Universidade Federal do Paraná

Departamento de Informática

Roberto A Hexsel

Redes de Dados: Tecnologia e Programação

Relatório Técnico
RT-DINF 005/2001

Curitiba, PR
2001

Conteúdo

1 Introdução **1**
- 1.1 Comunicação de Dados . . . 2
- 1.2 Modelo Conceitual Estratificado OSI/ISO . . . 8
- 1.3 Redes TCP/IP e a Internet . . . 12
- 1.4 Entidades Normatizadoras . . . 13
- 1.5 Bibliografia e Exercícios . . . 14

2 Transmissão de Sinais **15**
- 2.1 Meios Físicos . . . 16
- 2.2 Os Mundos Digital e Analógico . . . 17
  - 2.2.1 Sinais Analógicos . . . 19
  - 2.2.2 Séries de Fourier . . . 19
  - 2.2.3 Espectro de Freqüência . . . 22
  - 2.2.4 Filtros . . . 25
- 2.3 Atenuação, Distorção e Ruído . . . 27
  - 2.3.1 Condição para Transmissão sem Distorção . . . 28
  - 2.3.2 Atenuação versus Distância . . . 29
  - 2.3.3 Atenuação versus Freqüência . . . 30
  - 2.3.4 Distorção de Fase . . . 31
  - 2.3.5 Interferência entre Símbolos . . . 32
  - 2.3.6 Ruído . . . 33
  - 2.3.7 Capacidade de Canal . . . 34
- 2.4 Transmissão na Banda Básica . . . 36
  - 2.4.1 Códigos para Transmissão em Banda Básica . . . 37
  - 2.4.2 Espectro de Freqüência dos Códigos . . . 39
  - 2.4.3 Arquitetura dos Modems Digitais . . . 40
- 2.5 Transmissão por Sinais Modulados . . . 41
  - 2.5.1 Arquitetura dos Modems Analógicos . . . 41
  - 2.5.2 Técnicas de Modulação . . . 42
- 2.6 Recomendações CCITT Série V . . . 49
- 2.7 Bibliografia e Exercícios . . . 50

3 Nível Físico **52**
- 3.1 Equipamentos . . . 52
  - 3.1.1 Terminais . . . 52
  - 3.1.2 Controladores de terminais . . . 53
  - 3.1.3 Multiplexação estatística . . . 55
  - 3.1.4 Multiplexação por divisão de freqüência . . . 57
- 3.2 Transmissão Serial Assíncrona . . . 58
- 3.3 Interface ECD-ETD . . . 60

3.4 Interface EIA-232-D . . . 62
3.4.1 Interfaces Mecânica, Elétrica e Funcional . . . 63
3.4.2 Intertravamento entre ETD e ECD . . . 66
3.4.3 Conexão sem modem . . . 67
3.4.4 Diagnóstico e Testes na Interface EIA-232 . . . 69
3.4.5 Interfaces EIA-530, EIA-423-A, EIA-422-A . . . 70
3.5 Controlador de Interface Serial Assíncrona . . . 70
3.5.1 Descrição Funcional . . . 71
3.5.2 Programação . . . 74
3.6 Bibliografia e Exercícios . . . 77

4 Nível de Enlace 79
4.1 Detecção de Erros . . . 79
4.1.1 Teste de Paridade Vertical . . . 80
4.1.2 Teste de Paridade Longitudinal . . . 80
4.1.3 Teste da Divisão Polinomial . . . 81
4.2 Controle de Enlace . . . 83
4.2.1 Tipos de Linhas . . . 85
4.2.2 Disciplina de Linha . . . 85
4.2.3 Controle de Fluxo . . . 87
4.2.4 Recuperação de Erros . . . 91
4.3 Protocolos de Controle de Enlace . . . 95
4.3.1 Protocolo Kermit . . . 96
4.3.2 Protocolo BSC . . . 99
4.3.3 Protocolo HDLC . . . 103
4.3.4 A Família HDLC . . . 112
4.3.5 Padrão CCITT V.42 – LAPM . . . 112
4.3.6 Compressão de Dados – V.42bis . . . 113
4.4 Controlador de Interface Serial Síncrona . . . 114
4.4.1 Transmissão Serial Síncrona . . . 115
4.4.2 Descrição Funcional . . . 116
4.4.3 Programação . . . 121
4.5 Bibliografia e Exercícios . . . 126

5 Redes Locais 130
5.1 Nível Físico . . . 131
5.1.1 Topologias . . . 131
5.2 Nível de Enlace . . . 133
5.2.1 Controle de Acesso ao Meio . . . 134
5.2.2 Controle de Enlace Lógico . . . 144
5.3 Os Padrões IEEE 802 para Redes Locais . . . 149
5.4 Ethernet . . . 150
5.4.1 Arquitetura . . . 150
5.4.2 Nível Físico . . . 151
5.4.3 Nível de Enlace – Controle de Acesso ao Meio . . . 157
5.4.4 Nível de Enlace – Controle de Enlace Lógico . . . 161
5.4.5 Limites no Diâmetro de uma Rede Ethernet . . . 162
5.4.6 Comutadores Ethernet . . . 162
5.5 Fiber Distributed Data Interface . . . 164
5.5.1 Arquitetura . . . 166

5.5.2 Nível Físico . . . 167
5.5.3 Nível de Enlace – Controle de Acesso ao Meio . . . 168
5.5.4 Nível de Enlace – Controle de Enlace Lógico . . . 170
5.5.5 Topologia, Estações e Concentradores . . . 171
5.5.6 FDDI-II . . . 175
5.6 Scalable Coherent Interface . . . 175
5.6.1 Topologias . . . 176
5.6.2 Caches em Multiprocessadores . . . 179
5.6.3 Arquitetura . . . 181
5.6.4 Nível Físico . . . 182
5.6.5 Nível de Enlace – Controle de Acesso ao Meio . . . 183
5.6.6 Nível de Enlace – Controle de Enlace Lógico . . . 186
5.7 Bibliografia e Exercícios . . . 188

6 A Interface Rede–Computador 190
6.1 Vazão e Latência . . . 190
6.2 Que Interfaces? . . . 191
6.3 Interação entre Caches e E/S . . . 193
6.3.1 Entrada e Saída Programada . . . 196
6.3.2 Acesso Direto à Memória . . . 197
6.3.3 Interrupções . . . 200
6.4 Interações entre Memória Virtual e E/S . . . 201
6.4.1 Gerenciamento de Memória . . . 202
6.4.2 Custos Associados ao Sistema de Memória . . . 203
6.4.3 Eliminação de Cópias Através do Sistema de Memória Virtual . . . 207
6.5 Interações com o Sistema Operacional . . . 210
6.6 Bibliografia e Exercícios . . . 211

Capítulo 1

Introdução

Os sistemas de comunicação de dados, sejam em formato eletrônico, sejam em outros formatos, são omnipresentes em nossas vidas. Estamos tão habituados a eles que os usamos de forma automática sem pensar na complexidade e sofisticação dos mecanismos que mantém informações fluindo ao nosso redor.

Este capítulo apresenta algumas noções e conceitos básicos sobre sistemas eletrônicos de comunicação de dados. Estas noções serão desenvolvidas nos capítulos seguintes, onde os componentes dos sistemas de comunicação de dados são discutidos em mais detalhe. Dois exemplos são apresentados a seguir. Eles descrevem meios de transporte de dados que possuem muitas das características comuns nas redes de dados que serão apresentadas nos próximos capítulos.

Exemplo 1: Correio Terrestre ou Aéreo A Empresa Brasileira de Correios e Telégrafos (EBCT) oferece um conjunto de serviços de transporte de documentos e objetos. Um usuário da EBCT pode escolher a modalidade de serviço em função da urgência na entrega ou do custo do serviço. Uma carta simples tem um custo de entrega baixo mas o prazo de entrega pode chegar a vários dias. Além disso, existe a possibilidade de extravio da carta. Uma carta enviada pelo serviço de encomendas expressas tem um custo várias vezes maior mas o prazo é garantido dentro de limites rígidos e há uma probabilidade mínima de extravio.

A infra-estrutura de um serviço de entrega postal consiste de agências que funcionam como centros de coleta e distribuição de cartas e objetos. O usuário entrega suas cartas, devidamente endereçadas, na agência. Em função da localização do destinatário, uma carta pode ser transferida para o centro de distribuição local ou remetida para um centro de distribuição de outra cidade. Neste último caso, todas as cartas que se destinam àquela cidade são remetidas juntas e transportadas por um caminhão ou avião. Na cidade do destinatário, a carta é novamente transferida para um centro de distribuição, de onde será entregue por um carteiro.

Exemplo 2: Caixa Eletrônico Considere um sistema bancário de atendimento a clientes através de caixas eletrônicos. Cada caixa eletrônico, chamado de *terminal*, é ligado ao computador do banco através de uma linha de comunicação. Quando uma pessoa solicita o saldo de sua conta, o terminal envia uma mensagem ao computador com os dados do cliente e o pedido de saldo. Após verificar a veracidade dos dados fornecidos pelo cliente, o computador consulta seu banco de dados e envia ao terminal uma mensagem que contém o saldo solicitado.

Cada terminal é univocamente identificado pelo seu *endereço*, que neste caso, poderia ser um número codificado em 16 bits. As mensagens trocadas entre os terminais e o computador possuem um formato bem definido e que consiste dos endereços de remetente (terminal) e destinatário (computador central), de um campo que indica se a mensagem contém uma consulta (e seu tipo) ou se contém uma resposta, além de um certo número de bits ou caracteres no *campo*

de dados da mensagem. O conteúdo do campo de dados contém os dados do cliente e eventuais respostas do computador central.

Redes de Computadores Num sistema de transporte eletrônico de informações, o conjunto dos computadores e canais físicos que compõem o sistema é chamado genericamente de *rede de computadores*. Numa rede, um bloco de dados é chamado de *mensagem* e é transportado de *remetente* para *destinatário*. Cada ponto de acesso aos serviços propiciados pela rede possui um *endereço* que identifica cada computador que pode emitir ou receber mensagens.

Uma rede de computadores é organizada de maneira similar à descrita nos exemplos acima. Cada computador tem acesso à rede por um canal físico por onde circulam apenas as mensagens remetidas e destinadas aos usuários daquela máquina. Por razões de economia, este canal tem capacidade suficiente para atender somente ao tráfego de mensagens gerado por uma máquina. A outra extremidade do canal físico é ligada a um computador chamado de *roteador* e que distribui as mensagens que recebe em função dos endereços de seus destinatários. Os roteadores são interligados por canais de maior capacidade porque estes transportam o tráfego de muitas máquinas. A Figura 1.1 mostra uma rede interligando alguns dos departamentos de Informática e Ciência da Computação no Brasil. A rede é representada pela "nuvem" e as caixas dentro da nuvem são os roteadores. Na figura, a espessura das linhas é proporcional a sua capacidade.

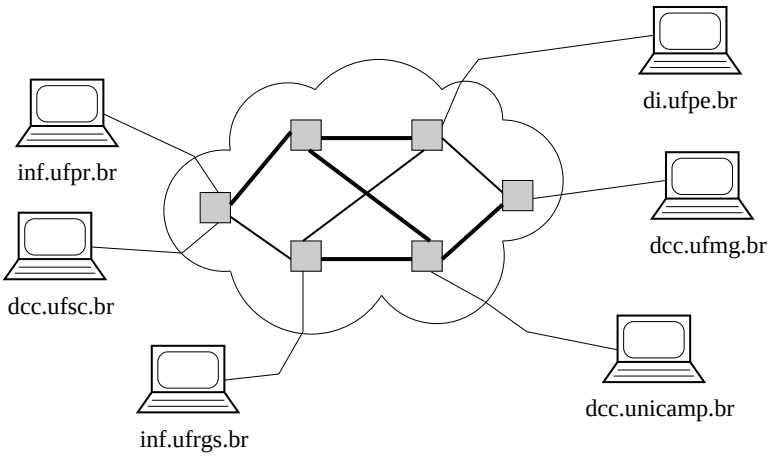


Figura 1.1: Rede interligando Departamentos de Informática e Ciência da Computação.

Da mesma forma que os correios, os serviços de transporte eletrônico de informação podem ser de alto ou baixo custo, dependendo das tecnologias empregadas na construção da rede. Fatores que influem no custo do sistema são a sua *capacidade*, medida pela quantidade de dados transferida por unidade de tempo, e o *tempo de entrega* de cada mensagem individual. A influência destes e outros fatores no projeto e implementação de redes de dados são discutidos no que segue.

1.1 Comunicação de Dados

A função de uma rede de dados é permitir a troca de informação entre dois ou mais equipamentos. Do ponto de vista de equipamentos, uma rede de dados consiste de, no mínimo, dois computadores e algum meio que permita a conexão entre eles. Do ponto de vista de programas, esta rede mínima inclui os programas controladores dos periféricos e um programa de comunicação que gerencie os periféricos e outros recursos da máquina de acordo com os desejos dos operadores. Redes mais complexas empregam técnicas diversas para interligar desde umas pou-

cas máquinas numa mesma sala (*e.g.* uma rede local) até milhões de máquinas espalhadas pelo planeta (*e.g.* a rede Internet).

Existem dois tipos básicos de equipamentos capazes de gerar e consumir dados numa rede: *computadores* e *terminais*. A diferenciação entre um computador e um terminal não é muito simples. Mesmo os terminais primitivos como máquinas de ponto de venda possuem um microprocessador que executa uma quantidade não desprezível de código. No contexto de redes de dados, esta diferenciação não é crucial. Nos casos em que ela é importante, o contexto evidencia quais os papeis que cabem a terminais e a computadores. A fim de evitar confusões, o termo técnico para descrever qualquer equipamento capaz de produzir ou consumir os dados presentes em uma rede é chamado de *equipamento terminal de dados*, abreviado por ETD. Se algum tipo de adaptação de sinais elétricos entre um ETD e o meio físico que transporta os dados é necessária, um *equipamento de comunicação de dados*, abreviado ECD, faz as conversões necessárias. Um ECD também pode ser chamado de *transceptor* porque tem as funções de transmissor e receptor. A combinação ETD+ECD é chamada de *estação*. Um *enlace* de dados corresponde a duas ou mais estações interligadas por um meio físico de comunicação.

Protocolos

Existem certas regras a ser obedecidas no relacionamento entre pessoas, por exemplo, quando se visita alguém essa pessoa é cumprimentada, pergunta-se como passa a família, para só então se entrar no assunto que motiva o encontro. O mesmo ocorre com correspondência escrita: cartas iniciam com a data e o local, nome e endereço do destinatário, seguidos de uma saudação, seguida do texto propriamente dito e finalizadas com o nome do remetente.

O estabelecimento de comunicação entre computadores implica em que as máquinas envolvidas sigam regras bem definidas de modo a que estas cooperem na transferência de informação. Um *protocolo* define os procedimentos necessários para o estabelecimento de comunicação entre dois ou mais sistemas. Protocolos consistem de um conjunto de regras que determinam *o quê* cada participante deve fazer e *quando* deve ser feito. Os procedimentos devem ser claramente descritos na definição do protocolo, de forma a evitar possíveis ambiguidades, assim permitindo que implementações diferentes de um mesmo protocolo possam interoperar de acordo com as regras nele definidas. A interoperabilidade é extremamente importante pois permite que computadores de vários fabricantes, usando sistemas operacionais e hardware distintos, cooperem no transporte de informação entre remetentes e destinatários.

Em última análise, a comunicação entre fonte e destinatário se dá através da cópia pelo destinatário dos bits emitidos pela fonte. Os protocolos de comunicação contém *regras de acordo*, que definem o formato de caracteres e dos endereços de máquinas, por exemplo. Os protocolos ainda contém *regras de implementação*, necessárias para relacionar elementos mais complexos da comunicação com os elementos de baixo nível que permitem a comunicação. Por exemplo, cada caracter do alfabeto latino é relacionado a uma seqüência particular de 8 bits que o representa.

Topologia

Os componentes de uma rede de comunicação de dados podem ser interconectados de várias maneiras. Estas formas de conexão são chamadas de *topologias*. A mais simples delas, mostrada na Figura 1.2, mostra dois computadores (ETDs) interconectados por uma única linha, numa ligação chamada de *ponto-a-ponto*. Os computadores de uma rede são chamados de *nós* ou *nodos* da rede. Uma forma de se conectar vários ETDs entre si consiste em escolher um deles para ser o nó central e todos os demais a ele se ligam por linhas ponto-a-ponto. Esta topologia é chamada de *estrela*. Na topologia em *anel*, os nós da rede são interligados por linhas ponto-a-ponto. Cada nó é responsável por passar adiante as mensagens endereçadas aos outros nós. Ainda, ligações

ponto-a-ponto podem ser usadas para interconectar nós de forma mais ou menos arbitrária em uma *malha*.

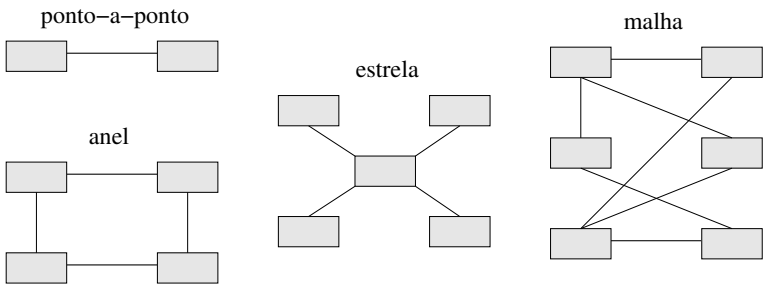


Figura 1.2: Topologias com enlaces ponto-a-ponto.

Ao invés de ligações ponto-a-ponto, se for empregado um meio físico que permita a *difusão* do sinal gerado por um transceptor para todos os outros ligados ao meio de difusão, as três topologias mostradas na Figura 1.3 podem ser usadas. A primeira delas é chamada de *barramento* porque os nodos se conectam a uma mesma "barra" ou cabo. Quando um nodo transmite uma mensagem, aquela é recebida por todos os outros nodos. Neste caso, os ETDs ligados ao barramento devem cooperar no acesso ao meio físico, uma vez que este é compartilhado entre todos eles. As outras duas topologias são baseadas em transmissão de ondas de rádio. Num *rádio enlace* ou enlace por *rádio difusão*, ao invés dos nodos injetarem sinais num cabo, aqueles transmitem as mensagens através da atmosfera via sinais de rádio adequados. Num *enlace via satélite*, cada nodo envia suas mensagens para o satélite que as retransmite para os outros nodos.

Quando um cabo ou barramento é usado para a transmissão de sinais, a transmissão é dita *guiada* porque os sinais elétricos ficam confinados aos limites físicos do cabo. Quando a difusão ocorre através do ar ou vácuo, a transmissão é dita *livre* porque os sinais se propagam livremente pelo espaço.

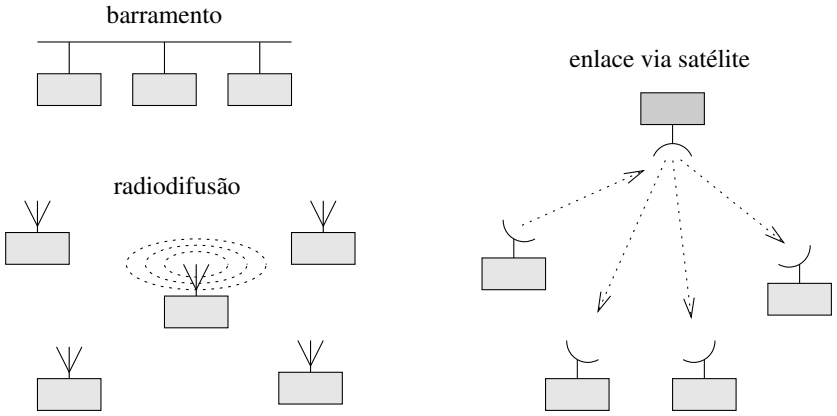


Figura 1.3: Topologias baseadas em difusão.

Códigos, Alfabetos e Símbolos

Dentro de um computador, todo o tipo de informação é representado por bits, bytes e palavras de 16 ou 32 bits. Quando computadores se comunicam, um *código* especial deve ser empregado para representar a informação a ser transferida entre eles. Cada código de comunicação consiste de um conjunto de palavras compostas por elementos do *alfabeto* do código. Obviamente, transmissor e receptor devem estar de acordo quanto ao código e alfabeto a ser usado para evitar ambiguidade na troca de informações. Um alfabeto é um conjunto de *símbolos* onde cada

símbolo representa uma quantidade mínima de informação. Os símbolos do alfabeto Português são os caracteres 'a' a 'z' e 'A' a 'Z'. A representação hexadecimal dos caracteres 'a' e 'z', num dos códigos comumente empregados, é 0x61 e 0x7a, respectivamente. Os elementos do alfabeto usado na transmissão binária são `1` e `0` e estes podem ser representados por uma grande variedade de símbolos. Por exemplo, +12V representando `0` e −12V representando `1` na interface EIA-232 (Seção 3.4); 0V representando `0` e 5V representando `1` em certas famílias de circuitos digitais.

O código ASCII ("American Standard Code for Information Interchange") é um dos códigos usados em comunicação de dados. Neste código, cada caracter alfa-numérico ou de acentuação é codificado em um número de 7 bits. Por exemplo, o caracter '0' (zero) equivale a 0x30, o caracter 'a' equivale a 0x61. O código ASCII contém ainda vários caracteres de controle usados em transmissão de dados.

A Tabela 1.1 mostra o mapeamento dos caracteres da versão extendida do alfabeto ASCII, que inclui a acentuação do Português, chamada de Código Brasileiro de Intercâmbio de Informações (CBII). No código CBII, os caracteres são codificados em números de 8 bits. Este código também é conhecido como ISO-latin (ISO 8859-1). Os caracteres CBII cujo bit $b_7 = 0$ *i.e.,* a metade esquerda da Tabela 1.1, são aqueles do alfabeto ASCII. A Seção 4.3.2 define o significado de alguns dos caracteres de controle.

	b_{7-4}															
b_{3-0}	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL	DLE	SP	0	@	P	‘	p			NBSP	°	À	Ð	à	ǒ
1	SOH	DC1	!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ
2	STX	DC2	”	2	B	R	b	r			¢	²	Â	Ò	â	ò
3	ETX	DC3	#	3	C	S	c	s			£	³	Ã	Ó	ã	ó
4	EOT	DC4	$	4	D	T	d	t			◇	´	Ä	Ô	ä	ô
5	ENQ	NAK	%	5	E	U	e	u			ψ	µ	Å	Õ	å	õ
6	ACK	SYN	&	6	F	V	f	v			|	π	Æ	Ö	æ	ö
7	BEL	ETB	’	7	G	W	g	w			§	·	Ç		ç	
8	BS	CAN	(	8	H	X	h	x			¨	b	È	Ø	è	ø
9	HT	EM	)	9	I	Y	i	y			©	1	É	Ù	é	ù
a	LF	SUB	*	:	J	Z	j	z				º	Ê	Ú	ê	ú
b	VT	ESC	+	;	K	[	k	{			≪	≫	Ë	Û	ë	û
c	FF	FS	,	<	L	\	l	|		¬	1/4	Ì	Ü	ì	ü
d	CR	GS	–	=	M	]	m	}			–	1/2	Í	Ý	í	ý
e	SO	RS	.	>	N	^	n	~			®	3/4	Î	*P*	î	*p*
f	SI	US	/	?	O	_	o	DEL		‘	—	¿	Ï	*β*	ï	ÿ

Tabela 1.1: Alfabeto CBII (ISO 8859-1).

Comunicação Serial

Devido aos custos e às distâncias envolvidas, é de grande conveniência que se use o menor número possível de fios para transmitir dados entre dois sítios. Assim, muitos dos sistemas de comunicação de dados de longa distância empregam *transmissões seriais*. A representação da informação interna aos computadores, chamada de *paralela*, deve ser convertida para uma representação *serial* onde a posição de cada bit numa palavra na representação paralela (um de N fios) corresponde a exatamente um de N intervalos consecutivos numa representação serial.

As duas representações são mostradas abaixo como sequências no tempo de valores binários.

$$\begin{array}{rl} \textit{paralela} & b_0 b_1 b_2 b_3 b_4 b_5 b_6 b_7(t);\ c_0 c_1 c_2 c_3 c_4 c_5 c_6 c_7(t+1);\ \ldots \\ \textit{serial} & b_0(t);\ b_1(t+1);\ b_2(t+2);\ \ldots\ c_0(t+8);\ c_1(t+9);\ \ldots \end{array}$$

Transmissor e receptor devem concordar sobre qual bit na representação serial corresponde a que bit na representação paralela. Por exemplo, na interface EIA-232, o bit menos significativo, que corresponde a 2^0, é o primeiro a ser emitido pelo transmissor – veja a Seção 3.5.

A *velocidade de transmissão* ou, *taxa de transmissão*, é definida como *o número de bits transmitidos por unidade de tempo* e sua unidade é dada em *bits por segundo* (bps). O intervalo entre dois bits consecutivos (*i.e.,* $b(t)$ e $b(t+1)$) é chamado de t_b ou *intervalo de bit* e é definido como a recíproca da taxa de transmissão.

Taxas de transmissão típicamente usadas em linhas telefônicas variam de 300 bps a 64000 bps. Se outros meios físicos forem empregados, taxas muito mais elevadas podem ser atingidas. Por exemplo, a rede local Ethernet emprega cabo coaxial para atingir a taxa de transmissão de 10×10^6 bps (Seção 5.4). O anel de fibra óptica FDDI transmite a 100×10^6 bps (Seção 5.5). Para taxas de transmissão da ordem de 10^3 bps o termo *kilo* bps (Kbps) é comummente usado; taxas de 10^6 bps correspondem a *mega* bps (Mbps) e taxas de 10^9 bps a *giga* bps (Gbps). Do ponto de vista do tempo de uso, um sistema baseado em linha telefônica, transmitindo a 9600 bps, com 10 bits por caracter, leva de 1 a 2 segundos para transmitir uma tela de 80 colunas por 24 linhas. Neste mesmo sistema, são necessários 40 segundos para transmitir uma imagem tipo facsímile, em preto e branco comprimida, com 4×10^5 bits.

Um conceito associado a velocidade de transmissão é o de capacidade de canal. Um dado meio físico, como uma linha telefônica, possui características físicas que impõem um limite na velociade máxima possível de transmissão. Esse limite é chamado de *capacidade de canal*. A palavra "canal" se refere ao meio físico empregado. Por exemplo, uma linha telefônica doméstica é chamada de "canal de voz" porque suas características de transmissão de sinais são adequadas ao transporte da voz humana. Da mesma forma, o sinal gerado por emissoras de rádio e televisão transmitem em seus próprios canais de rádio frequência. Por exemplo, um par de fios usado em linhas telefônicas pode transportar dados a taxas de até 1 Mbps em distâncias da ordem de 1000 m. Nesta mesma distância, um cabo coaxial usado no transporte de sinais de TV pode chegar a 800 Mbps. Mais detalhes na Seção 2.3.7.

Dois conceitos importantes relacionados à capacidade de canal são *vazão* e *tempo de propagação*. A capacidade de um canal determina o máximo teórico para a taxa de transmissão do canal enquanto que a vazão de um canal é a máxima velocidade de transmissão atingível no canal. A vazão também é determinada pelas características físicas do meio de transmissão mas são os equipamentos nas duas pontas do canal que provocam reduções na velocidade efetiva de transmissão. Os equipamentos são compostos por circuitos eletrônicos e por programas que controlam a operação destes circuitos. A sofisticação dos circuitos eletrônicos nas pontas do cabo também influi na vazão máxima possível. Quanto mais rápidos forem os circuitos de transmissão e recepção, mais estreito pode ser o intervalo de bit e consequentemente, maior a taxa de transmissão. Os programas de controle da comunicação e os aplicativos tem uma grande influência na vazão máxima possível. Mesmo que os equipamentos sejam rápidos, o custo do processamento pode limitar seriamente a vazão disponível aos aplicativos. O Capítulo 6 é devotado a este tema.

O *tempo de propagação* t_p de um canal depende da velocidade de propagação de ondas eletromagnéticas no meio físico (*i.e.,* da velocidade da luz). No vácuo, a velocidade da luz é $c = 3.0 \times 10^8$m/s. Num cabo de cobre a velocidade é aproximadamente $0.70\,c$. Em fibra óptica a velocidade de propagação é aproximadamente $0.65\,c$. Uma aproximação grosseira porém intuitiva é "a luz viaja um palmo por nanosegundo" em meios guiados como cabos e fios.

O tempo de propagação dos dados ao longo do canal de comunicação depende do tempo de viagem de um bit entre as duas extremidades do canal. Para distâncias pequenas, a influência

do tempo de propagação é desprezível. Para distâncias longas, o tempo de propagação pode ser um fator limitante no desempenho de um canal de comunicação. Considere um canal de fibra óptica que interliga duas máquinas a uma distância de 1000 Km. Neste caso, o tempo de propagação é da ordem de 5μs. Se a taxa de transmissão é de 200 Kbps, o tempo de propagação é igual a um intervalo de bit ($t_p = t_b$). Se a taxa de transmissão for 2 Mbps, quando o primeiro bit de uma mensagem chega na outra extremidade do enlace, o transmissor está transmitindo o décimo bit da mensagem. O tempo de propagação na fibra é equivalente ao tempo necessário para transmitir 10 bits ($t_p = 10t_b$). A relação entre tempo de propagação e desempenho será discutida novamente na Seção 4.2.

A *utilização* de um canal de comunicação é, para um dado intervalo de tempo, a razão entre o tempo em que que existem bits trafegando pelo canal e a duração do intervalo. A utilização indica a eficiência no uso do meio de comunicação. Alguns dos fatores que reduzem a utilização de um canal são a complexidade dos protocolos envolvidos e a quantidade e a forma de tratamento dos erros de transmissão.

Enlace de Dados

Um enlace é chamado de *simplex* se o tráfego de bits ocorre somente numa direção *e.g.* do computador para o periférico. Transmissões simplex não são muito úteis porque não há nenhuma realimentação entre transmissor e receptor. Num enlace *semi-duplex* o tráfego acontece nos dois sentidos mas nunca simultaneamente: quando um ETD transmite o outro recebe e vice-versa. Neste tipo de enlace, os protocolos contém meios de *reverter a linha*, invertendo a direção da transmissão. Num enlace *duplex*, o tráfego ocorre nos dois sentidos simultaneamente. Normalmente, somente um par de fios (uma linha) é empregada em comunicação semi-duplex. Na comunicação duplex, dependendo da técnica de transmissão, podem ser necessários quatro fios (dois pares de fios ou duas linhas) para a comunicação simultênea nos dois sentidos.

A transmissão de informação através de sinais elétricos (ou ópticos) envolve a adaptação destes sinais ao meio físico que vai transportar os sinais eletromagnéticos. Esta adaptação implica na transcodificação dos sinais digitais usados internamente aos computadores (*i.e.,* 1s e 0s) para uma forma de sinalização adequada ao meio físico. Se os sinais forem transmitidos através do sistema telefônico, o ECD que faz a transcodificação é chamado de *modem* – contração das palavras MOdulador e DEModulador. A Figura 1.4 mostra dois terminais interconectados por um enlace de dados composto por dois ECDs e uma linha telefônica. Existem várias técnicas para a transmissão de sinais digitais através de sistemas de transporte de voz. Algumas delas são descritas no Capítulo 3.

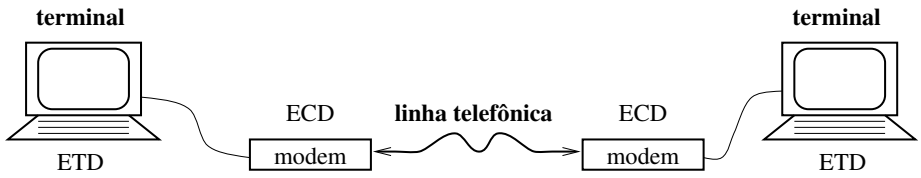


Figura 1.4: Enlace entre dois terminais.

A transmissão entre dois ETDs pode ser *assíncrona*: os ETDs transmitem dados representados por caracteres e não existe nenhuma limitação no intervalo entre o envio de dois caracteres consecutivos (Seção 3.2). Quando a transmissão é *síncrona*, os dados podem ser representados como caracteres individuais ou longas seqüências de bits. Neste caso, dados são transmitidos em blocos que contém vários caracteres ou sequências de bits. A identificação de blocos *i.e.,* onde termina um e onde começa o próximo, é definida pelo protocolo que rege a transmissão síncrona (Seção 4.4.1). Na transmissão assíncrona, a unidade de comunicação é um caracter individual.

Na transmissão síncrona a unidade de comunicação é um bloco de bits ou caracteres, chamado de *quadro*. O protocolo de transmissão deve possibilitar a identificação de quadros consecutivos emitidos pelo transmissor. A identificação só é possível se o *enquadramento* definido pelo protocolo indicar claramente quais são os bits do final de um quadro e os do início do quadro subsequente.

As concessionárias de serviços de telefonia oferecem dois tipos de linha para o transporte de dados. *Linhas privativas* são alugadas às concessionárias e a conexão entre os sítios nas duas pontas da linha é permanente. As tarifas pagas por linhas privativas são constantes e independem de seu uso. O nome técnico deste tipo de linha é "Linha Privativa para Comunicação de Dados" (LPCD). As concessionárias normalmente oferecem LPCDs ditas "normais" ou "condicionadas" sendo que estas últimas sofrem condicionamento elétrico que permite o uso de taxas de transmissão mais altas que em LPCDs normais. *Linhas comutadas*, ou *linhas discadas*, são linhas telefônicas comuns onde a ligação entre dois sítios deve ser estabelecida através da discagem do número do telefone correspondente ao ETD remoto. A discagem pode ser feita manual ou automaticamente pelo ECD.

Redes de Computadores

As redes de computadores são o produto da necessidade de interligação de um grande número de equipamentos. O uso de enlaces ponto-a-ponto deixa de ser prático quando se deseja interligar mais do que uma ou duas dezenas de terminais. A alternativa a um grande número de ligações ponto-a-ponto é o compartilhamento do meio físico por um grande número de terminais. Isso é possível porque o tráfego de dados é altamente variável e inconstante. Por exemplo, quando um programador executa um ciclo de edição, compilação e depuração, suas atividades ao terminal se alternam entre períodos de grande atividade (edição) e pouca atividade (compilação).

As *redes locais* surgiram em função das demandas descritas acima. Em geral, uma rede local pertence a uma única instituição, interliga de uma dezena a um milhar de máquinas e tem uma abrangência geográfica da ordem de uns poucos quilômetros. Uma *rede metropolitana* pode pertencer a uma entidade pública de transporte de dados ou a uma ou mais empresas privadas. Como o nome indica, a abrangência geográfica é da ordem de até 50 Km e estas redes são usadas para interligar as redes locais de várias instituições. Uma *rede dispersa* geralmente abrange um país ou continente e interliga redes metropolitanas e redes locais. A rede Internet é o exemplo mais óbvio de uma rede dispersa, que interliga redes em todo o planeta. Os Capítulos 4.2 e 5 exploram as tecnologias empregadas em redes dispersas e locais, respectivamente.

1.2 Modelo Conceitual Estratificado OSI/ISO

Dada a enorme diversidade dos tipos de sistemas computacionais em uso, algum tipo de acordo deve existir entre os fabricantes para permitir a comunicação entre sistemas que variam de microprocessadores isolados, que custam alguns reais, a supercomputadores que custam milhões de reais. O modelo conceitual amplamente aceito mas raramente implementado é o modelo chamado Open Sistems Interconnect (OSI) [ISO84a], padronizado pela International Standards Organization (ISO). Este modelo define sete níveis de funções a serem desempenhadas pelos sistemas de comunicação. Cada nível oferece ao nível imediatamente superior um certo conjunto de funções, que por sua vez, são implementadas usando serviços oferecidos pelos níveis inferiores. Este é um modelo conceitual adequado para a concepção de sistemas embora não leve necessariamente a implementações eficientes. Na prática, sistemas de comunicação de dados usam um número menor de camadas e/ou alocam algumas das tarefas a camadas diferentes daquelas indicadas pelo modelo OSI/ISO.

A Figura 1.5 mostra um diagrama com a pilha de níveis do modelo OSI/ISO. A pilha da esquerda corresponde a funcionalidade oferecida pelo subsistema de comunicação do computador local e a pilha da direita, aquela do computador remoto. Como indicado pelas setas, quando o computador local envia dados ao computador remoto, a informação flui do Nível de Aplicação passando pelos níveis intermediários até o Nível Físico do computador local. A transmissão de dados só ocorre de fato entre os níveis físicos dos dois computadores. No computador remoto, a informação flui do Nível Físico para o Nível de Aplicação. Cada camada implementa um meio de comunicação um pouco melhor que aquele oferecido pela camada imediatamente inferior. O Nível Físico implementa apenas um *infoduto* que transporta bits entre as duas pontas de um enlace. As camadas superiores atribuem significado aos bits que passam pelo infoduto.

Do ponto de vista lógico, uma *entidade* de Nível de Aplicação do computador local se comunica com sua entidade correspondente no Nível de Aplicação do computador remoto. O programa na máquina local, *e.g.* transferência de arquivos, se comunica com seu equivalante na máquina remota. Os níveis intermediários implementam toda a funcionalidade necessária para que os arquivos sejam transferidos corretamente. O programa de transferência de arquivos é a entidade de Nível de Aplicação e a comunicação lógica ocorre entre a entidade local e a remota. Da mesma forma, uma entidade do Nível de Apresentação no computador local se comunica logicamente com seu par no computador remoto; uma entidade do Nível de Sessão se comunica logicamente com seu par, e assim por diante em todas as camadas. A comunicação física somente acontece entre entidades de Nível Físico. Na Figura 1.5, a comunicação lógica entre entidades de mesmo nível é representada por linha pontilhada enquanto que a comunicação física é representada por uma linha contínua.

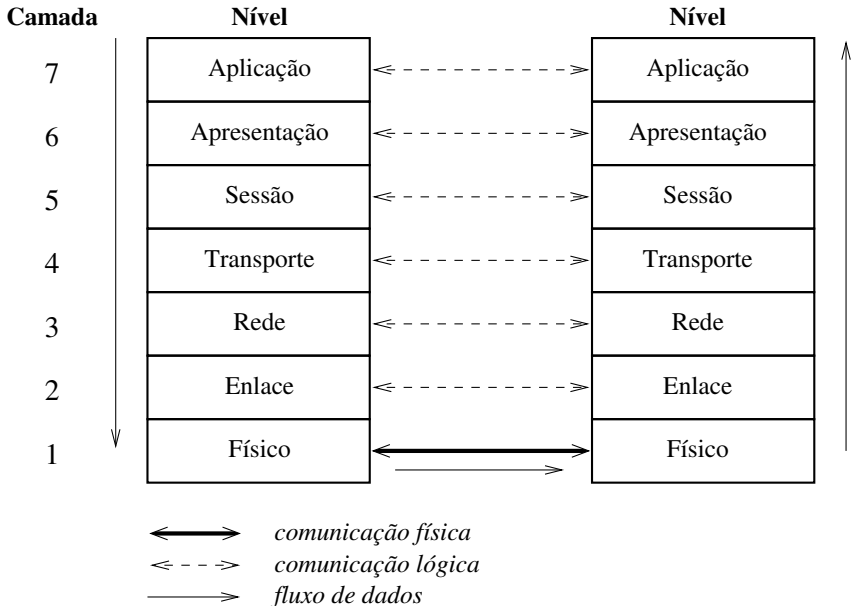


Figura 1.5: Modelo conceitual estratificado OSI/ISO.

Função das Camadas

Do ponto de vista da nomenclatura empregada nos vários níveis do modelo estratificado, os programas de aplicação enviam *mensagens* entre si, como mostrado na Figura 1.6. As entidades dos níveis de Apresentação e Sessão também trocam *mensagens* entre si. As entidades dos Níveis de Transporte e de Rede trocam *pacotes* (ou "datagramas") enquanto que entidades do Nível de Enlace trocam *quadros*. Entidades de Nível Físico se comunicam através de sinais elétricos ou ópticos chamados *símbolos*. A fim de evitar confusões, o termo *Unidade de Dados*

de Protocolo (UDP) ("Protocol Data Unit" – PDU) é empregado com frequência. Existem então UDPs de Aplicação, UDPs de Apresentação, e assim por diante.

O tamanho de uma unidade de dados de protocolo depende do nível onde a UDP é processada e do aplicativo que as produz. Nos três níveis mais altos, associados aos aplicativos, as mensagens tem tamanho variável e determinado pela natureza dos aplicativos. Nos níveis intermediários, responsáveis pelo transporte da informação entre fonte e destinatário, os datagramas, pacotes e quadros tem seu tamanho máximo determinado pelas características do protocolo de cada nível. Estas características dependem dos tipos de mensagens, da taxa de erros, e das capacidades dos meios físicos empregados na rede.

A Figura 1.6 também indica o que ocorre a uma UDP de Aplicação, a medida em que ele é transferida entre os níveis inferiores no computador local, e do Nível Físico até o Nível de Aplicação no computador remoto. Cada entidade envolvida na transmissão da mensagem original acrescenta um *prólogo* à UDP recebida da entidade superior antes de repassá-la à entidade inferior. Alguns níveis também acrescentam um *epílogo* à UDP. Os prólogos e epílogos contém informações de controle necessárias para a movimentação das mensagens através da rede. Na figura, os prólogos adicionados são denotados pela letra p no lado esquerdo de cada UDP. O tipo de informação nos prólogos e epílogos adicionados às UDPs depende do nível onde a adição ocorre. No Nível de Enlace, eles auxiliam no endereçamento das UDPs e na deteção e recuperação de erros ocorridos durante a viagem através do infoduto entre dois nodos adjacentes da rede. No Nível de Rede, o prólogo carrega o endereço lógico das máquinas fonte e destino da UDP. No Nível de Transporte, o epílogo permite a deteção e recuperação de erros ocorridos no percurso entre remetente e destinatário. É importante salientar que os dados transportados pela UDP do nível k consistem de uma UDP do nível $k+1$ *i.e.,* uma UDP de nível $k+1$ viaja *encapsulada* dentro de uma UDP de nível k.

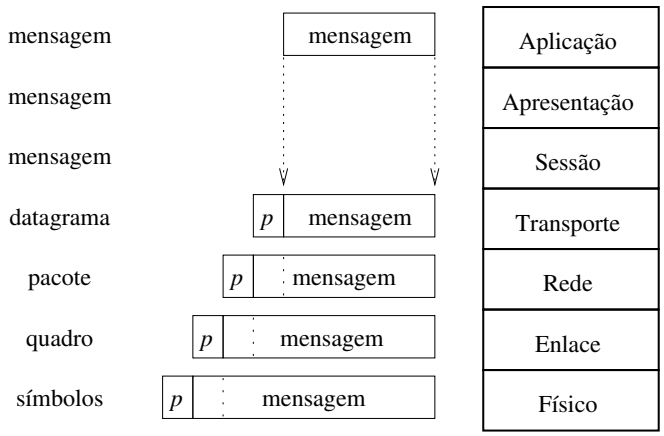


Figura 1.6: Comunicação no modelo conceitual estratificado OSI/ISO.

Nível Físico O principal serviço oferecido pelo Nível Físico é o transporte de bits entre os computadores através de um infoduto (ou "bit pipe"). Para tanto, a representação da informação interna aos computadores (*i.e.,* os sinais elétricos que representam os bits) devem ser transformada de acordo com as características elétricas ou ópticas do meio de transmissão. Dependendo destas características, são necessários equipamentos especiais para fazer a tradução entre a representação da informação interna ao computador e a representação necessária para a transmissão através do meio físico. ECDs são freqüentemente chamados de modems pois na transmissão, um sinal analógico deve ser modulado pelo sinal digital, e na recepção, o sinal analógico deve ser demodulado e re-convertido para a representação digital. O Capítulo 3 discute as razões para o uso de modems e transceptores e as técnicas de modulação empregadas

em transmissão de dados.

Além da tradução de sinais do domínio digital interno aos computadores para o domínio adequado à transmissão pelo meio físico, os protocolos de Nível Físico definem os conectores a ser usados entre os equipamentos, o intertravamento dos sinais das interfaces (*i.e.,* seus comportamentos), velocidades de transmissão, modos e dispositivos de teste, sincronia de envio e recepção, além de outras características. A função dos protocolos de Nível Físico é portanto permitir a interligação lógica e elétrica entre os equipamentos que vão trocar dados.

Nível de Enlace Este nível gerencia uma ligação entre duas máquinas, chamada de *enlace lógico*. Um enlace lógico é estabelecido entre duas máquinas interligadas por um segmento de cabo ou uma única rede local. As entidades deste nível devem ser capazes de estabelecer e terminar o enlace lógico, sincronizar e controlar a transmissão de dados bem como detectar eventuais erros ocorridos durante a transmissão pelo meio físico. Os protocolos de Nível Físico transportam bits entre dois terminais. Devido a influências externas, como ruído elétrico, bits que se encontram em viagem pelo meio físico podem ser distorcidos (*i.e.,* `0` para `1` e vice versa) e isso provoca erros de transmissão. No caso de ocorrência de erros, os protocolos de *Nível de Enlace* devem ser capazes de detectá-los e invocar algum procedimento para recuperar a informação perdida por causa do ruído. Além da detecção e recuperação de erros, o Nível de Enlace oferece serviços como o controle de tráfego entre terminais. O Capítulo 4 traz mais detalhes.

Nível de Rede Esta camada elimina a limitação decorrente de ligações ponto-a-ponto entre ETDs, ou ETDs confinados a uma rede local, permitindo a comunicação entre máquinas em diferentes sub-redes. Nesse contexto, uma sub-rede consiste de um segmento de cabo que interliga algumas máquinas. A função primordial do Nível de Rede é criar a abstração de uma "rede de redes" de forma a permitir a comunicação através de várias sub-redes e esconder a diversidade de tecnologias empregadas em cada sub-rede componente da rede de redes. Esta rede de redes é chamada de "internet" e o exemplo mais conhecido de internet é a rede mundial Internet. As entidades deste nível devem conhecer a topologia da internet e ser capazes de escolher uma rota ou caminho entre quaisquer duas máquinas ligadas à internet.

Nível de Transporte A função desta camada é oferecer o transporte confiável de dados entre duas máquinas em uma internet. Isto significa que as entidades desta camada devem garantir que todos os bits enviados pelo remetente sejam recebidos na ordem correta pelo destinatário. Para tanto, estas entidades devem negociar uma rota e corrigir quaisquer erros que possam ocorrer no transporte dos bits entre remetente e destinatário.

O Nível de Transporte isola os níveis superiores das questões relacionadas ao transporte de dados através da internet. Os níveis superiores suportam serviços associados aos aplicativos enquanto que os níveis de Transporte e inferiores suportam o movimento dos dados gerados e consumidos pelos aplicativos.

Nível de Sessão A função desta camada é gerenciar uma sessão de comunicação entre entidades de apresentação. As funções deste nível incluem o estabelecimento e manutenção da conexão lógica entre máquinas, a sincronização entre as entidades, o controle de fluxo de UDPs e a sinalização de condições anormais ocorridas no transporte de dados. As funções desta camada são subespecificadas no modelo OSI, e em algumas redes são implementadas nos outros níveis.

Nível de Apresentação A função desta camada é traduzir os dados de/para os formatos empregados pelas entidades de Aplicação e de Sessão. Por exemplo, os caracteres empregados

pelos aplicativos podem ser traduzidos do alfabeto ASCII para o EBCDIC (Extended Binary Coded Decimal Interchange Code). Esta camada suporta as abstrações de terminais e arquivos virtuais de forma a esconder os detalhes de implementação de terminais e arquivos em cada um dos muitos tipos de máquinas interligadas pela rede.

Nível de Aplicação Este nível é a interface entre aplicativos e serviço de transporte que é visível ao programador. Os serviços desta camada são normalmente implementados por chamadas de sistema, que são a interface de programação do sistema operacional. Os programas aplicativos executam chamadas de sistema para o estabelecimento de conexões lógicas entre as máquinas local e remota para a transferência de arquivos, por exemplo.

1.3 Redes TCP/IP e a Internet

A rede mundial Internet evoluiu de um projeto do Departamento de Defesa dos Estados Unidos chamado ARPANET, financiado pela Advanced Research Projects Agency (ARPA). A tecnologia Internet foi desenvolvida na mesma época que o modelo OSI/ISO e existem algumas semelhanças entre os dois projetos. O modelo estratificado da Internet é composto por cinco camadas apenas como mostra a Figura 1.7. Nesta figura, o Nível de Rede é denotado pela sigla do Internet Protocol (IP) e o Nível de Transporte pela sigla do Transmission Control Protocol (TCP). As funções dos Níveis Físicos e de Enlace são basicamente as mesmas que no modelo OSI/ISO e devem prover o transporte confiável de informação entre pares de máquinas interligadas pelo meio físico.

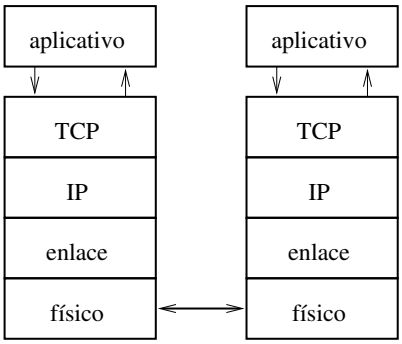


Figura 1.7: Modelo estratificado da Internet.

Nível de Rede – IP O modelo da Internet pressupõe que um certo número de máquinas individuais é interligado por uma rede local, e que pelo menos uma destas máquinas interliga duas ou mais redes locais. O Nível de Rede implementa portanto a abstração de uma *rede de redes*. O protocolo IP define os dois elementos fundamentais desta rede de redes. O primeiro elemento é um método de endereçamento universal que permite identificar cada máquina ligada à rede. O segundo elemento é um conjunto de estratégias de roteamento de mensagens através da rede. Além disso, as máquinas que executam os protocolos IP cooperam na base da boa vontade: se for possível entregar uma mensagem, uma quantidade de esforço razoável é dispendida na tentativa. Se a entrega não for possível, a mensagem é descartada. Estes três elementos, o endereçamento, o roteamento e a cooperação com boa vontade permitem que as implementações do protocolo IP sejam muito simples e eficientes. O protocolo IP é definido pelo padrão IAB 5 [Pos81a] publicado pela Internet Architecture Board.

Nível de Transporte – UDP e TCP As funções de Nível de Transporte na Internet são implementadas por dois protocolos, o User Datagram Protocol (UDP) e o Transmission Control Protocol (TCP). O protocolo UDP oferece transporte não-confiável entre duas máquinas na rede e também opera na base da boa vontade. A funcionalidade que o UDP acrescenta ao Nível de Rede é a capacidade de transportar mensagens oriundas de vários aplicativos diferentes através de um único canal lógico de comunicação. Se, devido a problemas de transmissão, mensagens forem perdidas, é obrigação dos aplicativos recuperar a informação perdida. Para aplicativos que suportem a perda eventual de mensagens, o UDP propicia um meio de transporte eficiente e simples. O protocolo UDP é definido pelo padrão IAB 6 [Pos80].

Se o aplicativo não tolera a perda de mensagens, o protocolo TCP deve ser usado na comunicação. Este protocolo garante a entrega ao destinatário, na ordem correta, de todas as mensagens enviadas pelo remetente. Além de transporte confiável, o TCP contém mecanismos que controlam o fluxo de mensagens entre pares de máquinas, evitando assim que um produtor rápido inunde um consumidor lento com dados. O TCP também contém mecanismos que controlam o tráfego de mensagens através da rede como um todo, evitando que a capacidade agregada da rede de transportar mensagens seja excedida. O protocolo TCP é definido pelo padrão IAB 7 [Pos81b].

Padronização Os padrões que definem a arquitetura e os protocolos empregados na comunicação através da Internet podem ser encontrados no endereço
`http://www.ietf.cnri.reston.va.us`.
Normalmente, as definições dos protocolos são identificadas pelo número do Request for Comments (RFC) que deu origem ao protocolo. Os RFCs são usados pela comunidade de pesquisa e desenvolvimento durante a fase de elaboração do protocolo e uma vez aprovado pela Internet Engineering Task Force e pelo IAB, o RFC passa a ser a definição oficial do protocolo.

1.4 Entidades Normatizadoras

Os protocolos empregados em redes de comunicação de dados definem precisamente as atividades envolvidas na troca de informações entre os nodos de uma rede. Para que os vários fabricantes possam produzir equipamentos e/ou programas que interoperem com os de outras marcas é necessário que todas as partes envolvidas na produção destes sistemas estejam de acordo quanto aos procedimentos descritos nos protocolos. Este acordo normalmente toma a forma de um *padrão* que, além de descrever precisamente o protocolo, define parâmetros de conformidade e limita o escopo de possíveis implementações do protocolo em questão.

O organismo de maior importância no que tange a publicação de padrões internacionais é a International Standards Organisation (ISO). A ISO tem como membros as organizações nacionais responsáveis pela publicação de padrões em cada país. Na área de telecomunicações, a ISO coopera com outras organizações responsáveis pela publicação de padrões. Publicações da ISO podem ser adquiridas de ISO Central Secretariat, Case Postale 56, 1 rue de Verambé, CH-1211, Genève 20, Suiça.

