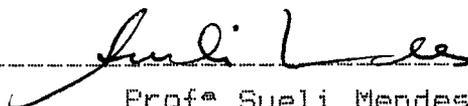


DESENVOLVIMENTO DE UM PROTOCOLO DE COMUNICAÇÃO ENTRE DOIS
SISTEMAS DISTRIBUÍDOS

Vera Lucia Lopes Monteiro

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE
PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE
JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO
DO GRAU DE MESTRE EM CIÊNCIAS (M. Sc.) EM ENGENHARIA DE SISTEMAS
E COMPUTAÇÃO

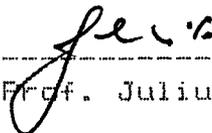
Aprovada por:



Profª Sueli Mendes
(Presidente)



Prof. Valmir Carneiro Barbosa



Prof. Julius César Barreto Leite

RIO DE JANEIRO, RJ - BRASIL
ABRIL DE 1988

MONTEIRO, VERA LUCIA LOPES

Desenvolvimento de um Protocolo de Comunicação
entre dois Sistemas Distribuídos [Rio de Janeiro]
1988.

XVI, 226 p. 29,7 cm (COPPE/UFRJ, M. Sc.,
Engenharia de Sistemas e Computação, 1988).

Tese - Universidade Federal do Rio de Janeiro,
COPPE.

1. Sistemas Distribuídos I. COPPE/UFRJ
II. Título (série).

Esta tese é dedicada:

- à memória de meu pai, Álvaro Vieira Monteiro;
- à minha mãe, Elisa da Purificação Lopes de Oliveira Monteiro.

AGRADECIMENTOS

À Professora Sueli Mendes, pela orientação no desenvolvimento deste trabalho.

À toda equipe de Furnas Centrais Elétricas S/A e da COMSIP ENGENHARIA S/A que participa deste projeto e que me apoiou e incentivou. Em especial à Sérgio Portela Romano Cotrim que me auxiliou no decorrer de todo o projeto e à Mário César S. de Oliveira pela ajuda nos assuntos relacionados ao Sistema do Despacho Central.

Aos amigos Luiza e Guilherme Fernandes de Souza pelo apoio e pela ajuda na confecção da tese.

À Francisco José de Macedo Jr. e Gerson Quintanilha pela elaboração dos desenhos.

À Liane de Jesus Costa pela ajuda na edição.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para obtenção do grau de Mestre em Ciências (M. Sc.)

**DESENVOLVIMENTO DE UM PROTOCOLO DE COMUNICAÇÃO ENTRE DOIS
SISTEMAS DISTRIBUÍDOS**

VERA LUCIA LOPES MONTEIRO

ABRIL, 1988

Orientadora: Sueli Mendes

Programa: Engenharia de Sistemas e Computação

Este trabalho trata da implementação de um protocolo de comunicação entre dois sistemas que fazem parte do Programa de Supervisão e Controle do Sistema Elétrico Nacional. Estes dois sistemas equiparão os Centros de Operação Regionais e o Despacho Central de Furnas Centrais Elétricas.

Visando um desenvolvimento estruturado e disciplinado dos Centros de Operação Regionais, investiu-se na elaboração de software básico e de suporte que facilitasse a implementação de um sistema aplicativo em tempo real numa arquitetura distribuída.

Os dois capítulos iniciais deste trabalho apresentam o resultado desses esforços, ou seja, as ferramentas disponíveis e o ambiente de hardware e de software dos sistemas em questão.

O principal objetivo deste trabalho é apresentar a comunicação entre esses dois sistemas, detalhando a implementação do protocolo do lado dos Centros de Operação Regionais. O protocolo desenvolvido utiliza a arquitetura proposta pela ISO (International Standard Organization) no Modelo de Referência OSI (Open Systems Interconnection), sendo que os três níveis inferiores seguem a Recomendação X.25 do CCITT (Comitê Consultivo Internacional em Telegrafia e Telefonia). Para os níveis superiores foi implementado um protocolo específico visando a aplicação em questão.

Abstract of Thesis presented to COPPE/UFRJ as partial fulfillment of the requirements for the degree of Master of Science (M. Sc.)

DEVELOPMENT OF A COMMUNICATION PROTOCOL BETWEEN TWO DISTRIBUTED SYSTEMS

VERA LUCIA LOPES MONTEIRO

APRIL, 1988

Advisor: Sueli Mendes

Department: System and Computation Engineering

This work presents the implementation of a communication protocol between two systems of the Supervision and Control Program of the National Power System. These systems will equip the Regional Operation Centers and the Central Dispatch of Furnas Centrais Elétricas.

Aiming at a structured and orderly development for the Regional Operation Centers, a software specification and design method was elaborated, as well as basic and support software that simplified the implementation of a real time application system on a distributed architecture.

The first two chapters of this work present the available tools and hardware and software environment of the systems in question.

The main purpose of this work is to discuss the communication between the two systems providing details of the protocol implementation from the view of the Regional Operation Centers. The protocol developed is based on the Reference Model of Open Systems Interconnection (OSI) from ISO (International Standard Organization) with the bottom three levels as defined by the X.25 Recommendation from CCITT (Consultative Committee for International Telegraph and Telephone). For the top levels a specific protocol was created to suit the application in question.

ÍNDICE

CAPÍTULO I - INTRODUÇÃO	1
CAPÍTULO II - DESENVOLVIMENTO DE SOFTWARE PARA SISTEMAS APLICATIVOS EM TEMPO REAL	6
II.1 - Introdução	6
II.2 - Metodologias de Projeto de Software	7
II.2.1 - Conceito de Abstração	7
II.2.2 - Modelo Algébrico e Modelo de Abstração de Dados e Procedimentos	8
II.2.3 - Abordagem Convencional e Abordagem Operacional para Desenvolvimento de Sistemas	9
II.3 - Descrição da Metodologia Utilizada	15
II.3.1 - 1ª Fase - Análise dos Requisitos	15
II.3.2 - 2ª Fase - Especificação Funcional	16
II.3.3 - 3ª Fase - Projeto ou Análise Orgânica	22
II.3.4 - 4ª Fase - Implementação e Testes	26
II.4 - Comunicação e Sincronização entre Processos	27
II.4.1 - Núcleo do Sistema Operacional	29
II.4.2 - Comunicação Através da Via de Interconexão dos Processadores	32

CAPÍTULO III - DESCRIÇÃO DOS CENTROS DE OPERAÇÃO REGIONAIS E DO DESPACHO CENTRAL	39
III.1 - Introdução	39
III.2 - Descrição dos Centros de Operação Regionais	41
III.2.1 - Subsistema de Comunicação com a Interface Homem-máquina (SCHM)	43
III.2.2 - Subsistema de Comunicação com o Sistema Elétrico Regional (SCER)	48
III.2.3 - Subsistema de Comunicação com o Sistema Central (SCSC)	52
III.2.4 - Subsistema de Observação do Sistema Computacional (SOSC)	54
III.3 - Descrição do Despacho Central	55
III.3.1 - Subsistema de Interface Homem-Máquina (IHM)	58
III.3.2 - Subsistema de Entradas e Saídas Locais (ESL)	60
III.3.3 - Subsistema de Comunicação de Dados (COD)	62
III.3.4 - Subsistema de Recuperação (REC)	64
CAPÍTULO IV - ESTRUTURAÇÃO DOS NÍVEIS DO PROTOCOLO DE COMUNICAÇÃO COR <=> DC	67
IV.1 - Introdução	67
IV.2 - Níveis do Modelo de Referência OSI	70
IV.2.1 - Nível Físico (Nível 1)	71

IV.2.2 - Nível de Enlace (Nível 2)	71
IV.2.3 - Nível de Rede (Nível 3)	72
IV.2.4 - Nível de Transporte (Nível 4)	72
IV.2.5 - Nível de Sessão (Nível 5)	73
IV.2.6 - Nível de Apresentação (Nível 6)	73
IV.2.7 - Nível de Aplicação (Nível 7)	73
IV.3 - Implementação do Protocolo COR <=> DC	74
IV.3.1 - Nível Físico (Nível 1)	77
IV.3.2 - Nível de Enlace (Nível 2)	77
IV.3.3 - Nível de Canal Lógico (Nível 3)	79
IV.3.4 - Nível de Segmentação (Nível 4)	82
IV.3.5 - Nível de Sessão (Nível 5)	83
IV.3.6 - Nível de Apresentação (Nível 6)	84
IV.3.7 - Nível de Aplicação (Nível 7)	87
CAPÍTULO V - IMPLEMENTANDO O NÍVEL DE CANAL LÓGICO	89
V.1 - Introdução	89
V.2 - Interface com o Nível 4	89
V.2.1 - Interface de Transmissão N4 -> N3	90
V.2.2 - Interface de Recepção N3 -> N4	92
V.3 - Interface com o Nível 2	94
V.3.1 - Comandos do Nível 3 (CMD_N3)	95
V.3.2 - Requisições do Nível 3 (REQ_N3)	97
V.3.3 - Indicações do Nível 3 (IND_N3)	97
V.3.4 - Respostas do Nível 3 (ANS_N3)	97
V.3.5 - Respostas do Nível 2 (ANS_N2)	98
V.3.6 - Indicações do Nível 2 (IND_N2)	98

V.4 - Implementação do Nível 3	99
V.4.1 - Máquina de Estados do Nível	103
V.4.1.1 - Descrição das Transições da Máquina de Estados do Nível	109
V.4.1.2 - Classificação dos Estados da Máquina de Estados do Nível nas suas Sub-fases	121
V.4.2 - Máquinas de Estados dos Canais Lógicos	122
V.4.2.1 - Descrição das Transições das Máquinas de Estados dos Canais Lógicos	127
V.4.2.2 - Classificação dos Estados das Máquinas de Estados dos Canais Lógicos nas suas Sub- fases	131
V.4.3 - Organização do Nível 3	132
V.4.3.1 - Módulo Responsável pelas Máquinas de Estados	134
V.4.3.2 - Módulo Monitora VGI	139
V.4.3.3 - Módulo Monitora Recursos	140
V.4.3.4 - Módulo de Recepção de Eventos Gerados pelo Nível 2	140
V.4.3.5 - Módulo de Transmissão de Pacotes ao Nível 2	141
V.4.3.6 - Módulo de Recepção de Segmentos	143
CAPÍTULO VI - IMPLEMENTANDO O PROTOCOLO DE ALTO NÍVEL	145
VI.1 - Introdução	145
VI.2 - Implementação do Nível 5	148

VI.2.1 - Módulo Responsável por Sessões no Sentido COR -> DC	149
VI.2.2 - Módulo Responsável por Sessões no Sentido DC -> COR	152
VI.2.3 - Módulo de Recepção de Mensagens Lógicas do DC	154
VI.2.4 - Módulo de Armazenamento de Números de Seqüência	154
VI.3 - Implementação do Nível 7	155
VI.3.1 - Procedimentos para os Dados de Iniciação	157
VI.3.2 - Procedimentos para os Dados de Controle Elétrico	159
VI.3.3 - Procedimentos para os Dados Adquiridos	160
VI.3.4 - Procedimentos para as Mensagens Operacionais	161
VI.4 - Especificação Funcional do Protocolo de Comunicação COR -> DC	162
REFERÊNCIAS BIBLIOGRÁFICAS	202
APÊNDICE A - EXTENSÕES DAS MENSAGENS	205
APÊNDICE B - ESTRUTURAÇÃO DO NÍVEL DE SEGMENTAÇÃO	223
APÊNDICE C - CÓDIGOS DE RETORNO AO USUÁRIO DA INTERFACE COM O NÍVEL 2	225

ÍNDICE DAS FIGURAS

CAPÍTULO II

II.1 - Abordagem Convencional	11
II.2 - Abordagem Operacional	13
II.3 - Comunicação Síncrona entre Processos	30
II.4 - Sessão Direta	36
II.5 - Sessão Normal	38

CAPÍTULO III

III.1 - Hierarquia dos Sistemas do Programa de Supervisão e Controle do Sistema Elétrico Nacional	40
III.2 - Arquitetura do Sistema dos Centros de Operação Regionais	42
III.3 - Ligação DC \Leftrightarrow COR	53
III.4 - Arquitetura do Sistema do Despacho Central	57
III.5 - Arquitetura do Subsistema de Comunicação de Dados do DC	63

CAPÍTULO IV

IV.1 - Arquitetura em Níveis - Modelo de Referência OSI	68
IV.2 - Arquitetura de Software X Arquitetura de Hardware	76
IV.3 - Tabela das Mensagens Trocadas na Ligação DC \Leftrightarrow COR	85

IV.4 - Formato Geral das Mensagens	86
IV.5 - Unidades de Transferência dos Níveis do Protocolo	88

CAPÍTULO V

V.1 - Interface de Transmissão N4 -> N3	91
V.2 - Interface de Recepção N3 -> N4	93
V.3 - Interface N3 <=> N2	96
V.4 - Fases e Sub-fases da Máquina de Estados do Nível ..	101
V.5 - Fases e Sub-fases da Máquina de Estados de Canal Lógico	102
V.6 - Máquina de Estados do Nível 3 - 1ª Parte	107
V.7 - Máquina de Estados do Nível 3 - 2ª Parte	108
V.8 - Máquina de Estados de um Canal Lógico	126
V.9 - Organização do Objeto COD	133

CAPÍTULO VI

VI.1 - Organização dos Níveis 5, 6 e 7	146
--	-----

APÊNDICE B

B.1 - Organização do N4	224
-------------------------------	-----

APÊNDICE C

C.1 - Códigos de Retorno ao Usuário do N2	226
---	-----

ÍNDICE DE DSA

CAPÍTULO II

II.1 - Diagrama de Sequência de Ações Exemplo	20
---	----

CAPÍTULO VI

VI.1 - Recepção de Comando de Iniciação do DC e Envio da Resposta	163
VI.2 - Recepção de Comando de Controle do DC e Envio da Resposta	167
VI.3 - Recepção do Resultado de Controle do SCER	171
VI.4 - Recepção da Resposta ao Resultado de Controle ...	175
VI.5 - Recepção de Dados XXX (DIG, ANL, TOT e SOE) do SCER	179
VI.6 - Recepção de Resposta do DC do Tipo de Dado XXX (DIG, ANL, TOT e SOE)	182
VI.7 - Recepção de Comando do Tipo de Dado XXX (DIG, ANL, TOT e SOE) e Envio da Resposta	185
VI.8 - Transmissão de Mensagens de CRT para o DC	187
VI.9 - Recepção de Mensagens de CRT do DC	192
VI.10 - Recepção de Comando do Tipo de Dado de Mensagens de CRT	198

CAPÍTULO I

INTRODUÇÃO

O setor elétrico brasileiro está implantando o Programa de Supervisão e Controle do Sistema Elétrico Nacional que será formado por um sistema hierárquico ligando os centros de operação de sistema das dez principais empresas geradoras de energia elétrica a um centro administrado pela ELETROBRÁS. No terceiro nível desta hierarquia aparecem os centros de operação regionais das empresas.

O escopo de desenvolvimento deste trabalho abrange os sistemas que constituirão o centro de operação de sistema, também chamado de Despacho Central, e os centros de operação regionais de Furnas Centrais Elétricas.

O sistema que equipará o Despacho Central é um sistema de grande porte distribuído, ou seja, é um agrupamento ("cluster") de computadores VAX 11/780 que se comunica com uma série de equipamentos, tais como, controladores de entrada e saída, controladores de comunicação, geradores de vídeo.

O sistema que equipará os centros de operação regionais é uma rede local de computadores baseada na tecnologia ETHERNET, ou seja, é composto por uma série de microprocessadores que se comunicam entre si via um barramento serial. A finalidade deste sistema, adquirir, supervisionar e controlar dados de um

sistema elétrico, configura uma aplicação em tempo real para controle de processos. As equipes que vêm participando deste projeto investiram muito, no início, no desenvolvimento de hardware e de ferramentas de software que possibilitassem a implantação de um sistema estruturado, de alta disponibilidade e confiabilidade, de fácil expansão e de degradação controlada. O sistema proposto é um sistema distribuído que se caracteriza pela distribuição de processamento, dados, programas e controle (KIRNER e MENDES [8]).

O objetivo deste trabalho é o desenvolvimento de um protocolo de comunicação entre esses dois sistemas, apresentando a implementação do mesmo do lado dos centros de operação regionais.

O protocolo implementado segue a estruturação do Modelo de Referência OSI para sistemas abertos da ISO. Os três níveis inferiores seguem a Recomendação X.25 do CCITT (CCITT [3]). Para os níveis superiores foi implementado um protocolo específico visando a aplicação em questão.

A implementação projetada procurou seguir os princípios determinados pela ISO a serem considerados na definição das camadas de um protocolo e das funções por elas executadas (ZIMMERMANN [17]). Estes princípios têm como principal objetivo a estruturação do projeto de modo a torná-lo simples, com interfaces de interações minimizadas, com as funções de cada nível bem definidas de maneira a facilitar um reprojeto sem afetar as interfaces com os outros níveis e os serviços oferecidos. Levando-se em conta estes princípios e as

particularidades da arquitetura do sistema chegou-se às soluções adotadas.

O ambiente de desenvolvimento é constituído por dois processadores cada um ligado ao Despacho Central via um canal de comunicação. Por outro lado, estes processadores encontram-se também ligados entre si e aos outros processadores da rede local que constitui o sistema regional, via um barramento serial. Logo, os principais objetivos deste protocolo são:

- Permitir que os processos da aplicação, localizados em diversos processadores distintos configurando uma aplicação distribuída, possam se comunicar com a aplicação do sistema do Despacho Central sem que haja perda de dados;
- Dividir o fluxo das informações enviadas ao Despacho Central pelos dois canais de comunicação;
- Manter, mesmo em modo degradado, a ligação entre os dois sistemas operante.

Cabe aqui ressaltar a importância do êxito deste projeto não só no que se refere à parcela dele que será aqui apresentada, mas a todo o sistema desenvolvido. Este sistema, totalmente nacional (software e projeto e industrialização da arquitetura de hardware), representa um grande passo para a nossa independência e total domínio da tecnologia de supervisão e controle de dados de um sistema elétrico, tecnologia esta amplamente dominada até a presente data apenas por grandes

potências mundiais, tais como, Alemanha, França, Estados Unidos, Suécia.

A seguir é apresentado um detalhamento da organização deste trabalho.

No capítulo II - "*Desenvolvimento de Software para Sistemas Aplicativos em Tempo Real*" - é feito um levantamento de metodologias ou métodos de desenvolvimento de software caracterizando-se aqueles que se adaptam ou não ao desenvolvimento de aplicações em tempo real numa arquitetura distribuída. O conjunto de métodos utilizado no decorrer de todo o projeto é apresentado, bem como as ferramentas de comunicação e sincronização desenvolvidas como suporte básico.

O capítulo III - "*Descrição dos Centros de Operação Regionais e do Despacho Central*" - apresenta a descrição da arquitetura de hardware e de software dos dois sistemas que constituem o ambiente de desenvolvimento deste trabalho.

No capítulo IV - "*Estruturação dos Níveis do Protocolo de Comunicação COR <=> DC*" - encontra-se uma descrição sucinta dos níveis em que o Modelo de Referência OSI é organizado. São apresentadas as características do sistema ambiente que particularizam esta implementação e a descrição dos níveis do protocolo conforme implementado.

O nível físico (nível 1) e o nível de enlace (nível 2) seguem integralmente a recomendação X.25 e nos foram fornecidos como um pacote fechado. No entanto, conhecemos bem suas

características, seu funcionamento e a interface que o mesmo oferece para comunicação com o nível superior.

O capítulo V - *"Implementando o Nível de Canal Lógico"* - apresenta a descrição da interface com o nível de enlace e da interface com o nível de segmentação, ou seja, respectivamente as interfaces com o nível inferior e superior ao nível de canal lógico (nível 3). Este capítulo descreve, também, a implementação deste nível a qual se baseia essencialmente em Máquinas de Estados ou Autômatos Finitos.

O capítulo VI - *"Implementando o Protocolo de Alto Nível"* - apresenta o protocolo de aplicação implementado (níveis 5, 6 e 7). Este protocolo é bem específico; foi criado visando atender às necessidades do sistema aplicativo, tendo em vista a arquitetura funcional do sistema que apresenta a aplicação distribuída fisicamente em relação ao protocolo de comunicação.

O apêndice A - *"Extensões das Mensagens"* - apresenta o formato das extensões (parte das mensagens que não é geral para todos os tipos de dados) das mensagens trocadas pelos dois sistemas.

O apêndice B - *"Estruturação do Nível de Segmentação"* - apresenta a organização do nível 4.

O apêndice C - *"Códigos de Retorno ao Usuário da Interface com o Nível 2"* - apresenta uma tabela explicativa dos códigos de retorno do nível 2, indicando os procedimentos que devem ser adotados pelo nível 3.

CAPÍTULO II

DESENVOLVIMENTO DE SOFTWARE PARA SISTEMAS APLICATIVOS EM TEMPO REAL

II.1 - INTRODUÇÃO

Este capítulo trata do desenvolvimento de software para sistemas distribuídos que visam acolher sistemas aplicativos do tipo tempo real.

Um sistema aplicativo tem por finalidade a solução do problema do usuário que neste caso constitui-se na supervisão e no controle de um Sistema Elétrico. O problema se apresenta, então, dependente do tempo e necessitando de um processamento contínuo. Estamos diante de um sistema aplicativo para Controle de Processos.

Há alguns anos existe um consenso geral sobre o projeto de desenvolvimento de software para qualquer que seja o sistema em questão. Deve ser seguida uma Metodologia de Projeto dividida em fases e que deve especificar para cada fase os passos a serem seguidos e as decisões a serem tomadas. A primeira preocupação, portanto, para o desenvolvimento de um sistema deve ser a escolha de uma metodologia de projeto que se adapte às características funcionais do software e do hardware do sistema a ser implementado.

No item (II.2) é feito um levantamento de algumas das metodologias de projeto existentes e para que tipo de aplicação elas se adaptam. No item (II.3) é descrito o método utilizado para o desenvolvimento do software aplicativo do sistema que constitui o ambiente de desenvolvimento discutido neste trabalho. Finalmente, no item (II.4) serão descritos os mecanismos de comunicação e sincronização de processos utilizados.

II.2 - METODOLOGIAS DE PROJETO DE SOFTWARE

II.2.1 - CONCEITO DE ABSTRAÇÃO

A abordagem do assunto Metodologias de Projeto de Software deve ser precedida por uma consideração a respeito de um conceito que é a base de toda uma família de métodos de especificação de sistemas. Estou me referindo ao conceito de ABSTRAÇÃO.

É através deste conceito que podemos definir uma metodologia como uma seqüência de fases sendo que cada fase possui um determinado grau de abstração. Em uma determinada fase pode ser feita uma descrição simplificada de um sistema que enfatize as características importantes do mesmo e ignore os detalhes desnecessários à sua compreensão em um certo nível. Ou seja, em cada fase existe o que é importante e deve ser detalhado e o que é irrelevante e deve, portanto, ser escondido.

Uma forma de abstrair é "Esconder Informação" (PARNAS [11]). Um elemento componente de um sistema em uma fase de desenvolvimento pode esconder o seu funcionamento interno dos demais elementos. Este mecanismo permite que os elementos nos quais o sistema se encontra decomposto possuam alta coesão, mas baixo acoplamento entre si.

II.2.2 - MODELO ALGÉBRICO E MODELO DE ABSTRAÇÃO DE DADOS E PROCEDIMENTOS

Uma importante linha de metodologias a ser abordada é aquela que se baseia em especificações formais ou semiformais. Dentro desta linha podemos destacar, entre outros, alguns modelos de especificação: Modelo Algébrico, Modelo de Abstração de Dados e Procedimentos, Modelos Operacionais baseados em máquinas abstratas.

Modelo Algébrico

Este método descreve as abstrações de tipos de dados e de procedimentos algebricamente. A especificação consiste na descrição algébrica da sintaxe das operações/funções e na definição dos axiomas que correspondem a essa álgebra. A sintaxe das operações/funções definem domínios e contradomínios. Os axiomas são escritos na forma de equações. Para que se possa provar que uma implementação é correta deve-se demonstrar que ela satisfaz a todos os axiomas.

Modelo de Abstrações de Dados e Procedimentos

Este modelo baseia-se principalmente no encapsulamento de dados e procedimentos em objetos de um determinado tipo e na definição das operações e funções sobre esses dados e procedimentos respectivamente. A sintaxe das operações/funções de um determinado objeto é definida da mesma forma do modelo algébrico. A diferença está na especificação da semântica das operações/funções. A semântica das operações/funções é descrita especificando-se como essas operações/funções afetam os objetos abstratos.

A utilização destes métodos é bastante conhecida e vem sendo muito publicada em artigos, mas sempre dizem respeito a modelos essencialmente matemáticos. Uma tentativa de ilustrar a importância dessas especificações na modelagem de banco de dados é apresentada por CLAYBROOK [4].

A utilização dessas especificações em sistemas que dependam essencialmente da comunicação entre objetos e da abstração de dados e procedimentos dentro de um ambiente de tempo real não nos parece a mais adequada.

II.2.3 - ABORDAGEM CONVENCIONAL E ABORDAGEM OPERACIONAL PARA DESENVOLVIMENTO DE SISTEMAS

Dentro do consenso geral de divisão do desenvolvimento de um projeto em fases bem definidas podemos destacar duas abordagens: a Convencional e a Operacional (ZAVE [16]).

A Convencional, como o próprio nome expressa, vem guiando os projetistas na percepção de problemas e na obtenção de soluções para os mesmos há muito tempo. No entanto, novas idéias para o desenvolvimento de software vêm surgindo e essas idéias não encontram espaço dentro da abordagem Convencional. Surge, então, uma abordagem alternativa, a Operacional, que procura, através de especificações executáveis e transformações programadas, facilitar a tarefa dos projetistas.

A seguir é apresentada uma descrição sucinta dessas duas abordagens.

Abordagem Convencional

A Abordagem Convencional baseia-se na decomposição do sistema em caixas pretas organizadas numa hierarquia "top down" como é mostrado na figura (II.1).

Nesta abordagem o desenvolvimento de um sistema é composto pelas seguintes fases:

Fase de Especificação de Requisitos - Nesta fase o problema a ser solucionado pelo sistema é descrito. São definidas todas as requisições do cliente, ou seja, o sistema é visto como uma grande caixa preta, importando apenas o seu comportamento externo;

Fase de Projeto - Determina a estrutura interna do sistema decompondo-o em módulos. A estrutura interna dos módulos é

ainda irrelevante, isto é, os módulos são as caixas pretas. Nesta fase é definida a interação entre os módulos de tal maneira que os requisitos do sistema sejam atendidos, e também trata-se da alocação dos recursos do sistema computacional entre os módulos componentes do sistema;

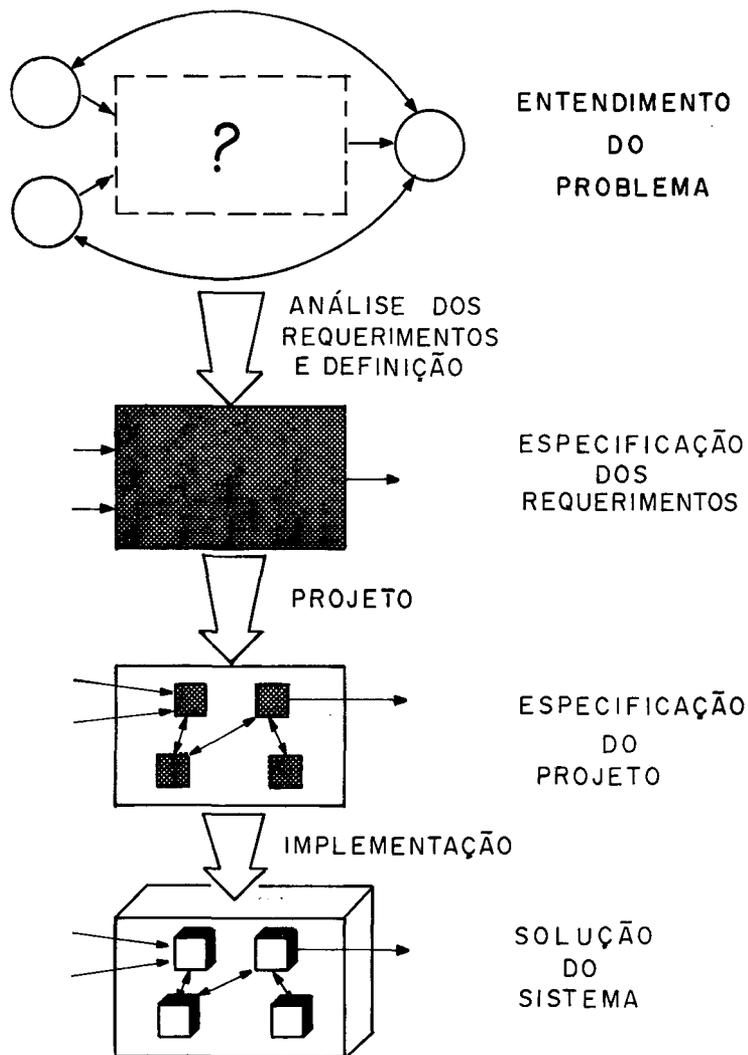


FIGURA II.1 - ABORDAGEM CONVENCIONAL

Fase de Implementação - Nesta fase a estrutura interna dos módulos é tornada explícita, ou seja, os módulos são transformados em código executável pelo sistema computacional.

Abordagem Operacional

Esta abordagem é mostrada na figura (II.2) e pode ser dividida da seguinte forma:

Especificação Operacional - Consiste na descrição do problema a ser solucionado em termos de estruturas que independem da forma de implementação. O comportamento interno dessas estruturas é explícito, enquanto que o externo é implícito, uma vez que é apenas obtido com a execução da especificação por um interpretador. Logo, a característica mais importante é que a Especificação Operacional é executável por um Interpretador;

Especificação Transformada - Consiste na transformação da especificação operacional em uma especificação orientada à implementação. As transformações introduzem representações explícitas da implementação e alocação dos recursos computacionais;

Solução - A especificação é mapeada na linguagem de implementação. O objetivo de pesquisas realizadas nesta área é que todos os problemas sejam resolvidos durante a especificação operacional e transformada para que o passo de realização do sistema seja o mais direto possível.

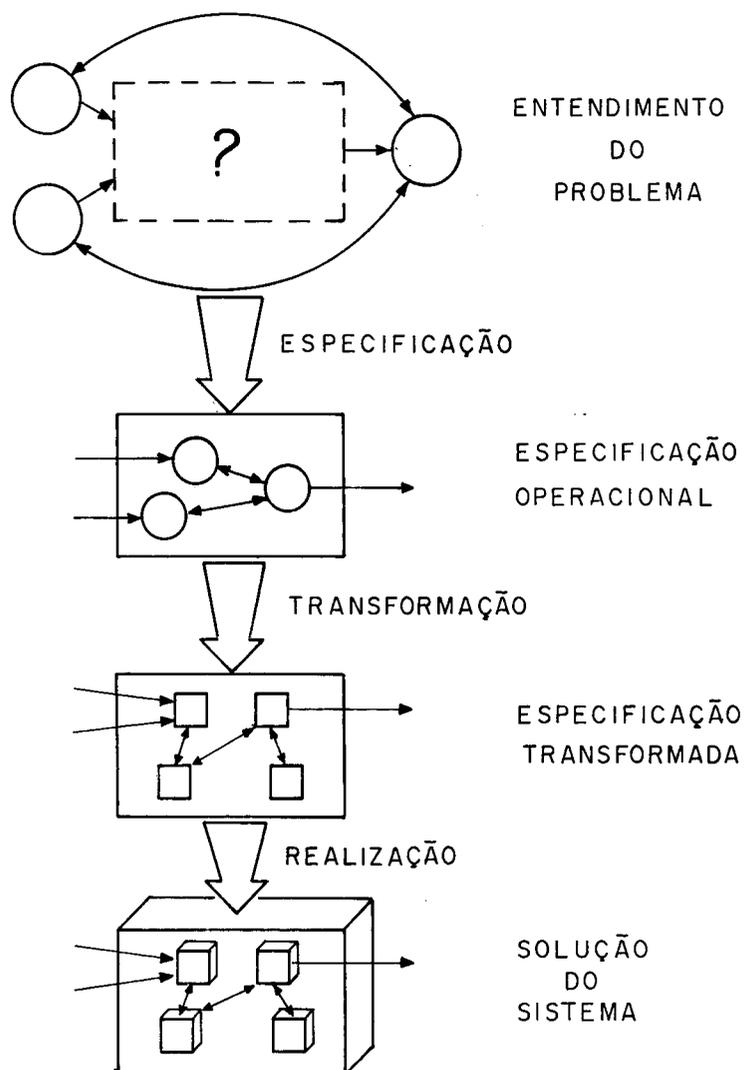


FIGURA II .2 - ABORDAGEM OPERACIONAL

A abordagem operacional é muito atraente para o desenvolvimento de sistemas em tempo real, principalmente por que possibilita a produção de protótipos num estágio prematuro do desenvolvimento do sistema. A produção de um protótipo com a abordagem convencional implica numa iteração de todo o ciclo de

desenvolvimento de software. Isso significa que é praticamente impossível se obter um protótipo mais rapidamente que o próprio produto.

Outro fato importante é que a abordagem operacional, por se basear em especificações formais, facilita a validação e verificação do sistema também num estágio prematuro do desenvolvimento. Com a abordagem convencional tem-se que esperar pela fase de implementação para a realização de testes exaustivos.

A abordagem operacional parece mais adequada ao desenvolvimento de sistemas em tempo real. As transformações seriam responsáveis pelo mapeamento das estruturas (objetos, abstrações) da especificação operacional em processos ou tarefas que se comunicariam de acordo com os mecanismos existentes no ambiente de implementação. O grande esforço de pesquisas realizadas nessa área consiste na automação das transformações.

Infelizmente o maior atrativo da abordagem operacional se transformou no maior empecilho à sua utilização prática. Ferramentas de suporte, tais como, interpretadores para execução de especificações operacionais, ferramentas para automação de transformações, estão apenas começando a ser desenvolvidas.

A abordagem convencional é amplamente utilizada por métodos de desenvolvimento de sistemas em tempo real juntamente com Análise Estruturada. A decomposição de um sistema em módulos leva a um projeto funcionalmente modular e altamente estruturado.

II.3 - DESCRIÇÃO DA METODOLOGIA UTILIZADA

A metodologia ou o conjunto de métodos utilizado é, na realidade, a união de uma série de conceitos já rotulados como essenciais ao desenvolvimento de um sistema simples, estruturado e eficiente. O principal objetivo era que essa metodologia conduzisse a um sistema com baixo grau de acoplamento entre seus diversos módulos, permitindo assim que o acréscimo de novas funções e a alteração de funções já existentes se fizesse com o menor esforço possível (SILVA, MOTTA, GARROFÉ, COTRIM [13]).

Numa primeira visão da metodologia podemos classificá-la como pertencente à linha convencional que organiza as fases de desenvolvimento numa hierarquia "top down". Cada fase possui um determinado grau de abstração, e os mecanismos implícitos de uma fase são tornados explícitos na fase seguinte.

Nos próximos itens é feita uma descrição detalhada das fases de desenvolvimento que compõem essa metodologia.

II.3.1 - 1ª FASE: ANÁLISE DOS REQUISITOS

Nesta fase é feita a descrição funcional do sistema com o detalhamento dos requisitos do cliente inclusive suas exigências quanto ao desempenho necessário ao produto final. Esta descrição é feita em português corrente e o sistema é apresentado como um todo, sem se configurar ainda qualquer tipo de estruturação para o mesmo.

II.3.2 - 2ª FASE: ESPECIFICAÇÃO FUNCIONAL

Nesta fase o sistema começa a ser estruturado. Em um primeiro plano ele é dividido em Subsistemas Funcionais. Cada subsistema é definido em termos das funções que lhe cabem executar.

As funções dos subsistemas são descritas em forma de Diagramas de Sequência de Ações (DSA). Os DSA são compostos por entidades abstratas denominadas OBJETOS e a comunicação entre esses objetos. Além disso é feita, para cada DSA, a descrição das ações executadas pelos objetos.

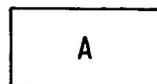
Objetos são entidades que se comunicam entre si através da troca de mensagens e devem compartilhar o mínimo de dados possível. Um objeto é definido pelo conjunto de dados que ele manipula e pelas ações que pode efetuar sobre esses dados. Essa definição é extremamente útil, pois permite a abstração do ambiente distribuído de implementação.

A identificação dos objetos que compõem os subsistemas é uma tarefa empírica inspirada nos tipos de dados e nas classes de ações a serem executadas.

Um objeto dentro do sistema pode ser único, quando existe apenas uma cópia no ambiente distribuído, ou pode ser replicado, no caso de existirem mais cópias.

A seguir são descritas as regras definidas para a construção dos DSA com suas respectivas representações gráficas.

- 1 - Objeto A - um objeto executa operações com um conjunto específico de dados (uma vista do banco de dados).



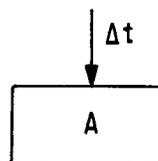
- 2 - Requisição de um serviço, pedido de dados ou comando.

[MENSAGEM]

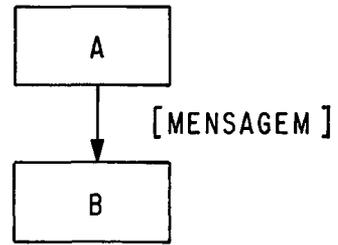
- 3 - Resposta a uma requisição de serviço, a um pedido de dados ou a um comando.

(MENSAGEM)

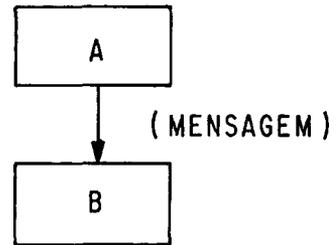
- 4 - Disparo de uma ação periódica interna ao objeto A.



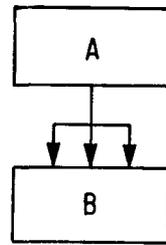
- 5 - Objeto A envia uma requisição de serviço, pedido de dados ou comando ao objeto B.



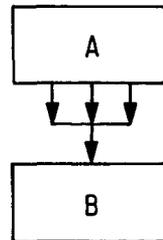
- 6 - Objeto A envia uma resposta a uma requisição de serviço, pedido de dados ou comando ao objeto B.



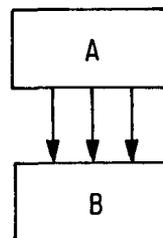
- 7 - Objeto A envia uma mensagem ao objeto B replicado; neste caso existem três cópias do objeto B.



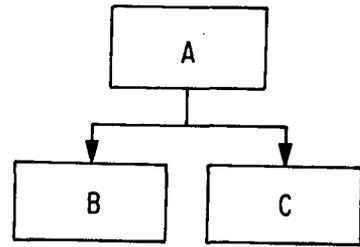
- 8 - Objeto A replicado envia uma mensagem ao objeto B.



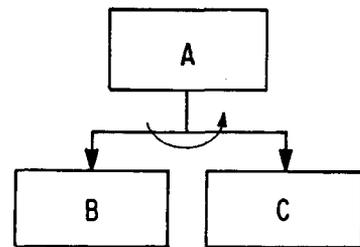
- 9 - Objeto A replicado envia uma mensagem ao objeto B também replicado.



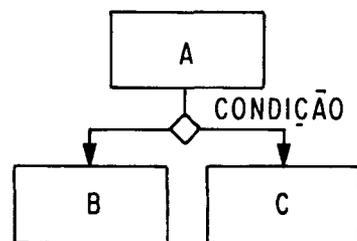
10 - Objeto A envia uma mensagem ao objeto B e uma mensagem ao objeto C; a ordem em que isso é feito não importa, podem até ser procedimentos realizados em paralelo.



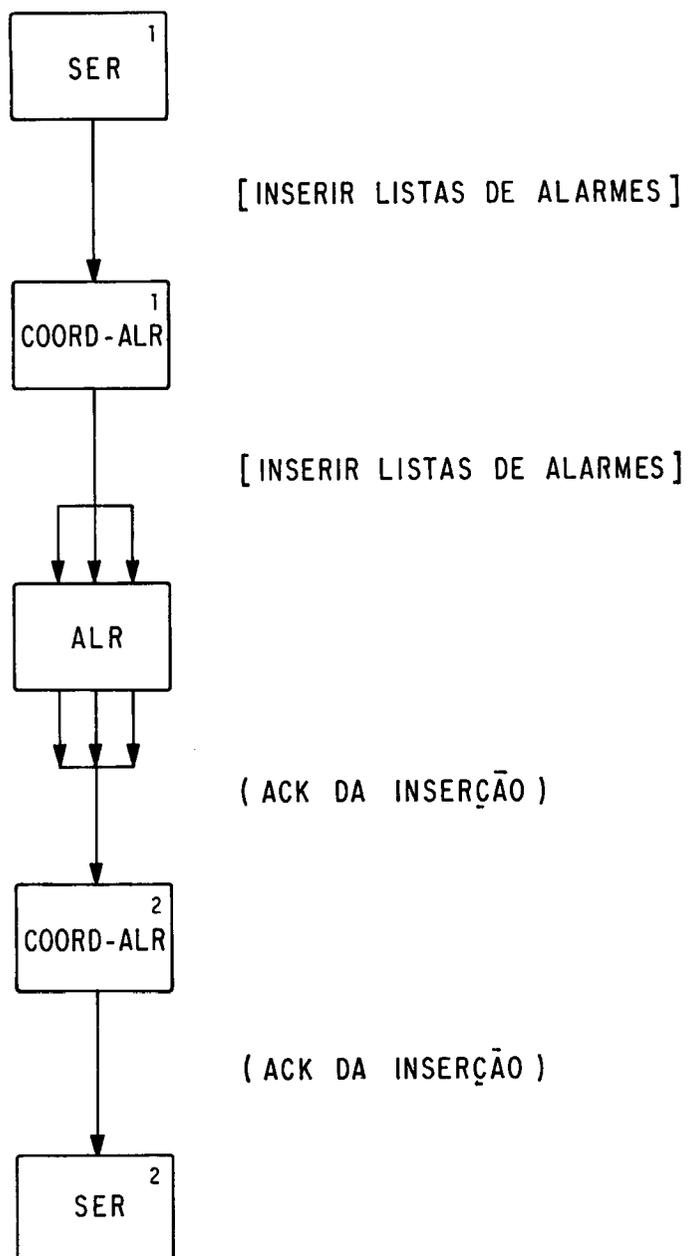
11 - Objeto A envia primeiro uma mensagem ao objeto B e depois envia uma mensagem ao objeto C.



12 - Objeto A envia, no caso da condição ser verdadeira, uma mensagem ao objeto B e, no caso de ser falsa, uma mensagem ao objeto C.



A título de ilustração é apresentado um exemplo de um DSA (DSA (II.1)) no qual um objeto pertencente ao Subsistema de Comunicação com o Sistema Elétrico envia listas de alarmes para serem inseridos, por objetos do Subsistema de Comunicação com a Interface Homem-Máquina, na memória de massa do sistema.



DSA II.1 - DIAGRAMA DE SEQUÊNCIA DE AÇÕES EXEMPLO.

Cada DSA é acompanhado por uma descrição na qual para cada instância de um objeto que aparece no DSA são especificadas as entradas, as saídas e as ações executadas pelo mesmo.

A seguir é apresentada a descrição do DSA do exemplo.

SER 1

Entrada: -

Saída: [Inserir listas de alarmes].

Efeito: O objeto SER, ao detectar a existência de alarmes no Sistema Elétrico, prepara uma lista contendo esses alarmes e envia um comando ao objeto COORD_ALR do Subsistema de Comunicação com a Interface Homem-Máquina para que o mesmo insira esses alarmes nas suas listas. O objeto SER fica aguardando uma confirmação da inserção.

COORD ALR 1

Entrada: [Inserir listas de alarmes].

Saída: [Inserir listas de alarmes].

Efeito: Ao receber um comando de inserção de listas de alarmes, o objeto COORD_ALR repassa, às três cópias do objeto ALR, esse comando, a fim de que esses alarmes sejam inseridos nas listas mantidas pelo sistema em três discos. COORD_ALR fica aguardando as respostas dos três ALR.

ALR

Entrada: [Inserir listas de alarmes].

Saída: (Ack da inserção).

Efeito: Ao receberem um comando de inserção de listas de alarmes, cada ALR insere no seu disco os alarmes e envia ao COORD_ALR um Ack de inserção.

COORD_ALR 2

Entrada: (Ack da inserção).

Saída: (Ack da inserção).

Efeito: Recebe a confirmação da inserção dos alarmes pelos três ALR e repassa essa resposta ao objeto SER.

SER 2

Entrada: (Ack de inserção).

Saída: -

Efeito: Mediante a recepção da confirmação de inserção dos alarmes, libera as listas de alarmes.

II.3.3 - 3ª FASE: PROJETO OU ANÁLISE ORGANICA

Nesta fase é levado em consideração o ambiente de implementação, ou seja, a arquitetura do sistema distribuído e o software básico e de suporte disponível. A implementação em si, linguagem de programação, ainda é irrelevante.

O primeiro passo desta fase constitui-se no mapeamento dos objetos dentro da arquitetura do sistema distribuído, ou seja,

na distribuição dos objetos pelos diversos processadores que compõem o sistema.

Na fase de especificação os subsistemas foram descritos em forma de objetos e das ações ou funções por eles executadas. No segundo passo da fase de projeto cada objeto com suas funções é projetado em termos de processos e de mensagens trocadas entre esses processos.

