



DEPARTAMENTO DE
ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

**FERRAMENTAS E MÉTODOS
PARA A AVALIAÇÃO
DO DESEMPENHO DE REDES
DE COMUNICAÇÃO**

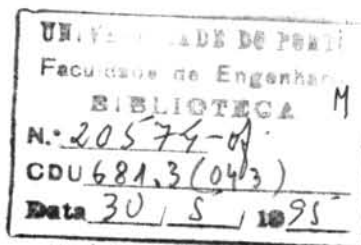
FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Rua dos Bragas, 4099 Porto Codex – PORTUGAL

hys. 760
Elet.

PEDRO JOÃO DE-FRANCESCO RESENDE FORTUNA ASSIS

FERRAMENTAS E MÉTODOS
PARA A AVALIAÇÃO
DO DESEMPENHO DE REDES
DE COMUNICAÇÃO



N.º 34570

Dissertação submetida à Faculdade de Engenharia da Universidade do Porto
para satisfação parcial dos requisitos do programa do Curso de Mestrado

em

Engenharia Electrotécnica e de Computadores
(Área de especialização de Informática Industrial)

Abril de 1994

043 M
A 866 f
lx. 2

Dissertação realizada sob a supervisão do
Prof. Doutor Eurico Manuel Elias de Moraes Carrapatoso,
Professor Auxiliar do Departamento de Engenharia Electrotécnica e de Computadores
da Faculdade de Engenharia da Universidade do Porto.

10/10/2011

À Graça e ao Diogo

Resumo

A avaliação do desempenho de sistemas de comunicação tem por objectivo não só melhorar o serviço que este oferece, como também fornecer pistas para as partes que deverão ser incluídas por forma a aumentar a sua funcionalidade. Devido às limitações das técnicas utilizadas na modelização dos sistemas de comunicações, modelos analíticos e simulações, é necessário realizar medidas durante a fase de desenvolvimento dos protótipos destes sistemas por forma a optimizar, e validar, as opções tomadas.

Conjugando as necessidades e objectivos descritos com a evolução que actualmente se verifica na área das redes de comunicações, surgiu a ideia de conceber uma ferramenta de teste cuja arquitectura lhe conferisse funcionalidades particulares. Os conceitos envolvidos na definição de tal arquitectura foram aplicados na construção de um sistema de teste denominado *Sistema Gerador e Detector de Tráfego* (SGDT), que foi desenvolvido no âmbito de dois projectos ESPRIT II (*European Strategic Programme for Research in Information Technology*) na área das redes de comunicação empresariais.

O modo de transferência assíncrono (ATM) apresenta-se como o meio de transmissão para as redes digitais com integração de serviços de banda larga (RDIS-BL), existindo, durante a fase de normalização que decorre, um grande interesse por parte da comunidade científica no estudo do desempenho que esta técnica fornecerá às redes que a usem. Desta forma é natural a migração do SGDT para ambientes ATM o que levou à realização de estudos de modelos analíticos conducentes à previsão do desempenho de redes baseadas nesta técnica. Assumiram-se como elementos chave nesta análise os comutadores e multiplexadores ATM, sendo apresentados estudos, baseados em modelos analíticos, para avaliar o desempenho esperado (teórico) de um comutador e de um multiplexador ATM, os resultados obtidos são relativos a um comutador ATM (16x16(c)).

Palavras-chave: Arquitecturas de Ferramentas de Teste; Sistema Gerador e Detector de Tráfego; Qualidade de Serviços; Modelos Analíticos; Filas de Espera; Modo de Transferência Assíncrono.

Abstract

There are two basic reasons for the time consumed in carrying out performance analysis: firstly it is necessary to improve the quality of services offered, and, secondly, to add functionality to the system. Due to the limitations of the analytical models and simulations techniques, it is important to build prototypes and thoroughly analyse them.

The conformance tests and the performance evaluation of high speed networks pose problems that cannot be solved by traditional equipment. In this work the concept of a powerful and versatile architecture is described, together with the model of a system capable of carrying out these tasks together and with an actual implementation of such a system. This system, which was called the *Traffic Generator/Detector System*, was developed with support of two ESPRIT II (European Strategic Programme for Research in Information Technology) projects.

Keeping in mind the migration of the TGDS towards the Asynchronous Transfer Mode (ATM) based networks, some theoretical studies were developed, which enable a better understanding of the requirements imposed to a test tool for such environments. Those studies include the analysis, using analytical methods, of the theoretical performance offered by an ATM switch and also an ATM multiplexer. Finally, some results are presented for the case study of an ATM switch (16x16(c)).

Key-words: Model of Test Tools; Traffic Generator/Detector System; Quality of Services; Analytical Models; Queueing Theory; Asynchronous Transfer Mode.

Índice de Texto

1	Introdução	1
1.1	Enquadramento	1
1.2	Descrição sumária dos capítulos	4
2	Uma Arquitectura para Ferramentas de Teste de Redes com Elevado Débito	5
2.1	Introdução	5
2.2	Princípio de funcionamento	6
2.3	Requisitos em termos de <i>hardware</i>	8
2.3.1	Arquitectura	8
2.3.2	Porquê um DSP?	9
2.4	Requisitos em termos de <i>software</i>	13
2.4.1	Funcionalidades	13
2.4.2	Métodos e ferramentas	14
2.5	Apresentação de alguns equipamentos de teste	14
2.5.1	<i>Fiber Distributed Data Interface - FDDI</i>	15
2.5.2	Uma ferramenta de teste Para redes FDDI-II: DA 30	16
2.5.3	Modo de transferência assíncrono - ATM	18
2.5.4	Uma ferramenta desenvolvida no âmbito do projecto RACE 1083 - PARASOL	19
3	SGDT - Sistema Gerador e Detector de Tráfego	21
3.1	Introdução	21
3.2	Apresentação do projecto DAMS	21
3.2.1	LSU - <i>Local Switch Unit</i>	22
3.2.2	BBU - <i>Backbone Unit</i>	25
3.2.3	SSCU - <i>Sub-System Control Unit</i>	26
3.3	SGDT - conceitos fundamentais	26
3.3.1	Conceito de sessão	27
3.3.2	Conceito de conexão	28
3.3.3	Avaliação do desempenho	28
3.3.3.1	Tráfego de pacotes	28

3.3.3.2	Tráfego de controlo	29
3.3.4	Funções de gestão	29
3.3.5	Monitoração	30
3.4	Arquitectura e modo de funcionamento de uma UPT	30
3.4.1	Unidade mestre - CPU 68030	30
3.4.2	Unidade escravo bloco servidor - DSP TMS320C25	31
3.4.2.1	Interface ao barramento VME	31
3.4.2.2	Núcleo do servidor	34
3.4.3	Unidade escravo bloco de interface ao DAMS	36
3.4.4	Aspectos tecnológicos relativos aos circuitos impressos realizados	37
3.5	Geração e detecção de tráfego de pacotes	38
3.5.1	Procedimentos para a geração de tráfego de pacotes	38
3.5.2	Procedimentos para a detecção de tráfego de pacotes	39
4	Modelos Analíticos para o Projecto e Optimização de Redes de Comunicação	43
4.1	Introdução	43
4.2	Processos estocásticos	44
4.2.1	Processos estacionários	44
4.2.2	Processos independentes	45
4.2.3	Processos de <i>Markov</i>	45
4.2.4	Caminhos aleatórios	45
4.2.5	Processos de renovação	46
4.3	Cadeias de <i>Markov</i>	46
4.3.1	Cadeias de <i>Markov</i> de tempo discreto estacionárias	46
4.3.2	Coefficiente egórdico	49
4.3.3	Cadeias de <i>Markov</i> de tempo contínuo	51
4.3.4	Análise computacional	52
4.4	Algumas noções necessárias para o estudo de filas de espera	54
4.4.1	Distribuições sem memória	54
4.4.2	Funções características e funções geradores de momentos	55
4.4.3	Aplicação de transformadas na determinação dos momentos	56
4.5	Sistemas de fila de espera simples	57
4.5.1	Filas de espera do tipo nascimento-morte em equilíbrio	58
4.5.2	Filas de espera do tipo: $M/M/m/K/M$	59
4.5.2.1	Filas de espera do tipo: $M/M/m/m$	60
4.5.2.2	Filas de espera do tipo: $M/M/\infty/M$	61
4.5.3	Filas de espera do tipo <i>Markov</i> em equilíbrio	61
4.5.4	Filas de espera do tipo: $M/E_r/1$	63

4.5.5	Filas de espera do tipo: $E_T/M/1$	64
4.5.6	Relação entre as filas de espera do tipo nascimento-morte e as baseadas em processos de <i>Markov</i>	65
5	Avaliação do Desempenho de Redes RDIS-BL/ATM	67
5.1	Introdução	67
5.2	Características do desempenho de uma ligação ATM	68
5.2.1	Transparência temporal	69
5.2.2	Transparência semântica	71
5.3	Mecanismos de controlo de situações de congestão em ATM	73
5.3.1	Modelo de referência adoptado pelo CCITT	74
5.3.2	Funções e procedimentos realizados pelo NRM	77
5.3.3	Funções e procedimentos realizados pelo CAC	77
5.3.4	Funções e procedimentos realizados pelo UPC/NPC	78
5.4	Avaliação do desempenho de ligações ATM usando os mecanismos de gestão em RDIS-BL	79
5.5	Tradução da QoS em parâmetros de desempenho do RDIS-BL/ATM	80
5.6	Caracterização de elementos da RDIS-BL/ATM	83
5.6.1	Avaliação do desempenho de um comutador ATM	84
5.6.1.1	Comprimento da fila de espera	85
5.6.1.2	Atraso experimentado pelo fluxo de células na fila de espera	87
5.6.1.3	Caso de estudo: comutador ATM (16x16(c))	88
5.6.1.4	Referências a outros métodos	91
5.6.2	Análise de um multiplexador ATM	92
6	Conclusões	95
	Referências	97
Apêndice A	Esquemáticos Relativos aos Circuitos Impressos Realizados	103
Apêndice B	Análise Computacional de Cadeias de <i>Markov</i> Finitas, de Índice Discreto e Estacionárias	129
Apêndice C	Determinação do Gráfico Relativo ao Atraso Médio de um Comutador ATM (16x16(c))	133

Índice de Figuras

2	Uma Arquitectura para Ferramentas de Teste de Redes com Elevado Débito	
2.1	Arquitectura de uma UPT.	7
2.2	Unidade MAC (genérica) de um CPU de 32 <i>bits</i> de vírgula flutuante.	12
2.3	Modelo OSPA.	12
2.4	A norma FDDI-II especifica três subcamadas que correspondem aos dois níveis inferiores do modelo OSI. Também é descrita a função SMT, que tem por objectivo a supervisão das restantes subcamadas.	16
2.5	Classes de estações no âmbito de uma rede FDDI.	17
2.6	Diagrama funcional da interface do DA 30.	18
2.7	Relação dos parâmetros NP com o modelo de referência protocolar ATM.	20
3	SGDT - Sistema Gerador e Detector de Tráfego	
3.1	Arquitectura do projecto DAMS.	23
3.2	Arquitectura da LSU.	23
3.3	Estrutura interna da LSI.	25
3.4	Diagrama de blocos da BBU.	26
3.5	Exemplos de configurações para a interligação do SGDT ao DAMS.	28
3.6	Diagrama funcional da interface (construída) ao VME.	32
3.7	Máquina de estados que controla a transferência de dados no DTB.	33
3.8	Máquina de estados relativa ao funcionamento do gerador de interrupções.	34
3.9	Formato do pacote SGDT.	37
3.10	Fluxograma relativo aos procedimentos da geração de pacotes.	38
3.11	Fluxograma relativo aos procedimentos da detecção de pacotes.	39
3.12	Circuito impresso do bloco servidor da entidade escravo.	41
3.13	Circuito impresso do bloco de interface da entidade escravo.	41
4	Modelos Analíticos para o Projecto e Optimização de Redes de Comunicação	
4.1	Grafo de transição de estados de um processo nascimento-morte.	52
4.2	Fluxograma descritivo do método soma-coluna.	53
4.3	Diagrama para a análise de cadeias de <i>Markov</i> finitas de tempo discreto estacionárias.	55

4.4	Esquema de funcionamento de um sistema baseado numa fila de espera com m servidores.	59
4.5	Servidor de Erlang com duas etapas (E_2).	62
4.6	Grafo de um sistema $M/E_r/1$.	63
4.7	Grafo de um sistema $E_r/M/1$.	64
4.8	Grafo de um sistema que permite introduzir conjuntos de clientes, de dimensão variável, em instantes de <i>Poisson</i> .	65
5	Avaliação do Desempenho de Redes RDIS-BL/ATM	
5.1a	Atrasos numa rede ATM (AE- Atraso de Empacotamento; AT- Atraso de Transmissão; AC- Atraso na Comutação; AD- Atraso de Desempacotamento).	70
5.1b	Atrasos numa rede que não é baseada em ATM (AE- Atraso de Empacotamento; AT- Atraso de Transmissão; AC- Atraso na Comutação; AD- Atraso de Desempacotamento).	70
5.2	Consequências dos erros no cabeçalho de uma célula ATM.	72
5.3a	Efeito das características do tráfego sobre a probabilidade de bloqueio de um multiplexador, efeito da intensidade de tráfego.	74
5.3b	Efeito das características do tráfego sobre a probabilidade de bloqueio de um multiplexador, efeito do valor médio da duração dos <i>bursts</i> de tráfego.	74
5.3c	Efeito das características do tráfego sobre a probabilidade de bloqueio de um multiplexador, efeito da distribuição da duração dos <i>bursts</i> de tráfego.	74
5.4	Configuração de referência para o controlo de tráfego e congestão numa rede ATM.	75
5.5	Configuração de referência e terminal equivalente para a definição do valor de pico do ritmo de transmissão numa ligação ATM.	76
5.6	Estabelecimento de uma ligação usando o conceito de VP.	78
5.7	Cenários possíveis para a utilização da capacidade de <i>loopback</i> .	81
5.8	Modelo hierárquico para a tradução dos parâmetros QoS num ambiente RDIS-BL/ATM.	81
5.9	Representação funcional de um comutador ATM ($4 \times 4(2)$).	85
5.10	Ocupação média total do comutador ATM ($16 \times 16(c)$).	89
5.11	Variância do tráfego injectado nas filas de espera do comutador ATM ($16 \times 16(c)$) ($V[A]$).	89
5.12	Ocupação média total (medida em células) do comutador ATM ($16 \times 16(c)$).	90
5.13a	Atraso médio numa (qualquer) fila de espera do comutador ATM ($16 \times 16(c)$).	90
5.13b	Valor médio do atraso (medido em <i>slots</i>) no comutador ATM ($16 \times 16(c)$).	91
5.14	Variância do atraso experimentado pelo fluxo de células injectado no comutador ATM ($16 \times 16(c)$).	91
Apêndice B	Análise Computacional de Cadeias de Markov Finitas, de Índice Discreto e Estacionárias	
B.1	Grafo orientado da cadeia de Markov em análise.	130

Índice de Tabelas

2	Uma Arquitectura para Ferramentas de Teste de Redes com Elevado Débito	
2.1	Processadores de vírgula flutuante: RISC vs DSP.	11
3	SGDT - Sistema Gerador e Detector de Tráfego	
3.1	Características da unidade mestre.	31
3.2	Modificadores de endereço utilizados.	33
3.3	Algumas características do TMS320C25.	35
5	Avaliação do Desempenho de Redes RDIS-BL/ATM	
5.1	Atraso (μs) para diferentes comprimentos da informação e velocidades de transmissão. Estes valores foram obtidos com a ligação a suportar uma carga igual a 80% da sua capacidade, e com uma probabilidade de perdas igual a 10^{-10} . A distância entre o nó de acesso e de destino é igual a 1000 km; assume-se que o atraso total na rede síncrona é igual a 900 μs , e que um total de 8 comutações de VP/VC foram realizadas.	71
5.2	Funções relativas ao controlo de tráfego e de congestão.	77
Apêndice C	Determinação do Gráfico Relativo ao Atraso Médio de um Comutador ATM (16x16(c))	
C.1	Raízes da equação ($z^c - A(z)$), para c igual a 2 e 4.	135

Lista de Abreviaturas

ALL	<i>ATM Adaptation Layer</i>
ANSI	<i>American National Standards Institute</i>
ARQ	<i>Automatic Repeat Request</i>
ATM	<i>Asynchronous Transfer Mode</i>
BBU	<i>Backbone Unit</i>
BOD	<i>Bandwidth On Demand</i>
CAC	<i>Connection Admission Control</i>
CATV	<i>Community Antenna TV</i>
CCITT	<i>Comité Consultatif International pour le Télégraphe et le Téléphone</i>
CDV	<i>Cell Delay Variation</i>
CEQ	<i>Customer Equipment</i>
CL	<i>Convergence Layer</i>
CLP	<i>Cell Loss Priority</i>
CPU	<i>Central Processing Unit</i>
CRC	<i>Cyclic Redundancy Check</i>
DAMS	<i>Dynamically Adaptable Multi-service System</i>
DAS	<i>Dual Attachment Station</i>
DBMAP	<i>Discrete-time Batch Markov Arrival Process</i>
DMA	<i>Direct Memory Access</i>
DSP	<i>Digital Signal Processor</i>
EDD	<i>Earliest-Due-Date</i>
ESPRIT	<i>European Strategic Programme for Research in Information Technology</i>
ETSI	<i>European Telecommunications Standards Institute</i>
FC	<i>Frame Control</i>
FDDI	<i>Fiber Distributed Data Interface</i>
FDM	<i>Frequency Division Multiplexing</i>
FIFO	<i>First In First Out</i>
FDP	<i>Função Distribuição de Probabilidade</i>
fdp	<i>Função Densidade de Probabilidade</i>
FPU	<i>Floating Point Unit</i>
GFC	<i>Generic Flow Control</i>

H-MAC	<i>Hybrid Media Access Control</i>
H-MUX	<i>Hybrid Multiplexer</i>
HEC	<i>Header Error Control</i>
HRC	<i>Hybrid Ring Controller</i>
I-MAC	<i>Isochronous Media Access Control</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IIS	<i>Internal Isochronous Switch</i>
IMC	<i>Imbedded Markov Chain</i>
ISCL	<i>Internal Signal & Control Layer</i>
ISDN	<i>Intregated Services Digital Network</i>
ISO	<i>International Standardization Organization</i>
LAN	<i>Local Area Network</i>
LBC	<i>Local Bus Control</i>
LLC	<i>Logic Link Control</i>
LSC	<i>Local Switch Control</i>
LSI	<i>Local Switch Interface</i>
LSU	<i>Local Switch Unit</i>
MAC	<i>Medium Access Control</i>
MACC	<i>Multiply-and- ACCumulate</i>
MAN	<i>Metropolitan Area Network</i>
MAX	<i>Multiple Array Matrix</i>
MMI	<i>Man-Machine Interface</i>
MMPP	<i>Markov Modulated Poisson Process</i>
NI-MAC	<i>Non-Isochronous Media Access Control</i>
NNI	<i>Network Node Interface</i>
NP	<i>Network Performance</i>
NPC	<i>Network Parameter Control</i>
NRM	<i>Network Resource Management</i>
NRZI	<i>Non Return to Zero Inverted</i>
OAM	<i>Operation And Maitenance</i>
OSI	<i>Open Systems Interconnection</i>
OSPA	<i>Open-Signal Processing Architecture</i>
P-MAC	<i>Packet Media Access Control</i>
PDU	<i>Protocol Data Unit</i>
PHY	<i>Physical Layer Protocol</i>
PHY	<i>Physical sub-layer</i>
PLD	<i>Programmable Logic Device</i>
PMD	<i>Physical Medium Dependent</i>

PORDOS	<i>Portable Real Time Distributed Operating System</i>
PT	<i>Payload Type</i>
PU	<i>Port Unit</i>
PVC	<i>Permanent Virtual Connection</i>
QoS	<i>Quality of Service</i>
RACE	<i>Research in Advanced Communications technologies for Europe</i>
RBL	Rede de Banda Larga
REDIS-BL	Rede Digital com Integração de Serviços de Banda Larga
RISC	<i>Reduced Instruction Set Computer</i>
ROAK	<i>Release on Acknowledge</i>
SAP	<i>Service Access Point</i>
SAPI	<i>Service Access Point Identifier</i>
SAR	<i>Segmentation and Reassembly</i>
SAS	<i>Single Attachment Station</i>
SDH	<i>Synchronous Digital Hierarchy</i>
SDL	<i>Specification and Description Language</i>
SGDT	Sistema Gerador Detector de Tráfego
SII	<i>Standard Internal Interface</i>
SMT	<i>Station Management</i>
SSCU	<i>Sub-System Control Unit</i>
STD	<i>Source Traffic Descriptor</i>
SVCC	<i>Signalling Virtual Channel Connection</i>
TDM	<i>Time Division Multiplexing</i>
UNI	<i>User Network Interface</i>
UPC	<i>Usage Parameter Control</i>
UPT	Unidade de Processamento de Tráfego
VCAS	<i>VCOS Application Server</i>
VCC	<i>Virtual Channel Connection</i>
VCI	<i>Virtual Channel Identifier</i>
VCOS	<i>Visible Caching Operating System</i>
VLSI	<i>Very Large Systema Integration</i>
VME	<i>Versa Module Europe</i>
VPC	<i>Virtual Path Connection</i>
VPI	<i>Virtual Path Identifier</i>

Capítulo 1

Introdução

1.1 Enquadramento

A tecnologia da informação assume-se actualmente como uma indústria florescente devido à crescente dependência da sociedade da troca de elevadas quantidades de informação, sendo necessário que esta se realize em tempo útil.

Diversas desvantagens estão associadas à especialização das redes de comunicação actuais: são dependentes do serviço que oferecem, são inflexíveis, ou seja, não estão preparadas para o aparecimento de novos serviços, e são ineficientes.

A evolução das redes de comunicação está a passar pela integração de serviços que eram, e ainda são em diversos casos, suportados por diferentes tipos de tecnologias e meios de comunicação. Exemplos dessa especialização são, no domínio público, as redes telefónicas, a transmissão de dados baseada em protocolos X.25/X.21 e as redes de transmissão de sinais de televisão, quer por satélite ou por cabo (CATV). Relativamente ao domínio privado, as redes locais apresentam-se como o meio de comunicação (de dados) mais difundido.

Um dos aspectos mais relevantes inerentes à evolução que se efectua é a integração de serviços que tradicionalmente usavam diferentes formas de comutação - comutação de circuitos e pacotes. Outros aspectos estão actualmente a merecer a atenção dos organismos de normalização, entre os quais se destaca a tendência de serem fornecidos às futuras redes de comunicação maior fiabilidade e facilidade na adaptação a novas normas, minimizando os custos de operação e manutenção.

A digitalização das redes telefónicas actuais prevê-se que irá tornar irrelevante o tipo de informação a transmitir, fornecendo a infra-estrutura para novos serviços e oferecendo uma maior conectividade entre diferentes sistemas.

A tendência a que se está a assistir no desenvolvimento das infra-estruturas da próxima geração será consequência dos avanços tecnológicos em diversas áreas (de que os VLSIs e a tecnologia óptica são exemplos importantes), que permitem uma maior disponibilidade de banda e um acréscimo considerável nas cadências de transmissão. Como consequência desta elevada capacidade, serão capazes de fornecer ao utilizador um novo conjunto de facilidades envolvendo componentes de vídeo/áudio, tais como distribuição de TV, teleconferência ou apli-

cações multimédia. Do ponto de vista do cliente, espera-se que estes novos serviços apresentem baixos custos devido à elevada capacidade disponível (custo por *bit* de informação).

Inicialmente as redes locais (LANs) caracterizavam-se por permitir o estabelecimento de comunicações entre utilizadores com um atraso na ordem dos *ms*, cobrindo uma área geográfica não superior a 10 km². Com a introdução do conceito de rede metropolitana (MAN) é possível abranger áreas geográficas superiores com velocidades de transmissão superiores às apresentadas pelas LANs.

As rede locais e as redes metropolitanas existentes não têm a capacidade exigida pelos novos serviços; assim, é necessária uma evolução no sentido de fornecer meios para suportar serviços que se prevê estarão disponíveis num futuro próximo. É neste âmbito que as redes de banda larga (RBL) terão de ser concebidas. A adopção pelo CCITT do modo de transferência assíncrono (ATM) irá determinar a base da arquitectura destas redes.

O ATM permite transmitir qualquer tipo de serviço independentemente das suas características, o que o torna ideal para ser usado como modo de transferência para a rede digital com integração de serviços (RDIS) de banda larga (RDIS-BL). Supõe-se que a curto prazo a RDIS-BL/ATM seja utilizada para interligar diferentes núcleos servidos por MANs. Na relação MAN - RDIS-BL, esta última irá desempenhar funções de encaminhamento, processamento e gestão das ligações estabelecidas. A interligação das MANs através da RDIS-BL/ATM poderá ser realizada em bases semi-permanentes.

Na concepção das arquitecturas das RBLs, é importante estudar as influências das opções tomadas tanto na qualidade de serviços oferecida ao utilizador (medida por este através de avaliações qualitativas), como no desempenho da rede de comunicação (atributos da ligação, ou seja, medidas quantitativas). No que diz respeito a este último aspecto, a avaliação dos atributos de uma ligação é tipicamente realizada em determinados elementos funcionais empregues na realização de um determinado serviço. A existência de duas definições concorrentes para avaliar a qualidade de serviço é necessária para caracterizar o comportamento quer do ponto de vista do utilizador (de quem é servido), quer do ponto de vista do sistema de comunicação (de quem serve).

Assim, é necessário, na fase de concepção das novas infra-estruturas de comunicações, recorrer à elaboração de *simulações* ou a *modelos analíticos* para antever a qualidade de serviços esperada, com o objectivo de obter resultados que facilitem a comparação de soluções alternativas no que respeita à arquitectura, aos protocolos, à sinalização e às tarefas de gestão.

A utilização de modelos analíticos será aconselhável no caso de ser possível obter uma fórmula fechada que descreva o funcionamento do sistema a modelizar, sem que tal implique aproximações grosseiras na descrição do mesmo. O que se verifica em grande parte dos casos é que estes são simplificações da realidade para que seja possível o tratamento matemático, não sendo possível garantir a exactidão dos resultados obtidos para situações reais.

No que respeita às simulações, é possível descrever os objectos em estudo com grande detalhe (minimizando as simplificações introduzidas), recorrendo ao uso de linguagens apropriadas, mas à custa de um elevado tempo de processamento.

Apesar da realização de estudos prévios, baseados em qualquer (ou ambos) dos procedimentos descritos, ser de grande utilidade na fase de concepção do sistema, estes não são suficientes. É necessário que a par da introdução de novas soluções para os sistemas de comunicação sejam efectuados desenvolvimentos no âmbito das ferramentas que irão permitir o teste destes sistemas: são exigidos instrumentos de teste cada vez mais poderosos e flexíveis por forma a tornar mais produtiva a fase de desenvolvimento, teste e validação dos protótipos. Por teste entende-se a verificação da funcionalidade (testes de conformidade) e a avaliação do desempenho de um determinado sistema, ou de módulos deste.

É neste âmbito que se desenvolve o trabalho descrito no presente texto, ou seja: na descrição de *modelos analíticos*, cuja aplicação forneça pistas para a optimização de aspectos relacionados com a arquitectura e dimensionamento dos elementos que compõem a rede de comunicação, e na apresentação de *ferramentas*, que permitam posteriormente a execução de tarefas relativas ao teste de conformidade e de desempenho de protótipos construídos.

Relativamente às ferramentas apresentados nos próximos capítulos a ênfase será colocada numa em particular, que foi desenvolvida com a colaboração do autor deste texto e foi denominada *Sistema Gerador e Detector de Tráfego* (SGDT). O SGDT surgiu no âmbito de dois projectos ESPRIT II, DAMS (*Dynamically Adaptable Multi-Service System*) e UCOL (*Ultra Wideband Coherent Optical LAN*), ambos baseados em redes ópticas de alto débito.

O projecto UCOL [Capol90] teve por objectivo a realização de uma RBL para ambientes industriais e comerciais, baseada numa estrela passiva que faz uso de técnicas coerentes. Tendo como requisito chave a flexibilidade, a arquitectura do sistema usava transmissão óptica multicanal, tendo sido tomadas as seguintes opções: gestão distribuída, modo de transferência assíncrono, protocolo de acesso ao meio eficiente para qualquer tipo de tráfego.

Quanto ao projecto DAMS, teve por finalidade o desenvolvimento de um sistema de comunicação para o mercado privado baseado num anel duplo, do tipo FDDI-II (*Fiber Distributed Data Interface*). A construção de um protótipo teve por objectivo demonstrar as seguintes capacidades: uso flexível da banda disponível (reserva dinâmica dos recursos), integração de diversos tipos de serviços (baseados em voz e dados), elevada fiabilidade (gestão eficiente e elevada robustez a falhas), adaptabilidade a futuros serviços, sistema aberto para suportar a evolução para o ATM e baixos custos inerentes ao desenvolvimento de novas aplicações.

1.2 Descrição sumária dos capítulos

No capítulo 2 é apresentada uma arquitectura de referência adoptada para sistemas de teste. Estes são definidos como um conjunto de unidades de processamento de tráfego (UPTs), que executam acções que contribuem para um fim comum. Estas acções podem visar funções como por exemplo testes de conformidade, avaliação do desempenho, monitoração e gestão dos recursos disponíveis. São apresentados os requisitos de uma UPT, quer do ponto de vista da arquitectura quer das suas necessidades em termos de *hardware* e *software*. Utilizando o modelo hierárquico mestre/servidor são descritos os princípios funcionais de uma UPT e identificadas as tarefas atribuídas a cada elemento desta unidade. Termina-se este capítulo com a descrição sucinta de alguns equipamentos existentes neste domínio.

No capítulo 3, descreve-se o projecto DAMS, apresentando os seus objectivos e, conseqüentemente, os requisitos e a especificação do SGDT. A constituição do protótipo desenvolvido pelo projecto é apresentada através da descrição da funcionalidade dos seus blocos principais. A descrição do SGDT é efectuada através do estudo da sua arquitectura e modos de funcionamento, sendo analisada, com algum pormenor, a funcionalidade de uma UPT e apresentados os circuitos impressos realizados, assim como o pacote de *software* desenvolvido.

O capítulo 4 descreve algumas das ferramentas matemáticas utilizadas habitualmente para o projecto e optimização de redes de comunicação. Neste contexto são descritos os *processos estocásticos* como sendo uma vertente da matemática utilizada massivamente na análise do comportamento dos sistemas de comunicação. É colocado um ênfase particular no estudo dos processos de *Markov* devido à sua vasta aplicabilidade no âmbito da modelação das redes de comunicação, sendo apresentado um programa que permite a análise de cadeias de *Markov*. Seguidamente são descritos alguns sistemas de filas de espera simples em equilíbrio. Não se apresentam filas de espera mais complexas, que utilizariam processos estocásticos genéricos para descrever o intervalo entre clientes ou de tempo de serviço, devido ao facto de posteriormente (capítulo 5) se efectuar a análise de um sistema discreto de filas de espera que faz uso destes.

No Capítulo 5 são apresentados alguns aspectos relacionados com a avaliação do desempenho de redes ATM. Inicialmente são descritos aspectos relativos à admissão de novas ligações assim como ao controlo das existentes. Este estudo permitirá determinar os mecanismos utilizados com o intuito de evitar situações de congestão assim como violações do contrato realizado entre o utilizador e a rede na fase de estabelecimento das ligações. Também são abordados os mecanismos de gestão das redes ATM, tendo em vista a possibilidade destes suportarem meios para executar a avaliação do desempenho das ligações existentes. Nesta área de avaliação do desempenho é apresentado um processo de tradução dos parâmetros de qualidade em parâmetros de desempenho da RDIS-BL/ATM. Conclui-se este capítulo com alguns modelos analíticos que permitem a análise de comutadores e multiplexadores ATM.

Por último, no capítulo 6, é efectuada a síntese das principais conclusões, indicando-se direcções possíveis para futuros desenvolvimentos.

Capítulo 2

Uma Arquitectura para Ferramentas de Teste de Redes com Elevado Débito

2.1 Introdução

A previsão da utilização generalizada de novos serviços e a inovação tecnológica a que se assiste actualmente estão a condicionar a evolução dos sistemas de telecomunicações. De facto, as redes de comunicação futuras terão que apresentar elevados níveis de desempenho resultantes de diversos factores, tais como a necessidade de interligação de redes locais, a utilização de computadores com elevado poder de processamento (o que implica elevados níveis de tráfego gerado), o desenvolvimento de plataformas que sustentem as aplicações multimédia distribuídas.

Tendo em vista estes factores, os sistemas de telecomunicações têm que evoluir por forma a acompanhar essas necessidades. É neste âmbito que o trabalho efectuado pelas entidades de normalização como o CCITT (*Comité Consultatif International pour le Télégraphe et le Téléphone*) e o ETSI (*European Telecommunications Standards Institute*), entre outras, se desenrola. Normas relativas às MANs, SDH (*Synchronous Digital Hierarchy*) ou ATM, encontram-se já, ou quase, definidas e conseqüentemente produtos comerciais estão a surgir ou surgirão muito brevemente.

Neste contexto surge a necessidade de ferramentas que permitam o teste e validação de protótipos, elaborados com o intuito de analisar a viabilidade das opções tomadas. As funções deste tipo de ferramentas englobam os procedimentos a realizar no âmbito dos testes de conformidade (verificação da funcionalidade), da avaliação do desempenho e por último, na gestão dos recursos da rede de comunicação.

Na literatura a avaliação de um sistema de telecomunicações é baseada na medição dos parâmetros QoS (*Quality of Service*) e NP (*Network Performance*) [RACE91]. Os parâmetros QoS são de natureza *subjectiva* (as opiniões dos utilizadores são determinantes na sua caracterização), sendo necessário quantificá-los através da sua tradução para grandezas objectivas. Os valores destas grandezas, que caracterizam a QoS de um determinado serviço, são previamente planeados (antes do serviço ser activado) e posteriormente medidos (inferidos) em condições normais de funcionamento desse mesmo serviço. Estes dois aspectos da QoS podem catalogados ser como a *QoS planeada* e a *QoS obtida*.

Relativamente aos parâmetros NP, estes definem-se pela capacidade do sistema de telecomunicações (ou porção deste) de fornecer as funções necessárias ao estabelecimento de ligações entre os utilizadores em condições aceitáveis. A avaliação dos componentes da rede que contribuem para a qualidade de serviço prestada é efectuada pela realização de um conjunto de *medidas objectivas*, sobre parâmetros tais como o atraso médio (extremo-a-extremo), a variância do atraso médio ou a probabilidade de perdas.

Para avaliar estes parâmetros é necessário promover a troca de informação entre a ferramenta de teste e os módulos do sistema de telecomunicações cujo desempenho se pretende avaliar, o que implica a geração e a detecção de informação a cadências elevadas sob a forma de pacotes, relativa a mensagens RDIS, de gestão ou de outro qualquer tipo, por forma a criar cenários de teste.

A avaliação de desempenho poderá ser executada para cada tipo de tráfego (isócrono e não-isócrono), devendo os resultados obtidos serem apresentados sob a forma de medidas estatísticas. Os parâmetros medidos são consequência do tipo de serviço em análise, por exemplo: em aplicações que envolvam ligações que suportem tráfego de vídeo, a variação do intervalo do tempo entre pacotes (*jitter*) é um dos parâmetros que reflectem a qualidade desse serviço.

Neste capítulo é apresentada a arquitectura que foi adoptada para desenvolver ferramentas capazes de realizar as funções descritas, e os requisitos a que estas devem obedecer. Descreve-se ainda o princípio de funcionamento baseado na configuração de referência apresentada e conclui-se com a análise de duas ferramentas disponíveis para avaliar o desempenho de redes baseadas, respectivamente, num duplo anel FDDI-II (DA 30, da W&G Gmgh) e em redes ATM (projecto PARASOL, RACE 1083).

2.2 Princípio de funcionamento

A definição apresentada no capítulo anterior para sistemas de teste pressupõe a existência de diversas unidades idênticas (*UPTs escravo*) que executam um conjunto de acções (previamente estabelecidas) para alcançar um fim comum, e que são coordenadas por uma única UPT, que se denomina *UPT mestre*.

Como já foi referido, o elemento básico (UPT) da arquitectura proposta para um sistema de teste (Fig. 2.1) consiste num cartão mestre e num número variável de cartões escravos. A principal função atribuída ao mestre é a gestão da UPT; é o escravo (bloco servidor e unidade de interface) que executa funções relativas à geração e detecção de tráfego [Assis93].

O mestre é a entidade responsável, entre outras funções, pela programação das acções a executar pelos escravos, que se realiza em dois níveis: a programação de dispositivos físicos existentes no circuito impresso (por exemplo, registos de controlo), e a transferência de código executável para a memória de sistema do escravo.

O escravo é divisível em dois blocos com funcionalidades distintas que, podem pertencer ao mesmo circuito impresso ou não, denominadas *bloco servidor* e *unidade de interface*. O bloco servidor realiza acções ditadas

pela programação previamente executada pelo mestre e pode comandar directamente as funções realizadas pela unidade de interface. Por forma a aumentar o desempenho deste sistema não é aconselhável que o mestre controle directamente a unidade de interface, situação que só é recomendada durante a fase de inicialização da UPT (cuja responsabilidade é do mestre).

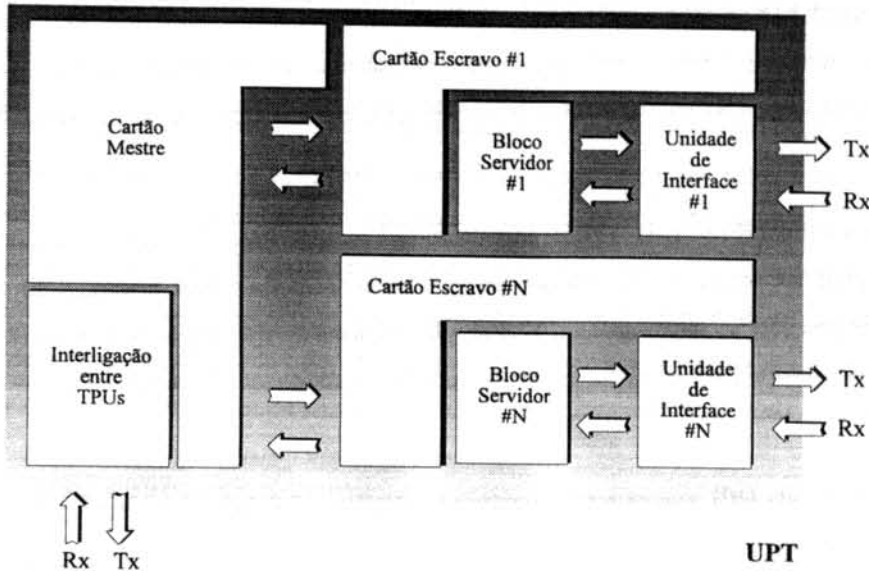


Figura 2.1 Arquitectura de uma UPT.

Durante a fase de inicialização o mestre executa alguns testes para determinar o número de cartões escravo em funcionamento, ou seja, verifica quantas unidades servidor/interface é que se encontram no barramento e respondem correctamente aos testes de inicialização. Supõe-se que o mestre é uma unidade auto-suficiente que não necessita de ajuda exterior para executar os procedimentos de inicialização.

As tarefas a executar após os testes de inicialização dependem da configuração actual (número de unidades escravo). Uma tarefa muito importante, mas que não é necessária no caso de só existir uma unidade escravo operacional, diz respeito à sincronização dos escravos. Este procedimento é determinístico pois todos as unidades escravos encontram-se ligadas ao mesmo barramento de ligação que, depende do mestre seleccionado. A sincronização reveste-se de interesse capital devido à necessidade de, em alguns casos existir na mensagem transmitida informação relativa ao seu tempo de geração (*time-stamp*). Só desta forma é possível medir algumas das grandezas que caracterizam a qualidade de serviços, como por exemplo o atraso experimentado pela mensagem na rede de comunicação.

Além dos testes executados pelo mestre às unidades escravo deve ser possível estabelecer um conjunto de funções de inicialização mais elaboradas (*built-in-self-tests*), que permitam o teste das unidades servidor/interface de uma forma mais exaustiva (o conjunto de testes a realizar podem ser programados pelo mestre). O diagnós-

tico dos cartões escravo é posteriormente comunicado ao mestre através de uma sequência de validação que é interpretada por este último.

É possível criar cenários de teste mais completos a partir da associação de várias UPTs em rede. A ligação entre as diversas unidades pode ser efectuada usando uma LAN (*Token-Ring*, *Ethernet*, etc) ou qualquer outra forma de interligação que permita cobrir a área geográfica pretendida não sendo o ritmo de transmissão importante. Neste tipo de configuração é necessário definir uma *UPT mestre*, isto é, uma unidade que estabeleça e coordene a sessão de testes, e que de preferência seja uma estação de trabalho com capacidades gráficas que permita a rápida configuração dos testes a executar através do uso de uma interface gráfica (MMI - *Man-Machine Interface*).

A possibilidade de interligar diversas UPTs em rede, sendo esta independente do próprio sistema de telecomunicações a testar, pode ser uma exigência na fase de teste de conformidade e de avaliação de desempenho. Quando se pretende proceder à monitoração ou a gestão dos recursos da rede, é possível, nesta fase, utilizar o próprio sistema como suporte para interligar as várias unidades (supõe-se que a fase de validação foi realizada com sucesso).

Relativamente à escolha da UPT mestre estão identificadas duas possibilidades: uma UPT pré-definida ou a utilização de um algoritmo de selecção entre as diversas UPTs candidatas. Este algoritmo pode ser baseado na transmissão de mensagens difundidas para todas as UPTs (*broadcast*). Desta forma cada UPT envia para o meio informação relativa à possibilidade de se assumir como *UPT mestre*, sendo depois a selecção efectuada através da análise do nível de prioridade indicado na mensagem transmitida e do instante de recepção da informação pelas diversas UPTs. A atribuição do nível de prioridade é consequência das potencialidades de cada UPT, como por exemplo: a UPT que suporta todo o conjunto de programas necessários para realizar e gerir uma sessão de testes deverá em princípio ter o nível de prioridade mais elevado.

Após a escolha da UPT mestre, esta é responsável pela transferência dos parâmetros que caracterizam os teste a realizar, pelo início e conclusão da sessão de testes e pela recolha dos resultados obtidos e pela sua representação gráfica utilizando a MMI.

2.3 Requisitos em termos de *hardware*

2.3.1 Arquitectura

As opções de cada UPT possuir diversos cartões escravo e de ser possível interligar diversas UPTs em rede têm reflexos na quantidade de tráfego que é possível gerar e detectar. Claramente cada escravo dentro de uma UPT pode ter funções distintas, ou seja, é possível programar um determinado escravo para só gerar tráfego com um dado perfil com o intuito de carregar a rede, enquanto que os restantes podem proceder às medições pretendidas, sendo o atraso máximo obtido ao estabelecer chamadas RDIS um exemplo.

A escolha do mestre terá que contemplar os aspectos relacionados com o suporte de vários cartões escravo e, por consequência, o sistema operativo terá que funcionar em tempo-real e ser multi-tarefa, com a possibilidade de suportar uma interface que permita a interligação de diversas UPTs em rede. A sua selecção deverá naturalmente atender à tecnologia disponível actualmente sendo as opções mais populares as baseadas em: VMEBus, VXI, Multibus II, EISA, S-Bus ou Futurebus+. Quando se refere um sistema em tempo-real é necessário ter cuidados adicionais com os mecanismos associados às interrupções: é exigido que estas tenham um tempo de serviço determinístico.

Uma condicionante no projecto do escravo é a elevada capacidade de cálculo requerida, ou seja, a necessidade de se dispor de uma máquina que tenha a possibilidade de realizar cálculos aritméticos em tempo útil: o processamento aritmético é fundamental para este tipo de aplicação, o que significa que é necessário utilizar um CPU que realize com eficácia cálculos matemáticos. Estes cálculos deverão ser executados durante os períodos de inactividade dos processos de geração/detecção, o que significa que o intervalo de tempo disponível para os executar depende da taxa de geração/detecção de tráfego, logo, para cargas elevadas este é reduzido. Dentro da gama de CPUs disponíveis encontram-se os DSPs (*Digital Signal Processor*), os quais parecem apresentar as características necessárias se se pretendem taxas elevadas.

2.3.2 Porquê um DSP?

Pode-se estabelecer um paralelo entre a evolução dos μ Ps e a dos DSPs. Inicialmente o desenvolvimento realizado destinava-se a criar arquitecturas que permitissem a resolução de um problema em particular. Esta fase pode-se chamar fase de especialização e caracterizou-se pela existência de uma grande gama de arquitecturas similares mas desenvolvidas para aplicações distintas.

Actualmente observa-se uma divisão do mercado em duas vertentes. A primeira resulta do desenvolvimento inicial dos DSPs e caracteriza-se por uma elevada produção a baixos custos: DSPs de vírgula fixa de 16 bits. A segunda vertente está a emergir sendo caracterizada por pequenas produções com custos elevados: DSPs de vírgula flutuante para processamento paralelo.

Os DSPs têm uma arquitectura adequada ao processamento de sinal, diferentes da dos μ Ps de uso genérico: as opções tomadas têm por consequência limitações ao nível das instruções disponíveis. Este facto irá criar dificuldades adicionais no desenvolvimento de programas, nomeadamente se está em causa um DSP da primeira geração. Na realidade os DSPs de vírgula fixa estão mais próximos das máquinas aritméticas do que dos μ Ps.

Um dos problemas tradicionais associados aos DSPs da primeira geração relaciona-se com as facilidades fornecidas para desenvolver aplicações a partir de linguagens de alto/médio nível, nomeadamente se se utilizar como referência a linguagem de desenvolvimento mais comum na família dos DSPs - Linguagem C.

Na realidade os processadores da primeira geração não foram projectados com o intuito de obter uma arquitectura que favorecesse o desenvolvimento de aplicações em C. Os compiladores existentes são ineficientes e sujeitos a um número elevado de erros. Este problema foi no entanto resolvido para a geração actual, de vórgula flutuante. DSPs como os TMS320C30/40 da *Texas Instruments* (TI) têm uma arquitectura que permite a utilização de ferramentas fiáveis para o desenvolvimento de aplicações usando a Linguagem C [Warre90].

Actualmente é cada vez mais importante avaliar o suporte disponível na escolha de um DSP: *assemblers*, simuladores, compiladores, *source-level debuggers* e circuitos de emulação. Existe já uma grande difusão de compiladores que procuram otimizar o código gerado. No entanto, e embora o nível de optimização esteja a aumentar, ainda não é possível realizar uma aplicação inteiramente baseada em linguagens de alto/médio nível, sendo as tarefas de gestão e controlo as que se prestam melhor a esse modo de implementação para os DSPs.

Outra possibilidade de incrementar a "quantidade" de programas escritos a partir de uma linguagem de alto/médio nível é melhorar a sua eficiência atendendo ao tipo das aplicações que são desenvolvidas. A *Analog Devices* propôs algumas extensões a introduzir na Linguagem C, denominada DSP/C, com o intuito de a transformar numa linguagem vectorial.

Os DSPs da segunda geração introduzidos recentemente combinam a capacidade de endereçamento disponível num μ P RISC (*Reduced Instruction Set Computer*) com a possibilidade de realizar cálculos com a eficiência que caracteriza os μ Ps vectoriais. A nova arquitectura destes DSPs fornece-lhes uma capacidade invulgar de paralelismo; como exemplo indica-se o TMS320C40 da TI, que está equipado com novas facilidades de comunicação que, lhe permitem estabelecer ligações com 6 CPUs TMS320C40 para troca de informação e coordenação até à velocidade máxima de 20 Mbps (arquitectura típica dos *Transputers*).

As características dos DSPs actuais permitem antever que estes serão a base das aplicações multimédia, especificamente realizando acções de co-processor. A concorrência que se apresenta aos DSPs vem dos μ Ps RISC, os quais têm características que favorecem a realização de aplicações de cálculo intensivo.

Entre os μ Ps RISC mais evoluídos apresenta-se no topo de gama o *Swordfish* desenvolvido pela *National Semiconductor* e o i860XP da *Intel*. Comparativamente aos DSPs, este último tem a vantagem de possuir um ciclo de instrução mais rápido embora não esteja equipado com os blocos funcionais necessários para o processamento de sinal em tempo-real que equipam o TMS320C40 ou o DSP96002 da Motorola. Na Tabela 2.1 estabelece-se uma comparação entre estes CPUs [Weiss91].