A International Eletrotechnical Commission (IEC) publica padrões, em associação com a ISO, de interesse na área de comunicação de dados e redes de computadores. Suas publicações podem ser adquiridas em IEC Sales Department, Case Postale 131, 3 rue de Verambé, CH-1211, Genève 20, Suiça. `http://www.iso.ch`

A International Telecommunications Union (ITU) estabeleceu uma secção para tratar do setor de telecomunicações, chamado de ITU-T e que subsitituiu o Comitê Consultativo Internacional de Telegrafia e Telefonia (CCITT). O ITU-T congrega os organismos e empresas estatais responsáveis pelo setor de telecomunicações de cada país membro da ONU e publica padrões em praticamente todas as áreas relacionadas com telecomunicações, incluindo telefonia, sistemas de

microondas, rádio e redes de dados. Publicações do ITU-T podem ser obtidas no International Telecommunications Union, General Secretariat, Sales Service, Place de Nation, CH 1211, Geneva 20, Suíça ou, United Nations Bookstore, Room 32B, United Nations General Assembly Building, New York, NY 10017, EUA. `http://www.itu.ch`

O Institute of Electrical and Electronic Engineers (IEEE) e a Electronics Industries Association (EIA) são organizações Norte-Americanas muito ativas na publicação de padrões relacionados a redes de dados. O IEEE é uma associação profissional e tem produzido padrões para redes de dados que terminam por ser adotados por organizações como a ISO e CCITT. Documentos publicados pelo IEEE podem ser obtidos no Institute of Electrical and Electronics Engineers, Inc., Service Center, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, EUA. Publicações da EIA podem ser obtidas de EIA Sales Order Department, Electronic Industries Association, 2001 Eye Street N W, Washington, DC 20006, EUA. `http://www.ieee.org`

1.5 Bibliografia e Exercícios

O modelo estratificado ISO/OSI [ISO84a] é discutido em [Tar86, Tan89, Sta91]. Os níveis intermediários do modelo estratificado da rede Internet [Pos81a] são extensivamente discutidos em [CS94, Com95]. [Com87] contém uma implementação dos protocolos TCP/IP no sistema operacional Xinu. [Tan92] discute o modelo estratificado ISO/OSI do ponto de vista de sistemas distribuídos.

Ex. 1.1 [§ 1] A *latência* de um meio de comunicação é o intervalo que decorre entre o instante do final da emissão de uma mensagem pelo transmissor o instante do final da aceitação da mensagem pelo receptor. A *vazão* de um meio de comunicação pode ser definida como o número de mensagens transportadas por unidade de tempo. Estime a vazão e a latência de um carteiro, no trajeto entre agência de correios e destinatários a várias distâncias. Repita para um avião que transporta cartas entre as maiores capitais do país. Suponha números razoáveis para o peso de uma carta e as velocidades do carteiro e do avião.

Ex. 1.2 [§ 1] O que pode ser deduzido do *produto vazão latência*, isto é, *vazão*×*latência*?

Ex. 1.3 [§ 1] Compare as características dos espaços de endereçamento empregados pelos serviços postais e pelo sistema de telefonia. Como os dois sistemas associam o 'endereço' à localização do 'endereçado'?

Ex. 1.4 [§ 1.1] Compare as topologias de rede, enumerando suas vantagens e desvantagens. (a) Estrela e anel. (b) Anel e malha. (c) Estrela e malha. (d) Estrela e barramento.

Ex. 1.5 [§ 1.1] Codifique a sentença "*a arara arou de araraquara para jabaquara*" nos códigos Morse e CBII. Qual o comprimento, em bits, das duas codificações? Suponha que o separador de caracteres no código Morse equivale a dois bits, e que o caracter 'espaço' equivale a quatro bits.

```
a .-        j .- - -     s ...       1 .- - - - -
b -...      k -.-        t -         2 ..- - -
c -.-.      l .-..       u ..-       3 ...- -
d -..       m - -        v ...-      4 ....-
e .         n -.         w . - -     5 .....
f ..-.      o - - -      x -..-      6 -....
g - -.      p .- -.      y -.- -     7 - -...
h ....      q - -.-      z - -..     8 - - -..
i ..        r .-.        0 - - - - - 9 - - - -.
```

Ex. 1.6 [§ 1.2] Como você implementaria os protocolos de rede do modelo ISO/OSI? Qual seria a arquitetura do software de rede?

Ex. 1.7 [§ 1.2] O que você pode dizer quanto à eficiência do código que implementa, de forma simplista, a pilha de protocolos do modelo ISO/OSI?

Capítulo 2

Transmissão de Sinais

Os dados a transferir entre duas máquinas são, geralmente, seqüências de caracteres ou sequências de bits que transportam a informação. Os computadores se acoplam ao meio físico através de interfaces de nível físico e estas interfaces, por sua vez, são parte de dispositivos periféricos ou dos dispositivos de Entrada/Saída dos computadores. Os programas de controle destes periféricos se encarregam de garantir a transferência dos dados entre periférico e memória em uma ponta do enlace até periférico e memória da máquina na outra ponta do enlace. Aos dispositivos de comunicação (ECDs) cabe traduzir os dados a transmitir do formato interno ao computador –bits e octetos– para um formato adequado à transmissão através do meio físico –sinais digitais ou analógicos.

As características eletromagnéticas de um dado meio físico impõem limites à máxima taxa de transmissão de dados através daquele meio físico. Por exemplo, quanto maior a distância entre os ETDs, e mais longo o cabo que os interliga, maior a atenuação sofrida pelos sinais que se propagam ao longo do cabo. A atenuação dos sinais aumenta o grau de dificuldade em separar o sinal que transporta dados de ruído elétrico captado pelos componentes eletrônicos que compõem o enlace. Isso implica em maior dificuldade para recuperar a informação contida nos sinais amostrados pelo receptor.

A Série de Fourier, introduzida na Seção 2.2, é uma ferramenta de análise de sinais que permite descrever o comportamento de sinais do ponto de vista de seu espectro de freqüências e da atenuação provocada pelas características dos meios de transmissão. As linhas telefônicas se comportam como verdadeiros filtros que permitem a passagem de sinais de voz mas dificultam a passagem de sinais digitais. O espectro de freqüência destes dois tipos de sinais, obtidos com auxílio das Séries de Fourier, explicam os comportamentos diferenciados de linhas telefônicas na transmissão dos dois tipos de sinais.

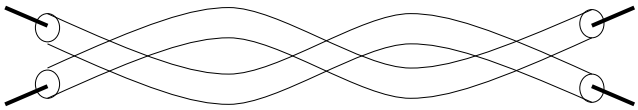
A máxima taxa de transmissão de um meio físico é determinada pelas suas características eletromagnéticas e pela quantidade de ruído elétrico (ou óptico) presente nos circuitos de comunicação. A influência do ruído eletromagnético na taxa máxima de transmissão é investigada na Seção 2.3, que discute os limites teóricos na velocidade de transmissão codificados nos Teoremas de Nyquist e Shannon.

As técnicas de tradução dos sinais digitais usados 'dentro' dos computadores para os sinais apropriados para a comunicação entre computadores são apresentadas nas Seções 2.4 e 2.5. A tradução consiste em modificar a representação da informação por 'bits' ou 'bytes' para uma representação que minimize os efeitos da atenuação e distorção dos sinais provocados pelos meios de comunicação.

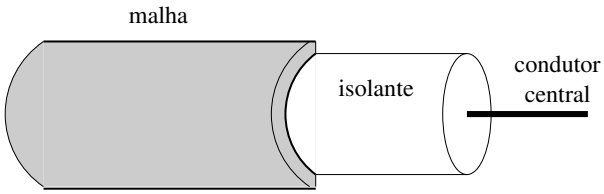
2.1 Meios Físicos

Dos três meios físicos mais empregados em redes de dados, dois transportam sinais elétricos e o terceiro transporta sinais ópticos. O meio físico mais simples e barato consiste de um par de fios trançados ou, *par trançado*. Com custo intermediário, o *cabo coaxial* é usado quando as características eletromagnéticas do par trançado tornam seu uso inapropriado. Finalmente, a *fibra óptica* é usada em redes de alta velocidade e é a mais cara das três alternativas. O preço-por-metro não é uma medida adequada para comparações de custo-benefício. Uma medida de custo mais adquada é o produto *alcance*×*vazão*. O alcance é distância máxima entre duas estações com taxas de erro suficientemente baixas. Na medida em que os sinais se propagam ao longo dos cabos, eles sofrem distorções provocadas pelas características eletromagnéticas do meio, como discutido na Seção 2.3. Praticamente, qualquer meio físico pode transferir dados com altas vazões (10^9 bps) em distâncias de uns poucos metros. Para transmissão em alta velocidade, o par trançado atinge o menor alcance, o cabo coaxial alcance intermediário e a fibra óptica, o maior alcance dos três.

Par trançado Este é o meio físico de menor custo por metro embora seu uso seja limitdo pelo seu reduzido produto *alcance*×*vazão*. Para distâncias de dezenas de metros, o par trançado pode ser usado com taxas de até 10^8 bps. Para distâncias da ordem de 1000 metros, o par trançado pode transmitir com taxas de até 10^6 bps. A Figura 2.1 mostra a configuração dos condutores em um segmento de par trançado e um segmento de cabo coaxial.



(a) Par trançado



(b) Cabo coaxial

Figura 2.1: Par trançado e cabo coaxial.

Cabo coaxial Como o nome sugere, a geometria dos cabos coaxiais é tal que os dois condutores compartilham o mesmo eixo. O condutor central consiste de um fio rígido de cobre. O condutor externo pode ser uma malha de fios de cobre ou uma fina tira de metal enrolada ao longo do isolamento que separa o condutor central da malha. Os dois condutores são separados um do outro por uma camada de material isolante. Cabos coaxiais também são chamados de *cabos blindados* porque sua geometria é tal que o condutor externo forma uma blindagem contra interferências eletromagnéticas. Cabos coaxiais apresentam portanto alta imunidade a ruído eletromagnético proveniente do ambiente onde o cabo se encontra. Isso significa que os sinais que trafegam através do cabo não interferem com o meio, nem sofrem de interferência provocada por sinais em outros condutores próximos. Para transmissão na banda básica (Seção 2.4), cabos coaxiais podem ser usados com taxas da ordem de 10^7 bps. Para transmissão de sinais modulados de alta frequência, cabos coaxiais transportam sinais de TV com espectro que se extende até

os 400 MHz em distâncias de uns poucos quilometros. Para a transmissão de dados nestas condições, taxas de 800 Mbps são teoricamente atingíveis.

Fibra óptica Ao contrário dos cabos elétricos, as fibras ópticas não conduzem eletricidade, apenas pulsos de luz. Isso aumenta em muito a seguraça de enlaces externos porque cabos com fibra óptica são imunes a interferência eletromagnética como relâmpagos, por exemplo. Em certas aplicações, como a interligação de edifícios ou equipamentos numa fábrica, a imunidade a interferências e o isolamento elétrico entre as pontas do enlace, compensam facilmente o custo mais elevado dos sistemas com fibra óptica.

A transmissão de sinais através de fibra óptica apresenta uma vantagem importante sobre a transmissão por cabos ou rádiodifusão: a faixa de passagem de uma fibra é muitíssimo maior que a de cabos ou canais de rádio. Velocidades de transmissão da ordem de 10^9 a 10^{10} bits por segundo são perfeitamente factíveis. Além disso, fibras ópticas apresentam maior alcance (*i.e.,* menor atenuação por quilômetro) e taxas de erros baixíssimas quando comparadas com outros meios. A Figura 2.2 mostra os componentes de um infoduto de fibra óptica. O *fotoemissor* pode ser um diodo emissor de luz (LED) ou um diodo laser. Sua função é converter os sinais elétricos em seus terminais para pulsos de luz. A fibra consiste de uma capa plástica para proteção mecânica (não mostrada), uma *casca* e um *núcleo* transparentes. O *fotodetector* converte os pulsos de luz novamente em sinais elétricos.

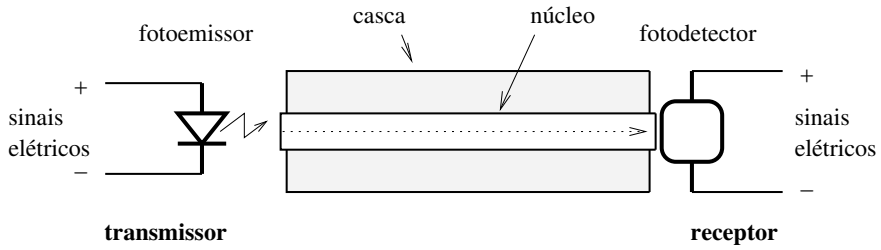


Figura 2.2: Infoduto de fibra óptica.

2.2 Os Mundos Digital e Analógico

A Engenharia de Computação emprega vários níveis de abstração para relacionar o mundo analógico em que vivemos ao mundo digital onde são projetados os computadores. No mundo analógico, os sinais de tensão e corrente variam continuamente com o tempo, assumindo todos os valores intermediários entre quaisquer dois pontos inicial e final. No mundo digital, os sinais variam discretamente entre dois pontos, chamados de *zero* e *um*, `1` e `0`, *Verdadeiro* e *Falso*, etc. A Figura 2.3 mostra a diferença entre sinais que variam com o tempo nos mundos analógico e digital.

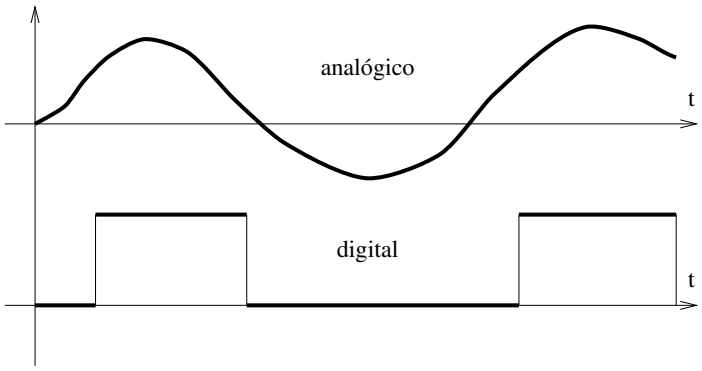


Figura 2.3: Sinais analógicos e digitais.

O projeto de sistemas digitais só se torna possível pelo uso de funções que mapeiam representações de um dado nível de abstração nos níveis imediatamente acima e imediatamente abaixo. Ignorando fenômenos de mecânica quântica, o comportamento de circuitos eletrônicos é modelado por equações diferenciais que descrevem o relacionamento entre as tensões e correntes em um dado circuito. Este nível de descrição do comportamento de circuitos é chamado de *concreto*. O projeto de circuitos digitais aproxima o comportamento digital ao analógico pelo uso de diagramas de tempo e regras de projeto onde as transições entre 0 e 1 ocorrem em um tempo finito e maior que zero. Neste nível de abstração, os circuitos tem um comportamento *idealizado*. No nível de descrição mais *abstrato*, o comportamento de um circuito digital é representado por funções booleanas. A Figura 2.4 mostra a progressão do concreto (sinais contínuos no tempo) para o abstrato (sinais discretos no tempo).

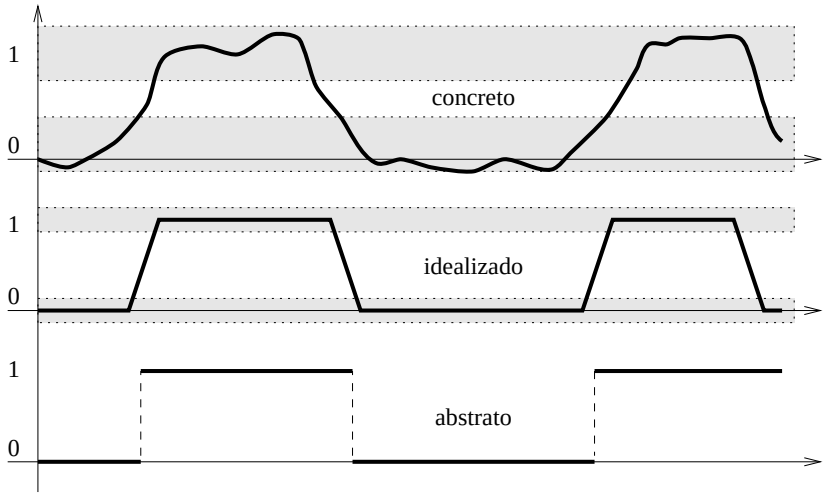


Figura 2.4: Níveis de abstração na representação de sinais.

No projeto de computadores, o comprimento dos caminhos elétricos entre os vários subsistemas são da ordem de 1 cm a 1 metro. Nesta faixa de distâncias, o uso de sinalização digital de alta velocidade é pouco problemático. Na comunicação de dados através de sinais elétricos, as distâncias são tais que a sinalização digital se torna pouco prática. Em especial, as características elétricas das linhas telefônicas são tais que a sinalização digital só é praticável em distâncias de uns poucos quilômetros. Para distâncias maiores, são necessários equipamentos especiais que fazem uma transcodificação entre a sinalização digital interna ao computador e a sinalização analógica necessária para transmissões de longa distância. Estes equipamentos são chamados de *modem*s porque sua função é MODular um sinal analógico em função do sinal digital

que se deseja transmitir e DEMOdular o sinal analógico recebido, assim recuperando a informação digital transmitida pela outra ponta do enlace. Algumas das técnicas de transcodificação são apresentadas nas próximas seções.

2.2.1 Sinais Analógicos

As técnicas de transmissão de dados por sinais analógicos são mais complexas que aquelas baseadas em sinalização digital. Parte da complexidade advém da própria natureza dos sinais analógicos. Ao invés de dois valores discretos (0s e 1s), o projeto de sistemas analógicos de comunicação deve levar em conta uma série de outras grandezas como o espectro de freqüência dos sinais. Mesmo na transmissão digital, essas grandezas devem ser consideradas porque o "digital" é somente uma abstração do mundo analógico. As Séries de Fourier são uma das ferramentas de análise de sistemas analógicos e se baseiam em séries de funções senoidais.

Senóides Uma função $x(t)$ é chamada de *periódica* se

$$x(t) = x(t+T), \quad T < T_0$$

Onde T_0 é o *período* da função $x(t)$ (*e.g.* o intervalo entre duas cristas consecutivas). A *freqüência* f_0 de $x(t)$ é definida como $f_0 = 1/T_0$. *Senóide* é o nome genérico para as funções trigonométricas $\text{sen}(x)$ e $\cos(x)$. A Figura 2.5 mostra as grandezas de interesse na função periódica

$$g(t) = A\,\text{sen}(\omega t + \phi)$$

A *amplitude* A da função é a distância entre o ponto mais negativo ou ponto mais positivo e o eixo horizontal. A *fase* é o deslocamento ϕ no ângulo da função com relação à abscissa zero. A *freqüência angular* ω é definida como $\omega = 2\pi f$. Note que as funções $\text{sen}(x)$ (linha cheia) e $\cos(x)$ (linha pontilhada) são defasadas de 90^o entre si.

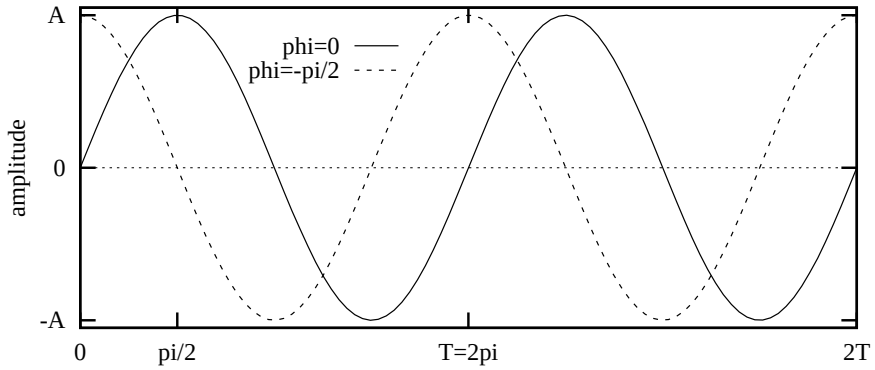


Figura 2.5: Funções $\text{sen}(x)$ e $\cos(x)$.

2.2.2 Séries de Fourier

Qualquer função contínua e periódica no tempo $g(t)$ pode ser descrita pela somatória de uma série de funções senoidais – uma *Série de Fourier*. As freqüências das componentes da série definem o *espectro de freqüência* da função $g(t)$. A largura e a forma da envoltória do espectro determinam a adequação de $g(t)$ para aplicações em comunicação de dados.

A Série de Fourier é definida por

$$g(t) \;=\; C_0 + \sum_{n=1}^{\infty}(a_n \cos\omega_n t + b_n\,\text{sen}\omega_n t), \quad 0 \le t \le T \tag{2.1}$$

A função periódica $g(t)$ pode ser representada por um valor constante adicionado a uma soma infinita de termos com freqüência cada vez mais altas. Os termos da série são:

$$C_0 = 1/T \int_0^T g(t)dt \tag{2.2}$$

$$a_n = 2/T \int_0^T g(t)\cos\omega_n t dt \tag{2.3}$$

$$b_n = 2/T \int_0^T g(t)\,\mathrm{sen}\omega_n t dt \tag{2.4}$$

$$\omega_n = 2\pi n f \tag{2.5}$$

O termo constante C_0 é o valor médio da função g em um período T, isto é, a média de sua amplitude em um período. Note que funções pares são representadas por séries em cosenos (porque todos os $b_n = 0$) enquanto que funções ímpares são representadas por séries em senos (porque todos os $a_n = 0$), as demais funções por séries que tem componentes representados por senos e cosenos.

Onda quadrada Como exemplo, considere seqüência binária `010101`$\cdots$. Ela pode ser representada por um sinal elétrico onde `0`s correspondem a -1Volt e `1`s correspondem a $+1$Volt. Se transmitido por uma linha serial, este *trem de pulsos* é descrito pela função $g(t)$, mostrada na Figura 2.6

$$g(t) = +1, \qquad 0 < t \leq T/4 \tag{2.6}$$

$$= -1, \qquad T/4 < t \leq 3T/4 \tag{2.7}$$

$$= +1, \qquad 3T/4 < t \leq T \tag{2.8}$$

Resolvendo-se as integrais para as componentes senoidais de $g(t)$ da Equação 2.1,

$$C_0 = 0 \tag{2.9}$$

$$a_n = 0, \qquad \text{para n par} \tag{2.10}$$

$$= (-1)^{(n-1)/2}\, 4/n\pi, \qquad \text{para n ímpar} \tag{2.11}$$

$$b_n = 0 \tag{2.12}$$

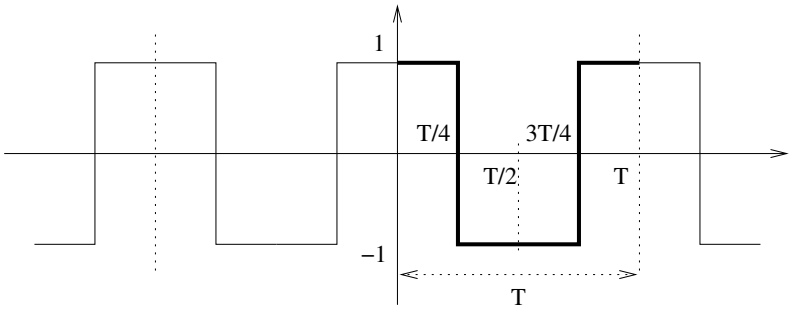


Figura 2.6: Onda quadrada expressa por $g(t)$.

A função $g(t)$ pode portanto ser representada pela Série de Fourier

$$g(t) = 4/\pi(\quad \cos\omega\, t - 1/3\cos 3\omega t + 1/5\cos 5\omega t \tag{2.13}$$
$$-1/7\cos 7\omega t + 1/9\cos 9\omega t \quad \ldots \quad)$$

A função $g(t)$ tem valor médio zero ($C_0 = 0$) já que ela é simétrica com relação ao eixo horizontal no intervalo $[0, T]$. Quando $n = 1$ a componente da série é chamada de componente *fundamental* já que ela tem a mesma freqüência de $g(t)$. A n-ésima componente é chamada de *n-ésima harmônica*. A fundamental tem amplitude $4/\pi$ e a amplitude da n-ésima harmônica é $4/n\pi$. A fase das harmônicas alterna-se entre 0^o e 180^o. Dependendo de n, a amplitude da n-ésima harmônica é multiplicada por $+1 \rightarrow \phi = 0^o$, ou $-1 \rightarrow \phi = 180^o$.

A Figura 2.7 mostra a fundamental, terceira, quinta e sétima harmônicas de $g(t)$ e a soma das suas amplitudes. A diferença de fase entre as harmônicas faz com que a sua soma "se pareça" mais com a forma de uma onda quadrada a cada harmônica que é adicionada. De cima para baixo: fundamental f e terceira harmônica ($3f$), fundamental+terceira e quinta harmônica ($5f$), fundamental+terceira+quinta e sétima harmônica ($7f$), fundamental+terceira+quinta+sétima e nona harmônica ($9f$). Note que na medida em que harmônicas de ordem mais elevada são adicionadas a onda fica com as bordas mais pronunciadas.

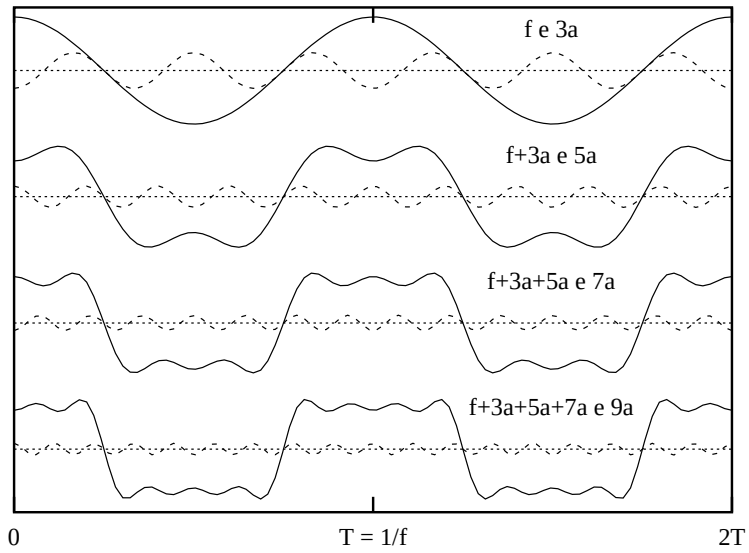


Figura 2.7: Conteúdo harmônico de $g(t)$.

Trem de pulsos Suponha um trem de pulsos periódico com período T onde um em cada b bits é 1, e os demais bits são 0. A Figura 2.8 mostra um trem de pulsos com período T e intervalo de duração de bit de p segundos. A Série de Fourier $h(t)$ que representa este trem de pulsos é dada por

$$\begin{aligned} h(t) \;=\; R[1 + 2\,(\;& (1/\pi R)\,\mathrm{sen}(\pi R)\cdot\cos(\omega t) \\ & +(1/2\pi R)\,\mathrm{sen}(2\pi R)\cdot\cos(2\omega t) \\ & +(1/3\pi R)\,\mathrm{sen}(3\pi R)\cdot\cos(3\omega t) \\ & +\ldots)], \qquad\qquad R = p/T \end{aligned} \tag{2.14}$$

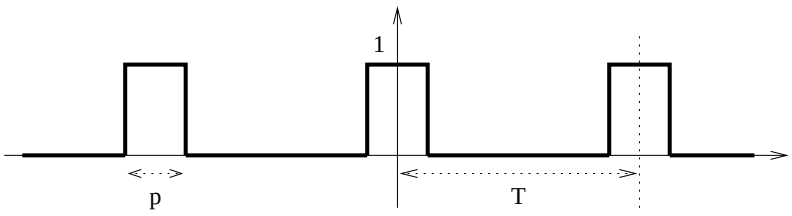


Figura 2.8: Trem de pulsos expresso por $h(t)$.

A Figura 2.9 mostra o conteúdo harmônico do sinal $h(t)$ quando $b = 4$, *i.e.,* o trem de pulsos $1000100010\cdots$. A figura mostra, de cima para baixo, a componente fundamental (f), a fundamental e a segunda harmônica (f+2), a fundamental, segunda e terceira harmônicas (f+2+3), e a fundamental, segunda e terceira e quarta harmônicas (f+2+3+4). Da mesma forma que na Figura 2.7, na medida em que harmônicas de ordem mais elevadas são adicionadas, o sinal resultante se torna "mais quadrado".

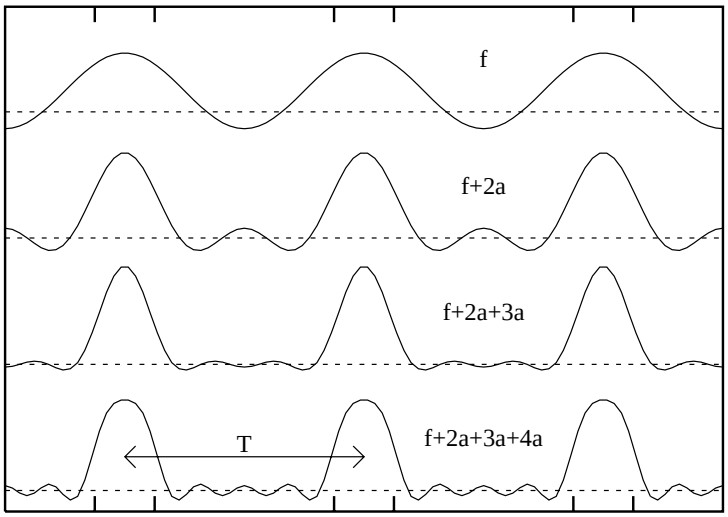


Figura 2.9: Harmônicos de $h(t)$.

2.2.3 Espectro de Freqüência

Há um relacionamento importante entre a forma de onda de uma função $g(t)$ – sua representação no *domínio tempo* – e seu espectro de freqüências – sua representação no *domínio freqüência*. A Série de Fourier nos permite trabalhar com sinais no domínio que seja mais conveniente. O espectro de freqüências de $g(t)$ é mostrado na Figura 2.10.

$$g(t) = 4/\pi(\quad \cos\omega\, t - 1/3\cos 3\omega t + 1/5\cos 5\omega t \\ -1/7\cos 7\omega t + 1/9\cos 9\omega t \quad \ldots \quad)$$

No eixo horizontal estão marcadas as harmônicas e no eixo vertical suas amplitudes. A *potência* P_n da harmônica n é $P_n = \sqrt{a_n^2 + b_n^2}$ onde a_n e b_n são os coeficientes da Série de Fourier correspondentes a harmônica n (Equações 2.3 e 2.4). Note que a maior parte da potência de uma onda quadrada é transferida pela fundamental porque a amplitude da fundamental é três vezes a da 3ª harmônica. A *envoltória* do espectro de freqüência da função $g(t)$ é mostrada superposta ao gráfico das amplitudes.

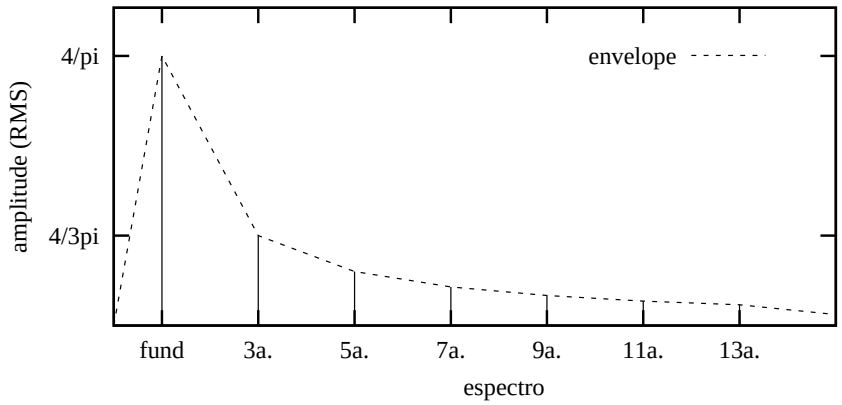
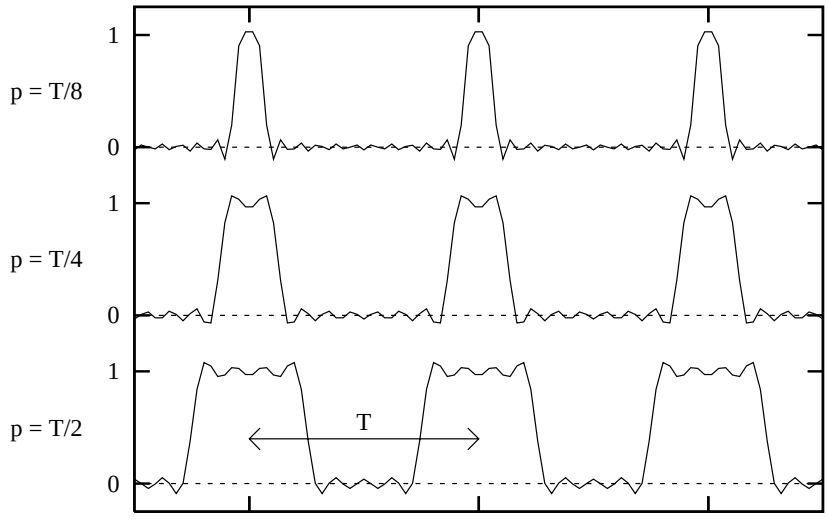


Figura 2.10: Espectro de freqüência de $g(t)$.

Como exemplo, considere o espectro de freqüências do trem de pulsos, quando a largura do pulso varia entre $T/2$ e $T/8$. Os gráficos na Figura 2.11 foram gerados com a Equação 2.14, usando as 16 primeiras harmônicas. O espectro dos trens de pulsos mostra que, para a onda quadrada ($p = T/2$), a maior parte da energia contida na onda está nas primeiras 3 harmônicas (fundamental, 3ª e 5ª harmônicas). No caso dos pulsos estreitos ($p = T/8$), as 7 primeiras harmônicas transportam a maior parte da energia da onda. Retornaremos a este ponto na Seção 2.3.7.

Outro ponto importante mostrado na Figura 2.11 é a existência da componente de freqüência zero nos trens de pulsos, mostrada como *dc* na figura. Esta componente é chamada de *nível DC do sinal* porque corresponde a um deslocamento do sinal com relação à referência de amplitude do sinal. O *DC* vem do Inglês "direct current" e significa corrente contínua. Compare com a onda quadrada das Figuras 2.7 e 2.10 que é simétrica com relação é referência vertical. Os sinais elétricos presentes dentro dos computadores possuem nível médio de DC diferente de zero porque isso simplifica a implementação de alguns tipos de circuitos integrados e das fontes de alimentação. Por outro lado, os sistemas empregados em telefonia podem ser danificados se os sinais nos circuitos de voz possuem uma componente DC. Por isso, os sinais empregados em transmissão de dados através de circuitos telefônicos não podem conter níveis DC devendo portanto ser simétricos com relação à tensão de referência.



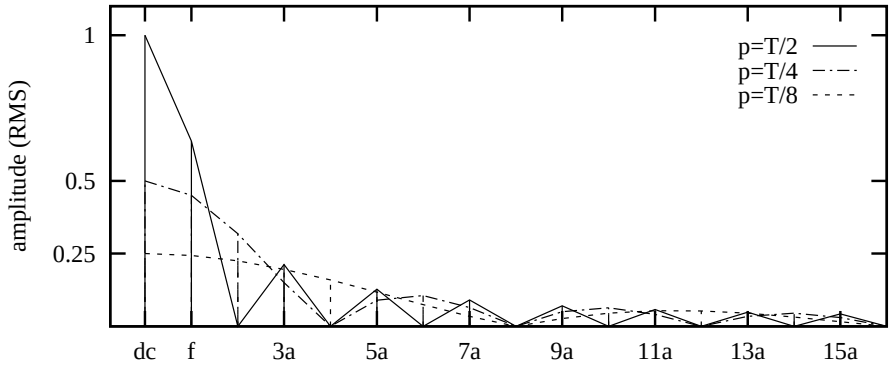


Figura 2.11: Trens de pulsos gerados por $h(t)$.

Amplificadores de áudio Um amplificador de áudio de alta-fidelidade é um dispositivo que amplifica os sinais na sua entrada até níveis de potência adequados para acionar os alto-falantes. Para que um aparelho de áudio se qualifique como alta-fidelidade, sua *resposta em freqüência* deve ser plana na faixa de 20 a 20000 Hz *i.e.,* o aparelho deve ou atenuar ou amplificar todas as freqüências do espectro audível uniformemente. Essa resposta ideal é mostrada na Figura 2.12 pela linha contínua (envoltória). O eixo horizontal é o das freqüências e o vertical o das amplitudes, normalmente em valores absolutos. A maioria dos amplificadores comerciais permite algum controle na resposta em freqüência através dos controles de "graves" e "agudos". A figura também mostra o efeito desses controles na resposta do amplificador (linhas pontilhadas). Estes controles são chamados de *filtros* porque alteram seletivamente a resposta em freqüência do amplificador, atenuando ou amplificando certas faixas de freqüências.

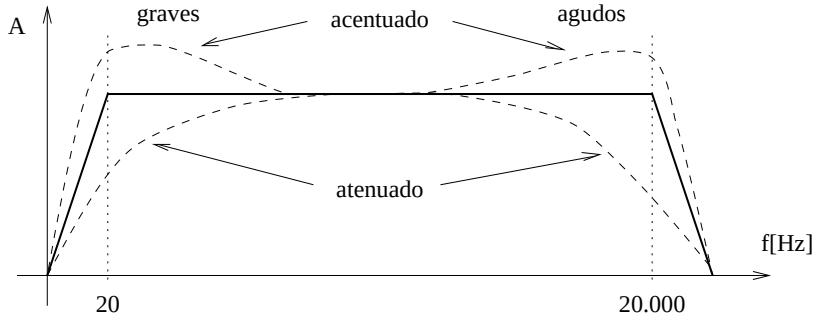


Figura 2.12: Espectro de freqüência de um amplificador de áudio.

Num amplificador de áudio, os controles de graves e agudos são usados para *equalizar* a resposta de freqüência do aparelho àquela da sala onde os alto-falantes se encontram. A equalização se torna necessária porque o meio de transmissão tem um comportamento não-uniforme com a freqüência. Se a sala contém tapetes e cortinas espessos, o comportamento acústico da sala *i.e.,* o meio de transmissão, é tal que as freqüências altas são atenuadas e as baixas acentuadas. Os controles de graves e agudos devem ser ajustados de forma a compensar o comportamento acústico da sala. A Figura 2.13 mostra o comportamento acústico da sala descrita acima (linha pontilhada), o reforço nos agudos e a atenuação nos graves (linha tracejada) e o resultado com o comportamento do amplificador alterado de forma a compensar as deficiências do meio de transmissão. A equalização de um meio físico consiste da alteração do comportamento de alguns componentes do sistema para compensar deficiências de outros.

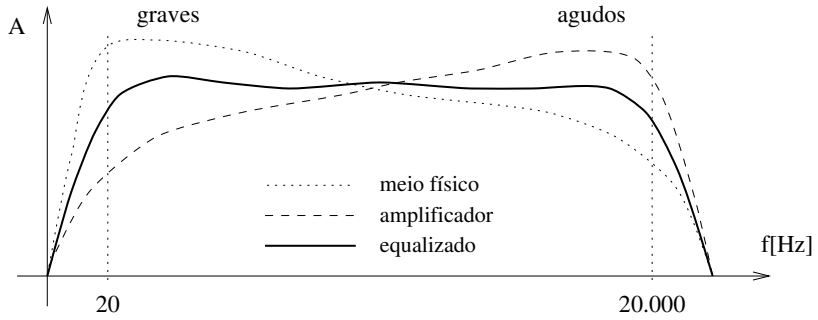


Figura 2.13: Equalização da resposta em freqüência de um amplificador de áudio.

2.2.4 Filtros

Um meio de transmissão (*e.g.* linha telefônica) pode ser considerado como uma caixa preta com uma entrada e uma saída e que produz na sua saída uma cópia mais ou menos fiel dos sinais injetados na entrada. A Figura 2.14 mostra um sistema de transmissão que reproduz, com distorções, o sinal digital aplicado a sua entrada. Os tipos de distorção causados pelos meios de transmissão comumente empregados em sistemas de comunicação de dados são o assunto da Seção 2.3.

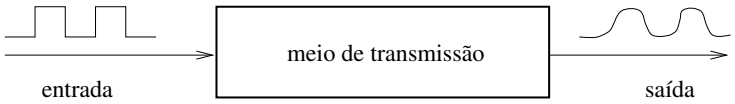
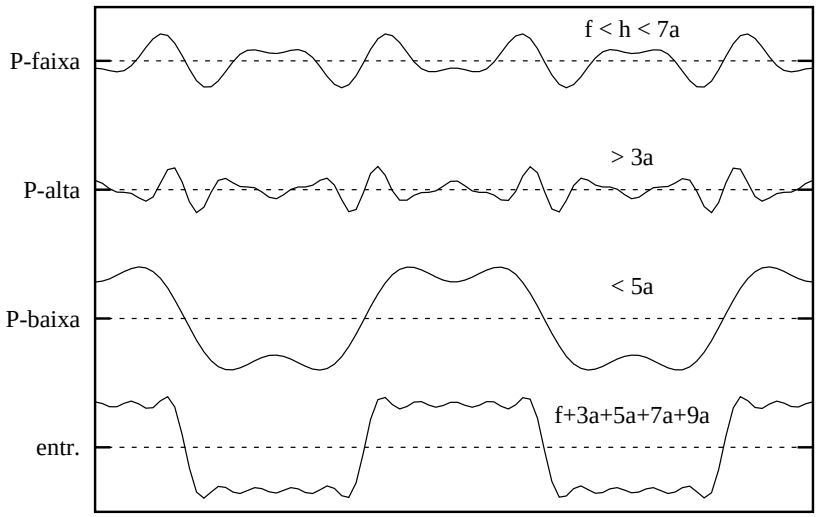


Figura 2.14: Modelo de comportamento de um meio de transmissão.

Todos os sistemas de transmissão introduzem um ou mais tipos de distorção no sinal que transportam. Quando se deseja quantizar a distorção como uma função da freqüência do sinal, o meio de transmissão pode ser considerado como um *filtro*. Filtros tem um comportamento que varia com a freqüência do sinal, atenuando ou acentuando algumas faixas de freqüência enquanto outras faixas passam através deles sem alteração.



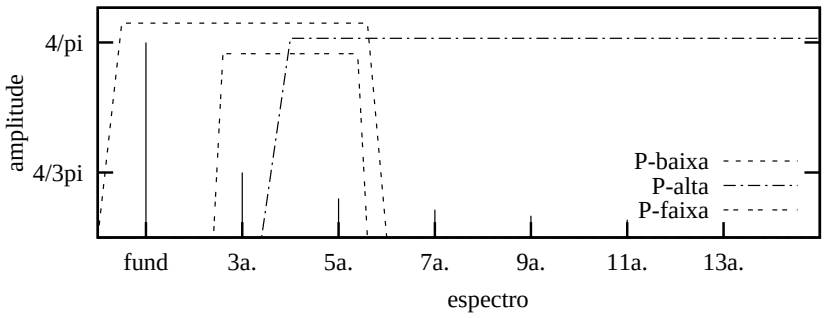


Figura 2.15: Efeito de filtros na forma de uma onda quadrada.

A Figura 2.15 mostra a forma de onda resultante da passagem de uma onda quadrada através de filtros. Embaixo, a figura mostra a forma de onda original, com a fundamental e as quatro primeiras harmônicas, sem qualquer filtragem (*entr.*). O efeito de um *filtro passa-baixas* é "arredondar os cantos" da onda pois as harmônicas de ordem elevadas são removidas. O efeito de um *filtro passa-altas* é o de remover as componentes de baixa freqüência (a fundamental e terceira harmônica, neste exemplo). O efeito de um *filtro passa-faixa* é remover as componentes fora da faixa de passagem do filtro (a fundamental e harmônicas de ordem maior que cinco). A figura também mostra o espectro da onda quadrada e a envoltória dos três filtros – os sinais "acima" das envoltórias são fortemente atenuados.

A *largura de faixa* de um sinal x pode ser definida como a faixa de freqüências $\Omega(x)$ tal que

$$\Omega(x) = \{\omega_{1,2} \mid \omega_1 < \omega < \omega_2,\ \sum P(\omega) < 0.99P(x)\} \tag{2.15}$$

Dado um sinal x com potência total $P(x)$, a largura de faixa de x consiste de todas as freqüências necessárias para transportar 99% da potência de x. Dependendo da aplicação esta definição é muito estrita e outras faixas de potência podem ser usadas (90% da potência ao invés de 99%).

A *faixa de passagem* P de um filtro passa faixa f pode ser definida como

$$P(f) = \{\omega_{1,2} \mid \omega_1 < \omega < \omega_2,\ |\omega| < |\omega_c|/\sqrt{2}\} \tag{2.16}$$

onde ω_c é a freqüência de centro do canal do filtro e $|\omega|$ é a amplitude da componente de freqüência ω. O filtro passa-faixa mostrado na Figura 2.15 tem sua freqüência de centro em $\omega_c = 4 \cdot \textit{fundamental}$ e sua faixa de passagem é aproximadamente $\omega_5 - \omega_3$ *i.e.,* a diferença entre as freqüências da 3ª e 5ª harmônicas.

Infelizmente, os canais telefônicos de voz empregados na comunicação de dados se comportam como filtros. Um canal de voz tem uma resposta (aproximandamente) plana entre os 300 e os 3500 Hz. Isso significa que um canal de voz não provoca distorção siginificativa na amplitude de sinais com freqüências entre 300 e 3500 Hz. Fora desta faixa os sinais são fortemente atenuados.

Considere um modem padrão CCITT V.22-bis operando duplex no modo origem (mais detalhes na Seção 2.5). Neste caso, o modem tem seu canal de recepção sintonizado numa portadora de 2400 Hz enquanto que seu canal de transmissão é sintonizado em 1200 Hz *i.e.,* o modem "ouve" em 2400 Hz e "fala" em 1200 Hz, como mostra a Figura 2.16. O modem remoto se comporta como a imagem especular do modem local: o circuito de recepção do modem remoto é sintonizado na freqüência de transmissão do modem local; seu circuito de transmissão é sintonizado na freqüência do canal de recepção do modem local. A figura mostra a representação em domínio freqüência normalmente usada para representar sistemas de transmissão de sinais.

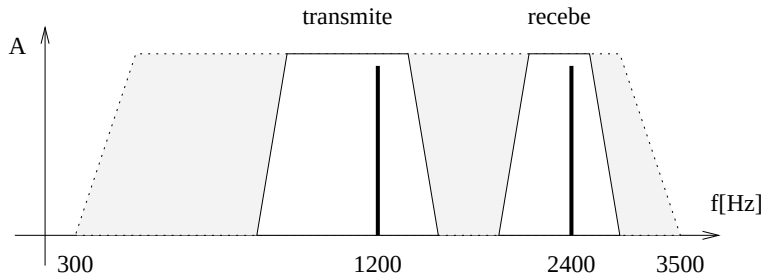


Figura 2.16: Espectros de freqüência de um canal de voz e do modem V.22bis.

O canal para a portadora de 1200 Hz compreende a faixa de 750 a 1650 Hz; o canal para a portadora de 2400 Hz ocupa a faixa de 1950 a 2850 Hz. O canal da portadora de 1200 Hz é implementado por um filtro passa-faixa centrado em 1200 Hz e que permite a passagem de freqüências entre 750 e 1650 Hz. Da mesma forma, o canal 2400 Hz é implementado por um filtro passa-faixa centrado em 2400 Hz permitindo a passagem de freqüências entre 1950 e 2850 Hz. As freqüências "fora do canal" são fortemente atenuadas.

2.3 Atenuação, Distorção e Ruído

Ao trafegar pelo meio físico, um sinal elétrico é atenuado por causa da resistência elétrica dos fios –quanto maior a distância, maior a atenuação. Esta atenuação não é uniforme em todo o espectro de freqüência de forma que frequências altas são atenuadas mais fortemente que frequências baixas. Ainda, a velocidade de propagação dos sinais varia conforme sua frequência, as frequências altas se propagam mais lentamente que as baixas e isso provoca distorção na forma de onda do sinal. Se isso não fosse o bastante, há o ruído elétrico omnipresente, tanto na forma de "ruído branco" como na forma de ruído impulsivo. Estes efeitos são discutidos no que segue.

Decibel A medida usada convencionalmente para expressar atenuação é o *Decibel*, abreviado como *dB – deci-Bell*. O dB expressa a relação entre duas grandezas, e para potência de sinais elétricos é definido como

$$A_{dB} = 10 \log(P/P_{\text{ref}}) \ . \tag{2.17}$$

Essa fórmula expressa a relação entre a potência P de um sinal com relação a potência P_{ref} de um sinal de referência. Poe exemplo, para um sinal transmitido por uma linha telefônica, P_{ref} é a potência do sinal nos terminais do transmissor enquanto que P é a potência do sinal recebido, medida nos terminais do receptor. Tipicamente, linhas telefônicas causam uma atenuação de 1dB/Km. Para modems em sítios a uma distância de 3Km um do outro, a atenuação provocada pela linha é de -3dB, o que corresponde a uma perda de potência de 50% ($-3 \approx 10 \log 1/2$). Por outro lado, se um amplificador é usado para aumentar a potência de um sinal, a saída do amplificador apresenta *ganho* em relação a sua entrada.

Se a grandeza em questão for tensão ao invés de potência, a definição de dB é

$$A_{dB} = 20 \log(V/V_{\text{ref}}) \tag{2.18}$$

porque $P = V^2/R$ para tensão V e resistor de carga R. Obviamente, a base de comparação deve ser mencionada *i.e.,* potência ou tensão e valor da resistência de referência. Por exemplo, em telefonia, a unidade de referência é *um miliwatt aplicado sobre uma resistência de 600 Ohms*. Neste caso, a atenuação é expressa em $dBmW$ ou dBm

$$A_{dBm} = 10 \log(P/1mW) \ . \tag{2.19}$$

A principal vantagem de se expressar ganho e atenuação em dB advém das propriedades dos logaritmos: uma multiplicação de números reais é equivalente a soma de seus logaritmos. Como exemplo considere um amplificador com ganho de 10dBm transmite por uma linha com atenuação de −7dB. A potência do sinal no receptor é dada por

$$P_{\text{rx}} = 10dBm - 7dB = 3dBm \ .$$

Resistência, capacitância e indutância A *resistência elétrica* é uma característica de materiais condutores de eletricidade que causa a dissipação de energia na forma de calor. Um exemplo de aplicação de *resistores* são os chuveiros elétricos. A resistência elétrica de um material é expressa em Ohms (Ω), é proporcional ao comprimento e inversamente proporcional à secção reta do condutor.

A *capacitância* dá a medida da capacidade de armazenagem de carga elétrica entre dois condutores. A carga armazenada entre os condutores cria um campo elétrico entre cargas positivas e negativas e este campo se opõe a mudança na configuração das cargas. O efeito da capacitância entre dois fios consiste na atenuação de sinais com frequências elevadas pois estes sinais mudam a configuração das cargas elétricas nos condutores com maior velocidade, o que encontra uma maior oposição pelo campo elétrico. A capacitância é expressa em Farads (micro- ou nano-Farads) e é proporcional à area paralela dos dois condutores e inversamente proporcional à distância entre eles.

Da mesma forma que capacitância, a *indutância* é uma característica de condutores elétricos que tende a provocar atenuação em sinais de frequências elevadas. A razão, contudo, é diferente. A passagem de corrente elétrica por um fio cria ao redor deste um campo magnético. Este campo se opõe a variações na corrente que lhe deu origem; quanto maior a velocidade da variação, maior a oposição. A indutância é expressa em Henrys (mili- ou micro-Henrys).

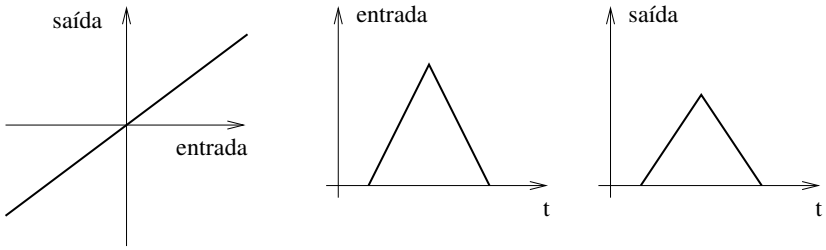
Estas três características contribuem para distorções nos sinais elétricos empregados em comunicação de dados. A resistência elétrica causa atenuação nos sinais. Capacitância e indutância atenuam os componentes de frequência elevada dos sinais fazendo com que um par de fios se comporte como um filtro passa-baixas. Os campos elétrico e magnético provocam *acoplamento eletromagnético* entre sinais trafegando por fios que estejam fisicamente próximos. Por exemplo, as conversas de fundo eventualmente ouvidas durante ligações telefônicas são um exemplo de acoplamento eletromagnético. Estas conversas de fundo são ruídos indesejáveis causados pela proximidade dos fios em cabos telefônicos. Um dos cabos se comporta como uma antena transmissora enquanto que o outro se comporta como uma antena receptora, numa situação análoga à radiodifusão em amplitude modulada.

2.3.1 Condição para Transmissão sem Distorção

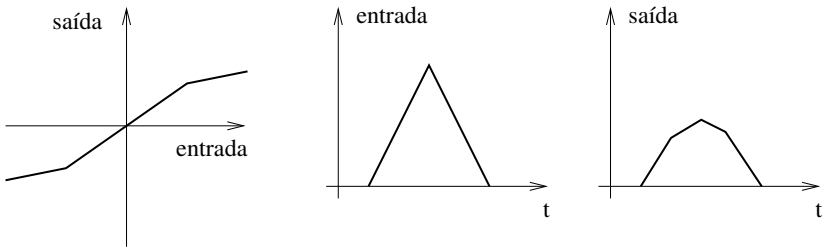
A Figura 2.14, pág. 25, mostra um modelo de um meio de transmissão cujo comportamento é o de um filtro que introduz alterações nos sinais que o atravessam. Idealmente, deseja-se um meio físico que não introduza nenhuma forma de distorção nos sinais que transporta. Isso significa que o comportamento de filtragem do meio físico deve ser *linear* tanto com relação à freqüência e fase quanto com relação à amplitude dos sinais.