Não é utilizado nenhum método específico de auxílio na estruturação de cada objeto em processos. Esta transformação é feita pelo projetista empiricamente. A principal consideração feita pelo projetista ao decompor o sistema em processos concorrentes deve ser a natureza assíncrona das funções do sistema. Ele precisa, portanto, identificar as funções dentro de um objeto que estão em posição de concorrência e as que devem ser executadas sequencialmente (GOMAA [6]).

Existe uma série de critérios que podem ajudar o projetista nesta tarefa, por exemplo:

Entrada/Saída de um periférico - se um objeto possui a função de realizar a entrada ou saída de um periférico, esta função deve constituir um processo, uma vez que a velocidade de execução da mesma depende da velocidade do periférico;

Funções Críticas no Tempo - uma função cuja execução é crítica no tempo deve ser executada com uma prioridade maior do que funções não dependentes do tempo e devido a

isso deve constituir um processo por si só, isto é, não deve ser embutida com outras funções em um processo;

Funções Computacionalmente Intensas - uma função cuja execução utiliza intensamente a UCP deve ser executada com baixa prioridade, portanto, deve constituir um processo a parte;

Execução Periódica - uma função que precisa ser executada periodicamente pode ser estruturada como um processo ativado a intervalos regulares de tempo;

Funções de Interface com outros Objetos - uma função cuja execução dependa da requisição de um serviço ou da sinalização por outro objeto do sistema deve constituir um processo.

Uma vez organizado o objeto em processos é necessário definir as estruturas de dados por eles utilizadas e as mensagens trocadas.

O terceiro passo, portanto, consiste em definir o banco de dados do objeto. Esse banco de dados é composto por uma série de tabelas, sendo que cada tabela possui a sua estrutura dividida em campos. Para cada campo é estabelecido o seu tamanho e a sua descrição funcional.

Neste ponto do projeto é hora de considerar as interfaces dos processos de um objeto entre si e as interfaces com os outros objetos. O quarto passo se refere, então, à troca de mensagens e à sincronização entre processos.

Para a definição das mensagens a serem trocadas e para a escolha da forma que as comunicações entre processos devem ser realizadas, o projetista precisa conhecer os recursos que o software básico lhe oferece. Para mensagens entre processos de um mesmo objeto ou entre processos de objetos diferentes, mas que estejam localizados em um mesmo processador, podem ser utilizadas as primitivas do núcleo do sistema operacional (NSO). Para a comunicação entre processos de objetos que se localizam em processadores diferentes o software básico oferece uma ferramenta que permite a abertura de sessões entre objetos (Nível de Sessão do protocolo de comunicação entre os processadores que compõem a rede do sistema distribuído). Esta segunda opção deve também ser utilizada pelo projetista, caso ele queira que as comunicações realizadas por um determinado objeto independam da localização desse objeto dentro da rede. Ou seja, a utilização das primitivas do NSO se restringe às comunicações internas a um processador, enquanto que as primitivas do nível de sessão podem envolver objetos no mesmo ou em diferentes processadores, logo elas permitem a abstração das comunicações entre objetos do ambiente distribuído de implementação. Por outro lado, o projetista deve também considerar que nos casos em que as primitivas do NSO possam ser utilizadas esta opção é a mais sensata, tendo em vista a sobrecarga causada pelas primitivas do nível de sessão.

As primitivas do NSO e do nível de sessão serão descritas com mais detalhes no item (II.4) que se refere à comunicação e sincronização entre processos.

Além desses passos que devem ser seguidos para a concepção do projeto de um objeto, existem outras ferramentas que o projetista pode lançar mão. Uma dessas ferramentas muito utilizadas são as Máquinas de Estados ou Autômatos Finitos. Elas podem auxiliar o projetista na estruturação de um objeto inteiro, ou de uma função ou parte de uma função de um objeto que pode incluir um processo, parte de um processo ou até mesmo vários processos.

Ao término desta fase foi gerada uma documentação na qual para cada objeto constam os seguintes itens:

- Descrição Geral das Funções do Objeto;
- Banco de Dados do Objeto;
- Descrição das Mensagens trocadas entre os processos do próprio objeto e com os processos de outros objetos;
- Diagrama de Processos do Objeto;
- Descrição de cada Processo do Objeto;
- Algoritmos dos Processos;
- Qualquer outra representação de mecanismos utilizados pelo projetista, tais como, Máquinas de Estados.

II.3.4 - 4ª FASE: IMPLEMENTAÇÃO E TESTES

Nesta fase é levado em consideração o ambiente de programação. O projeto da fase anterior é transformado em código executável.

Além disso o código executável é, finalmente, submetido a testes exaustivos. A princípio são realizados testes individuais de um objeto ou até mesmo de parte de um objeto. Para a realização desses testes, geralmente, é necessária a construção de ferramentas de teste tais como simuladores. Os simuladores são programas responsáveis pela criação do ambiente de teste para um determinado objeto, ou seja, eles, basicamente, simulam os objetos com os quais o objeto em teste se comunica. Estes simuladores não executam funções do sistema aplicativo, eles, simplesmente, enviam e recebem mensagens para e do objeto sendo testado. Esses simuladores podem ser configurados como um software executado em outro processador diferente daquele no qual o objeto em teste é executado ou podem ser, simplesmente, módulos executados juntamente com o objeto sendo testado.

Por fim, são realizados testes integrados, ou seja, aos poucos os simuladores vão sendo abandonados e os objetos vão sendo interligados e testados como um todo.

II.4 - COMUNICAÇÃO E SINCRONIZAÇÃO ENTRE PROCESSOS

Existem várias propostas de organização de sistemas distribuídos. A escolha do mecanismo de comunicação entre unidades concorrentes dentro de um sistema depende, essencialmente, dessa organização.

O sistema distribuído em questão é composto por vários processadores chamados OPERADORES, cada um responsável pelo gerenciamento de sua própria memória local. Não existe,

portanto, nenhuma memória central compartilhada pelos operadores.

Mediante o exposto acima, a escolha do mecanismo de troca de mensagens parece adequada como mecanismo de comunicação entre os objetos localizados em operadores distintos, embora se pudesse usar, também, chamada remota de procedimento (NELSON [10]). Além disso a metodologia utilizada no desenvolvimento do sistema é orientada a objetos, que são entidades autônomas que devem compartilhar o mínimo de dados possível, logo mesmo para comunicação entre objetos de um mesmo operador não é indicada a utilização de memória compartilhada como mecanismo de comunicação.

O mecanismo de memória compartilhada necessita, para garantir a integridade dos dados acessados pelos processos, de mecanismos de exclusão mútua, tais como, semáforos, monitores (HOARE [7], BRINCH HANSEN [2], SEGRE e MENDES [12]). Esta característica diminui a autonomia entre os processos. Já o mecanismo de troca de mensagens, embora apresente a desvantagem, no caso de comunicação em um mesmo operador, de necessitar que a mensagem seja copiada da memória do transmissor para a do receptor, permite um maior grau de paralelismo entre os processos. Além disso a utilização do mesmo mecanismo de comunicação e sincronização torna o projeto mais uniforme e estruturado. Em vista disso, o mecanismo de troca de mensagens foi adotado, também, para a comunicação entre processos de um mesmo objeto.

II.4.1 - NÚCLEO DO SISTEMA OPERACIONAL

As primitivas do Núcleo do Sistema Operacional (NSO) permitem a troca de mensagens entre processos localizados em um determinado operador, quer esses processos pertençam a um mesmo objeto ou a objetos diferentes.

O NSO administra processos concorrentes distribuindo o tempo do processador segundo uma estratégia baseada em prioridades. São fornecidas várias primitivas utilizadas pelos processos (APPEL e VALLE [1]).

A troca de mensagens entre processos envolve duas estruturas: a mensagem e o intercâmbio. Um intercâmbio possui sempre a ele associado uma fila. Esta fila pode ser de mensagens à espera de algum processo que as consuma ou de processos consumidores à espera de mensagens enviadas por processos produtores.

A primitiva de envio RQSEND transporta uma mensagem para um determinado intercâmbio. Esta primitiva é não bloqueante, ou seja, o processo produtor, após o envio da mensagem, continua a sua execução caso ele seja o mais prioritário da fila de prontos para executar, independentemente da recepção da mensagem pelo processo consumidor.

A primitiva de espera RQWAIT coloca um processo consumidor à espera de uma mensagem em um determinado intercâmbio. Caso não exista uma mensagem disponível o processo consumidor ficará bloqueado até que uma mensagem chegue a esse intercâmbio ou até que o tempo estipulado para espera se esgote. Se esse tempo

estipulado para espera for igual a zero o processo esperará indefinidamente por uma mensagem. Caso haja uma mensagem disponível o processo entrará na fila de prontos podendo entrar imediatamente em execução se sua prioridade for a maior entre os processos desta fila.

Estas primitivas implementam, naturalmente, comunicação assíncrona entre processos. Este tipo de comunicação oferece como grande vantagem o aumento do paralelismo do sistema. Mas, através da utilização dupla dessas primitivas, podemos implementar comunicação síncrona entre processos, como pode ser visualizado na figura (II.3).

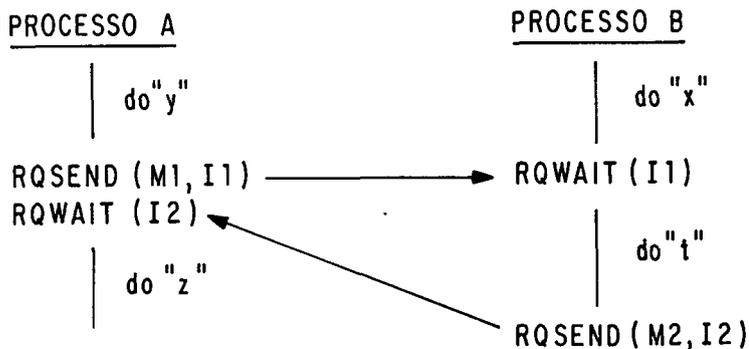


FIGURA II . 3 - COMUNICAÇÃO SÍNCRONA ENTRE PROCESSOS

O processo A, após o envio da mensagem M1, ficou bloqueado pela primitiva RQWAIT no intercâmbio I2 e, portanto, ficou à espera de sincronização com o processo B. Apenas quando o processo B receber a mensagem M1 e, talvez, até mesmo processá-la no trecho "t" a ser executado, é que será enviada a mensagem M2, cuja recepção pelo processo A o desbloqueará.

A primitiva RQACPT permite que um processo aceite uma determinada mensagem que por ventura tenha sido depositada em um intercâmbio. Mas caso o intercâmbio esteja vazio o processo retorna à sua execução normal. Esta primitiva implementa uma espera não bloqueada.

Além da comunicação entre processos as primitivas do NSD permitem também a sincronização entre eles. Um processo A é sincronizado com um processo B quando a sua execução depende da ocorrência de um evento sinalizado pelo processo B. Para que esta sincronização possa ser implementada o NSD oferece a primitiva RQISND. Esta primitiva permite o envio a um intercâmbio de uma sinalização, ou seja, a mensagem enviada a esse intercâmbio não possui conteúdo, ela apenas sinaliza a ocorrência de um evento. Este tipo de intercâmbio é denominado intercâmbio de interrupção e armazena em um determinado instante apenas uma sinalização. Não existe, portanto, uma fila de mensagens associada a ele, pode existir apenas uma fila de processos em espera nele. A primitiva de espera nesse intercâmbio é a mesma RQWAIT já descrita.

O intercâmbio de interrupção em conjunto com as primitivas RQISND e RQWAIT pode ser usado, também, para a implementação de

exclusão mútua. Como só existe uma mensagem disponível apenas um processo receberá esta mensagem com garantia de exclusividade. Este mecanismo pode ser utilizado pelos processos para o gerenciamento de recursos compartilhados. Cabe ressaltar que a fila de espera dos processos no intercâmbio é ordenada por chegada, não por prioridade. Assim, não há risco de espera indefinida ("starvation") de um processo ao qual o recurso é indispensável.

II.4.2 - COMUNICAÇÃO ATRAVÉS DA VIA DE INTERCONEXÃO DOS PROCESSADORES

A comunicação entre processos localizados em processadores distintos é mais facilmente visualizada como objetos se comunicando por meio de uma rede de interconexão (FREITAS e MONTEIRO [5]). Os objetos são referenciados por nomes, não havendo necessidade do conhecimento da distribuição dos mesmos entre os processadores que compõem a rede.

A troca de informações entre objetos é feita através de relações temporárias, denominadas sessões, controladas pelo que chamamos de Nível de Sessão do protocolo de comunicação entre os processadores da rede.

O conceito de intercâmbio, utilizado como estrutura de comunicação entre processos, é estendido para o conceito de portas de comunicação entre os objetos.

Um objeto, com respeito às sessões, pode operar como objeto origem, aquele que inicia a sessão, ou como objeto destino, aquele com o qual objetos origem querem se comunicar. Logo, para objetos origem são configuradas portas de comunicação de saída e para objetos destino portas de comunicação de entrada.

Um objeto destino pode ser único e residir em apenas um operador destino ou possuir réplicas residentes em diversos operadores destino, sendo denominado objeto replicado.

Um objeto destino pode ou não estar disponível para a realização de uma sessão. Caso esteja disponível, diz-se que está ATIVO, caso contrário INATIVO. Com este recurso é possível mascarar temporariamente a presença de certos objetos para o sistema.

Se o objeto origem é residente no mesmo operador que o objeto destino da sessão, esta sessão é denominada MONOLOGO.

Para a realização de uma sessão é necessário que se defina o conjunto de operadores nos quais o objeto destino se encontra residente. A esse conjunto dá-se o nome de CAST, que é composto pelos endereços de rede dos operadores. Cada operador possui um único endereço de rede que é a sua identificação na rede de interconexão.

Se os dados transferidos por uma sessão devem ser recebidos por todas as réplicas do objeto destino, ou seja, pelas réplicas localizadas em todos os operadores do CAST definido para a sessão, diz-se que esta sessão é MULTICAST. Se os dados devem apenas ser recebidos por uma réplica do objeto destino, ou seja,

pela primeira réplica encontrada ativa dentre os operadores do CAST, diz-se que a sessão é UNICAST.

Uma sessão cujo objeto destino é não replicado possui um CAST com apenas um operador destino. Logo, as sessões realizadas com objetos destino não replicados são sempre UNICAST.

As primitivas do Nível de Sessão implementam comunicação síncrona entre os objetos. O processo de um objeto origem que invoca a primitiva para realização de uma sessão fica bloqueado até o término da mesma. O processo de um objeto destino, através de uma primitiva de espera, fica, obviamente, bloqueado aguardando que uma sessão seja estabelecida com ele. São fornecidos, no entanto, tanto para objetos origem como para objetos destino mecanismos de saída por tempo ("time-out").

Um objeto destino pode esperar pelo estabelecimento de uma sessão indefinidamente ou por um determinado intervalo de tempo ("time-out").

Um objeto origem pode escolher a execução de uma sessão com retorno ou com insistência. No caso de uma sessão com retorno, uma vez esgotadas as tentativas de estabelecimento sem sucesso com todos os operadores do CAST, ou se ocorrerem falhas após o estabelecimento, a sessão é terminada sem sucesso e o controle é retornado ao processo do objeto origem. Logo, o objeto origem não fica indefinidamente bloqueado. No caso de uma sessão com insistência, os operadores do CAST são experimentados ciclicamente até que se consiga o estabelecimento com um operador destino, para sessão UNICAST, e pelo menos um operador destino, para sessão MULTICAST. Mesmo se após o estabelecimento

ocorrer alguma falha, volta-se novamente à fase de estabelecimento. Logo, neste caso, para sessões UNICAST o processo do objeto origem fica bloqueado até que a sessão seja executada com sucesso, e para sessões MULTICAST o processo do objeto origem fica bloqueado até que a sessão seja executada com sucesso (com todos os operadores do CAST) ou com sucesso parcial (com algum(ns) operador(es) do CAST). Uma sessão com insistência nunca é terminada com "status" igual a sem sucesso.

Existem dois modos básicos de realizar uma sessão: modo direto e modo normal.

O principal objetivo de uma sessão modo direto é fornecer uma comunicação rápida através da qual um objeto origem envia um comando a um objeto destino e obtém uma resposta a esse comando.

O esquema da troca de mensagens no modo direto está representado na figura (II.4). O objeto origem envia o pedido de estabelecimento juntamente com as informações ao objeto destino e recebe as informações de resposta juntamente com a ordem de término.

Este tipo de sessão deve ser utilizado para comunicações rápidas, uma vez que existe no operador origem um tempo máximo de espera ("time-out") pela resposta do objeto destino. Cabe ressaltar que se a sessão é com retorno, mediante o primeiro "time-out", o controle é retornado ao processo do objeto origem, mas se a sessão é com insistência após cada "time-out" o comando é novamente enviado, indefinidamente até que se obtenha uma resposta.

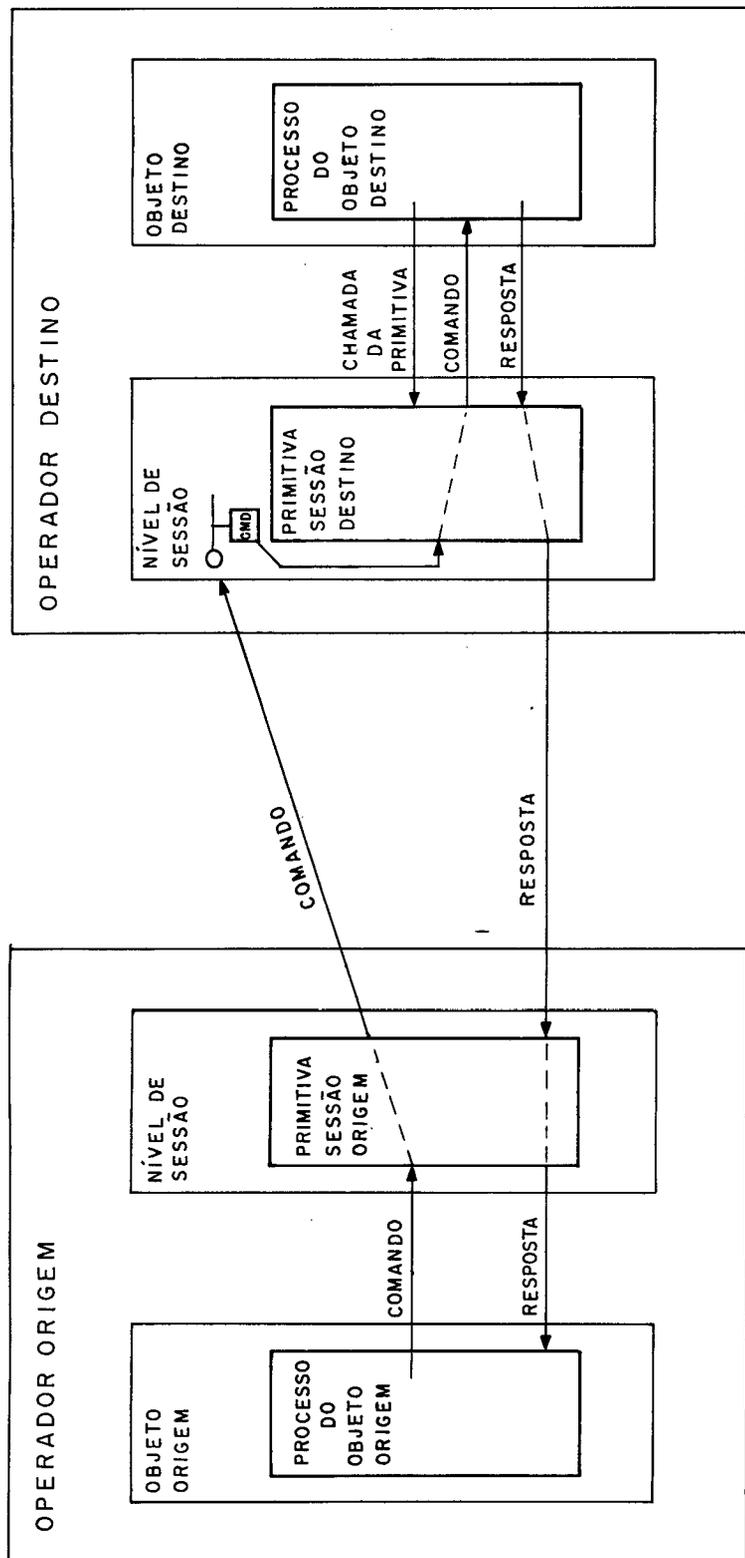


FIGURA II . 4 - SESSÃO DIRETA

O principal objetivo de uma sessão modo normal é a transferência de dados do objeto origem para o objeto destino replicado ou não. Esta transferência de dados é realizada em três etapas distintas como está esquematizado na figura (II.5): estabelecimento, transferência e término.

O objeto origem envia um pedido de estabelecimento de sessão à(s) cópia(s) do objeto destino. O operador destino, mesmo que o objeto destino em questão ainda não tenha requisitado a primitiva de espera para a realização da sessão, envia um "ACK" ao estabelecimento caso a cópia do objeto destino neste operador esteja ativa. Pedidos de estabelecimento para um objeto destino são enfileirados pelo operador destino. O objeto destino atenderá a esses pedidos segundo uma política FIFO.

O objeto destino, na etapa de transferência de dados, pede e recebe segmento a segmento dos dados e os passa um a um ao processo requisitante da primitiva.

Por fim o objeto destino envia uma ordem de término ao objeto origem e recebe do mesmo um "ACK" a essa ordem.

Para que a sessão seja totalmente realizada é preciso que a cada segmento transferido os recursos considerados necessários pelo objeto origem se encontrem disponíveis no objeto destino, caso contrário a sessão é interrompida e é retornado ao usuário um "status" de insucesso devido à falta de recursos (se a sessão for com retorno, porque se for com insistência só retorna mediante um "status" de sucesso total ou parcial).

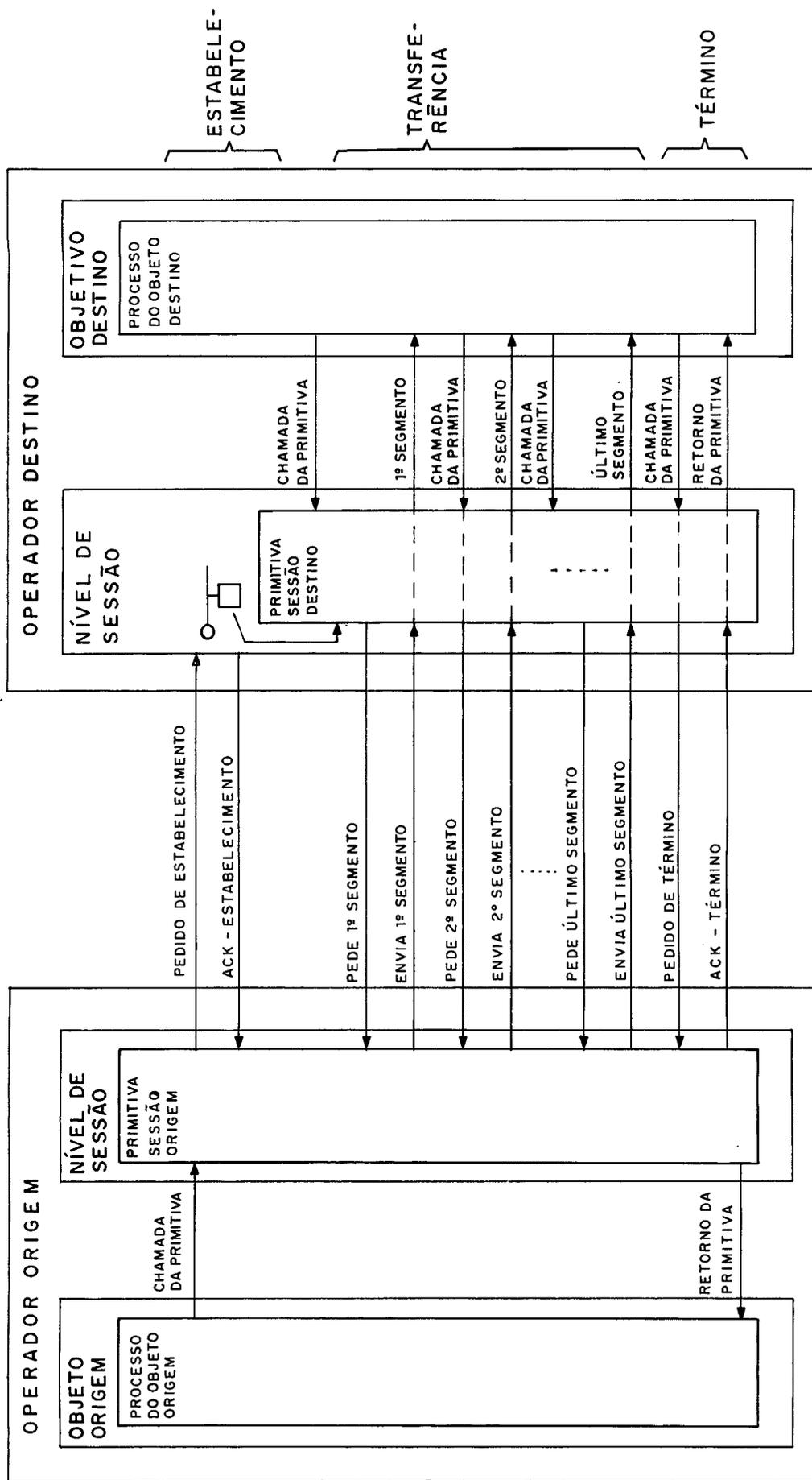


FIGURA II. 5 - SESSÃO NORMAL

CAPÍTULO III

DESCRIÇÃO DOS CENTROS DE OPERAÇÃO REGIONAIS E DO DESPACHO CENTRAL

III.1 - INTRODUÇÃO

Este capítulo engloba as descrições da arquitetura do hardware e da arquitetura funcional do software dos sistemas que constituem o ambiente de desenvolvimento deste trabalho.

Os sistemas acima mencionados fazem parte do Programa de Supervisão e Controle do Sistema Elétrico Nacional. Este programa prevê a ligação dos Despachos Centrais das dez principais empresas do Setor Elétrico ao Centro de Supervisão e Coordenação (CSC) administrado pela Eletrobrás também chamado de Sistema Nacional de Supervisão e Coordenação (SINSC).

O Despacho Central (DC) de cada empresa encontra-se ligado aos Centros de Operação Regionais (COR) da mesma os quais, por sua vez, se ligam às Unidades Terminais Remotas (UTR) responsáveis pela comunicação com o Sistema Elétrico.

O sistema se apresenta, então, organizado de forma hierarquizada como demonstra a figura (III.1). Os sistemas que serão descritos nos itens a seguir encontram-se destacados na figura.

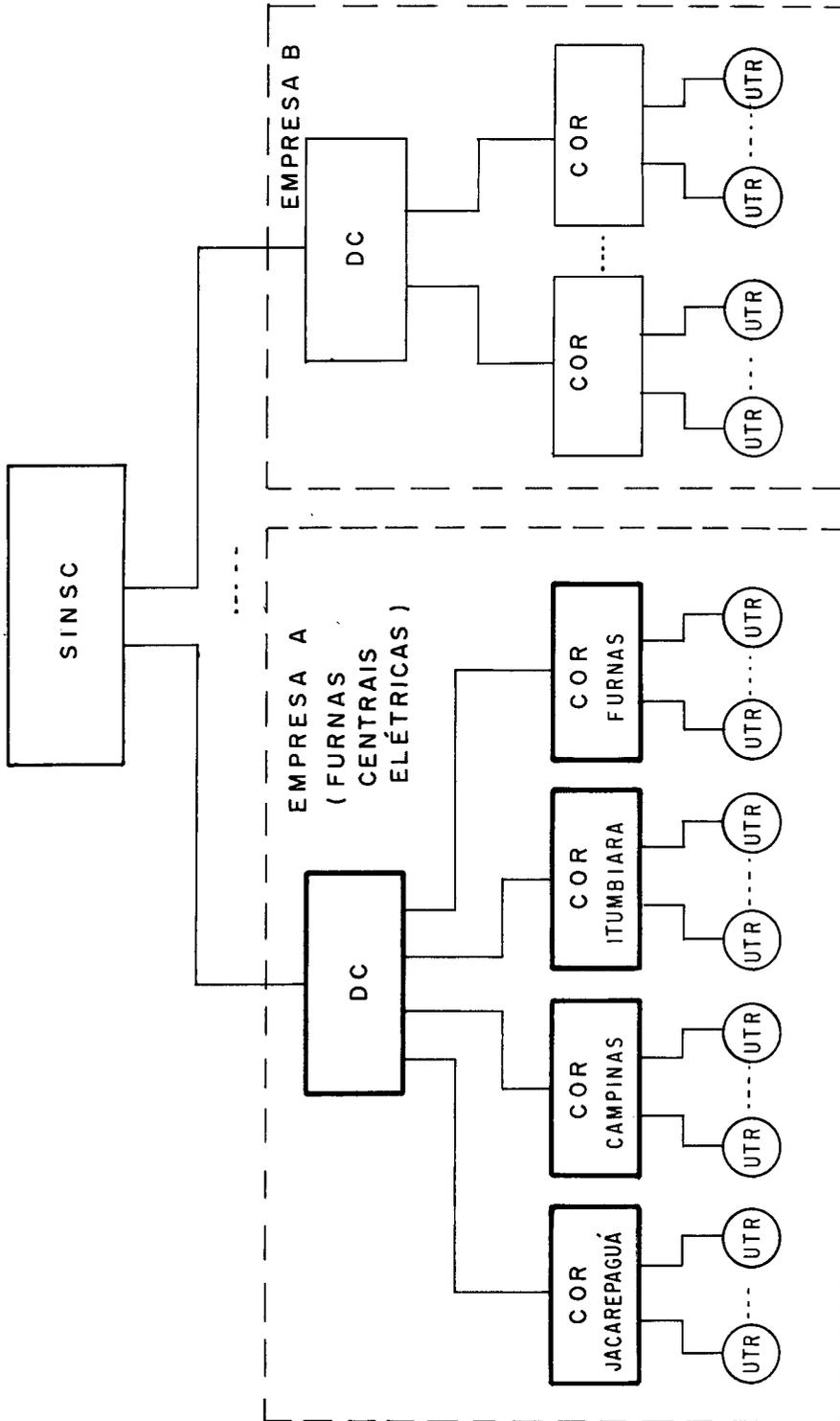


FIGURA III . 1 - HIERARQUIA DOS SISTEMAS DO PROGRAMA DE SUPERVISÃO E CONTROLE DO SISTEMA ELÉTRICO NACIONAL .

III.2 - DESCRIÇÃO DOS CENTROS DE OPERAÇÃO REGIONAIS

O Sistema de Supervisão, Controle e Aquisição de Dados (SCADA) das Usinas e Subestações de Furnas Centrais Elétricas possui quatro Centros de Operação Regionais (COR).

Os sistemas de supervisão que equiparão os COR são compostos por uma rede local de unidades independentes com capacidade de processamento, memória e interface com o meio externo chamadas Operadores. Esses operadores comunicam-se entre si via um barramento serial chamado Via Geral de Interconexão (VGI), ou seja, o Sistema Distribuído é uma rede local baseada na tecnologia ETHERNET. Uma visão geral da arquitetura do sistema é apresentada na figura (III.2).

Cada operador é responsável por determinada função do sistema aplicativo sendo que aqueles responsáveis por funções vitais encontram-se replicados, aumentando, assim, a disponibilidade do sistema. Além disso o sistema, por ser distribuído, oferece: facilidade de expansão, uma vez que novos operadores podem ser adicionados sem causar impacto nos já existentes; degradação controlada, uma vez que a falha de um operador não compromete o restante do sistema e alta confiabilidade.

Funcionalmente o sistema encontra-se dividido em quatro subsistemas a saber: Subsistema de Comunicação com a Interface Homem-Máquina (SCHM), Subsistema de Comunicação com o Sistema Elétrico Regional (SCER), Subsistema de Comunicação com o

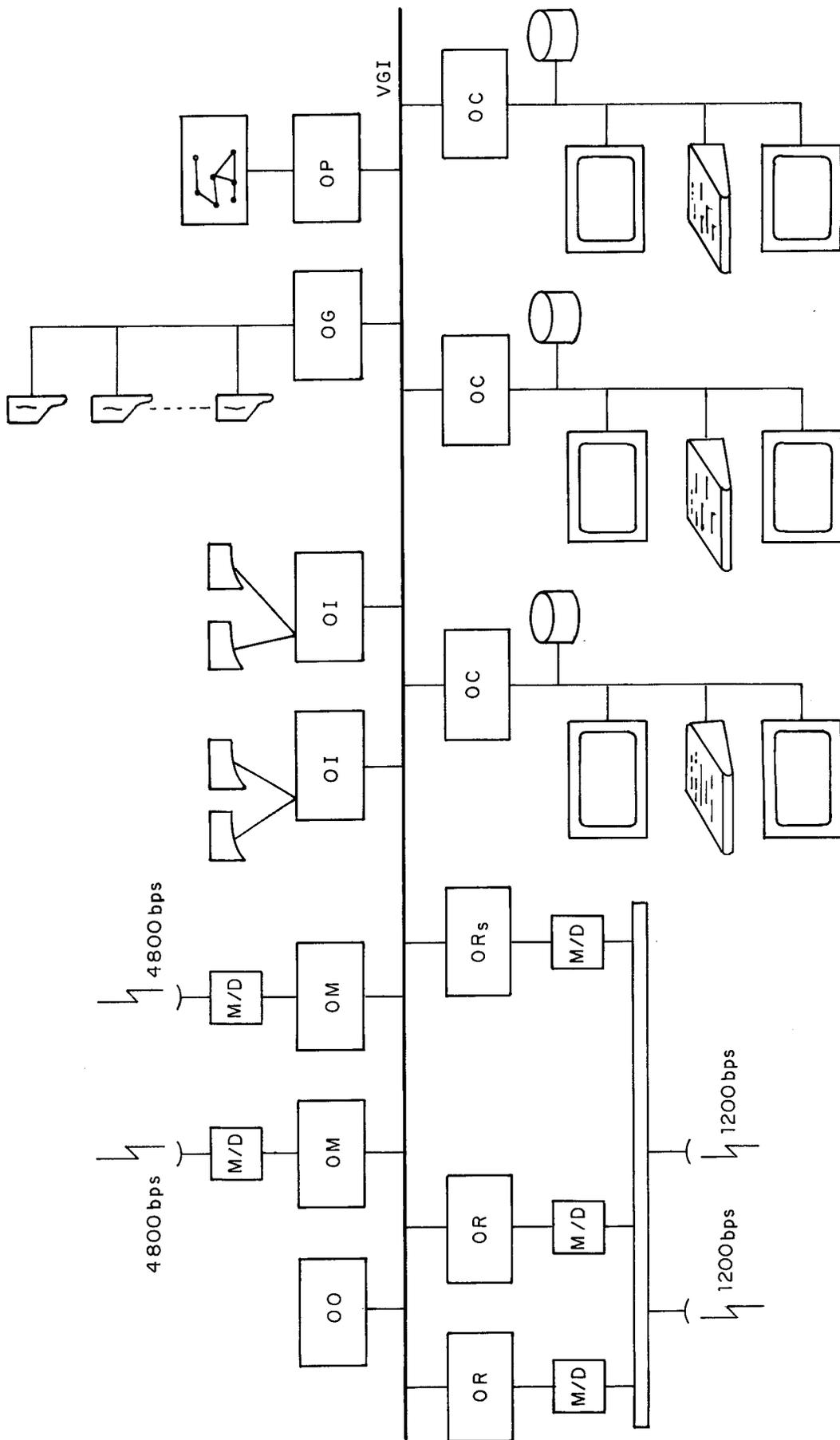


FIGURA III . 2 - ARQUITETURA DO SISTEMA DOS CENTROS DE OPERAÇÃO REGIONAIS .

Sistema Central (SCSC) e Subsistema de Observação do Sistema Computacional (SOSC).

Nos próximos itens serão descritas, separadamente, a configuração de hardware e as funções do sistema aplicativo de cada subsistema.

III.2.1 - SUBSISTEMA DE COMUNICAÇÃO COM A INTERFACE HOMEM-MÁQUINA (SCHM)

O Subsistema de Comunicação com a Interface Homem-Máquina é responsável pela interação com os despachantes do COR. A situação do Sistema Elétrico é controlada pelos despachantes através da observação de vários periféricos que exteriorizam os dados essenciais para a monitoração do sistema.

Este subsistema engloba quatro tipos de operadores: Operador de Console (OC), Operador de Painel Mímico (OP), Operador de Registradores Gráficos (OG) e Operador de Impressão (OI).

Cada Operador de Console (OC) encontra-se ligado a dois vídeos, um teclado e um disco Winchester que constitui a memória de massa do sistema. Existirão três OC com esta configuração em cada COR, ou seja, três postos de atuação e observação para os despachantes do sistema. Em situação normal o processamento necessário à execução das diversas funções é distribuído pelos três operadores, mas em caso de falha em um deles os demais

assumem os processamentos que seriam executados pelo operador com defeito.

Os OC são responsáveis pela execução de seis funções do SCHM: Função Acompanhamento, Função Controle, Função Alarme, Função Consulta e Alteração, Função Mensagem e Função Registro.

A Função Acompanhamento possibilita ao despachante o acompanhamento do Sistema Elétrico através de páginas apresentadas nos vídeos na forma de diagramas ou tabelas. Essas páginas são apresentadas com medidas de tensão, corrente, potência ativa e reativa. Além disso são exibidos, também, os estados de chaves seccionadoras, disjuntores e proteções. Uma página em apresentação é dinamicamente atualizada através da aquisição dos dados da mesma ao Subsistema de Comunicação com o Sistema Elétrico (SCER).

A Função Controle permite que os despachantes comandem remotamente equipamentos do Sistema Elétrico, ou seja, através desta função um despachante pode abrir ou fechar disjuntores, aumentar ou diminuir a tensão de transformadores ou de compensadores síncronos. Além disso o despachante pode realizar controles sobre os equipamentos do próprio Sistema Computacional, tais como, colocar ou retirar uma console de treinamento, colocar ou retirar de teste impressoras, registradores gráficos ou painel mímico e colocar ou retirar de serviço um canal de comunicação com uma unidade terminal remota (UTR).

A Função Alarme é responsável pela recepção de listas do SCER e pela apresentação desses alarmes em listas de prioridades

ao despachante. Esta função é responsável, também, pela atualização do macro-alarme de cada estação do Sistema Elétrico. O macro-alarme, exteriorizado na sigla da estação, indica, através da cor, a situação mais grave configurada na estação em um determinado instante. Até que o despachante fique ciente dessa situação o macro-alarme é exteriorizado através de um pisca-pisca ("blink"). Através desta função o despachante pode realizar o reconhecimento do macro-alarme, das listas de alarmes do SCHM e dos pontos em alarme de uma estação do SCER e ainda pode pedir a retirada dos alarmes reconhecidos das listas do SCHM. Além disso esta função realiza os mesmos procedimentos para os alarmes e macro-alarme do Sistema Computacional, ou seja, para os alarmes e macro-alarme enviados pelo Subsistema de Observação do Sistema Computacional (SOSC).

A Função Consulta e Alteração permite ao despachante obter informações sobre a configuração do Banco de Dados do COR e alterar os parâmetros, dados estáticos, do mesmo. Através desta função o despachante pode modificar os horários de emissão de relatórios, as grandezas e limites dos registradores gráficos, os limites de medidas analógicas; pode inibir a varredura de um ponto; pode inibir a geração de alarmes para um ponto; etc...

A Função Mensagem permite a troca de mensagens entre os despachantes do COR e os despachantes do DC. São três os tipos de mensagens que podem ser trocadas: mensagens de "crt" que constituem uma linha de oitenta (80) caracteres e são usadas para troca de informações rápidas entre os despachantes; mensagens estruturadas que portam dados de formato definido, por

exemplo, para composição de relatórios; e mensagens de "display" que são cópias de telas de vídeo.

A Função Registro é responsável pela montagem e envio para impressão de relatórios em horas programadas ou sob requisição do despachante; pela emissão de "hardcopy", ou seja, envio de cópias de telas de vídeo para impressão sob requisição do despachante; e pelo envio para impressão de alarmes e de ações dos despachantes, tais como, alterações de Banco de Dados, reconhecimento de alarmes, comandos de controle, etc...

Além dessas funções os Operadores de Console se encarregam de manter a consistência do Banco de Dados mantido nos três discos do sistema e da carga dos programas e dos bancos de dados dos demais operadores.

Existe um Operador de Painel Mímico (OP) em cada COR. Este operador é responsável pela atualização periódica dos estados das linhas, trechos de circuitos e macro-alarmes das estações representados no painel mímico. Além disso este operador pode ser comandado pelo SOSC para colocar ou retirar de teste o painel mímico.

Cada COR possui um Operador de Registradores Gráficos (OG). Este operador é responsável pela exteriorização de grandezas analógicas pré-definidas ou selecionadas através da função alteração dos OC. Um OG pode controlar até trinta e duas (32) penas de registradores gráficos. Além disso este operador pode ser comandado pelo SOSC para colocar ou retirar de teste uma determinada pena de um registrador gráfico.

Em cada COR existem dois Operadores de Impressão (OI). Cada OI é responsável pelo gerenciamento de duas impressoras. Um deles, normalmente, estará dedicado à impressão de alarmes e de ações de controle e alteração dos despachantes, sendo que uma impressora é utilizada para impressão desses dados do Sistema Elétrico e a outra do Sistema Computacional. Em caso de falha em uma das impressoras a outra assume a impressão de ambos os sistemas. O outro OI estará dedicado à impressão de relatórios e cópias de telas, sendo que uma impressora é utilizada para o Sistema Elétrico e a outra para o Sistema Computacional. A falha de uma impressora é tratada da mesma forma que é feito para o outro OI. Caso um dos OI fique fora de serviço deixa de existir a particularização para alarmes/ações do despachante e relatórios/cópias de telas. Com isso todas as impressões serão chaveadas para o OI em serviço, continuando a separação entre as duas impressoras dos dados do Sistema Elétrico e do Sistema Computacional.

Além disso um OI pode ser comandado pelo SOSC para colocar ou retirar de teste uma impressora. Se durante a impressão de um relatório ou tela for solicitada a colocação em teste da impressora em questão, o OI informa ao SOSC que a mesma não está disponível para teste. Se algum dado for endereçado a uma impressora em teste essa impressão é chaveada para a outra impressora.

III.2.2 - SUBSISTEMA DE COMUNICAÇÃO COM O SISTEMA ELÉTRICO REGIONAL (SCER)

O Subsistema de Comunicação com o Sistema Elétrico Regional é responsável pela interação do Sistema de Supervisão e Controle do COR com o Sistema Elétrico Regional através das Unidades Terminais Remotas.

Este subsistema engloba dois tipos de operadores: Operador de Remotas (OR) e Operador "Spare" (OS).

Cada COR possui dezesseis Operadores de Remotas e quatro Operadores "Spare". Cada OR é capaz de monitorar duas estações do Sistema Elétrico (usina ou sub-estação) sendo cada estação com, no máximo, duas remotas. Logo, um OR pode adquirir dados de até quatro remotas.

Para cada grupo de quatro OR existe um OS de reserva. O OS monitora, permanentemente, o estado de operação dos quatro OR, e no caso de ocorrência de uma falha em um dos OR do grupo, o OS assume os procedimentos do operador com defeito, ou seja, ele pede carga a um OC do programa e do banco de dados do operador em falha e assume o seu lugar na rede.

Funcionalmente o SCER é responsável pela aquisição de dados das unidades terminais remotas (UTR), pela manutenção do banco de dados das estações, pela distribuição desses dados aos demais subsistemas, pelo encaminhamento de comandos de controle do SCHM e do SCSC ao Sistema Elétrico e pela execução do controle que permite colocar ou retirar de serviço um canal de comunicação com uma UTR.

O SCER pode ser dividido em quatro funções distintas: Função de Varredura e Tratamento, Função de Envio de Dados, Função Controle e Função de Atendimento de Requisições.

A Função Varredura e Tratamento é responsável pela manutenção do banco de dados das estações atualizado. Existem quatro tipos de dados no banco de dados de uma estação: Dados Digitais Sem Memória, Dados Digitais Com Memória, Dados Analógicos e Dados Totalizados.

Os dados digitais sem memória e os analógicos são adquiridos e tratados continuamente, uma vez que as remotas não os monitoram permanentemente. Caso seja detectada alguma mudança de estado dos dados digitais sem memória, ou alguma variação significativa (filtro Graham Creep) ou ultrapassagem de limites nas medidas analógicas, essas informações são armazenadas no banco de dados para posterior utilização pela Função Envio de Dados.

Os dados digitais com memória são varridos mediante indicação da remota da existência de mudança de estado e também de dez em dez minutos numa varredura de integridade. Os dados tratados são armazenados para serem utilizados pela Função Envio de Dados. Além disso, caso a remota gere lista de seqüência de eventos, a mesma é adquirida e armazenada para posterior envio.

Os dados totalizados são varridos a cada dez minutos e acumulados durante o período de uma hora, quando, então, são congelados para utilização pela Função Envio de Dados e uma nova contagem é iniciada. O tratamento desses dados é análogo ao das medidas analógicas.

A Função Envio de Dados é responsável por manter o SCHM e o SCSC (conseqüentemente o DC) atualizados quanto às mudanças ocorridas nos bancos de dados das estações.

Para o SCHM são enviadas listas de alarmes de todos os tipos de dados do Sistema Elétrico e alarmes de entrada e saída do conjunto canal de comunicação/remota e listas de seqüência de eventos.

