requisição conforme determinada ordem então:

```
SELECT Request (amount) BY amount
      DO SKIP
ENDSEL.
```

As requisições pendentes são ordenadas de acordo com o valor de seus parâmetros reais; quando um Request é selecionado, a invocação cujo valor amount for mínimo será executada. Este comando implementa um "shortest-job-next Scheduler", se "amount" representa o tempo máximo de execução ou um "scheduler" por ordem de chegada se amount é um valor de relógio. [5].

4.2. "Rendezvous" Cliente/Servidor

Suponha que existem vários clientes requerendo serviços de um ou vários processos que possuam recursos para fornecer os serviços requeridos [5]. A maneira pela qual ocorre o encontro ("rendezvous") entre estes dois grupos de processos pode ser programada. Quando um cliente necessita um serviço, chama GetService, passando o parâmetro apropriado e recebendo o número de "rendezvous" no retorno. Algum tempo depois o cliente chama Waitdone para esperar a conclusão do serviço requerido. O número de "rendezvous" é passado como parâmetro.

Solução:

```
SERVER Servo [1..N];
UTILIZE Gettask (task-parâmetros decl.; VAR rendezvous id: INT);
      Taskdone (task-parametros decl.; VAR rendezvous id: INT);
PROCESS Serviço;
  WHILE TRUE DO
    CALL Gettask(task-parameters, rendezvous id);
    faz a tarefa
    SEND TaskDone (result-parameters, rendezvous id);
  ENDWHILE;
ENDPROS Serviço;
ENDSERVER;
SERVER Rendezvous;
OP Getservice (Client-params; VAR rid1:INT),
  GetTask (VAR Server-params; VAR rid2:INT),
  TaskDone (results1;rid3:INT);
  WaitDone (VAR results2; rid4:INT);
DEFINE GetService {CALL},Wait Done{CALL},Gettask {CALL}
  TaskDone;
PROCESS StartService;
VAR rid:INT;
  rid:=0
  WHILE TRUE DO
    SELECT GetService (Client-params; rid1) DO
      SELECT Gettask (Server-params; rid2) DO
        server-params:=client-params;
        rid1:=rid;
        rid2:=rid;
        rid3:=rid+1;

      ENDSEL;
    ENDSEL;
  ENDWHILE;
ENDPROS Startservice;
```

```

PROCESS Completion;
  WHILE TRUE DO
    SELECT TaskDone (results1; rid3;INT) DO
      SELECT WaitDone (results2; rid4:INT) WHEN
        rid3=rid4 DO results1:=results2;

      ENDSEL;
    ENDSEL;
  ENDWHILE;
ENDPROS Completion;
ENDSERVER Rendezvous;

CLIENT tarefas [1..M];
UTILIZE GetServe (Var Client-params; rendezvous id: INT):
  WaitDone (Var results; rendezvousid: INT)

PROCESS usuário;
  - Procedimentos
    Get Service (client-params, rendezvousid);
  - Outros procedimentos
    WaitDone (results, rendezvousid);
  - Procedimento

ENDPROS Usuário;
ENDCLIENT;

```

No processo StartService irá ocorrer "rendezvous" tão logo haja invocação para GetService e Gettask, durante este encontro será dado um número de "rendezvous" para ambos os processos. O processo Completion sincroniza a completação do serviço e o retorno dos resultados, de acordo com o número de "rendezvous". Conforme programado, nesta solu-ção só é possível processar uma completação de cada vez, po-dendo assim retardar outros clientes que estejam esperando serviços já realizados. Isto pode ser contornado fazendo com que o processo completion armazene os números de "rendezvous" dos serviços já realizados, de modo que possa aceitar Wait-Done de qualquer tarefa completada.

4.3. "Readers" e "Writers"

Dois grupos de processos, leitores e escritores, compartilham um banco de dados. Para proteger a integridade do banco de dados, no máximo um escritor pode acessá-lo de cada vez, e nenhum leitor pode examiná-lo enquanto um escritor o está alterando; leitores, contudo, podem acessar o banco de dados concorrente. Nenhum leitor ou escritor deve ser postergado indefinidamente. A solução é descrita a seguir [5]:

```
SERVER ReadersWrites;
VAR database [1..N]: item;
OP read (VAR v:item; i:INT) [1..M] {CALL}
    write (v:item;i:INT) [1..M]
    startread {CALL}, endread {SEND}, startwrite {CALL}, endwrite {SEND};
DEFINE read [1..M] {CALL}, write [1..M];

PROCESS RW [1..M];
    WHILE TRUE DO
        SELECT read(v,i) DO
            CALL startread;
            v:=database [i];
            SEND endread;

        ALSO write (v,i) DO
            CALL startwrite;
            database [i]:=v;
            SEND endwrite;

        ENDSEL;
    ENDWHILE;
ENDPROS RW;

PROCESS allocator;
VAR state:INT; writelast: BOOL;
    state:=0; writelast:=FALSE;
    WHILE TRUE DO
        SELECT startread WHEN (state > 0) AND (write last OR
            INVOC(startwrite)=0) DO
            state:=state+1, writelast:=FALSE;
```

```

    ALSO endread DO state:=state-1;
    ALSO startwrite WHEN (state=0) AND (NOT writelast
        OR INVOC (startread)=0) DO
        state:=-1; writelast:=TRUE,

    ALSO endwrite DO state:=0;
ENDSEL;
ENDWHILE;
ENDPROS allocator;
ENDSERVER Readers Writers;

```

É assumido que existe uma instância J do processo RW para cada usuário do banco de dados. Para acessar o banco de dados, o usuário chama read [J] ou write [J]. O processo allocator força determinados restrições de acesso. A variável "state" indica como o banco de dados, está sendo acessado (state=0 - nenhum leitor ou escritor, state=-1 - um escritor, state > 0 - state leitores); A variável "writelast" indica se um escritor foi o último a acessar o banco de dados. As condições de sincronização das operações do "allocator" asseguram a exclusão requerida, permitem a leitura concorrente e previnem a inanição (postergação indefinida). Um importante atributo desta solução é que encapsula o banco de dados. Nesta solução a granularidade é todo o banco de dados, porém é simples obter a granularidade de um item por exemplo. Para isto basta passar o número do item como parâmetro nas operações do "allocator" e ao invés de ter uma variável "state", ter um arranjo "state" com um elemento correspondente a cada item do banco de dados.

4.4. Transações

No problema anterior foi considerado o caso de leitura ou escrita de um único registro do banco de dados. Contudo, em aplicações de bancos de dados, o processamento de uma transação pode requerer a leitura e escrita de vários registros. Para manter a integridade, é necessário sincroni

zar toda a transação. Primeiro são feitas requisições de leitura ou escrita, depois é feito o acesso ao banco de dados. O pedido de requisições pode ser substituído por uma política de bloqueio ("locking policy") conforme o tipo de acesso desejado.

Solução [5]:

```

SERVER transactions;
  VAR database [1..N]:ítem;
  OP reqread [1..M] {CALL}
    read (VAR v:ítem; i:INT) [1:M] {CALL}
    relread [1..M], endread [1..M], startread [1..M],
    startwrite [1..M], endwrite [1..M], reqwrite [1..M] {CALL}
    relwrite [1..M],
    write (v:ítem; i:INT) [1..M];
  DEFINE reqread [1..M] {CALL}, read [1..M] {CALL},
    relread [1..M], reqwrite [1..M] {CALL},
    write [1..M], relwrite [1..M];
  PROCESS R [1..M];
  VAR reading: BOOL;
  WHILE TRUE DO
    SELECT reqread DO
      CALL startread; reading:=TRUE;
    ENDSSEL;
    WHILE reading DO
      SELECT read (V,i) DO v:=database [i]
      ALSO relread DO SEND endread;
      reading:=FALSE;
    ENDSSEL;
  ENDWHILE;

  ENDPROS R;
  PROCESS W[1..M];
  VAR writing: BOOL;
  WHILE TRUE DO
    SELECT rewrite DO CALL startwrite;
    writing:=TRUE;

```

```

        ENDSEL;
    WHILE writing DO
        SELECT write (v,i) DO database [i]:=v
        ALSO relwrite DO SEND endwrite;
        writing:=FALSE
    ENDSEL;
ENDWHILE;
ENDWHILE;
ENDPROS W;
PROCESS allocator;
    (corpo como na seção 4.3)
ENDPROS allocator;
ENDSERVER Transactions;

```

Esta solução também encapsula o banco de dados, também ilustra a utilidade de aceitar entrada de operações em diferentes partes do corpo do processo. Neste caso assegura que o usuário da transação primeiro faça a requisição, e então acesse o banco de dados, e depois o libere. Também é ilustrado como a construção de um processo pode ser usado para especificar uma família de co-rotinas, uma para cada usuário da transação.

4.5. Escalonamento para otimização de acesso a disco

Seja o problema de utilizar eficientemente uma unidade de disco com braço móvel. Para resolver este problema é necessário reduzir o excesso de movimentos do cabeçote, isto significa que é necessário ordenar a execução de operações de entrada e saída em disco, quando houver mais de uma pendente. Supondo que um dos parâmetros seja c , o índice do cilindro a ser acessado, a solução a seguir seleciona a invocação de operação cujo c seja o mais próximo da posição corrente do cabeçote no disco.

Solução [5]:

```
SERVER disk;
VAR buffers, etc...
OP do-IO (C:INT; ...);
DEFINE do-IO;

PROCESS Driver;
VAR position: INT;
position:=1;

WHILE TRUE DO
    SELECT do-IO(C,...) BY ABS (C-position)
        Position:=C;
        começa I/O em disco;
        aguarda interrupção do interface;

    ENDSEL;
ENDWHILE;
ENDPROS Driver;
ENDSERVER disk;
```

Embora a solução acima seja utilizada, ela pode levar à postergação de um pedido de entrada e saída para uma posição longe da que está sendo freqüentemente usada.

4.6. "Driver" de console

A seguir é ilustrado o uso de interface de hardware para controle da console. As operações podem ser executadas concorrentemente porque são implementadas por diferentes processos. O exemplo foi tomado de Andrews [8], onde era usado um "Real Resource". A grande vantagem do Interface é a clareza da programação.

```
INTERFACE HARDWARE Intcons;
COUPLE Console;
VAR keyboard: BOOL AT 60;% Keyboard interrupt
                                register
    Printer: BOOL AT 64;% Printer interrupt
                                register
```

```

TKS[0..7]: BOOL AT 177560; % reg.status keyboard
TPS[0..7]: BOOL AT 177564; % reg.status printer
TKB: CHAR AT 177568; % Keyboard data register
TPB: CHAR AT 177566; % Printer data register

ACCESS K-Int (TKB) WHEN Keyboard;
ACCESS P-Int      WHEN Printer;
OPER Startk; TSK[6]:=true, ENDOP;
OPER Starup; TPS[6]:=true; ENDOP;
OPER Ack-k; TKS[6]:= false; ENDOP;
OPEF Ackp; TPS[6]: false; ENDOP;
OPER Put (C:CHAR);TPB:=C; ENDOP;

ENDHARD;

SERVER Console:
CONST N = 80; % tamanho do buffer
OP getC(RES ch: CHAR) {CALL};
OP putC(ch: CHAR),
OP K-Int (TKB:CHAR) {INTR},
OP P-Int {INTR};
DEFINE getc {CALL}, putc;
PROCESS Consin;
VAR inbuf [0..N-1]: CHAR % Keyboard input buffer
    Busy: BOOL;
    Size, inf, ins: INT;
    Size,inf,ins,busy:=0,0,0,false;
WHILE size < N AND NOT busy DO
    Intcons.startk; busy:=true;
ELSE SELECT getc(ch) WHEN size > 0 DO
    ch:=inbuf[inf];
    size,inf:=size-1,(inf+1)MOD N;
ALSO K-Int(TKB) DO % interrupt
    Intcons.Ack-k; busy:=false;
    Inbuf[ins]:=TKB;
    size,ins:=SIZE+1, (ins+1) MOD N
ENDSEL;

ENDWHILE;
ENDPROS Consin;

```

```

PROCESS Consout;
VAR outbuf[0..N-1]:CHAR
    Busy: Bool;
    Size, outf, outs: INT;
    Size, outf, outs, Busy:=0,0,0, false;
WHILE size > 0 AND NOT Busy DO
    Intcons.put(outbuf[outf]);
    Intcons.startup; busy:=true;
    Size,outf:=Size-1, (outf+1) MOD N;
ELSE SELECT putc(ch) WHEN Size < N DO
    outbuf [outs]:=ch;
    size,outs:=size+1, (outs+1) MOD N;
    ALSO p-int DO
        Intcons.Ackp; Busy:=false;
    ENDSEL;
ENDWHILE;
ENDPROS consout;
ENDSERVER Console;

```

4.7. Carga de código executável

Finalmente é apresentado um exemplo do uso do INTERFACE SOFTWARE, para a carga de programas, e de uso de primitivas do sistema operacional sob forma de operações, por parte do programa carregado. Desta forma o interface também engloba a função do PREFIX do Pascal Concorrente [12].

No ambiente do sistema operacional o interface seria definido da seguinte forma:

```
INTERFACE SOFTWARE Programa;
UTILIZE Read (VAR C:CHAR) {CALL};
      Write (C:CHAR) {CALL};
      etc.
VAR MEM[0..10000]:PAL AT 10200;
COUPLE Loader;
OPER Carga (C:CHAR; i:INT); MEM[i]:=C; ENDOP;
OPER Executa; EXEC (ADDRESS (MEM)); ENDOP;
OPER Para; STOP (MEM); ENDOP;
ENDINT;
```

No ambiente do programa carregado teríamos:

```
INTERFACE SOFTWARE Prefix;
MASTER;
DEFINE Read (VAR C:CHAR) {CALL},
      Write (C:CHAR) {CALL};
      etc...
ENDSOFT;
```

Como pode ser visto no exemplo, é bastante simples e clara a definição da interação entre programa usuário e o sistema operacional.

5. ALGUMAS CONSIDERAÇÕES COM RESPEITO À IMPLEMENTAÇÃO

De um modo geral, no projeto do sistema, houve a preocupação de procurar simplificar os problemas de implementação. Assim o tratamento de entrada e saída e de interface é tratado em alto nível, aliviando grandemente o projeto do kernel. Deste modo ao Kernel cabe preocupar-se com a manipulação de interrupções e excessões, do "scheduling" de processos e das operações, da execução da comunicação entre processos, comunicação entre ambientes diferentes, manutenção de tabela de endereços de rede disponíveis, e tabela de estado dos processos de um mesmo ambiente.

Quanto à gramática da linguagem, procurou-se facilitar a construção do compilador. Assim existe um símbolo diferente para a indicação de fim de cada tipo de comando ou módulo. Isto facilita bastante a recuperação de erros no compilador. Também procurou-se facilitar o reconhecimento de estruturas sintáticas (como na declaração de tipo, por exemplo), de modo que não fosse necessário obter mais de um símbolo para saber que tipo de estrutura se trata, durante o reconhecimento (gramática tipo LL [1]).

A linguagem CONSTRUCTOR é um pouco mais complexa que a EDISON [9], para ambientes de processamento paralelo. Para aquela linguagem foi possível construir um compilador, bem menor que o compilador do Pascal Concorrente [12], com 4 passes, ocupando pouca memória. Para o compilador do CONSTRUCTOR também são previstos 4 passes: análise léxica e geração de tabelas, análise sintática, análise semântica, geração e otimização de código.

Os comandos guardados de Dijkstra [7], por definição, são selecionados em uma ordem aleatória quando dentro de um outro comando. No caso da implementação dos comandos da linguagem CONSTRUCTOR, esta ordem será a de escrita.

Ainda é previsto o projeto de um sistema operacional básico que ofereça as funções mínimas necessárias para carga de programas distribuídos. Estas funções podem incluir: carga do programa distribuído na rede, manutenção de tabelas de informação sobre o estado de elementos da rede, ou de hospedeiros, operações de entrada e saída, sistema para manuseio de bibliotecas, indicação e/ou recuperação de erros não detectados pelos programas distribuídos carregados. Este sistema básico deve ter condições de conviver com outros sistemas operacionais distribuídos construídos para as várias aplicações específicas possíveis de serem desenvolvidas na rede. Deste modo o desenvolvimento das aplicações é facilitado por uma série de funções de Sistemas Distribuídos já embutidos no Sistema CONSTRUCTOR.

Na implementação do sistema CONSTRUCTOR também haverá uma preocupação com portabilidade. Visto que em uma rede é comum ter hospedeiros de arquiteturas diferentes, é necessário minimizar as dificuldades de implementação, neste sentido, do CONSTRUCTOR em uma máquina. Assim, existem duas possibilidades: a definição de uma máquina virtual (máquina C) para a qual bastaria implementar um interpretador e um kernel no código do hospedeiro, ou o projeto de um compilador "tipo código parcial" onde bastaria trocar a tabela do gerador de código para o código do hospedeiro e programar o kernel no código do hospedeiro. As duas possibilidades podem ser aplicadas concomitantemente ou alternativamente. Como alternativa, é preferido o projeto da máquina virtual, principalmente pelo êxito observado com a máquina Pascal Concorrente (MPC).

6. CONCLUSÃO

O sistema CONSTRUCTOR, em projeto atualmente, fornece ao usuário de suporte dois tipos de visão de ambiente: o ponto de vista físico e o ponto de vista lógico. Deste modo o programador de um sistema distribuído deve ter condições de:

- Especificar a distribuição das unidades do programa.
- Estabelecer processos, programando partes que podem ser executadas em paralelo.
- Desenvolver os mecanismos de comunicação entre componentes remotos do programa.
- Providenciar estratégias para o manuseio de exceções.

Estas características introduzem um novo estilo de programação, que é diferente das áreas existentes tais como escrever programas seqüenciais e concorrentes [3].

A organização da linguagem CONSTRUCTOR possibilita uma descrição clara e concisa de qualquer sistema distribuído, e permite inclusive a detecção de vários erros lógicos já durante a compilação. A elaboração do modelo de ambientes antes da definição da linguagem viabilizou o projeto de estruturas tais, que se tornou possível disciplinar a liberdade de uso das potencialidades da linguagem sem tirar a sua versatilidade, como ocorre com o Pascal Concorrente [12], por exemplo. Estas estruturas também introduziram uma nova filosofia de programação, bastante útil para resolver problemas complexos como os de programação distribuída.

7. AGRADECIMENTOS

Ao Prof. Simão Sirineo Toscani pela revisão dos textos originais e pelas sugestões úteis.

8. BIBLIOGRAFIA

- [1] ROCHOL, J., NAVAUX, Ph., KEGLEVICH DE BUZIN, P., BRANCO. "Delineamento de um Projeto de Rede Local em Barra na UFRGS". XV Congresso Nacional de Informática-SUCESU Rio de Janeiro-RJ, outubro 1982 (442-446).
- [2] Advanced Course on Distributed Systems Architecture and Implementation, München, Mar.4-13, 1980. Edited. Besling, Springer-Verlag, 1981.
- [3] TOBIASCH, R. & RAFFLER, H. "Configurating Software a Method for bridging over the gap between concurrent processing and distributing processing". Infocom 82, Nevada, Las Vegas-USA, March-April /1982. (44-48)
- [4] WIRTH, Niklaus. "Toward a discipline of Real-time programming". Communications of the ACM, Vol. 20, N.8, August 1977. (577-583).
- [5] ANDREWS, Gregory R. "Synchronizing Resources". ACM Trans. on Programming Languages and Systems, Vol. 3, N.4, October 1981. (405-430).
- [6] HOARE, C.A.R. "Communicating Sequential Processes". Communications of the ACM, Vol. 21, N.8, August 1978. (666-677)
- [7] DIJKSTRA, E.W. "Guarded Commands, Nondeterminacy and formal derivation of Programs". Communications of the ACM, Vol. 18, N.8, August 1975 . (453-457).
- [8] ANDREWS, Gregory R. "The distributed Programming Language SR-Mechanisms, Design and Implementation". Software-Practice and Experience, Vol. 12, No.8, August, 1982. (719-753).
- [9] HANSEN, P.B. "The Design of Edison" Software-Practice and Experience, Vol.11, No.4, April, 1981 (363-396).
- [10] HANSEN, P.B. "EDISON - A Multiprocessor Language". Software - Practice and Experience, Vol. 11, No.4, April, 1981. (325-361).
- [11] WIRTH, N. "The Programming Language Pascal". Acta Informatica, Vol. 1, Fasc. 1, 1971 (35-63).
- [12] HANSEN, P.B. "The Architecture of Concurrent Programs". Englewood Cliffs, N.J., Prentice-Hall, 1977.
- [13] Data Processing - Open Systems Interconnection - Basic Reference Model. Computer Networks 5(1981).81-118.

TÍTULO: UM MODELO PARA SISTEMAS DE MENSAGENS POR COMPUTADOR

AUTORA: LIANE MARGARIDA ROCKENBACH TAROUÇO

RESUMO:

Vários modelos de sistemas de mensagens por computador têm sido desenvolvidos com diferentes perspectivas e em fases. Neste trabalho é apresentada uma tentativa de integrar os diferentes modelos baseado em algumas ideias adquiridas pela experiência da autora, aliada às discussões sobre tema, desenvolvidas no âmbito do grupo IFIP W.G. 6.5 - International Computer Message Systems.

PALAVRAS-CHAVES:

Modelo ISO, Sistemas de Mensagens por Computador, Correio Eletrônico, Padronização, Aplicações, Redes de Computadores.

1. INTRODUÇÃO

Um Sistema de Mensagens Baseado em Computador (SMBC) permite a comunicação entre entidades (usualmente pessoas) usando computadores. Computadores servem tanto para mediar a real comunicação entre os sistemas quanto para prover aos usuários recursos para criar e ler as mensagens.

Há cerca de dez anos os Sistemas de Mensagens por Computador estão sendo desenvolvidos e utilizados, e mais recentemente, passaram a receber maior grau de atenção por constituírem a base para a automação de escritório. Um crescente número de organizações está utilizando um Sistema de Mensagens Baseado em Computador para uso interno, implementado sobre redes locais mas também estão sendo utilizados em Redes de Computadores envolvendo conexões de longa distância. O projeto e a complexidade dos sistemas varia muito refletindo diferentes perspectivas e ênfases. Contudo, mais cedo ou mais tarde, surgirá a necessidade de interconectar esses diferentes sistemas de mensagens e essa tarefa será muito dificultada, se não impossível, devido as diferenças entre eles. Por esse motivo diversas organizações internacionais têm estado trabalhando na busca de um modelo de referência a ser seguido por todos os projetistas e implementadores de Sistemas de Mensagens Baseados em Computador, com vistas a assegurar sua futura interconectabilidade. O NBS - National Bureau of Standards dos Estados Unidos, CCITT - Comitê Consultivo Internacional de Telefonia e Telegrafia, a IFIP - International Federation of Information Processing são algumas das organizações que estão desenvolvendo, com muito de trabalho, nesta área.

O presente trabalho é uma tentativa para estudar o resultado do trabalho de todas essas organizações e comentá-las a vista das idéias oriundas da experiência da autora com o projeto e utilização de sistemas de mensagens bem como oriundas das discussões ocorridas no trabalho do subgrupo WG 6.5 da IFIP International Computer Message System.

2. UM MODELO SIMPLES DE MENSAGENS POR COMPUTADOR

A fim de apresentar um pano de fundo para discutir os diferentes propósitos dos órgãos será inicialmente descrito um modelo simples funcional para o Sistema de Mensagens por Computador. Esse modelo proporciona uma descrição dos recursos disponíveis para o usuário e da arquitetura para o sistema.

O Sistema de Mensagem por Computador provê a transferência de uma mensagem de um originador para um receptor. Uma mensagem é simplesmente a unidade de comunicação do originador para o receptor. O sistema de mensagem por computador oferece várias classes de funções a seus usuários:

- Criação das mensagens: As facilidades usadas pelo criador das mensagens para criar mensagens e especificar a quem elas devem ser enviadas.
- Transferência das mensagens: As facilidades usadas para levar as mensagens até seus receptores.
- Processamento de recepção: As facilidades usadas pelo receptor das mensagens para processá-las quando chegam.

Essas classes de funções são detalhadas diferentemente pelos órgãos padronizadores e serão comentadas posteriormente.

As características de um Sistema de Mensagem por Computador tornam-no diferente dos sistemas de comunicação ou de automação de escritório. Entre as causas destas diferenças pode-se destacar:

- Diferentemente de outros tipos de comunicação eletrônica, o Sistema de Mensagem por Computador envia mensagens a indivíduos em particular, e não a estações, ou a telefones. Se o receptor transfere-se para um local diferente, as mensagens enviadas a ele serão liberadas onde ele as chamar.
- A transmissão em Sistemas de Mensagens por Computador não é efetuada em tempo real, sendo denominada assíncrona, no sentido em que não há necessidade de sincronismo no acesso, isto é, o receptor e o originador não precisam combinar uma hora para estabelecer comunicação através do sistema. Quando a mensagem

deixa o originador o sistema receptor não precisa estar disponível pois os Sistemas de Mensagem por Computador incluem facilidades "Store-and-Forward".

- Um Sistema de Mensagem por Computador pode armazenar uma grande variedade de dados. Eles não são restritos a um só tipo de comunicação. As mensagens freqüentemente são simples memorandos mas não são restritas a isto podendo conter qualquer espécie de dados que o originador desejar enviar ao receptor.

- Sistema de Mensagens por Computador oferecem facilidades para criação da mensagem e isso é uma parte importante do sistema. Ele assiste o usuário na preparação das mensagens provendo facilidades para edição de texto e permitindo aos usuários, no momento da criação da mensagem, incluir na mensagem dados que estão armazenados em Bancos de Dados on-line. Alguns Sistemas de Mensagens por Computador também provêm interface para outras facilidades de automação de escritório tais como formatadores ou corretores de grafia.

- O Sistema de Mensagem por Computador também oferece facilidades de processamento no receptor como uma parte importante no sistema. Isto não acontece na maioria das demais formas de comunicação eletrônica. Por exemplo, Telex simplesmente imprime a mensagem no papel quando essas são recebidas, sem reter cópia. Processadores de palavra comunicantes podem notificar aos operadores se os documentos foram recebidos e estão armazenados on-line mas oferecem muito pouco mais do que isso. A maioria dos Sistemas de Mensagem por Computador oferecem ao menos as seguintes facilidades de processamento na recepção:

a. Habilidade de reter uma cópia de uma mensagem on-line depois que a mesma tiver sido lida.

b. Habilidade de examinar ou deletar uma a uma as mensagens armazenadas.

c. Habilidade de organizar as mensagens usando algumas forma de catalogação.

d. Habilidade de determinar se uma mensagem é recente no sistema ou ainda não foi lida.

e. Habilidade de resumir as mensagens armazenadas. O resumo geralmente inclui informação do tipo: se a mensagem é recente ou não lida, quando ela foi recebida, o seu comprimento, de onde ela vem, e sobre que trata.

f. Habilidade de recuperar uma mensagem armazenada em um ou mais de seus atributos (por exemplo, quando a mensagem foi recebida, se foi ou não lida ou deletada e os valores contidos em seus campos).

g. A facilidade de reencaminhamento permite aos usuários incluir algumas ou todas as partes de uma mensagem em uma nova mensagem sendo preparada.

h. A facilidade de resposta permite ao usuário responder a mensagem sem ter que informar uma nova relação de receptores.

3. O MODELO LÓGICO DE UM SISTEMA DE MENSAGEM POR COMPUTADOR

As facilidades num SMC para criação, transfêrencia e recepção de mensagens estão descritas num modelo lógico de SMC desenvolvimento pelo IFIP WG 6.5. Um modelo essencialmente idêntico está sendo usado pelo Grupo de Estudo VII do CCITT. O modelo consiste de um Sistema de Transferência de Mensagens (STM) e de uma série de Agentes de Usuário (AU) (Figura 1).

Um Agente de Usuário é uma entidade funcional que atua no atendimento de um usuário, assistindo-o na criação e processamento das mensagens e comunicando-o com o Sistema de Transferência de Mensagem.

O Sistema de Transferência de Mensagem é uma entidade que aceita uma mensagem de seu Agente Usuário originador e que a conduz para cada um dos Agentes Usuários receptores. O Sistema de Transferência de Mensagem pode efetuar funções de direcionamento e de armazenamento (entre outras) para atender a estes encargos.

O modelo divide as mensagens em duas partes: conteúdo e envoltório. O conteúdo da mensagem é a informação que o ori-

ginador deseja enviar ao receptor; o envoltório da mensagem consiste em todas as informações necessárias para que o Sistema de Transferência de Mensagem faça seu trabalho.

Nesse modelo funcional usando as facilidades de manipulação de mensagens um originador prepara a mensagem num Agente de Usuário que atua em seu auxílio. O agente do usuário através do interface com sistemas de transferência de mensagens, que é o mecanismo de encaminhamento e entrega, posta a mensagem colocando-a numa janela. Neste ponto o Sistema de Transferência de Mensagem assume a responsabilidade pela mensagem que é então reencaminhada internamente e finalmente liberada ao agente usuário receptor através de uma segunda janela. Neste ponto o Sistema de Transferência de Mensagens cede a responsabilidade pela mensagem ao Agente Usuário receptor que torna a mensagem disponível para o receptor. Os originadores e os receptores de mensagem são referidos como usuários.

A informação comunicada através de um Sistema de Transferência de Mensagens e sempre codificada digitalmente. Porém, embora exista essa exigência, o conteúdo da mensagem pode ser estruturado em um número ilimitado de modos. Mensagens de texto são comunicadas de maneiras muito diferentes, como por exemplo, dados não estruturados em cartas, dados estruturados em mensagens pessoais, documentos de intercâmbio e documentos trocados entre processadores de palavra ou máquinas de datilografia. O conteúdo das mensagens não está restrito a texto e pode incluir dados fac-símile, gráficos, etc. Além disso os usuários não são necessariamente seres humanos apenas. Processos eletrônicos e mecânicos podem também usar a facilidade de manipulação de mensagem para trocar dados do modo como quiserem ou necessitarem.

Somente aparece no envoltório a informação exigida para o encaminhamento e entrega da mensagem ou para acionar serviços especiais do Sistema de Transferência de Mensagem, por exemplo, entrega rápida. Todas as demais informações aparecem no conteúdo.

O Comitê dos Recursos para Usuário Final do CODASIL descreve um "end user", usuário final, como integrante de ao

menos uma das três principais categorias "de não especialistas em processamento de dados":

1. indireto;
2. intermediado;
3. direto.

Usuários finais indiretos ligam-se com os sistemas através de outras pessoas, usuários finais intermediados especificam exigências de informações comerciais para o sistema e os usuários finais diretos interagem por eles mesmos.

O Comitê CADASIL propõe uma abordagem baseada em formatos, formulários como sendo um interface mais natural entre o usuário final e os dados. O conceito e o modelo do usuário neste Comitê é baseado em objetos, por exemplo, formulários, pilhas, arquivos, localizações tais como escaninho de entrada e saída que contêm objetos e operações tais como movimentos dos objetos entre as localizações. "O usuário final neste contexto vê uma mensagem como um formulário com diferentes facetas dependendo das operações efetuadas sobre ele". O padrão proposto pelo NBS também visualiza uma mensagem como composta em campos com formatos e significados padrões.

4. MODELO DE UMA FACILIDADE DE MENSAGEM PARA USUÁRIO FINAL.

Cada usuário trabalha dentro de um ambiente organizacional em que ele se comunica com pessoas e acessa informação. Ele acessa seus recursos com facilidades de mensagens usando o terminal e com assistência de um agente de sessão interativa.

As capacidades de terminal de um usuário em conjunto com as facilidades do sistema para utilizá-lo determinam a forma de apresentação do sistema para o usuário, por exemplo, prolongando a tela ou apresentando formatos padrões de tamanho de página. O agente sessão interativa determina a forma de diálogo entre o usuário e o sistema para particular sessão, isto é, a linguagem de interação ou menu ou estilos orientados a comandos.

O usuário interage com uma facilidade final de mensagens para executar uma tarefa. A tarefa é subdividida em uma sequência de atividades que serão executadas durante a sessão interativa. O usuário executa algumas destas atividades diretamente dando instruções a seu agente de sessão interativa. Outras são executadas indiretamente, tais pontuações automáticas definidas pela semântica de um ambiente eletrônico e invocadas quando certas condições são satisfeitas. As operações automáticas podem ocorrer durante ou após a sessão interativa do usuário.

A medida em que o usuário executará bem ou mal uma particular tarefa usando este sistema de mensagem dependerá de: as capacidades de seu terminal; o estilo da sessão interativa de como ou quão apropriado isto é para o seu terminal; a semântica de seu ambiente eletrônico quão facilmente ele pode executar operações nele quão compatível isto é com os procedimentos organizacionais.

O usuário tipicamente trabalha através de seu agente de sessão interativa com ambiente eletrônico privado análogo ao seu próprio espaço no seu escritório. Em muitos sistemas de mensagens também tem acesso a ambientes compartilhados no âmbito da organização com itens tais como quadro de avisos, diretórios ou arquivos. Além disso ele ainda pode acessar informação compartilhada sobre ambiente organizacional e pode haver informação ou referência cruzada entre eles, por exemplo diretórios eletrônicos contendo referências a arquivos de correspondência mantidos em papel. O agente usuário pode ser visualizado como um assistente (mais do que como uma secretária eletrônica) com alguma independência delegada pelo usuário. Suas operações são iniciadas ou por uma explícita requisição do agente de sessão interativo, por exemplo enviar uma mensagem ou automaticamente em certas condições, por exemplo quando chega uma mensagem.

Para assistir o usuário são efetuadas as seguintes modalidades de processamento na recepção da mensagem

- Reter cópias;

- Examinar ou deletar as mensagens armazenadas, individualmente;
- Organizar as mensagens usando alguma forma de catalogação;
- Detectar mensagens recentes ou não lidas;
- Resumir as mensagens armazenadas recente, lida, quando chegou, comprimento, origem, assunto;
- Recuperar uma mensagem armazenada por seus atributos;
- Reencaminhar;
- Responder.

5. PADRONIZAÇÃO DOS SERVIÇOS PROVIDOS

Segundo a proposta do IFIP WG 6.5, os serviços providos ao usuário por um Sistema de Mensagem por Computador podem ser classificados como se segue:

5.1- Serviços Providos por ASM Cooperantes

- 5.1.1 - Pedido de endereço
- 5.1.2 - Informação sobre capacidade do receptor
- 5.1.3 - Audit Trail
- 5.1.4 - Tarifação
- 5.1.5 - Grupo fechado de usuários
- 5.1.6 - Indicação de conversão de conteúdo
- 5.1.7 - Indicação de incompatibilidade de tipo de conteúdo
- 5.1.8 - Cancelamento da entrega
- 5.1.9 - Notificação de entrega
- 5.1.10 - Lista de distribuição

- 5.1.11 - Criptografia
 - 5.1.12 - Entrega garantida
 - 5.1.13 - Entrega não prioritária
 - 5.1.14 - Multi-endereço
 - 5.1.15 - Nome/endereço/rota
 - 5.1.16 - Entrega normal
- 5.2 - Serviços Providos por AUs Cooperantes
- 5.2.1 - Conteúdo múltiplo
 - 5.2.2 - Referência cruzada
 - 5.2.3 - Circulação
 - 5.2.4 - Expiração (Data)
 - 5.2.5 - Intercepção de mensagem (ASM)
 - 5.2.6 - Reencaminhamento
 - 5.2.7 - Obsolescência
 - 5.2.8 - Designação de autor
 - 5.2.9 - Opções de apresentação
 - 5.2.10 - Notificação de recebimento
 - 5.2.11 - Geração de resposta
 - 5.2.12 - Data do usuário
- 5.3 - Serviços Providos pelo ASM Local
- 5.3.1 - Alarme
 - 5.3.2 - Autenticação
 - 5.3.3 - Informação sobre tarifas
 - 5.3.4 - Hora de entrega
 - 5.3.5 - Conversão de tipo de conteúdo
 - 5.3.6 - Recuperação histórica
 - 5.3.7 - Retenção para entrega posterior

5.4 - Serviços Providos pelo UA Local

5.4.1 - Edição

5.4.2 - Validação de campos

5.4.3 - Gerência de arquivos

5.4.4 - Programa Help

5.4.5 - Recebimento normal

5.4.6 - Recuperação

5.4.7 - Numeração, avisos

5.4.8 - Entrada do texto

5.4.9 - Leitura destrutiva

5.4.10 - Denominação de Mensagens

6. CAMPOS DAS MENSAGENS

Até o momento apenas o NBS apresentou uma proposta especificando o que deveria haver nas mensagens (campos).

Os campos das mensagens são de caracter obrigatória (tal como originador) ou opcional (tal como com cópias). Estes campos podem ser organizados nas seis categorias seguintes:

ORIGINADOR	DE RESPONDER PARA AUTOR EMISSOR
RECEPTOR	PARA CÓPIAS NÃO INFORMADAS PARA COM CÓPIAS PRÓXIMO A CIRCULAR CIRCULAR PARA

DATA	DE POSTAGEM LIMITE DE RECEPÇÃO DE INÍCIO DE ALERTA
REFERÊNCIAS CRUZADAS	EM RESPOSTA A IDENTIFICADOR DE MENSAGEM OBSOLETA NÚMERO SERIAL DO ORIGINADOR REFERÊNCIAS
MANIPULAÇÃO DE MENSAGENS	PRIORIDADE (Normal, Urgente...) CLASSE (Memo, Consulta a BD) TIPO DE REENCAMINHAMENTO RECEBIDA DE
CONTEÚDO	ASSUNTO TEXTO ADENDOS COMENTÁRIOS PALAVRAS-CHAVES

7. RELACIONAMENTO DO SISTEMA DE TRANSFERÊNCIA DE MENSAGENS AO MODELO DE REFERÊNCIA OSI-OPEN SYSTEM INTERCONNECTION.

O sub-comitê TC97/SC16 da ISO - International Standards Organization desenvolveu um modelo de referência para descrever a interconexão entre sistemas de computação integrantes de uma rede de computadores. Este modelo é conhecido como o modelo de referência OSI - Open System Interconnection. Ele organiza as funções e serviços essenciais para a comunicação entre os sistemas integrantes de uma rede de computadores. Para reduzir a complexidade do projeto desta infraestrutura, as funções e serviços necessários são sub-divididos em sete níveis. Cada nível provê

um certo conjunto de serviços e utiliza os serviços providos pelo nível inferior a ele. O nível mais baixo, conhecido como nível físico provê a transmissão dos bits no canal de comunicação. O nível 2 é responsável por garantir uma transmissão livre de erros e é denominado nível de linha. O nível 3, é o nível de rede e controla as operações da sub-rede de comunicações como um todo (direcionamento, interface com os computadores que se comunicam através da sub-rede). Estes três primeiros níveis de serviços são providos pela concessionária de um serviço de comutação de pacotes, por exemplo.

Mas, para que ocorra a comunicação entre os computadores são necessários ainda outros níveis, e protocolos denominados "de alto nível" que devem ser providos no próprio computador fonte e destino da comunicação. Estes níveis são: transporte, sessão, apresentação e aplicação (figura 6).

Um serviço de comunicação baseada em computadores que se comunicam é um serviço classificado como de nível 7 (nível de aplicação, no modelo ISO). Para a sua consecussão devem estar disponíveis alguns serviços do nível de sessão, que serão comentados a seguir. Os serviços relativos ao nível de apresentação estão englobados no próprio Serviço de Transferência de Mensagens.

Os serviços de nível de sessão necessários para garantir a transferência completa e confiável das mensagens são os seguintes:

- a- Estabelecimento da sessão
- b- Encerramento ordenado de sessão
- c- Interrupção iniciada pelo usuário
- d- Interrupção iniciada pelo provedor da comunicação
- e- Intercâmbio normal de dados
- f- Sincronização grossa
- g- Sincronização fina
- h- Ressincronização

Estes serviços são descritos à seguir:

Uma ou mais sessões podem ser estabelecidas entre dois Agentes de Transferência de Mensagem para a transferência das mensagens. A decisão sobre quando estabelecer as sessões é uma decisão local baseada no número de mensagens a transferir. Pode ocorrer a multiplexação de uma sessão ou diferentes sessões podem ser estabelecidas para transferir diferentes tipos de mensagens (cartas, fac-símile...). Um pedido de estabelecimento de uma sessão pode ser rejeitado devido a, por exemplo, congestionamento no Agente de Transferência de Mensagem Receptor.

O encerramento de uma sessão pode ocorrer logo que não houver mais mensagens a enviar. O estabelecimento e o encerramento de sessões é independente para cada direção de transferência.

O Agente de Transferência de Mensagem pode interromper uma sessão se são detectados problemas locais (por exemplo, problema na unidade de armazenamento em que estão sendo guardadas as mensagens).

Mas a sessão também pode ser interrompida se ocorrer uma falha no Serviço de Transporte ou outro elemento integrante do conjunto provedor da comunicação.

O intercâmbio normal de dados pode ser limitado por alguma forma de controle de fluxo para limitar a taxa de recebimento de mensagens em caso de congestionamento.

Quando um certo número de mensagens pode ser agrupado para envio, uma confirmação de recebimento pode ser enviada para indicar que a mensagem foi bem recebida e armazenada pelo Agente de Transferência de Mensagem Receptor. Esta sincronização grossa não indica que qualquer tipo de processamento de recepção tenha sido feito.

Por outro lado, uma mensagem longa (por exemplo, de fac-símile) pode ser sub-dividida em segmentos, e confirmados um a um (sincronização fina).

Após uma interrupção de qualquer tipo, o Agente de Transferência de Mensagem (ATM) que iniciou a sessão é responsável pelo estabelecimento de uma nova sessão. O ATM emissor usa então o serviço de ressincronização para estabelecer em que pontos de

sincronização grossa e fina deve iniciar a retransmissão. O ATM emissor deve ser capaz de retransmitir todos os dados após o último ponto de sincronização grossa ou fina.

A figura 6 mostra o relacionamento do sistema de Transferência de Mensagens com os demais níveis de serviços essenciais numa rede. O nível de sessão vai necessitar dos serviços de um nível de transporte que garanta a adequação da transmissão às classes de serviços disponíveis na rede.

8. CONCLUSÕES

A maneira como o modelo lógico para Sistemas de Mensagens por Computador foi apresentado oferece inúmeras vantagens. A liberdade para que o Agente de Usuário resida no equipamento do próprio usuário e possa ser por este desenvolvido propicia a implantação de interfaces mais humanas ou ergonômicas para o usuário. É num ambiente de automação de escritórios, onde certamente tais sistemas terão papel preponderante, ocorre uma grande variedade no grau de preparação do usuário. Usuários muito bem preparados e treinados no uso de sistemas computacionais, utilizarão os Sistemas de Mensagens por Computador, lado a lado, com usuários completamente leigos e despreparados. Mas, segundo o modelo ora proposto, desde que o Agente de Usuário intercomunique-se com o Sistema de Transferência de Mensagem, através de um interface padronizado com o Agente do Sistema de Mensagens, nada impede que este Agente de Usuário ofereça ao seu usuário a forma de comunicação mais adequada às suas necessidades e habilidades.

Por outro lado, a padronização no interface entre os AUs cooperantes, entre os ASMs cooperantes, entre o ASM e o AU a ele conectado viabilizará a interconectabilidade de diferentes sistemas projetados por equipes diferentes e implantados em máquinas diferentes. Desta forma, a criatividade de cada projetis-

ta poderá ser livremente exercida no projeto resguardando-se apenas os aspectos essenciais que garantirão a compatibilidade dos sistemas com os Serviços de Transferência de Mensagens públicos que porventura venham a existir no país e com os serviços propiciados pelas redes de computadores que proverão a comunicação de dados entre os ASM.