A Figura 2.17 mostra as diferenças entre comportamentos linear e não-linear. Os gráficos no lado esquerdo da figura contém a *função de transferência de amplitude* de dois filtros. A função de transferência linear relaciona valores da entrada do filtro a valores atenuados na sua saída. Os dois gráficos do lado direito mostram a atenuação sofrida por sinal com uma certa forma de onda que atravessa o filtro. A saída do filtro é uma cópia atenuada do sinal na entrada do filtro. Por outro lado, a função de transferência não-linear, além de atenuar, introduz distorção na *forma* de onda do sinal em sua saída: a saída é uma verão "achatada" da entrada. Além de reduzir a amplitude, este filtro altera a contúdo harmônico do sinal ao mudar sua forma de onda.

Da mesma forma que um filtro pode possuir uma função de transferência com não-linearidades de amplitude, filtros podem produzir alterações no espectro de freqüência e na fase dos sinais. As próximas seções discutem os problemas causados por estas formas de distorção.



(a) Meio de transmissão linear



(b) Meio de transmissão não–linear

Figura 2.17: Meios de transmissão lineares e não-lineares.

2.3.2 Atenuação versus Distância

A Tabela 2.1 mostra as características elétricas de cabos telefônicos [Mon90], onde R é a resistência em Ohms, C a capacitância em Farads e AWG o código que identifica o fio pelo seu diâmetro ("American Wire Gauge"). Note que esses são valores por kilômetro e portanto, quanto mais longa a linha, maior a atenuação resistiva e capacitiva apresentada pelo par de fios.

diâmetro [mm]	AWG	R [Ω/Km]	C [nF/Km]
0.40	26	288	49
0.50	24	184	51
0.65	22	106	51
0.90	19	56	51

Tabela 2.1: Resistência e capacitância de cabos telefônicos.

A Figura 2.18 mostra os efeitos combinados de atenuação resistiva e capacitiva em função da distância na forma de onda de um sinal digital. Suponha que linha de transmissão é uma caixa preta. Para uma entrada digital $E(t)$ que varia entre duas tensões (0V a +1V), a saída varia como $S(t) \propto e^{-\alpha t}$. De cima para baixo, a figura mostra a onda quadrada à entrada $E(t)$, a saída $S1(t)$ produzida por uma linha de curta distância (*i.e.,* $\alpha = 2/t_b$), e a saída $S2(t)$ produzida por uma linha longa (*i.e.,* $\alpha = 1/2t_b$). Note que t_b é o intervalo de duração de um bit.

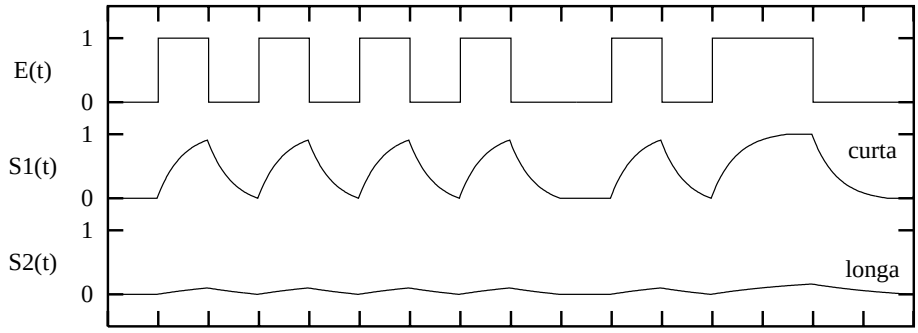


Figura 2.18: Efeito da atenuação resistiva e capacitiva em função da distância.

A curva para $S1(t)$ mostra a distorção introduzida pela capacitância distribuída ao longo da linha de curta distância. A atenuação capacitiva predomina sobre a atenuação resistiva: a amplitude do sinal varia pouco embora sua forma seja bastante distorcida. Um circuito relativamente simples seria capaz de recuperar a informação digital contida no sinal. A curva para $S2(t)$ mostra os efeitos combinados das atenuações resistiva e capacitiva. Neste caso, a atenuação resistiva predomina e o sinal é fortemente atenuado pela linha de transmissão. A recuperação da informação digital depende da amplificação do sinal recebido até níveis que permitam diferenciar 0s de 1s. A dificuldade neste caso é que o ruído presente na linha é amplificado junto com o sinal, e isso pode provocar a detecção errônea da informação contida no sinal.

2.3.3 Atenuação versus Freqüência

A Figura 2.19 mostra o gabarito de atenuação para uma linha telefônica discada [Mon90]. A concessionária promete a seus clientes que o comportamento elétrico de suas linhas não pode ser *pior* que o determinado pelos limites no gabarito. Linhas privativas tem características bastante melhores que aquelas mostradas na Figura 2.19. Note que a linha telefônica se comporta como um filtro passa-baixas (Figura 2.15, pág. 26).

A zona sombreada da Figura 2.19 indica os limites aceitáveis para atenuação dentro da faixa de frequências de um canal de voz. Entre os 500 e os 2400 Hz, a linha discada pode provocar diferenças de nível de até 10dB! Lembre que uma atenuação de 3dB correponde a uma perda de potência de 50%. Nos extremos da faixa de passagem do canal de voz, as diferenças de nível podem ser ainda maiores. As concessionárias empregam circuitos equalizadores para melhorar a resposta em frequência das linhas. As frequências nos dois extremos da faixa de passagem do canal de voz são acentuadas de forma a compensar as deficiências do meio de transmissão.

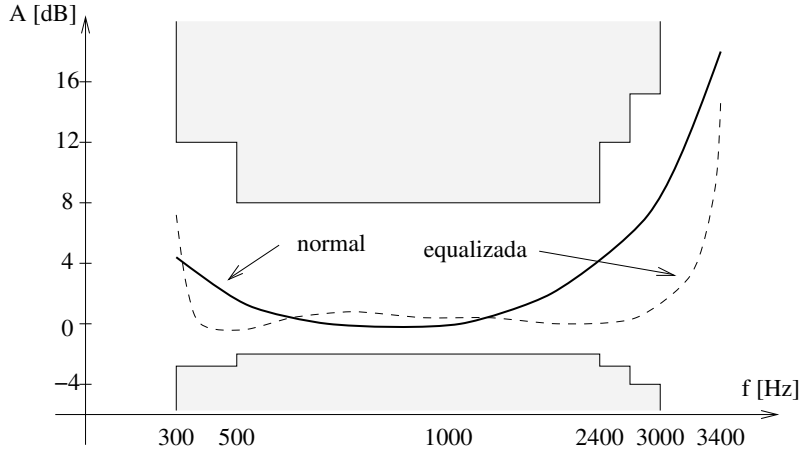


Figura 2.19: Gabarito de características de frequência de linha discada.

2.3.4 Distorção de Fase

O efeito da *distorção de fase*, também chamada de *atraso de fase*, é mostrado na Figura 2.20. A figura mostra, na linha contínua, uma onda "quadrada" de 1kHz (período de 1ms) com sua fundamental e terceira harmônicas com a fase correta (linhas tracejadas). Considere a transmissão deste sinal através de uma linha telefônica onde frequências na faixa dos 3kHz sofrem um atraso de 0.1ms, isto é, a terceira harmônica se propaga mais lentamente do que a fundamental. Devido ao atraso na propagação da terceira harmônica, a forma da onda na recepção fica bastante distorcida, com relação a original.

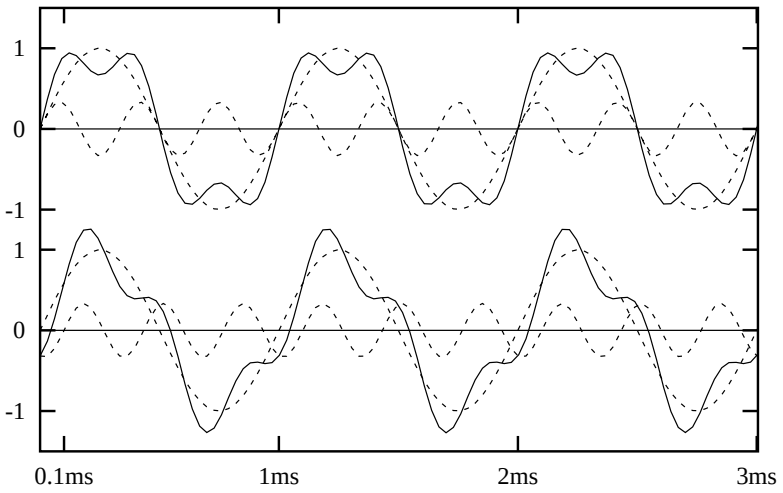


Figura 2.20: Efeito de atraso de fase na terceira harmônica.

O sistema auditivo consegue tolerar distorção de fase bastante bem. O mesmo não acontece com sistemas de comunicação de dados pois a recuperação da informação digital depende em grande medida da forma da onda que transporta a informação digital. As concessionárias adotam medidas para controlar as distorções de fase e amplitude através de amplificadores e filtros que compensam as deficiências da linha. Linhas que recebem este tipo de tratamento são chamadas de *linhas condicionadas*.

2.3.5 Interferência entre Símbolos

Outra forma de distorção, relacionada à distorção de fase, é a *interferência entre símbolos*. Considere os gráficos mostrados na Figura 2.21. As linhas tracejadas representam os limiares de decisão no circuito do receptor que identificam os valores binários `1` e `0`. Sinais entre as duas linhas tracejadas são considerados como sendo do último valor amostrado pelo circuito de detecção. O sinal $E(t)$ representa o sinal digital injetado no meio de transmissão. O sinal $S1(t)$ é o resultado de transmissão através de uma linha com faixa de passagem adequada e com atraso de fase mínimo. As bordas são um pouco arredondadas mas os valores de cada bit pode ser facilmente identificado pelo receptor.

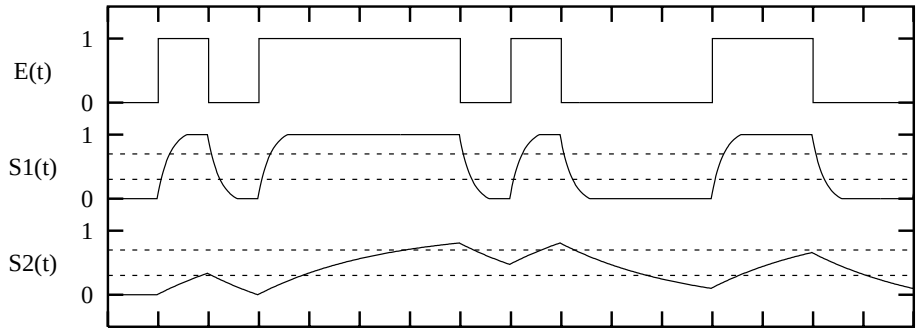
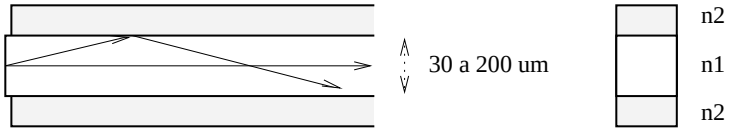


Figura 2.21: Efeito da interferência entre símbolos.

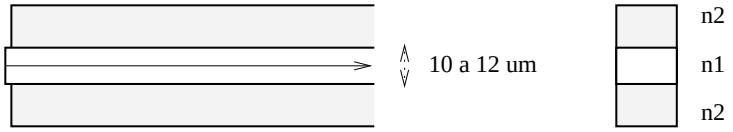
O sinal $S2(t)$ resulta da transmissão por um meio com faixa de passagem estreita e com atraso de fase considerável. O sinal $S2(t)$ sofre tamanha distorção que o circuito de detecção do receptor será incapaz de recuperar a informação contida em $E(t)$. O comportamento do meio de transmissão é tal que as harmônicas de ordem elevada sofrem tanto atraso que cada símbolo interfere com um ou mais de seus sucessores. Por exemplo, o segundo, quarto, quinto e sexto bits em $E(t)$ serão detectados erroneamente pelo receptor.

Interferência entre símbolos em fibras ópticas A Figura 2.22 mostra o percurso de raios de luz dentro de uma fibra óptica. A casca tem um índice de refração n_1 menor do que o índice de refração n_2 do núcleo. Ao se propagar pela fibra, os raios de luz se refletem nas paredes internas do núcleo. Os raios que incidem na fronteira entre casca e núcleo com um ângulo pequeno com relação ao eixo da fibra se propagam ao longo do eixo da fibra. Os raios de luz que incidem na fronteira com um ângulo grande com relação ao eixo da fibra são fortemente atenuados. A figura mostra as dimensões relativas de dois tipos de fibra, chamados de *multimodal* e *monomodal*.

As dimensões da fibra multimodal são tais que raios de luz podem se propagar ao longo de caminhos diferentes. Os raios que percorrem caminhos mais longos chegam atrasados com relação aos que percorrem caminhos curtos, o que provoca distorção na forma dos pulsos de luz. Esta distorção nos pulsos impõe limites no comprimento útil das fibras e na máxima taxa de transmissão possível por causa dos efeitos da interferência entre símbolos. A Figura 2.23 mostra os efeitos da propagação através de fibras multi- e monomodais na forma de um trem de pulsos, como detectado pelo receptor. Em fibras monomodais, as dimensões são tais que o número possível de caminhos é muito pequeno, o que reduz a distorção na forma dos pulsos de luz.



(a) Fibra multimodal



(b) Fibra monomodal

Figura 2.22: Modos de propagação na fibra.

(a) Fibra multimodal

(b) Fibra monomodal

Figura 2.23: Distorção causada pela propagação através da fibra.

2.3.6 Ruído

Além das distorções introduzidas pelas características elétricas dos meios físicos, outro fenômeno dificulta a transmissão de sinais elétricos: o omnipresente *ruído*. As fontes de ruído podem ser classificadas em três grupos:

- ruído térmico
- interferência eletromagnética
- ruído impulsivo.

A relação entre a potência do sinal transmitido e a potência do ruído existente no canal é dada pela *relação sinal-ruído*, expressa em dB por

$$S/R_{dB} = 10\log(P_{\text{sinal}}/P_{\text{ruido}}) \tag{2.20}$$

Valores típicos de relação sinal/ruído para linhas telefônicas são da ordem de 30dB (relação de 1.000 : 1). Valores típicos de relação sinal/ruído para equipamentos de áudio de boa qualidade são da ordem de 60dB (1.000.000 : 1).

Ruído térmico Como o nome indica, o *ruído térmico* é causado pela movimentação dos elétrons livres num condutor. Quanto maior a temperatura no condutor, maior a movimentação de seus elétrons livres. Este tipo de ruído não pode ser eliminado e portanto, impõe limites no desempenho de sistemas de comunicação. Se considerado como um sinal, o ruído térmico

contém componentes harmônicos de *todas* as frequências e por isso também é chamado de *ruído branco*. A potência do ruído térmico introduzido por qualquer dispositivo ou condutor elétrico, para uma largura de faixa F_1 de 1 Hz, isto é, a *densidade de potência do ruído*, é dada por

$$N_0 = kT \quad [\text{Watts/Hz}] \tag{2.21}$$

onde $k = 1.3803 \times 10^{-23} J/K$ é a constante de Boltzman e T a temperatura absoluta (Kelvin). Para uma faixa com largura F Hz, a potência do ruído é

$$N = kTF \quad [\text{Watts}] \tag{2.22}$$

Essa fórmula mostra que a potência do ruído aumenta com a largura de faixa do canal.

Interferência eletromagnética Este tipo de ruído é causado por acoplamento eletromagnético entre condutores e sistemas que estejam fisicamente próximos. Em geral, a potência dos sinais interferentes é da mesma ordem de grandeza que aquela do ruído térmico. O uso de cabos blindados tende a reduzir bastante os efeitos de interferência. Estes cabos contém um ou mais pares de condutores encobertos por uma fita ou malha de metal. Quando convenientemente aterrada, a malha forma uma "blindagem" muito eficiente contra interferência eletromagnética.

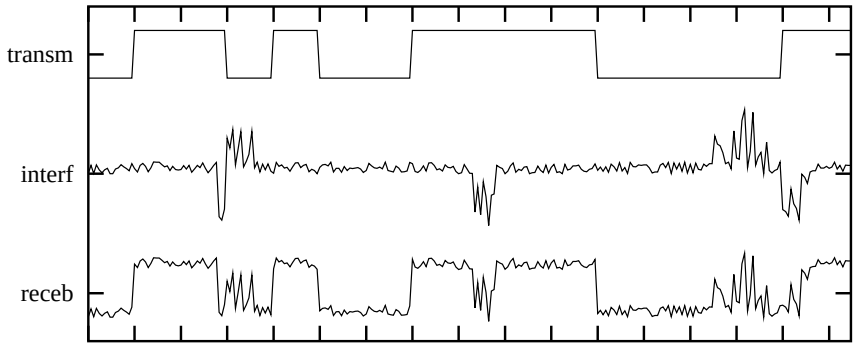


Figura 2.24: Efeito dos ruídos branco e impulsivo.

Ruído impulsivo Como seu nome indica, o ruído impulsivo consiste de pulsos de curta duração e amplitude relativamente alta. Estes pulsos podem ser causados por relâmpagos, motores elétricos ou contatos intermitentes em conectores, por exemplo. Estas fontes de ruído podem ser difíceis de localizar ou remover. A Figura 2.24 mostra os efeitos de ruído térmico e impulsivo numa linha de comunicação. O sinal emitido pelo transmissor (transm) sofre de contaminação por ruído branco ao trafegar pela linha –os pulsos irregulares de baixa intensidade– e por ruído impulsivo proveniente do meio em que a linha se encontra. Num sistema de comunicação de dados, os pulsos de ruído impulsivo de maior intensidade podem causar erros de detecção no receptor. Supondo que o receptor tome amostras de sua entrada (receb) no meio do intervalo correspondente a cada bit, pelo menos quatro dos bits recebidos poderiam ser amostrados com o valor errado. Em geral, pulsos de ruído duram alguns mili-segundos podendo portanto afetar dezenas a milhares de bits, dependendo da velocidade de transmissão.

2.3.7 Capacidade de Canal

Dado um canal de comunicação, *e.g.* uma linha telefônica, qual é a maior taxa de transmissão digital possível de ser usada neste canal? A Figura 2.18, pág. 30, indica que existem limites impostos pelas características elétricas do meio físico. O nível de ruído na linha também limita

a taxa de transmissão bem como a diferença de amplitude entre os sinais que representam a informação.

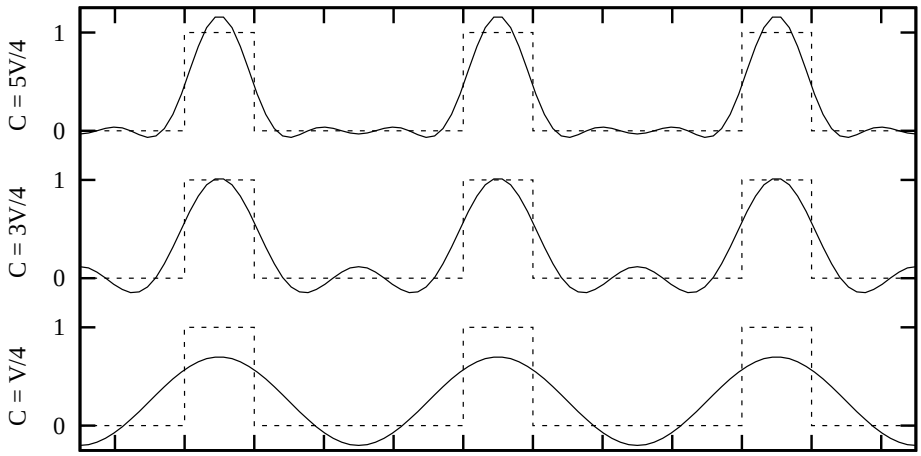


Figura 2.25: Efeito da capacidade de canal C em sinal digital com velocidade V.

A Figura 2.25 mostra outra limitação na taxa de transmissão de sinais digitais. Nela, são mostrados os efeitos da largura de faixa na transmissão de um sinal digital. Para uma capacidade de canal de C [Hz] e sinal digital com velocidade V [bps], a figura mostra o efeito de filtro passa-baixas de um canal de "faixa estreita". O sinal à entrada do canal é mostrado pela linha tracejada e as linhas contínuas mostram o sinal à saída. Um canal com largura de faixa similar à do sinal provoca pouca atenuação ($C = 5V/4$) enquanto que um canal com faixa estreita ($C = V/4$) provoca muita distorção. Uma regra de projeto de equipamentos de áudio reza que uma onda quadrada de freqüência F é fielmente reproduzida por um equipamento que se comporta como um canal com capacidade igual a $10F$.

Como visto na Seção 2.2.1, quanto maior o número de harmônicos presentes numa Série de Fourier, mais fielmente o sinal original é representado pela somatória dos harmônicos. Um meio de transmissão com faixa limitada atenua fortemente sinais com frequências fora da faixa de passagem. Assim, na transmissão de sinais digitais através de um canal de voz, a fundamental e as primeiras harmônicas tendem a ser pouco atenuadas enquanto que as harmônicas de ordem superior são praticamente eliminados do sinal. Visto de outra forma, quanto maior for a faixa de passagem de um meio físico, menor a distorção sofrida pelos sinais ao atravessá-lo.

Teorema de Nyquist Para um canal ideal *sem ruído*, a largura mínima de faixa de passagem necessária para transportar um sinal digital a uma taxa de B bits por segundo é $B/2$ Hz. Este é o *Teorema de Nyquist*. Um canal de voz *sem ruído* com faixa de passagem de 3 kHz não transmite dados binários a taxas superiores a 6000 bps. Intuitivamente, o sinal digital consistindo de 1's e 0's alternados *i.e.,* uma onda quadrada com frequência $B/2$, pode ser recuperado de uma senóide com frequência $B/2$.

Suponha que num dado sistema de transmissão sejam usados quatro níveis de tensão para representar os dados. Assim, a cada um dos níveis de tensão correspondem *dois* bis, ou um *dibit*, codificados como $-2V = 11$, $-1V = 10$, $1V = 00$, $2V = 01$. Com essa codificação, cada transição no sinal (*i.e.,* cada baud) transporta a informação correspondente a dois bits. Note que neste caso, a taxa de sinalização ("baud rate") é a metade da taxa de transmissão ("bit rate"). Esta codificação é equivalente a uma representação de números em base 4 ao invés de representação em base 2. A Figura 2.26 contém um diagrama que mostra a transmissão da

sequência 0011 0110 0011 00 usando o código acima. Quando este tipo de codificação multinível é empregada, a formulação do Teorema de Nyquist é

$$C = 2B \log_2 N \tag{2.23}$$

onde C é a capacidade do canal [bps], B é a faixa de passagem do canal [Hz] e N o número de níveis discretos do sinal. Assim, com os dados do exemplo acima, a máxima taxa de transmissão C é 12.000 bps. Infelizmente, existem limites no número de níveis que podem ser usados em códigos multinível. Os circuitos eletrônicos do receptor devem ser capazes de resolver todos os diferentes níveis do código e isso pode ser difícil face a todas as formas de distorção já vistas.

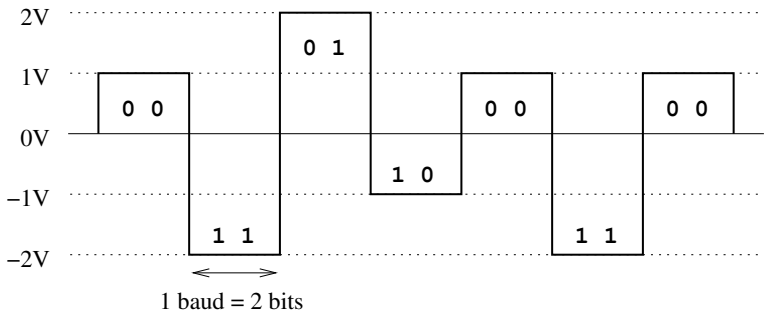


Figura 2.26: Codificação com dois bits por baud.

Teorema de Shannon O *Teorema de Shannon* estabelece que, para um canal com faixa de passagem F e relação sinal-ruído de S/R, a máxima taxa de transmissão possível é

$$T_{\max} = F \log_2(1 + S/R) \tag{2.24}$$

Assim, um canal de voz com faixa de passagem de 3 kHz e relação sinal-ruído de 30dB (1000 : 1), *não pode* transmitir dados a taxas superiores a 30000 bps. Note que este é um limite impossível de ser ultrapassado; taxas de transmissão de 9600 bps em canais de voz são consideradas excelentes – estas taxas implicam na transmissão de quatro bits para cada alteração no sinal transmitido. Mais detalhes na Seção 2.5.

2.4 Transmissão na Banda Básica

As características físicas e elétricas dos meios físicos de comunicação impõem limites na taxa máxima de transferência de dados através deles. Várias técnicas são usadas para minimizar os efeitos destas limitações. Estas técnicas consistem na alteração de um ou mais parâmetros do sinal que trafega pelo meio físico, como por exemplo, a transcodificação dos dados digitais em sinais analógicos que são então transmitidos pelo meio físico. Este tipo de transcodificação é chamado de *modulação* porque os dados digitais *modulam* um sinal analógico. O sinal analógico modulado é chamado de *portador* da informação contida nos dados digitais. A Seção 2.5 apresenta técnicas de modulação empregadas em Comunicação de Dados.

Esta seção discute algumas das técnicas de transcodificação dos dados digitais para adequá-los a transmissão sem o recurso da modulação. Estas transformações consistem basicamente na mudança da representação da informação digital. Esta forma de transmissão de dados digitais é chamada de *transmissão na banda básica* porque o espectro de freqüência dos sinais transcodificados não difere significativamente do espectro da codificação digital interna ao computador. Os circuitos eletrônicos empregados em transmissão na banda básica são mais simples,

e possivelmente mais baratos, que os moduladores e demoduladores empregados em transmisão analógica de dados digitais.

A *taxa de transmissão* de dados, cuja unidade é expressa em *bits por segundo* ou *bps*, indica o número de bits transferidos por unidade de tempo. A *taxa de sinalização*, expressa em *baud*, indica o número de vezes por segundo em que ocorrem alterações no sinal injetado no meio de transmissão. Um *símbolo* corresponde a um elemento do código (alfabeto) de sinalização empregado. A taxa de sinalização mede a velocidade com que símbolos são injetados no meio de transmissão. Não é necessário que a taxa de transmissão seja igual a taxa de sinalização, em geral, elas são diferentes. Se o código de sinalização for tal que a cada símbolo correspondem dois bits, a taxa de transmissão para este código é o dobro da taxa de sinalização.

Os códigos usados na transmissão em banda básica devem apresentar as seguintes características:

- A *transparência* de um código indica que este consegue representar qualquer seqüência de bits.
- A *unicidade de representação* implica em que deve haver uma correspondência biunívoca entre cada um dos símbolos do código e a seqüência de bits representada pelo símbolo.
- A *largura mínima da faixa* do espectro do sinal pode limitar a utilização de alguns códigos porque eles necessitam de um grande número de componentes harmônicos para ser decodificados. Outros códigos contém componentes de freqüência zero (corrente contínua), o que os torna inadequados para transmissão por circuitos de telefonia.
- A capacidade de *recuperação de sincronismo* do código indica a facilidade com que se recupera a informação de sincronismo inserida pelo transmissor. Códigos com transições freqüentes facilitam a sincronia entre transmissor e receptor.
- A *imunidade a ruído* do código permite uma melhor recuperação de informação em presença de ruído.
- Finalmente, a *complexidade* do código determina o custo dos codificadores e decodificadores.

Um código que não seja transparente e não contenha uma representação biunívoca não é muito útil porque a recuperação correta de informação pode ser impossível em códigos que não apresentam estas duas propriedades. A largura de faixa de passagem e a imunidade a ruído de um código são fatores determinantes na máxima velocidade de transmissão e distância que podem ser atingidos com um dado código e meio físico de transmissão.

2.4.1 Códigos para Transmissão em Banda Básica

A Figura 2.27 mostra alguns dos tipos de códigos empregados em sistemas de transmissão em banda básica. O *sinal de sincronismo* é usado para demarcar o início de cada bit transmitido (borda ascendente) e o ponto de amostragem de cada bit recebido (borda descendente). O tempo de duração de um bit é chamado de *intervalo de bit* t_b.

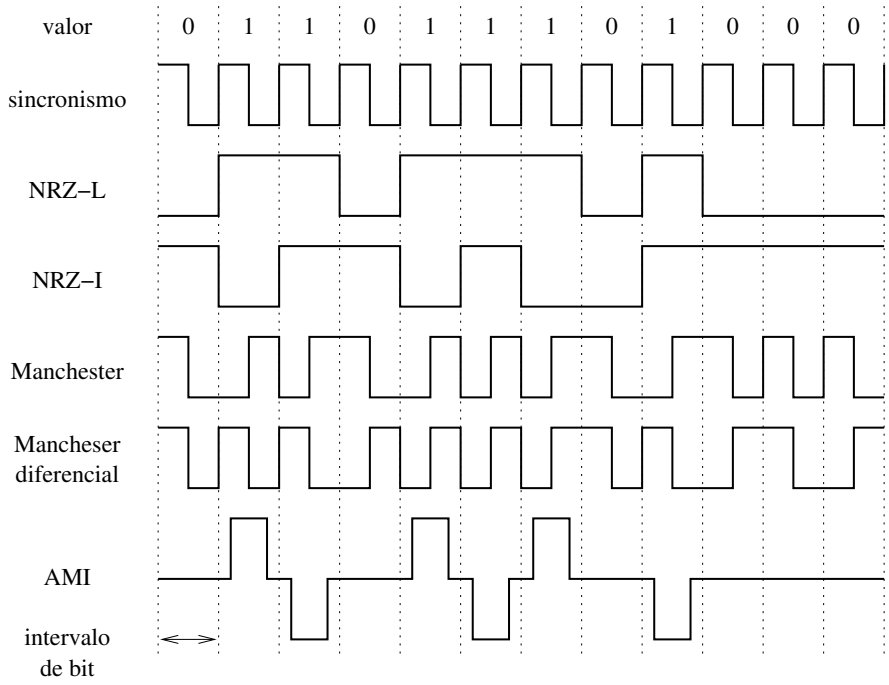


Figura 2.27: Códigos para transmissão em banda básica.

NRZ O primeiro código é chamado de *NRZ-L* "non-return-to-zero level". O bit `1` é representado por um nível alto durante t_b, e o bit `0` por um nível baixo durante o intervalo t_b. Este é o código usado na interface EIA-232 (Seção 3.4), mas com os níveis trocados. No segundo código, *NRZ-I* "non-return-to-zero, invert on ones" ocorre uma inversão de nível a cada bit em `1`; o nível permanece constante nos bits em `0`. Este tipo de código é chamado de *diferencial* porque os bits são representados pela diferença de nível no sinal entre dois intervalos de bit consecutivos. Códigos diferenciais apresentam melhor imunidade a ruído porque, em presença de ruído, a detecção de uma transição entre dois níveis pode ser feita com mais segurança do que a detecção de um nível que não permanece constante por causa do ruído. Os códigos NRZ são simples e eficientes embora a recuperação de sincronismo possa ser problemática se os dados contém longas cadeias de bits iguais (NRZ-L) ou de bits em `0` (NRZ-I). Estes códigos são chamados de "non-return-to-zero" para distingui-los dos códigos bipolares que "retornam a zero" entre dois símbolos consecutivos (*e.g.* o código AMI). Códigos NRZ são usualmente empregados no armazenamento de dados em meio magnético (*e.g.* discos e fitas magnéticas). O código NRZ-I0 é usado em transmissões com o protocolo BSC (Seção 4.3.2). NRZ-I0 é similar ao NRZ-I mas as inversões de nível ocorrem nos bits em `0`.

Manchester No código *Manchester* ocorre uma transição no meio do intervalo correspondente a cada bit e a transição denota simultaneamente informação e sincronismo – veja a Figura 2.27. Uma transição ascendente corresponde a `1` e uma transição descendente corresponde a `0`. No código *Manchester diferencial*, a transição no centro de t_b contém apenas informação de sincronismo. O bit `1` é denotado por uma transição no início do intervalo t_b; o bit `0` é denotado pela ausência de uma transição no início do intervalo. Estes códigos são chamados de auto-sincronizantes porque a informação de sincronismo é facilmente recuperável: sempre ocorre uma transição no meio do intervalo t_b. Uma desvantagem destes códigos é a sua taxa de sinalização. Para longas seqüências de `1`'s ou `0`'s, a taxa de sinalização é o dobro da taxa de transmissão. Estes códigos são usados em redes locais: o código Manchester é empregado na Ethernet (Seção 5.4) enquanto que o Manchester diferencial é adotado no padrão IEEE 802.5

(anel com passagem de bastão).

Códigos bipolares O código *AMI* ("alternate mark inversion") é chamado de *código bipolar* porque os pulsos que carregam informação são de duas polaridades: positivos e negativos com relação a uma referência de 0V – veja a Figura 2.27. No código AMI, bits em `0` são representados por uma tensão de 0V enquanto que bits em `1` são representados por pulsos que se alternam entre uma tensão positiva e uma tensão negativa. Este código não é auto-sincronizante porque longas cadeias de `0`'s podem provocar a perda de sincronismo entre transmissor e receptor. Por outro lado, este código produz um número menor de transições que o código Manchester. Para contornar o problema da perda de sincronismo, a técnica de *violação de código* é empregada na codificação *HDB3* ("high density binary – no more than 3 zeroes"). O HDB3 usa a mesma codificação do código AMI mas, quando uma seqüência de quatro `0`'s é transmitida, um pulso de mesma polaridade que o anterior é inserido pelo codificador no lugar do quarto `0`, causando uma violação do código. As violações subseqüentes usam polaridades invertidas com relação a anterior. Antes de qualquer processamento, o decodificador detecta a violação de código e remove o pulso adicional inserido pelo codificador.

2.4.2 Espectro de Freqüência dos Códigos

Os meios físicos de transmissão de dados impõem limitações na velocidade de transmissão. Na ausência de ruído, uma taxa de sinalização S [baud] pode ser atingida em um canal com faixa de passagem F [Hz], isto é $S = 2F$. Assim, quando mais estreito for o espectro de freqüência do código empregado na transmissão, maior a taxa de sinalização passível de ser usada num meio com uma dada faixa de passagem. A Figura 2.28 mostra o espectro de freqüência dos códigos NRZ, Manchester e AMI. Os códigos NRZ possuem componente DC (corrente contínua com freqüência zero), o que os torna inviáveis para uso em canais telefônicos. Seu espectro se espalha até $f = 2D$, para taxa de transmissão $D = 1/t_b$ [bps]. Os códigos Manchester não possuem componente DC mas seu espectro se espalha até $f = 1.7D$. Os códigos AMI e HDB3 também não possuem componente DC e seu espectro se concentra em torno de $f = D/2$, limitando-se até $f = D$. Os códigos bipolares são portanto apropriados para sistemas com altas taxas de transmissão. A desvantagem destes dois códigos está na sua menor imunidade a ruído por causa da codificação em três níveis distintos.

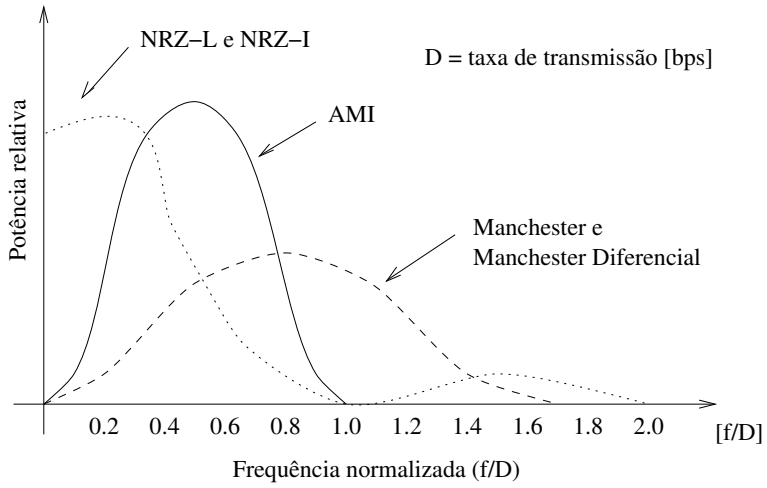


Figura 2.28: Espectro de freqüência dos códigos para transmissão em banda básica.

Taxa de Modulação

Para um dado código, sua taxa de sinalização S [baud] não é necessáriamente igual a sua taxa de transmissão $D = 1/t_b$ [bps]. A *taxa de modulação* M de um código mede a relação entre as taxas de transmissão e de sinalização daquele código $M = S/D$. Assim, além do espectro estreito, a *eficiência do código*, medida pela taxa de modulação, também é importante. A Figura 2.27 mostra que, para os códigos NRZ e AMI, ocorre no máximo uma transição no sinal durante o intervalo de bit t_b. Nos códigos Manchester, podem ocorrer até duas transições no sinal durante o intervalo t_b. A taxa de modulação dos códigos Manchester pode chegar até duas vezes a taxa de transmissão (*i.e.,* para transmitir um bit podem ser necessários até dois baud). A Tabela 2.2 mostra a relação entre as taxas de modulação e de transmissão para os cinco códigos mostrados na Figura 2.27.

	mínimo	101010	máximo
NRZ-L	0.0 (000 ou 111)	1.0	1.0 (1010)
NRZ-I	0.0 (000)	0.5	1.0 (111)
AMI	0.0 (000)	0.5	1.0 (111)
Manchester	1.0 (1010)	1.0	2.0 (000 ou 111)
Manch. diferencial	1.0 (111)	1.5	2.0 (000)

Tabela 2.2: Taxas de modulação dos códigos para transmissão em banda básica.

2.4.3 Arquitetura dos Modems Digitais

Os *modems digitais* empregam transmissão em banda básica e são mais simples e baratos que modems analógicos (Seção 2.5). Contudo, modems digitais somente podem ser usados em enlaces metropolitanos de até 40 Km de distância e em linhas privativas. As características físicas dos códigos banda básica e dos sistemas de telefonia, *e.g.* indução de ruído em linhas adjacentes e acoplamento por transformadores, impedem seu uso em linhas discadas. Os modems digitais podem ser síncronos e assíncronos e operam com taxas de transmissão de até 19200 bps.

A Figura 2.29 mostra o diagrama de blocos de um modem digital. Os dados a transmitir são fornecidos ao modem pelo terminal via sinal DTX da interface EIA-232 (Seção 3.4). O codificador transforma a seqüência de bits de acordo com o código de transmissão. O amplificador de saída entrega o sinal à linha com níveis de potência adequados. O condicionador de sinal amplifica e recupera a informação digital do sinal recebido. Na recepção, o decodificador recupera a seqüência original de bits. O circuito de sincronismo recupera a informação de sincronismo embutida no sinal recebido e cadencia a amostragem do sinal digital no sinal DRX da interface EIA-232.

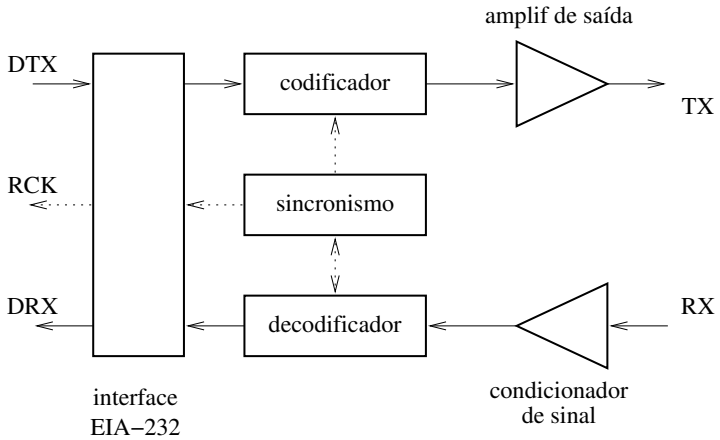


Figura 2.29: Diagrama de blocos de um modem digital.

A relação *taxa de transmissão*×*alcance* para modems banda básica é definida em função da máxima taxa de erros permitida. A taxa de erros t_E é medida pela relação entre o número de bits errados para um certo número de bits transmitidos. Assim, a norma Telebrás relativa a modems banda básica [Mon90] especifica que os modems devem operar a uma certa taxa de transmissão $D = 1/t_b$, com um certo alcance, e com uma taxa de erros $t_E < 5 \times 10^{-5}$ (50 bits errados em cada 10^6 bits transmitidos) em um par de fios tipo 26 AWG (0.40mm de diâmetro). A Tabela 2.3 mostra os valores mínimos especificados para transmissão em banda básica com estas características.

velocidade [bps]	alcance [Km]
600	30.0
1200	22.0
2400	16.0
4800	11.0
9600	8.0
19200	5.4

Tabela 2.3: Taxas de transmissão *versus* alcance para modems banda básica.

2.5 Transmissão por Sinais Modulados

Devido às características elétricas das linhas de transmissão usadas em telefonia, os sinais digitais usados por computadores devem ser adaptados para a transmissão de dados através daquelas linhas. Essa adaptação consiste em usar um sinal analógico como meio de transporte para o sinal digital. Esse sinal analógico, chamado de *portadora*, deve ser de tal natureza que possa trafegar por canais de voz sem sofrer distorção significativa. Os aparelhos que fazem a adaptação de sinais digitais para transmissão por canais de voz são os modems (MOduladores-DEModuladores). Na transmissão, o sinal digital é usado para modular a portadora; na recepção, o sinal analógico da portadora é demodulado para recuperar o sinal digital. O ITU-T (CCITT) emite as Recomendações da Série V que especificam os parâmetros de funcionamento de vários tipos de modems e cada classe de modem é designada pelo número da recomendação apropriada. A Seção 2.6 lista algumas destas recomendações.

2.5.1 Arquitetura dos Modems Analógicos

No mínimo, um modem analógico contém uma interface EIA-232, um modulador e um demodulador além de circuitos de condicionamento de sinal. A Figura 2.30 mostra um diagrama de blocos de um modem analógico. A interface EIA-232 traduz os níveis elétricos dos dados a transmitir DTX para níveis adequados ao funcionamento dos circuitos internos ao modem. O modulador usa o sinal digital para alterar a amplitude, freqüência ou fase da portadora, de acordo com o tipo de modulação empregado no modem. O filtro de transmissão (TX) remove harmônicos indesejáveis do sinal modulado de modo a que este se enquadre nos padrões de sinalização. O amplificador de saída aplica o sinal modulado com níveis de potência adequados à transmissão pela linha.

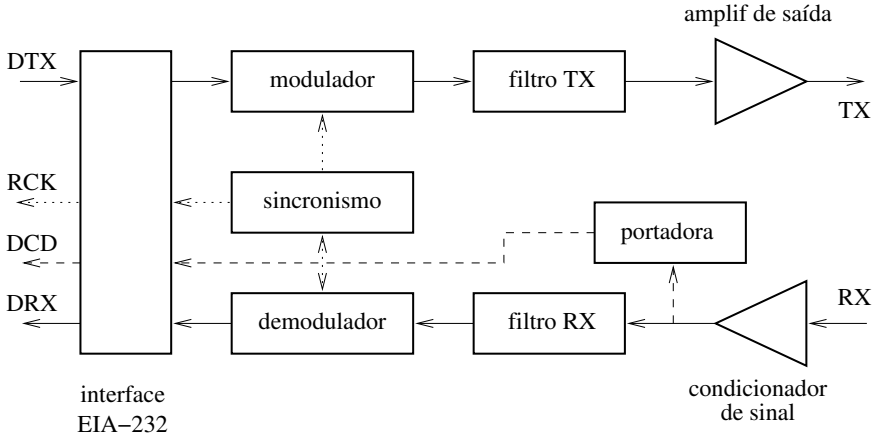


Figura 2.30: Diagrama de blocos de um modem analógico.

No lado da recepção, o sinal inicialmente passa por um condicionador, que o amplifica até um nível adequado para a demodulação. O filtro de recepção (RX) remove componentes de sinal devidos a ruído e equaliza o sinal recebido, compensando distorções introduzidas pela linha de transmissão. O demodulador recupera o sinal digital DRX a partir do sinal recebido e condicionado. O circuito de detecção de portadora informa ao modem sobre a presença de portadora (DCD) com nível e freqüência adequados. O circuito de recuperação de sincronismo fornece o sinal de relógio de recepção para o modem. A Seção 3.2 descreve a relação temporal entre RCK e DRX, veja a Figura 3.9, na página 59.

A discussão acima pressupõe o uso da interface EIA-232 entre ETD e ECD. Dependendo da velocidade de transmissão do modem, devem ser usadas interfaces de mais alta velocidade como a EIA-422 ou CCITT V.35.

2.5.2 Técnicas de Modulação

Os quatro tipos de modulação mais usados em comunicação de dados através de canais de voz são modulação em amplitude, modulação em freqüência, modulação em fase e modulação em fase e amplitude. Estas quatro técnicas são examinadas no que segue. f_p é a freqüência da senóide que transporta a informação digital e este sinal é chamado de *portadora*. A *freqüência angular* da portadora é definida como $\omega_p = 2\pi f_p$ e é uma abreviatura usada para simplificar expressões.

Modulação em Amplitude

A *modulação em amplitude* (MA) consiste em multiplicar uma senóide $P(t) = \cos(\omega_p t)$ pelo valor instantâneo do bit a ser transmitido *i.e.,* o valor do bit (1 ou 0) *modula a amplitude* da portadora $P(t)$. Modulação em amplitude também é chamada de "amplitude-shift keying" (ASK). A Figura 2.31 mostra a relação entre o sinal digital e sua representação por um sinal modulado em amplitude. Uma técnica de modulação em amplitude similar e esta é usada em modems para linhas especiais de 64 Kbps. Para um sinal $d(t)$ que descreve um trem de pulsos (0's e 1's), o sinal modulado $S_{MA}(t)$ é descrito por

$$S_{MA}(t) = d(t)\cos(\omega_p t)$$

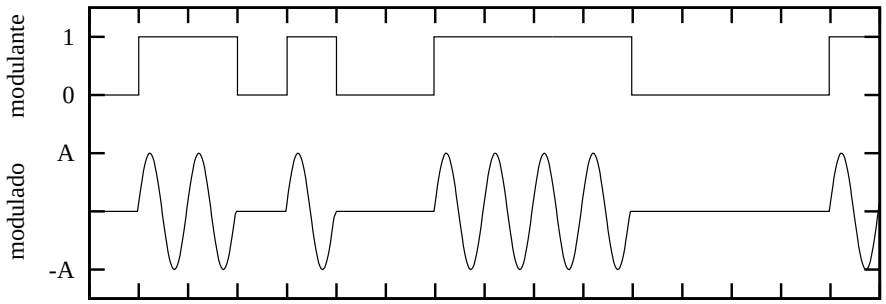


Figura 2.31: Sinal digital e sinal analógico correspondente modulado em amplitude.

Modulação em Freqüência

A *modulação em freqüência* (MF) consiste em alterar a freqüência da portadora em função do sinal digital a ser transmitido. Essa técnica de modulação também é conhecida por "frequency-shift keying" (FSK). Ao valor lógico 1 corresponde uma freqüência e ao valor lógico 0, outra. Se o sinal $d(t)$ descreve uma seqüência de bits (0's e 1's), o sinal $k(t)$ descreve a variação em freqüência Δf em torno da freqüência f_1, que é associada ao bit em 1.

$$k(t) = 2\pi(f_1 + (1 - d(t))\Delta f)t$$

O sinal modulado $S_{MF}(t)$ é descrito por

$$S_{MF}(t) = \cos k(t).$$

Por exemplo, o modem padrão CCITT V.23 usa, no canal principal, uma freqüência de 1300 Hz para representar 1 e de 2100 Hz para representar 0, como mostra a Figura 2.32.

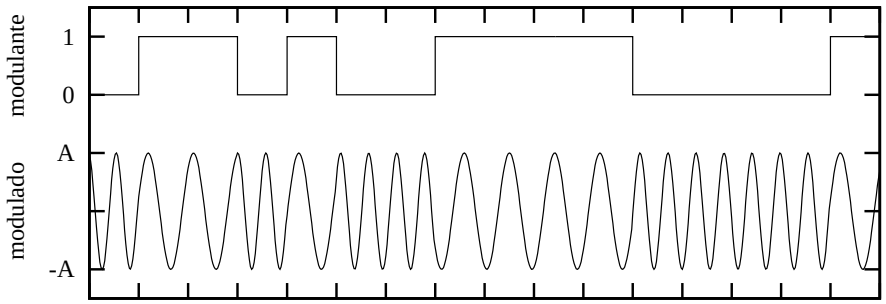


Figura 2.32: Sinal digital e sinal analógico correspondente modulado em freqüência.

Um dos circuitos usados para a modulação em freqüência consiste de um oscilador controlado por tensão (OCT). Para uma tensão de entrada correspondente à 1, o OCT produz em sua saída um sinal com a freqüência f_1. Para uma tensão de entrada correspondente à 0, o OCT produz um sinal de freqüência f_0. Na demodulação, o sinal recebido passa por dois filtros passa-faixa. O filtro F_1 é centrado em f_1 e filtro F_0 em f_0. A saída do filtro F_1 somente apresenta sinal durante os intervalos correspondentes aos bits em 1. O filtro F_0 somente apresenta sinal em sua saída durante os intervalos correspondentes aos bits em 0. A saída destes filtros é combinada para reproduzir o sinal digital que foi transmitido. A Figura 2.33 mostra o espectro de freqüência destes dois filtros – estes atenuam fortemente freqüências na área sombreada.

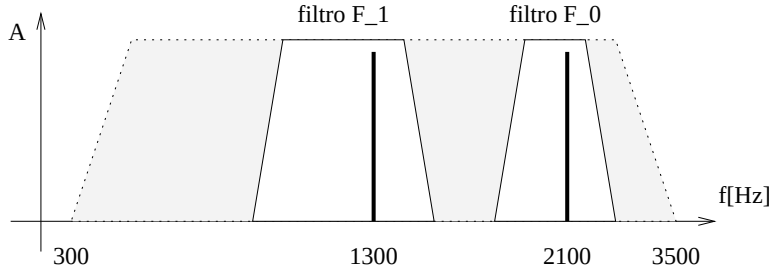


Figura 2.33: Filtros de demodulação do modem padrão V.23.

Modulação em Fase

Na *modulação em fase*, a informação digital é representada por mudanças na fase da portadora. A Figura 2.34 mostra as alterações na fase de uma senóide provocadas por deslocamentos em fase de 180º e 90º. Da esquerda para a direita, a figura mostra um ciclo completo de 360º, uma inversão de 180º, um ciclo completo, um quarto de ciclo, um deslocamento de fase de 90º seguido de um ciclo completo. O deslocamento de 180º corresponde a eliminação de metade de um ciclo enquanto que o deslocamento de 90º corresponde a eliminação de um quarto de ciclo. As linhas pontilhadas mostram a forma da onda caso não houvessem os deslocamentos de fase.

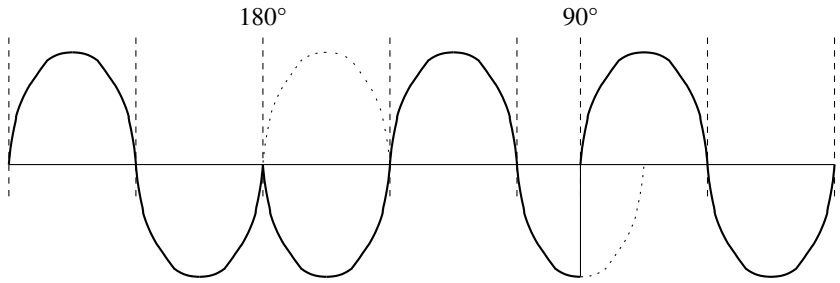


Figura 2.34: Efeito do deslocamento de fase em sinal senoidal.

Esta técnica de modulação é também conhecida como *modulação por deslocamento de fase* (MDF) ("phase-shift keying" ou PSK). A modulação em fase normalmente usada em modems emprega deslocamentos diferenciais de fase *i.e.,* a fase da portadora é deslocada com relação ao valor da fase correspondente ao bit imediatamente anterior, ao invés de em relação a um valor de referência fixo. Esta técnica é chamada de *modulação por deslocamento diferencial de fase* ou MDDF ("differential phase-shift keying" ou DPSK). Se o sinal $d(t)$ descreve uma seqüência de bits (0's e 1's), o sinal $\phi(t)$ descreve o valor de fase no instante t em função da fase no instante $(t-1)$. Para um código com dois valores de deslocamento de fase (*i.e.,* 0º e 180º), a fase da portadora varia no tempo como

$$\phi(t) = \pi d(t-1)$$

Para uma portadora de freqüência f_p, a um bit em 1 corresponde uma mudança de fase de $\Delta_1\phi = \pi$; a um bit em 0 corresponde uma mudança de fase de $\Delta_0\phi = 0$. Assim, o sinal modulado por deslocamento diferencial de fase $S_{MDDF}(t)$ é descrito por

$$S_{MDDF}(t) = \cos(\omega_p t + \phi(t))$$

A eficiência da sinalização MDDF pode ser aumentada se, a uma mudança de fase forem associados dois ou mais bits (*i.e.,* a cada baud correspondem dois ou mais bits). Por exemplo, se forem usados quatro valores para o deslocamento de fase, cada um deles vai corresponder

a um *dibit*, de acordo com a tabela abaixo. Estes são os valores especificados para o modem padrão CCITT V.22bis. A Figura 2.35 mostra um sinal digital e o sinal modulado em fase que o representa.

dibit	$\Delta\phi$
00	90^o
01	0^o
10	180^o
11	270^o

Da mesma forma, tribits podem ser associados a mudanças de 45^o e tetrabits a mudanças de 22.5^o e assim por diante. Devido a dificuldade em recuperar o sinal original quando as variações de fase são pequenas, em geral não são usadas variações menores que 22.5^o com modulação MDDF pura.