Encontram-se diferenças entre os DSPs e os μ Ps RISC ao nível das suas arquitecturas: os DSPs utilizam por norma uma arquitectura *Harvard* que lhes permite um acesso dual à memória. Uma das partes mais aperfeiçoadas da família dos DSPs é o bloco MACC (*multiply-and-accumulate*) (Fig. 2.2), que realiza ciclos aritméticos iterativos com elevado rendimento (operações deste tipo surgem, com bastante frequência, em técnicas de processamento de sinal). É também oferecido ao utilizador o processamento automático de controlo de ciclos re-

petitivos. Muitos dos DSPs têm blocos que efectuam este tipo de controlo de forma transparente para o utilizador, o qual não necessita de escrever código específico para esse fim.

Em termos da realização de aplicações baseadas em instruções disponíveis na família dos DSPs, existe um vasto conjunto de instruções simples que englobam um número elevado de operações (num único ou em vários ciclos de relógio) que facilita o desenvolvimento de programas que façam uso destas.

Tabela 2.1 Processadores de vírgula flutuante: RISC vs DSP.

i860XP (RISC)	DSP96002 & TMS320C40 (DSP)
<ul style="list-style-type: none"> • Suporta UNIX • Aritmética com 64 bits • Ciclo de instrução de 25 ns • Comutação de contexto lenta • Mecanismo de interrupção normal • Um barramento externo • Perda de desempenho ao aceder à memória externa • Memória <i>cache</i> sem retenção • FPU não suporta <i>pipeline</i> • Não suporta multiplicações com inteiros • Suporta gráficos 	<ul style="list-style-type: none"> • Não suporta UNIX, não têm MMU • Aritmética com 32 bits • Ciclo de instrução de 50 ns • Comutação de contexto rápida • Interrupções rápidas • Dois barramentos externos • Um ciclo de acesso sem perda de desempenho • Memória <i>cache</i> com retenção • FPU realiza operações num só ciclo • Realiza multiplicações com inteiros (32 bits) • Não suporta gráficos

Sendo os DSPs baseados em arquiteturas que fornecem múltiplos barramentos é possível realizar mais do que uma determinada acção por ciclo de instrução. O DSP96002 é capaz de executar operações em vírgula flutuante utilizando diversos registos e, em simultaneidade, realizar uma transferência de informação utilizando o canal de DMA (*Direct Memory Access*), tudo num único ciclo de instrução.

Um dos aspectos mais importantes no que se refere às aplicações em tempo-real é a latência no atendimento de um sinal de interrupção. Os DSPs tem um tempo de resposta determinístico e, reduzido, inerente ao número de instruções que são suportadas em *pipeline*. Existem μ Ps, por exemplo o i860XP, que permitem realizar rotinas de interrupção que não façam uso do *pipeline*; desta forma não é necessário guardá-lo e restaurá-lo, diminuindo-se o tempo necessário para executar o acesso e o retorno de uma rotina de interrupção.

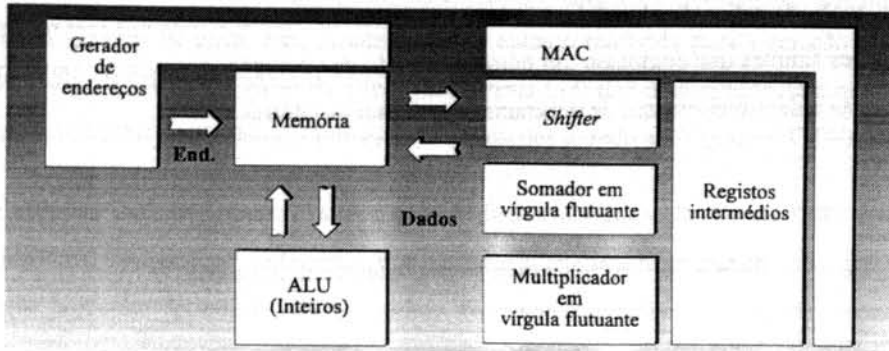


Figura 2.2 Unidade MACC (genérica) de um CPU de 32 bits de vírgula flutuante.

No âmbito dos sistemas operativos, assiste-se à tentativa de normalizar uma arquitectura que permitirá a portabilidade de aplicações desenvolvidas em diferentes suportes físicos. Neste momento existem duas interfaces concorrentes para a mesma norma, nomeadamente a OSPA (*Open-Signal Processing Architecture*), que se encontra a ser desenvolvida pela Microsystem, e a VCOS (*Visible Caching Operating System*) da AT&T [Shear91].

A OSPA resume-se a um conjunto de protocolos e interfaces que permitem ao DSP estabelecer comunicações com a aplicação final. O controlo das tarefas executadas pelo DSP é realizado pelo servidor de *SpoX* existente no sistema *host*, que se baseia na transferência de mensagens entre o sistema operativo *SpoX* (localizado no DSP) e o servidor de *SpoX*. Com este sistema é possível separar as funções realizadas ao nível da aplicação, das funções realizadas ao nível do algoritmo (Fig. 2.3).

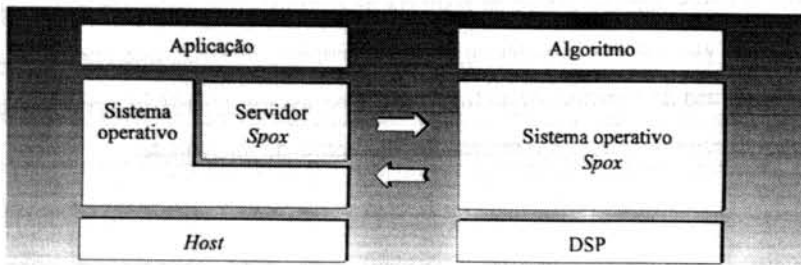


Figura 2.3 Modelo OSPA.

A VCOS é outra aproximação com o intuito de realizar os mesmos objectivos da OSPA. Esta arquitectura pressupõe a existência de um suporte físico baseado no DSP3210 da AT&T, a funcionar como co-processor no sistema onde se encontra inserido, caracteriza-se por ser multi-tarefa/multi-processamento e tempo-real, e da mesma forma que no caso anterior, foi desenvolvida uma aplicação, VCAS (*VCOS Application Server*), que funciona como servidor de VCOS.

O núcleo da VCOS é um pequeno programa executado no DSP, e é responsável pelo controlo da execução, pela gestão da memória *cache* e pela gestão do fluxo de informação de I/O. Relativamente ao servidor, é executado no sistema *host* e tem a tarefa de inicializar, carregar e activar os processos destinados ao DSP.

2.4 Requisitos em termos de *software*

2.4.1 Funcionalidades

A função principal do mestre da UPT é controlar as sessões de testes, executando funções análogas às de um "demónio" que fornece serviços, tais como a configuração da UPT através da programação das unidades escravos, definição dos dados a utilizar na sessão de testes, caracterização das mensagens em cada nível da pilha protocolar e recolha dos resultados no final da sessão. No caso de se interligarem diversas UPTs, um mecanismo *cliente-servidor* pode ser usado para permitir o controlo de todo o sistema de teste de uma forma integrada. O utilizador deverá poder visualizar e controlar o comportamento do sistema através de uma interface gráfica disponível na *UPT mestre*.

Para fornecer a funcionalidade desejada o sistema operativo do mestre tem que oferecer diversas facilidades, tais como tempo-real, multi-tarefa, gestão de memória e de I/O. Deve ser suportada uma pilha de mensagens que no seu conjunto devem incluir funções que permitam o envio de informação (com ou sem confirmação) pelo mestre para todos os escravos existentes numa UPT. Este tipo de funções permitem o controlo integrado de todos os elementos de uma UPT.

O escravo é responsável pelas acções de avaliação do desempenho e de verificação de conformidade. No primeiro caso a geração e detecção de tráfego fazem-se segundo funções estatísticas. Relativamente aos testes de conformidade, estes são usualmente realizados através do manuseamento de mensagens sendo necessário utilizar funções relativas à correcção de erros e à utilização de relógios para fornecer referências temporais.

Pelo que foi exposto anteriormente, não é desejável ocupar o tempo de processamento das unidades escravo com procedimentos que não resultem das atribuições referidas. Assim, só é necessário desenvolver um pequeno núcleo que permita executar funções relativas à comunicação com a entidade mestre, rotinas de auto-teste (fase de inicialização) e um gestor (*dispatcher*) de processos. Este último irá efectuar a comutação entre processos (sendo esta associada a acontecimentos), realizando a comunicação entre estes através de mensagens, facilidade esta que será utilizada no caso de se testar um pilha de protocolos.

2.4.1 Métodos e ferramentas

Uma metodologia orientada aos objectos deve ser utilizada no desenvolvimento dos programas para permitir a sua reutilização de tal modo que a construção de novos testes (desempenho ou conformidade) seja efectuada a partir do conjunto de objectos escritos anteriormente. Assim, a análise dos programas a desenvolver deve ser orientada aos objectos e não às estruturas, o que significa que a definição das tarefas a executar tem que ser efectuadas sob a perspectiva de um conjunto de elementos que interactuam e não como um conjunto estruturado de funções.

Na descrição das tarefas pode ser utilizada uma linguagem de especificação como o SDL (*Specification and Description Language*). Esta linguagem é particularmente adequada à descrição de protocolos, podendo ser posteriormente convertida para Linguagem C, ou C++, pois permite a especificação de tipos abstratos, de classes assim como o manuseamento de dados e funções.

A eficiência de tal procedimento é afectada pelo desempenho das ferramentas actuais, no que se refere principalmente ao código obtido a partir da tradução de SDL para a Linguagem C, ou C++. Devido à ineficiência destas ferramentas é necessário proceder à optimização do código gerado, através da inclusão de rotinas em *Assembly* nos pontos críticos do algoritmo.

A metodologia proposta para o desenvolvimento de programas é baseada no uso de ferramentas SDL (que realizam a verificação semântica e de sintaxe), a tradução automática do SDL para a Linguagem C ou C++ não implica a optimização código gerado desde que esta seja efectiva, utilizando este procedimento é possível reduzir substancialmente o tempo necessário no desenvolvimento do *software*.

2.5 Apresentação de alguns equipamentos de teste

Existem actualmente no mercado diversas ferramentas que permitem o teste de redes de comunicação com elevado débito, duas das quais vão ser apresentadas. As ferramentas escolhidas foram as seguintes: DA 30 da *Wandel & Goltermann* para o teste de redes baseadas em FDDI-II, e uma segunda ferramenta destinada ao teste de sistemas baseados em ATM, desenvolvida pelo projecto RACE 1083 - PARASOL. Através destas ferramentas pretendem-se fornecer exemplos práticos de alguns dos conceitos desenvolvidos neste capítulo.

Antes da descrição sucinta de cada uma destas ferramentas são apresentadas algumas características das redes a testar para permitir uma melhor compreensão dos requisitos de tais ferramentas.

2.5.1 Fiber Distributed Data Interface - FDDI

A norma FDDI (X3T9, desenvolvida pelo ASC - *Accredited Standards Committee*) define uma cadência de transmissão de 100 Mbps utilizando como suporte para a comunicação a fibra óptica. O protocolo de acesso ao meio é do tipo *multiple token* e a rede pode cobrir uma área geográfica de 100 Km² e suportar 500 estações distanciadas até ao máximo de 2 Km [Ross86].

A norma FDDI-II é baseada na anterior, tendo sido estabelecida com o intuito de suportar a comutação de circuitos para aplicações que utilizem vídeo e áudio. A especificação foi realizada no âmbito do modelo dos sistemas abertos (OSI) adoptado pela ISO (*International Standards Organization*). Assim, foram normalizadas as seguintes especificações:

- *Physical Medium Dependent (PMD)*: define o tipo de fibra a usar (multimodo com um diâmetro de 62.5/125 µm), assim como o comprimento de onda dos emissores (1300 nm), e tipo de repetidores de sinal. Tendo por objectivo a obtenção de uma probabilidade de erro (na transmissão) não superior a 10⁻⁹;
- *Physical Layer Protocol (PHY)*: especifica o tipo de codificação/descodificação e o método para recuperar o relógio. Estabelece a utilização do código 4B/5B que é posteriormente convertido para NRZI (sendo possível atingir cadências da ordem de 125 Mbps);
- *Media Access Control (MAC)*: define o protocolo de acesso ao meio, *Multiple Token*. De igual forma são estabelecidos os diversos campos que constituem a trama. A informação enviada é precedida de um conjunto de 16 ou mais símbolos *IDLE* e 2 símbolos de começo de trama (estes são destinados à extracção do relógio);
- *Station Management (SMT)*: esta norma, que não é mapeada no modelo OSI (Fig. 2.4), realiza funções de supervisão, controlo e gestão, procedimentos de inicialização e controlo de erro;
- *Hybrid Ring Controller (HRC)*: descreve o modo de transporte do tráfego isócrono; este aspecto não era contemplado na norma inicial.

Uma rede FDDI-II tem a capacidade de transportar não só tráfego isócrono como também tráfego de pacotes. A informação é estruturada em tramas com a cadência de 8 kHz, sendo a largura de banda reservada dinamicamente em unidades de 6.144 Mbps, denominadas WBC (*Wideband Channel*). Cada WBC pretendido para tráfego síncrono pode ser subdividido em canais de velocidade inferior.

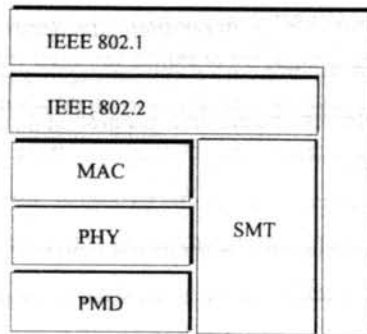


Figura 2.4 A norma FDDI-II especifica três subcamadas que correspondem aos dois níveis inferiores do modelo OSI. Também é descrita a função SMT, que tem por objectivo a supervisão das restantes subcamadas.

A banda reservada para informação síncrona por cada octeto de dados, em cada canal, é de 64 kbps, correspondendo a um canal B da RDIS. Além de suportar canais RDIS, é possível dividir um WBC em três ou quatro canais obtendo-se canais com as seguintes cadências: 2.048 Mbps e 1.536 Mbps. Estas velocidades são as requeridas para o acesso primário no E.U.A. e na Europa.

Existem dois tipos de estações: classe A e classe B (Fig. 2.5). As estações classe A, designadas por DAS - *Dual Attach Station*, são providas de duas interfaces físicas as quais se ligam ao anel primário e secundário (por exemplo, todas as estações ligadas a uma rede que funcione como *backbone* são classe A). Estas realizam uma das seguintes funções: controlador (adapta um dispositivo ao anel), concentrador (permite a interligação de múltiplas estações ao anel), adaptador de velocidades e/ou protocolos e/ou encaminhamento (*Gateway, Bridge e Router*). Em funcionamento normal uma estação classe A está logicamente ligada ao anel primário enquanto que no anel secundário opera como um repetidor.

No que respeita às estações classe B, denominadas SAS - *Single Attach Station* só têm uma única interface física e normalmente realizam funções de controladores.

2.5.2 Uma ferramenta de teste para redes FDDI-II: DA 30

Esta ferramenta suporta conexões do tipo SAS e DAS, e pode ser usada para algumas das funções anteriormente enumeradas, nomeadamente: teste de conformidade (analisa os protocolos relativos às 7 camadas do modelo OSI), avaliação do desempenho e monitoração do tráfego e determinação e isolamento de falhas [Wande92].

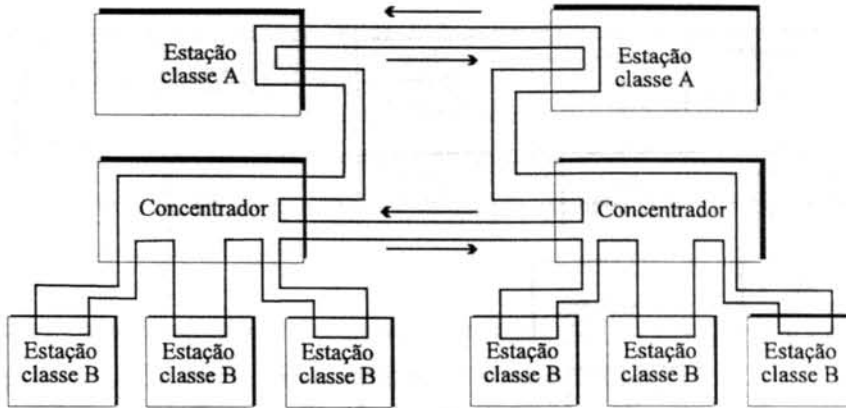


Figura 2.5 Classes de estações no âmbito de uma rede FDDI.

Enquanto que uma estação ligada à rede FDDI processa as tramas cujo endereço coincida com o seu, uma ferramenta com os objectivos expostos no parágrafo anterior necessita de analisar todo o tráfego independentemente do seu destino. O que implica receber e processar (em tempo-real) tramas à cadência de 100 Mbps, justificando desta forma o uso de duas unidades para processamento da informação recebida da rede.

É utilizado para controlador nível MAC o AM79C830 FORMAC Plus, o qual se encarrega de analisar as tramas armazenadas na memória de sistema e de as colocar numa memória de duplo porto. É da responsabilidade da unidade principal do bloco de recepção (*transputer T425*) a sua remoção da memória de duplo porto, e adicionar-lhes uma referência temporal (relativa ao instante de leitura por parte deste elemento). Posteriormente esta informação é enviada para o módulo de análise de resultados (Fig. 2.6). Enquanto se realiza este processo um outro elemento da interface (*Rx gate array*) adquire (após a conversão O/E) e adiciona às tramas recebidas da rede uma etiqueta (32 bits) relativa ao instante de recepção, sendo as tramas posteriormente armazenadas num bloco de memória de elevada dimensão (porto único).

É possível realizar funções de filtragem baseadas nos endereços de origem e de destino, no campo de controlo de erros e no tipo de protocolo utilizado.

Para enviar tramas para o duplo anel, a informação é colocada num elemento (*Tx buffer*), pelo FORMAC Plus, que as armazena temporariamente antes de proceder à sua transmissão.

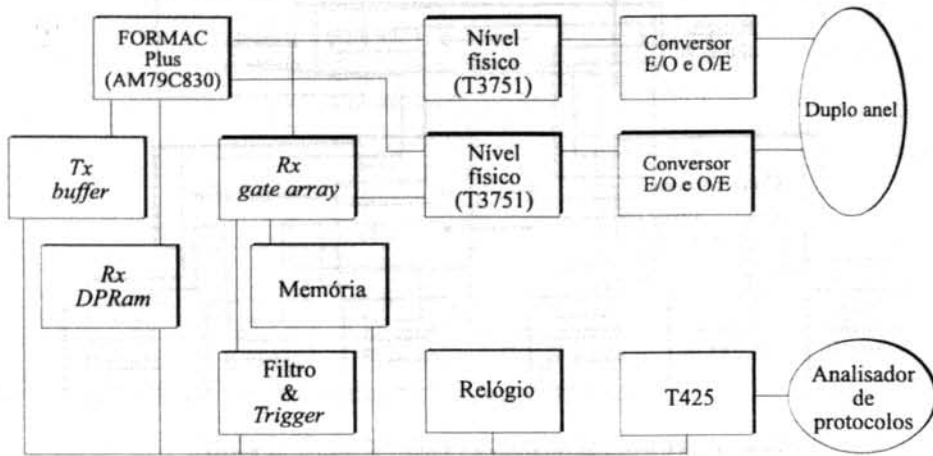


Figura 2.6 Diagrama funcional da interface do DA 30.

2.5.3 Modo de transferência assíncrono - ATM

Estando em fase final a normalização da RDIS-BL e sendo esta baseada em técnicas de transferência assíncronas, é essencial contemplar, nas redes que se encontram em fase de desenvolvimento, arquitecturas que permitam migrar para ambientes ATM.

O ATM é um modo de transferência de informação orientado às conexões, sendo estabelecidos identificadores para cada ligação: *Virtual Path Identifier* (VPI) e *Virtual Channel Identifier* (VCI). A informação é organizada em pacotes de dimensão fixa denominados células, cujo comprimento é igual a 53 octetos.

Cada célula consiste num cabeçalho e num campo de informação. A função principal do cabeçalho é a identificação de células que pertencem ao mesmo canal virtual. A reserva de recursos (para a transferência de informação) é negociada tendo por base os requisitos do utilizador e os recursos disponíveis. A sequência de células é preservada numa ligação virtual, estando inerente a este processo de transmissão a fragmentação e agrupamento da informação enviada cuja dimensão seja superior a 48 octetos (dimensão do campo de informação). Estas funções são realizadas pela camada de adaptação ALL (*ATM Adapation Layer*).

O campo de informação da célula ATM é transportado sem processamento pela camada ATM, o que significa que não se realiza controlo de erro. Esta camada é responsável pela multiplexagem e comutação dos canais virtuais.

A camada ALL além das funções descritas anteriormente realiza a adaptação entre o fluxo existente ao nível físico e os serviços mapeados em níveis superiores, sendo por isso dependente do serviço, o que implica que

existe para cada tipo de serviço uma determinada camada ALL (não foi desenvolvida uma única norma que abranja todos os requisitos). Assim, foram identificadas quatro classes associadas a cinco tipos de camada ALL:

- Classe A: ritmo de transmissão constante, orientada às conexões e sensível à variabilidade do atraso (ALL do tipo 1), de que a emulação de circuitos é um exemplo;
- Classe B: ritmo de transmissão variável, orientada às conexões e sensível à variabilidade do atraso (ALL do tipo 2 - neste momento não está contemplada na norma). Um exemplo de serviço que pertença a esta classe é a transmissão de vídeo a ritmo variável;
- Classe C: ritmo de transmissão variável, orientada às conexões (ALL do tipo 3/4 e 5). Um exemplo de serviço que pertença a esta classe é a transmissão de dados orientada às conexões;
- Classe D: ritmo de transmissão variável, orientada aos pacotes (ALL do tipo 3/4), de que a transferência de dados é um exemplo.

2.5.4 Uma ferramenta desenvolvida no âmbito do projecto RACE 1083 - PARASOL

No âmbito do projecto PARASOL (RACE 1083) foi desenvolvida uma ferramenta de teste para redes ATM. Esta baseia-se num barramento VXI através do qual se encontram ligados diversos módulos, que combinados formam instrumentos virtuais (gerador e analisador de tráfego, multiplexador/demultiplexador e emulador de rede ATM). Estes destinam-se a executar as seguintes funções:

- Teste de conformidade de nível físico e camada ATM;
- Avaliação do desempenho;
- Emulação de falhas no fluxo de células para medição dos parâmetros de qualidade (QoS) resultantes.

O gerador de tráfego (ATG) produz fluxos baseados em *bits* ou células para o teste de conformidade e de desempenho até à cadência máxima de 155 Mbps.

Relativamente ao analisador de tráfego (ATA), este observa e avalia a reacção dos módulos submetidos ao teste através de estímulos enviados pelo ATG. Este modo de funcionamento permite a medição simultânea das características do tráfego, do número de células perdidas e inseridas, do número de *bits* errados, do valor médio e da variância do atraso experimentado. No entanto, estando estas medições correlacionadas, a degradação do desempenho da rede não pode ser avaliada pela simples consulta dos valores obtidos para estas grandezas, e é necessário proceder à análise estatística dos resultados.

Para estes dois instrumentos virtuais, ATG e ATA, são admitidos três modos de funcionamento: ponto-a-ponto (considerando dois geradores A e B a transmissão é realizada do gerador A para o gerador B), em anel (por exemplo, do gerador A para o gerador B e retornando ao gerador A) e em série. Neste último caso o gerador A

recebe um fluxo de células (gerado por outra fonte externa ao PARASOL), e modifica as células do tipo *idle* pertencentes a um determinado canal virtual, sendo o fluxo original restaurado no gerador B.

O emulador de rede destina-se a introduzir alterações no fluxo de células (controladas através de sequências de dados programadas previamente) com o intuito de medir a taxa de células removidas e a variância do atraso obtido. Estas medidas destinam-se a caracterizar a relação entre a qualidade de serviço (QoS) e os parâmetros NP.

Quanto ao módulo de multiplexagem/demultiplexagem, destina-se a fundir fluxos de células ATM (gerados por uma fonte exterior e pela ferramenta) num único fluxo. Esta função tem por objectivo a substituição de determinadas células do fluxo original (externo) por outras geradas internamente.

As medidas efectuadas [RACE92] baseiam-se nos parâmetros NP relacionados com o desempenho do fluxo de células como é definido na recomendação do CCITT I.35B (Fig. 2.7). De qualquer forma a arquitectura elaborada permite a evolução para futuras sessões de teste baseadas em medidas relativas à camada ALL [Crawf91].

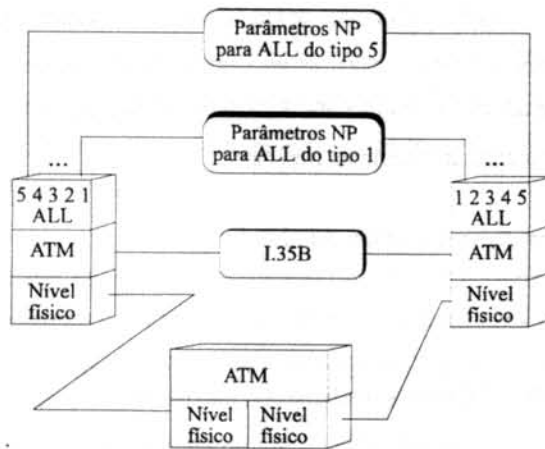


Figura 2.7 Relação dos parâmetros NP com o modelo de referência protocolar ATM.

Capítulo 3

SGDT - Sistema Gerador e Detector de Tráfego

3.1 Introdução

Apresenta-se neste capítulo uma ferramenta de teste para redes com elevado débito denominada Sistema Gerador e Detector de Tráfego (SGDT). É colocada ênfase na descrição dos circuitos impressos realizados e em alguns programas implementados para a execução das funções atribuídas à UPT (tarefas parcialmente realizadas pelo autor do presente texto). Foram ainda definidos e vão ser apresentados os conceitos básicos que caracterizam os modos de funcionamento do SGDT.

Os objectivos do protótipo do sistema de comunicação construído pelo consórcio impôs diversos requisitos à arquitectura do SGDT: era necessário que este fosse capaz de gerar e detectar tráfego de pacotes para avaliação do desempenho por forma a responder adequadamente a situações distintas, nomeadamente:

- Realização de testes quando o protótipo se encontra sob condições normais de funcionamento, isto é, os resultados obtidos deveriam poder passar a servir de referência;
- Execução de testes nos quais o protótipo se encontra em situações próximas da saturação dos seus recursos ou mesmo em sobrecarga.

Por forma a reduzir o número de UPTs necessárias para realizar os testes descritos foram escolhidos perfis de tráfego que sendo aproximações da realidade permitissem obter elevados débitos.

Adicionalmente foi necessário gerar tráfego de controlo, este era constituído pela sinalização necessária para o estabelecimento e terminação de chamadas RDIS, destinando-se a avaliar limitações do sistema operativo e *drivers* envolvidos na gestão dessas chamadas.

3.2 Apresentação do projecto DAMS

O projecto DAMS teve por objectivo o desenvolvimento de um sistema integrado de comunicações para o mercado privado. Foi dada particular atenção ao estudo dos problemas associados à integração de serviços baseados em diferentes formas de comutação, assim como ao desenvolvimento de terminais (BOD - *Bandwidth On*

Demand) que realizam a reserva de banda dinamicamente (múltiplos de 64 kbps). Este terminal foi desenvolvido em conformidade com a recomendação IEEE (*Institute of Electrical and Electronics Engineers*) 802.9.

Os requisitos identificados na fase de planeamento deste projecto (que decorreu durante os dois primeiros anos, período no qual o grupo que desenvolveu o SGTD não participou) foram os seguintes:

- Flexibilidade no uso dos recursos disponíveis;
- Integração de serviços;
- Fácil adaptação face às novas necessidades dos utilizadores;
- Elevada fiabilidade;
- Modular (necessidade de se poder proceder facilmente sua expansão);
- Custos reduzidos.

Os conceitos inerentes aos requisitos apresentados foram validados através da realização de um protótipo. Este consistia em três subsistemas autónomos interligados por uma rede FDDI-II (*backbone*), que foi utilizada com o intuito de fornecer maiores potencialidades ao protótipo construído.

Um subsistema é constituído por diversos elementos (Fig. 3.1): uma unidade que promove a ligação de cada subsistema à rede FDDI-II (BBU - *Backbone Unit*); um comutador que realiza o encaminhamento local (dentro de cada subsistema), ou então, usando a BBU, encaminhando a informação para outro(s) subsistema(s) (LSU - *Local Switch Unit*); uma entidade para controlar as ligações estabelecidas (SSCU - *Sub-System Control Unit*); e diversas unidades que promovem a ligação dos utilizadores ao subsistema (PUs - *Port Units*).

No que diz respeito aos tipos de interfaces permitidas para o exterior só um número limitado de opções é que foram contempladas (devido aos limites impostos quer em termos de tempo quer pelo orçamento disponível para realizar o projecto). Assim, cada subsistema é capaz de estabelecer ligações do tipo acesso básico RDIS (2B+D), que suportam comunicações baseadas em comutação de circuitos. As comunicações baseadas em comutação de pacotes são fornecidas através da interface para LANs 802.3 e, por último, é possível estabelecer ligações utilizando a recomendação IEEE 802.9 (comutação de circuitos e pacotes).

3.2.1 LSU - *Local Switch Unit*

A LSU [DAMS90a] é baseado num barramento dobrado (Fig. 3.2), ao qual se encontram ligadas todas as unidades descritas anteriormente, e faz uso de técnicas híbridas de comutação, o que permite a integração de tráfego isócrono e não-isócrono de uma forma eficiente.

Tendo em vista a evolução para o ATM a comutação é orientada às ligações, e para tal é adicionado o cabeçalho ATM a todos os pacotes, permitindo que se processe o seu encaminhamento baseado nos VPI/VCI atribuídos.

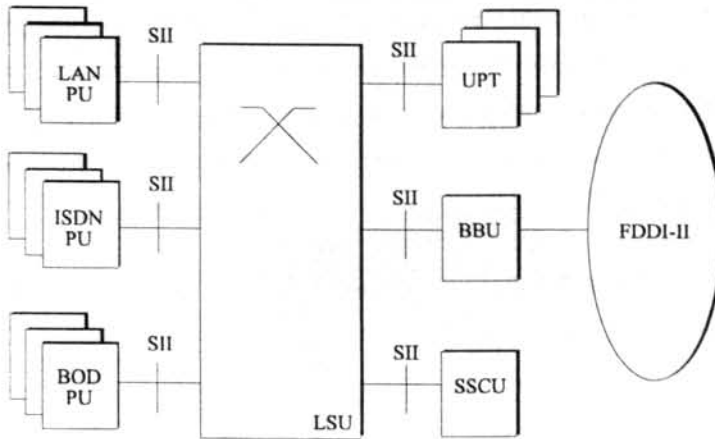


Figura 3.1 Arquitectura do projecto DAMS.

O facto de se usar um meio distribuído como técnica de comutação confere a este comutador uma elevada flexibilidade em termos de protocolo de acesso e na distribuição de tráfego (nomeadamente a transmissão de mensagens de *broadcast*). A principal desvantagem deste tipo de arquitectura é a sua expansão em termos de capacidade oferecida, pois implica o aumento do número de linhas que formam o barramento. O controlo da LSU é realizado por dois blocos: pelo LBC (*Local Bus Control*), responsável pela geração de sinais de controlo, e pelo LSC (*Local Switch Control*), para funções de gestão (de nível mais elevado).

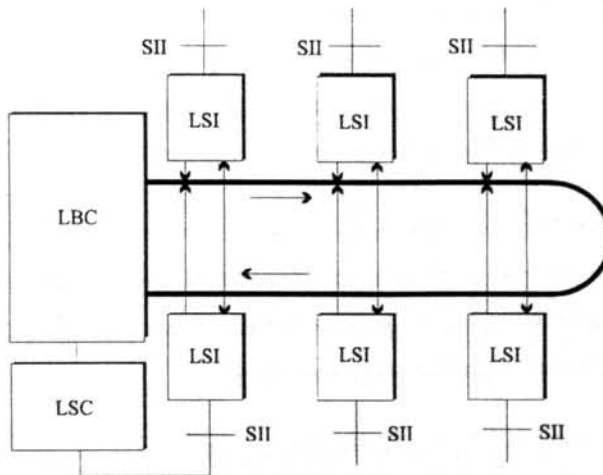


Figura 3.2 Arquitectura da LSU.

É possível transportar tráfego isócrono e tráfego não-isócrono no LSU; no primeiro caso é utilizada a técnica de comutação de circuitos entre o multiplexador temporal e a PU, sendo fornecida a possibilidade de alterar a banda requisitada durante determinada ligação. Para o tráfego não-isócrono é utilizada a comutação baseada no conceito de canal virtual, assim todas as PUs que se encontrem ligadas ao comutador são designados por um endereço IEEE, sendo os SAPs (*Service Access Points*), identificados pelos endereços, mapeados em canais virtuais. Estes são reservados pelo NI-MAC (*Non-Isochronous Media Access Control*).

As unidades externas (PUs) estão ligadas à LSU através da LSI (*Local Switch Interface*), bloco que realiza todo o processamento necessário para o encaminhamento da informação através do comutador. A sua constituição é representada na Fig. 3.3.

Os blocos funcionais que constituem a LSI executam um conjunto de acções de entre as quais se destacam:

- Nível físico - executa funções relacionadas com as interfaces utilizadas;
- *Hybrid Medium Access Control* (H-MAC) - promove a adaptação dos níveis ATM e IIS (*Internal Isochronous Switch*) ao nível físico;
- *Internal Isochronous Switch* (IIS) - permite a transferência de *slots* de informação, em ambos os sentidos, da interface SII (*Standard Internal Interface*) para slots do barramento dobrado. Este procedimento é efectuado sob o controlo do LSC através de mensagens de sinalização, sob a forma de células, utilizando o ISCL (*Internal Signal & Control Layer*);
- *Non-Isochronous Media Access Control* (NI-MAC) - entidade do bloco H-MAC que realiza serviços requisitados pela camada ATM. As funções realizadas relacionam-se com o protocolo a implementar para requisitar permissão para transmitir para o meio. De igual modo gera e verifica o campo HEC (*Header Error Control*) do cabeçalho da célula. Não realiza a delimitação de células, na realidade não existem PUs que gerem fluxos de tráfego contínuos (tal como é descrito pelo CCITT);
- *Asynchronous Transfer Mode Layer* (camada ATM) - realiza a extracção do cabeçalho e a conversão dos canais virtuais utilizados para o SAPI (*Service Access Point Identifier*) apropriado. Transfere os pacotes de informação para a camada de convergência e os de sinalização para a entidade ISCL. Implementa um mecanismo de inibição de transmissão utilizando o bloco NI-MAC;
- *Convergence Layer* (CL) - promove a ligação entre o tráfego de pacotes, transferidos via SII, para a LSI, recorrendo à comutação orientada às ligações. Realiza a verificação do endereço IEEE da PU à qual se encontra ligado a LSI com o endereço fornecido pela camada ATM. Caso a verificação esteja correcta o campo de informação é transferido via SII para a PU;
- *Internal Signal & Control Layer* (ISCL) - os procedimentos de controlo do comutador são baseados em células ATM, e por isso todos LSIs têm um VPI/VCI para o estabelecimento de comunicação com o LSC;

- *Standard Internal Interface (SII)* - é a interface comum a todos os blocos ligados ao computador. Está dividida em três sectores: distribuição de relógio de sistema, interface para o tráfego isócrono e a interface para o tráfego não-isócrono (*full-duplex*).

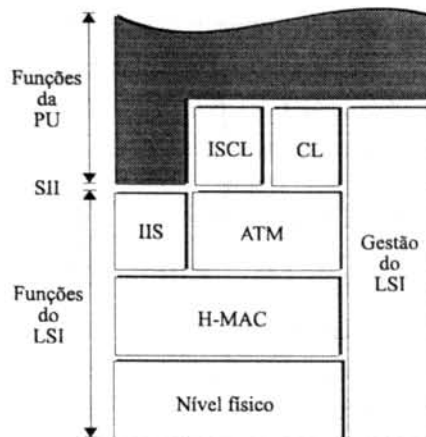


Figura 3.3 Estrutura interna da LSI.

3.2.2 BBU - Backbone Unit

Cada subsistema do projecto DAMS tem uma unidade (BBU) [Hills91] que promove a ligação deste ao anel duplo, comportando-se como uma estação do tipo DAS e fornecendo uma dupla interface e um só nível MAC.

O funcionamento da BBU é representado na Fig. 3.4: pode-se observar que o bloco H-MUX/PHY (*Hybrid Multiplexer/Physical Layer Protocol*) promove a interface com o meio, extraíndo os dados e fornecendo-os ao nível MAC numa base de 8 bits paralelo. Ambas as interfaces encontram-se ligadas ao bloco H-MUX, o qual contém a configuração do fluxo de dados ao longo das estações baseadas na informação fornecida pelo SMT.

O bloco H-MUX/PHY está ligado aos blocos P-MAC (*Packet Media Access Control*) e I-MAC (*Isochronous Media Access Control*) pelo barramento de dados FDDI. Este bloco fornece o suporte para os diversos tipos de dados (isócronos e não-isócronos) entre as entidades H-MUX e MAC. O tráfego não-isócrono é armazenado e convertido para uma interface interna no módulo P-MAC. O mesmo ocorre para o tráfego isócrono mas utilizando a entidade I-MAC. A informação é removida destes blocos quer pelo cartão de processamento quer pelo módulo que promove a interface ao sistema DAMS (módulo de aplicação). Para transmitir utiliza-se o caminho inverso, do nível MAC para o H-MUX/PHY.

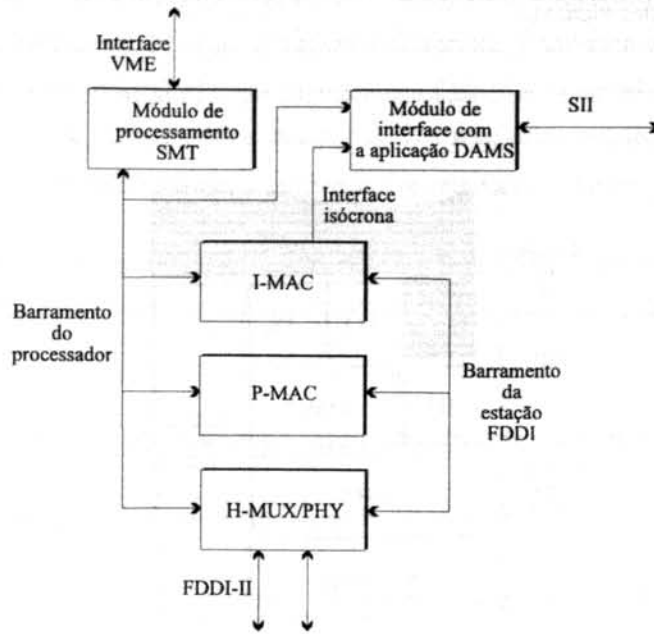


Figura 3.4 Diagrama de blocos da BBU.

No bloco I-MAC são realizadas funções de comutação temporal e espacial sob o controlo do cartão de processamento, o qual implementa as funções de SMT (*Station Management*). A entidade P-MAC realiza as funções necessárias para implementar o protocolo de acesso ao meio.

3.2.3 SSCU - *Sub-System Control Unit*

Esta unidade é baseada num processador 68030 da Motorola, o qual realiza funções de gestão, de estabelecimento e de terminação das ligações isócronas sobre canais B, recorrendo aos procedimentos (*call control*) de sinalização descritos na recomendação Q.931. Estas funções são suportadas pelo sistema operativo denominado PORDOS (*Portable Real Time Distributed Operating System*), que fornece um conjunto de serviços que permitem criar o ambiente adequado à execução das aplicações necessárias no âmbito do projecto (por exemplo, o *call control*).

3.3 SGDT - conceitos fundamentais

Basicamente o SGDT é uma ferramenta que permite a avaliação e o teste dos sub-sistemas e meios de transmissão do sistema DAMS. Além dos modos de funcionamento indicados existe ainda a possibilidade de monitorar o estado dos diversos módulos da rede sob teste. Deste ponto de vista o SGDT pode ser encarado

como uma ferramenta de gestão desde que lhe seja atribuída a potencialidade de interactivar com as entidades de controlo da rede. Do que foi exposto, resultam os modos de funcionamento seguintes:

- Avaliação do desempenho através da elaboração de medidas estatísticas realizadas sobre o tráfego de pacotes e de sinalização. Este modo deve ser utilizado durante a fase de teste por forma a otimizar globalmente o protótipo através da optimização das suas partes. Para ser possível realizar esta avaliação é necessário gerar e detectar tráfego sob a forma de pacotes até à cadência máxima de 64 Mbps (limite imposto pela interface SII), iniciar e terminar chamadas (subconjunto da recomendação Q.931 do CCITT) até ao número máximo de 500 chamadas simultâneas, e executar medidas;
- Gestão, através da avaliação do estado do sistema, sendo realizadas trocas de informação com as unidades de controlo;
- Monitoração, isto é, avaliação periódica do estado da rede quando esta se encontra a operar sob condições normais de tráfego. Caso se detectem situações de excepção o SGDT deverá ser capaz de activar mecanismos de protecção. Neste modo de funcionamento o SGDT irá gerar tráfego de baixa cadência, sendo as medidas realizadas comparadas com as obtidas durante a avaliação do desempenho.

Em termos genéricos o SGDT caracteriza-se por ser um sistema distribuído que funciona em tempo-real. É distribuído porque é composto por diversas UPTs autónomas executando as tarefas de uma forma concorrential. É tempo-real pois responde a solicitações com um atraso limitado, sendo coerente com a evolução do fenómeno controlado.

A configuração de referência utilizada é composta por duas UPTs e um posto de trabalho (WS - *Workstation*) onde são definidos os testes e visualizados os resultados, interligados por uma rede local - *Ethernet*. O SGDT faz uso de uma interface gráfica (MMI) que permite a definição dos testes a realizar; posteriormente os dados introduzidos são traduzidos para comandos a enviar para as UPTs, operação esta que é da responsabilidade do gestor de experiências. A comunicação entre a MMI e o gestor de experiências faz uso de mecanismos de IPC (*Inter-Process Communications*), enquanto que a comunicação entre o posto de trabalho e as UPTs é realizada usando mecanismos de RPCs (*Remote Procedure Calls*).

3.3.1 Conceito de sessão

As experiências são designadas por sessões e são definidas à custa de ligações ponto-a-ponto entre um gerador e um detector (no caso do tráfego de controlo, entre um originador de chamada e um destinatário). Estas ligações são usadas para transportar tráfego de pacotes ou de sinalização, e são caracterizadas por quatro elementos: a UPT de origem, a UPT de destino, o perfil de tráfego e as medidas a efectuar.

Uma sessão tem uma duração pré-definida durante a qual um conjunto de UPTs geram e detectam tráfego, existindo quatro estados possíveis em que esta se pode encontrar: em definição, à espera de execução, em execução e concluída.

3.3.2 Conceito de conexão

A definição de experiências é realizada a partir de ligações ponto-a-ponto entre UPTs que se designam por conexões. Uma conexão é uma ligação lógica que permite identificar quais as UPTs que se encontram interligadas, o tipo de tráfego que suporta e as medidas a realizar. Na Fig. 3.5 são apresentados exemplos relativos a algumas configurações possíveis para interligar o SGDT e o DAMS.

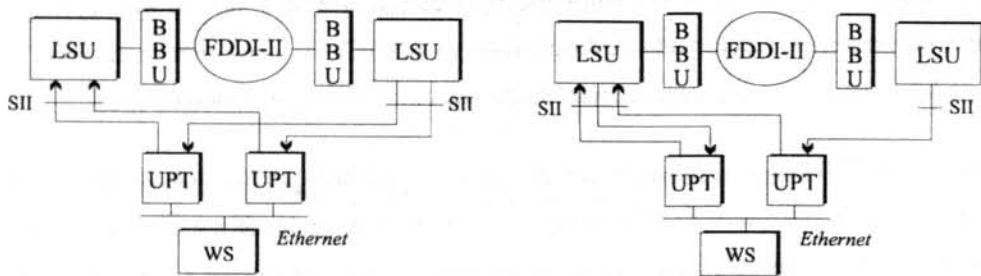


Figura 3.5 Exemplos de configurações para a interligação do SGDT ao DAMS.

Cada conexão é identificada inequivocamente pela UPT que gera o tráfego de pacotes ou que estabelece a chamada (originador). Relativamente ao destino, este pode ou não existir; no caso de ser especificada a UPT de destino esta só pode pertencer a uma conexão, se o destino não é indicado o tráfego destina-se somente a "carregar" a rede.

3.3.3 Avaliação do desempenho

Neste modo de funcionamento a sessão definida não poderá ser alterada quando em execução e pode suportar dois tipos de conexão: tráfego de pacotes e chamadas RDIS.

3.3.3.1 Tráfego de pacotes

Os perfis de tráfego suportados pelo SGDT neste modo de operação podem-se catalogar de um e de dois estados, e irão caracterizar o comprimento dos pacotes e o intervalo de tempo entre pacotes consecutivos. Para o tráfego cujo perfil é designado por dois estados, é necessário definir um parâmetro adicional (relativamente aos parâmetros especificados para os modelos de um estado) para descrever o seu funcionamento, ou seja, é

necessário especificar o tempo de permanência num determinado estado antes de se efectuar a transição para o restante.

Os parâmetros referidos (que caracterizam os modelos de um e dois estados) são definidos pela sua função de distribuição de probabilidade (FDP), média e variância (quando for caso disso). Assim, para o modelo de dois estados é necessário caracterizar para cada estado o intervalo entre pacotes consecutivos, o comprimento do pacote e o número de pacotes gerados neste estado (o que é equivalente a dizer o tempo de permanência nesse estado). Para ambos os modelos existem por defeito um conjunto de perfis de tráfego que podem ser utilizados (assim como novos perfis podem ser construídos a partir destes): *determinístico*, de *Markov*, *Gaussiano* e *AFAP* (*As Fast As Possible*).

Os resultados obtidos neste modo de funcionamento são: o atraso extremo-a-extremo, o intervalo entre pacotes consecutivos (estas medidas só são possíveis quando é a mesma UPT que gera e detecta o tráfego, condicionante que decorre do facto de não estar previsto nenhum mecanismo de sincronização entre as UPTs utilizando a *Ethernet*); o número de pacotes gerados/detectados (inclui o número de pacotes perdidos e inseridos); e o número de pacotes com erro. As medidas apresentadas são visualizadas utilizando a MMI e apresentam-se sob a forma de histogramas, média, variância e valor máximo (obviamente nem todas estas representações serão aplicadas a todas as medidas).

3.3.3.2 Tráfego de controlo

Como já foi referido as conexões que suportam este tipo de tráfego realizam o estabelecimento e terminação de chamadas, só sendo geradas as mensagens de sinalização (canal D) e não se efectuando a transmissão de informação relativa a estas (tráfego isócrona, canal B).

Em cada conexão podem ser estabelecidas chamadas com três larguras de banda distintas: 64 kbps, 2 Mbps e 8 Mbps. O perfil de tráfego é caracterizado, para além da largura de banda reservada, pelo intervalo de tempo entre chamadas, pela sua duração (intervalo de tempo entre a mensagem de *CONNECT* e *DISCONNECT*) e pelo número que as identifica. Estes parâmetros são definidos recorrendo ao uso dos perfis de tráfego referidos anteriormente (secção 3.3.3.1).

Os resultados obtidos para o tráfego de sinalização são os seguintes: atraso no estabelecimento e terminação da chamada (distinguindo os casos da terminação no originador e no destinatário), intervalo de tempo entre chamadas e duração das mesmas, número de chamadas estabelecidas com sucesso e rejeitadas, e número de falhas no estabelecimento e na terminação das chamadas.

3.3.4 Funções de gestão

Neste modo de funcionamento a sessão é interactiva não estando pré-definida a sua duração, sendo possível modificar os parâmetros que a caracterizam durante a execução. Este modo de operação utiliza os mesmos

conceitos apresentados para o modo de avaliação de desempenho, não sendo no entanto fornecidos resultados após a conclusão deste tipo de sessões. As conexões definidas em modo de avaliação de desempenho, neste caso, são destinadas a impor um certo volume de carga ao sistema para criar as situações de teste pretendidas. Adicionalmente são geradas mensagens específicas para comunicar com as entidades do sistema que se encontram em análise.