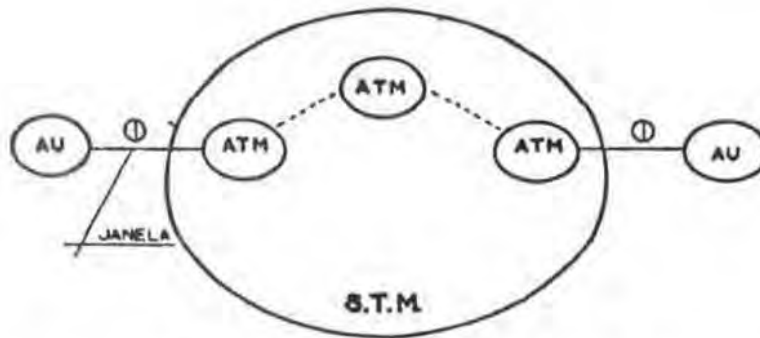
MODELO IFIP PARA SISTEMAS DE MENSAGEM POR COMPUTADOR

RESPONSABILIDADE DIVIDIDA ENTRE

AU - AGENTE DO USUÁRIO

ATM - AGENTE DE TRANSFERÊNCIA DE MENSAGEM

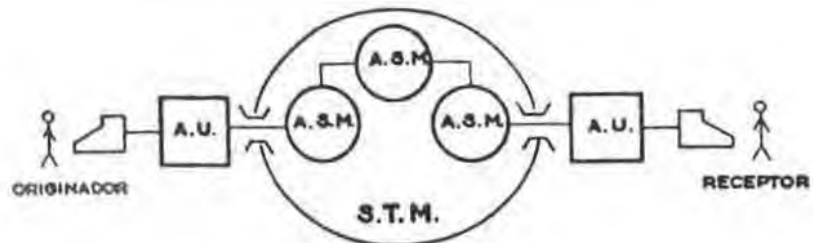
S.T.M. - SISTEMA DE TRANSFERÊNCIA DE MENSAGEM



TRANSFERÊNCIA DE RESPONSABILIDADE NA JANELA

Fig. 1

SISTEMA DE MENSAGEM POR COMPUTADOR

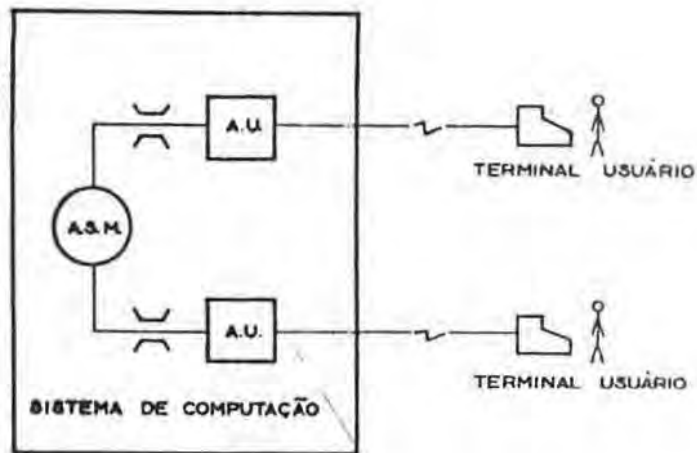


A.U. = AGENTE DO USUÁRIO

A.S.M. = AGENTE DO SISTEMA DE MENSAGEM

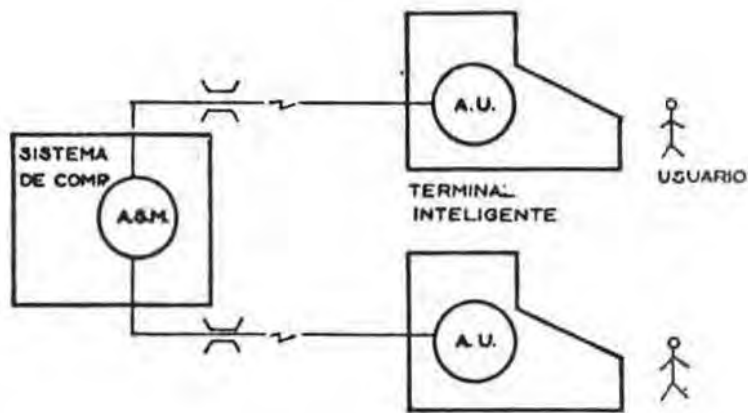
S.T.M. = SISTEMA DE TRANSFERÊNCIA DE MENSAGEM

Fig. 2



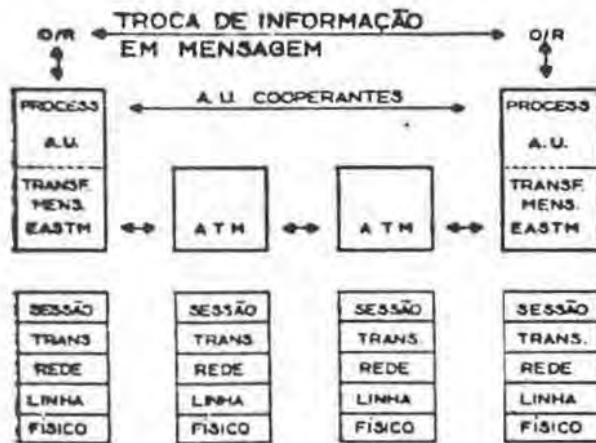
A.S.M. E A.U. CO-RESIDENTES

Fig. 3



A.U. STANDALONE

Fig. 4



OS PROTOCOLOS DE TRANSFERÊNCIA DE MENSAGENS
NO MODELO DE REFERÊNCIA DA ISO

EASTM - ENTIDADE DE ACESSO AO SISTEMA DE
TRANSFERÊNCIA DE MENSAGEM

ATM - AGENTE DE TRANSFERÊNCIA DE MENSAGEM

ISO: NATIONAL BUREAU OF STANDARDS

Fig. 5

ESTRUTURA DA MENSAGEM



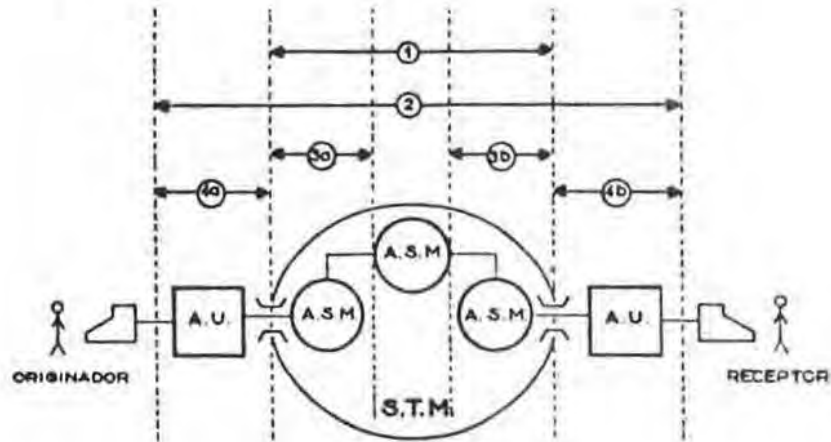
A UNIDADE DE TRANSFERÊNCIA DE INFORMAÇÃO
É A MENSAGEM

UMA MENSAGEM É NATURALMENTE DIVIDIDA EM
DUAS PARTES:

CONTEÚDO
ENVELOPE

Fig. 6

CLASSIFICAÇÃO DOS SERVIÇOS E FUNÇÕES



- 1 - A.S.M. COOPERANTES
- 2 - A.U. COOPERANTES
- 3a - A.S.M. LOCAL ORIGINADOR
- 3b - A.S.M. LOCAL RECEPTOR
- 4a - A.U. LOCAL ORIGINADOR
- 4b - A.U. LOCAL RECEPTOR

Fig. 7



Fig. 8

TÍTULO: Sistemas Operacionais para Redes de Computadores

AUTOR: José Antônio Monteiro de Queiróz

UFRGS/UFPE

RESUMO

Considerando a evolução dos sistemas de computação atuais nos defrontamos com um crescimento destacado das redes de computadores. Um grande número de trabalhos de pesquisa buscam uma estrutura básica para gerenciamento de redes, o Sistema Operacional para Redes de Computadores. Este trabalho visa fazer uma introdução ao estudo desses sistemas operacionais, apresentando a sua conceituação básica.

1. INTRODUÇÃO

O avanço no desenvolvimento tecnológico da indústria de computadores juntamente com o barateamento dos equipamentos tem contribuído para que, nos últimos anos, o campo das comunicações de dados tenha um crescimento significativo. As redes de computadores evoluíram desde os projetos experimentais até os sistemas operacionais muito sofisticados. Nesta década assistiremos uma, ainda mais, impressionante evolução desta área, com um grande número de redes de computadores se tornando operacionais.

As redes de computadores oferecerão aos seus usuários uma variedade crescente de serviços cada vez mais sofisticados, serviços estes que serão estendidos a uma comunidade ampla de usuários, em muitos casos, pouco sofisticados tecnicamente. Os recursos para que os serviços oferecidos sejam realizados ficam dispersos pela rede, aumentando a necessidade de esquemas de controle distribuídos. Estas tendências que são observadas levam à busca de uma estrutura básica para o gerenciamento de redes, o Sistema Operacional para Redes de Computadores. O sistema operacional fornecerá os mecanismos efetivos para que os usuários tenham facilidades que permitam um acesso eficiente aos serviços da rede.

Este trabalho visa apresentar a conceituação básica sobre Sistemas Operacionais para Redes de Computadores.

2. EVOLUÇÃO DAS REDES DE COMPUTADORES

2.1 Sistemas e Ambientes de Computação

Os sistemas de computação são usados para simular sistemas físicos e abstratos existentes, auxiliando os usuários na resolução de seus problemas. A simulação desses sistemas é feita através da programação de computadores.

Podemos observar até recentemente que um cenário tradicional para os ambientes de computação, de uma maneira geral, é constituído de instalações de computadores auto contidos. Estes sistemas são fechados e não demonstram uma preocupação maior com a possibilidade de ligação a outros sistemas de computadores.

O crescimento da importância dos computadores na nossa sociedade é um fato incontestável. A cada dia que passa mais facilidades de computação são colocadas à disposição de camadas cada vez maiores da sociedade. O custo das unidades de processamento vem decrescendo a grandes proporções com o avanço tecnológico da eletrônica, contribuindo assim para o desenvolvimento acentuado da computação.

Agora o cenário tradicional está se transformando em um outro. Uma empresa constituída por diversos departamentos terá distribuído por entre estes a sua capacidade de computação. Micro computadores, dispositivos inteligentes dotados de micro computadores, mini computadores e mesmo computadores de médio e/ou grande porte são usados para atender as diversas necessidades da empresa e se encontram espalhados pelos diversos departamentos que podem se localizar em um edifício ou conjunto de edifícios próximos, um campus ou seja uma área geográfica pequena.

Um sistema de processamento de informações é constituído de muitos recursos em sua maioria caros e subutilizados. Para diminuir os efeitos da subutilização e custo é necessário tornar os recursos compartilháveis e ampliar o número de usuários a utilizar o sistema. Os sistemas de tempo compartilhado, nos quais os recursos do computador estão disponíveis a um grande número de usuário, são exemplos de uma melhor utilização de recursos.

Antes os computadores seguiam duas rotas bem definidas para a sua evolução que eram a do aumento da velocidade de processamento e a do aumento da capacidade de armazenamento. Agora, além destas, uma outra rota está se avolumando e crescendo em importância na evolução dos sistemas de computação, a da utilização dos computadores em redes.

2.2. Redes de Computadores

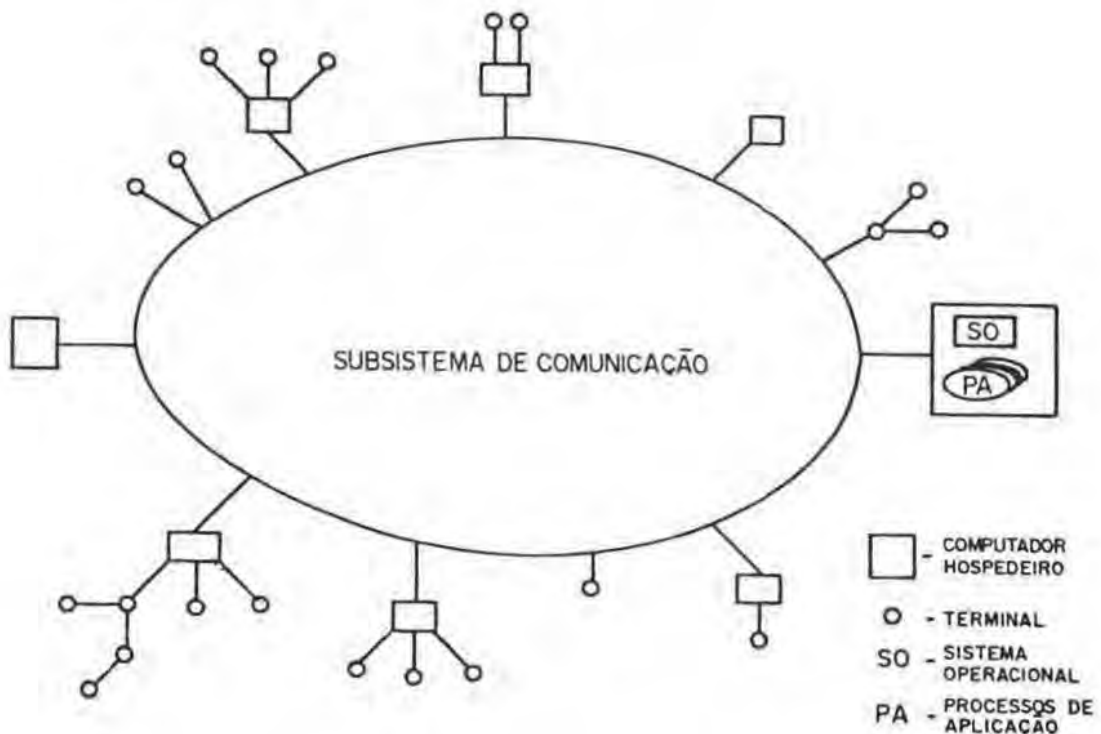


Figura 2.1 - Rede típica de computadores

Uma rede de computadores (Figura 2.1) é constituída por um conjunto de computadores que estão interconectados através de um subsistema de comunicação que fornece os mecanismos necessários para suportar a comunicação entre os diversos componentes da rede. Computadores componentes da rede, os hospedeiros, fornecem os serviços de processamento de informações.

Através das redes de computadores é possível distribuir o grande poder de computação que se obtém pela interconexão das máquinas, a uma comunidade ampla de usuários que pode compartilhar os recursos que se acham espalhados por toda a rede.

A comunicação de dados teve nos últimos anos um desenvolvimento acentuado e com isso uma grande variedade de redes de computadores foi produzida. Redes de comunicação se espalharam por grandes áreas geográficas, as redes de longa distância, ou se limitaram a pequenas áreas de abrangência, as redes locais. As redes locais se desenvolveram para atender a uma demanda crescente por uma alta taxa de transmissão de dados a um baixo custo de comunicação.

As redes de comunicação têm por objetivo fazer com que usuários, ou processos atuando em seu favor, tenham acesso a recursos que podem estar localizados em um outro computador da rede e não naquele ao qual está vinculado o usuário. As redes levaram ao desenvolvimento de protocolos que suportam o acesso a computadores remotos e a transferência de arquivos entre computadores. Apenas com isto não é possível obter um compartilhamento de recursos ou ainda uma distribuição de processamento eficiente já que se está utilizando apenas um pouco da grande potencialidade que estas arquiteturas oferecem. Esta potencialidade não pode ser atingida totalmente porque [WAT 80] e [FOR 78]:

- a) os computadores componentes da rede podem ter sistemas operacionais diferentes, sendo os seus recursos incompatíveis entre si, obrigando o usuário a conhecer, além dos mecanismos de acesso à rede, os sistemas operacionais dos computadores aos quais estão ligados os recursos que ele quer usar. O usuário também teria dificuldades para obter informações sobre a localização e utilização de recursos;
- b) o suporte para que novos recursos ou serviços sejam fácil e modularmente criados não é fornecido e com isto o programador que deseja criar ou utilizar um novo recurso terá que enfrentar todos os problemas gerados pela não modularização; e
- c) a contabilização, cobrança e outros procedimentos administrativos são, geralmente, mantidos por computador e com isto o usuário teria que manter com cada componente da rede uma conta separada.

Considerando isto, é fácil observar que uma rede de comunicação, mesmo nos oferecendo protocolos que permitam acesso por terminal a recursos remotos ou transferência de arquivos entre computadores, não pode ser vista como um sistema operacional integrado.

As redes atuais suportam facilidades para comunicação de dados e para processamento de informações, facilidades estas que permitem que arquivos, programas, dados, mensagens e transações sejam transferidos de um ponto a outro e sejam também processados. Estas redes podem ser melhor denominadas de Redes de Informações [POU 80]. As redes de informações são elementos da maior importância no desenvolvimento e implantação dos sistemas de informações distribuídas.

3. SISTEMAS OPERACIONAIS PARA REDES DE COMPUTADORES

3.1. Sistemas Operacionais - SO

As instalações modernas de computadores são compostas por uma variedade de dispositivos de hardware tais como terminais, consoles de operação, leitora de cartões, discos magnéticos, fitas magnéticas, impressoras, traçadores de gráficos, outros dispositivos de entrada e/ou saída e processadores. O sistema operacional deverá garantir o uso eficiente destes componentes e também que eles forneçam convenientemente os serviços aos seus usuários.

Um sistema operacional é formado por um conjunto de módulos de software. Estes módulos recebem as requisições dos usuários e devem escalonar os componentes do sistema para atender a estes pedidos. A tarefa básica executada pelo sistema operacional é a de multiplexar os recursos disponíveis em seu sistema. Através de técnicas variadas os reursos físicos disponíveis são transformados em recursos lôgicos que podem ser utilizados pelos processos ativos que estão em execução sob controle do sistema operacional.

3.2. Desenvolvimento dos SO para Redes de Computadores

Acompanhando a evolução das redes de computadores podemos observar que o seu desenvolvimento variou desde os projetos experimentais até os projetos de sistemas com mecanismos sofisticados de gerenciamento e controle. Nesses projetos podemos observar que uma tendência evidenciada é a do crescimento de serviços oferecidos aos usuários, também de vemos notar que aumenta a sofisticação destes serviços. Tudo isto contribui para que o interesse das pessoas em utilizar as redes aumente, fazendo com que a comunidade de usuários seja cada vez mais ampliada, gerando uma comunidade onde aumenta o número de membros pouco sofisticados tecnic

mente. Para que estas redes sejam viáveis é necessário dotá-las de uma estrutura que permita aos usuários facilidades para a utilização dos serviços oferecidos. Esta estrutura também deve ser dotada de mecanismos que conduzam a uma operação eficiente dos recursos, que se encontram dispersos, para que com isto se obtenha uma otimização nos custos de utilização. Estas estruturas básicas que estão em desenvolvimento e visam solucionar os problemas gerados pelo aumento de complexidade e também tornar econômica a utilização da rede, são os Sistemas Operacionais para Redes de Computadores.

Um sistema operacional para rede de computadores pode ser implementado através de uma das seguintes maneiras [FOR 78], [WAT 80] e [ROW 82]: ou diretamente sobre o hardware básico ou usando as funções básicas dos sistemas operacionais existentes como blocos de construção do sistema operacional para a rede. A implementação de um sistema operacional base pode nos dar funções básicas melhor adaptadas às necessidades da rede. A construção em cima de sistema operacional base existente nos dá a vantagem de contarmos com o grande investimento feito no sistema operacional base e no software de aplicação, já testados e conhecidos, minimizando o volume do suporte de software que deve ser desenvolvido. Uma desvantagem dessa construção é a de que aumenta a complexidade do sistema.

Muitos sistemas operacionais para redes de computadores surgiram nos últimos anos. Eles se desenvolveram experimentalmente para estudar os problemas da integração em redes, como o sistema RSEXEC (Resource Sharing EXECutive) [FOR 78], primeiro projeto orientado a redes, inicialmente voltado para um sistema homogêneo, e o sistema XNOS (eXperimental Network Operating System) [KIM 78] dirigido a uma rede de computadores heterogêneos de propósito geral. Sistemas operacionais surgiram para suportar aplicações específicas. O sistema NSW (National Software Works) [FOR 78]

foi projetado para suportar o desenvolvimento de software. Gunton [GUN 78] apresentou um sistema para suporte de aplicações de telecomunicações. O Squire [CHE 81] e o ACCENT [RAS 81] são sistemas orientados a comunicações. As arquiteturas de sistemas distribuídos Eden [LAZ 81] e CHORUS [GUI 82] com sistema operacional integrado, fornecem suporte para uma grande gama de aplicações. O Roscoe [SOL 79] foi desenvolvido para uma rede de micro computadores de propósito geral. O MICROS [TIL 80]b e [WIT 80] foi desenvolvido em Pascal Concorrente [BRI 77] para a rede MICRONET [TIL 80]a constituída de um computador multi-micro. O MIKE [LIU 82] e [TSA 83] foi projetado para uso em sistemas distribuídos em geral, e para uso na rede DDLCN em particular. Como o sistema operacional UNIX [RIT 78] assumiu uma posição de grande importância, com um número enorme de instalações em operação, muitos projetos têm se desenvolvido para integrar estes computadores [CHE 75] em redes de computadores [COL 82], [LUD 81] e [MAR 82]. Características como acesso transparente (ACCENT, e LOCUS [POP 81]), carga balanceada (ECN [HWA 82]), bancos de dados distribuídos [ROW 82], autonomia local (LOCUS) e tolerância a falhas (ACCENT) são encontradas nestes projetos de sistemas baseados ou compatíveis com o UNIX. Arquiteturas orientadas ao modelo ISO para 'Open System Interconnection' (OSI) [des 80] e [MOU 80] foram propostas [COA 81], [MAR 82] e [POU 80].

3.3. Conceitos de um SO para Redes de Computadores

Um sistema operacional para rede de computadores pode ser visto como uma extensão do conceito de sistema operacional convencional, ao qual se aplicam os problemas de comunicação gerados pela natureza distribuída dos recursos, pela heterogeneidade dos seus componentes e pela administração descentralizada.

Seria apropriado conceituar os sistemas operacionais para redes de computadores através do objetivo funcional desses sistemas [KIM 78] que é o do fornecimento de mecanismos de suporte, que simplifiquem o acesso aos serviços oferecidos, e de bases que facilitem a construção e subsequente incorporação de novos serviços à rede, sem a necessidade de alterar o código residente ou privilegiado, isto através de uma simplificação de interação entre sistemas e entre sistemas e usuários.

3.4. Componentes de um SO para Redes de Computadores

Os sistemas operacionais para redes de computadores são constituídos de dois componentes principais. O primeiro deles é um conjunto de processos de software ou módulos que se encargam das tarefas de gerenciamento e controle. O segundo componente é o conjunto de protocolos de alto nível que governam a comunicação e interação entre os módulos. Estes protocolos suportam a comunicação entre processos e se apoiam nos protocolos de baixo nível que fornecem o transporte de dados entre os nodos da rede.

3.5. Funções de um SO para Redes de Computadores

As funções do sistema operacional para uma rede de computadores se localizarão em dois grupos principais: funções de gerenciamento de serviços e funções de gerenciamento do sistema. As funções de gerenciamento de serviços são motivadas pelas necessidades dos usuários e nelas se enquadram a assistência na localização de recursos remotos, a alocação dos recursos, o controle de acesso aos recursos e funções para facilitar o compartilhamento de recursos. As funções de gerenciamento do sistema são motivadas pelas necessidades do próprio sistema e estão relacionadas às diretrizes operacionais para a rede. Nelas estão incluídas os re

latórios de status do sistema, as funções para gerenciamento de tabelas e bases de dados e as funções para execução de controle do sistema.

3.6. Funcionamento de um Sistema

Uma rede de computadores deve estar pronta para atender às necessidades de seus usuários. Serviços são oferecidos e recursos estão disponíveis para atender às requisições de serviços. O usuário que desejar a prestação de serviços da rede deverá fazer a sua requisição e, para que isto possa ser feito, é necessário, antes de mais nada, que o usuário peça o estabelecimento de uma sessão e se identifique perante a administração a qual ele está vinculado. Após a verificação dos direitos de utilização, a administração estabelecerá a sessão e providenciará a contabilização da utilização da rede. Tendo a requisição de serviço do usuário o sistema operacional, no papel de gerente de recursos da rede, se encarregará de localizar os recursos que podem ser locais ou remotos, e solicitar a alocação dos mesmos, podendo ser necessário para isto que um controle de direito de acesso seja feito. De posse dos recursos necessários, o serviço requisitado será atendido. Não é importante para o usuário saber onde está localizado o recurso que foi utilizado para o seu atendimento, se local ou remoto.



Figura 3.1 - Sistema de Computação

3.7. Um Modelo de SO para Redes de Computadores

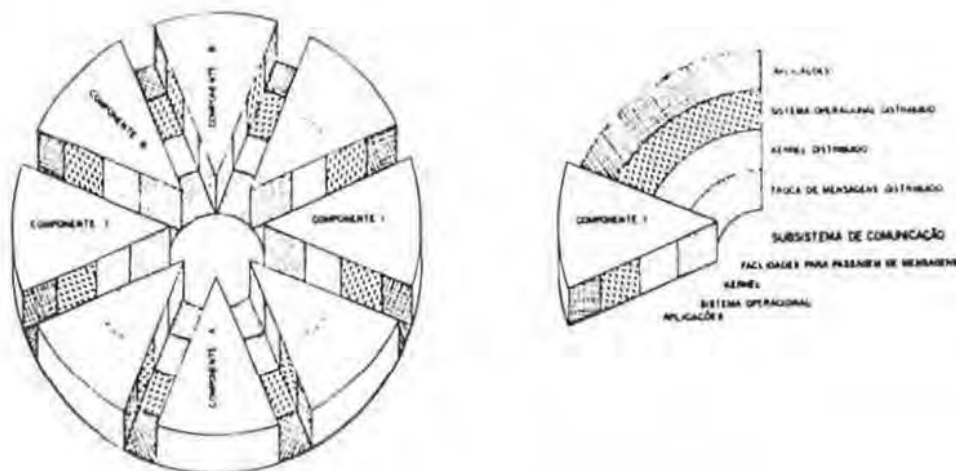


Figura 3.2 – Estrutura de um SO para Redes de Computadores

3.7.1. Estrutura do Modelo

Para a definição de uma arquitetura de rede é necessário fazer a escolha de um modelo de sistema operacional para redes. A estrutura escolhida é baseada no modelo de objeto ou recurso, de um sistema operacional [JON 78]. A criação de uma arquitetura em níveis requer que o sistema seja decomposto em níveis, que sejam especificados os serviços oferecidos por cada nível e que se especifique os serviços que cada um deles requer [des 80].

O acesso a objetos representa uma das metas principais dos projetos de sistemas operacionais para redes. Os objetos se classificam em objetos reais que são os processadores, as memórias secundárias, os dispositivos de E/S e outros, e em objetos abstratos, ou recursos, que são os processos, os arquivos, os diretórios, os dispositivos virtuais de E/S, as bases de dados e outros. Estes objetos representam blocos básicos de construção que são úteis na constru-

ção de outros objetos. A utilização de recursos apenas poderá ser feita através de funções ou operações bem definidas.

A implementação de um determinado tipo de objeto pode ser feita por um ou mais módulos chamados de servidores. Clientes são módulos encarregados das requisições, de acesso a recursos, para os servidores. Processos podem operar tanto no modo de servidor como no de cliente, em tempos distintos.

Os processos se comunicam através de objetos chamados mensagens. Uma mensagem é a menor unidade de dados que deve ser trocada entre dois processos para a realização de uma atividade. Mensagens podem ser de controle ou de dados. Mensagens de dados são aquelas que contêm informações para serem, simplesmente, armazenadas, impressas ou passadas, sem necessidade de interpretação.

Um processo cliente pode acessar um recurso através do envio de uma requisição para o servidor apropriado. O servidor se encarregará de atender à requisição acessando as estruturas locais a ele ou, se necessário, expedindo requisições adicionais a outros servidores para ajudá-lo na execução da tarefa solicitada. Quando a tarefa é considerada cumprida, é enviada uma resposta contendo resultados, se existirem, e indicação do sucesso ou não da operação.

O modelo escolhido [NES 81] é constituído por um conjunto de sistemas de computadores autônomos, interconectados através de um subsistema de comunicação (figura 3.2). Cada computador juntamente com o seu sistema operacional constitui um componente do sistema distribuído. Cada componente é formado por um kernel que suporta um conjunto de processos privilegiados, que compõem o sistema operacional, mais facilidades para passagem de mensagens, que em cooperação com o subsistema de comunicação formam o sistema distribuído de troca de mensagens. O kernel distribuído é constituído pelo consórcio de todos os kernels de componentes in-

terconectados através do sistema distribuído de troca de mensagens.

3.7.2. Níveis Componentes do Modelo

A composição do modelo, que está representando pela estrutura apresentada na figura 3.2, é feita pelos níveis de aplicação, de sistema operacional para redes, de kernel distribuído, de sistema distribuído de troca de mensagens, e pelo subsistema de comunicação.

No nível de aplicações estão localizados os processos que fornecem os serviços para o atendimento das aplicações. Estes serviços são necessários para suportar o usuário final (um usuário em seu terminal ou um processo). Os serviços oferecidos por este nível são dependentes de aplicações específicas e determinam os serviços a serem requeridos ao nível do sistema operacional. Neste nível os dados são informações com significado. Os problemas de estruturação da aplicação envolvem a distribuição de dados e de processamento, a organização dos processos de controle e comunicação e os mecanismos de suporte às organizações diferentes de dados e de processamento. Os problemas de estruturação da aplicação assim como os problemas de linguagens para sistemas distribuídos são inerentes ao nível de aplicações. Exemplos de serviços específicos de redes são: transferência eletrônica de fundos, base de dados distribuída, correio eletrônico, automação de escritórios, teleconferência e processamento distribuído em geral.

O nível de sistema operacional para redes é formado pela conjunção de todos os serviços fornecidos pelos diversos servidores, que se encontram dispersos pela rede. Neste nível são fornecidos serviços para uma grande variedade de aplicações e são implementadas políticas para alocação e compartilhamento de recursos básicos. Gerenciamento de pro

cessos, gerenciamento de informações, E/S virtual, comunicação, contabilização de utilização de recursos e clocks, são alguns dos serviços usuais fornecidos pelo sistema operacional, para atendimento das diversas aplicações. Os serviços que este nível necessita são requeridos ao kernel do componente. Isto é feito através dos processos mais primitivos do nível, ou seja, daqueles logicamente muito restritos e privilegiados que se comunicam com os dispositivos através de interrupções e comandos privilegiados, e que se comunicam com os objetos de um nível mais alto através das facilidades de comunicação entre processos.

O kernel distribuído é o nível seguinte do modelo. Ele é constituído pela cooperação de todos os kernels componentes, que se comunicam através do sistema distribuído de troca de mensagens. O nível de kernel distribuído do sistema operacional para redes fornece: os drivers que atuam como interfaces com os dispositivos de E/S, a base para criação de processos, arquivos, ou outros recursos primitivos e os mecanismos básicos para proteção, compartilhamento e segurança dos componentes do sistema. Os serviços fornecidos pelo kernel componente variam de acordo com o suporte de hardware, segurança e aplicações a serem suportadas. Um kernel componente deve possuir apenas o suporte necessário ao atendimento dos serviços fornecidos pelo componente.

Os serviços fornecidos pelo sistema distribuído de troca de mensagens são definidos pela necessidade de comunicação entre processos. Um serviço é definido através de recursos e de operações sobre estes recursos. Processos se comunicam por meio de mensagens enviadas através de canais, pares (origem, destino) de identificadores. Este nível deve suportar, além das facilidades para passagem de mensagens, facilidades para sincronização entre processos.

O subsistema de comunicação fornece a interface através da qual o componente do sistema distribuído tem aces

so às facilidades de comunicação da rede. Através dele são feitos o controle de erros, o controle de fluxo e é garantida a sequenciação de entrega de mensagens.

4. CONCLUSÕES

Os projetos de redes de computadores buscam alcançar metas que tornem ampla e fácil a utilização de sistemas distribuídos. Transparência, flexibilidade e facilidade para crescimento são metas a serem alcançadas.

Ao usuário deve ser dada a ilusão de operar um sistema composto de uma única máquina, sem distribuição de recursos (figura 3.1). Isto é conseguido através de níveis de abstração. Apesar de ter esta ilusão, o usuário deve poder especificar o local onde o processamento deve ser feito ou onde dados devem ser armazenados.

Um sistema operacional para rede de computadores pode ser entendido como uma extensão do conceito convencional de sistema operacional aplicado ao ambiente de redes. Com isto os conceitos desenvolvidos para sistemas centralizados podem ser aproveitados ao máximo, o que é vantajoso pois estes conceitos são bem definidos e entendidos.

Os sistemas que se obtêm da integração das redes de computadores com os sistemas operacionais próprios a elas, oferecerão ambientes simulados com um maior grau de correspondência com os sistemas físicos e abstratos existentes.

5. AGRADECIMENTOS

O meu agradecimento ao professor Simão Sirineo Toscani pelos comentários e pela atenção especial dedicada durante a elaboração deste trabalho, me ajudando a tornar mais claras as idéias apresentadas.

6. REFERÊNCIAS

- [BRI 77] BRINCH HANSEN, Per. The Architecture of Concurrent Programs. Englewood Cliffs, Prentice-Hall, 1977.
- [CHE 75] CHESSON, Gregory L. The Network UNIX System. In: SYMPOSIUM ON OPERATING SYSTEMS PRINCIPLES 5., Austin, Nov. 19-21, 1975. Proceedings. New York, ACM, 1975. p.60-6.
- [CHE 81] CHESLEY, Harry R. & HUNT, V. Bruce. Squire - A Communications-Oriented Operating System. Computer Networks, 5(5): 333-9, Sep. 1981.
- [COA 81] COATES, K. E., DVORAK, D. L. & WATTS, R. M. An Overview of BLN: A Bell Laboratories Computing Network. In: DATA COMMUNICATIONS SYMPOSIUM, 7., Mexico City, Oct. 27-29, 1981. Proceedings. New York, IEEE, 1981. p.224-9.
- [COL 82] COLLINSON, R. P. A. The Cambridge Ring and UNIX. Software-Practice and Experience, 12(6): 583-94, Jun. 1982.
- [des 80] desJARDINS, Richard & WHITE, George W. ISO/ANSI Reference Model of Open Systems Interconnection. In: COMPUTER NETWORK PROTOCOLS TRENDS & APPLICATIONS, Gaithersburg, May 29, 1980. Proceedings. New York, IEEE, 1980. p.47-53.
- [FOR 78] FORSDICK, Harry C., SCHANTZ, Richard E. & THOMAS, Robert H. Operating Systems for Computer Networks. Computer, 11(1):48-57, Jan. 1978.
- [GUI 82] GUILLEMONT, Marc. The CHORUS Distributed Operating System: Design and Implementation. In: INTERNATIONAL SYMPOSIUM ON LOCAL COMPUTER NETWORKS, Florence, Italy, Apr.19-21, 1982. Local Computer Networks. Amsterdam, North-Holland/IFIP, 1982. p.207-23.

- [GUN 78] GUNTON, Tony. Network Operating System for Telecommunications Applications. Computer Communications, 1(5):253-6, Oct. 1978.
- [HWA 82] HWANG, Kai et alii. A UNIX-Based Local Computer Network with Load Balancing. Computer, 15(4):55-66, Apr. 1982.
- [ISR 78] ISRAEL, Jay E., MITCHELL, James G. & STURGIS, Howard E. Separating Data from Function in a Distributed File System. In: INTERNATIONAL SYMPOSIUM ON OPERATING SYSTEMS THEORY AND PRACTICE, 2., Rocquencourt, France, Oct. 2-4, 1978. Operating Systems. Amsterdam, North-Holland, 1979. p.17-27.
- [JON 78] JONES, Anita K. The Object Model: A Conceptual Tool for Structuring Software. In: ADVANCED COURSE ON OPERATING SYSTEMS, München, Mar.29-Apr. 6, 1978. Berlin, Springer-Verlag, 1978. cap. 2A. p.8-16. (Lecture Notes in Computer Science, 60)
- [KIM 78] KIMBLETON, Stephen R., WOOD, Helen M. & FITZGERALD, M. L. Network Operating Systems - An Implementation Approach. In: NATIONAL COMPUTER CONFERENCE, 1978. Proceedings. Montvale, AFIPS, 1978. p.773-82.
- [LAU 78] LAUER, Hugh C. & NEEDHAM, Roger M. On the Duality of Operating System Structures. In: INTERNATIONAL SYMPOSIUM ON OPERATING SYSTEMS THEORY AND PRACTICE, 2., Rocquencourt, France, Oct. 2-4, 1978. Operating Systems. Amsterdam, North-Holland, 1979. p.371-84.
- [LAZ 81] LAZOWSKA, Edward D. et alii. The Architecture of the Eden System. In: SYMPOSIUM ON OPERATING SYSTEMS PRINCIPLES, 8., Pacific Grove, Dec. 14-16, 1981. Proceedings. New York, ACM, 1981. p.148-59.

- [LIU 82] LIU, Ming T., TSAY, Duen-Ping & LIAN, Richard C. Design of a Network Operating System for the Distributed Double-Loop Computer Network. In: INTERNATIONAL SYMPOSIUM ON LOCAL COMPUTER NETWORKS, Florence, Italy, Apr. 19-21, 1982. Local Computer Networks. Amsterdam, North-Holland/IFIP, 1982. p.225-47.
- [LUD 81] LUDERER, G. W. R. et alii. A Distributed UNIX System Based on a Virtual Circuit Switch. In: SYMPOSIUM ON OPERATING SYSTEMS PRINCIPLES, 8., Pacific Grove, Dec. 14-16, 1981. Proceedings. New York, ACM, 1981. p.160-8.
- [MAR 82] MARTIN, Marleen R. UNIX and Local Computer Networking. In: IEEE COMPUTER SOCIETY INTERNATIONAL CONFERENCE, 24., San Francisco, Feb. 22-25, 1982. High Technology in the Information Industry. New York, IEEE, 1980. p.318-22. (COMPCON Spring 80)
- [MOU 80] MOUTON, James. High Level Protocol Boundaries in the ISO Model. In: COMPUTER NETWORK PROTOCOLS TRENDS & APPLICATIONS, Gaithersburg, May 29, 1980. Proceedings. New York, IEEE, 1980. p.54-8.
- [NES 82] NESSET, D. M. Identifier Protection in a Distributed Operating System. Operating Systems Review, 16(1):26-31, Jan. 1982.
- [POP 81] POPEK, G., WALKER, B., CHOW, J. & EDWARDS, D. LOCUS: A Network Transparent, High Reliability Distributed System. In: SYMPOSIUM ON OPERATING SYSTEMS PRINCIPLES, 8., Pacific Grove, Dec. 14-16, 1981. Proceedings. New York, ACM, 1981. p.169-77.
- [POU 80] POULOS, John C. & BEAVERS, Alex N. Network Operating Systems: A Concept and Protocol. In: COMPUTER NETWORK PROTOCOLS TRENDS & APPLICATIONS, Gaithersburg, May 29, 1980. Proceedings. New York, IEEE, 1980. p.64-9.

- [RAS 81] RASHID, Richard F. & ROBERTSON, George G. ACCENT: A Communication Oriented Network Operating System Kernel. In: SYMPOSIUM ON OPERATING SYSTEMS PRINCIPLES, 8., Pacific Grove, Dec. 14-16, 1981. Proceedings. New York, ACM, 1981. p.64-75.
- [RIT 78] RITCHIE, Dennis M. & THOMPSON, Ken. The UNIX Time-Sharing System. The Bell System Technical Journal, 57(6):1905-29, Jul.-Aug. 1978.
- [ROW 82] ROWE, Lawrence A. & BIRMAN, Kenneth P. A Local Network Based on the UNIX Operating System. IEEE Transactions on Software Engineering, 8(2):137-46, Mar. 1982.
- [SOL 79] SOLOMON, Marvin H. & FINKEL, Raphael A. The Roscoe Distributed Operating System. In: SYMPOSIUM ON OPERATING SYSTEMS PRINCIPLES, 7., Pacific Grove, Dec. 10-12, 1979. Proceedings. New York, ACM, 1979. p.108-114.
- [TIL 80]a TILBORG, André M. van & WITTIE, Larry D. Packet Switching Using Concurrent PASCAL in a Network Computer. In: IEEE COMPUTER SOCIETY INTERNATIONAL CONFERENCE, 21., Washington, Sep. 23-25, 1980. Distributed Computing. New York, IEEE, 1980. p.358-65. (COMPCON Fall 80)
- [TIL 80]b TILBORG, André M. van & WITTIE, Larry D. A Concurrent PASCAL Operating System for a Network Computer. In: INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE, 4., Chicago, Oct. 27-31, 1980. Proceedings. New York, IEEE, 1980. p.757-63. (COMPSAC 80)
- [TSA 82] TSAY, Duen-Ping & LIU, Ming T. MIKE: A Network Operating System for the Distributed Double-Loop Computer Network. IEEE Transactions on Software Engineering, 9(2):143-54, Mar. 1983.

- [WAT 80]a WATSON, Richard W. & FLETCHER, John G. An Architecture for Support of Network Operating System Services. Computer Networks, 4(1)33-49, Feb. 1980.
- [WAT 80]b WATSON, Richard W. Distributed System Architecture Model. In: ADVANCED COURSE ON DISTRIBUTED SYSTEMS - ARCHITECTURE AND IMPLEMENTATION, München, Mar. 4-13, 1980. Berlin, Springer-Verlag, 1981. cap.2. p.10-43. (Lecture Notes in Computer Science, 105)
- [WIT 80] WITTIE, Larry D. & TILBORG, André M. van. MICROS, A Distributed Operating System for MICRONET, A Reconfigurable Network Computer. IEEE Transactions on Computers, 29(12):1133-44, Dec. 1980.

TÍTULO: Metodologia para Desenvolvimento de Software em Sistemas Distribuídos

AUTORES: Antonio Luiz Bogado Fernandes
Maurício Moszkowicz
Lucia Della Valle

ENTIDADE: CEPEL - Centro de Pesquisas de Energia Elétrica
Departamento de Eletrônica DPET)
Cidade Universitária - Ilha do Fundão
C. Postal 2754
21.941-Rio de Janeiro-RJ
Tel.: 270-0112 R. 143

RESUMO

O CEPEL vem desenvolvendo um sistema de supervisão para sistemas elétricos que incorpora características peculiares tal como arquitetura com multimicroprocessadores organizados segundo um barramento serial.

Tendo em vista esta restrição e a grande complexidade do software envolvido, apresentamos uma metodologia ao desenvolvimento do software que tem entre outros, os seguintes macro-objetivos:

- Redução do custo de desenvolvimento e manutenção do software;
- Tornar mais observável (aumentar a capacidade de administração) o desenvolvimento do projeto, criando um meio de comunicação fracamente conexo entre as pessoas que participam do desenvolvimento.

A metodologia divide o desenvolvimento do software em seis fase distintas:

- Análise dos requisitos: apresenta um método para a definição das propriedades gerais do sistema, verificando os contornos do projeto, analisando o contexto e as funções principais;
- Especificação: apresenta métodos de decomposição das funções em objetos, define as interfaces e descreve os efeitos de cada função;
- Projeto: define a alocação dos objetos no sistema, fornece um plano de implementação para os objetos, refina suas estruturas de controle descrevendo as suas entradas/saídas/efeitos.
- Implementação: consiste na elaboração dos algoritmos, rotinas e programas de uma forma geral;
- Teste: observa o comportamento do software implementado e examina a sua disponibilidade;
- Operação: é a entrega do produto final ao usuário, podendo envolver uma atualização/manutenção do sistema.

Como conclusão do trabalho serão apresentados os resultados obtidos pela aplicação da metodologia a partes do sistema de supervisão ora em desenvolvimento no CEPEL.

Palavra Chave: Engenharia de Software e Sistemas Distribuídos

1. INTRODUÇÃO

O CEPEL vem desenvolvendo um sistema de supervisão para sistemas elétricos para ser utilizado nos centros de despacho regional de concessionárias de energia elétrica. Os despachos regionais têm como encargo a coordenação das manobras, o controle de tensão e a normalização após perturbações de uma área pré-determinada do sistema elétrico de uma empresa.

Para fazer face às necessidades especificadas por FURNAS Centrais Elétricas S.A. para seus centros de supervisão regional o CEPEL propôs a esta empresa o desenvolvimento de um centro de arquitetura modular à base de microprocessadores. Esta solução permitiu um pleno domínio da tecnologia de projeto de centros de supervisão dentro da filosofia do processamento distribuído. As vantagens desta estratégia são óbvias:

- o usuário, no caso, FURNAS possui total conhecimento do sistema em todos os seus aspectos de hardware e software podendo portanto moldá-lo às suas necessidades;
- a estrutura modular, de grande flexibilidade, permite, quando necessária, a introdução de novas funções no sistema e assegura uma longa vida útil ao mesmo devido às facilidades de expansão.

O sistema protótipo está em fase final de implementação e já se tem uma versão em operação no centro regional de FURNAS de Jacarepaguá. A transferência da tecnologia para a indústria também já foi iniciada. A primeira das duas indústrias que fabricarão o sistema já iniciou o processo de absorção da tecnologia e a segunda será escolhida até o final de 1982.

A Figura 1 mostra o esquema de um centro de supervisão dentro da concepção modular proposta.