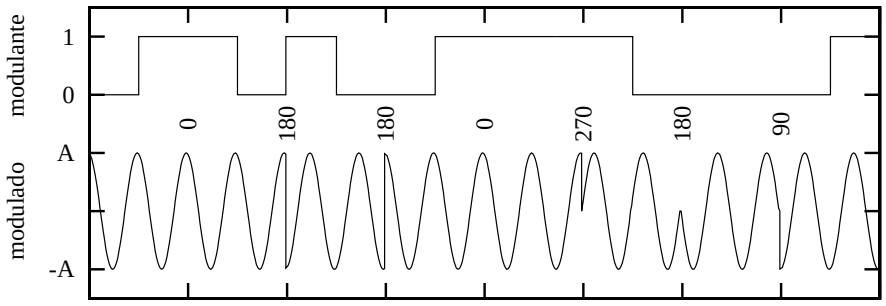


Figura 2.35: Sinal digital e sinal analógico com modulação diferencial em fase.

Modulação em Amplitude e Quadratura de Fase

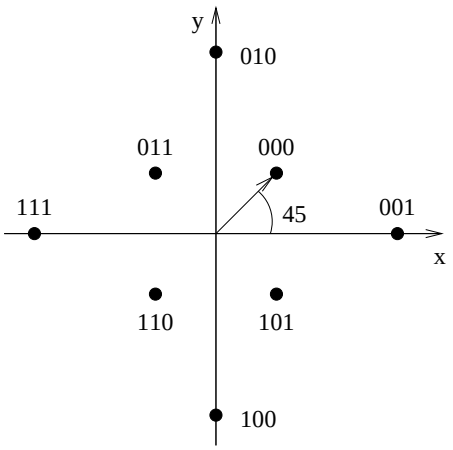
Uma eficiência de modulação ainda maior que aquela possível com MDDF pode ser obtida modulando-se a fase e a amplitude da portadora. Esta técnica é chamada de *modulação em amplitude e quadratura de fase* (MAQF). O nome em Inglês é "quadrature amplitude modulation" (QAM). A "quadratura" se refere a diferença de fase de 90^o entre as duas portadoras usadas na modulação: as funções $\text{sen}(t)$ e $\cos(t)$.

Para uma portadora de freqüência f_p e sinal modulante $\phi(t)$, definido como na modulação em fase (*i.e.,* $\Delta_0 = 0$ e $\Delta_1 = \pi$), o sinal modulado $S_{MAQF}(t)$ é descrito por

$$\begin{aligned} \phi(t) &= \pi d(t-1) \\ S_{MAQF}(t) &= \cos(\omega_p t + \phi(t)) \\ &= \cos\phi(t)\cos(\omega_p t) - \text{sen}\phi(t)\,\text{sen}(\omega_p t) \\ &= x\cos(\omega_p t) - y\,\text{sen}(\omega_p t) \end{aligned}$$

Usando-se valores adequados para x e y podem ser geradas inúmeras combinações de fase e amplitude, cada uma delas correspondendo a um *n-bit*. Por exemplo, o modem padrão CCITT V.29 modula tetrabits com diferenças de fase de 45^o e duas amplitudes. Este modem também transmite dados a 7200 bps a uma taxa de sinalização de 2400 baud com uma portadora f_p de 1700 Hz, onde cada símbolo (baud) corresponde a um tribit. A Figura 2.36 mostra a *constelação* de valores x e y especificados para esta velocidade de transmissão. Para gerar o símbolo (baud) correspondente ao tribit **000**, o modulador deve gerar o sinal correspondente a $(x, y) = (1, 1)$ *i.e.,* amplitude $\sqrt{2}$ e fase 45^o. Para o tribit **111**, $(x, y) = (-3, 0)$ *i.e.,* amplitude 3 e fase 180^o. Os valores de x e y são usados para modular a amplitude de duas portadoras de freqüência f_p defasadas

em 90º *i.e.,* $x\cos(\omega_p)$ e $y\,\mathrm{sen}(\omega_p)$. O circuito que gera os valores de x e y é chamado de *gerador de quadratura* porque as duas portadoras estão a 90º uma da outra.



tribit	amplitude	fase
000	1.42	45
001	3	0
010	3	90
011	1.42	135
100	3	270
101	1.42	315
110	1.42	225
111	3	180

Figura 2.36: Constelação do modem padrão V.29 transmitindo a 7200 bps.

Na demodulação, após filtragem e condicionamento, o sinal $S_{MAQF}(t)$ é processado em dois caminhos diferentes. O sinal $x(t)$ é recuperado indiretamente pela multiplicação de $S_{MAQF}(t)$ por $\cos(\omega_r t)$, onde $\omega_r = \omega_p$.

$$\begin{aligned} S_{MAQF}(t) &= \cos(\omega_p t + \phi(t)) \\ S_{MAQF}(t)\cos(\omega_r t) &= \cos(\omega_p t + \phi(t))\cos(\omega_r t) \\ &= 1/2[\cos(\omega_r t + \phi(t) - \omega_r t) + \cos(2\omega_r t + \phi(t))] \\ &= 1/2[\cos(\phi(t)) + \cos(2\omega_r t + \phi(t))] \\ &= x'(t) + 1/2\cos(2\omega_r t + \phi(t)) \end{aligned} \tag{2.25}$$

Estas derivações baseiam-se nas igualdades

$$\begin{aligned} \cos(a)\cos(b) &= 1/2[\cos(a-b) + \cos(a+b)] \\ \cos(a)\,\mathrm{sen}(b) &= 1/2[\,\mathrm{sen}(a-b) + \,\mathrm{sen}(a+b)] \end{aligned}$$

Da mesma forma, o sinal $y(t)$ é recuperado indiretamente pela multiplicação de $S_{MAQF}(t)$ por $\mathrm{sen}(\omega_r t)$.

$$\begin{aligned} S_{MAQF}(t)\,\mathrm{sen}(\omega_r t) &= \cos(\omega_p t + \phi(t))\,\mathrm{sen}(\omega_r t) \\ &= 1/2[\,\mathrm{sen}(\omega_r t + \phi(t) - \omega_r t) + \,\mathrm{sen}(2\omega_r t + \phi(t))] \\ &= 1/2[\,\mathrm{sen}(\phi(t)) + \,\mathrm{sen}(2\omega_r t + \phi(t))] \\ &= y'(t) + 1/2\,\mathrm{sen}(2\omega_r t + \phi(t)) \end{aligned} \tag{2.26}$$

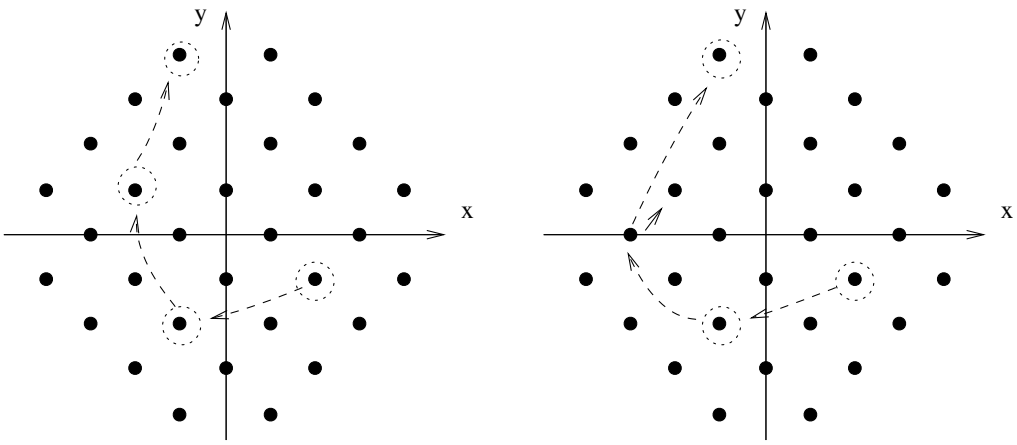
Os termos em $2\omega_p t$ nas Equações 2.25 e 2.26 são as segundas harmônicas de $\omega_r t$ produzidos na demodulação e são eliminados através de filtragem. De maneira similar à modulação em amplitude e fase, os termos $x' = 1/2\cos(\phi(t))$ e $y' = 1/2\,\mathrm{sen}(\phi(t))$ são níveis de tensão com freqüência zero. A informação digital pode ser recuperada convertendo-se os valores de $1/2\cos(\phi)$ e $1/2\,\mathrm{sen}(\phi)$ para os x e y que correspondem aos tribits originais.

Modulação por Codificação em Treliça

A modulação em amplitude e fase permite a transmissão a taxas de até 9600 bps. Acima desta velocidade, o número de pontos na constelação de sinais do modem fica muito grande e

a discriminação entre as várias fases e amplitudes se torna cara e complexa. Uma variação da modulação MAQF consiste em usar uma constelação com pontos redundantes. O mapeamento de n-bits nos pontos da constelação é feito de tal forma que apenas algumas seqüências de pontos da constelação representam seqüências válidas de bits transmitidos. O transmissor gera uma seqüência de símbolos (pontos da constelação) que corresponde a uma dada seqüência de bits de dados. Ao trafegar pelo meio físico, o sinal sofre distorções e alguns dos símbolos podem ser recuperados erroneamente pelo receptor. Na recepção, o modem é capaz de "adivinhar" a seqüência correta e corrigir muitos dos erros ocorridos no meio físico. Esta técnica é chamada de *modulação por codificação em treliça* (MCT) porque a constelação lembra uma treliça ("trellis coded modulation" ou TCM).

A Figura 2.37 mostra a constelação de pares [amplitude, fase] de um modulador MCT. A figura também mostra a transmissão de uma seqüência de quadrabits e a recepção da seqüência com um dos símbolos recuperado erroneamente. Por causa da codificação com redundância, o modem receptor consegue recuperar o erro e entregar a seqüência correta de quadrabits ao terminal. Neste exemplo, modem padrão CCITT V.32, para cada quatro bits de dados, são transmitidos cinco bits pela linha. O valor de quinto bit é calculado com base nos valores de dois bits do quadrabit a ser transmitido e de dois bits do quadrabit anterior. A grande vantagem da modulação MCT consiste em um ganho de 4dB a 6dB na relação sinal-ruído da transmissão – isso corresponde a uma taxa de erros 1000 vezes menor: de $t_E \leq 1 \times 10^3$ para $t_E \leq 1 \times 10^6$. Esse ganho na relação sinal-ruído permite a operação com velocidades de transmissão maiores que as possíveis com as outras técnicas de modulação.



(a) Seqüência transmitida (b) Seqüência recebida

Figura 2.37: Constelação de modem MCT V.32 transmitindo a 9600 bps.

Teorema de Shannon e Limite de Velocidade de Transmissão

A Seção 2.3.7 enuncia o Teorema de Shannon. Após o estudo das técnicas de modulação, um exame mais detalhado das implicações daquele teorema se torna possível. Para um canal com faixa de passagem F [Hz] e relação sinal-ruído de $S/(N_0F)$, o Teorema de Shannon estabelece que a taxa de transmissão máxima possível $T_{\max}$ [bps] é

$$T_{\max} = F \log_2(1 + S/(N_0F)) \tag{2.27}$$

onde F é a faixa de passagem do canal em hertz, S é o nível de potência máxima de sinal admissível pelo receptor [watts] e N_0F é a potência do ruído térmico [watts] para a faixa de passagem F. Segundo o Teorema de Nyquist, se k bits podem ser codificados em cada símbolo, Fk bits podem ser transmitidos pelo canal (Seção 2.3.7).

A interpretação do Teorema de Shannon que segue é baseada em [BG92]. Deve existir um espaçamento mínimo entre os pontos vizinhos de uma constelação MAQF para permitir a discriminação dos símbolos em presença de ruído. Isso significa que a cada ponto deve corresponder uma fração da área total da constelação e esta área é proporcional a N_0F. A área total ocupada pela constelação é proporcional à potência máxima S do sinal. O número de pontos da constelação 2^k é proporcional a $S/(N_0F)$ e k é proporcional a $\log_2(S/(N_0F))$ *i.e.,* k é proporcional à relação sinal-ruído do meio físico. Quanto melhor a qualidade da linha, maiores são a relação sinal-ruído, os valores possíveis para k e conseqüentemente, as taxas de transmissão possíveis. O limite máximo de 30 Kbps para modulação MAQF pura, discutido na Seção 2.3.7, pode ser aproximado, ou até ultrapassado, por modems MCT porque esta técnica de modulação propicia um ganho na relação sinal-ruído de 4 a 6dB com relação a modulação em amplitude e fase.

Cancelamento de Eco

Para que a transmissão duplex seja possível, são necessários dois canais distintos para transportar simultaneamente os sinais emitidos pelos dois ETDs. Em sistemas com *multiplexação em freqüência*, cada canal consiste de uma faixa do espectro de freqüências e a separação entre canais decorre do uso de filtros que bloqueiam os sinais indesejáveis (*i.e.,* os sinais do outro canal). Outra técnica, chamada de *transmissão a quatro fios*, consiste no uso de dois pares de fios onde cada par transporta os sinais de um canal. A multiplexação em freqüência tem a desvantagem de oferecer dois canais de faixa mais estreita e portanto, com taxas máximas de transmissão menores que as possíveis se toda a largura de faixa do meio físico fosse usada, como mostra a Figura 2.33 (pág. 44). A desvantagem da transmissão duplex a quatro fios é o custo de duas linhas ao invés de uma só. Os modems mais sofisticados e de projeto recente empregam a técnica de *cancelamento de eco* para a transmissão duplex a dois fios.

Quando se fala ao telefone, o sinal transmitido pela linha é atenuado e retornado ao alto-falante. Esse sinal é chamado de *eco* e permite que quem fala também ouça o que está dizendo. Pelo menos dois ecos são gerados nos sistemas telefônicos. O "primeiro" eco é gerado propositalmente no próprio aparelho, para que se ouça o que se fala. O "segundo" eco decorre da reflexão do sinal no aparelho do outro lado da linha. Como o segundo eco viaja uma distância maior, há um atraso entre a emissão de um som e a percepção de seu segundo eco. Em ligações de longa distância, o atraso e a amplitude do segundo eco podem ser tais que uma conversação se torna difícil. A Figura 2.38 mostra dois aparelhos telefônicos T1 e T2, e a geração de ecos pelas híbridas H1 e H2. As híbridas fazem parte dos próprios telefones e são transformadores especiais que permitem o retorno do primeiro eco ao mesmo tempo que separam os sinais provenientes do microfone local dos sinais provenientes do microfone remoto.

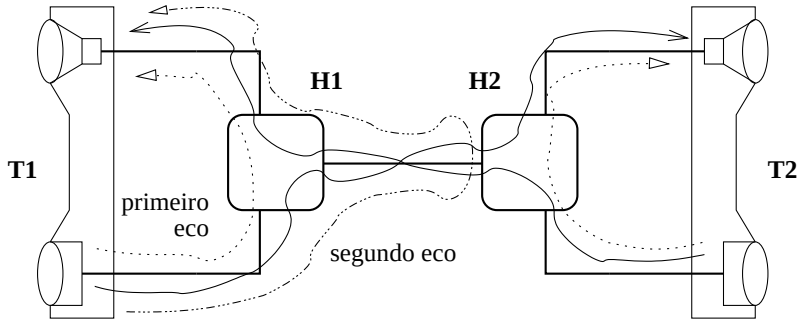


Figura 2.38: Fontes de eco em linhas telefônicas.

O sistema auditivo consegue tolerar bastante bem a ocorrência de eco desde que o atraso entre a emissão do som e a percepção do eco seja menor que uns poucos milisegundos (de 15 a 20ms).

Nestas condições, o sistema auditivo se adapta de tal forma a cancelar o eco. Por exemplo, uma conversação num local com muita reverberação (*e.g.* uma garagem) normalmente se inicia com pouca compreensão do que é dito pela outra pessoa. Após alguns segundos, o sistema auditivo se adapta e efetivamente cancela a maior parte dos ecos. O nível de compreensão melhora porque os ecos dos sons produzidos por quem fala são ignorados enquanto que o sistema auditivo se sintoniza nos sons produzidos pela outra pessoa.

Assim como em conversas telefônicas, os modems também recebem ecos do sinal que transmitem. Se o *circuito de cancelamento de eco* do modem for capaz de eliminar os ecos do sinal que ele próprio produz, enquanto que os sinais produzidos pelo modem remoto são amplificados, os dois modems podem transmitir ao mesmo tempo, estabelecendo assim comunicação duplex [Dou90]. O circuito de cancelamento de eco mede o atraso e as amplitudes dos ecos do sinal que o modem está injetando na linha. O sinal que o modem recebe pela linha contém ecos do sinal que ele próprio emite, além do sinal transmitido pelo modem remoto. Os ecos podem ser eliminados se cópias do sinal emitido, devidamente atrasadas e atenuadas forem subtraídas do sinal recebido pela linha. Este tipo de tratamento de sinais só se tornou possível com o advento de circuitos com integração em muito larga escala (VLSI) que permitem a implementação em silício de técnicas sofisticadas de processamento digital de sinais.

2.6 Recomendações CCITT Série V

O ITU-CCITT é o organismo responsável pela publicação das recomendações que especificam as características operacionais de modems tais como as taxas de transmissão e sinalização e o tipo de modulação empregado. Estas recomendações tornam-se efetivamente padrões que permitem a um certo fabricante projetar equipamentos com a garantia de compatibilidade com os equipamentos de outros fabricantes.

O CCITT publicou uma série de especificações de modems e algumas das recomendações da *Série V* são mostradas na Tabela 2.4. A menos do V.22bis, todos os padrões anteriores ao V.32 são praticamente obsoletos. Pode-se comprar um modem de alta velocidade (*i.e.,* $\geq$ 9600 bps) por uma fração do preço de um computador pessoal. Os padrões V.21 e V.22 foram publicados no início dos anos 50. O padrão V.22bis é de meados da década de 60. O padrão V.32 é do início da década de 80 e os modems de mais de 19200 bps são do início da década de 90.

A primeira coluna da Tabela 2.4 mostra o número de referência da recomendação. A segunda coluna indica a taxa de transmissão definida para o tipo de modem. Algumas das recomendações especificam mais de uma taxa de transmissão. Se as condições do canal forem ruins (*e.g.* demasiado ruído) o equipamento pode reduzir a taxa transmissão. A terceira coluna "duplx" indica a duplexidade do modem: 'd' se duplex, 's' se semi-duplex. A quarta coluna "canais" mostra o mecanismo empregado para a separação de canais na transmissão duplex. A separação pode ser obtida com o uso de filtros, com o uso de quatro fios '4f' ou por cancelamento de eco 'c. eco'. A quinta coluna, "mod", indica o tipo de modulação empregado. A sexta coluna "codif" mostra o número de bits transmitido para cada símbolo, isto é, a relação entre a taxa de transmissão [bps] e a taxa de sinalização [baud]. A sétima coluna "sincr" indica se o modem opera em modo síncrono 's', assíncrono 'a', ou multiponto 'mp'. A oitava coluna mostra o tipo de "linha" recomendada: 'disc'ada ou 'priv'ada.

Padrão	bps	duplx	canais	modul	codif	sincr	linha
V.21	300	d	filtros	MF	1:1	s,a	disc
V.22	1200	d	filtros	MDDF	2:1	s,a	disc
	600	d	filtros	MDDF	1:1	s,a	disc
V.22bis	2400	d	filtros	MAQF	4:1	s,a	disc
	1200	d	filtros	MAQF	2:1	s,a	disc
V.23	1200	s	–	MF	1:1	s,a	disc
	600	s	–	MF	1:1	s,a	disc
V.26	2400	d	4f	MDDF	2:1	s,mp	disc
V.26bis	2400	s	–	MDDF	2:1	s	disc
	1200	s	–	MDDF	1:1	s	disc
V.26ter	2400	s, d	c. eco	MDDF	2:1	s,a	disc
	1200	s, d	c. eco	MDDF	1:1	s	disc
V.27	4800	s, d	2f, 4f	MDDF	3:1	s	priv
V.27bis	4800	s, d	2f, 4f	MDDF	3:1	s	priv
	2400	s, d	2f, 4f	MDDF	2:1	s	priv
V.27ter	4800	s	–	MDDF	3:1	s	disc
	2400	s	–	MDDF	2:1	s	disc
V.29	9600	s, d	4f	MAQF	4:1	s	priv
	7200	s, d	4f	MDDF	3:1	s	priv
	4800	s, d	4f	MAQF	2:1	s	priv
V.32	9600	d	c. eco	MAQF	4:1	s	disc
	9600	d	c. eco	MCT	5:1	s	disc
	4800	d	c. eco	MAQF	2:1	s	disc
V.32bis	14400	d	c. eco	MAQF	6:1	s	disc
V.33	14400	d	4f	MCT	6:1	s	priv
	12200	d	4f	MCT	5:1	s	priv
V.34	19200	d	c. eco	MCT	7:1	s	disc
	28800	d	c. eco	MCT	8:1	s	disc

Tabela 2.4: Recomendações Série V do CCITT.

2.7 Bibliografia e Exercícios

O texto clássico sobre sistemas de comunicação é Taub & Schilling [TS86]. Livros de Matemática para Engenharia tratam Séries de Fourier em profundidade. O material sobre resistência, capacitância e indutância pode ser encontrado em livros de circuitos elétricos, como [Bob87]. Outra formulação do Teorema de Nyquist, conhecida como *Teorema da Amostragem* é discutida em [Sta91].

Exercícios

Ex. 2.1 [§ 2.2.1, 2.3.7] Calcule a potência das componentes harmônicas dos sinais descritos pela Equação 2.14 para $R = 1/2, 1/4, 1/8, 1/16$. Usando estes valores, trace um gráfico semelhante ao da Figura 2.10. Com base neste gráfico, o que se pode concluir sobre a largura de faixa de sinais digitais? Como estes resultados se relacionam com aqueles previstos pelos Teoremas de Nyquist (Eq. 2.23) e Shannon (Eq. 2.24)?

Ex. 2.2 [§ 2.3.6] Um experimento interessante consiste em operar uma calculadora eletrônica próxima a um receptor de rádio AM. O que ocorre quando uma tecla da calculadora que é apertada? Que tipo de ruído é esse?

Ex. 2.3 [§ 2.3.7] Algumas emissoras de rádio transmitem a mesma programação simultameamente em AM e em FM. Um canal de radiofrequência para AM tem uma largura de faixa de aproximadamente 5 kHz (200 a 5 kHz) enquanto que um de FM tem largura de faixa de aproximadamente 15 kHz (50 a 15 kHz). Ignorando os diferentes níveis de ruído, o que se pode dizer sobre a transmissão de música clássica nesses dois meios? E sobre a transmissão de vozes masculinas?

Ex. 2.4 [§ 2.3.6, 2.5] Ainda na comparação de transmissões de rádio em AM e FM, considere a transmissão de vozes masculinas (ignore portanto a limitação de faixa de passagem). Estude a Figura 2.24, pág 34, e com base na codificação da informação empregada (AM e FM), justifique a diferença nos níveis de ruído das duas formas de transmissão.

Ex. 2.5 [§ 2.5, 2.6] Compare a eficiência da transmissão a 9600 bps com modems V.29 e V.32 com MCT. Qual a vantagem do V.32 nesta velocidade?

Capítulo 3

Nível Físico

Este capítulo descreve algumas das tecnologias comumente empregadas no Nível Físico de redes de dados. Exemplos representativos dos equipamentos e técnicas empregados em comunicação serial de dados são brevemente discutidos na Seção 3.1. Particularmente importante é a multiplexação estatística, que é uma técnica de comunicação que permite o uso simultâneo de um único canal físico por vários canais lógicos, permitindo um uso mais racional e econômico do meio físico.

A técnica de transmissão serial assíncrona é apresentada com algum detalhe por ser amplamente empregada para interligar equipamentos através de linhas telefônicas. Esta é uma alternativa de baixo custo para interligar computadores pessoais, por exemplo. A Seção 3.4 descreve uma interface padronizada de uso quase-universal para a conexão entre terminais e ECDs: a interface EIA-232-D ou CCITT V.24-V.28. O funcionamento desta interface é simples o bastante para um primeiro estudo de caso, e ao mesmo tempo demonstra vários dos problemas a ser enfrentados na implementação de interfaces de comunicação digitais.

A Seção 3.5 descreve o funcionamento e a programação de um dispositivo de E/S que implementa a parte funcional da interface EIA-232-D. Um Controlador de Interface Serial Assíncrona (CISA) faz o acoplamento elétrico entre os sinais empregados entre ETD e ECD bem como permite ao processador controlar o fluxo de dados entre sua memória e o equipamento ligado à outra ponta do enlace. Os componentes de um CISA são descritos bem como o inter-relacionamento destes componentes com o programa que permite a operação do CISA como um dispositivo periférico ao processador. O capítulo se encerra com uma versão simplificada de um programa de comunicação através de linha serial e interfaces ETD-ECD tipo EIA-232-D. Este programa relaciona as funções de envio e recepção de caracteres com a programação do CISA descrito anteriormente.

3.1 Equipamentos

Há uma enorme diversidade de equipamentos que empregam alguma forma de comunicação de dados. O espectro vai desde controladores de bombas de postos de gasolina até supercomputadores. Esta variedade de aplicações também se reflete na variedade de equipamentos empregados em sistemas de comunicação de dados. O que segue contém uma pequena amostra.

3.1.1 Terminais

Em muitas aplicações comerciais, um grande número de terminais fornecem e recebem informações de um computador de grande porte. Estes terminais podem ser máquinas de ponto de venda, onde são digitados os dados da operação, estes são enviados ao computador que os processa e responde completando a operação. Este tipo de terminal é chamado de *terminal*

burro por causa de sua limitada capacidade de processamento. Em geral, terminais burros são controlados pelo computador a que estão ligados, onde o computador controla o posicionamento do cursor, desenha telas, etc. Um exemplo detalhado com o código que implementa as funções de controle de terminais em ambiente similar ao Unix pode ser encontrado em [Com84].

Em alguns sistemas comerciais como bancos, por exemplo, as transações do dia são processadas localmente na agência e um resumo destas é transmitido para o centro de processamento da matriz ao final do expediente. Do ponto de vista do sistema de comunicação de dados, o computador da agência é um *terminal inteligente* já que este possui alguma/grande capacidade de processamento e armazenamento.

A Figura 3.1 mostra um enlace de dados entre duas máquinas interconectadas por uma linha telefônica. O terminal (ETD) se conecta ao modem (ECD) através da *interface ETD-ECD*. Esta interface obedece a um protocolo de baixo nível que controla o tráfego da informação entre ETD e ECD. Este tipo de protocolo somente permite que o terminal inicie suas transmissões se o modem estiver pronto para transmitir dados pela linha telefônica, por exemplo. Os dois ECDs se comunicam através do meio físico de acordo com um mesmo protocolo de comunicação, que define a velocidade de transmissão e a codificação dos sinais digitais em sinais adequados à transmissão por linha telefônica.

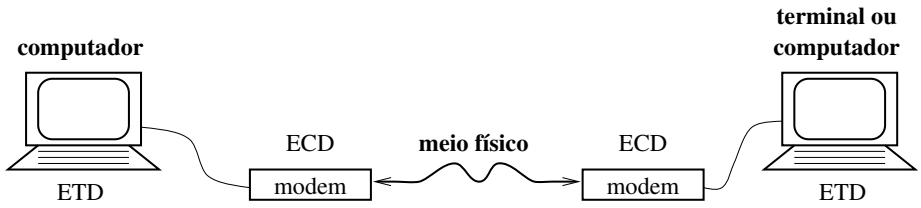


Figura 3.1: Enlace ponto-a-ponto simples.

3.1.2 Controladores de terminais

Em sistemas onde um computador se comunica com grande número de terminais pode ser necessário o emprego de um *controlador de terminais* para aliviar o processador das operações de baixo nível relacionadas com a comunicação de dados. Um controlador é um periférico especializado que se encarrega de efetuar todas as operações de amostragem, controle, detecção e recuperação de erros nas linhas de comunicação sob sua responsabilidade. A interface entre controlador e processador trata apenas de blocos de dados a nível de aplicativo como a consulta a um sistema de venda de passagens aéreas, por exemplo. Controladores de terminais são também conhecidos como *canais* de E/S serial. A Figura 3.2 mostra uma instalação contendo um controlador de terminais. Note que alguns dos terminais estão em sítios distantes ao computador e por isso aqueles são conectados ao computador de através de ECDs e linhas de comunicação.

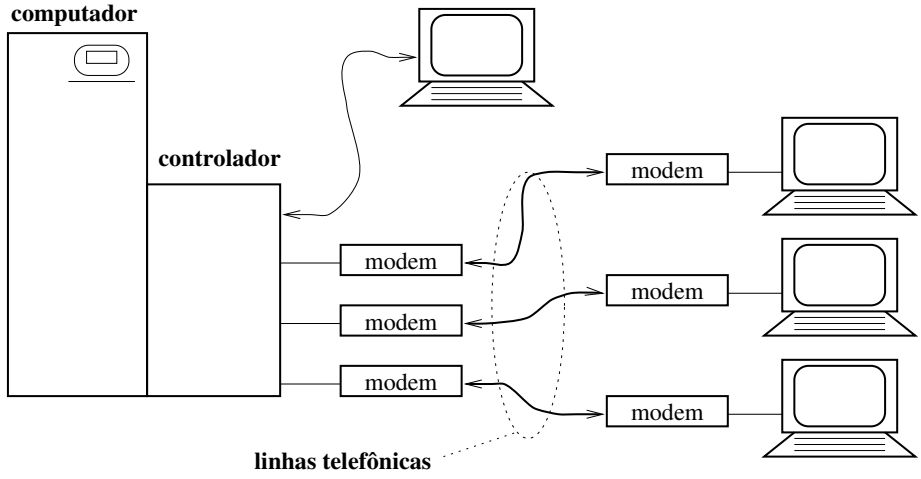


Figura 3.2: Sistema de comunicação com um controlador de terminais.

Multiplexação por divisão de tempo

No sistema mostrado na Figura 3.2, existe uma linha que conecta cada um dos terminais ao computador central. Suponha que os três terminais remotos estão no mesmo sítio. Se a soma das taxas de transmissão dos três terminais for menor que a capacidade de uma linha, eles poderiam compartilhar a linha pois aquela é um recurso caro. Um *multiplexador* é mais ou menos equivalente a um funil: os dados dos três terminais são combinados pelo multiplexador local, transmitidos através de uma única linha e separados novamente pelo multiplexador remoto. Desta forma, dois ou mais pares de terminais se comunicam através de uma única linha física. A Figura 3.3 mostra uma instalação com 4 terminais remotos ligados ao computador através de uma única linha telefônica compartilhada entre os quatro.

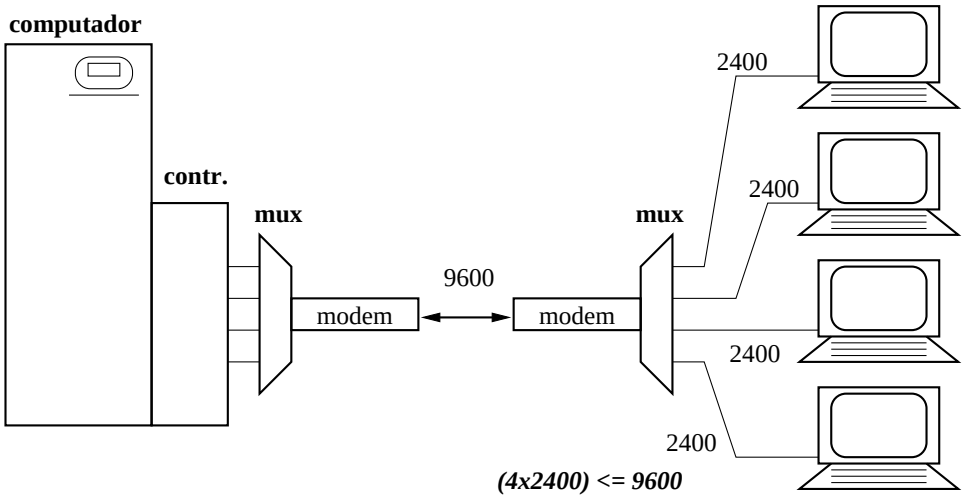


Figura 3.3: Compartilhamento de linha física.

Alguns concentradores de terminais são baseados na *multiplexação por divisão de tempo* porque cada terminal tem direito a transmitir, periodicamente, durante uma fatia fixa de tempo. A Figura 3.4 mostra quatro terminais ligados a um multiplexador por divisão de tempo. A somatória das taxas de transmissão das entradas do multiplexador deve ser menor ou igual a capacidade da linha de transmissão. O multiplexador amostra todas as suas entradas periodicamente e cada terminal transmite quando chega a sua vez. Uma analogia para a multiplexação por

divisão de tempo consiste de um trem onde os vagões são numerados seqüencialmente. Para N terminais, cada vagão cujo número de seqüência $mod\, N = i$ carrega dados do terminal t_i, ou seja, cada i-ésimo vagão transporta dados do terminal t_i.

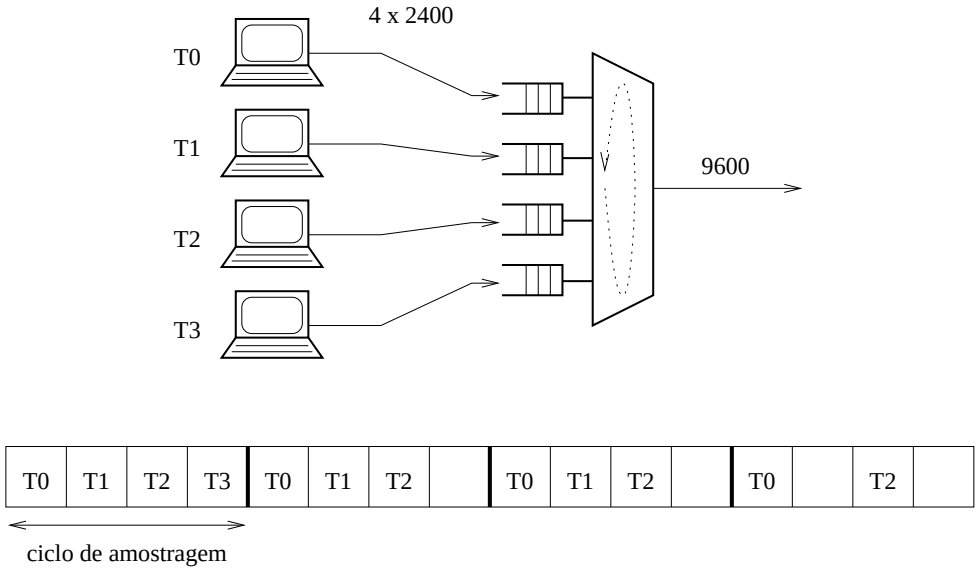


Figura 3.4: Transmissão via multiplexador por divisão de tempo.

3.1.3 Multiplexação estatística

A maior desvantagem dos multiplexadores por divisão de tempo é que o tempo disponível a cada um de N terminais é fixo e invariável. Se um terminal não tem o que transmitir, a capacidade de canal a ele alocada não pode ser utilizada por nenhum outro terminal. Isto é remediado com um *multiplexador estatístico* ou *statmux*. Em transmissões via statmux, a capacidade disponível para cada um dos terminais depende do tráfego gerado por eles. Terminais que transmitem pouco tem menores oportunidades de transmitir enquanto que terminais que geram muitas mensagens podem usar uma fração maior da capacidade do canal. A Figura 3.5 mostra quatro terminais ligados a um multiplexador estatístico. A somatória das taxas de transmissão dos terminais pode ser maior que a capacidade da linha de transmissão.

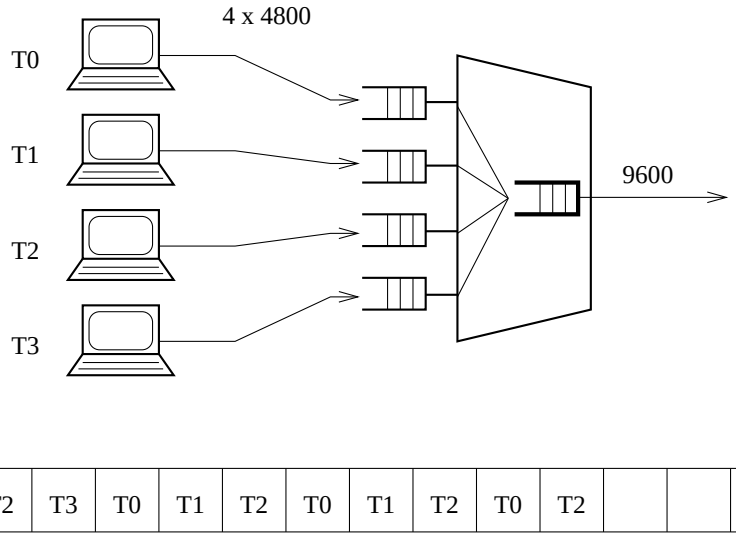


Figura 3.5: Transmissão via multiplexador estatístico.

A capacidade disponível a cada terminal é re-calculada periodicamente e a capacidade disponibilizada a um terminal é re-ajustadas em função de seu uso do canal. De volta a analogia do trem, no caso da multiplexação por divisão de tempo, quando o terminal t_i não tem dados a transmitir, os vagões do trem de dados alocados a t_i viajam vazios. Com multiplexação estatística isto pode ser remediado fazendo com que os vagões sejam marcados com uma placa que indica a quem sua carga pertence. Assim, quando um terminal está inativo, seus vagões podem ser usados pelos demais.

Tipicamente, um terminal operado por humanos envia e/ou recebe dados 25% do tempo. Esse tráfego não é contínuo e sua intensidade varia muito com o passar do tempo. Assim, um multiplexador estatístico pode transmitir o tráfego gerado por quatro ou cinco terminais operando a 4800 bps através de uma linha de 9600 bps sem que os usuários percebam maiores flutuações na qualidade do serviço. Se o tráfego gerado por algum dos terminais for contínuo, multiplexadores estatísticos são contra-indicados porque eles introduzem ineficiência na transmissão sem o benefício correspondente ao compartilhamento da linha.

A Figura 3.6 mostra a diferença entre os dois tipos de multiplexação. Na figura, quatro terminais (T_0 a T_3) transmitem um certo número de mensagens ou blocos de caracteres. Na multiplexação por divisão de tempo, uma "fatia" de tempo é alocada a cada um dos terminais. Se um terminal não tem nada a transmitir, a fatia correspondente é desperdiçada. Na multiplexação estatística, cada bloco de caracteres é precedido do endereço do terminal que o transmitiu. Isso representa informação adicional com relação a multiplexação por divisão de tempo. Por outro lado, a utilização da linha é maior pois, enquanto algum terminal estiver pronto a transmitir, poderá fazê-lo imediatamente se houver alguma fatia vazia, ao invés de ter que esperar pela sua vez na seqüência.

T0	T1	T2	T3	T0	T1	T2		T0	T1	T2		T0		T2	

(a) Multiplexação por divisão de tempo.

T0	T1	T2	T3	T0	T1	T2	T0	T1	T2	T0	T2				

(b) Multiplexação estatística.

Figura 3.6: Multiplexação por divisão de tempo e multiplexação estatística.

Num multiplexador estatístico, existem duas maneiras de escolher qual é o próximo terminal a transmitir, relacionadas com a implementação das filas nas entradas do multiplexador. A primeira consiste em amostrar seqüencialmente as N filas de entradas do multiplexador. Quando uma fila de entrada está vazia, o statmux examina a próxima entrada na seqüência. Filas que ficam vazias com freqüência passam a ser examinadas a intervalos maiores. A segunda maneira consiste em inserir todos as mensagens por transmitir numa única fila, *i.e.,* na Figura 3.5, as quatro filas de entrada são organizadas em uma única fila. As mensagens são transmitidas na ordem em que foram emitidas pelos terminais.

Existe a possibilidade de que, durante intervalos de tempo, os terminais gerem mais tráfego do que a capacidade da linha. Neste caso, as filas mostradas na Figura 3.5 podem ficar cheias e algo deve ser feito para evitar a perda de mensagens. A solução mais simples consiste em descartar novas mensagens se as filas estão cheias. Neste caso, o programa de comunicação deve conter algum mecanismo para a retransmissão de mensagens descartadas. A solução mais interessante emprega o mecanismo descrito abaixo para controlar o fluxo de dados entre terminais

e multiplexador.

Controle de fluxo Xon–Xoff Devido a diferença nas velocidades de processamento entre terminais e computadores, existe a necessidade de alguma forma de controle do fluxo de dados entre estes. Alguns sistemas usam um método simples de controle de fluxo, chamado de *Xon-Xoff*. Quando o ETD está apto a receber dados, este transmite o caracter 'XON' (ASCII 'DC1'). Quando o ETD necessita interromper o fluxo de dados que está recebendo, transmite o caracter 'XOFF' (ASCII 'DC3'). Quando um ETD recebe um 'XOFF', este pára de transmitir até que receba um 'XON'. Dessa forma, ETDs conseguem manter a comunicação mesmo quando há grande disparidade nas capacidades de processamento entre eles. Em alguns sistemas, 'XOFF' é representado pelo comando CTRL-S e 'XON' pelo comando CTRL-Q. Métodos mais sofisticados de controle de fluxo são discutidos na Seção 4.2.

No caso de terminais ligados a um multiplexador, os dois, terminal e multiplexador, se comportam como ETDs. Considere um conjunto de terminais ligados a um multiplexador e este a um modem. O modem remoto está ligado a um computador de grande porte. Se uma das filas de transmissão do multiplexador está próxima de sua capacidade máxima, o multiplexador envia um XOFF para o terminal correspondente, que cessa suas transmissões. Se a fila de recepção do computador remoto está próxima de sua capacidade máxima, este envia um XOFF para o multiplexador, que cessa suas transmissões. Quando há espaço suficiente na(s) fila(s), os terminais são liberados pela emissão de um XON. Se o fluxo recebido por um terminal é muito alto, o terminal envia um XOFF para o multiplexador que o repassa para o computador.

3.1.4 Multiplexação por divisão de freqüência

A terceira forma de multiplexação é chamada de *multiplexação por divisão de freqüência* porque a faixa de passagem do meio físico é dividida num certo número de canais de faixa mais estreita. Cada terminal transmite através de um destes canais, de maneira similar aos canais de televisão aberta ou por cabo, como mostra a Figura 3.7. Cada transmissor emite sinais dentro de sua faixa de freqüências e cada receptor emprega filtros para selecionar o canal que deseja receber.

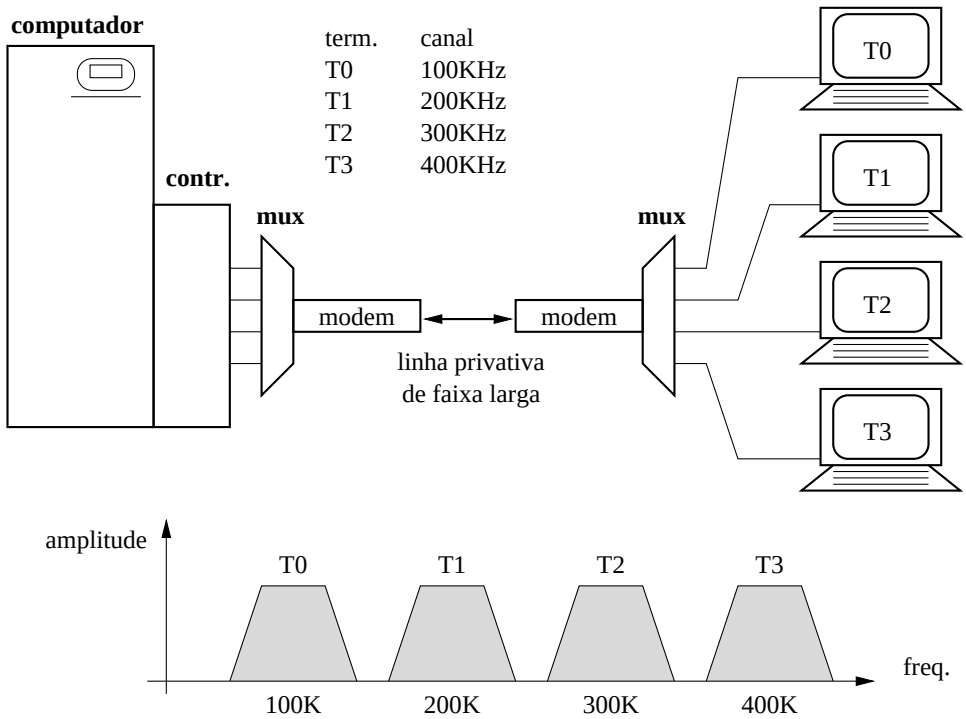


Figura 3.7: Multiplexação por divisão de freqüência.

3.2 Transmissão Serial Assíncrona

Os sistemas de transmissão de dados normalmente empregam o formato serial para a comunicação porque este formato usa, como mínimo, um par de fios para transportar os sinais contendo a informação digital. Internamente aos computadores, a informação é armazenada e processada em grupos 8 bits (ou bytes) quando se trata de texto ou grupos de 16, 32 ou 64 bits quando se trata de código executável, por exemplo. A transmissão de dados em paralelo, supondo códigos de 8 bits exige, no mínimo, 9 fios (8 de sinal e um de referência de tensão). Por medida de economia, a transmissão serial é preferida porque necessita de somente dois fios.

A Figura 3.8 mostra a conversão paralelo-serial efetuada em uma interface serial por ocasião da transmissão através de um par de fios. Na recepção, os dados recebidos em formato serial são re-convertidos para o formato paralelo. A figura, bastante simplificada, não mostra o fio de referência de tensão nem os circuitos de controle necessários ao correto funcionamento deste sistema. Os oito bits de um caracter são carregados em paralelo no registrador de deslocamento. A cada pulso do relógio de transmissão, o valor correspondente a um novo bit é emitido pela interface serial. No receptor, o relógio de recepção amostra o valor presente na linha e a cada pulso do relógio, desloca os bits amostrados pelas posições consecutivas do registrador. Quando, após oito pulsos do relógio de recepção, os oito bits do caracter original são recebidos, estes são apresentados em paralelo ao processador.

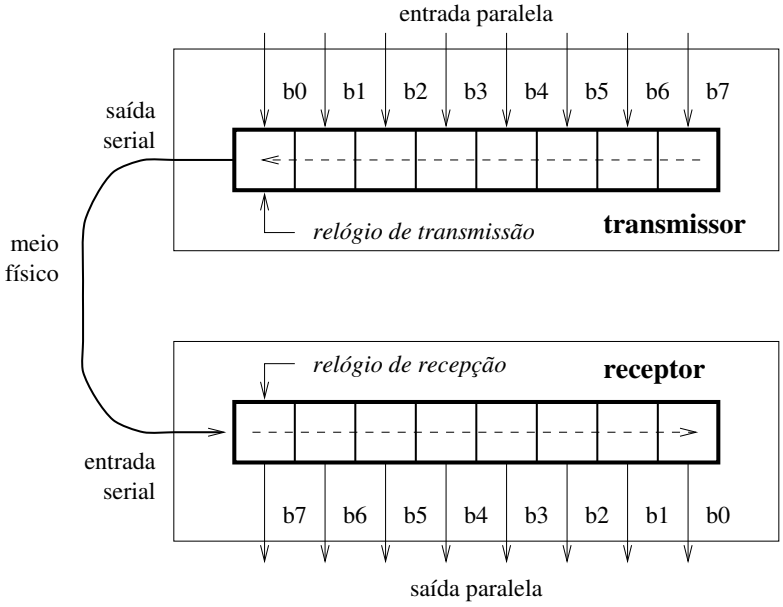


Figura 3.8: Conversão paralelo-serial, transmissão e re-conversão serial-paralelo.

O transmissor e o receptor devem estar de acordo quanto à taxa de sinalização a ser empregada nas transferências de dados entre eles. Isso significa que ambos devem empregar um sinal de relógio de mesma freqüência para emitir os bits na transmissão e para amostrá-los na recepção. Os circuitos lógicos da interface serial tomam amostras do sinal recebido na borda descendente dos pulsos do relógio de recepção, como mostra a Figura 3.9. Estas amostras devem ocorrer no centro do intervalo de duração de cada bit t_b ($t_b = 1/\text{taxa de transmissão}$) para que os sinais elétricos já estejam estáveis nas tensões correspondentes a 0 e 1. Na prática, as transições dos sinais entre os níveis correspondentes a 0 e 1 não são abruptas e bem definidas como mostra

a Figura 3.9. Assim, o ponto mais seguro para a amostragem dos bits pelo receptor é o centro do intervalo de bit t_b.

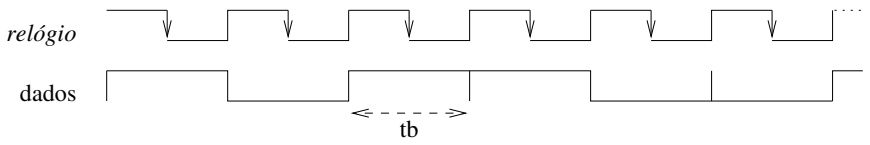


Figura 3.9: Amostragem de dados pela interface serial.

É possível que haja uma pequena diferença nas freqüências dos relógios de transmissão e de recepção. Valores típicos para a tolerância dos cristais usados para gerar os sinais de sincronismo são da ordem de 50 partes por milhão (ppm). No pior caso, a diferença de freqüências é de 100ppm, o que acarretaria em um bit amostrado erradamente a cada 5000 bits transmitidos pois a amostragem ocorre no centro do intervalo de bit t_b. Este tipo de erro é chamado de *erro de amostragem*. A Figura 3.10 mostra, com exagero, erros causados pela assincronia entre os relógios de transmissão e recepção. As partes sombreadas denotam o erro de amostragem causado pela diferença de freqüência entre o relógio de transmissão que cadencia a emissão dos bits e o relógio de recepção que os amostra nas posições erradas.

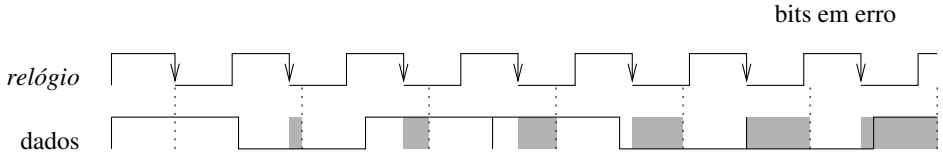


Figura 3.10: Erro de amostragem devido a assincronia de relógios.

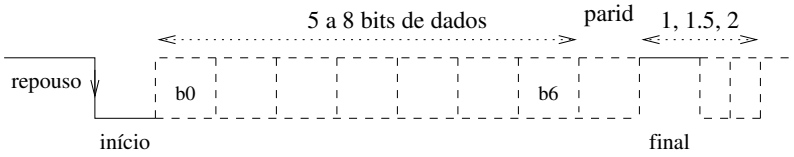
Por razões históricas, o nível de tensão ou corrente correspondente ao bit 0 é chamado de espaço ("space") e o nível de tensão correspondente ao bit 1 é chamado de marca ("mark"). Se a linha de comunicação está em repouso, ela é normalmente mantida no estado de marca. Estes nomes e convenções tem origem nos teletipos eletromecânicos e permanecem em uso até o presente.

No modo de transmissão assíncrono, o transmissor insere informação de sincronismo *a cada* caracter que ele envia pela linha serial. O enquadramento na transmissão assíncrona é determinado por um bit de início ("start bit") e um bit de final ("stop bit"), como mostra a Figura 3.11. Os bits que representam um caracter são emoldurados pelos bits de início e de final. A função do bit de início é sincronizar o relógio do receptor ao caracter que inicia no bit seguinte *i.e.,* a transição de marca para espaço no bit de início separa um caracter do anterior. A função do bit de final é colocar a linha num estado de repouso de forma a garantir uma transição de marca para espaço no próximo bit de início. Outra função do bit de final é permitir que o receptor aumente ou reduza sua cadência de amostragem para adequá-la à cadência do transmissor. Geralmente, o número de bits de final pode ser escolhido dentre um, um e meio ou dois intervalos t_b.

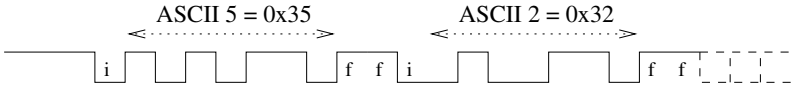
Não há nenhuma restrição quanto ao intervalo entre o envio de caracteres consecutivos, isto é, os caracteres são emitidos *assincronamente* pelo transmissor e é essa característica que dá o nome a esse modo de transmissão. A Figura 3.11 descreve graficamente o modo de transmissão assíncrona. Ao detectar a transição de 1 para 0, o receptor passa a amostrar o estado da linha nos intervalos determinados pela taxa de transmissão. Note que o bit menos significativo (b0) é o primeiro bit do caracter emitido pelo transmissor.

Existe a possibilidade de que um pulso de ruído cause um erro de amostragem pelo receptor. Isso ocorre quando o pulso ruído distorce a forma de onda do sinal transmitido de tal forma

que um 0 seja interpretado erroneamente como 1, e vice-versa. Um *bit de paridade* pode ser transmitido após um caracter, como mostra a Figura 3.11. O valor do bit de paridade (0 ou 1) é calculado de tal forma que o número total de bits em 1 entre os bits de início e de final seja par (ou ímpar). Por exemplo, ao transmitir o caracter ASCII correspondente a letra 'A'=0x41, o valor do bit de paridade deve ser 0 porque o número de 1's em 0x41 é par. O receptor re-calcula o valor do bit de paridade e compara o valor calculado com o valor recebido. Se os dois forem iguais, o caracter foi recebido sem erros. Mais detalhes na Seção 4.1.



(a) Enquadramento de um caracter



(b) Transmissão de "52", com dois bits de final

Figura 3.11: Enquadramento e transmissão assíncrona.