3.3.5 Monitoração

A monitoração caracteriza-se pelo estabelecimento prévio de grupos de sessões destinadas a serem executadas periodicamente com o intuito de avaliar o desempenho da rede durante o seu funcionamento normal. Desta forma é necessário definir os parâmetros que se revestem de interesse, não sendo aconselhável a medição exaustiva de todos os parâmetros apresentados no modo de avaliação do desempenho devido à elevada quantidade de tráfego que seria necessário gerar, o que prejudicaria o funcionamento da rede. O tráfego gerado neste modo de funcionamento caracteriza-se por ter cadências reduzidas.

3.4 Arquitectura e modo de funcionamento de uma UPT

Como já foi referido (capítulo 2), os elementos que constituem a arquitectura proposta para a UPT são um cartão mestre e um número variável de cartões escravos. Para o SGDT foi seleccionada uma arquitectura baseada num mestre e dois escravos. Esta opção privilegia a divisão das tarefas atribuídas a cada cartão escravo, sendo a geração e detecção funções que são executadas por escravos diferentes. O mestre é baseado num processador 68030 da Motorola que suporta uma interface *Ethernet*, e a escolha do barramento para interligar as três unidades da UPT recaiu sobre o VME (*Versa Module Europe*).

Os dois escravos dedicados às funções de geração/detecção são baseados num DSP de vírgula fixa da TI - TMS320C25 e terão que suportar uma interface, para o VME, que realiza as funções descritas na norma ANSI/IEEE 1014 [VITA87]. O VME caracteriza-se pelo modo de transferência assíncrono e não-multiplexado possibilitando débitos até 40 Mbps, e é utilizado como suporte para a transferência de informação de dados e controlo entre o mestre e os escravos.

A norma ANSI/IEEE 1014 descreve um sistema de interfaces para interligar equipamentos que realizem o processamento e armazenamento de dados e o controlo de periféricos. Esta norma apresenta a lógica necessária para suportar fisicamente a transferência de dados em quatro barramentos, com funcionalidades distintas, e numa colecção de blocos funcionais que podem ser interligados através deste suporte.

3.4.1 Unidade mestre - CPU 68030

Foram identificadas três tarefas a executar pela UPT: a gestão e controlo, o processamento dos dados e o acesso ao meio. A primeira tarefa é realizada pela unidade mestre que actua como supervisor da UPT. A Tabela 3.1

resume as características mais importantes desta unidade, tendo sido esta escolhida entre os equipamentos disponíveis baseados em VME (remete-se o leitor para os manuais que acompanham este equipamento [Micro90]).

Tabela 3.1 Características da unidade mestre.

- Baseado num processador 68030 da Motorola (32 bits)
- Formato 6U
- 4 Mbyte de memória de duplo porto (com teste de paridade)
- EPROM (formato JEDEC - *Joint Electron Device Engineering Council*)
- 256 bytes de RAM (*Random Access Memory*) não volátil
- Suporta um barramento de extensão
- Capacidade de multiprocessamento, suportando diversos utilizadores
- Interface VME, DTB (*Data Transfer Bus*) A16, A24, A32, D8, D16 e D32
- Árbitro com quatro níveis de prioridade ou com prioridades rotativas
- *Interrupter handler* e *interrupter* (sete níveis cada)
- Sistema operativo OS-9 (tempo-real e multi-tarefa)

A unidade mestre suporta um controlador de *Ethernet*, compatível com a norma IEEE 802.3, e utiliza o programa OS-9/NET para implementar o protocolo TCP/IP, o qual faz parte dos programas incluídos no pacote de *software* que acompanha o sistema operativo.

Foi necessário desenvolver programas para realizar a interface entre o sistema operativo OS-9 e as placas escravo (baseadas no DSP TMS320C25), que implementaram um conjunto de comandos (funções) que fazem uso do *device driver* desenvolvido [Proen91].

O *device driver* é constituído pelas funções de *Init* e *Getstat*. A função *Init* instala e inicializa vectores de interrupção que serão utilizados pelos escravos, sendo o acesso ao *device driver* realizado pela função *Getstat*.

3.4.2 Unidade escravo bloco servidor - DSP TMS320C25

3.4.2.1 Interface ao barramento VME

As unidades que geram e detectam tráfego sob a forma de pacotes têm exactamente a mesma arquitectura por forma a tornar o SGGT mais flexível. Sendo estas escravos do ponto de vista do VME, não podem requisitar a utilização do barramento (neste tipo de bloco funcional não é incluída esta função - *requester*), mas podem detectar e participar em ciclos de transferência de dados iniciados pelo mestre (usando o DTB - *Data Transfer Bus*) sempre que seja requisitada a sua participação.

Devido ao facto das unidades escravo não terem a capacidade de requisitar o barramento, é fornecida a possibilidade de gerar interrupções à unidade mestre para garantir a transferência de dados sempre que estas assim o

entendam. A interrupção gerada é programável por *hardware* (nível e vector), sendo enviado um pedido de interrupção pelo barramento PIB (*Priority Interrupt Bus*). Este pedido posteriormente traduz-se na leitura por parte do mestre da identificação do escravo - *Status/Id*, identificação esta que não é mais do que o vector de interrupção que cada escravo utiliza.

A implementação da interface VME das unidades escravo foi realizada utilizando lógica programável, nomeadamente PALs (*Programmable Array Logic*) e PLDs (*Programmable Logic Device*) da Altera (família 5000). A utilização destes componentes destinou-se à descodificação e regeneração dos sinais do VME, bem como à descrição através de máquinas de estados dos protocolos a executar.

Assim, foram construídos dois blocos de interface relativos à transferência de dados (A24 e D16) e à geração de interrupções (gerador de interrupções do tipo ROAK - *Release On Acknowledge*) (Fig. 3.6).

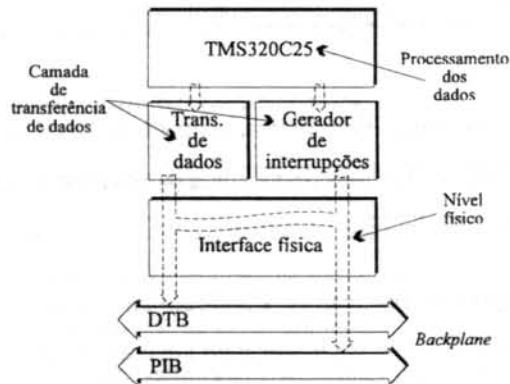


Figura 3.6 Diagrama funcional da interface (construída) ao VME.

Os tipos de ciclos de transferência de dados suportados pela interface desenvolvida são os seguintes: leitura e escrita (A24, D16 e D8, este último só é admitido em ciclos resultantes de pedidos de interrupção), leitura-modifica-escrita (a competição pelo acesso ao barramento só é necessária para realizar a leitura), escrita e leitura de uma página de memória (até ao máximo de 256 posições de memória). Estes ciclos são diferenciados através da emissão, no início do ciclo, dos modificadores de endereços por parte do mestre. Os valores destes modificadores aos quais a placa tem a capacidade de responder encontram-se descritos na Tabela 3.2.

As placas escravo encontram-se mapeadas em memória do mestre em posições correspondentes ao VME I/O (A1-A23 e D0-D15) e SHORT I/O (A1-A15 e D0-D15), sendo necessário descodificar além do endereço base e do corresponde modificador de endereço, as seguintes linhas de controlo:

$$\overline{DS0} = "0", \overline{DS1} = "1", \overline{IACK} = "1", \overline{LWORD} = "1".$$

Se os níveis lógicos das linhas de controlo referidos anteriormente estiverem correctos, então o escravo selecciona o bloco correspondente e coloca o sinal \overline{DTACK} activo; caso contrário coloca o sinal \overline{BERR} activo (só no caso do endereço se encontrar correcto). A sequência de operações pode ser descrita pelo diagrama da máquina de estados da Fig. 3.7.

Tabela 3.2 Modificadores de endereço utilizados.

Códigos Hex.	AM5	AM4	AM3	AM2	AM1	AM0
3F	"1"	"1"	"1"	"1"	"1"	"1"
3E	"1"	"1"	"1"	"1"	"1"	"0"
3D	"1"	"1"	"1"	"1"	"0"	"0"
3B	"1"	"1"	"1"	"0"	"1"	"1"
3A	"1"	"1"	"1"	"0"	"1"	"0"
39	"1"	"1"	"1"	"0"	"0"	"1"
2D	"1"	"0"	"1"	"1"	"0"	"1"
39	"1"	"0"	"1"	"0"	"0"	"1"

Os sinais referidos na Fig. 3.7 são os sinais de controlo fornecidos pelo VME, exceptuando os seguintes:

- DTBC - resulta da seguinte equação lógica: $DTBC = \overline{DS0} \& \overline{DS1} \& \overline{LWORD} \& \overline{IACK}$;
- \overline{CS} - é obtido das PALs de descodificação e está activo quando é descodificado um endereço válido, assim como o correspondente modificador de endereço;
- \overline{HOLDA} - pode-se observar que só se atinge o estado 4 se este sinal estiver activo, o que corresponde à resposta do TMS320C25 ao pedido de DMA.

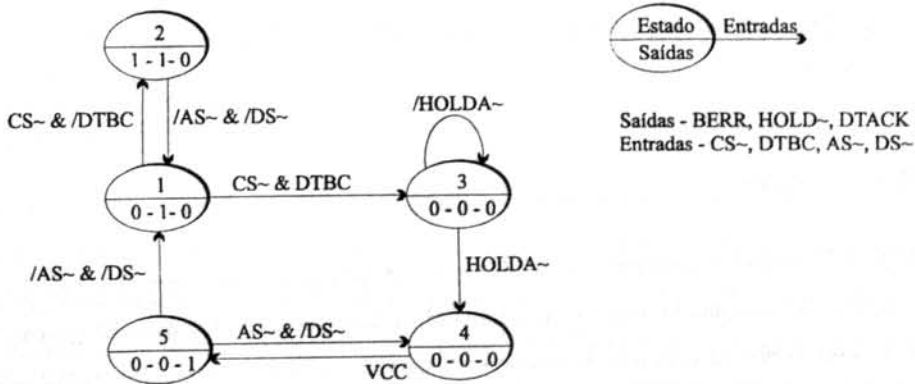


Figura 3.7 Máquina de estados que controla a transferência de dados no DTB.

Relativamente aos ciclos de leitura e escrita por transferência de página, o procedimento de controlo é idêntico ao apresentado anteriormente (Fig. 3.7), sendo necessário captar os 7 bits menos significativos do endereço base (no início da primeira transferência) e incrementá-los sempre que o escravo valida (\overline{DTACK}) cada acesso do mestre (mantêm-se constantes durante a transferência os endereços fornecidos pelo mestre).

O VME dispõe de um esquema de múltiplos níveis (7) de interrupção denominados $\overline{IRQ1} / \overline{IRQ7}$ tendo esta última a prioridade mais elevada. Todas as linhas de interrupção com o mesmo nível encontram-se unidas ("OU" físico), o que implica que no caso de vários escravos terem gerado interrupções (com o mesmo nível) simultaneamente, a ordem pela qual serão servidas depende da sua posição física face ao mestre (*Daisy-chain*).

Após a decisão de atender um determinado escravo é necessário obter a sua identificação, ou seja, ler o *Status/Id* (vector de interrupção) que o caracteriza e é da responsabilidade do mestre executar este ciclo. O escravo irá monitorar o sinal \overline{IACKIN} para que, quando este for activado, descodificar o nível da interrupção admitida a serviço através da análise das linhas A1-A3 (cujo resultado é transmitido pelo sinal \overline{ADDR}). Caso a interrupção não corresponda ao nível utilizado pelo escravo, este deve activar o sinal $\overline{IACKOUT}$ o qual irá actuar como sinal \overline{IACKIN} do próximo circuito impresso localizado no barramento. Este procedimento é descrito pela máquina de estados da Fig. 3.8.

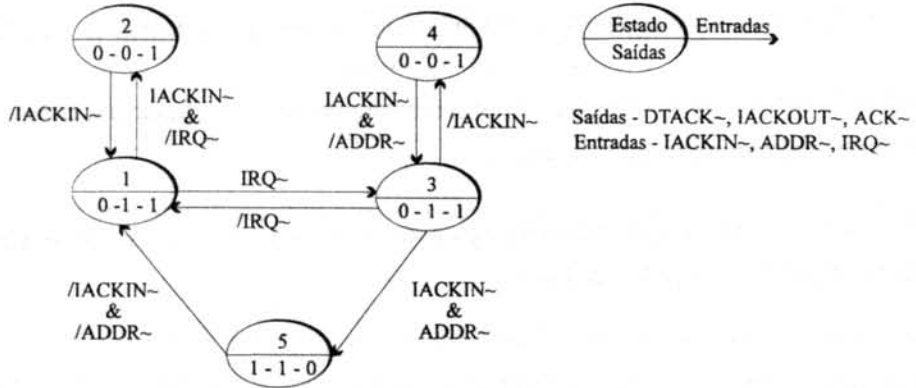


Figura 3.8 Máquina de estados relativa ao funcionamento do gerador de interrupções.

3.4.2.2 Núcleo do servidor

O núcleo de processamento do escravo é baseado, como já foi referido anteriormente, no DSP TMS320C25 [Texas86], [Lin87]. Esta escolha foi baseada na possibilidade de suportar cálculo numérico intensivo, resultando o bom nível de desempenho do conjunto de instruções disponíveis (como por exemplo: *multiply and accumulate* com opção de *data move* num único ciclo de relógio). Estas instruções permitem suportar operações em vírgula flutuante, assim como de oito registos auxiliares com uma unidade aritmética dedicada. Na tabela seguinte (Tabela 3.3) são resumidas as características principais deste DSP.

Sendo necessário obter a melhor eficácia possível o desenvolvimento da arquitectura das placas dos escravos foi realizado tendo por objectivo torná-las auto-suficientes, no sentido de minimizar a quantidade de programação que é necessária efectuar. Neste contexto, todos os blocos que interactivam com o DSP representam, para este, o mínimo de atraso possível, ou seja, o número de ciclos de espera foi minimizado por forma a afectar o menos possível o desempenho da UPT.

Tabela 3.3 Algumas características do TMS320C25.

- 100 ns de ciclo de instrução (alimentado por um oscilador externo de 40 MHz)
- 10 MIPS, dado que quase todas as instruções são de ciclo único
- Arquitectura *Harvard*, o que permite três níveis de *pipeline*
- 544 *words* de RAM interna
- 4 *keywords* de ROM interna
- 128 *keywords* de memória de dados/programa externa
- ALU (*Arithmetic and Logic Unit*) e acumulador de 32 *bits*
- Oito registos auxiliares
- Pilha em *hardware* com oito níveis
- Gera estados de espera automáticos utilizando os sinais *READY* e \overline{MSC}
- Suporta DMA, utilizando os sinais \overline{HOLD} e \overline{HOLDA}
- Permite até 256 vezes a repetição automática de uma única instrução

Cada escravo dispõe de 128 *keywords* de memória de dados e programa (referidas como memória de sistema), cujo acesso não introduz estados de espera, assim como o respectivo bloco de descodificação. É de realçar que o acesso às últimas 16 páginas de memória de dados (2 *keywords*) está restrito ao bloco de interface ao DAMS, podendo ser utilizada pelo DSP caso o acesso seja concedido pela máquina de transmissão.

Para se obter maior precisão nas medidas efectuadas implementou-se um relógio de 32 *bits* de largura para gerar as referências temporais (*time-stamps*) relativas aos instantes de geração e detecção dos pacotes. Tendo sido também desenvolvido um temporizador (*Timer*) de 32 *bits* (em detrimento do *Timer* interno de DSP, de 16 *bits*). Este módulos são alimentados pelo relógio de sistema, distribuído pela LSU, o que permite a sincronização em frequência das diversas UPTs. Está prevista a possibilidade de sincronização temporal dos relógios dos escravos que pertencem à mesma UPT, este procedimento é implementado pela função *synchronise*, e faz uso do *device driver* desenvolvido para os escravos, através da escrita nos 16 *bits* inferiores do relógio.

Foi implementada uma caixa-de-correio para permitir a troca de mensagens entre os escravos e o mestre [Kirmm85], que faz uso do gerador de interrupções descrito anteriormente e se destina fundamentalmente à troca de mensagens entre as duas entidades. Exemplos de mensagens são os comandos (*Init_session*, *Start_session*, *Stop_session*, etc) enviados pelo mestre, assim como as respostas do escravo a estes comandos (sob a forma de códigos de erro). Associada a esta facilidade está acoplada uma fila de mensagens bidireccional (2 *keywords*) destinada a promover a troca de informação mais detalhada entre o mestre e os escravos, sem que estes cessem a sessão de testes que se encontra a decorrer. Caso não existisse esta facilidade, para o mestre (sistema operativo)

obter os resultados actualizados necessitava de aceder à memória de sistema do escravo por DMA, o que implicaria parar o processamento neste.

É possível controlar, por *software*, a configuração dos escravos através de um conjunto de registos denominados de estado e controlo. Recorrendo a estes é possível programar todos os módulos existentes incluindo o modo de funcionamento do bloco da interface ao DAMS (secção 3.4.3).

Antes de terminar esta secção é de realçar que todos os blocos descritos estão acessíveis ao mestre, o que implica que todas eles estão mapeados na memória deste.

3.4.3 Unidade escravo bloco de interface ao DAMS

A interface SII [DAMS90b] define o protocolo utilizado para interligar ponto-a-ponto (cobrindo pequenas distâncias) as unidades externas (como o SGDT) à LSU. A sua estrutura pode ser dividida em três partes: global, isócrona e não-isócrona. Do ponto de vista do SGDT é particularmente relevante a que se relaciona com o tráfego não-isócrono, sendo a secção relativa aos sinais globais de uso obrigatório por todas as unidades externas. Os sinais desta última incluem o relógio de sistema (8.192 MHz), que é distribuído para efeitos de sincronização dos relógios locais de todas as unidades externas, o sinal de inicialização (RESET) da LSU e o sinal IAM (*Indication of Annoucement Message*). Este último identifica a mensagem UAM (*Unit Annoucement Message*), que é utilizada pelo elemento de gestão da LSU por forma a permitir a transmissão de informação dessa unidade externa desbloqueando a LSI ao qual se encontra ligada.

A informação é transmitida numa base 8 *bits* paralelo simultaneamente em ambos os sentidos, utilizando um relógio de 8.192 MHz, de que resulta uma cadência de 64 Mbps. As características eléctricas recomendadas são destinadas a distâncias inferiores a 1 metro, utilizando tecnologia TTL-FAST. A terminação das linhas é efectuada no receptor, sendo cada octeto transmitido protegido por um bit de paridade (ímpar, activo alto).

Estão contemplados modos de operação que permitem a transmissão de um único ou múltiplos pacotes, assim como mecanismos de recuperação de erros detectados durante a transmissão. Esta pode ser abortada quer pelo emissor quer pelo receptor, estando definidos os procedimentos a realizar em cada caso.

A arquitectura estabelecida [Assis91] decorre da norma SII e baseia-se em dois circuitos independentes para realizar o fluxo de informação em ambos os sentidos. O protocolo foi implementado em duas máquinas de estados (Tx e Rx) utilizando PLDs da Altera (EPM5032DC-1). Este tipo de componentes conjuga a velocidade à flexibilidade, possuindo uma estrutura de sector único com 32 macro-células, estrutura essa que não apresenta problemas no estabelecimento de realimentações entre diferentes sectores [Meyer90], [Conne92], [Leibs90], [Small91].

As características principais do bloco de transmissão resultam da utilização de uma memória SRAM (organizada em palavras de 16 *bits*) para reter o pacote a enviar. Esta opção permitiu aumentar o desempenho devido ao

facto do conteúdo do pacote ser praticamente invariável durante uma sessão. A construção do pacote é feita na fase de inicialização do escravo, deixando em branco os campos relativos à informação que é actualizada, nomeadamente o campo variável que contém o instante de geração (*time-stamp*). Na Fig. 3.9 é possível observar a constituição do pacote enviado sendo de realçar que o campo de informação (cujo comprimento é variável) contém um padrão pré-definido, que é utilizado para determinar os erros introduzidos na transmissão. É ainda possível enviar neste campo outro tipo de informação relevante para os testes a executar, como por exemplo o número de ordem do pacote.

Para controlar o comprimento do pacote (fornecido pela FDP relativa a esta grandeza) existe um registo de 10 bits programável (o comprimento máximo admitido no protótipo realizado é igual a 2048 octetos), este é actualizado antes da transmissão de cada pacote (se tal for necessário).

FC (1 octeto)	DA (6 octetos)	SA (6 octetos)	<i>Time-stamp</i> (4 octetos)	Informação (variável)
------------------	-------------------	-------------------	----------------------------------	--------------------------

Figura 3.9 Formato do pacote SGDT.

A arquitectura implementada no bloco de recepção é baseada numa memória FIFO (*First-In-First-Out*) que contém extractos dos pacotes recebidos. Este processo inclui um filtro, que permite a cópia de um único segmento com um comprimento máximo igual a 127 octetos (este valor é superior à maior mensagem enviada pelo PORDOS para o SGDT). Resulta desta arquitectura a possibilidade de armazenar um número elevado de pacotes (sem que o bloco servidor os tenha processado), sem que a máquina de estados esteja inibida.

A todos os pacotes recebidos é adicionado um campo de controlo de erro por forma a informar o bloco servidor da ocorrência de situações de excepção que foram detectadas durante a recepção de um determinado pacote. Assim, a toda a informação recebida e transmitida ao bloco servidor para processamento, são adicionados um dos seguintes códigos: "AAAA", recepção correcta; "0FOF", pacote abortado; e "5555", pacote com erro (por exemplo, erro de paridade).

3.4.4 Aspectos tecnológicos relativos aos circuitos impressos realizados

A opção de utilizar tecnologia multi-camada (4 camadas) nos circuitos impressos realizados foi consequência da complexidade envolvida, tendo sido mesmo necessário recorrer a 6 camadas na versão inicial do cartão servidor. O uso deste tipo de tecnologia possibilita a utilização de um plano para GND (massa) e outro para VCC (alimentação), que permitem diminuir os problemas de radiação existente em circuitos digitais através da diminuição dos anéis de corrente [Ott88], [HP89]. Sendo as linhas de relógio aquelas que devido à sua elevada frequência e periodicidade apresentam maiores índices de radiação, é necessário minimizar os traços relativos a estas linhas, tendo sido consideradas áreas de obstrução nos locais dos osciladores. Também nos circuitos de

regeneração de sinal (*drivers*), dadas as elevadas correntes nas saídas, poderiam surgir emissões de radiação, mas devido às características aleatórias dos sinais que os percorrem (sinais de controlo e dados), a energia do espectro emitido ir-se-à distribuir por um largo conjunto de frequências.

Questões relativas à adaptação das pistas e desacoplamento dos componentes foram consideradas no desenho destes circuitos impressos. Relativamente à adaptação das linhas, as situações mais críticas foram detectadas nas entradas dos componentes activadas à transição em que t_s e t_h eram na ordem de poucos ns (terminação série) [Motor90].

São apresentadas nas Figs. 3.12 e 3.13 os circuitos impressos realizados (cujos esquemas se encontram no apêndice A), estando identificadas as localizações de cada um dos blocos descritos neste texto.

3.5 Geração e detecção de tráfego de pacotes

Para se realizar a geração e detecção de tráfego sob a forma de pacotes, no servidor e em cooperação com o módulo de interface, são necessários vários procedimentos que se irão apresentar resumidamente, sendo a descrição acompanhada de fluxogramas elucidativos (uma descrição mais exhaustiva pode ser encontrada em [DAMS92]).

3.5.1 Procedimentos para a geração de tráfego de pacotes

Após a recepção do comando de inicialização de sessão (proveniente do mestre), o escravo gera tabelas, que para as FDPs escolhidas representam os intervalos entre pacotes consecutivos e os comprimentos destes, inicializa as variáveis, onde os resultados irão se armazenados e constroi os campos constantes do pacote SGDT.

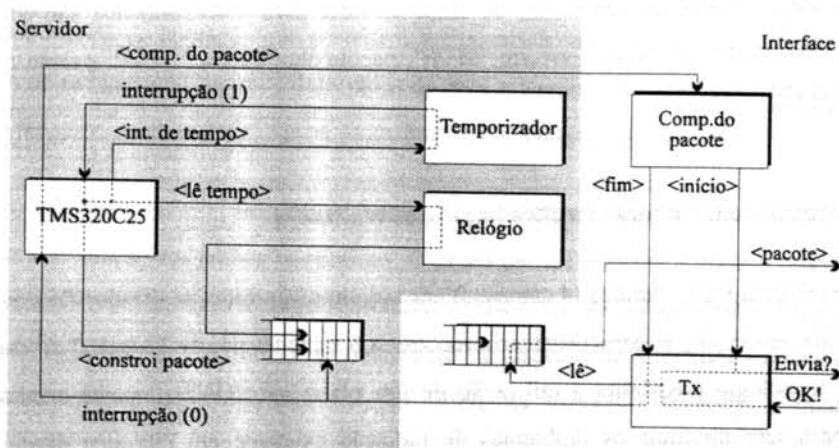


Figura 3.10 Fluxograma relativo aos procedimentos da geração de pacotes.

A geração é descrita pela Fig. 3.10 sendo iniciada após a recepção do comando de início de sessão. Este processo compreende os seguintes acontecimentos:

- Ocorrência de uma interrupção (nível 1), o que significa que irá ser gerado um novo pacote;
- Preenchimento dos campos variáveis do pacote (*time-stamp* e por exemplo número do pacote), este pacote espera, armazenado em memória de sistema do bloco servidor, que a máquina de estados que controla a transmissão (Tx) sinalize (interrupção nível 0) o TMS320C25 que deve enviar a informação relativa ao próximo pacote (campos variáveis e comprimento deste);
- Geração de outro intervalo de tempo (consulta aleatória da tabela correspondente aos intervalos de tempo);
- Envio do pacote através da SII para a LSI e no fim da transmissão o DSP é novamente sinalizado pela interrupção de nível 0.

3.5.2 Procedimentos realizados no âmbito da detecção de tráfego de pacotes

A fase de inicialização dos módulos servidor e interface é idêntica à apresentada para a geração de tráfego. Após a recepção do comando de início de sessão, a interface é informada relativamente ao segmento do pacote a copiar para o FIFO de recepção (início e comprimento). Quando a máquina Rx detecta um pedido de transmissão por parte da LSI é estabelecida a ligação utilizando o protocolo definido na norma do SII.

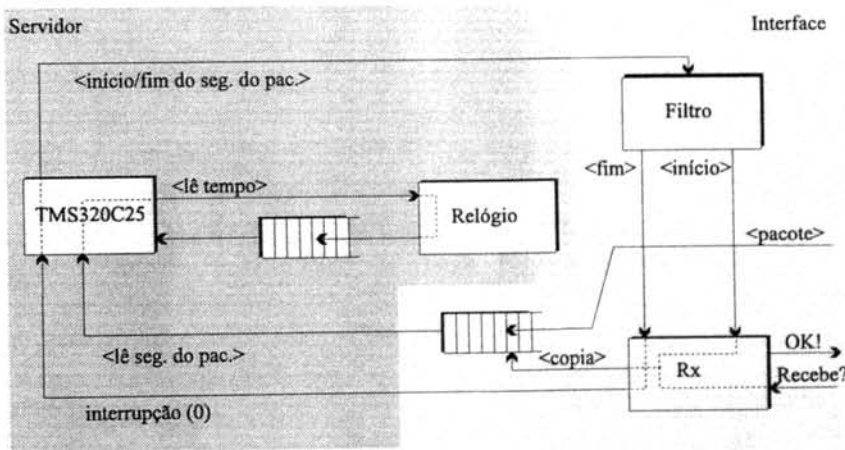


Figura 3.11 Fluxograma relativo aos procedimentos da detecção de pacotes.

Após a recepção do pacote o TMS320C25 é informado desta ocorrência, via interrupção nível 0, sendo necessário que sinalize, via registo de controlo, que atendeu a interrupção; posteriormente a máquina Rx desactiva esta saída. Neste momento o servidor pode actuar de duas formas: se está a realizar tarefas com prioridade mais

elevada não processa esta informação, caso contrário retira o segmento do pacote do FIFO e efectua as medidas. Em qualquer dos casos é retido o tempo actual, faz parte da rotina de interrupção ler o relógio e colocar o tempo obtido numa fila cuja ordenção é idêntica à do FIFO de recepção.

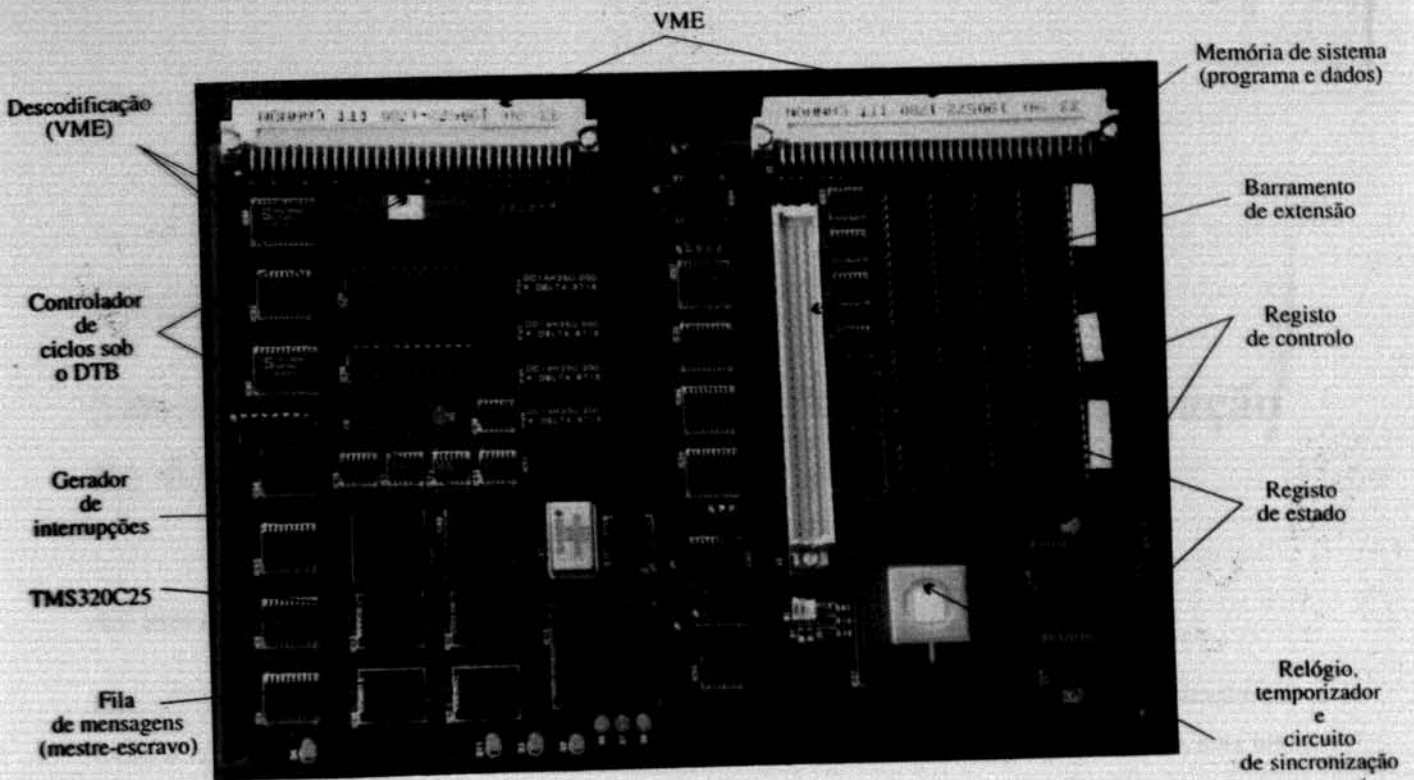


Figura 3.12 Circuito impresso do bloco servidor da entidade escravo.

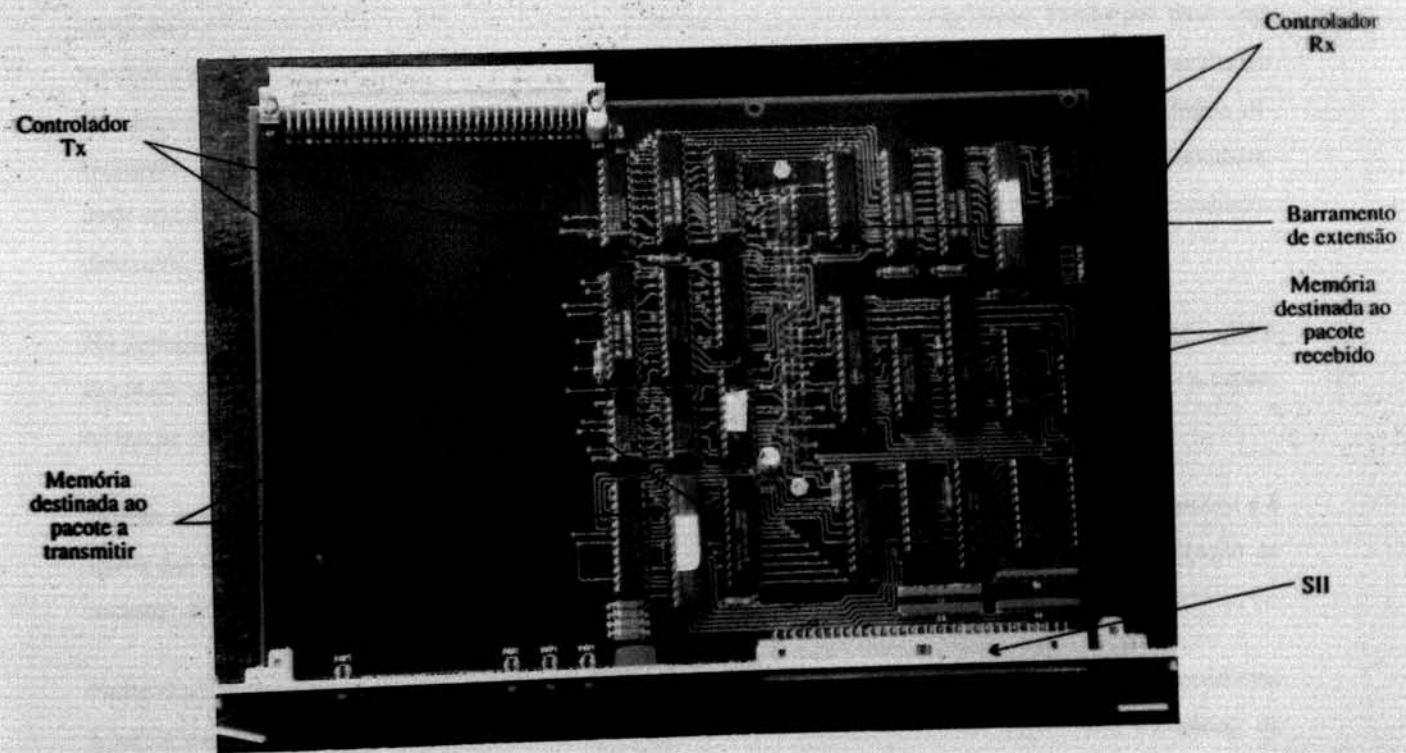


Figura 3.13 Circuito impresso do bloco de interface da entidade escravo.

Capítulo 4

Modelos Analíticos para o Projecto e Optimização de Redes de Comunicação

4.1 Introdução

Para determinar o nível de desempenho de um sistema de comunicações é usual recorrer-se à utilização de modelos matemáticos que o descrevem, na totalidade ou apenas alguns dos seus blocos mais relevantes, com um grau de profundidade adequado. Pretende-se habitualmente, através da quantificação da influência de algumas grandezas no desempenho global do sistema, prever a qualidade dos serviços oferecidos ao utilizador.

Quando se avalia um sistema de comunicação é usual adoptar determinados critérios que são na maioria dos casos subjectivos, sendo necessário traduzi-los para um conjunto de medidas objectivas. Tendo por base uma medida objectiva, é possível em alguns casos encontrar uma expressão ou expressões que permitam a avaliação do comportamento do sistema sob essa perspectiva. Como alternativa, no caso de se revelar de extrema dificuldade a determinação de tal expressão (ou mesmo na impossibilidade), ou nos casos de maior complexidade, pode-se recorrer a algoritmos, que através de uma análise computacional (simulação), forneçam os resultados desejados.

Na elaboração do modelo analítico distinguem-se duas fases: a primeira está relacionada com o tráfego que é injectado na rede, ou seja, que tipo de fontes é que se encontram ligadas ao sistema, e a segunda com a caracterização dos componentes do sistema, como por exemplo comutadores e multiplexadores.

Devido ao seu carácter aleatório a caracterização de tais modelos é obtida recorrendo a processos estocásticos e à teoria das filas de espera, sendo os processos de *Markov* largamente utilizados para realizar a modelização de sistemas de comunicação (o trabalho de [Daigl86] é exemplo dessa utilização).

Neste capítulo serão apresentadas algumas noções relativas à classificação e definição de processos estocásticos, à teoria das probabilidades (nomeadamente função geradora de momentos e distribuições sem memória), de transformadas e sistemas de filas simples.

4.2 Processos estocásticos

Um processo estocástico define-se como sendo uma família de variáveis aleatórias $\{X(t) \mid t \in T\}$ em que T designa o conjunto de valores possíveis para o índice t (o significado dado no texto a este índice é o tempo). As variáveis aleatórias $X(t)$ assumem valores pertencentes a um conjunto S , denominado espaço de estados. A classificação dos processos estocásticos depende das características do espaço de estados, do índice t e das dependências estatísticas entre as variáveis aleatórias $X(t)$ para diferentes valores do índice t .

No caso do espaço de estados (conjunto de valores que as variáveis aleatórias podem assumir) ser finito e enumerável, o processo estocástico designa-se por cadeia (conjunto de estados discreto). Por oposição, se o conjunto de valores assumidos for não enumerável, então denomina-se processo de estados contínuos.

Se os instantes de tempo t em que se avalia o estado do processo estocástico formarem um conjunto finito e enumerável, o processo estocástico é qualificado como sendo de índice discreto; caso contrário é designado de índice contínuo. No primeiro caso a notação a utilizar é X_t , que indica que é uma sequência estocástica, em oposição com $X(t)$, que se representa um processo estocástico.

O processo estocástico não fica completamente classificado se não for especificada a relação entre os seus elementos (variáveis aleatórias). Assim, é necessário determinar a função de distribuição conjunta, $F_{\mathbf{X}}(\mathbf{x}; \mathbf{t})$, em que \mathbf{X} (impressão do tipo carregado) indica um vector cujas componentes são as variáveis aleatórias, $\mathbf{X} = [X(t_1), X(t_2), \dots]$, \mathbf{x} representa os valores assumidos por \mathbf{X} , $\mathbf{x} = [x_1, x_2, \dots]$, e \mathbf{t} os valores do índice t , $\mathbf{t} = [t_1, t_2, \dots]$. De acordo com o tipo de relação entre as variáveis aleatórias, representada por $F_{\mathbf{X}}(\mathbf{x}; \mathbf{t})$, os processos estocásticos podem classificar-se em estacionários, independentes ou de incrementos independentes, de *Markov* (entre os quais a classe dos processos nascimento-morte é uma das mais importantes), caminhos aleatórios, e de renovação (entre os quais se destacam os processos *semi-Markov*).

4.2.1 Processos estacionários

Um processo estocástico é estacionário se $F_{\mathbf{X}}(\mathbf{x}; \mathbf{t})$ for invariante relativamente a deslocamentos constantes no tempo, τ . Em notação matemática

$$F_{\mathbf{X}}(\mathbf{x}; t + \tau) = F_{\mathbf{X}}(\mathbf{x}; t). \quad (\text{Eq. 4.1})$$

Um processo estacionário no sentido lato caracteriza-se pela independência de $E[X(t)]$, primeiro momento de $X(t)$, relativamente ao índice t , e à dependência de $E[X(t)X(t + \tau)]$, segundo momento, relativamente ao deslocamento τ .

4.2.2 Processos independentes

Se uma sequência X_t forma um conjunto de variáveis aleatórias independentes, o que significa que não existe nenhuma dependência ou relação entre elas, então a função densidade de probabilidade (fdp) conjunta pode ser factorizada:

$$f_X(x;t) = f_{X_1 \dots X_n}(x_1 \dots x_n; t_1 \dots t_n) = f_{X_1}(x_1; t_1) \dots f_{X_n}(x_n; t_n). \quad (\text{Eq. 4.2})$$

Uma sequência X_t é de incrementos independentes se, para todo o domínio do índice t , as diferenças

$$X_{t_1} - X_{t_0}, \dots, X_{t_n} - X_{t_{n-1}} \quad (\text{Eq. 4.3})$$

forem independentes.

4.2.3 Processos de Markov

Um processo estocástico denomina-se de *Markov* no caso de verificar a propriedade de *Markov* (Eq. 4.4), e se o conjunto formado pelo espaço de estados for discreto então este processo de *Markov* designa-se por cadeia de *Markov*. A propriedade de *Markov* permite condensar toda a sequência de acontecimentos passados, que influenciam o futuro, no estado actual. Analiticamente a propriedade de *Markov* é descrita da seguinte forma

$$P[X(t_{n+1})=x_{n+1} | X(t_n)=x_n, \dots, X(t_0)=x_0] = P[X(t_{n+1})=x_{n+1} | X(t_n)=x_n]. \quad (\text{Eq. 4.4})$$

As cadeias de *Markov* podem-se classificar relativamente ao tipo do conjunto formado pelo índice t , resultando a designação de cadeias de *Markov* de tempo discreto e tempo contínuo, respectivamente se as transições para um novo estado se efectuarem em instantes de tempo discretos ou contínuos.

Conhecendo os instantes de transição de estado, a determinação do tempo que o processo residiu num estado particular é imediata, bastando para isso ter conhecimento das probabilidades de transição de estado. Devido à propriedade de *Markov*, a distribuição probabilística desta grandeza tem que ser exponencial (caso contínuo) ou geométrica (caso discreto).

Um caso particular das cadeias de *Markov* são os processos *nascimento-morte* usados no desenvolvimento da teoria das filas de espera e na construção de modelos que descrevem os sistemas de comunicação. Este tipo de processos estocásticos caracteriza-se por só admitir transições entre estados adjacentes.

4.2.4 Caminhos aleatórios

Neste tipo de processos estocásticos visa-se a determinação do estado em que estes processos se encontram num determinado momento: em cada instante de transição procede-se à avaliação do próximo estado, sendo este

baseado no estado actual e no valor assumido por uma grandeza aleatória cuja distribuição probabilística é arbitrária e independente do processo.

É usual descrever este tipo de processos fazendo a analogia com a trajetória de uma partícula: o caminho percorrido pode ser delimitado por estados que impõem certas condições à sua mobilidade, tais como barreiras elásticas e absorventes. Considere-se um caminho aleatório sem restrições a uma dimensão $\{X_t \mid t \in T\}$: em cada instante existe a probabilidade p da partícula se mover para a direita e a probabilidade $q = 1 - p$ de se mover para a esquerda, sendo p constante e independente do estado.

O processo descrito é de incrementos independentes, ou seja, é independente do estado inicial ($P[X_{t_n} = i + j \mid X_{t_0} = j] = P[X_{t_n} = i \mid X_{t_0} = 0]$), e é independente relativamente a deslocamentos constantes no tempo ($P[X_{t_n + \tau} = i \mid X_{t_n} = 0] = P[X_{t_\tau} = i \mid X_{t_0} = 0]$). Se o espaço de estados for constituído por $S_{t_n} \in \{-1, 1\}$ e $P[S_{t_n} = -1] = q$ e $P[S_{t_n} = 1] = p$, então o estado num determinado instante de tempo é dado por,

$$X_{t_n} = S_{t_0} + S_{t_1} + \dots + S_{t_n} \quad (\text{Eq. 4.5})$$

4.2.5 Processos de renovação

Ao relaxar-se a restrição relativa à distribuição probabilística do tempo de permanência num determinado estado, aplicada aos processos de *Markov*, um novo processo estocástico pode ser definido: uma família de variáveis aleatórias, $\{X_{t_n}\}$ que definem os instantes de tempo relativa às ocorrências de um qualquer fenómeno, em que as diferenças $Z_{t_n} = X_{t_n} - X_{t_{n-1}}$ (sendo $X_{t_0} = 0$) formam uma sequência de variáveis aleatórias independentes e identicamente distribuídas. Então o processo $\{X_{t_n} \mid n > 0\}$ é um processo de renovação.

Aos instantes de tempo $\{X_{t_n}\}$ chama-se instantes de renovação e aos intervalos $\{Z_{t_n}\}$ períodos de renovação. Se os instantes de renovação forem seleccionados de acordo com uma cadeia discreta de *Markov*, em que as transições de estado obedecem a uma distribuição probabilística arbitrária, então ao processo que resulta chama-se processo *semi-Markov*. Estes processos utilizam-se no estudo de sistemas de filas do tipo *M/G/1* (notação de *Kendall*), sendo a técnica utilizada denominada método das cadeias de *Markov embebidas*.

4.3 Cadeias de *Markov*

As cadeias de *Markov* são processos estocásticos que encontram grande aplicabilidade na modelização de sistemas de comunicação, e por isso, merecem uma referência mais detalhada.

4.3.1 Cadeias de *Markov* de tempo discreto estacionárias

Neste tipo de processos estocásticos, a transição entre estados é determinada pelas probabilidades condicionais, ($P[X_n = j \mid X_{n-1} = i]$), ou seja, a probabilidade de no instante n o processo evoluir do estado i para o estado j , sendo a notação adoptada $p_{ij}^{(n-1, n)}$. Esta notação pode ser simplificada, p_{ij} , no caso das cadeias de *Markov* de tempo discreto serem estacionárias, ou homogéneas, no tempo, o que significa que as probabilidades condicionais são independentes do instante de tempo de transição de estado.

Denominam-se probabilidades de transição as probabilidades condicionais que governam a transição entre os estados, e são representadas utilizando notação matricial. O resultado é uma matriz de transição de estados, \mathbf{P} , em que o elemento da linha i coluna j fornece a probabilidade de transição p_{ij} . Esta matriz pertence à classe das matrizes estocásticas, pois todas os seus elementos são positivos e a soma ao longo de uma linha é igual à unidade.

A partir do conhecimento de \mathbf{P} é possível avaliar em todos os instantes o estado do processo, sendo só necessário indicar o estado inicial. Um vector ($\boldsymbol{\pi}^0$) é válido para descrever as condições iniciais se o somatório das suas componentes for igual a um. Para obter a distribuição de permanência após n passos (n instantes de transição de estado) é necessário calcular

$$\boldsymbol{\pi}^n = \boldsymbol{\pi}^0 \mathbf{P}^n. \quad (\text{Eq. 4.6})$$

Sob determinadas condições $\boldsymbol{\pi}^n$ converge para um vector $\boldsymbol{\pi}$, cujas componentes são constantes e cujo somatório é igual a um, o qual é independente das condições iniciais (cadeia de *Markov* ergódica). A este vector chama-se distribuição de probabilidade estacionária e fornece as probabilidades de no limite a cadeia se encontrar num determinado (qualquer) estado. Resta estabelecer em que condições é que existe este limite, o que leva ao estudo da classificação dos estados de uma cadeia de *Markov*.

Uma cadeia de *Markov* é irredutível se e só se num qualquer número de passos for possível a transição entre quaisquer dois estados que lhe pertençam. Caso contrário, pode-se encontrar uma ou mais subcadeias irredutíveis se forem constituídas por um conjunto fechado de estados, não contendo outro subconjunto de estados fechado. Entenda-se por conjunto fechado uma colecção de estados que têm probabilidade nula de transitar para outro estado que não lhe pertença. Se, por hipótese, o conjunto fechado contiver um só elemento, este é chamado estado absorvente.

Se se quantificar a probabilidade do primeiro retorno (após n passos) ao estado j , $f_j^{(n)}$, medida a partir do momento em que se deixou esse estado, então, no limite esta probabilidade de retorno a esse estado é fornecida por

$$f_j = \sum_{n=1}^{\infty} f_j^{(n)}. \quad (\text{Eq. 4.7})$$

De acordo com o valor de f_j , pode-se classificar o estado j em estado recorrente (ou persistente), $f_j = 1$, ou então em estado transitório, $f_j < 1$. Por último, se o retorno ao estado j só é possível em múltiplos de γ , sendo $\gamma \neq 1$, então o estado j é periódico com período igual a γ . Para todos os estados persistentes calcula-se o tempo médio de retorno ao estado j como sendo

$$\mu_j = \sum_{n=1}^{\infty} n f_j^{(n)}. \quad (\text{Eq. 4.8})$$

Baseado no tempo médio de retorno ser infinito ou finito, podem-se classificar os estados persistentes em persistentes nulos e persistentes positivos.