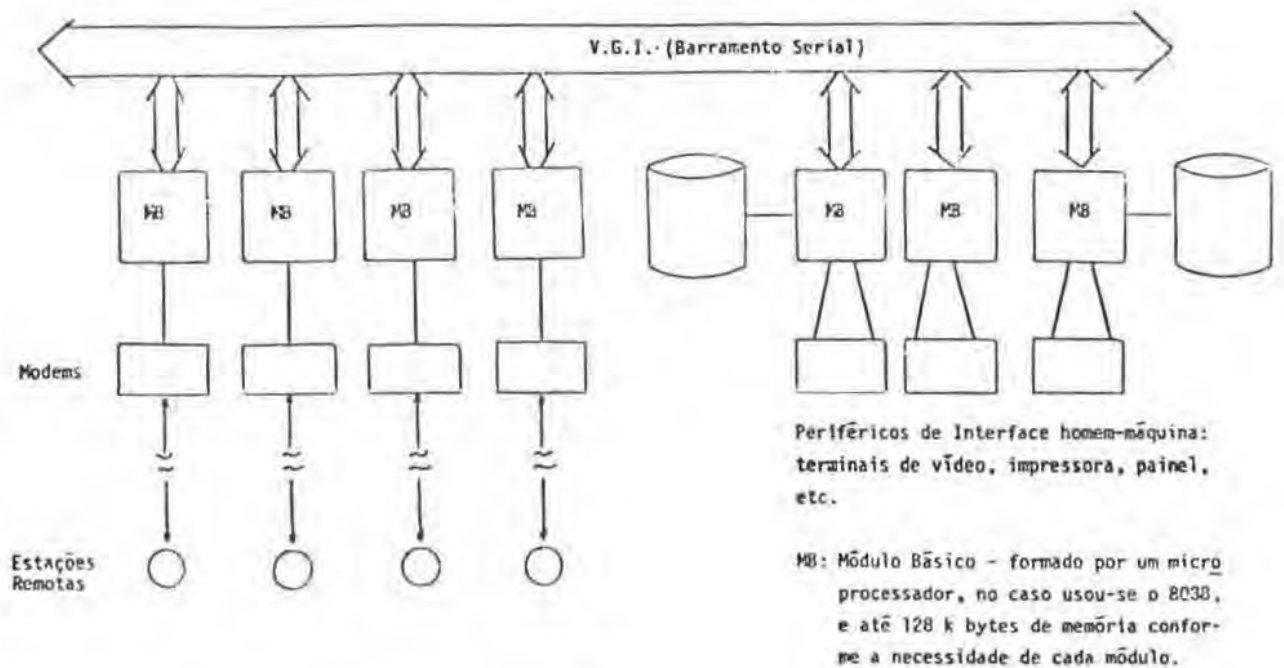


Figura 1

O projeto do sistema básico foi conduzido em paralelo com o do sistema de aplicação, o que permitiu o "overlapping" de etapas, facilitado pela arquitetura modular. Por outro lado, a constante interação entre os requisitos da aplicação e as capacitações alcançáveis pelo sistema básico permitiram definir-se um casamento ótimo que atendesse aos diversos condicionamentos econômicos e cronológicos.

A complexidade do projeto, que ao ser concluído terá durado cerca de 6 anos e envolvido recursos humanos da ordem de 72 engenheiros-mês e materiais da ordem de 1 milhão de dólares, merece toda uma sistemática de desenvolvimento. Esta sistemática que, a bem da verdade, está sendo consolidada "on the job" é hoje um dos principais sub-produtos da realização do trabalho. Nas páginas a seguir procura-se descrever os aspectos mais relevantes da metodologia utilizada.

2. METODOLOGIA GERAL

As motivações para a adoção de uma metodologia definida em projetos de engenharia de software são bem conhecidas: custos crescentes para desenvolvimento do software, idem para sua manutenção, a tradicional dificuldade de se obter boa documentação, problemas decorrentes da alta coesão normalmente encontrada entre os participantes do projeto, baixa visibilidade pela gerência do andamento das diversas frentes de projeto, etc.

Inúmeros métodos de trabalho vêm sendo propostos para fazer face a esses problemas. Os enfoques variam nos detalhes de implementação mas há uma certa unanimidade no tocante às fases básicas do desenvolvimento: estabelecimento dos requisitos, análise dos requisitos, busca de soluções, especificação da solução, projeto, implementação, testes e manutenção. Uma documentação paralela a todas estas etapas também é uma necessidade de reconhecimento geral.

O trabalho que vimos realizando está seguindo esta cronologia básica como não poderia deixar de ser. Sentiu-se entretanto, por diversas vezes, necessidade de melhor caracterizar as diversas etapas e, principalmente, ao tratarmos de um sistema que já tem uma estrutura frouxamente conexa (vide barramento serial comum na figura 1) tirar proveito da mesma para criar ferramentas de desenvolvimento que favorecessem ao máximo o gerenciamento das configurações do projeto.

As seções a seguir descrevem cada uma das etapas seguidas, dando para cada uma: a caracterização que nos pareceu a mais correta, exemplos das atividades do projeto do centro de supervisão realizadas dentro dela e as ferramentas criadas para o trabalho com uma arquitetura do tipo distribuída.

3. DEFINIÇÃO DOS REQUISITOS

O trabalho de definição dos requisitos foi conduzido basicamente por FURNAS, com a participação do CEPEL. Seu resultado consta de extenso documento designado CEP-170 e que rege os termos da concorrência aberta por FURNAS para escolha do fabricante de seus centros de supervisão regional.

Os requisitos estabelecem as diversas funções do software de aplicação, a configuração dos equipamentos periféricos, tempos de resposta, índices de desempenho, ocupação, confiabilidade, reparabilidade, etc.

Por carência de espaço não nos cabe aqui fazer esta descrição. Ao longo das seções seguintes serão feitas referências aos requisitos envolvidos nos exemplos abordados.

4. ANÁLISE DE REQUISITOS

Esta fase dos trabalhos consiste em um processo de submeter cada um dos requisitos de performance (oriundos do usuário) a uma espécie de crivo formado por um outro conjunto de requisitos, alguns também oriundos do usuário e outros da experiência da engenharia. A Figura 2 ilustra este procedimento.

As recomendações produzidas por este processo vão constituir o quadro de restrições a que a fase seguinte - a Busca da Solução - deverá obedecer ao procurar a solução ótima. Ou seja, para cada requisito funcional r_{fj} será obtido um conjunto de recomendações $\{r_{1,j} \dots r_{l,j}\}$ a partir de um mapeamento dos conjuntos $\{r_{f1} \dots r_{fp}\}$, $\{r_{o1} \dots r_{ok}\}$ e $\{r_{e1} \dots r_{em}\}$.

Vejam os agora como podem ser caracterizados os conjuntos acima, como o processo de crivagem se realiza e, usando a aplicação mencionada, mostrar um exemplo deste tipo de trabalho.

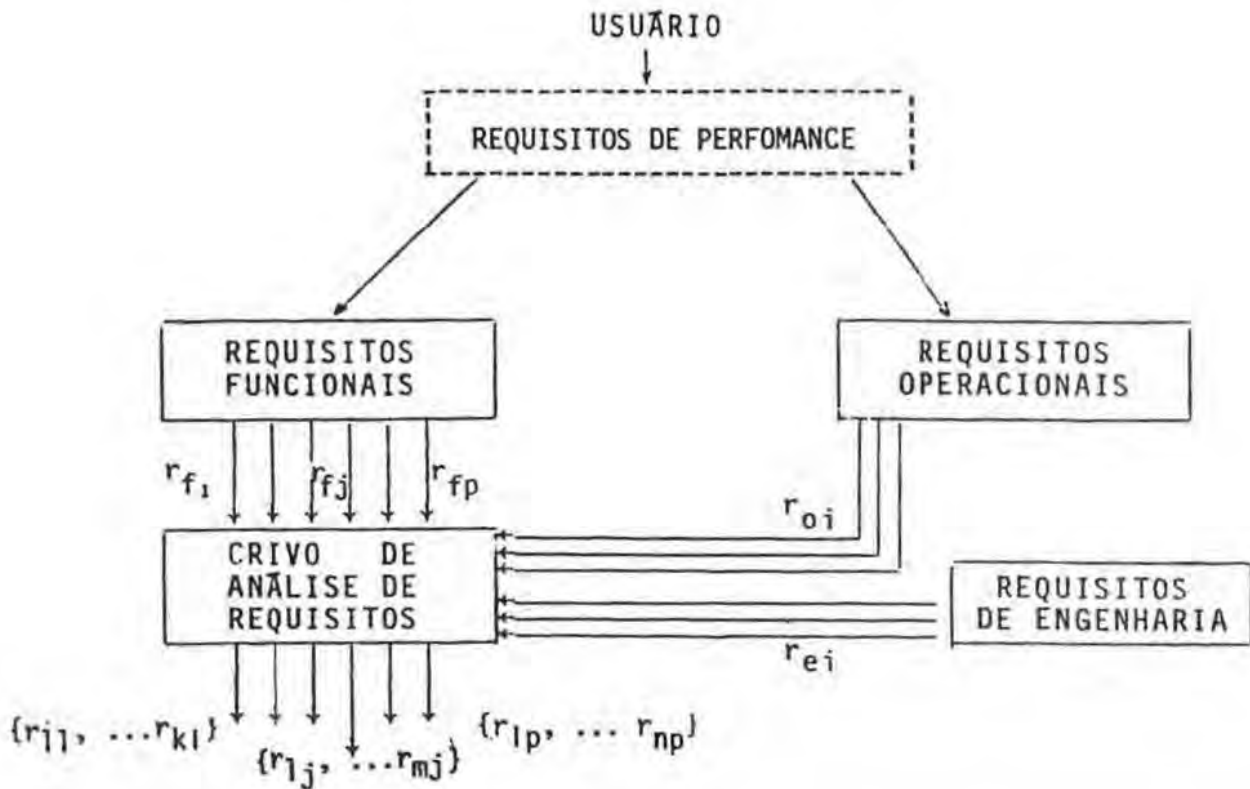


Figura 2

Em primeiro lugar é necessário separar dentro dos requisitos definidos pelo usuário quais os requisitos funcionais e quais os operacionais. Os primeiros são normalmente mais explícitos e de fácil caracterização. São considerados funcionais porque podem ser descritos pelo conceito matemático de função: espaço de partida, espaço de chegada e mapeamento do primeiro no segundo. Englobam: variáveis a tratar, tipos de tratamento, taxas de tratamento, formas de exteriorização dos resultados, parâmetros, formas de modificação dos parâmetros, prioridades para tratamento e exteriorização, etc.

Os requisitos operacionais são mais imprecisos e muitas vezes são omitidos pelo usuário já que não constam das suas preocupações básicas que são cumprir os objetivos últimos do sistema. A forma de descrição destes requisitos não é padronizada como a dos funcionais pela diversidade de noções que envolvem. Alguns exemplos destes requisitos são: índices de disponibilidade do fluxo de informações, índices de observabilidade, índices de recuperabilidade, critérios para expandibilidade quantitativa, critérios para expandibilidade funcional, linguagens para comunicação homem-máquina (ergonomia), etc.

Finalmente, os requisitos de engenharia são aqueles ditados pelos recursos tecnológicos disponíveis, funções do custo admitido para o projeto, maturidade industrial, etc, adicionados da chamada "experiência" adquirida em projetos anteriores. Salvo os dados técnicos dos elementos de hardware, os requisitos de engenharia também fazem parte de uma coleção de difícil definição pela sua heterogeneidade. Por exemplo: custos admissíveis, recursos humanos disponíveis, conceitos arquitetônicos tais como: multiprocessamento, programação estruturada, programação concorrente, sistemas operacionais, bancos de dados, objetos, monitores, etc.

O processo de crivagem será examinado a seguir tomando por base alguns elementos do sistema de supervisão. Cabe porém, a ressalva de que as recomendações produzidas pela Análise de Requisitos são por vezes conflitantes e caberá a etapa seguinte (Busca de Soluções) administrar estes possíveis conflitos.

Alguns dos principais requisitos funcionais do Centro de Supervisão são:

- 1 - Recolher dados de unidades terminais remotas;
- 2 - Tratá-los (filtrá-los, convertê-los);
- 3 - Verificar condições de alarmes;

- 4 - Enviar estes dados e condições a diversos dispositivos de exteriorização;
- 5 - Conservar uma imagem permanentemente atualizada dos dados;
- 6 - Responder a consultas sobre esta imagem;
- 7 - Enviar dados/comandos indicados pelo operador.

Alguns requisitos operacionais básicos são:

- 1 - Conservar um índice de disponibilidade do fluxo de dados superior a 99,8%;
- 2 - Permitir a observação de condições de erro com uma cobertura superior a 95% e com um retardo inferior a 5 segundos;
- 3 - Recuperar a plena capacidade de operação na presença de falhas isoladas em menos de 1 minuto e de falhas duplas em menos de 1 hora;
- 4 - O sistema deverá poder atender a um volume de dados variáveis numa escala de 1:3;
- 5 - Não está prevista expandibilidade funcional, além das apresentadas na figura 3, devendo-se contudo poder admitir um acréscimo da complexibilidade sub-funcional numa escala de 3:4;
- 6 - A linguagem de comunicação homem-máquina terá sua semântica definida a priori devendo ser possível aumentar o seu vocabulário numa escala de 1:2.

Quanto aos requisitos de engenharia, no exemplo considerado, os principais se referem à escolha de uma arquitetura do tipo distribuído.

A realização completa do processo de crivagem é bastante trabalhosa e, a nosso ver, maiores desenvolvimentos precisam ser feitos nesta área para aumentar a eficiência deste tipo de atividade.

Consideremos por exemplo o requisito funcional da detecção de condições de alarmes e sua notificação ao homem através de diversos dispositivos de exteriorização (CRT's, impressoras, painéis mímicos, etc.). Obtivemos as seguintes recomendações:

- 1 - As detecções de alarmes deverão ser feitas nos próprios módulos (tanto de hardware quanto de software) em que ocorrerem ou onde foram primeiramente observados.
- 2 - O alarme detectado deve ser codificado também a nível do módulo de tector.
- 3 - Após a codificação o aviso de alarme deve ser expedido:
 - i) a um coordenador central para inserção do mesmo em uma lista ordenada por prioridade. Como esta lista deve ser triplicada (por questões de segurança), surge a necessidade de um coordenador que garanta a homogeneidade das três listas.
 - ii) aos diversos dispositivos de exteriorização escalados para a emissão de alarmes (CRT's, painel mímico e "logger" de alarmes).

Como se pode observar, no caso do requisito em questão, as recomendações obtidas encaminham as soluções a serem buscadas sob uma ótica descentralizante visando a máxima modularidade e confiabilidade.

Procedendo-se desta forma para todo o conjunto de requisitos funcionais obteve-se um documento de natureza conceitual, não muito extenso, mas de profundo significado para a realização da etapa seguinte.

Outro subproduto desta fase é a obtenção de um primeiro nível de particionamento funcional onde são discriminadas funções básicas e as funções de aplicação (figura 3).

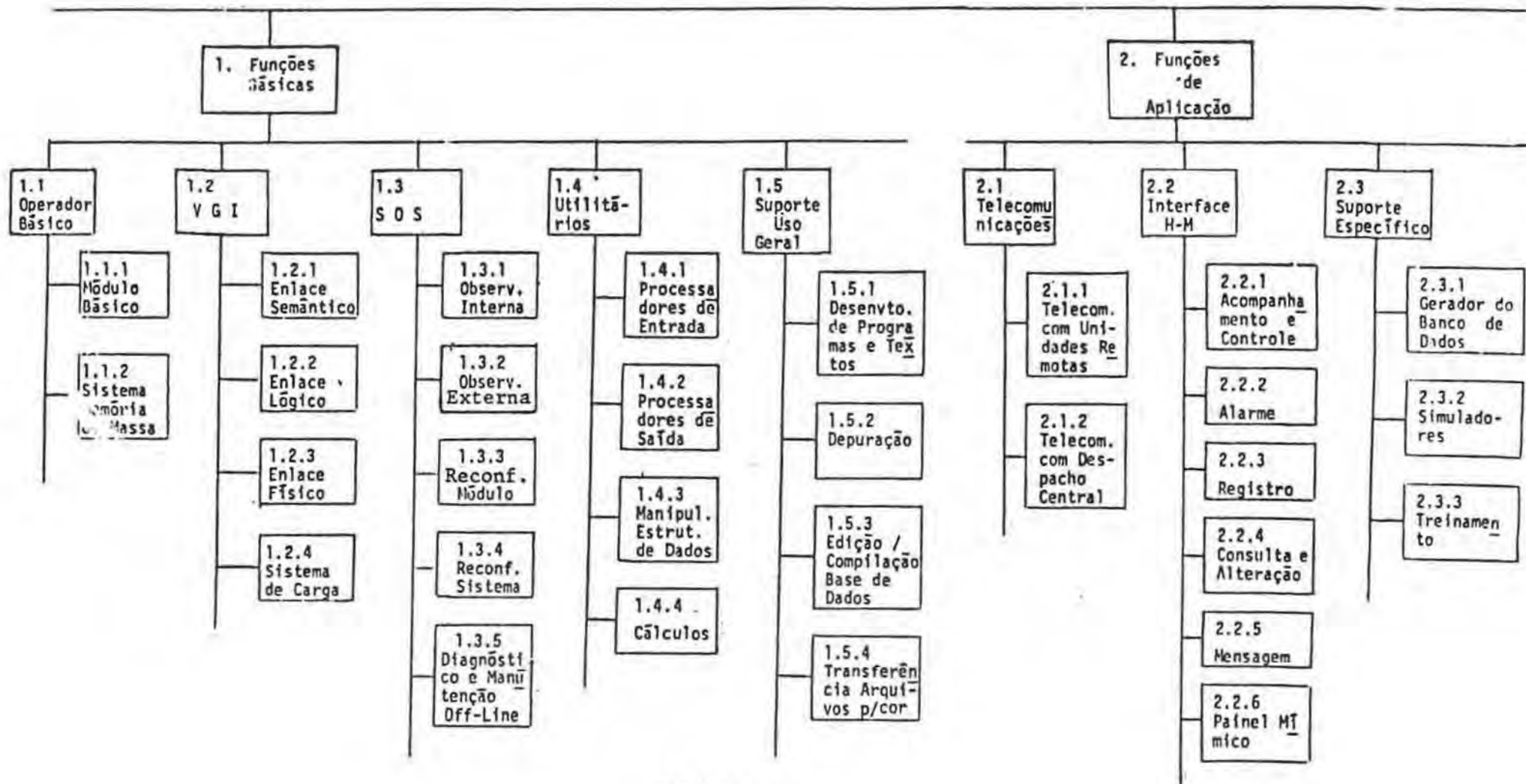


Figura 3

5. BUSCA DE SOLUÇÕES

Esta é a etapa onde se obtêm as soluções para o projeto do sistema. Os insumos são os requisitos e as recomendações para o sistema obtidos anteriormente.

O processo de obtenção das soluções é iterativo, onde um conjunto de possíveis soluções passa por uma cadeia de decisão cujos nodos são os requisitos ou as recomendações (figura 4)

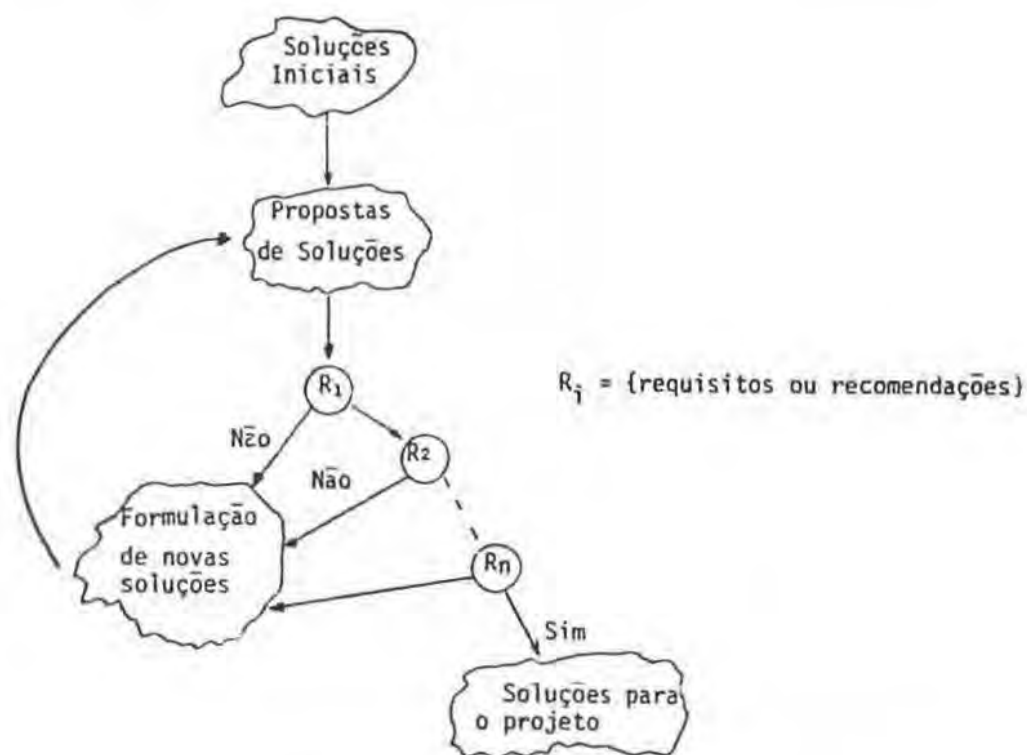


Figura 4

Para o processo de busca de soluções foi adotada uma linguagem de descrição que possui duas entidades notáveis:

- i) objetos - são os blocos básicos da descrição. Um objeto representa um conjunto de dados e as operações (métodos) que podem ser executadas sobre estes dados, sendo somente acessível ao mundo externo através de mensagens. A utilização destes blocos básicos se justifica facilmente em sistemas distribuídos, dado o baixo grau de coe

são resultante. Este baixo grau é indispensável quando se deseja poder evoluir o sistema sem causar maiores abalos às partes já construídas. Uma das recomendações que nortearam o projeto foi a de se poder inserir ou extrair objetos sem a reescrita dos demais. Outra vantagem do uso de objetos é a de proporcionar um isolamento entre os diversos projetistas ou programadores, permitindo que estes conheçam o microcosmo do objeto e ~~o~~ ^{sem necessitar conhecer} o macrocosmo do sistema.

- ii) Diagramas de sequência de ações ("Action Sequence Diagram"-ASD). Os ASD's são compostos por objetos ligados de forma tal a representarem o desempenho de uma determinada função.

A simbologia adotada nessa primeira proposta foi a seguinte:

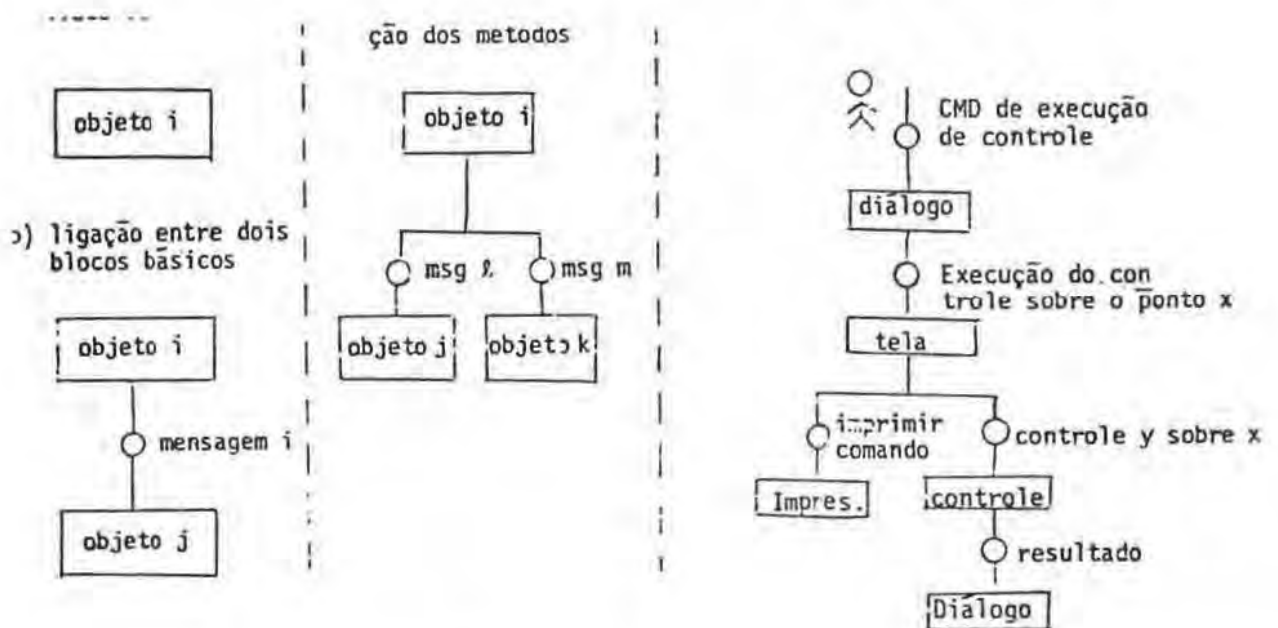


Figura 5

5.1 - Individualização dos Objetos do Sistema

Para completar a linguagem adotada para a descrição das soluções, fica apenas faltando como se determinam os objetos do sistema. O método utilizado consistiu em se fazer um mapeamento de todas as operações necessárias à realização das funções (passo 1). Associa-se a cada operação o tipo de dado que deve operar (passo 2). Grupa-se as operações segundo categorias de abstração comuns (passo 3) obtendo-se, então, os objetos (Figura 6).

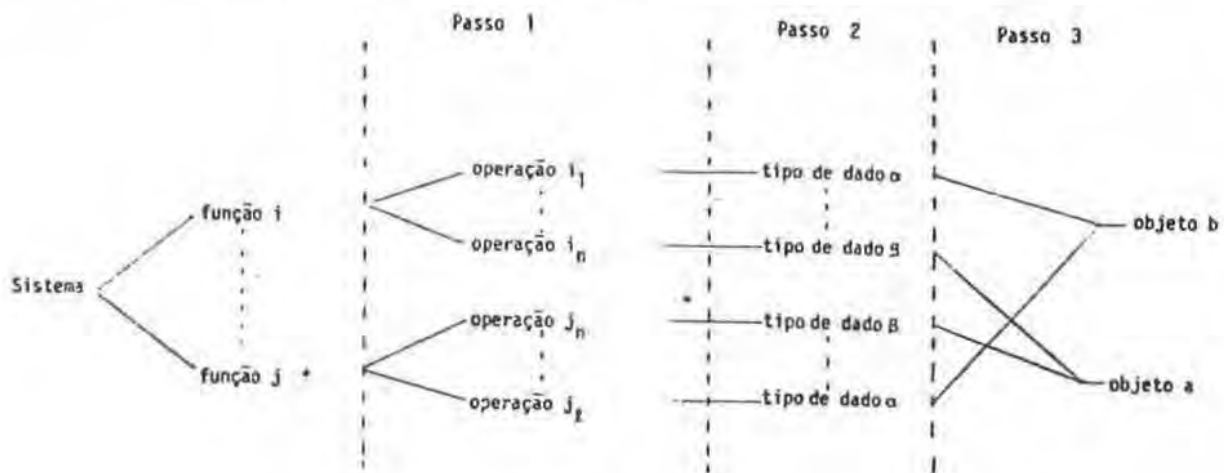


Figura 6

A partir da decomposição de funções como as da Figura 3, realizada de acordo com os requisitos, as novas funções (subfunções) foram descritas através de um processo iterativo de definição dos ASD's e objetos respectivos.

As figuras 7, 8 e 9 descrevem um exemplo de resultado da aplicação deste processo às funções de Acompanhamento e Controle.

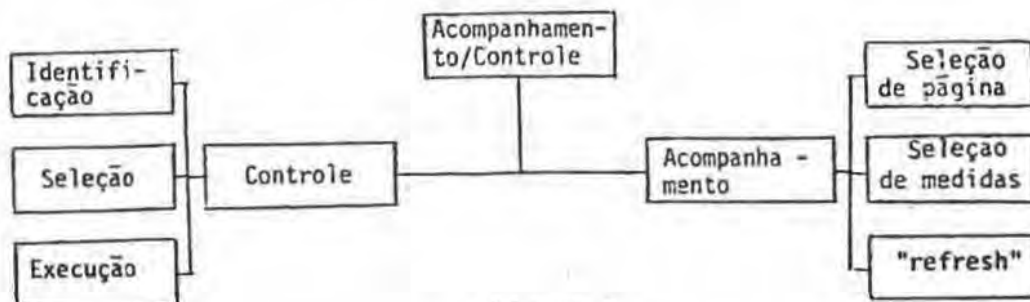
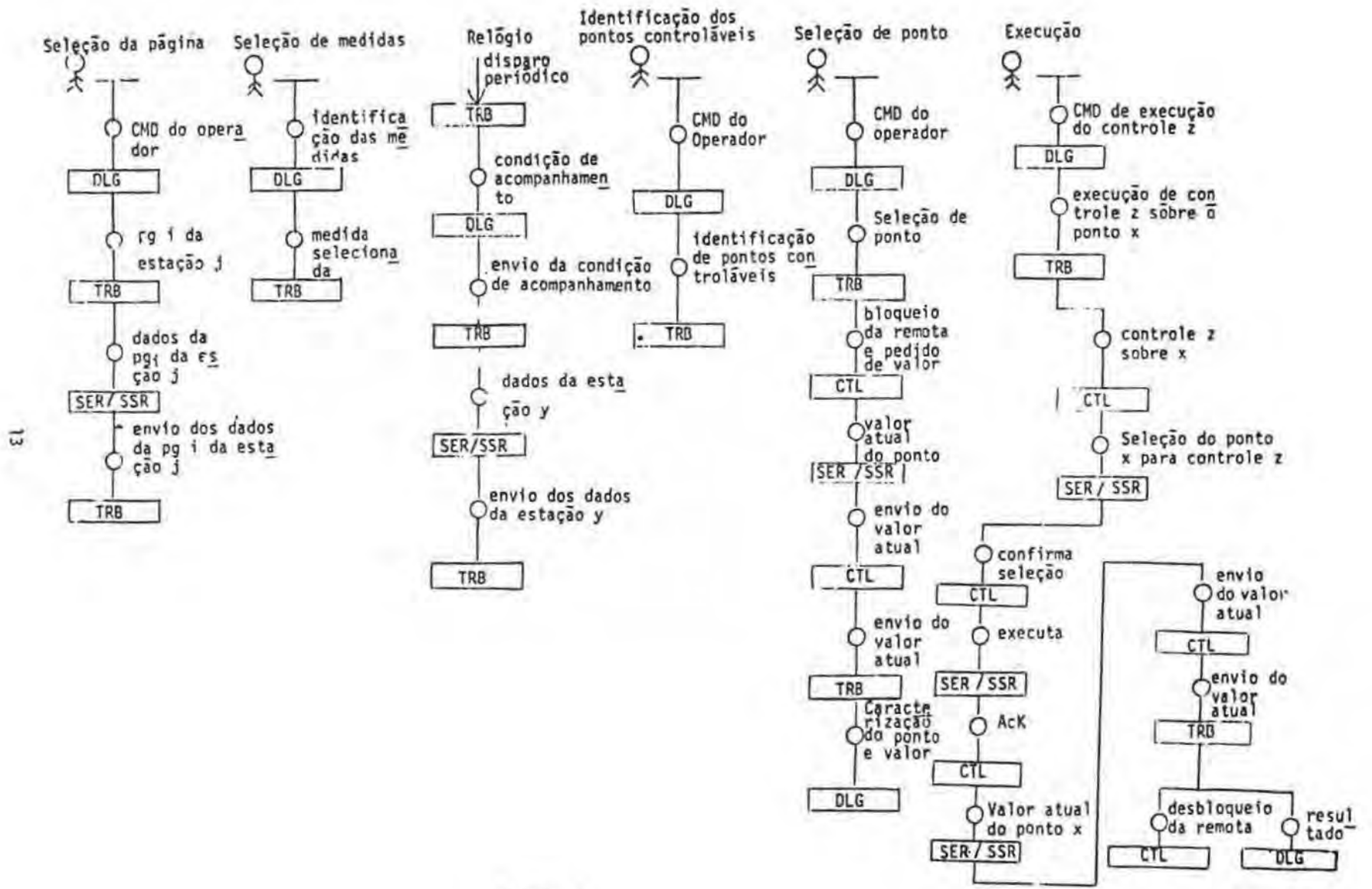


Figura 7



13

Figura 8

OBJETOS	OPERAÇÕES	DADOS
Diálogo	<ul style="list-style-type: none"> - Análise sintática e semântica dos comandos recebidos do despachante - Consulta e alteração do estado interno de diálogo 	<ul style="list-style-type: none"> - Definição sintática da linguagem - Vocabulário de comandos - Estado interno de diálogo
Tela de Trabalho (TRB)	<ul style="list-style-type: none"> - Busca e exteriorização da página no vídeo - Alteração do estado de exteriorização da página 	<ul style="list-style-type: none"> - Diretório das páginas - Conteúdo das páginas - Estado da exteriorização da página (inclusão ou retirada de dados da página).
Controle (CTL)	<ul style="list-style-type: none"> - Bloqueio e desbloqueio de uma estação para controle - Execução da lógica necessária à efetuação do controle 	<ul style="list-style-type: none"> - Estado do controle - Dicionários de sequências de controle
SER/SSR	<ul style="list-style-type: none"> - Aquisição de dados do sistema elétrico e de supervisão - Envio de comandos ao sistema elétrico e de supervisão 	<ul style="list-style-type: none"> - Estado do sistema elétrico e de supervisão

Figura 9

6. ESPECIFICAÇÃO DAS SOLUÇÕES DO PROJETO

Esta etapa apresenta a descrição das soluções que passaram pelo crivo da etapa de Busca de Soluções. Esta descrição apresenta minuciosamente as operações e os dados dos objetos dos ASD's escolhidos. Cada operação i identificada deverá ser descrita por seu conjunto de entrada, de saída, seu efeito e condições de exceção (opcional), conforme mostra a tabela 1.

As descrições dos dados também deve ser feita cuidadosamente, identificando formatos, volumes, formas de acesso e sobretudo analisando possíveis inconsistências.

Tabela 1

Exemplo: ESPECIFICAÇÃO DA OPERAÇÃO 'SELEÇÃO DE PÁGINA'	
<u>ENTRADAS:</u> Conjunto de comandos possíveis:	
{ S_i S_i = sigla de estação, acesso direto a uma página definida, página anterior/seguinte, colocação/retirada dos nomes dos dispositivos no diagrama }	
<u>SAÍDAS:</u>	- pedidos de página à Tela de Trabalho - envio de críticas ao despachante sobre comandos recebidos
<u>EFEITOS:</u>	- fixação do status da estação sob acompanhamento, onde status = < nome da estação, página em exibição, medidas selecionadas, exibição ou não dos nomes dos dispositivos > - estados de exceção

7. PROJETO

Esta etapa é responsável pela seguinte atividades:

- i) Mapeamento dos objetos no sistema distribuído;
- ii) Definição dos "chips de software" que implementem os objetos;

iii) Definição do sistema de interligação dos chips para a construção do sistema.

7.1 - Mapeamento dos objetos no sistema distribuído

O mapeamento dos objetos no sistema distribuído poderá ser feito "off-line", "on-line" ou de forma mista. No modo "off-line" cabe ao projetista determinar de forma rígida a localização dos objetos no sistema. Esta localização é fixa.

No modo "on-line", os objetos podem migrar no sistema de acordo com a disponibilidade dos recursos computacionais. Na forma mista, alguns objetos são localizados de forma "off-line" e outros, podem migrar no sistema.

Os principais critérios a serem obedecidos no mapeamento dos objetos são os seguintes:

- i) indivisibilidade do objeto: este critério deve ser perseguido de forma tal a manter mínima a coesão entre os diversos componentes do sistema.

- ii) atendimento às recomendações da análise de requisitos e disponibilidade de recursos computacionais: no caso em questão, existem operadores que possuem funções semelhantes no sistema tais como os operadores de comunicação com as estações remotas (vide figura 1). Esta replicação de operadores dará origem ao particionamento dos dados de um objeto, mantendo-se porém, os métodos replicados e independentes entre si.

7.2 - Definição dos chips de software que implementarão os objetos

Para o projeto dos objetos utilizamos uma disciplina rígida que resultou nos chamados chips de software.

Um chip é constituído, basicamente, das partes mostradas na Figura 10.

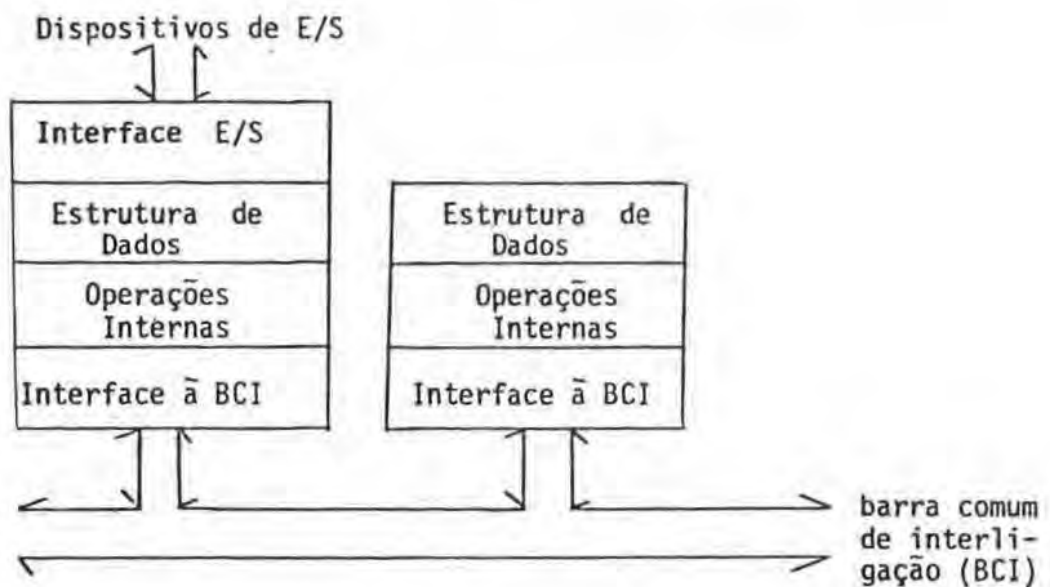


Figura 10

A filosofia subjacente à criação do conceito de chip foi a de se criar uma analogia entre o software e o hardware. Vê-se, no futuro, a existência de bibliotecas (manuais) que contenham os chips já implementados. Visualizamos que a fase de projeto deverá ser a da individualização dos chips para a realização do sistema, eventualmente criando-se um novo chip. A fase de implementação terá apenas de unir os chips de forma conveniente para a realização do sistema.

7.2.1 - Operações Internas

Representam o conjunto de procedimentos e funções necessários à execução dos métodos do chip. Internamente ao chip os métodos poderão ser executados de forma puramente combinacional (o conjunto de saída é função do de entrada), ou sequencial (o conjunto de saída é função do de entrada e do de estados internos ao chip). O projetista deverá utilizar estratégias de sincronismo interno que se fizerem necessárias à execução dos métodos. Por exemplo: um pedido de leitura pode ser inibido ou postergado se estiver em curso um processo de alteração de uma área de memória.

7.2.2 - Estrutura de Dados Interno

A organização dos dados internamente ao chip é totalmente livre, ficando esta tarefa a cargo do programador que irá implementar o chip. Estes dados não são acessíveis ao sistema, criando-se portanto, um escopo restrito ao chip.

7.2.3 - Interface à BCI

Esta é a única parte "visível" de um chip aos demais do sistema (ponto de observação do chip). Por apresentar esta propriedade, qualquer alteração nesta interface repercutirá nos demais chips.

Sua função é a de selecionar o conjunto de operações internas ao chip para a realização de um método. Por exemplo: o chip de comunicação com a estação remota ao receber um pedido de execução do método "varredura dos pontos analógicos" deverá executar as seguintes operações internas:

- i - Montagem das mensagens a serem enviadas às remotas para obtenção dos seus dados analógicos;
- ii - Comunicação com a remota para a obtenção de dados;
- iii - Preparação do buffer de dados recebidos da remota.

7.2.4 - Interface com Dispositivos de E/S

Representa o conjunto de "handlers" que especializam o chip a um determinado tipo de dispositivo de E/S. Por exemplo: terminal de vídeo. A troca de um dispositivo por outro deve resultar apenas na alteração desta interface, mantendo-se intacto o resto do chip.

7.3 - Sistema de Interligação dos Chips para a Construção do Sistema

Idealmente, este sistema deve ser passivo. Na prática, porém, o que se observou foi que este sistema agregou todos os recursos comuns (utilitários) aos chips, tais como: sistema operacional, sistema de comunicação entre chips, etc.

OBS.: O sistema de comunicação entre os chips (enlace semântico) foi um utilitário concebido de forma tal a permitir com que o programador se abstraia da localização física de um chip. Caso os chips estejam localizados em operadores diversos, o enlace semântico funcionará como um adaptador de bus entre os operadores (Figura 11)

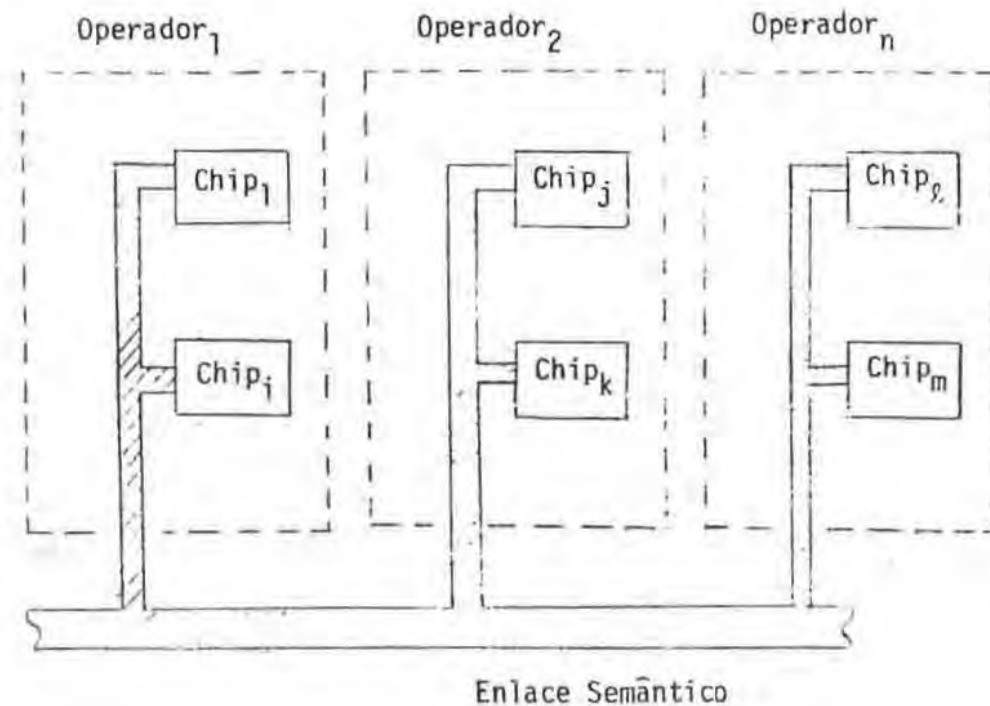


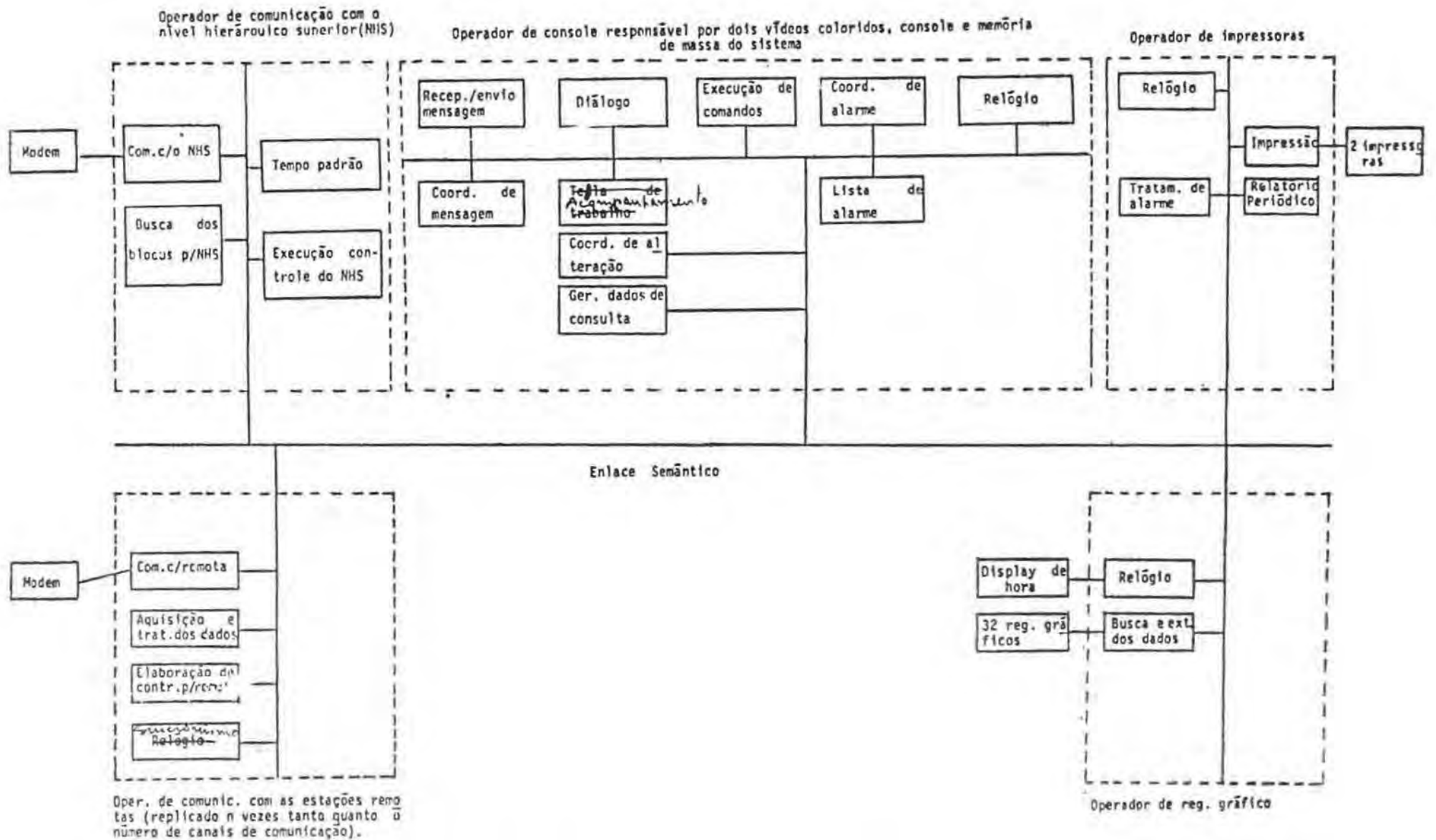
Figura 11

Na prática os utilitários são rotinas que se agregam ao chip, podendo desta forma estar presentes em todos os chips. Ao se agregarem, os utilitários passam a apresentar as características do chip, tal como a sua prioridade, stack, etc.

