

2º SIMPÓSIO BRASILEIRO SOBRE REDES DE COMPUTADORES (2º SBRC)

**COMPORTA PARA CONEXÃO DE UM COMPUTADOR DEC-10  
A UMA REDE PÚBLICA DE COMUNICAÇÃO DE DADOS**

José A.S. Monteiro

Merval A. Jurema Filho

Paulo R. Freire Cunha

Departamento de Informática da UFPE  
Centro de Ciências Exatas e da Natureza  
Cidade Universitária  
50.000 - RECIFE - PE

**RESUMO:**

Dentro do esforço de conectar os recursos computacionais da UFPE aos de outras Universidades, encontra-se em desenvolvimento, pelo Grupo de Redes de Computadores e Sistemas Distribuídos da UFPE, uma Comporta de acesso do sistema DEC-10 a uma rede de comutação de pacotes. O presente documento apresenta uma descrição detalhada da estrutura de software da COMPORTA X.25, seguida da situação atual de desenvolvimento do projeto.

## 1. INTRODUÇÃO:

É de grande atualidade na área de informática aqui, no Brasil, o desenvolvimento de equipamento e de programas com vistas à interligação de sistemas computacionais através de redes de computadores. A motivação do uso de redes de computadores provém da necessidade de se compartilhar recursos básicos ou informações entre diversas instituições, ou numa única, se estiver distribuída geograficamente. Uma vez implantada a sub-rede, ou seja, os seus nós de comutação, não está resolvido todo o problema. É preciso que cada um dos computadores que devem se conectar à rede sejam adaptados para este fim. É necessária a utilização de um equipamento específico, assim como, de programas encarregados de controlar a troca de dados, os protocolos.

Se estes programas são executados pelo processador principal do sistema, temos um desperdício na capacidade de processamento devido ao tempo gasto para o desempenho destas funções . A solução para este problema é transferir estas funções a um processador dedicado, de custo mais barato, que deste modo alivia o processador principal. Um processador dedicado à função acima é denominado de comporta , pois é através dele que "fluem" os dados do processador principal para a rede e vice-versa.

A interligação de computadores numa rede foi particionada em 7 (sete) níveis funcionais (modelo ISO) [ ZIMM 80] que vão desde a ligação elétrica e mecânica (nível 1) até os programas que fornecem serviços aos usuários como, por exemplo, os de transferência de arquivos (nível 7). Cada um destes níveis fornece serviços ao nível imediatamente superior, fazendo uso dos serviços fornecidos pelo nível imediatamente inferior. Alguns destes níveis, os de 1 ao 3, cuidam do estabelecimento/fim de conexões e da transmissão correta de pacotes de dados através da sub-rede de comunicações. Estes três níveis são os de relacionamento de

um computador com o nó de comutação ao qual está conectado. Os demais níveis correspondem às interações entre os computadores parceiros na comunicação. Portanto, qualquer computador que deva se conectar à rede pública deverá ter implantadas as funções padronizadas correspondentes aos níveis de 1 a 3. A implementação e padronização dos níveis superiores ficam a cargo das instituições que desejam compartilhar seus recursos. Naturalmente, a comunicação será possível apenas entre computadores que obedecem à mesma padronização.

Um comporta possui duas vias por onde fluem os dados. Uma que relaciona com a sub-rede de comunicações e a outra que a relaciona com o computador principal, também denominado "hospedeiro". O relacionamento com o nó de comutação é padronizado pela administração da rede, e portanto, único, seja qual for o computador que desejamos conectar. Já o relacionamento entre a comporta e o computador dependerá de seus recursos de comunicação disponíveis. Como, em geral, conecta-se a uma rede computadores heterogêneos, inclusive de fabricantes diferentes, é bastante interessante o desenvolvimento de uma comporta que, com um pequeno esforço adicional, permita a conexão de qualquer computador à Rede, ao menos no que diz respeito aos três níveis inferiores.

## 2. O PROJETO DA COMPORTA X.25

Atualmente destacam-se no Brasil o projeto CEPINNE - que visa interligar os computadores das universidades federais do Norte-Nordeste; e a RENPAC - Rede Nacional de Comutação de Pacotes - que encontra-se em fase de implantação pela EMBRATEL.

Dentro do esforço de conectar os recursos computacionais da UFPE aos de outras universidades, foram iniciados os estudos com vistas à implementação de uma COMPORTA de acesso do sistema DEC-10 a uma rede de comutação por pacotes. Para este fim, a UFPE conta com um microcomputador da EBC, o SDE 40 para o qual

já foi desenvolvida uma interface que implementa o nível 1 e parte do nível 2 da recomendação X.25 da CCITT [CCIT 80] e [TELE 81] que é o protocolo de acesso a redes públicas. Na via de acesso à rede deve ser ainda codificados e testados os programas que implementam o restante do nível 2 e o nível 3 da recomendação X.25.