Neste momento já nos encontramos em posição de obter a distribuição estacionária π de uma cadeia de *Markov* irredutível (o que implica que todos os estados sejam positivos persistentes) e aperiódica. Esta satisfaz as condições seguintes

$$\pi_j > 0, \quad \sum_{j \in S} \pi_j = 1, \quad \sum_{i \in S} \pi_i p_{ij} = \pi_j. \quad (\text{Eq. 4.9})$$

O que foi exposto no parágrafo anterior é aplicável ao caso de subcadeias irredutíveis; para tal basta identificar os estados positivos persistentes, reordenando a matriz de transição de estados para o formato de bloco,

$$P^* = \begin{bmatrix} P_i & 0 \\ R_i & Q \end{bmatrix}. \quad (\text{Eq. 4.10})$$

P_i é um conjunto de matrizes estocásticas correspondentes às transições entre estados das sub-cadeias irredutíveis. Q refere-se às transições entre os estados transitórios, e é uma matriz sub-estocástica (elementos todos positivos ou nulos mas a soma ao longo da linha é inferior à unidade). Às matrizes R_i correspondem as transições dos estados transitórios para as subcadeias irredutíveis.

Uma questão central, que surge no âmbito deste estudo, é a determinação do tempo médio de absorção de um estado transitório, por um conjunto de estados persistentes, e qual a probabilidade dessa absorção se realizar por uma dada subcadeia irredutível. Para calcular o tempo médio de absorção, para cada estado transitório, calcula-se a soma ao longo da linha da matriz fundamental N ,

$$N = (I - Q)^{-1}. \quad (\text{Eq. 4.11})$$

Este processo é denominado *método da matriz fundamental* e, ao contrário do método directo, não exige o cálculo de $q_{ij}^{(n)}$. Prova-se que a matriz fundamental existe para toda a matriz sub-estocástica Q que resulte das probabilidades de transições entre os estados transitórios de uma cadeia de *Markov* finita.

Resta obter as probabilidades de absorção de um estado transitório para um estado persistente. Este procedimento baseia-se novamente na matriz fundamental. O produto entre a matriz N e a matriz que descreve as transições entre estes dois tipos de estados (R) fornece os resultados pretendidos. Desta forma, a probabilidade de absorção de um estado transitório por uma subcadeia irredutível é obtida pela soma das entradas, na matriz produto, que pertencem a esse estado e aos estados da subcadeia irredutível em questão.

4.3.2 Coeficiente egórdico

Uma cadeia de *Markov* não estacionária é completamente descrita pelo vector de distribuição de probabilidades inicial, $\mathbf{f}^{(0)}$, e pela sequência das matrizes de transição, $\{\mathbf{P}_n\}$, resultando daí o interesse de estudar o comportamento do vector de distribuição de probabilidades após n passos.

Existem diversos comportamentos que tal sequência pode assumir, entre os quais se destaca a convergência para um determinado limite, independentemente das condições iniciais (perda de memória). As cadeias que se encontram nestas condições denominam-se por cadeias de *Markov* não estacionárias fortemente egórdicas. Outros tipos de comportamento englobam a convergência sem perda de memória ou então perda de memória sem convergir.

Neste momento surge a necessidade de introduzir conceitos relativos à norma de um vector e de uma matriz, para posteriormente serem usados na definição de cadeias de *Markov* não estacionárias, fortemente e fracamente egórdicas. Assim, norma de um vector \mathbf{f} é definida por

$$\|\mathbf{f}\| = \sum_{i=1}^{\infty} |f_i|, \quad (\text{Eq. 4.12})$$

sendo a norma de uma matriz quadrada, \mathbf{A} , dada por

$$\|\mathbf{A}\| = \sup_i \sum_{j=1}^{\infty} |a_{ij}|. \quad (\text{Eq. 4.13})$$

Recorrendo às definições apresentadas define-se uma cadeia de *Markov* não estacionária como fracamente egórdica se verificar a seguinte igualdade:

$$\lim_{k \rightarrow \infty} \sup_{f^0, g^0} \|f^{(m,k)} - g^{(m,k)}\| = 0, \quad \forall m. \quad (\text{Eq. 4.14})$$

Devido à dificuldade em aplicar a definição anterior foi introduzida a noção de *coeficiente egórdico*, definido por *Dobrushin*. Este é calculado através da seguinte relação:

$$\alpha(\mathbf{P}) = 1 - \sup_{i,k} \sum_{j=1}^{\infty} [\max(0, p_{ij} - p_{kj})], \quad (\text{Eq. 4.15})$$

em que \mathbf{P} é uma matriz quadrada com dimensões finitas, que gera um conjunto de estados enumerável. O coeficiente definido obedece às seguintes relações:

$$\alpha(P) = 1 - \frac{1}{2} \sup_{i,k} \sum_{j=1}^{\infty} |p_{ij} - p_{kj}|, \quad (\text{Eq. 4.16})$$

$$0 \leq \alpha(P) \leq 1. \quad (\text{Eq. 4.17})$$

Em alguns casos é conveniente utilizar o *coeficiente delta*, $\delta(P)$,

$$\delta(P) = 1 - \alpha(P). \quad (\text{Eq. 4.18})$$

Este é utilizado para medir a proximidade de uma matriz estocástica da matriz constante, isto é, de linhas iguais. Quando, para uma determinada matriz, o coeficiente delta for nulo isso significa que a matriz é constante. Fazendo uso deste coeficiente é possível simplificar o processo da determinação de ergodicidade fraca; para tal basta verificar se para todo m o valor de $\delta(P^{(m,k)})$ tende a anular-se para k crescente.

Relativamente ao caso de uma cadeia de *Markov* não estacionária ser fortemente ergódica é necessário determinar uma matriz constante Q tal que

$$\lim_{k \rightarrow \infty} \|P^{(m,k)} - Q\| = 0. \quad (\text{Eq. 4.19})$$

Para determinar a matriz Q pode-se estabelecer a analogia com o caso estacionário, em que as linhas de P^n são iguais ao vector de probabilidade estacionária, π ; este vector representa o vector próprio à esquerda, associado ao valor próprio unitário. Desta forma é imediata a transposição para o caso não estacionário, em que o vector próprio à esquerda de P_n será ψ_n , que se relaciona com as linhas de Q .

Seja $\{P_n\}$ uma sequência de matrizes de transição de estados de uma cadeia de *Markov* não estacionária, em que, para cada matriz P_n , exista um vector próprio, à esquerda, associado ao valor próprio unitário, tal que a sua norma seja igual a um. Para se poder dizer que uma determinada cadeia é fortemente ergódica é necessário que

$$\lim_{n \rightarrow \infty} \|\psi_n - \psi\| = 0, \quad (\text{Eq. 4.20})$$

$$D < \infty, \quad \sum_{j=0}^{k-1} \delta(P^{(j,k)}) \leq D. \quad (\text{Eq. 4.21})$$

As condições anteriores refletem a necessidade de P_n convergir para P , sendo esta última fracamente ergódica. Neste caso é automática a convergência do vector próprio e, por consequência, a cadeia é fortemente ergódica.

4.3.3 Cadeias de Markov de tempo contínuo

Se numa cadeia de Markov os instantes de transição de estado não formarem um conjunto discreto de elementos, mas sim constituírem uma grandeza contínua, estamos perante uma cadeia de Markov de índice (tempo) contínuo. Esta é representada simbolicamente por $X(t)$ e respeita a (Eq. 4.4). A validade da propriedade de Markov neste tipo de processos leva a que o tempo que a cadeia permanece num determinado estado tenha uma distribuição particular: exponencial. Mais à frente neste texto (secção 4.4.1) irá ser apresentada como a única função distribuição de probabilidade (FDP) que, em conjunto com a função de distribuição geométrica, no caso discreto, goza desta propriedade.

Neste tipo de cadeias de Markov a matriz de probabilidades de transição a um passo é substituída por uma matriz cujos elementos descrevem a frequência de transição entre estados. Esta representação conduz à determinação da derivada em ordem ao tempo de $p_{ij}(s, t)$ quando $t \rightarrow s$. A probabilidade estacionária de permanência é obtida através das expressões (Eq. 4.22), em que \mathbf{P} é a matriz de transição de estados e \mathbf{Q} o gerador infinitesimal (matriz de frequência de transição). A dedução das expressões apresentadas recorre à determinação das equações de Chapman-Kolmogorov [Klein75]

$$\pi \mathbf{Q} = \mathbf{0}, \quad \sum_j \pi_j = 1. \quad (\text{Eq. 4.22})$$

$$\mathbf{Q}(t) = \lim_{\Delta t \rightarrow 0} \frac{\mathbf{P}(t) - \mathbf{I}}{\Delta t}. \quad (\text{Eq. 4.23})$$

Um caso particular deste tipo de cadeias são os processos nascimento-morte, os quais só admitem transições entre estados adjacentes. Assim, estes processos utilizam duas grandezas que definem as taxas de nascimento (λ) e de morte (μ), ou seja, que fornecem as probabilidades de se transitar para um estado adjacente de índice superior ou inferior (em termos da população existente esta é identificada pelo índice do estado presente $P_k(t)$). Considerando que a cadeia é estacionária, isto é λ e μ não dependem do índice tempo mas sim do estado em que se encontra a cadeia, pode-se construir o gerador infinitesimal \mathbf{Q} em função destas taxas (Eq. 4.24). De realçar que a soma ao longo da linha é nula.

$$\mathbf{Q} = \begin{bmatrix} -\lambda_0 & \lambda_0 & 0 & 0 & \dots \\ \mu_1 & -(\lambda_1 + \mu_1) & \lambda_1 & 0 & \dots \\ 0 & \mu_2 & -(\lambda_2 + \mu_2) & \lambda_2 & \dots \\ 0 & 0 & \mu_3 & -(\lambda_3 + \mu_3) & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} \quad (\text{Eq. 4.24})$$

A matriz \mathbf{Q} pode ser representada através de um grafo de transição de estados, que permite estudar a cadeia de Markov com bastante simplicidade desde que a sua representação seja periódica. Esta representação pode ser usada para determinar a dinâmica do sistema através do estabelecimento da equação diferencial das diferenças

do sistema. A taxa de entrada num estado genérico E_k pode ser representada por $\lambda_{k-1}P_{k-1}(t) + \mu_{k+1}P_{k+1}(t)$ e a taxa de saída desse estado pode ser obtida por $(\lambda_k + \mu_k)P_k(t)$. A diferença descreverá a dinâmica do sistema:

$$\frac{\partial P_k(t)}{\partial t} = \lambda_{k-1}P_{k-1}(t) + \mu_{k+1}P_{k+1}(t) - (\lambda_k + \mu_k)P_k(t). \quad (\text{Eq. 4.25})$$

Um caso particular é o estudo de um processo puramente de nascimento: a resolução da equação (Eq. 4.25), supondo condições iniciais apropriadas, fornece um processo de *Poisson* que descreve as épocas de nascimentos.

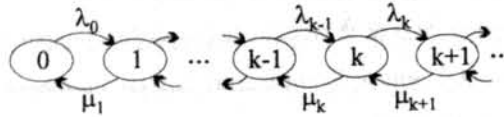


Figura 4.1 Grafo de transição de estados de um processo nascimento-morte.

4.3.4 Análise computacional

Um algoritmo para o análise de cadeias de *Markov* finitas, estacionárias, e de tempo discreto deve-se a Isaacson [Isaac76]. O primeiro passo é classificar a matriz de transição de estados quanto à egordicidade. Para avaliar esta característica pode-se recorrer ao coeficiente egórdico: para a matriz de transição ser egórdica é necessário que $\alpha(P^k) > 0$, para qualquer k .

Uma questão que se coloca é até quando se deve repetir este processo (cálculo do coeficiente egórdico), visto que para uma matriz não egórdica nunca se irá verificar tal condição. A resposta aceite é que para uma matriz estocástica, com uma dimensão $n \times n$, ser egórdica é necessário que $\alpha(P^k) > 0$ para qualquer $k \leq n(n-1)/2$. Os valores numéricos obtidos não são relevantes bastando determinar se o valor calculado é diferente de zero. Este facto permite representar os seus elementos em base 2 (0 e 1), matriz adjacente (Z). Esta transformação baseia-se na substituição dos valores positivos pela unidade, o que implica a alteração da definição dada para o coeficiente egórdico, visto que a matriz Z não é uma matriz estocástica (Eq. 4.26),

$$\alpha(Z^k) = \min_{i, m} \sum_j z_{ij}^k z_{mj}^k. \quad (\text{Eq. 4.26})$$

Para impedir que os elementos da matriz Z^k sejam de grande dimensão, executa-se a transformação descrita (matriz adjacente) em cada iteração da potenciação da matriz Z^k . Este procedimento pode ser bastante ineficaz se, por exemplo, a matriz de transição não for egórdica. Para obviar a este problema Isaacson propôs um método, denominado *soma-coluna*, que se baseia no cálculo de potências de dois da matriz Z e na observação dos valores obtidos ao longo de uma coluna. Este método apresenta-se sumariado no fluxograma seguinte (Fig. 4.2).

A análise da cadeia de *Markov* depende da classificação da matriz de transição de estados; para o caso da matriz não ser egórdica o principal objectivo é escrever a matriz sob a forma de bloco. Para isso é obrigatório determinar os estados persistentes e, consequentemente, os estados transitórios. Um estado i é persistente se a soma ao longo da coluna i , da matriz Z^M (M é a potência em que foi decidida a egordicidade da cadeia), é máxima, isto é, superior à soma de qualquer outra coluna. Na condição do estado i ser persistente, existe um conjunto de estados que podem ser classificados devido a pertencerem ao conjunto irredutível e fechado gerado pelo estado i , C_i .

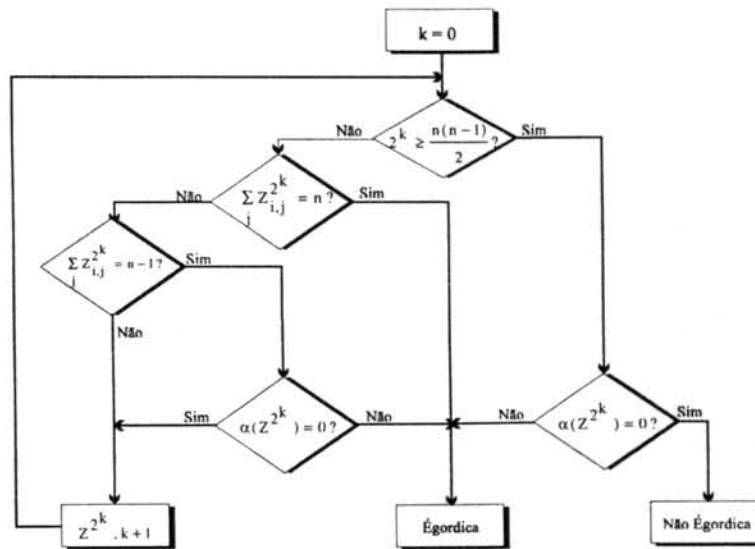


Figura 4.2 Fluxograma descritivo do método da soma-coluna.

O conjunto de estados gerados pelo estado i são persistentes e podem ser calculados pela expressão seguinte:

$$C_i = \bigcup_{j=0}^{d-1} K_j, \tag{Eq. 4.27}$$

em que d é o período de C_i , isto é, o menor inteiro para o qual $K_d = K_0$, sendo K_t definido da seguinte maneira:

$$K_t = \{j: z_{ij}^{M+t} = 1\}. \tag{Eq. 4.28}$$

Para obter outro estado persistente podem-se omitir os estados já classificados, C_i . Assim, antes de se executarem outras operações sobre a matriz Z^M , esta é alterada por forma a colocar zeros em todas as linhas e colunas correspondentes a esses estados. Após esta operação aplica-se o mesmo processo para determinar outro estado persistente.

No entanto os estados transitórios, para os quais existem transições para a subcadeia constituída pelo conjunto fechado de estados definidos por C_i (absorção), também podem ser classificados, o que significa que as suas entradas podem ser anuladas. Este novo conjunto é definido pelas relações seguintes, e irá conter os estados persistentes pertencentes a C_i , assim como todos os estados transitórios através dos quais se atinge i (Eq. 4.29), (Eq. 4.30).

$$D = \bigcup_{l=0}^{d-1} D_l \quad (\text{Eq. 4.29})$$

$$D_l = \left\{ j: z_{j,K_l}^M = 1 \right\} \quad (\text{Eq. 4.30})$$

A etapa final será encontrar o tempo médio de absorção e as probabilidades de absorção dos estados transitórios pelos estados persistentes. Para tal basta considerar o cálculo da matriz fundamental (Eq. 4.11), sabendo que a matriz de transição de estados se encontra sob a forma de bloco.

Retomando as cadeias egórdicas, o objectivo é a determinação da distribuição estacionária. Para tal basta resolver o sistema de ordem n descrito pelas equações (Eq. 4.9). O cálculo do tempo médio de absorção é idêntico ao que foi descrito para as cadeias não egórdicas, bastando observar que os estados transitórios são aqueles identificados por $\pi_j = 0$.

Um problema que se põe são os arredondamentos efectuados durante o cálculo das probabilidades estacionárias, os quais podem introduzir erros na determinação dos estados transitórios. Assim, é necessário implementar modificações ao nível da sequência dos cálculos a efectuar, tais como realizar diversos testes aos resultados obtidos após cada operação concluída. Existem algoritmos mais precisos para avaliar os estados transitórios, os quais não serão aqui abordados dado que implicam a não utilização do método da soma-coluna.

No apêndice B é apresentado um exemplo que pretende elucidar a forma de utilizar o programa MK elaborado pelo autor deste trabalho. Este exemplo inclui a descrição do formato dos ficheiros de dados e dos resultados fornecidos.

4.4 Algumas noções necessárias para o estudo das filas de espera

4.4.1 Distribuições sem memória

Neste tipo de distribuições, o comportamento actual do processo é independente dos intervalos de tempo precedentes ou subsequentes, isto é, só depende da duração do intervalo de tempo em questão. Este conceito é apresentado por diversos autores [Klein75], [King90], sendo verificado pelas funções de distribuição geométrica e exponencial. A primeira revela-se importante no estudo das filas de espera baseadas em cadeias de *Markov* de tempo discreto, ou seja, nestas cadeias o número de unidades de tempo que o sistema passa num determinado

estado é governado por uma distribuição geométrica, enquanto que a distribuição exponencial é relevante para o estudo das filas baseado em tempo contínuo.

Exemplifica-se esta propriedade para o caso de uma cadeia de *Markov* de tempo discreto, assumindo-se que esta transitou para o estado E_i . A probabilidade de se manter neste estado é denominada por p_{ii} , sendo a probabilidade de transitar para outro estado qualquer, $1-p_{ii}$. A probabilidade da cadeia permanecer neste estado após residir k instantes de tempo no estado E_i é novamente dada por p_{ii}^k ; isto é, o conhecimento de que a cadeia permaneceu k instantes de tempo no estado E_i não afecta as probabilidades em instantes futuros (*falta de memória*), assim, dado que estas probabilidades são independentes, a probabilidade de transitar para outro estado após k instantes de tempo no estado E_i é dada pela função de distribuição geométrica $(1-p_{ii})p_{ii}^k$.

Argumentos idênticos poderiam ser utilizados para mostrar que a função de distribuição exponencial rege a duração de cada estado numa cadeia de *Markov* de tempo contínuo.

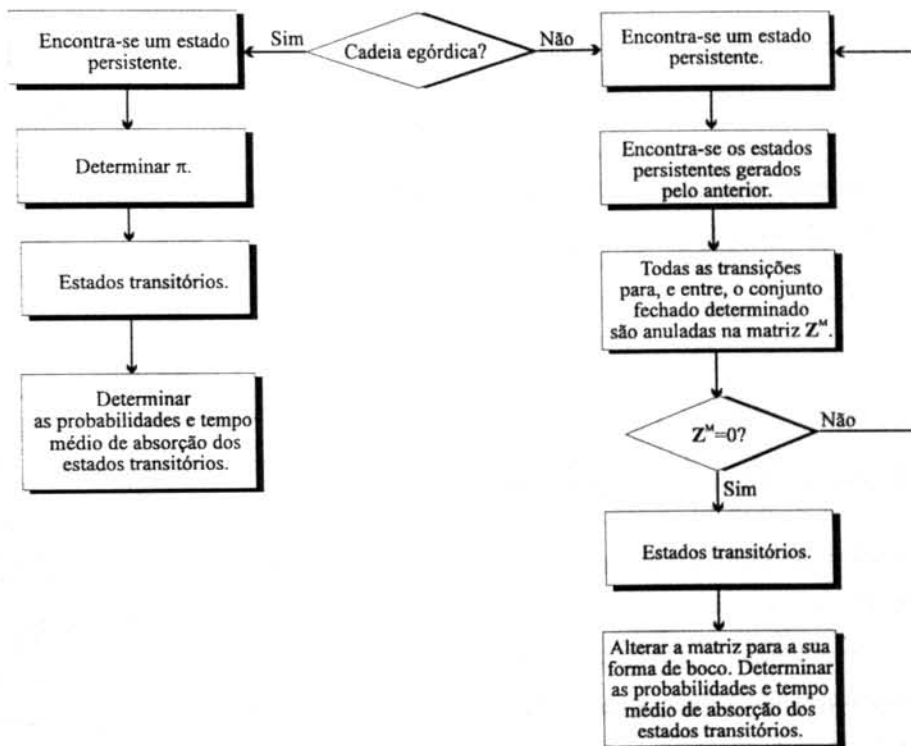


Figura 4.3 Diagrama para análise de cadeias de *Markov* finitas de tempo discreto estacionárias.

4.4.2 Funções características e funções geradoras de momentos

Entende-se por função característica de uma variável aleatória X a função $\phi_X(s)$ tal que

$$\phi_X(s) = \int_{-\infty}^{\infty} e^{jsx} f_X(x) dx. \tag{Eq. 4.31}$$

Relativamente à definição fornecida note-se que esta é igual à transformada de *Fourier* da FDP de \mathbf{X} , com uma única diferença no sinal do expoente. Uma propriedade importante resulta da expansão em série de *Taylor* da função exponencial, realizando-se a integração relativamente a cada um dos seus termos: observa-se que a função característica depende dos momentos da variável aleatória \mathbf{X} , que são definidos pelo valor esperado da potência de ordem n da variável aleatória e caracterizam estatisticamente o comportamento da variável aleatória em questão,

$$\phi_X(s) = 1 + js\bar{X} + \frac{(js)^2}{2!} \overline{X^2} + \dots \quad (\text{Eq. 4.32})$$

Anulando a variável s e calculando a derivada de ordem n da função característica para este ponto (condições estacionárias), resulta uma expressão genérica que permite obter a função característica, para este ponto, em função dos momentos da variável aleatória \mathbf{X} ,

$$\phi_X^{(n)}(0) = \left. \frac{d^n \phi_X(s)}{ds^n} \right|_{s=0} = j^n \overline{X^n}. \quad (\text{Eq. 4.33})$$

A função geradora de momentos, $M_X(s)$, é definida na Eq. 4.34, e a sua derivada de ordem n para s nulo é idêntica à derivada de ordem n da função característica, $\overline{X^n}$, no mesmo ponto.

$$M_X(s) = \int_{-\infty}^{\infty} e^{sx} f_X(x) dx \quad (\text{Eq. 4.34})$$

4.4.3 Aplicação de transformadas na determinação dos momentos

A utilização de transformadas, nomeadamente a transformada de *Laplace* (caso contínuo) e a transformada Z (caso discreto), permite a determinação dos momentos de uma variável aleatória \mathbf{T} a partir de relações muito simples. Exemplifica-se recorrendo à distribuição exponencial; nesta a variável aleatória \mathbf{T} representa o intervalo de tempo entre acontecimentos consecutivos (função de densidade exponencial) sendo $A(t)$ e $a(t)$ respectivamente a FDP e a fdp de \mathbf{T} . A notação usada para representar a transformada de *Laplace* de $a(t)$ será $A^*(s)$, enquanto que a notação para representar a transformada Z de $a(t)$ seria $A(z)$. Então

$$A^*(s) = \int_0^{\infty} e^{-st} a(t) dt = \int_0^{\infty} e^{-st} \lambda e^{-\lambda t} dt = \frac{\lambda}{s + \lambda}, \quad (\text{Eq. 4.35})$$

os momentos são determinados a partir das derivadas de ordem n da transformada de *Laplace* da função de densidade exponencial. Considerando que condições estacionárias ocorrem quando $t \rightarrow \infty$, então $s \rightarrow 0$.

$$A^{*(n)}(0) = (-1)^n \overline{T^n} \quad (\text{Eq. 4.36})$$

Logo, o primeiro e segundo momentos são

$$E[t] = -\left. \frac{dA^*(s)}{ds} \right|_{s=0} = \frac{1}{\lambda}, \quad (\text{Eq. 4.37})$$

$$E[t^2] = -\left. \frac{d^2 A^*(s)}{ds^2} \right|_{s=0} = \frac{2}{\lambda^2}. \quad (\text{Eq. 4.38})$$

É utilizado um processo idêntico quando se trabalha com variáveis aleatórias discretas, ou seja, neste caso aplica-se a transformada Z fazendo uso da função geradora de probabilidade, $G(z)$. Neste momento é clara a relação entre a função característica, a função geradora de momentos e a transformada de *Laplace* de uma variável aleatória contínua, sendo possível relacioná-las da seguinte forma:

$$\phi_X(sj) = M_X(-s) = A^*(s). \quad (\text{Eq. 4.39})$$

4.5 Sistemas de filas de espera simples

No estudo de sistemas de filas é usualmente utilizada uma notação que condensa a descrição relativa a um sistema particular, a notação de *Kendall*. Esta consiste numa sequência de caracteres e números separados em campos com significado distinto através de um símbolo, "/".

No caso geral a notação tem a estrutura $A/B/c/n/p$, em que os dois primeiros caracteres fornecem indicação quanto ao tipo de processo que descreve a chegada de novos elementos ao sistema (A) e ao tipo de serviço que é prestado (B). O número de servidores é indicado por c ; relativamente ao quarto campo, este fornece informação sobre o espaço de armazenamento de que o sistema dispõe (fila mais servidores). O último campo indica a dimensão da população de potenciais clientes.

Os dois primeiros campos podem assumir o seguinte conjunto de caracteres:

D: de determinístico - o intervalo de tempo entre clientes consecutivos e de serviço é constante;

M: de *Markov* - no caso contínuo os intervalos de tempo de serviço têm uma distribuição exponencial, e no caso discreto têm uma distribuição geométrica. O processo de chegada de novos clientes forma um processo de *Poisson*;

G: de geral - significa que não são colocadas restrições quanto ao tipo de distribuição que caracteriza o comportamento do sistema.

É comum que um dos dois últimos campos, ou ambos, não sejam representados, o que significa que se assume que sejam infinitos. Este valor não deve ser tomado no sentido literal, mas sim relativamente às dimensões particulares do sistema em causa.

4.5.1 Filas de espera do tipo nascimento-morte em equilíbrio

A primeira questão que importa estudar numa fila do tipo nascimento-morte é a existência de probabilidades estacionárias, anteriormente denominadas por π_k e que se passam a chamar p_k . Observando o grafo da Fig. 4.1, é possível por inspecção estabelecer a equação de diferenças para um estado genérico k :

$$\lambda_{k-1}p_{k-1} + \mu_{k+1}p_{k+1} = (\lambda_k + \mu_k)p_k. \quad (\text{Eq. 4.40})$$

A partir da expressão anterior, e utilizando a indução, determinam-se as probabilidades estacionárias de uma fila nascimento-morte em função da sua probabilidade inicial p_0 (Eq. 4.41). A existência deste limite coloca condições sobre o quociente entre λ_k e μ_k (ρ), denominado intensidade. A necessidade de ser menor do que a unidade reflecte a condição deste processo ser egórdico, e, por consequência, a imposição dos estados da cadeia serem persistentes. O significado desta condição está associado à possibilidade desta cadeia retornar a qualquer estado que lhe pertença, facto que não era possível se estes fossem transitórios,

$$p_k = p_0 \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \text{ e } k = 1, 2, \dots \quad (\text{Eq. 4.41})$$

Esta expressão pode ser imediatamente aplicada ao estudo de uma fila do tipo M/M/1, com $\lambda_k = \lambda$ e $\mu_k = \mu$. Os intervalos de tempo entre nascimentos e entre mortes têm distribuição exponencial, sendo os seus valores médios λ^{-1} e μ^{-1} respectivamente. A probabilidade estacionária para este sistema é descrita pela Eq. 4.42, que resulta da aplicação directa da Eq. 4.41. De notar que p_k converge se e só se $\rho < 1$.

$$p_k = p_0 \left(\frac{\lambda}{\mu} \right)^k \text{ e } k \geq 0 \quad (\text{Eq. 4.42})$$

Para determinar p_0 basta realizar o somatório para todo o domínio (k) e igualar à unidade, condição de normalização. Desta forma a probabilidade da cadeia residir no estado inicial é igual a $1 - \rho$, sendo possível representar a probabilidade estacionária do sistema residir no estado k , como se indica na Eq. 4.43. Esta indica claramente um comportamento de acordo com a distribuição geométrica (sem memória),

$$p_k = (1 - \rho)\rho^k \text{ e } k \geq 0. \quad (\text{Eq. 4.43})$$

É possível, através da escolha dos coeficientes de nascimento e morte, modificar o funcionamento da fila do tipo M/M/1 para reflectir outro tipo de comportamento, como por exemplo: é normal que os potenciais clientes sejam desencorajados de entrar para uma fila quando esta apresenta um número elevado de clientes à espera, e mais ainda se se verificar que esta está a crescer. Assim, a adesão de novos clientes, coeficiente de nascimento, irá diminuir com o crescimento de clientes que esperam na fila (Eq. 4.44),

$$\lambda_k = \frac{\alpha}{k+1} \text{ e } k \geq 0. \quad (\text{Eq. 4.44})$$

4.5.2 Filas de espera do tipo: M/M/m/K/M

Este é o caso geral das filas do tipo nascimento-morte: os processo de adesão e de serviço da fila são caracterizados por distribuições exponenciais, existem m servidores que irão servir uma população de M clientes e a capacidade de armazenamento (só na fila, não se considerando os servidores) é igual a $K-m$.

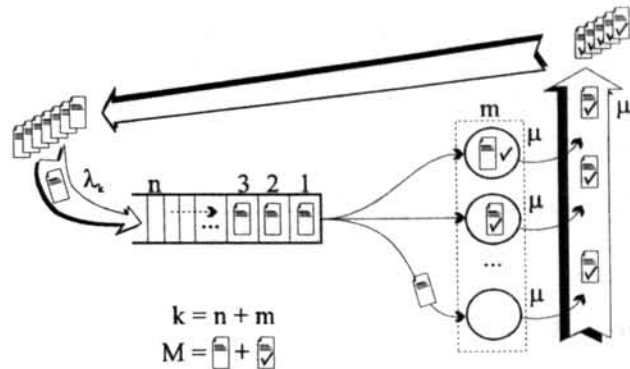


Figura 4.4 Esquema de funcionamento de um sistema baseado numa fila de espera com m servidores.

Em termos de funcionamento podem-se distinguir duas situações. A primeira quando existe um servidor disponível para o cliente que entra no sistema: nesta situação o serviço é imediato, sendo o tempo que o cliente se atrasa no sistema igual ao tempo de serviço. A outra situação contempla o caso de não se encontrar nenhum servidor livre sendo o cliente colocado à espera: o tempo de atraso neste caso é igual à soma do tempo de serviço com o tempo de espera na fila.

Nas situações descritas não foi contemplado o caso do cliente ser recusado devido a não existir um servidor livre e da fila se encontrar cheia - situação de bloqueio. O que foi exposto conduz aos seguintes coeficientes de nascimento e morte:

$$\lambda_k = \begin{cases} \lambda(M-k) & 0 \leq k \leq K-1, \\ 0 & k > K \end{cases} \tag{Eq. 4.45}$$

$$\mu_k = \begin{cases} k\mu & 0 \leq k \leq m \\ m\mu & k \geq m \end{cases} \tag{Eq. 4.46}$$

A probabilidade estacionária tem de considerar as situações apresentadas, determinando-se expressões que só são válidas dentro dos limites referidos,

$$p_k = p_0 \rho^k \binom{M}{k}, \quad 0 \leq k \leq m-1, \tag{Eq. 4.47}$$

$$p_k = p_0 \rho^k \binom{M}{k} \frac{k!}{m!} m^{m-k}, \quad m \leq k \leq K. \quad (\text{Eq. 4.48})$$

Para calcular a probabilidade inicial recorre-se à condição de normalização, sendo a relação obtida um pouco complexa. Assim, a expressão do comprimento médio da fila, \bar{N} , foi determinada em função de p_0 por forma a simplificar a escrita da mesma. Este sistema genérico pode ser utilizado para determinar a probabilidade estacionária de casos mais simples, alguns dos quais se apresentam nas secções seguintes.

$$p_0 = \left[(1+\rho)^{m-1} + \frac{m^m M!}{m!} \sum_{k=m}^K \left(\frac{\rho}{k}\right)^k \frac{1}{(M-k)!} \right]^{-1} \quad (\text{Eq. 4.49})$$

$$\bar{N} = \sum_{k=0}^K k p_k = p_0 \rho \left[(m-1)(1+\rho)^{m-2} + \frac{m^m M!}{m!} \frac{d}{d\rho} \left[\sum_{k=m}^K \left(\frac{\rho}{k}\right)^k \frac{1}{(M-k)!} \right] \right] \quad (\text{Eq. 4.50})$$

4.5.2.1 Fila de espera do tipo: M/M/m/m

Estes sistema tem m servidores, e não tem capacidade de armazenamento, ou seja, quando todos os servidores estão ocupados o cliente é rejeitado. Os coeficientes deste sistema são:

$$\lambda_k = \begin{cases} \lambda & 0 \leq k \leq m \\ 0 & k > m \end{cases}, \quad (\text{Eq. 4.51})$$

$$\mu_k = k\mu, \quad 0 \leq k \leq m. \quad (\text{Eq. 4.52})$$

Como é óbvio a existência de probabilidades estacionárias está garantida devido ao facto do sistema ser finito. Aplicando as relações já determinadas, nomeadamente a Eq. 4.41, é imediato calcular as probabilidades estacionárias assim como os restantes parâmetros. A expressão obtida para p_0 só é válida (sendo $\rho < 1$) para $m \geq 2$.

$$p_k = p_0 \rho^k \frac{1}{k!}, \quad 0 \leq k \leq m \quad (\text{Eq. 4.53})$$

$$p_0 \equiv e^{-\rho} \quad (\text{Eq. 4.54})$$

$$\bar{N} \equiv \rho^2 \quad (\text{Eq. 4.55})$$

Neste sistema, em particular, é de grande importância determinar a fracção de tempo em que todos os servidores estão ocupados, p_m ; esta expressão é denominada fórmula de perdas de Erlang (em notação, $E_1, m(\rho)$).

$$p_m \equiv \frac{\rho^m}{e^\rho m!} \quad (\text{Eq. 4.56})$$

4.5.2.2 Filas de espera do tipo: M/M/∞/M

Pode-se dizer que o conceito de número infinito de servidores está relacionado com a impossibilidade do sistema bloquear; esta afirmação baseia-se no conhecimento do número de clientes existentes (cuja população é finita). Assim, para este caso, os coeficientes de nascimento e morte são:

$$\lambda_k = \begin{cases} \lambda(M-k) & 0 \leq k \leq M \\ 0 & k > M \end{cases}, \quad (\text{Eq. 4.57})$$

$$\mu_k = k\mu. \quad (\text{Eq. 4.58})$$

Seguindo os mesmos passos do caso anterior é possível obter as expressões para as probabilidades estacionárias, a probabilidade inicial e o comprimento médio da fila, como se apresentam seguidamente:

$$p_k = p_0 \rho^k \binom{M}{k}, \quad 0 \leq k \leq M, \quad (\text{Eq. 4.59})$$

$$p_0 = \frac{1}{(1+\rho)^M}, \quad (\text{Eq. 4.60})$$

$$\bar{N} = \frac{M\rho}{1+\rho}. \quad (\text{Eq. 4.61})$$

4.5.3 Filas de espera do tipo Markov em equilíbrio

Até ao momento foram apresentadas filas baseadas em processos aleatórios do tipo nascimento-morte; é agora a altura de introduzir sistemas de filas que são descritos por processos aleatórios mais abrangentes, os processos de Markov. Neste tipo de filas é permitida a transição entre estados que não sejam vizinhos. A eliminação desta restrição não impossibilita o desenvolvimento de uma forma fechada para as probabilidades estacionárias.

Para realizar o cálculo da condição de equilíbrio é necessário introduzir um procedimento, o *método das etapas*, que permita o estudo dos sistemas de filas do tipo Markov. Este método é devido ao trabalho de A. K. Erlang, surgindo devido à falta de aplicabilidade em certos casos da distribuição exponencial para caracterizar os intervalos de tempo entre clientes consecutivos, e os de serviço, apresentados por certos sistemas.

O método das etapas é baseado na decomposição do servidor numa sequência de entidades cujas distribuições do tempo de serviço são exponenciais. Em cada etapa de serviço pode-se fazer uso da propriedade de perda de memória apresentada por esta distribuição (o tempo de serviço que resta é independente do tempo de serviço já prestado).

Considere-se o caso em que para realizar um determinado serviço é necessário executar duas operações distintas. Assim, é possível subdividir o servidor numa série de duas unidades independentes, mas que se complementam

(servidor de *Erlang* com duas etapas). Cada uma pode ser representada por um servidor cujo serviço é determinado por uma distribuição exponencial cujo valor médio é metade do inicial. No servidor de *Erlang*, tal como no original, só é admitido um novo cliente quando o anterior for removido pela última unidade de serviço.

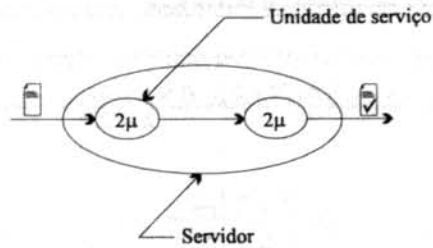


Figura 4.5 Servidor de *Erlang* com duas etapas (E_2).

A distribuição do tempo total de atraso introduzido pelo servidor de *Erlang* é obtido a partir da soma de duas variáveis aleatórias independentes, que estão associadas ao tempo de serviço de cada uma das etapas através da convolução das suas fdps. Para simplificar o cálculo aplica-se a transformada de *Laplace* às fdps, realizando o seu produto, e posteriormente o cálculo da transformada inversa. Conclui-se que a distribuição do tempo total de atraso introduzido é fornecida por

$$b(x) = 2\mu(2\mu x)e^{-2\mu x}, \quad x \geq 0. \quad (\text{Eq. 4.62})$$

A partir do conhecimento da distribuição do tempo total de serviço é possível obter os dois primeiros momentos através das propriedades da transformada de *Laplace* de $b(x)$, $B^*(s)$, e assim calcular a média e a variância dessa distribuição:

$$E[\bar{x}] = \frac{1}{x}, \quad (\text{Eq. 4.63})$$

$$\sigma_b^2 = \frac{1}{2\mu^2}. \quad (\text{Eq. 4.64})$$

De realçar que o tempo médio de serviço é o mesmo, como seria de esperar, mas a variância é metade da anterior. Estas conclusões são bastante interessantes pois permitem supor que, num servidor de *Erlang* com r etapas, a média é idêntica mas a variância será r vezes inferior à variância obtida para um servidor com uma única etapa. O coeficiente de variação, C_b , será sempre inferior à unidade, nomeadamente para o caso de r etapas (Eq. 4.65). A fdp obtida para o tempo total de serviço denomina-se por distribuição de *Erlang*, (Eq. 4.66).

$$C_b = \frac{1}{\sqrt{r}} \quad (\text{Eq. 4.65})$$

$$b(x) = \frac{r\mu(r\mu x)^{r-1} e^{-r\mu x}}{(r-1)!}, \quad x \geq 0 \tag{Eq. 4.66}$$

4.5.4 Filas de espera do tipo: M/E_r/1

Para este tipo de filas de espera considera-se um sistema cuja fdp do intervalo de tempo entre clientes consecutivos é uma distribuição exponencial e a fdp do número de etapas (do servidor) que necessitam de ser completadas é uma distribuição de Erlang (E_r). Assim, a descrição do sistema é baseada no número de etapas necessárias para completar o serviço de todos os clientes que se encontram no sistema, quer à espera quer actualmente a receber serviço. Cada cliente que se encontre na fila é responsável pela introdução de *r* etapas na variável de estado, Fig. 4.6.

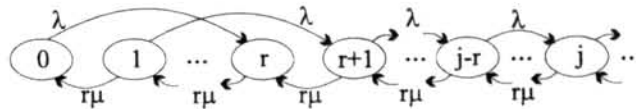


Figura 4.6 Grafo de um sistema M/E_r/1.

O número total de etapas que faltam executar para servir todos os *k* clientes do sistema será obtido considerando que um deles se encontra actualmente a ser atendido, etapa *i*, e os restantes representam (*k*-1)*r* etapas,

$$j = (k-1)r + (r-i+1). \tag{Eq. 4.67}$$

Se o número de processos no sistema é igual a *k* e se a probabilidade estacionária de tal ocorrer for definida por *p_k*, sendo *P_j* o número de etapas a executar, é possível estabelecer uma relação entre estes dois parâmetros:

$$p_k = \sum_{j=(k-1)r+1}^{kr} P_j, \quad k = 1, 2, 3, \dots \tag{Eq. 4.68}$$

Estabeleceram-se as equações de estado do sistema usando a inspecção da conservação de fluxo no estado E_j (Fig. 4.6). Para determinar a solução destas equações utilizou-se a transformada Z (*P(z)* é definida pela Eq. 4.71).

$$\lambda P_0 = r\mu P_1 \tag{Eq. 4.69}$$

$$(\lambda + r\mu) P_j = \lambda P_{j-r} + r\mu P_{j+1}, \quad j = 1, 2, \dots \tag{Eq. 4.70}$$

$$P(z) = \sum_{j=0}^{\infty} P_j z^j \tag{Eq. 4.71}$$

Para aplicar a Eq. 4.71, realiza-se a soma em todo o domínio admissível (Eq. 4.70) e multiplicam-se os termos por z^j , identificando-se os termos que resultam em $P(z)$. Para avaliar P_0 sabe-se que $P(1) = 1$, sendo necessário usar a regra de *L'Hôpital* para remover a indeterminação que resulta, sendo a transformada Z da distribuição do número de etapas que faltam realizar apresentada na Eq. 4.72.

$$P(z) = \frac{r\mu(1-\rho)(1-z)}{r\mu + \lambda z^{r+1} - (\lambda + r\mu)z} \quad (\text{Eq. 4.72})$$

A transformada inversa de $P(z)$ para valores arbitrários de r é obtida pela determinação dos zeros e pólos de $P(z)$ (expansão em fracções parciais), o que torna possível representar a função por uma soma de fracções que facilmente podem ser invertidas por inspecção, resultando a expressão seguinte:

$$P_j = (1-\rho) \sum_{i=1}^r A_i (z_i)^{-j}, \quad j = 1, 2, \dots, r, \quad (\text{Eq. 4.73})$$

$$A_i = \prod_{\substack{n=1 \\ n \neq i}}^r \frac{1}{\left(1 - \frac{z_i}{z_n}\right)}. \quad (\text{Eq. 4.74})$$

4.5.5 Filas de espera do tipo: $E_r/M/1$

O sistema em análise tem uma distribuição de *Erlang* para caracterizar a fdp do intervalo de tempo entre clientes consecutivos, sendo a fdp do tempo de serviço exponencial. Este sistema encontra-se descrito na Fig. 4.7 sendo baseado na existência de r etapas que cada cliente tem que atravessar para ser admitido no sistema. Só é possível existir um único cliente nesta estrutura de r etapas, como já se verificava no caso anterior. A variável de estado reflecte o número de etapas da estrutura de recepção, contribuindo cada cliente já em fila com r etapas.

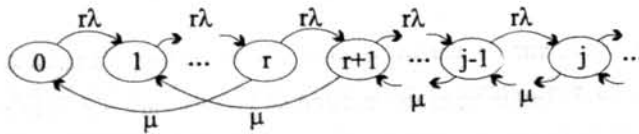


Figura 4.7 Grafo de um sistema $E_r/M/1$.

Como anteriormente, P_j é definida pela probabilidade do número de etapas para completar ser igual a j , e p_k será a probabilidade estacionária do número de processos ser igual a k , estando as duas grandezas relacionados através da Eq. 4.75. Por inspecção do fluxo do estado E_j podem-se obter as equações de equilíbrio, (Eq. 4.76) à (Eq. 4.78).

$$P_k = \sum_{j=rk}^{r(k+1)-1} P_j \tag{Eq. 4.75}$$

$$r\lambda P_0 = \mu P_r \tag{Eq. 4.76}$$

$$r\lambda P_j = r\lambda P_{j-1} + \mu P_{j+r}, \quad 1 \leq j \leq r-1 \tag{Eq. 4.77}$$

$$(r\lambda + \mu) P_j = r\lambda P_{j-1} + \mu P_{j+r}, \quad r \leq j \tag{Eq. 4.78}$$

Aplicando a transformada Z, como no caso da fila M/E_r/1, e através de procedimentos idênticos, determina-se P(z). O cálculo da função inversa é novamente obtido pelo cálculo dos pólos e dos zeros, sendo cada fracção invertida por inspeção. A expressão obtida é a distribuição geométrica com excepção do primeiro termo.

$$P_k = \begin{cases} 1-\rho & k=0 \\ \rho(z_0^r - 1)z_0^{-rk} & k>0 \end{cases} \tag{Eq. 4.79}$$

4.5.6 Relação entre as filas de espera do tipo nascimento-morte e as baseadas em processos de Markov

Numa fila do tipo M/E_r/1 cada clientes é servido através de r etapas discretas que são percorridas antes de se completar o tempo total de serviço. A este procedimento corresponde um equivalente nas filas do tipo M/M/1: num instante de Poisson é admitido, não um único cliente, mas um conjunto de r clientes, que têm um tempo de serviço fornecido por uma distribuição exponencial.

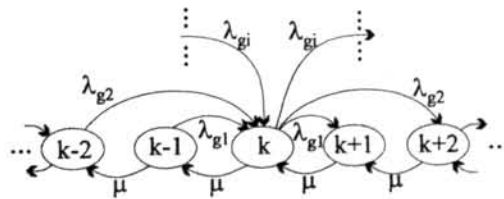


Figura 4.8 Grafo de um sistema que permite introduzir conjuntos de clientes, de dimensão variável, em instantes de Poisson.

Uma generalização do que foi exposto anteriormente é considerar a possibilidade do conjunto de clientes não ser de dimensão fixa mas aleatória, g_i. Na Fig. 4.8 é apresentado um diagrama correspondente a este sistema; concentrando a atenção no estado E_k observa-se que são admitidas entradas para este estado de qualquer estado de índice inferior. Da mesma forma é possível transitar deste estado para outro de índice superior. A transição entre dois estados contíguos ocorre sempre que o serviço de um processo é executado.

$$(\lambda + \mu)p_k = p_{k+1} + \sum_{i=0}^{k-1} p_i \lambda g_{k-i}, \quad k \geq 0 \quad (\text{Eq. 4.80})$$

$$\lambda p_0 = \mu p_1 \quad (\text{Eq. 4.81})$$

Como é usual aplica-se a transformada Z às equações às diferenças, procurando os termos que correspondem a $P(z)$ e $G(z)$, Eq. 4.82. Note-se que o termo $P(z)G(z)$ corresponde à convolução das sequências $\{p_k\}$ e $\{g_k\}$. Para se obter p_k , isto é, realizar a inversão, é necessário eliminar o termo p_0 , aplicando a condição fronteira (Eq. 4.81), e conhecer a função $G(z)$. Considerando o caso particular da dimensão do conjunto ser fixa e igual a r o resultado obtido é igual ao apresentado para a fila de espera do tipo $M/E_r/1$, como seria de esperar já que os dois sistemas apresentados são equivalentes.

$$(\lambda + \mu)[P(z) - p_0] = \frac{\mu}{z}[P(z) - p_0 - p_1 z] + \lambda P(z)G(z) \quad (\text{Eq. 4.82})$$

Fazendo uso de um procedimento idêntico é possível estabelecer uma relação entre os sistema $E_r/M/1$ e $M/M/1$ em que o serviço é prestado não numa única etapa mas num conjunto de r etapas. De igual forma, a generalização deste sistema para grupos de etapas de dimensão aleatória seria simples e imediato.