Para o SCSC são enviadas listas contendo todas as informações necessárias para que o DC mantenha o seu banco de dados atualizado. Essas listas são caracterizadas por dois tipos de informação, aquelas geradas pelo próprio Sistema Elétrico e aquelas geradas por ações dos despachantes do COR que provocam mudanças de atributos nos pontos monitorados pelo sistema. As listas do primeiro grupo são: listas de alarmes de todos os tipos de dados, listas de variações de medidas analógicas e totalizadas, listas de seqüências de eventos, listas de alarmes do canal de comunicações/remota, lista de integridade enviada de hora em hora de todos os tipos de dados, sendo que para os dados totalizados esta lista também informa o valor de um período completo de acumulação dos contadores totalizados, e listas de iniciação seqüencial contendo todos os dados do Sistema Elétrico. As listas do segundo grupo informam mudança de atributo, tais como, varredura de uma medida inibida e valor inserido pelo despachante do COR, varredura desinibida e valor atualizado pela UTR (iniciação não seqüencial).

A Função Controle é responsável pela recepção de comandos de controle para equipamentos do Sistema Elétrico do SCHM e do

SCSC e envio desses comandos às UTR para sua execução. Além disso a função controle pode ser comandada pelo SCHM para colocar ou retirar de serviço um canal de comunicação com uma remota.

A Função de Atendimento de Requisições é responsável pelas seguintes subfunções: Pedido de Dados, Alteração, Pedido de Alarmes Não Reconhecidos, Pedido de Reconhecimento de Alarmes e Pedido de Iniciação do DC.

A subfunção Pedido de Dados consiste no envio dos valores atualizados de pontos de qualquer tipo de dado. Esta subfunção é utilizada pela Função Acompanhamento do OC quando a mesma deseja obter dados dinâmicos para apresentação de alguma tela, pelo OG e OP para manterem as exteriorizações dos seus periféricos atualizadas e pela Função Registro para confecção de relatórios.

A subfunção Alteração recebe os comandos provenientes da Função Consulta/Alteração do OC e executa as mudanças de atributos em pontos do Sistema Elétrico requisitadas.

A subfunção Pedido de Alarmes Não Reconhecidos é utilizada pela Função Consulta/Alteração do OC quando o despachante requisita uma consulta que lhe mostre quais os pontos de um determinado tipo que se encontram em alarme e não foram ainda reconhecidos por algum despachante.

A subfunção Pedido de Reconhecimento de Alarmes é utilizada pela Função Alarme do OC quando o despachante requisita o reconhecimento de alarmes de uma determinada estação.

A subfunção Pedido de Iniciação do DC é encaminhada pelo SCSC mediante um pedido do DC para que o COR envie todos os dados do Sistema Elétrico, a fim de que o DC possa iniciar o seu banco de dados. Este pedido desencadeia o envio pela Função Envio de Dados de listas de iniciação sequencial de todos os dados do Sistema Elétrico.

III.2.3 - SUBSISTEMA DE COMUNICAÇÃO COM O SISTEMA CENTRAL (SCSC)

O Subsistema de Comunicação com o Sistema Central é responsável pela recepção de dados dos demais subsistemas (SCHM e SCER) e envio desses dados ao DC e vice-versa.

Este subsistema engloba apenas um tipo de operador, o Operador de Estação Mestre (OM).

Cada COR possui dois Operadores de Estação Mestre. Cada OM encontra-se ligado ao DC através de um canal de comunicação como demonstra a figura (III.3). Em situação normal ambos os OM transferem dados de/para o DC dividindo, assim, o fluxo de informações pelos dois canais, mas em caso de falha em um deles o outro assume as funções do operador com defeito. Além disso uma falha em um OM ou em um modem de um dos conjuntos OM/modem/canal de comunicação juntamente com uma falha no canal de comunicação do outro conjunto, que causaria interrupção na comunicação com o DC, pode ser recuperada (um conjunto apenas) através de chaveamento manual dos canais de comunicação.

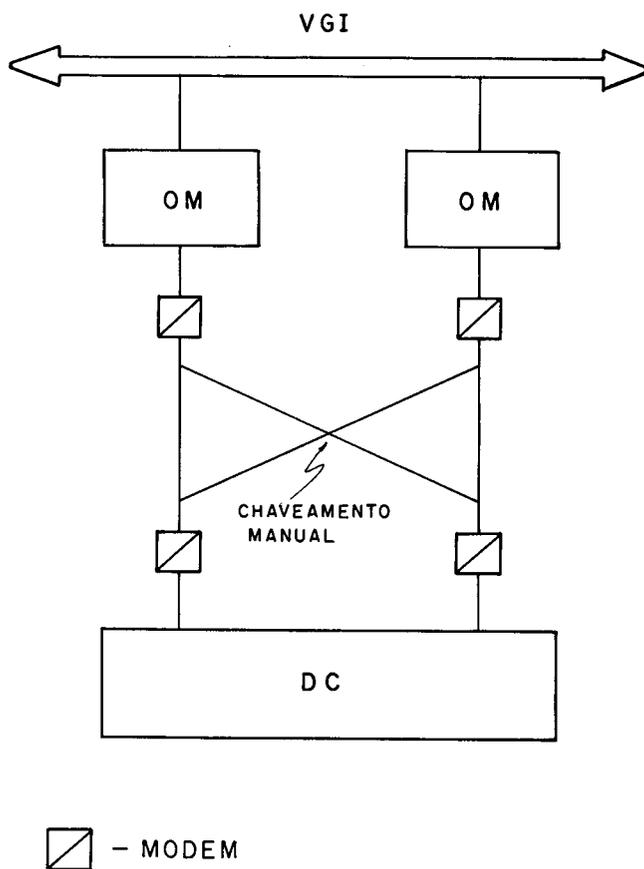


FIGURA III. 3 - LIGAÇÃO DC \Leftrightarrow COR

Funcionalmente o SCSC é responsável pela comunicação entre o COR e o DC. Três tipos de informação são trocados entre os dois sistemas:

- Mensagens de Dados em Tempo Real,
- Mensagens Operacionais,
- Mensagens de Sistema.

Os Dados em Tempo Real se dividem em dois grupos: Dados de Controle Elétrico e Dados Adquiridos. As mensagens de controle elétrico se referem a comandos de controle gerados pela interface homem-máquina do DC. As mensagens de dados adquiridos constituem-se nos dados do Sistema Elétrico enviados ao SCSC pelo SCER, ou seja, são informações referentes aos dados digitais, analógicos, totalizados e seqüência de eventos.

As Mensagens Operacionais também se dividem em dois grupos: Mensagens Formatadas e Mensagens Não Formatadas. As mensagens formatadas constituem-se nas mensagens de "crt" e nas mensagens estruturadas (relatórios) enviadas de e para o SCHM. As mensagens não formatadas são telas ou "hardcopies" que constituem as mensagens de "display" também enviadas de e para o SCHM.

As Mensagens de Sistema dizem respeito aos Sistemas Computacionais e se caracterizam como mensagens de iniciação.

III.2.4 - SUBSISTEMA DE OBSERVAÇÃO DO SISTEMA COMPUTACIONAL (SO SC)

O Subsistema de Observação do Sistema Computacional é responsável pela monitoração dos equipamentos do próprio Sistema Computacional.

Este subsistema engloba apenas um tipo de operador, o Operador de Observação (OO). Cada COR possui apenas um OO.

Funcionalmente o SDSC é responsável pela monitoração de todos os operadores da rede e de todos os periféricos a eles acoplados.

Em linhas gerais o SDSC é responsável por duas funções: Função Alarme para o Sistema Computacional e Função Controle para o Sistema Computacional.

A Função Alarme é responsável pela aquisição periódica do estado de todos os operadores da rede e de todos os periféricos do sistema. Ao detectar alguma mudança de estado, ela envia listas de alarmes com essas mudanças ao SCHM.

A Função Controle é responsável pela recepção do SCHM de comandos de controle para impressoras, painel mímico e penas de registradores gráficos e encaminhamento desses comandos ao operador em questão.

III.3 - DESCRIÇÃO DO DESPACHO CENTRAL

O Sistema de Supervisão e Controle que equipará o Despacho Central de FURNAS constitui-se basicamente numa rede de três computadores VAX 11/780 ligados entre si formando um sistema "cluster".

A configuração VAX-cluster é um arranjo altamente integrado de sistemas VAX/VMS que se comunicam via hardware do barramento CI ("Computer Interconnect"), usando o software de comunicação DECNET, ou seja, DECNET-CI.

As duas primeiras das três UCP do sistema (VAX-A e VAX-B) formam um sistema homogêneo de tempo real, executando as funções de supervisão e controle do sistema elétrico (Sistema "On-line"), e a terceira UCP (VAX-C) opera em um ambiente de tempo compartilhado, executando as funções de desenvolvimento e treinamento (Sistema "Off-line").

A configuração "cluster" utilizada está representada na figura (III.4), assim como a arquitetura de todo o sistema.

As UCP VAX 11/780 operam interligadas por meio da interface CI780 ("Computer Interconnect"). Cada UCP é considerada como um nó ativo do agrupamento ("cluster"). O CI780 é uma interface inteligente que conecta uma UCP no barramento CI.

O SC ("Star Coupler") é o ponto de conexão comum para todos os nós conectados no agrupamento. O SC conecta os CI de todos os nós num arranjo em estrela que suporta conexão ou desconexão física de qualquer nó durante a operação normal do agrupamento sem afetá-lo.

Através do SC as UCP têm acesso aos três HSC50 ("Hierarchical Storage Controller") do sistema. O HSC50 é um controlador inteligente de disco. Cada disco é acessível por dois dos três HSC50 do sistema; diz-se que os discos são "dual pathed".

O sistema comporta, ainda, três conjuntos de fita magnética cada um acessível por duas UCP. Cada UCP encontra-se ligada via barramento SBI a um controlador MASSBUS (MBA) que se liga a dois conjuntos de fita magnética (MF).

Cada VAX encontra-se ligado a uma via (UNIBUS) respectivamente vias A, B e C. Cada via possui um grau de redundância, ou seja, é composta por uma via primária e uma via reserva. Através dessa via é que os VAX se ligam aos demais equipamentos que compõem o sistema os quais serão descritos a seguir, dentro do escopo do subsistema apropriado. Alguns desses equipamentos se ligam à via PRIMÁRIA/RESERVA diretamente, como é o caso dos geradores de vídeo. Os demais se ligam através de uma via comutada ("Switched Unibus"). Esta via é assim chamada, pois através de duas chaves de comutação DT07 ela se conecta à via do VAX que esteja operando como primária (vias A, B ou C).

Funcionalmente o Sistema "On-line" do DC encontra-se dividido em quatro (4) subsistemas a saber: Subsistema de Interface Homem-Máquina, Subsistema de Entradas e Saídas Locais, Subsistema de Comunicação de Dados e Subsistema de Recuperação.

O Sistema "Off-line" possui uma configuração semelhante, embora reduzida, ao Sistema "On-line". A arquitetura de hardware deste sistema, bem como sua arquitetura de software, não serão descritas por fugirem ao escopo a que este trabalho se propõe.

III.3.1 - SUBSISTEMA DE INTERFACE HOMEM-MÁQUINA (IHM)

O Subsistema de Interface Homem-Máquina proverá recursos para apresentação de informações aos despachantes acerca do estado corrente e da segurança da operação do Sistema Elétrico,

bem como para o intercâmbio de mensagens com os COR e o CSC da Eletrobrás.

A IHM do sistema "on-line" é dotada de três consoles de despacho. Os vídeos das três consoles são controlados por dois geradores de vídeo DG 5215A. Cada gerador está conectado às três UCP através da interface DR11-W. Cada canal do gerador está conectado a um dos dois terminais de vídeo a cores 8830 de cada console de despacho. O teclado 5115A e o dispositivo de movimentação do cursor 5576 ("trackball") de cada console podem ser chaveados entre qualquer um dos dois canais em que estão conectados os seus vídeos.

O subsistema de IHM conta ainda com uma copiadora de vídeo LXY22, duas impressoras de linha e duas impressoras seriais EI-9051.

Todos estes equipamentos de impressão encontram-se conectados a duas chaves 6503 que chaveiam a uma das duas interfaces de comunicação DZ11 ligadas à via comutada.

O subsistema de IHM é responsável pelas seguintes funções do sistema aplicativo:

- Implementação do Diálogo Homem-Máquina, ou seja, atendimento aos pedidos dos despachantes de visualização de telas, de cópias de telas, etc e tratamento das ações dos despachantes;
- Tratamento de alarmes, apresentação e reconhecimento dos mesmos pelos despachantes;

- Tratamento e impressão de seqüências de eventos;
- Geração de relatórios nas impressoras de linha;
- Impressão de eventos do sistema, tais como as ações dos despachantes nas impressoras seriais;
- Troca de mensagens entre os despachantes do DC e os despachantes dos COR e do CSC através do Subsistema de Comunicação de Dados;
- Tratamento dos comandos de controle supervisiório comandados pelos despachantes e encaminhamento desses comandos e recepção dos resultados do Subsistema de Comunicação de Dados;
- Gerenciamento de um dispositivo de alarme sonoro integrado à operação do painel mímico para as funções de alarme sobre eventos no Sistema Elétrico; de registradores gráficos; de "displays" digitais e de painel mímico através do interfaceamento com o Subsistema de Entradas e Saídas Locais.

III.3.2 - SUBSISTEMA DE ENTRADAS E SAÍDAS LOCAIS (ESL)

O Subsistema de Entradas e Saídas Locais usa um sistema baseado no processador PDP-11/24 que utiliza dois discos RL02 para armazenamento de dados e programas.

A conexão do PDP-11/24 com os VAX-780 é feita através de dois controladores de comunicação síncronos DMR11 operando com

software DECNET para comunicação ponto a ponto. Cada um dos dois DMR11 do PDF-11/24 implementa uma ligação independente com os correspondentes DMR11 conectados à via comutada.

Há ainda um controlador de entrada e saída IP11 ligado à via do PDP responsável pelo interfaceamento com equipamentos de entradas ou saídas digitais ou analógicas, tais como, Central Horária, Painel de Configuração dos Equipamentos do Sistema Computacional, Painel Mímico, Registradores Gráficos, Alarme Sonoro, Mostradores Digitais, Equipamentos de Telemedicação Analógica.

O ESL é responsável pelas seguintes funções do sistema aplicativo:

- Controle Automático de Geração (CAG) através da leitura de valores de telemedicação fornecidos pelo sistema analógico de FURNAS e da emissão de pulsos digitais para o sistema de transmissão de sinais de FURNAS;
- Leitura periódica da hora do dia fornecida pela Central Horária e encaminhamento dessa hora para os demais subsistemas;
- Ativação da sinalização luminosa do painel de configuração a partir de pedido do Subsistema de Recuperação;
- Ativação da sinalização luminosa do painel mímico e do alarme sonoro; e atualização dos mostradores digitais e dos registradores gráficos a partir de pedido do Subsistema de Interface Homem-Máquina;

- Detecção de eventos, tais como, falha do canal conversor de telemedicação, comutação de periféricos, e encaminhamento desses eventos aos subsistemas apropriados.

III.3.3 - SUBSISTEMA DE COMUNICAÇÃO DE DADOS (COD)

O Subsistema de Comunicação de Dados é composto por controladores de comunicação KMS11-BD/BE da DEC num total de seis para comunicação com os Centros de Operação Regionais e para a comunicação com o Centro de Supervisão e Coordenação (CSC).

A comunicação com um COR é feita ligando-se cada um dos dois canais de comunicação a uma das entradas de uma chave de comutação GS03K-SB. Cada entrada desta chave pode ser comutada entre um dos dois KMS11 a ela ligados, sendo que um deles encontra-se conectado à via primária da via comutada e o outro à via reserva.

A comunicação com o CSC é feita de forma semelhante através de três canais de comunicação.

A configuração de hardware deste subsistema encontra-se representada em detalhe na figura (III.5).

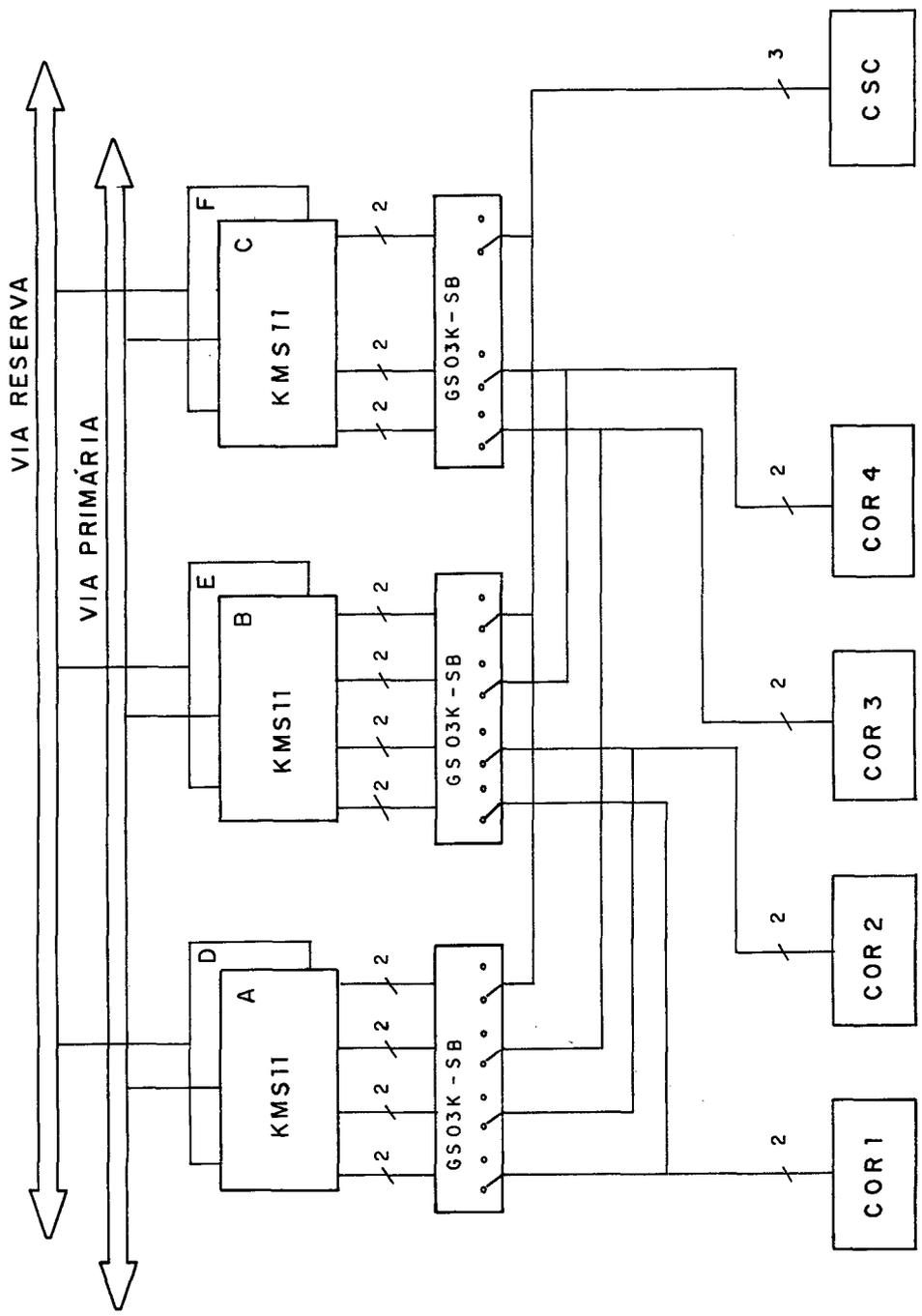


FIGURA III. 5 - ARQUITETURA DO SUBSISTEMA DE COMUNICAÇÃO DE DADOS DO DC.

O COD é responsável pelas seguintes funções do sistema aplicativo:

- Recepção de dados de tempo real (dados analógicos, digitais, seqüência de eventos e totalizados) dos COR e encaminhamento desses dados aos subsistemas apropriados;
- Troca de mensagens entre o DC e os COR com encaminhamento de mensagens de e para o subsistema de IHM;
- Envio de comandos de controle aos COR a pedido do IHM, e recepção e encaminhamento dos resultados ao IHM;
- Envio de medições analógicas e digitais ao CSC devido a variações ou ao ciclo de integridade;
- Troca de mensagens entre o DC e o CSC com encaminhamento de mensagens de e para o subsistema de IHM;
- Acionamento do programa aplicativo de Análise de Redes quando ocorre alguma variação de interesse do mesmo;
- Sincronismo a partir de dados enviados pelo CSC da hora do dia mantida pelo DC.

III.3.4 - SUBSISTEMA DE RECUPERAÇÃO (REC)

O Subsistema de Recuperação é responsável por detectar falhas de hardware e de software que interferem no funcionamento normal do sistema e, após identificá-las, executar as funções de

recuperação necessárias para que este possa continuar a sua operação normal sem degradação de desempenho.

Para que se atinja a disponibilidade especificada está sendo empregada uma configuração de processadores e de outros equipamentos redundantes no sistema computacional.

A redundância de processadores permite que um deles assuma as funções executadas pelo outro automaticamente quando ocorre alguma falha.

Em condições normais um dos processadores VAX-A ou VAX-B do sistema "on-line" é chamado de primário e tem o controle das funções em tempo real do Sistema de Supervisão e Controle do DC. O outro é chamado de processador secundário e encontra-se pronto para assumir o sistema ("hot stand by") em caso de falha ("failover").

Caso ocorra a situação de falha simultânea dos processadores VAX-A e VAX-B está prevista a entrada em operação do processador VAX-C do sistema "off-line".

O REC implementa as seguintes funções:

- Determinação do estado operativo dos computadores, detecção de falhas em equipamentos do sistema "on-line" e comunicação dessas informações aos subsistemas apropriados;
- Determinação da necessidade de execução de "failover" visando a recuperação de uma falha crítica;

- Manutenção de informações atualizadas no banco de dados sobre o funcionamento dos equipamentos do sistema computacional para anúncio nos sistemas de ESL e IHM;
- Execução de comandos de comutação de controladores, canais de E/S e dispositivos de chaveamento da configuração do sistema computacional, a partir de pedidos oriundos dos subsistemas de ESL e IHM;
- Controle das operações de cópia dos dados do banco de dados ou arquivos genéricos do computador primário para o secundário;
- Partida e parada dos processadores do sistema computacional do DC juntamente com a partida e a parada do sistema operacional VAX/VMS e de todo o software usado pelo Sistema de Supervisão e Controle;
- Reiniciação automática e manual ("restart") dos processadores do sistema computacional, após a ocorrência de falhas não críticas de hardware e de software.

CAPÍTULO IV

ESTRUTURAÇÃO DOS NÍVEIS DO PROTOCOLO DE COMUNICAÇÃO

COR <=> DC

IV.1 - INTRODUÇÃO

Este capítulo e os demais subsequentes tratam, efetivamente, da implementação da comunicação entre os Centros de Operação Regionais (COR) e o Despacho Central (DC) do Sistema de Supervisão e Controle de Furnas, ou seja, da implementação do Subsistema de Comunicação com o Sistema Central (SCSC) dos COR.

Visando a uniformização das interconexões entre os diversos sistemas que compõem a rede de computadores (MENASCÉ e SCHWABE [9], TANENBAUM [14] e TAROUÇO [15]) do Programa de Supervisão e Controle do Sistema Elétrico Nacional, optou-se pela utilização do modelo elaborado pela ISO para interconexão de sistemas abertos, denominado Modelo de Referência OSI.

O modelo OSI define, para a implementação de protocolos de comunicação, uma arquitetura que se baseia na estratificação de níveis organizados de forma hierarquizada. Cada nível constitui uma camada e oferece um serviço para a camada imediatamente superior. Para que um nível possa executar esse serviço ele se utiliza dos níveis inferiores.

Apenas as camadas mais inferiores de duas hierarquias interconectadas encontram-se fisicamente ligadas; as demais

possuem uma ligação lógica, ou seja, a camada de nível N de uma hierarquia se comunica, logicamente, com a camada de nível N da hierarquia a ela interconectada.

A figura (IV.1) ilustra a arquitetura em níveis do Modelo de Referência OSI.

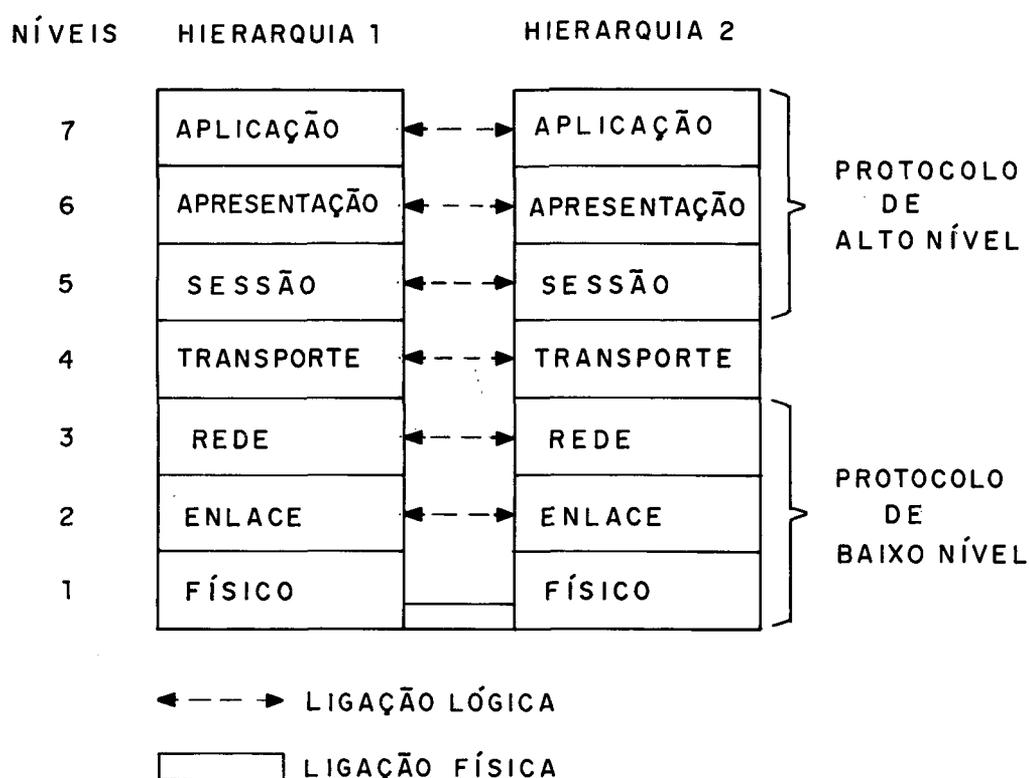


FIGURA IV. 1 - ARQUITETURA EM NÍVEIS - MODELO DE REFERÊNCIA OSI

Os três níveis superiores, Aplicação, Apresentação e Sessão, são tratados genericamente como camadas da aplicação, ou seja, correspondem ao protocolo de alto nível. Os três níveis inferiores, Rede, Enlace e Físico, são conhecidos como camadas

de comunicação, correspondendo ao protocolo de baixo nível. O nível intermediário, o de Transporte, é responsável pelas compatibilizações necessárias entre os dois protocolos.

No que se refere à padronização para implementação do protocolo de alto nível não existe muito trabalho em desenvolvimento. As aplicações são variadas e, geralmente, os sistemas são concebidos no escopo de um grupo pequeno e fechado, conseqüentemente os protocolos de alto nível desenvolvidos são muito específicos, atendendo apenas às necessidades do grupo envolvido. A tendência, no entanto, é que essa situação mude. Classes de aplicações podem ser delineadas e os serviços dos níveis superiores padronizados de acordo com a aplicação em questão.

Já no que diz respeito às camadas de comunicação, protocolo de baixo nível, o trabalho de padronização vem se desenvolvendo intensamente ao longo dos últimos dez anos. Esse trabalho começou pelos níveis mais inferiores e foi subindo na hierarquia até o nível 3. Existem vários padrões estabelecidos para o nível físico (V.24, V.35, X.20, X.21, X.21 bis, EIA RS 445) e para o nível de enlace (HDLC da ISO, X.25 nível 2 do CCITT, SDLC da IBM, LLC do IEEE 802).

A recomendação X.25 do CCITT especifica um protocolo de baixo nível para acesso a redes de comutação de pacotes, sendo que o nível 1 é compatível com a recomendação X.21 do próprio CCITT e o nível 2 é um subconjunto do HDLC. Ou seja, tanto o nível 1 como o nível 2 se baseiam em protocolos já existentes, é, portanto, o nível 3 que caracteriza a identidade do X.25.

O protocolo X.25 foi escolhido como protocolo de comunicação na interconexão dos COR com o DC por questão de uniformização, uma vez que este protocolo vem sendo amplamente utilizado no setor de comunicações e, também, pelo fato da ELETROBRÁS ter definido como protocolo de comunicação na interconexão do SINSC com os DC das empresas, o X.75 que é uma extensão do X.25 que abrange multiligação (sub-nível que permite multiplexação e demultiplexação de quadros para diversas ligações simples, ou seja, mais de um canal de comunicação físico).

O item (IV.2) apresenta uma descrição sucinta das funções de cada nível segundo a definição do Modelo de Referência OSI. O item (IV.3) apresenta cada nível conforme a visão da implementação alvo deste trabalho. São apresentadas as diferenças existentes nesta implementação e é explicado o porquê dessas diferenças.

IV.2 - NÍVEIS DO MODELO DE REFERÊNCIA OSI

A técnica básica de estruturação da arquitetura do Modelo de Referência OSI consiste na organização em níveis, cada um com um conjunto bem definido de funções. A seguir é apresentada uma descrição das funções de cada nível do Modelo de Referência.

IV.2.1 - NÍVEL FÍSICO (NÍVEL 1)

O Nível Físico é responsável pela transmissão e recepção de bits através do meio físico de comunicação, é, portanto, responsável pela especificação das características elétricas e mecânicas da ligação com o modem, tais como, níveis de tensão, conectores, etc... Este nível tem por objetivo prover meios para que duas entidades de enlace possam estabelecer uma conexão física e transmitir/receber bits por ela.

IV.2.2 - NÍVEL DE ENLACE (NÍVEL 2)

A função principal do Nível de Enlace é garantir a integridade dos dados que passam pelo canal de comunicação, ou seja, este nível é responsável pelo controle de erros na linha física (detecção e recuperação através, por exemplo, da retransmissão de dados). Este nível implementa um enlace lógico que é visto pelos níveis superiores como sendo imune a erros. Além disso é responsável pelo controle de fluxo de informações que trafegam no enlace. Este nível tem, portanto, por objetivo prover meios para que duas entidades de rede possam estabelecer uma conexão lógica e transmitir/receber dados sem se preocuparem com a integridade dos mesmos.

IV.2.3 - NÍVEL DE REDE (NÍVEL 3)

O Nível de Rede é responsável pela multiplexação do enlace lógico em várias conexões de rede, ou seja, permite que em uma conexão lógica trafeguem dados referentes a várias conexões de rede. Este nível provê, portanto, condições para que duas entidades de transporte estabeleçam uma ligação através da qual possam trocar dados. A essa ligação dá-se o nome de *circuito virtual*, e o ponto de acesso dele é denominado *canal lógico*. Desta maneira, otimiza-se a utilização do enlace lógico que tem uma correspondência de um para um com a linha física. Logo, o Nível de Rede é responsável pelo estabelecimento de conexões de rede para transferência de dados e pelo encaminhamento e direcionamento desses dados dentro da rede. Além disso tem a função de controle de fluxo, isto é, compatibiliza qualquer diferença de velocidade entre os equipamentos que executam o protocolo de transporte. O controle de fluxo é realizado separadamente para cada circuito virtual.

IV.2.4 - NÍVEL DE TRANSPORTE (NÍVEL 4)

O Nível de Transporte é responsável pela comunicação fim-a-fim entre dois computadores usuários da rede. Cabe ressaltar que todos os computadores envolvidos na transmissão de dados executam o protocolo de comunicação (níveis 1, 2 e 3), mas somente os computadores usuários da rede precisam executar os níveis superiores (níveis 4, 5, 6 e 7). Logo, o nível 4 é o primeiro da hierarquia (de baixo para cima) em condições de

garantir que uma mensagem atingiu seu destino. Além disso este nível é responsável pela segmentação das mensagens em fragmentos que caibam na unidade de transferência do nível de rede e no sentido oposto pela junção dos fragmentos que formam uma mensagem. Este nível provê meios para que duas entidades de sessão estabeleçam uma conexão de transporte.

IV.2.5 - NÍVEL DE SESSÃO (NÍVEL 5)

O Nível de Sessão é responsável pelo diálogo entre processos. Ele implementa uma conexão um a um sem multiplexação. Este nível provê suporte para o estabelecimento e manutenção de relações lógicas cooperativas e temporárias denominadas sessões.

IV.2.6 - NÍVEL DE APRESENTAÇÃO (NÍVEL 6)

O Nível de Apresentação é responsável pela conversão de códigos e formatos de representação dos dados sendo transmitidos/recebidos em uma sessão.

IV.2.7 - NÍVEL DE APLICAÇÃO (NÍVEL 7)

O Nível de Aplicação é a interface com os programas de aplicação, ou seja, com os processos que realmente utilizam,

geram e consomem os dados que trafegam em uma comunicação. Este nível representa a porta de entrada no ambiente OSI.

IV.3 - IMPLEMENTAÇÃO DO PROTOCOLO COR <=> DC

A implementação do protocolo COR <=> DC segue, como já foi dito, a estruturação do Modelo de Referência OSI em toda a sua hierarquia, utilizando para os níveis inferiores a padronização X.25 do CCITT. No entanto, o sistema ambiente desta implementação apresenta certas particularidades de hardware e software que levam a uma série de diferenças, às vezes simplificando a implementação em relação à padronização e às vezes complicando.

As principais características que particularizam esta implementação são:

- A arquitetura de hardware da comunicação não constitui uma configuração típica de rede, na realidade o que temos são dois sistemas ligados por meio de dois canais de comunicação, ou seja, duas ligações ponto-a-ponto (vide figura (III.3));
- Os COR se comunicam com o DC através de duas unidades de processamento independentes (dois operadores OM) e o tráfego de informações deve ser dividido pelos dois canais de comunicação, cada um gerenciado por uma dessas unidades de processamento;

- Em caso de falha em um canal de comunicação ou em um OM, todo o tráfego deve ser desviado para o sistema em funcionamento, e quando da entrada em serviço do equipamento em falha a comunicação deve retornar à sua configuração original;
- Os processos de aplicação, que são os usuários do serviço oferecido pelo protocolo, se encontram distribuídos pela rede local que constitui um COR, ou seja, se localizam em operadores diferentes dos OM, mas precisamente nos OR e OC (ver figura (IV.2));
- Uma falha na via de interconexão (VGI) que cause o isolamento de um OM dos demais operadores da rede local, impede, conseqüentemente, que os processos de aplicação se comuniquem com o operador OM que oferece o serviço de comunicação, logo também neste caso o fluxo de dados deve ser desviado para o operador OM em funcionamento.

Estas características influenciam, principalmente, a concepção dos níveis de Aplicação, Transporte e Rede. Este último passará, doravante, a ser referenciado por Nível de Canal Lógico e o penúltimo por Nível de Segmentação, uma vez que para a implementação em questão estas denominações são mais apropriadas.

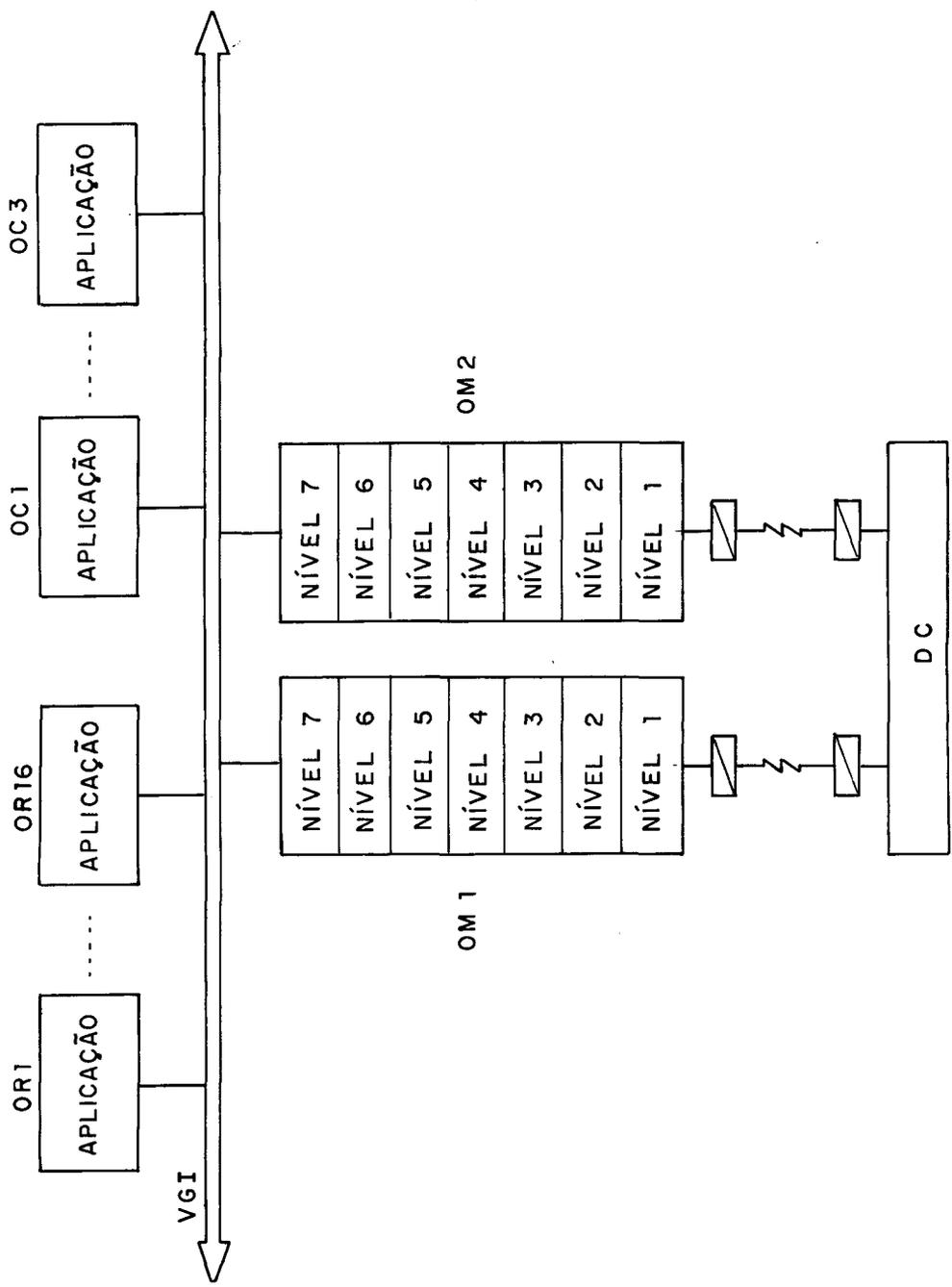


FIGURA IV.2 - ARQUITETURA DE SOFTWARE x ARQUITETURA DE HARDWARE

IV.3.1 - NÍVEL FÍSICO (NÍVEL 1)

A implementação do nível 1 tem por finalidade o oferecimento do suporte físico necessário ao estabelecimento do protocolo X.25 na comunicação de dados. Ela provê para os níveis superiores no enlace uma comunicação serial orientada a bit, síncrona, full-duplex, ponto-a-ponto, transmitindo e recebendo a 4800 bps e trabalhando com portadora constante. São utilizados dois circuitos independentes (dois canais de telecomunicações) com estas características, as quais estão de pleno acordo com o nível físico da recomendação X.25 opção X.21 bis que é compatível com a recomendação V.24 do CCITT e RS232-C da EIA. O oferecimento de tais serviços baseia-se na utilização do controlador de comunicações 8274 da INTEL.

IV.3.2 - NÍVEL DE ENLACE (NÍVEL 2)

A função básica do nível 2 consiste na transformação do enlace físico sujeito a erros em um enlace lógico relativamente livre de erros. Na sua implementação é empregado o método de transmissão de bits síncrona, formatação HDLC compatível, com geração e detecção automática (por hardware - 8274 da INTEL) de "flags", inserção e retirada também automática de zeros (mecanismo de "bit stuffing"), cálculo na transmissão e teste na recepção, também automático, de CRC, transmissão e detecção de sequência de aborto.

Adicionalmente, provê mecanismos de controle de fluxo do enlace, permitindo que ambas as pontas ajustem a carga de tráfego às respectivas disponibilidades de recursos ("buffers", capacidade de processamento,...). O procedimento para controle do enlace empregado é o Modelo Assíncrono Balanceado, LAPB da recomendação X.25, com o DC nas funções de ETD e os COR nas funções de ECD. Foram implementados os procedimentos de conexão, desconexão e transferência de informação com rejeição e retransmissão de quadros quando necessário, como descrito no nível 2 da recomendação X.25.

Não é o intuito deste trabalho transcrever os procedimentos descritos nesta recomendação, cabe apenas ressaltar que os parâmetros nela definidos podem ser reprogramados, se desejável, em cada partida fria do sistema através da re-edição de um arquivo de configuração. Abaixo são enumerados esses parâmetros:

- N1: número máximo de bits em um quadro;
- N2: número máximo de retransmissões;
- T1: temporizador de retransmissão;
- K: número máximo de quadros que podem ser transmitidos sem que ocorra aceitação por parte do sistema remoto;
- A,B: endereços dos comandos e respostas transmitidos e recebidos.

O nível 2 oferece ao seu usuário, o nível 3, os seguintes serviços:

- Detecção de erros;
- Recuperação de erros baseada na retransmissão de quadros;
- Procedimento de configuração dos parâmetros do nível 2;
- Procedimentos para conexão, desconexão e reiniciação do enlace lógico;
- Transporte de pacotes de/para o nível 3 na seqüência de entrega;
- Comunicação ao nível 3 quanto a ocorrência de situações de anormalidade irrecuperáveis;
- Reporte de estatísticas de desempenho do canal de comunicação.

IV.3.3 - NÍVEL DE CANAL LÓGICO (NÍVEL 3)

O nível 3 tem como principal objetivo a multiplexação do enlace lógico em vários circuitos virtuais. Denomina-se *canal lógico* a identificação ou endereço local de um circuito virtual. Nesta implementação os circuitos virtuais se estabelecem segundo os procedimentos do nível 3 do protocolo X.25 para Circuitos Virtuais Permanentes (CVP), ou seja, sem a necessidade de procedimentos de conexão/desconexão ("Set up / Clear"), devido à própria configuração lógica, ponto-a-ponto, dos enlaces CDR \Leftrightarrow DC e devido às características funcionais do diálogo entre esses dois sistemas. Além disso cada circuito virtual possui o mesmo número de canal lógico (mesmo endereço) tanto no

COR quanto no DC. Devido a isso o termo canal lógico pode ser usado intercambiavelmente com o termo circuito virtual, ou seja, estabelecer um canal lógico de número 1, por exemplo, significa estabelecer um circuito virtual cujo canal lógico no COR é o de número 1 e no DC também é o de número 1. São estabelecidos vários canais lógicos relativos aos tipos de dados definidos nos níveis superiores. O número de canais lógicos estabelecidos pode ser reprogramado a cada partida fria do sistema através da re-edição de um arquivo de configuração.

O sistema, como já foi ressaltado na apresentação das características que o particularizam, não apresenta uma configuração típica de rede, no entanto a função do nível 3 de direcionamento e encaminhamento de pacotes existe, uma vez que o fluxo de dados deve ser dividido pelos dois enlaces da ligação. Para isso foi criado o conceito de alocação e desalocação de canal lógico. Determinados canais lógicos estão prioritariamente alocados a um determinado enlace lógico, diz-se, então, que estes canais lógicos são de *alocação primária* a esse enlace. Os demais são de *alocação secundária*, ou seja, são alocados a ele apenas se houver algum problema com o outro enlace ou com o operador OM responsável por ele ou, ainda, com a VGI desse operador OM.

A gerência da alocação/desalocação dos canais lógicos não é um mecanismo simples. Um nó de uma rede (ECC) para realizar o encaminhamento e direcionamento de pacotes mantém tabelas com as rotas alternativas para um pacote atingir o seu ETD destino. Ou seja, toda a informação necessária encontra-se armazenada internamente ao nó. Já para o controle da alocação/desalocação

dos canais lógicos isto não é possível. Os dois enlaces pelos quais os pacotes podem trafegar são controlados por dois processadores diferentes, dois operadores OM, logo a alocação/desalocação dos canais lógicos depende de informações externas ao operador, mais precisamente do estado de funcionamento dos dois enlaces, dos dois operadores OM e da VGI dos dois operadores OM. Para implementá-la foi criada uma entidade denominada COORDENADOR BÁSICO que monitora os recursos (enlace, operador e VGI) e as condições necessárias para que os canais lógicos possam estar alocados ao enlace lógico controlado pelo OM em questão.

Outra particularidade desta implementação é a associação de prioridades na transmissão e recepção de dados dos canais lógicos. Na configuração desses canais os mesmos devem ser organizados a partir do de mais alta prioridade ao de mais baixa prioridade, ou seja, a ordem em que os canais lógicos são configurados implica nas prioridades a eles associada. Além disso os "buffers" de recepção devem ser distribuídos pelos canais lógicos de acordo com as prioridades dos mesmos, ou seja, canais lógicos de alta prioridade devem possuir mais "buffers" do que os de baixa prioridade.