Com relação à via de acesso ao computador hospedeiro, o que pretende-se nesta fase inicial é desenvolver a interface com o DEC-10 partindo-se dos recursos de comunicação que já se encontram implantados nele, ou seja, o ANF-10. Com este procedimento poderia ser feita a comunicação de imediato com os outros computadores da DIGITAL, como os DEC-10 instalados na UFBA, UFCE e UFPA. Numa etapa posterior, com um pequeno adicional, será possível conectar um computador qualquer à rede pública, através desta comporta, bastando para isto fazer as adaptações necessárias no sistema operacional do computador hospedeiro e instalar uma interface serial caso não exista nenhuma disponível.

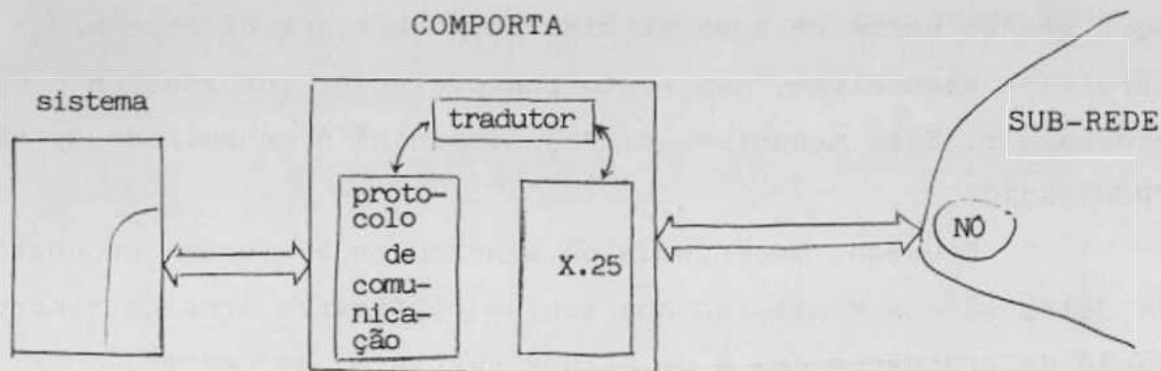
### 3. ESTRUTURA DO SOFTWARE DA COMPORTA

O tipo de Comporta que estamos desenvolvendo é aquele que efetua a tradução do protocolo de comunicação do sistema que desejamos conectar à rede, para os protocolos da Recomendação X.25 que são, justamente, os protocolos de acesso às Redes Públicas [HOLL 82].

Deste modo, podemos visualizar o software da Comporta como sendo dividido em três grandes blocos:

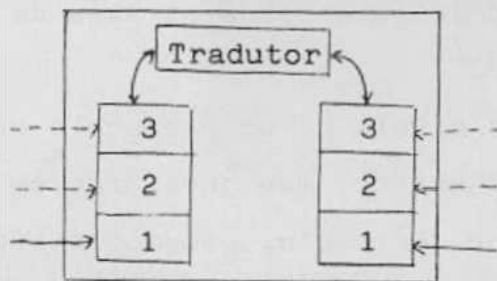
- o protocolo de comunicação do sistema
- o tradutor
- os protocolos da X.25

Como pode-se observar na figura, existem duas portas de comunicação para a comporta: uma com o nó da sub-rede de comunicação e a outra com o sistema. Para sistemas que utilizem pro



protocolos de comunicação é preciso construir implementações diferentes para os mesmos (no lado da comporta) e para o bloco tradutor. Como os protocolos de comunicação são, em geral, divididos em níveis, o próximo passo de detalhamento da estrutura do software é o definir os blocos nestes níveis.

A divisão dos blocos em níveis resultaria no seguinte:



Nesta fase do projeto, a Comporta deverá ser utilizada para a interligação do Sistema DEC-1091, da UFPE, a uma rede pública. Os protocolos de comunicação do DEC-10 são denominados globalmente como ANF-10. A seguir, serão introduzidos os conceitos de Processos Concorrentes e de Monitores e, logo em seguida, a estrutura dos diversos módulos que compõem o "software" da comporta, que teve como base a proposta apresentada em [LULA 81] e [BOCH 79].

Dois ou mais processos são ditos concorrentes se as suas execuções se sobrepõem no tempo [HANS 73]. Dois processos são ditos disjuntos se forem independentes um do outro, isto é, se em nenhum ponto de suas execuções devem esperar pelo outro para prosseguir. Muitas vezes os processos são não disjuntos e em

bora grande parte de suas atividades possam ser desempenhadas em paralelo, necessitam, num certo ponto, trocar informações para prosseguir. Este mecanismo de "ajustamento" é denominado de "SINCRONIZAÇÃO".

Diversos mecanismos de sincronização foram propostos. Um deles são os Monitores que tentam unificar a área de sincronização de processos com a de "tipos abstratos de dados".