Uma deficiência da transmissão assíncrona decorre da detecção de um *falso bit de início*. Tal acontece se um pulso de ruído provoca a detecção de um falso bit de início *i.e.,* um pulso de ruído dispara o circuito de recepção, e a partir da falsa detecção, o receptor toma amostras da linha como se um caracter houvesse realmente sido enviado. A detecção de falso bit de início causa um ou mais *erros de enquadramento* porque o receptor não consegue distinguir quais bits são bits de início e de final e quais são bits de dados. Em geral, o receptor consegue recuperar o sincronismo sem muita demora. Se o intervalo entre dois caracteres consecutivos for maior que a duração de um caracter mais bits de início e final, o segundo caracter será amostrado corretamente. Mesmo que o espaçamento seja curto, após alguns caracteres o receptor volta a enquadrar os caracteres corretamente. Note que erros de amostragem também podem causar erros de enquadramento.

A transmissão assíncrona simplifica a implementação dos dispositivos de E/S porque os circuitos de elasticidade de relógio para intervalos de uns poucos bits são de projeto e construção relativamente fácil. Contudo, a transmissão assíncrona é ineficiente. Para cada octeto, no mínimo dois bits adicionais são enviados apenas para sincronizar transmissor e receptor. Isso corresponde a uma redução de 20% na taxa efetiva de envio. Em códigos que usam um menor número de bits por caracter, o desperdício é proporcionalmente maior.

3.3 Interface ECD-ETD

Para tornar as discussões que seguem um pouco mais concretas, uma implementação bastante comum da interface ETD-ECD é apresentada abaixo. Tipicamente, o *adaptador de interface serial* consiste de um módulo de entrada e saída (E/S) ligado ao barramento do computador. Este módulo contém os circuitos que implementam a parte elétrica do protocolo de comunicação empregado na interface ECD-ETD. Esta define as dimensões e pinagem dos conectores, níveis de tensão e corrente nos circuitos que transportam os sinais e o seqüenciamento das ações efetuadas por ETD e ECD. Além da interface elétrica, circuitos integrados em larga escala implementam

pelo menos parte da funcionalidade definida pelo protocolo da interface. A Figura 3.12 mostra os componentes físicos de uma interface entre ETD e ECD. A parte do protocolo que cabe ao ECD deve ser implementada naquele equipamento.

A Figura 3.12 mostra os componentes de uma interface ETD-ECD. Um diagrama de blocos de um adaptador de interface é mostrado na Figura 3.13. Sob o controle do programa de comunicação, o processador transfere octetos de um armazenador em memória para o registrador de transmissão no adaptador. O adaptador serializa cada octeto e emite os bits que compõem o octeto, um a um, pela linha DTX. No circuito de recepção, o adaptador amostra os bits recebidos serialmente e reconverte-os em octetos. O processador examina o estado do periférico, e quando houver um ou mais octetos disponíveis, estes são copiados do registrador de recepção para um armazenador em memória. Dependendo da sofisticação do adaptador, os registradores de transmissão e recepção podem ser implementados como filas com capacidade para vários octetos. Estas filas dissociam a velocidade do processador da velocidade da comunicação serial porque acomodam uma certa assincronia entre aos circuitos dos ETDs+ECDs.

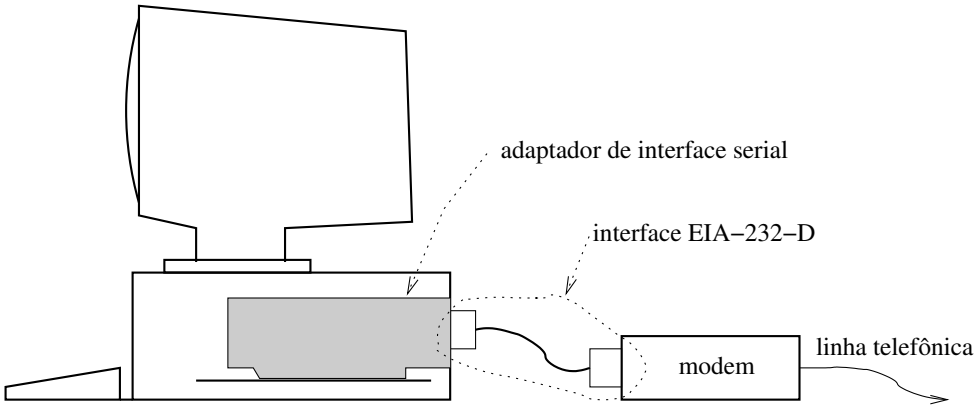


Figura 3.12: Componentes físicos de uma interface ECD-ETD.

Tipicamente, a interface ETD-ECD contém os três conjuntos de sinais mostrados na Figura 3.14. Os *sinais de dados* (DTX, DRX) conduzem a informação a ser transmitida entre os terminais local e remoto. Os *sinais de controle* (CONTR) indicam e/ou modificam o estado de funcionamento do ECD e do ETD. Sua função é manter o ETD ciente da condição de funcionamento do ECD bem como de alterar esta condição, sob comando do programa controlador da interface serial do ETD. Os *sinais de sincronismo* (SINCR) controlam a temporização dos sinais de dados nas linhas DTX e DRX.

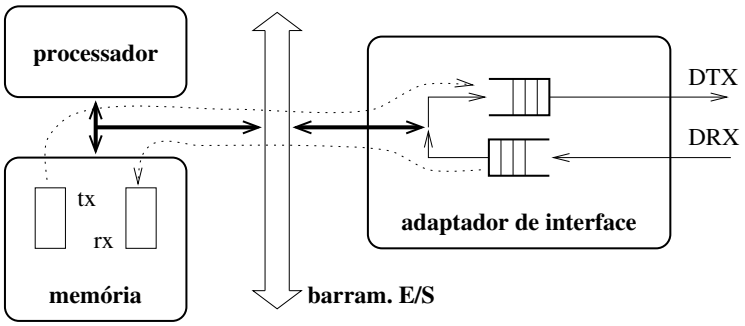


Figura 3.13: Diagrama de blocos de um ETD.

Os ECDs geralmente são implementados por dois circuitos, um de transmissão e um de recepção. A Figura 3.15 mostra o diagrama de blocos de um modem. Os bits recebidos pela

linha DTX são transcodificados de acordo com a técnica de modulação do ECD. O sinal elétrico resultante é condicionado ao passar através de um filtro e do amplificador de saída. Este condicionamento garante que os sinais emitidos pelo modem se enquadram nos limites prescritos pela técnica de modulação empregada. No circuito de recepção, o sinal recebido é amostrado e condicionado antes de ser decodificado e entregue ao ETD, através da linha DRX. A filtragem no circuito de recepção elimina ruídos espúrios que podem confundir o demodulador, causando erros de decodificação.

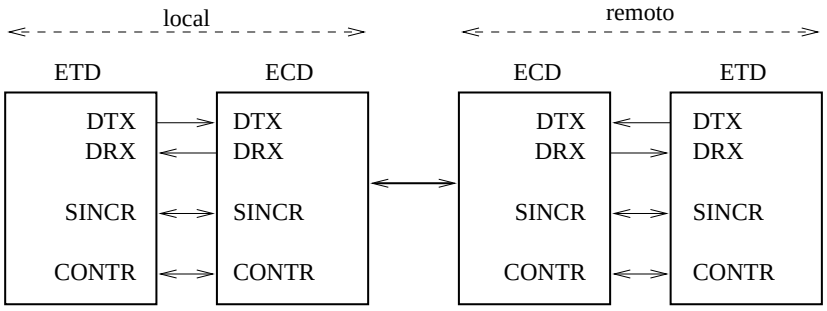


Figura 3.14: Interface ECD-ETD.

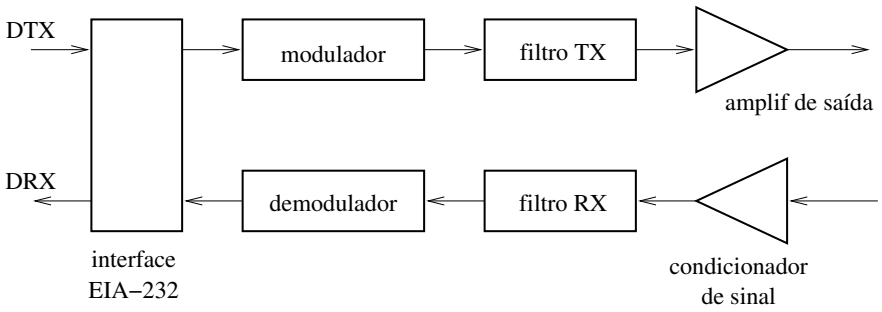


Figura 3.15: Diagrama de blocos de um ECD.

3.4 Interface EIA-232-D

A interface EIA-232-D é usada para interligar equipamentos terminais de dados a equipamentos de comunicação de dados. Normalmente, os ECDs são modems, necessários para a transmissão de dados através de canais de voz como linhas telefônicas –veja a Figura 3.12, pág. 61. Esta interface também é conhecida como RS-232-D, por ser este o nome das versões anteriores do padrão. Interfaces tipo "232" são amplamente usadas para interligar terminais assíncronos a computadores, bem como em computadores pessoais para a comunicação através de linhas telefônicas. O uso de interfaces EIA-232-D é tão difundido que vários fabricantes produzem circuitos integrados que implementam em silício os protocolos definidos pelo padrão. Estes dispositivos são discutidos na Seção 3.5. Os seguintes padrões são equivalentes:

EIA-232-D Electronics Industries Association EIA-232-D com título "Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange", 1987.

CCITT V.24 & V.28 Comitê Consultivo Internacional de Telefonia e Telegrafia, Livro Vermelho, Vol VIII, 1984. A recomendação V.24 lista definições para circuitos de interconexão entre Equipamento Terminal de Dados e Equipamento

de Comunicação de Dados. A recomendação V.28 define as características elétricas para circuitos de interconexão desbalanceados.

Telebrás 225-540-730 A prática Telebrás 225-540-730 contém "Especificações gerais de circuitos de interconexão entre equipamento terminal de dados (ETD) e equipamento de comunicação de dados (ECD) para velocidades de transmisssão de sinal de dados de até 20000 bits/s". Emissão 03, 1986.

3.4.1 Interfaces Mecânica, Elétrica e Funcional

O padrão EIA-232-D define uma interface mecânica (dimensões dos conectores), uma interface elétrica (níveis e comportamento temporal dos sinais elétricos) e uma interface funcional (comportamento dos sinais lógicos na interface ECD-ETD). Estas são discutidas no que segue.

Interface Mecânica

Os conectores empregados na interface EIA-232-D são definidos pela norma ISO-2110-1980 e conhecidos como DB-25, com 25 pinos. O conector macho é o ponto de acesso no ETD e o conector fêmea, no ECD. A Figura 3.16 mostra as dimensões do conector macho e a posição dos pinos no mesmo. No cabo que conecta ECD a ETD, o conector fêmea no cabo encaixa "dentro" do conector macho no ETD. Existe uma versão reduzida da interface com somente sete sinais de dados e controle, que utiliza um conector de 9 pinos do tipo DB-9.

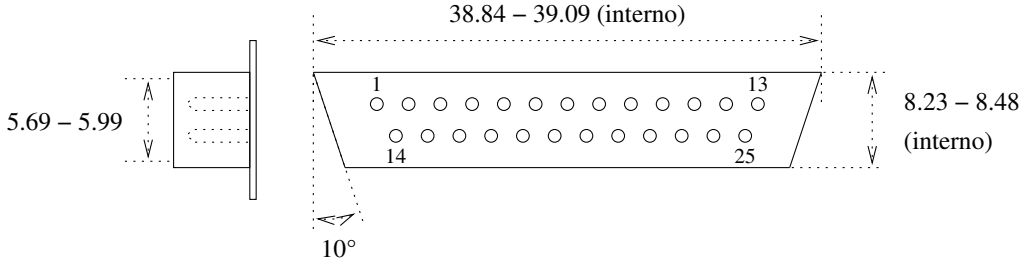


Figura 3.16: Vista externa do conector DB-25 macho.

Interface Elétrica

O padrão EIA-232-D define os níveis de tensão e corrente elétricas correspondentes aos níveis lógicos como mostrado na Tabela 3.1. Uma transição nas linhas de dados deve durar menos que 1ms ou 4% da duração de um bit ($dv/dt < 30V/\mu s$). Uma transição nos sinais de controle ou relógio deve durar menos que 1ms. Note que o tempo de transição especificado impõe um limite na velocidade de transmissão máxima possível. A interface permite ligações entre ECD e ETD em distâncias menores que 15m em velocidades menores que 20kbps. Estes são os valores especificados no padrão, embora transmissão a distâncias maiores e a velocidades mais altas são possíveis em alguns equipamentos.

Classe de sinal		nível lógico	faixa de tensão
Sinais de dados	**1**	marca	−3V a −15V
	0	espaço	+3V a +15V
Sinais de controle, sincronismo	**1**	ativo, ativ	+3V a +15V
	0	inativo, inatv	−3V a −15V
Faixa de transição	**X**	indefinido	−3V a +3V

Tabela 3.1: Níveis de tensão dos sinais na interface EIA-232-D.

Interface Funcional

Os sinais definidos pelos padrões EIA-232-D e CCITT V.24 são mostrados na Tabela 3.2. Da esquerda para a direita a tabela mostra o pino do conector DB-25, o identificador no padrão V.24, o identificador no padrão EIA-232-D, descrição do sinal além de sua categoria e fonte. A existência de duas vias de dados (DRX e DTX) permite a operação em modo duplex. O padrão define um canal secundário, cuja operação como canal reverso também permite a operação duplex. Em geral, a velocidade de transmissão do canal secundário é menor que a do canal principal. No que segue, somente o canal principal é descrito.

DB25	V.24	EIA-232	Descrição	Categoria	Fonte
2	103	DTX	dados a transmitir	dados	ETD
3	104	DRX	dados recebidos	dados	ECD
15	114	TCK	sincronismo de transmissão	sincron.	ECD
17	115	RCK	sincronismo de recepção	sincron.	ECD
24	113	TCKE	sincr. de transmissão externo	sincron.	ETD
20	108	DTR	ETD operacional/pronto	controle	ETD
6	107	DSR	ECD operacional/pronto	controle	ECD
11	126	MOD	seleção de modo (orig/resp)	controle	ETD
23	111	VEL	seleção de velocidade	controle	ETD
4	105	RTS	solicitação para transmitir	controle	ETD
5	106	CTS	pronto para transmitir	controle	ECD
8	109	DCD	portadora detectada	controle	ECD
22	125	RING	linha chamando	controle	ECD
21	140	LDR	enlace digital remoto	teste	ETD
18	141	LAL	enlace analógico local	teste	ETD
25	142	TST	ECD em teste	teste	ECD
7	102	0V	nível de referência de tensão		
1		GND	terra de proteção (0V)		

Tabela 3.2: Sinais da interface EIA-232-D.

Sinais de dados Estas linhas transportam os dados binários serializados nas interfaces seriais dos ETDs local e remoto. Os bits a transmitir devem ser amostrados pelo ECD com o sinal TCK e os bits recebidos devem ser amostrados com o sinal RCK.

103 – DTX dados gerados pelo ETD local. Quando em repouso, a linha deve permanecer em 1. Dados são transmitidos pelo ETD se e somente se DTR=DSR=RTS=CTS=ativ.

104 – DRX dados binários recebidos de ECD/ETD remotos e repassados pelo ECD local. O ECD deve manter DRX=1 se DCD=inatv.

Sinais de sincronismo São usados para amostrar os bits nas linhas de dados. Em comunicação síncrona, o sincronismo de recepção RCK é extraído dos dados recebidos pelo ECD.

114 – TCK *Relógio de transmissão.* As transições de 1 para 0 devem ocorrer no centro dos bits em DTX.

115 – RCK *Relógio de recepção.* As transições de 1 para 0 devem ocorrer no centro dos bits em DRX.

113 – TCKE *Relógio de transmissão externo.* Gerado pelo ETD em transmissões síncronas. As transições de 1 para 0 devem ocorrer no centro dos bits em DTX.

Sinais de controle As linhas de controle podem ser divididas em dois grupos. No primeiro grupo estão os sinais que informam o estado geral de funcionamento de ECD e ETD (DTR, DSR) e os sinais que forçam o ECD a operar na velocidade e modo escolhidos pelo ETD (VEL, MOD). No segundo grupo estão os sinais de controle e indicação de estado associados à transmissão de dados propriamente dita. Estes são o pedido e a autorização para transmitir (RTS, CTS), o indicador de recepção de dados (DCD) e o indicador de chamada telefônica (RING). *Portadora* é o sinal modulado injetado no meio físico pelo ECD.

108 – DTR "Data Terminal Ready" ("Data Terminal Equipment Ready"), indica *ETD pronto* para operar e solicita ao ECD que se conecte à linha telefônica. DTR=inatv causa a desconexão do ECD. Em linhas comutadas, uma transição para inatv em DTR não pode ser seguida de uma transição para DTR=ativ até que DSR=inatv.

107 – DSR "Data Set Ready" ("Data Circuit-terminating Equipment Ready"), indica *ECD operacional*, que o ECD está conectado à linha telefônica, que não está em modo de teste e que o protocolo de ligação à linha já foi concluído.

111 – VEL "Data signal rate select", quando ativo, o ETD está indicando ao ECD para transmitir na velocidade mais alta de duas possíveis.

126 – MOD "Mode of operation", se aplicável, o ETD indica ao ECD para operar no *modo origem* (MOD=inatv) ou no *modo resposta* (MOD=ativ).

105 – RTS "Request To Send", o ETD *deseja transmitir*.

106 – CTS "Clear To Send", em resposta a RTS, indica ECD *pronto para transmitir*. A condição CTS=DSR=ativ indica que o ECD está transmitindo os dados presentes em DTX.

109 – DCD "Data Carrier Detected", indica *portadora detectada* pelo ECD. DCD=inatv causa DRX=1.

125 – RING "Ring indicator", indica que o ECD está recebendo o sinal *indicador de chamada* telefônica.

Sinais de teste Estas linhas permitem o diagnóstico de falhas no equipamento, discutido na Seção 3.4.4. Num *enlace digital remoto*, o ECD remoto conecta a sua linha DRX diretamente a sua linha DTX; dessa forma, todos os dados enviados pelo ECD local são recebidos de volta. O ETD pode avaliar a qualidade da linha ao comparar os bits recebidos com os enviados. Num *enlace analógico local*, o ECD local conecta a saída de seu circuito de transmissão à entrada de seu circuito de recepção e desconecta-se da linha telefônica. Isso permite ao ETD avaliar o funcionamento do ECD local – os bits recebidos pelo ETD devem ser iguais aos transmitidos.

140 – LDR "Remote Loopback", ou *enlace digital remoto*. Se aplicável, o ETD indica ao ECD local que envie ao ECD remoto um comando para que aquele opere no modo de enlace digital remoto.

141 – LAL "Local Analog Loopback", ou *enlace analógico local*. O ETD indica ao ECD que redirecione os sinais analógicos de sua saída para a sua própria entrada analógica.

142 – TST "Test mode", quando ativo, indica ao ETD que o ECD está em *modo de teste*, atendendo a um pedido de enlace digital remoto.

3.4.2 Intertravamento entre ETD e ECD

O padrão EIA-232-D define as interações entre os sinais da interface. Por exemplo, a transmissão só pode iniciar após o ETD avisar ao ECD que deseja transmitir e o ECD responder que está pronto para fazê-lo. Para acompanhar a descrição do funcionamento da interface, veja a Figura 3.17; a seqüência de eventos é listada abaixo:

1. terminal local pede para transmitir ("request to send");
2. modem transmite seqüência de treinamento (portadora);
3. modem remoto detecta portadora ("data carrier detected");
4. modem local permite transmissão ("clear to send");
5. terminal local envia dados; modem modula e transmite-os;
6. terminal local encerra transmissão;
7. modem remoto detecta fim de transmissão;
8. terminal remoto detecta fim de transmissão (DCD inativo).

Inicialmente, o ETD e o ECD indicam que estão prontos para operar fazendo DTR=DSR=ativ. Quando o ETD deseja transmitir, este ativa RTS, e em resposta, o ECD injeta sinal (portadora) na linha. Após decorrer o intervalo *RTS-CTS*, o ECD ativa CTS, indicando ao ETD que este pode iniciar a transmissão. O intervalo *RTS-CTS* é definido no padrão específico ao ECD. Durante este intervalo, o ECD local transmite uma *sequência de treinamento* para que o ECD remoto ajuste seus circuitos de recepção à qualidade do sinal recebido. Ao perceber CTS=ativ, o ETD local inicia a transmissão pela linha DTX. Ao concluir a transmissão, o ETD local faz RTS=inatv, o que causa CTS=inatv e DTX=1.

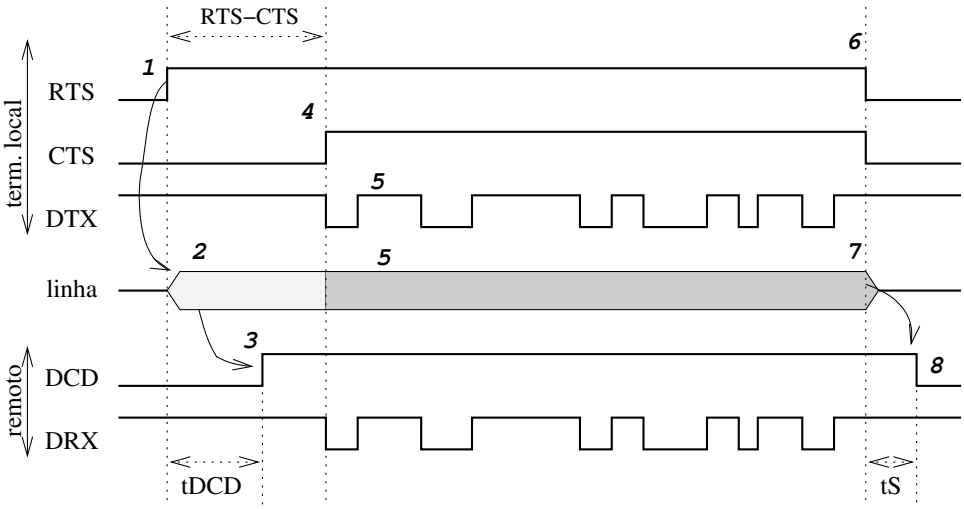


Figura 3.17: Estabelecimento de conexão entre ETDs local e remoto.

O ECD remoto, ao reconhecer portadora em nível e qualidade adequados e após ajustar seu circuito de recepção, faz DCD=ativ, informando ao ETD remoto que este está prestes a receber dados. O intervalo entre detecção da portadora e a ativação de DCD é marcado na Figura 3.17 como t_{DCD}. Da mesma maneira, decorre um intervalo, t_S, entre a remoção da portadora pelo ECD local e a desativação de DCD pelo ECD remoto. O intervalo t_S é necessário para evitar que perdas momentâneas de sinal, causadas por ruído, interrompam a comunicação entre os dois ETDs. Note que t_S é independente do tempo de propagação do sinal pela linha. As setas nas Figuras 3.17 e 3.18 indicam uma dependência funcional entre dois sinais. Por exemplo, o ECD somente inicia a sequência de treinamento após perceber RTS=ativ.

Na operação em modo duplex, a sequência de eventos mostrada na Figura 3.17 ocorre nas duas pontas da linha. A Figura 3.18 mostra a operação da interface em modo semi-duplex. A seqüência de eventos é mostrada abaixo:

1. terminal local pede para transmitir ("request to send");
2. modem transmite seqüência de treinamento (portadora);
3. modem remoto detecta portadora ("data carrier detected");
4. modem remoto detecta silêncio (DCD inativo);
5. terminal remoto pede para transmitir ("request to send");
6. terminal local detecta portadora (DCD ativo);
7. terminal remoto encerra transmissão.

No modo de operação semi-duplex, somente um ETD transmite a cada vez e o direito de transmitir é passado de um ETD para o outro quando o transmissor encerra a transmissão de um bloco de dados. O ECD ligado ao primeiro ETD a transmitir opera em *modo origem* e o ECD ligado ao outro ETD opera em *modo resposta*. Quando o ETD em modo origem encerra a transmissão de seu primeiro bloco de dados, este faz RTS=inatv, o que força seu ECD a remover a portadora da linha. O ECD em modo resposta avisa ao ETD remoto que este pode iniciar sua transmissão ao fazer DCD=inatv. O ETD remoto então ativa sua linha RTS e ao detectar CTS=ativ, pode iniciar a sua transmissão. O *intervalo de reversão de linha* t_R é o tempo que decorre desde que o ECD percebe a ausência de sinal na linha e faz DCD=inatv até que seu ETD ative RTS. Note que, nos intervalos *RTS-CTS*, t_S (Figura 3.17) e t_R, não são transmitidos dados. Estes intervalos devem portanto ser minimizados já que eles reduzem a taxa efetiva de transmissão de dados.

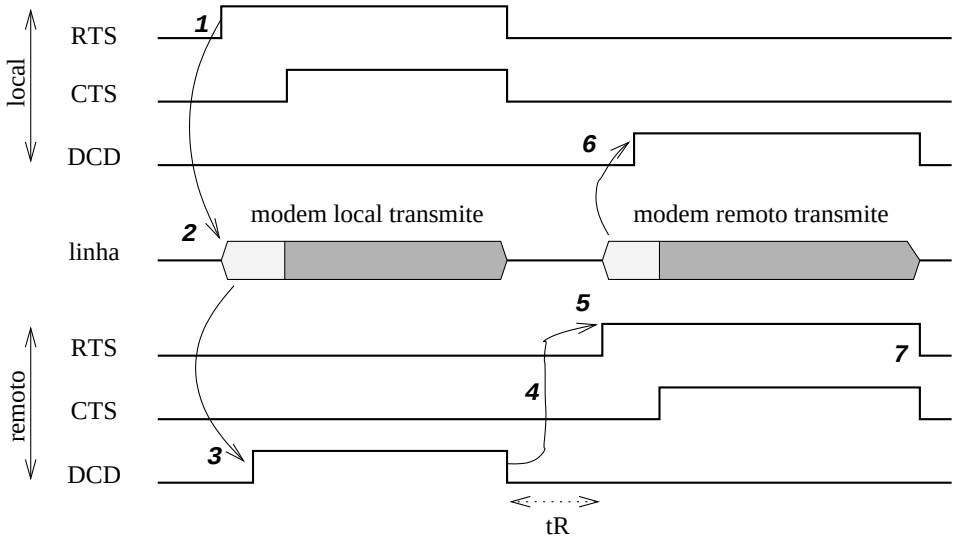


Figura 3.18: Comutação de linha em transmissão semi-duplex.

3.4.3 Conexão sem modem

A interface EIA-232-D pode ser usada para a interconexão direta de dois ETDs, se estes estiverem fisicamente próximos. A ligação mais simples é estabelecida através de um cabo onde o sinal DTX de um ETD é ligado à entrada DRX do outro e vice-versa *i.e.,* o cabo com as ligações invertidas se comporta como um "modem nulo". A Figura 3.19 mostra dois ETDs ligados diretamente com um modem nulo. Além da ligação do pino 7 (0V) nos dois conectores, algumas implementações da interface necessitam da ligação do pino 1 (GND) nos dois conectores. Ainda

outras implementações somente funcionam se os pinos 7 e 1 forem interligados – *isso é fora de padrão e pode causar sérios danos aos equipamentos.* Quando um fabricante informa que seu equipamento é compatível com o padrão EIA-232-D, isso em geral não significa que a compatibilidade é total. Apenas alguns dos sinais da interface podem estar presentes e mesmo nestes, a implementação não necessariamente funciona como o especificado pelo padrão.

A Figura 3.20 mostra as ligações necessárias para transmissões com e sem controle de fluxo. Os números ao lado dos sinais indicam os pinos do conector DB-25 que devem ser interligados. A diferença entre os dois esquemas de conexão está na possibilidade de intertravamento entre os dois ETDs através do uso dos sinais de controle –veja o Exercício 3.6. Dependendo das ligações internas ao equipamemto e da interpretação do significado dos sinais pelo fabricante, a fiação do modem nulo deve ser ligada diferentemente *e.g.* o sinal DTR é ligado a RING no mesmo conector e a DSR e DCD no outro conector. Ainda para transmissões síncronas, pode ser necessária a ligação do sinal TCKE a uma fonte externa de temporização.

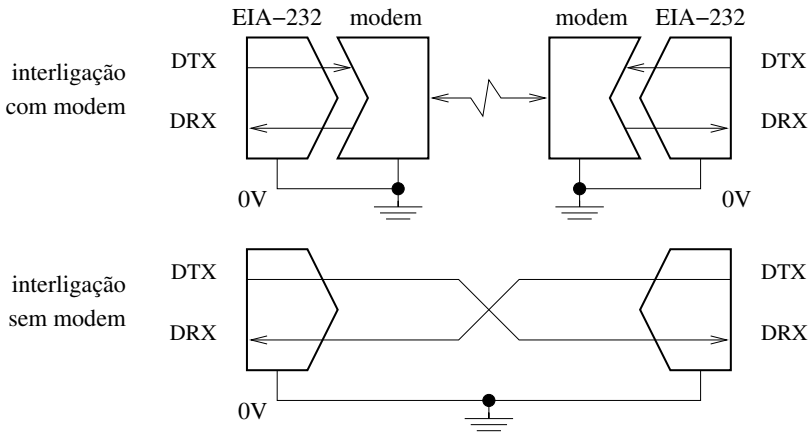


Figura 3.19: Interligação de ETDs com e sem modem.

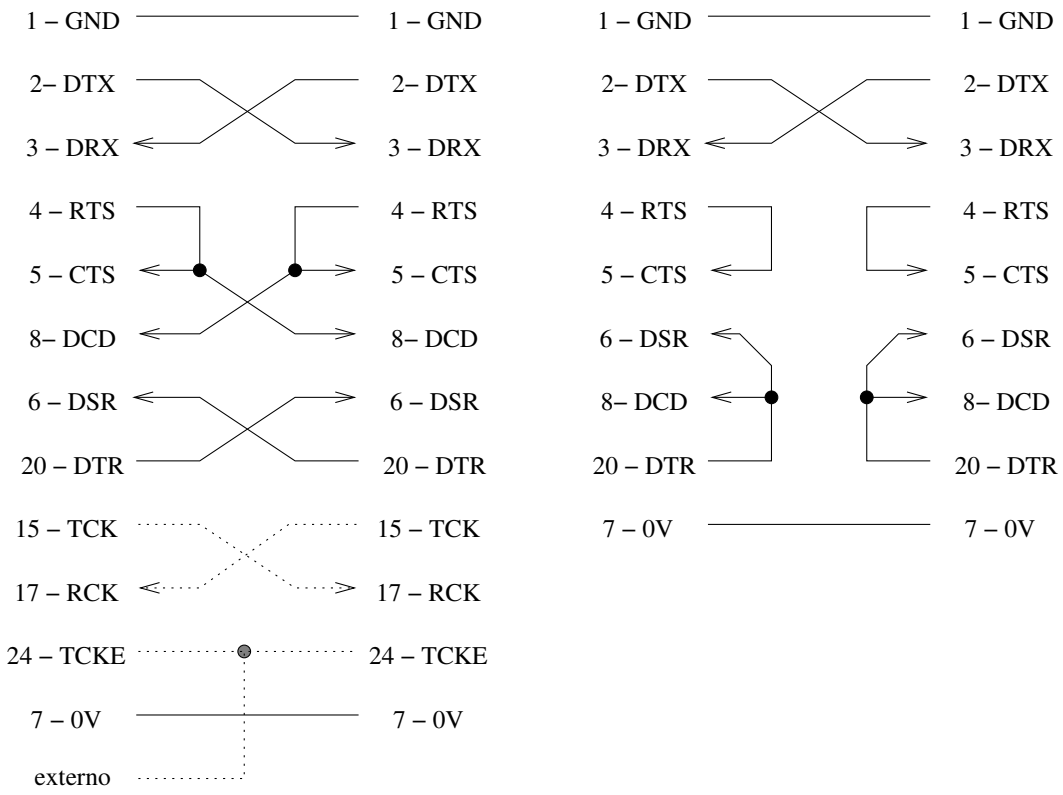


Figura 3.20: Modem nulo para ligações com e sem controle de fluxo.

3.4.4 Diagnóstico e Testes na Interface EIA-232

A função dos sinais de teste presentes na interface EIA-232-D é descrita na página 65. O sinal 141–LAL permite a verificação do funcionamento do ECD local – veja a Figura 3.21. Quando o ECD é colocado em *enlace analógico local*, a saída analógica de seu circuito de transmissão é ligada à entrada analógica de seu circuito de recepção. Se o ECD está funcionando corretamente, dados inseridos na linha DTX devem ser idênticos àqueles entregues pela linha DRX.

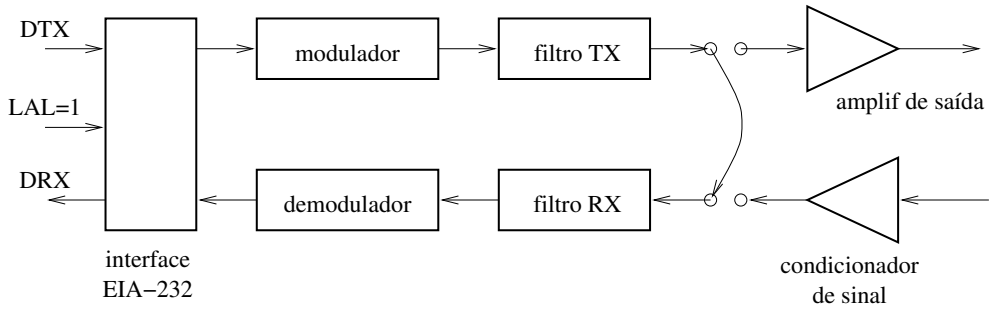


Figura 3.21: Enlace analógico local.

Alguns ECDs contém um dispositivo que permite o teste do enlace de dados envolvendo ETD e ECD locais, a linha e o ECD remoto. Quando o ETD ativa o sinal 140–LDR, o ECD local envia ao ECD remoto um pedido de *enlace digital remoto* –veja a Figura 3.22. O ECD remoto desconecta-se do ETD remoto e liga, internamente, suas linhas DRX a DTX. Assim, todos os dados recebidos pelo ECD remoto são re-transmitidos para o ECD local. Esse teste permite a detecção de falhas no ECD remoto, na linha e no ECD local.

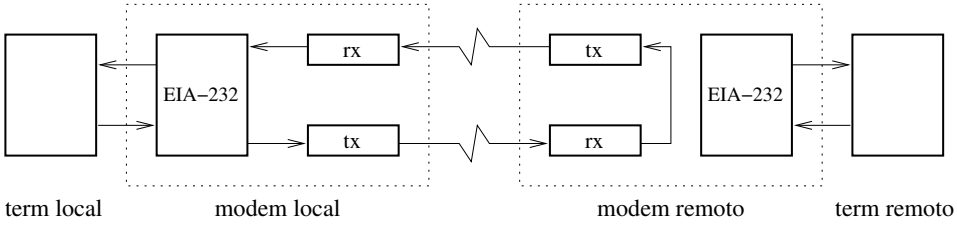


Figura 3.22: Enlace digital remoto.

Alguns modens possuem um gerador de padrões de teste que facilita a condução de testes nos enlaces locais e remotos. O gerador de padrões emite uma cadeia de caracteres continuamente, o que exercita o enlace de uma forma bem determinada. Algumas das cadeias de caracteres empregadas nestes testes são o "padrão dot" *i.e.,* bits 1's e 0's alternados, ou o "padrão fox", que contém todas as letras do alfabeto ASCII: "the quick brown fox jumps over the lazy dog".

Uma *testador de interface* ("breakout box") é uma ferramenta extremamente útil na instalação e diagnóstico de problemas em conexões via EIA-232-D. A versão mais simples do testador consiste somente de um conector com os pinos ligados de forma a eliminar uma parte do enlace *i.e.,* DTX–DRX, RTS–CTS, DTR–DSR–DCD. Este conector permite testes simples de funcionamento do ETD e do cabo entre ETD e ECD. Dispositivos mais sofisticados permitem a visualização do estado dos sinais na interface através de LEDs (diodos emissores de luz) e a reconfiguração das ligações entre os sinais da interface. Alguns testadores são "inteligentes" e podem executar vários testes de enlace automaticamente.

3.4.5 Interfaces EIA-530, EIA-423-A, EIA-422-A

Alguns padrões foram publicados mais recentemente que o EIA-232-D definindo interfaces de maior velocidade para a comunicação serial de dados entre ETDs e ECDs –veja [Bla93] para detalhes. Estes novos padrões refletem os avanços na microeletrônica ocorridos desde a introdução do padrão RS-232-A.

EIA-530, EIA-449 – CCITT V.24, V.54 interface mecânica baseada em conectores DB-37 e DB-9; as interfaces funcional e lógica são semelhantes àquelas de EIA-232-D.

EIA-422-A – CCITT V.11, X.27 interface elétrica que emprega um par de fios para cada sinal, com transmissão balanceada em distâncias de até 1200m, a taxas de transmissão de até 10Mbps, por exemplo, 1200m com taxa de 100kbps, ou 12m com taxa de 10Mbps.

EIA-423-A – CCITT V.10, X.26 interface elétrica compatível com a interface EIA-232-D; os limites de desempenho desta interface são 1000m com taxa de 3kbps ou 10m com taxa de 300kbps.

3.5 Controlador de Interface Serial Assíncrona

Esta seção descreve o dispositivo periférico que implementa os protocolos de nível físico especificados pela interface EIA-232. A interface entre as partes do protocolo implementadas em silício e o código que implementa as funções restantes *i.e.,* de envio e recepção de caracteres, também é discutida. A Seção 3.5.1 descreve um circuito integrado fictício porém similar aos encontrados em computadores pessoais. A Seção 3.5.2 contém fragmentos de código que implementam, de forma muito simplificada, as funções `enviac()` e `esperac()` tendo por base o Controlador de Interface Serial Assíncrona da Seção 3.5.1.

3.5.1 Descrição Funcional

O *Controlador de Interface Serial Assíncrona* (CISA) ("Universal Asynchronous Receiver-Transmitter" ou UART) é o dispositivo de entrada e saída (E/S) que implementa a interface entre ETD e ECD. A funcionalidade de um CISA pode ser implementada num circuito integrado ou pode fazer parte de um controlador de E/S que contém várias interfaces de E/S. A função do CISA é converter os dados em formato paralelo usados internamente ao computador (8, 16 ou 32 bits) para o formato serial usado pelos ECDs EIA-232. O CISA serializa os dados a transmitir e adiciona os bits de início e final e calcula o bit de paridade, se for o caso. Na recepção, os dados recebidos serialmente são paralelizados, os bits de início e final são removidos e a paridade é re-calculada e comparada com o bit de paridade recebido, se for o caso.

O CISA implementa o protocolo assíncrono usado em interfaces EIA-232. Este protocolo consiste de uma máquina de estados que determina o comportamento dos sinais na interface ECD-ETD (RTS, CTS, etc.). Uma parte dessa máquina de estados determina a sequência de transmissão/recepção de bits pela linha serial (DTX, DRX). Os parâmetros de transmissão e recepção (*i.e.,* número de bits de final, geração ou não de bit de paridade, número de bits por caracter, velocidade) podem ser selecionados através de atribuição de valores a registradores do periférico.

Usualmente, CISAs tem uma interface paralela ligada ao barramento interno do computador e uma interface serial onde é conectado o ECD. A Figura 3.23 mostra a ligação de um CISA ao barramento de um computador. O processador executa instruções de escrita ou leitura que referenciam o endereço dos registradores do CISA para programar o modo de operação do periférico ou para verificar seu estado de operação. A interface com o barramento contém sinais normalmente encontrados em dispositivos de E/S.

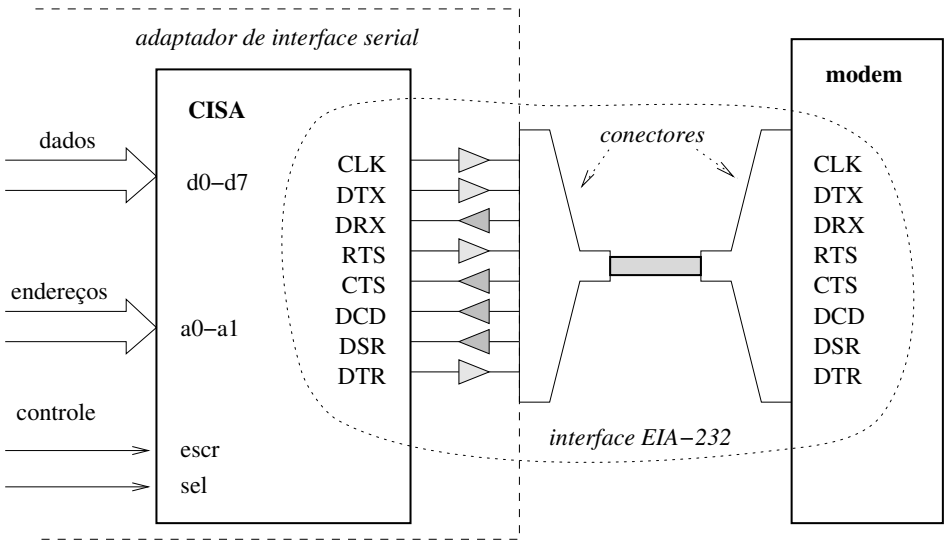


Figura 3.23: Ligações CISA–barramento e CISA–ECD.

A organização interna de um CISA básico é mostrada na Figura 3.24. Pelo lado da interface com o barramento de E/S são visíveis quatro registradores. O *registrador de controle* (Rcontr) armazena a programação do modo de operação do CISA. O *registrador de status* (Rstat) contém bits que descrevem o estado de funcionamento do CISA, *i.e.,* ocorrência de erros, espaço disponível nos registradores de recepção e transmissão, etc. Dados a transmitir pela linha serial são depositados no *registrador de transmissão* (Rtx). Dados recebidos pela linha serial são retirados do *registrador de recepção* (Rrx). As duas linhas de endereço a0 e a1 selecionam um de quatro registradores internos ao periférico de acordo com a Tabela 3.3. A linha de controle escr (escrita)

indica ao periférico se o acesso pelo processador é uma leitura ou escrita; a linha sel (selecionado), quando ativa, indica que a instrução corrente de E/S está referenciando o endereço físico do CISA. O endereço "base" do periférico é o valor contido nas linhas de endereço do processador que torna a linha selecionado ativa.

registradores		a1 a0	endereço
controle	Rcontr	00	base+0
status	Rstat	01	base+1
recepção	Rrx	10	base+2
transmissão	Rtx	11	base+3

Tabela 3.3: Endereçamento dos registradores do CISA.

Do lado da interface serial, são visíveis apenas as linhas de dados e de controle da interface EIA-232, isto é, CLK, RTS, CTS, DCD, DRX, DTX, etc. Os bits recebidos na linha DRX são armazenados num registrador de deslocamento interno que, quando contiver todos os bits de um caracter, recebidos sem erros de transmissão, este caracter é transferido para o registrador Rrx, de onde será lido pelo processador. Cada caracter a ser transmitido deve ser gravado pelo processador no registrador Rtx. Deste registrador o caracter é copiado para um registrador de deslocamento interno de onde será transmitido pela linha DTX, depois de emoldurado pelos bits de início, final e paridade. Os registradores de deslocamento são similares aos da Figura 3.8, na pág. 58. O *gerador de sincronismo* é um circuito divisor programável que gera os sinais de relógio de recepção e de transmissão. O programador escolhe a velocidade de transmissão ao gravar um número binário de valor adequado no registrador do gerador de sincronismo. Os triângulos nas linhas da interface EIA-232 são amplificadores que fazem a converão entre os níveis elétricos dos sinais internos ao ETD e aqueles dos sinais da interface EIA-232.

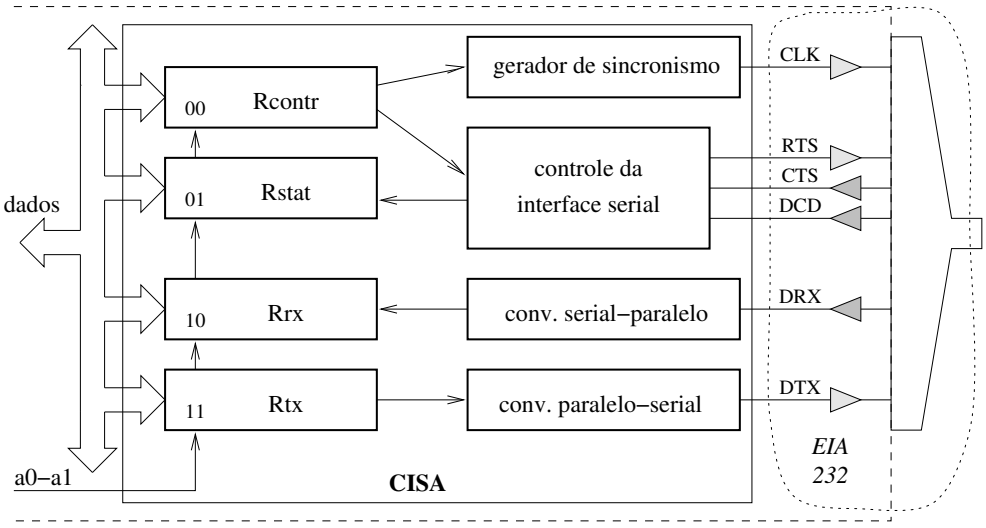


Figura 3.24: Organização interna de um CISA.

A Tabela 3.4 define o significado dos bits do registrador de controle Rcontr. A tabela mostra o efeito do bit se programado como 0 e como 1. Se o bit Rcontr.sincr é 1, o próximo valor escrito no CISA é gravado no registrador do gerador de sincronismo e vai portanto definir a velocidade de operação do periférico. O bit Rcontr.final define o número de bits de final. O bit Rcontr.parid determina se o bit de paridade é transmitido e o bit Rcontr.pimpr define se a paridade é par ou ímpar. Se o bit Rcontr.parid é 0, o bit Rcontr.pimpr é ignorado. Os bits Rcontr.rts e Rcontr.dtr

fazem com que os sinais da interface EIA-232 RTS (solicitação para transmitir) e DTR (ETD operacional) fiquem no estado ativo (ativ) ou inativo (inatv). O bit Rcontr.RXint habilita a geração de uma interrupção ao processador pelo CISA sempre que houver um caracter disponível no registrador de recepção. O bit Rcontr.TXint habilita a geração de uma interrupção sempre que houver espaço disponível no registrador de transmissão. A escrita do octeto 0x00 em Rcontr inicializa o CISA com todos os sinais inativos e com interrupções desabilitadas.

O programa que controla o envio de caracteres no ETD não pode fazer DTR=inatv até que todos os caracteres que possam estar armazenados nos registradores de dados a transmitir, visíveis ou internos, do CISA tenham sido esvaziados. Algumas interfaces seriais possuem armazenadores internos que podem conter vários caracteres esperando para ser transmitidos. Se o ETD faz DTR=inatv antes que o conteúdo destes registradores tenha sido transmitido, estes caracteres serão perdidos. Alguns ECDs necessitam de um certo intervalo entre o instante em que o último caracter é inserido no infoduto e o sinal DTR passar para a condição inatv.

bit	nome	em 0	em 1
0	sincr		gerador de sincronismo
1	final	1 bit de final	2 bits de final
2	parid	sem paridade	com paridade
3	pimpr	paridade par	paridade ímpar
4	rts	faz RTS inatv	faz RTS ativ
5	dtr	faz DTR inatv	faz DTR ativ
6	RXint	não interrompe	interrompe na recepção
7	TXint	não interrompe	interrompe após transmissão

Tabela 3.4: Significado dos bits do registrador de controle Rcontr.

O significado dos bits do registrador de status Rstat é definido na Tabela 3.5. Quando Rstat.caRX=1, há um novo caracter disponível no registrador Rrx. Este caracter c_i deve ser removido de Rrx o mais rápido possível para evitar que o próximo caracter c_{i+1} recebido pelo CISA seja gravado "em cima" dele. Caso isso ocorra o caracter c_i é perdido e um erro de *assincronia na recepção* ("overrun") é sinalizado pelo bit Rstat.ersin. Quando o caracter c_j é copiado pelo CISA do registrador Rtx para o registrador de deslocamento interno, o bit Rstat.dsTX indica que o caracter c_{j+1} pode ser copiado da memória para o registrador Rtx.

bit	nome	significado (quando em 1)
0	caRX	indica que há um novo caracter disponível em Rrx
1	dsTX	indica que há espaço disponível em Rtx
2	erenq	erro de enquadramento
3	ersin	falta de sincronia na recepção
4	dsr	o sinal DSR está ativo (ECD operacional)
5	cts	o sinal CTS está ativo (pronto para transmitir)
6	dcd	o sinal DCD está ativo (portadora detectada)
7	erpar	ocorreu erro de paridade no último caracter recebido

Tabela 3.5: Significado dos bits do registrador de status Rstat.

Se o número de bits amostrados pela interface serial for diferente do esperado, isto é sinalizado como um *erro de enquadramento* ("framing") pelo bit Rstat.erenq. Esta condição pode ocorrer se parâmetros de comunicação, tais como número de bits por caracter ou paridade, não sejam selecionados da mesma forma pelos dois ETDs nas pontas da linha. Este erro também

pode ocorrer se os relógios dos dois ETDs operam com velocidades aproximadamente iguais, mas não idênticas. Caso os caracteres estejam sendo transmitidos com paridade, o bit Rstat.erpar sinaliza a ocorrência de eventuais *erros de paridade* –para a definição de deteção de erros com bit de paridade veja a Seção 4.1. Os bits Rstat.dsr, Rstat.cts e Rstat.dcd indicam o estado das linhas da interface EIA-232 *i.e.,* se estas linhas estão ativas (ativ) ou inativas (inatv).

Além de sinalizar a recepção ou o envio de um caracter, as interrupções podem também sinalizar alterações no estado das linhas de controle da interface EIA-232. Por exemplo, se o sinal DCD passar de inatv para ativ, o ECD está informando ao ETD que o ECD remoto está prestes a enviar dados. Antes de iniciar uma transmissão, o ETD faz RTS=ativ para colocar o ECD em modo de transmissão. A transmissão só pode iniciar quando o ECD atender ao pedido de transmissão, condição essa sinalizada com CTS=ativ. Em dispositivos mais sofisticados que o descrito aqui, estas mudanças de estado também podem ser indicadas ao processador por interrupções.

3.5.2 Programação

A Figura 3.24, na pág. 72, mostra a organização interna de um CISA. Dos oito registradores do periférico, apenas cinco são visíveis ao programador. Os conversores serial-paralelo e o controle da interface serial não podem ser acessados diretamente pelo processador. Os registradores visíveis são acessados para leitura e/ou escrita dependendo das instruções contidas no programa de controle da interface serial.

O fragmento de pseudo-código mostrado abaixo indica, de forma simplificada, a funcionalidade de um programa de comunicação através de uma interface serial assíncrona. O modo de funcionamento do CISA deve ser programado durante a inicialização do sistema, quando os bits adequados do registrador de controle são ligados/desligados. Para tanto, são gravados em Rcontr uma ou mais palavras de controle que determinam o modo de funcionamento do periférico.

Os caracteres recebidos via interface serial são transferidos do periférico para um armazenador em memória (`armazenRX[ ]`) por uma função que espera pela chegada de um novo caracter e o copia para variável determinada pelo programa. No exemplo abaixo, a função `esperac()` retorna um caracter recebido da interface cujo endereço é `serial1`. A função `enviac()` espera até que haja espaço disponível no registrador de transmissão e transfere para o periférico endereçado pelo primeiro argumento o valor indicado no segundo argumento (`armazenTX[j]`).

```
configura dispositivos de E/S       /* velocidade, etc...  */
...
while (comunicando) {
  ...
  if (querReceber)
    armazenRX[i++] = esperac(serial1);
  ...                               /* usa caracter recém-recebido */
  if (querTransmitir)
    enviac(serial1, armazenTX[j++]);
  ...                               /* produz novo caracter a transmitir */
}
```

Definições

Uma versão primitiva de um programa de comunicação via interface serial é apresentada a seguir, codificada na linguagem C. Note que os valores usados neste exemplo não correspondem a nenhum CISA em particular. Para programar um CISA comercial, consulte a folha de dados do fabricante do circuito integrado.