Fazendo-se uma analogia com o hardware, o sistema de interligação dos chips pode ser comparado ao conjunto placa de circuito impresso (elemento passivo) e fontes de alimentação (elemento ativo).

7.4 - Resultado da fase de projeto

A figura 12 apresenta um possível mapeamento dos objetos, a nível de chips, do sistema de supervisão.



EXEMPLO DE MAPEAMENTO DOS OBJETOS NO SISTEMA DE SUPERVISÃO

Figura 12

8. IMPLEMENTAÇÃO E TESTE

A implementação do sistema deve seguir um plano diretor elaborado na fase de projeto. Este plano estabelece o seguinte:

- i - a nível de sistema: define camadas sucessivas de chips a serem implementadas/testadas para a obtenção das funções. Camadas sucessivas podem acrescentar mais chips ou métodos nos chips existentes.
- ii - a nível de chip: estabelece a prioridade com que as diversas operações deverão ser implementadas tendo em vista os requisitos das diversas camadas do item i.

A técnica para a confecção deste plano diretor é a do tipo zig-zag (zig-zag approach), que combina os modos "top-down" e "bottom-up". Esta técnica se justifica em sistemas de controle de processos pela necessidade de se atender a requisitos de tempo de resposta restritos. Estes requisitos enfatizam a necessidade de consolidação das partes dos chips responsáveis pelas interfaces de entrada e saída.

Para a construção dos chips foram adotadas técnicas que visam auxiliar a administração da implementação, tais como:

- i - criação de um módulo de descrição do chip ou utilitário. Este módulo deve conter as seguintes informações: título do módulo, data de entrega, descrição suscintada, arquivos de dados e programas que compõem o chip, procedures ou funções externas ao módulo referenciadas neste;
- ii - descrição dos algoritmos das rotinas utilizando-se linguagem estruturada antes de se partir para a codificação. O código deverá possuir como comentário frases do algoritmo em linguagem estruturada (pretende-se garantir desta forma a estruturação do código).

9 - CONCLUSÃO

O trabalho aqui apresentado foi uma primeira iniciativa visando criar uma mentalidade de visualização do software como uma área de engenharia no âmbito do projeto de Centros de Supervisão. Deve-se notar porém, que diversas das técnicas mencionadas já vinham sendo adotadas por iniciativa pessoal. Nosso objetivo é sistematizar tais procedimentos.

Os resultados até agora obtidos já comprovam a validade desta iniciativa. Observa-se na equipe do projeto uma maior polarização em níveis de competência, resultando uma maior eficiência, melhor documentação das diversas fases e maior observabilidade.

Prevê-se como extensão deste trabalho as seguintes atividades:

- Adaptação das técnicas mencionadas a outros projetos de controle em tempo real;
- Aperfeiçoamento da metodologia proposta principalmente no que tange a: Sistemática de Análise de Requisitos, sistema convencional de definição de requisitos, linguagem para descrição dos ASD's e objetos, criação de bibliotecas de objetos.

Agradecimento:

Agradecemos a todos os participantes do projeto do Centro de Operação Regional de FURNAS pela sua ativa participação no desenvolvimento dos métodos expostos no artigo, e em especial, aos engenheiros Antonio Joaquim Setubal, Cesar A. Boavista, Homero G. Andrade, José A. Motta e Sergio P. Cotrim.

BIBLIOGRAFIA

- (1) BERGLANI, G.D.; "A Guided Tour of Program Design Methodologies"; Computer, October 1981.
- (2) BERSOFF, E.H.; HENDERSON, V.P.; SIEGEL, S.G.; "Software Configuration Management: An Investment in Product Integrity"; Prentice Hall, Englewood Cliffs, N.J., 1980.
- (3) DALY, E.; "Management of Software Development"; IEEE Transactions on Software Engineering, May 1977.
- (4) ENOS, J.C.; VAN Tilburg, R.L.; "Software Design"; Computer, February 1981.
- (5) FERNANDES, A.L.B.; ANDRADE, H.G.; GRACIA, J.; MONTEIRO, M.C.B.; MOSZKOWICZ, M.; APPEL, O.; COSTA, R.S.; FREITAS, C.A.B.; MOTTA, J.A.P.; GARROFÉ, P.H.S.; COTRIM, S.P.R.; "Projeto de Centros de Supervisão de Sistemas de Energia Elétrica"; Anais 19 SICOP, Maio 1981.
- (6) FURNAS - Centrais Elétricas S.A.; "Expansão do Sistema de Furnas, Sistemas de Supervisão e Controle - CEP 170", Especificações Técnicas, junho 1981.
- (7) LISKOV, B.; ZILLES, S.; "An Introduction to formal Specifications of Data Abstractions"; Current Trends in Programming Methodology, Prentice Hall Inc., 1977.
- (8) ROSS, D.T.; SCHOMAN K.E.; "Structured Analysis for Requirements Definition"; IEEE, Transactions on Software Engineering, January 1977.
- (9) XEROX LEARNING RESEARCH GROUP, "The Small Talk - 80 System"; Byte, August 1981.
- (10) YOURDON, E. Constantine, L.; "Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design"; Prentice Hall Inc., 1979.

CONCURSO DE TRABALHOS
LIVRO DE ATAS DO 1º SIMPÓSIO
SOBRE REDES DE COMPUTADORES
MÚLTIPLAS PARTES

STATMUX: ESPECIFICAÇÕES,
DESENVOLVIMENTO E APLICAÇÕES.

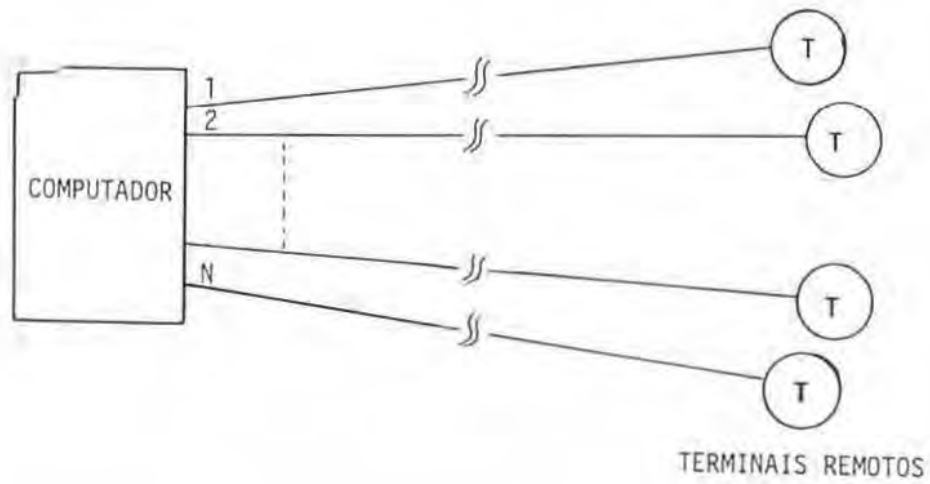
1º SIMPÓSIO SOBRE REDES DE COMPUTADORES

AUTOR: JONAS BARCELLOS DE MORAES

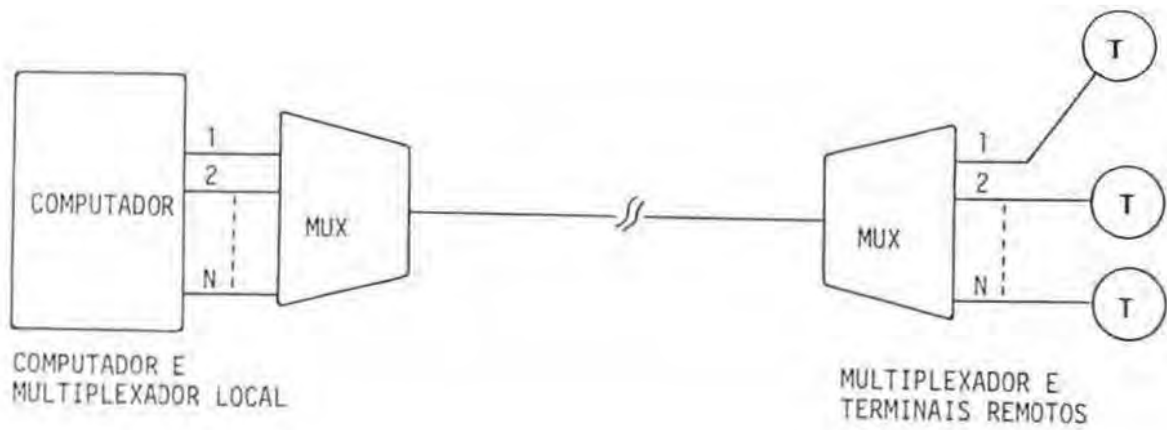
UFRGS - DIGITEL

STATMUX: ESPECIFICAÇÕES, DESENVOLVIMENTO E APLICAÇÕES

- Histórico
- Razão da Multiplexação Estatística
- Aplicações STATMUX
- Futura Família de Equipamentos
- STATMUX
 - Especificações
 - Programabilidade
 - Protocolos nos Canais Principais e Técnica de Alocação de Banda
 - Aproveitamento do(s) Canal(is) de Comunicação
 - Hardware
 - Software



Teleprocessamento sem Multiplexação

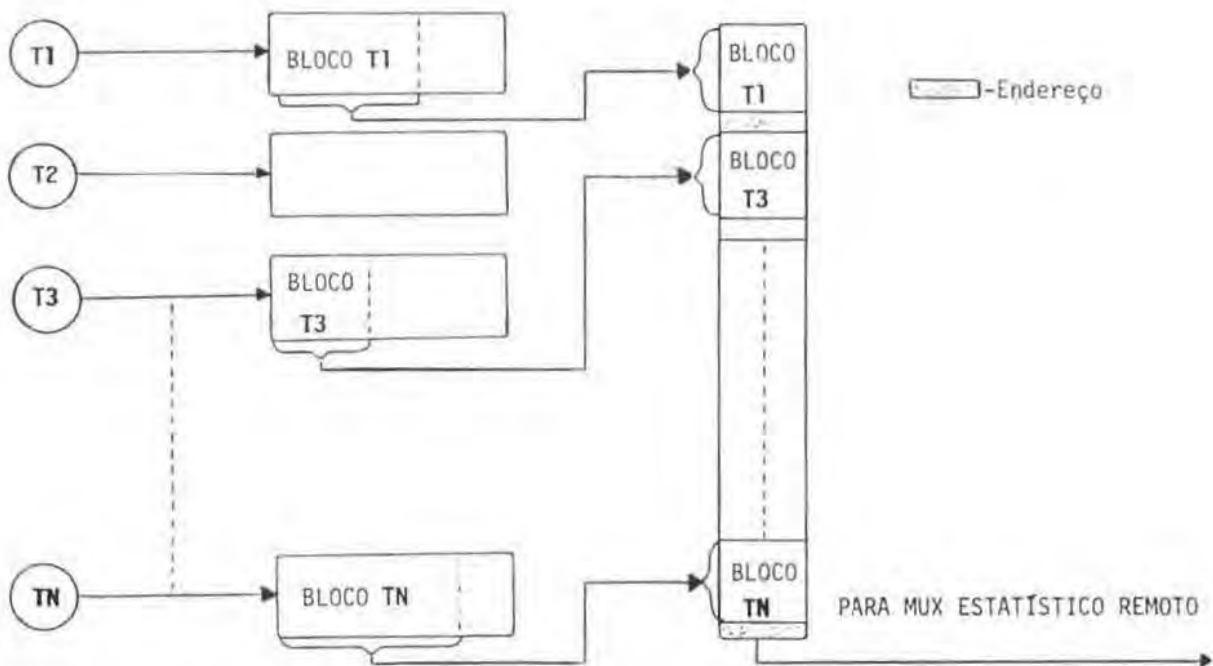


Teleprocessamento com Multiplexação

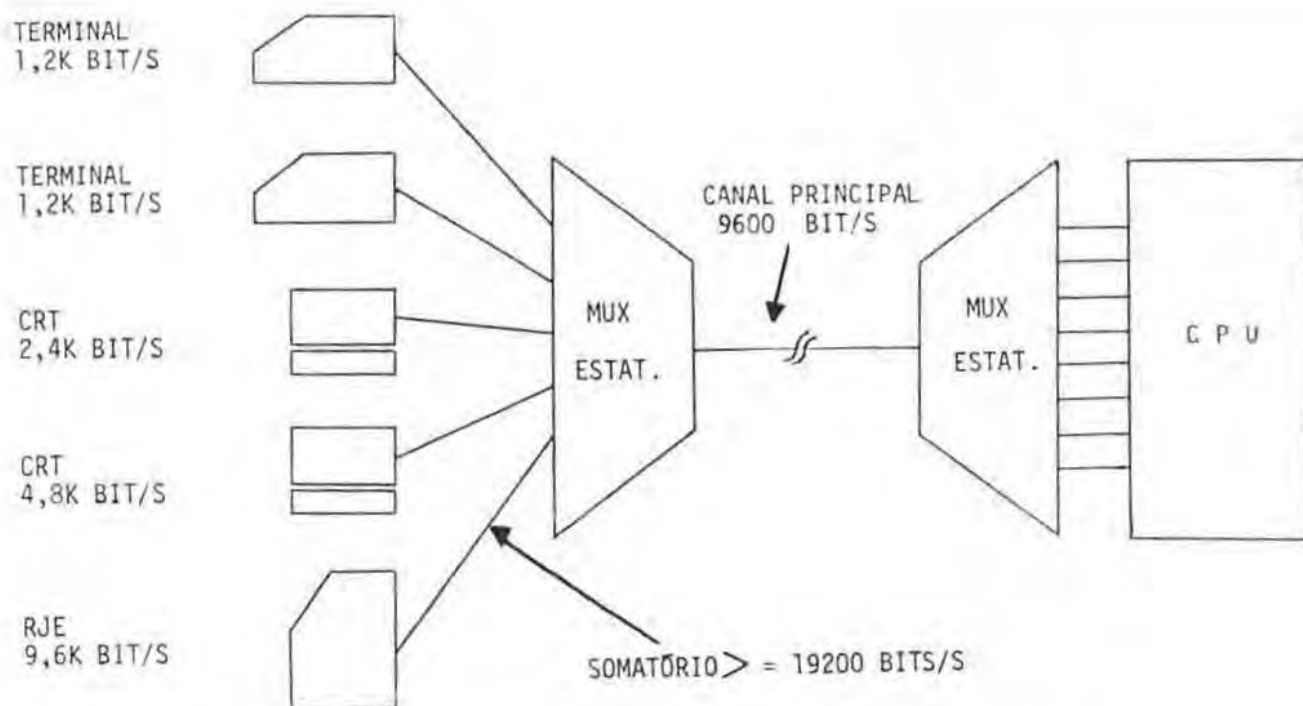
MULTIPLEXAÇÃO ASSÍNCRONA (ORIENTADA A MULTIFATIAS OU BLOCOS)

- Atribui várias fatias de tempo a um mesmo terminal, transmitindo um bloco inteiro.
- O endereçamento é por blocos, aumentando a eficiência da transmissão em relação a técnica TDM.
- Necessidade de armazenamento de dados antes da transmissão.

A FIGURA ABAIXO ILUSTRA ESTA TÉCNICA.

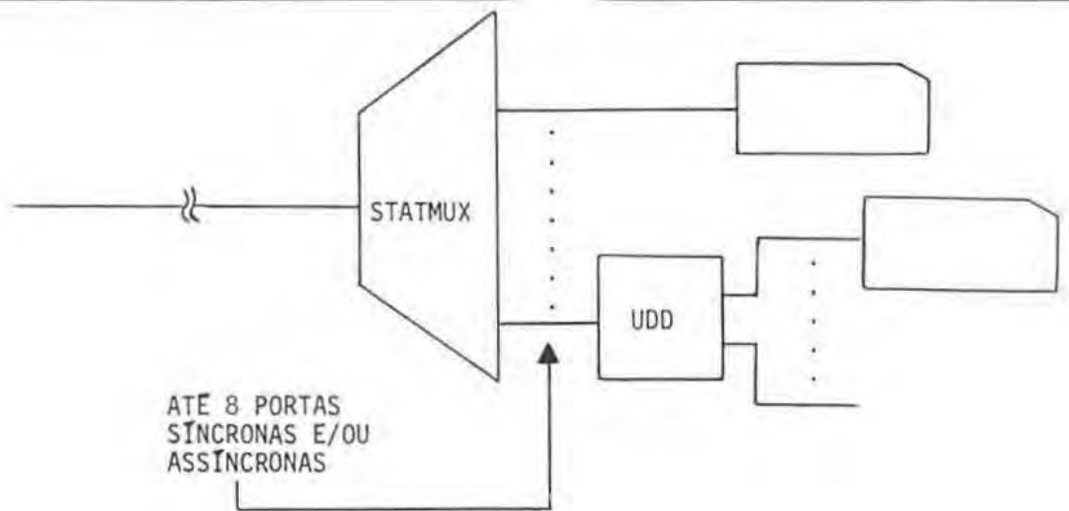
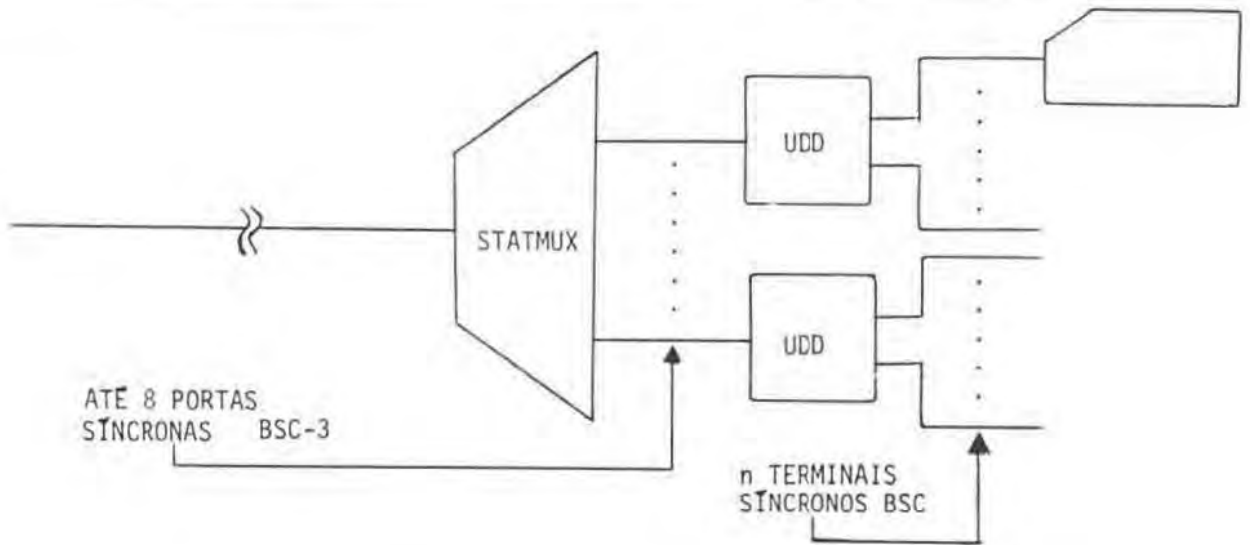
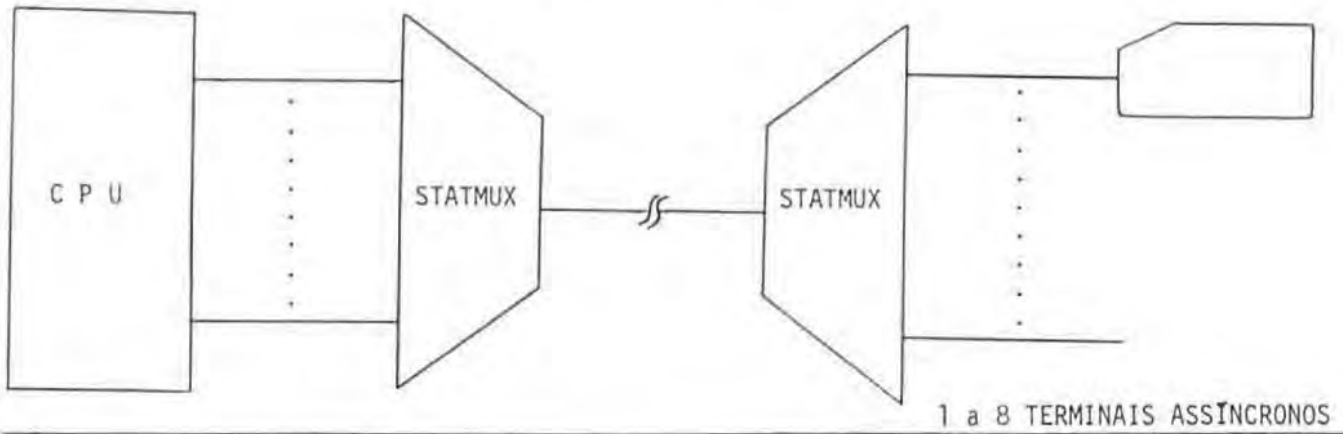


Multiplexação Estatística Assíncrona

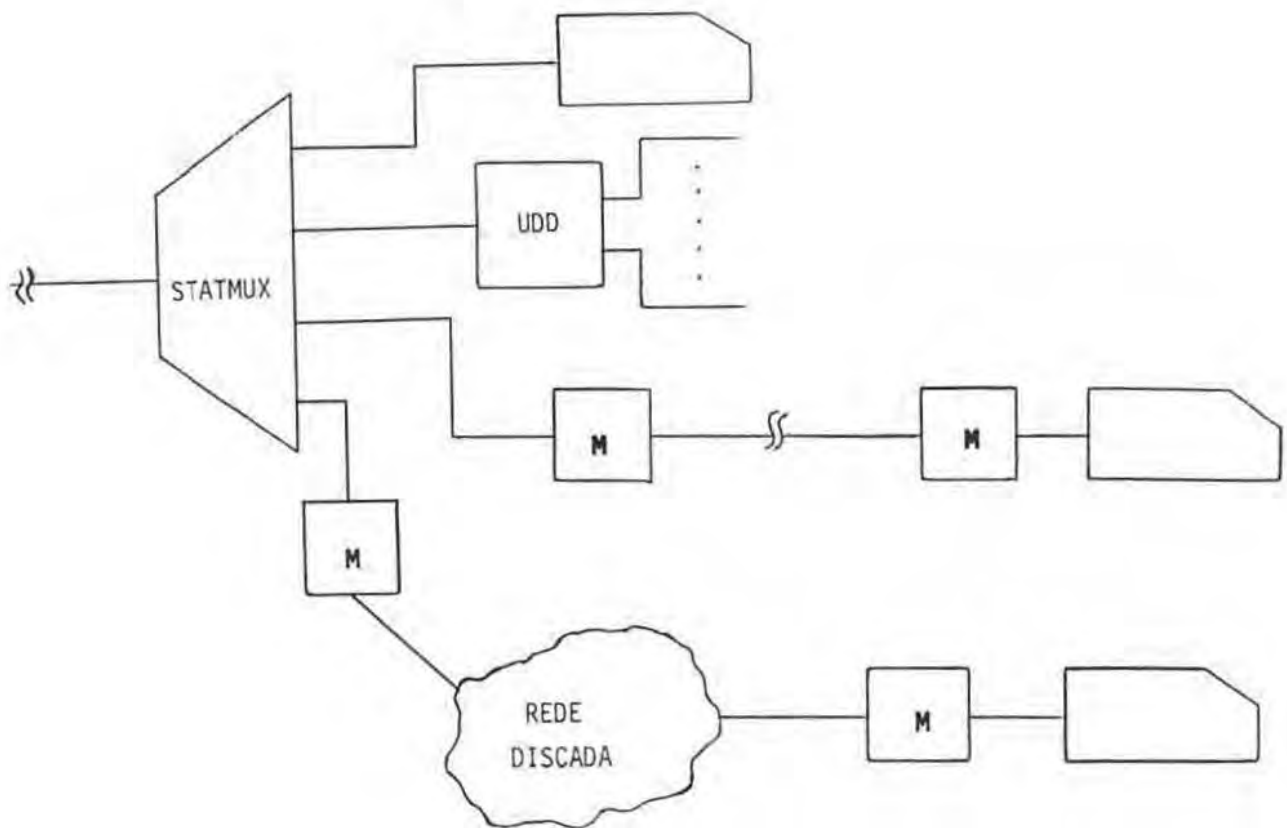


Resultado Prático da Utilização de um Multiplexador Estatístico

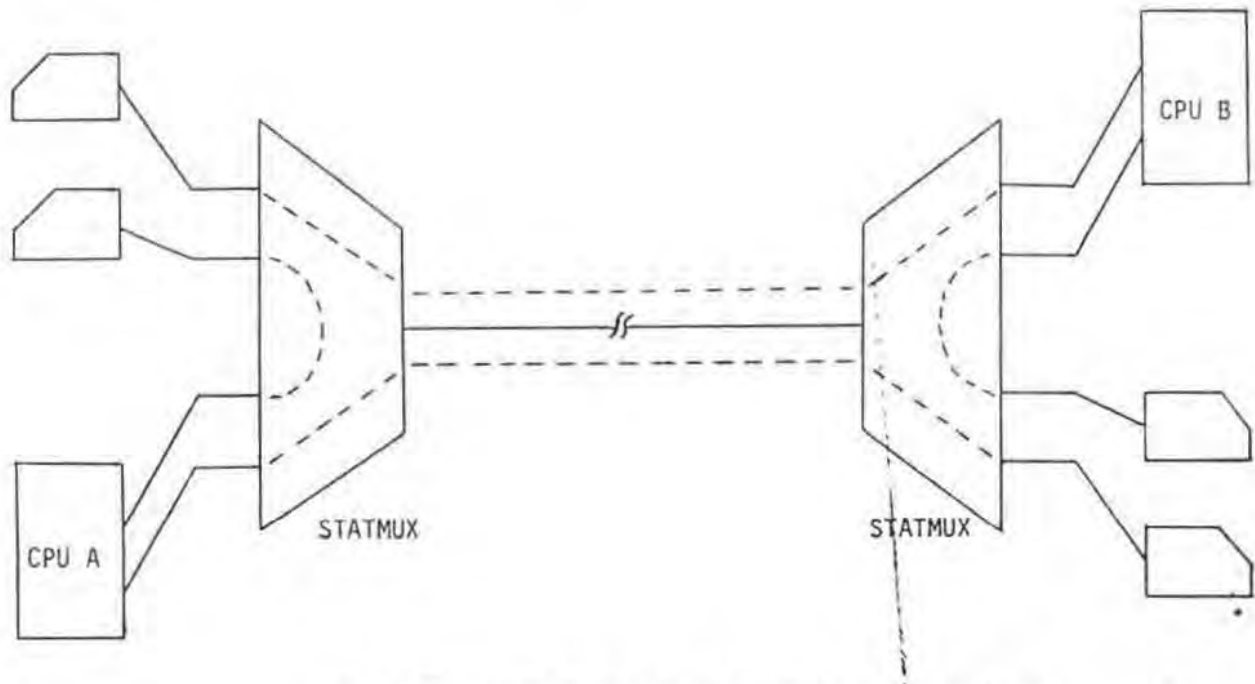
APLICAÇÕES STATMUX



APLICAÇÕES STATMUX

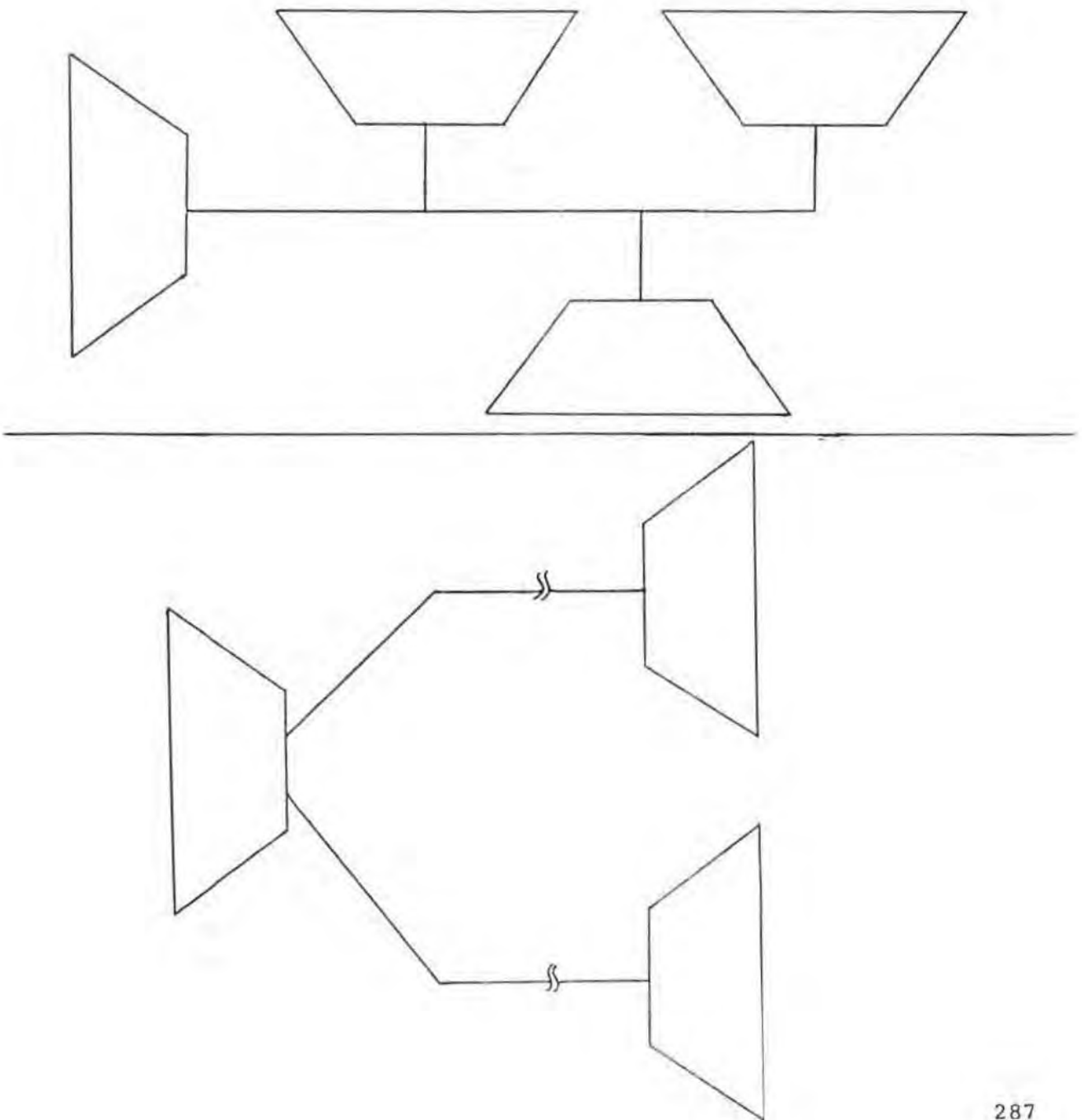


STATMUX - Futura Família

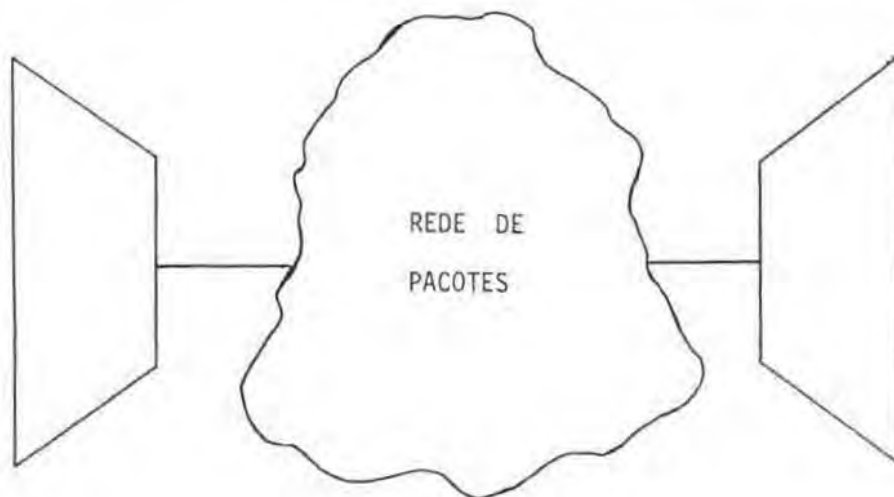
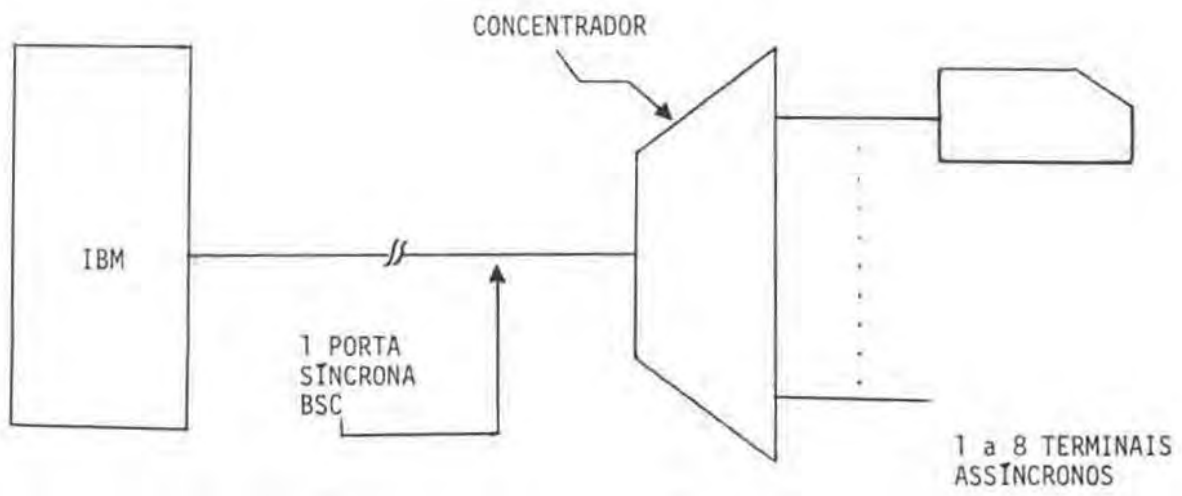


CHAVEAMENTO LOCAL E REMOTO NOS CANAIS SECUNDÁRIOS

STATMUX - Futura Família



STATMUX : Futura Família



Definição das características e funções do Multiplexador Estatístico STATMUX.

LADO BAIXA VELOCIDADE:

- nº básico portas: 4
- Extensão: 4
- Modo Sincronismo: Assíncrono, síncrono, isócrono.
- Velocidades: 50 a 9600 bps (TX/RX P/ASSINC.)
- Código: 5 a 8 bits/caracter
- Interface elétrica: V.24/V.28 (EIA-RS-232-C)
- Modo Transmissão: FDX/HDX
- Paridade: Par, Impar, Nenhuma
- Nº Bits Stop: 1, 1 1/2, 2.

LADO ALTA VELOCIDADE:

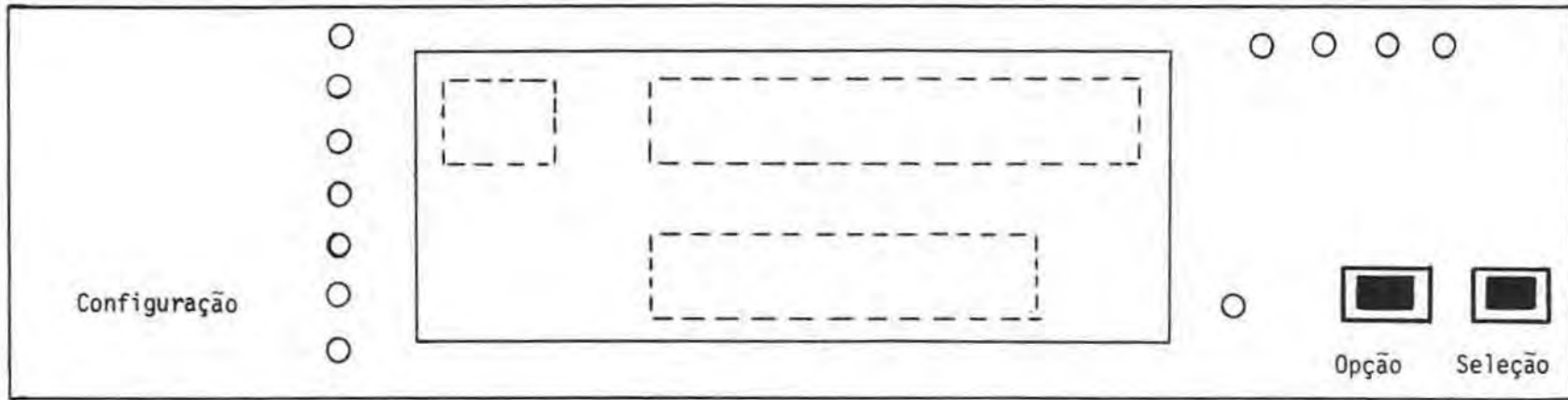
- Nº básico portas: 1
- Extensão: 1
- Modo sincronismo: síncrono
- Velocidades: 1200, 2400, 4800, 9600
- Interface Elétrica: V24 / V.28
- Modo Transmissão: FDX
- Origem Relógio: interna / externa

FUNÇÕES DO MULTIPLEXADOR ESTATÍSTICO STATMUX.

- Tratamento de erro no Canal Principal.
- Controle de fluxo Local e Remoto.
- Resposta ao controle de fluxo.
- Laços locais e remotos para os canais secundários e principais.
- Auto-teste.
- Echoplex.
- Reconhecer e repassar break.
- Down-loading.
- Atender porta de controle ou supervisora.
- Flyback buffering.
- Relatórios de rede (fornecer)
- Conversão de velocidade.
- Atendimento à linha discada nos canais secundários.

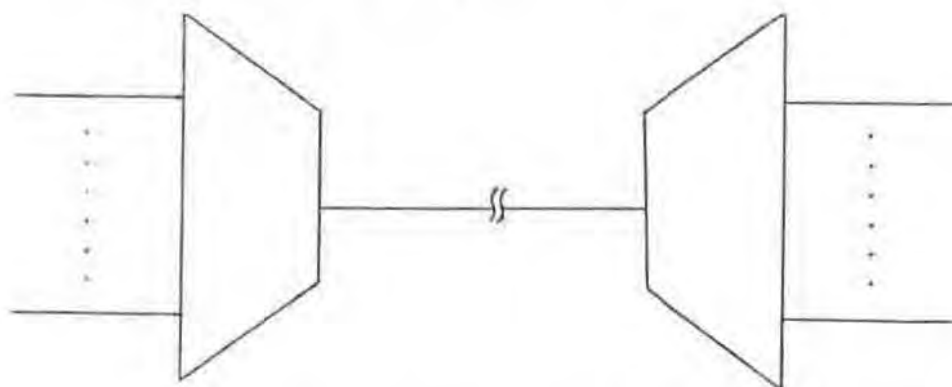
STATMUX : PROGRAMAÇÃO DA REDE

- Por Painel
- Por Console



- 13 -

Protocolo do Canal Principal STATMUX



- Coordena, define como serão as trocas de informações entre os MUX.
- Determina a maior ou menor eficiência no compartimento do Canal Principal.

O protocolo do Canal Principal é dividido em 3 níveis:

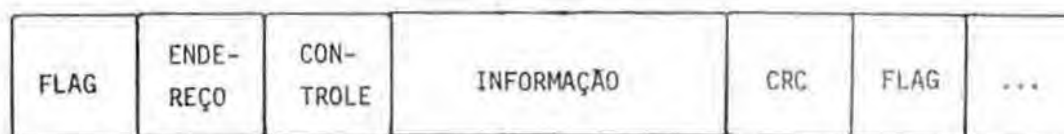
Nível 1: V24 / V28

Nível 2: CCITT X.25 - nível 2

Nível 3: Estudado e definido

PROTOCOLO DE NÍVEL 2:

Estrutura do Frame:



Procedimento Geral:



PROTOCOLO DE NÍVEL 3:

- Consiste, basicamente, da técnica de endereçamento das informações relativas a cada Canal Secundário.
- Características desejáveis no protocolo:
 - Mínimo overhead
 - Overhead diminua quando a carga aumenta.
 - Terminais ativos possam, se necessário, ocupar a banda dos terminais inativos dentro de um mesmo frame.

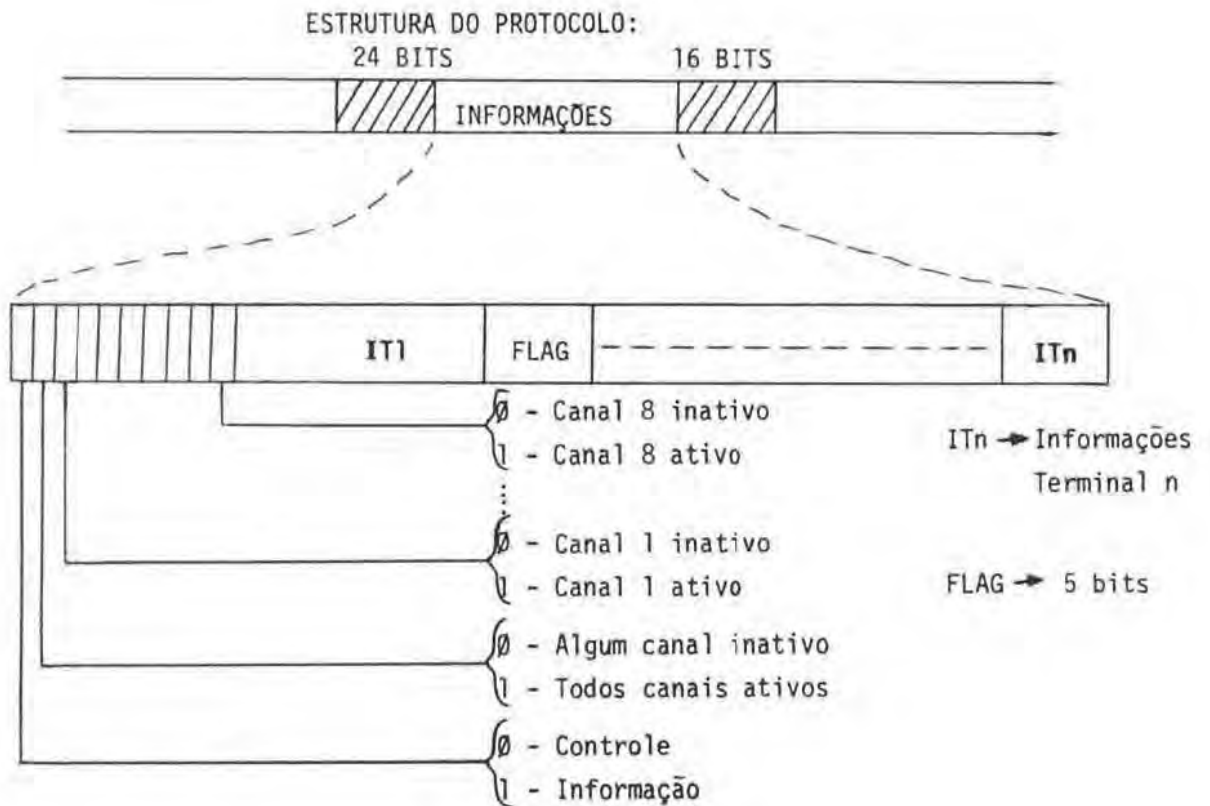


TABELA ALOCAÇÃO BANDA

<u>Nº TERMINAL</u>	<u>BANDA MÁXIMA</u>
1	IT1 máx
2	IT2 máx
⋮	⋮
n	ITn máx

ITn máx DEPENDE:

- Velocidade Terminal
- Utilização média do mesmo
- Fluxo total líquido informação passando pelo MUX (Médio).

$$ITn \text{ máx} = \frac{\text{Fluxo Líquido Médio Inform. Term. } n \times \text{Tam.Frame}}{\text{Fluxo Total Líquido Médio no MUX}}$$

Se nem todos os terminais estão ativos num determinado frame, ITn máx de cada terminal ativo muda para:

$$ITn \text{ máx} = \frac{ITn \text{ máx da Tabela Alloc.} \times \text{Tam.Frame}}{\sum \text{ITn máx dos term. ativos (da Tabela)}}$$

ANÁLISE DO PROTOCOLO:

OBJETIVO: Avaliar, em função do tamanho do frame (maior ou menor atraso) e do overhead (informações de controle), qual a quantidade efetiva de informação que passará pelo link.