**Monitor** é um conjunto de procedimentos que operam sobre variáveis comuns a vários processos. Segue a idéia básica de tipos abstratos de dados que é a ENCAPSULAÇÃO dos dados, isto é, as únicas operações permitidas sobre o objeto são as incorporadas na definição do tipo de dados abstrato. Um objeto definido pelo monitor representa variáveis comuns a processos paralelos e só poderão ser acessadas pelos processos através de chamadas aos procedimentos do monitor.

Cada um dos níveis de um protocolo são entidades independentes, num certo sentido, mas que fornecem serviços ao nível imediatamente superior. Portanto, podemos visualizar o "software" da comporta como sendo composto por, no mínimo, um processo para cada módulo (nível), onde a sincronização entre eles é feita através das interfaces entre os mesmos.

Com o uso de **monitores** para implementar a troca de interfaces entre os níveis, teríamos para cada nível um processo com acesso, externamente, a quatro monitores: o monitor de transmissão de interfaces para o nível inferior, o monitor de recepção de interfaces do nível inferior e os monitores de transmissão e de recepção de interfaces com o nível superior.

MTX<sub>n</sub> - Monitor de Transmissão do nível n

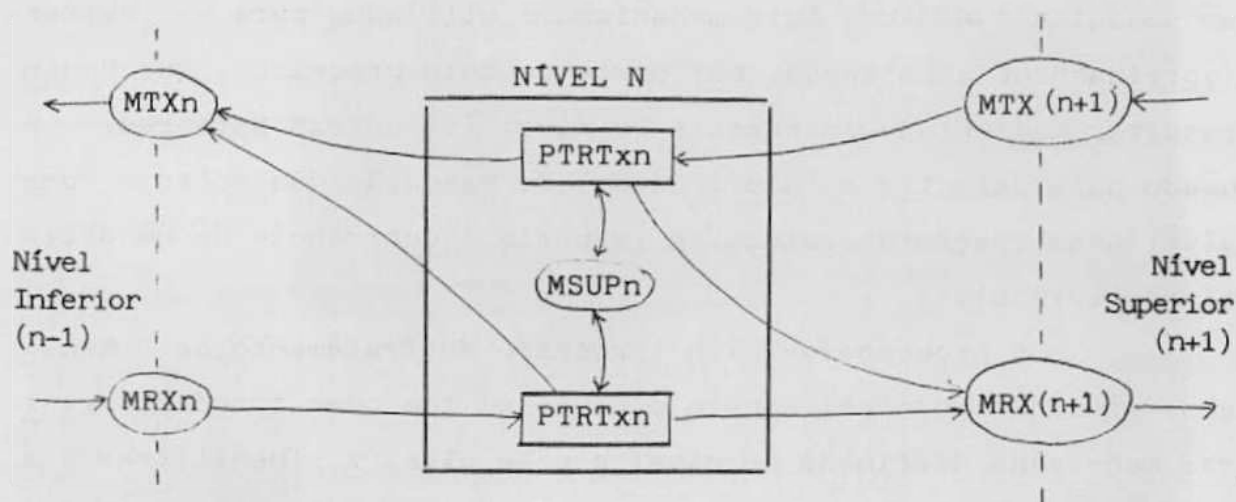
MRX<sub>n</sub> - Monitor de Recepção do nível n

MTX<sub>(n+1)</sub> - Monitor de Transmissão do nível (n+1)

MRX<sub>(n+1)</sub> - Monitor de Recepção do nível (n+1)

O nível N pode ser implementado não como um único processo, mas como alguns. A divisão mais natural é entre um processo que trate a recepção de interfaces do nível superior, e de outro que trate a recepção de interfaces do nível inferior. Neste caso, é preciso, ao menos, mais um monitor interno ao nível, para sincronizar estes dois processos.

A implementação de cada nível de protocolo será feita basicamente do seguinte modo:



(Note que círculos envolvem monitores e retângulos envolvem processos).

Os arcos direcionados representam fluxo de dados e não de controle. Os monitores são sempre elementos passivos, enquanto que os processos são os elementos ativos. Como citado no item anterior, a comunicação entre níveis é feita através de quatro monitores. Cada um destes monitores consiste, basicamente, de uma fila de mensagens (comandos ou respostas) originadas em um nível e destinadas a um dos níveis adjacentes. Ou seja, um dos níveis "produz" mensagens e o outro "consome" mensagens de acordo com o sentido dos arcos. As operações permitidas sobre esta fila são a de "enfileiramento" (colocar na fila) e a de retirada da fila. Em alguns casos, são permitidas também outras operações e/ou fun

ções associadas. Estas serão detalhadas na descrição do monitor específico.

Em MSUP<sub>n</sub> (Monitor de Supervisão do nível n) são guardadas as informações referentes ao estado de cada uma das conexões do nível. As operações permitidas sobre este monitor são, basicamente, de consulta e de atualização destas informações. Algumas consultas podem bloquear o acesso a estas informações por parte de um outro processo, até que o processo que fez a primeira consulta a atualize. Este mecanismo é utilizado para evitar "corridas" de atualização por parte de dois processos, que podem resultar num estado incoerente do nível (em outras palavras, é usado para garantir a "atomicidade" da execução das ações e possível atualização de estado em resposta à ocorrência de um determinado evento).