É importante referir a possibilidade de colocar sistemas do tipo $M/E_r/1$ e $E_r/M/1$ em série e em paralelo, de que resultam as distribuições hiperexponencial, ou ligadas através de outra qualquer topologia. A consideração de redes de filas do tipo *Markov* (redes de *Jackson*) fornece também resultados de grande utilidade prática [Stuck85], [King90], [Harri93].

Capítulo 5

Avaliação do Desempenho de Redes RDIS-BL/ATM

5.1 Introdução

O panorama das redes de comunicação actuais caracteriza-se pela especialização no transporte de informação proveniente de um único tipo de serviço. Os custos inerentes à excessiva especialização reflectem-se num conjunto de desvantagens das quais se destacam: a inflexibilidade na modernização das redes de comunicação possibilitada pelos avanços tecnológicos verificados, a ineficiência na utilização dos recursos disponíveis e a dependência de um único tipo de serviço, que não permite o transporte de outros para os quais a rede não foi projectada.

A introdução do conceito de rede de comunicação com integração de serviços (RDIS) representou um primeiro passo na direcção da realização de uma única rede universal para o transporte da informação proveniente de serviços que requerem uma largura de banda reduzida (voz e dados). O progresso tecnológico realizado em diversas áreas da ciência (semicondutores, óptica e arquitectura das redes de comunicação), assim como a crescente procura de novos serviços por assinantes domésticos e empresariais, vieram possibilitar e determinar a evolução da RDIS para RDIS de banda larga (RDIS-BL).

O modo de transferência assíncrono (ATM) surge como o mais adequado para a rede RDIS-BL, esta escolha deve-se à elevada flexibilidade obtida resultante do transporte da informação segmentada em pacotes de dimensão fixa (células), assim como à maior rentabilização da rede devido à partilha dinâmica dos seus recursos.

A ligação do utilizador (rede privada) à RDIS-BL está normalizada e denomina-se UNI (*User-Network Interface*). Esta é relativa ao ponto de referência T_B e oferece uma única ligação física que suporta um conjunto de ligações virtuais (caracterizadas por identificadores denominados VPI/VCI - *Virtual Path Identifier/Virtual Channel Identifier*), em que cada uma transporta informação resultante dos mais diversos tipos de serviços, orientados às ligações ou não.

A enorme variedade de informação transportada por uma RDIS-BL reflecte-se nas múltiplas classes de tráfego definidas para satisfazer a capacidade e modo de transferência, assim como nos objectivos de desempenho propostos para cada ligação ATM. O utilizador pode requisitar até duas classes de tráfego distintas numa única ligação ATM; estas são possíveis dada a existência de duas prioridades relativas à remoção de células, CLP (*Cell Loss Priority*). A qualidade de serviço (QoS), tal como pode ser avaliada na camada ATM, é consequência do

mapeamento dos objectivos de desempenho solicitados à camada ALL (*Adaptation ATM Layer*) no ATM SAP (*ATM Service Access Point*), e, conseqüentemente, dos parâmetros de desempenho de uma aplicação em particular.

Os elementos que pertencem a uma determinada ligação ATM têm que respeitar a qualidade de serviço previamente contratada. Para caracterizar as situações em que não é possível respeitar os parâmetros de qualidade previamente acordados, quer para novas ligações quer em ligações já existentes, define-se, no âmbito do CCITT, *congestão*. Esta situação existe para os elementos da rede que se encontrem sob tais condições, podendo ser consequência de falhas da própria rede ou devido a flutuações imprevisíveis no tráfego gerado.

O controlo do tráfego, e por consequência o controlo das situações de congestão, é realizado na camada ATM por dois blocos funcionais distintos: controlo de tráfego e controlo de congestão. Estes distinguem-se por executarem um conjunto de acções que têm por finalidade impedir o aparecimento de situações de congestão (controlo de tráfego) e, quando estas se verificarem, minimizar o seus efeitos (controlo de congestão).

O desempenho de uma ligação ATM é consequência dos elementos que a constituem, sendo o ATM uma técnica de comutação/multiplexagem o seu desempenho é necessariamente determinado pelos comutadores VP/VC (*VP/VC switching*) e pelos multiplexadores. Neste capítulo alguns métodos analíticos para a avaliação de ambos serão apresentados.

Um aspecto relevante no estudo do desempenho das ligações ATM é a verificação da qualidade de serviço contratada pelo utilizador (monitoração da QoS), procedimento que deve ser realizado sem afectar a QoS medida. Este tema assume importância vital para o bom funcionamento de uma rede ATM; assim o estudo da arquitectura e funcionalidade das entidades destinadas a executarem tais mecanismos (controlo de tráfego e de congestão) é também apresentado neste texto.

5.2 Características do desempenho de uma ligação ATM

Sendo o ATM baseado no transporte de pacotes de dimensão fixa, distingue-se dos tradicionais modos de transferência orientados à comutação de pacotes por não realizar controlo de erro e fluxo nos nós intermédios de uma ligação. Estes tipos de mecanismos não são necessários devido à elevada qualidade do meio de transmissão e ao modo como a rede será gerida, que permitem (prevê-se) obter valores extremamente baixos para a probabilidade de perda de células (10^{-8} a 10^{-12}). Embora não sejam tomadas acções para a detecção e recuperação da informação perdida (CRC ou ARQ), estão previstos procedimentos preventivos para minimizar tais ocorrências. Estes incluem a reserva de recursos apropriada ao estabelecimento da ligação e a monitoração da QoS (controlo de tráfego).

Durante a fase de estabelecimento de uma ligação a rede determina a disponibilidade dos recursos necessários para suportar o tráfego (previamente caracterizado) e a QoS requisitada pelo utilizador. No caso de se verificar a possibilidade do estabelecimento de tal ligação os recursos necessários são reservados e afectados à ligação. É esta propriedade, serviço orientado às ligações, que permite manter dentro de certos limites o número de células

perdas, minimizando-se a possibilidade de se exceder a capacidade de armazenamento em qualquer elemento que pertença à ligação.

Pelo que foi exposto é fácil compreender a reduzida dimensão do cabeçalho de uma célula ATM: a sua funcionalidade é quase exclusivamente reservada ao estabelecimento do caminho/canal virtual (VP/VC) a ser usado no transporte da informação (encaminhamento estático). Os identificadores VPI/VCI não representam um endereço explícito mas sim um identificador local estabelecido em consonância com as tabelas de encaminhamento. A utilização de dois identificadores significa que o encaminhamento de uma ligação ATM é efectuado em dois níveis distintos, o que permite simplificar o processo, permitindo a comutação simultânea de todos os canais virtuais pertencentes ao mesmo caminho virtual.

O estabelecimento de uma ligação ATM é executado fazendo uso de canais ATM específicos denominados SVCC (*Signalling Virtual Channel Connection*). Estes são estabelecidos ponto-a-ponto entre o terminal e a entidade responsável pela atribuição dos recursos da rede, sendo a atribuição de um canal virtual de sinalização realizada durante a fase de activação do terminal (procedimentos de meta-sinalização). Estas funções fazem parte do plano de controlo do modelo de referência adoptado pelo CCITT.

Uma descrição mais detalhada sobre ATM pode ser obtida nos seguinte textos, que constituem um conjunto de recomendações elaboradas pelo CCITT:

- I.113 *Vocabulary of Terms for Broadband aspects of ISDN*, 1992
- I.121 *Broadband aspects of ISDN*, 1991
- I.150 *BISDN ATM Functional Characteristics*, 1992
- I.211 *BISDN Service Aspects*, 1991
- I.311 *BISDN General Network Aspects*, 1992
- I.321 *BISDN Protocol Reference Model and its Application*, 1990
- I.327 *BISDN Network Functional Architecture*, 1990
- I.361 *BISDN ATM Layer Specification*, 1992
- I.362 *BISDN ATM Adaptation Layer (AAL) Functional Description*, 1991
- I.363 *BISDN ATM Adaptation Layer (AAL) Specification*, 1990
- I.371 *Traffic Control and Congestion Control in BISDN*, 1992
- I.413 *BISDN User-Network Interface*, 1990
- I.432 *BISDN User-Network Interface - Physical Layer Specification*, 1991
- I.610 *BISDN Operation and Maintenance Principles and Functions*, 1992.

5.2.1 Transparência temporal

O atraso verificado num determinado serviço, devido ao transporte através de uma rede de comunicação, é reflexo do desempenho dos elementos que a constituem e que intervêm no transporte da informação. Numa ligação (unidireccional) puramente ATM (Fig. 5.1a) a informação no extremo origem é transformada

(empacotada) por forma a ser transportada em células ATM; no extremo destino existirá um processo idêntico mas simétrico com o intuito de recuperar a informação original.

Relativamente ao transporte da informação, sob a forma de células ATM, através de um meio físico baseado numa rede síncrona (SDH) (Fig. 5.1b), é necessário processamento adicional para adaptar o fluxo de células ao formato utilizado pela SDH. O facto da dimensão do campo de informação de uma trama SDH não ser múltiplo da dimensão de uma célula ATM, o que implica alterações no algoritmo de delineação da célula, é um exemplo de um procedimento adicional a efectuar.

Os vários parâmetros, que contribuem para o atraso total verificado, estão identificados nas figuras apresentadas, bem como o local a que estão associados. Na Tabela 5.1 [Pryck91] apresentam-se os valores estimados para cada um dos parâmetros, considerando as diferentes velocidades de transmissão que se prevê venham ser adoptadas.



Figura 5.1a Atrasos numa rede ATM (AE- Atraso de Empacotamento; AT- Atraso de Transmissão; AC- Atraso na Comutação; AD- Atraso de Desempacotamento).

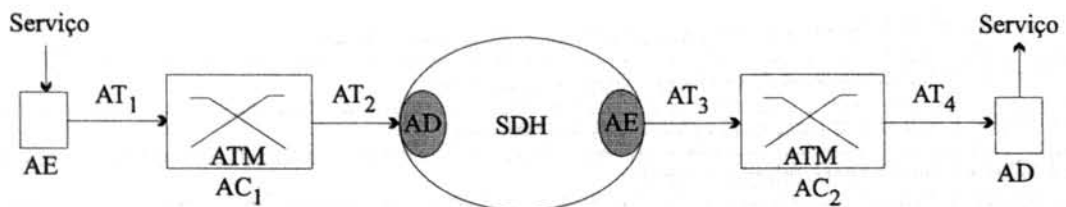


Figura 5.1b Atrasos numa rede que não é baseada em ATM (AE- Atraso de Empacotamento; AT- Atraso de Transmissão; AC- Atraso na Comutação; AD- Atraso de Desempacotamento).

Atraso na Transmissão (AT) - é variável e depende da distância e meio de transmissão utilizado (é independente do modo de transferência).

Atraso de Empacotamento (AE) - é necessário distinguir duas situações: a primeira resulta da conversão (numa das extremidades da ligação) da informação de um determinado serviço em células, e é dependente deste; a segunda situação resulta da conversão de formatos em pontos interiores da ligação, que representam a fronteira entre o meio ATM e o meio não ATM, e também neste caso o atraso é dependente do tipo de rede em presença.

Atraso na Comutação (AC) - é constituído por duas parcelas: uma delas é fixa (PF) e resulta do encaminhamento que é necessário executar baseado no conhecimento dos identificadores VPI/VCI; a segunda parcela é devida ao tempo que a célula se encontra armazenada nas filas de espera do comutador, sendo este atraso variável (PV) porque depende do tráfego imposto ao comutador.

Atraso de Desempacotamento (AD) - tempo associado à reconstituição do formato da informação original, quer num extremo quer em pontos interiores à ligação (ATM/não ATM). O valor assumido por esta grandeza no local de destino da informação pode ser elevado (relativamente aos restantes atrasos), sem que isso seja prejudicial à aplicação, por forma a minimizar o carácter aleatório da atraso total (*jitter*).

Tabela 5.1 Atraso (μ s) para diferentes comprimentos da informação e velocidade de transmissão. Estes valores foram obtidos com a ligação a suportar uma carga igual a 80% da sua capacidade, e com a probabilidade de perdas igual a 10^{-10} . A distância entre o nó de acesso e de destino é igual a 1000 km, assume-se que o atraso total na rede síncrona igual a 900 μ s, e que um total de 8 comutações de VP/VC foram realizadas.

Velocidade (Mbits/s)	155.52			622.08		
Informação (bytes)	16	32	64	16	32	64
AT	4000	4000	4000	4000	4000	4000
AC (PF)	64	128	256	16	32	64
AC (PV)/AD	200	400	800	50	100	200
AE	2000	4000	8000	200	4000	8000

5.2.2 Transparência semântica

Devido à particularidade das redes baseadas em ATM transmitirem pacotes de dimensão fixa através de ligações pré-estabelecidas e fixas durante toda a sessão de transferência de informação, o estudo do comportamento dos erros resume-se aos seguintes pontos: erros em *bits* de informação devido ao meio de transmissão, células removidas nos multiplexadores/comutadores devido a não existir capacidade de armazenamento disponível nas suas filas de espera e células perdidas ou inseridas devido a erros de encaminhamento.

A análise relativa ao primeiro ponto não é relevante devido a ser independente do modo de transferência usado. Na Fig. 5.2 é apresentado um fluxograma que descreve as consequências do aparecimento de erros no cabeçalho das células ATM.

O dimensionamento apropriado das filas de espera, que se encontram nos comutadores/multiplexadores, pode minimizar o número de células removidas do fluxo de informação por momentaneamente não existir capacidade de armazenamento disponível. Para realizar esta análise é necessário avaliar qual é o valor máximo do número de células removidas sem que a qualidade de serviço oferecida seja afectada de modo visível. A partir deste valor é possível determinar o valor médio do comprimento da fila utilizando, por exemplo, as ferramentas introduzidas no capítulo anterior. Este tema será novamente abordado na secção 5.6.

O número de células perdidas e inseridas são termos que reflectem, na óptica do utilizador, situações em que as células foram encaminhadas para um destino diferente do pretendido (perdeu células que lhe eram destinadas e recebeu células que não lhe eram destinadas), devido a erros no cabeçalho resultantes do meio de transmissão. Assim, um erro num *bit* no campo VCI/VPI pode originar dois tipos de erros ao nível da célula (multiplicação dos erros): o utilizador não recebeu a informação que foi recebida por outro destinatário (caso o VPI/VCI resultante não exista a célula é removida originando um único erro ao nível da célula). Um estudo efectuado que contemple este e outros tipos de situações é apresentado em [Pryck91].

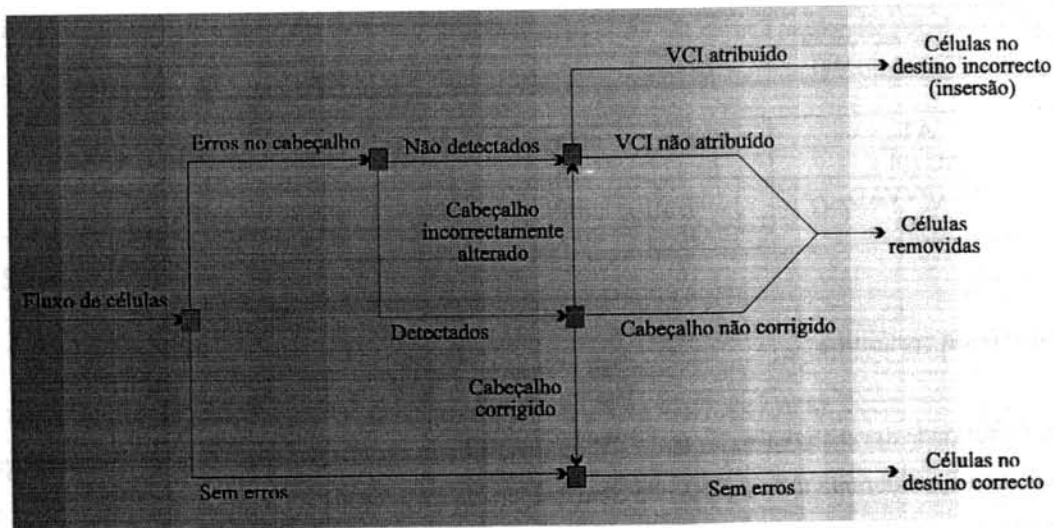


Fig. 5.2 Consequências dos erros no cabeçalho de uma célula ATM.

5.3 Mecanismos de controlo de situações de congestão em ATM

O ATM tem a potencialidade de gerir a largura de banda disponível com elevada eficiência através do uso de multiplexagem estatística. Ou seja, o transporte da informação resultante de serviços com requisitos distintos, que se reflectem em tráfego de ritmo variável (fontes do tipo ON/OFF), é realizado num canal cuja capacidade é inferior à soma dos valores de pico do tráfego gerado por cada fonte. A desvantagem do uso desta técnica, face à multiplexagem determinística, é a possibilidade de perda da informação e o atraso variável na transmissão da mesma. Para que o uso da multiplexagem estatística resulte em ganhos que superem as desvantagens, é necessário que as características estatísticas do tráfego gerado permitam que os recursos disponíveis sejam utilizados pelos diversos serviços sem que resultem situações de congestão.

O dimensionamento da capacidade de armazenamento de qualquer elemento de uma rede de comunicação é vital para o seu desempenho. Assim, a análise da situação de congestão num multiplexador, assumindo que o tráfego é gerado por fontes do tipo ON/OFF [Rober91], depende da dimensão da capacidade de armazenamento, sendo composta por duas componentes distintas denominadas: componente de célula, e componente de *burst*. A primeira resulta da recepção simultânea de células de informação proveniente de fontes independentes, mas que não excedem a capacidade total de transmissão, e pode ser definida pela probabilidade conjunta dos seguintes acontecimentos: {capacidade de armazenamento nula e ritmo de recepção menor que a capacidade de transmissão}. A segunda resulta do facto de momentaneamente o pico do ritmo de tráfego gerado superar a capacidade de transmissão, e pode ser definida pela seguinte probabilidade conjunta: {capacidade de armazenamento nula e ritmo de recepção superior à capacidade de transmissão}. As consequências da congestão por *burst* são bastante mais graves, introduzindo atrasos que se traduzem na perda da qualidade de serviço oferecida.

Na Fig. 5.3 é apresentada a dependência entre a probabilidade de bloqueio e diversos parâmetros, nomeadamente: intensidade de tráfego, duração do *burst* de tráfego (valor médio) e efeito da distribuição da duração da *burst* (exponencial e determinística).

Como consequência natural do que foi referido anteriormente, o conhecimento prévio das características do tráfego gerado por um utilizador irá permitir uma correcta avaliação relativamente à possibilidade de estabelecimento da ligação requerida. Dos parâmetros que tipicamente caracterizam o tráfego de fontes ON/OFF consta a duração média e as distribuições do tráfego gerado no estado activo (ON) e silencioso (OFF), bem como a correlação entre sucessivos estados ON e OFF. Estes parâmetros, entre outros, podem constar de uma lista relativa ao contrato que é realizado entre a rede e o utilizador, que é fornecida pelo utilizador à entidade responsável pelo estabelecimento da ligação (controlo de admissão). Esta lista descreve as exigências do ponto de vista do tráfego e da QoS (que devem ser respeitadas durante a ligação); os parâmetros que nela estão incluídos devem ser *suficientes* para avaliar se existem recursos disponíveis para suportar a ligação, devem ser de *interpretação* fácil pelo utilizador e devem conter grandezas que sejam *controláveis* pela rede por forma a verificar a validade do contrato durante a ligação (policiamento).

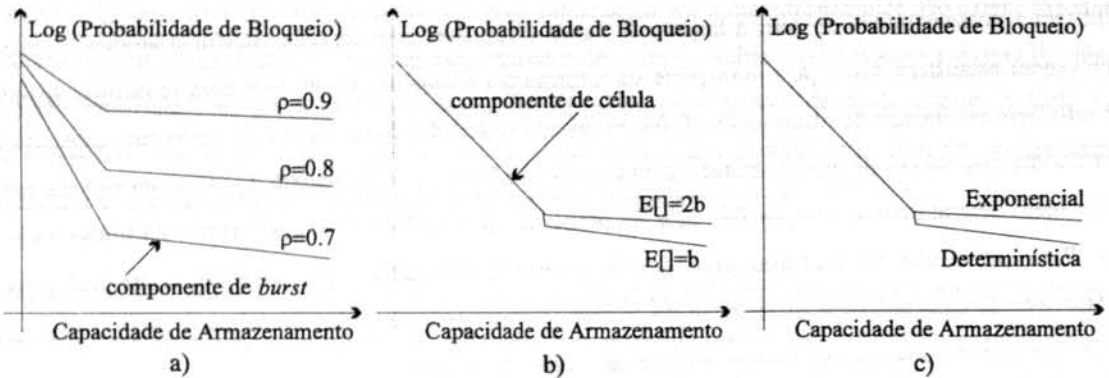


Figura 5.3 Efeito das características do tráfego sobre a probabilidade de bloqueio de um multiplexador, a) efeito da intensidade de tráfego, b) efeito do valor médio da duração dos *bursts* de tráfego, c) efeito da distribuição da duração dos *bursts* de tráfego.

O ATM partilha com as redes convencionais de comutação de pacotes, conceitos e modos de funcionamento que foram inicialmente desenvolvidos e aplicados nestas últimas. Destes aspectos um que foi sujeito a intensivo estudo diz respeito ao controlo de situações de congestão e desse estudo resultou um conjunto de técnicas que se encontram hoje em uso com bons resultados, embora o trabalho desenvolvido anteriormente não possa ser aplicado directamente às elevadas velocidades de transmissão associadas à RDIS-BL/ATM.

De seguida é apresentada a visão do CCITT relativamente aos aspectos de congestão e controlo de tráfego para a RDIS-BL/ATM, estudo este que pode ser confrontado com trabalhos anteriores relativos à implementação de um sistema distribuído de controlo de tráfego e de congestão [Burgi91], [Eckbe91], [Gilbe91], [Habib91], [Okada91].

5.3.1 Modelo de referência adoptado pelo CCITT

No âmbito das funções necessárias para suportar os mecanismos de controlo e gestão em redes baseadas no modo de transferência assíncrono, vão ser apresentados os blocos funcionais já identificados pelo CCITT, e que fornecem actualmente a base de trabalho para os desenvolvimentos realizados no campo da avaliação do desempenho de uma rede baseada em ATM.

Network Resource Management (NRM) - executa funções inerentes à reserva de recursos com a finalidade de separar o tráfego de acordo com as suas características, entre as quais se pode citar a especificação do tipo de classe a que cada serviço pertence.

Connection Admission Control (CAC) - é definido pelo conjunto de acções realizadas pelos elementos da rede durante a fase de estabelecimento, ou posterior re-negociação, de uma ligação para determinar se esta pode ou não ser aceite. Uma das funções que lhe pertence é o encaminhamento das ligações.

Usage/Network Parameter Control (UPC/NPC) - realiza funções de monitoração e controlo do tráfego nos pontos de acesso definidos pela RDIS-BL: UNI (*User Network Interface*) e NNI (*Network Node Interface*).

Com o intuito de regular o tráfego estão estabelecidas formas de realimentação da informação entre os elementos da rede e o utilizador. Este procedimento permite controlar o tráfego submetido à rede por forma a estar de acordo com o estado dos seus elementos. A utilização do *bit* de prioridade (CLP) poderá ser útil nos elementos da rede que se encontrem em congestão: nestes, parte do tráfego (células com baixa prioridade) poderá ser removida com o objectivo de manter os parâmetros de desempenho dos serviços correspondentes às células com prioridade superior. No esquema da Fig. 5.4 é apresentada a configuração de referência tal como é descrita na recomendação I.371 do CCITT.

O tráfego numa ligação ATM é descrito, qualitativamente e quantitativamente, através de uma lista de parâmetros (*ATM Traffic Descriptor*). Durante a fase de estabelecimento de uma ligação o utilizador fornece à rede os elementos necessários para descrever as características da ligação requerida (STD - *Source Traffic Descriptor*). Estes são utilizados pelo CAC para reservar os recursos necessários para suportar a ligação, e para determinar os parâmetros que irão caracterizar o funcionamento dos módulos UPC/NPC. É com base nos dados fornecidos pelo STD, na QoS e conhecimento da tolerância estipulada para a variação máxima do atraso das células, CDV (*Cell Delay Variation*), que é definido o contrato, no ponto de referência T_B , entre o utilizador e a rede. Por parte da rede, os intervenientes na decisão do estabelecimento de uma ligação são os módulos CAC e *UPC/NPC*.

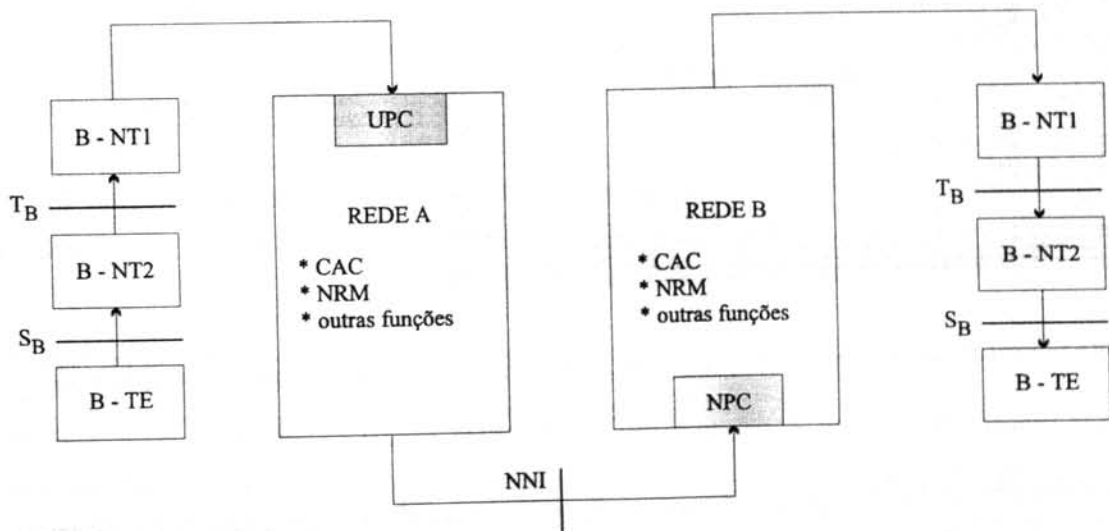


Figura 5.4 Configuração de referência para o controlo de tráfego e congestão numa rede ATM.

No parágrafo anterior foi introduzido um parâmetro relativo à variação máxima do atraso admitida pelas células numa ligação ATM (CDV); o conhecimento desta restrição implica que a atribuição dos recursos por parte do UPC/NPC seja realizada tendo em consideração este parâmetro. Situações que se reflectem em CDV encontram-

se ao nível das camadas ATM e ALL: a multiplexagem de células de diferentes ligações e a introdução de células do tipo OAM (*Operation And Maintenance*) na camada ATM irá alterar de uma forma aleatória o atraso imposto a um determinado conjunto de células. Ao nível da camada de adaptação (AAL) pode-se realizar o *agrupamento* de serviços, o que irá originar atrasos variáveis nas células transmitidas. A aplicação de certas operações (*Traffic Shaping*) sobre o fluxo de células pode minorar os efeitos de CDV; um exemplo é o espaçamento dinâmico de células realizado a partir do conhecimento do valor de pico do ritmo de transmissão.

A normalização da tolerância (CDV) será aplicada entre certos pontos de referência (medida extremo-a-extremo) e será obtida através de um conjunto discreto de valores. A escolha de um valor para a CDV é realizada no estabelecimento da chamada ou então através de assinatura.

Entre os parâmetros que constam da lista de elementos fornecida pelo utilizador à rede inclui-se o valor de pico do ritmo de transmissão numa VPC/VCC (*Virtual Path Connection/Virtual Channel Connection*). A sua definição, recorrendo à configuração de referência adoptada pelo CCITT para definição do mesmo, é relativa ao ponto de acesso do nível físico (PHY SAP) de um terminal equivalente (Fig. 5.5), configuração que é utilizada como representação de uma VPC/VCC. O valor de pico é obtido através do inverso do intervalo de tempo mínimo entre dois pedidos consecutivos de transmissão de ATM_PDUs (*ATM Protocol Data Unit*).

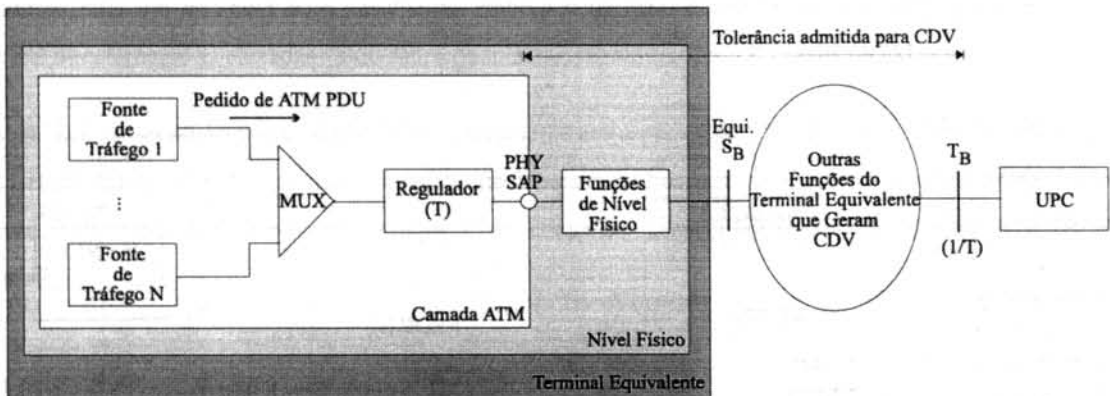


Figura 5.5 Configuração de referência para a definição do valor de pico do ritmo de transmissão numa ligação ATM.

Como cada ligação ATM pode suportar dois tipos de classes de tráfego distintas, que corresponde a diferentes valores do campo CLP, assim como o fluxo de células OAM, é necessário estabelecer o valor de pico do ritmo de transmissão para cada tipo de fluxo de células. Só desta forma é que os recursos necessários para o caminho e canal virtuais são correctamente atribuídos.

Do que foi exposto pode-se concluir que as funções de controlo de tráfego e de congestão têm por objectivo garantir a qualidade de serviço previamente contratada; neste âmbito a atribuição de tarefas tal como é descrita

pelo CCITT (Tabela 5.2) transparece a aplicabilidade de cada uma dessas à existência ou não de uma situação de congestão.

Tabela 5.2 Funções relativas ao controlo de tráfego e de congestão.

Controlo de tráfego	Controlo de congestão
<ul style="list-style-type: none"> • <i>NRM</i>, gestão de recursos da rede • <i>CAC</i>, controlo de admissão de ligações • <i>UPC/NPC</i>, controlo e monitoração de ligações • Controlo de células baseado na sua prioridade • Regulação do tráfego • Gestão de recursos em tempo real 	<ul style="list-style-type: none"> • Remoção de células • Indicação de congestão

5.3.2 Funções e procedimentos realizados pelo NRM

Outras funções realizadas por este bloco funcional, para além das descritas na secção anterior, englobam a distribuição de mensagens de controlo, num determinado caminho virtual, quando, por exemplo, ocorrem situações de congestão. A gestão dos recursos da rede por forma a simplificar as funções realizadas por outros blocos de controlo de tráfego (*CAC* e *UPC/NPC*), é também um dos objectivos deste bloco.

A utilização das VPCs no controlo de tráfego e na gestão dos recursos irá simplificar os procedimentos a executar por estas entidades [Burgi91]. A aplicação dos VPCs, no controlo do tráfego, poderá aumentar a eficiência dos blocos funcionais *UPC/NPC* através da agregação das VCCs, pertencentes a uma determinada ligação, no mesmo caminho virtual. Assim, as acções executadas quer pelo *UPC* quer pelo *NPC* são aplicadas ao tráfego que percorre a VPC e não a cada VCC que lhe pertença. Também na gestão dos recursos a utilização de VPCs irá permitir simplificar a atribuição de VCCs, só sendo necessário avaliar a possibilidade do estabelecimento de tais ligações nos elementos da rede que terminam as VPCs.

5.3.3 Funções e procedimentos realizados pelo CAC

O *CAC* executa funções relativas à negociação das ligações (Fig. 5.6), que são estabelecidas quando é possível respeitar a qualidade de serviço requerida, assegurando que tal facto não implique a quebra da QoS contratada para as restantes (já previamente estabelecidas). Assim, é necessário garantir a QoS contratada para todas as ligações existentes (extremo-a-extremo), ou seja, como as ligações num ambiente RDIS-BL são constituídas por diversos segmentos, o que significa diversas VPCs e VCCs, é necessário garantir o desempenho de cada VPC/VCC.

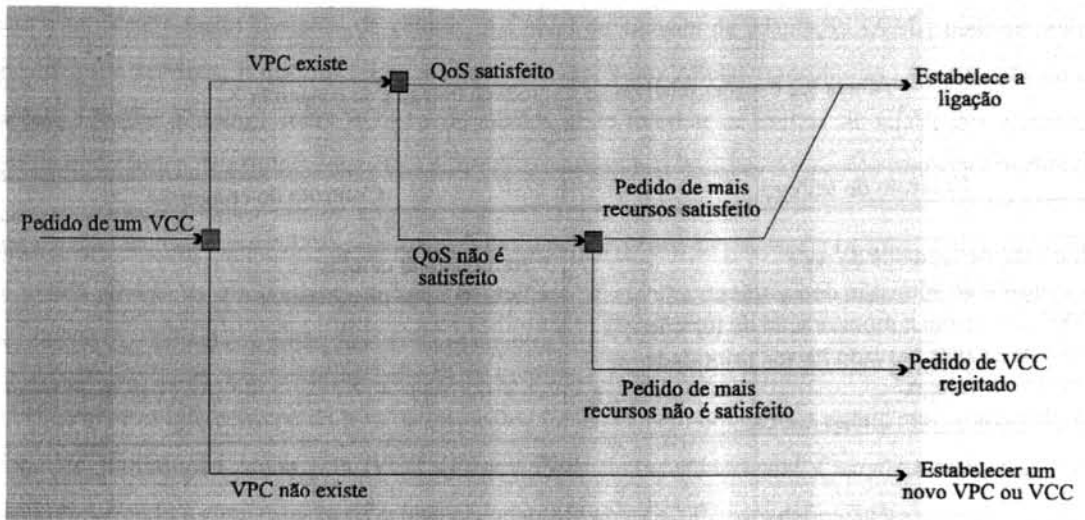


Figura 5.6 Estabelecimento de uma ligação usando o conceito de VP

5.3.4 Funções e procedimentos realizados pelo UPC/NPC

As funções atribuídas a estes blocos são semelhantes, sendo executadas em pontos de *interface* distintos, nomeadamente na UNI (UPC) e na NNI (NPC). As funções de ambos são relativas à monitoração e controlo do tráfego (policimento) com o intuito de detectar violações da QoS contratada nas ligações estabelecidas, o que inclui tanto canais VPC/VCC como canais de sinalização. A monitoração de canais de meta-sinalização e gestão (OAM) encontra-se em fase de estudo pelo CCITT.

Para realizar as funções descritas é necessário executar um conjunto de procedimentos que satisfaçam os seguintes requisitos:

- Capacidade de detectar situações de uso abusivo relativo ao previamente contratado (por exemplo, tráfego gerado superior ao valor de pico contratado);
- Existência de tolerâncias para determinados parâmetros;
- Tempo de resposta limitado na detecção e comunicação de violações;
- Simplicidade de realização.

Para avaliar o desempenho destes mecanismos foram identificadas dois tipos de medidas, que contemplam dois aspectos importantes na acção destes módulos, nomeadamente a rapidez na determinação das ligações que não respeitam a QoS contratada e a transparência das acções realizadas para as ligações que cumpram o contrato estipulado. Eventualmente poderão ocorrer situações em que, erradamente, acções sejam executadas sobre

ligações que cumpram a QoS estipulada, este tipo de degradação tem que estar previsto no desempenho global da rede.

Os parâmetros que estão sujeitos à acção destas unidades são os que se encontram descritos na lista de elementos fornecida pelo utilizador no estabelecimento da ligação. Este tipo de acções refletem-se no fluxo de células nas ligações ATM da seguinte forma: se uma ligação (VPC/VCC) excede o fluxo de tráfego contratado então as células que lhe pertencem podem ser marcadas (alteração da prioridade) ou mesmo removidas. Relativamente às ligações (VPC/VCC) que cumpram os parâmetros estabelecidos no contrato, o fluxo de células pode eventualmente ser modificado (*Traffic Shaping*), mas mantendo as características requisitadas pelo utilizador.

5.4 Avaliação do desempenho de ligações ATM usando mecanismos de gestão

Existem situações, de que a congestão e as falhas são exemplos, que conduzem à degradação do desempenho de uma rede de comunicação. Para prevenir e controlar estas situações é necessário estabelecer mecanismos para supervisionar, testar e avaliar o desempenho das ligações existentes. No âmbito da RDIS-BL/ATM, um conjunto de recomendações sobre o funcionamento e gestão estão a ser estudadas por diversas entidades de normalização, nomeadamente o CCITT (séries I.600).

Numa rede baseada em ATM os mecanismos de gestão são relativos às VPCs/VCCs e incluem o teste, a avaliação do desempenho, e a detecção e sinalização de falhas. Naturalmente que estes procedimentos não serão executados em todos às VPCs/VCCs existentes, mas sim naquelas que devido às suas características, nomeadamente duração, permitam a recolha de um conjunto de dados significativo ou a execução de acções de gestão. Assim, são as ligações virtuais permanentes (PVCs) as indicadas para a aplicação dos conceitos apresentados nesta secção.

A gestão das ligações virtuais (camada ATM) faz uso de um tipo de células específicas denominadas OAM, que formam um fluxo de informação bidireccional designado por F4, no caso das VPCs, e por F5 no caso das VCCs. Estes fluxos existem entre elementos que realizem funções de terminação quer de VPCs quer de VCCs (o tráfego correspondente a cada fluxo é constituído só por células OAM). As células destes fluxos distinguem-se das restantes através de valores particulares no seu cabeçalho: no caso do fluxo do tipo F4 as células diferenciam-se por alguns valores reservados para o campo VCI, e no caso do fluxo do tipo F5 as células distinguem-se pelo valor do campo PTI (*Payload Type Identifier*). Os fluxos dos tipos F4 ou F5 são caracterizados por serem estabelecidos extremo-a-extremo, entre pontos que terminam a ligação virtual VP/VC, ou de segmento, entre pontos que pertencem à mesma VPC ou a uma associação de VPCs que estejam abrangidas pela mesma administração de rede.

Dos mecanismos de gestão apresentados é a avaliação do desempenho numa VPC/VCC que se enquadra no âmbito deste texto e por esse motivo, são descritos seguidamente os procedimentos realizados por esta função.

A avaliação do desempenho de uma ligação VPC/VCC é realizada através da transmissão de células OAM entre os pontos extremos da ligação ou entre pontos extremos de um segmento. Antes da transmissão se estabelecer é

efectuado um pedido de activação da ligação, por parte de um dos extremos, sendo esta confirmada, ou não, pelo outro extremo da ligação (ou segmento).

Uma célula OAM é gerada a partir de um código de detecção de erro obtido a partir de um bloco de informação cuja dimensão N é fixa (embora seja admitida uma variação de 50% para minimizar o atraso da informação), estando normalizado um conjunto discreto de valores (128, 256, 512 e 1024). Este valor é acordado entre os dois extremos durante a fase de activação de uma sessão de avaliação de desempenho. O extremo que recebe o fluxo de informação realiza o cálculo do código de detecção de erro que é posteriormente comparado com o que está contido na célula OAM enviada na posição imediatamente a seguir ao bloco de informação cuja dimensão é conhecida. Para realizar a comparação dos dois códigos pode-se recorrer ao cálculo da distância de *Hamming*, conseguindo-se obter uma grandeza que indica os erros introduzidos durante a transmissão.

Tabém é possível obter uma noção do atraso introduzido no transporte da informação se na célula OAM for introduzida uma referência temporal (*Time Stamp*); no entanto, para ser possível efectuar os cálculos do atraso entre os dois extremos, é necessário que estes estejam sincronizados. O resultados podem ser comunicados do extremo de destino para o extremo de origem utilizando os fluxos F4 e F5 (bidireccionais), podendo o relatório emitido ser consultado por todos os nós que pertencam à ligação (segmento).

Uma facilidade importante fornecida pela camada ATM é a capacidade de *loopback* das células ATM: a célula é transmitida dum ponto acessível pertencente à VPC/VCC sendo redireccionada num outro ponto da ligação para o inicial, sem quebra da ligação virtual. Esta capacidade é suportada por um conjunto de campos cujos valores, entre outras utilizações, identificam quais as células OAM que devem ser ou não redireccionadas num determinado ponto da ligação. Um dos potenciais usos desta facilidade é a realização de medidas relacionadas com o atraso, e sua variação, a que a informação está sujeita nessa ligação. Também permite o teste de conectividade antes do estabelecimento de ligações. Na Fig. 5.7 são apresentados alguns cenários possíveis identificados pelo CCITT. Uma descrição mais profunda sobre este tema, englobando outros aspectos que não foram apresentados, pode ser obtida em [Ander91], [Breue91], [Farko93], [Pryck91].

5.5 Tradução da QoS em parâmetros de desempenho da RDIS-BL/ATM

Nesta secção é estudada a tradução dos parâmetros QoS que descrevem a qualidade de serviço desejada para parâmetros de avaliação do desempenho relativos à camada ATM. É introduzido um conjunto de parâmetros que caracterizam a QoS e cujas propriedades indicam a sua natureza e requisitos sob a forma de uma estrutura hierárquica em camadas [Jung93], baseada no modelo OSI. Os parâmetros são definidos para cada uma das camadas de tal forma que cada uma forneça a QoS solicitada pela camada hierarquicamente superior. Assim, este estudo (tradução dos parâmetros QoS) baseia-se na tradução dos parâmetros QoS da camada ALL (que são dependentes do serviço) para os parâmetros QoS da camada ATM. Estes últimos são por sua vez incluídos nos parâmetros NP da ligação e medidos extremo-a-extremo.

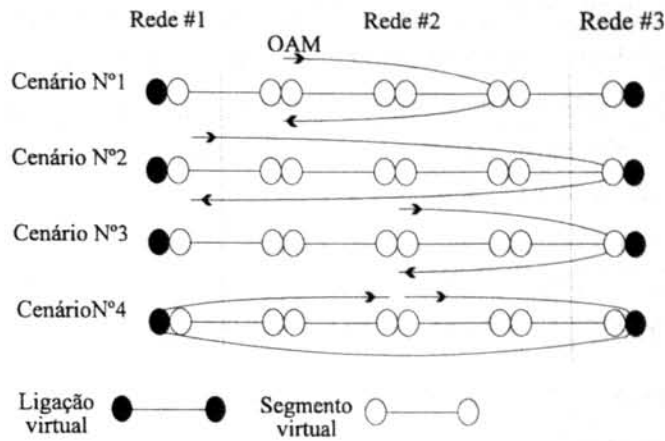


Figura 5.7 Cenários possíveis para a utilização da capacidade de loopback

A dificuldade que se apresenta neste tipo de estudo é a caracterização do desempenho dos futuros serviços que vão ser suportados pela RDIS-BL. Mesmo aqueles que neste momento se conhecem com alguma profundidade (os serviços disponíveis na RDIS) ainda não foram totalmente caracterizados. Também a gama de serviços que se espera sejam transportados pela RDIS-BL/ATM levanta dificuldades adicionais ao estabelecimento dos parâmetros de avaliação de desempenho necessários para os caracterizar, mas pode dizer-se que o desempenho de certos serviços será determinado pela sua sensibilidade ao atraso sofrido no transporte da informação, enquanto outros serão sensíveis às perdas de informação ou mesmo à variabilidade do atraso introduzido pela rede de comunicação. É necessário identificar um conjunto de parâmetros que avaliem o desempenho dos serviços requisitados pelo utilizador, assim como é necessário proceder à tradução destes para parâmetros NP por forma a possibilitar a medição de grandezas objectivas na rede de comunicação. Na Fig. 5.8 é apresentado o modelo utilizado para desenvolver este trabalho, tal como é descrito por [Jung93].

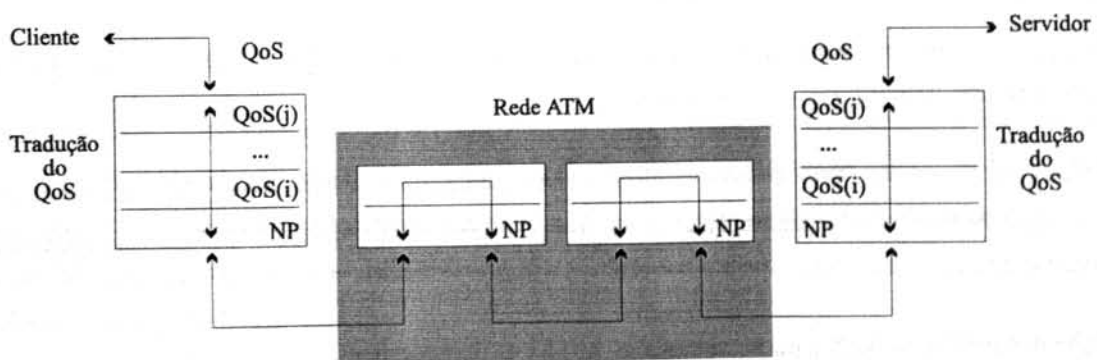


Figura 5.8 Modelo hierárquico para tradução dos parâmetros QoS num ambiente RDIS-BL/ATM.

Os requisitos do utilizador são especificados através de limites admissíveis para os parâmetros QoS, limites esses que podem ser determinísticos ou estatísticos. No caso do limite ser determinístico este é definido da seguinte forma: {parâmetro QoS \leq limite sup.} ou {parâmetro QoS \geq limite inf.}. Para o caso do limite ser estatístico, é admitida uma determinada tolerância, sendo o limite definido por {P[parâmetro QoS \leq limite sup.] \geq limite prob.}. A seguir são apresentados alguns exemplos relativos a serviços das classes A e B, sendo os parâmetros descritos os que serão aplicados no domínio da camada ALL.

- $FER_{m\acute{a}x}$ (*Frame Error Ratio Bound*) - fornece indicação relativa ao número de tramas (SDUs) com erros recebidas das camadas superiores; em qualquer instante $FER(t) \leq FER_{m\acute{a}x}$.
- $W_{m\acute{i}n}$ (*Throughput Bound*) - especifica a capacidade de transmissão de SDUs pela camada AAL; esta é limitada por um valor imposto pela camada hierarquicamente superior.
- $D_{m\acute{a}x}$ (*Delay Bound*) - o atraso experimentado no transporte de uma trama nunca poderá ultrapassar este limite.
- $J_{m\acute{a}x}$ (*Delay Jitter Bound*) - restringe a variabilidade do atraso admitida por qualquer trama (J_i), com base no atraso ideal (D) que a trama experimenta no seu transporte pela rede ($J_i = |D_i - D| \leq J_{m\acute{a}x}$).

Relativamente à camada ATM, alguns dos parâmetros QoS definidos a seguir têm uma correspondência directa com os parâmetros NP definidos pelo CCITT (I.35B). Não se verifica esta relação para o caso dos parâmetros BER (Bit Error Ratio) e CER (Cell Error Ratio), por estes não serem específicos do ATM.

- $CLR_{m\acute{a}x}$ (*Cell Loss Ratio Bound*) - indica o número máximo admissível de células perdidas, como resultado de erros no cabeçalho (5.2.2) e de situações de congestão nos elementos da rede.
- $W'_{m\acute{i}n}$ - é necessário distinguir este parâmetro do definido anteriormente para a camada ALL: neste caso o limite refere-se à capacidade de transferência de células numa ligação ATM.
- $D'_{m\acute{a}x}$ e $J'_{m\acute{a}x}$ - têm a mesma definição dos parâmetros $D_{m\acute{a}x}$ e $J_{m\acute{a}x}$ definidos na camada ALL.
- $CIR_{m\acute{a}x}$ (*Cell Insertion Rate Bound*) - número máximo de células inseridas devido a erros no cabeçalho da célula ATM não detectados ou não corrigidos (5.2.2).
- $CDR_{m\acute{a}x}$ (*Cell Discard Ratio Bound*) - indica o número máximo de células removidas num bloco de informação de dimensão N ; esta restrição é imposta pela camada ALL devido à limitada capacidade de correcção de erros.

A selecção dos parâmetros QoS para as camadas ATM está dependente do tipo de serviço requisitado, o que significa que é consequência do tipo de camada ALL em presença. Usando a título de exemplo um serviço de vídeo (ALL1), e não considerando a sub-camada de convergência (CS), a tradução dos limites é realizada entre a sub-camada SAR (*Segmentation and Reassembly*) e a camada ATM. Os limites dos parâmetros NP gerados a

partir da tradução dos limites dos parâmetros QoS da camada ALL ($FER_{máx}$, $W_{mín}$, $D_{máx}$ e $J_{máx}$) foram os seguintes: $CLR_{máx}$, $W'_{mín}$, $D'_{máx}(NP)$ e $J'_{máx}(NP)$.