EQUAÇÕES:

NOMENCLATURA

F - Comprimento do frame (bits)

OH - Overhead no frame (bits)

L - Carga de dados no frame (bits/s)

C - Capacidade do link (bits/s)

Supondo $L < C$, temos:

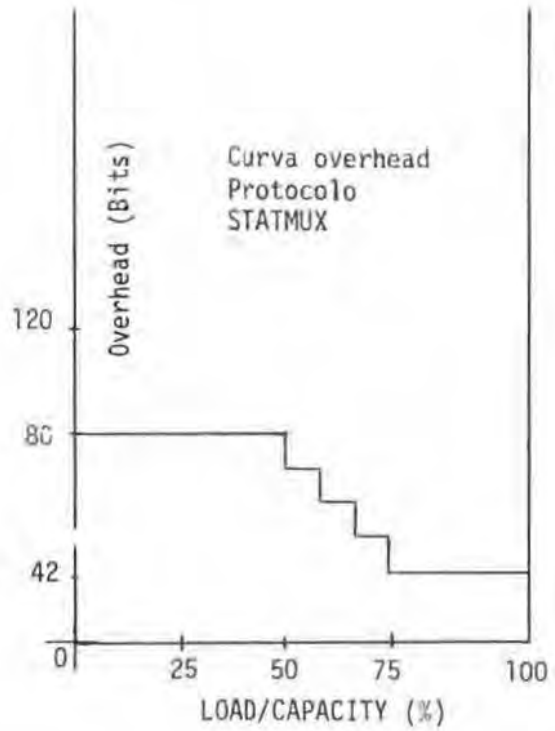
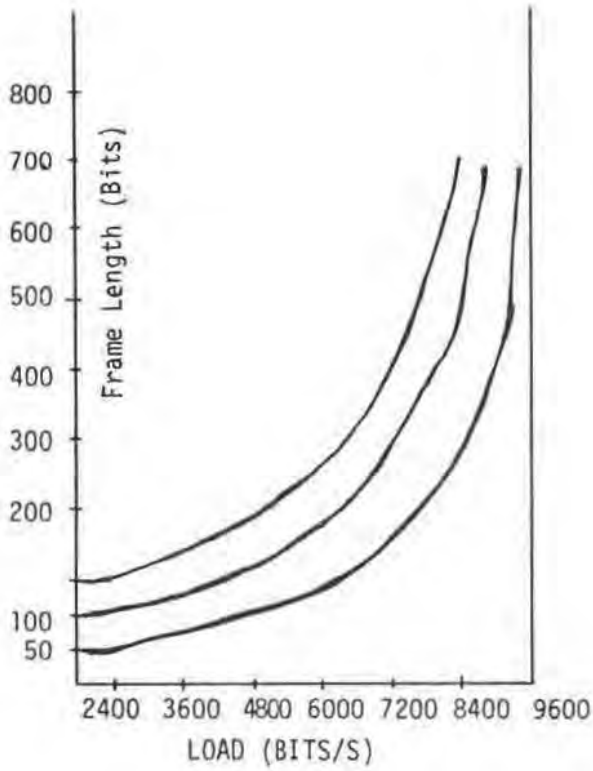
$L/C \rightarrow$ Percentagem utilização link

$1 - L/C \rightarrow$ Parcela desperdiçada link

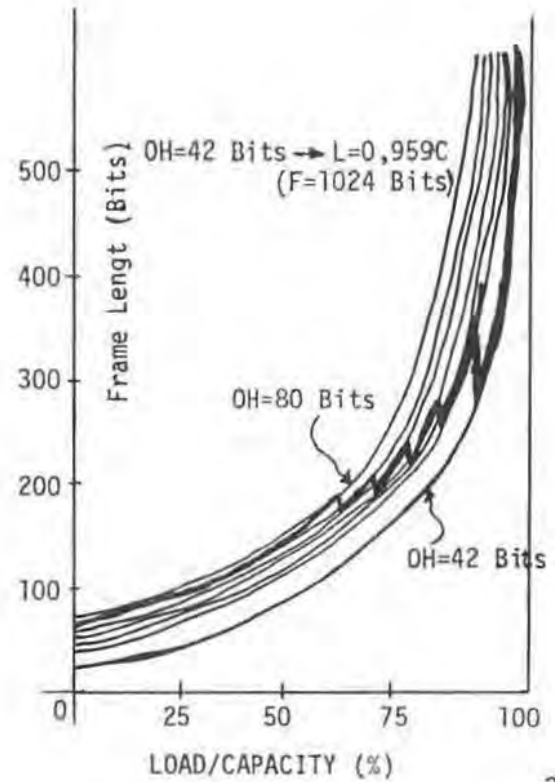
$OH = (1 - L/C) F$

$$F = (OH) (C) / (C - L)$$

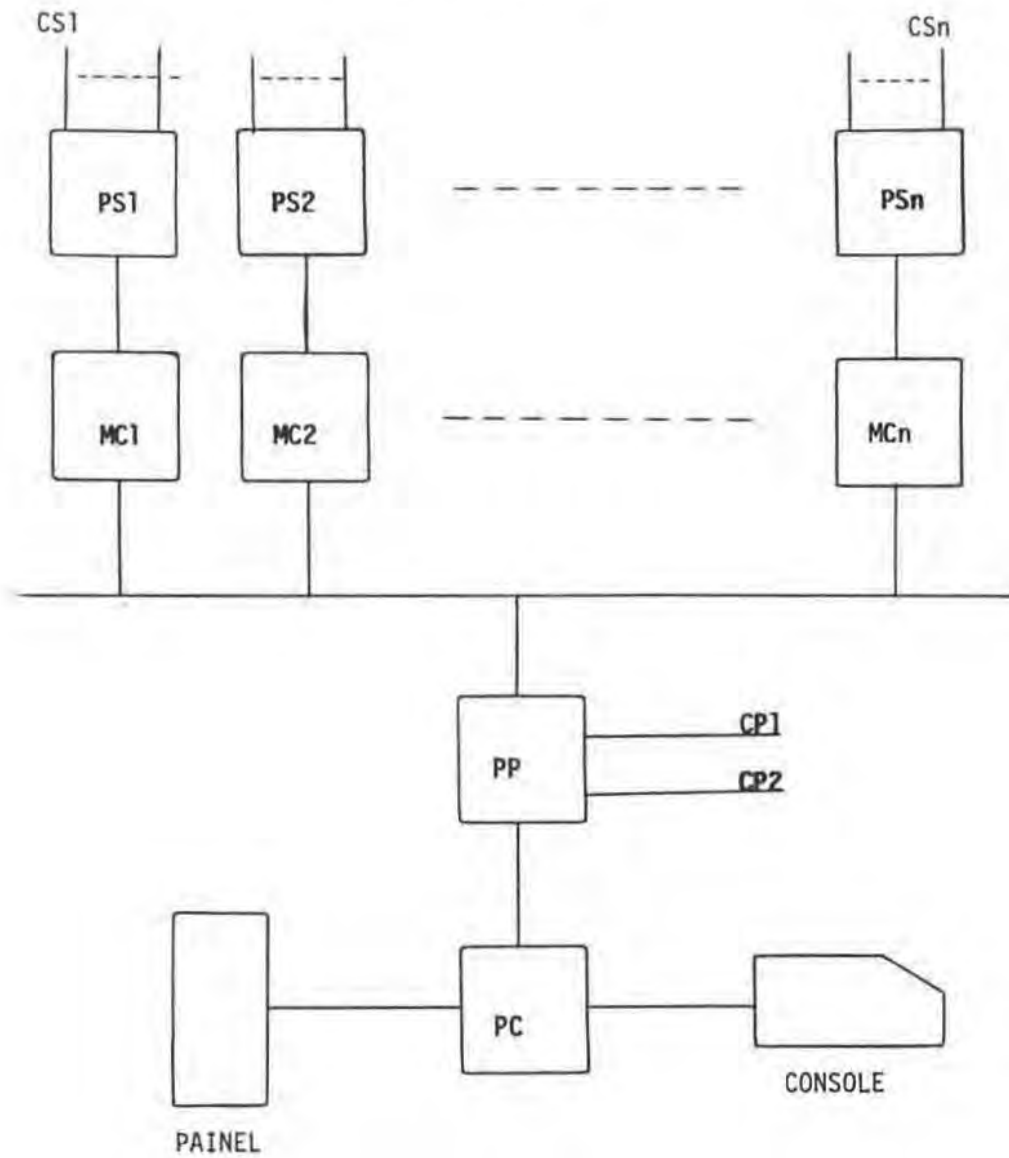
CURVAS



CONCLUSÃO: **STATMUX**
PODE UTILIZAR 95,9%
DO CANAL DE COMUNICAÇÕES

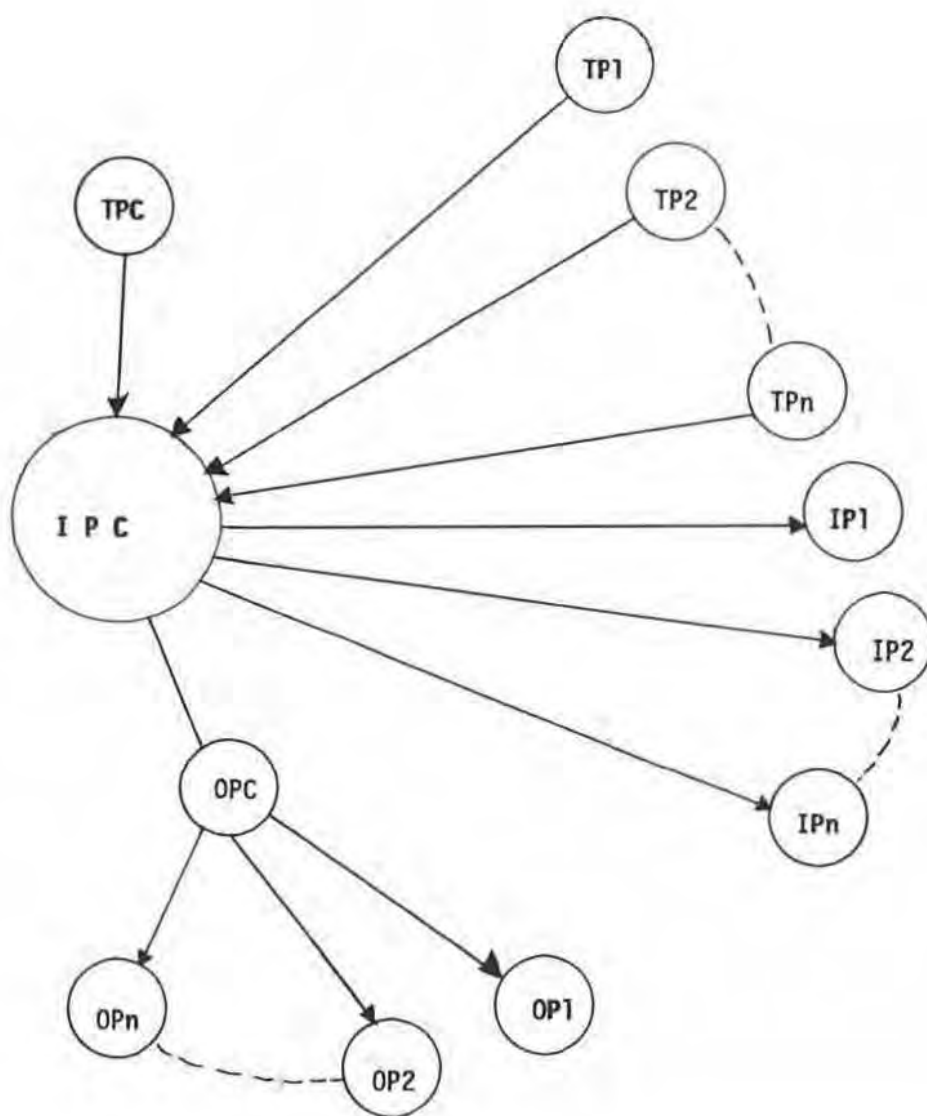


STATMUX - HARDWARE

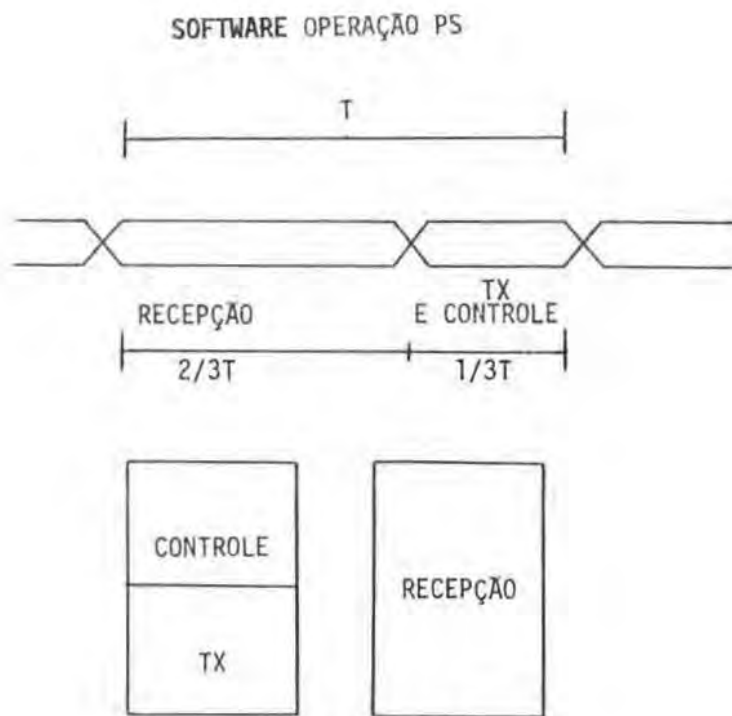


- PS - Processador Secundário
- PP - Processador Principal
- PC - Processador Console
- MC - Memória Compartilhada
- CP - Canal Principal
- CS - Canal Secundário

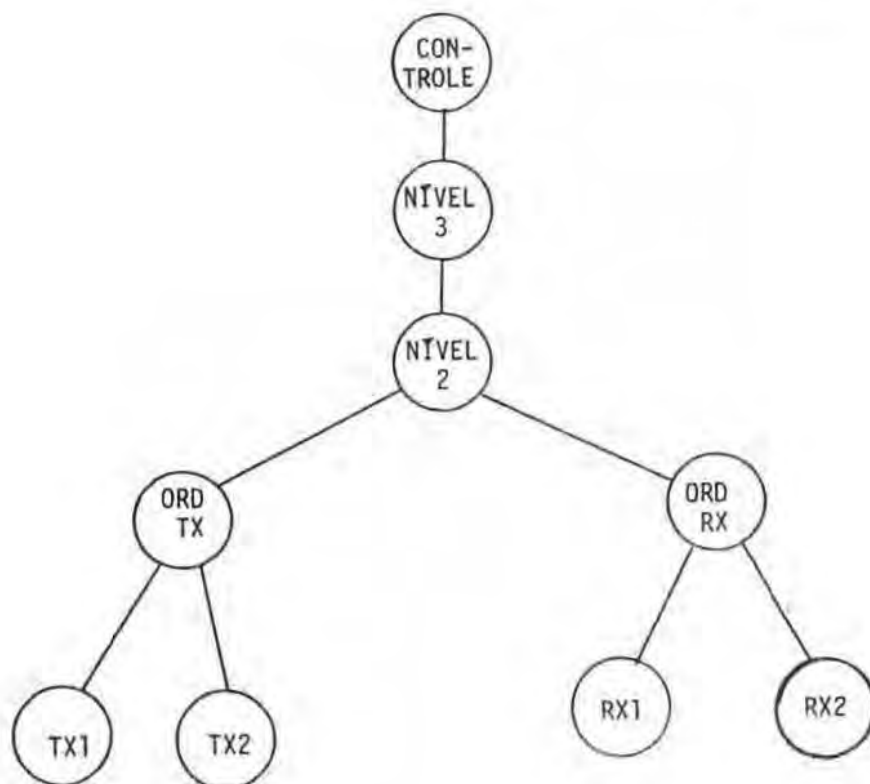
SOFTWARE - OPERAÇÃO GERAL



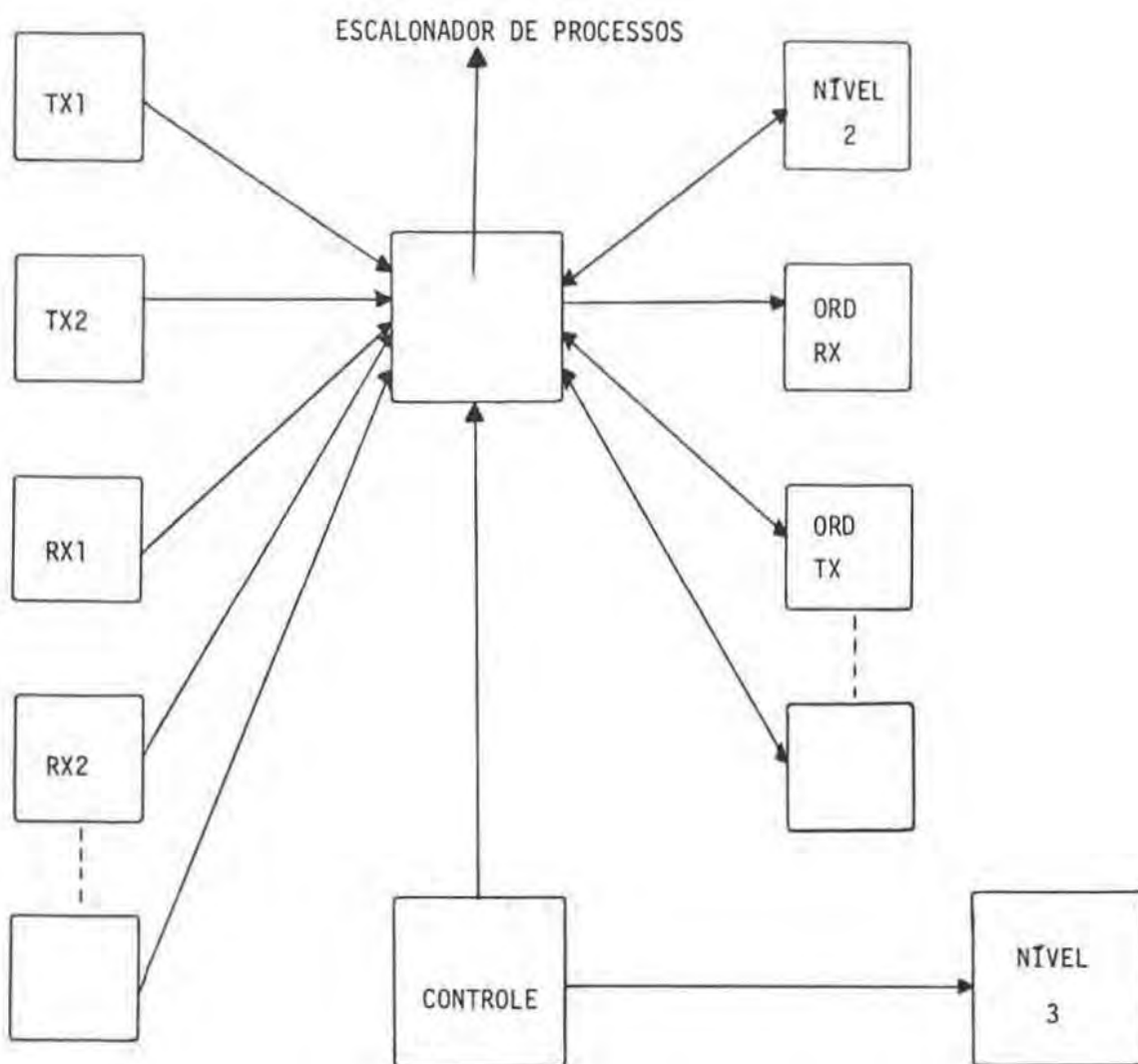
TP - Teste Processador
IP - Inicialização Processador
OP - Operação Processador



SOFTWARE PP - DEFINIÇÃO LÓGICA



SOFTWARE PP - IMPLEMENTAÇÃO



Teletex em Rede de Pacotes

Roberto Gadelha Pinheiro

José Roberto Emiliano Leite

Rosana Jamal Francisco dos Santos

Centro de Pesquisa e Desenvolvimento da Telebrás

Campinas - SP

I. Introdução

O Serviço Telex, por maior que seja sua importância e penetração, não está realmente em sintonia com o atual estágio da tecnologia digital.

A fim de oferecer uma alternativa ao Telex, mais condizente com o atual estado da tecnologia, O CCITT especificou o Serviço Teletex.

A introdução do Serviço Teletex no Brasil está sendo analisada com muito cuidado pelo Grupo Telebrás para que se garanta realmente o interesse nacional em seus amplos aspectos. O CCITT, apesar de ser um órgão internacional ligado à CNU, não deixa de refletir necessidades de sociedades mais desenvolvidas que a brasileira. Quando e como soluções preconizadas pelo CCITT, tal como o Serviço Teletex, devem ser introduzidas no Brasil devem ser objeto de uma decisão consciente de todas implicações, principalmente as que dizem respeito à dependência externa, seja tecnológica, seja financeira.

A empresa do Grupo Telebrás que estuda a viabilidade da introdução do Serviço Teletex no Brasil é a Embratel. O Programa de Comunicação de Dados do CPqD tem subsidiado a Embratel com estudos sobre alguns aspectos. O presente trabalho é resultado dessa interação entre CPqD e Embratel.

TELEBRAS

O objetivo específico desse trabalho é caracterizar o problema de introdução do Serviço Teletex tendo-se como rede de suporte uma rede de dados com comutação de pacotes. Em suas seções iniciais, o trabalho fornece também uma visão geral sobre o Serviço Teletex.

II. Serviço Teletex

O Serviço Teletex está destinado a ser uma das ferramentas fundamentais na automatização de escritórios, agilizando a preparação, manipulação, armazenamento e distribuição de documentos.

Mais concretamente, o Serviço Teletex permite terminais trocarem correspondência memória a memória, em modo automático, através de rede pública de telecomunicações.

O Serviço Teletex pode ser oferecido em cima de rede telefônica ou rede de dados com comutação de circuito ou rede de dados com comutação de pacotes. A comunicação entre terminais Teletex é realizada em cima da rede suporte sem qualquer modificação nessa rede.

A Fig. 2.1 ilustra a interação entre terminais Teletex tendo-se como suporte uma rede de pacotes.

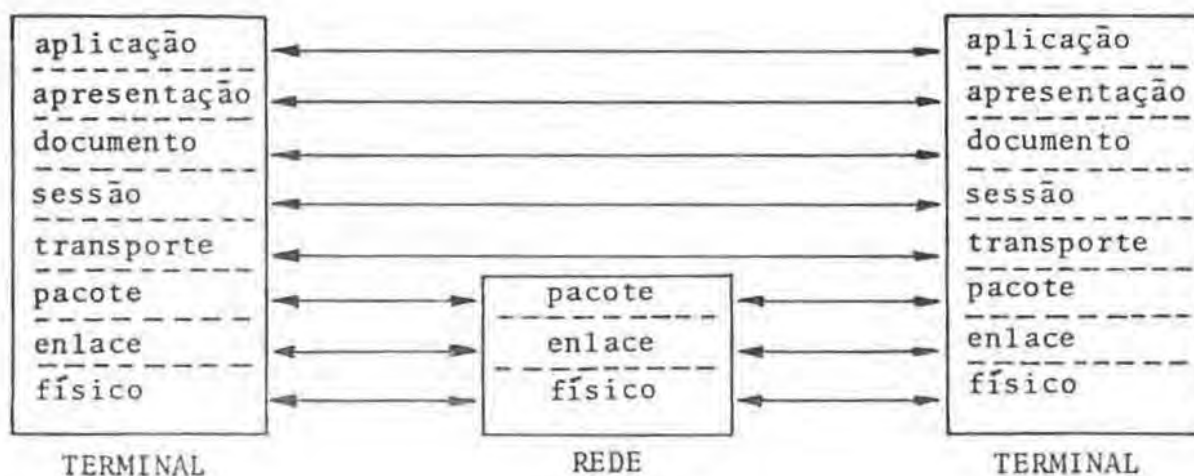


Fig. 2.1

Ligação Teletex via Rede de Pacotes

A Fig. 2.1 mostra as camadas de protocolo especificadas pelo CCITT.

As três camadas mais básicas (físico, enlace e pacote) são regidas pela Recomendação X.25 e idênticas às de qualquer outro terminal da rede, seja Teletex ou não. São camadas dependentes da rede em oposição às camadas acima que são totalmente independentes de rede.

Existe um nível básico de compatibilidade entre todos os terminais Teletex nas camadas acima da rede. Níveis mais sofisticados só são adotados após uma negociação em tempo real entre os terminais em comunicação.

A camada de transporte é definida pela Recomendação S.70.

O nível básico da camada de transporte, denominada classe zero, é extremamente simples. Possui apenas blocos de: conexão, rejeição de blocos e dados de transporte.

Como não existe explicitamente a desconexão de transporte, a duração da conexão de transporte está diretamente relacionada com a duração da conexão de rede. Não existe mecanismo de recuperação de erro. Ao detetar erros, o serviço de transporte apenas informa ao nível superior sobre o fato ocorrido.

As camadas de sessão e documento são definidas pela Recomendação S.62.

A camada de sessão tem por finalidade estabelecer uma conexão lógica entre dois terminais Teletex. Durante a fase de conexão da sessão são negociadas as facilidades opcionais a serem utilizadas durante a sessão: características do terminal (espaço entre caracteres ou linhas não "default"), negociação de janela de recepção e transmissão, além do serviço a ser utilizado (por exemplo, interfuncionamento com Telex).

A camada de documento tem por finalidade controlar a transmissão de um texto por meio de um mecanismo de controle de fluxo a nível de página. Através deste controle, podemos interromper a transmissão de um documento e continuá-la posteriormente em outra sessão, pedir para o receptor descartar o texto recebido ou definir o conjunto de caracteres específico do texto.

TELEBRÁS

A camada de apresentação é definida pelas Recomendações S.60 e S.61.

Mais de que um protocolo, trata-se da caracterização de formatos e caracteres válidos no Serviço Teletex e da especificação das características básicas do terminal Teletex.

A S.60 define as características básicas e opcionais do terminal: velocidade de transmissão (2400 bps), transmissão duplex, formato da página A4.

A intercomunicação com terminais Telex também faz parte do escopo da S.60: ela define o tamanho da linha (69 caracteres) e a tabela de conversão de caracteres do repertório Teletex para o repertório Telex (ITA 2).

A S.61 define o repertório de caracteres do Teletex que inclui letras maiúsculas, minúsculas, todos os tipos de acento, caracteres especiais, sinais financeiros, funções de controle, etc.

Em resumo, o Serviço Teletex, enquanto uma interação entre terminais Teletex, consiste essencialmente de uma interação fim-a-fim entre terminais. Nesse escopo, nada teria que se fazer em uma rede (telefônica, pacotes ou circuito) para implantar o Serviço Teletex. Requisitos, entretanto, de interfuncionamento com o Serviço Telex complicam a situação.

III. Interfuncionamento com Telex

É um requisito importante e fundamental que o Serviço Teletex deva interagir, em ambas as direções, com o Serviço Telex através de uma Facilidade de Conversão.

O terminal Teletex é implicitamente um terminal inteligente: processa protocolos complexos e se comunica a 2400 bps. Em oposição, o terminal Telex é um terminal não inteligente, operando a 50 bauds.

TELEBRÁS

O CCITT exige que na interação entre um terminal Teletex e um terminal Telex, o terminal Teletex utilize suas funções básicas, cabendo a uma Facilidade de Conversão a tarefa de "explicar" tudo para o terminal Telex.

Essa Facilidade de Conversão seria instalada e operada pela empresa que oferece o Serviço Teletex, sendo o único recurso público explicitamente associado ao Serviço Teletex.

A Facilidade de Conversão seria um equipamento dotado de capacidade de processamento e armazenamento capaz de simular a operação básica de um determinado número (tipicamente entre 16 e 64) de terminais Teletex.

A Fig. 3.1 ilustra a operação de uma Facilidade de Conversão.

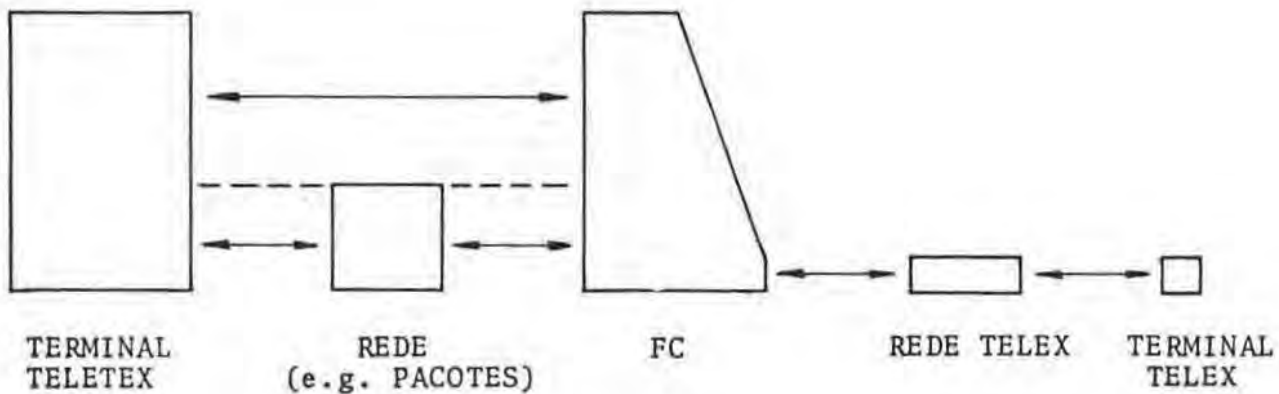


Fig. 3.1

Interfuncionamento com Telex via Facilidade de Conversão (FC)

TELEBRÁS

A função básica da Facilidade de Conversão (FC) seria a de dialogar nas camadas transporte, sessão, etc com o terminal Teletex, e em padrão Telex com o terminal Telex. Além disso, para que o terminal Teletex possa enviar/receber em seu throughput habitual, torna-se necessário um armazenamento temporário na Facilidade de Conversão.

No sentido do terminal Teletex para o terminal Telex, a mensagem é armazenada rapidamente na FC, o terminal Teletex é liberado e a mensagem é transferida lentamente para o terminal Telex.

No sentido do terminal Telex para o terminal Teletex, a mensagem é armazenada lentamente na FC, a mensagem é enviada para o terminal Teletex e ambos terminais são liberados.

A Fig. 3.2 ilustra a conversão de throughput através da FC na interação Teletex/Telex.

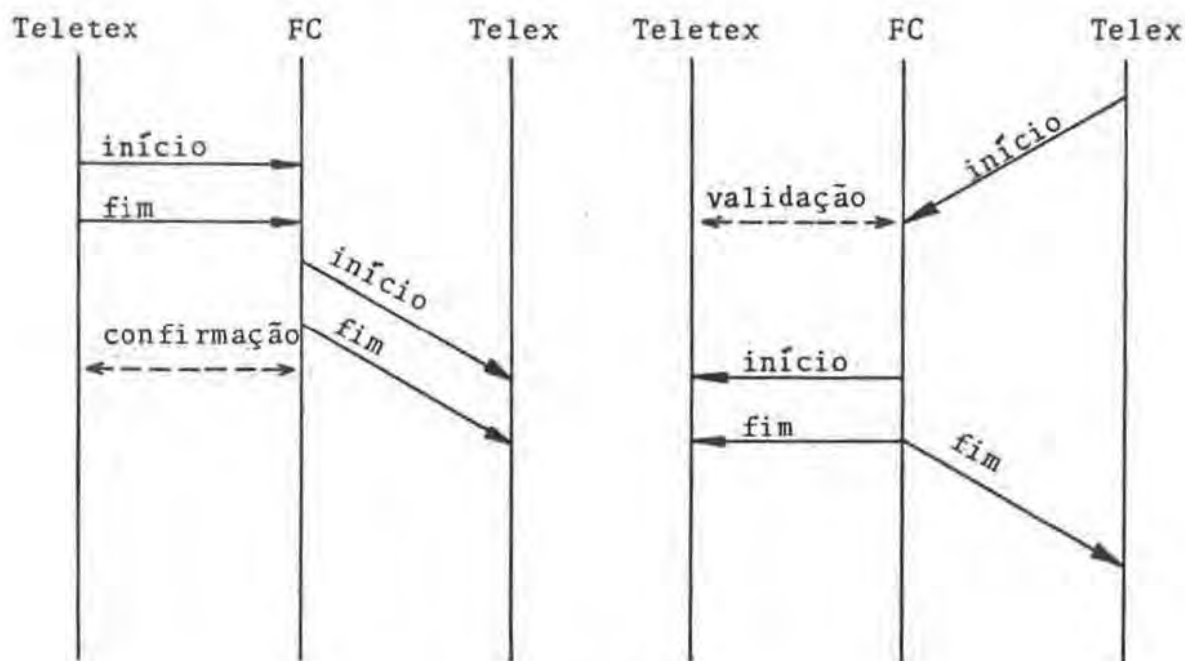


Fig. 3.2

Conversão de Throughput via FC

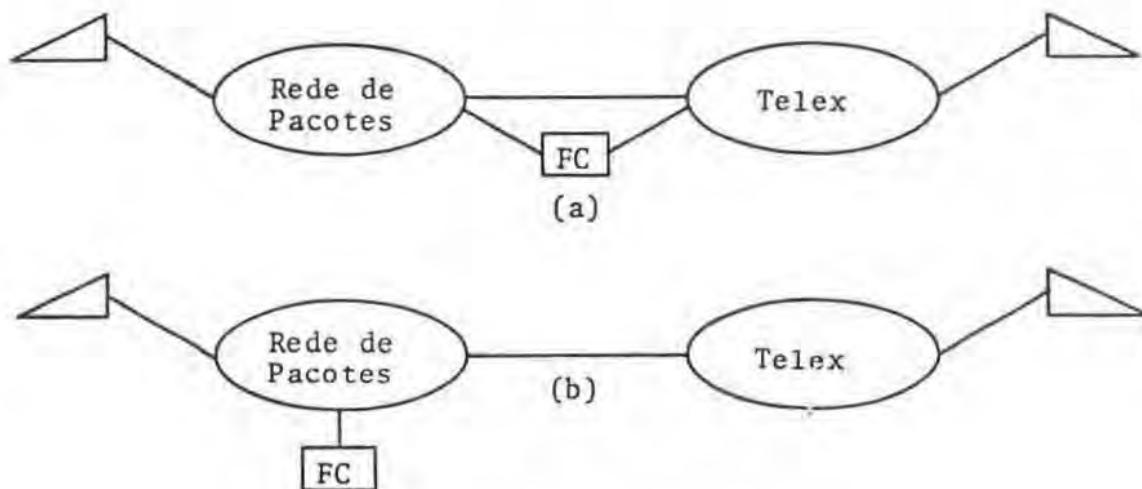
Obs: ligações opcionais em tracejado

IV. Facilidade de Conversão

Para o desenvolvimento e a implantação de uma Facilidade de Conversão em uma rede de pacotes a primeira etapa é especificar sua operação e requisitos necessários.

Esse trabalho deve ser realizado a partir de algumas hipóteses.

Em primeiro lugar, é conveniente tomar uma decisão quanto à localização da FC relativa às redes de pacotes e Telex. Há duas alternativas, como mostra a Fig. 4.1.



Na Fig. 4.1 (a) tem-se uma FC que interfaceia ambas as redes. Por exemplo, a ligação com a rede de pacotes seria em X.25 enquanto a ligação com a rede Telex poderia ser feita conforme a interface X.28.

Um terminal Teletex ao desejar se comunicar com um terminal Telex discaria o número Telex precedido de um prefixo específico que resultaria em um circuito virtual entre o terminal Teletex e a FC.

A FC receberia a mensagem e logo após a encaminharia ao terminal Telex destinatário.

TELEBRAS

No sentido do Telex para o Teletex, o terminal Telex discaria um número Telex correspondente ao acesso à FC. A FC atenderia a essa chamada, receberia a mensagem e o respectivo endereço destinatário. A seguir, enviaria a mensagem para o terminal Teletex.

A desvantagem desse método é o interfaceamento da FC com duas redes. Essa desvantagem pode ser eliminada aproveitando-se uma interface já existente entre a rede de pacotes e a rede Telex. Essa é a situação mostrada na Fig. 4.1 (b).

Agora a FC é única e exclusivamente um ETD (equipamento terminal de dados) da rede de pacotes. A única interface de comunicação dessa ETD seria uma interface X.25 com a rede de pacotes.

Na alternativa (b) há uma diferença em relação à alternativa (a) digna de nota. O assinante Telex, após acessar a rede de pacotes através de um PAD, deve solicitar conexão com a FC e não com o terminal Teletex destinatário. Esse procedimento não seria convencional de modo que o acesso internacional a terminais Teletex deveria ser feito pela rede de pacotes e não pela rede Telex a fim de manter a compatibilidade com normas do CCITT.

Além de caracterizar a localização da FC é interessante definir o computador que suportará a FC.

Para essa tarefa é necessário estimar:

- a) demanda de conversões em cada sentido
- b) número típico de páginas associado a cada conversão
- c) quantidade de software residente
- d) quantidade de recursos consumidos na conversão de uma página típica

A fim de se dar prosseguimento à linha de raciocínio desse trabalho será necessário manipular alguns números que devem ser tomados meramente como ilustrativos. Quando necessário, utiliza-se parâmetros compatíveis com os dos mini computadores nacionais existentes no mercado.

TELEBRÁS

O software associado a enlace (X.25 nível 2) ou, se for o caso, a interface X.28, seria colocado em placas especiais. O software residente na memória principal seria, com alguma folga, o seguinte:

X.25 nível 3	:	20K
S.70 transporte	:	10K
S.62 sessão	:	20K
S.62 documento	:	20K
interface disco	:	20K
operação e tarifação	:	<u>50K</u>
		140Kbytes

A quantidade de recursos consumidos por página pode ser estimada sob três aspectos:

- i) recursos de comunicação
- ii) recursos de processamento (tempo de CPU)
- iii) recursos de armazenamento (memória e tempo de acesso)

Em uma ligação a 9600 bps, o tempo de ocupação de uma página é de aproximadamente 1 segundo. É viável a utilização de até 6 (seis) linhas síncronas a 9600 bps em paralelo.

A quantidade de instruções de programa a que é submetido uma página pode ser estimada em 15000 instruções incluindo entrada e saída. Assumindo-se um tempo médio de instrução de 2 μ sg, a ocupação de CPU associada a uma página é de 30 msg. Em outras palavras, a CPU seria capaz de processar cerca de 30 páginas por segundo.

A área ocupada por uma página em disco seria de 2kbytes. Supondo-se uma taxa de entrada de 1 página/seg e permanência de 10 minutos, a capacidade necessária em disco seria de 1,2 Mbytes.

O tempo de acesso a um disco (30 msg de procura e 8 msg de latência) por uma página pode ser estimado em 150 msg. Como há a escrita e a leitura, para cada disco há uma capacidade máxima de trabalhar com 3 páginas por segundo.

Um mini computador pode ser configurado tipicamente com até 8 (oito) discos de 80 Mbytes.

TELEBRAS

A partir dessas alternativas chega-se à conclusão de que para situações em que a demanda de conversão é da ordem de 1 (uma) chamada por segundo, a cada chamada correspondendo tipicamente 1 (uma) página, a FC pode ser perfeitamente implementada tendo-se por base um minicomputador nacional.

V) Conclusão

O Serviço Teletex pode ser suportado por uma rede telefônica, por uma rede de pacotes ou uma rede de circuitos.

Esse trabalho não procurou provar que a alternativa pacote é melhor que as demais. O objetivo do trabalho foi simplesmente mostrar que a implantação do Serviço Teletex em uma rede de pacotes é perfeitamente possível tecnicamente.

Sob nosso ponto de vista, a decisão de implantação do Serviço Teletex em uma ou outra rede é uma questão política e econômica que deve ser analisada dentro do âmbito de cada realidade nacional sem se deixar influenciar por pressões externas.

A discussão em torno da rede mais indicada para suportar o Serviço Teletex não deve em hipótese alguma ofuscar os enormes problemas associados ao terminal Teletex.

Muitas indústrias nacionais têm condições de participar desse mercado promissor desde que haja uma implantação bem planejada. Qualquer pressa iria beneficiar enormemente grupos multinacionais que acompanham o fenômeno Teletex desde suas origens e estando, portanto, já prontos para lançar seus produtos no mercado.

VIDEOTEXTO: Desenvolvimento de Terminais de Usuários

Ricardo Menna Barreto Felizzola

Av. Santos Dumont, 1791

Bairro São Geraldo

90.000-Porto Alegre-RS

RESUMO

O texto aborda o estágio atual de implantação do Sistema Videotexto no Brasil pela TELESP (Telecomunicação de São Paulo S.A.). São introduzidos os conceitos básicos dos processos de atualização e acesso ao banco de dados do Sistema via rede telefônica, bem como são abordadas as características principais de Terminais de Fornecedores de Serviço e Terminais de Usuário.

Ao desenvolvimento de terminais com características especiais para o mercado nacional é dedicada uma ênfase salientando-se aspectos de arquitetura, custo e atualização tecnológica.

Por fim, conclusões de mercado e política de utilização do Sistema são enfocadas permitindo a situação no "estágio da arte" atual.

I - O QUE É VIDEOTEXTO?

II - QUAIS AS CARACTERÍSTICAS DE UM PROJETO DE DESENVOLVIMENTO DESTES TERMINAIS?