As constantes definidas abaixo são máscaras para extrair informações do registrador de status. Por exemplo, o resultado da conjunção do conteúdo de Rstat com `CISAdcdON` indica se o ECD está ou não recebendo sinal de portadora. Da mesma forma, se o resultado da conjunção de Rstat com `CISArx` for 1, isto indica a presença de um novo caracter em Rrx.

```
#define CISArx 0x01          /* caracter disponível em Rrx */
#define CISAtx 0x02          /* espaço para novo caracter em Rtx */
#define CISAenquadr 0x04     /* erro de enquadramento */
#define CISAoverrun 0x08     /* erro de sincronismo na recepção */
#define CISAdsrATIV 0x10     /* DSR – ECD operacional */
#define CISActsATIV 0x20     /* CTS – pronto para transmitir */
#define CISAdcdATIV 0x40     /* DCD – portadora detectada */
#define CISAparERROR 0x80    /* erro de paridade no último caracter */
```

Os elementos do tipo `CISAconfigTipo` ativam/desativam características operacionais do CISA *i.e.,* especificam o número de bits de final, determinam o tipo de detecção de erros pelo bit de paridade, controlam os sinais da interface EIA-232 e habilitam ou não interrupções na recepção e envio de caracteres.

```
typedef unsigned char uchar;  /* octetos, oito bits sem sinal */
typedef enum CISAconfig {
  uchar reset = 0x00,         /* inicializa periférico */
  uchar sincr = 0x01,         /* inicializa gerador de sincronismo */
  uchar stops1 = 0x00,        /* bits de final - um bit de final */
  uchar stops2 = 0x02,        /* bits de final - dois bits de final */
  uchar paridNao = 0x00,      /* sem paridade */
  uchar parid = 0x04,         /* com paridade */
  uchar paridPar = 0x00,
  uchar paridImpar = 0x08,
  uchar rtsATIV = 0x10,       /* ativa sinal RTS – solicitação para transmitir */
  uchar rtsINAT = 0x00,
  uchar dtrATIV = 0x20,       /* ativa sinal DTR – ETD operacional */
  uchar dtrINAT = 0x00
  uchar RXint = 0x40,         /* habilita interrupções na recepção */
  uchar RXnaoI = 0x00,
  uchar TXint = 0x80,         /* habilita interrupções na transmissão */
  uchar TXnaoI = 0x00
} CISAconfigTipo;
```

Os elementos do tipo `CISAvelTipo` são as constantes que devem ser gravadas no registrador do gerador de sincronismo para selecionar a taxa de transmissão em que o periférico vai operar. Quanto maior o valor do divisor, menor a velocidade de operação.

```
typedef enum CISAvel {
  uchar v38400 = 0x00,
  uchar v19200 = 0x01,
  uchar v9600 = 0x02,
  uchar v4800 = 0x04,
  uchar v2400 = 0x08,
  uchar v1200 = 0x10,
  uchar v600 = 0x20,
  uchar v300 = 0x40,
} CISAvelTipo;
```

O endereço do registrador de controle pode ser acessado tanto para a escrita de valores de configuração do periférico como para programação do gerador de sincronismo. A estrutura `CISAcontrTipo` diferencia estes dois tipos de acesso.

```
typedef union CISAcontr {
  CISAconfigTipo config;       /* modo de operação */
  CISAvelTipo vel;             /* velocidade de operação */
} CISAcontrTipo;
```

O programador deve forçar o compilador a alocar a estrutura de dados que descreve os registradores do CISA aos endereços físicos adequados. No código que segue, os dispositivos de E/S são *mapeados em memória*. A alternativa, com os dispositivos de E/S mapeados num espaço de endereçamento especial para E/S, é empregada em algumas máquinas *e.g.* a família Intel 80x86. Para a programação de periféricos naquelas máquinas, as idéias básicas são as mesmas que as discutidas aqui embora os detalhes de codificação sejam diferentes: a escrita e leitura dos registradores dos periféricos somente ocorre com o uso de instruções de máquina reservadas para E/S.

O tipo `CISAtipo` descreve os registradores visíveis do CISA. Note a correspondência exata entre os elementos deste tipo e a Tabela 3.3 (pág. 72).

```
/* registradores do CISA mapeados em octetos consecutivos */
typedef struct CISA {
  CISAcontrTipo Rcontr;                /* base + 0 */
  uchar Rstat;                         /* base + 1 */
  uchar Rrx;                           /* base + 2 */
  uchar Rtx;                           /* base + 3 */
} CISAtipo;
```

Declarações

Os quatro registradores visíveis do CISA são mapeados nos endereços de octetos consecutivos, iniciando nos endereços base `serial1` e `serial2`. Aos endereços dos registradores vão corresponder os octetos da estrutura `CISAtipo`, alocados pelo compilador, conforme o endereçamento mostrado na Tabela 3.3.

```
CISAtipo serial1;
CISAtipo serial2;
```

Inicialização

No exemplo que segue, a interface `serial1` é inicializada com um comando `reset` e programada para operar a 9600bps, com dois bits de final ("stop bits"), paridade par e a linha DTR é ativada, informando ao ECD que o ETD está pronto para iniciar suas transmissões. O periférico vai operar com as interrupções desabilitadas.

```
  serial1.Rcontr.config = reset;          /* inicializa periférico */
  serial1.Rcontr.config = ( sincr | stop2 | parid | paridPar | dtrATIV );
  serial1.Rcontr.vel = v9600;             /* programa gerador de sincronismo */
```

Recepção

Antes de tentar transmitir ou esperar pela chegada de algum caracter, o ETD deve verificar se o ECD está pronto para operar. Esta condição é sinalizada por `CISAdsrATIV`=1. O ECD deve estar recebendo sinais válidos do ECD remoto *i.e.,* do tipo e qualidade especificados pelo padrão específico ao tipo de ECD em uso. Esta condição é sinalizada por `CISAdcdATIV`=1 (portadora detectada). Ao esperar pela recepção de um caracter, o processador testa continuamente o registrador de status, até que o bit `CISArx` seja 1. Se não ocorreu nenhum erro de paridade no caracter recebido, ele pode ser retornado. Se ocorreu um erro, o tratador de erros é invocado.

```
char esperac(CISAtipo s) {
  while ( ! ((s.Rstat & CISAdsrATIV) && (s.Rstat & CISAdcdATIV)) )
    ;                                   /* ECD pronto e recebendo?  */
  while ( ! s.Rstat & CISArx )
    ;                                   /* espera até receber caracter */
  if ( s.Rstat & CISAparERROR )         /* erro de paridade?  */
    panico("erro de paridade");
  else
    return s.Rrx;
}
```

Transmissão

Antes de transmitir, o ETD informa ao ECD que a transmissão vai iniciar, ativando a linha RTS da interface EIA-232 (`rtsATIV`). Note que o código abaixo pressupõe que ambos escrita e leitura sejam possíveis em Rcontr. A transmissão somente inicia após o ECD ativar o sinal CTS (`CISActsATIV`). A cada caracter, o processador espera por espaço no registrador de transmissão do CISA, testando o bit `CISAtx`. Quando há espaço disponível no registrador, o novo caracter é depositado no registrador de transmissão.

```
void enviac(CISAtipo s, char c) {
  s.Rcontr.config |= rtsATIV;           /* pede para transmitir */
  while ( ! s.Rstat & CISActsATIV )     /* pode transmitir?  */
    ;
  while ( ! s.Rstat & CISAtx )          /* espera por espaço */
    ;
  s.Rtx = c;
}
```

Este exemplo usa a técnica de *consulta* ("polling") para detectar a recepção ou a transmissão de um caracter *i.e.,* o processador fica testando continuamente o conteúdo do registrador de status, esperando por uma mudança no estado do periférico. Obviamente, este método é *muito* ineficiente porque, enquanto espera pela ocorrência de um evento relacionado ao periférico, o processador não faz nada além de ficar esperando. Em um programa minimamente realista, a recepção de um caracter (ou bloco de caracteres) é sinalizada ao processador por uma interrupção. O tratador da interrupção é quem copia o(s) caracter(es) do registrador de recepção para um armazenador do sistema ou usuário.

3.6 Bibliografia e Exercícios

[McN88] é um exelente texto que discute vários tópicos em comunicação de dados via interface serial. Os protocolos da família [XYZ]modem são muito usados em aplicativos de computadores pessoais – veja [Kie94] para uma implementação de [XY]modem. Para exemplos realistas de programa controlador para interface serial veja [Com84, Wel94]. Sobre a Interface EIA-232, veja [McN88, Mon90]. O CISA empregado nos computadores pessoais tipo PC é o circuito integrado Intel 8250 [Int79] ou National Semiconductor 16450 ou 16550A [NSC95].

Exercícios

Ex. 3.1 [§ 3.1] Projete dois multiplexadores de 4 canais de entrada e um de saída, um síncrono e outro estatístico. Defina, nos dois casos, o algoritmo de escolha do próximo canal com direito a transmitir. Calcule a utilização do canal de saída dos dois multiplexadores quando os canais de entrada transmitem

com as seguintes freqüências (durante o intervalo necessário para enviar 100 caracteres, envia N, N≤100): canal A - 25%, canal B - 50%, canal C - 15% e canal D - 10%.

Ex. 3.2 [§ 3.2] Suponha que as palavras "transmissão assíncrona" estejam sendo transmitidas por uma linha serial, codificadas em ASCII com 7 bits/caracter e dois bits de final. Mostre a sequência de re-sincronização, após a detecção de um falso bit de início na letra "t". Pista: o receptor sempre espera que um bit de final seja seguido por um bit de início.

Ex. 3.3 [§ 3.2] Faça esquemas das máquinas de estados que controlam as converões série-paralelo e paralelo-série. Considere todas as possibilidades de tamanho de caracter (5-8 bits), cálculo de paridade, número de bits de final, erros de enquadramento e sincronismo.

Ex. 3.4 [§ 3.4.2] Faça um esquema da máquina de estados dos sinais de controle da interface (DTR, DSR, RTS, CTS, DCD).

Ex. 3.5 [§ 3.4.4] Dois computadores estão ligados via interface EIA-232 sem modem mas, a comunicação serial não funciona. Determine quais ligações são necessárias no conector EIA-232 para colocar uma interface em "enlace digital local". Quais problemas podem ser diagnosticados com este testador de interface?

Ex. 3.6 [§ 3.4.3] Explique o funcionamento dos cabos mostrados na Figura 3.20, pág. 69. Em particular, mostre a sequência de eventos nos sinais da interface que permite o controle de fluxo entre os dois ETDs. Por que a ligação a quatro fios não permite o controle de fluxo?

Ex. 3.7 [§ 3.5.1] Usando os indicadores de erro de enquadramento e de sincronia de recepção, descreva um algoritmo capaz de medir a taxa de transmissão dos dados recebidos. Suponha que durante a inicialização do enlace o ETD remoto envia um mesmo caracter n vezes. Determine um valor mínimo para n. Existem caracteres que facilitam esta tarefa?

Ex. 3.8 [§ 3.5.2] Suponha que você está projetando um programa controlador de interface serial baseado em interrupções. Um dos requisitos de projeto é que o processador mantenha a máxima taxa efetiva de transmissão e recepção. Quantas interrupções por segundo ocorrem numa transmissão a 9600bps? E a 64Kbps? E no caso de transmissão e recepção simultâneas? Suponha uma interrupção por caracter.

Ex. 3.9 [§ 3.5.2] Escreva o código das duas funções de tratamento de interrupção de transmissão e recepção. Cada uma das funções copia um caracter de/para um vetor separado em memória e incrementa o apontador apropriado. Compile o código fonte até o estágio de geração de código de máquina (`gcc -S f.c -o f.s`) e conte o número de instruções que o processador vai executar em cada caso.

Ex. 3.10 [§ 3.5.2] Sabendo as respostas dos Exercícios 3.8 e 3.9, quanto tempo sobra para o processador executar outras tarefas? Suponha que o processador executa uma instrução a cada 10ns.

Ex. 3.11 [§ 3.5.2] Suponha que, ao invés de um registrador, o CISA armazene vários caracteres em duas filas, uma de transmissão e uma de recepção. Qual seria um tamanho adequado para estas filas, levando em conta as respostas aos Exercícios 3.8–3.10? Esta é a diferença mais importante entre os CISAs 16450 e 16550A, usadas nos PCs.

Proj. 3.1 [§ 3.4.4, 3.5.2] Escreva um programa para a medição da taxa de erro em uma linha de comunicação serial assíncrona. Seu programa deve colocar o modem em enlace digital/analógico local/remoto e transmitir uma quantidade suficiente de dados para que as medidas sejam confiáveis (10^6 a 10^{10} caracteres). Com os resultados de medições tomadas em intervalos longos, trace um gráfico da variação da taxa de erros ao longo de um dia.

Proj. 3.2 [§ 3.2] Projete os circuitos lógicos associados aos registradores de deslocamento para a transmissão e recepção serial de caracteres de 8 bits. Como o seu circuito de recepção faz a detecção do início de um caracter?

Proj. 3.3 [§ 3.2] Projete os circuitos lógicos que implementam as máquinas de estado do Exercício 3.3 bem como dos registradores necessários para a implementação de um CISA simples.

Proj. 3.4 [§ 3.2] Ao amostrar o último bit do caracter, o processador deve copiar o caracter recém recebido do registrador de deslocamento durante a duração do bit de final. Caso isso não ocorra, o caracter será sobreposto pelo caracter seguinte. Acrescente componentes ao circuito do Projeto 3.3 de modo a resolver este problema.

Capítulo 4

Nível de Enlace

Os capítulos anteriores descrevem algumas das formas de comunicação através de enlaces físicos, empregando as funções atribuídas ao Nível Físico no modelo estratificado OSI/ISO. Um enlace físico possibilita apenas o transporte de bits entre os dois ETDs que compõem o enlace. Se ocorrerem erros de transmissão durante o trajeto entre os dois ETDs, os circuitos encarregados da comunicação não tem a responsabilidade de detectá-los nem de corrigí-los. A *detecção e a recuperação de erros* é uma das funções pertinentes ao Nível de Enlace. A funcionalidade embutida no Nível de Enlace esconde algumas das deficiências inerentes ao transporte de bits fornecido pelo Nível Físico. O tratamento de erros de transmissão é um dos serviços mais importantes do Nível de Enlace porque propicia, às camadas superiores, o transporte confiável de dados entre as duas pontas do enlace. Por esta razão, um sistema com as funções do Nível de Enlace é freqüentemente chamado de *enlace de dados* ou de *enlace lógico*.

Além da detecção e recuperação de erros, o *controle do fluxo* de dados entre transmissor e receptor também é uma função importante de um enlace de dados. O controle do fluxo é necessário para impedir que um ETD rápido envie dados para um ETD lento a taxas maiores do que este conseguiria absorvê-los. Os mecanismos de controle de fluxo mantém transmissor e receptor sincronizados, evitando que uma nova mensagem seja remetida quando o receptor não tem condições de aceitá-la.

Nos enlaces físicos discutidos até agora, somente dois ETDs estão interligados por um meio de comunicação. Existem várias ciscunstâncias onde é conveniente compartilhar o meio físico entre três ou mais ETDs. Considere um escritório com alguns computadores e uma impressora. Geralmente, nem todos os computadores estão imprimindo todo o tempo, e portanto, a impressora é um recurso que pode ser compartilhado entre os computadores. O mesmo se aplica ao meio físico que interliga estes computadores à impressora: a cada instante somente um dos computatores está transmitindo seus dados para a impressora, e o meio físico fica sem uso quando não há nada a imprimir. Ao invés de ligar cada computador diretamente à impressora, o mesmo cabo pode ser compartilhado por todos os computadores desde que exista alguma forma de arbitragem para decidir sobre o direito de transmissão. O *controle de acesso ao meio* é o mecanismo do controle de enlace lógico que decide qual de vários candidatos tem o direito de inserir seus dados no meio físico a cada instante.

4.1 Detecção de Erros

As taxas de erro de linhas telefônicas discadas são da ordem de um bit em erro para cada 10^3 a 10^5 bits transmitidos, isto é, um caracter errado para cada 1000 enviados por transmissão assíncrona. Essas taxas de erro são inaceitáveis na maioria das aplicações. Assim sendo, os protocolos de controle de enlace possuem mecanismos para a detecção e recuperação de erros de transmissão. As técnicas de detecção de erros se baseiam numa idéia simples: um

número C_t é computado em função dos dados que se deseja transmitir sem erro. Na transmissão, o número C_t é enviado junto com os dados. O receptor, usando o mesmo algoritmo que o transmissor, calcula C_r em função dos dados recebidos. Se o receptor obtém $C_r = C_t$, os dados foram recebidos sem erro. Algumas das técnicas usadas para calcular C_t e C_r são descritas nos próximos parágrafos.

4.1.1 Teste de Paridade Vertical

O método mais simples de detecção de erros de transmissão é o uso de um *bit de paridade*, com a aplicação da técnica de detecção de erros do *teste de paridade vertical* (TPV). A paridade de um caracter é a contagem módulo-2 do número de bits em 1 no caracter, *e.g.* se o número de 1's é par, o bit de paridade é 0. Se o número de bits em 1 é ímpar, o bit de paridade deve ser 1 para que o total de bits em 1 seja par. Com detecção de erros por bit de paridade, a paridade de cada caracter é computada pelo transmissor e o bit de paridade é transmitido junto com o caracter. O bit de *paridade vertical* v_j do caracter j é computado tomando-se o ou-exclusivo ($\oplus$) de todos os n bits do caracter:

$$v_j = b_{1j} \oplus b_{2j} \oplus b_{3j} \oplus \ldots \oplus b_{nj}$$

Outra possibilidade consiste em contar o número de bits em 1 no caracter a ser transmitido e gerar o bit de paridade de tal forma que o número total de bits em 1 seja *par i.e.,* se a contagem de bits em 1 no caracter for ímpar, o bit de paridade deve ser 1. Se um bit do caracter "virar" durante a transmissão, o cômputo da paridade pelo receptor vai indicar a ocorrência do erro. Contudo, se dois bits forem invertidos, o duplo erro não é detectado. Este método não é adequado para uso em linhas discadas porque os pulsos de ruído naquelas linhas tendem a afetar vários bits adjacentes, o que reduz a eficácia da detecção de erros.

4.1.2 Teste de Paridade Longitudinal

O método de detecção de erros pelo *teste de paridade longitudinal* (TPL) consiste de enviar, junto a cada bloco de caracteres, um caracter adicional tal que cada bit deste caracter denota a paridade computada longitudinalmente sobre todos os bits daquela posição no bloco de caracteres. Para um bloco com m caracteres, o bit i do caracter de paridade l é computado tomando-se o ou-exclusivo sobre todos os m caracteres do bloco de dados:

$$l_i = b_{i1} \oplus b_{i2} \oplus b_{i3} \oplus \ldots \oplus b_{im}$$

Normalmente, são usados bits de paridade vertical em conjunto com um caracter de paridade longitudinal, como mostrado na Figura 4.1.

Este método de detecção de erros é extrememente simples de se implementar, sendo extensamente empregado em programas de comunicação de dados. Os dois métodos TPV e TPL quando combinados possibilitam melhoras da ordem de 10^2 a 10^4 na taxa de detecção de erros em linhas discadas [Bla93]. O protocolo BSC emprega esta combinação de testes de paridade quando o alfabeto empregado é o ASCII com 7 bits [IBM70].

$$\begin{array}{rccccccc}
\text{primeiro caracter} \rightarrow & b_{11} & b_{21} & b_{31} & b_{41} & \ldots & b_{n1} & v_1 \\
 & b_{12} & b_{22} & b_{32} & b_{42} & \ldots & b_{n2} & v_2 \\
 & b_{13} & b_{23} & b_{33} & b_{43} & \ldots & b_{n3} & v_3 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\text{último caracter} \rightarrow & b_{1m} & b_{2m} & b_{3m} & b_{4m} & \ldots & b_{nm} & v_m \\
\text{TPL} \rightarrow & l_1 & l_2 & l_3 & l_4 & \ldots & l_n & v_l
\end{array}$$

Figura 4.1: Bloco de dados com caracter do TPL.

Teste de paridade nos protocolos IP Outra versão do teste de paridade longitudinal é empregada para a deteção de erros de transmissão nos protocolos IP, UDP e TCP [BBP88, MK90, Rij94, Com95]. O teste de paridade é computado efetuando-se a soma módulo-65535 de todas as palavras da mensagem a ser enviada. O corpo da mensagem é tratado como um vetor de palavras de 16 bits e o complemento da soma é transmitido junto com a mensagem. O receptor refaz a soma e se o valor que ele calcula, incluindo a palavra de paridade recebida, for igual a zero, não ocorreram erros de transmissão. Da mesma forma que paridade longitudinal, este teste não detecta o reordenamento das palavras na mensagem. O código da função `testeParidade()`, que computa este teste de paridade, é mostrado abaixo. Este código usa palavras de 16 bits –ao invés de 32– porque, quando os protocolos TCP/IP foram desenvolvidos, as máquinas que compunham a Internet eram baseadas em processadores de 16 bits.

```
typedef unsigned short uint16;          /* inteiro com 16 bits */
typedef unsigned int uint32;            /* inteiro com 32 bits */

uint16 testeParidade(uint16 *msgm, uint32 tam) {
   uint32 soma;

   for (soma=0; tam > 0; tam--, msgm++) {
      soma += *msgm;
      if (soma > 0xffff)                /* soma não cabe em 16 bits */
         soma = 1;
   }
   return (uint16)(~(soma & 0xffff));
}
```

4.1.3 Teste da Divisão Polinomial

O método de detecção de erros pelo *Teste da Divisão Polinomial* (TDP) ("cyclic redundancy code" – CRC) consiste em dividir o número binário representado por uma cadeia com k bits de dados (a mensagem propriamente dita) por um divisor D adequado. O resto da divisão, com n bits, é concatenado ao bloco de dados, que fica então com $k + n$ bits. O receptor repete a divisão do número binário representado pela seqüência de bits recebidos por D e se o resto da divisão for zero, o bloco foi recebido sem erros.

A mensagem a ser transmitida é representada pelo polinômio

$$M^{k-1} + M^{k-2} + \cdots + M^1 + M^0 \ .$$

O primeiro bit da mensagem corresponde ao coeficiente a_{k-1} do termo M^{k-1}. Os dois últimos bits da mensagem correspondem aos coeficientes a_1 e a_0 dos termos M^1 e M^0, respectivamente. Da mesma forma, o número usado como divisor é representado por um polinômio na variável X tal que os coeficientes binários diferentes de zero correspondem aos bits em 1 no divisor D. Por exemplo, o divisor

$$\texttt{1 0001 0000 0010 0001} \text{ é representado por } X^{16} + X^{12} + X^5 + X^0 \ .$$

Este método de detecção de erros se baseia em aritmética módulo-2 em campos finitos. Isso significa que somas e subtrações são efetuadas *sem* o vai-um e vem-um, isto é, somas e subtrações são efetuadas tomando-se o ou-exclusivo bit-a-bit dos operandos. A idéia é transmitir um número binário T que seja exatamente divisível por D. A construção de T se dá em três passos:

1. Para um divisor D com $n + 1$ bits (com coeficiente de X^n diferente de zero), T_0 é obtido multiplicando-se M por 2^n –isso corresponde a concatenar n zeros à direita de M;

2. Divide-se $T_0 = 2^n M$ por D. O resto necessariamente contém menos de n bits significativos;

3. O resto R da divisão é subtraído de T_0, resultando em $T = 2^n M - R$. A subtração é efetuada pelo ou-exclusivo do resto R com os n bits em zero à direita de T_0.

T é exatamente divisível por D porque o resto da divisão foi subtraído de T_0 no terceiro passo. T é então transmitido através do enlace, quando alguns de seus bits podem ser invertidos.

O receptor divide o número recebido T' por D. Se a mensagem foi recebida sem erros, $T' = T$ e T'/D produz resto zero. Se ocorreram erros de transmissão, estes serão detectados por causa do resto diferente de zero. Existe contudo uma pequena probabilidade de os bits errados tornarem T' exatamente divisível por D. A escolha apropriada do divisor D reduz drasticamente a probabilidade desta ocorrência (veja abaixo).

Suponha, a título de exemplo, que a mensagem `10111` deva ser transmitida intacta através de um enlace. O divisor a ser empregado é `101`. Efetuando a divisão, obtém-se quociente `100` e resto `11`. O resto da divisão é concatenado à mensagem que é então transmitida, neste caso, `1011111`. O receptor divide `1011111` por `101`. Se o resto da divisão for zero, a mensagem original foi recebida sem erros.

Após a transmissão, os bits errados vão corresponder ao polinômio $E^p + E^q + \cdots + E^r$ para erros nas posições p, q e r da mensagem T'. A ocorrência de erros de transmissão implica na adição do polinômio em E ao polinômio transmitido T. O erro de transmissão não é detectado se e só se o polinômio em E for divisível exatamente pelo polinômio em X.

O polinômio $X^{16} + X^{12} + X^5 + X^0$ é conhecido como TDP-CCITT (padrão V.41), produz uma seqüência de resto com n=16 bits e é muito usado em transmissões de caracteres (octetos). O uso deste polinômio permite a detecção de todos os erros com um único grupo de não mais de 16 bits afetados e mais de 99.9% de todos erros em grupos de mais de 16 bits. Isso equivale a uma taxa de erros não-detectados de um bit em cada 10^{14} transmitidos. Outros polinômios empregados com frequência são

$$\begin{array}{ll} \text{TDP-12} & X^{12} + X^{11} + X^3 + X^2 + X^1 + X^0 \\ \text{TDP-16} & X^{16} + X^{15} + X^2 + X^0 \\ \text{TDP-32} & X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} \\ & + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + X^0 \end{array}$$

O TDP-12 é usado com caracteres de 6 bits e gera uma seqüência de resto n de 12 bits. Aplicações que demandam maior segurança se valem do TDP-32 (seqüência de resto com $n = 32$), que permite a detecção de todos os grupos em erro de até 32 bits. As redes locais Ethernet e FDDI, discutidas no Capítulo 5, empregam o polinômio TDP-32. Estes TDPs podem detectar a ocorrência de erros em todos os blocos B com um ou dois bits errados; em todos os blocos B com uma seqüência de m bits errados se $m < n$, bem como a grande maioria dos demais casos.

O circuito que implementa o TDP é relativamente simples. A Figura 4.2 mostra o circuito usado para computar e testar o TDP-CCITT ($X^{16} + X^{12} + X^5 + X^0$). O circuito consiste de um registrador de deslocamento de 16 bits e três portas ou-exclusivo ($\oplus$). Para os outros TDPs, muda apenas o número de flip-flops no registrador de deslocamento e número e posição das portas ou-exclusivo. O registrador é inicializado com `0`s e os bits a transmitir são amostrados pela porta à direita da figura. A cada bit transmitido, o registrador é deslocado uma posição para a direita. Após a transmissão do último bit dos dados, o conteúdo do registrador é transmitido do bit 0 para o bit 15. Duas regras sobre o comportamento das portas ou-exclusivo são importantes para se entender o funcionamento do circuito:

$$\begin{array}{ll} a \oplus a = 0 & \textit{XOR-id – entradas idênticas} \\ a \oplus 0 = a & \textit{XOR-0 – transparência} \end{array}$$

A segunda regra, *XOR-0*, mostra que o ou-exclusivo de algo com 0 é o mesmo algo *i.e.,* a porta com uma entrada em 0 fica transparente.

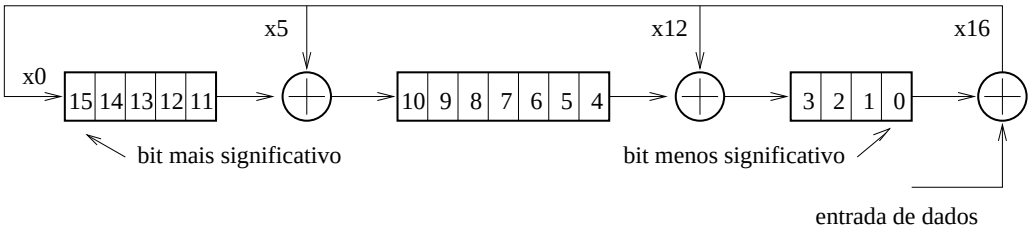


Figura 4.2: Circuito que computa TDP-CCITT.

A Figura 4.3 mostra o registrador de deslocamento após a transmissão do primeiro bit do bloco, neste caso 1. Ao ser inserido no registrador, este bit altera o conteúdo dos flip-flops 3, 10 e 15. A medida que os demais bits são transmitidos, os bits iniciais da seqüência vão sendo deslocados para a direita e vão alterando o conteúdo dos flip-flops na medida que são deslocados. Após a transmissão de todos os bits de dados, os 16 bits no registrador contém o TDP_{tx} que é então transmitido.

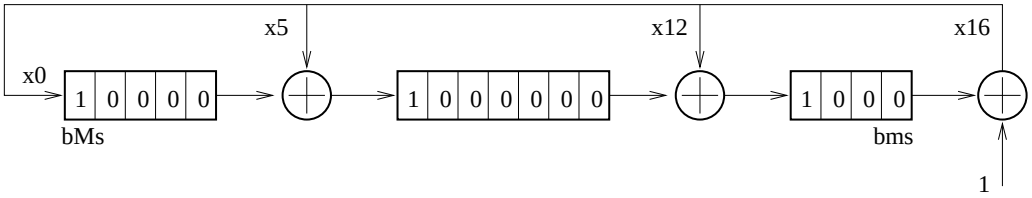


Figura 4.3: Cálculo do TDP-CCITT.

No receptor, os bits recebidos são também inseridos na porta à direita do registrador e deslocados para a direita. Quando o último bit de dados é recebido, o registrador contém um valor calculado para o TDP_{rx}. Quando o primeiro bit do TDP enviado pelo transmissor TDP_{tx} é comparado com o primeiro bit de TDP_{rx}, se os dois bits forem iguais, o resultado 0 é inserido no registrador – regra *XOR-id*. Como este bit é 0, as outras portas ou-exclusivo ficam transparentes – regra *XOR-0* – e o segundo bit de TDP_{rx} é então comparado com o segundo bit de TDP_{tx}. Se os dois forem iguais, novo 0 é inserido no registrador e assim por diante. Se todos os bits do bloco foram recebidos sem erro, o TDP_{rx} computado é igual ao TDP_{tx} recebido e o registrador de deslocamento contém 16 0s.

4.2 Controle de Enlace

O *Nível de Enlace* consiste de um conjunto de serviços oferecidos às entidades do Nível de Rede e esses serviços, por sua vez, são implementados com base nos serviços oferecidos pelo Nível Físico. A Figura 4.4 mostra o relacionamento entre as camadas do Modelo ISO/OSI e as unidades de dados de protocolo (UDP) transferidas pelos níveis em questão. No ETD local, o *datagrama* do Nível de Transporte é encapsulado pela entidade do Nível de Rede que, após adicionar o prólogo p_3 e epílogo e_3, entrega o *pacote* resultante à entidade do Nível de Enlace. Esta entidade, por sua vez, também adiciona seus prólogo p_2 e epílogo e_2 ao pacote antes de passar o *quadro* que disso resulta para transmissão pelo Nível Físico. Aquele, após concatenar o quadro com p_1 e e_1, injeta o resultado no infoduto. Na outra ponta do enlace, o Nível Físico do ETD remoto amostra o sinal recebido e entrega o quadro à entidade do Nível de Enlace remota. Ao passar da camada k para a camada $k+1$, a mensagem recebida perde o prólogo p_k

e epílogo e_k adicionados pela entidade da camada k na outra extremidade do enlace. A entidade remota de nível k recebe exatamente a mesma seqüência de bits que foi enviada pela entidade local de nível k. Cabe às entidades dos níveis inferiores a transferência dos bits entre as duas pontas do enlace. Neste contexto, o par ETD+ECD é chamado de *estação*.

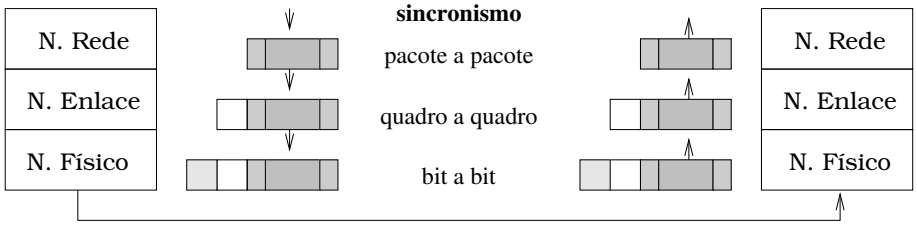


Figura 4.4: Relação da camada de enlace de dados com as camadas de rede e física.

A funcionalidade de cada nível pode ser especificada em termos da granularidade da sincronização entre as entidades de cada nível. As entidades de Nível Físico se sincronizam a cada *bit* transferido entre os ECDs. Isso é necessário para que os bits transferidos sejam amostrados no instante correto pelo receptor. As entidades de Nível de Enlace se sincronizam a cada *quadro* transferido entre os ETDs. As entidades de Nível de Rede se sincronizam a cada *pacote* transferido.

Um quadro consiste de um certo número de bits, devidamente enquadrados, que transportam informação de usuário e de controle de enlace. As informações de controle de enlace podem conter os endereços de Nível Físico de remetente e destinatário além de outras, discutidas adiante. A Figura 4.5 mostra os campos de um quadro de um protocolo de enlace. Dependendo da tecnologia do Nível Físico, um pacote do Nível de Rede pode ser maior que o maior quadro suportado pelo Nível de Enlace. Neste caso, o Nível de Rede divide o pacote em vários *segmentos* que são então transmitidos dentro dos quadros da entidade de enlace. A entidade de rede na outra ponta do enlace deve *remontar* o pacote original a partir dos segmentos recebidos da entidade de enlace. O processamento adicional envolvido nas atividades de segmentação e remontagem diminui a eficiência da comunicação entre as entidades de transporte. Em geral, os protocolos de enlace são implementados em silício enquanto que os protocolos dos Níveis de Rede e superiores são implementados por programa.

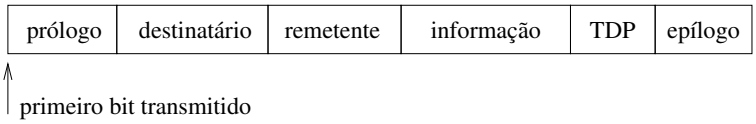


Figura 4.5: Formato de quadro do Nível de Enlace.

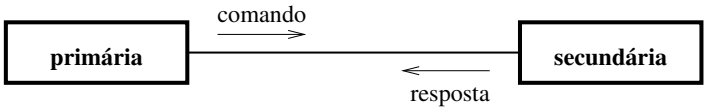
Várias funções importantes de um sistema de comunicação de dados não são cumpridas pelos componentes de Nível Físico de um sistema de transmissão (*e.g.* modem e interface EIA-232). Dentre estas, estão o *controle de fluxo* entre transmissor e receptor, para evitar que o receptor perca dados caso não consiga absorvê-los na mesma taxa em que são emitidos pelo transmissor, mantendo a sincronia bit-a-bit entre os dois ETDs. Além do controle de fluxo, o Nível Físico também não implementa nenhum mecanismo para a *detecção e recuperação de erros* ocorridos durante a viagem pelo infoduto. Se o enlace interligar várias estações numa linha multiponto, alguma forma de *endereçamento* deve ser empregada para identificar os participantes do enlace.

Um protocolo de controle de enlace deve satisfazer a um conjunto de requisitos, alguns deles conflitantes. Os protocolos de enlace devem ser flexíveis *i.e.,* devem operar tanto em enlaces ponto-a-ponto como multiponto, em modo semi-duplex e duplex e devem ser capazes de controlar

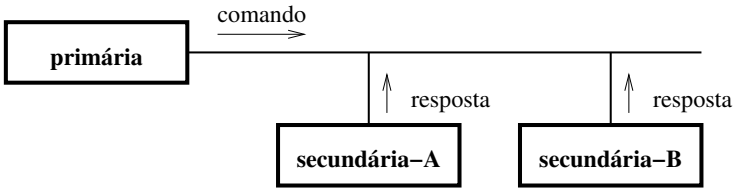
as interações entre estações com capacidades equivalentes ou em interações tipo mestre-escravo. Um mesmo protocolo deve operar correta e eficientemente em enlaces com estações próximas ou distantes umas das outras, e de forma independente do tempo de propagação dos sinais através do meio físico. A maioria das aplicações exige alta confiabilidade na transmissão de dados *i.e.,* eventuais erros devem ser eliminados pelos mecanismos de deteção e recuperação de erros. O que segue discute técnicas normalmente empregadas para resolver os problemas mencionados acima de forma a satisfazer os requisitos quanto a flexibilidade, correção, eficiência e confiabilidade.

4.2.1 Tipos de Linhas

A *topologia* de um enlace de dados descreve a configuração das linhas físicas e as conexões entre as estações através destas linhas. Uma linha *ponto-a-ponto* permite a ligação de somente duas estações (*e.g.* computador+terminal). Uma linha *multiponto* permite a ligação de várias *estações secundárias* a uma *estação primária*. Note que linhas multiponto somente são viáveis quando nenhuma estação puder transmitir continuamente. A Figura 4.6 mostra uma linha ponto-a-ponto e uma linha multiponto com uma estação primária e duas secundárias.



(a) Ponto–a–ponto



(b) Multiponto

Figura 4.6: Linhas ponto-a-ponto e multiponto.

A *duplexidade* de um enlace descreve o sentido e a simultaneidade da troca de dados entre estações. Quando a transmissão é *simplex*, o sentido é sempre o mesmo, do computador para o periférico, por exemplo. Transmissões simplex são pouco usadas na prática porque não há nenhuma realimentação entre transmissor e receptor. Numa transmissão *semi-duplex*, as transmissões ocorrem nos dois sentidos mas alternadamente: uma estação transmite, a(s) outra(s) escuta(m). Em transmissões *duplex*, duas estações podem transmitir e receber simultaneamente.

4.2.2 Disciplina de Linha

Uma das funções de uma *disciplina de linha* é determinar se há algum tipo de hierarquia entre as estações conectadas à linha. Se todas as estações tem os mesmos direitos para iniciar transmissões, a comunicação é dita *entre iguais* ou entre estações *equivalentes*. Se há uma estação que centraliza o controle de acesso à linha, esta estação é chamada de *primária* (mestre) e as demais de *secundárias* (escravos). Nos dois casos, existe a necessidade de endereçamento para que destinatário e/ou remetente das mensagens sejam identificados.

A comunicação através de um enlace de dados compreende três fases. A primeira delas é o *estabelecimento* da conexão lógica entre as estações, quando se determina quem transmite

e quem recebe e se o receptor está preparado para receber dados. Na segunda fase ocorre a *transferência de dados* em um ou mais quadros. Finalmente, na terceira fase a conexão lógica entre as estações é *terminada*. Estas atividades são descritas abaixo com os detalhes pertinentes para linhas ponto-a-ponto e multiponto.

Devido a possibilidade de a segunda estação responder com uma mensagem inválida ou, simplesmente não responder, os protocolos de controle de enlace normalmente especificam um intervalo de espera por resposta. Este intervalo, chamado de *temporizador* ("timeout"), deve ser especificado de acordo com as características do protocolo, das distâncias envolvidas e com a velocidade de processamento das estações. Caso decorra o intervalo especificado no temporizador sem que uma resposta válida seja recebida, o transmissor deve executar um procedimento de recuperação de erros. Normalmente, isso implica na retransmissão da última mensagem.

Disciplina de linha ponto-a-ponto As estações interconectadas por um enlace de dados devem cooperar no uso do meio de comunicação. Numa linha semi-duplex, somente uma estação pode transmitir em cada instante; o direito de transmitir passa de estação para estação segundo alguma regra. Além disso, a estação $\mathcal{T}$, com direito a transmitir, somente envia dados se o receptor $\mathcal{R}$ está pronto para recebê-los. Normalmente, a cada quadro q_i transmitido pela estação $\mathcal{T}$, um quadro de resposta $r_i[\,]$ é remetido pela estação $\mathcal{R}$. Quadros de resposta indicam a *aceitação* (ACT), ou a *rejeição* (REJ) do quadro recém-recebido. Um quadro de resposta informa ao transmissor $\mathcal{T}$ se o quadro q_i foi recebido sem erro (r_i[ACT]) ou, se o quadro q_i foi recebido com erros e portanto, rejeitado, (r_i[REJ]). No caso de rejeição, a primeira estação retransmite o quadro q_i. O mecanismo de *recuperação de erros* define em que condições a estação $\mathcal{T}$ pode transmitir o quadro q_{i+1}, se antes, ou somente após a aceitação do quadro q_i pela estação $\mathcal{R}$.

Uma disciplina de linha simples para enlaces ponto-a-ponto é a seguinte. Antes de uma estação enviar dados para a outra, a primeira *consulta* a segunda sobre sua disponibilidade para receber dados. Se houver espaço disponível no armazenador em memória do receptor, a resposta à consulta é positiva, indicando que a segunda estação está pronta, e a transmissão pode iniciar. Após o envio do quadro q_i por $\mathcal{T}$, o receptor responde ao transmissor, com um quadro de resposta $r_i[\,]$. Após transmitir todos os seus quadros, a estação $\mathcal{T}$ deve avisar a estação $\mathcal{R}$ do fim de suas transmissões. Quando isso ocorre, as duas estações podem trocar de papéis.

Disciplina de linha multiponto A disciplina de linha chamada *consulta* e *seleção* ("poll-select") é comumente empregada em protocolos de controle de enlace multiponto. A estação primária *consulta* a secundária sobre o seu desejo de transmitir. Se a secundária tem dados a transmitir, estes são enviados. Senão, uma resposta negativa (REJ) é retornada à primária. Quando a primária tem dados para enviar a uma secundária, aquela deve ser *selecionada* pela primária. Se pronta para receber, a secundária responde afirmativamente (ACT), ao que a primária responde com os dados. A secundária completa a transmissão informando a primária se os dados foram recebidos corretamente (ACT). Se ocorreram erros na transmissão, os dados devem ser retransmitidos.

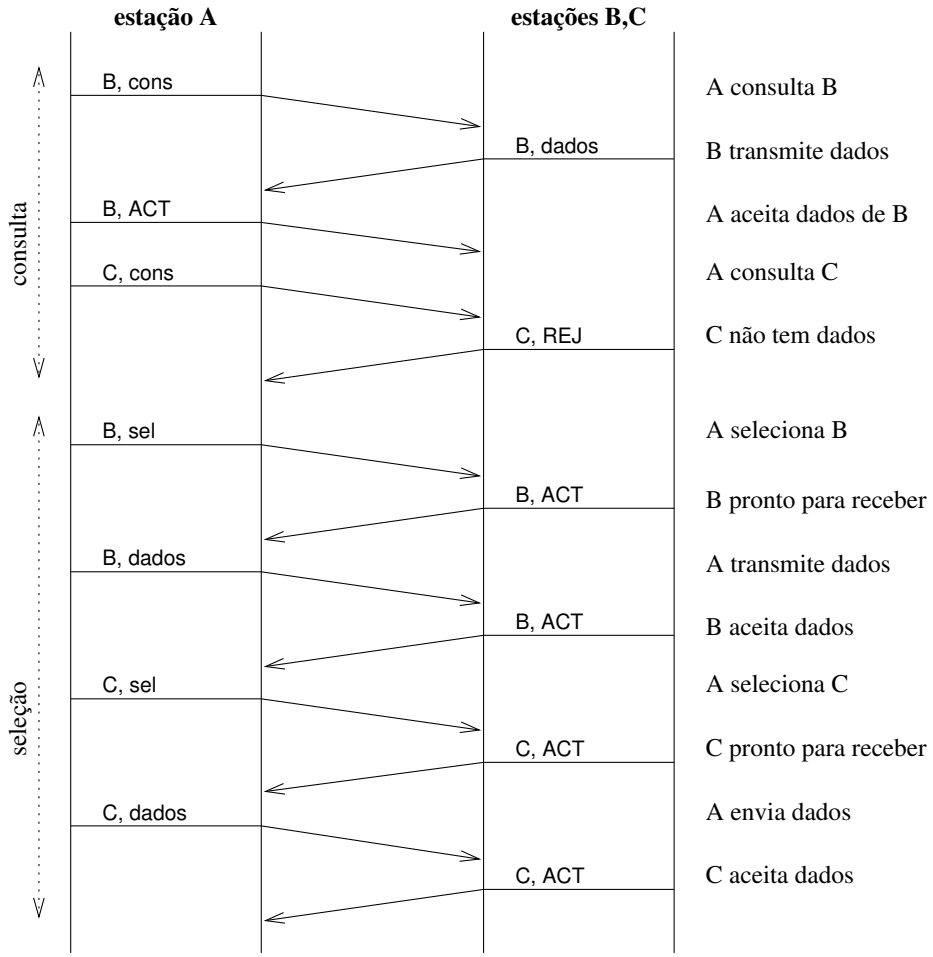


Figura 4.7: Operação de um enlace multiponto.

A Figura 4.7 mostra um exemplo de consulta e seleção. Naquela figura, o eixo do tempo é vertical e este decorre de cima para baixo. As linhas horizontais indicam o instante da inserção do primeiro bit do quadro pelo transmissor. As pontas das setas indicam o momento em que o último bit do quadro é amostrado pelo receptor. Estação primária $\mathcal{A}$ consulta e seleciona secundárias $\mathcal{B}$ e $\mathcal{C}$. As mensagens consistem de [endereço, comando] ou [endereço, dados]. No caso de seleção, são necessárias quatro quadros para o envio de uma mensagem da primária para uma secundária. Numa linha com umas poucas secundárias, uma fração enorme da capacidade da linha pode ser consumida durante um ciclo de seleção se nenhuma das secundárias tiver quadros a transmitir. Em geral, os protocolos de controle de enlace possuem modos de operação que fazem uso mais eficiente da capacidade de canal do enlace.

4.2.3 Controle de Fluxo

O tempo de propagação t_p de um bit através de um enlace é proporcional ao comprimento do enlace[1]. O produto entre o tempo de propagação e a vazão V através do enlace define o número de bits contido entre as duas pontas do enlace num dado instante.

$$\text{número de bits no enlace} = t_p V \qquad \textit{[bits]} = \textit{[s]}\ \textit{[bits/s]}$$

Considere, por exemplo, um sistema que transmite a 1 Mbps. A duração de cada bit é portanto de 1μs. Cada bit demora 5μs para atravessar um enlace com 1 Km de comprimento

[1] Ondas eletromagnéticas viajam um palmo por nanosegundo em meios guiados.

(5 ns/metro×1000 metros). Neste caso, o enlace pode conter 5 bits entre suas pontas. Um enlace de 1000 Km pode conter 5000 bits entre suas pontas. Só se atinge a máxima utilização do enlace quanto este é mantido constantemente "cheio" de bits, isto é, as transmissões mantém o número máximo possível de bits "dentro" do enlace. Isso significa que o método de controle de fluxo deve evitar interrupções desnecessárias no fluxo dos quadros.

A parte de um protocolo de enlace que controla o fluxo de dados pela linha deve executar três funções. A primeira é manter a utilização da linha pelas estações a mais alta possível já que a linha é um recurso caro. Isso significa que o tempo gasto na reversão de linhas semi-duplex deve ser minimizado pois durante os intervalos de reversão de linha nenhum dado é transmitido – veja a Seção 3.4.2. A segunda função é garantir que nenhuma estação use menos da capacidade do canal que o razoável. A terceira função do controle de fluxo é garantir que estações não recebam mais tráfego do que podem aceitar porque os quadros em excesso são descartados pelo receptor, causando suas retransmissões.

Para-e-espera Esta técnica de controle de fluxo é usada com linhas semi-duplex pois ela somente permite tráfego alternado entre duas ou mais estações. Basicamente, sob a disciplina de controle de fluxo por *para-e-espera*, o remetente envia um quadro ao destinatário que, após verificar se ocorreram erros de transmissão, envia um quadro de resposta aceitando ou rejeitando a quadro recém recebido. Até que a resposta do destinatário seja recebida pelo remetente, este não pode enviar outro quadro. Em caso de rejeição, o remetente retransmite o último quadro enviado. Um exemplo da operação de controle de fluxo com para-e-espera é mostrado na Figura 4.7 onde as estações só transmitem um novo quadro após a aceitação do último transmitido.

Apesar de sua simplicidade, o controle de fluxo por para-e-espera tende a ser ineficiente por causa do tempo perdido na comutação da linha quando do envio do quadro de resposta com aceitação ou rejeição pelo destinatário. A Figura 4.8 mostra a seqüência de eventos relativa ao envio de um quadro sob controle de fluxo com para-e-espera. O tempo de propagação p de um bit de uma ponta a outra do enlace é o intervalo que decorre desde a inserção do bit no infoduto pelo transmissor até sua amostragem pelo receptor. O tempo de inserção de quadro q é o número d de bits no quadro multiplicado pelo tempo de bit $q = d \cdot t_b = d/(\textit{taxa de transmissão})$. A figura mostra um quadro sendo inserido no infoduto no instante t. No instante $t + p$, o primeiro bit do quadro é amostrado pelo receptor e no instante $t + p + q$, o último bit do quadro é amostrado e o quadro de aceitação (ACT) é emitido pelo receptor. Em $t + 2p + q$, o transmissor recebe o quadro ACT. Para simplificar a discussão, desconsidera-se o tempo consumido na verificação de erros e de inserção do quadro de aceitação.

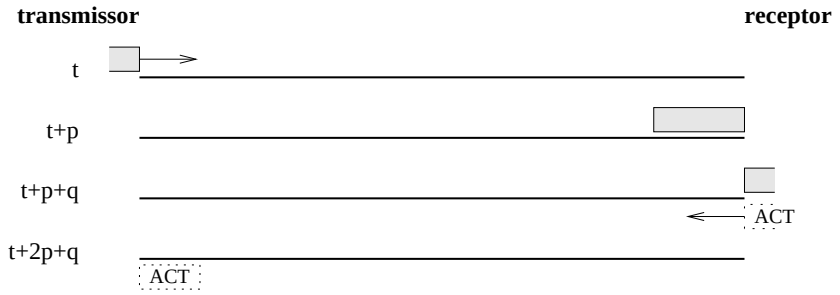


Figura 4.8: Eficiência de controle de fluxo com para-e-espera.

A eficiência de protocolos sob controle de fluxo com para-e-espera é calculada dividindo-se o intervalo q decorrido na inserção de d bits de dados, pelo intervalo entre inserção do primeiro bit daquele quadro e a chegada do quadro de aceitação no transmissor. Esse intervalo é dado

por $(t - (t + 2p + q))$. Assim,

$$E_{\text{pee}} = min[q \,/\, (q + 2p), 1] \tag{4.1}$$

No melhor caso, as estações estão próximas uma da outra, $q \gg p$, e a eficiência no uso da linha tende à unidade. Se as estações estão distantes, o tempo de propagação do sinal pela linha tem uma influência considerável, $q \ll p$, resultando em uso ineficiente da linha *i.e.,* eficiência próxima de zero.

Janela deslizante Quando são empregadas linhas duplex, o controle de fluxo por *janela deslizante* deve ser usado porque ele é mais eficiente que aquele com para-e-espera. O fluxo de mensagens com um protocolo de janela deslizante pode ser mantido praticamente constante e ininterrupto se o transmissor for capaz de manter o meio físico saturado com quadros em trânsito e o receptor for capaz de aceitá-los.

Os quadros são numerados e transmitidos seqüencialmente. O transmissor mantém uma *janela de transmissão*, que determina quais quadros na seqüência podem ser enviados. O receptor mantém uma *janela de recepção*, que indica quais os quadros que ainda vão ser recebidos. O "deslizamento" das duas janelas mantém transmissor e receptor sincronizados, ao mesmo tempo em que permite mais de um quadro em trânsito pelo meio físico. A janela consiste de um contador módulo-N, onde N corresponde ao número máximo de quadros em trânsito (ou amostrados pelo receptor) antes que o transmissor deva parar de transmitir e esperar por uma resposta. Ao receber a aceitação de um quadro, o transmissor "desliza" a janela de transmissão e envia então outro quadro.

O controle de fluxo com janela deslizante usa o mecanismo de *aceitação inclusiva*. Com aceitação inclusiva, o receptor pode aceitar vários quadros com uma única mensagem de aceitação. Por exemplo, a aceitação do quadro com número de seqüência 4 indica que todos os quadros dentro da janela de recepção, até o de número 3 inclusive, foram recebidos sem erros e que o receptor está esperando pelo quadro com número de seqüência 4.

Quando o transmissor envia todos os quadros em sua janela, ele *deve* sustar as transmissões até que receba uma mensagem de aceitação. Quando a janela de recepção fica vazia, o receptor *deve* enviar uma mensagem de aceitação. A janela de transmissão vazia significa que (1) ou a distância é muito longa e a aceitação ainda não retornou, ou (2) o receptor não tem condições de aceitar novos quadros. A janela de recepção vazia indica que o transmissor já enviou todos os quadros de sua janela e está bloqueado, esperando por uma aceitação. Este é o mecanismo que controla o fluxo porque impede que o transmissor envie quadros a uma taxa maior do que o receptor pode aceitá-los e força o receptor a enviar aceitações dentro de intervalos razoáves. Dependendo das distâncias e taxas de transmissão envolvidas, o tamanho da janela deve ser determinado de tal forma que o fluxo de quadros seja constante e ininterrupto. No caso extremo em que $N = 1$, a operação degenera para o controle com para-e-espera e a comunicação se torna semi-duplex.

A Figura 4.9 mostra janelas de transmissão e recepção de tamanho $N = 4$. A janela de transmissão encolhe da esquerda para a direita na medida em que quadros são transmitidos. A janela cresce para na direita na medida em que mensagens de aceitação são recebidas. O último quadro aceito fica fora da janela, à sua esquerda. O próximo quadro por transmitir, após a chegada de uma aceitação, fica fora da janela, à sua direita. O comportamento da janela de recepção é similar. Os quadros na janela foram recebidos mas o quadro com sua aceitação ainda não foi enviado ao transmissor. A janela encolhe quando quadros são recebidos e cresce quando aceitações são enviadas.

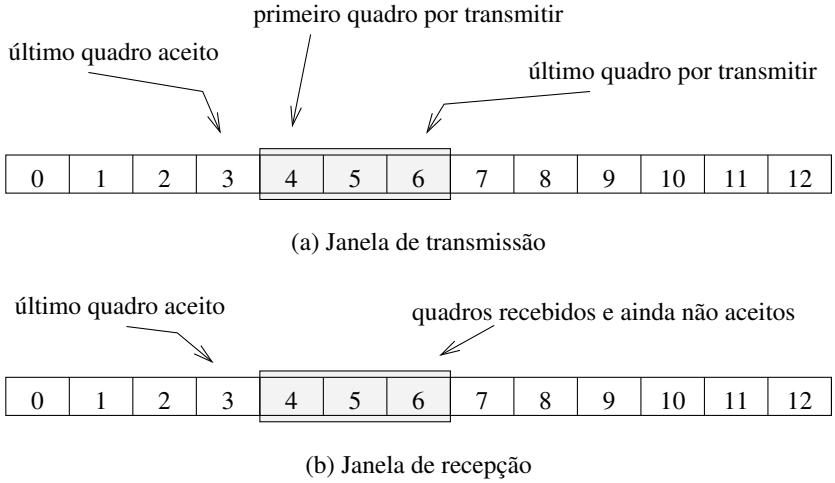


Figura 4.9: Janelas de transmissão e recepção.