Os processos PTRTxn (Processo de Tratamento de Mensagens enviadas do nível x para o nível n) têm como funções: retirar mensagens destinada ao nível n pelo nível x; identificar a mensagem (análise sintática da mesma); determinar qual evento foi disparado; consultar o estado do nível (em MSPUn) e executar as ações correspondentes que podem consistir, além da atualização do estado, em transmissões de mensagens para o nível inferior e/ou para o nível superior.

É comum, em protocolos de comunicação, a existência de temporizadores para limitar o tempo de espera pela ocorrência de um dado evento (em geral, a chegada de uma mensagem de reconhecimento ou resposta). Para a implementação destes temporizadores é preciso fazer uso de um relógio de "hardware", que gere pedidos de interrupção em intervalos regulares de tempo, correspondentes a uma unidade de tempo. O tratamento deste pedido de interrupção consistiria na decretação das variáveis, que apresentam os diversos temporizadores. Se um destes temporizadores for decrementado até  $\emptyset$  (correspondendo ao estouro de temporiza



ção), então, o nível correspondente deverá ser notificado da ocorrência deste evento.

Seguindo o esquema apresentado para a recepção de mensagens do nível inferior ou do superior, a rotina de tratamento de interrupção do relógio, ao detectar o estouro de um determinado temporizador insere, num monitor especial, o MTMRn, a indicação correspondente. Haverá, então, para cada nível que faça temporização um Processo de Tratamento destes eventos (PTMRn) que possui funções análogas às dos PTRTxn.

Apresentamos nos anexos 1 e 2 a estrutura de cada um dos níveis, as funções dos diversos elementos (processos e monitores) e as suas estruturas de dados e operações de acesso (no caso dos monitores).

#### 4. ESTÁGIO ATUAL E CONCLUSÃO

O atraso na previsão inicial para a conclusão do projeto foi decorrente, principalmente, da ausência de um grupo estável com uma maior dedicação ao projeto, uma vez que o trabalho, embora, não demasiado complexo, envolve algumas novidades e um volume considerável de trabalho. A equipe de apoio consistiu basicamente de estudantes da graduação em Ciência da Computação e do Mestrado em Informática. Boa parte do tempo foi gasto em familiarização com o problema. Além do mais, sendo estudantes e sem nenhum compromisso financeiro com o projeto, suas dedicações variaram de acordo com a solicitação dos professores, nos respectivos cursos. De qualquer modo, todo o trabalho encontra-se esquemático e, com mais alguns meses de trabalho, é possível chegar à conclusão, incluindo a elaboração de relatórios mais detalhados e completos.

De acordo com as etapas apresentadas na proposta do projeto, a situação atual do projeto é a seguinte:

1ª ETAPA - Estudo da Recomendação do CCITT referente aos níveis 1,2 e 3 do protocolo X.25.

**CONCLUÍDA**

2ª Etapa - Especificação formal dos níveis 2 e 3 do protocolo X.25, através de Gramáticas de Transmissão [ TENG 80].

**CONCLUÍDA** (sujeita a alterações).

3ª Etapa - Programação da Comporta no SDE-40.

**REALIZADO CERCA DE 40% DO TOTAL**

**EM TERMOS DE X.25 FOI REALIZADO 70% DO TOTAL**

4ª Etapa - Testes e Acertos Finais da Comporta

**REALIZADOS APENAS OS TESTES INDIVIDUAIS REFERENTES ÀS PARTES JÁ PROGRAMADAS.**

5ª Etapa - Estudo para extensão da utilização da comporta a outro computador.

**NAO INICIADA**

A grande vantagem da construção de uma comporta do tipo proposto reside na facilidade que proporciona à conexão de um computador qualquer a uma Rede Pública de Comutação de Pacotes . Todos os níveis de protocolo necessários ao acesso à rede estariam já prontos, restando ao usuário a tarefa de implementar os níveis superiores de acordo com os serviços que deseja utilizar. A outra grande vantagem é a redução do tempo de processamento, do processador principal, em relação ao que ele consumiria, caso tivesse que cuidar de todos os níveis do protocolo. Em síntese, facilitando a conexão de um computador qualquer a uma rede pública, é de utilidade a toda e qualquer instituição que deseje conectar seus sistemas computacionais à mesma.

## 5. AGRADECIMENTOS

Este trabalho tem contado com a colaboração significativa de outros membros do Grupo de Redes de Computadores e Sistemas Distribuídos da UFPE (REDIS/UFPE), tais como: Clylton J. G.

Fernandes, Emílio de B. Lucena, Ivanilton L. Silva, José Rodrigues, Ruy J.G. Queiroz e o apoio inestimável do CLADI( Centro Latino-Americano de Desenvolvimento da Informática). Ressaltamos também a colaboração do Núcleo de Computação Eletrônica da UFRJ e CNPq.