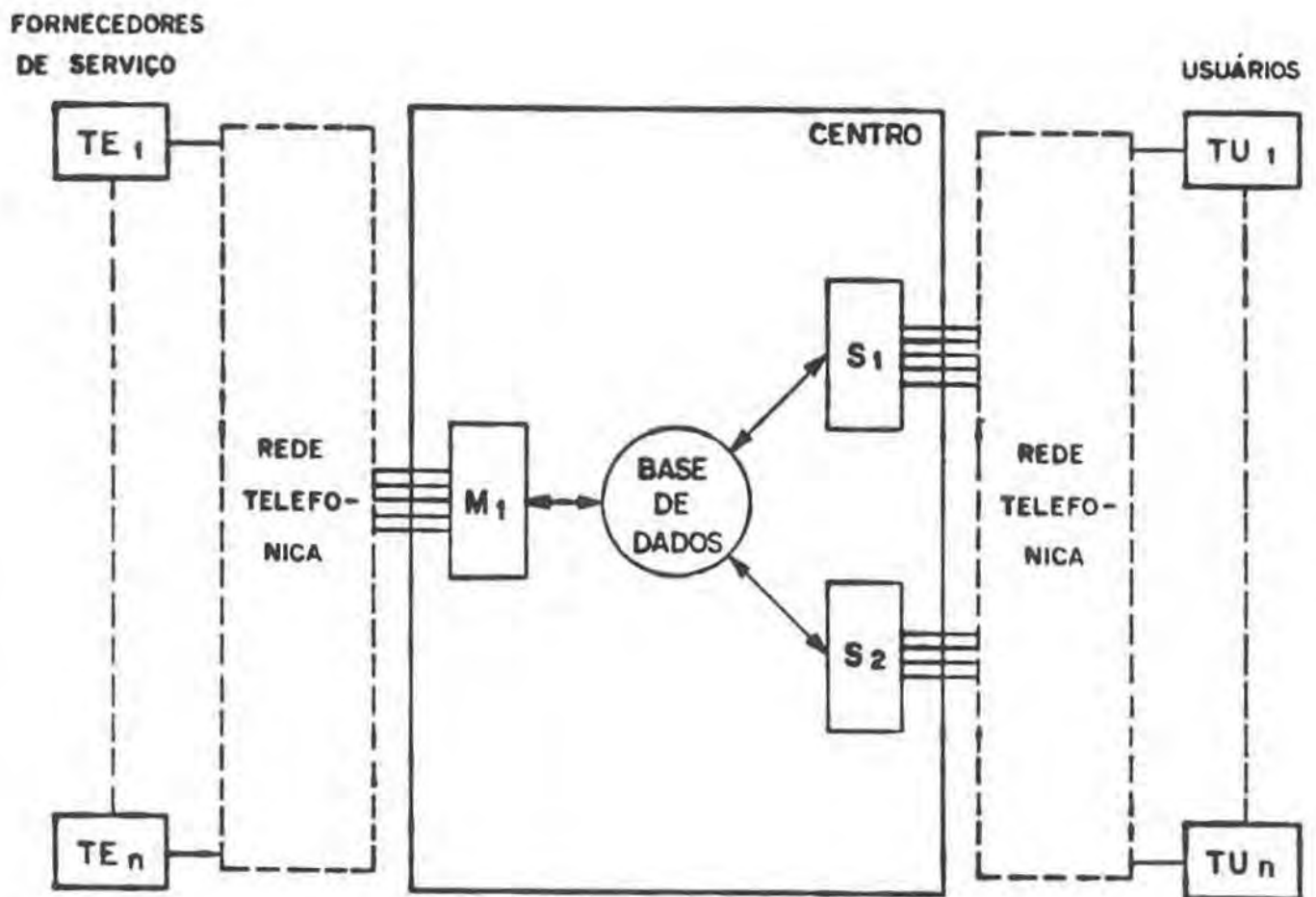
III - COMO O PROJETO ESTÁ CONFIGURADO?

IV - É DAQUI PARA FRENTE?

I. O QUE É VIDEOTEXTO?

I.A. CARACTERÍSTICAS BÁSICA DA REDE

- CENTRAL VIDEOTEXTO;
- TERMINAIS DE FORNECEDORES;
- TERMINAIS DE USUÁRIOS;
- UTILIZAÇÃO DA LINHA TELEFÔNICA;



M_1, S_1, S_2 : COMPUTADORES
 $TU_1 - TU_n$: TERMINAIS DE USUÁRIO
 $TE_1 - TE_n$: TERMINAIS DE EDITAGEM

Figura 2.1 - Blocodiagrama do Sistema Videotexto

I.B. OPERAÇÃO DO TERMINAL DE USUÁRIO

- CONEXÃO A CENTRAL;
- ENVIO DA 1ª PÁGINA;
- MENU DE OPÇÕES;
- PÁGINAS ALFA-GRÁFICAS;
- PÁGINAS ESPECIAIS (D.R.C.S.);
- TELESOFTWARE.

II. QUAIS AS CARACTERÍSTICAS DE UM PROJETO DE DESENVOLVIMENTO DESTES TERMINAIS?

II.A. OBJETIVOS PRINCIPAIS;

- BAIXO CUSTO;
- BAIXO CUSTO;
- BAIXO CUSTO;

II.B. OUTROS OBJETIVOS:

- ENGENHARIA DE PRODUTO VOLTADO AO MERCADO DE CONSUMO;
- EXPANSÃO DO TERMINAL VISANDO AUMENTO DE VIDA DO PROJETO.

II.C. MODELOS

- VDT-I (ESCRITÓRIO);
- VDT-R (RESIDENCIAIS);

II.D. MÓDULOS DE EXPANSÃO.

- IMPRESSORA;
- MICROCOMPUTADOR;

III. COMO O PROJETO ESTÁ CONFIGURADO?

III.1. HARDWARE

- PLACA LÓGICA CONTENDO:
 - 1 C.P.U. (8039);
 - CONTROLADOR DE VÍDEO;
 - INTERFACE P/ K7;
 - U.A.R.T;
- MODEM V.23
- PLACAS ACESSÓRIAS:
 - RECEPTOR DE INFRA-VERMELHO;
 - MONITOR DE VÍDEO;
 - MICROCOMPUTADOR (Z-80);
 - MODULADOR PAL;

III.2. SOFTWARE

- MÓDULO DE CONTROLE;
- MÓDULO DE COMUNICAÇÕES;

IV. E DAQUI PARA FRENTE?

IV.1. TECNOLOGIA

- PRODUTO DE CONSUMO;
- VLSI DEDICADOS;
- SOFTWARE COM CARACTERÍSTICAS PECULIARES PARA AS APLICAÇÕES NO PAÍS.

IV.2. MERCADO

- NOVA MÍDIA DE COMUNICAÇÕES;
- A REDE JA ESTÁ COMPLETAMENTE INSTALADA

PROGRAMA DE INFORMÁTICA DO
RIO GRANDE DO SUL

BADESUL - Banco de Desenvolvimento
do Estado do Rio Grande do Sul

Econ. Antonio Ernani Martins Lima
Eng. Newton Braga Rosa

Porto Alegre, maio de 1983.

1. INTRODUÇÃO

O BADESUL é um Banco de Desenvolvimento integrante do sistema financeiro nacional, com sede em Porto Alegre, cuja atuação compreende a área do Estado do Rio Grande do Sul.

O seu objetivo é proporcionar o suprimento adequado dos recursos financeiros necessários ao financiamento, a médio e longo prazo, de programas e projetos que visam a promover o desenvolvimento econômico e social do Rio Grande do Sul, cabendo-lhe apoiar, prioritariamente, o setor privado.

As linhas de crédito e os programas operados pelo BADESUL abrangem os seguintes objetivos de apoio financeiro:

- a) elaboração de projetos de viabilidade;
- b) desenvolvimento tecnológico;
- c) investimentos fixos na lavoura e na pecuária;
- d) investimentos fixos industriais e comerciais;
- e) realocização de empresas;
- f) aumento das exportações;
- g) capitalização empresarial;
- h) capital de giro;
- i) saneamento financeiro;
- j) substituição de combustíveis derivados do petróleo.

O BADESUL pode também efetuar, exclusiva e privadamente, operações de arrendamento mercantil contratadas com o próprio vendedor dos bens ou com as pessoas jurídicas a ele vinculadas (lease back).

A prestação de garantias sob a modalidade de aval ou fiança se constitui outra alternativa de apoio financeiro do BADESUL a empresas interessadas em implantar ou expandir as suas atividades econô-

2. O Programa de Informática do Estado

2.1. Objetivos e Características

Complementarmente, o Banco recebeu da Secretaria de Coordenação e Planejamento a atribuição de apoiar e coordenar a implantação do Pólo Eletroeletrônico no Estado do Rio Grande do Sul, tendo eleito

como principal prioridade a consolidação e o desenvolvimento das empresas industriais e de prestação de serviços, vinculadas ao setor de informática.

Neste sentido, foi elaborado, no decorrer de 1982, o "Programa de Informática do Rio Grande do Sul", que buscou reunir e sistematizar as vantagens locacionais, financeiras e econômicas oferecidas pelo Estado para a implantação e/ou expansão de empresas do setor, através de:

- identificação de novas oportunidades de mercado para as empresas locais;
- atração de novos investimentos para o Rio Grande do Sul;
- concessão de apoio financeiro e institucional às empresas localizadas no Estado, visando sua consolidação e expansão;
- orientação de investimentos em áreas de alta tecnologia consideradas prioritárias.

A justificativa central para a implantação de um programa de desenvolvimento da indústria de informática no Estado do Rio Grande do Sul está baseada na crescente importância assumida por este setor no aumento da produtividade e da competitividade industrial e econômica em geral, assim como decorre da própria continuidade natural do processo de desenvolvimento histórico da economia gaúcha.

Um setor industrial e de serviços de informática se constitui em excelente oportunidade de diversificação de sua atividade econômica, cuja evolução histórica tem se pautado pelo constante processo de agregação tecnológica.

Em termos financeiros, este programa se justifica pela possibilidade de aplicação de recursos otimizada, programada e coerente com a política nacional para o setor de informática.

Destaca-se igualmente a capacidade e a infraestrutura do BADESUL.

para a detecção de oportunidades de mercado e vocações locais de produção de equipamentos e serviços, bem como para a execução e o acompanhamento de projetos de informática.

2.2. Plano de Aplicação para 1983/84

Muito embora ações de fomento ao setor, por parte do Governo Estadual, tenha se desenvolvido desde 1979, o Programa de Informática do Estado do Rio Grande do Sul possibilitou a consolidação dos objetivos e a formalização destas ações de forma integrada e programada.

Iniciou-se então diversos contatos com o objetivo de obter a adesão ao Programa de organismos oficiais de desenvolvimento econômico e tecnológico, de forma a assegurar o fluxo dos recursos financeiros necessários à execução dos investimentos programados.

Neste sentido, obteve-se do BNDES o enquadramento do Programa no âmbito do POC (Programa de Operações Conjuntas), o que representa, em última análise, o enquadramento automático de projetos do setor, pelo próprio BADESUL. Este apoio inclui a concessão de créditos às indústrias de informática e de seus insumos, destinados à elaboração de projetos, investimentos fixos e mistos, realocização, fortalecimento financeiro e modernização.

O BADESUL obteve, também, junto à FINEP, o enquadramento global do Programa, o que significa, além da sensível agilização do processo de análise, a garantia de recursos financeiros para o apoio creditícios aos projetos que visem o desenvolvimento de tecnologia.

Considerados unicamente os dois organismos referidos, o Programa dispõe, através do BADESUL, de recursos dotados da ordem de Cr\$ 2,2 bilhões para apoio a projetos, no decorrer dos exercícios de 1983 e 1984, com possibilidades de ampliação, conforme identificação da demanda.

No tocante ao aspecto operacional do programa, o BADESUL destaca duas formas principais de atuação: Financeira e Institucional.

Esta ordem de colocação não significa a de prioridade, tudo depende dos objetivos que o empresário pretende atingir e das características de sua empresa.

3. Sistemática Operacional - apoio financeiro

Em relação ao apoio financeiro, o BADESUL, trabalha repassando recursos de diversas entidades federais ou com recursos próprios. Estes recursos são denominados linhas de crédito.

Cada linha de crédito é constituída de forma a atender determinados objetivos específicos.

No setor de informática estes objetivos, na maioria dos casos são:

- Investimentos Fixos:
 - Obras Cíveis
 - Equipamentos
 - Produção, Pesquisa e Desenvolvimento; Controle e dualidade.
 - Software
- Capital de Giro
- Saneamento Financeiro
- Elaboração de Projetos
- Desenvolvimento de Tecnologia

As principais linhas de crédito são:

1. POC - Programa de Operações Conjuntas:

Linha de Crédito do BNDES que financia investimentos fixos, capital de giro, saneamento financeiro, elaboração de projetos e desenvolvimento tecnológico.

2. FINAME - Especial e Longo Prazo - Linha de crédito da Agência Especial de Financiamentos Industrial que financia compra de equipamentos nacionais.

3. Convênio BNDES/SEI/DIGIBRÁS

Linha de crédito que financia o desenvolvimento de software. Be

neficia Software e Houses, Produtoras de Hardware e Usuários Finais.

4. Convênio CEF/DIGIBRAS

Linha de crédito que financia capital de giro com recursos da Caixa Econômica Federal.

5. ADTEN - Apoio ao Desenvolvimento Tecnológico da Empresa Nacional

Linha de crédito do FINEP que financia desenvolvimento de novos produtos, montagem de laboratórios de pesquisa e implantação ou ampliação de sistemas de controle de qualidade.

6. PROMICRO/CEBRAE

Linha de crédito que financia investimentos fixos, capital de giro e desenvolvimento tecnológico de microempresas.

2.4. Sistemática Operacional - apoio institucional

Referente ao apoio Institucional. o BADESUL vem realizando:

- apoio a constituição de novas empresas no setor;
- atração de novos investimentos para a informática;
- realização de encontros e debates;
- contatos com órgãos federais, como SEI, DIGIBRAS, BNDES, FINEP, etc, para defender interesses específicos ou coletivos das empresas do setor.
- orientação de investimentos em áreas de alta tecnologia consideradas prioritárias.

A forma de beneficiar-se de todo este leque de alternativas de apoio consiste em, definidos os objetivos a serem alcançados pela empresa, o empresário procurar o Departamento de Planejamento do BADESUL (coordenação do Programa de Informática) para discutirmos, em carácter sigiloso, as características do projeto e qual a alternativa mais adequada.

3. Discussão de alguns resultados do Programa de Informática do RS

Para uma melhor compreensão da atuação do BADESUL através de seu

Programa de Informática, apresentamos a seguir, aos senhores, uma breve exposição dos resultados já obtidos.

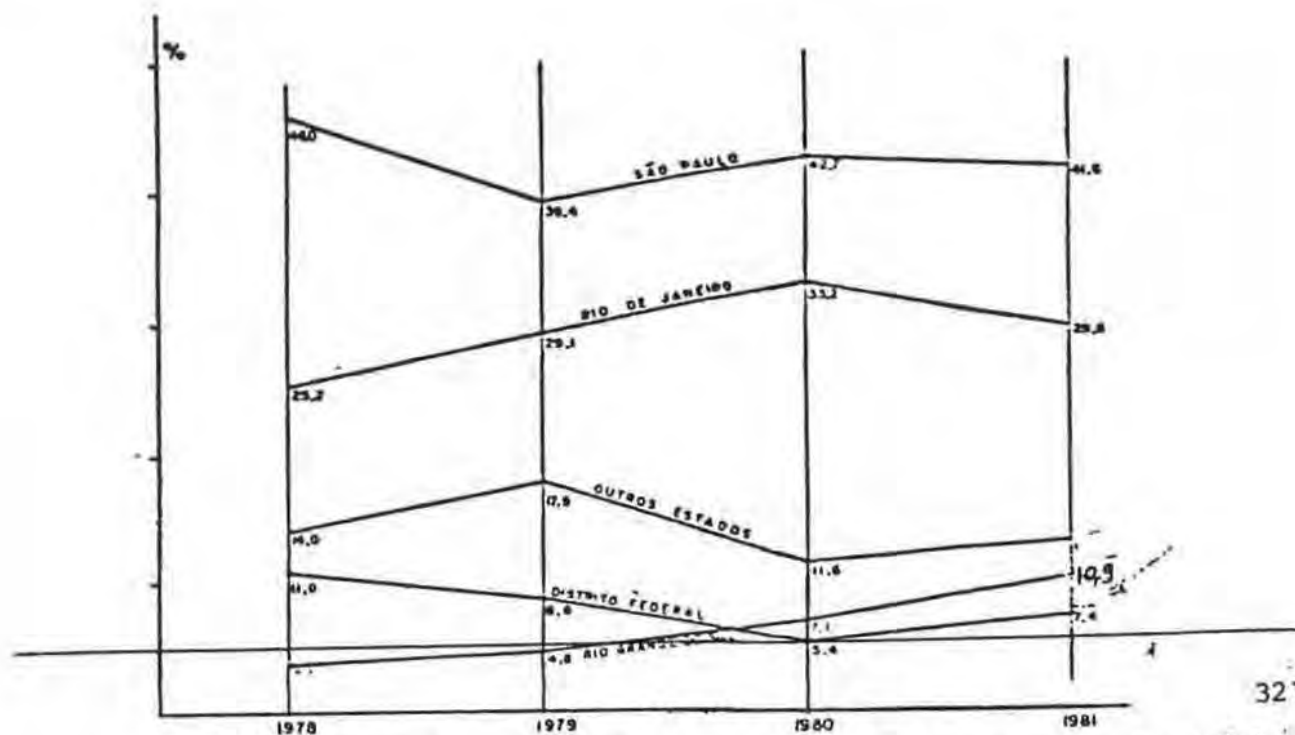
O fato da informática no Estado e mesmo no Brasil ser uma atividade nova, torna difícil avaliar o seu desempenho de 1982 pela série histórica de dados.

Entretanto, o cotejo dos dados da Informática do RS com os da informática no Brasil e com outros segmentos da Economia do próprio RS, são bastante elucidativos como mostram os gráficos abaixo.

O gráfico seguinte mostra o desempenho do faturamento da informática no RS, face a outros estados. Embora não seja o maior, o faturamento do RS, é o que mais está crescendo e isto de uma forma que parece segura e regular. Parece portanto razoável a previsão de aumento da participação em 82 em mais 1,5% do total nacional, atingindo 12%. Entretanto, a confirmação deste dado só será possível em julho, quando a DIGIBRÁS estará lançando a edição 82 do "Panorama da Indústria Nacional de Computadores e Periféricos".

Gráfico 1-DISTRIBUIÇÃO GEOGRÁFICA DAS VENDAS
PERCENTUAL DO FATURAMENTO

16

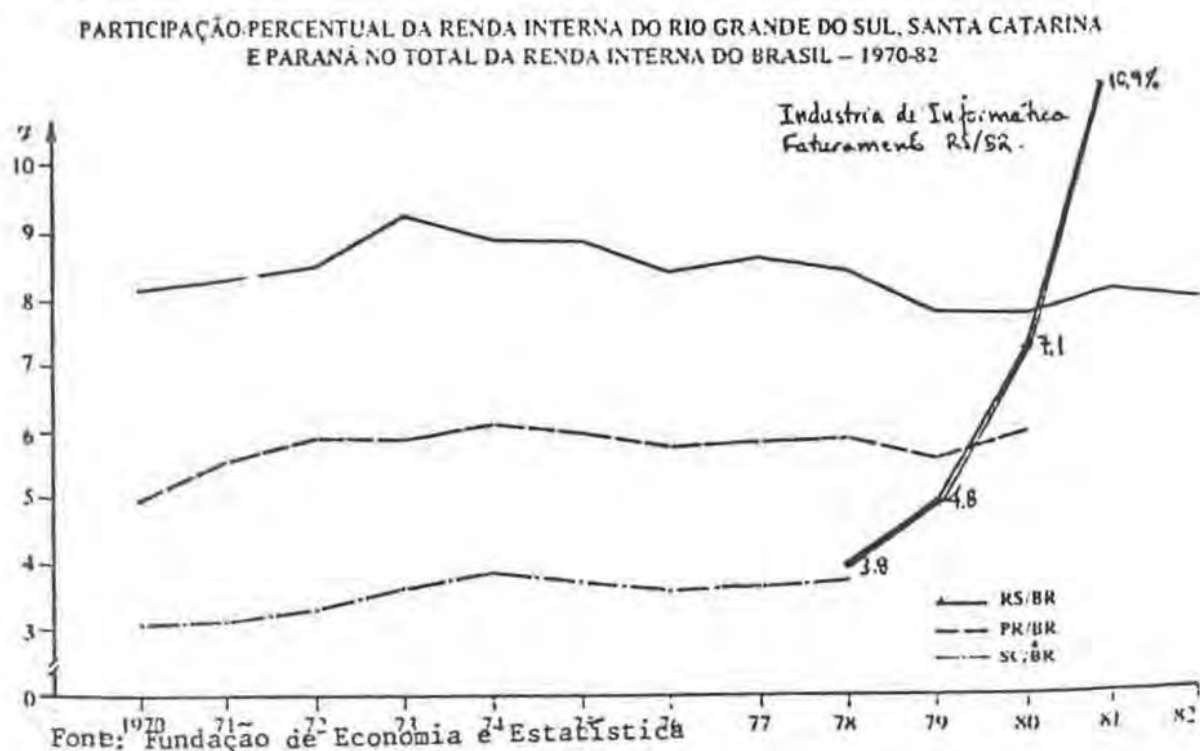


327

Fonte: Panorama da indústria nacional de Computadores e Periféricos

Mais surpreendentes, entretanto, são os dois gráficos abaixo. O gráfico 2 mostra a participação percentual da renda interna do RS na renda interna do Brasil como um todo e o equivalente para a Informática. O RS com cerca de 7,5% do Produto Interno Bruto representou, em 1981, quase 11% do faturamento da indústria nacional. Este índice parece tender a crescer.

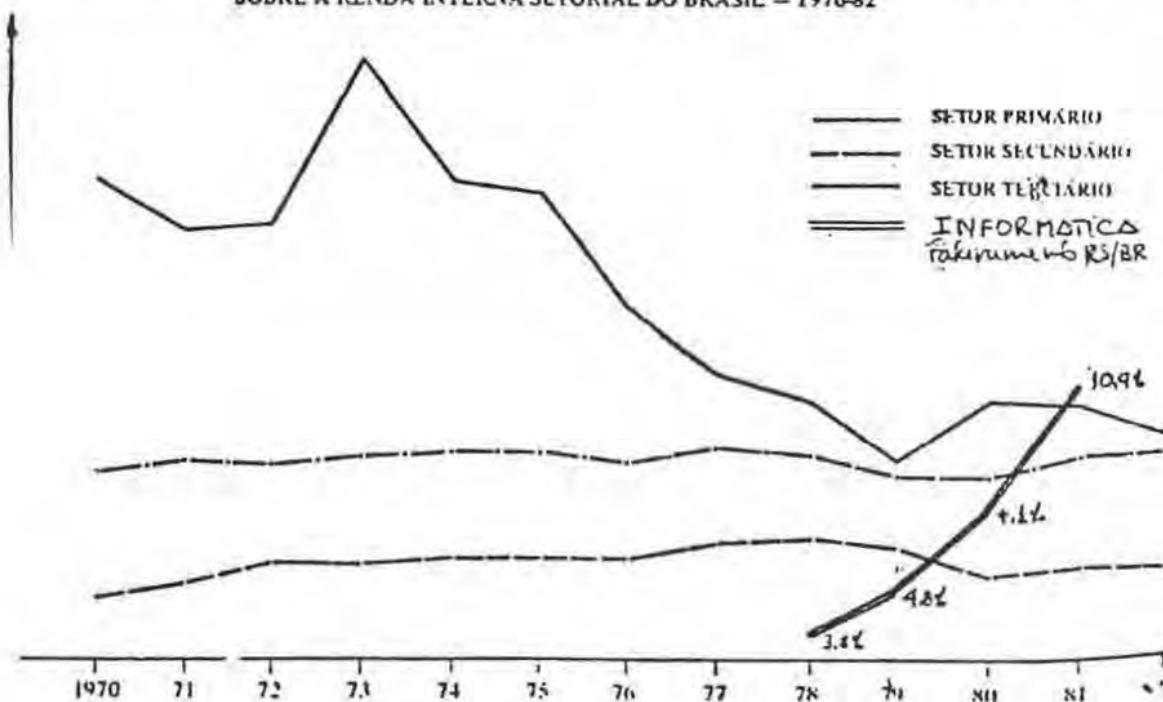
Gráfico 2



O gráfico 3 mostra que indústria de Informática do RS já superou, em importância relativa, a participação dos setores primários, secundário e terciário na renda interna estadual.

Gráfico 3

PARTICIPAÇÃO PERCENTUAL DA RENDA INTERNA SETORIAL DO RIO GRANDE DO SUL
SÓBRE A RENDA INTERNA SETORIAL DO BRASIL - 1970-82



Fonte: Fundação de Economia e Estatística

estima-se em 23.000 o número de empregos diretos oferecidos pelas 30 empresas do Rio Grande do Sul que atuam no setor. O faturamento global estimado para 1982 situou-se em torno de 40 bilhões de cruzeiros.

Atividades Desenvolvidas:

O apoio à constituição de novas empresas no setor, o que identifica a preocupação do Banco em não se ater exclusivamente aos aspectos tecnológicos, mas também aos aspectos mercadológicos e políticos do setor. Apesar da conjuntura econômica desfavorável, foram criadas 18 novas empresas no Rio Grande do Sul em 1982, algumas delas fabricando produtos muito próximos ao estado da arte mundial;

A atração de novos investimentos para a informática também mereceu a atenção do BADESUL, que veiculou os resultados e as perspectivas do setor às empresas de outros setores da economia do Estado, identificadas como possuidoras de um potencial para diversificação de investimentos. Alguns acordos importantes já foram selados entre empresas do setor metal-mecânico e de informática para fabricação de equipamentos de automação controle de processos;

- No plano creditício, o BADESUL aprovou, em 1982, projetos destinados a inversões fixas à suplementação de capital de giro, apresentados por empresas do setor de informática, no montante de Cr\$ 674 milhões. As liberações no mesmo período totalizaram Cr\$ 337 milhões, contemplando 9 empresas do setor;
- O Rio Grande do Sul conquistou uma posição de destaque nacional ao tornar-se 3º Estado brasileiro na produção de equipamentos para a informática, representando, segundo dados da própria DIGIBRÁS, 11% da produção nacional;
- Neste 1983 estamos promovendo a aproximação do setor de informática com o de mecânica de precisão visando utilizar a participação mais efetiva do Estado na produção de equipamentos para automação. Neste contexto caberá um importante papel ao CETEMP - Centro Tecnológico de Mecânica de Precisão, do SENAI, inaugurado na semana passada.

Na continuação das atividades realizadas, estabeleceu-se uma programação de trabalhos a serem desenvolvidos no ano de 1983, que compreende:

- Redefinição do detalhamento e do montante dos investimentos a serem realizados e previstas pelas empresas, ao nível de intenções, por ocasião da elaboração do Programa;
- Formalização participação do BNDES no aporte de recursos ao Programa através da assinatura de Contrato de Abertura de Crédito - CAC;
- Captação dos projetos finais elaborados pelas empresas e a serem avaliados a nível de análise;
- Operacionalização dos processos de análise, aprovação, contratação e liberação de recursos para os projetos captados;
- Estabelecimento de diretrizes de atuação futura com vistas a assegurar a continuidade e ampliação dos objetivos fixados pelo Programa.

Esperamos, através desta e de outras iniciativas no âmbito do Programa de Informática do Rio Grande do Sul dar a parcela de contribuição do Estado à consolidação e expansão deste novo e estratégico setor da economia nacional.

OS SISTEMAS DE INFORMAÇÕES

E A EMBRATEL

JORGE LUIZ CHARBEL
DEPT. DE COORDENAÇÃO COMERCIAL

I N D I C E

1. Introdução	333
2. Os segmentos dos "Sistemas de Informações" para a consulta a Bancos de Dados.	333
3. O Papel da EMBRATEL no apoio a "Sistemas de Informações".	335
4. Sistemas de Informações - Uma visão a nível nacional	338
4.1. Em âmbito público	338
4.2. Em âmbito restrito	342
5. Telecomunicações - Alternativas para apoio a Sistemas de Infor- mações.	342

1. INTRODUÇÃO

É um fato indiscutível que, no Brasil, os Sistemas de Informações estão se desenvolvendo em acelerada progressão, a exemplo do que vem ocorrendo em outros países.

Informação, Computador e Telecomunicações são os alicerces dos sistemas de informações.

A EMBRATEL como empresa responsável pela Comunicação de Dados no País, vem envidando os maiores esforços em prol do desenvolvimento dessa nova área de atividades.

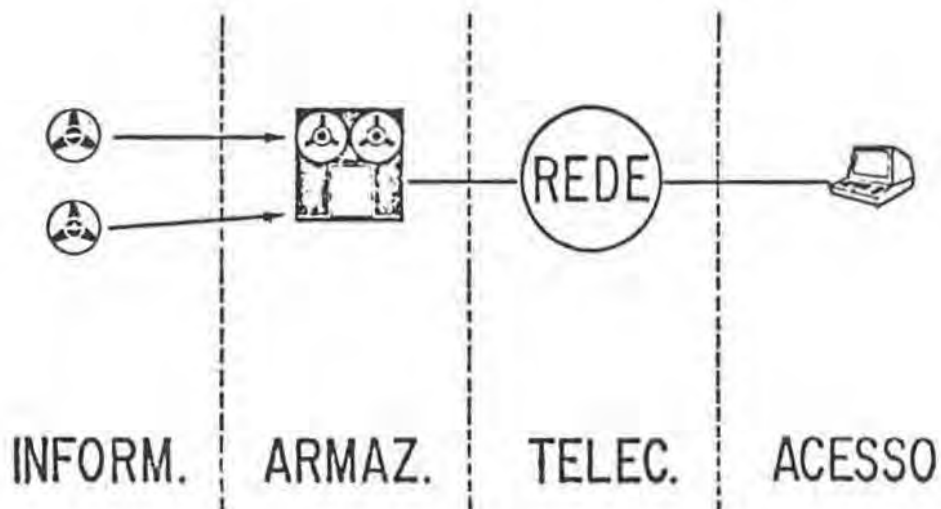
O objetivo do presente trabalho é apresentar a postura e os serviços oferecidos pela EMBRATEL, e citar alguns exemplos de resultados concretos de desenvolvimento de Sistemas de Informações.

2. OS SEGMENTOS DOS "SISTEMAS DE INFORMAÇÕES" PARA CONSULTA A BANCOS DE DADOS

Para o desenvolvimento de Sistemas de Informações, a participação de empresas com diferentes atividades torna-se necessário no sentido de planejamento das tarefas relacionadas, desde o usuário final até a informação armazenada num sistema eletrônico de dados.

Daí surge a necessidade da definição dos segmentos para os serviços de Consulta a Bases de Dados.

Basicamente resumem-se em 04 (quatro) segmentos conforme mostra a figura abaixo:



- O segmento INFORMAÇÃO, se refere as empresas responsáveis na elaboração de informações, no sentido de colocá-las disponíveis para acesso público ou restrito a uma coletividade, ou seja, consiste na produção e estruturação dos dados.

Como resultado final do segmento de Informações tem-se as BASES DE DADOS, mantidas atualizadas e prontas a serem consultadas.

Exemplificando podemos citar a SEADE (Fundação Sistema Estadual de Análise de Dados), empresa responsável pelo desenvolvimento do Sistema SIM (Sistema de Informações Municipais do Estado de São Paulo); a EDITORA BOLETIM DE CUSTOS, responsável por informações contidas no SBC (Sistema Boletim de Custos) abrangendo toda a área de construção civil.

- O segmento ARMAZENAMENTO/PROCESSAMENTO, define os responsáveis pelo tratamento das informações (arquivar, atualizar, processar) elaboradas pelos fornecedores de informação. No Brasil podemos exemplificar algumas empresas como:

- . SERPRO (Serviço Federal de Processamento de Dados), responsável por projetos tais como POLVO, ARUANDA, IMPOSTO

DE RENDA etc...

- . PROBAM (Processamento de Dados Bancários) responsáveis por informações referentes a área bancária.
- . PRODASEM (Processamento de Dados do Senado Federal), responsável por informações da área governamental.
- O Segmento TELECOMUNICAÇÕES, é um monopólio governamental, sendo prestado pelo Sistema TELEBRÁS, tendo a EMBRATEL como empresa responsável pelos Sistemas de Comunicações de Dados e Telex, e as Empresas Polos (Concessionárias Locais) responsáveis pela Rede Telefônica.
- O segmento ACESSO completaria o ciclo, definido os responsáveis no desenvolvimento de TERMINAIS (Vídeo, Teleimpressores) para acesso às informações. Neste caso o recurso oferecido é o Terminal de Dados, que permite ao usuário final alcançar, instantaneamente, desde o local onde vive ou trabalha, informações selecionadas ou meios de tratá-las. Embora no Brasil já exista uma grande variedade de equipamentos, a sua utilização para os vários Sistemas de Informações, principalmente a nível público, ainda é bastante difícil, pelo elevado custo de fornecimento dos terminais para os usuários finais.
Uma boa alternativa está surgindo no mercado através do serviço VIDEOTEXTO, que basicamente é um terminal de dados para uso doméstico.

3. O PAPEL DA EMBRATEL NO APOIO A "SISTEMAS DE INFORMAÇÕES"

A existência de grandes iniciativas no Brasil em termos de, CONSULTAS À BASE DE DADOS E APLICAÇÕES EM TIMESHARING, levou a EMBRATEL a ter um papel muito importante na participação dos trabalhos que estão sendo desenvolvidos no país. Disto resulta a enumerar-mos resumidamente o papel da EMBRA

TEL no apoio a SISTEMAS DE INFORMAÇÕES, no seguinte:

a) Incentivar iniciativas de terceiros

Ou seja, procurar apoiar, informar, desenvolver junto, Sistemas de Informações que possam ser divulgados para um grande número de usuários, procurando usar um meio de Te_lecomunicações mais rápido e com menor custo.

b) Fornecer Serviços de Telecomunicações

Sendo o objetivo fim da empresa, procurar os seus servi_ços, como TRANSDATA, TELEX, REPARTE, e futuramente a REN_PAC, na conexão com os Sistemas de Informações existentes no país.

c) Divulgar os Serviços Existentes

Atualmente a EMBRATEL tem uma participação mais concreta, procurando divulgar estes sistemas por diversos meios de informações, tais como:

- . Folhetos - onde trabalhando em conjunto com o responsá_{vel} pelo "Sistema de Informação", divulga através de suas Regiões e Distritos folhetos referentes ao serviço que está sendo implantado.
- . Artigos em revistas - informa através de notícia, ou ar_tigos descritos por elementos pertencentes ao seu qua_{dro} de pessoal.
- . Guia de Bancos de Dados - sentindo a necessidade impos_{ta} pelo mercado, de não existir um documento oficial

sobre "Sistemas de Informações" a nível nacional, a EMBRATEL desenvolveu um documento denominado "GUIA NACIONAL DE BANCOS DE DADOS".

Este guia, para cada sistema descrito, informa os nomes das BASES DE DADOS que o constituem, o TIPO DE ACESSO, a FORMA DE ACESSO, a REDE DE COMUNICAÇÃO utilizada.

A sua atualização é feita pela EMBRATEL, e para participar no documento, basta contactar o órgão comercial mais próximo da EMBRATEL e descrever as informações referentes ao sistema.

d) Estudo de novos Serviços facilitando a divulgação de informações

Procurando ainda mais facilitar a divulgação de informações, a EMBRATEL vem participando no desenvolvimento de novos serviços na Área de Teleinformática, visando principalmente facilitar o usuário final, na pesquisa e divulgação de informações.

Nesse particular, encontra-se em estudos na empresa serviços como TELETEX, TELEMENSAGENS POR COMPUTADOR, etc., que em breve virão a se tornar novas alternativas para o usuário final na pesquisa e divulgação de informações.

e) A participação em empreendimentos conjuntos

Procurando ainda mais facilitar os Serviços de Informática, abrangendo a diferentes tipos de mercado, como setor bancário, setor comercial e de empresas que tradicionalmente operam com informações, a EMBRATEL vem participando conjuntamente na implantação desses Sistemas, nas funções de armazenamento, faturamento e cobrança principalmente

para Sistemas conectados a sua Rede Nacional de Telex. Co
mo exemplo podemos citar:

- Sistema SIM (Sistema de Informações Municipais do Esta
do de São Paulo)
- Sistema SBC (Sistema Boletim de Custos)
- Sistema ARUANDA
- Sistema TELCOMM

4. SISTEMAS DE INFORMAÇÕES - Uma Visão a Nível Nacional

Apresentaremos neste tópico uma visão geral dos Sistemas de
Informações desenvolvidos no Brasil, identificando alguns
sistemas de acesso público, e outros de acesso restrito.

4.1. EM ÂMBITO PÚBLICO:

a) SISTEMA DE INFORMAÇÕES MUNICIPAIS E DADOS ESTATÍSTI COS - SIM

- Informações contidas nas Bases de Dados
 - . Dados sobre o Estado de São Paulo nos setores:
agropecuária; características físicas; comércio
exterior-câmbio; comunicações; construção civil;
contas nacionais; cultura; demografia etc.
- Como acessar as informações contidas no SIM:
 - . Acesso através da Rede Nacional de Telex, digi
tando-se o n. (019)1321+

b) SISTEMA DE INFORMAÇÃO ARUANDA

- Bases de Dados disponíveis

- . Indicadores de Desempenho da Economia Nacional - IDEN.
 - . Serviço Automático de Consultas para Intermediação de produtos e Serviços - SACI.
 - . Informações sobre a indústria extrativa mineral brasileira - IEM.
- Como acessar as informações contidas no SISTEMA DE INFORMAÇÕES ARUANDA:
- . Acesso através da Rede Nacional de Telex, digitando-se os n. (061)2446+ ou (01131080+.

c) SISTEMA BOLETIM DE CUSTOS - SBC

- Informações contidas nas Bases de Dados: Construção Civil, abrangendo:
- . Tabelas de gabaritos, contendo custos de obras por m² em diversos municípios;
 - . Tabelas de custos básicos dos produtos que constituem os gabaritos, atingindo diversos municípios;
 - . Tabelas contendo os insumos correspondentes a cada composição de custos desejados;
 - . Tabelas contendo o custo unitário de cada produto desejado, em determinadas cidades.
- Como acessar as informações contidas no SBC:
- . Acesso através da Rede Nacional de Telex, digitando-se o n. (021)30730+.

d) SISTEMA DA BOLSA DE VALORES DE SÃO PAULO

- Informações contidas nas Bases de Dados:
 - . BOLSA DE VALORES DE SÃO PAULO:
Informações sobre o mercado de ações, nas Bolsas de Valores de São Paulo, Extremo Sul, Minas Gerais, Espírito Santo e Brasília.
 - . BOLSA DE MERCADORIAS DE SÃO PAULO:
Informações sobre Cotações de negócios da Bolsa de Mercadoria de São Paulo.
 - . BANCO CENTRAL DO BRASIL:
Informações sobre Cotações de câmbio de moedas, taxas de mercado aberto, notícias do mercado financeiro.
- Como acessar as informações contidas no Sistema da Bolsa de Valores de São Paulo:
 - . Acesso através da Rede Nacional de Telex, digitando-se o n. (011) 21458+, ou através de LP's próprias mediante entendimento com a BOVESPA.

e) SISTEMA DA BOLSA DE VALORES DO RIO DE JANEIRO

- Informações contidas nas Bases de Dados:
 - . Cotações de títulos e seus vencimentos
 - . Volumes negociados
 - . Boletins de fechamento e Índice de Lucratividade e preços.
 - . Mercado à Vista, Futuro, Termo e Opções.
- Como acessar as informações contidas no Sistema da

Bolsa de Valores do Rio de Janeiro:

Acesso através da Rede Nacional de Telex, digi-
tando-se o n. (021)33737+.

f) SISTEMA DE INFORMAÇÕES SOBRE MERCADORIAS - TELCOMM

- Informações contidas na Base de Dados:

. Cotações Nacionais e Internacionais de Commodity
ties (Mercadorias) das Bolsas de Nova Iorque,
Chicago, Londres, Paris e Brasil.

- Como acessar as informações contidas no TELCOMM:

. Acesso através da Rede Nacional de Telex, digi-
tando-se os números (011)33266+ ou (011)35027+
ou através de LP's próprias mediante entendimen-
to com a CMA-Engenharia de Sistemas.

g) SISTEMA DE INFORMAÇÃO "QUEM É QUEM" - GRUPO VISÃO

- Informações contidas nas Bases de Dados:

. Balanço e L & P de uma Empresa
. Índices de uma empresa e do Subsetor
. Índices de um Subsetor
. Definição dos Índices
. Lista de Subsetores

- Como acessar as informações contidas no Sistema
"Quem é Quem":

. Acesso através da Rede Nacional de Telex, digi-
tando-se o n. (011)36541+.

h) SISTEMA DA BIBLIOTECA REGIONAL DE MEDICINA

- Informações contidas nas Bases de Dados:

. MEDLINE:

Informações sobre referências bibliográficas de artigos da literatura biomédica mundial.

. IMLA:

Contém referências de artigos de literatura biomédica latino-americana.

- Como acessar as informações contidas no sistema:

. Através de LP's próprias, mediante entendimento com a Biblioteca Regional de Medicina (BIREME).

4.2. EM ÂMBITO RESTRITO:

a) SISTEMA DE INFORMAÇÃO DO BANCO NOROESTE DO ESTADO DE SÃO PAULO S.A.

- Bases de Dados disponíveis:

- . Cobrança de Títulos
- . Conta Corrente
- . Risco.

- Como acessar as informações contidas nas Bases de Dados:

. Acesso através da Rede Nacional de Telex, ou LP's próprias, mediante entendimento com o Banco Noroeste do Estado de São Paulo S.A.

b) SISTEMA DE INFORMAÇÃO DA EMPRESA DE PROCESSAMENTO DE DADOS DA PREVIDÊNCIA SOCIAL - DATAPREV

- Bases de Dados disponíveis:

- . Contribuições de Empresas
- . Codificações e Recodificações de Benefícios.

- Como acessar as informações contidas nas Bases de Dados:

- . Através da Rede Nacional de Telex, mediante entendimento com a DATAPREV.

c) SISTEMA DE INFORMAÇÃO DA PRODAM

- Bases de Dados disponíveis:

- . PMSP - Prefeitura Municipal de São Paulo
- . CMTC - Cia. Municipal de Transportes Coletivos.

- Como acessar as informações contidas nas Bases de Dados.

- . Através LP's próprias, mediante entendimento com PRODAM - Companhia de Processamento de Dados do Município de São Paulo.

5. TELECOMUNICAÇÕES - ALTERNATIVAS PARA APOIO A SISTEMAS DE INFORMAÇÕES

Nesta fase do trabalho vamos tecer breves comentários sobre alguns tipos de Serviços prestados pela EMBRATEL, enfatizando o seu uso na Área de Informática.

Procuramos com isto dar alternativas para os interessados na divulgação, rapidez de transmissão de informações contidas

num determinado equipamento de processamento de dados.

Os serviços, alguns já implantados, e outros em desenvolvimento na EMBRATEL, são os seguintes:

a - SERVIÇO ESPECIALIZADO DE COMUNICAÇÃO DE DADOS - TRANSDATA.

b - SERVIÇO DE COMUNICAÇÃO DE DADOS À DISTÂNCIA - CDD.

c - ALUGUEL DE INTERFACE TELEX PARA ACESSO A COMPUTADORES.

d - SERVIÇO DE COMUNICAÇÃO DE DADOS POR COMUTAÇÃO DE PACOTES.

e - SERVIÇO INTERNACIONAL DE COMUNICAÇÃO DE DADOS - INTERDATA.

a) SERVIÇO ESPECIALIZADO DE COMUNICAÇÃO DE DADOS - TRANSDATA

O Serviço TRANSDATA tem como finalidade básica possibilitar a transferência da informação de um ponto a outro, ou outros, em âmbito urbano ou interurbano, por meios de circuitos permanentemente conectados e especificamente projetados para comunicação de dados. Ou seja, trata-se de um serviço destinado à constituição de Redes Privativas de Dados.

Neste serviço, a EMBRATEL é a responsável pelo fornecimento, instalação e manutenção de todo o circuito de comunicação de dados, de MODEM, tendo o usuário apenas de conectar os seus Terminais e CPU's.

Atualmente, o serviço Transdata atinge a cerca de 280 localidades de atendimento pela EMBRATEL no país, sendo

suas tarifas cobradas em função da distância e da velocidade.

É um serviço que atende, de imediato, os setores cujas atividades são, a curto prazo, desenvolvidas com auxílio de processamento eletrônico da informação. Entre outros setores podemos citar:

- Indústria
- Transportes
- Finanças e Bancos
- Educação e Pesquisa
- Comércio
- Governo
- Serviços Públicos
- Serviços de Processamento de Dados

1) SERVIÇO DE COMUNICAÇÃO DE DADOS A DISTÂNCIA - CDD

É um serviço em que a EMBRATEL oferece todo o apoio técnico para a transmissão de dados através da Rede Telefônica, alugando os MODEM's para a conexão na linha telefônica do usuário e também em seu equipamento de processamento de dados. Se justificada por causa da grande penetração geográfica que tem a Rede de Telefonia, e também por sua grande simplicidade.

O seu custo é em função do uso, ou seja, de maneira semelhante a uma ligação telefônica.

Atualmente o serviço está disponível para as velocidades:

- 300 bps e 1200 bps - assíncrono
- 1200 e 2400 bps - síncrono

Como principais aplicações para utilização do CDD, podemos citar:

- . Consulta a Bases de Dados
- . Timesharing
- . Batch remoto (pequenos arquivos).