O mecanismo de controle de fluxo implementado, apesar de ser idêntico ao do nível 3 da recomendação X.25, desempenha um papel um pouco diferente daquele definido para uma rede de comutação de pacotes. Numa rede podemos ter dois computadores hospedeiros (ETD) que desejam trocar dados entre si que acessem a rede a velocidades diferentes e que possuam capacidades de processamento também diferentes. O mecanismo de controle de

fluxo é responsável pela compatibilização dessas diferenças. Nesta implementação, devido à ligação ser ponto-a-ponto, não existe diferença na velocidade de transmissão e recepção de dados (este controle de fluxo recai apenas no controle do nível 2). O mecanismo de controle de fluxo do nível 3 tem por objetivo a compatibilização das velocidades de processamento, não só devido à diferença de capacidade das próprias máquinas, como também devido à diferença de velocidade no tratamento dos dados de cada canal lógico, uma vez que cada um possui uma prioridade diferente.

IV.3.4 - NÍVEL DE SEGMENTAÇÃO (NÍVEL 4)

Um protocolo de transporte tem como principal objetivo o controle fim-a-fim das mensagens lógicas trocadas entre dois computadores hospedeiros. No caso desta implementação, por não haver uma rede, mas apenas ligações ponto-a-ponto, como já foi dito acima, este controle não é tão crítico. É claro que existe ainda a sua necessidade, uma vez que nem o nível 2 nem o nível 3 realizam reinfileiramento de dados, não transmitidos ou à espera de aceitação, quando ocorre reiniciação de enlace (que leva a um "restart" do nível 3) ou de canal lógico. Mas devido à simplicidade da arquitetura utilizada e tendo em vista que o Nível de Sessão (nível 5) implementa um protocolo do tipo comando/resposta, optou-se por não implementar um protocolo de transporte específico, deixando que o nível 5 acumule esta função. A implementação de um protocolo de transporte nestas

circunstâncias pareceu redundante, implicando em perda de desempenho do sistema.

A função do nível 4 ficou, então, reduzida à segmentação das mensagens lógicas em fragmentos que caibam nos pacotes do nível 3 e no sentido contrário, à junção dos fragmentos que formam uma mensagem lógica. Isto explica o fato do nível 4 ser denominado Nível de Segmentação para esta implementação.

IV.3.5 - NÍVEL DE SESSÃO (NÍVEL 5)

O nível 5 tem por objetivo prover meios para que os níveis superiores executem sessões. Uma sessão é uma relação lógica temporária do tipo comando/resposta.

Para cada tipo de dado especificado pelo sistema aplicativo o Nível de Sessão permite que seja realizada uma sessão no sentido CDR -> DC e uma no sentido DC -> CDR simultaneamente. Logo, este nível recebe uma mensagem lógica dos níveis superiores, encaminha-a aos níveis inferiores para transmissão e fica aguardando uma resposta que ao ser recebida é repassada ao nível superior. No sentido oposto o nível 5 aguarda a recepção de uma mensagem lógica dos níveis inferiores, encaminha-a ao nível superior e fica aguardando uma resposta que ao ser recebida é transmitida ao nível inferior. Este serviço é oferecido para cada tipo de dado existente, sendo que cada um desses tipos corresponde a um canal lógico do nível 3.

Para realização dessas sessões o nível 5 provê mecanismos de sequencialização de mensagens e de respostas e de retransmissão no caso de erro nessa sequencialização ou devido à ocorrência de "time-out" na espera da resposta. O número de retransmissões é configurado para cada tipo de dado. Estes mecanismos realizam também a função de controle fim-a-fim na transmissão/recepção das mensagens. Logo, caso ocorra alguma reiniciação nos níveis inferiores, que leva ao descarte dos dados armazenados nesses níveis, o nível 5 detectará esse fato pela ocorrência de "time-out" na espera da resposta e providenciará a retransmissão da mensagem em questão.

IV.3.6 - NÍVEL DE APRESENTAÇÃO (NÍVEL 6)

O nível 6 é responsável pela formatação das mensagens enviadas pela aplicação em mensagens lógicas a serem transmitidas pelo Nível de Sessão. Ele organiza as mensagens em duas partes: o cabeçalho e a extensão. O cabeçalho obedece a um formato geral que inclui o tipo de dado, o tipo de operação para o tipo de dado em questão, a ligação, ou seja, o COR que está gerando a mensagem e a estação dentro do COR. A extensão porta os dados propriamente ditos da mensagem e segue um formato pré-definido para cada tipo de operação de cada tipo de dado definido pela aplicação.

Na figura (IV.3) é apresentada uma tabela com um resumo de todas as mensagens trocadas na ligação COR \Leftrightarrow DC, onde os campos TIPO, GRUPO e SUB-GRUPO codificam o tipo de dado da

TIPO	GRUPO	SUB-GRUPO	OPERAÇÃO	DC - COR	CANAL LÓGICO	CÓDIGO HEXA
SISTEMA (0)	INICIAÇÃO (0)	---	(5) REQ-INIC	→	1	05
			(6) ACK-INIC	←		06
DADOS EM TEMPO REAL (1)	CONTROLE ELÉTRICO (0)	---	(1) EXEC-CTRL	→	2	41
			(2) RSLT-CTRL	←		42
			(6) ACK-CTRL	→ ←		46
	AQUISIÇÃO (1)	ESTADOS DIGITAIS (0)	(1) INICS-DIG	←	3	61
			(2) INIC-DIG	←		62
			(3) VAR-DIG	←		63
			(4) INTGR-DIG	←		64
			(5) REQ-DIG	→		65
			(6) ACK-DIG	→ ←		66
	MEDIDAS ANALÓGICAS (1)		(1) INICS-ANL	←	4	69
(2) INIC-ANL			←	6A		
(3) VAR-ANL			←	6B		
(4) INTGR-ANL			←	6C		
(5) REQ-ANL			→	6D		
(6) ACK-ANL			→ ←	6E		
TOTALIZADORES (2)		(0) RELAT-TOT	←	5	70	
		(1) INICS-TOT	←		71	
		(2) INIC-TOT	←		72	
		(3) VAR-TOT	←		73	
		(4) INTGR-TOT	←		74	
		(5) REQ-TOT	→		75	
(6) ACK-TOT	→ ←	76				
SEQUÊNCIA DE EVENTOS (3)		(1) MSG-SOE	←	6	79	
		(5) REQ-SOE	→		7D	
		(6) ACK-SOE	→ ←		7E	
MENSAGENS OPERACIONAIS (2)	FORMATADAS (0)	CRT	(1) MSG-CRT	→ ←	7	81
			(5) REQ-CRT	→		85
			(6) ACK-CRT	→ ←		86
	ESTRUTURADAS (1)		(1) MSG-ESTR	→ ←	8	89
			(5) REQ-ESTR	→		8E
			(6) ACK-ESTR	→ ←		8F
NÃO FORMATADAS (1)	DISPLAY (0)		(1) MSG-DSP	→ ←	9	A1
			(5) REQ-DSP	→		A5
			(6) ACK-DSP	→ ←		A6

FIGURA IV.3 - TABELA DAS MENSAGENS TROCADAS NA LIGAÇÃO DC ↔ COR

mensagem e o campo OPERAÇÃO representa o tipo de operação dentro de um determinado tipo de dado.

A figura (IV.4) apresenta a organização geral da mensagem com os campos de cabeçalho e de extensão. O apêndice (A) contém a organização da extensão das mensagens para cada tipo de operação de cada tipo de dado existente.

TIPO	GRUPO	SUB - GRUPO	OPERAÇÃO
LIGAÇÃO		ESTAÇÃO	
EXTENSÃO			

LIGAÇÃO - COR JACAREPAGUÁ → 1
 COR CAMPINAS → 2
 COR FURNAS → 3
 COR ITUMBIARA → 4

ESTAÇÃO - NÚMEROS DE 1 A 32 PARA AS ESTAÇÕES DE CADA COR.

FIGURA IV.4 - FORMATO GERAL DAS MENSAGENS.

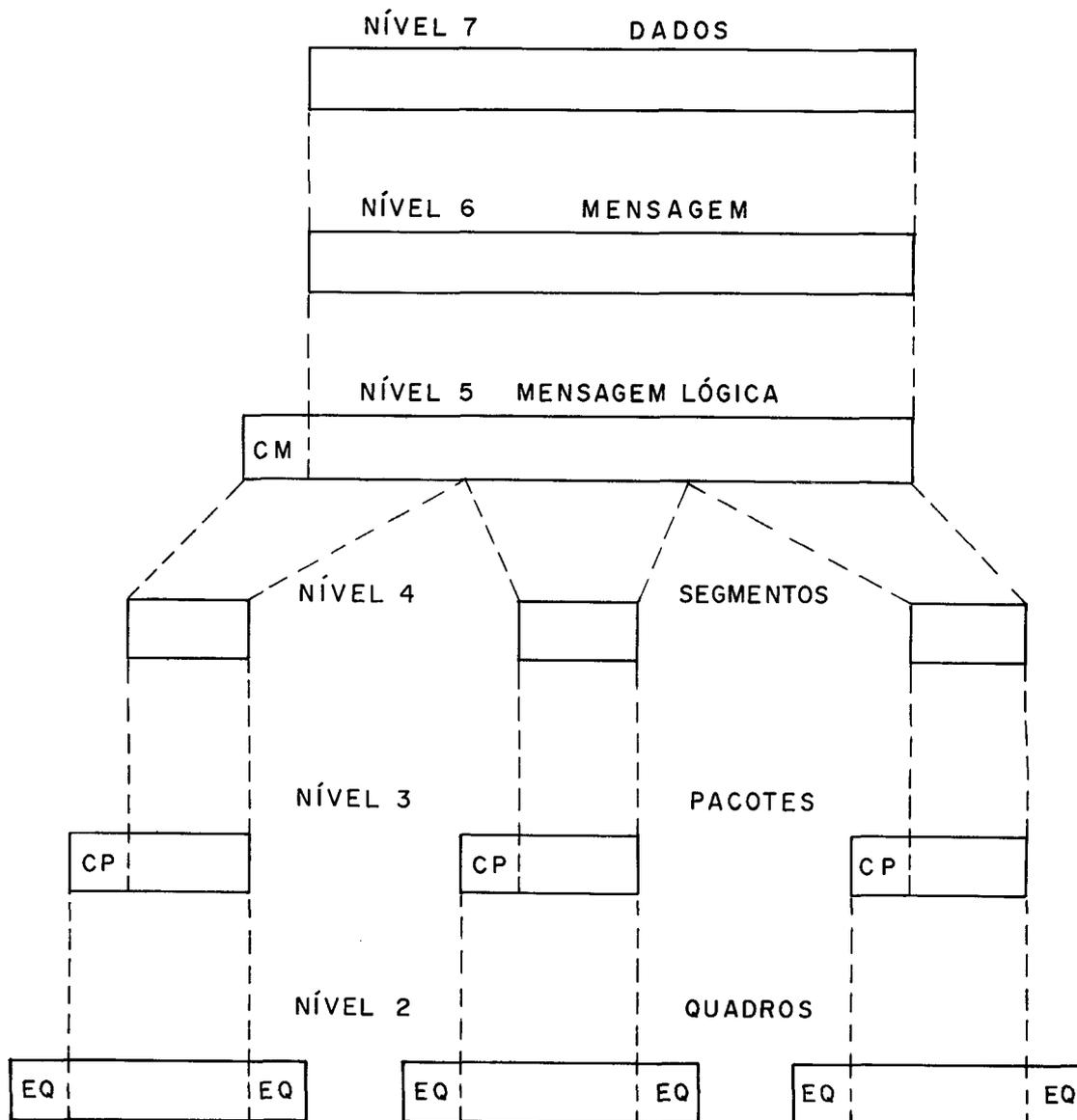
IV.3.7 - NÍVEL DE APLICAÇÃO (NÍVEL 7)

O nível 7 é a interface com os processos da aplicação. O COR é um sistema distribuído no qual cada operador executa determinada função. Logo, os processos da aplicação encontram-se distribuídos pela rede local, mais precisamente pelos operadores OR e OC, como já foi ressaltado na descrição das características do sistema e na figura (IV.2).

Este nível é responsável pela recepção dos dados a serem transmitidos dos processos da aplicação. Estes dados devem ser armazenados em ambos os operadores OM, uma vez que caso ocorra alguma falha no conjunto VGI/OM/CANAL DE COMUNICAÇÃO o outro conjunto acumula as funções do conjunto com defeito. Para isso foi criada uma entidade para cada tipo de dado denominada COORDENADOR DA APLICAÇÃO. O coordenador da aplicação, em um dado momento, encontra-se ativo em apenas um dos OM, o OM dito primário para o tipo de dado que ele representa. Em caso de falha este coordenador migra para o outro operador OM e no restabelecimento do conjunto a configuração retorna à original. Este mecanismo está intimamente ligado ao mecanismo de alocação/desalocação de canais lógicos do nível 3, portanto, a mesma entidade, o COORDENADOR BÁSICO, que gerencia a alocação/desalocação dos canais lógicos, gerencia a ativação/desativação dos COORDENADORES DA APLICAÇÃO.

Os coordenadores da aplicação são responsáveis pelo controle da seqüencialização dos dados que trafegam na ligação com os operadores OR e OC via VGI, tanto para os dados do sentido COR -> DC como do sentido DC -> COR.

A figura (IV.5) apresenta um esquema das unidades de transferência para cada nível do protocolo.



EQ - ENVELOPE DE QUADRO
 CP - CABEÇALHO DE PACOTE
 CM - CABEÇALHO DE MENSAGEM LÓGICA

FIGURA IV.5 - UNIDADES DE TRANSFERÊNCIA DOS NÍVEIS DO PROTOCOLO.

CAPÍTULO V

IMPLEMENTANDO O NÍVEL DE CANAL LÓGICO

V.1 - INTRODUÇÃO

Este capítulo trata da implementação do Nível de Canal Lógico (nível 3) do protocolo de comunicação.

Basicamente, este nível recebe do nível superior, Nível de Segmentação, segmentos, transforma-os em pacotes e os encaminha ao Nível de Enlace para transmissão e no sentido contrário, recebe pacotes do Nível de Enlace, transforma-os em segmentos e os repassa ao Nível de Segmentação. Portanto, a descrição deste nível é feita em três partes distintas. No item (V.2) é apresentada a descrição da interface de transmissão e recepção (TX/RX) com o nível 4 (Nível de Segmentação) e no item (V.3) a descrição da interface de TX/RX com o nível 2 (Nível de Enlace). No item (V.4) é descrita toda a implementação do nível, incluindo as suas funções e as respectivas interfaces.

V.2 - INTERFACE COM O NÍVEL 4

A interface com o nível superior, Nível de Segmentação (N4), pode ser dividida em duas partes: interface de transmissão (TX) e interface de recepção (RX).

V.2.1 - INTERFACE DE TRANSMISSÃO N4 -> N3

A interface de transmissão N4 -> N3 é realizada através de uma exclusão mútua entre o N3 e o N4, de uma fila de segmentos e de uma rotina do N4 chamada pelo N3, como pode ser visto na figura (V.1).

A exclusão mútua é implementada pelo intercâmbio *permissão_de_transmissão_xxx#iex*, que é um intercâmbio de interrupção, ou seja, funciona como uma sinalização (ver capítulo (II) item (II.4.1)) que indica que o canal lógico xxx encontra-se pronto para transmissão.

O N4 quando deseja transmitir segmentos de uma mensagem lógica de um tipo de dado ao N3, se apossa da permissão do canal lógico correspondente, deposita os segmentos na *fila_de_segmentos* do canal lógico e libera a permissão. Cada segmento vem acompanhado das seguintes informações: tamanho do segmento, indicação se o segmento é ou não o último de uma mensagem lógica e, no caso de ser o último, o endereço de um "flag" para liberação da área do N4 da mensagem lógica sendo transmitida. O N3, ao receber a aceitação do último pacote da mensagem lógica, libera a área do N4 através da chamada da rotina *libera_áreas_n4*.

Caso o N4 não obtenha a permissão para transmitir ele informa esse fato, *impossibilidade de transmissão*, ao nível imediatamente superior a ele, o Nível de Sessão (N5).

O N3, quando deseja transmitir um pacote de um determinado canal lógico ao N2, se apossa da permissão deste canal lógico, retira

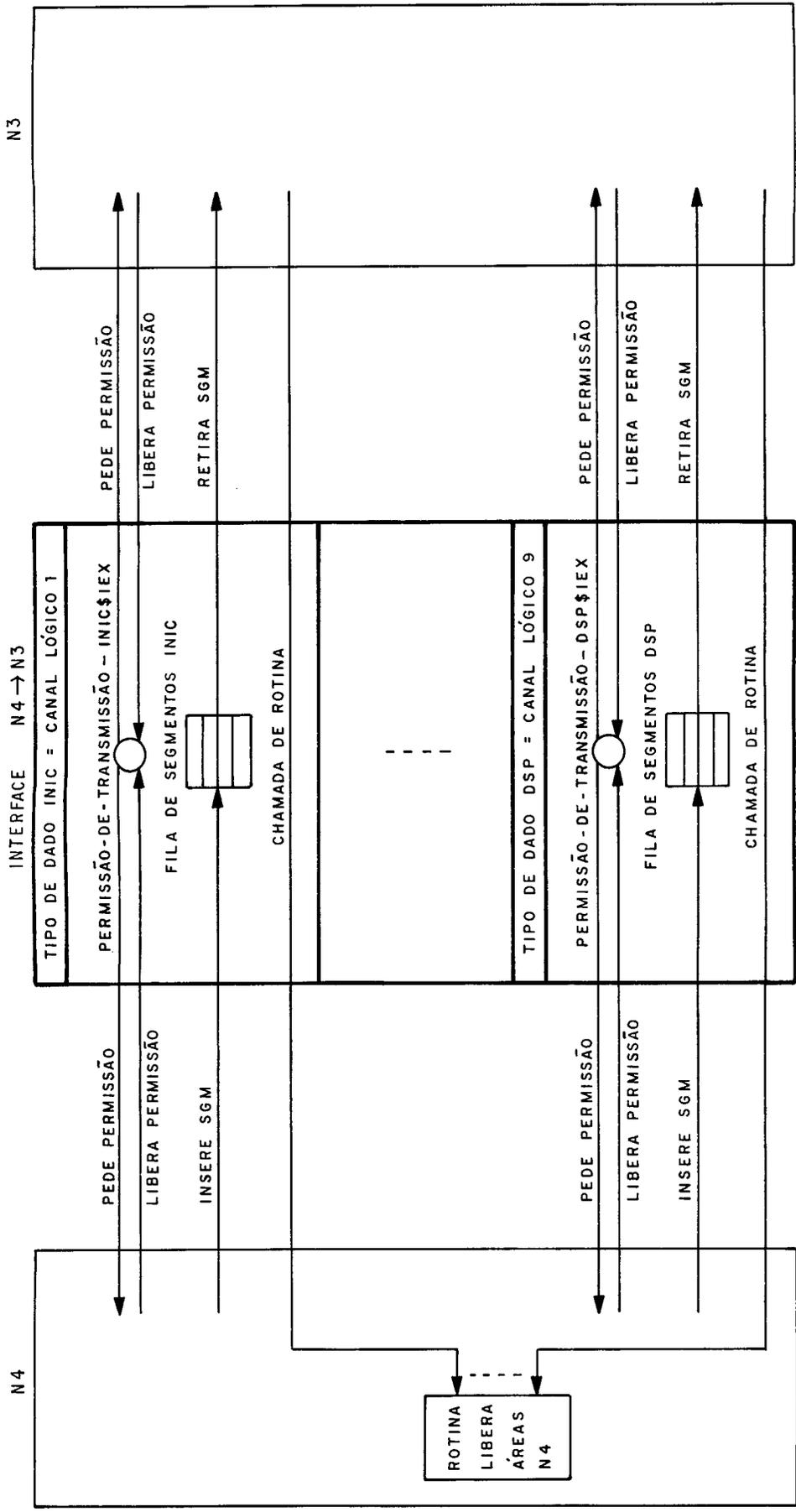


FIGURA V. 1 - INTERFACE DE TRANSMISSÃO
N4 → N3

um segmento da *fila_de_segmentos*, prepara o pacote, transmite-o ao N2 e o coloca à espera de aceitação. Em seguida ele libera a permissão.

O controle da permissão de transmissão é feito pelo N3. A existência de permissão significa que o canal lógico correspondente se encontra na fase de controle de fluxo. A permissão é concedida pelo N3 quando o canal lógico entra nesta fase e é retirada quando o mesmo a abandona. No item (V.4), que se refere à implementação propriamente dita, este mecanismo é explicado mais detalhadamente.

V.2.2 - INTERFACE DE RECEPÇÃO N3 -> N4

A interface de recepção N3 -> N4 é realizada através de uma tabela denominada *estado_rx_sg* que indica para cada tipo de dado do N4 (que corresponde a cada canal lógico do N3) se o mesmo encontra-se bloqueado ou desbloqueado para recepção de segmentos de uma mensagem lógica e através de uma chamada de rotina do N4 pelo N3, como pode ser visto na figura (V.2).

O N3, quando quer transferir segmentos de uma mensagem lógica para o N4, verifica na tabela o estado de recepção do tipo de dado correspondente. Caso o mesmo esteja desbloqueado, ele chama a rotina *recebe_segmento* do N4 passando o tipo de dado e o endereço da área onde o segmento se encontra. Ao retornar dessa chamada essa área é liberada pelo N3.

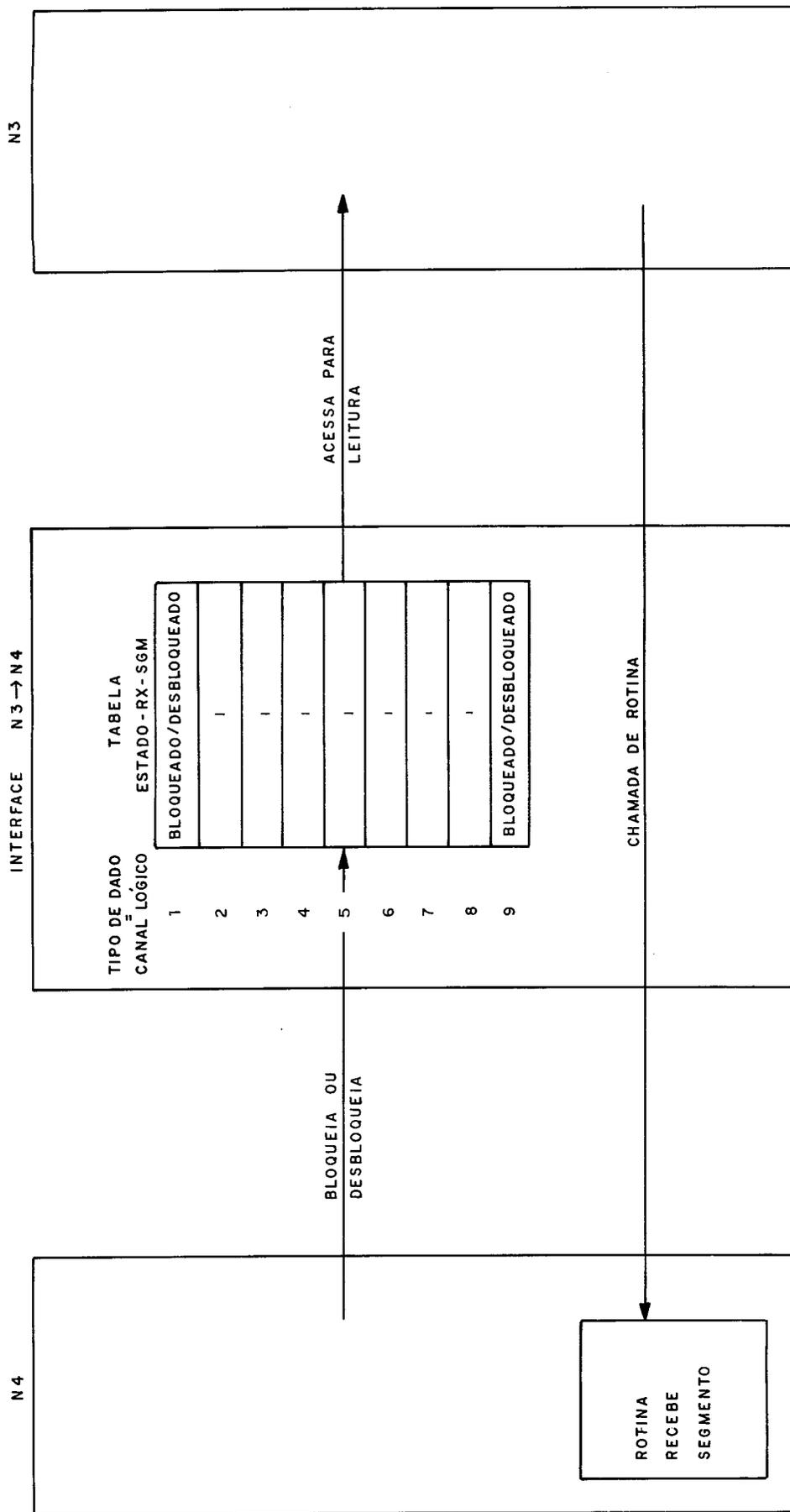


FIGURA V. 2 - INTERFACE DE RECEPÇÃO N3 → N4

O N4, ao receber um segmento, copia-o para uma área interna e dependendo da sua disponibilidade de áreas bloqueia ou não a recepção de novos segmentos. Quando o N4 está de posse de uma mensagem lógica completa, ele a transfere ao N5 (Nível de Sessão) e quando o mesmo liberar a área correspondente a essa mensagem ele desbloqueia a recepção de novos segmentos do N3, caso a mesma se encontre bloqueada.

A implementação do N4, devido à sua simplicidade, não constitui um capítulo desta descrição, mas no apêndice (B) pode ser encontrada uma figura que esquematiza a sua organização de software e uma descrição sucinta de como são realizadas as suas funções.

V.3 - INTERFACE COM O NÍVEL 2

A interface com o nível inferior, Nível de Enlace, é realizada através de quatro filas implementadas por quatro intercâmbios definidos pelo nível 2 (N2) e acessíveis pelo nível 3 (N3). As mensagens trocadas através desses intercâmbios estão divididas em classes, a saber:

- Classes de Mensagens enviadas ao N2 pelo N3

- . comandos -> CMD_N3
- . requisições -> REQ_N3
- . indicações -> IND_N3
- . respostas -> ANS_N3

- Classes de Mensagens enviadas ao N3 pelo N2
 - . respostas -> ANS_N2
 - . indicações -> IND_N2

A figura (V.3) ilustra a troca de mensagens da interface N3 <=> N2.

V.3.1 - COMANDOS DO NÍVEL 3 (CMD_N3)

São três os comandos que podem ser enviados ao N2 pelo N3:

- SET\$LINE\$ON -> solicita o estabelecimento do enlace físico ao N2 e, além disso, configura os parâmetros desse nível (N1, N2, T1, K e os endereços A e B);
- SET\$LINK\$ON -> solicita o estabelecimento do enlace lógico;
- SET\$LINK\$OFF -> solicita a desconexão do enlace lógico.

Esses comandos são encaminhados ao N2 via o intercâmbio FLM\$N1_N3\$EX numa estrutura do próprio N3. Esta estrutura é aproveitada pelo N2 para encaminhar uma resposta (ANS_N2) ao N3 via intercâmbio FLM\$N2_N4\$EX.

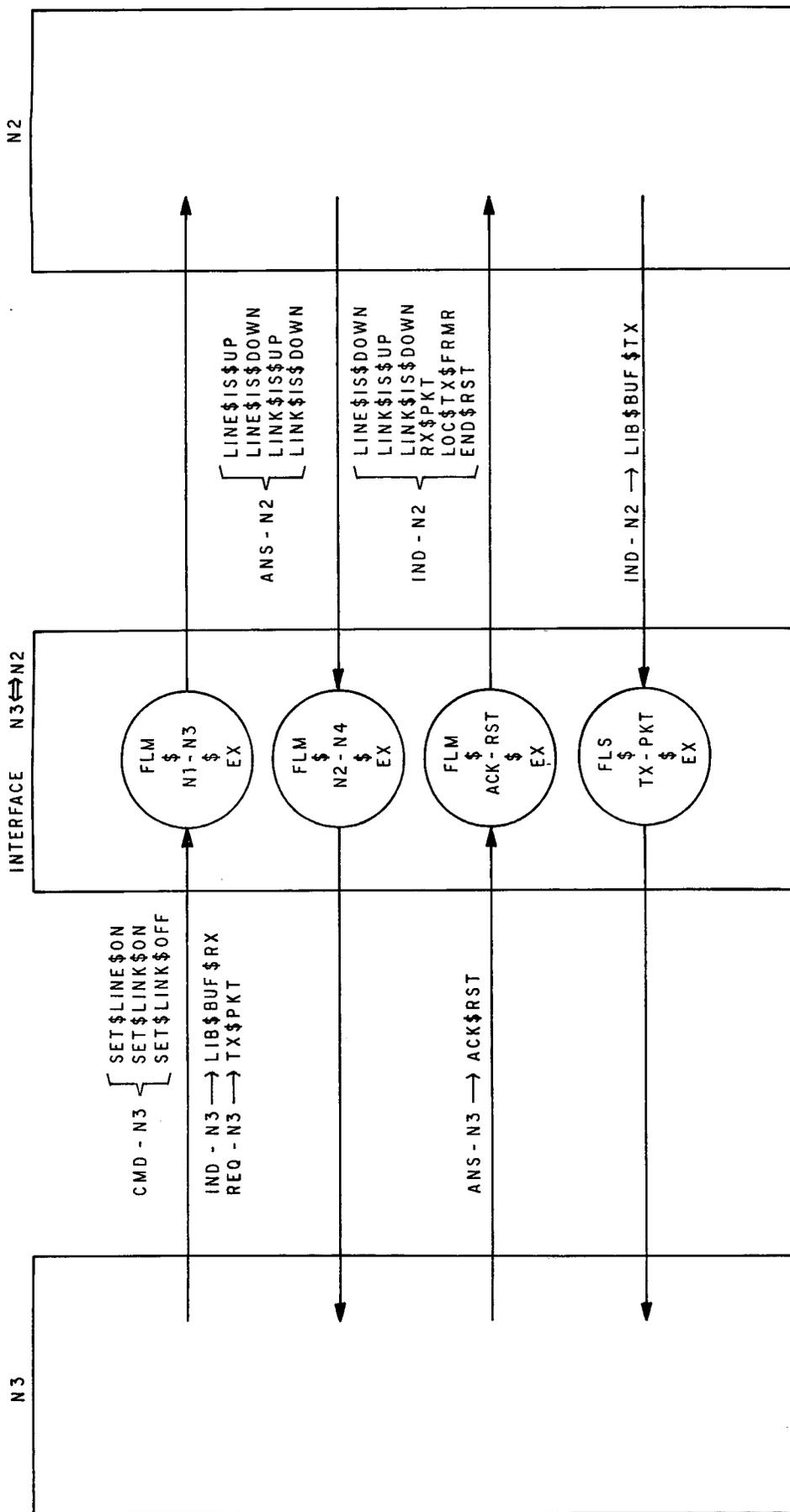


FIGURA V. 3 - INTERFACE N3 ↔ N2

V.3.2 - REQUISIÇÕES DO NÍVEL 3 (REQ_N3)

Existe apenas uma requisição do N3 para o N2, é a TX\$PKT que solicita a transmissão de um pacote através do enlace lógico. O N3 obtém estruturas para esta requisição através do intercâmbio FLS\$TX_PKT\$EX, o qual funciona como uma fila de áreas livres para transmissão. O N2 utilizando-se de uma indicação (IND_N2) denominada LIB\$BUF\$TX envia a esse intercâmbio estruturas que contêm o endereço de uma área livre na qual o N3 coloca o pacote a ser transmitido.

V.3.3 - INDICAÇÕES DO NÍVEL 3 (IND_N3)

Há, também, apenas uma indicação do N3 que é a LIB\$BUF\$RX que informa a liberação de uma área através da qual o N3 recebeu um pacote a ele encaminhado pelo N2 através de uma indicação (IND_N2) RX\$PKT via intercâmbio FLM\$N2_N4\$EX. Esta indicação é enviada ao N2 via intercâmbio FLM\$N1_N3\$EX.

V.3.4 - RESPOSTAS DO NÍVEL 3 (ANS_N3)

Existe somente uma resposta enviada pelo N3 ao N2. Sempre que o N3 recebe uma IND_N2 chamada LINK\$IS\$UP com código de indicação (INCCOD) igual a REM\$CONEC ou LOC\$RESET significa que o N2 remoto ou local reiniciou o enlace lógico. O N3 paraliza os seus procedimentos e responde (ANS_N3) ACK\$RST via o intercâmbio FLM\$ACK_RST\$EX, indicando reconhecimento de reiniciação do

enlace lógico e aguarda que o N2 reinicie a interface N3 <=> N2. O N3 é avisado do término da reiniciação através da indicação (IND_N2) END*RST via intercâmbio FLM\$N2_N4\$EX.

V.3.5 - RESPOSTAS DO NÍVEL 2 (ANS_N2)

Estas mensagens são enviadas pelo N2 como respostas a comandos do N3, via o intercâmbio FLM\$N2_N4\$EX.

Existem quatro tipos de respostas possíveis: LINE\$IS\$UP, LINE\$IS\$DOWN, LINK\$IS\$UP e LINK\$IS\$DOWN. LINE\$IS\$UP indica que o enlace físico encontra-se estabelecido, LINE\$IS\$DOWN informa ao N3 que o enlace físico não está ativo, LINK\$IS\$UP indica o estabelecimento do enlace lógico ou que o mesmo está ativo e LINK\$IS\$DOWN informa a desconexão do enlace lógico ou que o mesmo já está desconectado. Junto com cada resposta é retornado um código explicativo.

V.3.6 - INDICAÇÕES DO NÍVEL 2 (IND_N2)

As mensagens de indicações do N2 são enviadas espontaneamente mediante eventos ocorridos. A indicação LINE\$IS\$DOWN informa ao N3 a queda do enlace físico; LINK\$IS\$UP informa o estabelecimento ou reiniciação do enlace lógico por iniciativa da estação remota ou reiniciação do enlace por iniciativa local; LINK\$IS\$DOWN informa desconexão do enlace lógico; RX\$PKT informa a chegada de um pacote; END*RST indica o

fim do procedimento de reiniciação e LOC\$TX\$FRMR informa que o sistema local está rejeitando um quadro recebido e que o remoto deverá, brevemente, reiniciar ou desconectar o enlace lógico. Nestas condições o N3 não comanda mais o N2 até a finalização do procedimento de reiniciação ou desconexão.

Estas indicações são enviadas ao N3 via o intercâmbio FLM\$N2_N4\$EX. Elas contêm um código explicativo assim como as respostas do N2.

Existe ainda uma indicação LIB\$BUF\$TX que é enviada ao intercâmbio FLS\$TX_FKT\$EX, que libera uma área para transmissão de pacotes.

O apêndice (C) apresenta uma tabela explicativa de todos os códigos de retorno possíveis nas indicações e respostas do N2.

V.4 - IMPLEMENTAÇÃO DO NÍVEL 3

A implementação do Nível de Canal Lógico está fundamentada, basicamente, no mecanismo de Máquinas de Estados ou Autômatos Finitos.

Os procedimentos definidos neste nível podem se referir a todo o nível ou somente a um canal lógico específico. Foram desenvolvidas, portanto, duas máquinas de estados: a Máquina de Estados do Nível e a Máquina de Estados do Canal Lógico.

O nível pode estar em uma entre quatro fases: Estabelecimento do Enlace, "Restart", Normal ou

Desestabelecimento do Enlace. Cada fase encontra-se subdividida em sub-fases como pode ser visto na figura (V.4).

Um determinado canal lógico pode estar em uma entre duas fases: Espera ou Transferência de Dados. Cada fase encontra-se subdividida em sub-fases como pode ser visto na figura (V.5).

Cada canal lógico é controlado por uma Máquina de Estados de Canal Lógico. Inicialmente estas máquinas encontram-se na fase de Espera e aí permanecem até que a máquina de estados do nível atinja a fase Normal sub-fase Pronto. Quando isso ocorre, a máquina de estados do nível ativa procedimentos que levarão as máquinas de estados dos canais lógicos alocados ao enlace à fase de Transferência de Dados. Caso o nível saia da sub-fase Pronto, as máquinas de estados dos canais lógicos também saem da fase de Transferência de Dados, retornando à fase de Espera. Conseqüentemente, os canais lógicos só se encontram em Controle de Fluxo se o nível estiver na sub-fase Pronto.

Na implementação cada sub-fase corresponde a um ou mais estados da máquina de estados correspondente.

Os estados definidos nas máquinas de estados podem ser de dois tipos: PERMANENTES ou TRANSITÓRIOS. Os estados permanentes aguardam eventos gerados por entidades externas à máquina de estados e os transitórios aguardam eventos internos, ou seja, eventos gerados por uma ação desencadeada pela própria transição que levou a máquina a esse estado transitório.

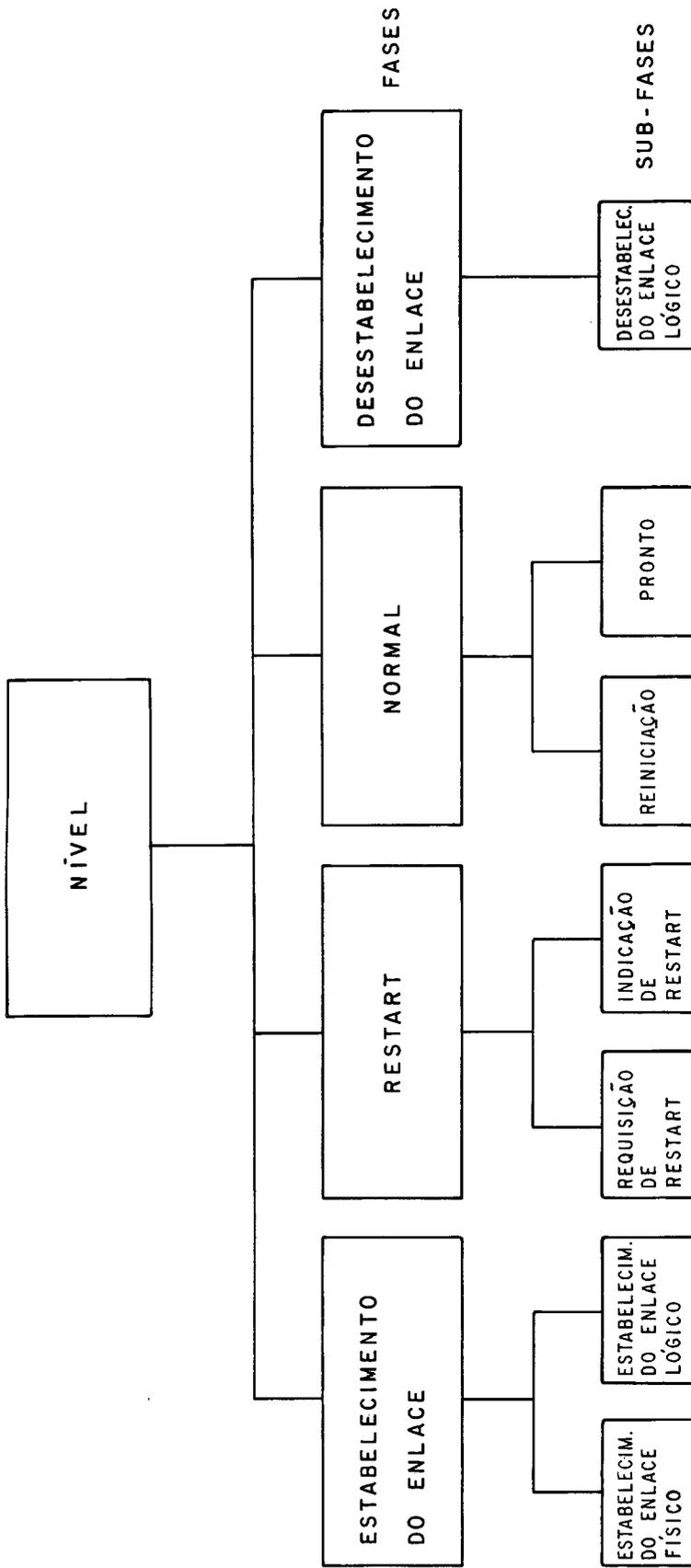


FIGURA V.4 - FASES E SUB-FASES DA MÁQUINA DE ESTADOS DO NÍVEL

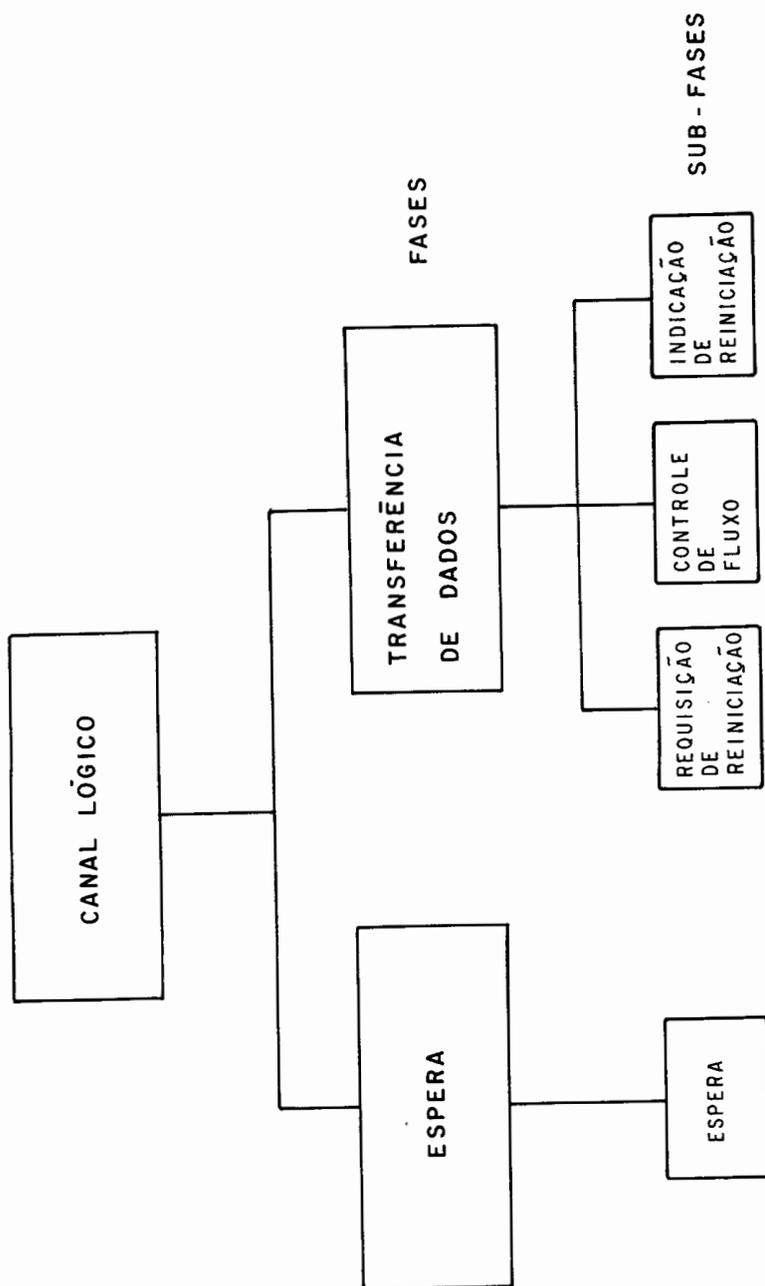


FIGURA V.5 - FASES E SUB-FASES DA MÁQUINA DE ESTADOS DE CANAL LÓGICO

V.4.1 - MÁQUINA DE ESTADOS DO NÍVEL

O objetivo desta máquina de estados é colocar o nível na sub-fase Pronto para que as máquinas dos canais lógicos possam iniciar seus procedimentos. Para que isto ocorra é necessário:

- estabelecer o enlace físico através da interface com o nível 2 (Nível de Enlace), procedimento de SET\$LINE\$ON;
- estabelecer o enlace lógico através da interface com o nível 2, procedimento de SET\$LINK\$ON;
- que a VGI do operador esteja operante.

Uma vez estas condições satisfeitas, o nível entra na sub-fase de Requisição de "Restart" na qual ele envia um pacote de RESTART e recebe um ACK_REST. Mediante a recepção do pacote ACK_REST, o nível entra na sub-fase Pronto que corresponde ao estado de Pronto da máquina de estados.

Caso o nível se encontre no estado de Pronto e receba um pacote de RESTART, ele entra na fase de Indicação de "Restart", na qual ele envia um pacote de ACK_REST.

É importante ressaltar que a execução de um "restart" de nível, seja por iniciativa local (Requisição de "Restart") ou por iniciativa remota (Indicação de "Restart"), implica na reiniciação de todos os canais lógicos alocados ao enlace.

Como já foi dito, para que o "restart" do nível seja completado é necessário que a VGI do operador esteja operante. Por outro lado, o nível pode estar no estado de Pronto, com o

enlace em perfeitas condições, e ser detectada uma falha na VGI. Quando isso ocorre é necessário derrubar o enlace lógico (procedimento de SET#LINK#OFF na interface com o nível 2) para que o sistema remoto, DC, tome conhecimento deste fato. Ou seja, a máquina de estados do nível entra na fase de Desestabelecimento do Enlace e, em seguida, torna a estabelecer o enlace lógico (sub-fase de Estabelecimento do Enlace Lógico), no entanto, uma vez que a VGI encontra-se inoperante, não será completado o procedimento de "restart". Assim, tanto o DC quanto o COR realocarão os canais lógicos deste enlace para o outro, caso este último esteja em condições de assumir essa função.