A Figura 4.10 mostra o diagrama de tempo referente ao envio dos primeiros 7 quadros de uma longa seqüência. A janela tem tamanho $N = 4$, o que significa que podem existir no máximo 3 quadros em viagem, a cada instante (o Exercício 4.4 explora esta questão). Os quadros são numerados segundo uma seqüência com módulo 4. Inicialmente, as janelas de transmissão e recepção estão abertas para quadros com números de 0 a 3. O transmissor envia os quadros 0, 1 e 2 e fica bloqueado a espera de uma mensagem de aceitação. Após receber as três mensagens, o receptor fica com sua janela com tamanho zero e é forçado a enviar uma mensagem de aceitação, liberando assim o envio de novas mensagens pelo transmissor. O receptor aceita os dois primeiros quadros, ao avisar que está esperando pelo quadro 2 (ACT-2). Note que neste exemplo, apesar de ter já recebido o quadro de número 2, o receptor somente aceita os quadros 0 e 1. O transmissor envia então os quadros 3 e 4 ($4 \bmod 4 = 0$). Ao aceitar o quinto quadro com o (ACT-1), o receptor aceita implicitamente todos os quadros pendentes (2, 3 e 0), e permite que a janela de transmissão aumente três quadros. Esta aceitação também aumenta a janela do receptor em três quadros. A Seção 4.3.3 contém diagramas de janelas deslizantes com transmissão duplex que mostram vários quadros simultaneamente em viagem, quando então as qualidades do controle de fluxo por janela deslizante ficarão mais evidentes.

A Figura 4.11 mostra a transmissão de uma série longa de quadros sob controle de fluxo com janela deslizante. A figura mostra um quadro sendo inserido no infoduto no instante t. No instante $t + p$, o primeiro bit do quadro com número de seqüência 1 é amostrado pelo receptor. No instante $t + p + q$, o quadro 1 é aceito e o quadro de aceitação (ACT) é emitido pelo receptor, ao mesmo tempo em que os quadros de número 2 a $n + 1$ estão trafegando pelo infoduto. Em $t + 2p + q$, o transmissor recebe o quadro ACT, ao mesmo tempo em que o último bit do quadro $n + 1$ é amostrado pelo receptor. Assim a eficiência do controle de fluxo com janela deslizante é dada por

$$E_{\text{jd}} = min[(n + 1)q \,/\, (q + 2p), 1] \tag{4.2}$$

pois $n + 1$ quadros são transferidos entre a inserção do primeiro quadro e sua aceitação ser detectada pelo transmissor. Mantidas as mesmas condições, o controle de fluxo por janela deslizante é $(n+1)$ vezes mais eficiente que o controle com para-e-espera (Equação 4.1). Se $q \gg p$, a eficiência se aproxima da unidade. Se o tamanho da janela de transmissão $N = mq \geq 2p$ quadros, para $m \geq 1$, o transmissor pode inserir quadros continuamente no infoduto sem nunca ter de sustar transmissões enquanto espera pela aceitação de quadros.

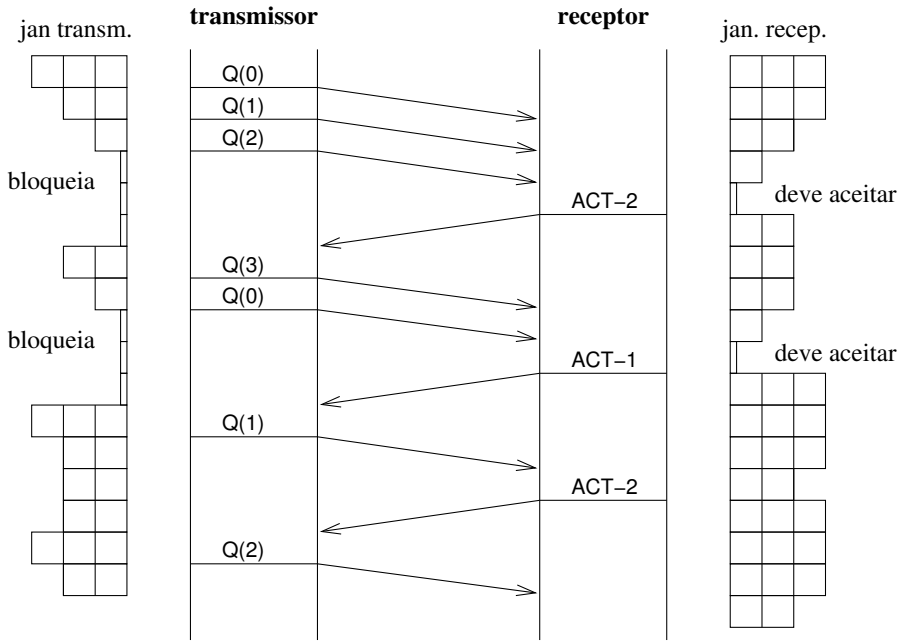


Figura 4.10: Sincronização de transmissor e receptor com janelas deslizantes.

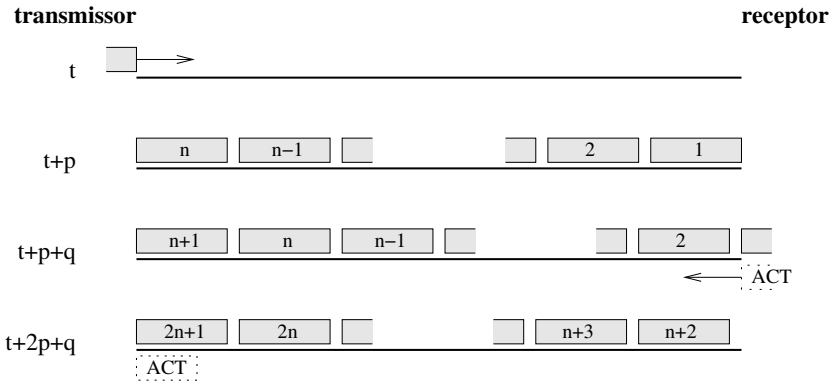


Figura 4.11: Eficiência de controle de fluxo com janela deslizante.

O controle de fluxo com janelas deslizantes é empregado em enlaces duplex. Uma técnica simples mas que permite máxima utilização do enlace consiste em enviar a aceitação dos quadros "de carona" nos quadros que contém dados mas viajam na direção contrária. Num enlace com duas estações, os quadros enviados pela estação $\mathcal{A}$ transportam dados para a estação $\mathcal{B}$ bem como as aceitações das mensagens recebidas por $\mathcal{A}$. Dessa forma, evitam-se quadros cuja única finalidade é transportar a aceitações.

4.2.4 Recuperação de Erros

Uma função importante de um protocolo de enlace é a detecção e a recuperação de erros ocorridos na transmissão. A detecção de erros é decorrência do uso de técnicas descritas na Seção 4.1. Após a detecção, a *recuperação de erros* envolve a retransmissão de um ou mais quadros. Os métodos de recuperação de erros empregados em protocolos de controle de enlace são *para-e-espera*, *volta-N* e *repetição seletiva* e empregam três técnicas diferentes de recuperação de erros. Estas técnicas são exemplos da classe de protocolos onde a recuperação de erros decorre de *pedidos automáticos de repetição* (PAR) ("Automatic Repeat reQuest" abreviado ARQ).

Para-e-espera O método de recuperação de erros por *para-e-espera* é intimamente associado ao controle de fluxo por para-e-espera. Quando o transmissor envia um quadro, este não pode iniciar outra transmissão até receber uma aceitação ou rejeição. Se o último quadro foi rejeitado, provavelmente porque o receptor detectou erros de transmissão, o quadro rejeitado é re-transmitido.

Quando um quadro que contenha uma mensagem com informação ou uma mensagem de aceitação é danificado durante o trajeto e se torna irreconhecível pelo destinatário, a mensagem perdida deve ser retransmitida. Existe a possibilidade de que a máquina destinatária tenha sofrido uma pane e esteja fora de operação, ou que ela esteja momentaneamente sobrecarregada e portanto incapaz de tratar as mensagens recebidas. Se o remetente envia um quadro ao destinatário e este não devolve um quadro de aceitação, existe a possibilidade de que o remetente fique eternamente a espera. Esta é uma situação obviamente indesejável e para evitar que ela ocorra, os protocolos de comunicação fazem uso de temporizadores de transmissão. A cada mensagem transmitida, um temporizador é inicializado com um intervalo de espera razoável. Se a aceitação é recebida antes que o intervalo se esgote, novo quadro pode ser enviado. A Figura 4.12 mostra esta situação no primeiro caso.

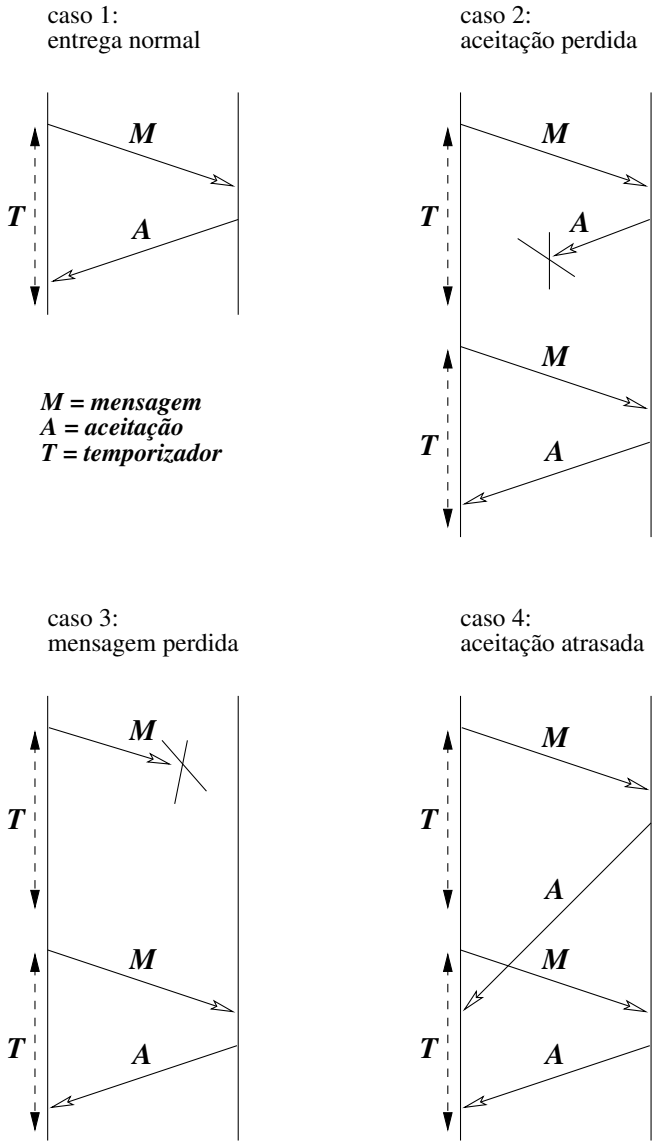


Figura 4.12: Perda e duplicação de mensagens com para-e-espera.

Se o intervalo se esgota sem que haja alguma reação do destinatário, a última mensagem é retransmitida. Existem duas possíveis causas para esta ocorrência. O segundo caso na Figura 4.12 mostra uma retransmissão causada pela perda do quadro de aceitação enquanto que o terceiro caso mostra uma retransmissão causada pela perda do quadro com a mensagem.

Finalmente, existe a possibilidade de que o destinatário demora a reagir a um quadro recebido e a enviar a sua aceitação, como mostra o quarto caso na Figura 4.12. Do ponto de vista do remetente, o temporizador expira sem que a aceitação tenha sido recebida e portanto, uma cópia do quadro supostamente perdido é retransmitida. Quando a aceitação do quadro original é finalmente recebida, o remetente já enviou a cópia daquele quadro. Neste caso, o destinatário recebe duas cópias do mesmo quadro e o remetente recebe duas cópias da mesma aceitação. Pare evitar confusão entre cópias e originais, o cabeçalho dos quadros contém um número de seqüência. Quando um quadro é recebido, seu número de seqüência é verificado. Se um quadro com o mesmo número já foi recebido, este é resultado de duplicação acidental e pode ser descartado. Note que esta discussão também é válida para o método de controle de fluxo com janelas deslizantes.

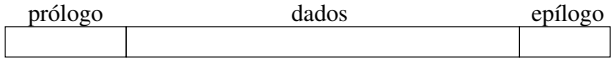
Volta-N A descrição do controle de fluxo por janela deslizante não menciona o tratamento de erros de transmissão. Suponha que a segunda resposta do receptor na Figura 4.10 fosse REJ-3, ao invés de ACT-1. Neste caso, o receptor está indicando que o quadro 3 foi recebido com erros e portanto, rejeitado. No método de recuperação de erros por *volta-N*, o transmissor deve retransmitir todos os quadros enviados desde aquele que foi rejeitado. Assim, os quadros 3 e 0 seriam retransmitidos *i.e.,* o transmissor volta atrás e retransmite os quadros 3 a 0. A retransmissão dos quadros 3 a 0 é um desperdício de tempo. Contudo, a retransmissão de todos os quadros desde quadro o rejeitado é um método muito simples de manter a seqüência dos quadros recebidos. Isso também simplifica a implementação do protocolo, seja em código seja em silício.

Repetição seletiva Outra forma de recuperação de erros em protocolos de janela deslizante consiste na retransmissão somente dos quadros onde foram detectados erros. Assim, no exemplo anterior (volta-N), somente o quadro 3 seria retransmitido. Enquanto protocolos com *repetição seletiva* evitam retransmissões desnecessárias, sendo portanto mais eficientes, a lógica de seqüenciamento é mais complicada. O receptor deve manter espaço em armazenadores para todos os quadros que poderiam ser recebidos dentro de uma janela de recepção de forma a garantir o seqüenciamento dos quadros. No exemplo acima, até que o quadro 3 fosse recebido corretamente, os quadros 3 e 0 deveriam ser retidos pelo controle de enlace do receptor e somente após a recepção correta do quadro 3 é que os quadros 3 e 0 seriam repassados para a entidade do Nível de Rede.

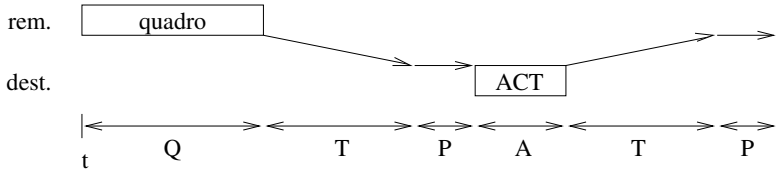
Eficiência da Transmissão Assíncrona Semi-Duplex

A transmissão semi-duplex implica em que, após enviar um quadro, o remetente espere por um quadro contendo a aceitação ou rejeição do último quadro enviado. A Figura 4.13 contém um diagrama de tempo com as ações necessárias para o envio e aceitação de um quadro. Esta seqüência de ações se repete para cada quadro enviado através do enlace. Um quadro transporta um bloco de dados que é precedido pelo prólogo e sucedido pelo epílogo. O quadro que transmporta um pacote com B_d bits de dados, é precedido por um prólogo com B_p bits e terminado por um epílogo com B_e bits. Assim, o quadro contém $B_p + B_d + B_e$ bits. O último bit do quadro é injetado no meio físico em $t + Q$, após o tempo de propagação T é processado pelo destinatário (P), que devolve um comando de aceitação, que é então processado pelo remetente. A uma velocidade de transmissão V bps, um quadro é transmitido em $Q = (B_d + B_p + B_e)/V$ segundos; desse tempo, $D = B_d/V$ segundos correspondem aos dados. Da mesma forma, o quadro

com a resposta contém B_r bits e é transmitido em $A = B_r/V$ segundos. Os dois ETDs consomem P segundos processando cada quadro recebido (computando o bloco de paridade, copiando o pacote para memória, etc). Finalmente, um quadro leva T segundos trafegando pelo meio físico. Nas derivações abaixo, o tempo de processamento P está incluido no tempo de transmissão T.



(a) Formato de um quadro.



(b) Diagrama de tempo de uma transação.

Figura 4.13: Diagrama de tempo da transmissão semi-duplex.

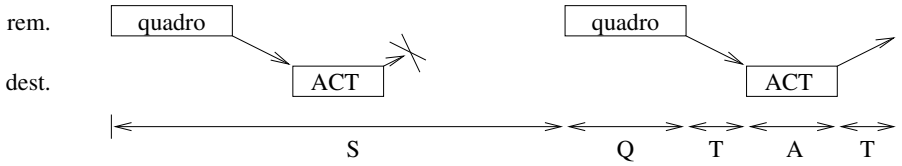
Na ausência de erros, a *utilização* U, do enlace de dados é definida como sendo o tempo usado para transmitir o bloco de dados com B_d bits, dividido pelo tempo gasto na transação completa *i.e.,* da inserção do quadro até o processamento do quadro com a aceitação. Esta eficiência é dita "ideal" pois não inclui os custos relativos à recuperação de erros.

$$U_{\text{ideal}} = D \;/\; (Q + A + 2T) \tag{4.3}$$

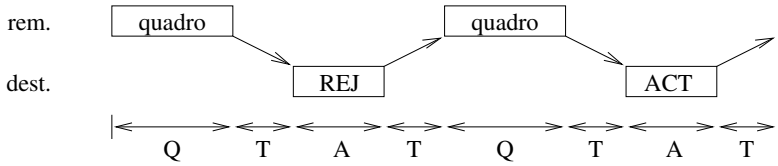
Ignorando o termo $2T$, a eficiência do protocolo é determinada somente pelos tamanhos de prólogo, epílogo e quadro de aceitação. Estes representam o custo adicional devido ao uso do protocolo de enlace de dados. A Equação 4.3 é muito similar à Equação 4.1, a menos do maior detalhamento na atribuição dos custos.

O tempo de propagação pelo meio físico T é pequeno se os ETDs estão geograficamente próximos (<10 milisegundos), podendo chegar ao entorno de 300 milisegundos para transmissões via satélite. O tempo de reversão de linha semi-duplex não pode ser ignorado em linhas discadas por causa dos supressores de eco instalados naquelas linhas (veja a Seção 2.5.2). De qualquer forma, o tempo de reversão de linha t_r reduz a eficiência da comunicação semi-duplex (veja a Figura 3.18, pág. 67). T depende da velocidade de propagação de ondas eletromagnéticas e dos equipamentos que implementam o nível físico (tipo de modem, velocidade de transmissão, etc). Acima de uma certa velocidade de transmissão, as ineficiências devidas ao protocolo de enlace semi-duplex tornam investimentos em modems mais rápidos pouco atrativos porque estes protocolos são mais eficientes a baixas velocidades. A fração do tempo perdida por causa do protocolo tende a aumentar em velocidades de transmissão mais altas. Isto é, D, Q e A diminuem mas T e P permanecem constantes. Se a velocidade de transmissão dobrar, o tempo de uma transação não cai para a metade por causa daqueles termos.

A Figura 4.14 mostra as duas situações onde são necessárias retransmissões. Se o quadro de resposta é danificado, após o decurso do temporizador de transmissão S, o último quadro deve ser retransmitido. Se ocorrerem erros no quadro de dados, o destinatário rejeita o quadro recebido com um quadro de resposta do tipo REJ. O remetente então retransmite o quadro rejeitado. Veja Exercícios 4.9 e 4.10 para o cálculo da eficiência de enlaces semi-duplex em presença de erros.



(a) Retransmissão por perda de resposta.



(b) Retransmissão por erros no quadro.

Figura 4.14: Causas de retransmissões em protocolos semi-duplex.

4.3 Protocolos de Controle de Enlace

Existem três grandes famílias de protocolos de controle de enlace. Os protocolos da primeira família são simples e normalmente empregados na comunicação assíncrona entre computadores pessoais. Tipicamente, estes protocolos permitem a transferência de arquivos entre duas máquinas através de suas interfaces seriais. Um arquivo a ser transferido é dividido em blocos. Cada bloco de dados é precedido por um prólogo e seguido de um epílogo, formando um quadro. Cada quadro é transmitido e reconhecido individualmente. O *tamanho* do bloco de dados é enviado no prólogo de cada bloco. Após o prólogo, o receptor espera pelo número de octetos/caracteres nele especificado. Um protocolo deste tipo é o Kermit, descrito em mais detalhe na Seção 4.3.1.

Os protocolos da segunda família são chamados de protocolos *baseados em caracter* porque caracteres de controle demarcam os campos de cada quadro. O prólogo consiste de um ou mais *caracteres de início*, informações de controle e um *caracter de final*. O bloco de dados é precedido de um caracter de início e seguido de um terminador. O teste de paridade completa o quadro. Devido a possibilidade da ocorrência de uma seqüência de bits no campo de dados igual àquela de um caracter de controle, este tipo de protocolo deve incluir uma forma de diferenciar as ocorrências válidas de caracteres de controle das ocorrências acidentais. Um protocolo é dito *transparente* quando quaisquer seqüências de bits podem ocorrer no campo de dados de um quadro. O protocolo "Binary Synchronous Control" (BSC) é um protocolo desta família, usado em transmissões síncronas em linhas semi-duplex. O BSC é descrito na Seção 4.3.2.

Finalmente, em protocolos da terceira família são *baseados em bit*, porque os campos de controle e de dados ocupam *posições fixas no quadro i.e.,* o campo de dados inicia no n-ésimo bit a partir do início do quadro. Os bits usados para o teste de integridade dos dados são os r-ésimos bits antes do final do quadro. A vantagem de protocolos baseados em bit sobre os protocolos baseados em caracter é a economia na codificação das informações de controle. Além disso, a transparência dos protocolos baseados em bit pode ser obtida de uma forma mais simples do que em protocolos baseados em caracter. A Seção 4.3.3 descreve o "High-level Data Link Control" (HDLC). Este protocolo é usado em transmissões síncronas em linhas duplex e é similar aos protocolos empregados na rede de comutação de pacotes X.25, e em protocolos de comunicação serial como o padrão V.42+V.42bis, discutidos na Seção 4.3.6.

Do ponto de vista de enquadramento, as diferenças principais entre as três famílias estão na forma de sincronização em nível de caracter e em nível de quadro. Nos protocolos assíncronos

(Kermit), cada caracter é enquadrado por bits de início e de final e o intervalo entre dois caracteres pode ser arbitrariamente longo, a menos da temporização de erros. Nos protocolos síncronos (BSC e HDLC), centenas ou milhares de bits são transmitidos sem interrupção. A separação de um quadro do quadro seguinte depende de seqüências especiais de bits. Em protocolos baseados em caracter, caracteres de controle precedem e sucedem os quadros. Nos protocolos baseados em bit, seqüências especiais de bits emolduram os quadros.

As seções seguintes não contém descrições completamente detalhadas dos protocolos. A inteção é mostrar as diferentes técnicas usadas nas implementações de protocolos de comunicação serial ao invés de descrever os protocolos em seus mínimos detalhes.

4.3.1 Protocolo Kermit

O protocolo Kermit foi concebido para a transferência de arquivos entre micro-computadores e computadores de médio e grande porte da Digital e IBM [dC87]. Kermit suporta enlaces ponto-a-ponto com transmissões assíncronas semi-duplex. Embora sua eficiência não seja muito alta, Kermit é simples, robusto e permite a transferência de quaisquer tipos de dados, tais como arquivos de caracteres ASCII de 7 e 8 bits, arquivos binários e executáveis.

Formado dos Quadros

A Figura 4.15 mostra o formato dos quadros empregados no Kermit. Todos os campos do cabeçalho consistem de caracteres visíveis do alfabeto. A marca denota o início de um quadro, e normalmente consiste do caracter SOH=`0x01` ("Start Of Header"). O campo tamanho define o número de caracteres que o seguem no quadro e inclui o campo TDP. Normalmente, o tamanho máximo de um quadro não pode exceder 96 caracteres. Esta limitação decorre da necessidade de se usar caracteres visíveis em todos os campos do quadro. O campo sequência é usado para ordenar os quadros e é um contador de módulo-64. Um contador é usado no sequenciamento dos quadros em cada direção. O campo tipo indica o conteúdo do quadro e contém caracteres que identificam quadros de dados, aceitação, rejeição, etc. O campo dados contém dados, informações de controle relacionadas à transferência de arquivos ou é vazio. O TDP é usado para detectar erros de transmissão e pode ser o teste de paridade longitudinal ou o da divisão polinomial. O valor no campo TDP é computado sobre todos os carateres do quadro exceto marca e o próprio TDP.

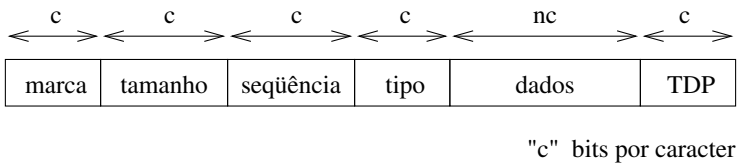


Figura 4.15: Formato de quadro do Kermit.

A Tabela 4.1 mostra os tipos de quadro. Um quadro do tipo D transporta dados do arquivo que está sendo transferido. Um quadro Y indica a aceitação pelo receptor do último quadro enviado. Este quadro também é usado como resposta a um pedido de transmissão. Um quadro N indica a rejeição do último quadro recebido e que este deve ser retransmitido. Um quadro S contém parâmetros a ser usados durante a transferência de arquivos. Um quadro F contém o nome do arquivo a ser transferido.

Um quadro B indica o final da transmissão de um grupo de arquivos (EOT="End Of Transmission"). O fim da transmissão de um arquivo é sinalizado por um quadro Z (EOF="End Of File"). O campo de dados de quadros do tipo X deve ser mostrado no terminal do receptor.

Os atributos de um arquivo (*e.g.* dono, datas, permissões) são transportados por quadros A. O tipo T é reservado. Finalmente, erros catastróficos são sinalizados por quadros do tipo E.

Quadros tipo S são usados por transmissor e receptor para definir os parâmetros operacionais que vão ser usados durante a transferência de arquivos. Nesta *negociação* de parâmetros, dentre outras variáveis, o receptor estabelece o tamanho máximo de quadro em função de sua capacidade de tratamento dos quadros recebidos e o intervalo máximo de espera antes de o receptor iniciar tratamento de erros é negociado.

Todos os caracteres no cabeçalho de um quadro devem ser visíveis. Portanto, a contagem de tamanho de quadro inicia em 32, ou seja, o caracter "espaço" `0x20`). Os quadros mais curtos (*e.g.* ACT, REJ) contém somente os campos sequência, tipo e TDP, tem em seu campo tamanho o caracter '#' (`0x23`). O quadro mais longo tem 96 caracteres, transporta 91 caracteres em dados e o campo tamanho contém o caracter '}' (`0x7d`). Da mesma forma, a sequência de quadros inicia no caracter "espaço" (`0x20`) e termina no caracter '_' (`0x5f`).

tipo	significado
D	dados
Y	aceitação (ACT)
N	rejeição (REJ)
S	"send initiate", inicia transmissão
F	"file header", descritor de arquivo
B	final de transmissão de grupo de arquivos (EOT)
Z	final de transmissão de arquivo (EOF)
X	mostre o texto na tela
A	atributos
T	reservado
E	erro catastrófico

Tabela 4.1: Tipos de quadros do Kermit.

Transparência do protocolo Antes de iniciar a transmissão de dados, o Kermit pré-processa o conteúdo do arquivo para evitar que algum caracter ASCII (ou octeto) presente no arquivo seja mal interpretado pelos sistemas de comunicação das duas máquinas. Os octetos/caracteres que podem ser interpretados como caracteres de controle são precedidos do caracter '#' e o octeto em questão é transformado através de um ou-exclusivo com o valor 0x40. Note que isto deixa o bit mais significativo intacto. Antes de gravar o arquivo em disco, o Kermit do receptor desfaz as modificações introduzidas pelo transmissor. Esta e outras operações garantem que arquivos com caracteres de 7 ou 8 bits sejam transmitidos e recebidos corretamente.

Funcionamento

A interface de usuário do Kermit varia de versão para versão. Na versão para Unix (veja `man kermit`), a velocidade de transmissão, número de bits por caracter, número de bits de final, uso de bit de paridade, etc podem ser escolhidos. Uma vez estabelecida a conexão entre as máquinas, a transferência de arquivos pode iniciar. Como o modo de comunicação é semi-duplex, as duas máquinas se alternam nos papéis de transmissor e receptor. Para evitar confusão, a máquina que envia o arquivo é chamada de *remetente* e a que recebe de *destinatário*.

A Figura 4.16 mostra a sequência de eventos descrita no que segue. O processo de transferência de arquivos se inicia com o envio de um quadro tipo N (REJ) pelo destinatário. As duas máquinas então negociam parâmetros de transmissão (*e.g.* tamanho máximo de quadro, temporização). Quando entram num acordo, o remetente envia ao destinatário um quadro tipo F com

o descritor do arquivo, com seu nome e outros atributos. Se o descritor é recebido corretamente, o remetente inicia a transferência do arquivo. Em geral, um arquivo deve ser segmentado em vários quadros tipo D, que são transmitidos, aceitos e reconhecidos individualmente pelo destinatário. Ao terminar a transferência, o remetente envia um quadro tipo Z (EOF), sinalizando o fim-de-arquivo. Quando o receptor detecta um erro no quadro recém-recebido, este transmite um quadro tipo N (REJ). O transmissor retransmite então o último quadro que enviou.

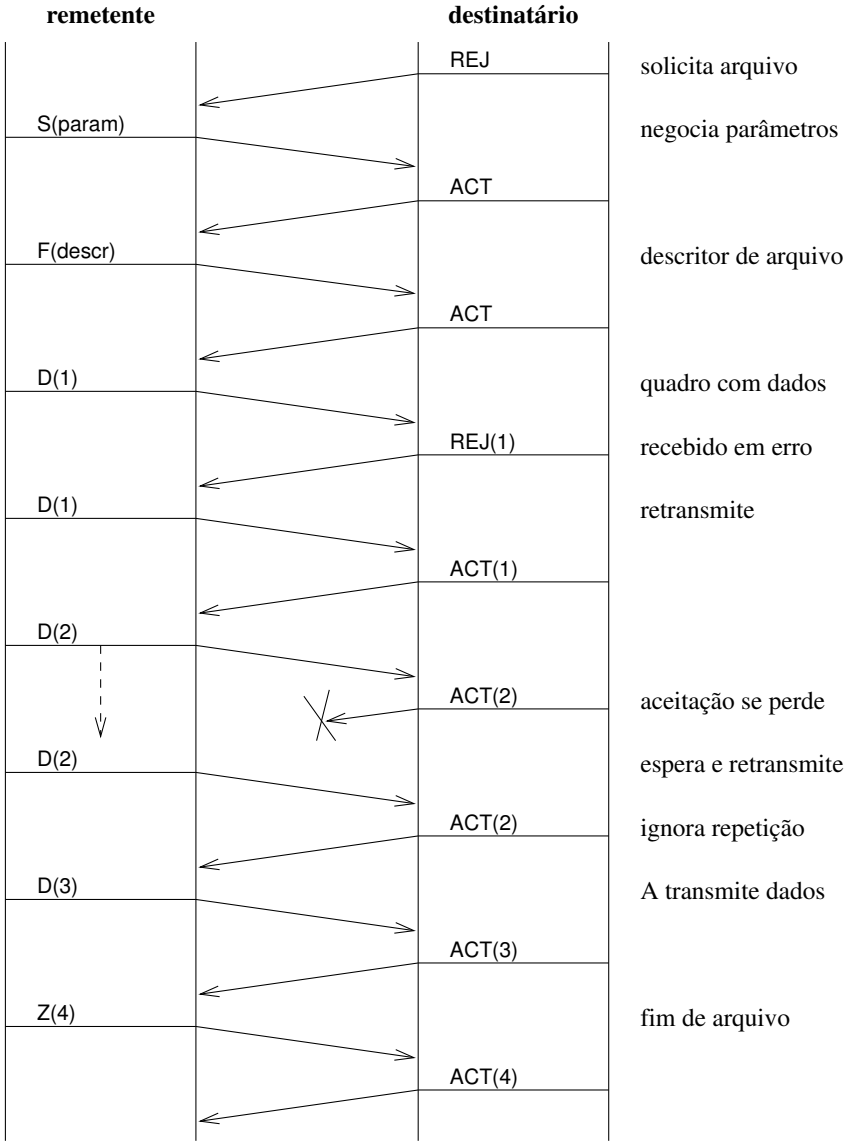


Figura 4.16: Transferência de arquivo com Kermit.

Caso um quadro tipo Y (ACT) seja danificado por um pulso de ruído, o transmissor espera pelo decurso do intervalo especificado no temporizador e retransmite o último quadro. Se aquele já havia sido recebido corretamente pelo destinatário, os dois quadros (original e retransmitido) possuem o mesmo número de sequência. O destinatário ignora o quadro retransmitido, envia novo quadro Y (ACT) e a transferência prossegue normalmente.

Um problema com protocolos baseados no *tamanho* dos quadros, como é o caso do Kermit, decorre da dependência na correção do campo que determina o tamanho do quadro. O protocolo só funciona corretamente se os dois terminais transmitem e recebem o número correto de caracteres em cada quadro. Se o caracter que indica o tamanho do quadro for danificado por

causa de ruído, os protocolos nos dois terminais podem perder o sincronismo. Neste caso, a transferência deve ser re-inicializada.

Existem versões do Kermit para praticamente todos os sistemas operacionais em uso corrente. O executável e os arquivos fonte pode ser encontrados em vários repositórios na Internet. A distribuição oficial pode ser obtida de Kermit Distribution, Columbia University Center for Computing Activities, Watson Laboratory, 7th Floor, 612 West 115th Street, New York, NY 10025, EUA.

4.3.2 Protocolo BSC

Os protocolos de controle de enlace síncronos baseados em caracter são mais eficientes que os protocolos assíncronos (*e.g.* Kermit). Nos protocolos síncronos baseados em caracter, um ou mais caracteres especiais são transmitidos no prólogo e epílogo das mensagens com a finalidade de sincronizar transmissor e receptor enquanto que os caracteres no corpo das mensagens contém somente informação de controle ou dados. A posição de certos campos que contém dados ou controle nos quadros é indicada pela presença de caracteres de controle que os identificam *i.e.,* a posição dos campos de controle e dados nos quadros é variável. Para evitar confusões entre caracteres de controle e caracteres de dados com os mesmos valores, os protocolos contém mecanismos que distinguem os dois tipos de ocorrências.

O protocolo "Binary Synchronous Control" (BSC) foi desenvolvido pela IBM no início da década de 60 e foi muito usado na comunicação entre máquinas de grande porte daquele fabricante [IBM70]. Atualmente, com o aumento da capacidade de processamento dos terminais, o BSC vem sendo substituído pelos protocolos SDLC e HDLC (Seção 4.3.3), que permite uso mais eficiente das linhas de comunicação. O protocolo BSC define um conjunto de regras de controle de enlace de dados baseado em caracter, com transmissão serial síncrona. A transmissão é semi-duplex e o protocolo opera em linhas ponto-a-ponto ou multiponto. Note que o modo de operação é semi-duplex mesmo que sejam usadas duas linhas (quatro fios), e os DTEs devem recuperar o sincronismo a cada reversão de linha. BSC funciona com caracteres dos alfabetos ASCII, EBCDIC ou Transcode. Obviamente, todas as estações num mesmo enlace devem usar o mesmo alfabeto. O receptor deve examinar cada caracter recebido para decidir se o mesmo é um caracter de controle ou dado.

Formato dos Quadros

O formato dos quadros definidos pelo protocolo BSC depende do modo de operação do enlace. Contudo, para garantir a operação correta da transmissão síncrona semi-duplex, alguns caracteres especiais são sempre inseridos antes e depois de cada quadro. A Tabela 4.2 contém a codificação ASCII dos comandos BSC. Do ponto de vista de sincronização em nível de quadro, os dois formatos básicos são mostrados na Figura 4.17. Os caracteres PAD servem para sincronizar os circuitos de relógio de transmissor e receptor. O PAD inicial pode ser `0x55` ou SYN e contém os `0`s e `1`s necessários para sincronizar os relógios. O PAD final deve ser `0xff` e coloca a linha em repouso. Para simplificar a exposição, os caracteres PAD serão omitidos daqui para a frente.

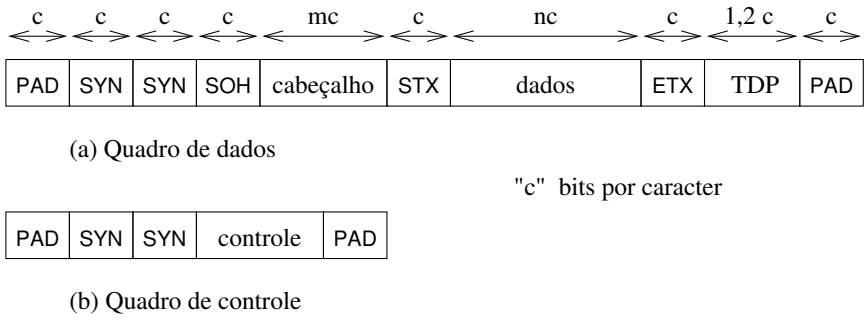


Figura 4.17: Formato dos quadros do protocolo BSC.

Nos quadros de dados, o cabeçalho é opcional e transporta informação utilizada por protocolos de alto nível. O campo de dados é opcional, se houver um de cabeçalho. Quadros de controle não contém um campo para teste de paridade. Se o alfabeto usado for ASCII, cada caracter é acompanhado por um bit de paridade e o bloco de paridade é um octeto com o teste de paridade longitudinal (TPL). Se o alfabeto for EBCDIC (8 bits), dois octetos contém o TDP-16; se o alfabeto for Transcode (6 bits), dois caracteres contém o TDP-12.

Os caracteres de controle que definem os campos dentro dos quadros dependem do modo de operação do enlace. Nos quadros de dados, os campos com o cabeçalho e dados são identificados pelos caracteres especiais SOH 'início de cabeçalho' ("Start of Heading"), STX 'início de texto' ("Start of Text") e ETX 'fim de texto' ("End of Text"). Ou, o campo com cabeçalho ou o campo com dados deve ser não-vazio. Os demais caracteres de controle são mostrados na Tabela 4.2.

comando	significado
SYN	"Synchronize", início de um quadro
SOH	"Start of Header", início do campo de cabeçalho
STX	"Start of Text", início do campo de dados
ETX	"End of Text", indica o final do campo de dados
ETB	"End of Transmission Block", indica o final de um bloco
ACK0	"Positive Acknowledgement", aceite de quadro par (DLE 0)
ACK1	"Positive Acknowledgement", aceite de quadro ímpar (DLE 1)
NAK	"Negative Acknowledgement", rejeita quadro; estação ocupada
ENQ	"Enquiry", requisição de linha, consulta/seleção
DLE	"Data Link Escape", usado para tornar protocolo transparente
WACK	"Wait Before Transmitting", aceite, espere antes de transmitir
TTD	"Temporary Text Delay", transmissão em breve (STX ENQ)

Tabela 4.2: Comandos do protocolo BSC.

Se uma mensagem longa deve ser segmentada em blocos, o campo de dados ou cabeçalho inicia com SOH ou STX e termina com um ETB ("End of Transmission Block"). O último bloco da mensagem deve ser terminado com um ETX. A detecção de ETB ou ETX pelo receptor causam a comutação da linha e o envio de um quadro de resposta aceitando ou rejeitando o último bloco recebido. Uma mensagem pode ainda ser segmentada de forma a melhorar as características de detecção de erros pelo receptor. O cabeçalho ou campo de dados é terminado por um ITB ("Intermediate Transmission Block") e seguido pelo bloco de paridade. Após detectar o ITB, o receptor re-inicializa o cômputo de bloco de paridade. O último bloco é terminado por um ETB ou ETX. Os blocos transmitidos desta forma devem ser tratados como uma única mensagem pelo receptor.

Transparência de Protocolo A presença de caracteres de controle do protocolo (*e.g.* SOH, ENQ) no campo de dados pode causar problemas no funcionamento do enlace. Para tornar o protocolo transparente, *i.e.,* os dados de usuário podem conter *qualquer* seqüência de bits, o protocolo BSC possui um *modo de operação transparente*. Em modo transparente, os comandos do protocolo são precedidos pelo caracter DLE. Assim, os pares DLE-STX e DLE-SOH denotam o início do campo de dado e DLE-ETX e DLE-ETB marcam o final do campo de dados. Se os dados contém o caracter DLE, o montador de quadros no transmissor duplica aquele caracter. Ao detectar o par DLE-DLE, o receptor ignora o primeiro DLE e aceita o segundo como sendo parte dos dados. Os caracteres adicionais não entram no cômputo do bloco de paridade.

Funcionamento

No protocolo BSC, a comunicação somente pode ser estabelecida entre um par de estações a cada vez. O modo de transmissão entre estações é semi-duplex. A estação com o direito de transmitir dados é chamada de *transmissora* e a outra estação é a *receptora*. A cada vez que um quadro com dados é enviado, a linha deve ser revertida para que a estação receptora possa responder se aceita ou não o quadro recém recebido. A estação transmissora abdica do controle da linha ao enviar o quadro de controle com SYN SYN EOT.

O BSC foi projetado para operar de modo diferente em enlaces ponto-a-ponto e multiponto. Em enlaces *ponto-a-ponto*, as duas estações no enlace competem pelo direito de transmitir. Quando uma estação deseja enviar quadros, ela transmite a seqüência SYN SYN ENQ. A outra estação responde com (1) SYN SYN ACK0 se está pronta para receber, (2) SYN SYN NAK, se não está pronta ou (3) SYN SYN WACK se não está pronta mas o transmissor deve tentar novamente em breve. Caso as duas estações desejem transmitir ao mesmo tempo, existe a possibilidade de bloqueio ("deadlock"). Para evitar que isso ocorra, as estações são designadas *primária* e *secunária* e a estação primária tem prioridade para transmitir em caso de bloqueio.

Em enlaces *multiponto*, a troca de informações ocorre sempre entre a primária e uma das secundárias. A estação primária adquire dados das secundárias ao consultar cada uma delas individualmente. A estação primária seleciona uma secundária quando deseja lhe transmitir dados. O quadro com a consulta e suas possíveis respostas são

SYN SYN EOT PAD SYN SYN endereço ENQ	consulta
SYN SYN [cabeçalho] [dados] paridade	cabeçalho/dados
SYN SYN EOT	nada a transmitir
SYN SYN STX ENQ	"Temporary Text Delay" (TTD)

O endereço consiste de uma seqüência de 1 a 7 caracteres e pode identificar um de vários dispositivos em uma estação. A estação pode enviar um quadro com dados, avisar que não tem nada a transmitir ou, avisar que está temporariamente impossibilitada de transmitir mas que necessita reter o controle da linha (TTD).

A seleção de uma secundária pela primária usa o mesmo tipo de quadro que uma consulta. As respostas contudo são diferentes:

SYN SYN EOT PAD SYN SYN endereço ENQ	seleção
SYN SYN ACK0	pronta para receber
SYN SYN NAK	não está pronta para receber
SYN SYN WACK	espere e tente novamente

A secundária pode responder que está pronta para receber dados (ACK0), que não está pronta para receber (NAK) ou, que a primária deve tentar novamente. Neste último caso, a primária envia o caracter ENQ até que a secundária responda com ACK0 ou, a primária interrompe a transação com um comando EOT.

Um enlace BSC entre duas estações funciona em um de dois modos. O enlace fica em *modo controle* quando é usado por uma estação primária para controlar a operação do enlace, durante consulta e seleção, por exemplo. O modo *mensagem* é usado para a transmissão de dados entre duas estações. Quando uma secundária recebe uma consulta, se ela tiver dados a transmitir, estes são transmitidos e o enlace permanece no modo mensagem até que a secundária envie um EOT sinalizando o final dos dados, quando o enlace retorna ao modo controle. Enquanto um par de estações estiver em um enlace no modo mensagem, todas as demais estações devem permanecer inativas. A Figura 4.18 mostra um exemplo do funcionamento do protocolo BSC numa linha multiponto. O enlace consiste de três estações, a estação primária $\mathcal{A}$ e duas secundárias $\mathcal{B}$ e $\mathcal{C}$. Os caracteres SYN foram omitidos para simplificar o diagrama.

Note que existem dois comandos de aceitação de quadros, ACK0 e ACK1. Ambos são necessários para evitar a duplicação de quadros em caso de perda de uma aceitação. Considere um protocolo com somente um comando de aceitação. A seguinte seqüência de eventos pode provocar a duplicação de um quadro. A estação $\mathcal{A}$ envia o quadro m_4 para a estação $\mathcal{B}$, que responde com um ACK. Ao receber o ACK, a estação $\mathcal{A}$ envia o quadro m_5, que tem sua recepção reconhecida com outro ACK. Devido a ruído na linha, este ACK não é recebido por $\mathcal{A}$. Após expirar o intervalo de espera por resposta, $\mathcal{A}$ retransmite m_5. A estação $\mathcal{B}$ aceita m_5 e fica assim com duas cópias daquele quadro. Para evitar que isso ocorra, o protocolo BSC emprega números de seqüência nos quadros. Quadros e ACKs recebidos fora de ordem indicam a ocorrência de algum problema. Como a operação do BSC é semi-duplex, somente pode haver um quadro em trânsito a cada instante e uma seqüência com dois números é o suficiente para evitar o problema descrito acima. Ao receber um quadro com número de seqüência par, o receptor responde com um ACK0. Ao receber um quadro com número de seqüência ímpar, o receptor responde com ACK1. A Figura 4.19 mostra um diagrama de tempo com os eventos descritos acima.

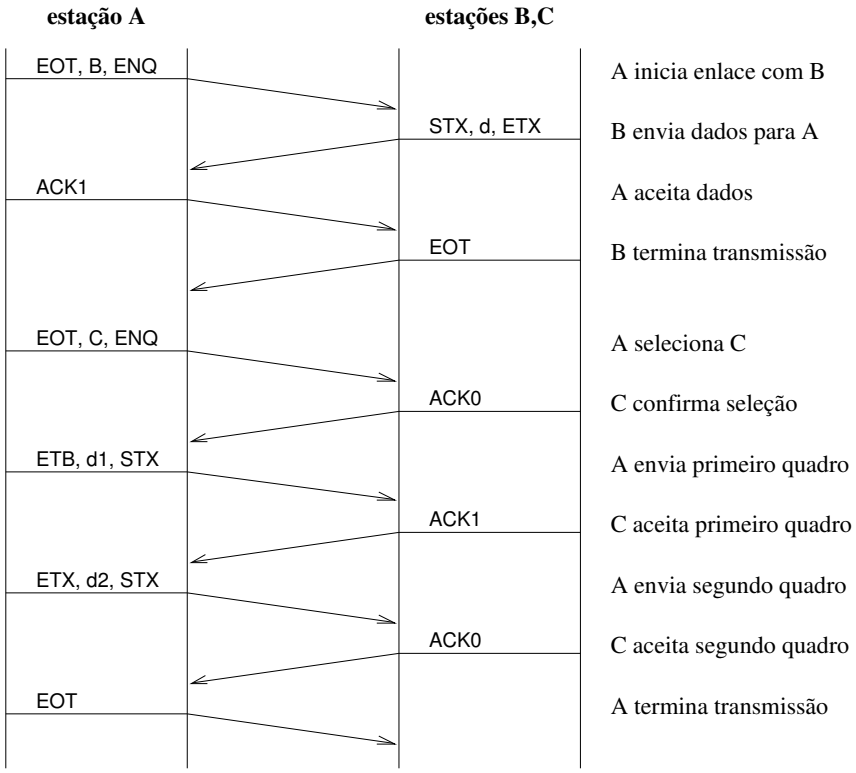


Figura 4.18: Operação de um enlace BSC multiponto.

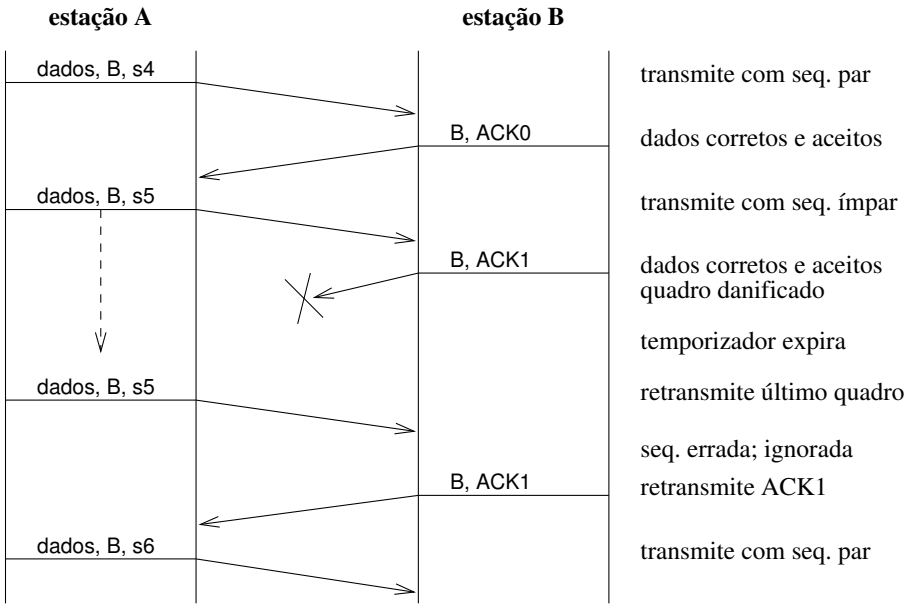


Figura 4.19: Operação de um enlace BSC com perda de quadro de aceitação.

4.3.3 Protocolo HDLC

O protocolo *High Level Data Link Control* (HDLC) e seus descendentes são o resultado da padronização internacional do protocolo *Synchronous Data Link Control* (SDLC), introduzido pela IBM em meados da década de 70 [IBM92]. Este trabalho resultou nos padrões ISO/IEC 3309, 4335 e 7809 [ISO84b, ISO93b, ISO93a]. As diferenças entre SDLC, HDLC e seus descendentes são discutidas no final desta seção.

Os protocolos da família HDLC são síncronos e suportam a detecção e recuperação de erros com o método de pedido automático de repetição em caso de erros (PAR). Estes protocolos de controle de enlace são mais eficientes que os protocolos baseados em caracter como o BSC. Nos protocolos *baseados em bit*, a posição dos campos de um quadro é fixa em relação ao seu início e os comandos são representados por combinações de bits em posições específicas. Além da codificação dos comandos, os protocolos da família HDLC operam em modo duplex e usam janelas deslizantes para melhorar a utilização da linha. A combinação destas técnicas resulta em maior eficiência do que aquela dos protocolos baseados em caracter.

Formato dos Quadros

O protocolo HDLC emprega transmissão síncrona com um único formato de quadro para a troca de dados e controle. A Figura 4.20 mostra o formato dos quadros HDLC. O tamanho mínimo de um quadro é 48 bits mais os dois marcadores. O *marcador* ("flag") consiste da seqüência de bits 01111110 e delimita os quadros. Um mesmo marcador pode ser usado para marcar o final de um quadro e o início do quadro subseqüente. Quando não há quadros a enviar, marcadores são transmitidos continuamente para evitar que os relógios das estações saiam de sincronismo. O TDP é computado sobre todos os bits do quadro exceto os marcadores. O teste usado é o TDP-CCITT.

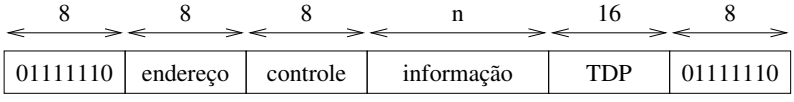


Figura 4.20: Formato dos quadros HDLC.

O campo de *endereço* identifica a estação secundária que enviou o quadro, ou a secundária que irá recebê-lo. O octeto 0xff é o endereço para difusão e quadros com este endereço são aceitos por todas as estações secundárias no enlace. Em operação no modo extendido (veja abaixo), o campo de endereço pode ter um número variável de octetos. O conteúdo e o tamanho do campo de *informação* não é especificado. O campo de informação pode ter tamanho zero; o tamanho máximo é limitado pela técnica de detecção de erros. Usualmente, este campo contém um número inteiro de octetos.

O campo de *controle* diferencia três tipos de quadro, mostrados na Figura 4.21. Os dois primeiros bits do campo identificam o tipo de quadro. Os *quadros de informação* (quadros-I) tem campo de informação não-vazio e transportam os dados propriamente ditos. O campo NS contém o *número de seqüência* do quadro. Quadros-I também transportam informação de controle de fluxo e recuperação de erros (PAR) com janelas deslizantes e aceitação inclusiva. O número de seqüência do próximo quadro esperado pelo destinatário é transportado "de carona" no campo NA, *número de aceitação*, em um quadro que viaja do destinatário para o remetente. Dessa forma, a aceitação de um ou mais quadros pelo destinatário pode viajar de carona em um quadro que transmporta dados na direção do remetente.

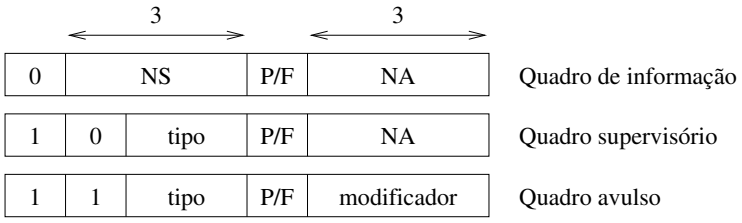


Figura 4.21: Formato do campo de controle.

Os *quadros supervisórios* (quadros-S) transportam informação de PAR no campo NA para a aceitação ou rejeição de quadros recebidos quando o método da carona não pode ser usado (*i.e.,* quando a estação não tem quadros de informação a transmitir). Os comandos transportados em quadros-S tem a função de controle de fluxo, recuperação de erros e consulta e seleção.

Os *quadros avulsos* (quadros-U "unnumbered") transportam informações de controle especiais para o estabelecimento dos enlaces e seleção do modo de operação. Os comandos transportados por quadros-U servem para inicializar e terminar um enlace lógico, determinar o modo de funcionamento do enlace e para resolver impasses e situações de erros catastróficos.