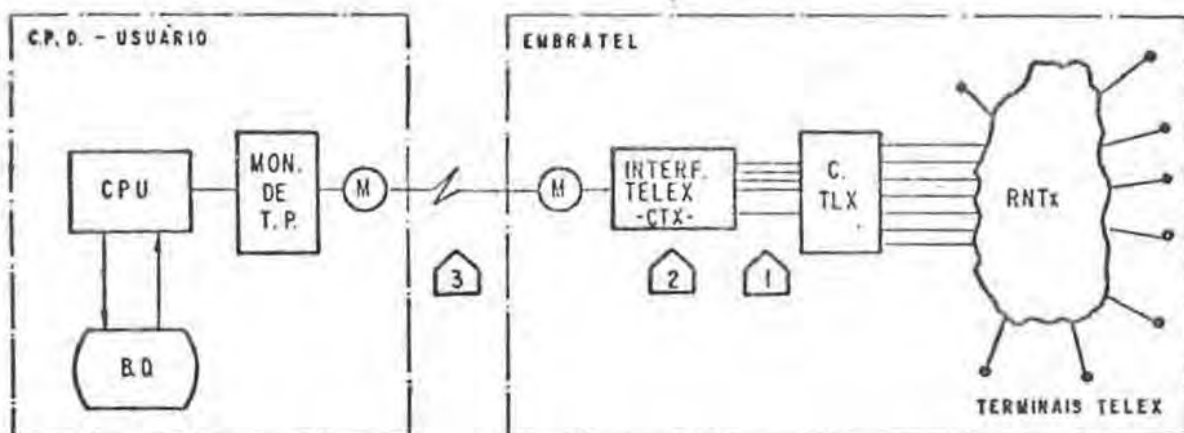
c) ALUGUEL DE INTERFACE TELEX PARA ACESSO A COMPUTADORES

É uma facilidade que permite a comunicação de dados entre a Rede Nacional de Telex (RTNx) e computadores.

Para que isto possa ser realizado o usuário terá que utilizar um equipamento denominado INTERFACE TELEX, onde do lado da Rede Telex funciona como um terminal telex, e do lado do computador como um terminal de vídeo.

Nesta facilidade prestada pela EMBRATEL, esta aluga as Terminações Telex, o Concentrador Telex e o Circuito Especializado Urbano ou Interurbano para conexão ao computador, já que o equipamento fica localizado na própria EMBRATEL.

Para uma maior visualização, apresentamos abaixo a sua configuração típica:



- 1 - Terminações Telex
- 2 - Concentrador Telex (CTX)
- 3 - Circuito Especializado (Veloc. de até 4800 bps).

Entre outras aplicações para utilização desta facilidade, podemos ressaltar:

- . Controle de estoques
- . Reservas diversas
- . Informações sobre áreas específicas:
 - Medicina
 - Engenharia
 - Agricultura
 - Economia, etc.

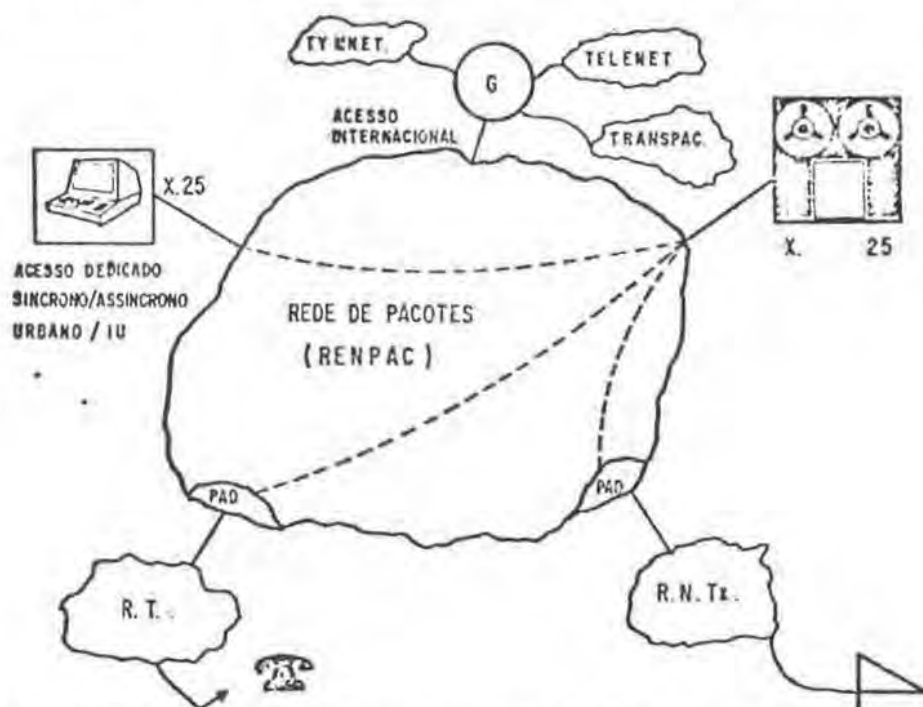
d) SERVIÇO PÚBLICO DE COMUNICAÇÃO DE DADOS POR COMUTAÇÃO DE PACOTES

A partir do 2º semestre de 1984, está prevista para entrada comercial a Rede Nacional de Comutação por Pacotes, projeto conduzido pela EMBRATEL, utilizando moderna tecnologia de comutação por pacotes, baseado em estruturas multimicroprocessadoras.

Com a entrada deste novo serviço, mais uma alternativa será oferecida para o usuário, para acessar computadores utilizando vários terminais, através de diferentes redes de comunicação (Telex, Telefonia, etc.).

A Rede de Pacotes está prevista para acessos dedicados ou comutados. No caso de acesso comutado a conexão poderá ser estabelecida por meio das redes públicas de telefonia e de telex, segundo procedimentos padronizados de discagem ao número de acesso à Rede de Pacotes. Para avaliar as possibilidades de acesso comutado, via rede telefônica e rede telex, estão previstos acessos públicos e acessos restritos. Na modalidade de acesso público haverá um número telex, único em âmbito nacional, e ou telefônico por localidade, para acesso de qualquer usuário.

A figura abaixo apresenta as várias modalidades de acesso à Rede de Pacotes, inclusive o acesso a redes de comunicação estrangeiras, através do serviço INTERDATA.



Esta rede oferecerá aos seus usuários um conjunto de facilidades opcionais, sendo algumas delas constantes do con

trato de prestação do serviço, e outras que poderão ser solicitadas cada vez que o assinante origina uma chamada (facilidades solicitadas por chamada). Além destas, estarão disponíveis aos usuários das classes assíncronas (telex, rede telefônica - de 300 a 1200 bps) as facilidades de PAD previstas pelo CCITT.

Dentre as facilidades contratadas estão:

- grupo fechado de assinantes
- grupo fechado com saída permitida
- grupo fechado com entrada permitida
- aceitação de tarifação reversa

Sob aspectos tarifários definidos pela administração responsável pela rede, concluiu-se que o usuário poderá ser cobrado em função dos seguintes pontos:

- Taxa de instalação:

Depende da estrutura adotada, sendo função de fatores como velocidade do acesso, distância do usuário ao ponto de acesso à rede e se o circuito é analógico/digital a 2 ou 4 fios, sendo cobrada uma única vez quando da solicitação do serviço.

- Taxa de acesso:

Depende dos mesmos fatores da taxa de instalação e representa o aluguel mensal do circuito de acesso e da porta da rede dedicada ao acesso.

- Canais lógicos adicionais:

Se solicitados serão cobrados.

- Taxa de utilização:

Onde estão incluídos os fatores TEMPO (duração da chamada e DISTÂNCIA (distância entre os assinantes chamado e chamador), mas, sendo o principal fator o VOLUME DE DADOS TRANSMITIDOS.

- Taxa de Facilidades opcionais:

Onde são consideradas as facilidades contratadas para cada canal lógico.

e) SERVIÇO INTERNACIONAL DE COMUNICAÇÃO DE DADOS - INTERDATA

A partir de DEZEMBRO/82, entrou em sua fase comercial o Serviço Interdata.

Sendo um serviço público, a cargo da EMBRATEL, o Interdata permite ao usuário estabelecido no Brasil o acesso às redes de comunicação de dados localizadas no exterior, bem como aos bancos de dados conectados a estas redes.

Os usuários poderão acessar as bases de dados existentes no exterior, através:

- . Da Rede de Telefonia
- . Da Rede Nacional de Telex
- . De circuitos privativos de Comunicação de Dados, ligados ao Sistema Internacional de Comunicação de Dados.

Para isto basta entrar em contato com qualquer Órgão Comercial da EMBRATEL, para que seja feito um contrato de utilização do serviço, e os códigos de acessos as bases de dados possam ser oferecidos.

O Interdata entra em operação oferecendo de imediato aos seus futuros usuários, o acesso a dois bancos de dados:

- ORBIT
- QUESTEL

O primeiro, de propriedade do System Development Corporation - SDC, dos EUA, e o outro, propriedade da Telesystemes, da França, de acordo com contratos específicos firmados com a EMBRATEL e estas duas entidades.

A estrutura de tarifação com acesso pela redes públicas de telefonia e de telex é definida por:

- 1 - Taxa de registro por código de identificação (NUI).
- 2 - Tempo de conexão (minuto ou fração) de duração de cada chamada efetuada, sendo que a chamada deve ser arredondada para 1 (um) minuto.
- 3 - Tarifa de volume, aplicada por kilocaracteres, ou fração, computados aqueles transmitidos mais os recebidos, relativos a cada chamada efetuada, sendo que a fração deve ser arredondada para 1 (um) kilocaracter.

Utilizando-se do acesso dedicado - Rede Transdata, a estrutura define-se do seguinte modo:

- 1 - Taxa de registro
- 2 - Tarifa mensal fixa (franquia), uso mínimo de 1450 kilocaracteres para 300 bps/2400 minutos, e 2360 kilocaracteres para 1200 bps/2400 minutos.

Deve-se observar que a nível nacional as tarifas são uniformes, ou seja, são as mesmas para qualquer localidade do território, e definidas em FRANCO-OURO. E também nessas tarifas já estão incluso o fornecimento, manutenção e

instalação, pela EMBRATEL, dos MODEM's nas dependências do usuário.

Algumas idéias para quando
a Rede de Pacotes chegar.

Renato Brea
Depto. Coord. Comercial
EMBRATEL

SERVIÇO DE COMUTAÇÃO DE PACOTES

Um serviço com características novas

gerando

- Novas aplicações de TP
- Novas condições de projeto de sistemas
- Novas características de equipamentos
- Novos equipamentos
- Novas condições de gerência de rede
- Novos serviços de Teleinformática

Características do serviço

COMUTAÇÃO

CONEXÃO RÁPIDA

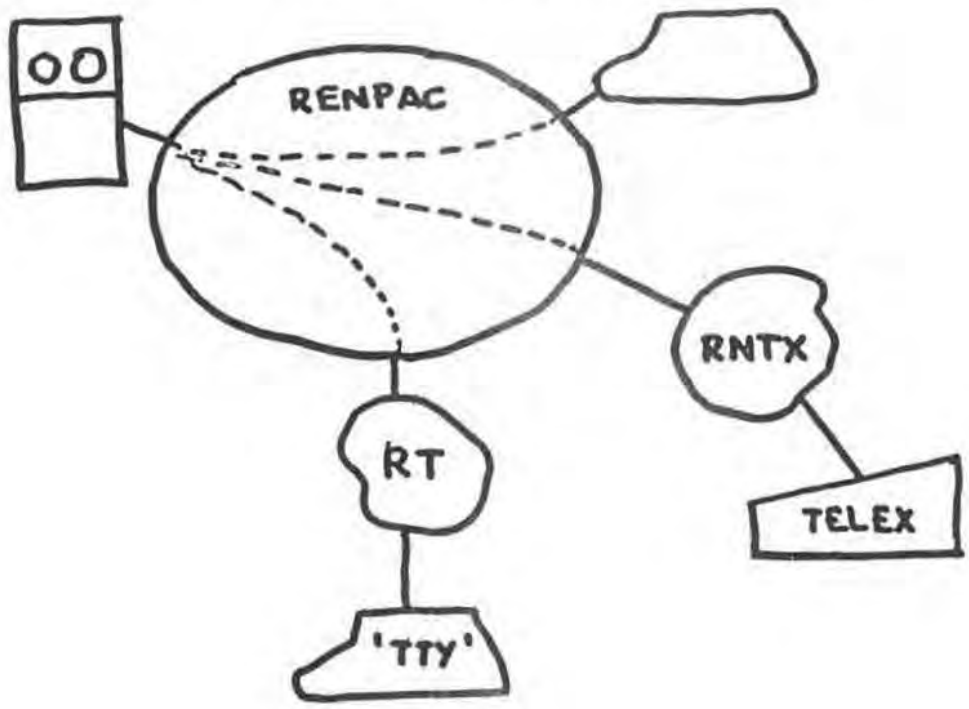
ABRANGENCIA NACIONAL

ACESSOS

DIRETOS

VIA REDE TELEFONICA

VIA REDE TELEX



CARACTERÍSTICAS DO SERVIÇO

PADRONIZAÇÃO DE PROCEDIMENTOS

ACESSO MODO PACOTE (X25)

MULTIPLEXAÇÃO
E
CONTROLE DE ERROS } NO ACESSO À REDE

ACESSO MODO CARACTER (PAD)

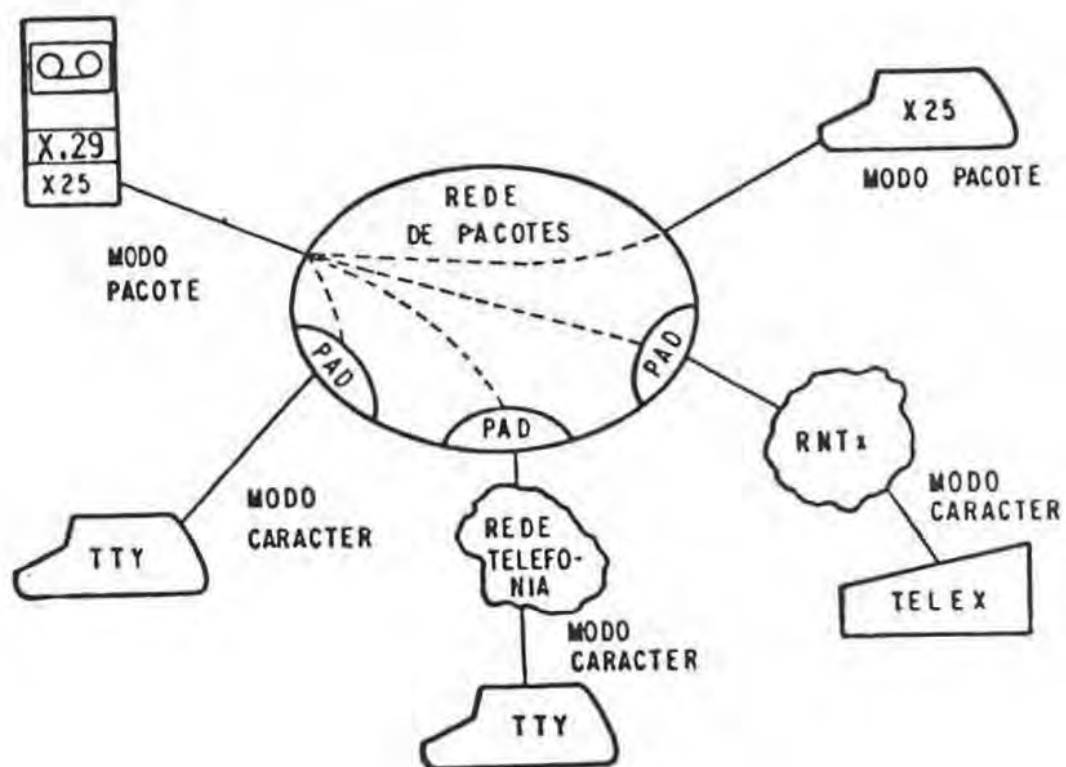
FACILIDADE DE CONEXÃO E DE USO

COMPATIBILIDADE

RAPIDEZ DE IMPLANTAÇÃO E DE EXPANSÃO

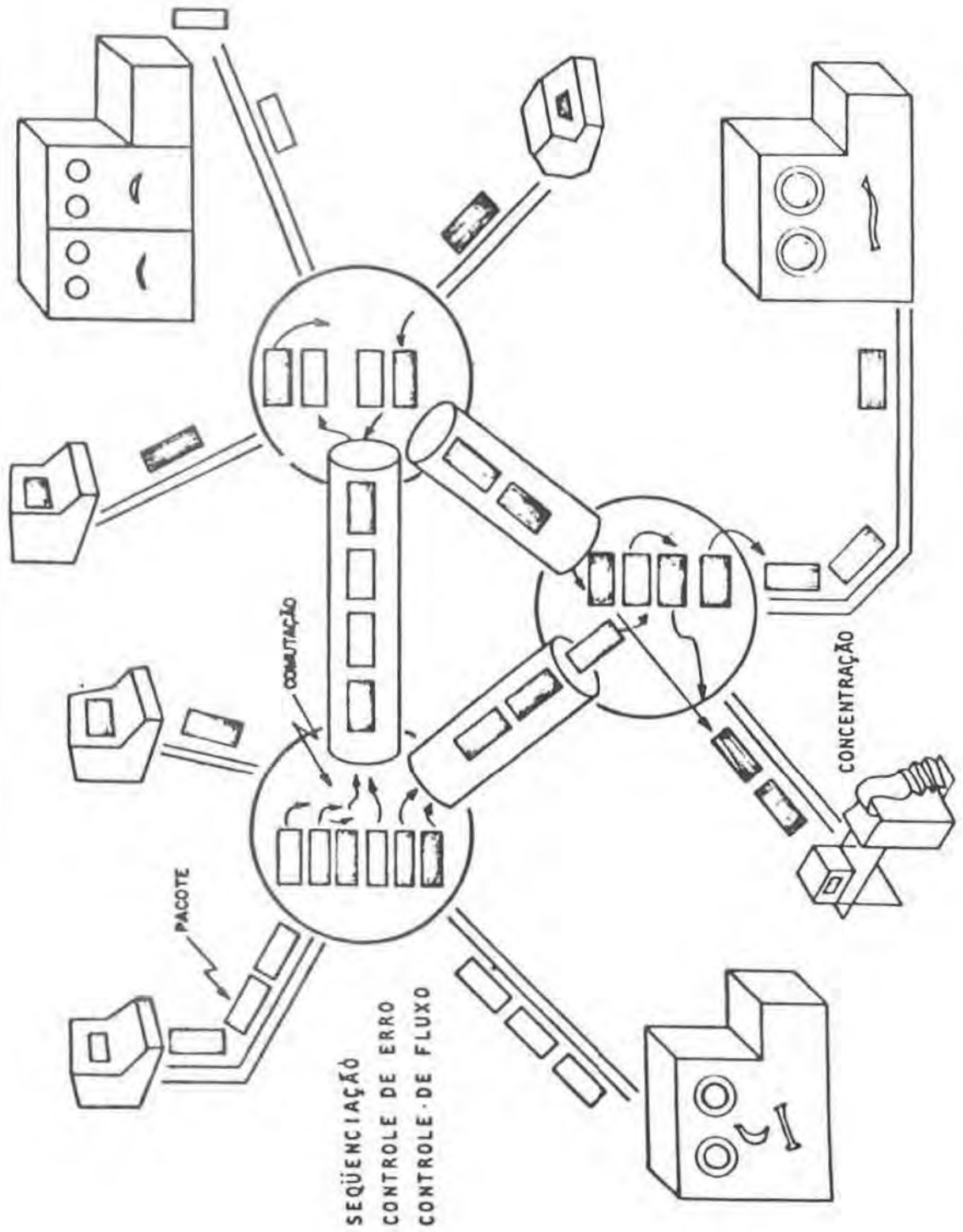
COMO LIGAR-SE À REDE

ESQUEMA REAL



EQUIPAMENTOS MODO PACOTE E MODO CARACTER

TECNOLOGIA DE COMUTAÇÃO DE PACOTES



NOSSA REDE

ACESSO	VELOCIDADE BPS	PROCEDIMENTO INTERFACE	POSSIBILIDADE DE "CHAMAR"
DEDICADO	SÍNCRONA	X.25	DEDICADO
	2400 4f		
	4800 4f		
	9600 4f		
DEDICADO	ASSÍNCRONA	COMPATÍVEL COM "TELETYPE"	TELEX
	300 2f		
	1200 4f		
COMUTADO TELEFÔNICO	ASSÍNCRONA	COMPATÍVEL COM "TELETYPE"	DEDICADO
	300 2f		
	1200 /75 2f		
COMUTADO TELEX	50	TELEX	DEDICADO

CARACTERÍSTICAS DO SERVIÇO

CONFIABILIDADE E DISPONIBILIDADE

BAIXO CUSTO UNITÁRIO

TARIFAÇÃO EM FUNÇÃO DO VOLUME

FLEXIBILIDADE

COMUTAÇÃO

ACESSOS COM Nº UNICO

GRUPOS FECHADOS

CHAMADAS ENTRANTES OU SAINTES BARRADAS

TARIFAÇÃO REVERSA

SENHAS

PARAMETROS DO PAD

NEGOCIAÇÕES

NOVAS APLICAÇÕES

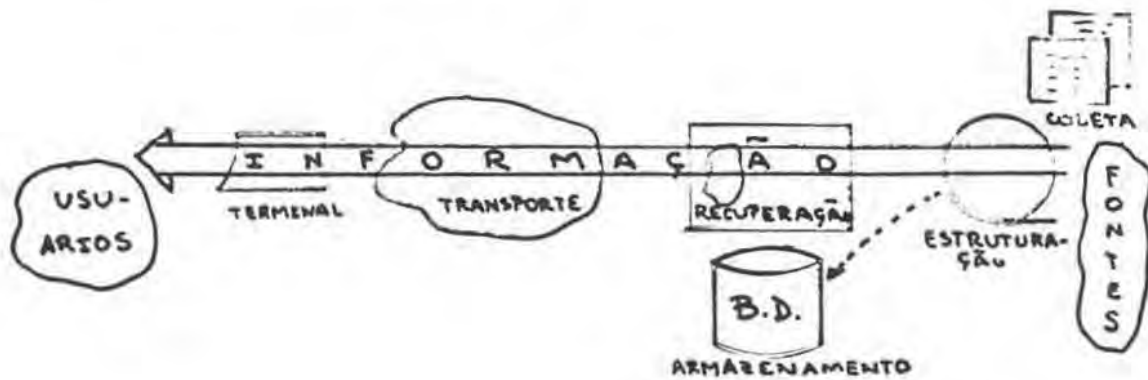
ACESSO A BANCOS DE DADOS

TIME-SHARING

CONSULTA
E
COLETA DE DADOS } ON-LINE

COMPARTILHAMENTO DE TERMINAIS

ACESSO A BANCOS DE DADOS



BASES DE DADOS

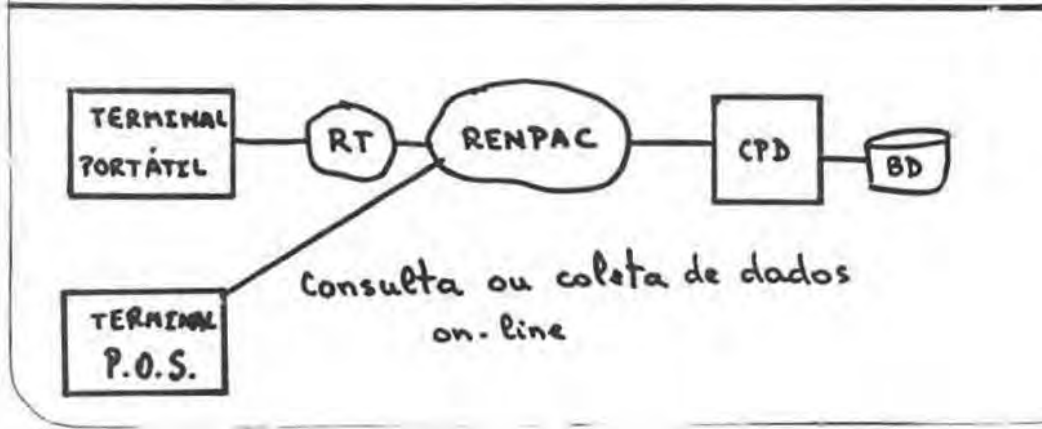
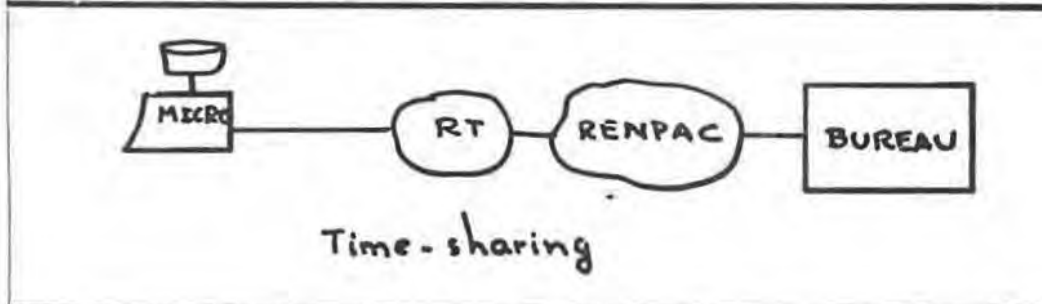
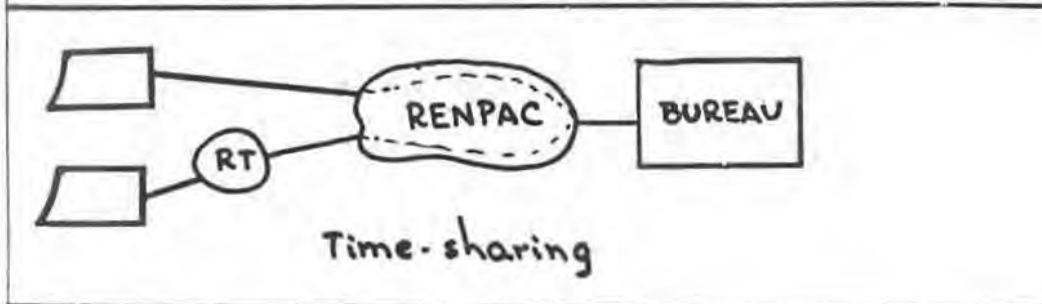
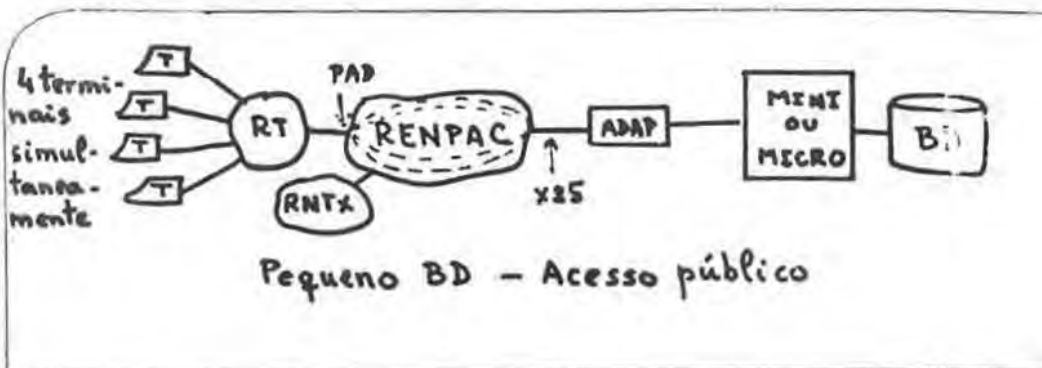
DE ACESSO PÚBLICO

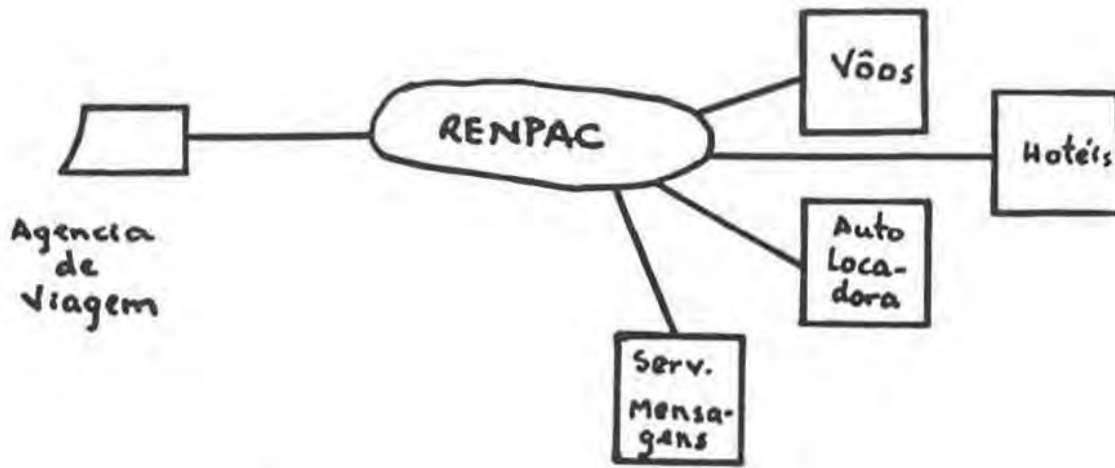
DE ACESSO LIMITADO

DE USO INTERNO

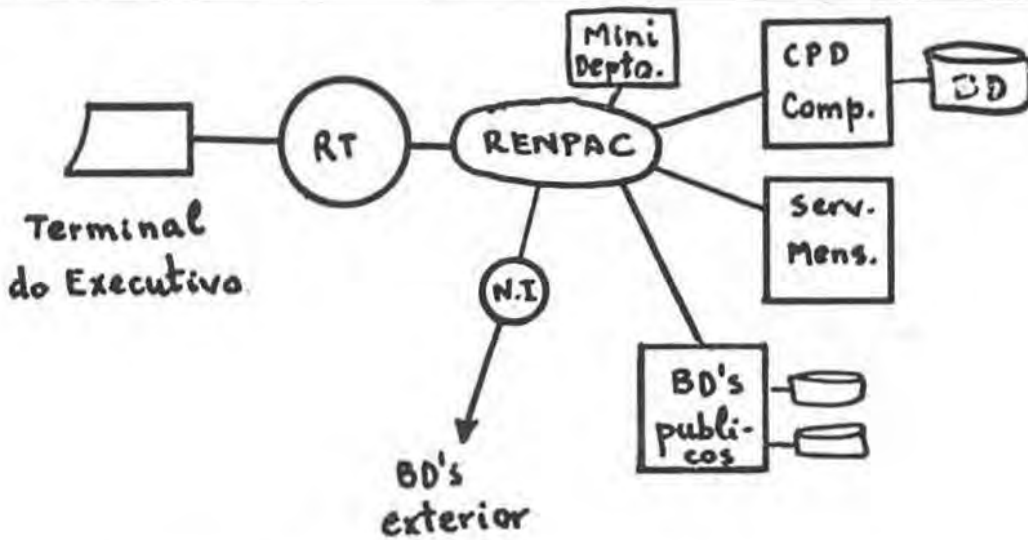
PANORAMA ATUAL

PERSPECTIVAS





COMPARTILHAMENTO DE TERMINAIS



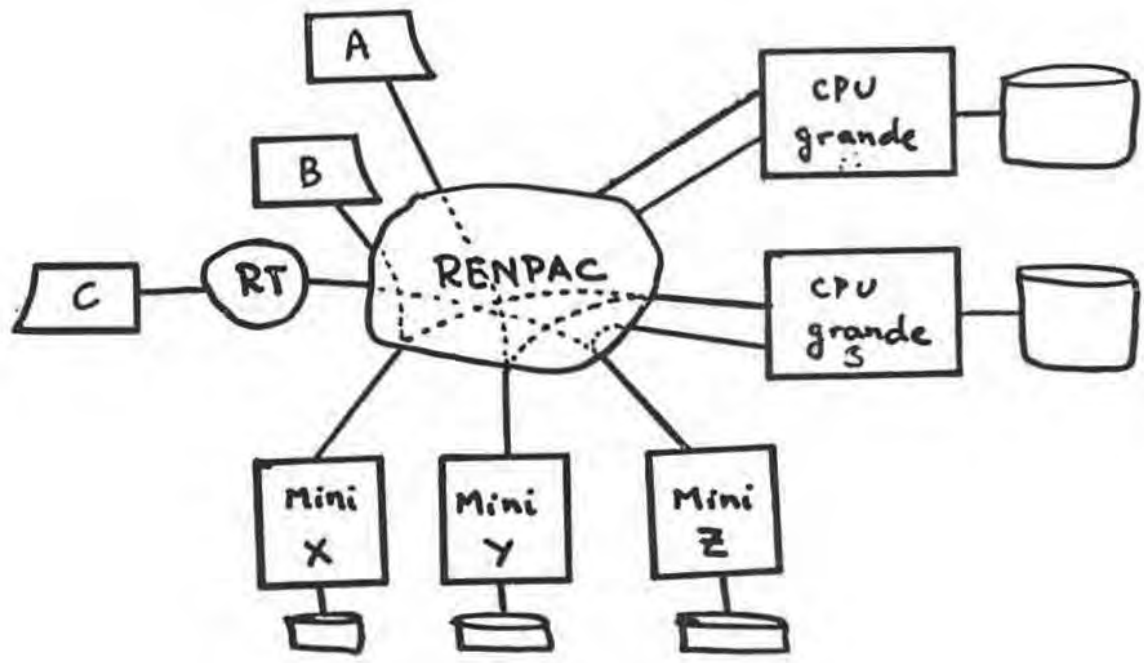
COMPARTILHAMENTO DE TERMINAIS

NOVAS CONDIÇÕES DE PROJETO DE SISTEMAS

COMUTAÇÃO
CONEXÃO RÁPIDA
ACESSOS COM N.º ÚNICO

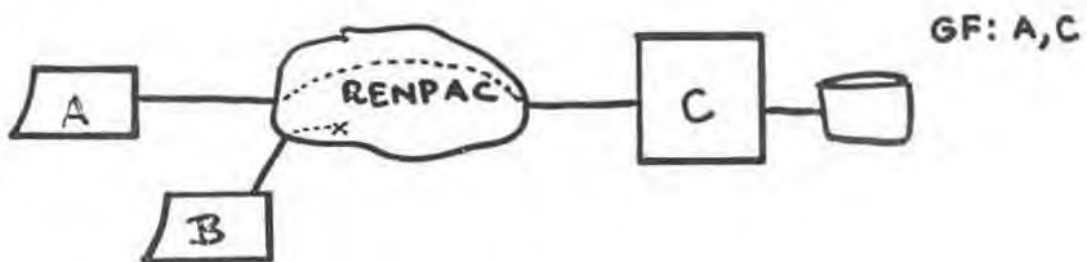


SISTEMAS DISTRIBUIDOS
ESQUEMAS DE BACK-UP
BALANCEAMENTO DE CARGA
ESPECIALIZAÇÃO DE CPU'S

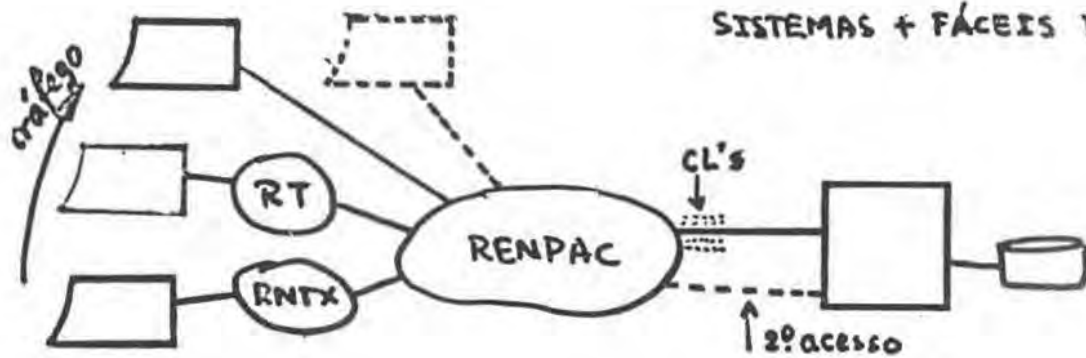


NOVAS CONDIÇÕES DE PROJETO DE SISTEMAS

GRUPOS FECHADOS
CHAMADAS BARRADAS
SENHAS } → PARTE DA SEGURANÇA JÁ
NA REDE



ABRANGENCIA NACIONAL
ACESSOS VIA RT & TELEX } → IMPLANTAÇÃO RÁPIDA
EVOLUÇÃO CONFORME O USO
CUSTO E DESEMPENHO CONTRA A-
DOS
SISTEMAS + FÁCEIS DE USAR



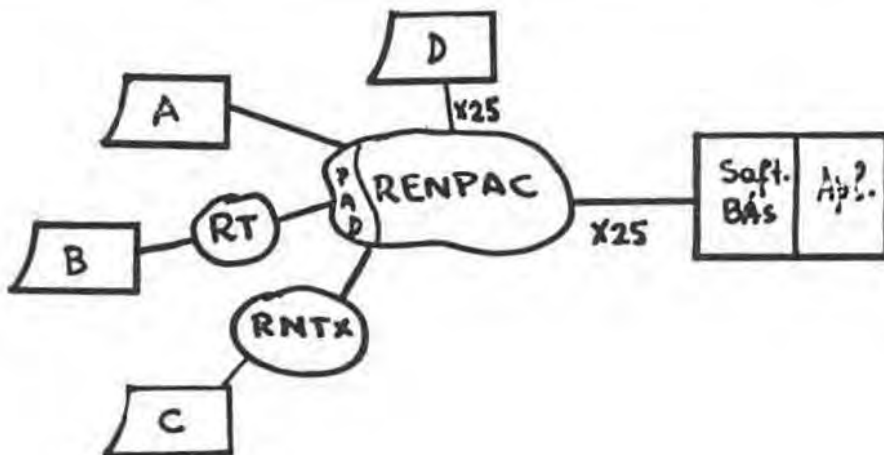
NOVAS CONDIÇÕES DE PROJETO DE SISTEMAS

TARIFAÇÃO POR VOLUME
BAIXO CUSTO UNITARIO

→ SISTEMAS + OBJETIVOS
MSGs. + CURTAS
NOVA COMUNIDADE DE USUAR

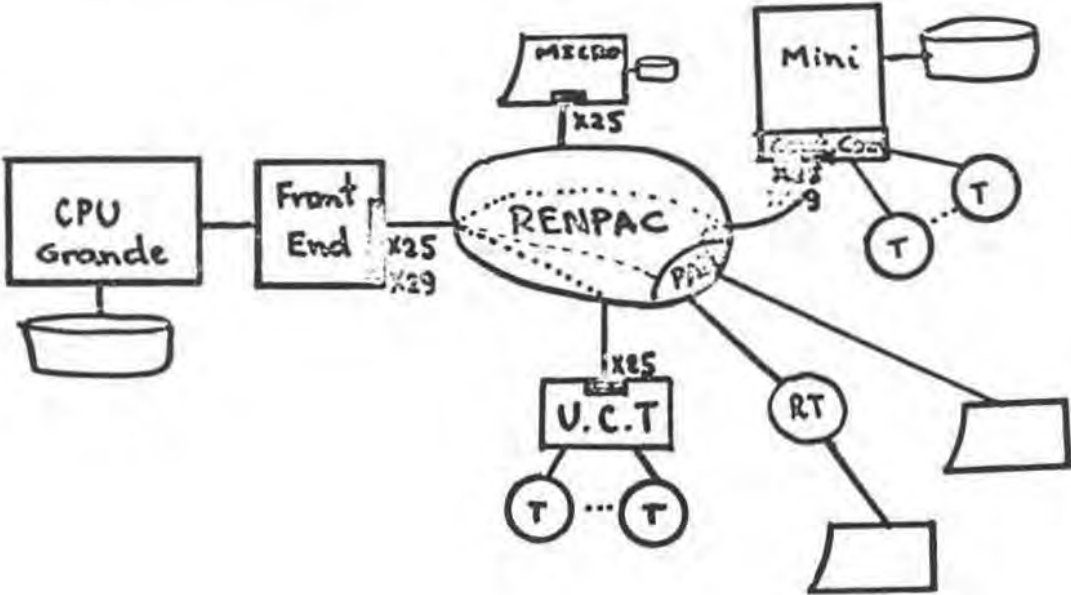
TADRONIZAÇÃO
PARAMETROS DO PAD

→ SISTEMAS ABERTOS
INDEPENDENCIA DO TERMINA.
SOFTWARE BÁSICO + INTELIGE



NOVAS CARACTERÍSTICAS
DE EQUIPAMENTOS

SUORTE AO X25 e X29 (INTEGRADO)