O mecanismo de alocação/desalocação dos canais lógicos é controlado pela entidade COORDENADOR BASICO, mas para isso é necessário que a máquina de estados do nível envie a esse coordenador uma informação a respeito do estado dos recursos do operador, ou seja, se os recursos necessários a essa alocação encontram-se disponíveis ou não. Os recursos são colocados disponíveis sempre que o nível entra no estado de Pronto, uma vez que isso só ocorre se o enlace estiver estabelecido e a VGI operante. Quando o nível sai do estado de Pronto devido a problemas com o enlace, os recursos continuam disponíveis no caso do N2 estar, ainda, tentando reiniciar o mesmo (sub-fase de Reiniciação). Quando o enlace sai realmente de serviço, ou seja, quando o restabelecimento do enlace depende de um novo comando do N3, os recursos são colocados não disponíveis, caso o outro OM esteja com os seus recursos disponíveis. Se o outro OM também estiver sem recursos, os recursos deste operador são

mantidos disponíveis porque apesar do enlace não estar funcionando a VGI está, logo o OM pode ficar armazenando os dados enviados pela aplicação (isto porque o COORDENADOR BÁSICO, que utiliza a informação do estado dos recursos, também é responsável pela ativação/desativação dos coordenadores da aplicação, ou seja, do nível 7 - ver capítulo (IV) item (IV.3.7)).

Outra condição que faz com que os recursos fiquem não disponíveis é uma falha na VGI. Neste caso os recursos são colocados não disponíveis qualquer que seja o estado dos recursos do outro operador OM, porque a falha na VGI isola o operador na rede, logo ele não consegue receber/enviar dados de/para a aplicação.

Para manter a situação desses recursos coerente, mediante a ocorrência de transições no estado dos recursos do outro operador OM, o N3 monitora tais recursos em duas condições a saber:

- 1ª condição -> . recursos do operador disponíveis
 - . enlace do operador com problema
 - . VGI do operador operante
 - . recursos do outro operador não disponíveis,

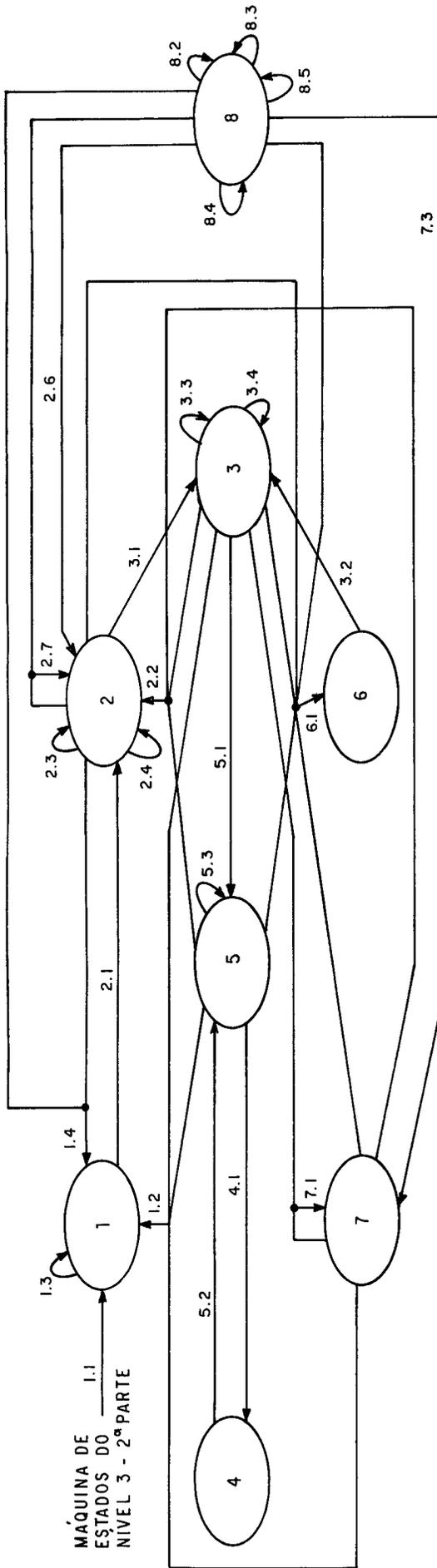
o N3 monitora o estado dos recursos do outro operador para que, caso a situação do mesmo se restabeleça e os recursos dele se tornem disponíveis, o N3 possa colocar os seus recursos não disponíveis e os canais lógicos possam migrar para o outro operador, ou seja, possam ser alocados ao outro enlace;

- 2ª condição -> . recursos do operador não disponíveis
- . enlace do operador com problema
 - . VGI do operador operante
 - . recursos do outro operador disponíveis,

o N3 monitora o estado dos recursos do outro OM para que, caso os mesmos se tornem não disponíveis, ele possa colocar os seus recursos disponíveis e este OM possa, pelo menos, armazenar os dados enviados pela aplicação.

As figuras (V.6) e (V.7) apresentam a máquina de estados do nível com seus estados e transições. Devido à complexidade dessa máquina, ela encontra-se representada em duas partes. A figura (V.6) apresenta as transições dos estados nos quais a VGI encontra-se operante. A figura (V.7) apresenta as transições dos estados nos quais a VGI encontra-se inoperante e as transições entre esses dois grupos de estados, ou seja, as transições referentes a própria condição da VGI.

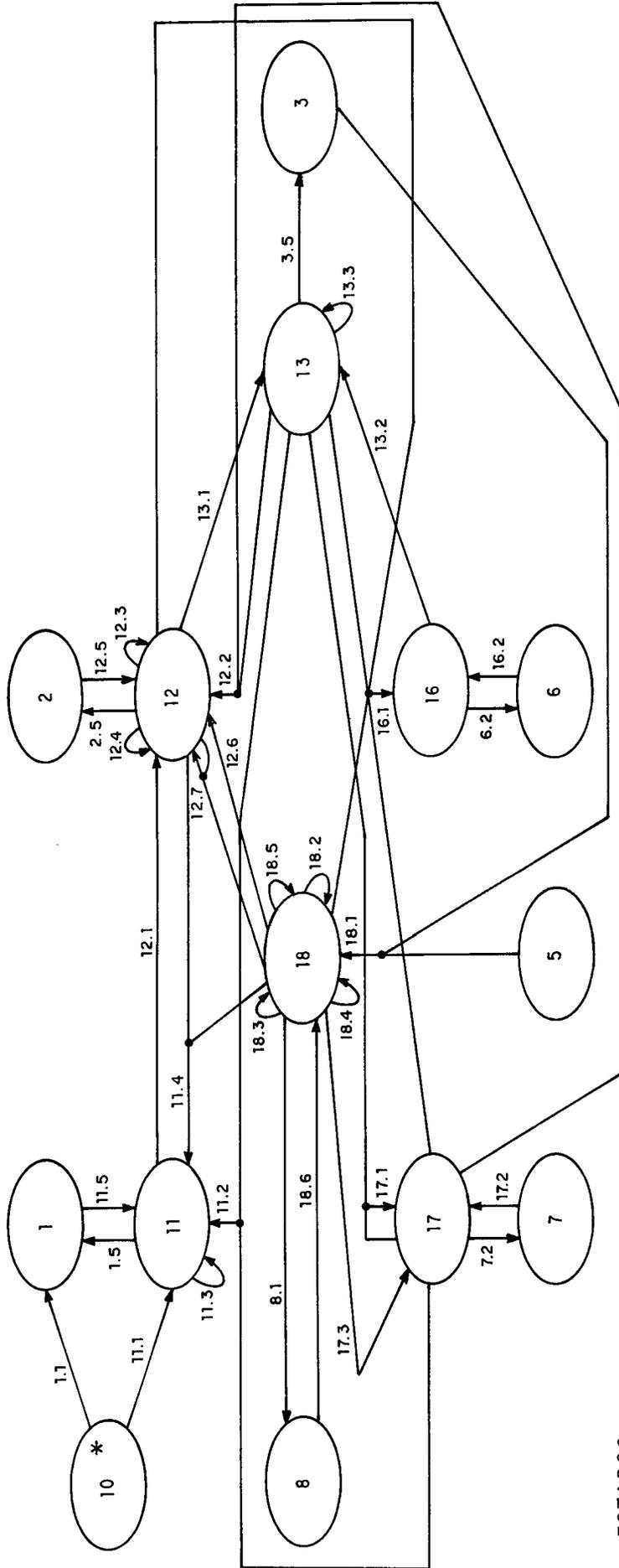
A seguir cada transição da máquina de estados do nível é descrita em termos do evento que a gera e das ações por ele desencadeadas. Cada evento e cada ação possui um código associado que é representado por ENi (evento de nível i, onde i é o número do evento de uma lista de eventos) ou ENIi (evento de nível interno i, onde i é o número do evento de uma lista de eventos) e Ai (ação i, onde i é o número da ação de uma lista de ações) respectivamente.



ESTADOS

- 1 - ESTABELECIMENTO DO ENLACE FÍSICO.
- 2 - ESTABELECIMENTO DO ENLACE LÓGICO.
- 3 - REQUISICÃO DE RESTART.
- 4 - INDICAÇÃO DE RESTART (TRANSITÓRIO).
- 5 - PRONTO.
- 6 - REINICIAÇÃO.
- 7 - ESPERA N2 REINICIAR.
- 8 - DESESTABELECE ENLACE LÓGICO.

FIGURA V.6 - MÁQUINA DE ESTADOS DO NÍVEL 3 - 1ª PARTE



ESTADOS

- 10 - ESTADO INICIAL (*)
- 11 - ESTABELECIMENTO DO ENLACE.
FÍSICO COM VGI INOPERANTE.
- 12 - ESTABELECIMENTO DO ENLACE
LÓGICO COM VGI INOPERANTE.
- 13 - ENLACE OK COM VGI.
INOPERANTE.
- 14 - NÃO EXISTE.
- 15 - NÃO EXISTE.
- 16 - REINICIAÇÃO COM VGI.
INOPERANTE.
- 17 - ESPERA N2 REINICIAR
COM VGI INOPERANTE.
- 18 - DESESTABELECE ENLACE LÓGICO
COM VGI INOPERANTE.

FIGURA V.7 - MÁQUINA DE ESTADOS DO NÍVEL 3 - 2ª PARTE

V.4.1.1 - DESCRIÇÃO DAS TRANSIÇÕES DA MÁQUINA DE ESTADOS DO NÍVEL

Transição 1.1

E - VGI OK (EN0).

A - Enviar ao N2 o CMD#N3 SET\$LINE\$ON (A0).

Transição 1.2

E - Recepção do N2 de uma IND#N2 LINE\$IS\$DOWN com INDCDD CTS\$NOK ou DCD\$NOK (EN2).

A - Desabilitar TX e RX de pacotes de todos os canais lógicos, quando a transição é procedente do estado 5. Esta ação inclui a retirada da permissão de transmissão de todos os canais lógicos da interface de transmissão com o N4 (A9). Desativar TRS (temporizador de "restart"), quando a transição é procedente do estado 3 (A13). Ativar monitoração dos recursos do outro OM, caso a mesma esteja desativada. Esta monitoração colocará os recursos deste OM não disponíveis, caso o outro esteja com os recursos disponíveis; e recolocará os recursos disponíveis, caso os recursos do outro OM se tornem não disponíveis (A5). Indicar às máquinas de estados dos canais lógicos que o nível saiu do estado de pronto, quando a transição é procedente do estado 5 (A11). Enviar ao N2 o CMD#N3 SET\$LINE\$ON (A0).

Transição 1.3

E - Recepção do N2 de uma ANS#N2 LINE#IS#DOWN com ANSCOD CTS#DCD#NOK ou CTS#NOK ou DCD#NOK (EN7).

A - Enviar ao N2 o CMD#N3 SET#LINE#ON (A0).

Transição 1.4

E - Recepção do N2 de uma ANS#N2 LINE#IS#DOWN com ANSCOD CMD#NOK_N2#NOT#OPER (EN8).

A - Enviar ao N2 o CMD#N3 SET#LINE#ON (A0).

Transição 1.5

E - VGI OK (EN0).

A - Enviar ao N2 o CMD#N3 SET#LINE#ON (A0).

Transição 2.1

E - Recepção do N2 de uma ANS#N2 LINE#IS#UP com ANSCOD CMD#ACCEPT (EN11).

A - Enviar ao N2 o CMD#N3 SET#LINK#ON (A1).

Transição 2.2

E - Recepção do N2 de uma IND#N2 LINK#IS#DOWN com INDCOD REM#DISC ou REM#DM ou COL#U#DIF#UP ou LOC#N2#SABM (EN3).

A - Desabilitar TX e RX de pacotes de todos os canais lógicos, quando a transição é procedente do estado 5 (A9). Desativar TRS, quando a transição é procedente do estado 3 (A13). Ativar monitoração dos recursos do outro OM, caso a mesma esteja desativada (A5). Indicar às máquinas de estados dos canais lógicos que o nível saiu do estado de

pronto, quando a transição é procedente do estado 5 (A11). Enviar ao N2 CMD#N3 SET#LINK#ON (A1).

Transição 2.3

E - Recepção do N2 de uma ANS#N2 LINK#IS#DOWN com ANSCOD COL#U#DIF#UP ou LOC#N2#SABM (EN9).

A - Enviar ao N2 o CMD#N3 SET#LINK#ON (A1).

Transição 2.4

E - Recepção do N2 de uma IND#N2 LINE#IS#DOWN com INDCOD CTS#NOK ou DCD#NOK (EN2).

A - Ação Nula.

Transição 2.5

E - VGI OK (EN0).

A - Ativar monitoração dos recursos do outro OM, caso a mesma esteja desativada (A5).

Transição 2.6

E - Recepção do N2 de uma ANS#N2 LINK#IS#DOWN com ANSCOD CMD#NOK_DISC#PHASE ou COL#U#DIF#DOWN ou LOC#DESC ou LOC#N2#DISC (EN10).

A - Enviar ao N2 CMD#N3 SET#LINK#ON (A1).

Transição 2.7

E - Recepção do N2 de uma ANS#N2 LINK#IS#DOWN com ANSCOD REM#DM (EN11).

A - Enviar ao N2 o CMD#N3 SET#LINK#ON (A1).

Transição 3.1

E - Recepção do N2 de uma ANS#N2 LINK#IS#UP com ANSCOD LOC#CONEC (EN12).

A - Enviar pacote de RESTART e disparar TRS (A3).

Transição 3.2

E - Recepção do N2 de uma IND\$N2 END\$RST (EN6).

A - Enviar pacote de RESTART e disparar TRS (A3).

Transição 3.3

E - N_ésimo "time-out" de TRS, onde n é um parâmetro configurável (EN15).

A - Ativar monitoração dos recursos do outro OM, caso a mesma esteja desativada (A5). Incrementar número de "time-outs" de TRS (A12). Enviar pacote de RESTART e disparar TRS (A3).

Transição 3.4

E - Qualquer "time-out" de TRS, exceto o n_ésimo (EN14).

A - Incrementar número de "time-outs" de TRS (A12). Enviar pacote de RESTART e disparar TRS (A3).

Transição 3.5

E - VGI OK (ENO)

A - Ativar monitoração dos recursos do outro OM, caso a mesma esteja desativada (A5). Enviar pacote de RESTART e disparar TRS (A3).

Transição 4.1

E - Recepção de pacote de RESTART (EN16).

A - Desabilitar TX e RX de pacotes de todos os canais lógicos (A9). Indicar às máquinas de estados dos canais lógicos que o nível saiu do estado de

pronto (A11). Enviar pacote de ACK_REST e gerar evento interno de término de "restart" remoto (A4).

Transição 5.1

- E - Recepção de pacote de ACK_REST ou RESTART (EN17 ou EN16 respectivamente).
- A - Desativar TRS (A13). Desativar monitoração dos recursos do outro OM, caso a mesma esteja ativada (A6). Esperar término de iniciação do operador e colocar os recursos deste OM disponíveis (A7). Indicar às máquinas de estados dos canais lógicos que o nível entrou no estado de pronto (A10).

Transição 5.2

- E - Término de "restart" remoto (EN18).
- A - Indicar às máquinas de estados dos canais lógicos que o nível entrou no estado de pronto (A10).

Transição 5.3

- E - Recepção de pacote de ACK_REST (EN17).
- A - Descartar pacote recebido (ação nula).

Transição 6.1

- E - Recepção do N2 de uma IND\$N2 LINK\$IS\$UP com INDCOD REM\$CONEC, quando a transição é procedente do estado 2, 3 ou 5; ou com INDCOD REM\$CONEC ou LOC\$RESET quando a transição é procedente do estado 7 ou 8 (EN5).
- A - Desabilitar TX e RX de pacotes de todos os canais lógicos quando a transição é procedente do estado

5 (A9). Desativar TRS, quando a transição é procedente do estado 3 (A13). Indicar às máquinas de estados dos canais lógicos que o nível saiu do estado de pronto, quando a transição é procedente do estado 5 (A11). Enviar ao N2 uma ANS#N3 ACK\$REST (A14).

Transição 6.2

E - VGI OK (ENO).

A - Ativar monitoração dos recursos do outro DM, caso a mesma esteja desativada (A5).

Transição 7.1

E - Recepção do N2 de uma IND#N2 LINK\$IS\$DOWN com INDCOD SOFT\$ERROR, FR\$REJ, LOC#N2\$I\$REJ, PROT\$BREAK, REM#N2\$REJ ou LOC#N2\$FRMR ou de uma IND#N2 LOC\$TX\$FRMR com INDCOD C\$INV, LENG\$REJ ou NR\$INV (EN4).

A - Desabilitar TX e RX de pacotes de todos os canais lógicos, quando a transição é procedente do estado 5 (A9). Desativar TRS, quando a transição é procedente do estado 3 (A13). Indicar às máquinas de estados dos canais lógicos que o nível saiu do estado de pronto, quando a transição é procedente do estado 5 (A11). Ação nula, quando a transição é procedente dos demais estados.

Transição 7.2

E - VGI OK (ENO).

A - Ativar monitoração dos recursos do outro OM, caso a mesma esteja desativada (A5).

Transição 7.3

E - Recepção do N2 de uma ANS#N2 LINK\$IS\$UP com ANSCOD CMD#NOK_LINK\$SET\$UP ou CMD#NOK_REJ\$COND (EN13).

A - Ação nula.

Transição 8.1

E - VGI OK (ENO).

A - Ativar monitoração dos recursos do outro OM, caso a mesma esteja desativada (A5).

Transição 8.2

E - Recepção do N2 de uma IND#N2 LINE\$IS\$DOWN com INDCOD CTS#NOK ou DCD#NOK (EN2).

A - Ação nula.

Transição 8.3

E - Recepção do N2 de uma IND#N2 LINK\$IS\$DOWN com INDCOD REM#DISC, REM#DM, COL\$U\$DIF\$UP ou LOC#N2\$SABM (EN3).

A - Ação nula.

Transição 8.4

E - Recepção do N2 de uma IND#N2 LINK\$IS\$DOWN com INDCOD SOFT\$ERROR, FR\$REJ, LOC#N2\$I\$REJ, PROT\$BREAK, REM#N2\$REJ ou LOC#N2\$FRMR ou de uma IND#N2 LOC\$TX\$FRMR com INDCOD C\$INV, LENG\$REJ ou NR\$INV (EN4).

A - Ação nula.

Transição 8.5

E - Recepção de pacote de RESTART ou ACK_REST (EN16 ou EN17 respectivamente).

A - Descartar pacote recebido (ação nula).

Transição 11.1

E - VGI trancada (EN1).

A - Enviar ao N2 o CMD#N3 SET\$LINE\$ON (A0).

Transição 11.2

E - Recepção do N2 de uma IND#N2 LINE\$IS\$DOWN com INDCOD CTS#NOK ou DCD#NOK (EN2).

A - Enviar ao N2 o CMD#N3 SET\$LINE\$ON (A0).

Transição 11.3

E - Recepção do N2 de uma ANS#N2 LINE\$IS\$DOWN com ANSCOD CTS\$DCD#NOK ou CTS#NOK ou DCD#NOK (EN7).

A - Enviar ao N2 o CMD#N3 SET\$LINE\$ON (A0).

Transição 11.4

E - Recepção do N2 de uma ANS#N2 LINE\$IS\$DOWN com ANSCOD CMD#NOK_N2#NOT\$OPER (EN8).

A - Enviar ao N2 o CMD#N3 SET\$LINE\$ON (A0).

Transição 11.5

E - VGI trancada (EN1).

A - Desativar monitoração dos recursos do outro OM, caso a mesma esteja ativada (A6). Colocar recursos não disponíveis, caso os mesmos estejam disponíveis (A8).

Transição 12.1

E - Recepção do N2 de uma ANS#N2 LINE\$IS\$UP com ANSCOD
CMD\$ACCEPT (EN11).

A - Enviar ao N2 o CMD#N3 SET\$LINK\$ON (A1).

Transição 12.2

E - Recepção do N2 de uma IND#N2 LINK\$IS\$DOWN com
INDCOD REM\$DISC ou REM\$DM ou COL\$U\$DIF\$UP ou
LOC#N2\$SABM (EN3).

A - Enviar ao N2 o CMD#N3 SET\$LINK\$ON (A1).

Transição 12.3

E - Recepção do N2 de uma ANS#N2 LINK\$IS\$DOWN com
ANSCOD COL\$U\$DIF\$UP ou LOC#N2\$SABM (EN9).

A - Enviar ao N2 o CMD#N3 SET\$LINK\$ON (A1).

Transição 12.4

E - Recepção do N2 de uma IND#N2 LINE\$IS\$DOWN com
INDCOD CTS\$NOK ou DCD\$NOK (EN2).

A - Ação nula.

Transição 12.5

E - VGI trancada (EN1).

A - Desativar monitoração dos recursos do outro OM,
caso a mesma esteja ativada (A6). Colocar
recursos não disponíveis, caso os mesmos estejam
disponíveis (A8).

Transição 12.6

E - Recepção do N2 de uma ANS#N2 LINK\$IS\$DOWN com ANSCOD CMD\$NOK_DISC\$PHASE ou COL\$U\$DIF\$DOWN ou LOC\$DESC ou LOC\$N2\$DISC (EN10).

A - Enviar ao N2 o CMD\$N3 SET\$LINK\$ON (A1).

Transição 12.7

E - Recepção do N2 de uma ANS#N2 LINK\$IS\$DOWN com ANSCOD REM\$DM (EN11).

A - Enviar ao N2 o CMD\$N3 SET\$LINK\$ON (A1).

Transição 13.1

E - Recepção do N2 de uma ANS#N2 LINK\$IS\$UP com ANSCOD LOC\$CONEC (EN12).

A - Ação nula.

Transição 13.2

E - Recepção do N2 de uma IND#N2 END\$RST (EN6).

A - Ação nula.

Transição 13.3

E - Recepção de pacote de RESTART ou ACK_REST (EN16 ou EN17 respectivamente).

A - Descartar pacote recebido (ação nula).

Transição 16.1

E - Recepção do N2 de uma IND#N2 LINK\$IS\$UP com INDCOD REM\$CONEC quando a transição é procedente do estado 12 ou 13 ou com INDCOD REM\$CONEC ou LOC\$RESET quando a transição é procedente do estado 17 ou 18 (EN5).

A - Enviar ao N2 uma ANS#N3 ACK\$REST (A14).

Transição 16.2

E - VGI trancada (EN1).

A - Desativar monitoração dos recursos do outro OM, caso a mesma esteja ativada (A6). Colocar recursos não disponíveis, caso os mesmos estejam disponíveis (A8).

Transição 17.1

E - Recepção do N2 de uma IND#N2 LINK\$IS\$DOWN com INDCOD SOFT\$ERROR, FR\$REJ, LOC#N2\$I\$REJ, PROT\$BREAK, REM#N2\$REJ ou LOC#N2\$FRMR ou de uma IND#N2 LOC\$TX\$FRMR com INDCOD C\$INV, LENG\$REJ ou NR\$INV (EN4).

A - Ação nula.

Transição 17.2

E - VGI trancada (EN1).

A - Desativar monitoração dos recursos do outro OM, caso a mesma esteja ativada (A6). Colocar recursos não disponíveis, caso os mesmos estejam disponíveis (A8).

Transição 17.3

E - Recepção do N2 de uma ANS#N2 LINK\$IS\$UP com ANSCOD CMD\$NOK_LINK\$SET\$UP ou CMD\$NOK_REJ\$COND (EN13).

A - Ação nula.

Transição 18.1

E - VGI trancada (EN1).

A - Desabilitar TX e RX de pacotes de todos os canais lógicos, quando a transição é procedente do estado 5 (A9). Desativar TRS, quando a transição é procedente do estado 3 (A13). Desativar monitoração dos recursos do outro DM, caso a mesma esteja ativada (A6). Colocar recursos não disponíveis, caso os mesmos estejam disponíveis (A8). Indicar às máquinas de estados dos canais lógicos que o nível saiu do estado de pronto, quando a transição é procedente do estado 5 (A11). Enviar ao N2 o CMD#N3 SET\$LINK\$OFF (A2).

Transição 18.2

E - Recepção do N2 de uma IND#N2 LINE\$IS\$DOWN com INDCOD CTS\$NOK ou DCD\$NOK (EN2).

A - Ação nula.

Transição 18.3

E - Recepção do N2 de uma IND#N2 LINK\$IS\$DOWN com INDCOD REM\$DISC, REM\$DM, COL\$U\$DIF\$UP ou LOC#N2\$SABM (EN3).

A - Ação nula.

Transição 18.4

E - Recepção do N2 de uma IND#N2 LINK\$IS\$DOWN com INDCOD SOFT\$ERROR, FR\$REJ, LOC#N2\$I\$REJ, PROT\$BREAK, REM#N2\$REJ ou LOC#N2\$FRMR ou de uma IND#N2 LOC\$TX\$FRMR com INDCOD C\$INV, LENG\$REJ ou NR\$INV (EN4).

A - Ação nula.

Transição 18.5

E - Recepção de pacote de RESTART ou ACK_REST (EN16 ou EN17 respectivamente)

A - Descartar pacote recebido (ação nula).

Transição 18.6

E - VGI trancada (EN1).

A - Desativar monitoração dos recursos do outro OM, caso a mesma esteja ativada (A6). Colocar recursos não disponíveis, caso os mesmos estejam disponíveis (A8).

V.4.1.2 - CLASSIFICAÇÃO DOS ESTADOS DA MÁQUINA DE ESTADOS DO NÍVEL NAS SUAS SUB-FASES

Sub-fase Estabelecimento do Enlace Físico

- . Estado 10 - Estado Inicial
- . Estado 1 - Estabelecimento do Enlace Físico
- . Estado 11 - Estabelecimento do Enlace Físico com VGI Inoperante

Sub-fase Estabelecimento do Enlace Lógico

- . Estado 2 - Estabelecimento do Enlace Lógico
- . Estado 12 - Estabelecimento do Enlace Lógico com VGI Inoperante

Sub-fase Requisição de "Restart"

- . Estado 3 - Requisição de "Restart"
- . Estado 13 - Enlace OK com VGI Inoperante

Sub-fase Indicação de "Restart"

- . Estado 4 - Indicação de "Restart" (Transitório)

Sub_fase Reiniciação

- . Estado 6 - Reiniciação
- . Estado 7 - Espera N2 Reiniciar
- . Estado 16 - Reiniciação com VGI Inoperante
- . Estado 17 - Espera N2 Reiniciar com VGI Inoperante

Sub-fase Pronto

- . Estado 5 - Pronto

Sub_fase Desestabelecimento do Enlace Lógico

- . Estado 8 - Desestabelece Enlace Lógico
- . Estado 18 - Desestabelece Enlace Lógico com VGI
Inoperante

V.4.2 - MÁQUINAS DE ESTADOS DOS CANAIS LÓGICOS

As máquinas de estados dos canais lógicos encontram-se, inicialmente, na fase de Espera. Para que essas máquinas possam abandonar essa fase e entrar em Transferência de Dados, é necessário que:

- A máquina de estados do nível esteja no estado de pronto;
- Os recursos necessários à alocação dos canais lógicos a esse enlace estejam disponíveis;
- O canal lógico correspondente esteja alocado a esse enlace.

Quando a máquina de estados do nível entra no estado de pronto, ela envia uma indicação às máquinas dos canais lógicos sobre este fato e coloca os recursos do COORDENADOR BÁSICO como disponíveis.

O COORDENADOR BÁSICO possui a informação de quais canais lógicos são de alocação primária a este enlace e quais são de alocação secundária (esta informação é armazenada na iniciação do sistema em um arquivo de configuração para o COORDENADOR BÁSICO). Mediante esta informação e mediante a monitoração do estado do outro operador OM e do estado dos seus recursos, o coordenador decide se o canal lógico deve ser ou não alocado a este enlace. Ou seja, os canais lógicos de alocação primária, uma vez que os recursos deste operador encontram-se disponíveis, são alocados a este enlace. Já os de alocação secundária só serão aqui alocados se o outro operador estiver fora de serviço ou se os seus recursos não estiverem disponíveis.

Uma vez decidido quais os canais lógicos que devem ser alocados a este enlace o COORDENADOR BÁSICO envia às máquinas de estados desses canais uma indicação de alocação de canal lógico.

Logo, a essa altura as três condições necessárias para que um canal lógico entre na fase de Transferência de Dados estão satisfeitas. A máquina de estados do canal lógico entra, então, na sub-fase de Requisição de Reiniciação, na qual ela enviará um pacote de REINIC e aguardará um ACK_REINIC. Mediante a recepção do pacote ACK_REINIC, o canal lógico entra na sub-fase de Controle de Fluxo.

Na sub-fase de Controle de Fluxo o canal lógico está com a transmissão e a recepção de pacotes habilitadas, logo foi atingido o objetivo principal da máquina de estados. Enquanto o canal lógico permanecer nesta sub-fase pacotes de dados e de aceitação são trocados entre o COR e o DC.

Existem quatro fatos que fazem com que uma máquina de estados de canal lógico saia da sub-fase de Controle de Fluxo, a saber:

- Ocorrência de algum erro no mecanismo de recepção ou transmissão de pacotes;
- Recepção de um pacote de REINIC;
- Saída do nível da sub-fase de Pronto;
- Desalocação do canal lógico feita pelo COORDENADOR BÁSICO.

A ocorrência de algum erro na recepção ou na transmissão de pacotes, tais como, erro de sequencialização de pacotes recebidos ou a não aceitação de pacotes transmitidos, detectada pelo estouro do temporizador de transmissão (TT), leva a máquina de estados à sub-fase de Requisição de Reiniciação na qual o canal lógico será reiniciado e um pacote de REINIC será enviado. Mediante a recepção do pacote de ACK_REINIC, a máquina volta à sub-fase de Controle de Fluxo.

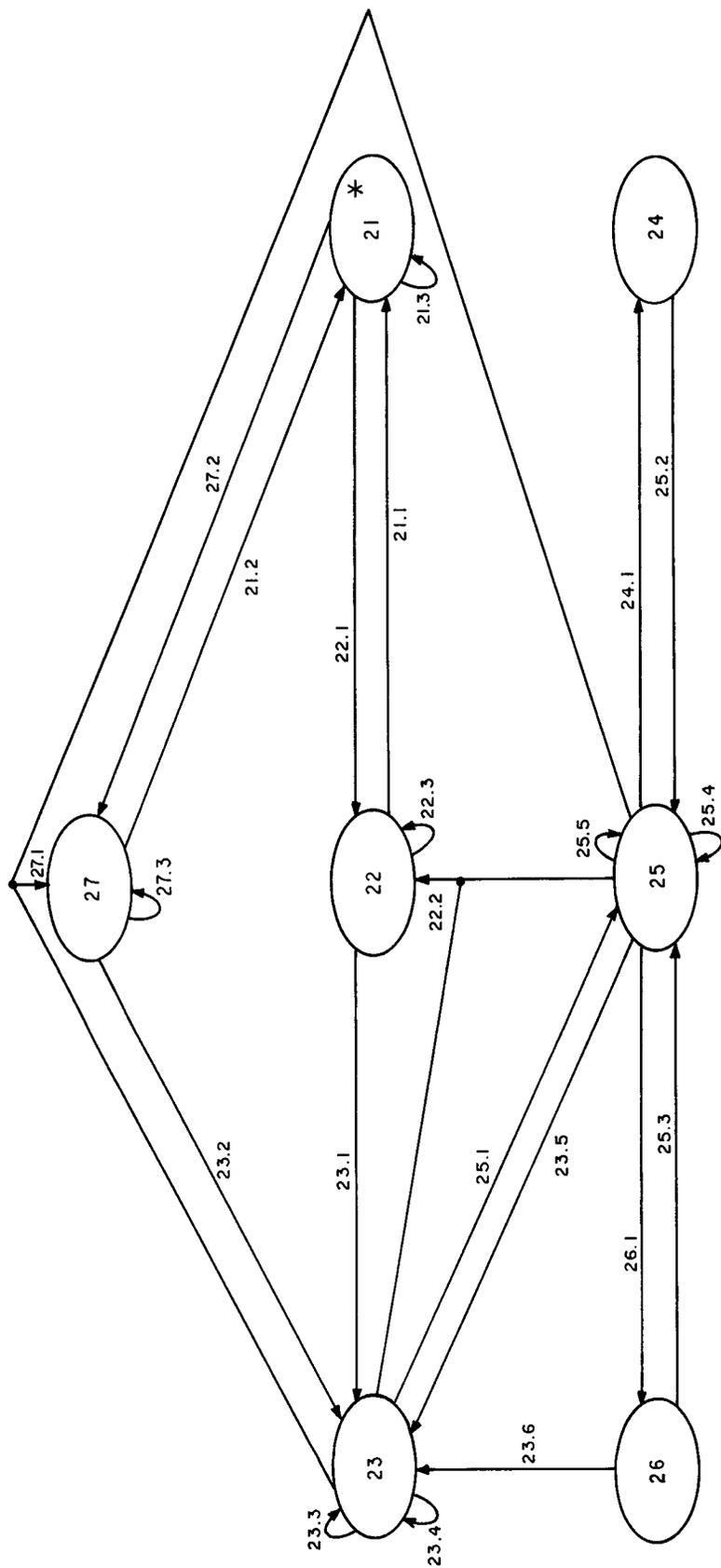
A recepção de um pacote de REINIC indica que o sistema remoto está reiniciando o canal lógico devido a algum erro por ele

detectado. A máquina de estados entra na sub-fase de Indicação de Reiniciação na qual ela reinicia o canal lógico, envia um ACK_REINIC e retorna à sub-fase de Controle de Fluxo.

Nos dois casos anteriores a máquina de estados do canal lógico não chega a sair da fase de Transferência de Dados, uma vez que os eventos ocorridos dizem respeito apenas ao canal lógico correspondente. A saída do nível da sub-fase Pronto leva as máquinas de todos os canais lógicos a saírem da fase de Transferência de Dados, quer eles estejam na sub-fase de Controle de Fluxo quer estejam em qualquer outra sub-fase desta fase. Os canais lógicos voltam, então, à fase de Espera.

A desalocação de um canal lógico feita pelo COORDENADOR BÁSICO também leva a máquina de estados deste canal a abandonar a fase de Transferência de Dados, qualquer que seja a sub-fase que ela se encontra, e a ir para a fase de Espera. Cabe ressaltar que a recepção de uma indicação de desalocação do canal lógico sem que o nível tenha saído do estado de Pronto, ou seja, os recursos deste operador continuam disponíveis, só ocorre para canais lógicos de alocação secundária a este enlace. Isto é, o COORDENADOR BÁSICO está desalocando o canal lógico porque o seu operador primário encontra-se em condições de assumi-lo e, portanto, o mesmo deve migrar para o outro operador.

A figura (V.8) apresenta a máquina de estados de um canal lógico com seus estados e transições. No item a seguir cada transição desta máquina é descrita em termos do evento que a gera e das ações por ele desencadeadas. Cada evento e cada ação



ESTADOS

- 21 - ESPERA NÃO PRONTO DESALOCADO (ESTADO INICIAL (*)).
- 22 - ESPERA PRONTO DESALOCADO.
- 23 - REQUISICÃO DE REINIC
- 24 - INDICAÇÃO DE REINIC (TRANSITÓRIO).
- 25 - CONTROLE DE FLUXO OK.
- 26 - TRATA PACOTE RECEBIDO (TRANSITÓRIO).
- 27 - ESPERA NÃO PRONTO ALOCADO.

ESTADO DO CL	NÍVEL	CANAL LÓGICO	RECURSOS
21	NÃO PRONTO	DESALOCADO	DISPONÍVEIS, / NÃO DISPONÍVEIS
22	PRONTO	DESALOCADO	DISPONÍVEIS
23,24,25,26	PRONTO	ALOCADO	DISPONÍVEIS
27	NÃO PRONTO	ALOCADO	DISPONÍVEIS

FIGURA V.8 - MÁQUINA DE ESTADOS DE UM CANAL LÓGICO.

possui um código associado que é representado por ECI (evento de canal lógico i, onde i é o número do evento de uma lista de eventos) ou ECII (evento de canal lógico interno (associado a um estado transitório) i, onde i é o número do evento de uma lista de eventos) e Ai (ação i, onde i é o número da ação de uma lista de ações) respectivamente.

V.4.2.1 - DESCRIÇÃO DAS TRANSIÇÕES DAS MÁQUINAS DE ESTADOS DOS CANAIS LÓGICOS

Transição 21.1

E - Indicação de que o nível saiu do estado de pronto (EC20).

A - Ação nula.

Transição 21.2

E - Indicação do COORDENADOR BÁSICO de desalocação do canal lógico (EC22).

A - Liberar o COORDENADOR BÁSICO (A19).

Transição 21.3

E - Recepção de qualquer pacote (EC24 ou EC25 ou EC26).

A - Descartar pacote recebido (ação nula).

Transição 22.1

E - Indicação de que o nível entrou no estado de pronto (EC19).

A - Ação nula.

Transição 22.2

- E - Indicação do COORDENADOR BÁSICO de desalocação do canal lógico (EC22).
- A - Desativar TRN (temporizador de reiniciação), quando a transição é procedente do estado 23 (A18). Desabilitar TX e RX de pacotes, quando a transição é procedente do estado 25. Esta ação inclui a retirada da permissão de transmissão deste canal lógico da interface de transmissão com o N4 (A21). Reiniciar o canal lógico, quando a transição é procedente do estado 25 (A17). Liberar COORDENADOR BÁSICO (A19).

Transição 22.3

- E - Recepção de qualquer pacote (EC24 ou EC25 ou EC26).
- A - Descartar pacote recebido (ação nula).

Transição 23.1

- E - Indicação do COORDENADOR BÁSICO de alocação do canal lógico (EC21).
- A - Enviar pacote de REINIC e disparar TRN (A15).

Transição 23.2

- E - Indicação de que o nível entrou no estado de pronto (EC19).
- A - Enviar pacote de REINIC e disparar TRN (A15).

Transição 23.3

- E - "Time-out" de TRN (EC23).
- A - Enviar pacote de REINIC e disparar TRN (A15).

Transição 23.4

- E - Recepção de pacote de dados ou aceitação (EC26).
- A - Descartar pacote recebido (ação nula). Desativar TRN (A18). Enviar pacote de REINIC e disparar TRN (A15).

Transição 23.5

- E - "Time-out" de TT (temporizador de transmissão) (EC30).
- A - Desabilitar TX e RX de pacotes (A21). Reiniciar canal lógico (A17). Enviar pacote de REINIC e disparar TRN (A15).

Transição 23.6

- E - Tratamento de pacote recebido nok (EC129).
- A - Desabilitar TX e RX de pacotes (A21). Reiniciar canal lógico (A17). Enviar pacote de REINIC e disparar TRN (A15).

Transição 24.1

- E - Recepção de pacote de REINIC (EC24).
- A - Desabilitar TX e RX de pacotes (A21). Reiniciar canal lógico (A17). Enviar pacote de ACK_REINIC e gerar evento interno de *término de reiniciação remota* (A16).

Transição 25.1

- E - Recepção de pacote de ACK_REINIC ou REINIC (EC25 ou EC24).
- A - Desativar TRN (A18). Habilitar TX e RX de pacotes. Esta ação inclui o envio da permissão de

transmissão do canal lógico para a interface de transmissão com o N4 (A20).

Transição 25.2

E - Término de reiniciação remota (ECI27).

A - Habilitar TX e RX de pacotes (A20).

Transição 25.3

E - Tratamento de pacote recebido ok (ECI28).

A - Ação nula.

Transição 25.4

E - Evento nulo.

A - Transmitir pacote (ação nula).

Transição 25.5

E - Recepção de pacote de ACK_REINIC (EC25).

A - Descartar pacote recebido (ação nula).

Transição 26.1

E - Recepção de pacote de dados ou aceitação (EC26).

A - Tratar pacote e gerar evento interno de *tratamento de pacote recebido ok* ou *tratamento de pacote recebido nok* conforme resultado obtido no tratamento do pacote (A22).

Transição 27.1

E - Indicação de que o nível saiu do estado de pronto (EC20).

A - Desativar TRN, quando a transição é procedente do estado 23 (A18). Desabilitar TX e RX de pacotes, quando a transição é procedente do estado 25

(A21). Reiniciar canal lógico, quando a transição é procedente do estado 25 (A17).

Transição 27.2

E - Indicação do COORDENADOR BÁSICO de alocação do canal lógico (EC21).

A - Ação nula.

Transição 27.3

E - Recepção de qualquer pacote (EC24 ou EC25 ou EC26).

A - Descartar pacote recebido (ação nula).

V.4.2.2 - CLASSIFICAÇÃO DOS ESTADOS DAS MÁQUINAS DE ESTADOS DOS CANAIS LÓGICOS NAS SUAS SUB-FASES

Sub-fase Espera

- . Estado 21 - Espera Não Pronto Desalocado (Estado Inicial)
- . Estado 22 - Espera Pronto Desalocado
- . Estado 27 - Espera Não Pronto Alocado

Sub-fase Requisição de Reiniciação

- . Estado 23 - Requisição de "Reinic"

Sub-fase Indicação de Reiniciação

- . Estado 24 - Indicação de "Reinic" (Estado Transitório)

Sub-fase Controle de Fluxo

- . Estado 25 - Controle de Fluxo
- . Estado 26 - Trata Pacote Recebido (Estado Transitório)

V.4.3 - ORGANIZAÇÃO DO NÍVEL 3

O Nível de Canal Lógico juntamente com Nível de Segmentação (Nível 4), o Nível de Enlace e o Nível Físico (Nível 1) constituem um objeto denominado COD, Objeto de Comunicação de Dados, responsável pela implementação do protocolo de comunicação.

Em termos de comunicação com outros objetos, no que se refere ao nível 3, ele apenas se comunica com o objeto COORDENADOR BÁSICO, como ficou claro na descrição das máquinas de estados, e com o objeto IOM, Objeto de Iniciação do Operador OM. Quando a máquina de estados do nível quer entrar no estado de Pronto, ela espera que o objeto IOM termine a iniciação do operador (Ver item (V.4.1.1) transição (5.1)). A comunicação com os demais objetos é feita pelo nível 4.

A figura (V.9) apresenta a organização do objeto COD com as interfaces entre os níveis que o compõem, as interfaces internas do N3 e as interfaces do N3 com outros objetos.

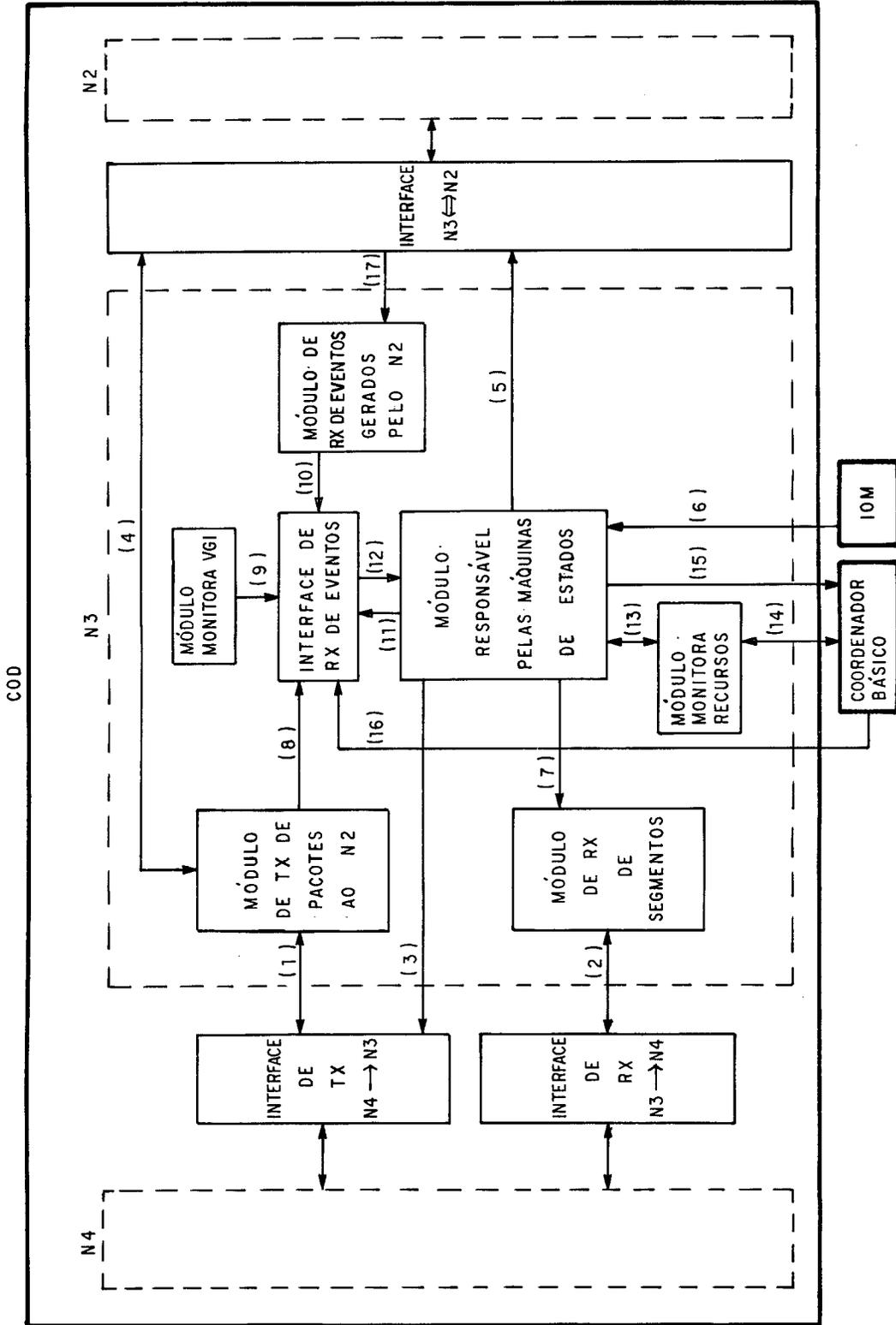


FIGURA V.9 - ORGANIZAÇÃO DO OBJETO COD

O N3 é composto por seis módulos distintos:

- Módulo Responsável pelas Máquinas de Estados,
- Módulo Monitora VGI,
- Módulo Monitora Recursos,
- Módulo de Recepção de Eventos Gerados pelo Nível 2,
- Módulo de Transmissão de Pacotes ao Nível 2,
- Módulo de Recepção de Segmentos.