Resta obter uma relação entre os limites dos parâmetros da camada ALL e os obtidos para a camada ATM. O cálculo destes limites encontra-se descrito em bibliografia diversa [Anagn92], [Boyer92], [Jung93], apresentando-se neste texto somente o cálculo de $D'_{máx}$ e $J'_{máx}$, para exemplificar a relação existente entre estes e os que os originaram.

Os limites $D'_{máx}$ e $J'_{máx}$ são garantidos pela camada ATM mas dependem da capacidade de armazenamento da entidade, na camada ALL, que recebe a informação. Desta forma os limites negociados pela camada ALL podem ser posteriormente corrigidos pela camada ATM, correcção esta que se reflecte no reforço do limite $D'_{máx}$ e relaxando a restrição em $J'_{máx}$. O limite $D'_{máx}$ é afectado pelo processos de segmentação da subcamada SAR (Eq. 5.1). Nesta expressão l representa o intervalo máximo entre SAR-PDUs (segmentação de uma trama), sendo M é o número de SAR-PDUs gerado a partir de uma trama e $d_{máx}(\alpha)$ é o atraso máximo introduzido pelo armazenamento da informação na fila da entidade receptora da informação.

$$D'_{máx} = D_{máx} - l(M-1) - d_{máx}(\alpha). \quad (\text{Eq. 5.1})$$

O limite $D'_{máx}(NP)$ é o atraso máximo para o parâmetro NP da camada ATM sendo fornecido pela Eq. 5.1, ao qual é subtraído o atraso de propagação. Relativamente aos limites $J'_{máx}$ e $J'_{máx}(NP)$, estes são obtidos pelas Eq. 5.2 e Eq. 5.3, em que $J_{mín}(\alpha)$ indica a compensação mínima efectuada na variabilidade do atraso pelo armazenamento da informação na fila da entidade receptora.

$$J'_{máx} = J_{máx} + J_{mín}(\alpha) \quad (\text{Eq. 5.2})$$

$$J'_{máx}(NP) = J'_{máx} - J_{máx}(UNI) \quad (\text{Eq. 5.3})$$

5.6 Caracterização de elementos da RDIS-BL/ATM

Com a introdução da RDIS novas áreas de interesse surgiram, nomeadamente o estudo da atribuição de recursos ao tráfego de voz e ao tráfego de dados. A formulação deste problema resulta da procura de mecanismos que permitam acomodar simultaneamente ligações em modo circuito e em modo pacote.

Inicialmente foi utilizada uma trama TDM (*Time-Division Multiplexing*) para suportar ambos tipos de tráfego sendo a questão colocada na seguinte forma: como distribuir a capacidade da trama entre tipos de tráfego com características heterogéneas? A solução proposta foi baseada num esquema que permitia a divisão desta em duas secções distintas, uma reservada ao tráfego de voz e outra para ser utilizada pelo tráfego de dados (*Movable Boundary Scheme*) [Willi84].

No esquema adoptado a fronteira entre os dois tipos de tráfego é móvel no sentido em que a capacidade que não seja ocupada pelo tráfego de voz poderá ser aproveitada para transmitir tráfego de dados. O mecanismo proposto pode ser representado através de um processo de Markov (egórdico) bi-dimensional $(X(t), Y(t))$, $0 \leq X(t) \leq N$ e 0

$\leq Y(t) \leq M$, em que $X(t)$ descreve o processo de chamadas de voz e $Y(t)$ descreve o processo de tráfego de dados. A limitação deste esquema é a impossibilidade de o tráfego de voz utilizar a capacidade de transmissão reservada ao tráfego de dados quando esta não é totalmente ocupada.

Num ambiente ATM a capacidade de transmissão é partilhada (estatisticamente) entre todos os utilizadores por forma a maximizar o desempenho e a flexibilizar a gestão dos recursos disponíveis. A determinação do desempenho dos elementos de comutação e de multiplexagem é fundamental para conhecer as características das ligações estabelecidas. A presente secção destina-se a estudar estes elementos utilizando modelos de *tempo discreto*, estes são caracterizados pela divisão da escala dos tempos em parcelas, denominadas por *slots*, e pela existência de um serviço síncrono, ou seja, o serviço é comandado pelas fronteiras do *slot*. As mensagens que chegam num $slot_i$ só podem ser servidas a partir do $slot_{i+1}$, esta restrição distingue este modelo dos ditos contínuos (capítulo 4).

Na análise que se segue são adoptadas as seguintes convenções: um *slot* corresponde a uma unidade de tempo e são numerados sequencialmente através de inteiros positivos; por j^- e j^+ representa-se o instante de tempo anterior, e posterior, relativos ao início do $slot_j$; uma mensagem que complete o serviço abandona o elemento entre os instantes de tempo j^- e j , e uma mensagem que inicie o serviço realiza-o entre os instantes de tempo j e j^+ .

Decorre da utilização deste tipo de modelo que alguns dos parâmetros que caracterizam os modelos contínuos não são adequados para os modelos discretos, nomeadamente o processo de chegada de mensagens ao sistema. Neste caso o processo é descrito pelo acréscimo do número de mensagens no intervalo de tempo de um *slot* e não pela distribuição do tempo entre mensagens consecutivas. Consequentemente a notação utilizada para a descrição deste modelos distingue-se da utilizada nos modelos contínuos na sua apresentação: A-B-m-K (é preservado o significado dos caracteres).

5.6.1 Avaliação do desempenho de um comutador ATM

A funcionalidade oferecida por um comutador ATM baseia-se no encaminhamento de células que se apresentam numa das suas entradas (N) para uma ou mais saídas (N), utilizando para tal o conhecimento dos identificadores VPI/VCI. Num comutador com dimensão $N \times N$ considera-se que existem N/c destinos diferentes (Fig. 5.9) e que as saídas do comutador estão proporcionalmente distribuídas pelos destinos existentes, logo existem c saídas por cada destino (este comutador é designado por $N \times N(c)$).

Cada entrada pode admitir uma célula por cada unidade de tempo, assim como cada saída pode transmitir uma única célula por *slot*. No caso de diversas células serem encaminhadas para o mesmo destino e o grupo de saídas associadas a este destino ter capacidade inferior à requisitada, algumas dessas células têm que ser armazenadas temporariamente (Fig. 5.9) em filas que por hipótese se localizam nas saídas.

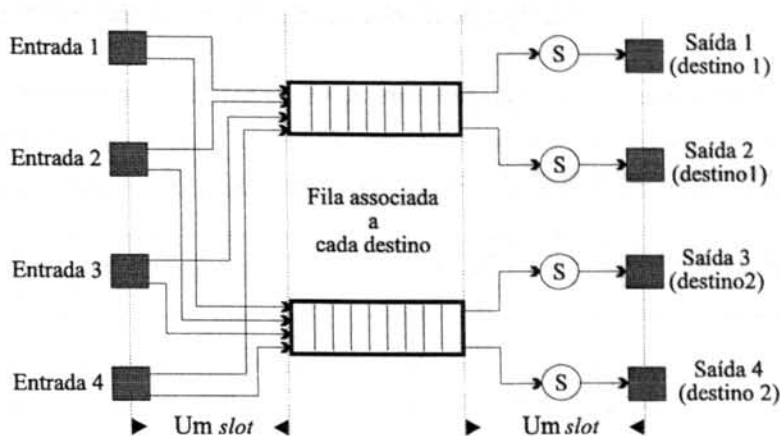


Figura 5.9 Representação funcional de um comutador ATM (4x4(2)).

A fila de espera que está associada a cada destino pode ser representada (em termos da teoria das filas de espera) por um sistema de tempo discreto (*slot*) com c servidores, cujo processo de admissão de clientes é dependente das características das fontes de tráfego, sendo o acréscimo máximo admitido igual a N .

O tráfego é gerado (em cada entrada) segundo um processo de *Bernoulli* com probabilidade p . Se o fluxo de células em cada uma das entradas for independente das restantes então o número total (por *slot*) de células admitidas em cada uma das filas de espera é fornecido por uma distribuição *binomial*. As células são encaminhadas até à fila de espera relativa ao destino pretendido utilizando recursos exclusivos associados à entrada que as admitiu. Resulta que a probabilidade de uma célula (proveniente de qualquer entrada) ser admitida por uma das filas de espera é igual a $p \cdot c / N$ (considera-se que as células são encaminhadas independentemente e uniformemente através do comutador). Conclui-se que os parâmetros que caracterizam a distribuição binomial são: N e $p \cdot c / N$ (caso $N \rightarrow \infty$ a distribuição binomial transforma-se numa distribuição de *Poisson*).

Considerando que o serviço efectuado é determinístico, o sistema descrito tem a seguinte classificação: *Binom-D-c* (se $N \rightarrow \infty$ então *Pois-D-c*) [Brune93]. Pode-se determinar seguidamente a função geradora de probabilidade (*fg*) da ocupação da fila de espera, considerando para esta análise que o fluxo de células que são admitidas pela fila de espera (entre *slots* consecutivos) forma um processo independente, cuja distribuição é genérica (englobando os dois casos anteriores), e sendo a sua *fg* denominada por $A(z)$.

5.6.1.1 Comprimento da fila de espera

Representando por S_k o número total de células pertencentes à fila de espera associada a um determinado destino como às células que se encontram nos seus servidores (pertencentes a essa fila de espera) no princípio do *slot* _{k} . Por $S(z)$ representa-se a *fg* de S_k em equilíbrio, e por $Q(z)$ a *fg* do conteúdo da fila de espera (sem contemplar os servidores) em equilíbrio. Então as equações que descrevam a dinâmica do sistema são as seguintes:

$$S_{k+1} = (S_k - c)^+ + A_k \text{ se } S_k \geq c \quad (\text{Eq. 5.4})$$

$$S_{k+1} = A_k \text{ se } S_k < c. \quad (\text{Eq. 5.5})$$

Recorrendo à transformada Z consegue-se obter a expressão para $S_{k+1}(z)$,

$$S_{k+1}(z) = E[z^{S_{k+1}}] = P[S_k \geq c]E[z^{S_k - c + A_k} | S_k \geq c] + P[S_k < c]E[z^{A_k} | S_k < c]$$

$$S_{k+1}(z) = \sum_{n=0}^{\infty} \sum_{j=c}^{\infty} P[S_k = j]P[A_k = n]z^{j-c+n} + \sum_{n=0}^{\infty} \sum_{j=0}^{c-1} P[S_k = j]P[A_k = n]z^n$$

$$S_{k+1}(z) = \frac{A_k(z)}{z^c} \left[S_k(z) - \sum_{j=0}^{c-1} P[S_k = j]z^j \right] + A_k(z) \sum_{j=0}^{c-1} P[S_k = j]. \quad (\text{Eq.5.6})$$

Impondo condições estacionárias ($k \rightarrow \infty$) na Eq. 5.6 resulta a fg para $S(z)$,

$$S(z) = \frac{A(z) \sum_{j=0}^{c-1} (z^c - z^j) P[S = j]}{z^c - A(z)}. \quad (\text{Eq. 5.7})$$

Na equação anterior existem c termos desconhecidos ($P[S = j]$); para os determinar recorre-se ao teorema de Rouché [Churc90]. Na Eq. 5.8 as quantidades z_j são zeros complexos da expressão ($z^c - A(z)$), localizados no interior da circunferência de raio unitário do plano complexo Z . Sendo $A'(1)$ o valor médio de $A(z)$, é necessário, para se atingir o equilíbrio, isto é para que não se verifique o crescimento da fila, que este valor seja inferior a c .

$$S(z) = (c - A'(1)) \frac{(z-1)A(z)}{z^c - A(z)} \prod_{j=1}^{c-1} \frac{z - z_j}{1 - z_j} \quad (\text{Eq. 5.8})$$

Para determinar a fg do comprimento da fila de espera basta observar que esta grandeza é obtida a partir de S_k através da seguinte relação: $Q_k = (S_k - c)^+$. Desta forma é possível, a partir de $S(z)$, obter imediatamente $Q(z)$ ($S(z) = A(z) * Q(z)$), assim como o comprimento médio da fila ($Q'(1)$).

$$Q(z) = (c - A'(1)) \frac{(z-1)}{z^c - A(z)} \prod_{j=1}^{c-1} \frac{z - z_j}{1 - z_j} \quad (\text{Eq. 5.9})$$

$$Q'(1) = \frac{1}{2} \prod_{j=1}^{c-1} \frac{z + z_j}{1 - z_j} + \frac{A''(1) - (c-1)A'(1)}{2(c - A'(1))} \quad (\text{Eq. 5.10})$$

5.6.1.2 Atraso experimentado pelo fluxo de células na fila de espera

O atraso de uma determinada célula pode ser visto como sendo o número de *slots* entre o instante de admissão no sistema e o fim do *slot* no qual é admitida pelo servidor, visto que o tempo associado ao serviço é determinístico e por isso não é necessário incluí-lo na análise efectuada. Representa-se pela variável aleatória D o atraso verificado na fila de espera, correspondendo $D(z)$ à sua transformada Z .

Para se estabelecer uma relação que determine o atraso de uma célula particular (previamente seleccionada) é necessário conhecer a dimensão da fila de espera associada ao destino pretendido, $Q(z)$, assim como o número de células, $F(z)$, que entram, antes da célula marcada, na fila de espera durante o slot de admissão desta e que, irão ser transmitidas antes dela (admitindo uma disciplina FIFO). O conteúdo da fila de espera (R) antes da célula marcada ser admitida é dado por: $F + Q - 1$ (note-se que F inclui a célula em estudo e que no caso da disciplina da fila ser do tipo LIFO o número de células que seriam admitidas ao serviço antes da célula marcada seria fornecido por: $N - (F + 1)$). A *fg* da expressão anterior é dada por

$$R(z) = \frac{F(z)Q(z)}{z}, \tag{Eq. 5.11}$$

sendo $F(z)$ [Brune93]

$$F(z) = z \frac{A(z) - 1}{(z - 1)A'(1)}. \tag{Eq. 5.12}$$

É então possível obter $D(z)$ a partir do conhecimento da distribuição de probabilidade de R ,

$$D(z) = \sum_{i=1}^{\infty} P[D = i]z^i = \sum_{i=1}^{\infty} z^i \sum_{j=0}^{c-1} P[R = (i-1)c - j]. \tag{Eq. 5.13}$$

Na Eq. 5.14 é apresentada a *fg* de D em que $a = e^{j2\pi/c}$ (j , imaginário); a partir desta expressão é possível obter através de derivação os momentos da variável aleatória D . Um parâmetro importante é o valor médio do atraso a que a célula marcada está sujeita na fila de espera; no seu cálculo é necessário atender ao facto do argumento da *fg* de D não ser z mas sim z^c

$$D(z^c) = \frac{c - A'(1)}{cA'(1)} \sum_{s=0}^{c-1} \frac{z^c - 1}{1 - (a^s z)^{-1}} \frac{A(a^s z) - 1}{z^c - A(a^s z)} \prod_{j=1}^{c-1} \frac{a^s z - z_j}{1 - z_j}, \tag{Eq. 5.14}$$

$$\bar{D} = \frac{1}{c} \left[\frac{dD(z^c)}{dz} \right]_{z=1} = 1 + \frac{\bar{Q}}{A}. \tag{Eq. 5.15}$$

Existem serviços para os quais não só o valor médio da atraso é importante mas também, e especialmente, a sua variabilidade (*jitter*). Para se obter a variância (segundo momento central) do atraso recorre-se às expressões seguintes:

$$V[D] = E\left[(D - \bar{D})^2\right] = D'(1) + D'(1) - (D'(1))^2, \quad (\text{Eq. 5.16})$$

$$D'(1) = \frac{1}{c^2} \left[\frac{d^2 D(z^c)}{dz^2} \right]_{z=1} - \frac{c-1}{c} D'(1). \quad (\text{Eq. 5.17})$$

As expressões, aproximadas, da distribuição de probabilidade do comprimento da fila de espera, $P[Q = n]$, e do atraso experimentado na fila de espera, $P[D = n]$, são obtidas por inversão das *fgs* através do teorema dos resíduos de Cauchy. Estas não são apresentadas devido à complexidade envolvida no seu cálculo estar além do âmbito deste texto (ver [Brune93]).

5.6.1.3 Caso de estudo: comutador ATM 16x16(c)

Utilizando as expressões anteriores é possível estimar a capacidade de armazenamento das filas de espera, caracterizada pelo seu valor médio, assim como o atraso médio, e a variância, que o fluxo de células sofre ao atravessar este elemento. Neste exemplo considera-se um comutador com 16 entradas e 16 saídas ($N = 16$) sendo o número de destinos possíveis variável (depende do número de servidores, saídas, por destino: $c = 1, 2$ e 4). Sendo $A(z)$ (Eq. 5.18) a transformada Z da distribuição do tráfego injectado nas filas de espera associadas a cada destino, que resulta da sobreposição de 16 entradas em que cada uma gera tráfego segundo uma distribuição de *Bernoulli* ($p(0) = p$), então obtém-se uma distribuição binomial cujos parâmetros são: $N = 16$ e $P = p*c/N$.

$$A(z) = \left(1 - \frac{p*c}{16} + \frac{p*c*z}{16} \right)^{16} \quad (\text{Eq. 5.18})$$

No gráfico da Fig. 5.10, que representa a ocupação média total (em todas as N/c filas de espera) do comutador, é visível que esta depende inversamente do número de servidores por destino. A evolução das curvas é determinada por dois factores distintos: a existência de maior capacidade de processamento (aumento do número de servidores por fila de espera) e a diminuição do número de destinos (consequentemente do número de filas de espera) implicam o aumento da probabilidade de congestão devido ao aumento da variabilidade do tráfego injectado nas filas de espera (Fig. 5.11). Dos resultados obtidos conclui-se que o aumento da variância do tráfego injectado é suplantado pelo aumento do número de servidores.

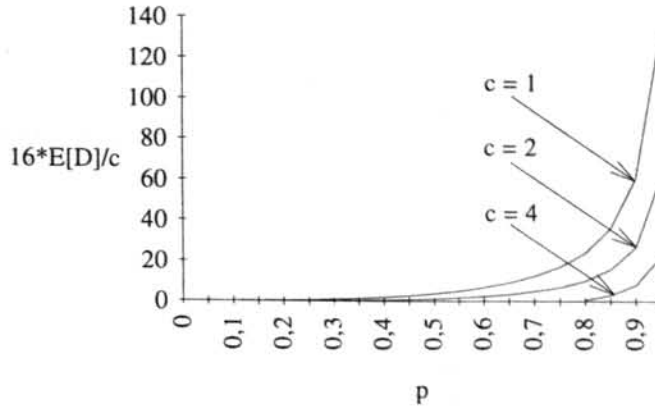


Figura 5.10 Ocupação média total do comutador ATM (16x16(c)).

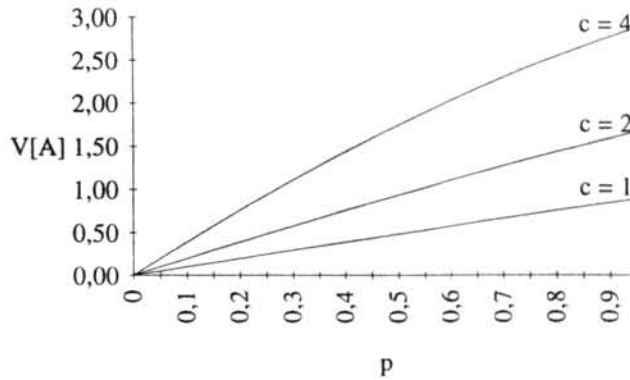


Figura 5.11 Variância do tráfego injectado nas filas de espera do comutador ATM (16x16(c)) (V[A]).

Sendo a ocupação média das filas de espera uma variável discreta, medida pelo número de células armazenadas à espera de serviço (grandeza inteira), e não uma variável contínua como é apresentado no gráfico da Fig. 10, é necessário determinar o inteiro superior ao valor calculado para cada ponto das referidas curvas. Os resultados obtidos estão apresentados no gráfico da Fig. 12.

Para analisar o atraso experimentado no comutador considera-se que todas as filas de espera são estatisticamente equivalentes (as células são encaminhadas deterministicamente e a probabilidade de chegarem células para diferentes destinos é idêntica), o que permite estudar uma das filas de espera independentemente das restantes. Nesse caso, o desempenho deste elemento é determinado quer pela sua velocidade de funcionamento, quer pelo *jitter* introduzido, e estas medidas podem ser obtidas pelo estudo da média e variância do atraso (Fig. 5.13a e 5.14).

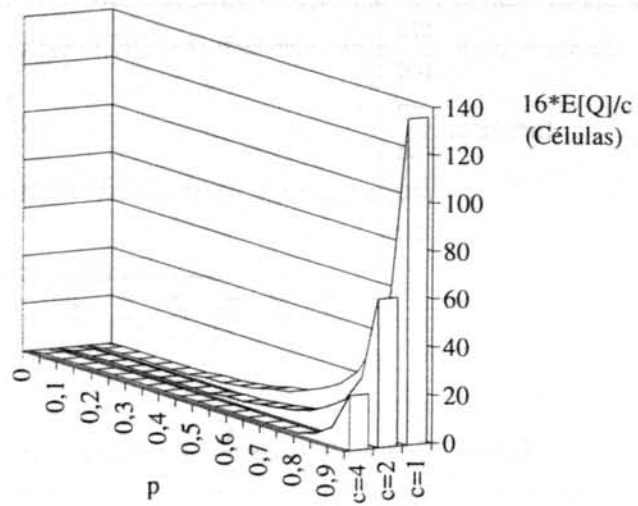


Figura 5.12 Ocupação média total (medida em células) do comutador ATM (16x16(c)).

Constata-se que o valor médio do atraso introduzido, para a mesma carga (p) imposta ao comutador, se reduz substancialmente para valores crescentes do número de servidores por fila de espera. De igual forma a variância do atraso é menor quando se aumenta o número de servidores, resultado que é consequência da sobreposição de diversas fontes de tráfego independentes (regulação das características estatísticas). Na Fig. 5.13b encontra-se representada o gráfico do atraso médio medido em slots (variável discreta).

No apêndice C são indicados, a título exemplificativo, os cálculos realizados e os resultados numéricos obtidos que serviram de base para os gráficos apresentados nesta secção. Na execução destes cálculos foi utilizado um pacote de *software* público denominado PARI-GP, instalado numa SPARCstation 10.

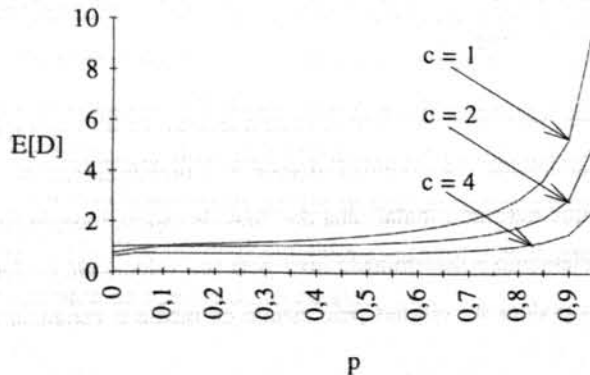


Figura 5.13a Atraso médio numa (qualquer) fila de espera do comutador ATM (16x16(c)).

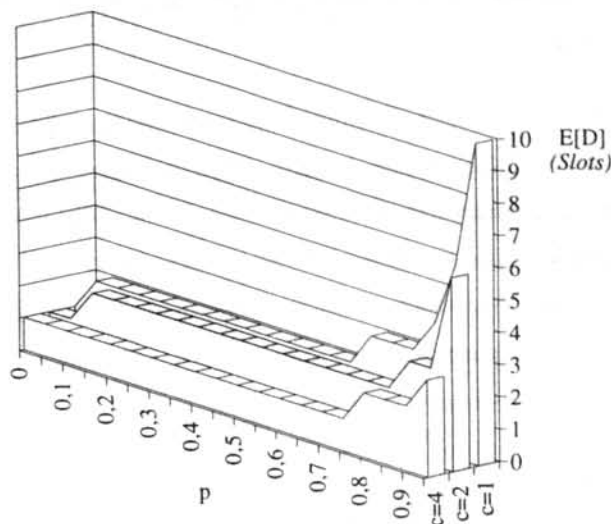


Figura 5.13b Valor médio do atraso (medido em slots) num comutador ATM (16x16(c)).

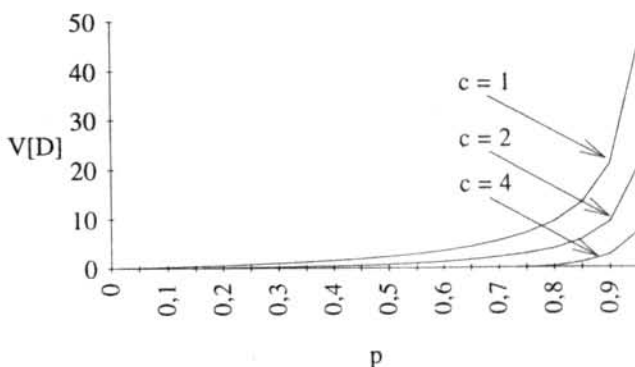


Figura 5.14 Variância do atraso experimentado pelo fluxo de células injectado no comutador ATM (16x16(c)).

5.6.1.4 Referências a outros métodos

Os resultados apresentados podem ser comparados com outros obtidos por diversos autores, que recorreram a diferentes métodos para prever o comportamento das filas de espera, usando outros tipos de serviços fornecido e de processos de chegada de clientes.

Saito [Saito90] refere a utilização de uma nova disciplina para governar o funcionamento das filas de espera num nó de comutação, denominada por EDD (*Earliest-Due-Date*). Esta é apresentada como um critério óptimo para determinar qual o próximo cliente a ser seleccionado para serviço, tendo por objectivo minimizar o número de células cujo atraso é superior ao máximo admitido, ou seja, minimiza o número de células removidas, e baseia-se na atribuição de uma constante T_j a cada classe de tráfego, que reflecte o atraso máximo admitido.

Desta forma, a escolha do próximo cliente é baseada na EDD mais restritiva, mantendo-se a sequência das células para cada classe de tráfego.

Outros autores [San-Qi90] modelizaram a correlação existente entre mensagens consecutivas, resultante do tráfego gerado por fontes de voz e vídeo, através de cadeias de *Markov* discretas bi-dimensionais. Cada fonte de tráfego é representada por dois estados possíveis (ON/OFF), tendo o intervalo de tempo que esta permanece no estado inactivo uma distribuição geométrica. Relativamente ao estado activo, o tráfego gerado caracteriza-se por apresentar intervalos de tempo constantes entre células consecutivas. No caso de se considerar que as cadeias de *Markov* discretas bi-dimensionais são independentes e identicamente distribuídas então esta representação pode ser aproximada por um processo nascimento-morte.

Para representar o tráfego resultante da sobreposição de fluxos de mensagens com características heterogéneas podem utilizar-se processos estocásticos do tipo MMPP (*Markov Modulated Poisson Process*) [Heffe86], processos estes que são duplamente estocásticos, ou seja, resultam da modulação do valor médio de um processo de *Poisson* por uma cadeia de *Markov* (contínua). Recentemente Baiocchi [Baioc93] apresentou a análise exacta de uma fila de espera do tipo MMPP/G/1/K utilizando a abordagem clássica das filas de espera do tipo M/G/1/K, IMC (*Imbedded Markov Chain*) [Klein75]. A análise deste sistema é baseada na determinação das probabilidades estacionárias de uma variável bi-dimensional, $S(t) = [X(t), J(t)]$, em que $X(t)$ representa o número de utilizadores no sistema (toma valores entre 0 e K) e $J(t)$ representa o estado da cadeia de *Markov* que modula o processo de chegada de clientes à fila de espera (do tipo de *Poisson*). As épocas de renovação estão associadas aos instantes de serviço.

5.6.2 Análise de um multiplexador ATM

A análise de desempenho de um multiplexador ATM depende do tipo de comportamento das fontes ATM que lhe estão ligadas. Este comportamento caracteriza-se [Coste91] em três escalas de tempo distintas: chamada, *burst* e célula. O tráfego gerado pelas fontes têm modelos distintos consoante a escala de tempos que se considera. Naturalmente que sendo a capacidade de armazenamento partilhada (estatisticamente) por todas as fontes, existe a probabilidade do multiplexador bloquear devido a não existirem os recursos necessários para armazenar o fluxo de clientes (células) que pretendem entrar no sistema (5.3).

Reconhecendo que diferentes escalas de tempo têm consequências na análise do comportamento do multiplexador é comum adoptar diferentes modelos para os diversos casos. De seguida apresenta-se um estudo relativo ao uso de processos MMPP para caracterizar o comportamento de fontes de tráfego ATM, que podem posteriormente ser utilizados para determinar o comportamento de um multiplexador ATM. O estudo deste elemento resume-se à análise de um sistema de fila de espera do tipo MMPP/D/1/K, ou, no caso da análise ser discreta, DBMAP-D-1-K (*Discrete-time Batch Markov Arrival Process*).

O processo DBMAP pode ser utilizado para descrever o comportamento do fluxo de informação resultante da sobreposição de várias fontes ATM (domínio discreto). À transição, no $slot_k$, do estado i (do processo de entrada

no sistema) para o estado j , no $slot_{k+1}$, corresponde a probabilidade de serem geradas n células, $(a_n)_{i,j}$, sendo $n \geq 0$. Estas probabilidades podem ser representadas sob a forma matricial, por

$$\mathbf{A}_n = \left[(a_n)_{i,j} \right] \text{ e } (0 \leq i, j \leq S, n \geq 0). \quad (\text{Eq. 5.19})$$

Para estabelecer a correspondência entre a expressão anterior e o significado associado a este tipo de processos, refira-se que o índice i indica o número de fontes que estão activas no $slot_k$, assumindo-se que as células geradas, i , são uniformemente distribuídas pela trama, cuja dimensão é M . A probabilidade $(a_n)_{i,j}$ é dada por uma distribuição binomial,

$$(a_n)_{i,j} = \binom{i}{j} \left(\frac{1}{M} \right)^j \left(1 - \frac{1}{M} \right)^{i-j}. \quad (\text{Eq. 5.20})$$

A ocupação da fila de espera é avaliada nos instantes de fim de serviço e pode ser descrita pela probabilidade $(c_n)_{i,j}$. Esta representa a probabilidade da fila de espera ficar vazia no estado i , correspondente ao instante de início da inactividade, e de no estado j , correspondente ao instante de início de actividade, ter uma ocupação igual a n . Analogamente a $(a_n)_{i,j}$ $(c_n)_{i,j}$ representa-se sob a forma matricial \mathbf{C}_n .

Para obter a probabilidade $(c_n)_{i,j}$ basta observar que entre o estado i e o estado j têm que entrar n células no sistema. Assim se chegarem $j + 1$ ($0 \leq j \leq n$) células no último slot para o qual o sistema ainda está vazio, e entrarem na fila de espera $n - j$ células no slot em que a primeira célula é servida (após um período de inactividade), então

$$\mathbf{C}_n = \sum_{i=0}^{\infty} (\mathbf{A}_0)^i \sum_{j=0}^{\infty} \mathbf{A}_{j+1} \mathbf{A}_{n-j} = (\mathbf{I} - \mathbf{A}_0)^{-1} \sum_{j=0}^n \mathbf{A}_{j+1} \mathbf{A}_{n-j},$$

$$\mathbf{C}_n = z^{-1} (\mathbf{I} - \mathbf{A}_0)^{-1} (\mathbf{A}(z) - \mathbf{A}_0) \mathbf{A}(z). \quad (\text{Eq. 5.21})$$

Denominando por (L_k, J_k) a ocupação da fila de espera e o estado do processo de entrada de novos clientes após o $slot_k$, é possível determinar o vector de probabilidades estacionárias

$$\bar{\mathbf{L}} = [\bar{L}_0, \bar{L}_1, \dots], \quad (\text{Eq. 5.22})$$

em que se associa o índice ao número de células armazenadas na fila de espera. Assim, no caso deste índice ser igual a m , o significado fornecido por \bar{L}_m é a probabilidade estacionária de existirem m células em fila de espera. As componentes do vector $\bar{\mathbf{L}}_m$ representam a probabilidade para um determinado estado do processo de entrada.

A partir da Eq. 5.21 é possível determinar o vector de probabilidades estacionárias utilizando procedimentos algorítmicos e operações analíticas desenvolvidas por [Neuts81], cuja solução final é:

$$\bar{L}_m = \left(\bar{L}_0 C_m + \sum_{i=1}^{m-1} \bar{L}_i \bar{A}_{m+1-i} \right) (\mathbf{I} - \bar{A}_1)^{-1}, \quad (\text{Eq. 5.23})$$

$$\bar{A}_w = \sum_{i=w}^{\infty} \mathbf{A}_i \mathbf{G}^{i-w} \quad \text{e } w \geq 0. \quad (\text{Eq. 5.24})$$

em que a matriz \mathbf{G}_n é definida por

$$\mathbf{G}_n = \left[(g_n)_{i,j} \right] \text{ e } (0 \leq i, j \leq S, n \geq 0), \quad (\text{Eq. 5.25})$$

e fornece a probabilidade de serem transmitidas n células no período de tempo compreendido entre o estado i e o estado j do processo de entrada.

De entre a vasta bibliografia existente nesta área são destacados os seguintes artigos publicados sobre o estudo de fontes de tráfego ATM. Estes estudos são importantes pois aplicam-se na avaliação do desempenho de multiplexadores ATM, como também no estudo da admissão de tráfego gerado por tais fontes a uma rede ATM (policiamento) [Harta93], [Lee93], [Norro91], [Rathg91], [Reiss93], [Saito91], [Sauln92], [Sohra92, 93], [Stavr92], [Yamad91], [Zhang92], [Zhang93].

Capítulo 6

Conclusões

A evolução dos sistemas de comunicação requiere a realização de estudos prévios baseados em simulações e modelos analíticos, que permitam avaliar os conceitos e técnicas usadas com o intuito de fornecer pistas para otimizar o serviço fornecido. Devido às limitações das técnicas de modelização quando aplicadas aos sistemas reais, é necessário construir protótipos dotados da funcionalidade requerida para suportar os serviços que se prevêem existir no futuro, e realizar medidas que caracterizem o desempenho obtido quando este se encontra sob as mais variadas condições de funcionamento. É neste contexto que surgiu a ideia de desenvolver um sistema de teste adequado às exigências das futuras redes de comunicação.

Apresentou-se uma arquitectura para ferramentas de teste de redes de comunicação com elevado débito, tendo sido esta concretizada no desenvolvimento de um sistema denominado SGDT, que foi utilizado para testar a funcionalidade e avaliar o desempenho do protótipo desenvolvido pelo projecto DAMS. Os testes elaborados compreendiam a geração e detecção de tráfego de pacotes, e mensagens de sinalização para o estabelecimento de chamadas RDIS, sendo esperado obter valores para os parâmetros que caracterizam o desempenho das ligações. A construção do SGDT foi orientada por um requisito fundamental - a necessidade de gerar e detectar tráfego a elevadas cadências, o que levou ao desenvolvimento de uma arquitectura modular para permitir facilmente a sua expansão.

Durante o projecto DAMS foram estabelecidos alguns contactos em conferências internacionais, com responsáveis de empresas activas neste domínio. Reconhecendo méritos ao SGDT foi identificada uma potencial desvantagem relacionada com a utilização de uma interface para o exterior que não está normalizada, facto que elimina a possibilidade de tornar o SGDT num produto comercial a curto prazo. Esta interface determina o local onde se realizam as medidas, tendo-se revelado uma opção correcta, pois permite testar os subsistemas que constituem o protótipo assim como o meio de transmissão utilizado.

A grande modularidade conferida ao SGDT permitiu a aplicação dos desenvolvimentos realizados a outros ambientes de que, o projecto ESPRIT III denominado MANTIS (*Metropolitan Area Network for Integrated Services*) é um exemplo, este traduz a aposta dos projectos ESPRIT neste sistema de validação. O projecto MANTIS tem por objectivo o desenvolvimento de um sistema de telecomunicações de banda larga capaz de oferecer serviços SMDS (*Switched Multi-megabit Data Service*) baseados em interfaces DQDB (*Distributed Queue Dual Bus*). Os nós construídos serão interligados por uma rede baseada em técnicas ATM.

As alterações preconizadas para adaptar o SGDT a este projecto limitam-se ao desenvolvimento de um novo circuito impresso para o bloco de interface ao meio (2.048 Mbps), assim como de parte do *software* destinado ao controlo deste. Uma vantagem óbvia para o grupo que desenvolve o novo SGDT, é o facto deste ser passível de ser comercializado, dado que oferece uma interface normalizada.

O trabalho que se desenrola no projecto MANTIS permitirá obter os conhecimentos necessários para fazer evoluir o SGDT para ambientes ATM. Simultaneamente encontram-se a ser realizados estudos teóricos sobre modelos analíticos utilizados na avaliação do desempenho de sistemas de comunicação baseados em ATM. Foi evidente a preocupação em analisar os módulos cujas funções caracterizam a técnica de transmissão utilizada, e por consequência que se consideram fundamentais no desempenho previsto para segmentos ATM - o comutador e o multiplexador. Outros modelos analíticos são possíveis, além dos apresentados, sugerindo-se a realização de estudos comparativos do desempenho teórico previsto para um comutador ATM, fazendo uso de outras disciplinas para a gestão das filas de espera, como na utilização de filas de espera à entrada do comutador em vez destas se encontrarem na saída deste. Pretende-se continuar o estudo dos modelos analíticos para multiplexadores ATM.

A análise do desempenho é importante para definir quais os parâmetros de qualidade e seus valores; estes são negociados durante a fase de estabelecimento da ligação e monitorados (policiamento) pelas entidades competentes durante a ligação, para detectar possíveis violações do contrato estabelecido. Diversos algoritmos têm sido estudados para implementar um mecanismo de monitoração de admissão de tráfego. Assume-se de grande importância a definição de um conjunto de medidas, para cada serviço que irá ser suportado pela RDIS-BL, que traduzam adequadamente os requisitos dos utilizadores. Resultando na necessidade de traduzir a QoS estabelecida pelos diversos serviços na camada de adaptação ATM, para medidas de qualidade relativas aos procedimentos efectuados pela camada ATM. Esta tradução baseia-se num modelo de camadas sendo cada medida pertencente a uma determinada camada traduzida em medidas relativas às camadas inferiores. De igual forma são obtidos os limites determinísticos ou estatísticos que caracterizam cada medida, utilizando para tal o mesmo processo de tradução.

O conhecimento adquirido pela participação em projectos ESPRIT no domínio dos sistemas de teste, permite afirmar que estão criadas as condições necessárias à concepção de ferramentas para a avaliação do desempenho de redes baseadas em técnicas ATM. Neste âmbito encontra-se em fase de negociação com as operadoras a participação num projecto piloto ATM destinado a estabelecer um conjunto de medidas que caracterizem o desempenho de ligações ATM.

Referências

- [Anagn92] M. E. Anagnostou, M. E. Theologou e E. N. Protonotarios, "Cell insertion ratio analysis in ATM networks," *Computer Networks and ISDN Systems*, vol. 24, pp. 335-344, Maio 1992.
- [Ander91] J. Anderson e M. D. Nguyen, "ATM-layer OAM implementation issues," *IEEE Communications Magazine*, pp. 79-81, Setembro 1991.
- [Assis91] P. Assis, "Notas relativas à utilização da placa de interface TGD/LSI," INESC Doc. interno, Novembro 1991.
- [Assis93] P. Assis, A. Almeida, M. Henriques, M. Ricardo, P. Ferreira, P. Proença e E. Carrapatoso, "TGDS - a new tool for performance evaluation of HSLAN's." *Proceedings of the SPIE93*, vol. 1975, pp. 246-256.
- [Baioc93] A. Baiocchi e N. Bléfari-Melazzi, "Steady-state analysis of the MMPP/G/1/K queue," *IEEE Trans. Commun.*, vol. 41, pp. 531-534, Abril 1993.
- [Boyer92] P. E. Boyer e D. P. Tranchier, "A reservation principle with applications to the ATM traffic control," *Computer Networks and ISDN Systems*, vol. 24, pp. 321-334, Maio 1992.
- [Breue91] H. Brueuer, "ATM-layer OAM: principles and open issues," *IEEE Communications Magazine*, pp. 75-78, Setembro 1991.
- [Brune93] H. Bruneel e B. G. Kim, *Discrete-time Models For Communication Systems Including ATM*, Kluwer Academic Publishers, 1993.
- [Burgi91] J. Burgi e D. Dorman, "Broadband ISDN resource management: the role of the virtual paths," *IEEE Communications Magazine*, pp. 44-48, Setembro 1991.
- [Capol90] D. Capolupo e F. del Castello, "A flexible integrated multiservice optical coherent network based on TDMA techniques," *Proceedings of the EFOC/LAN90*, pp. 97-102.
- [Caste91] P. Castelli, R. Melen e A. Tonietti, "Performance evaluation and call admission rules in ATM networks," *CSELT Technical reports*, vol. XIX, pp. 17-24, Fevereiro 1991.

- [Churc90] R. V. Churchill e J. W. Brown, *Complex Variables and Applications*, 5ª Ed., McGraw-Hill, 1990.
- [Conne92] D. Conner, "High-Density PLDs," *EDN*, pp. 77-88, 2 de Janeiro 1992.
- [Crawf91] G. Crawford e W. Grupp, "Testing ATM," *Communications International*, pp. 131-140, Outubro 1991.
- [Daigl86] J. N. Daigle e J. D. Langford, "Models for analysis of packet voice communications systems," *IEEE J. Select. Areas Commun.*, vol. SAC-4, pp. 847-855, Setembro 1986.
- [DAMS90a] M. Bleichrodt, W. Schmidt e K. Sauer, "The local switch unit," E.2146:DAMS/TN/DEL 2.1, Março 90.
- [DAMS90b] W. Schmidt, "Definitive SII specification," E.2146:DAMS/TN/DEL 3.2, Junho 90.
- [Eckbe91] A. Eckberg, B. T. Doshi e R. Zoccolillo, "Controlling congestion in B-ISDN/ATM: issues and strategies," *IEEE Communications Magazine*, pp. 64-70, Setembro 1991.
- [Farko93] S. C. Farkouh, "Managing ATM-based broadband networks," *IEEE Communications Magazine*, pp. 82-86, Maio 1993.
- [Gilbe91] H. Gilbert, O. Aboul-Magd e V. Phung, "Developing a cohesive traffic management strategy for ATM networks," *IEEE Communications Magazine*, pp. 36-45, Outubro 1991.
- [Habib91] I. W. Habib e T. N. Saadawi, "Controlling flow and avoiding congestion in broadband networks," *IEEE Communications Magazine*, pp. 46-53, Outubro 1991.
- [Harri93] P. G. Harrison e N. M. Patel, *Performance Modelling of Communication Networks and Computer Architectures*, Addison-Wesley, 1993.
- [Harta93] V. F. Hartanto e H. R. Sirisena, "User-network policer: a new approach for ATM congestion control," *Proceedings of the IEEE INFOCOM 93*, pp. 376-383.
- [Heffe86] H. Heffes e D. M. Lucantoni, "A Markov characterization of packetized voice and data traffic and related statistical multiplexer performance," *IEEE J. Select. Areas Commun.*, vol. SAC-4, pp. 856-868, Setembro 1986.
- [Hills91] A. R. Hills, "FDDI-II - an implementor's perspective," *Proceedings of the EFOC/LAN91*, pp. 194-200.
- [Isaac76] D. L. Isaacson e R. W. Madsen, *Markov Chains Theory and Applications*, John Wiley & Sons, 1976.

- [Jung93] J. Jung e D. Seret, "Translation of QoS parameters into ATM performance parameters in B-ISDN," *Proceedings of the IEEE INFOCOM 93*, pp. 748-755.
- [King90] P. J. B. King, *Computer and Communication Systems Performance Modelling*, Prentice-Hall, 1990.
- [Kirrm85] H. Kirrmann, "Events and interrupts in tightly coupled multiprocessors," *IEEE Micro*, pp. 53-66, Fevereiro 1985.
- [Klein75] L. Kleinrock, *Queueing Systems I: theory*, John Willey & Sons, 1975.
- [Lee93] H. W. Lee e J. W. Mark, "ATM network traffic characterization using two types of ON-OFF sources," *Proceedings of the IEEE Globecom 92*, pp. 152-1459.
- [Leibs90] S. H. Leibson, "PLD development software," *EDN*, pp. 100-116, 2 de Agosto 1990.
- [Lin87] K.-S. Lin, G. A. Frantz e R. Simar, "The TMS320 family of digital signal processors," *Proceedings of the IEEE*, vol. 75, No. 9, pp. 1143-1159, Setembro 1987.
- [Meyer90] E. Mayer, "Tools help eliminate PLD timing errors," *Computer Design*, pp. 55-64, 1 de Abril 1990.
- [Micro90] Microsys, *Technical Description User's Manual CPU 22/32*, Microsys Electronics GmbH, Janeiro 1990.
- [Neuts81] M. F. Neuts, "Matrix -geometric solutions in stochastic models: an algorithmic approach," The Johns Hopkins University Press, 1981.
- [Norro91] I. Norro, J. W. Roberts, A. Simonian e J. T. Virtamo, "The superposition of variable bit rate sources in an ATM multiplexer," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 378-387, Abril 1991.
- [Okada91] T. Okada, H. Ohnishi e N. Morita, "Traffic control in asynchronous transfer mode," *IEEE Communications Magazine*, pp. 58-62, Setembro 1991.
- [Proen91] P. Proença, "Descrição de um device-driver para OS-9," INESC Doc. interno, Novembro 1991.
- [Pryck91] M. de Prycker, *Asynchronous Transfer Mode: Solution for Broadband ISDN*, Ellis Horwood, 1991.
- [RACE91] *RACE Common Functional Specifications, Doc. 3, Networking and Interconnecting 2, Issue B*, Dezembro 1991.
- [RACE92] *Perliminary Version PARASOL Consortium, System Level Description*, Agosto 1992.