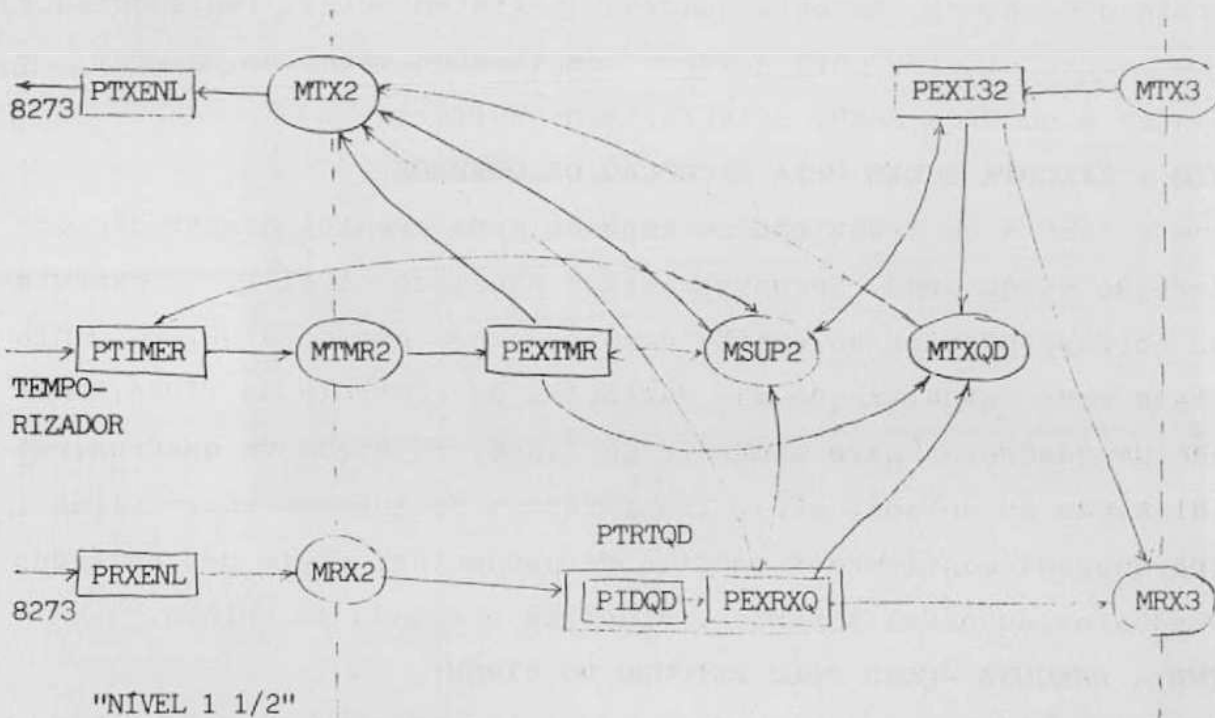
## 6. BIBLIOGRAFIA

- [ BOCH 79 ] - BOCHMANN,G.V.& JOACHIM,T. - Development and Structure of an X.25 Implementation. IEE Transactions on Software Engineering,5 (5): 429-39,Sept.,1979.
- [ CCIT 80 ] - CCITT International Telegraph & Telephone Consultative Comitee - Draft Revised Recommendation X.25. Computer Communication Review,10 (1&2): 56-129,Jan. April, 1980
- [ HANS 73 ] - HANSEN,P.B. - Concurrent Programming Concepts. Computing Surveys,5 (4): 233-45, Dec.,1973.
- [ HOLL 82 ] - HOLLANDA,T.C.B. - Como interligar computadores à rede de pacotes: o caso IBM. In: XV CONGRESSO NACIONAL DE INFORMÁTICA, Rio de Janeiro,1982. Anais. Rio de Janeiro, SUCESU, 1982. p. 799-808.
- [ LULA 81 ] - LULA,S.T.T. - Um processador de Comunicação para ligação de um Computador a uma Rede Comutada por Pacotes. Dissertação de Mestrado apresentada ao Departamento de Engenharia Elétrica da Pontifícia Universidade Católica do Rio de Janeiro, 1981.
- [ TELE 81 ] - TELEBRÁS - Telecomunicações Brasileiras S.A. - Campinas.  
REXPAC - Rede Experimental de Comunicação de Pacotes, Relatório Técnico, novembro, 1981.
- [ TENG 80 ] - TENG,A.Y. - The Transmission Grammar Model for Protocol Construction. Procedings Trends & Applications-Computer Network Protocols, p.110-20, may 1980.

[ ZIMM 80 ] - ZIMMERMANN, H. - OSI Reference Model - The ISO Model  
of Architecture for Open Systems Interconnection .  
IEEE Transactions on Communicatios, 28(4):425-32 ,  
Apr., 1980.