V.4.3.1 - MÓDULO RESPONSÁVEL PELAS MÁQUINAS DE ESTADOS

O Módulo Responsável pelas Máquinas de Estados implementa a máquina de estados do nível e as máquinas de estados dos canais lógicos configurados. Qualquer evento destinado a essas máquinas é enviado a este módulo através da Interface de Recepção de Eventos. Esta interface aceita apenas um tipo de evento em um determinado instante. O evento recebido é tratado pela máquina de estados devida e só, então, a interface é liberada para recepção de um novo evento.

O mecanismo deste módulo é basicamente o seguinte:

- Recebe um evento e o classifica dentro de uma lista de eventos que inclui eventos de nível e de canal lógico (ENi, ENIi, ECI, ECII);
- De acordo com o evento recebido e o estado atual da máquina de estados a que o evento se destina, determina através de uma tabela a transição a ser executada;

- Executa as ações desta transição;
- Atualiza o estado atual da máquina de estados correspondente com o estado a ser assumido após esta transição;
- Libera a interface para recepção de um novo evento, caso a própria transição não tenha gerado um evento.

Logo as máquinas de estados são configuráveis, uma vez que são descritas na forma de uma tabela que contém para cada estado os eventos que podem ocorrer e para cada um desses eventos as ações a serem executadas e o próximo estado que a máquina deve assumir.

Este módulo recebe eventos através da interface (12). Pela interface (11), ele pode liberar a interface de recepção de eventos ou enviar um evento interno (evento gerado pela própria máquina de estados para sair de um estado transitório) ou, ainda, a máquina de estados do nível pode enviar eventos às máquinas de estados dos canais lógicos. Este tipo de evento é replicado, ou seja, é enviada uma cópia do evento para cada máquina de estados de canal lógico.

Através da interface (13) é possível ativar ou desativar a monitoração dos recursos do outro operador OM. As condições e a finalidade deste mecanismo foram explicadas no item (V.4.1).

A máquina de estados do nível coloca os recursos do COORDENADOR BÁSICO disponíveis ou não através da interface (15) e espera o término de iniciação do operador pela interface (6). As máquinas de estados enviam comandos ao N2 (CMD_N3) e pacotes

de RESTART, ACK_REST, REINIC e ACK_REINIC pela interface (5); pela interface (3) estas máquinas habilitam ou desabilitam a transmissão de pacotes pelos canais lógicos.

O COORDENADOR BÁSICO envia a este módulo eventos de alocação/desalocação de canais lógicos para as máquinas de estados dos mesmos via a Interface de Recepção de Eventos (interface (16)).

Além de manter o controle das máquinas de estados, este módulo também controla a recepção de pacotes dos canais lógicos, quando os mesmos se encontram na sub-fase de Controle de Fluxo. Nesta sub-fase a recepção de um pacote de dados ou de aceitação é um evento que leva a máquina de estados do canal lógico a um estado transitório (Estado 26 - Trata Pacote Recebido) e desencadeia a ação de tratamento do pacote recebido, a qual gera um evento interno ou de *tratamento Ok* ou de *tratamento Nok* (ver descrição da Máquina de Estados dos Canais Lógicos item (V.4.2.1)) enviado à máquina de estados do canal lógico via interface (11).

Os mecanismos de RX e TX de pacotes implementados por este módulo e pelo Módulo de Transmissão de Pacotes ao Nível 2 respectivamente, utilizam uma série de informações em comum. Essas informações encontram-se agrupadas em um Bloco de Controle de Canal Lógico (BCL) e o acesso a esse bloco é controlado por exclusão mútua. As informações contidas em cada BCL são:

PS_TX - P(S) (número de seqüência de transmissão de um pacote) do próximo pacote de dados a ser transmitido;

PS_RX - P(S) do próximo pacote de dados a ser recebido;

PS_RX_AUX - PS_RX auxiliar; usado para controlar a aceitação de dados enquanto o sistema local estiver ocupado até a expiração da janela;

PS_RX_A - P(S) do primeiro pacote de dados com recepção autorizada;

PS_RX_NA - P(S) do primeiro pacote de dados com recepção não autorizada;

PS_TX_A - Primeiro P(S) com transmissão já autorizada pelo outro sistema;

PS_TX_NA - Primeiro P(S) com transmissão não autorizada pelo outro sistema;

ULT_PR_RX - Último P(R) (número de sequência do pacote esperado pelo sistema remoto) recebido e já processado na recepção de pacotes;

LOCAL_OCUP - Indica se o sistema local está ou não ocupado para recepção de pacotes;

REM_OCUP - Indica se o sistema remoto está ou não ocupado para recepção de pacotes;

TT_ATIVO - Indica se o temporizador de transmissão encontra-se ou não ativado;

TR_ATIVO - Indica se o temporizador de recepção encontra-se ou não ativado;

JT - Janela de Transmissão;

JR - Janela de Recepção;

NUM_BUF_LIVRES - Número de "buffers" disponíveis para recepção de pacotes;

LOC_OCUP#DESOCUP - Indica se houve ou não transição do sistema local de ocupado para desocupado ou vice-versa;

IND_EXP_TR - Indica que o temporizador de recepção expirou;

IND_M_RX_ZERO - Indica recepção de um pacote com $M = 0$, ou seja, último pacote de uma mensagem;

IND_JR_QFECH - Indica que a janela de recepção está quase fechada.

Utilizando estas informações este módulo controla a recepção de pacotes dentro da janela de recepção e com número de sequência correto, ativando o temporizador de recepção caso o mesmo esteja desativado.

Além disso ele faz a aceitação dos pacotes transmitidos através do P(R) enviado nos pacotes de dados ou nos pacotes de aceitação (RR ou RNR). Ou seja, retira da fila de espera de aceitação os pacotes lá colocados pelo Módulo de Transmissão de Pacotes ao Nível 2, atualiza a janela de transmissão e desativa o temporizador de transmissão quando não há mais pacotes a serem aceitos para o canal lógico.

Este módulo é responsável, também, pelo disparo do envio de pacotes de aceitação pelo Módulo de Transmissão de Pacotes ao Nível 2, marcando indicações em três casos:

- Expiração do temporizador de recepção _ marca a indicação IND_EXP_TR;
- Recepção do último pacote de uma mensagem lógica _ marca a indicação IND_M_RX_ZERO;
- Janela de recepção quase fechada _ marca a indicação IND_JR_QFECH.

Na recepção de pacotes este módulo utiliza áreas livres e quando não há áreas suficientes para recepção de tantos pacotes quanto os estipulados pela janela de recepção o sistema local é colocado ocupado e a indicação LOC_OCUP\$DESOCUP é marcada para que o Módulo de Transmissão de Pacotes ao Nível 2 envie um pacote RNR indicando ao sistema remoto a condição do sistema local. A transição inversa é realizada pelo Módulo de Recepção de Segmentos quando o mesmo libera essas áreas de recepção.

V.4.3.2 - MÓDULO MONITORA VGI

Este módulo é responsável pela monitoração do estado da VGI do operador. Quando é detectada uma mudança de estado, seja de VGI ok para VGI trancada ou vice-versa, este módulo gera o evento respectivo e o envia, através da Interface de Recepção de Eventos (interface (9)), para a máquina de estados do nível implementada pelo Módulo Responsável pelas Máquinas de Estados.

V.4.3.3 - MÓDULO MONITORA RECURSOS

Este módulo, quando comandado pelo Módulo Responsável pelas Máquinas de Estados, monitora os recursos do outro operador OM. Ele obtém a informação dos recursos do outro operador através da interface (14) com o objeto COORDENADOR BÁSICO.

Caso os recursos do outro OM estejam disponíveis e o deste operador estiverem, também, disponíveis ele coloca os recursos deste OM não disponíveis, e caso os recursos do outro estejam não disponíveis e os deste também, ele coloca os recursos deste OM disponíveis.

Cabe ressaltar que esta monitoração só é disparada em caso de haver falha no enlace do operador (ver condições descritas no item (V.4.1)). Desta maneira, a menos nos casos em que a VGI de ambos os OM estejam trancadas ou ambos os OM estejam fora de serviço, qualquer que seja a configuração de falhas, existe um operador (com VGI ok) com os recursos disponíveis que assume os canais lógicos.

V.4.3.4 - MÓDULO DE RECEPÇÃO DE EVENTOS GERADOS PELO NÍVEL 2

Este módulo aguarda, na interface com o N2 (interface (17)), a chegada de eventos e os encaminha ao Módulo Responsável pelas Máquinas de Estados via a Interface de Recepção de Eventos (interface 10)).

Estes eventos são as respostas do N2 (ANS_N2) e as indicações do N2 (IND_N2) descritas nos itens (V.3.5) e (V.3.6) respectivamente.

V.4.3.5 - MÓDULO DE TRANSMISSÃO DE PACOTES AO NÍVEL 2

Este módulo é responsável pela transmissão de pacotes ao N2. Através da interface N3 <=> N2 (interface (4)) ele obtém uma área para transmissão e envia um pacote ao N2. Este mecanismo encontra-se descrito no item (V.3.2) sob o título Requisições do Nível 3. A interface (1), interface de TX N4 -> N3, já foi descrita no item (V.2.1).

A escolha do pacote a ser transmitido segue uma política de prioridades de acordo com o canal lógico ao qual o pacote pertence. Os canais lógicos são configurados do de mais alta para o de mais baixa prioridade. No entanto, a transmissão de pacotes não pode considerar apenas a prioridade do canal lógico, pois a constante existência de pacotes de um canal de alta prioridade levaria os canais de baixa prioridade à espera indefinida ("starvation"). Este módulo implementa, então, um mecanismo de transmissão em ciclos. Para cada canal lógico configura-se o número de pacotes a serem transmitidos em um ciclo. Este número deve ser maior para os canais de alta prioridade e menor para os de baixa. Preferencialmente são transmitidos os pacotes do canal de mais alta prioridade e a cada pacote transmitido é incrementado um contador (por canal lógico). Quando este contador atingir o número máximo de

pacotes a serem transmitidos em um ciclo, a transmissão de pacotes deste canal fica temporariamente bloqueada. Isso se repete sucessivamente com todos os canais lógicos. Quando os contadores de todos os canais lógicos atingirem o seu máximo, ou se aqueles que não tenham atingido não possuírem pacotes a transmitir ou estiverem bloqueados por outra razão qualquer, atinge-se o fim de um ciclo e os contadores de todos os canais são reiniciados com zero.

Para que um pacote de um canal lógico seja transmitido é necessário que o canal lógico:

- seja escolhido pelo mecanismo de transmissão em ciclos,
- esteja com a transmissão de pacotes habilitada,
- esteja desocupado no sistema remoto,
- esteja com a janela de transmissão aberta para o pacote a ser transmitido.

Quando um pacote é transmitido o Temporizador de Transmissão do canal lógico correspondente é ativado, caso esteja desativo, e o pacote é colocado numa fila de espera pela aceitação do mesmo pelo sistema remoto.

Na transmissão de um pacote é feita a aceitação implícita de pacotes recebidos (mecanismo de "piggy-back"), logo são desmarcadas as indicações IND_EXP_TR, IND_M_RX_ZERO e IND_JR_QFECH evitando, assim, o envio de pacotes de aceitação.

Através da interface (8), este módulo pode enviar para a máquina de estados de algum canal lógico um evento que implique na reiniciação do mesmo. Isso é feito quando ocorre a expiração do Temporizador de Transmissão, ou seja, existe algum ou alguns pacotes enviados que não foram aceitos dentro do tempo estipulado por este temporizador.

Este módulo é responsável, também, pela transmissão de pacotes de aceitação. Estes pacotes, qualquer que seja o canal lógico correspondente, possuem prioridade de transmissão maior que qualquer pacote de dados, logo são transmitidos independentemente do mecanismo de ciclos. Um pacote de aceitação de um canal lógico é enviado se alguma das indicações que constam do Bloco de Controle de Canal Lógico estiver marcada. Estas indicações são: LOC_OCUP\$DESOCUP, IND_EXP_TR, IND_M_RX_ZERO e IND_JR_QFECH; descritas no item (V.4.3.1). De acordo com a condição do sistema local, ocupado ou desocupado, é enviado um pacote RNR ou RR respectivamente.

Mediante a aceitação de pacotes recebidos, seja ela implícita ou explícita, este módulo atualiza a janela de recepção.

V.4.3.6 - MÓDULO DE RECEPÇÃO DE SEGMENTOS

Este módulo é responsável pela recepção de segmentos de uma mensagem lógica e transferência desses segmentos para o N4 via a interface de recepção N3 -> N4 (interface (2)).

O Módulo Responsável pelas Máquinas de Estados, que também controla a recepção de pacotes, ao receber o último pacote de uma mensagem transfere todos os pacotes dessa mensagem, transformando-os em segmentos, ao Módulo de Recepção de Segmentos.

O mecanismo de transferência dos segmentos ao N4 é análogo ao de transmissão de pacotes, ou seja, é feito de acordo com a prioridade do canal lógico e o mecanismo de ciclos mediante a contagem dos segmentos transferidos até um número máximo definido na configuração para cada canal lógico.

Para que um segmento seja transferido ao N4 é necessário que o canal lógico:

- seja escolhido no mecanismo de transferência em ciclos,
- esteja desbloqueado na interface com o N4.

É este módulo que libera as áreas de recepção e, portanto, ele é responsável pela transição do sistema local da condição de ocupado para desocupado quando o número dessas áreas torna-se superior ao número de pacotes permitidos pela janela de recepção deste canal lógico. Ele marca, portanto, a indicação LOC_OCUP\$DESOCUP e para isso utiliza o BCL do canal lógico em questão.

CAPÍTULO VI

IMPLEMENTANDO O PROTOCOLO DE ALTO NÍVEL

VI.1 - INTRODUÇÃO

Este capítulo engloba a descrição da implementação dos três níveis superiores do protocolo, ou seja, dos níveis que correspondem ao protocolo de alto nível, níveis estes também chamados de camadas da aplicação.

Estes níveis implementam os seus mecanismos independentemente para cada tipo de dado especificado. Logo, para cada tipo de dado, estes três níveis constituem dois objetos: o COORD_XXX, coordenador da aplicação, e o NHS_XXX, objeto de armazenamento de informações para o nível hierárquico superior, onde XXX é substituído pelo mnemônico do tipo de dado correspondente (INIC, CTL, DIG, ANL, TOT, SOE, CRT, EST ou DSP).

A figura (VI.1) apresenta a organização desses níveis com as interfaces entre eles e as externas a eles para um tipo de dado genérico.

Os procedimentos destes três níveis para um determinado tipo de dado só são realizados quando o objeto COORDENADOR DA APLICAÇÃO correspondente estiver ativo em um dos operadores OM. Como já foi dito no item (IV.3.7) do capítulo (IV), o mecanismo

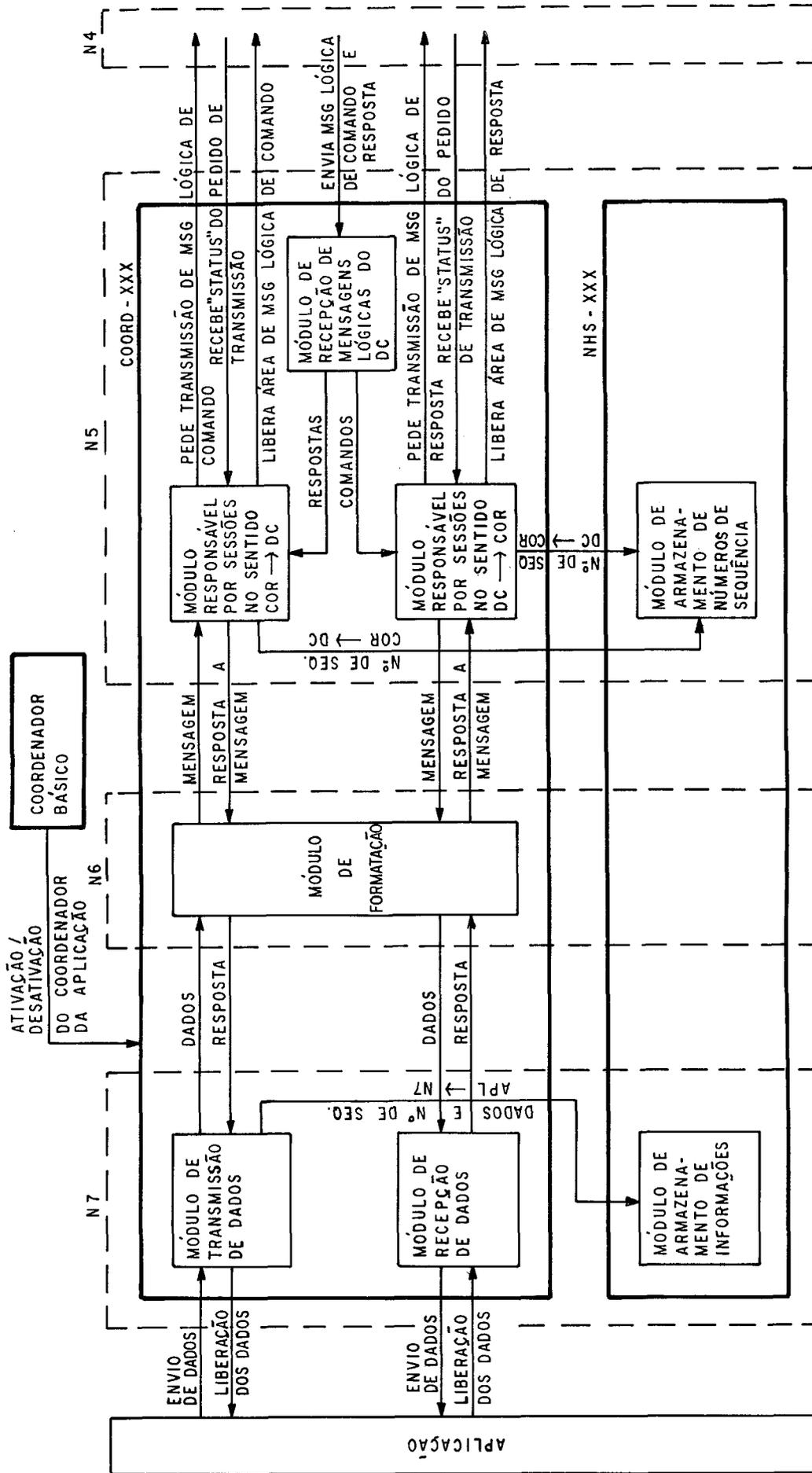


FIGURA VI.1 - ORGANIZAÇÃO DOS NÍVEIS 5,6 e 7.

de ativação/desativação dos tipos de dados nos OM está intimamente ligado ao mecanismo de alocação/desalocação dos canais lógicos do nível 3. Logo, quando o COORDENADOR BASICO envia uma mensagem ao nível 3 indicando alocação/desalocação de canal lógico, ele envia, também, ao objeto COORDENADOR DA APLICAÇÃO do tipo de dado correspondente, uma mensagem de ativação/desativação respectivamente.

O objeto COORDENADOR DA APLICAÇÃO de um tipo de dado está ativo em um determinado instante em apenas um dos operadores OM, logo os procedimentos dos níveis 5, 6 e 7 executados pelos módulos pertencentes a esse objeto só são realizados no operador OM no qual o objeto coordenador encontra-se ativo.

Já o objeto de armazenamento de informações para o nível hierárquico superior, ou seja, o objeto NHS de um tipo de dado encontra-se ativo sempre nos dois operadores OM. Toda informação enviada pelo COORDENADOR DA APLICAÇÃO ao objeto NHS é recebida por ambas as suas cópias. Isto é necessário para que os mecanismos acima referenciados (alocação/desalocação e ativação/desativação) possam ser realizados sem que ocorra perda de dados e de continuidade na comunicação COR \Leftrightarrow DC. Ou seja, quando o COORDENADOR DA APLICAÇÃO é desativado em um OM e ativado no outro, este último encontra-se de posse (no seu objeto NHS) dos dados ainda não enviados e dos números de seqüência da próxima mensagem lógica a ser enviada/recebida para/do DC e dos próximos dados a serem enviados/recebidos para/dos processos de aplicação. Além disso antes da reativação de um COORDENADOR DA APLICAÇÃO no seu OM de origem é feita a consistência dessas informações para o objeto NHS desse

operador. Isto é, se o COORDENADOR DA APLICAÇÃO foi desativado porque o OM ou a VGI saiu de serviço, uma vez a falha restabelecida, este coordenador só volta a ser reativado neste OM quando o objeto NHS correspondente possuir todos os dados que estão armazenados no outro OM e os números de seqüência atuais. Esta consistência é feita pelo objeto de iniciação, IOM.

Nos itens (VI.2) e (VI.3) são descritos os níveis 5 e 7, respectivamente, para um tipo de dado genérico. O nível 6 realiza apenas a formatação das mensagens, formatos estes já apresentados no capítulo (IV) item (IV.3.6) e no apêndice (A). Cabe apenas ressaltar que uma mensagem lógica recebida com formato geral errado não é repassada ao nível 7, o próprio nível 6 encaminha a resposta apropriada ao nível 5 para transmissão.

VI.2 - IMPLEMENTAÇÃO DO NÍVEL 5

O nível 5 é responsável pela execução de sessões do tipo comando/resposta. É permitida a execução de uma sessão no sentido COR -> DC e outra no sentido DC -> COR simultaneamente.

Este nível, como foi explicado no capítulo (IV) item (IV.3.5), é responsável pelo controle fim-a-fim na transmissão/recepção das mensagens lógicas, ou seja, ele garante a não ocorrência de perda de dados na troca de mensagens. Para isso ele implementa mecanismos de retransmissão e de sequencialização de mensagens lógicas.

Este nível encontra-se organizado em quatro módulos:

- Módulo Responsável por Sessões no Sentido COR -> DC,
- Módulo Responsável por Sessões no Sentido DC -> COR,
- Módulo de Recepção de Mensagens Lógicas do DC,
- Módulo de Armazenamento de Números de Sequência.

VI.2.1 - MÓDULO RESPONSÁVEL POR SESSÕES NO SENTIDO COR -> DC

Este módulo recebe uma mensagem do N6, acrescenta a ela um número de sequência, encaminha-a ao N4 para que seja transmitida ao DC, dispara o temporizador de transmissão (TTM) e fica aguardando uma resposta do DC via a interface com o Módulo de Recepção de Mensagens Lógicas do DC.

O número de sequência de uma mensagem lógica transmitida deve ser repetido pelo sistema remoto na resposta por ele enviada. Essa resposta, chamada ACK_XXX, possui um campo denominado CAUSA o qual é analisado pelo N5 segundo três critérios: SUCESSO (1), NÚMERO DE SEQUÊNCIA ERRADO (2) ou qualquer outra causa diferente (0, 3, 4, 5, 6 ou 7) (ver descrição deste campo no apêndice (A)). A codificação dessas outras causas importa apenas para o nível 6 (apenas FORMATO GERAL ERRADO) e para o nível 7, Nível de Aplicação.

O procedimento normal deste módulo é, portanto, transmitir uma mensagem lógica e receber uma resposta com causa igual a SUCESSO. Neste caso o número de sequência para sessões no sentido COR -> DC é incrementado via a interface com o Módulo de

Armazenamento de Números de Sequência e a resposta é repassada aos níveis superiores para que a mensagem transmitida seja liberada.

Cabe ressaltar que o Módulo de Armazenamento de Números de Sequência pertence ao objeto NHS_XXX. Este objeto, como já foi dito, está sempre ativo nos dois operadores OM e os números de sequência de sessões no sentido CDR -> DC são armazenados em ambos. O Módulo Responsável por Sessões no Sentido CDR -> DC, ao transmitir uma mensagem lógica, obtém o número de sequência para ela através de acesso direto ao NHS_XXX deste operador.

No entanto, várias situações anormais podem ocorrer e os procedimentos adotados variam de acordo com o caso.

No pedido de transmissão de mensagem lógica ao N4 o mesmo pode retornar um "status" indicando *impossibilidade de transmissão*. O procedimento adotado pelo N5 é tentar transmitir indefinidamente, pois, ou os níveis inferiores estão em iniciação e quando a mesma terminar vão liberar a transmissão, ou ocorreu algum problema nos níveis inferiores e o COORDENADOR DA APLICAÇÃO vai acabar sendo desativado pelo COORDENADOR BÁSICO e os seus procedimentos interrompidos. Neste último caso a mensagem lógica em questão vai acabar sendo transmitida pelo coordenador da aplicação do outro OM, uma vez que o mesmo será ativado.

Caso a mensagem seja aceita para transmissão pelo N4 e a resposta não chegue dentro do tempo estipulado pelo temporizador de transmissão (TTM), este módulo retransmite a mensagem e incrementa um contador de mensagens transmitidas (CTTM). Este

procedimento pode ser repetido até um número máximo de vezes (NTTM); este número é um parâmetro configurável para cada tipo de dado. A ocorrência de "time-outs" sucessivos pode se dar devido a algum problema na comunicação, mas nesse caso os níveis inferiores vão acabar detectando o problema e o coordenador será desativado. Ou então, devido a algum erro na mensagem lógica, por exemplo, um erro de formato geral que impeça o DC de reconhecer a mensagem e o leve a descartá-la. Neste caso, quando o número de tentativas (CTTM) atingir NTTM, este módulo repassa aos níveis superiores uma indicação para liberar a mensagem e zera o número de seqüência do sentido COR -> DC via a interface com o Módulo de Armazenamento de Números de Seqüência.

Uma mensagem lógica com número de seqüência zero é sempre aceita pelo sistema remoto e leva o mesmo a reiniciar o seu número de seqüência de recepção.

Ao receber uma resposta este módulo verifica se a mesma se refere à última mensagem lógica transmitida através do seu número de seqüência. Se este número for o esperado menos um, esta resposta é descartada e o módulo permanece à espera da resposta correta. Isto porque esta resposta pode ser um ACK_XXX atrasado da mensagem anterior, ou seja, uma resposta repetida pelo sistema remoto devido a retransmissão precipitada, por parte do sistema local, de uma mensagem lógica que o remoto já havia respondido e que o sistema local teria em seguida aceito. Isto ocorre caso o TTM local esteja mal dimensionado em relação ao tempo gasto pelo sistema remoto para responder a uma mensagem. Se o número de seqüência for completamente errado, a

resposta é ignorada e o procedimento adotado é o mesmo do caso de "time-out".

Mediante a recepção de uma resposta com o número de seqüência correto, é analisada a causa enviada. Um ACK_XXX com causa SUCESSO configura o procedimento normal já descrito. A causa NÚMERO DE SEQUÊNCIA ERRADO pode configurar um erro de protocolo. Esta situação é recuperada zerando-se o número de seqüência e retransmitindo-se a mensagem lógica. Como já foi dito, toda mensagem com número de seqüência zero é aceita pelo sistema remoto e o seu número de seqüência reiniciado. Qualquer outra causa de erro recebida não pode ser recuperada pelo N5, pois não se encontra dentro do seu escopo de conhecimento do protocolo de aplicação. Logo, neste caso a resposta é repassada para os níveis superiores e o número de seqüência é zerado por precaução.

Finalmente, este módulo libera a área do N4 na qual ele recebeu a resposta a uma mensagem lógica.

VI.2.2 - MÓDULO RESPONSÁVEL POR SESSÕES NO SENTIDO DC -> COR

O procedimento normal deste módulo é receber uma mensagem lógica com o número de seqüência correto do N4, via a interface com o Módulo de Recepção de Mensagens Lógicas do DC, repassar esta mensagem aos níveis superiores e receber deles uma resposta a ser enviada com causa igual a SUCESSO. Essa resposta é, então, transmitida ao N4 e o número de seqüência de sessões no

sentido DC -> COR é incrementado via a interface com o Módulo de Armazenamento de Números de Sequência.

Caso a mensagem recebida possua um número de sequência igual ao da anterior, esta mensagem é descartada pelo NS, o número de sequência de sessões no sentido DC -> COR é mantido e o ACK da mensagem anterior é retransmitido.

A recepção de uma mensagem com o número de sequência completamente errado leva ao descarte dessa mensagem e à transmissão de uma resposta com causa NÚMERO DE SEQUENCIA ERRADO. O número de sequência é mantido inalterado.

Uma mensagem com número de sequência correto é repassada aos níveis superiores. Pode ocorrer, no entanto, que a resposta enviada pela aplicação para ser transmitida possua uma causa diferente de SUCESSO. Neste caso a resposta é transmitida, mas o número de sequência é mantido inalterado. Isto é feito porque caso o sistema remoto tenha condições de recuperar a mensagem, ele a retransmitirá com o mesmo número de sequência e a mesma será aceita por este módulo. Caso ele não tenha condições de fazer esta recuperação ele pode descartar a mensagem e enviar a próxima com o número de sequência zerado, a qual também será aceita. É desta segunda forma que este sistema local age nessas condições, como foi descrito no item anterior.

Cabe ressaltar que a recepção de uma mensagem com número de sequência igual a zero é aceita, ou seja, é repassada aos níveis superiores. Se a resposta enviada pela aplicação para ser transmitida possuir causa igual a SUCESSO o número de sequência é zerado e incrementado, isto é, passa a ser um (1). Se a

resposta tiver outra causa diferente o número de seqüência é simplesmente zerado.

Finalmente, este módulo libera a área do N4 na qual ele recebeu a mensagem lógica.

VI.2.3 - MÓDULO DE RECEPÇÃO DE MENSAGENS LÓGICAS DO DC

Este módulo é responsável pela interface de recepção de mensagens lógicas do N4. Ao receber uma mensagem, ele analisa a mesma e caso ela seja uma mensagem lógica propriamente dita, isto é, seja um comando de uma sessão, ele a encaminha ao Módulo Responsável por Sessões no Sentido DC -> COR; caso ela seja uma resposta encaminha-a ao Módulo Responsável por Sessões no sentido COR -> DC.

VI.2.4 - MÓDULO DE ARMAZENAMENTO DE NÚMEROS DE SEQUÊNCIA

Este módulo, pertencente ao objeto NHS ativo em ambos os OM, é responsável pela manutenção dos números de seqüência de sessões no sentido COR -> DC e DC -> COR.

Excluindo-se o tipo de dado de iniciação, para o qual não existem sessões no sentido COR -> DC, os demais requerem um mecanismo que evite perda de dados e de continuidade na troca de mensagens neste sentido. Daí os números de seqüência sentido COR -> DC serem armazenados em ambos os NHS_XXX.

Já no sentido DC -> COR isto só é feito para os tipos de dados referentes às Mensagens Operacionais. Para os demais o número de seqüência sentido DC -> COR é armazenado apenas no NHS_XXX do operador no qual o coordenador correspondente esteja ativo. No NHS_XXX do outro OM o número de seqüência é mantido zerado e caso o coordenador migre para esse operador, ele aceitará o primeiro número de seqüência enviado pelo sistema remoto e atualizará o seu número de seqüência. Isto porque as sessões neste sentido para os tipos de dados de INIC, DIG, ANL, TOT e SOE não justificam uma comunicação via VGI com o outro NHS_XXX, uma vez que não portam dados importantes para o sistema aplicativo. Para o tipo de dado de controle este mecanismo de seqüencialização não basta, pois é necessário, também, garantir que um mesmo comando não seja repetido, logo optou-se por colocar NTTM deste tipo igual a um (1) e não manter o número de seqüência nos dois NHS_CTL.

Este mecanismo e suas implicações no procedimento de ativação e desativação dos coordenadores é melhor explicado no item (IV.4) que se refere à Especificação Funcional do Protocolo de Comunicação.

VI.3 - IMPLEMENTAÇÃO DO NÍVEL 7

Este nível, o mais alto na hierarquia das camadas do protocolo, é responsável pela interface com os processos da aplicação os quais, como já foi dito, encontram-se localizados em outros operadores do sistema distribuído.

Os procedimentos deste nível, como ficou claro na introdução deste capítulo, fazem parte de dois objetos: COORD_XXX e NHS_XXX.

O objeto COORD_XXX, no que diz respeito a este nível, é composto por dois módulos: Módulo de Transmissão de Dados e Módulo de Recepção de Dados. O Módulo de Transmissão de Dados é responsável pela recepção dos dados da aplicação, envio dos mesmos para transmissão e espera da resposta enviada pelo DC. O Módulo de Recepção de Dados, analogamente, é responsável pela recepção de dados do DC, através da interface com o N6, envio desses dados, quando for o caso, para os processos de aplicação do COR e transmissão da resposta.

O objeto NHS_XXX é composto pelo Módulo de Armazenamento de Informações ativo nos dois OM. Ele é responsável pelo armazenamento de dados e/ou armazenamento de números de seqüência da ligação PROCESSOS DA APLICAÇÃO <=> NÍVEL 7, uma vez que a troca de dados entre estas duas entidades se dá através da VGI devido à localização física das mesmas. Este objeto é, também, responsável pela manutenção do Banco de Dados do operador.

É importante ressaltar a necessidade do mecanismo de seqüencialização controlado pelo N7. Pode parecer estranho, uma vez que o N5 também implementa seqüenciamento das mensagens lógicas. O seqüenciamento do N5 se refere à ligação COR <=> DC; o do N7 é necessário uma vez que os processos aplicativos encontram-se localizados em outros operadores, logo este seqüenciamento se refere à ligação PROCESSOS DA APLICAÇÃO <=>

PROTOCOLO DE COMUNICAÇÃO para o qual o N7 é a porta de entrada. A importância desses mecanismos de seqüencialização para que seja mantida a consistência dos dados trocados entre os dois sistemas fica mais clara no item (VI.4).

Para cada tipo de dado os procedimentos dos objetos correspondentes diferem de acordo com as características inerentes a cada um deles. Em vista disso, esses procedimentos serão descritos em quatro itens. O item (VI.3.1) apresenta os procedimentos para o tipo de dado de *iniciação* e o item (VI.3.2) para os dados referentes ao *controle* do Sistema Elétrico. O item (VI.3.3) descreve os procedimentos para os dados adquiridos, ou seja, para quatro tipos de dados a saber: *estados digitais*, *medidas analógicas*, *totalizadores* e *seqüência de eventos*. Finalmente, o item (VI.3.4) se refere às mensagens operacionais, ou seja, mensagens de "*crt*", *estruturadas* e de "*display*". O significado desses dados, bem como a origem dos mesmos dentro do sistema aplicativo foi descrito no capítulo (III). A figura (IV.3) do capítulo (IV) apresenta um resumo das mensagens trocadas para cada tipo de dado.

VI.3.1 - PROCEDIMENTOS PARA OS DADOS DE INICIAÇÃO

Para que a troca de informações entre os dois sistemas, DC e COR, se inicie é necessária a execução do procedimento de iniciação. Logo, primeiramente, somente este tipo de dado encontra-se liberado para execução de sessões de troca de informações. Este procedimento é realizado na iniciação do

sistema ou numa reiniciação caso a ligação DC <=> COR tenha ficado fora de operação durante algum tempo, ou devido a falha no DC e/ou no COR (nos dois OM) e/ou nos canais de comunicação.

A iniciação começa com o envio pelo DC de uma mensagem de REQ_INC na qual é enviada a versão do banco de dados e a versão do protocolo de comunicação vigente.

Caso alguma das versões, ou as duas, não coincidam com as versões mantidas no banco de dados do objeto NHS_INIC, o N7 (Módulo de Recepção de Dados) envia para transmissão uma resposta com o campo causa apropriado (VERSÃO DE BD ERRADA ou VERSÃO DE PC ERRADA). Neste caso os outros tipos de dados permanecerão impedidos de trocarem informações e este fato é avisado através da exteriorização, pelo operador OM, de uma determinada configuração dos "leds" existentes no seu painel frontal.

Estando ambas versões corretas, após a reiniciação interna do operador OM, a mensagem de iniciação é enviada a cada objeto responsável pela manutenção dos dados de uma estação do sistema elétrico. Estes objetos denominados SERi, Objeto do Sistema Elétrico Regional, onde i é o número da estação correspondente, encontram-se localizados nos operadores OR. Este procedimento vai disparar o envio de dados de iniciação sequencial digital, analógica e totalizada de todas as estações do sistema elétrico, ao respectivo objeto coordenador. O N7 envia, então, uma resposta ACK_INIC com causa igual a SUCESSO para transmissão ao DC.

Para este tipo de dado existe apenas o Módulo de Recepção de Dados, uma vez que não há troca de mensagens no sentido COR -> DC.

VI.3.2 - PROCEDIMENTOS PARA DADOS DE CONTROLE ELÉTRICO

A realização de um controle pelo DC em um equipamento do sistema elétrico se dá através de duas sessões, uma no sentido DC -> COR e a outra no sentido oposto.

O DC envia o comando de controle através da mensagem de EXEC_CTRL, que é repassada pelo N7, Módulo de Recepção de Dados, ao objeto CTL da estação a qual o comando se refere (este objeto também encontra-se localizado em um operador OR). Mediante a confirmação de recepção do comando por parte do CTL, o N7 envia uma resposta ACK_CTRL com causa SUCESSO para transmissão. Caso não obtenha sucesso na comunicação com o objeto CTL é enviado um ACK_CTRL com causa ESTAÇÃO FORA DE SERVIÇO.

Uma vez terminado o controle, o objeto CTL envia ao N7 um resultado. Este dado é recebido pelo Módulo de Transmissão de Dados, pertencente ao COORD_CTL ativo, e enviado para armazenamento em ambos os NHS_CTL (Módulo de Armazenamento de Informações). Logo após essa inserção dos dados nos dois operadores, o objeto CTL é liberado e pode descartar o resultado do controle. Existe, portanto, uma fila de mensagens RESLT_CTRL para todas as estações do sistema elétrico.

O Módulo de Transmissão de Dados é responsável, também, pela transmissão de cada mensagem desta fila e pela recepção da resposta. Essa resposta ou é um ACK_CTRL com causa SUCESSO ou é uma resposta cuja causa já foi tratada pelos níveis inferiores.

VI.3.3 - PROCEDIMENTOS PARA OS DADOS ADQUIRIDOS

Os dados adquiridos incluem os dados digitais, analógicos, totalizados e sequência de eventos. Os objetos SER referentes às estações do sistema elétrico enviam esses dados para o Módulo de Transmissão de Dados respectivo e os mesmos são armazenados pelo Módulo de Armazenamento de Informações em ambos os objetos NHS correspondentes. Após o armazenamento das informações o objeto SER que as enviou é liberado.

Da mesma forma que ocorre com os resultados de controle, existe uma fila de mensagens para cada tipo de dado e o Módulo de Transmissão de Dados se encarrega, também, de enviar um a um desses dados para transmissão e aguardar a resposta.

A resposta recebida é um ACK_XXX (onde XXX pode ser DIG, ANL, TOT ou SOE) com causa SUCESSO ou {NDICE DE PONTO ERRADO. Neste segundo caso deve ser exteriorizado um alarme indicando a existência de alguma inconsistência nos bancos de dados dos dois sistemas.

No sentido DC -> COR é trocada uma mensagem REQ_XXX cujo objetivo é verificar se o tipo de dado encontra-se em perfeitas condições de funcionamento. O DC envia este comando caso ele

permaneça um determinado tempo sem receber nenhuma mensagem do COR para um determinado tipo de dado. O Módulo de Recepção de Dados responde a esse comando com um ACK_XXX com causa SUCESSO.

VI.3.4. - PROCEDIMENTOS PARA AS MENSAGENS OPERACIONAIS

As mensagens operacionais incluem as mensagens de "crt", as mensagens estruturadas e as mensagens de "display".

O procedimento de envio de dados do COR para o DC para estes tipos de dados difere dos demais, uma vez que os objetos da aplicação (MSG, Objeto de Mensagens, e SPR, Objeto "Spooler"), localizados em um operador DC, ficam presos até que os dados sejam enviados e a resposta recebida. Para os demais tipos de dados isto não ocorre porque a informação armazenada é pequena e o objeto da aplicação que a envia tem que estar dinamicamente monitorando mudanças, pois as mesmas ocorrem frequentemente. Logo, não é concebível que este objeto seja mantido preso até que as informações atinjam seu destino. As mensagens operacionais, no entanto, são grandes, principalmente os relatórios e as telas, e o armazenamento de várias delas para cada tipo de dado seria inviável, devido a problemas de memória. Por outro lado, também não é comum que um despachante fique enviando várias mensagens sem ter recebido a resposta à primeira delas. Mediante o exposto acima o procedimento de troca de mensagens sentido COR -> DC ocorre da seguinte forma: o objeto da aplicação envia a mensagem ao Módulo de Transmissão de Dados o qual transmite os mesmos aos níveis inferiores e fica

aguardando a resposta. Com a recepção do ACK com causa SUCESSO o processo da aplicação é liberado.

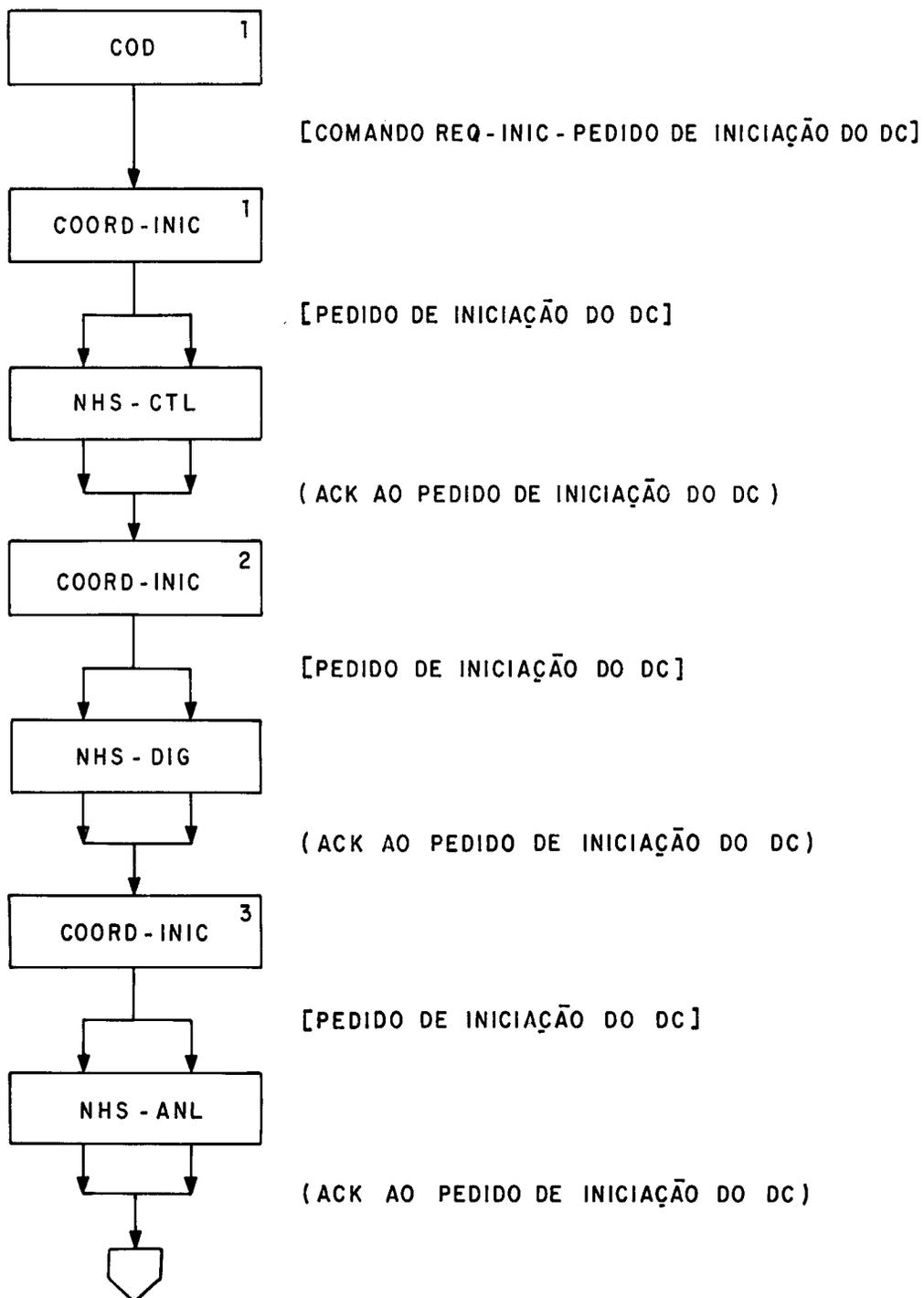
No sentido DC -> COR o Módulo de Recepção de Dados, ao receber uma mensagem, realiza duas comunicações com o objeto da aplicação CMG, Coordenador de Mensagens, do DC. Na primeira, ele adquire os "status" (ocupada/desocupada) das áreas da aplicação para recepção de mensagens e na segunda ele envia a mensagem. Caso essas duas comunicações sejam realizadas com sucesso, este módulo envia um ACK com causa SUCESSO contendo os "status" das áreas, caso contrário ele envia um ACK com causa POSTO FORA DE SERVIÇO.