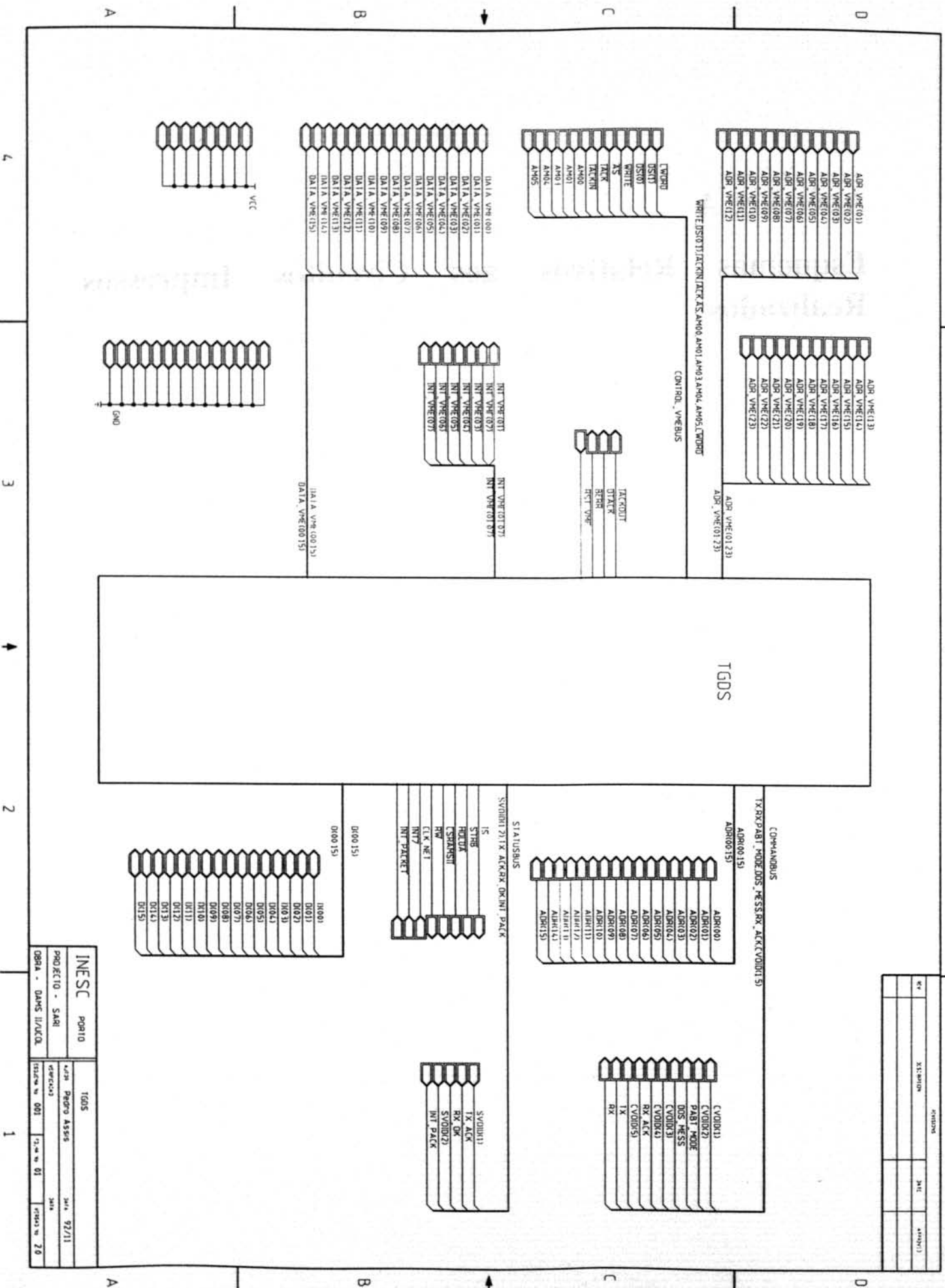
- [Rathg91] E. P. Rathgeb, "Modeling and performance comparison of policing mechanisms for ATM networks," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 325-334, Abril 1991.
- [Reiss93] L. K. Reiss e L. F. Merakos, "Shaping of virtual path traffic for ATM B-ISDN," *Proceedings of the IEEE INFOCOM 93*, pp. 168-175.
- [Rober91] J. Roberts, "Variable-bit-rate traffic control in B-ISDN," *IEEE Communications Magazine*, pp. 50-56, Setembro 1991.
- [Ross86] F. E. Ross, "FDDI - a tutorial," *IEEE Communications Magazine*, pp. 10-17, Maio 1986.
- [Saito90] H. Saito, "Optimal queueing discipline for real-time traffic at ATM switching nodes," *IEEE Trans. Commun.*, vol. 38, pp. 2131-2136, Dezembro 1990.
- [Saito91] H. Saito, M. Kawarasaki e H. Yamada, "An analysis of statistical multiplexing in an ATM transport network," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 359-367, Abril 1991.
- [San-Qi90] S-Q. Li e J. W. Mark, "Traffic characterization for integrated services networks," *IEEE Trans. Commun.*, vol. 38, pp. 1231-1243, Agosto 1990.
- [Sauln92] E. T. Saulnier e K. S. Vastola, "A HI-LO Markov chain model for multimedia traffic in ATM networks," *Proceedings of the IEEE Globecom 92*, pp. 1450-1454.
- [Shear91] D. Shear, "EDN DSP chip directory," *EDN*, pp. 104-133, 1 de Outubro 1991.
- [Small91] C. H. Small, "FPGA design methods," *EDN*, pp. 114-122, 5 de Agosto 1991.
- [Sohra92] K. Sohraby, "Heavy traffic multiplexing behavior of highly-bursty heterogeneous sources and their admission control in high-speed networks," *Proceedings of the IEEE Globecom 92*, pp. 1518-1523.
- [Sohra93] K. Sohraby, "On the theory of general ON-OFF sources with applications in high-speed networks," *Proceedings of the IEEE INFOCOM 93*, pp. 401-410.
- [Stavr92] I. Stavrakakis, "Statistical multiplexing of correlated slow traffic sources," *Proceedings of the IEEE Globecom 92*, pp. 1540-1544.
- [Stuck85] B. W. Stuck e E. Arthurs, *A Computer & Communications Network Performance Analysis Primer*, Prentice-Hall, 1985.
- [Texas86] Texas Instruments, *TMS320C25 Users's Guide (perliminary)*, Texas Instruments, 1986.

- [Wande92] "Wandel & Golterman, news and information," *Wandel & Golterman GmbH & Co, Issue 63*, pp. 8-11, Julho 1992.
- [Warre90] A. Warren, "Designers pack intelligent, memory speed into new DSPs," *Computer Design*, pp. 82-97, 1 de Abril 1990.
- [Weiss91] R. Weiss, "EDN special report: 32 bit floating-point DSP processors," *EDN*, pp. 127-146, 7 de Novembro 1991.
- [Willi84] G. F. Williams e A. Leon-Garcia, "Performance analysis of integrated voice and data hybrid-switched links," *IEEE Trans. Commun.*, vol. COM-32, pp. 695-706, Junho 1984.
- [Yamad91] H. Yamada e S. Sumita, "A traffic measurement method and its application for cell loss probability estimation in ATM networks," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 315-324, Abril 1991.
- [Zhang92] Z. Zhang e A. S. Acampora, "Effect of ON/OFF distribution on the cell loss probability in ATM networks," *Proceedings of the IEEE Globecom 92*, pp. 1533-1539.
- [Zhang93] J. Zhang, "Performance study of Markov modulated fluid flow models with priority traffic," *Proceedings of the IEEE INFOCOM 93*, pp. 10-17.



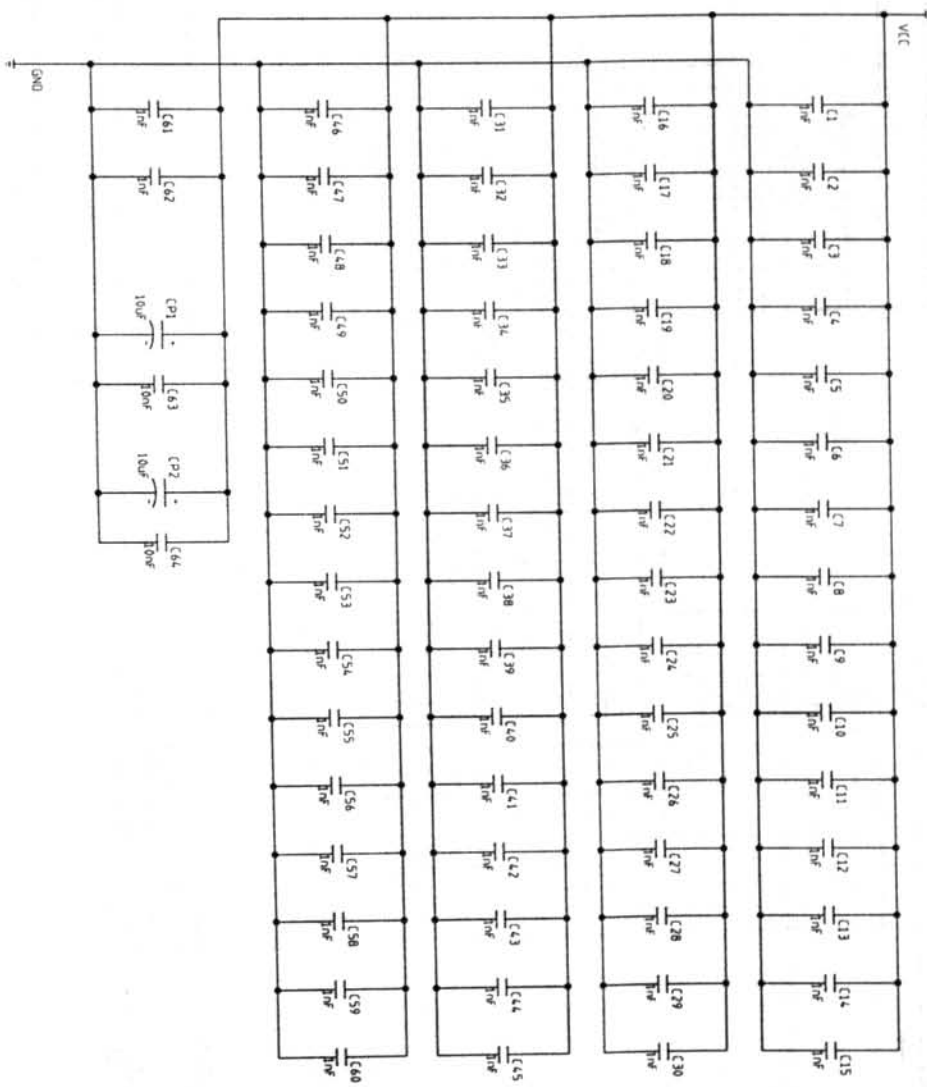
Apêndice A

Esquemas Relativos aos Circuitos Impressos Realizados



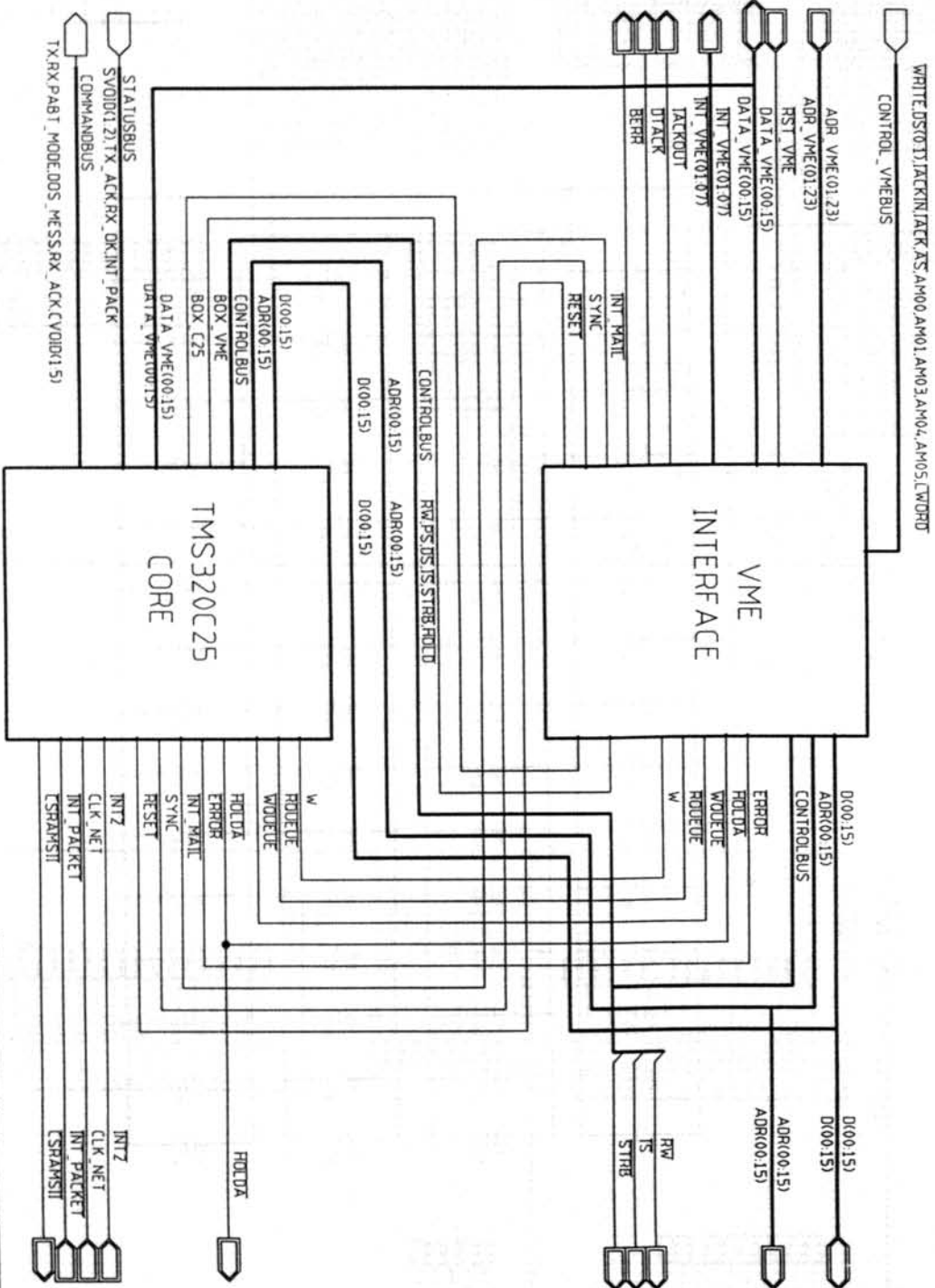
REV	31.05.2010	341	4449(2)
FUNÇÕES			

INESC		PQ0110	
PROJETO - SARI			
1605		341 - 9/2/11	
PROJETO ASSIS			
VERSÃO	PROJ. Nº	3.ª. Nº	PRIMEIRO Nº
001A	001	01	20

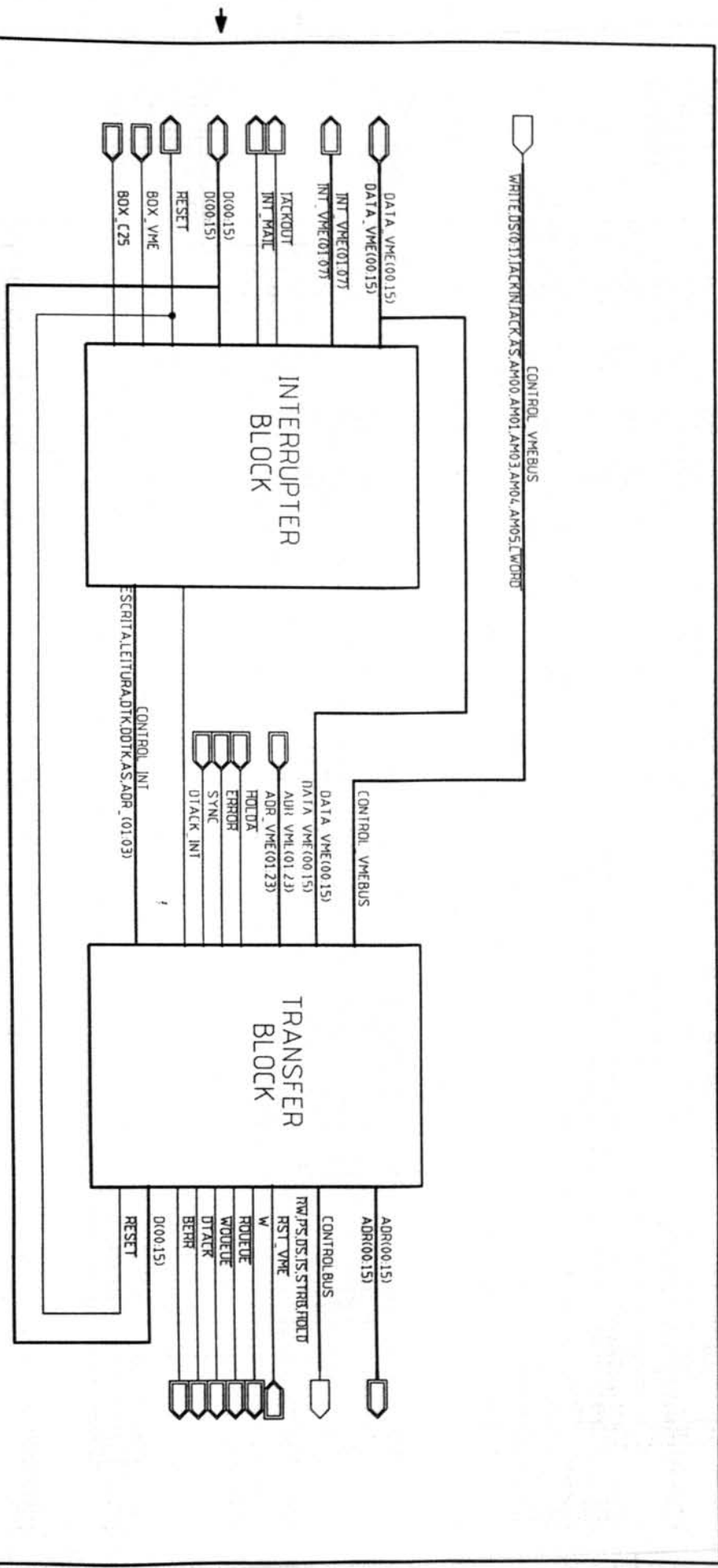


REV	REVISÃO	DATA	ELABORADO

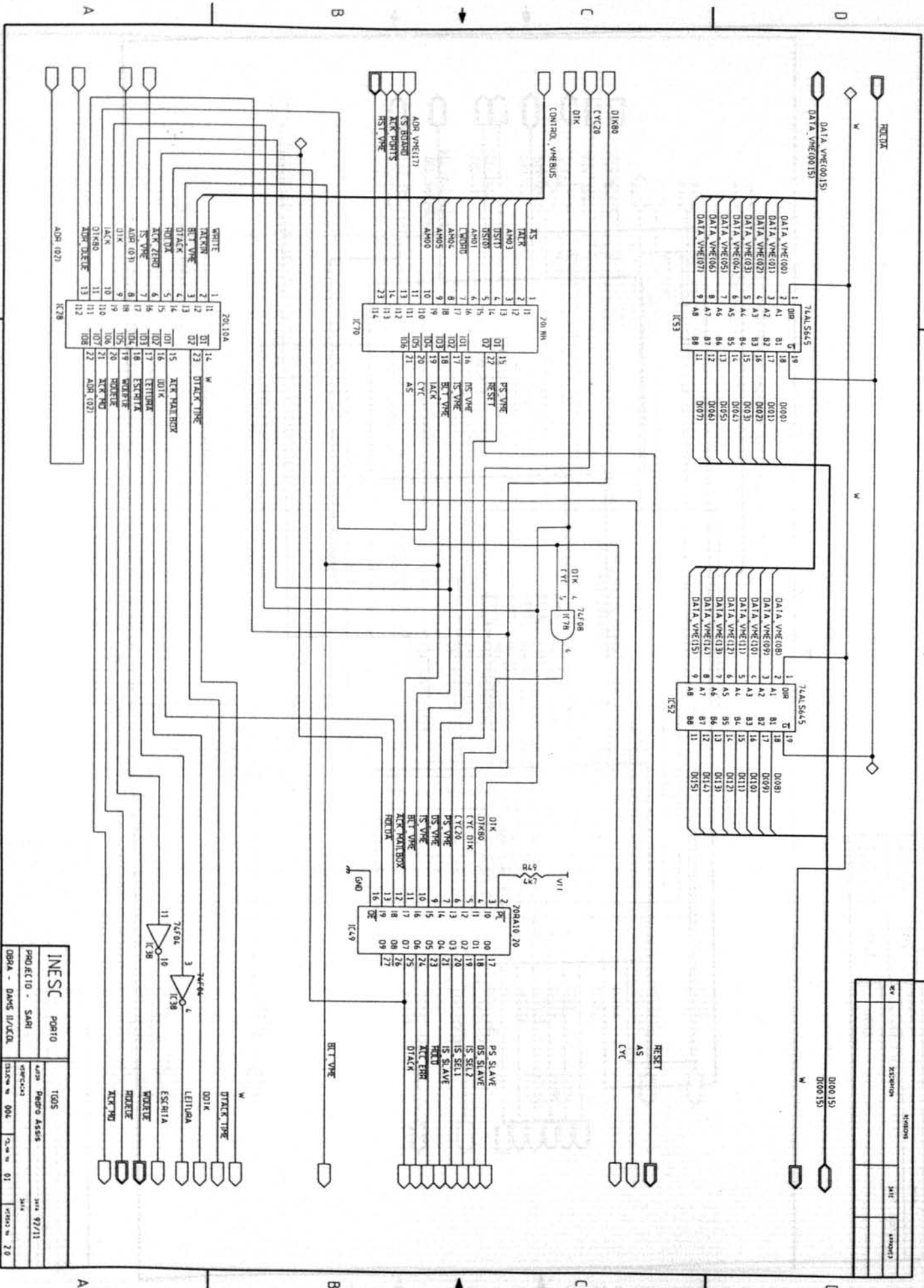
INESC - PDR10		T005	
PROJETO - SARI		DATA - 02/11	
PROFESSOR		DATA	
OBRA - DAVS II/LC/LC		VERSÃO - 2.0	



INESC NORTE		T605	
PROJECIO - SARI	Autor Pedro ASSIS	DATA 92/11	
OBRA - DAMS II/JCOL	VERIFICADO	DATA	
	ESPECIA No 002	FOLHA No 01	VERSÃO No 2.0

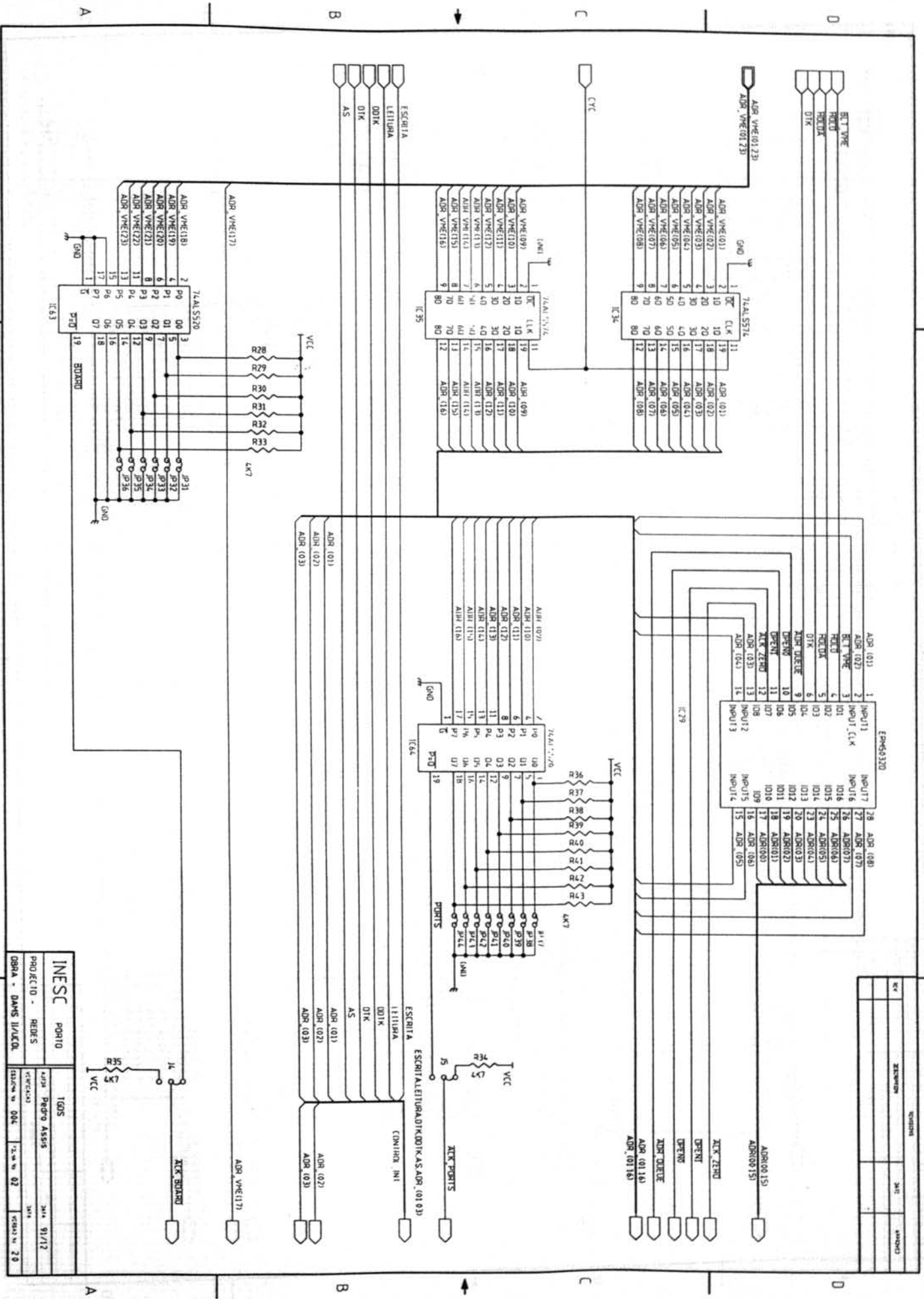


INESC NORTE		TGDS	
PROJECTO - SARI		Autor Pedro ASSIS	
OBRA - DAMS II/UCOL		VERIFICAÇÃO	
ESPECIM. Nº 003		FOLHA Nº 01	
		DATA 92/11	
		VERSÃO Nº 2.0	



REV	ALTERAÇÃO	DATA	PROJETO

INESC PORTIO		TIGOS	
PROJETO - SARI	DESIGNADO	ASSINADO	DATA 92/11
OPERA - DANEI B/U/C/D	VERSÃO 004	DATA 01	FOLHA 20



INESC PORTIO

IGOS

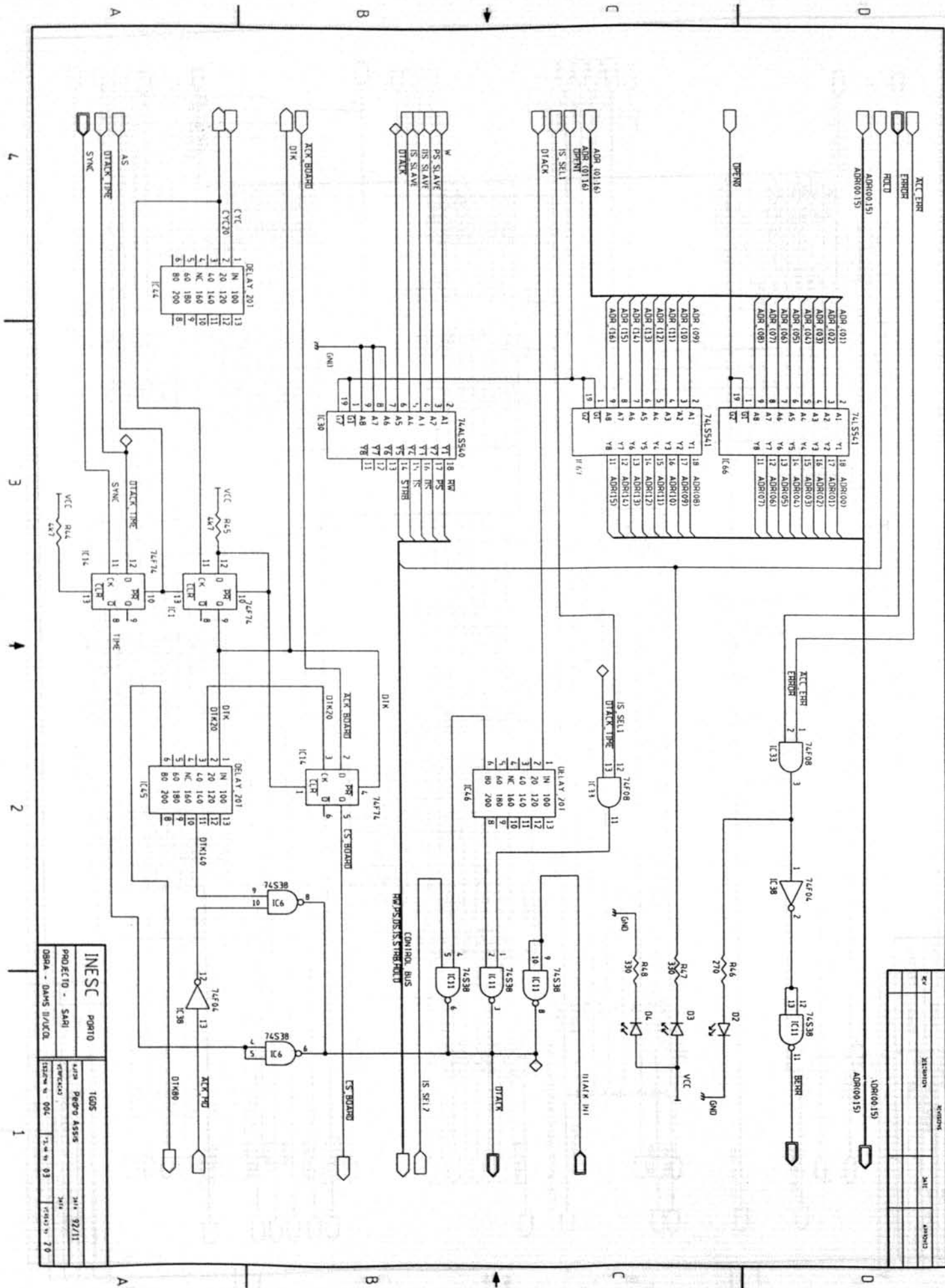
PROJETO - REDES

OBRA - DAVS II/UCOL

DATA 91/12

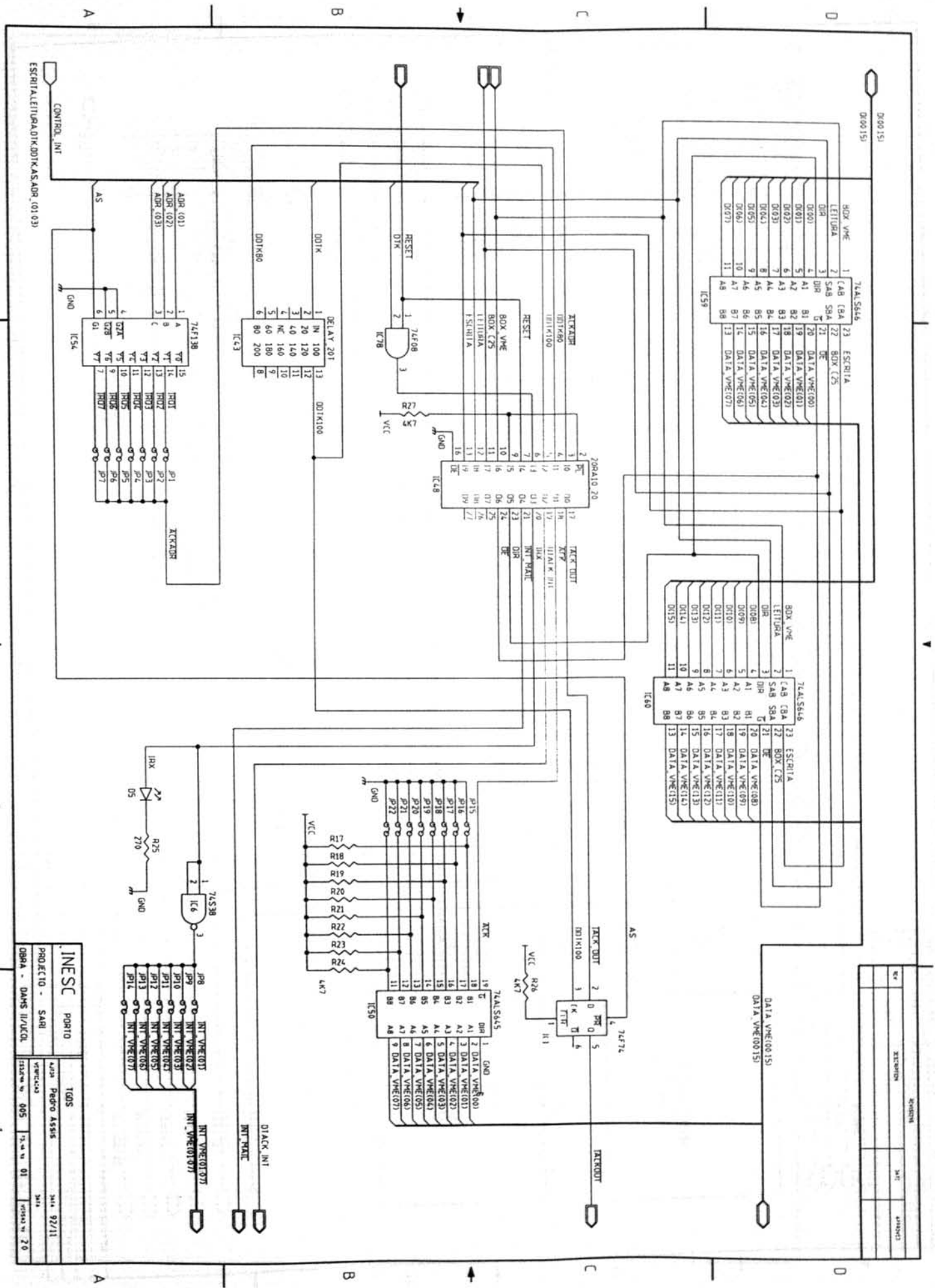
15.000 02

20



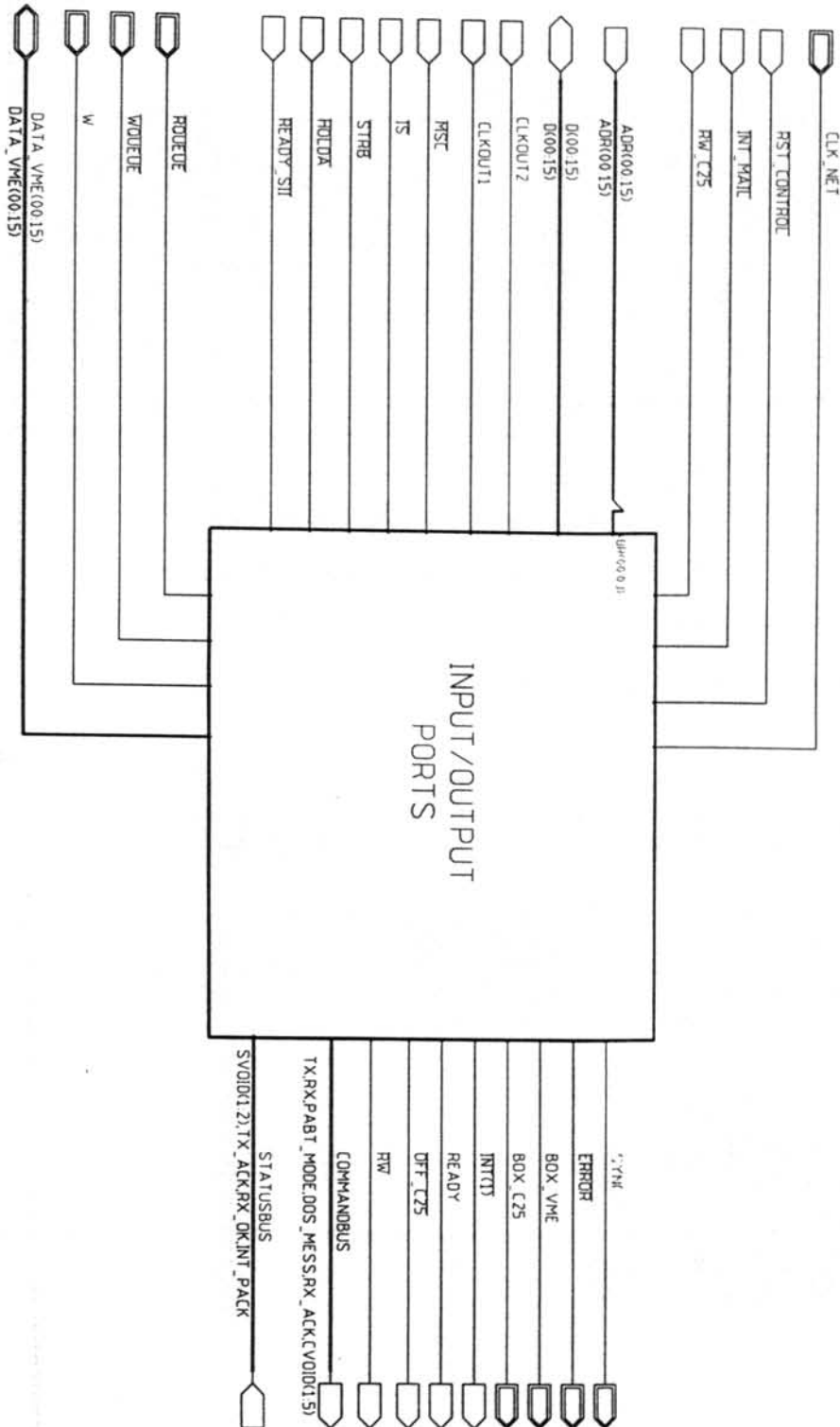
REV	ALTERAÇÃO	DATA	AUTORIA

INESC		1005	
PDR10		1005	
PROJETO - SAEL		DATA - 02/11	
ORCA - DAHS D/UCLO		REVISÃO - 03	
		FECHA - 20	

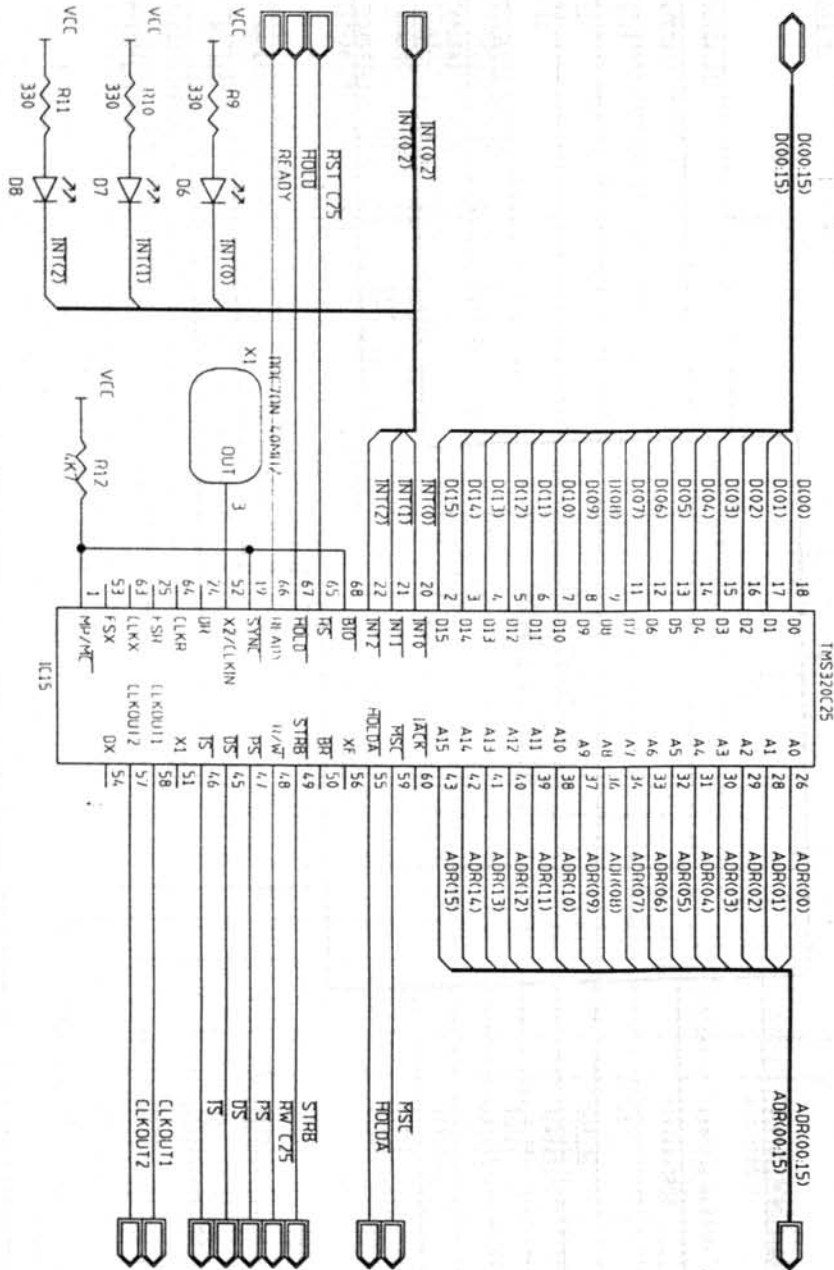


REV.	REVISÃO	DATA	ELABORADO

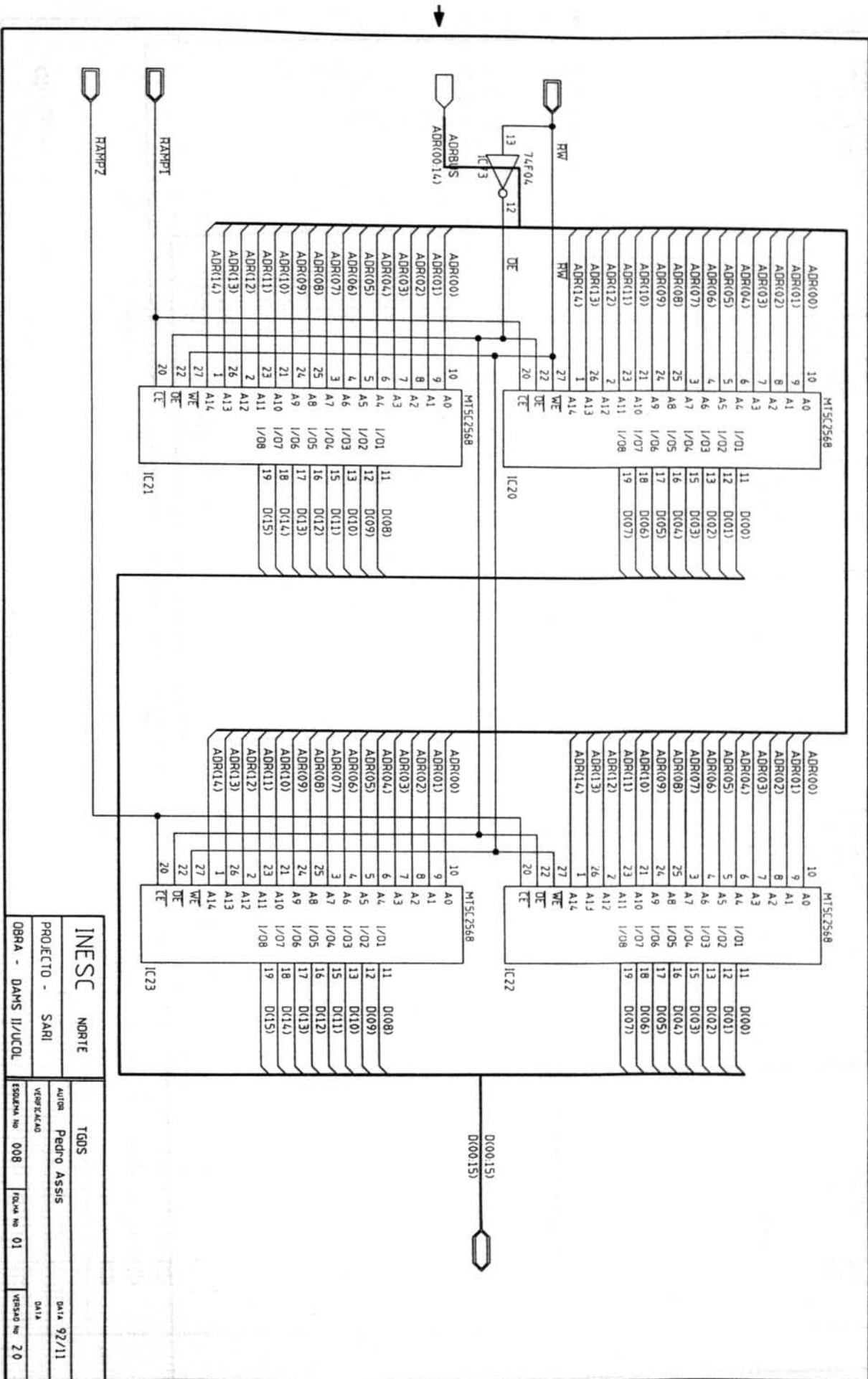
INESC		TGDS	
PORTIO		Pedro Assis	
PROJECTO - SARI		DATA 92/11	
VERIFICACAO		DATA	
OBRA - DAWS II/UCOL		FOLHA Nº 01	
TOTAL Nº 095		TOTAL Nº 20	



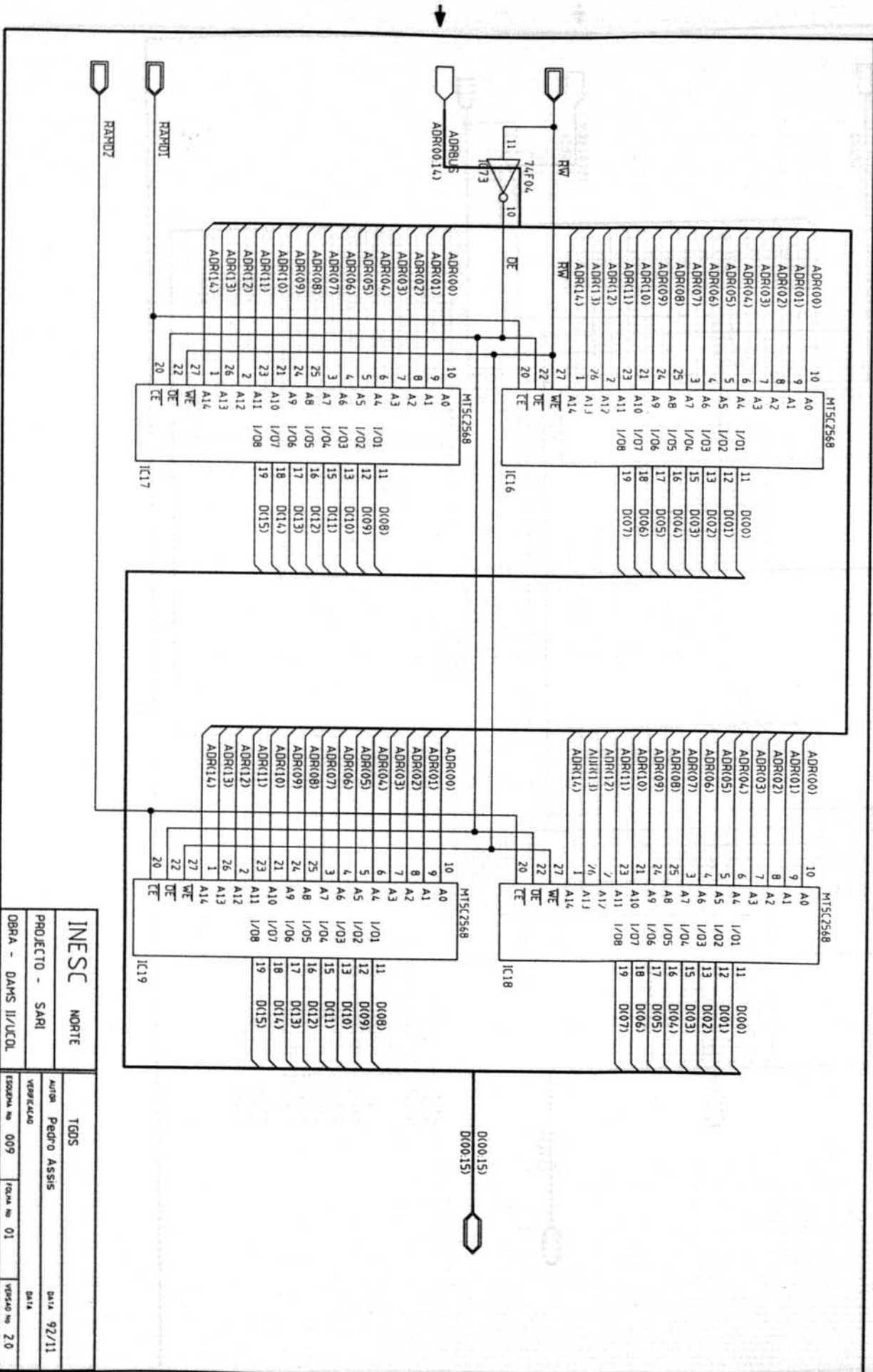
INESC NORTE		TGDS	
PROJECTO - SARI	AUTOR Pedro Assis	FECHA 92/11	DATA
VERIFICACAO	ESQUEMA Nº 006	FOLHA Nº 02	VERSÃO Nº 2.0
OBRA - DAHS 11/UJOL			