ANEXO 1

ESTRUTURA DE SOFTWARE DO NÍVEL 2 DO X. 25



PROCESSOS

**PTXENL = TRANSMISSÃO PARA O ENLACE**

Retira quadros da fila de transmissão do MTX2: liga, se necessário, o temporizador T1 (temporizador de quadros transmitidos). Desliga, se necessário, o temporizador T2 (temporizador de confirmação remota). Transmite os quadros através de enlace. Efetua a programação de 8273 e manipula a transferência de dados.

**PRXENL = RECEPÇÃO DO ENLACE**

Programa a recepção de um quadro (inicialização), alocando um buffer para isto. Efetua a transferência, byte a byte, dos dados recebidos do 8273 para a memória. Detecta quadros com tamanho inválido e falta de buffer.

**PTIMER = TRATAMENTO DE INTERRUPÇÕES DO TIMER**

Identifica o estouro do temporizador. Uma vez identificado, sendo o mesmo do nível 2, especifica este acontecimento no MTMR2.

**PIDQD = IDENTIFICAÇÃO DE QUADRO**

Faz a identificação dos quadros recebidos, examinando o campo de controle dos mesmos. Detecta quadros inexistentes (não implementados) Libera buffer alocado para quadros com tamanho excedido, quadros inexistentes e quadros FRMR. Seta falta de buffers.

**PEXRQ = EXECUTA AÇÕES PELA RECEPÇÃO DE QUADROS**

Contém a tabela de transição de estados para eventos disparados com a recepção de quadros. Recupera estado atual do nível 2. Executa ações correspondentes ao evento disparado e o estado atual do enlace, tais como: atualização das variáveis de controle de fluxo; transmissão de quadros ( para controle de fluxo, rejeição de quadros, reinicialização do enlace, etc); transferência de quadros entre filas ; retira quadros confirmados da fila de pendentes; envio de comandos ou respostas ao nível superior. Atualiza o estado de enlace.

**PEXTMR = EXECUTA AÇÕES PELO ESTOURO DO TIMER**

Contém a tabela de transição de estados para eventos disparados com o estouro de temporizadores. Recupera estado atual do nível 2. Executa ações correspondentes ao evento disparado e o estado atual do enlace, tais como: envio de quadros (confirmação remota, retransmissão, etc); teste de retransmissão de quadros; reinicialização do enlace; envio de comandos ou respostas ao nível superior. Atualiza estado atual do enlace.

**PEXI32 = EXECUTA AÇÕES PELA RECEPÇÃO DE COMANDOS/RESPOSTAS (OU PRIMITIVAS DO N 3) DE INTERFACE COM O NÍVEL 3**

Contém tabela de transição de estados para eventos disparados com a recepção de primitivas do nível 3. Recupera estado atual do nível 2. Executa as ações correspondentes ao evento disparado e o estado atual do enlace, tais como: transmissão de quadros (inicialização do enlace , desconexão, informação, etc); envio de comandos/ respostas ao nível superior. Atualiza o estado atual do enlace.

## MONITORES

### **MTX2 = TRANSMISSÃO DO NÍVEL 2**

Contém a fila de transmissão de quadros (I,S e U). Contém a fila de quadros (I) fora da janela, bem como a variável que indica quantos quadros I ainda podem ser transmitidos na janela. Cada elemento das filas possui os seguintes campos: indicação de comando/resposta temporizado ou não; controle; comprimento utilizado do buffer de dados e ponteiro para o mesmo.

Os procedimentos de acesso são os seguintes: Colocação de quadros S e U, Colocação de quadros I, Retirada de quadros, Adição e Reset de créditos.

### **MRX2 = RECEPÇÃO DO NÍVEL 2**

Contém fila de quadros recebidos, bem como uma variável que indica a falta de buffer de dados. Cada elemento da fila contém os seguintes campos: indicação de comando, resposta ou erro e os demais campos são idênticos aos da fila de MTX2.

Os procedimentos de acesso são: Colocação/Retirada de quadros, Atualização/Leitura da variável que indica a falta de buffer.

### **MSUP2 = SUPERVISÃO DO NÍVEL 2**

Contém a variável que indica o estado do enlace, variáveis de controle de fluxo (borda inferior da janela, N(S) do próximo quadro a ser transmitido, N(S) do próximo quadro esperado, nº de retransmissões dos quadros), temporizadores T1 (temporizador de quadros transmitidos) e T2 (temporizador para confirmação remota) e a fila de quadros não confirmados. Cada elemento da fila possui campos idênticos aos da fila de MTX2.

Os procedimentos de acesso ao MSUP2 são basicamente os seguintes: Recuperação/Atualização das variáveis de controle de fluxo, Recuperação/Atualização do estado do enlace, Coloca quadro na fila, Copia fila de quadros não confirmados para o MTX2, Disparo/Desligamento de temporizadores.

**MTXQD = TRANSMISSÃO DE QUADROS**

Monta os campos de controle e comando/resposta dos quadros a serem transmitidos (indicando se os mesmos são temporizados ou não). Para quadros de informação, coloca quadro na fila de quadros fora da janela (MTX2) e na fila de quadros não confirmados (MSUP2). Para quadros tipo S ou U, coloca quadro na fila de transmissão de quadros (MTX2). Monta o campo de informação do quadro FRMR.