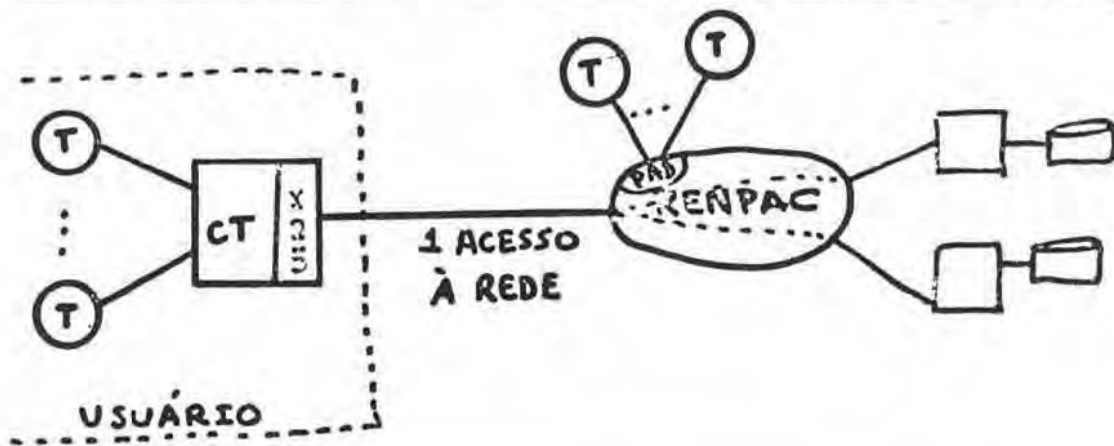


ARQUITETURAS DE REDE

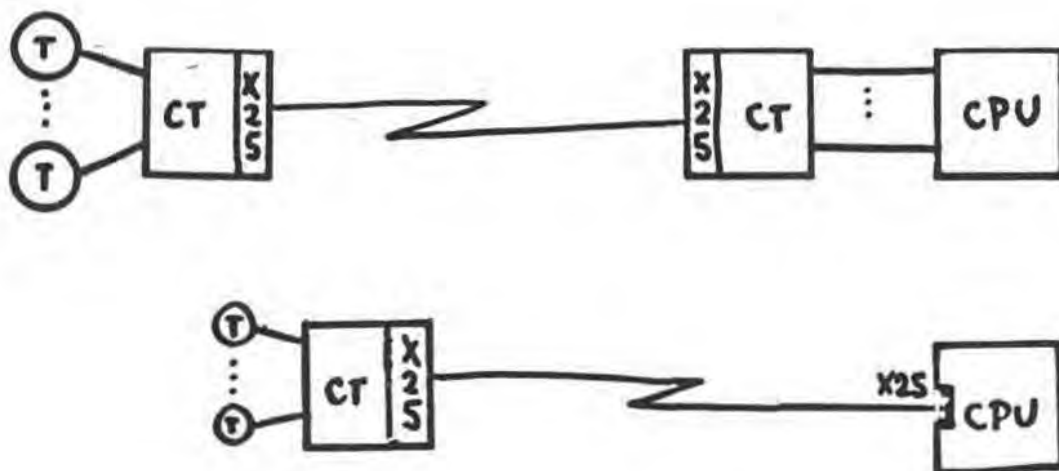
PROCOLOS DE ALTO NÍVEL

NOVOS EQUIPAMENTOS

CONCENTRADOR DE TERMINAIS ASSÍNCRONOS



MULTIPLEXADOR ESTATÍSTICO

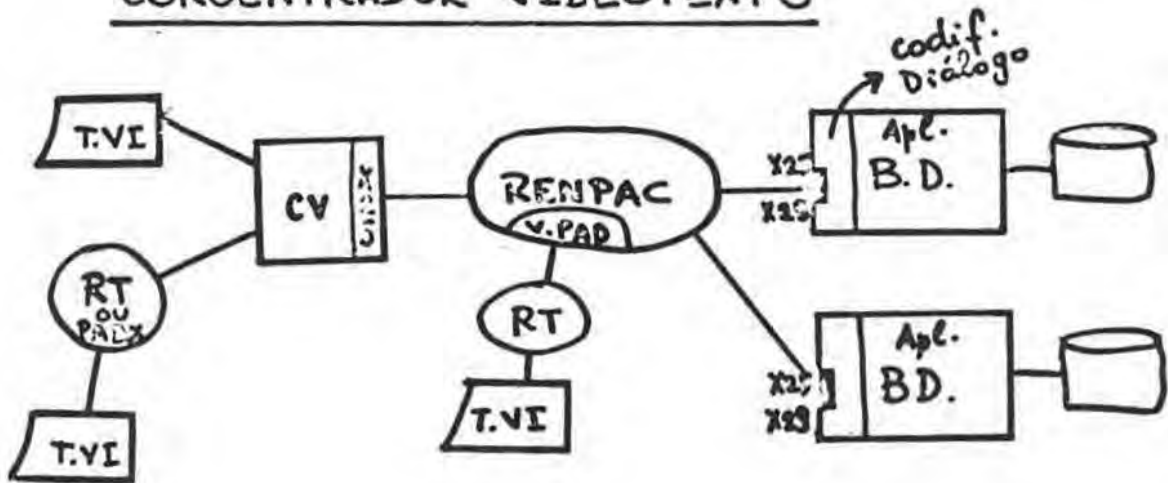


PROCESSADOR SATÉLITE

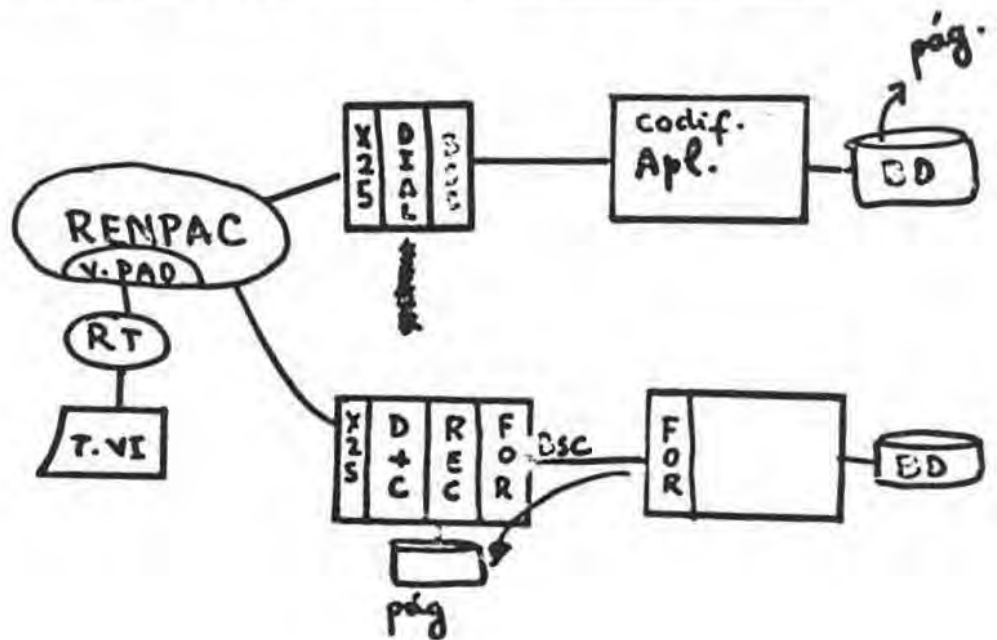


NOVOS EQUIPAMENTOS

CONCENTRADOR VIDEOTEXTO



ADAPTADORES / FORMATADORES VIDEOTEXTO

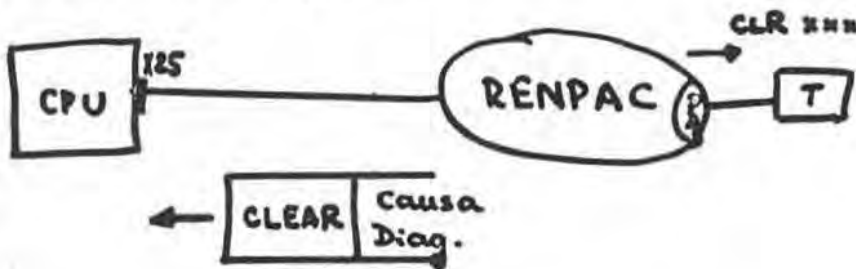


NOVAS CONDIÇÕES DE GERENCIA DE REDE

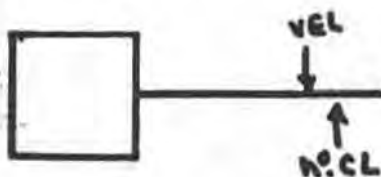
POUCAS PREOCUPAÇÕES COM TOPOLOGIA



PREOCUPAÇÃO EM ANALISAR CAUSAS E DIAGNOSTICOS ENVIADOS PELA REDE

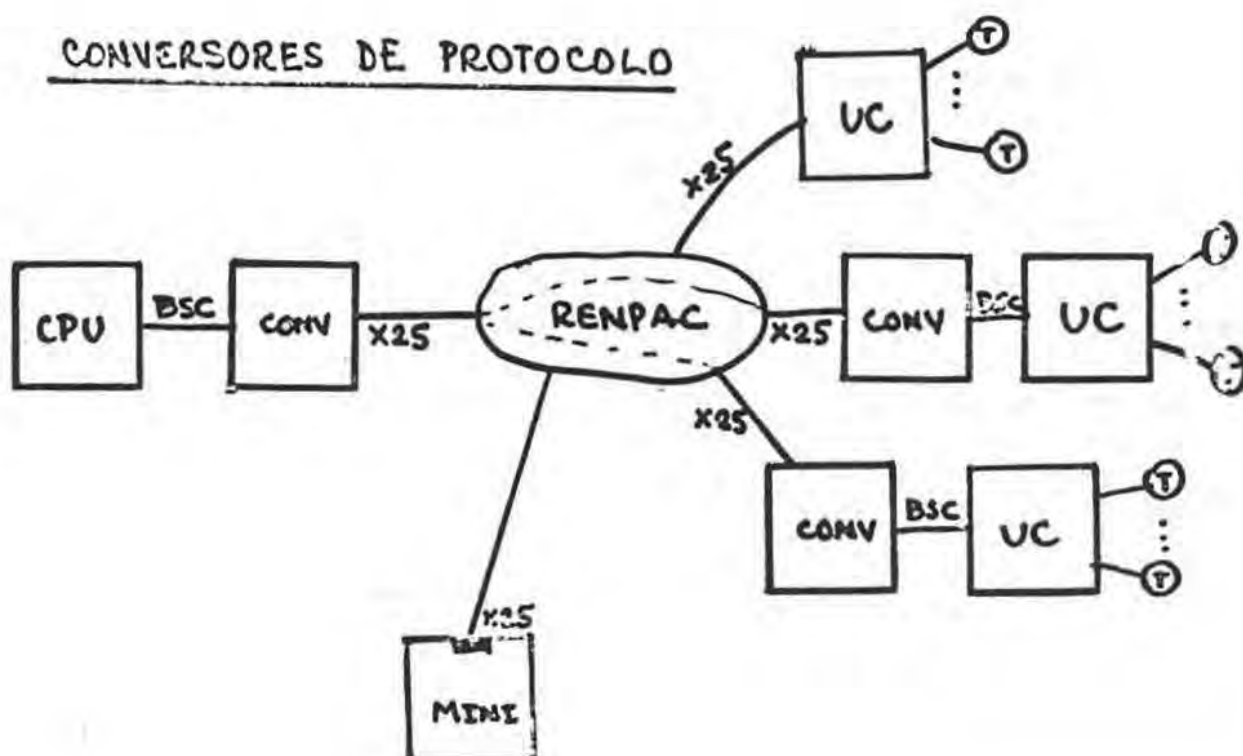


PREOCUPAÇÃO EM MANTER DIMENSIONAMENTO ADEQUADO DOS ACESSOS



NOVOS EQUIPAMENTOS

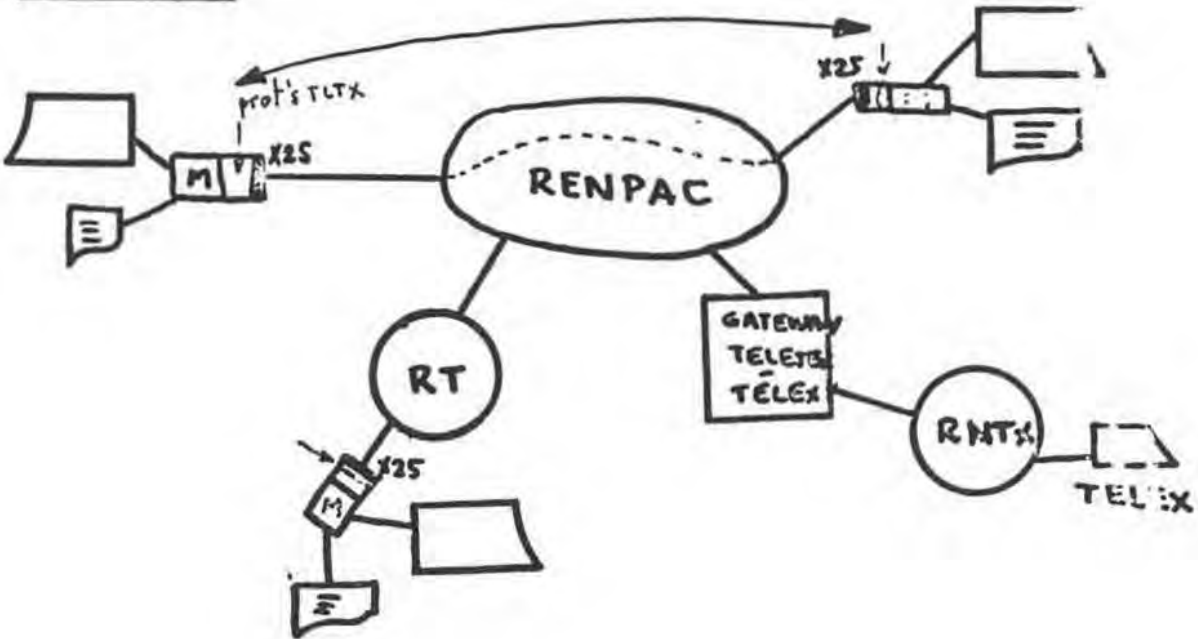
CONVERSORES DE PROTOCOLO



• ACORDO SOBRE PROTOCOLO DE TRANSPORTE

NOVOS SERVIÇOS DE TELEINFORMÁTICA

TELETEX

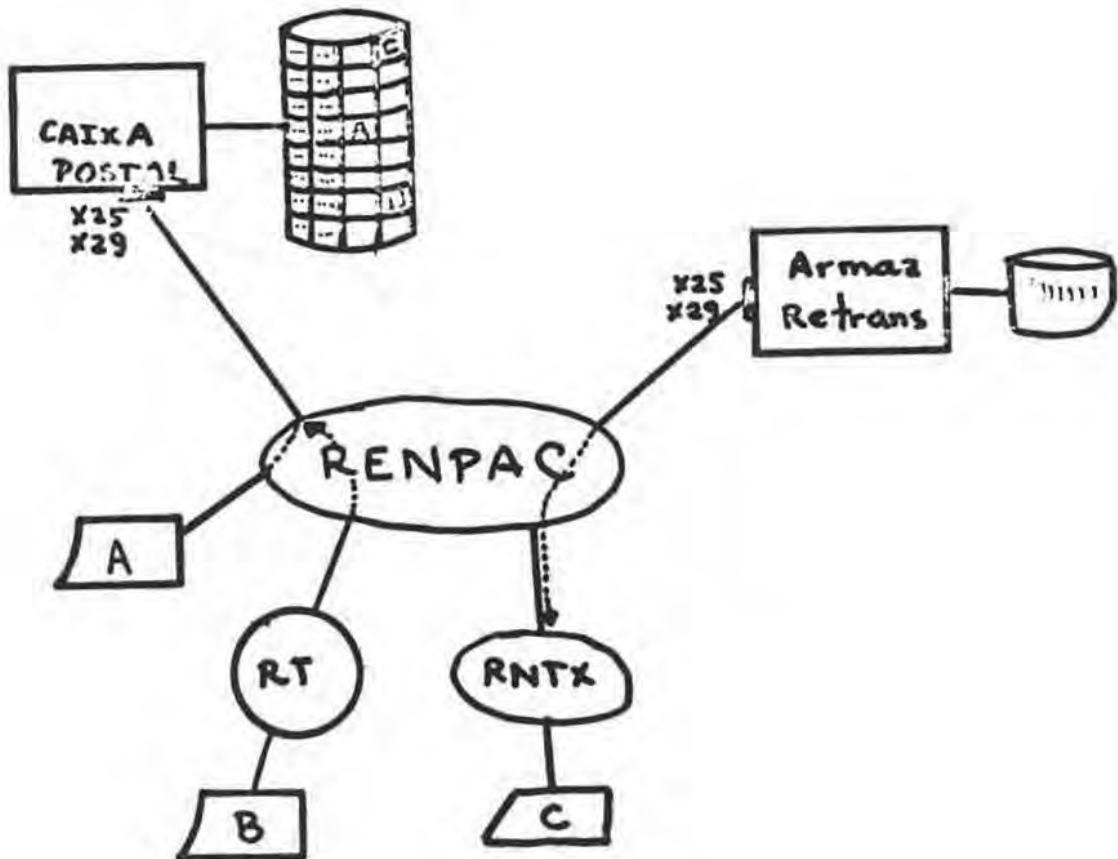


- TERMINAL TELETEX
- GATEWAY TELETEX-TELEX
- CHAMADAS PARA A RT
- X25 VIA RT

NOVOS SERVIÇOS DE TELEINFORMÁTICA

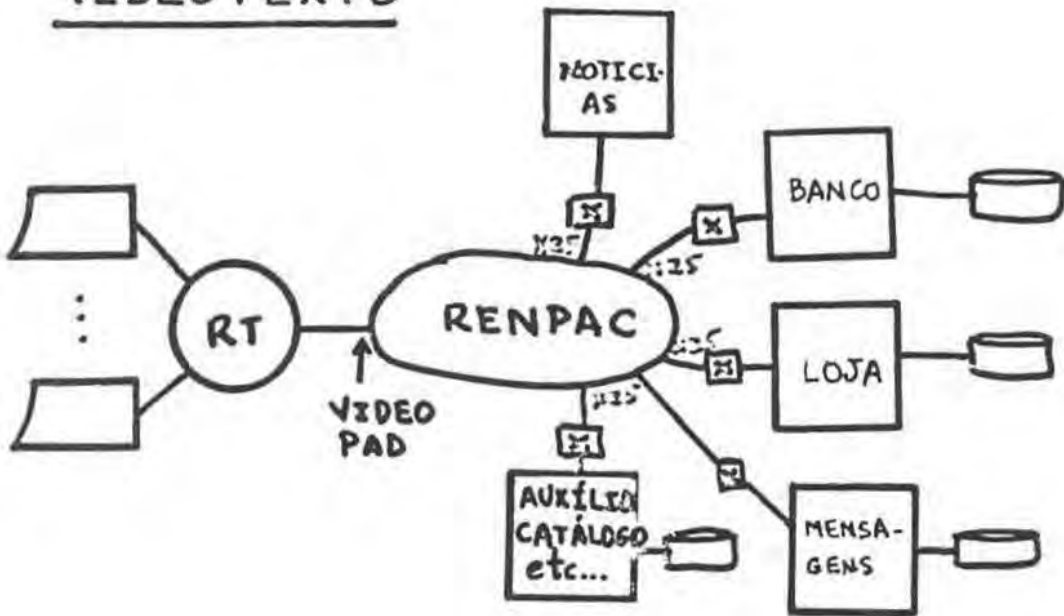
TELEMENSAGENS

- 1-TIPO ARMAZ. + RETRANSMISSÃO
- 2-TIPO CAIXA POSTAL



NOVOS SERVIÇOS DE TELEINFORMÁTICA

VIDEOTEXTO



Rede Edisa para Automação de Bancos

Rolf Harm Hinrichs

EDISA - Eletrônica Digital S.A.

Distrito Industrial de Gravataí - BR290 - Km 22

94.000-GRAVATAÍ-RS

RESUMO

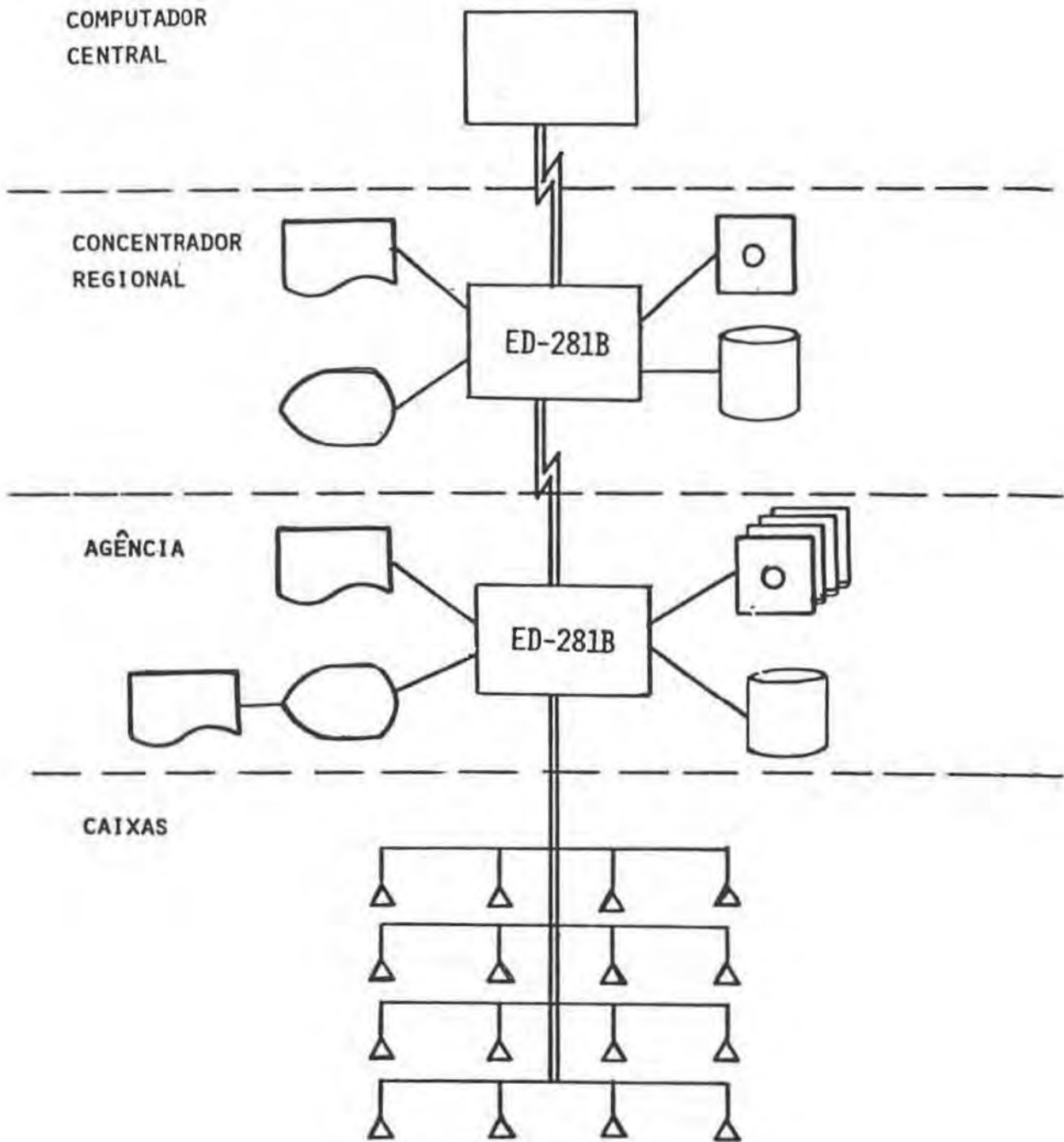
Este trabalho descreve a utilização de uma rede de micro e minicomputadores para automatizar os serviços de um banco.

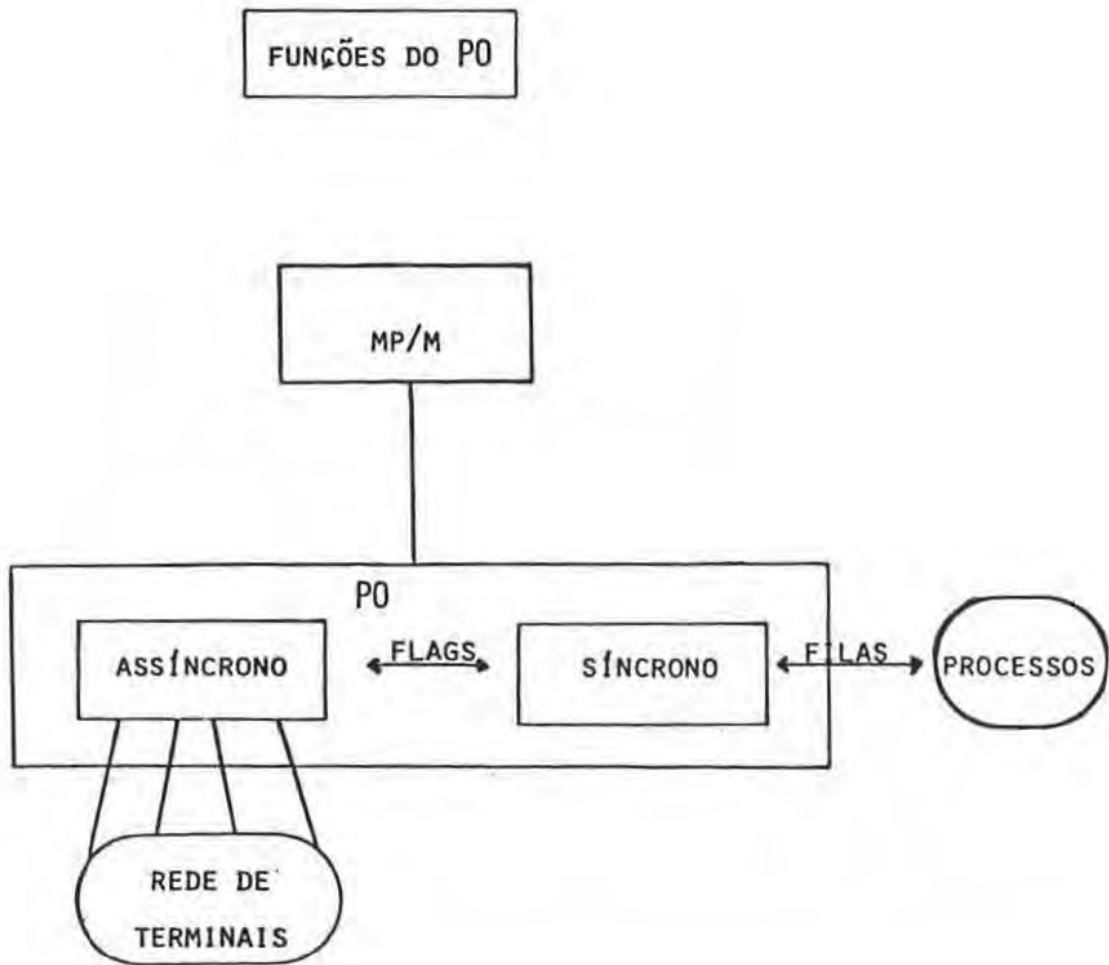
Inicialmente é descrito o histórico desta rede, e quais, os princípios objetivos que foram levados em conta na sua elaboração.

A seguir a rede física é apresentada, sob ponto de vista de equipamentos utilizados, topologia e aplicação. São descritos vários exemplos de transações que ocorrem entre os vários nodos da rede.

A implementação é descrita em maior detalhe: protocolo utilizado e como foi implementado nas diversas máquinas visando portabilidade e fácil manutenção. Também a adaptação de um sistema operacional a esta rede é apresentado.

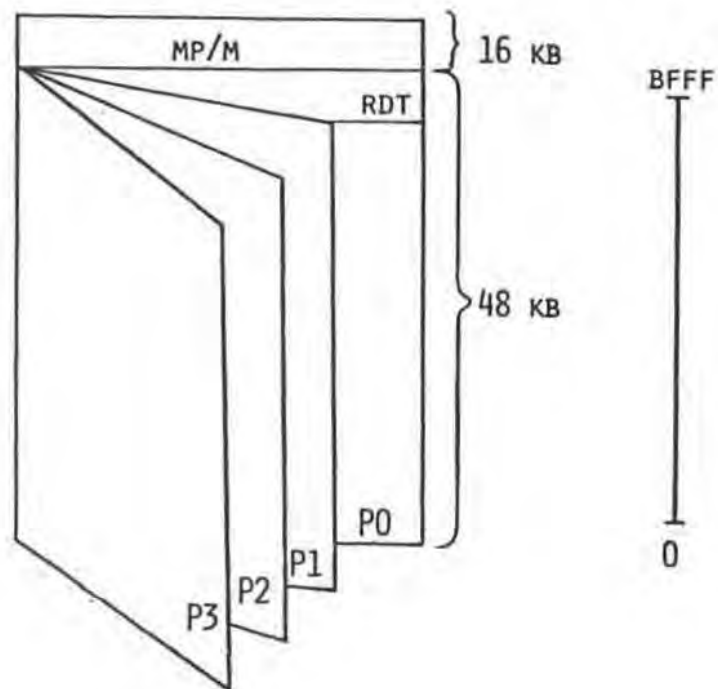
MODULARIDADE





- INTERFACE ENTRE O MP/M E UMA REDE DE TERMINAIS

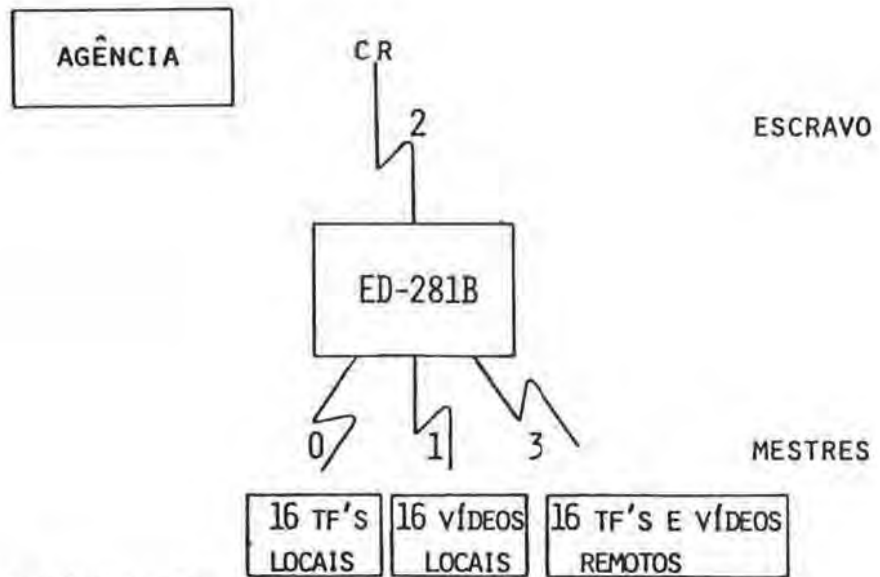
UTILIZAÇÃO DA MEMÓRIA



- P0: SEMPRE NO BANCO 0
- RDT: PROGRAMA DE DEPURAÇÃO
OCUPA SEGMENTO NO TOPO DO BANCO 0 (RDT,PRL)
- ENDEREÇAMENTO NO MP/M: ATÉ 64 KB (0000-FFFF)
- CARGA AUTOMÁTICA

CONTROLE DA COMUNICAÇÃO

CANAIS DE COMUNICAÇÃO



VELOCIDADES:

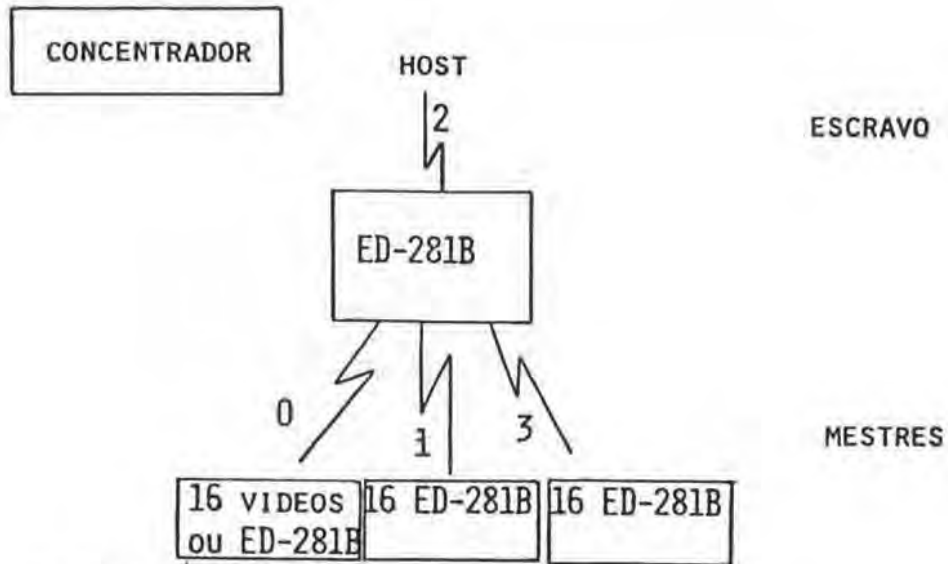
- CANAL 0 - 4800 BPS
 - CANAL 1 - 9600 BPS
 - CANAL 2 - 1200 BPS
 - CANAL 3 - 1200 BPS
- } RESTRIÇÃO DE LINHA
MODEM'S ASSÍNCRONOS

LIMITES

- TEÓRICO :16 TERMINAIS/CANAL MESTRE
- REAL: { ATÉ 16 TF'S
 { ATÉ 4 VÍDEOS
- IDEAL: { ATÉ 8 TF'S
 { 1 VÍDEO

CONTROLE DA COMUNICAÇÃO

CANAIS DE COMUNICAÇÃO



VELOCIDADES:

CANAL 0 }
CANAL 1 } 1200 BPS
CANAL 2 }
CANAL 3 }

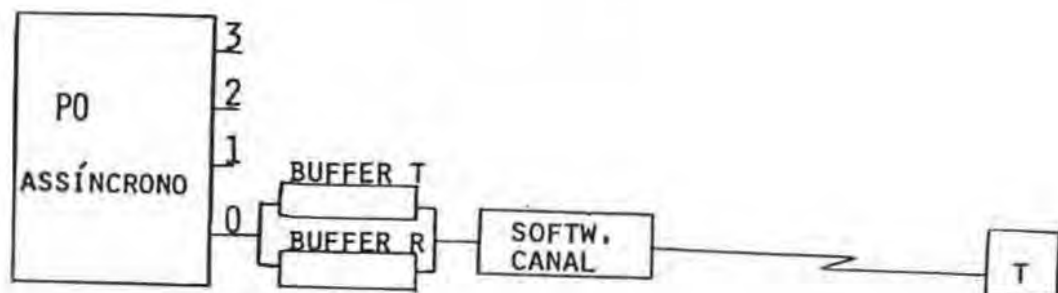
LIMITES:

TEÓRICO: ATÉ 16 AGÊNCIAS/CANAL MESTRE

REAL : ATÉ 15 AGÊNCIAS NO TOTAL E 1 VÍDEO

OBS: DEVE-SE BALANÇEAR O PORTE DAS AGÊNCIAS DE CADA CONCENTRADOR E O PORTE DAS AGÊNCIAS DE CADA CANAL.

CONTROLE DA COMUNICAÇÃO



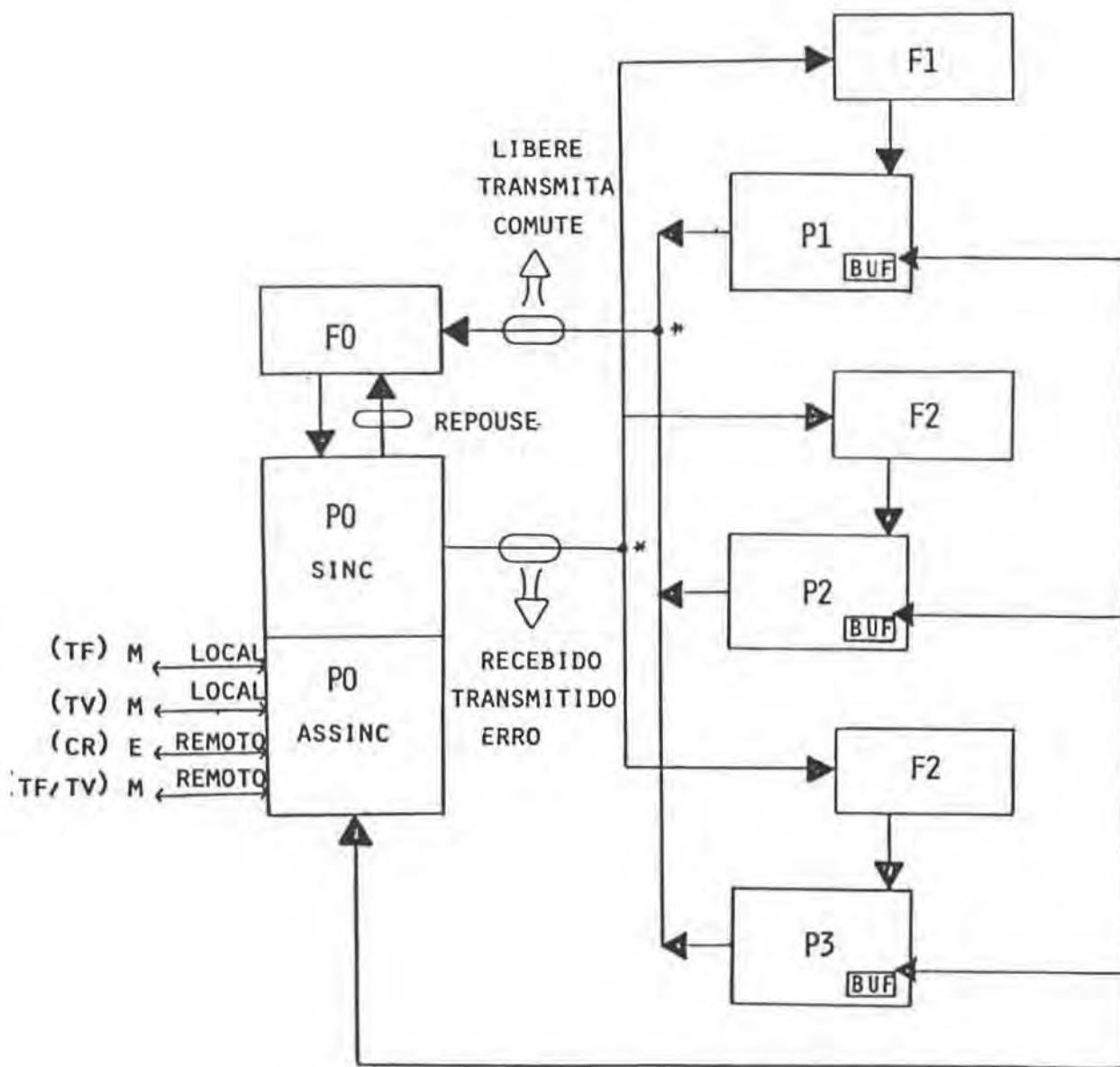
NA TRANSMISSÃO:

-P0 RETIRA MENSAGEM DO BUFFER DO PROCESSO E COLOCA NO BUFFER DE TRANSMISSÃO DO CANAL.

NA RECEPÇÃO:

-P0 RETIRA MENSAGEM DO BUFFER DE RECEPÇÃO DO CANAL E COLOCA NO BUFFER DO RESPECTIVO PROCESSO.

TRÁFEGO DE MENSAGENS



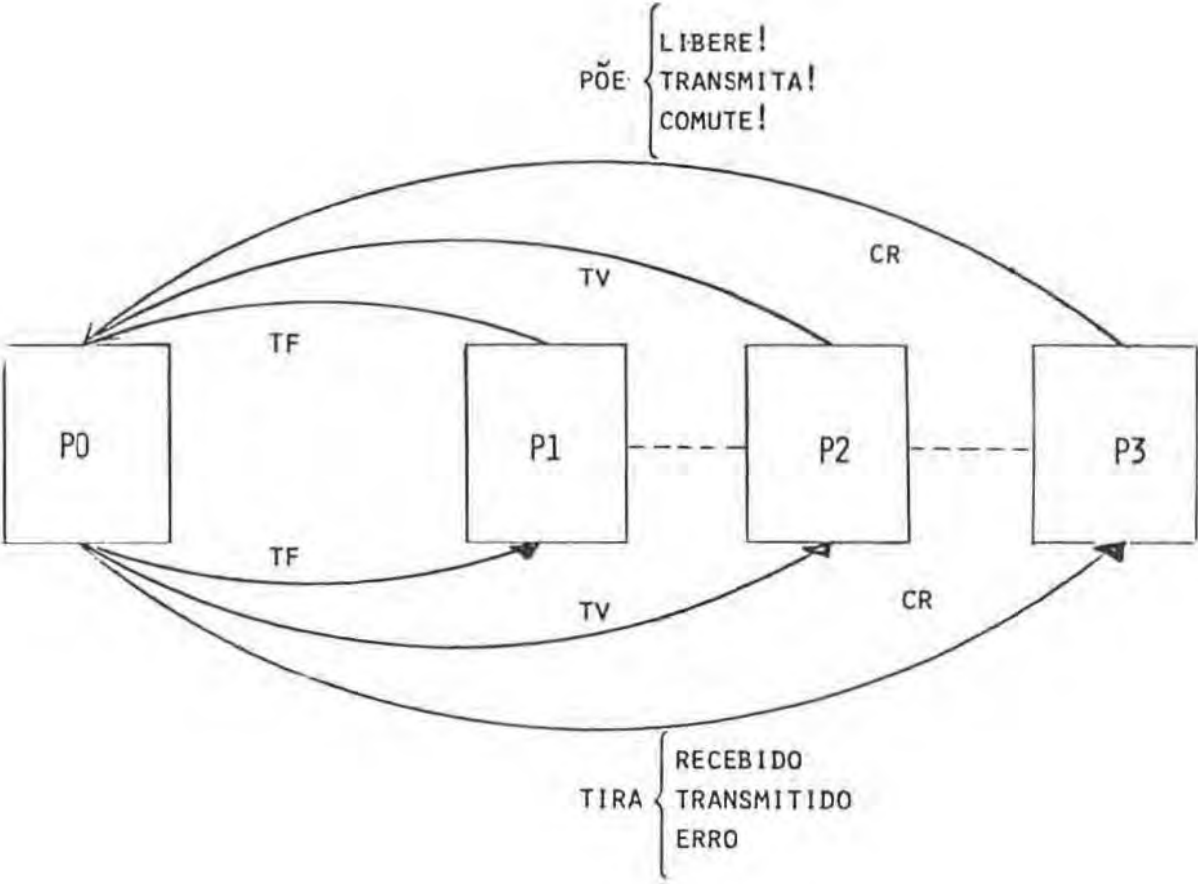
* CAIXA DE MONITORAÇÃO

L = DO APLICATIVO PARA O PO

E = DO PO PARA O APLICATIVO

EX: L [msg] * E [msg] *

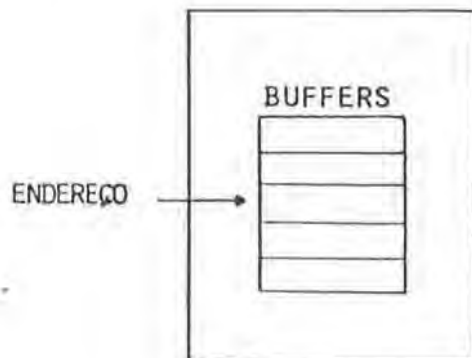
TRÁFEGO DE MENSAGENS



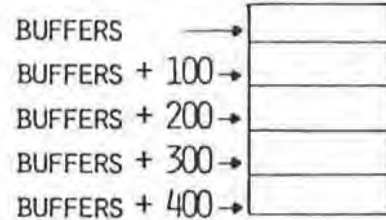
FORMATO DAS MENSAGENS

ENDEREÇO

ENDEREÇO DO BUFFER; CALCULADO PELO PO DE ACORDO COM O NÚMERO DA OCORRÊNCIA INFORMADO.



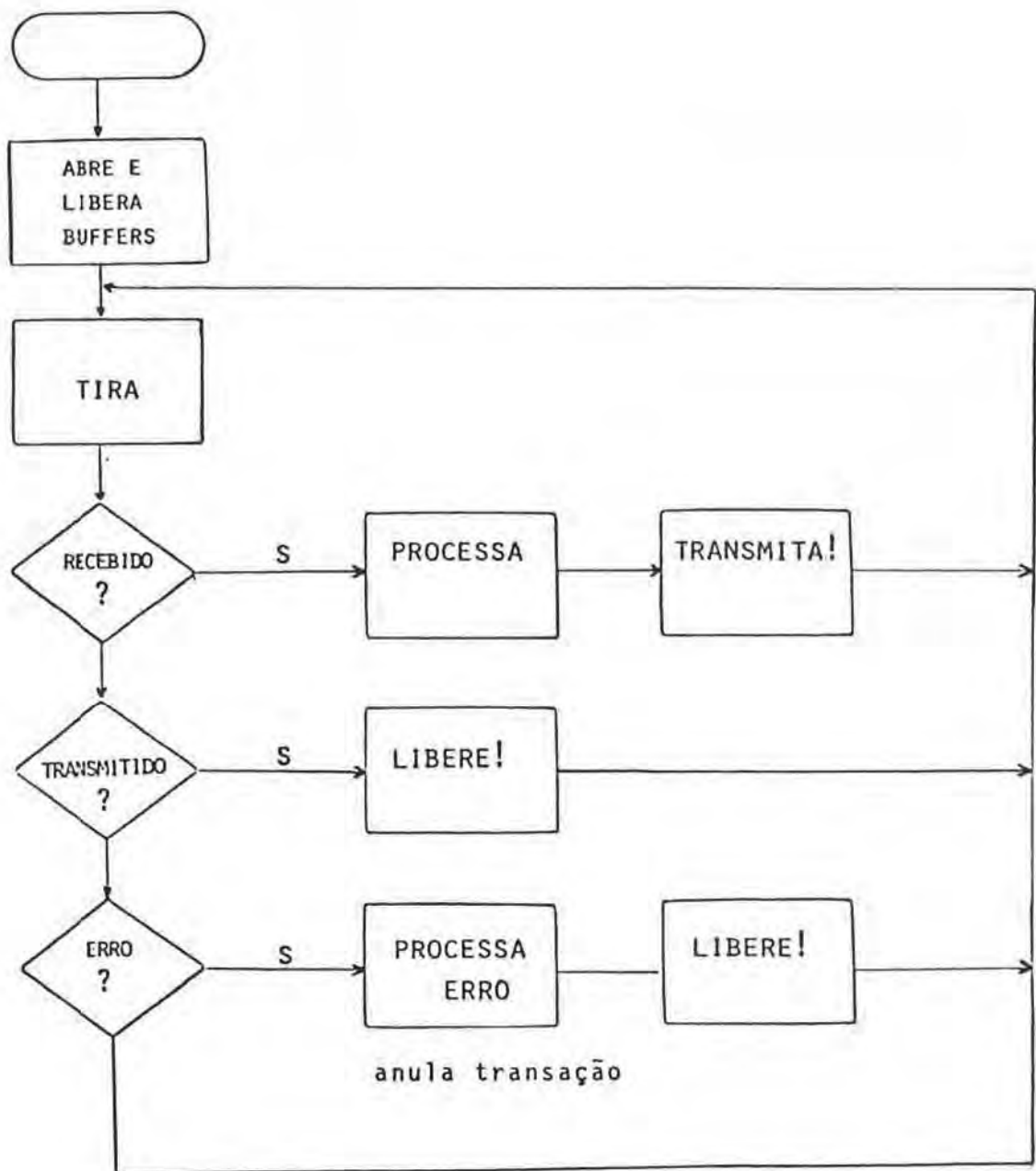
```
01 BUFFERS
03 - OCCURS 5
05 - PIC X (100)
```



TAMANHO

LIBERE - POE - TAMANHO DO BUFFER QUE PODE SER PREENCHIDO
RECEBIDO - TIRA - TAMANHO DA MENSAGEM RECEBIDA
TRANSMITA - POE - TAMANHO DA MENSAGEM A TRANSMITIR
TRANSMITA - TIRA - TAMANHO DA MENSAGEM TRANSMITIDA

FLUXO BÁSICO DE PROCESSAMENTO



GRÁFICA
1983 UFRGS
30
ANOS



Composto e Impresso na Gráfica da
UFRGS - Rua Jacinto Gomes, 540 -
1º and. - Fone 23-6800 e 23-6909 R. 10
Porto Alegre - RS — Brasil

1983