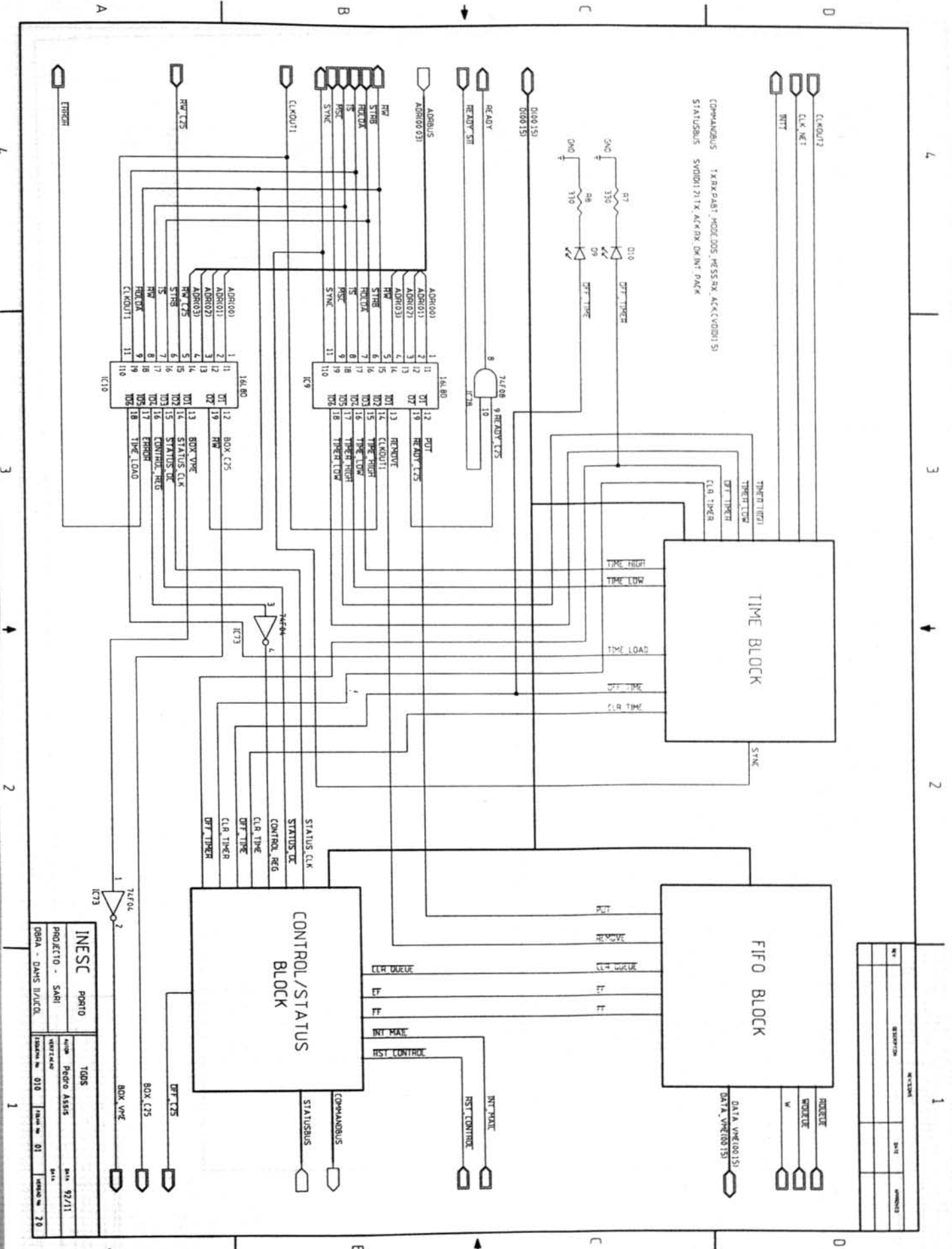
INESC		NDRITE	
PROJECTO - SARI		TGDS	
OBRA - DAMS II/UCOL		AUTOR Pedro ASSIS	
ESQUEMA No 007		DATA 92/11	
FOLHA No 01		DATA	
VERSÃO No 2.0			



INESC		NDPTE	
PROJECTO - SARI		TGDS	
AUTOR - PEDRO ASSIS		DATA - 92/11	
VERIFICADO		DATA	
OBRA - DAMS II/UCOL	ESQUENA No 008	FOLHA No 01	VERSÃO No 2 0

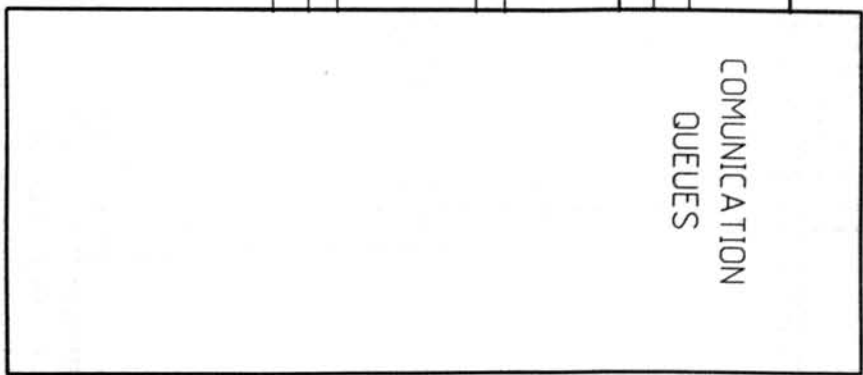
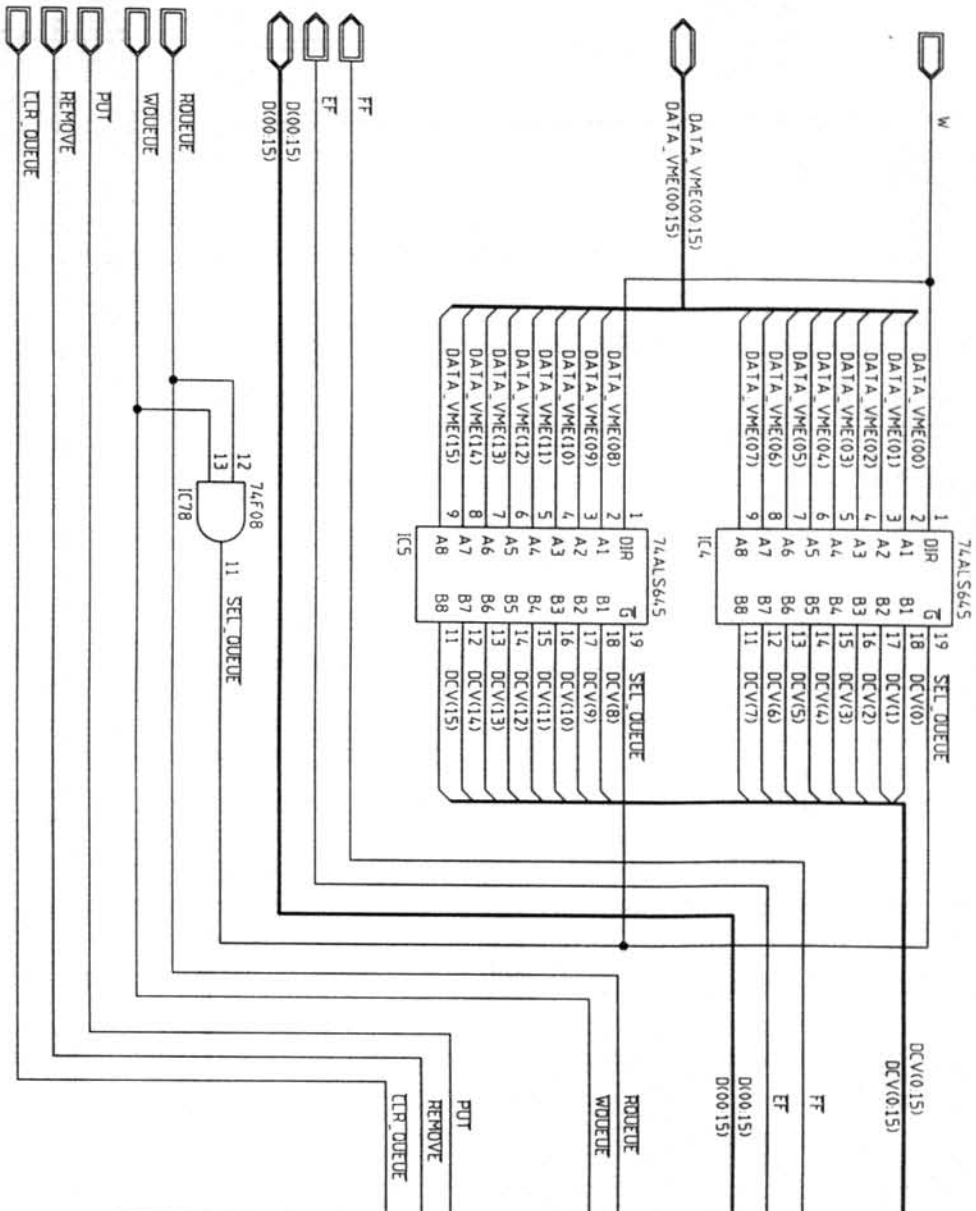


INESC NORTE		TGDS	
PROJECTO - SARI	AUTOR Pedro Assis	DATA 92/11	
OBRA - DAMS II/UCOL	VERIFICADO	DATA	
	ESCALA No 009	FOLHA No 01	VERSÃO No 2.0

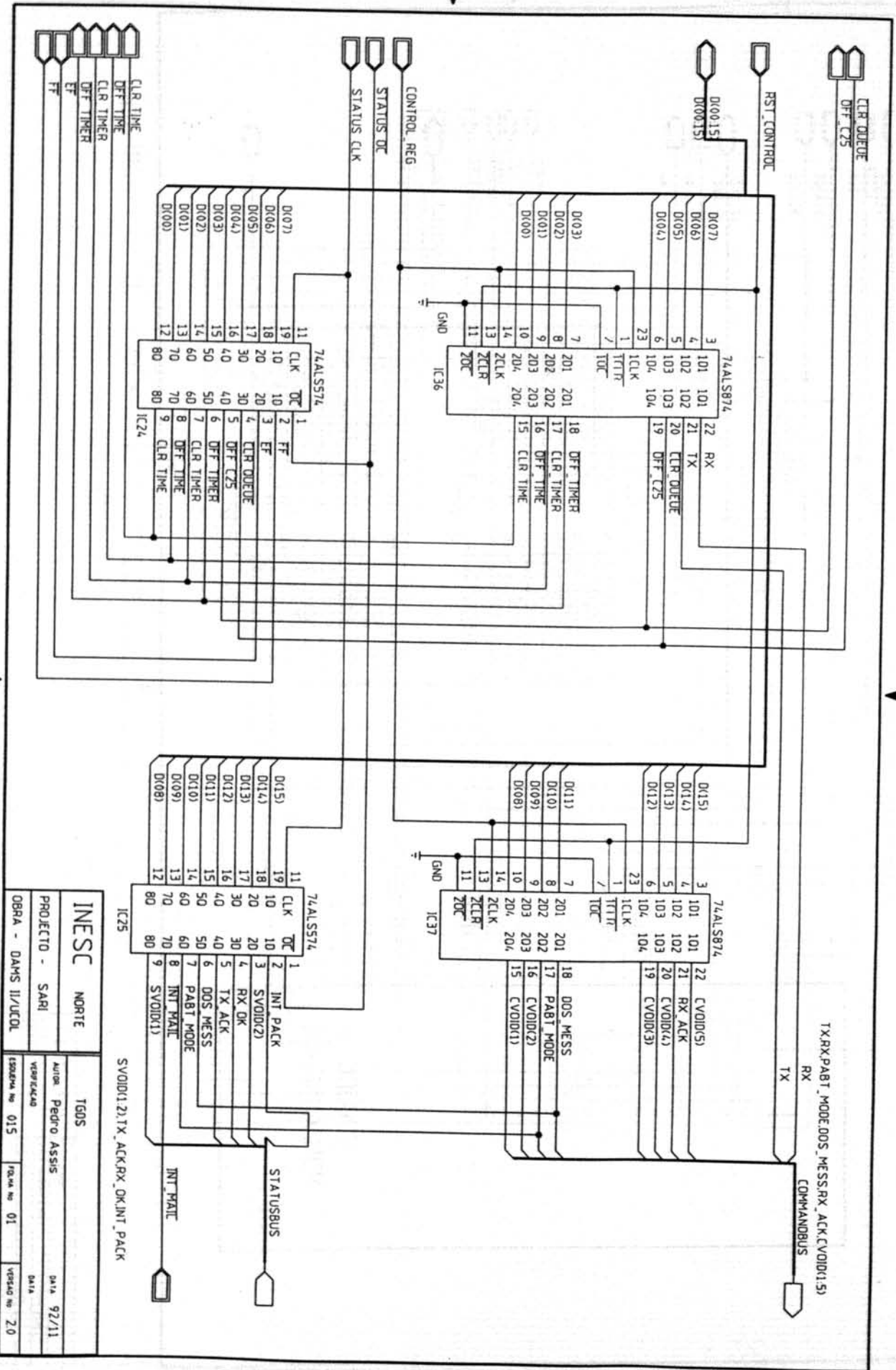


REV	DESCRIPTION	DATE	APPROVED

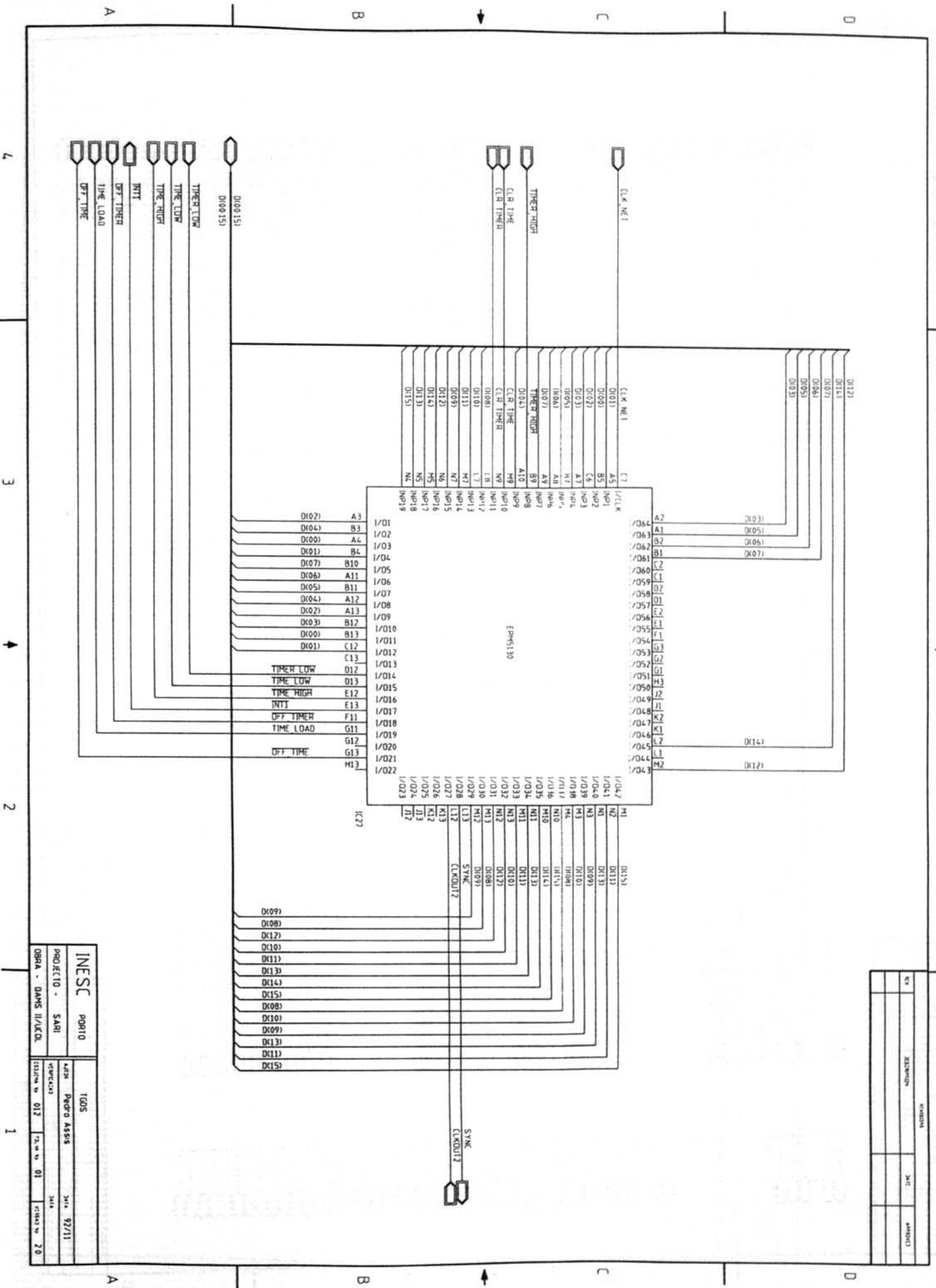
INESC			
PORTO	TGDS	DATA VERR(0015)	
PROJECTO - SARI	supp Pedro Assis	DATA VERR(0015)	
versão	versão	DATA VERR(0015)	
000A - DAHS I/JULIO	010	01	7.0

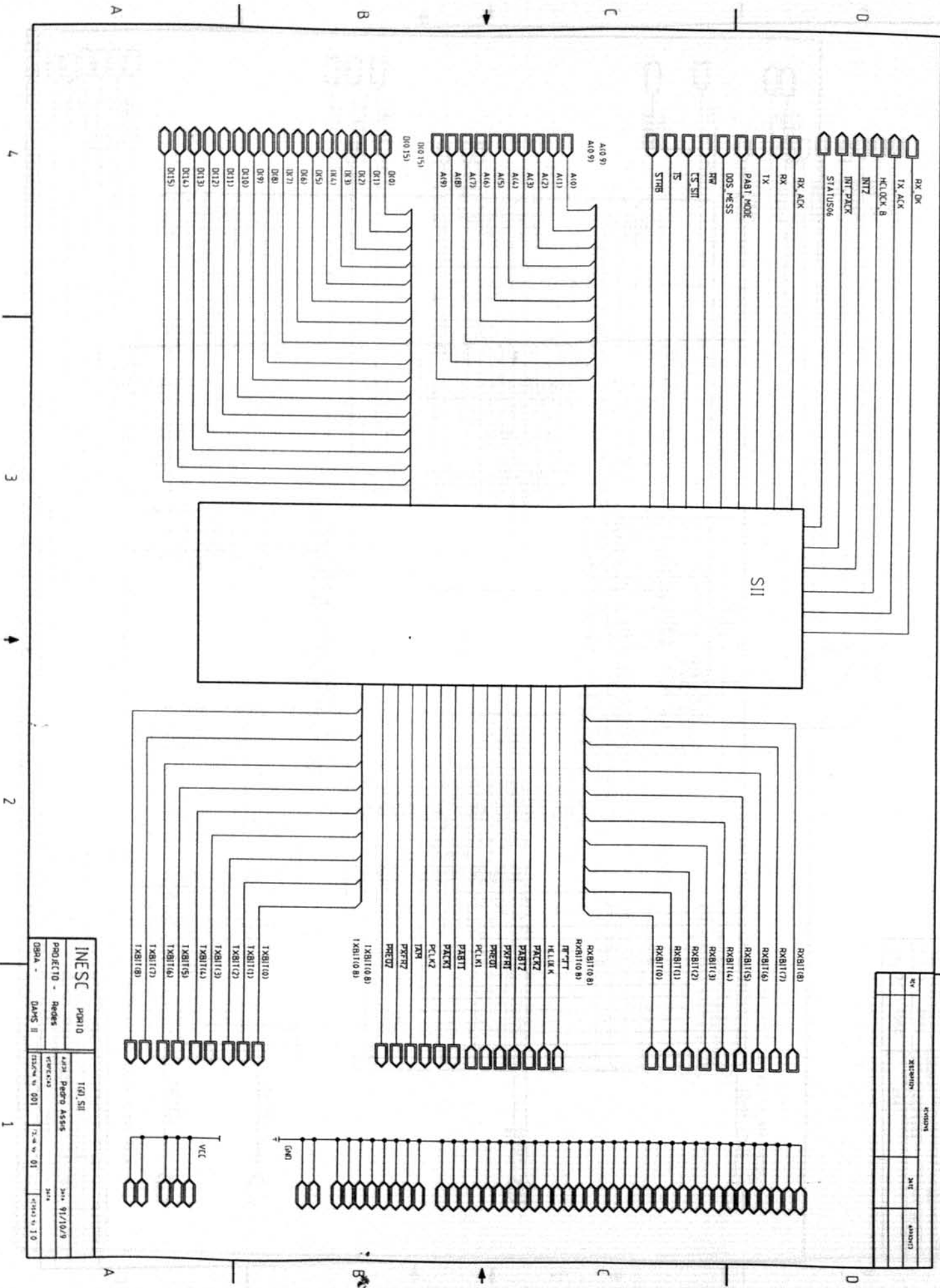


INESC NORTE		TG05	
PROJECTO - SARI		AUTOR - Pedro Assis	
OBRA - DAMS II/UCOL		VERIFICAO	
		1502/94 No 013	
		10/94 No 01	
		DATA	
		1992/11	
		VERSAO No 2.0	



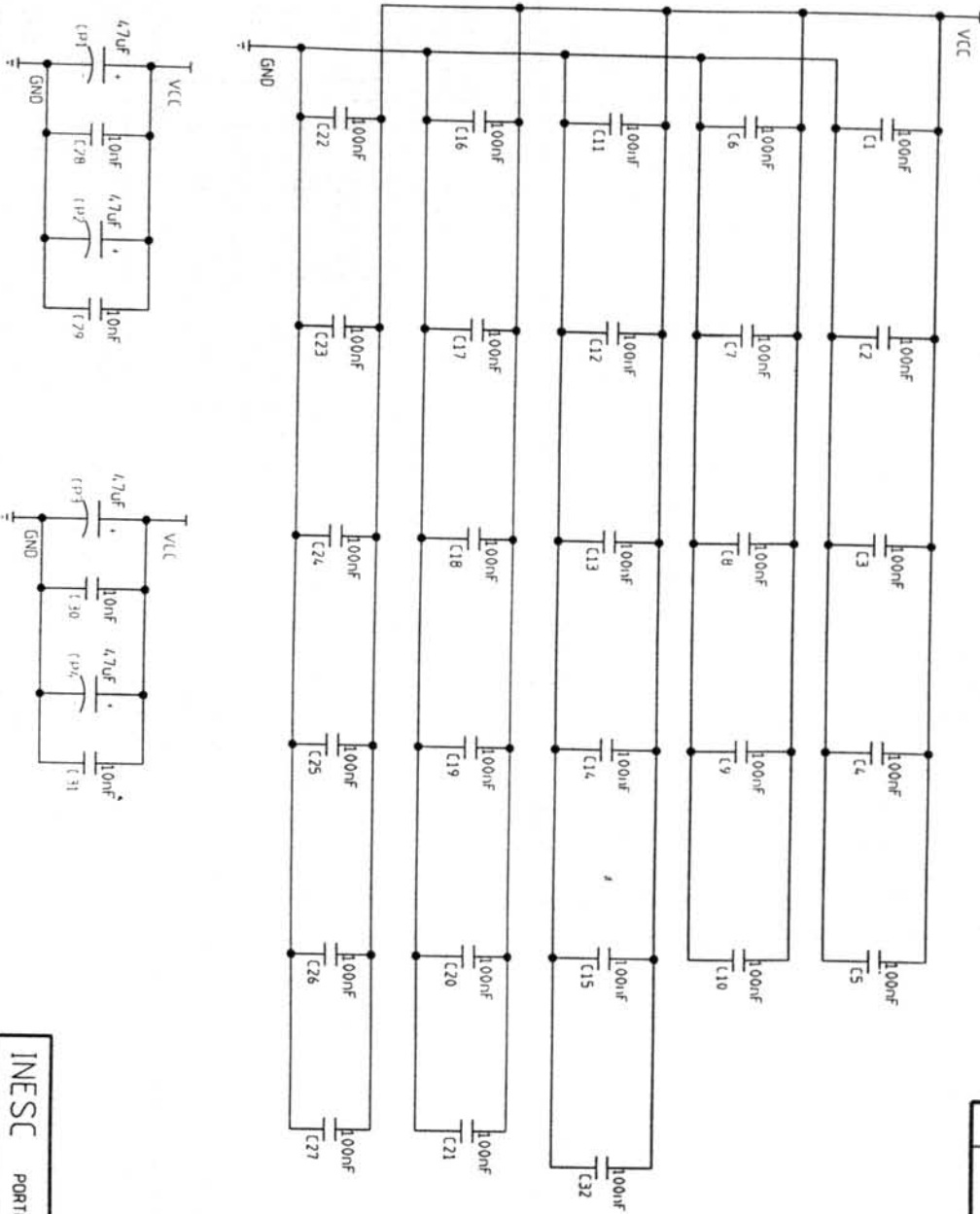
INESC		NDORITE		TGDS	
PROJECTO - SARI		AUTOR Pedro ASSIS		DATA 92/11	
OBRA - DAMS II/JCOL		VISTOR/ELABO		DATA	
		ESQUEMA No 015		FOLHA No 01	
				VERSÃO No 20	





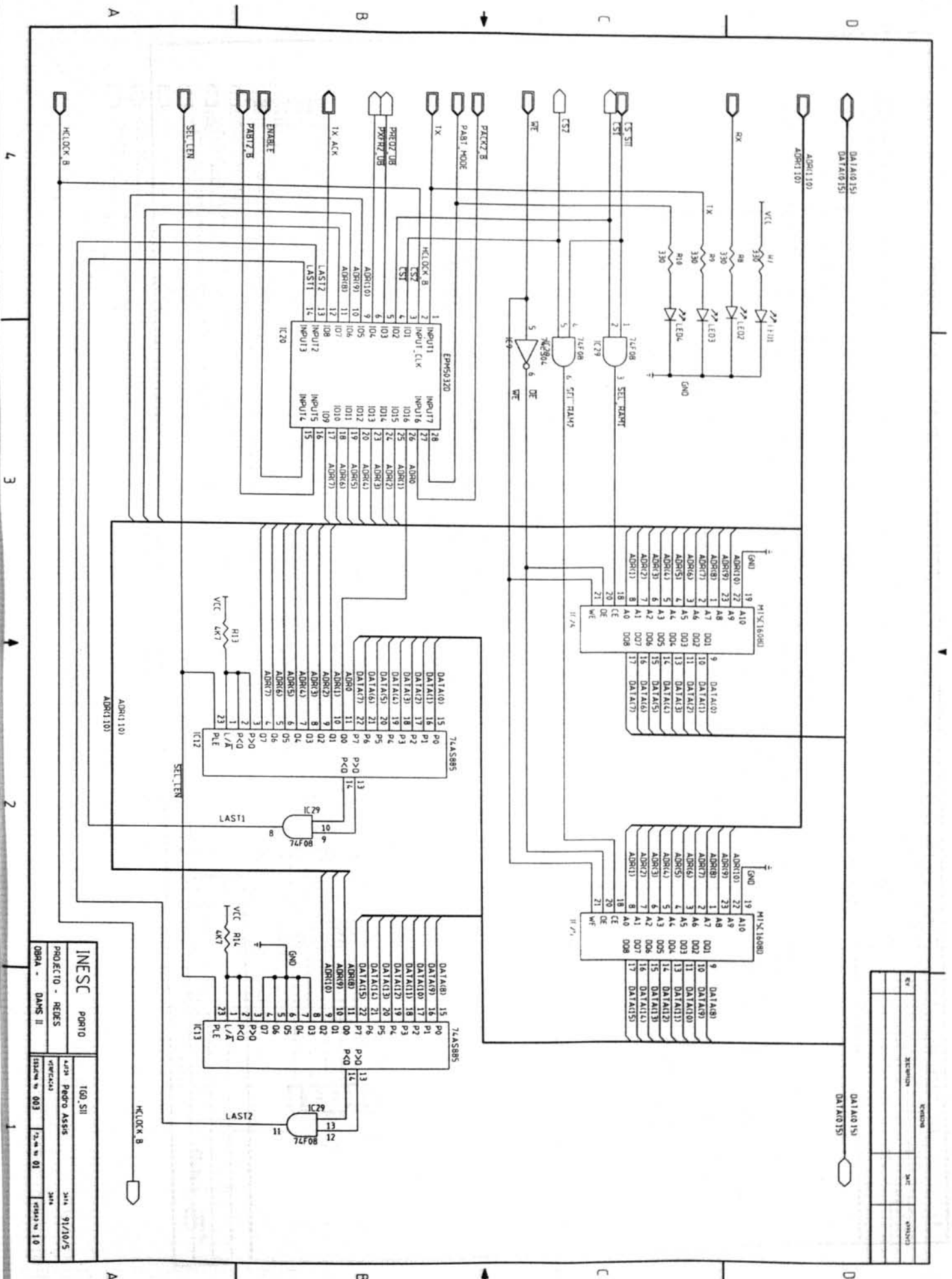
REV	DESCRIPTION	DATE	APPROVED

INESC PDR110
 PROJEITO - Redes
 OBRAS - OANS II
 100 SII
 Edm. Pedro Assis
 2011 01/10/09
 13200 n. 001
 2.0 n. 01
 1.0



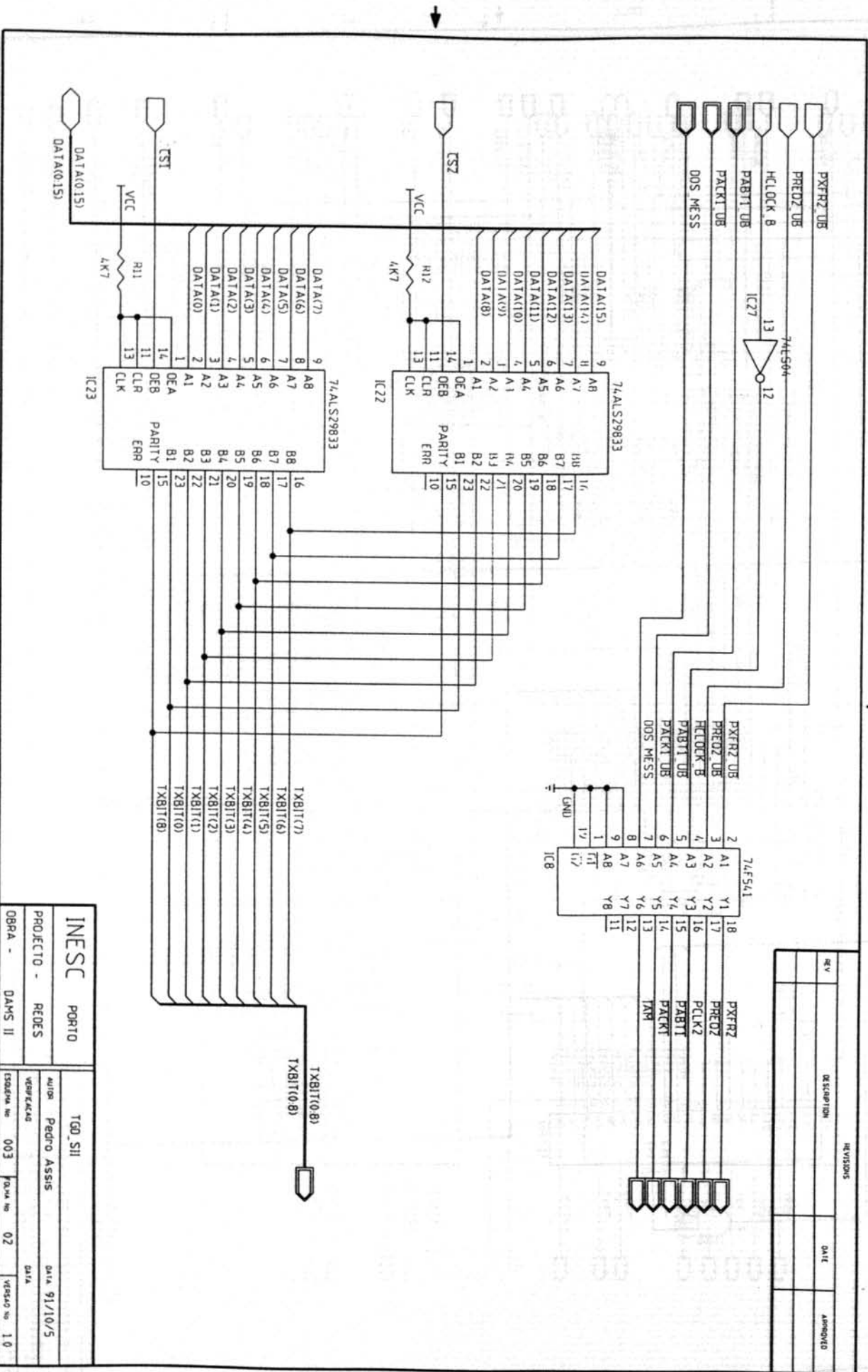
REVISIONS		
REV	DESCRIPTION	DATE

INESC PORTIO		TGD_SII	
PROJECTO - REDES		AUTOR Pedro ASSIS	
OBRA - DANIS II		VERIFICAO	
15000PM No 001	DATA No 02	DATA 91/10/5	VERSIC No 10



INESC PORTO		TGO, SII	
PROJETO - REDES		MIRA Pedro ASSIS	
VERSÃO	003	DATA	91/10/5
OBRA -	DANS II	FECHA	01
		FECHA	10

REV	REVISÃO	DATA	REVISÃO



REV	DESCRIPTION	DATE	APPROVED

INSC		PORTO		TGO SII	
PROJECTO - REDES		AUTOR - Pedro Assis		DATA - 91/10/5	
OBRA - DAVS II		VERIFICAD		DATA	
		ESTADIA No 003		FOLIA No 02	
				VERSÃO No 1.0	

Apêndice B

Análise Computacional de Cadeias de *Markov* Finitas, de Índice Discreto e Estacionárias

Para a análise computacional de cadeias de *Markov* egórdicas e não egórdicas foi desenvolvido um programa denominado MK. Os resultados fornecidos dependem da classificação da cadeia de *Markov*, ou seja, no caso da cadeia de *Markov* ser egórdica é calculada a distribuição estacionária e o tempo de médio de absorção (se existirem estados transitórios), para cadeias de *Markov* não egórdicas os seus estados são classificados sendo a matriz de transição reescrita sob a forma de bloco e, finalmente, o tempo médio e a probabilidade de absorção dos estados transitórios são obtidos.

Os procedimentos utilizados para determinar os resultados indicados são baseados nos fluxogramas apresentados no capítulo 4. O programa elaborado determina, num primeiro passo, o coeficiente egórdico da cadeia de *Markov* (método da soma coluna) permitindo a classificação da mesma, e consequentemente os cálculos a efectuar para obter os resultados desejados.

Os dados são introduzidos através da utilização de dois ficheiros, com extensões *par* e *tpl*, que contêm valores que caracterizam a cadeia de *Markov*, no primeiro são descritos parâmetros como o número de estados (nós) que a cadeia contém. No segundo tipo ficheiro de dados (com extensão *tpl*) descreve-se a topologia da cadeia, isto é, as ligações existentes (nós que se encontram interligados com indicação do sentido da ligação), assim como o peso afecto a cada uma dessas ligações descritas.

Na concepção do programa MK foram contemplados diversos tipos de protecções, que englobam a validação dos dados fornecidos, como a verificação dos operandos antes da execução de certas operações. Para exemplificar este ponto apresentam-se algumas das mensagens existentes no ficheiro *err.h* (contém as mensagens de erro fornecidas pelo programa MK):

```
/* err.h */
```

```
#define utilização ("Utilização: MK [-t nome do ficheiro com a topologia da cadeia de Markov] [-p nome do ficheiro que contém parâmetros descritivos da cadeia de Markov]n")
```

```
#define erro1 ("nErro: Formato dos dados do ficheiro de topologia é inválido. \n")
```

```
#define erro2 ("Erro: Ficheiro não existe. \n\n")
```

```
#define erro5 ("Erro: Divisão por zero, matriz singular. \n\n")
```

Para descrever o funcionamento deste programa é apresentado um exemplo de uma cadeia de Markov finita. Esta é representada através do grafo orientado representado na Fig. B.1, e pela matriz de transição de estados (P) (Eq. B.1),

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 0 & 1/4 & 0 & 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 \end{bmatrix} \quad (\text{Eq. B.1})$$

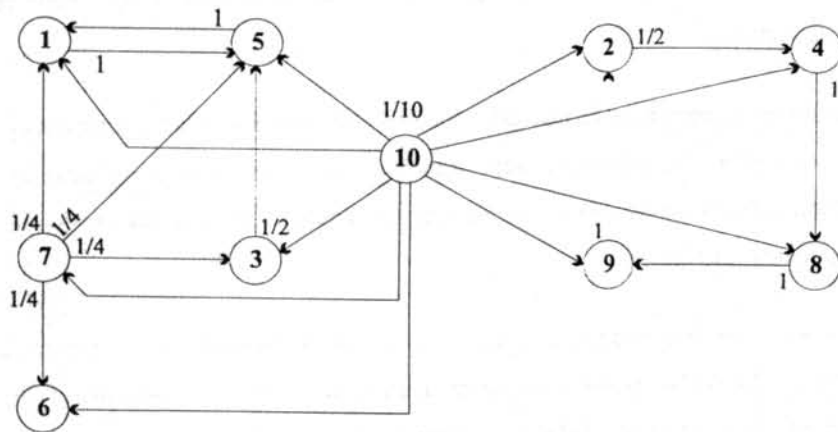


Fig. B.1 Grafo orientado da cadeia de Markov em análise.

Para este exemplo, o conteúdo e a sintaxe dos ficheiros referidos são apresentados a seguir.

- *exemplo.par*

Indica o número de nós

nr_nodes 10

- *exemplo.tpl*

O dados têm o seguinte formato: *i* <espaço> *j* <espaço> *k* <nova linha>

em que *i* e *j* são os nós que se encontram ligados e *k* é o peso associado à transição *i* para *j*

1 5 1
 2 2 0.5
 2 4 0.5
 3 3 0.5
 3 5 0.5
 4 8 1
 5 1 1
 6 6 1
 7 1 0.25
 7 3 0.25
 7 5 0.25
 7 6 0.25
 8 9 1
 9 2 1
 10 1 0.25
 10 2 0.25
 10 3 0.25
 10 4 0.25
 10 5 0.25
 10 6 0.25
 10 7 0.25
 10 8 0.25
 10 9 0.25
 10 10 0.25

De notar que só são introduzidas as ligações existentes

Os dados fornecidos foram analisados numa estação de trabalho *SPARCstation 10*, tendo-se obtidos os seguintes resultados, os quais podem ser facilmente verificados pela análise da Fig. B.1:

- *exemplo.res*
 Programa em execução ...
 Leitura dos ficheiros de dados → início → fim
 Reserva de memória → início → fim
 Coeficiente egórdico → início
 n = 1 k = 1
 n = 2 k = 2
 n = 4 k = 4
 n = 8 k = 8
 n = 16 k = 16
 n = 32 k = 32
 n = 64 k = 64
 Cadeia não é egórdica → fim
 Análise da cadeia → início
 Estado persistente: 2 → início
 Conjunto gerado pelo estado persistente 2
 K0 : 2
 K0 : 4
 K0 : 8
 K0 : 9
 K1 : 2
 K1 : 4
 K1 : 8
 K1 : 9
 Períodicidade do estado 2: 1

Estado transitório: 10

Estado persistente: 2 → fim

Número de estados para analisar: 5

Estado persistente: 1 → início

Conjunto gerado pelo estado persistente 1

K0 : 1

K1 : 5

K1 : 1

Periodicidade do estado 1: 2

Estado transitório: 3

Estado transitório: 7

Estado persistente: 1 → fim

Número de estados para analisar: 1

Estado persistente: 6 → início

Conjunto gerado pelo estado persistente 6

K0 : 6

K1 : 6

Periodicidade do estado 6: 1

Estado persistente: 6 → fim

Número de estados para analisar: 0 → fim

Resultados:

Matriz P reescrita sob a forma de bloco

										estado
0.50	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2
0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4
0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	8
1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	9
0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	1
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	5
0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	6
0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	10
0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.50	0.00	3
0.00	0.00	0.00	0.00	0.25	0.25	0.25	0.00	0.25	0.00	7

Tempo de médio de absorção:

10: 1.5

3: 2.0

7: 1.5

Matriz NR

							estado
0.111	0.111	0.111	0.111	0.138	0.277	0.138	10
0.000	0.000	0.000	0.000	0.000		0.000	3
0.000	0.000	0.000	0.000	0.250	0.500	0.250	7

Liberta memória → início → fim.

Apêndice C

Determinação do Gráfico Relativo ao Atraso Médio de um Comutador ATM (16x16(c))

Os cálculos que se seguem destinam-se a exemplificar os procedimentos utilizados para determinar os gráficos apresentados quer no que respeitam à sequência dos cálculos efectuados, quer na utilização do pacote de *software* PARI-GP.

Repete-se, por conveniência, a expressão do atraso médio, $E[D]$, experimentado pelo fluxo de células ATM ao atravessar o comutador ATM (16x16(c)),

$$\bar{D} = \frac{1}{c} \left[\frac{dD(z^c)}{dz} \right]_{z=1} = 1 + \frac{\bar{Q}}{A}. \quad (\text{Eq. C.1})$$

Desenvolvendo a expressão anterior resulta,

$$\bar{D} = D'(1) = \frac{1}{A'(1)} \left[\frac{1}{2} \prod_{j=1}^{c-1} \frac{z+z_j}{1-z_j} + A'(1) + \frac{A''(1) - (c-1)A'(1)}{2(c-A'(1))} \right], \quad (\text{Eq. C.2})$$

em que $A'(1)$ e $A''(1)$ são obtidos por derivação no ponto $z = 1$ da *fg* do tráfego injectado na(s) fila(s) de espera (função binomial),

$$A'(1) = \left[\frac{dA(z)}{dz} \right]_{z=1} = p * c, \quad (\text{Eq. C.3})$$

$$A''(1) = \frac{N-1}{N} (p * c)^2 = 0.9375 (p * c)^2, \quad (\text{Eq. C.4})$$

substituindo na Eq. C.2, obtém-se

$$\bar{D} = D'(1) = \frac{1}{2 * p[n] * c} \prod_{j=1}^{c-1} \frac{z + z_j}{1 - z_j} + 1 + \frac{0.9375 * p[n] - 1 + 1/c}{2 * (1 - p[n])} \quad \text{e } n = 0, 1, \dots, 19. \quad (\text{Eq. C.5})$$

O cálculo da expressão anterior foi efectuado para um conjunto discreto de 20 pontos (n). Na avaliação desta expressão é necessário determinar as raízes z_j , para cada ponto, da seguinte equação,

$$z^c - \left(1 - \frac{p[n] * c}{16} + \frac{p[n] * c * z}{16} \right)^{16} = 0 \quad \text{e } n = 0, 1, \dots, 19. \quad (\text{Eq. C.6})$$

Para realizar este cálculo (3*16*20 raízes complexas) foi utilizado o pacote PARI-GP, este permite obter de uma forma fácil (e rápida) as raízes desejadas como é ilustrado pela sequência de comandos apresentada a seguir,

`\r raízes de f(z) = z^c - A(z)`

`p = vector(20, n, 0);`

`for(n = 1, 20, p[n] = (n - 1)/20);`

`for(j = 1, 3, c = 2^(j - 1); for(n = 1, 20, r = roots(z^c - (1 - p[n]*c/16 + p[n]*c*z/16)^16); pprint(r)));`

O número de raízes existentes dentro do círculo unitário pode ser determinado, formalmente, pelo teorema de *Rouché* utilizando a circunferência de raio unitário para o contorno C e para f(z) e g(z) as seguintes expressões (recordar a Eq. 5.7):

$$f(z) = \sum_{j=0}^{c-1} z^c * P[S = j], \quad (\text{Eq. C.7})$$

$$g(z) = - \sum_{j=0}^{c-1} z^j * P[S = j]. \quad (\text{Eq. C.8})$$

Aplicando este teorema conclui-se que o número de raízes dentro do círculo unitário para $c = 1, 2$ e 4 é respectivamente 0, 1 e 3 (existindo em todos os casos uma raiz para $z = 1$). Estas conclusões reflectem-se na Eq. 5.9, existência do termo $(z - 1)$ e valores assumidos pelos limites do produto.

O cálculo numérico, baseado no extracto de comandos apresentado anteriormente, forneceu os seguintes resultados: para c igual a 1 não existem raízes dentro da circunferência de raio unitário; para c igual a 2 existe uma única raiz real negativa dentro do círculo unitário e para c igual a 4 existem três raízes dentro da circunferência de raio unitário sendo uma real (negativa) e as restantes complexas conjugadas (a parte real é negativa). Na Tabela C.1 apresenta-se as raízes determinadas para c igual a 2 e 4.

Voltando à Eq. C.5 verifica-se que para carga (p) nula o valor fornecido por esta expressão para c qualquer é uma indeterminação do tipo $0/0$. Para levantar esta indeterminação utiliza-se a regra de *L'Hôpital*; o valor calculado é facilmente validado devido ao facto do atraso mínimo ser igual a um *slot* (atraso no encaminhamento das células).

Tabela C.1 Raízes da equação $(z^c - A(z))$, para c igual a 2 e 4.

n	Raízes para $c = 2$	Raízes para $c = 4$	n	Raízes para $c = 2$	Raízes para $c = 4$
0	-1	-1 i $-i$	10	-0,464	-0,449 $-0,16 + 0,553i$ $-0,16 - 0,553i$
1	-0,908	-0,907 $-0,045 + 0,947i$ $-0,045 - 0,947i$	11	-0,435	-0,419 $-0,156 + 0,481i$ $-0,156 - 0,481i$
2	-0,831	-0,829 $-0,082 + 0,893i$ $-0,082 - 0,893i$	12	-0,409	-0,309 $-0,151 + 0,449i$ $-0,151 - 0,449i$
3	-0,764	-0,760 $-0,11 + 0,833i$ $-0,11 - 0,833i$	13	-0,384	-0,366 $-0,146 + 0,42i$ $-0,146 - 0,42i$
4	-0,705	-0,7 $-0,131 + 0,783i$ $-0,131 - 0,783i$	14	-0,362	-0,342 $-0,139 + 0,392i$ $-0,139 - 0,392i$
5	-0,654	-0,647 $-0,146 + 0,732i$ $-0,146 - 0,732i$	15	-0,341	-0,32 $-0,133 + 0,366i$ $-0,133 - 0,366i$
6	-0,608	-0,599 $-0,155 + 0,682i$ $0,155 - 0,682i$	16	-0,321	-0,299 $-0,126 + 0,343i$ $-0,126 - 0,343i$
7	-0,566	-0,556 $-0,16 + 0,636i$ $-0,16 - 0,636i$	17	-0,303	-0,28 $-0,119 + 0,321i$ $-0,119 - 0,321i$
8	-0,529	-0,517 $-0,163 + 0,593i$ $-0,163 - 0,593i$	18	-0,286	-0,262 $-0,112 + 0,3i$ $-0,112 - 0,3i$
9	-0,495	-0,482 $-0,162 + 0,553i$ $-0,162 - 0,553i$	19	-0,270	-0,245 $-0,106 + 0,281i$ $-0,106 - 0,281i$

Neste momento é possível calcular os pontos do gráfico a partir da utilização das raízes apresentadas na tabela anterior. Seguidamente descrevem-se os comandos executados para determinar os pontos $(p[n], E[Q])$.

\ Gráfico de E[Q]

for(w = 0, 2, initrect(w, 350, 350); scale(w, 0, 1, 0, 10); move(w, 0, 0); box(w, 350, 350));

p = vector(20, n, 0);

```

q = vector(20, n, 0);
qa = vector(60, n, 0);
qb = vector(60, n, 0);

for(n = 1, 20, p[n] = (n - 1)/20);

index1 = 21; index2 = 1;

// Cálculo da 1ª parcela

for(j = 1, 3, c = 2^(j - 1); index1 = index1 + 1; index2 = index2 + 2^c - 1; for(n = 2; 20, qa[index1] =
0.5*(p[n]*2^c); for(nc = 1, 2^c - 1, qa[index1] = qa[index1]*(1 + raízes[index2])/(1 - raízes[index2]));
index2 = index2 + 1); index1 = index1 + 1));

\\ Cálculo da 2ª parcela

index1 = 1;

for(j = 1, 3, c = 2^(j - 1), for(n = 1; 20, qb[index1] = (0.9375*p[n] - 1 + 1/2^c)/(2*(1 - p[n])); index1 =
index1 + 1));

\\ Cálculo de E[Q]

for(j = 1, 3, c = 2^(j - 1), for(n = 1; 20, q[n] = (real(qa[index1]) + 1 + qb[index1])); index1 = index1 + 1);

draw([0,0,0,1,0,0,2,0,0])

```



Agradecimentos

Após a conclusão deste trabalho não posso deixar de agradecer a todas as pessoas que para ele contribuíram, referindo de um modo especial:

- O Prof. Doutor Eurico Carrapatoso pelo apoio e disponibilidade manifestada durante o desenvolvimento deste trabalho de investigação;
- O Instituto de Engenharia de Sistemas e Computadores (INESC) pelas facilidades concedidas durante a sua realização;
- A Junta Nacional de Investigação Científica e Tecnológica (JNICT) pelo apoio financeiro concedido através da atribuição de uma bolsa de estudo;
- O Centro de CAD pela colaboração no desenvolvimento dos circuitos impressos;
- Os elementos que realizaram o Sistema Gerador e Detector de Tráfego (SGDT).

Finalmente desejo agradecer a todos os Familiares e Amigos pelo apoio e amizade que me dedicaram.



FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

BIBLIOTECA



000034580