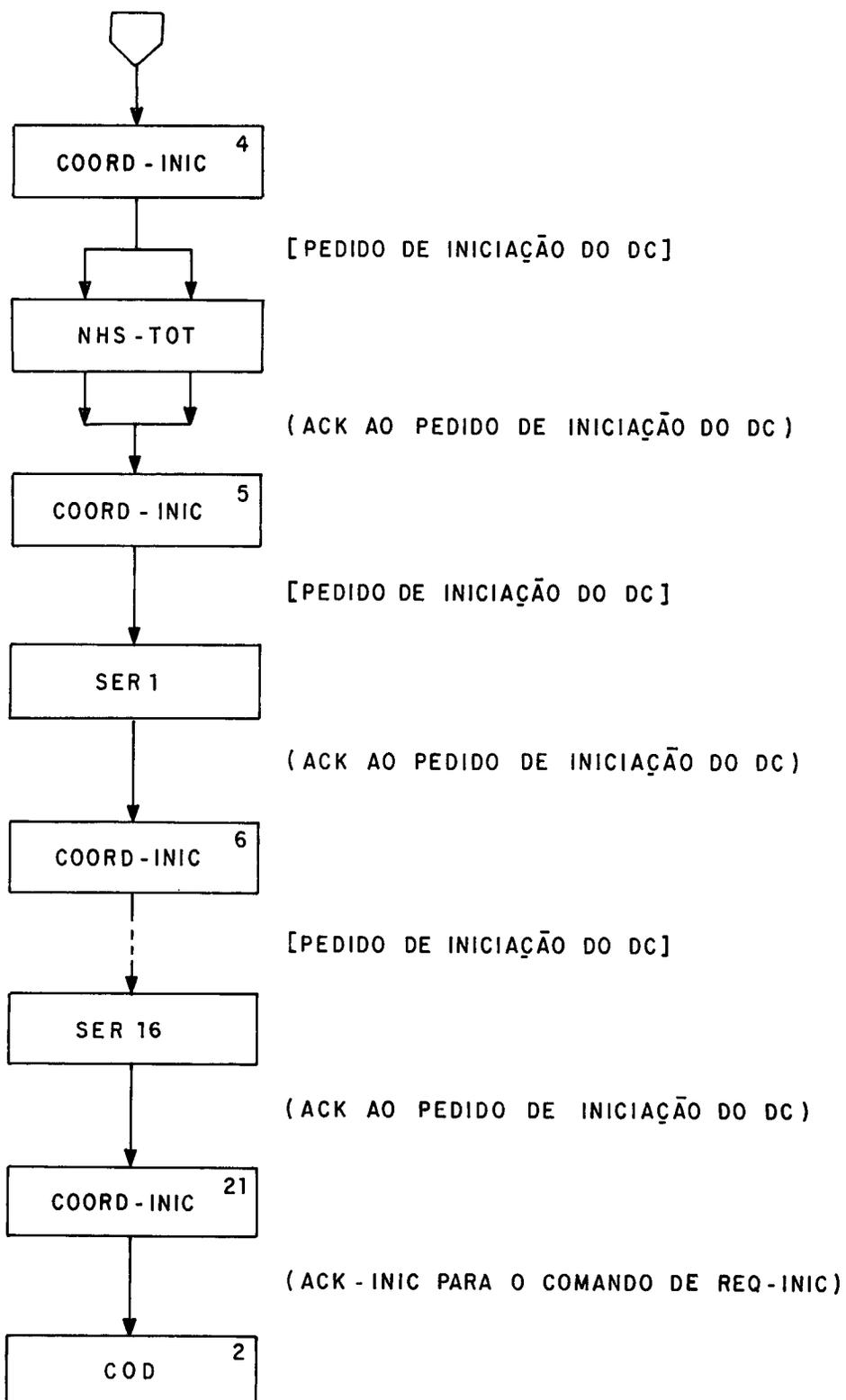
VI.4 - ESPECIFICAÇÃO FUNCIONAL DO PROTOCOLO DE COMUNICAÇÃO

COR <=> DC

Este item apresenta a Especificação Funcional, segunda fase da metodologia utilizada (ver capítulo (II) item (II.3), do SCSC (Sistema de Comunicação com o Sistema Central)).

As informações contidas nas descrições dos DSA são, por vezes, redundantes em relação às apresentadas nas descrições dos níveis 5 e 7. A intenção deste item é apresentar as camadas da aplicação e de comunicação como um todo, enfatizando a relação entre os objetos que compõem todo o subsistema. Os procedimentos descritos com mais detalhes são os referentes ao protocolo de alto nível, explicando como esse protocolo se recupera mediante a ocorrência de certas falhas.





DSA VI.1 - RECEPÇÃO DE COMANDO DE INICIAÇÃO DO DC E ENVIO DA RESPOSTA.

RECEPÇÃO DE COMANDO DE INICIAÇÃO DO DC E ENVIO DA RESPOSTA

COD_1

Entrada: -

Saída: [Comando REQ_INIC - pedido de iniciação do DC].

Efeito: Ao receber um comando REQ_INIC, COD encaminha esse comando ao COORD_INIC.

COORD_INIC 1, 2, 3, 4

Entrada: [Comando REQ_INIC - pedido de iniciação do DC] para o COORD_INIC 1 e (Ack ao pedido de iniciação do DC) para os demais.

Saída: [Pedido de iniciação do DC].

Efeito: Ao receber um pedido de iniciação do DC, COORD_INIC abre, sequencialmente, sessões com os objetos NHS_CTL, NHS_DIG, NHS_ANL e NHS_TOT para que os mesmos possam se reiniciar.

NHS_CTL, NHS DIG, NHS ANL

Entrada: [Pedido de iniciação do DC].

Saída: (Ack ao pedido de iniciação do DC).

Efeito: Ao receberem um pedido de iniciação do DC estes objetos descartam todas as mensagens que estão armazenadas para transmissão.

NHS_TOT

Entrada: [Pedido de iniciação do DC].

Saída: (Ack ao pedido de iniciação do DC).

Efeito: NHS_TOT descarta todas as mensagens, exceto as de integridade que devem ser enviadas ao DC como relatório horário dos valores totalizados.

COORD INIC 5 a 20

Entrada: (Ack ao pedido de iniciação do DC).

Saída: [Pedido de iniciação do DC].

Efeito: Após a iniciação feita nos objetos do SCSC, COORD_INIC envia, sequencialmente, a todos os objetos SER do SCER, um pedido de iniciação. Este pedido vai disparar o envio de todos os dados digitais, analógicos e totalizados pelo SCER ao respectivo objeto coordenador do SCSC.

COORD INIC 21

Entrada: (Ack ao pedido de iniciação do DC).

Saída: [ACK_INIC para o comando de REQ_INIC].

Efeito: COORD_INIC envia ao COD um ACK_INIC para ser transmitido ao DC.

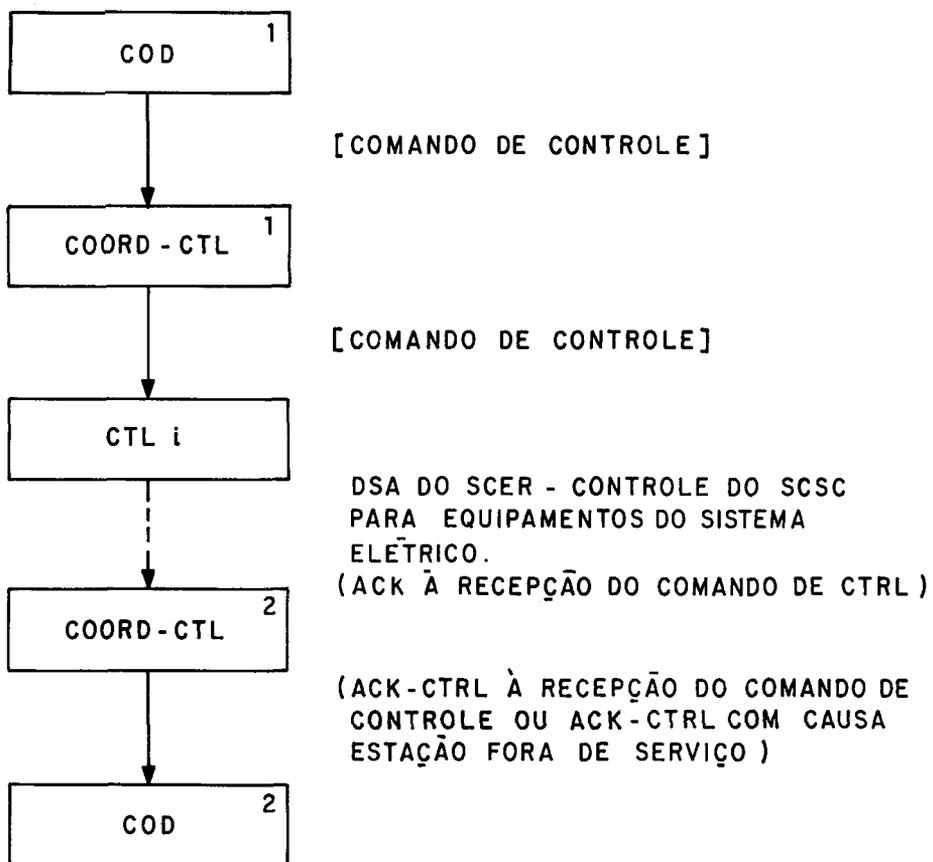
Cabe ressaltar que os procedimentos descritos acima se referem à recepção de um comando REQ_INIC correto, logo o ACK_INIC enviado neste caso indica causa igual a SUCESSO. No entanto o comando é analisado e, dependendo do caso, o ACK_INIC enviado indica a causa apropriada (FORMATO GERAL ERRADO, NÚMERO DE SEQUÊNCIA ERRADO, VERSÃO DE BD ERRADA ou VERSÃO DE FC ERRADA). Nestes casos os procedimentos acima descritos não são efetuados.

COD_2

Entrada: (ACK_INIC para o comando de REQ_INIC).

Saída: -

Efeito: Transmite ao DC o ACK_INIC apropriado.



DSA VI.2 - RECEPÇÃO DE COMANDO DE CONTROLE DO DC E ENVIO DA RESPOSTA.

RECEPÇÃO DE COMANDO DE CONTROLE DO DC E ENVIO DA RESPOSTA

COD 1

Entrada: -

Saída: [Comando de Controle].

Efeito: Ao receber do DC um comando de controle, o COD encaminha esse comando ao COORD_CTL.

COORD_CTL 1

Entrada: [Comando de Controle].

Saída: [Comando de Controle].

Efeito: De acordo com a estação à qual o comando de controle está endereçado, COORD_CTL envia-o ao objeto CTL respectivo e fica aguardando a resposta à recepção do comando.

COORD_CTL 2

Entrada: (Ack à recepção do comando de controle).

Saída: (ACK_CTRL à recepção do comando de controle ou ACK_CTRL com causa ESTAÇÃO FORA DE SERVIÇO).

Efeito: COORD_CTL recebe de CTLi a resposta indicando sucesso na recepção do comando de controle. É enviado, então, ao COD um ACK_CTRL para ser transmitido ao DC.

Como uma sessão (COMANDO/RESPOSTA do nível 5 do protocolo) tem que ser iniciada e terminada pelo mesmo enlace (caminho lógico estabelecido em um determinado canal físico), não há necessidade desse ACK_CTRL ser enviado aos objetos NHS_CTL.

Caso a comunicação com CTLi não seja realizada com sucesso, COORD_CTL envia ao COD um ACK_CTRL com causa ESTAÇÃO FORA DE SERVIÇO.

O comando de controle, enviado pelo DC, possui um número de sequência associado ao tipo de dado de controle sentido DC -> COR que é repetido na resposta. Esse número de sequência para controle serve apenas como garantia para a associação da resposta ao comando respectivo. Não há necessidade de repassar esse número para o CTLi, uma vez que não existe redundância "on line" nos OR.

Também não há necessidade de manter esse número de sequência nos dois OM, uma vez que o DC nunca repete um comando de controle (NTTM = 1 no DC para o tipo de dado de controle), logo COORD_CTL testa esse número apenas internamente, ou seja, com o número de sequência mantido pelo NHS_CTL deste OM. Comandos de controle não podem ser repetidos (NTTM = 1), pois a saída de serviço do OM ou do canal de comunicação no qual COORD_CTL está ativo, após o envio do comando para o CTLi, não impede que este objeto execute o controle. A reativação do COORD_CTL no outro OM poderia implicar na repetição do controle pelo DC, caso NTTM não tivesse sido atingido, fazendo com que o controle fosse novamente executado pelo CTLi. Essa nova execução de um mesmo controle seria feita indevidamente para pontos do tipo compensador

síncrono, uma vez que o controle dos mesmos é feito através de ações do tipo AUMENTAR ou DIMINUIR.

NTTM é um parâmetro do nível 5 configurável por tipo de dado que significa número de tentativas de retransmissão de uma mensagem lógica devido a estouros sucessivos do temporizador TTM.

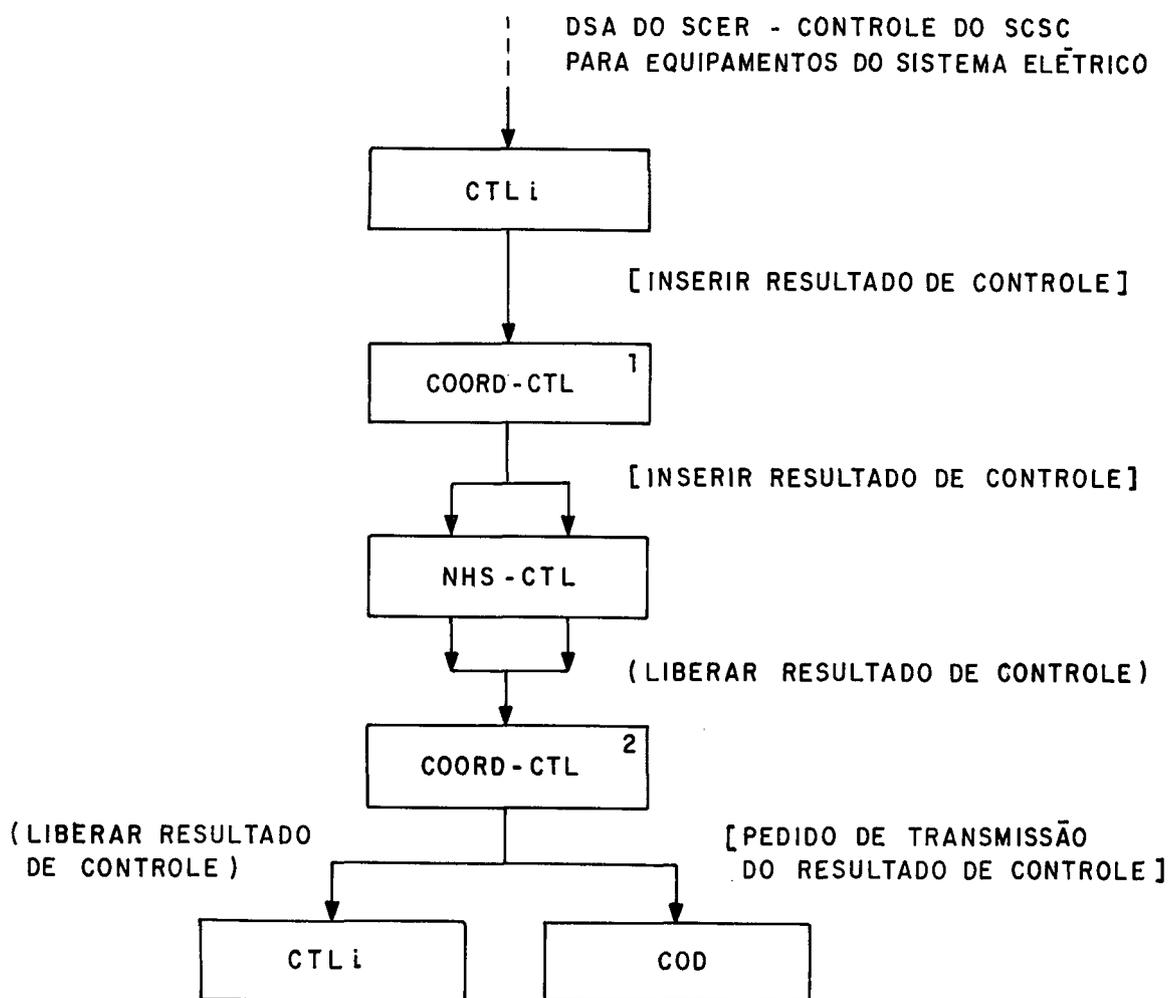
Caso o comando de controle possua número de seqüência ou formato geral errado, o comando não chega a ser enviado ao SCER e a resposta enviada possui causa apropriada.

COD_2

Entrada: (ACK_CTRL à recepção do comando de controle ou ACK_CTRL com causa ESTAÇÃO FORA DE SERVIÇO).

Saída: -

Efeito: COD transmite ao DC a resposta à recepção do comando de controle.



DSA VI.3 - RECEPÇÃO DO RESULTADO DE CONTROLE DO SCER.

RECEPÇÃO DO RESULTADO DE CONTROLE DO SCER

COORD_CTL 1

Entrada: [Inserir resultado de controle].

Saída: [Inserir resultado de controle].

Efeito: COORD_CTL recebe de CTLi o resultado de um controle e o envia aos dois NHS_CTL para que seja transmitido ao DC.

O DC pode enviar vários comandos de controle para diferentes estações e ficar esperando os resultados. Como as sessões para envio dos resultados não ocorrem, necessariamente, na mesma ordem dos comandos, o número de seqüência dos resultados tem que ser outro, ou seja, existe um número de seqüência no sentido DC → COR (para os comandos de controle) e outro no sentido COR → DC (para os resultados de controle). Esse número de seqüência do sentido COR → DC é mantido pelos dois NHS_CTL e sua consistência feita pelo IOM na iniciação do OM. Isso é necessário porque na queda de um OM antes da recepção do ACK_CTRL para um resultado de controle, e com a reativação do coordenador no outro OM, é feita a retransmissão desse resultado que será descartado pelo DC através do número de seqüência enviado.

O DC verifica a que controle um determinado resultado se refere através da estação e do índice do ponto enviado pelo COR no resultado.

Deve-se ressaltar que o DC bloqueia comandos de controle múltiplos para uma mesma estação, ou seja, o COR só recebe um controle para uma determinada estação depois que a sessão do resultado do controle anterior for terminada.

Caso o SCSC receba outro comando de controle para uma mesma estação ele o envia ao SCER. No SCER pode ocorrer "time-out" ou caso o controle anterior já tenha terminado, o controle acaba sendo realizado.

NHS_CTL

Entrada: [Inserir resultado de controle].

Saída: (Liberar resultado de controle).

Efeito: NHS_CTL insere o resultado do controle para transmissão e libera esse resultado para COORD_CTL para que este possa liberar o SCER. A retenção do resultado de controle no SCER até a inserção nos dois NHS_CTL é necessária para garantir que uma queda de um OM nessa fase não provoque perda de dados.

Cada objeto CTL faz o seqüenciamento dos resultados de controle enviados ao COORD_CTL, ou seja, existe um número de seqüência para controle para cada estação do sistema. Este número de seqüência é utilizado pelo NHS_CTL para que um mesmo resultado de controle não seja inserido mais de uma vez e pelo IOM para fazer a consistência dos dados por ocasião da iniciação de um OM.

Caso não haja nenhuma outra sessão em andamento no sentido COR -> DC para o tipo de dado de controle, NHS_CTL coloca um resultado de controle à disposição do COORD_CTL ativo para transmissão.

COORD_CTL_2

Entrada: (Liberar resultado do controle).

Saída: (Liberar resultado do controle) para CTLi e [Pedido de Transmissão do Resultado de Controle].

Efeito: COORD_CTL libera o resultado de controle para CTLi.

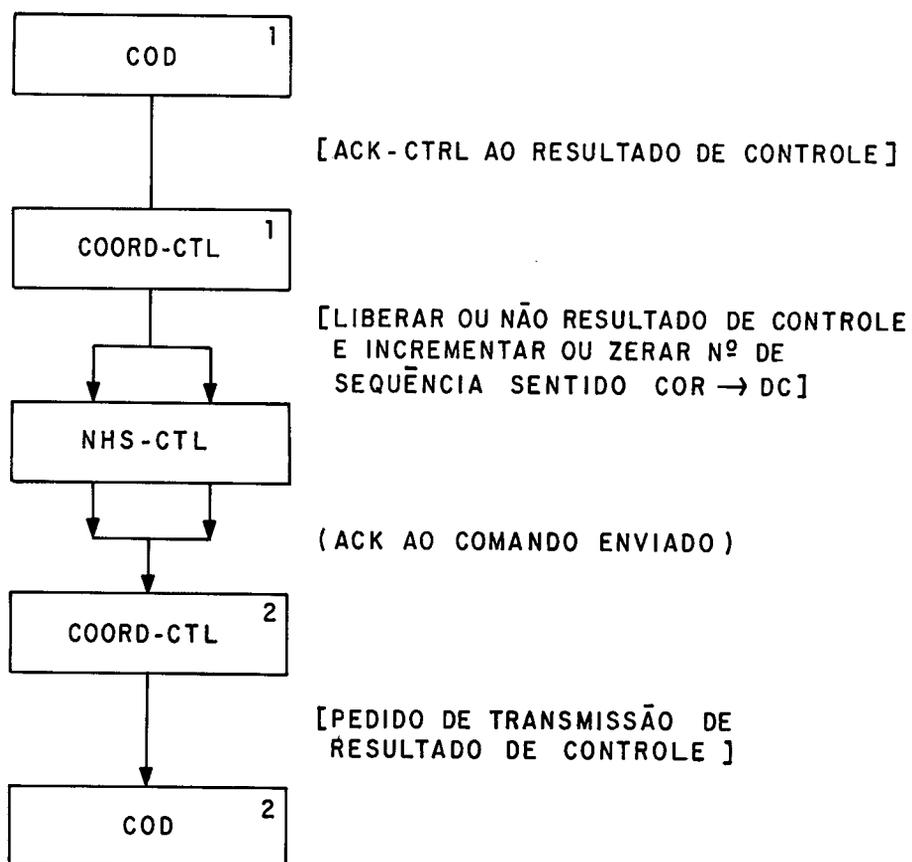
Não havendo sessão no sentido COR -> DC em andamento, COORD_CTL envia, ao objeto COD, o resultado de controle destinado pelo NHS_CTL à transmissão e coloca esse resultado à espera de aceitação.

COD

Entrada: [Pedido de transmissão do resultado de controle].

Saída: -

Efeito: COD transmite ao DC o resultado de controle.



DSA VI.4 - RECEPÇÃO DA RESPOSTA AO RESULTADO DE CONTROLE.

RECEPÇÃO DA RESPOSTA AO RESULTADO DE CONTROLE

COD 1

Entrada: -

Saída: [ACK_CTRL do resultado de controle].

Efeito: Ao receber do DC um ACK_CTRL do resultado de um controle, o COD o encaminha ao COORD_CTL.

COORD_CTL 1

Entrada: [ACK_CTRL do resultado de controle].

Saída: [Liberar ou não resultado de controle e incrementar ou zerar número de seqüência sentido COR -> DC].

Efeito: COORD_CTL recebe o ACK_CTRL e envia um comando aos dois NHS_CTL indicando as ações a serem tomadas a respeito do resultado de controle transmitido e do número de seqüência sentido COR -> DC.

Caso o ACK_CTRL indique que o resultado de controle foi recebido com sucesso, COORD_CTL envia um comando para liberar esse resultado que estava esperando aceitação e incrementar o número de seqüência do tipo de dado sentido COR -> DC.

Caso o ACK_CTRL indique número de seqüência errado, COORD_CTL envia um comando para zerar o número de seqüência sentido COR -> DC. COORD_CTL retransmitirá a mesma mensagem com o número de seqüência zerado, se o contador de tentativas não tiver atingido NTTM. Esse contador é incrementado e continua a contabilizar o número de retransmissões. Se o contador atingiu o máximo o

comando enviado ao NHS_CTL indicará, também, liberação do resultado de controle e o COORD_CTL não retransmitirá a mensagem.

Caso o ACK_CTRL indique outra causa qualquer de erro, NHS_CTL envia um comando para zerar o número de seqüência e liberar o resultado de controle à espera de aceitação.

NHS_CTL

Entrada: [Liberar ou não resultado de controle e incrementar ou zerar número de seqüência sentido COR -> DC].

Saída: (Ack ao comando enviado).

Efeito: NHS_CTL executa as ações indicadas no comando recebido e caso o resultado do controle à espera de aceitação tenha sido liberado e exista algum outro a ser transmitido, ele coloca esse outro resultado à disposição do COORD_CTL ativo para transmissão.

COORD_CTL_2

Entrada: (Ack ao comando enviado).

Saída: [Pedido de Transmissão de resultado de controle].

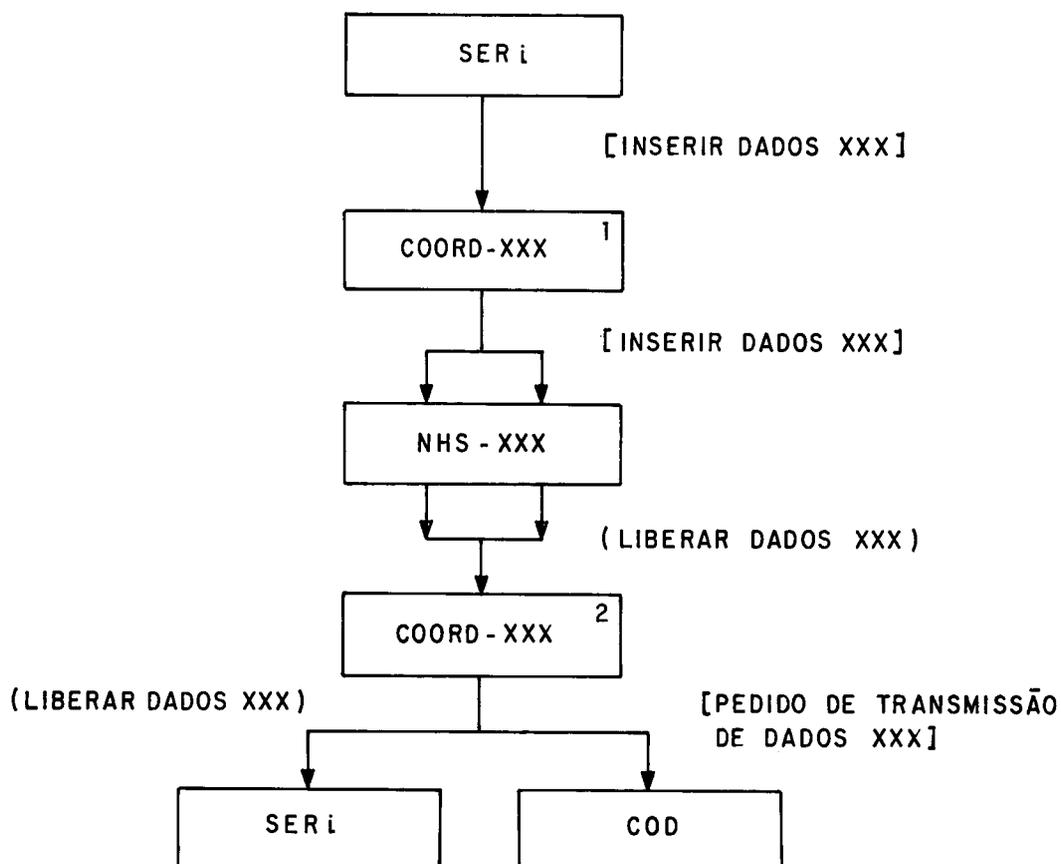
Efeito: COORD_CTL ou retransmite um resultado de controle ou transmite um outro resultado de controle colocado para transmissão pelo NHS_CTL, dependendo do caso.

COD_2

Entrada: [Pedido de transmissão de resultado de controle].

Saída: -

Efeito: COD transmite ao DC um resultado de controle.



DSA VI.5 - RECEPÇÃO DE DADOS XXX (DIG, ANL, TOT E SOE)
DO SCER.

RECEPÇÃO DE DADOS XXX (DIG, ANL, TOT e SOE) DO SCER

COORD XXX 1

Entrada: [Inserir dados XXX].

Saída: [Inserir dados XXX].

Efeito: Envia para os dois NHS_XXX os dados recebidos de SERi a fim de que sejam armazenados para posterior envio ao DC e fica aguardando a liberação desses dados.

Cada objeto SER faz o seqüenciamento das listas enviadas ao COORD_XXX, ou seja, existe um número de seqüência para cada estação do sistema. Este número é utilizado pelo NHS_XXX para que uma mesma lista de dados não seja inserida mais de uma vez e pelo IOM para fazer a consistência dos dados por ocasião da iniciação de um OM.

O seqüenciamento enviado ao DC tem que ser o mesmo para as listas de todas as estações, para isso é utilizado o número de seqüência do tipo de dado sentido COR -> DC. Este número de seqüência é necessário porque na queda de um OM, antes da recepção do ACK_XXX para uma lista de dados, e reativação do coordenador no outro OM, é feita a retransmissão dessa lista que será descartada pelo DC através do número de seqüência enviado. Este número deve ser atualizado pelo IOM na iniciação de um OM.

NHS XXX

Entrada: [Inserir dados XXX].

Saída: (Liberar dados XXX).

Efeito: Armazena os dados recebidos para transmissão ao DC e libera esses dados para que COORD_XXX também possa liberá-los para o SCER. A retenção dos dados no SCER até a inserção nos dois NHS_XXX é necessária para garantir que uma queda de um OM nessa fase não provoque perda de dados.

Caso não haja nenhuma outra sessão em andamento no sentido COR -> DC para o tipo de dado XXX, NHS_XXX coloca um resultado de controle à disposição do COORD_XXX ativo para transmissão.

COORD_XXX 2

Entrada: (Liberar dados XXX).

Saída: (Liberar dados XXX) para SERi e [Pedido de Transmissão de dados XXX] para o COD.

Efeito: COORD_XXX libera os dados XXX para SERi.

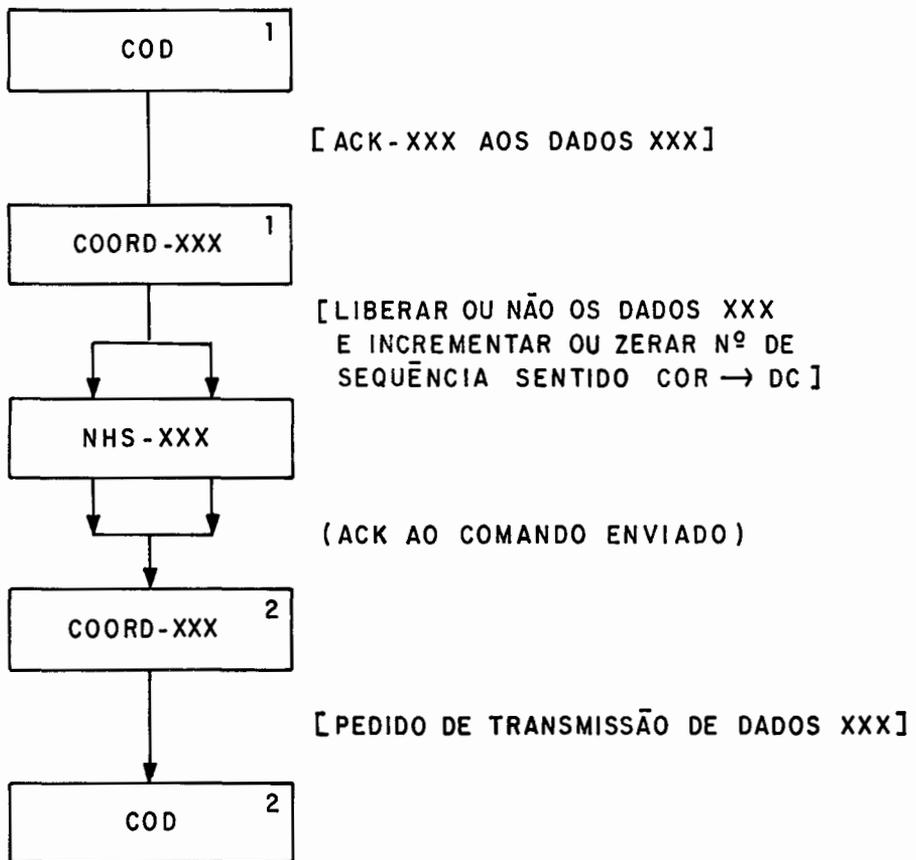
Não havendo sessão no sentido COR -> DC em andamento, COORD_XXX envia, ao objeto COD, os dados XXX destinados pelo NHS_XXX à transmissão e coloca esses dados à espera de aceitação.

COD

Entrada: [Pedido de transmissão de dados XXX].

Saída: -

Efeito: COD transmite ao DC dados XXX.



DSA VI.6 - RECEPÇÃO DE RESPOSTA DO DC DO TIPO DE DADO
XXX (DIG, ANL, TOT E SOE).

RECEPÇÃO DE RESPOSTA DO DC DO TIPO DE DADO XXX

(DIG, ANL, TOT e SOE)

COD 1

Entrada: -

Saída: [ACK_XXX aos dados XXX].

Efeito: Ao receber do DC um ACK_XXX aos dados XXX enviados, o COD o encaminha ao COORD_XXX.

COORD XXX 1

Entrada: [ACK_XXX aos dados XXX].

Saída: [Liberar ou não os dados XXX e incrementar ou zerar número de seqüência sentido COR -> DC].

Efeito: COORD_XXX recebe o ACK_XXX e envia um comando aos dois NHS_XXX indicando as ações a serem tomadas a respeito dos dados transmitidos e do número de seqüência sentido COR -> DC.

Caso o ACK_XXX indique que os dados foram recebidos com sucesso, COORD_XXX envia um comando para liberar os dados que estavam esperando aceitação e incrementar o número de seqüência do tipo de dado sentido COR -> DC.

Caso o ACK_XXX indique número de seqüência errado, COORD_XXX envia aos NHS_XXX um comando para zerar o número seqüência sentido COR -> DC. COORD_CTL retransmitirá a mesma mensagem com o número de seqüência zerado se o contador de tentativas não tiver atingido NTTM, esse contador é incrementado e continua a contabilizar o número de retransmissões; se o contador atingiu o máximo o

comando enviado ao NHS_XXX indicará, também, liberação dos dados e o COORD_XXX não retransmitirá a mensagem.

Caso o ACK_XXX indique outra causa qualquer de erro, NHS_XXX envia um comando para zerar o número de seqüência e liberar os dados à espera de aceitação.

NHS_XXX

Entrada: [Liberar ou não os dados XXX e incrementar ou zerar número de seqüência sentido COR -> DC].

Saída: (Ack ao comando enviado).

Efeito: NHS_XXX executa as ações indicadas no comando recebido e caso os dados à espera de aceitação tenham sido liberados e existam mais dados a serem transmitidos, ele coloca novos dados à disposição do COORD_XXX ativo para transmissão.

COORD_XXX 2

Entrada: (Ack ao comando enviado).

Saída: [Pedido de Transmissão de Dados XXX].

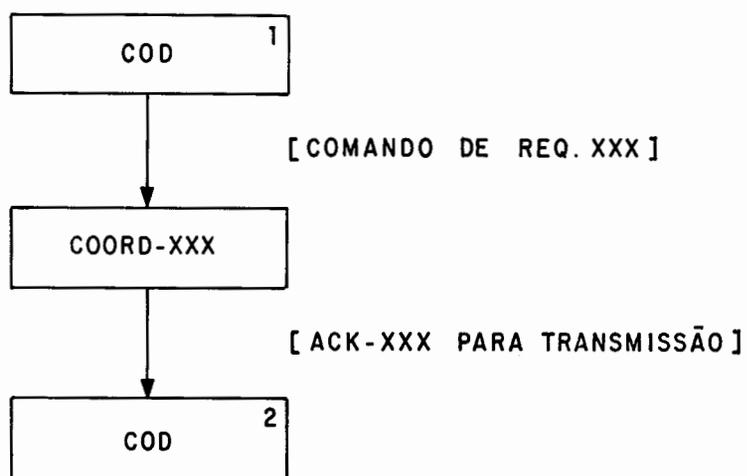
Efeito: COORD_XXX ou retransmite os dados ou transmite novos dados colocados para transmissão pelo NHS_XXX, dependendo do caso.

COD 2

Entrada: [Pedido de transmissão de dados XXX].

Saída: -

Efeito: COD transmite ao DC os dados XXX.



DSA VI.7 - RECEPÇÃO DE COMANDO DO TIPO DE DADO XXX
(DIG, ANL, TOT E SOE) E ENVIO DA RESPOSTA.

RECEPÇÃO DE COMANDO DO TIPO DE DADO XXX (DIG, ANL, TOT e SOE) E
ENVIO DA RESPOSTA

COD_1

Entrada: -

Saída: [Comando de REQ_XXX].

Efeito: Ao receber do DC um comando REQ_XXX, o COD encaminha esse comando ao COORD_XXX.

COORD_XXX

Entrada: [Comando de REQ_XXX].

Saída: [ACK_XXX para transmissão].

Efeito: O comando REQ_XXX tem por finalidade averiguar se o tipo de dado XXX encontra-se ativo e em perfeitas condições de funcionamento.

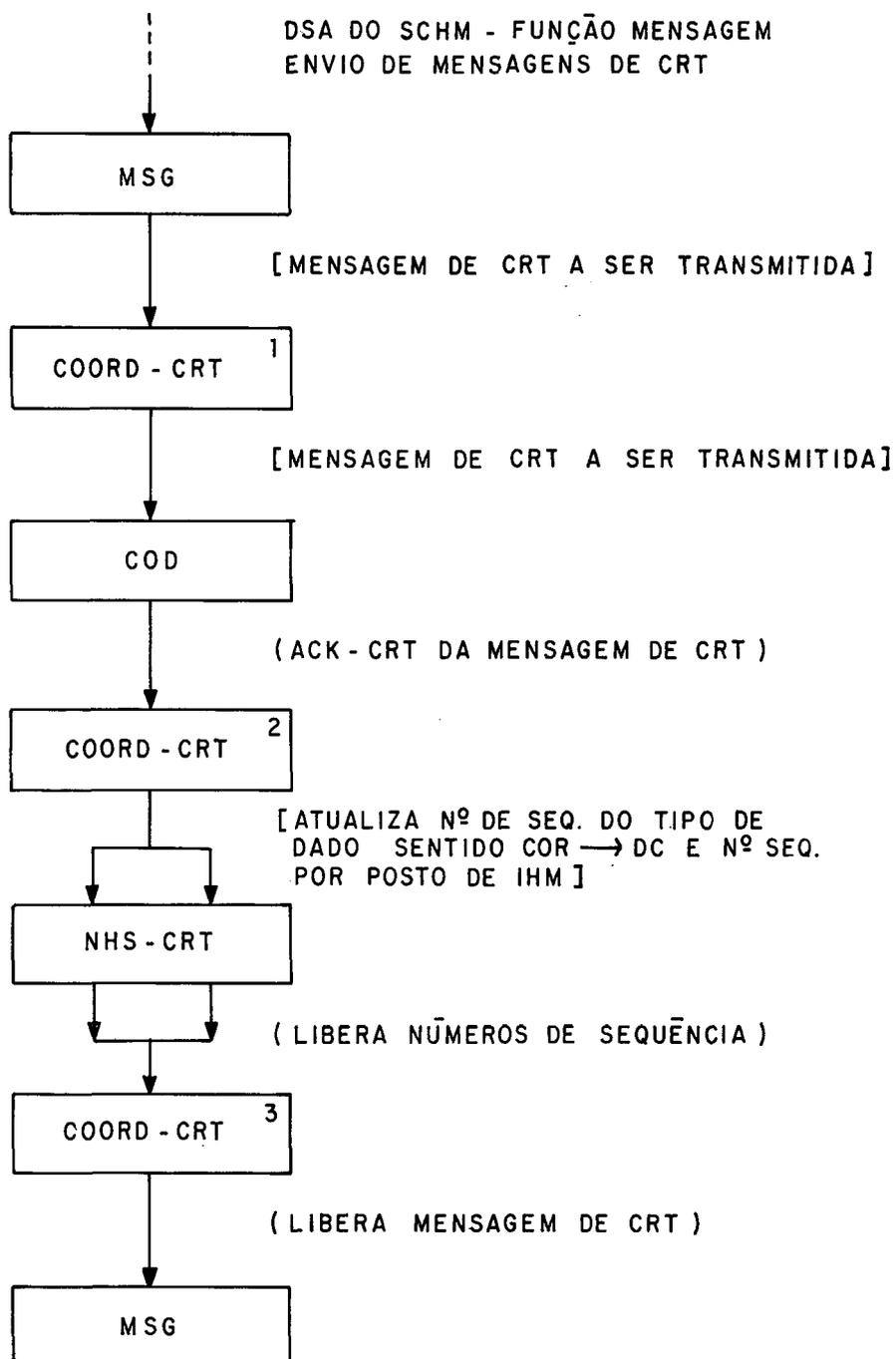
COORD_XXX envia um ACK_XXX, como resposta ao comando REQ_XXX, ao objeto COD para ser transmitido ao DC. Como uma sessão tem que ser iniciada e terminada pelo mesmo enlace, não há necessidade desse ACK_XXX ser enviado aos objetos NHS_XXX. A causa enviada pode ser SUCESSO, NÚMERO DE SEQUÊNCIA ERRADO ou FORMATO GERAL ERRADO de acordo com o comando recebido.

COD_2

Entrada: [ACK_XXX para transmissão].

Saída: -

Efeito: COD transmite ao DC a resposta ao comando REQ_XXX.



DSA VI.8 - TRANSMISSÃO DE MENSAGENS DE CRT PARA O DC

TRANSMISSÃO DE MENSAGENS DE CRT PARA O DC

COORD CRT 1

Entrada: [Mensagem de "crt" a ser transmitida].

Saída: [Mensagem de "crt" a ser transmitida].

Efeito: Cada objeto MSG envia ao COORD_CRT as mensagens de "crt" referentes a um posto de IHM. Logo, existem três números de seqüência para as mensagens de "crt" enviadas do SCHM para o SCSC. Esses números de seqüência são mantidos pelos objetos NHS_CRT.

Ao receber uma mensagem de MSG, COORD_CRT acessa esse número de seqüência através do NHS_CRT residente no mesmo operador e aceita a mensagem caso o número de seqüência esteja correto (número de seqüência diferente do anterior). COORD_CRT envia, então, a mensagem ao COD para que seja transmitida ao DC.

Caso o número de seqüência seja o mesmo da mensagem anterior, a mensagem é descartada e para o MSG é repetido o Ack anterior.

Além desses três números de seqüência, existe um outro que é utilizado na comunicação com o DC para seqüenciar todas as mensagens de "crt" que trafegam no sentido COR -> DC para esse tipo de dado. Esse número de seqüência tem sua consistência feita pelo IOM na iniciação de um OM. Cabe ressaltar que uma queda do OC, hospedeiro deste MSG, a partir desse ponto pode fazer com que o DC receba e aceite a mesma mensagem como se

fosse uma nova, caso o despachante do COR repita a mesma mensagem por outro posto de IHM.

COD

Entrada: [Mensagem de "crt" a ser transmitida].

Saída: (ACK_CRT da mensagem de "crt").

Efeito: COD transmite a mensagem de "crt" para o DC e fica aguardando a resposta do DC. Ao receber essa resposta, COD a envia para COORD_CRT para que este possa liberar a mensagem para o MSG.

COORD_CRT 2

Entrada: (ACK_CRT da mensagem de "crt").

Saída: [Atualiza número de seqüência do tipo de dado sentido COR -> DC e número de seqüência por posto de IHM].

Efeito: Caso o ACK_CRT recebido indique que a mensagem transmitida foi recebida com sucesso, COORD_CRT envia para os dois NHS_CRT um comando para atualizar o número de seqüência recebido do MSG (número de seqüência por posto de IHM) e o número de seqüência do tipo de dado de mensagem de "crt" sentido COR -> DC.

Uma queda do OM, hospedeiro deste COORD_CRT, antes que os NHS_CRT recebam este comando faz com que o MSG repita a mensagem e conseqüentemente haja a retransmissão da mesma ao DC, que a descartará pelo número de seqüência do tipo de dado, sentido COR -> DC.

Caso o ACK_CRT indique número de seqüência errado, COORD_CRT envia aos NHS_CRT um comando para zerar o número de seqüência do tipo de dado, sentido COR -> DC, e não atualizar o número de seqüência do posto de IHM. COORD_CRT retransmitirá a mesma mensagem com o número de seqüência zerado se o contador de tentativas não tiver atingido NTTM, o contador CTTM é incrementado e continua a contabilizar o número de tentativas de retransmissão; se CTTM atingiu NTTM o comando indicará, também, a liberação dos dados.

Caso o ACK_CRT indique outra causa qualquer de erro, COORD_CRT envia aos NHS_CRT um comando para atualizar apenas o número de seqüência do posto de IHM, o número de seqüência do tipo de dado é zerado. COORD_CRT não retransmite a mensagem.