**MTMR2 = TEMPORIZADORES DO NÍVEL 2**

Contém as variáveis que indicam se os temporizadores estouraram ou não.

**MRX3 = INTERFACE DO NÍVEL 2 COM O NÍVEL 3**

Contém a fila de comandos/respostas do nível 2 para o nível 3. Cada elemento da fila contém os seguintes campos: identificador do comando/resposta, ponteiro e tamanho utilizado do buffer de dados.

Os procedimentos de acesso são os de Colocação e Retirada da fila.

**MTX3 = INTERFACE DO NÍVEL 3 COM O NÍVEL 2**

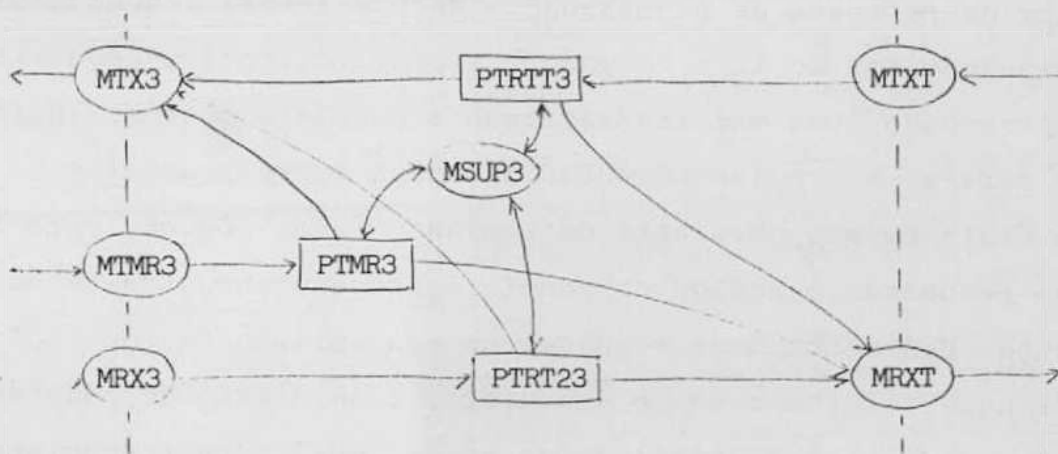
Contém fila de comandos/respostas do nível 3 para o nível 2. Os campos de cada elemento da fila assim como os procedimentos de acesso são idênticos aos de MRX3.



## A N E X O 2

### ESTRUTURA DE SOFTWARE DO NÍVEL 3 DO X.25

A estrutura do nível 3 em termos de processos e monitores é a que se segue:



O nível 3 do X.25 fornece uma conexão lógica entre pares de "hosts", denominada de CANAL LÓGICO.

Como vimos, anteriormente, o nível 2 possui uma única conexão, o ENLACE. No caso do nível 3, ao invés, é permitida a existência de 4.095 canais lógicos. Na prática, é utilizada apenas uma parcela deste total, estabelecida quando da subscrição de serviços à Rede Pública.

Embora cada um destes canais lógicos tenham um mesmo comportamento, o estado de cada um deles, em geral, difere um do outro, num dado instante. Daí a necessidade de se guardar, separadamente, o estado de cada um deles, embora haja uma única cópia da tabela de transição de estados e das rotinas associadas.

Após estas considerações gerais, vejamos as funções de cada um dos elementos do nível 3:

#### **MRX3 = MONITOR DE RECEPÇÃO DO NÍVEL 3**

Em termos de nível 3, o único procedimento acessível é o de retirada de pacotes ou indicações/respostas do nível 2.

A única estrutura de dados existentes neste monitor é uma fila que contém os seguintes campos: tipo de interface, comprimento e ponteiro do buffer de dados.

### **PTRT23 = PROCESSO DE TRATAMENTO DE MENSAGENS ENVIADAS DO NÍVEL 2 PARA O NÍVEL 3**

Este processo pode ser subdividido em diversas etapas. A primeira consiste na retirada de mensagens de MRX3 e identificação do seu tipo. Se a mensagem for do tipo pacote é preciso, então, identificar o pacote recebido. Uma vez identificado o pacote e o canal lógico a que se refere, é preciso executar as ações correspondentes à ocorrência deste evento para este determinado canal lógico. Para isto, é preciso recuperar o estado do canal lógico que encontra-se no MSUP3. A execução das ações pode resultar na transmissão de indicação para o nível superior (no caso, o tradutor), transmissão de pacotes para a entidade parceira de nível 3 (no nó da rede) e/ou transmissão de comando para o nível inferior (nível 2).

### **MTXT = MONITOR DE TRANSMISSÃO DO TRADUTOR**

Contém uma fila de comandos do tradutor para o nível 3. O único procedimento de acesso permitido ao nível 3 é o de remoção de elemento da fila. Cada elemento possui os seguintes campos: Tipo de comando, Identificação do C.L., Comprimento e ponteiro para o buffer de dados.

### **MRXT = MONITOR DE RECEPÇÃO DO TRADUTOR**

Contém uma fila de indicações/respostas do nível 3 para o tradutor. O único procedimento de acesso visível ao nível 3 é o de colocação na fila de mensagens existente em MRXT.

Um elemento desta fila é composto por: Tipo de indicação/resposta e os demais campos são idênticos aos de MTXT.

### **PTRT3 = PROCESSO DE TRATAMENTO DE MENSAGENS ENVIADAS DO TRADUTOR PARA O NÍVEL 3**

Este processo tem funções análogas ao PTRT23. Inicialmente deve identificar o comando recebido do módulo tradutor (retirado de MTXT) e,

em seguida, de acordo com o estado do canal lógico (se houver algum específico) ou do nível, executar as ações correspondentes: atualização do estado, transmissão de pacotes para a entidade de nível 3 parceira, transmissão de comandos para o nível 2 e/ou de respostas para o módulo tradutor.

### **MSUP3 = MONITOR DE SUPERVISÃO DO NÍVEL 3**

A estrutura de dados deste monitor consiste numa tabela com uma entrada para cada canal lógico. Cada uma destas entradas corresponde ao ESTADO de um determinado canal lógico. Estas entradas compõem-se de:

1. sub-estados
2. parâmetros, tais como: número de sequência do próximo pacote a ser transmitido, número de sequência do próximo pacote a ser aceito, número de retransmissões de pacotes, etc.
3. temporizadores: variáveis que indicam o instante de temporizações usadas em estados temporizados como o de espera pela confirmação de estabelecimento de conexão.
4. fila de pacotes não confirmados: devido ao mecanismo de janela é possível transmitir um certo número de pacotes sem que tenha chegado a confirmação de recebimento do primeiro deles. Caso não seja efetuado este reconhecimento, é preciso retransmitir estes pacotes e como são característicos de cada canal lógico, daí, a necessidade de mantê-los na tabela.

Os procedimentos de acesso ao monitor são, basicamente, os seguintes:

- i) recuperação/atualização de estado e parâmetros
- ii) pedido de canal lógico disponível
- iii) disparo/desligamento de temporizador
- iv) decretação dos temporizadores (acessível, unicamente, à rotina de tratamento de interrupção do relógio do sistema).
- v) atualização de pacotes confirmados
- vi) retransmissão de pacotes ainda não confirmados

### **MTX3 = MONITOR DE TRANSMISSÃO DO NÍVEL 3**

Contém uma fila de comandos e pacotes prontos para transmitir e um conjunto de Filas de Pacotes Fora da Janela sendo uma para cada canal lógico.

Associada a cada canal lógico há também uma variável que indica o número de créditos de transmissão de pacotes de dados.

Os elementos da fila de comandos e pacotes prontos para transmitir compõem-se dos mesmos campos da fila de MRX3. Enquanto que os elementos das Filas de Pacotes Fora da Janela compõem-se apenas do comprimento e do ponteiro para o buffer de dados.

Esta fila conteria os pacotes de dados que o tradutor solicitou a transmissão, mas que não foram inseridas na fila de transmissão por falta de créditos suficientes.

Ao receber novos créditos para o canal lógico, seriam retirados alguns pacotes (correspondentes ao crédito) desta fila e colocados na fila de transmissão.

Os procedimentos de acesso (visíveis ao nível 3) deste monitor são:

i) colocação de comandos e pacotes (não de dados) na fila de transmissão.

ii) colocação de pacotes de dados: caso haja créditos e não haja nenhum pacote na Fila de Pacotes fora da janela, o pacote é colocado diretamente na fila de transmissão e o número de créditos é decrementado.

iii) atribuição de créditos: enquanto o número de créditos de transmissão for maior que zero e houver pacotes na Fila de Pacotes fora da janela, é transferido um pacote desta fila para a fila de transmissão e o número de créditos é decrementado.

### **MTMR3 = MONITOR DOS TEMPORIZADORES DE NÍVEL 3**

Quando a rotina de tratamento de interrupção do relógio do sistema deteta a ocorrência do estouro de temporização, de um dos temporizadores do nível 3, esta insere na fila de MTMR3 a identificação do temporizador que estourou, juntamente, com a identificação do canal

lógico a que se refere.

Há também um procedimento de retirada de elementos desta fila.

**PTMR3 = PROCESSO DE TRATAMENTO DE ESTOURO DE TEMPORIZAÇÕES DO NÍVEL 3**

Este processo retira indicações de estouro de temporização de MTMR3, recupera o estado do canal lógico a que se refere a indicação e executa a rotina correspondente que, além de atualizar o estado do canal lógico poderá, também, solicitar a transmissão de pacotes ou comandos para o nível 2 e/ou indicações para o tradutor.