NHS_CRT

Entrada: [Atualiza número de seqüência do tipo de dado sentido COR -> DC e número de seqüência por posto de IHM].

Saída: (Libera número de seqüência).

Efeito: NHS_CRT atualiza o número de seqüência do tipo de dado de mensagem de "crt" sentido COR -> DC e o número de seqüência do posto de IHM em questão, conforme instruções do COORD_CRT.

Uma queda do OM a partir deste ponto faz com que o MSG repita a mensagem para o outro OM que a

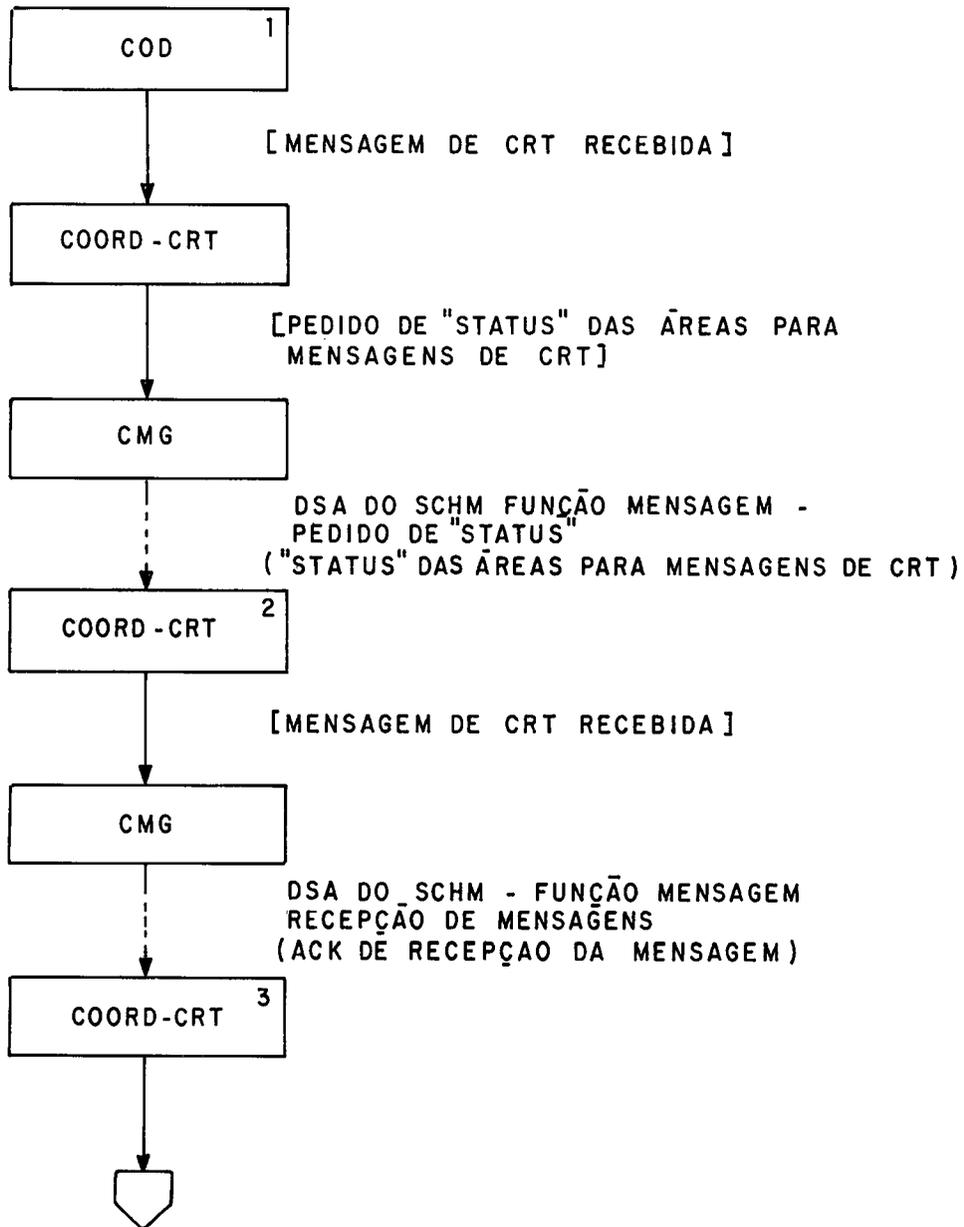
descartará pelo número de seqüência do posto de IHM enviado pelo MSG.

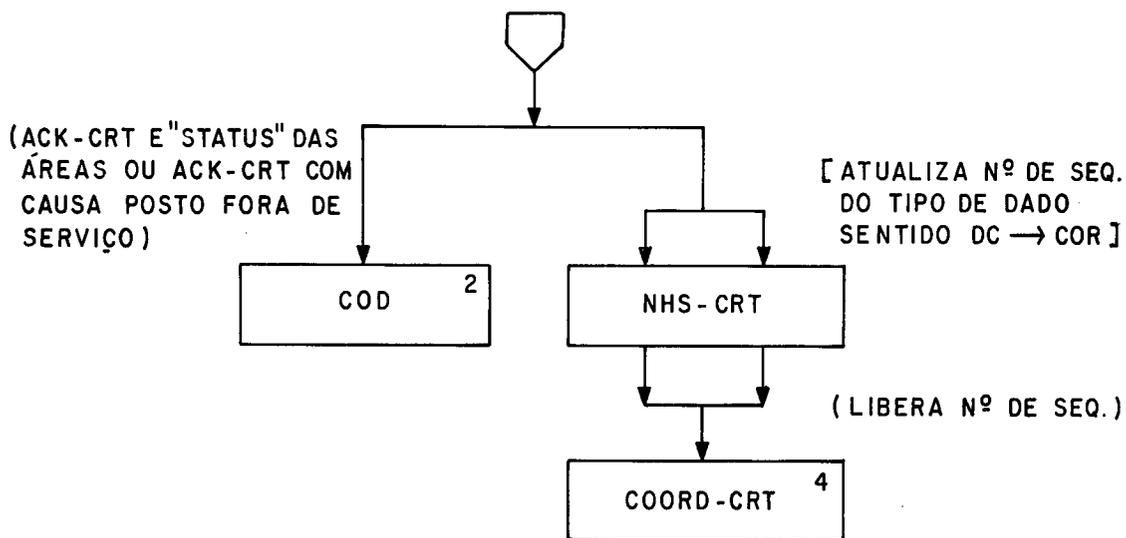
COORD CRT 3

Entrada: (Libera números de seqüência).

Saída: (Libera mensagem de "crt").

Efeito: Uma vez atualizados os números de seqüência, COORD_CRT libera a mensagem para o MSG.





DSA VI.9 - RECEPÇÃO DE MENSAGENS DE CRT DO DC .

RECEPÇÃO DE MENSAGENS DE CRT DO DC

COD 1

Entrada: -

Saída: [Mensagem de "crt" recebida].

Efeito: Ao receber do DC uma mensagem de "crt", o COD encaminha esse comando ao COORD_CRT.

COORD_CRT 1

Entrada: [Mensagem de "crt" recebida].

Saída: [Pedido de "Status" das áreas para mensagens de "crt"].

Efeito: Após ter verificado que o número de seqüência é correto, através do acesso ao número de seqüência mantido pelo NHS_CRT residente no mesmo operador e também o formato geral, envia um pedido de "status" das áreas para CMG, objeto do SCHM, e fica aguardando a resposta.

Caso o número de seqüência enviado pelo DC seja o mesmo da mensagem anterior, a mensagem é descartada, ou seja, não é enviada ao CMG e o ACK_CRT da mensagem anterior é novamente enviado ao COD para ser retransmitido ao DC.

Caso o número de seqüência seja completamente errado a mensagem também é descartada e ao DC é enviado um ACK_CRT com causa NÚMERO DE SEQUENCIA ERRADO. Além disso, nesse caso o número de seqüência não é atualizado em NHS_CRT. O mesmo acontece para o caso de FORMATO GERAL ERRADO.

COORD CRT 2

Entrada: ("Status" das áreas para mensagens de "crt").

Saída: [Mensagem de "crt" recebida].

Efeito: Após ter recebido o "status" das áreas de mensagens de "crt" COORD_CRT envia a mensagem recebida para CMG e fica aguardando a resposta.

A área para a qual a mensagem enviada estava endereçada deve ser marcada ocupada pelo COORD_CRT.

Uma queda do OM antes que a resposta à mensagem de "crt" seja enviada ao DC e antes da atualização do número de seqüência no NHS_CRT, provoca retransmissão da mensagem pelo DC, no entanto, a mesma será descartada por CMG através do número de seqüência enviado pelo DC que é repassado ao CMG.

COORD CRT 3

Entrada: (Ack de recepção da mensagem).

Saída: (ACK_CRT e "status" das áreas ou ACK_CRT com causa POSTO FORA DE SERVIÇO) para o COD e [Atualizar número de seqüência do tipo de dado, sentido DC -> CORJ, para NHS_CRT.

Efeito: Recebe de CMG o Ack da mensagem de "crt" e o envia juntamente com o "status" das áreas, para o COD para ser transmitido ao DC. Como uma sessão tem que ser iniciada e terminada pelo mesmo enlace não há necessidade desse ACK_CRT ser enviado aos objetos NHS_CRT.

O número de seqüência enviado na mensagem no sentido DC -> COR é repetido na resposta.

Caso qualquer uma das comunicações com o CMG não seja realizada com sucesso (três OC fora de serviço ou CMG inativo) COORD_CRT envia ao COD um ACK_CRT com causa POSTO FORA DE SERVIÇO, indicando que a mensagem não foi recebida pela interface homem-máquina do COR. Neste caso o número de seqüência do tipo de dado sentido DC -> COR não é atualizado em NHS_CRT. Cabe ressaltar que neste caso, quando qualquer um dos OC voltar a serviço e assumir o CMG, ele deve aceitar o número de seqüência da primeira mensagem recebida como correto.

COD 2

Entrada: (ACK_CRT e "status" das áreas ou ACK_CRT com causa POSTO FORA DE SERVIÇO).

Saída: -

Efeito: COD transmite ao DC o ACK_CRT.

NHS_CRT

Entrada: [Atualiza número de seqüência do tipo de dado sentido DC -> COR].

Saída: (Libera número de seqüência).

Efeito: Atualizar o número de seqüência do tipo de dado sentido DC -> COR.

Uma queda do OM após a atualização do número de seqüência e antes da transmissão do ACK_CRT pelo COD provoca retransmissão da mensagem pelo DC, mas

neste caso o descarte será feito pelo COORD_CRT ativo no outro DM.

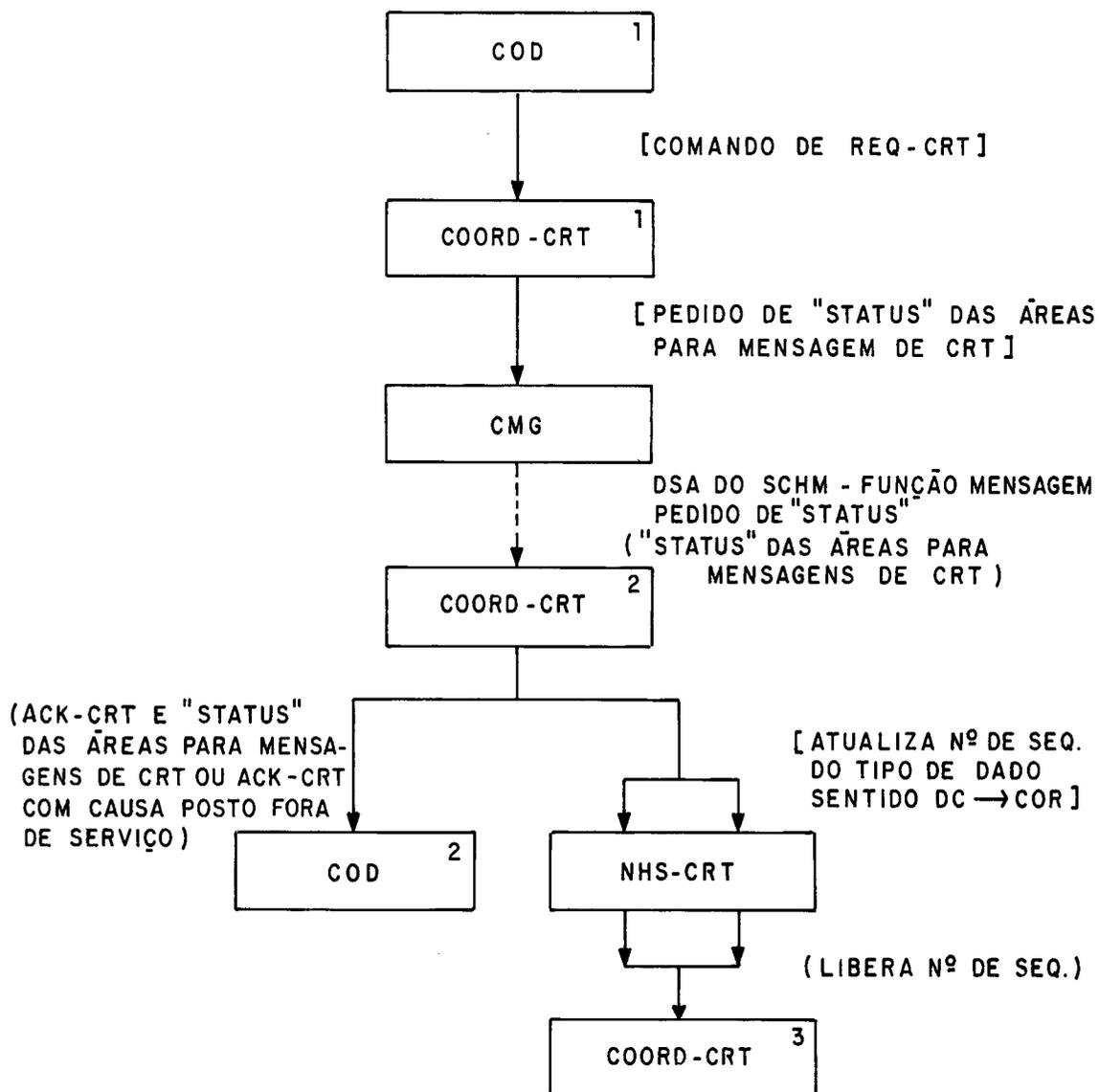
Uma queda do DM após a transmissão do ACK_CRT mas antes da atualização do número de sequência provoca a rejeição, pelo COORD_CRT ativo no outro DM, da próxima mensagem enviada (número de sequência errado), mas o DC repetirá a mesma mensagem com número de sequência zero e a mesma será aceita e o número de sequência sentido DC -> CDR reiniciado.

COORD CRT 3

Entrada: (Libera número de sequência).

Saída: -

Efeito: COORD_CRT fica liberado para recepção de outra mensagem lógica.



DSA VI.10 - RECEPÇÃO DE COMANDO DO TIPO DE DADO DE MENSAGENS DE CRT.

RECEPÇÃO DE COMANDO DO TIPO DE DADO DE MENSAGENS DE CRT

COD 1

Entrada: -

Saída: [Comando de REQ_CRT].

Efeito: Ao receber do DC um comando REQ_CRT, COD o encaminha a COORD_CRT.

COORD CRT 1

Entrada: [Comando de REQ_CRT].

Saída: [Pedido de "Status" das áreas para mensagens de "crt"].

Efeito: Após a verificação do número de seqüência e do formato geral, envia ao CMG um pedido de "status" das áreas para mensagens de "crt" e fica aguardando a resposta.

Os comandos REQ_CRT são seqüenciados junto com as mensagens de "crt" enviadas pelo DC, ou seja, o número de seqüência utilizado é o do tipo de dado sentido DC -> COR. Valem então, para esta descrição e as subseqüentes deste DSA, todas as observações sobre número de seqüência e sobre queda de operadores feitas na descrição do DSA anterior.

COORD CRT 2

Entrada: ("Status" das áreas para mensagens de "crt").

Saída: (ACK_CRT e "status" das áreas para mensagens de "crt" ou ACK_CRT com causa POSTO FORA DE SERVIÇO)

para o COD e [Atualizar número de seqüência do tipo de dado sentido DC -> COR] para NHS_CRT.

Efeito: Recebe de CMG o "status" das áreas para mensagens de "crt" e o envia para o COD para ser transmitido ao DC. Como uma sessão tem que ser iniciada e terminada pelo mesmo enlace, não há necessidade dessa resposta ser enviada aos objetos NHS_CRT.

Caso a comunicação com CMG não seja realizada com sucesso COORD_CRT envia a NHS_CRT um ACK_CRT com causa POSTO FORA DE SERVIÇO.

COD_2

Entrada: (ACK_CRT e "status" das áreas para mensagens de "crt" ou ACK_CRT com causa POSTO FORA DE SERVIÇO).

Saída: -

Efeito: COD transmite ao DC o ACK_CRT.

NHS_CRT

Entrada: [Atualiza número de seqüência do tipo de dado sentido DC -> COR].

Saída: (Libera número de seqüência).

Efeito: Atualiza número de seqüência do tipo de dado sentido DC -> COR.

COORD_CRT_3

Entrada: (Libera número de seqüência).

Saída: -

Efeito: COORD_CRT fica liberado para a recepção de um outro comando REQ_CRT.

Para mensagens estruturadas e de "display" os DSA e suas descrições são análogos aos apresentados para as mensagens de "crt". Diferem apenas nos objetos da aplicação que as enviam e na quantidade de números de seqüência da ligação PROCESSO DA APLICAÇÃO <=> NÍVEL 7.

As mensagens estruturadas que constituem relatórios são enviadas pelo objeto SPR e possuem um número de seqüência a elas associado; as mensagens estruturadas, propriamente ditas, são enviadas pelos mesmos objetos MSG e possuem, como as mensagens de "crt", um número de seqüência por posto de IHM, ou seja, um número de seqüência para cada um dos três MSG.

As mensagens de "display" que constituem os "hardcopies" são enviadas pelos objetos SPR e possuem três números de seqüência cada um de um posto de IHM; as telas são enviadas pelos MSG e, também, possuem um número de seqüência por posto de IHM.

Na recepção não há diferenças.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] - APPEL, OSIAS e VALLE, LUCIA - "Software Básico para Microprocessadores em Tempo Real". *II Seminário de Microprocessadores para a Indústria - SEMICRO*, CEPTEL, RJ, 1981.
- [2] - BRINCH HANSEN, P. - "The Architecture of Concurrent Programs". Englewood Cliffs, N. J., Prentice Hall, Inc., 1977.
- [3] - CCITT, "Interface Between Data Terminal Equipment (DTE) and Data Circuit Terminating Equipment (DCE) for Terminals Operating in the Packet Mode on Public Data Networks - Recommendation X25". Fascicle VIII.2, Geneva, 1980.
- [4] - CLAYBROOK, BILLY G. - "A Specification Method for Specifying Data and Procedural Abstractions". *IEEE Transactions on Software Engineering*, vol. SE-8, no 5, p. 449 a 459, September 1982.
- [5] - FREITAS, CÉSAR A. e MONTEIRO, VERA L. L. - "Comunicação entre Administradores de Objetos - Uma Filosofia de Especificação". *Conferência Latino Americana IEEE*, Buenos Aires, Argentina, 1988.

- [6] - GOMAA, H. - "A Software Design Method for Real Time Systems". *Communications of the ACM*, vol. 27, no 9, p. 938 a 949, September 1984.
- [7] - HOARE, C. A. R. - "Monitor: An Operating System Structuring Concept". *Communications of the ACM*, vol. 17, no 10 p. 549 a 557, October 1974.
- [8] - KIRNER, CLAUDIO e MENDES, SUELI - "Sistemas Operacionais Distribuídos Aspectos Gerais e Análise de sua Estrutura". Editora Campus Ltda, 1988.
- [9] - MENASCÊ, DANIEL A. e SCHWABE, DANIEL - "Redes de Computadores - Aspectos Técnicos e Operacionais". *Terceira Escola de Computação*, Departamento de Informática, PUC/RJ, Rio de Janeiro, 1982.
- [10] - NELSON, B. J. - "Remote Procedure Call". *PHD Thesis Department of Computer Science, Carnegie-Mellon University*, (Report CMV-C8-81-119), 201p, May 1981.
- [11] - PARNAS, D. L. - "On the Criteria To Be Used in Decomposing Systems into Modules". *Communications of the ACM*, vol. 15, no 12, p. 1053 a 1058, December 1972.
- [12] - SEGRE, LÍDIA e MENDES, SUELI - "O Conceito de Monitor como Instrumento de Sincronização em Programação Concorrente". *Revista Brasileira de Computação*, vol. 1, no 1, 1981.

- [13] - SILVA, ANTONIO (CEPEL) e MOTTA, JOSÉ e GARROFÊ, PAULO e
COTRIM, SÉRGIO (FURNAS) - "Projeto do Software
Aplicativo e de Suporte para o Centro Regional de
Supervisão e Controle de Furnas". *VIII Seminário
Nacional de Produção e Transmissão de Energia
Elétrica* - Grupo V, São Paulo, 1986.
- [14] - TANENBAUM, ANDREW - "Computer Networks". Prentice Hall,
Inc., Englewood Cliffs, New Jersey, 1981.
- [15] - TAROUCCO, LIANE - "Redes de Computadores Locais e de Longa
Distância". Mc Graw Hill, 1986.
- [16] - ZAVE, PAMELA - "The Operational Versus The Conventional
Approach to Software Development". *Communications of
the ACM*, vol. 27, no 2, p. 104 a 118, February
1984.
- [17] - ZIMMERMANN, HUBERT - "OSI Reference Model - The ISO Model
of Architecture for Open Systems Interconnection".
IEEE Transactions on Communications, vol. COM-28,
no 4, p. 425 a 432, April 1980.

APÊNDICE A

EXTENSÕES DAS MENSAGENS

i) REQ_INIC

```

+-----+-----+-----+-----+-----+-----+-----+
|                Versão do Banco de Dados                |
+-----+-----+-----+-----+-----+-----+-----+
|                Versão do Protocolo de Comunicações        |
+-----+-----+-----+-----+-----+-----+-----+

```

ii) ACK_INIC

```

+-----+-----+-----+-----+-----+-----+-----+
|                                Causa                                |
+-----+-----+-----+-----+-----+-----+-----+

```

Causa	-	Formato Geral Errado	0
		Sucesso	1
		Número de Sequência Errado	2
		Versão de Protocolo Errada	6
		Versão de BD Errada	7

iii) EXEC_CTRL

	índice do Ponto	
+-		-+
+-		-+
	Nome do Ponto	
	(20 bytes)	
+-		-+
+-	Ação/Valor	-+
+-		-+

Ação - abrir/aumentar.... AAAA (hex)
 fechar/diminuir... 7777 (hex)

iv) RESULT_CTRL

	índice do Ponto	
	Resultado do Controle	
+-		-+

v) ACK_CTRL

```

+-----+-----+-----+-----+-----+-----+-----+
|                                     Causa                                     |
+-----+-----+-----+-----+-----+-----+-----+
    
```

```

Causa - Formato Geral Errado      .... 0
        Sucesso                    .... 1
        Número de Sequência Errado .... 2
        Estação Fora de Serviço    .... 5
    
```

vi) INICS_DIG e INTGR_DIG

```

+-----+-----+-----+-----+-----+-----+-----+
|                                     |                                     |
+---+                               Número de Pontos                               +---+
|                                     |                                     |
+-----+-----+-----+-----+-----+-----+-----+
|                                     |                                     |
+---+                               índice do Ponto 'n'                               +---+
|                                     |                                     |
+-----+-----+-----+-----+-----+-----+-----+
| Estado/Flags de 'n+1' | Estado/Flags de 'n' |
+-----+-----+-----+-----+-----+-----+-----+
| Estado/Flags de 'n+3' | Estado/Flags de 'n+2' |
+-----+-----+-----+-----+-----+-----+-----+
|                                     |
|                                     |
|                                     |
|                                     |
|                                     |
|                                     |
|                                     |
|                                     |
+-----+-----+-----+-----+-----+-----+-----+
| Estado/Flags de 'n+m' |
+-----+-----+-----+-----+-----+-----+-----+
    
```

```

Estado/Flags - +-----+-----+-----+-----+
                | Fm |         | E | Fi |
                +-----+-----+-----+-----+
                |         |         |         |
                |         |         | +--> Flag 'Inválido'
                |         |         | +-----> Estado do ponto
                +-----+-----+-----+-----+-----> Flag 'Manual'
    
```

```

Válido/Aberto/Inativo/Normal.....0
Inválido/Fechado/Ativo/Manual.....1
    
```

vii) VAR_DIG e INIC_DIG

```

+-----+-----+-----+-----+-----+-----+
|                                     |
|           Hora do dia               |
|-----+-----+-----+-----+-----+
|                                     |
|           Minuto da hora            |
|-----+-----+-----+-----+-----+
|                                     |
|           Segundo do minuto         |
|-----+-----+-----+-----+-----+
|                                     |
|                                     |
|---           Número de Pontos       ---|
|                                     |
|-----+-----+-----+-----+-----+
|                                     |
|---           [ndice de 'i'          ---|
|                                     |
|-----+-----+-----+-----+-----+
|                                     |
|---           [ndice de 'j'          ---|
|                                     |
|-----+-----+-----+-----+-----+
| Estado/Flags de 'j' | Estado/Flags de 'i' |
|-----+-----+-----+-----+-----+
|                                     |
|                                     |
|           e t c . . .               |
|                                     |
|-----+-----+-----+-----+-----+
|                                     |
|---           [ndice de 'k'          ---|
|                                     |
|-----+-----+-----+-----+-----+
|                                     |
|---           -----                 ---|
|                                     |
|-----+-----+-----+-----+-----+
|           ---           | Estado/Flags de 'k' |
|-----+-----+-----+-----+-----+

```

Estado/Flags - idem ao das mensagens de INICS_DIG
e INTGR_DIG

viii) REQ_DIG

Não tem Extensão

ix) ACK_DIG

```

+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Causa                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

```

Causa - Formato Geral Errado      ..... 0
        Sucesso                    ..... 1
        Número de Sequência Errado ..... 2
        Índice de Fonte Errado     ..... 4

```

x) INICS_ANL e INTGR_ANL

Número de Medidas
Índice de 'n'
Valor de 'n'
Flags de 'n'
Valor de 'n+1'
Flags de 'n+1'
etc ...
Valor de 'n+m'
Flags de 'n+m'

Flags Analógicos -

Condição	Fm	Fi
		+--> Flag 'Inválido'
	+--> Flag 'Manual'	
<----->		
	+--> 1 Superior overflow	
	2 Superior máxima	
	3 Superior normal	
	4 Normal	
	5 Inferior normal	
	6 Inferior mínima	
	7 Inferior overflow	

xi) INIC_ANL e VAR_ANL

```

+-----+-----+-----+-----+-----+-----+
|                                     |
|           Hora do dia               |
+-----+-----+-----+-----+-----+-----+
|                                     |
|           Minuto da hora            |
+-----+-----+-----+-----+-----+-----+
|                                     |
|           Segundo do minuto         |
+-----+-----+-----+-----+-----+-----+
|                                     |
|           Número de Medidas         |
+-----+-----+-----+-----+-----+-----+
|                                     |
|           índice de 'i'            |
+-----+-----+-----+-----+-----+-----+
|                                     |
|                                     |
|-----+-----+-----+-----+-----+-----+
|                                     |
|           Valor de 'i'              |-----+
|                                     |
+-----+-----+-----+-----+-----+-----+
|                                     |
|           Flags de 'i'             |
+-----+-----+-----+-----+-----+-----+
|                                     |
|           índice de 'j'            |
+-----+-----+-----+-----+-----+-----+
|                                     |
|                                     |
|-----+-----+-----+-----+-----+-----+
|                                     |
|           Valor de 'j'              |-----+
|                                     |
+-----+-----+-----+-----+-----+-----+
|                                     |
|           Flags de 'j'             |
+-----+-----+-----+-----+-----+-----+
|                                     |
|           e t c ...                 |
+-----+-----+-----+-----+-----+-----+
|                                     |
|           índice de 'k'            |
+-----+-----+-----+-----+-----+-----+
|                                     |
|                                     |
|-----+-----+-----+-----+-----+-----+
|                                     |
|           Valor de 'k'              |-----+
|                                     |
+-----+-----+-----+-----+-----+-----+
|                                     |
|           Flags de 'k'             |
+-----+-----+-----+-----+-----+-----+

```

Flags de Analógicos - idem ao das mensagens de INICS_ANL
e INTGR_ANL

xii) REQ_ANL

Não tem Extensão

xiii) ACK_ANL

```
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Causa                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

```
Causa - Formato Geral Errado      .... 0
      Sucesso                       .... 1
      Número de Sequência Errado   .... 2
      Índice de Ponto Errado       .... 4
```


xv) VAR_TOT e INIC_TOT

```

+-----+-----+-----+-----+-----+-----+-----+
|           Dia do mês           |
+-----+-----+-----+-----+-----+-----+-----+
|           Hora do dia          |
+-----+-----+-----+-----+-----+-----+-----+
|           Minuto da hora       |
+-----+-----+-----+-----+-----+-----+-----+
|           Segundo do minuto    |
+-----+-----+-----+-----+-----+-----+-----+
|           Número de Totalizadores
+-----+-----+-----+-----+-----+-----+-----+
|           índice de 'i'       |
+-----+-----+-----+-----+-----+-----+-----+
|           |                   |
+---          Valor de 'i'          ---+
|           |                   |
+-----+-----+-----+-----+-----+-----+-----+
|           Flags de 'i'        |
+-----+-----+-----+-----+-----+-----+-----+
|           índice de 'j'       |
+-----+-----+-----+-----+-----+-----+-----+
|           |                   |
+---          Valor de 'j'          ---+
|           |                   |
+-----+-----+-----+-----+-----+-----+-----+
|           Flags de 'j'        |
+-----+-----+-----+-----+-----+-----+-----+
|           |                   |
|           e t c ...           |
+-----+-----+-----+-----+-----+-----+-----+
|           índice de 'k'       |
+-----+-----+-----+-----+-----+-----+-----+
|           |                   |
+---          Valor de 'k'          ---+
|           |                   |
+-----+-----+-----+-----+-----+-----+-----+
|           Flags de 'k'        |
+-----+-----+-----+-----+-----+-----+-----+

```

Flags de Totalizadores - idem ao das mensagens de
INICS_ANL e INTGR_ANL

xvi) REQ_TOT

Não tem extensão

xvii) ACK_TOT

```

+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Causa                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

```

Causa - Formato Geral Errado      .... 0
        Sucesso                    .... 1
        Número de Sequência Errado .... 2
        Índice de Ponto Errado     .... 4

```

xviii) MSG_SOE

```

+-----+-----+-----+-----+-----+-----+-----+-----+ <--+
|                                     | |                                     |
|           Dia do mês                 | |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     | |                                     |
|           Hora do dia                 | | Data                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     | |                                     |
|           Minuto da hora              | | hora                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     | |                                     |
|           Segundo do minuto           | | base                               |
+-----+-----+-----+-----+-----+-----+-----+-----+ <--+
|                                     | |                                     |
+---+                               | |                               +---+
|                                     | |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     | |                                     |
|           Milisegundo de 'i'         | |                               +---+
| E | FIT |                           | |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     | |                                     |
+---+                               | |                               +---+
|                                     | |                                     |
|           Índice de 'j'              | |                               +---+
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     | |                                     |
|           Milisegundo de 'j'         | |                               +---+
| E | FIT |                           | |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     | |                                     |
|                                     | |                                     |
|                                     | |                                     |
|           e t c ...                   | |                                     |
|                                     | |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     | |                                     |
+---+                               | |                               +---+
|                                     | |                                     |
|           Índice de 'k'              | |                               +---+
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     | |                                     |
|           Milisegundo de 'k'         | |                               +---+
| E | FIT |                           | |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

E Estado do ponto
FIT Flag de imprecisão de tempo

xxii) ACK_CRT

Direção COR -> DC

Causa	
8	1
16	9
Estado das Áreas	
24	17
32	25

Causa - Formato Geral Errado	0
Sucesso	1
Número de Sequência Errado	2
Posto Fora de Serviço	3

Estado das Áreas - Disponível	0
Não disponível	1

Direção DC -> COR

Causa	
-------	--

Causa - idem às do ACK_CRT na direção COR -> DC

xxiii) REQ_CRT

Não tem extensão

xxv) ACK_ESTR

Direção COR -> DC

Causa	
24	17
Estado das Áreas	
32	25

Causa - idem às do ACK_CRT

Estados das Áreas - Disponível 0
 Não disponível 1

Direção DC -> COR

Causa	
-------	--

Causa - idem às do ACK_CRT

xxvi) REQ_ESTR

Não tem extensão

xxviii) ACK_DSP

Direção COR -> DC

Causa	
8	1
Estado das Áreas	
16	9

Causa - idem às do ACK_CRT

Estado das Áreas	-	Disponível 0
		Não disponível 1

Direção DC -> COR

Causa	
-------	--

Causa - idem às do ACK_CRT

xxix) REQ_DSP

Não tem extensão

APÊNDICE B

ESTRUTURAÇÃO DO NÍVEL DE SEGMENTAÇÃO

O nível 4 trata da segmentação das mensagens lógicas, ou seja, ele é responsável pela divisão e pela junção dos fragmentos de uma mensagem.

Para cada tipo de dado ele mantém áreas de transmissão e de recepção suficientes para que o Nível de Sessão possa realizar uma sessão no sentido DC -> COR e uma no sentido COR -> DC simultaneamente. Logo, as transmissões de um comando e de uma resposta ocorrem independentemente, analogamente a recepção de um comando não interfere na de uma resposta. Isto é essencial para que a recepção de um comando de uma sessão no sentido DC -> COR não influencie no tempo estipulado pelo N5 (TTM) para recepção de uma resposta a um comando enviado ao DC. O N4 gerencia a alocação dessas áreas, não permitindo a transmissão de mensagens caso a área respectiva não se encontre liberada (retorna ao N5 o "status" *impossibilidade de transmissão*) e bloqueando ou desbloqueando a recepção de segmentos do N3 de acordo com a disponibilidade das áreas de recepção.

Para a realização dessas funções este nível possui quatro módulos distintos, como pode ser visto na figura (B.1), a saber:

- Módulo de Transmissão de Segmentos;
- Módulo de Liberação das Áreas de Transmissão;
- Módulo de Recepção de Segmentos;
- Módulo de Liberação das Áreas de Recepção.

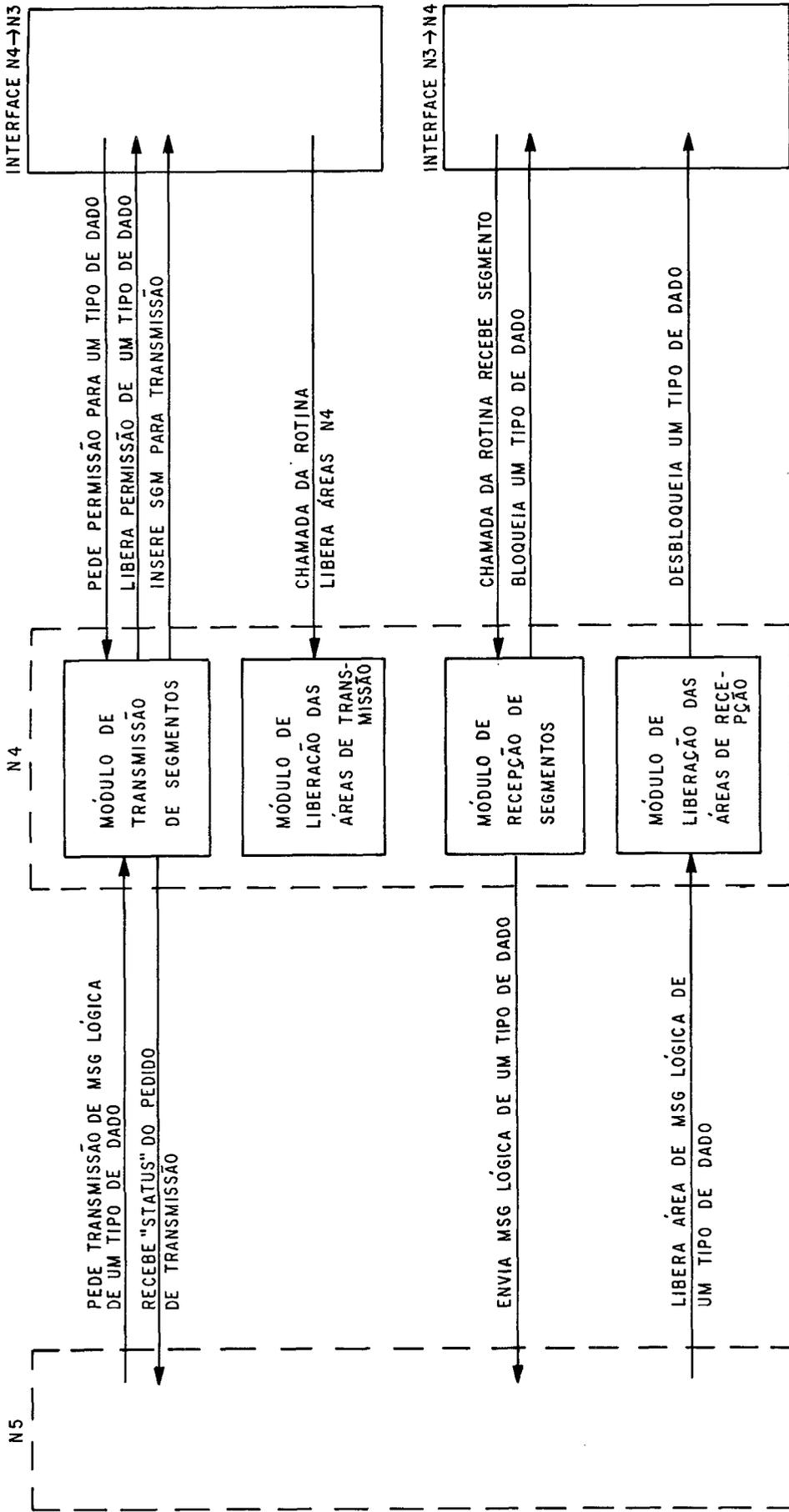


FIGURA B.1 - ORGANIZAÇÃO DO N4

APÊNDICE C

CÓDIGOS DE RETORNO AO USUÁRIO DA INTERFACE COM O NÍVEL 2

RETCOD	CMD-N3	ANS-N2	IND-N2	CAUSA	PROCEDIMENTO N3
CMD\$NOK-N2\$OPER	SET\$LINE\$ON	LINE\$IS\$UP		ENLACE FÍSICO JÁ ATIVO	AJUSTAR MÁQUINA DE ESTADO
CMD\$ACCEPT	SET\$LINE\$ON	LINE\$IS\$UP		SET\$LINE\$ON ACEITO	RESPOSTA ESPERADA
CTS\$DCD\$NOK	SET\$LINE\$ON	LINE\$IS\$DOWN		AUSÊNCIA DE CTS E DCD NA INICIAÇÃO DO N1	RETENTAR INDEFINIDAMENTE
CTS\$NOK	SET\$LINE\$ON	LINE\$IS\$DOWN	LINE\$IS\$DOWN	AUSÊNCIA DE CTS NA INICIAÇÃO DO N1	RETENTAR INDEFINIDAMENTE
DCD\$NOK	SET\$LINE\$ON	LINE\$IS\$DOWN		PERDA DE CTS	ENVIAR SET\$LINE\$ON E SET\$LINK\$ON
CMD\$NOK-N2\$NOT\$OPER	SET\$LINK\$ON	LINE\$IS\$DOWN		AUSÊNCIA DE DCD NA INICIAÇÃO DO N1	RETENTAR INDEFINIDAMENTE
LOC\$CONEC	SET\$LINK\$OFF SET\$LINK\$ON	LINE\$IS\$DOWN LINK\$IS\$UP	LINE\$IS\$DOWN	PERDA DA PORTADORA	ENVIAR SET\$LINE\$ON E SET\$LINK\$ON
CMD\$NOK-LINK\$SET\$DOWN	SET\$LINK\$ON SET\$LINK\$OFF	LINK\$IS\$UP LINK\$IS\$DOWN		CMD N3 INCOERENTE	AJUSTAR MÁQUINA DE ESTADO
CMD\$NOK-INFO\$TRANSFER	SET\$LINK\$ON	LINK\$IS\$UP		CMD N3 INCOERENTE	AJUSTAR MÁQUINA DE ESTADO
CMD\$NOK-REJ\$COND	SET\$LINK\$ON SET\$LINK\$OFF	LINK\$IS\$UP LINK\$IS\$DOWN		ESTABELECEMENTO DE ENLACE LÓGICO PELO ECD	RESPOSTA ESPERADA
CMD\$NOK-DISC\$PHASE	SET\$LINK\$OFF	LINK\$IS\$DOWN		ECD JÁ ESTÁ DESCONECTANDO A PEDIDO DO NÍVEL 3	AJUSTAR MÁQUINA DE ESTADO
CMD\$NOK-LINK\$SET\$UP	SET\$LINK\$ON	LINK\$IS\$DOWN		ECD JÁ ESTÁ CONECTADO	AJUSTAR MÁQUINA DE ESTADO
COL\$U\$DIF\$DOWN	SET\$LINK\$OFF	LINK\$IS\$DOWN		N2 JÁ INDICOU AO N3 QUE IRÁ DESCONECTAR OU REINICIAR O ENLACE	AJUSTAR MÁQUINA DE ESTADO
				ECD JÁ ESTÁ DESCONECTADO	AJUSTAR MÁQUINA DE ESTADO
				ECD JÁ ESTÁ TENTANDO (RE) INICIAÇÃO	AJUSTAR MÁQUINA DE ESTADO
				COLISÃO DE CMD U DIFERENTES COM ECD TENTANDO DESCONEXÃO	RESPOSTA ESPERADA

RETCOD	CMD-N3	ANS-N2	IND-N2	CAUSA	PROCEDIMENTO N3
REM\$DM	SET\$LINK\$ON	LINK\$IS\$DOWN		ETD JÁ ESTÁ DESCONECTANDO E RECUSA CONEXÃO	RETENTAR INDEFINIDAMENTE
	SET\$LINK\$OFF	LINK\$IS\$DOWN	LINK\$IS\$DOWN	ETD JÁ ESTÁVIA DESCONECTADO OU TENTANDO CONEXÃO	RESPOSTA ESPERADA
LOC\$DESC	SET\$LINK\$OFF	LINK\$IS\$DOWN		ETD ESTÁ DESCONECTADO E RECUSA REINICIAÇÃO	ENVIAR SET\$LINK\$ON
COL\$U\$DIF\$UP	SET\$LINK\$ON	LINK\$IS\$DOWN	LINK\$IS\$DOWN	ETD ACEITOU DESCONEXÃO	RESPOSTA ESPERADA
REM\$CONEC			LINK\$IS\$UP	COLISÃO DE CMD U DIFERENTES COM ECD (RE) INICIANDO O ENLACE	RETENTAR INDEFINIDAMENTE
LOC\$RESET			LINK\$IS\$UP		ENVIAR SET\$LINK\$ON
LOC\$N2\$SABM			LINK\$IS\$DOWN	ETD (RE) INICIOU	RESTART
REM\$DISC	SET\$LINK\$ON	LINK\$IS\$DOWN	LINK\$IS\$DOWN	ECD(N2) REINICIOU	RESTART
FR\$REJ			LINK\$IS\$DOWN	ECD TENTOU (RE) INICIAR	ENVIAR SET\$LINK\$ON
LOC\$N2\$REJ			LINK\$IS\$DOWN	N2 VEZES COMUNICAÇÃO INTERROMPIDA	
SOFT\$ERROR			LINK\$IS\$DOWN	ETD COMANDOU DESCONEXÃO	ENVIAR SET\$LINK\$ON
PROT\$BREAK			LINK\$IS\$DOWN	ETD ENVIOU FRMR	OBS.1
			LINK\$IS\$DOWN	ECD TENTOU TIMEOUT RECOVERY OU RECUPERAÇÃO DE SEQUENCIA N2 VEZES	OBS.1
REM\$N2\$REJ			LINK\$IS\$DOWN	ERRO DE SOFT NO ECD	OBS.1
C\$INV			LINK\$IS\$DOWN	RECEPÇÃO DE RESPOSTA UA OU DM NÃO SOLICITADA OU REJ COM NR = VS	OBS.1
LENG\$REJ			LOC\$TX\$FRMR	ETD TENTOU RECUPERAÇÃO DE SEQUENCIA N2 VEZES	OBS.1
NR\$INV			LOC\$TX\$FRMR	ECD ENVIOU FRMR	OBS.1
LOC\$N2\$FRMR			LOC\$TX\$FRMR	ECD ENVIOU FRMR	OBS.1
LOC\$N2\$DISC	SET\$LINK\$OFF	LINK\$IS\$DOWN	LINK\$IS\$DOWN	ECD ENVIOU FRMR N2 VEZES	OBS.1
				ECD TENTOU DESCONEXÃO N2 VEZES COMUNICAÇÃO INTERROMPIDA.	RESPOSTA ESPERADA

OBS.1 - O N3 não deverá mandar b N2 até receber as indicações LINK\$IS\$UP: (LOC\$RESET), LINK\$IS\$DOWN(LOC\$N2\$SABM, REM\$DISC, REM\$DM, COL\$U\$DIF\$UP) ou LINE\$IS\$DOWN(CTS\$NOK, DCD\$NOK)

FIGURA C.1 - CÓDIGOS DE RETORNO AO USUÁRIO DO N2