O bit P/F ("poll/final") tem dupla função. Em quadros emitidos por uma estação primária, este bit é uma consulta ("poll") que deve ser respondida pela secundária indicada no campo de endereço. Em quadros emitidos por uma estação secundária, se este bit é 1, ele sinaliza o último de uma série de quadros ("final").

Quadros-I e quadros-S tem números de seqüência NA de 3 bits e as janelas deslizantes tem capacidade para 7 quadros; o uso de 7 bits nos números de seqüência também é possível no modo extendido, permitindo o uso de janelas com 127 quadros. Os quadros de supervisão transportam quatro comandos: RR, RNR, REJ e SREJ (veja abaixo). Os quadros avulsos transportam os demais comandos. O número de comandos avulsos é menor que as 32 combinações possíveis com os 2 bits do campo tipo e os 3 bits do modificador.

Transparência de protocolo Para evitar confusão entre os marcadores que delimitam os quadros e eventuais seqüências de bits no campo de informação com exatamente os bits 01111110, o HDLC emprega a técnica de *enxerto de bits* ("bit stuffing"). O transmissor sempre insere

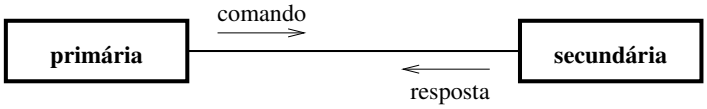
um 0 extra após cada ocorrência de cinco 1s consecutivos, exceto nos marcadores. Após detectar o marcador, o receptor observa as seqüências de cinco 1s. Se o sexto bit é um 0, o 0 é removido porque ele foi enxertado pelo transmissor. Se o sexto bit é 1 e o sétimo 0, a seqüência é um marcador e delimita um quadro. Se o sexto e sétimo bits são 1, ou transmissor está enviando um comando abort, ou ocorreu um erro de transmissão. Nos dois casos o quadro é descartado e o receptor fica a espera do próximo marcador. Se o receptor detecta uma seqüência de quinze 1s, o transmissor está colocando a linha em estado inativo, talvez em preparação para reverter a linha em enlaces semi-duplex.

Os bits enxertados pelo transmissor não são considerados no cálculo do TDP. Os bits enxertados são inseridos imediatamente antes da transmissão do quadro pelo meio físico e são removidos pelo receptor antes de qualquer processamento (*i.e.,* cálculo do TDP). Esta é uma técnica simples para prover a transparência do protocolo. Nenhum procedimento especial é necessário para o envio de dados que possam confundir o receptor. Para um quadro com 1000 bits, [BG92] mostra que o enxerto de bits introduz, em média, 23 bits adicionais. Note que a ocorrência de erros (*i.e.,* bits invertidos) pode causar erros de sincronização ao criar ou destruir marcadores.

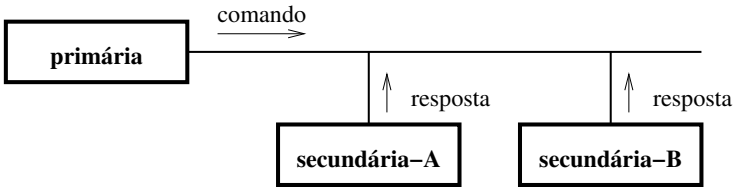
Funcionamento

O protocolo HDLC define uma *estação* como sendo a combinação de um DTE com um DCE. São três os tipos de estações. Uma estação *primária* controla a operação do enlace. Os quadros emitidos por uma estação primária são chamados de *comandos*. Uma estação *secundária* opera sob o controle da estação primária. Quadros emitidos por estações secundárias são chamados de *respostas*. Em enlaces multiponto, a estação primária mantém um enlace lógico separado com cada uma das estações secundárias. Uma estação *combinada* tem as funções de primária e secundária, podendo emitir tanto comandos como respostas.

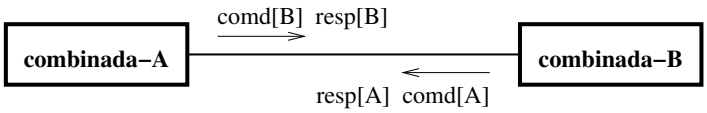
O HDLC pode ser usado com enlaces em uma de duas configurações possíveis. Uma configuração *desbalanceada* consiste de uma estação primária e uma ou mais estações secundárias. Enlaces desbalanceados podem ser usados com ligações ponto-a-ponto ou multiponto e as transmissões podem ser semi-duplex ou duplex. Uma configuração *balanceada* consiste de duas estações combinadas, com transmissões semi-duplex ou duplex. Configurações balanceadas podem ser usadas somente em ligações ponto-a-ponto. A Figura 4.22 mostra as três configurações.



(a) Desbalanceada ponto–a–ponto



(b) Desbalanceada multiponto



(c) Balanceada

Figura 4.22: Configurações de enlaces HDLC.

O HDLC suporta três modos de transferência de dados, dependendo se a configuração do enlace é balanceada ou não e da disciplina que controla o acesso ao meio de transmissão. As transferências em *"Normal Response Mode"* (NRM), ou Modo Normal de Resposta (MNR), podem ser usadas com configurações desbalanceadas. Neste modo, a primária pode iniciar uma transferência para uma secundária, mas a secundária só pode transmitir quadros após uma consulta pela primária. Este modo é usado principalmente em linhas multiponto onde uma estação primária controla várias estações secundárias sub uma disciplina de linha tipo consulta e seleção ("poll/select"). O modo *"Asynchronous Balanced Mode"* (ABM), ou Modo Balanceado Assíncrono (MBA), é usado em configurações balanceadas em linhas ponto-a-ponto. Neste caso, as duas estações são do tipo combinada e podem transmitir sem consulta prévia, o que aumenta a utilização da linha. Finalmente, transferências em *"Asynchronous Response Mode"* (ARM), ou Modo de Resposta Assíncrona (MRA), são usadas em configurações desbalanceadas onde qualquer estação secundária pode transmitir sem consulta prévia pela estação primária. Tipicamente, MRA é usado com uma topologia em estrela onde as secundárias podem iniciar as transações independentemente de consulta prévia pela primária.

Comandos em quadros de informação Os quadros de informação não transportam comandos propriamente ditos. Contudo, o bit `P/F` é usado pela primária para solicitar uma resposta de uma secundária. Quando a secundária recebe um quadro com uma consulta (`P=1`), esta pode transmitir tantos quadros quantos sua janela de transmissão permitir. Em enlaces duplex, enquanto a primária aceitar quadros, a secundária pode seguir transmitindo. Em enlaces semiduplex, quando a janela de transmissão se esvazia, a secundária *deve* sustar suas transmissões e reverter a linha. Quando da transmissão do último quadro de uma série, o bit `F=1` serve de indicação à primária de que esta pode reverter a linha.

Comandos em quadros supervisórios Os comandos suportados pelo protocolo HDLC são descritos a seguir. A Tabela 4.3 contém os comandos contidos em quadros supervisórios. Estes

comandos são usados para implementar controle de fluxo e a recuperação de erros com pedidos automáticos de repetição do tipo volta-N e repetição seletiva. Como o HDLC usa aceitação inclusiva, os quadros recebidos com número de seqüência menor que o indicado no campo NA são todos implicitamente aceitos. A estação espera receber o quadro cujo número de seqüência é o indicado no campo NA.

comd.	significado
RR	"Receive Ready", aceite, espera quadro de informação
RNR	"Receive Not Ready", aceite, estação não está pronta
REJ	"Reject", rejeição, volte atrás N quadros
SREJ	"Selective Reject", rejeição seletiva

Tabela 4.3: Comandos contidos em quadros supervisórios.

Os comandos empregados no controle de fluxo são "Receive Ready" (RR) e "Receive Not Ready" (RNR). O comando RR é usado para a aceitação dos quadros já recebidos e indica que a estação espera pelo quadro com o número de seqüência indicado no campo NA. Este comando também é usado pela estação primária para consultar as secundárias. O comando RNR indica que a estação não está apta a receber novos quadros. A mudança na condição indicada por RNR pode ser sinalizada por um quadro com o comando RR. Quadros com o comando RNR também indicam a aceitação de quadros previamente recebidos. O comando "Reject" (REJ) indica que os quadros com numeração maior ou igual àquela indicada no campo NA devem ser retransmitidos. Os quadros com numeração menor ou igual a $\mathsf{NA}-1$ são aceitos. O comando "Selective Reject" (SREJ) indica que o quadro de número indicado no campo NA deve ser retransmitido. Da mesma forma que com o comando REJ, quadros com numeração menor ou igual a $\mathsf{NA}-1$ são aceitos.

Comandos em quadros avulsos A Tabela 4.4 mostra os comandos do protocolo HDLC transportados em quadros avulsos ("unnumbered"). Estes comandos podem ser divididos em quatro grupos. Os comandos que estabelecem o modo de operação do enlace são SNRM, SARM, SABM, SIM, DISC. Além destes, os comandos SNRME, SARME e SABME determinam a operação em *modo extendido*. Neste modo, o campo de controle dos quadros tem 16 bits ao invés de 8 e os campos NS e NA tem 7 bits ao invés de 3. Dessa forma, as janelas deslizantes comportam 127 quadros ao invés dos 7 quadros do modo não-extendido. No segundo grupo estão os comandos UI e UP que são usados para a transmissão de dados em quadros avulsos. No terceiro grupo, o comando RSET é usado para a recuperação de erros. Finalmente, os comandos XID e TEST são usados para o estabelecimento e manutenção do enlace.

comd.	significado
	Comandos
SNRM	"Set Normal Response Mode", Modo Normal de Resposta
SARM	"Set Asynchronous Response Mode", Modo de Resposta Assíncrono
SABM	"Set Asynchronous Balanced Mode", Modo de Resp. Assíncr. Balanceado
SIM	"Set Initialization Mode", inicializa controle de enlace na estação
DISC	"Disconnect", termina enlace lógico
UP	"Unnumbered Poll", consulta não seqüenciada
RSET	"Reset", inicia recuperação de erros, faz NS = NA = 0
	Respostas
UA	"Unnumbered Acknowledgement", aceita comando de seleção de modo
DM	"Disconnect Mode", secundária está logicamente desconectada
RD	"Request Disconnect", pedido de desconexão (DISC)
RIM	"Request for Initialization Mode", necessita re-inicialização (SIM)
FRMR	"Frame Reject", indica recepção de quadro inválido
	Comandos e/ou Respostas
UI	"Unnumbered Information", quadro com informação não seqüenciada
XID	"Exchange Identification", solicita e/ou informa identidade e status
TEST	"Test", troca quadros com campo de informação idênticos

Tabela 4.4: Comandos e respostas contidos em quadros avulsos.

O comando SNRM ("Set Normal Response Mode") coloca a secundária indicada pelo campo de endereço em Modo Normal de Resposta (MNR). A secundária não pode enviar nenhum quadro sem solicitação prévia pela primária. Em MNR, a primária tem controle total sobre o tráfego de quadros no enlace. O comando SARM ("Set Asynchronous Response Mode") coloca a secundária em Modo de Resposta Assíncrono (MRA), o que permite que a secundária transmita sem consulta prévia. Este comando cria dois enlaces desbalanceados e portanto deve ser emitido pelas duas estações que se conectam ao enlace lógico ($\mathcal{A} \rightarrow \mathcal{B}$ e $\mathcal{B} \rightarrow \mathcal{A}$). O comando SABM ("Set Asynchronous Balanced Mode") coloca o enlace em Modo Balanceado Assíncrono (MBA) estabelecendo a igualdade entre as estações no enlace. As duas estações combinadas podem transmitir sem consulta prévia. O comando SIM ("Set Initialization Mode") inicializa controle de enlace na estação selecionada pelo campo de endereço. A resposta a esse comando deve ser um quadro com o comando UA. O comando DISC ("Disconnect"), emitido pela primária, coloca a secundária em modo desconectado, terminando o enlace lógico. A resposta deve um comando UA.

O comando UP ("Unnumbered Poll") é uma consulta não-seqüenciada que prescinde de resposta se o bit P=0. Um quadro com o comando UI ("Unnumbered Information") transporta informação não seqüenciada. O comando RSET ("Reset") faz com que NS = 0 no transmissor e NA = 0 no receptor. Os quadros que não haviam sido aceitos até a emissão do RSET permanecem sem aceitação. O comando XID ("Exchange Station Identification") é usado para solicitar e/ou informar a identidade e o status da estação que o emite. Este comando é usado em linhas discadas para identificar o ETD remoto. O comando TEST solicita um quadro de resposta cujo campo de informação é idêntico ao do comando TEST.

Operação desbalanceada Após a inicialização do enlace no modo de operação apropriado, as estações podem iniciar suas transmissões. Num enlace desbalanceado operando em Modo Normal de Resposta (MNR), a primária consulta as secundárias ao transmitir um quadro de informação (quadro-I) ou um quadro supervisório (quadro-S) com o bit P=1 ("poll"). Ao receber um quadro com P=1, a secundária deve responder com (1) um ou mais quadros-I contendo dados,

ou (2) com um quadro-S indicando que não tem dados a transmitir, ou (3) que está ocupada e não pode receber quadros naquele momento. A secundária pode transmitir vários quadros-I a cada consulta. O último quadro de uma série deve ter o bit F=1 ("final").

A Figura 4.23, na página 110, mostra uma troca de dados entre uma estação primária $\mathcal{A}$ e duas estações secundárias $\mathcal{B}$ e $\mathcal{C}$, em Modo Normal de Resposta. Para cada quadro, são mostrados o campo de endereço, o comando (em minúsculas), os números de seqüência de envio e de aceitação (NS, NA) e P ou F se o bit "poll/final" é 1. Para simplificar a explicação, a figura mostra operação semi-duplex no enlace. Nada impede de a primária consultar a estação $\mathcal{B}$ ao mesmo tempo em que recebe quadros de $\mathcal{C}$. Note que os números de seqüência são diferentes para as estações $\mathcal{B}$ e $\mathcal{C}$. Nos quadros emitidos pela primária, o endereço é o da secundária a que o quadro se destina; nos quadros emitidos pelas secundárias, o endereço identifica a estação remetente. O campo NS contém o número de seqüência de transmissão do quadro-I; o campo NA indica o número de seqüência do quadro esperado pelo receptor e a aceitação implícita dos quadros de número menor que NA. A aceitação dos quadros emitidos pela estação $\mathcal{Y}$ que são recebidos corretamente pela estação $\mathcal{X}$, viajam de carona em quadros enviados de $\mathcal{X}$ para $\mathcal{Y}$.

No diagrama da Figura 4.23, a primária estabelece enlace MNR com a estação $\mathcal{B}$. O primeiro quadro transmitido para $\mathcal{C}$ se perde, o temporizador de $\mathcal{A}$ expira, o quadro é então retransmitido e aceito por $\mathcal{C}$. A primária envia três quadros para $\mathcal{B}$, com uma consulta no terceiro quadro. O quadro [B(1, 0)] se perde na transmissão. A estação $\mathcal{B}$ responde com dois quadros-I, um com a aceitação do quadro [0], que se perde, e outro indicando que é o último quadro (F=1). Nesse meio tempo, a primária consulta $\mathcal{C}$, que responde com um quadro. A primária volta atrás e retransmite os dois quadros que ainda não foram aceitos pela estação $\mathcal{B}$, enquanto solicita a retransmissão do quadro [B(0, 1)] e subseqüentes. $\mathcal{B}$ retransmite os dois primeiros quadros e mais um novo, que são aceitos pela primária [B(rr, 3)]. A estação $\mathcal{A}$ envia um quadro-I para $\mathcal{C}$ e um quadro-U para $\mathcal{B}$, solicitando a interrupção do enlace lógico. A estação $\mathcal{C}$ aceita o último quadro recebido.

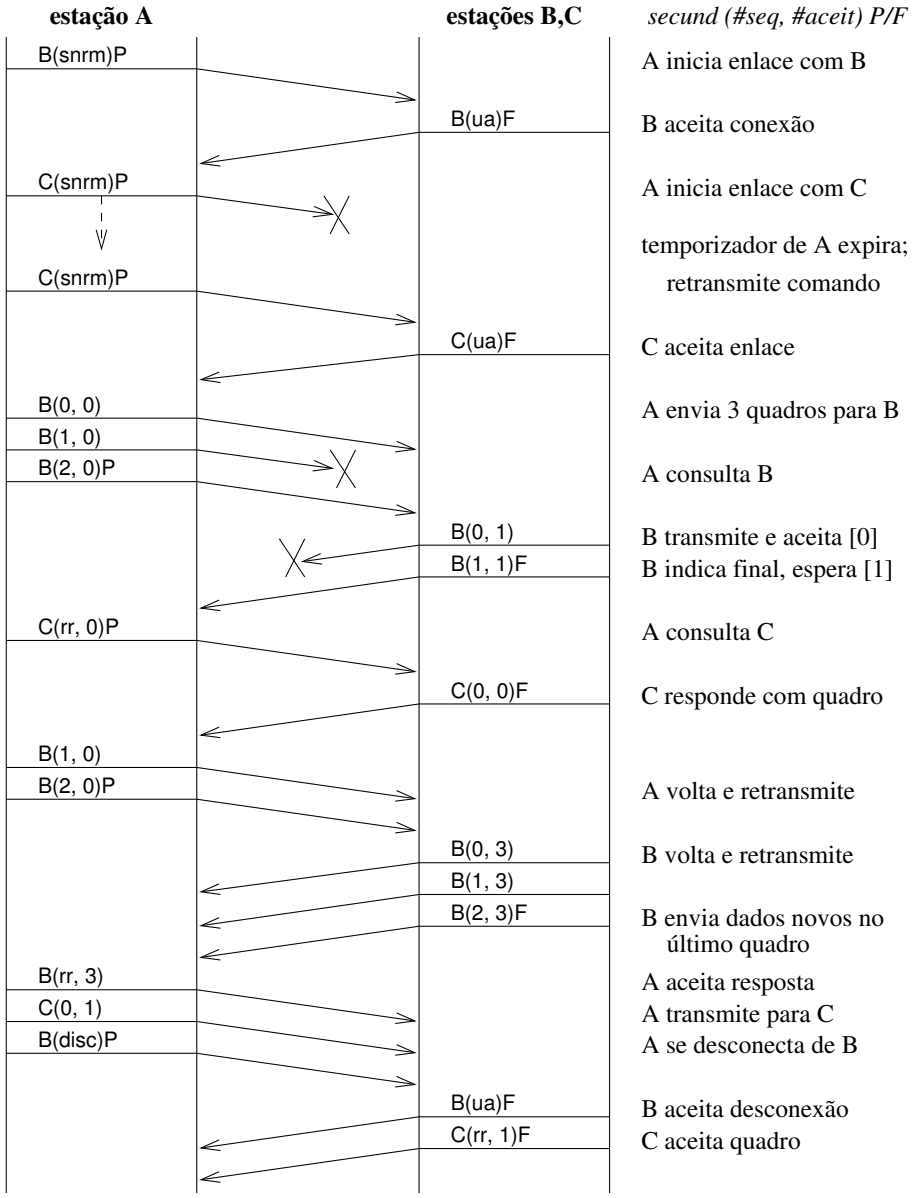


Figura 4.23: Comunicação em Modo Normal de Resposta (MNR).

Operação balanceada A operação de um enlace duplex entre duas estações combinadas em Modo Balanceado Assíncrono (MBA) é mostrada na Figura 4.24 (pág 111). De cada quadro, a figura mostra o campo de endereço, o comando (em minúsculas), os números de seqüência de envio e de aceitação (NS, NA) e P ou F se o bit "poll/final" é 1. O modo de operação MBA é o que permite uma maior utilização da linha porque a troca de quadros entre as duas estações independe de consultas e/ou seleções (*i.e.,* o estado do bit P/F não subverte o fluxo de mensagens). A figura também mostra os métodos de controle de fluxo volta-N e repetição seletiva, vistos na Seção 4.2.3.

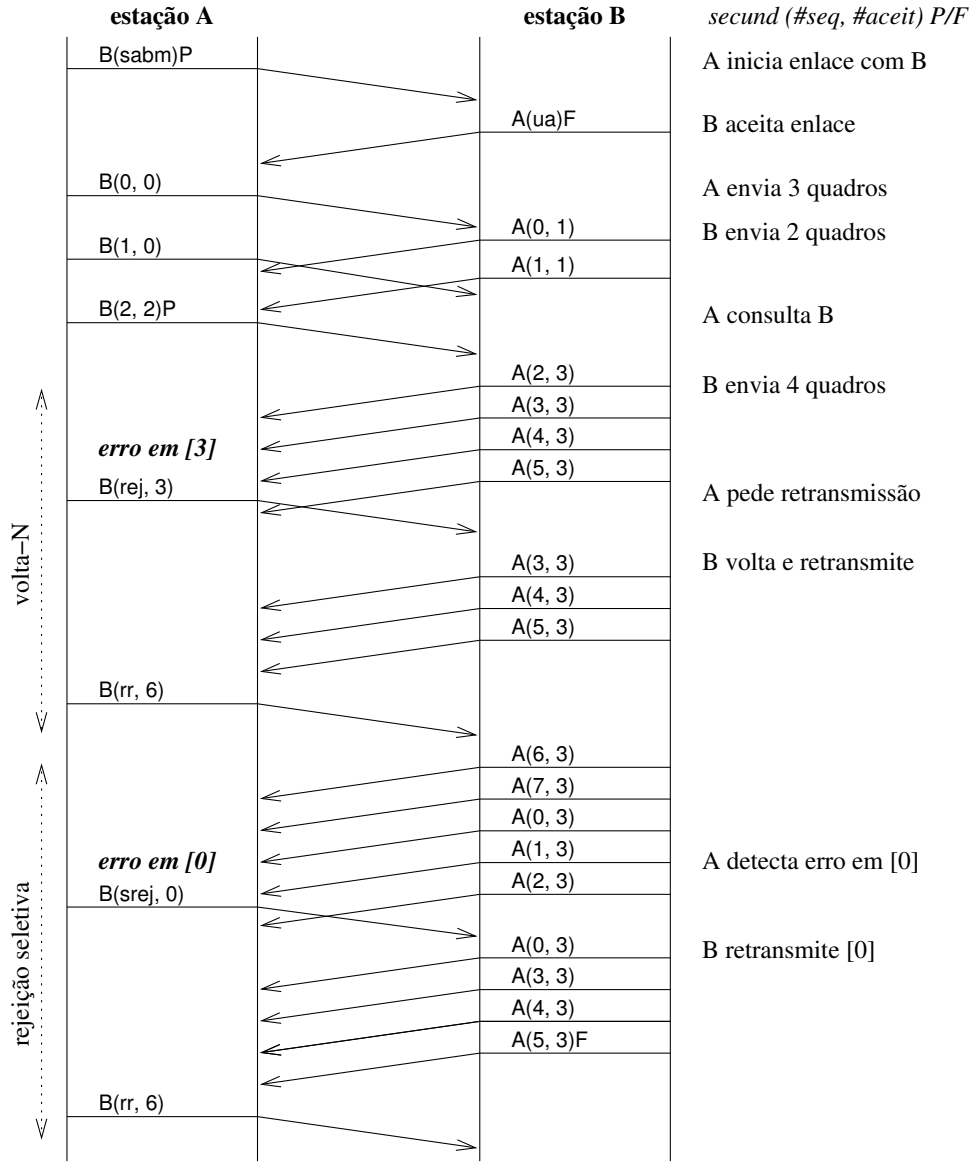


Figura 4.24: Comunicação em Modo Balanceado Assíncrono (MBA).

A estação $\mathcal{A}$ inicializa o enlace ao enviar um quadro avulso com o comando SABM e solicitando uma resposta (P=1). A estação $\mathcal{B}$ responde com outro quadro avulso aceitando o estabelecimento do enlace. Os contadores correspondentes ao número de seqüência NS e ao número de seqüência de aceitação NA são inicializados em 0. As duas estações iniciam então a troca de quadros em modo duplex. Os eventos mostrados na parte central da Figura 4.24 são um exemplo de controle de fluxo com volta-N. Com este método, quando o destinatário detecta erro em um quadro recebido, este envia um quadro ao remetente com a aceitação dos quadros que imediatamente precedem o quadro em erro. O remetente volta atrás e retransmite o quadro em erro e todos que o sucedem. Esse método de controle de fluxo garante que a seqüência dos quadros em posse do destinatário esteja sempre correta. Por outro lado, os quadros transmitidos após o quadro em erro mas que foram recebidos corretamente devem ser retransmitidos. Quando a estação $\mathcal{A}$ detecta um erro de transmissão no quadro [3], ela envia um quadro-S com o comando REJ aceitando implicitamente o quadro [2] porém rejeitando o quadro [3]. A estação $\mathcal{B}$ volta atrás e retransmite os quadros [3] a [5], apesar de que os quadros [4] e [5] haviam sido recebidos sem erro.

O controle de fluxo com retransmissão seletiva é mostrado no terço inferior da Figura 4.24. Após a aceitação dos quadros até o de número [5], a estação $\mathcal{B}$ envia oito quadros, numerados de [6] a [5]. A estação $\mathcal{A}$ detecta um erro no quadro [0] e envia um quadro supervisório com o comando SREJ, pedindo a retransmissão daquele quadro. A estação $\mathcal{B}$ ainda envia os quadros [1] e [2] antes de receber o comando SREJ. Após a retransmissão do quadro [0], $\mathcal{B}$ prossegue com os quadros [3] a [5], que são aceitos por $\mathcal{A}$.

4.3.4 A Família HDLC

O protocolo HDLC resultou da padronização internacional do protocolo *Synchronous Data Link Control* (SDLC) de propriedade da IBM [IBM92]. SDLC está na base do Systems Network Architecture (SNA), que é uma família de produtos da IBM para serviços de teleprocessamento e redes de computadores. As primeiras redes de dados instaladas consistiam de ligações seriais ponto-a-ponto entre computadores de grande porte ou de linhas multiponto onde um controlador gerenciava vários terminais. Este tipo de aplicação se reflete nas características e serviços oferecidos pelo SDLC.

A própria IBM propôs a padronização internacional do SDLC com algumas alterações. Deste processo resultou o HDLC, definido pelos padrões ISO 3309, 4335 e 7809. O CCITT introduziu algumas alterações no HDLC e publicou o padrão *Link Access Procedure* (LAP), para ser usado no controle de enlace de dados da rede de comutação de pacotes X.25 [Sta92]. O LAP foi por sua vez modificado até resultar no protocolo *Link Access Procedure Balanced* (LAPB), posteriormente adotado para uso na rede X.25, ao invés do LAP. O IEEE introduziu modificações no LAPB e publicou o padrão *IEEE 802.2 – Logical Link Control*. O protocolo *Link Access Procedure – D Channel* (LAPD) também resulta de modificações e extensões do HDLC e é usado na Rede Digital de Serviços Integrados (RDSI) [Mon94]. O protocolo *Link Access Procedure for Modems* (LAPM), publicado pelo ITU-T, é descrito abaixo.

Com o advento das redes locais, a funcionalidade dos níveis inferiores da pilha de protocolos teve de ser adaptada aos vários meios físicos empregados. Apesar da variedade de implementações possíveis para o Nível Físico, é interessante que a interface com o Nível de Rede seja independente do Nível Físico. Tendo em vista estas considerações, o padrão IEEE 802.2 para o Nível de Enlace para redes locais foi publicado pelo IEEE e posteriormente adotado pela ANSI e ISO/IEC [ISO94]. O padrão 802.2 é baseado no HDLC e será discutido na Seção 5.2. Outros dois protocolos usados em redes locais e de longa distância e que também são baseados no HDLC, são o Serial Line IP (SLIP) [Rom93] e o Point-to-Point Protocol (PPP) [Sim94]. Ambos são usados para transportar datagramas IP [Pos81a] através de enlaces em linhas seriais.

4.3.5 Padrão CCITT V.42 – LAPM

A recomendação V.42 do CCITT define o protocolo "Link Access Procedure for Modems" (LAPM) para o controle de enlace entre modems. O LAPM usa o Modo Balanceado Assíncrono (MBA) para prover a detecção e correção de erros de transmissão entre pares de modems V.42. O uso da recomendação V.42 implica na transformação da transmissão assíncrona que ocorre através da interface EIA-232 em transmissão síncrona. Essa transformação é especificada na recomendação CCITT V.14. A Figura 4.25 mostra a ligação entre dois ETDs através de ECDs V.42.

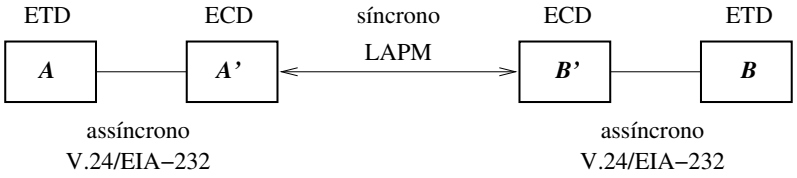


Figura 4.25: Enlace entre ETDs através de ECDs V.42.

A comunicação entre os ETDs $\mathcal{A}$ e $\mathcal{B}$, interligados por ECDs V.42 se processa em dois modos. O ETD $\mathcal{A}$ se comunica assincronamente com o ECD $\mathcal{A}'$ através de sua interface EIA-232-D. O mesmo ocorre com o ETD $\mathcal{B}$ e o ECD $\mathcal{B}'$. Os dois ECDs se comunicam síncronamente em modo de resposta assíncrono balanceado. Dados enviados assincronamente por $\mathcal{A}$ são convertidos para a transmissão síncrona através da linha por $\mathcal{A}'$, reconvertidos para o formato assíncrono por $\mathcal{B}'$ e entregues a $\mathcal{B}$. A conversão assíncrono-síncrono ocorre internamente ao modem antes do estágio de modulação.

Pelo lado assíncrono, os ECDs usam 10 bits/caracter (8 bits/caracter mais bits de início e final). No lado síncrono, os ECDs se comunicam através de quadros com três octetos de cabeçalho, um bloco de dados e dois octetos do TDP-CCITT. Cada quadro é precedido e sucedido de um ou mais marcadores. Quando ocorrem erros em um bloco, ele deve ser retransmitido e isso obviamente reduz a vazão do enlace. Contudo, o uso de ECDs V.42 reduz em muito a necessidade de recuperação de erros pelos programas de comunicação. Isso é, o tratamento de erros de transmissão ocorre no próprio ECD, assim reduzindo o tempo desperdiçado na recuperação de erros pelos programas. Note que ainda podem ocorrer erros de transmissão nas interfaces assíncronas entre ETD e ECD e estes somente são detectáveis por programa.

4.3.6 Compressão de Dados – V.42bis

Devido às limitações de velocidade da interface EIA-232 (*i.e.,* 20 Kbps), uma maneira de aumentar a taxa efetiva de transmissão através de linhas seriais consiste em comprimir os dados a transmitir. Além do aumento da vazão, a compressão de dados reduz o tempo de conexão em linhas discadas e melhora a utilização de linhas alugadas. A transmissão de dados comprimidos pode ser feita de duas maneiras. Na primeira delas, a compressão é feita no ETD pelo usuário. Os arquivos que contém os dados são comprimidos antes da transmissão por programas como `compress` ou `gzip`. O arquivo comprimido é transmitido e posteriormente expandido pelo receptor. A outra possibilidade consiste em dotar o ECD da capacidade para comprimir dados durante a transmissão. O ECD receptor expande os dados enquanto os recebe.

A recomendação CCITT V.42bis define uma técnica de compressão de dados baseada num algoritmo de substituição textual [ZL78]. Os dados a ser transmitidos são agrupados em cadeias de um certo número de octetos. O algoritmo de substituição textual constrói um dicionário de cadeias e ao invés das cadeias, os índices no dicionário é que são transmitidos. Para cadeias de 16 octetos e um dicionário com 2^{11} elementos, no melhor caso, a taxa de compressão seria $(16 * 8)\ /\ 11 = 11.6 : 1$. Para textos em Inglês, V.42bis atinge reduções da ordem de $2.3 : 1$ [Tho92].

Algoritmo de compressão O transmissor e o receptor contróem seus dicionários ao mesmo tempo, na medida em que as cadeias são transmitidas. No exemplo acima, um dicionário com 2^{11} elementos permite a substituição de cadeias de 16 caracteres pelo índice da cadeia no dicionário. Inicialmente, as primeiras 256 posições do dicionário são preenchidas com cadeias de um caracter, na ordem do alfabeto usado (*e.g.* CBII, Tabela 1.1, pág. 5), como mostrado abaixo.

$$
\begin{aligned}
\mathrm{d}[0] &= 0\mathrm{x}00\ (\textsc{nul}) \\
\mathrm{d}[1] &= 0\mathrm{x}01\ (\textsc{soh}) \\
&\cdots \\
\mathrm{d}[0\mathrm{x}41] &= \mathrm{A} \\
\mathrm{d}[0\mathrm{x}42] &= \mathrm{B} \\
&\cdots \\
\mathrm{d}[0\mathrm{x}100] &= \emptyset \\
\mathrm{d}[0\mathrm{x}101] &= \emptyset \\
&\cdots \\
\mathrm{d}[0\mathrm{x}3\mathrm{ff}] &= \emptyset
\end{aligned}
$$

A cada cadeia transmitida, é acrescentada ao dicionário uma nova cadeia, consistindo da cadeia recém-transmitida concatenada com o primeiro caracter ainda não transmitido. Quando todas as cadeias do dicionário contém 16 caracteres, as cadeias usadas no passado mais distante são substituídas por cadeias que ainda não existem no dicionário. Dessa forma, a compressão se adapta ao texto sendo transmitido. Por exemplo, considere uma transmissão que inicia com a palavra "araraquara".

1. o primeiro 'a' é transmitido como `0x061` e a cadeia 'ar' inserida na posição 0x100;
2. o primeiro 'r' é tramsmitido como `0x072` e a cadeia 'ra' inserida na posição 0x101;
3. a cadeia 'ar' é transmitida como `0x100` e 'ara' inserido na posição 0x102;
4. o terceiro 'a' é transmitido como `0x061` e a cadeia 'aq' inserida na posição 0x103;
5. 'q' é transmitido como `0x071` e 'qu' inserido na posição 0x104;
6. 'u' é transmitido como `0x075` e 'ua' inserido na posição 0x105;
7. finalmente, 'ara' é transmitido como `0x102`, ao invés de `0x061 0x072 0x061`.

Obviamente, a taxa de compressão aumenta com o número de cadeias no dicionário que contém um número de caracteres próximo de 16. A transmissão do código fonte de programas via equipamentos V.42+V.42bis atinge taxas de compressão de 2.6 a 3.2 : 1 porque existem muitas repetições de cadeias relativemente simples. O uso de compressão V.42bis para a transmissão de dados previamente comprimidos pode causar *expansão* nos dados, sendo portanto desaconselhada.

Comprimir antes de transmitir Como uma alternativa ao uso de modems V.42bis, o usuário pode comprimir seus arquivos com `compress` ou `gzip` antes de transmiti-los. As vantagens advindas disso são uma melhor taxa de compressão porque os algoritmos implementados por estes programas são mais eficientes que o usado no V.42bis (embora sejam computacionalmente mais caros: estes algoritmos consomem mais memória e tempo de processador que V.42bis). Além disso, o usuário pode escolher o melhor algoritmo de compressão, dependendo do tipo de dados (binários, executáveis, texto, bancos de dados). Outra vantagem decorre de que dados já comprimidos não sofrem com o estrangulamento na interface EIA-232. As desvantagens de se fazer a compressão antes da transmissão são a necessidade de haver o mesmo programa de compressão nas duas máquinas e tanto a compressão pelo remetente quanto a descompressão pelo destinatário consomem tempo de máquina.

4.4 Controlador de Interface Serial Síncrona

O Capítulo 3 discute as técnicas empregadas para a transmissão serial assíncrona. Enquanto a transmissão assíncrona pode ser implementada com circuitos relativamente simples, ela é ineficiente. Para um código com caracteres de 8 bits, os bits de início, final e paridade acabam

por causar a transmissão de 37.5% bits adicionais, que servem apenas para manter transmissor e receptor sincronizados e detectar erros. Para códigos com menos de 8 bits por caracter, o desperdício é ainda maior.

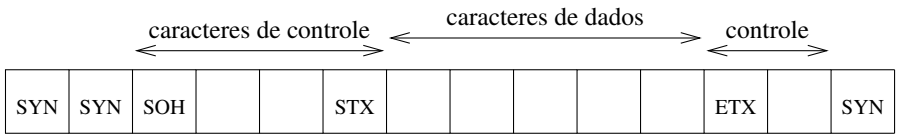
Se, ao invés de transmissor e receptor sincronizarem seus circuitos de emissão e detecção a cada caracter, isso ocorrer entre blocos de caracteres, a eficiência da transmissão pode ser muito aumentada. Por ser mais eficiente, a comunicação serial síncrona tem sido empregada em aplicações mais sofisticadas do que aquelas reservadas para transmissão assíncrona. Para tanto, protocolos de controle de enlace de dados implementam níveis de funcionalidade razoavelmente complexos quando comparados com protocolos de comunicação assíncrona. Compare, por exemplo, Kermit com BSC ou HDLC.

4.4.1 Transmissão Serial Síncrona

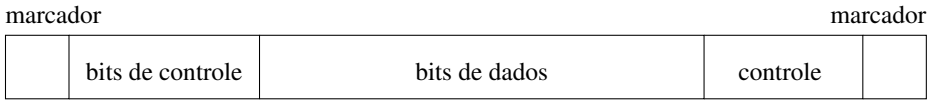
Na Seção 3.2 foi visto que o modo de transmissão assíncrono é ineficiente porque dois bits adicionais são transmitidos para cada caracter, além do bit de paridade, se for o caso. Uma maneira mais eficiente de sincronizar as taxas de amostragem nas duas pontas da linha consiste em enquadrar *grupos* de caracteres, chamados de *quadros*, entre bits de sincronismo. Além disso, a cadência de amostragem que deve ser usada pelo receptor para amostrar o sinal na linha DRX é recuperada do próprio sinal transmitido *i.e.,* a informação de sincronismo deve permear os dados durante a transmissão. As técnicas usadas para tanto são examinadas na Seção 2.4.1. Este modo de transmissão é dito *síncrono* porque os bits transmitidos são amostrados pelo receptor em sincronia com sua emissão pelo transmissor. Durante a transmissão de um quadro, não existe nenhum intervalo arbitrário entre a transmissão de dois caracteres consecutivos, como é o caso na transmissão assíncrona. Ao contrário da transmissão assíncrona, o sincronismo entre transmissor e receptor é mantido *sempre*, mesmo que não hajam dados a transmitir. Portanto, em transmissões síncronas, os circuitos de transmissor e receptor se mantém sincronizados *sempre* e marcadores especiais sinalizam o início e final de cada quadro.

Do ponto de vista do Nível Físico, o receptor faz a amostragem dos bits recebidos da mesma forma que na transmissão serial assíncrona, vista na Seção 3.2. Do ponto de vista do Nível de Enlace, a sincronização entre transmissor e receptor acontece quando o receptor detecta o *prólogo* de um quadro, consistindo de um certo número de bits com uma sequência definida de `0`'s e `1`'s. O final de um quadro é sinalizado pelo *epílogo*. Um quadro também pode ser enquadrado por caracteres reservados para esta finalidade. A Figura 4.26 mostra a posição dos bits de controle e de dados na transmissão síncrona e a delimitação de um quadro pelos caracteres SYN e marcadores.

Os dados transmitidos sincronamente podem ser interpretados de duas maneiras. Na transmissão *baseada em caracter*, os dados e os comandos são codificados em caracteres do alfabeto ASCII, por exemplo. O mesmo ocorre com o prólogo e o epílogo, representados por um ou mais caracteres SYN. Ao detectar o prólogo (caracter de sincronismo), o receptor se prepara para receber um bloco de caracteres que termina no epílogo *i.e.,* um quadro completo. Outra alternativa consiste no envio, logo em seguida ao prólogo, de um campo de controle que indica o tamanho do quadro *i.e.,* o número de caracteres do quadro. Nesse caso, após detectar o SYN, o receptor determina o tamanho do quadro e lê da entrada serial aquele número de caracteres.



(a) Quadro em transmissão síncrona baseada em caracter



(b) Quadro em transmissão síncrona baseada em bit

Figura 4.26: Quadros usados em transmissão síncrona baseadas em caracter e bit.

Na transmissão *baseada em bit*, os campos de dados e controle são interpretados como uma sequência de bits, e não como caracteres. Uma sequência particular com oito bits é usada como *marcador* ("flag") e é usada no prólogo e no epílogo. Geralmente, o marcador é a sequência 01111110. O receptor espera pela detecção do marcador, o que indica o início de um quadro. Após o marcador, seguem campos de controle, um campo de tamanho variável com dados, outros campos de controle e o quadro é terminado por outro marcador. O campo de controle tem tamanho fixo e a interpretação dos bits de controle depende da sua posição com relação aos marcadores de início e final de quadro.

A eficiência da transmissão síncrona é muito maior que a da assíncrona para blocos com muitos caracteres. A transmissão assíncrona com código de 8 bits requer 20% de bits adicionais apenas para sincronizar transmissor e receptor (*i.e.,* os bits de início e final). Na transmissão síncrona, o desperdício é muito menor. O campo de controle do protocolo HDLC (Seção 4.3.3) e os dois marcadores perfazem 48 bits. Para uma mensagem com 1000 bits, os bits adicionais de sincronização e controle perfazem 4.6% dos bits enviados ($48/1048 = 0.046$). No outro extremo, quadros que transportam menos de 48 bits de dados incorrem em desperdício maior que 100%.

4.4.2 Descrição Funcional

Um *Controlador de Interface Serial Síncrona* (CISS) implementa um ou mais protocolos de comunicação síncrona, como BSC ou HDLC, por exemplo. Nos dispositivos mais sofisticados, além das funções de comunicação, um controlador de acesso direto à memória (ADM) pode ser programado para carregar/descarregar os armazenadores de transmissão/recepção com um mínimo de intervenção do processador. Esta seção apresenta um CISS simplificado para ilustrar os elementos mais importantes do funcionamento e programação de dispositivos deste tipo. O protocolo implementado pelo CISS aqui descrito é um subconjunto do HDLC (Seção 4.3.3).

O controlador de interface serial assíncrona (CISA) visto na Seção 3.5 é relativamente simples porque os circuitos do transmissor e do receptor se re-sincronizam a cada caracter transferido. Como consequência da baixa sofisticação dos periféricos, os protocolos de comunicação assíncrona devem ser implementados por programa (*e.g.* Kermit).

Por outro lado, a comunicação síncrona é usada com velocidades de transmissão mais altas, a re-sincronização das entidades de Enlace ocorre a cada quadro transferido. Essas duas características fazem com que os protocolos de comunicação síncrona sejam frequentemente implementados em silício. Em geral, CISSs suportam acesso direto à memória para liberar o processador das funções de cópia de dados entre memória e periférico. O periférico é programado para interromper o processador ao final da transmissão ou recepção de um quadro completo, por exemplo.

A Figura 4.27 mostra o diagrama de blocos de um CISS. A *interface com processador*

consiste dos sinais de controle do barramento para acessos de escrita e leitura aos registradores do periférico e geração e tratamento de interrupções. O *controlador de ADM* contém os sinais necessários para a operação em modo de acesso direto à memória *i.e.,* a tomada de controle do barramento durante os ciclos de transferência de dados entre memória e periférico.

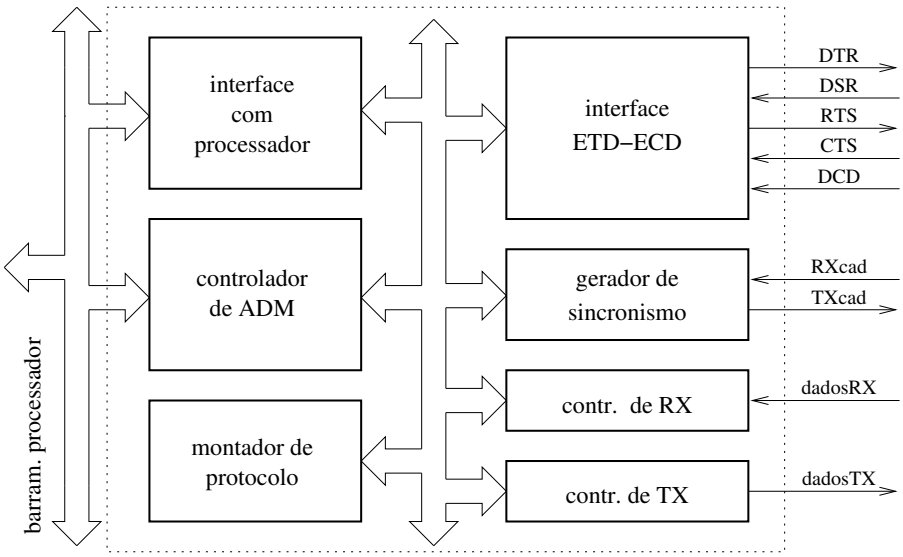


Figura 4.27: Organização interna do CISS.

A *interface ETD–ECD* contém os sinais necessários para o controle do tráfego de dados entre ETD e ECD. Uma vez que a velocidade de operação da interface EIA-232-D é limitada a 20 Kbps, ETDs e ECDs síncronos usam interfaces mais rápidas, como aquelas definidas pelos padrões CCITT V.35 ou EIA-422. A funcionalidade dos sinais de interface DTR, DSR, RTS, CTS e DCD é aquela descrita na Seção 3.4.1. O *gerador de sincronismo* gera a cadência de emissão dos dados a transmitir e usa a cadência de amostragem fornecida pelo ECD para amostrar os dados recebidos, como discutido nas Seções 2.4 e 2.5.

O *montador de protocolo* implementa a máquina de estados do protocolo de enlace e fornece os sinais de controle para os circuitos responsáveis pela transmissão e recepção de quadros. Sob controle do processador, o montador de protocolo gerencia o estabelecimento, manutenção e terminação dos enlaces lógicos, controla o endereçamento e resolve as situações de erro. O *controlador de recepção* detecta o início dos quadros, verifica o endereçamento dos quadros, transfere os octetos recebidos para a interface do processador/ADM e re-calcula o teste de paridade dos quadros recebidos. Se algum erro for detectado, este é sinalizado no registrador de status do periférico. O *controlador de transmissão* gera os octetos necessários para sincronizar o receptor, adiciona os campos de controle e endereço, transmite os dados e calcula e transmite o teste de paridade. Se ocorrerem erros, estes também são sinalizados no registrador de status.

A Figura 4.28 mostra o modelo de programação do periférico. Os registradores do CISS contém 32 bits e são endereçados em palavras consecutivas de 32 bits. O programador configura o modo de operação do periférico ao escrever uma palavra apropriada no registrador de controle `Rcontr`. O estado de operação do periférico é acessível através do registrador de status `Rstat`. Outros parâmetros de operação relacionados ao protocolo HDLC (*e.g.* Modo Balanceado Assíncrono, endereço da estação) devem ser gravados no registrador `Rparam`. Quando da operação com ADM, os registradores `RXender`, `TXender`, `RXtam` e `TXtam` devem ser programados com os endereços e tamanhos máximos dos armazenadores em memória.

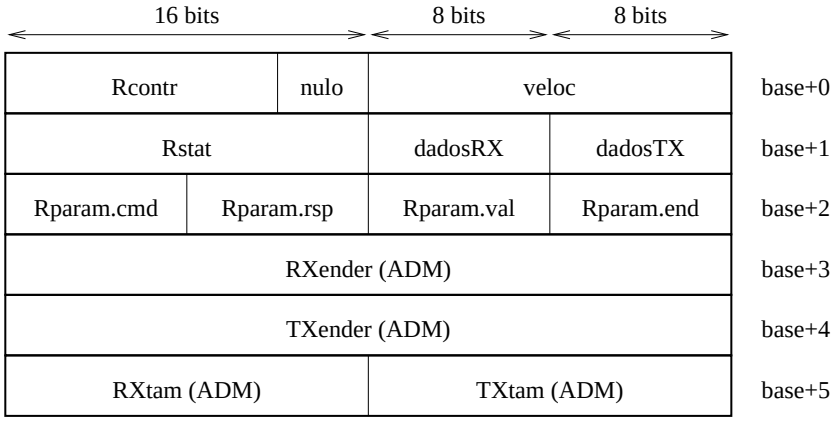


Figura 4.28: Registradores do CISS.

A Tabela 4.5 mostra o significado dos bits do registrador de controle. Quando os bits estão em 1, a função mostrada na tabela é ativada; quando em 0, a função é desativada. Os bits dtr e rts controlam os sinais da interface ETD-ECD de mesmo nome. O bit abort, quando ativado pelo processador, interrompe a transmissão normal do quadro e inicia a transmissão de um comando ABORT (inserção de 8 a 14 bits em 1). O bit secund indica ao periférico que este deve operar como uma estação secundária. Os bits RXint e TXint habilitam a geração de interrupções quando o periférico recebeu um quadro completo e quando completou a transmissão de um quadro completo, respectivamente. O campo nulo não é usado. O campo veloc deve ser programado com a velocidade de operação desejada.

Ao contrário da transmissão assíncrona, não pode haver nenhum intervalo entre a transmissão de dois caracteres consecutivos. Caso o processador ou controlador de ADM não consiga transferir dados entre a memória e o registrador `dadosTX` a uma taxa adequada, um *erro de transmissão* deve ser sinalizado. A taxa de transferência é determinada pela velocidade de transmissão. Quanto maior a velocidade de transmissão, mais rápida deve ser a transferência de dados da memória para o periférico. O bit errTXint habilita a geração de uma interrupção se, durante a transmissão de um quadro, o processador não for capaz de escrever um caracter novo no registrador `dadosTX` a tempo, assim causando a transmissão de um ABORT.

bit	nome	quando em 1
0	dtr	faz DTR ativo (terminal pronto)
1	rts	faz RTS ativo (pedido de transmissão)
2	abort	transmite comando ABORT
3	secund	opera como estação secundária
4	RXint	interrompe na recepção
5	TXint	interrompe na transmissão
6	errTXint	interrompe se transmissão lenta
7-15	nulo	não usado (sempre 0)
16-31	veloc	velocidade de operação

Tabela 4.5: Significado dos bits do registrador de controle Rcontr.

A Tabela 4.6 identifica os bits do registrador de status `Rstat`. Este registrador pode ser dividido em 5 grupos funcionais. O primeiro deles indica o estado dos sinais da interface ETD-ECD. Os bits dsr, cts e dcd indicam o estado dos sinais de mesmo nome. O bit ECDmud indica se houve qualquer mudança no estado dos outros três bits. Este bit permanece ativo mesmo

que um dos sinais tenha sofrido duas alterações de estado (*e.g.* ativ → inatv → ativ). Após um acesso ao registrador de status, o bit ECDmud fica inatv.

O segundo grupo de bits indica o estado do protocolo. O bit abort indica que o receptor detectou um comando ABORT. O bit procSyn indica que o receptor está tentando se sincronizar pelos caracteres recebidos. Quando o periférico se encontra neste estado, o circuito receptor amostra cada bit recebido e compara os últimos 8 bits amostrados com o caracter SYN. Quando um SYN é detectado, o receptor se prepara para receber um quadro e ativar o bit idq (início de quadro) assim que o primeiro caracter diferente de SYN for recebido.

Os bits do terceiro grupo identificam o tipo de interrupção ocorrida. O periférico gera uma interrupção e o processador, ao atendê-la, deve examinar o registrador de status para identificar a causa da interrupção. O bit `RXint` indica a ocorrência de uma interrupção causada pela recepção de um quadro completo. `TXint` indica a ocorrência de uma interrupção causada pelo término da transmissão de um quadro completo. `errTXint` indica interrupção por perda de sincronismo na transmissão (*i.e.,* ADM lento com relação à taxa de transmissão).

O quarto grupo identifica a causa de erros. Da mesma forma que o controlador de ADM pode demorar demais para escrever um novo octeto no registrador `dadosTX`, condição esta indicada pelo bit `errTX`, o controlador de ADM pode causar a perda de um ou mais octetos recebidos por não removê-los do registrador `dadosRX` com frequência suficientemente alta. O CISS transfere os octetos recebidos para o registrador `dadosRX` a medida em que estes são recebidos. Se estes octetos não são copiados para a memória a tempo, o octeto r_{i+1} pode ser colocado pelo receptor "em cima" do octeto r_i. Esta situação é sinalizada pelo bit `errRX`. O bit `errTDP` indica que o TDP calculado é diferente do recebido e portanto o quadro foi recebido em erro e deve ser descartado. O bit `errTemp` indica que o temporizador expirou sem que uma resposta à última transmissão tenha sido recebida.

O quinto grupo de bits contém o bit `RXTXdisp` que indica a disponibilidade de um novo octeto no registrador `dadosRX` e/ou que o último octeto gravado em `dadosTX` já foi transmitido. Este bit permite que o CISS seja usado com "polling". Se o periférico está operando com acesso direto à memória, os octetos recebidos/transmitidos são copiados automaticamente do periférico para a memória e vice-versa.

bit	nome	significado (quando em 1)
0	dsr	sinal DSR está ativo (ECD operacional)
1	cts	sinal CTS está ativo (pronto para transmitir)
2	dcd	sinal DCD está ativo (portadora detectada)
3	ECDmud	um dos sinais da interface com ECD mudou
4	fdq	detectou final de quadro
5	idq	detectou início de quadro
6	abort	detectou comando ABORT
7	procSyn	receptor está procurando SYN
8	RXint	ocorreu interrupção de recepção
9	TXint	ocorreu interrupção de transmissão
10	errTXint	ocorreu interrupção de transmissão lenta
11	errRX	erro: falta de sincronia na recepção
12	errTX	erro: falta de sincronia na transmissão (lenta)
13	errTDP	erro: paridade na recepção
14	errTemp	erro: temporizador expirou
15	RXTXdisp	espaço/octeto disponível em `dadosTX`/`dadosRX`
16-23	dadosRX	último octeto recebido
24-31	dadosTX	próximo octeto a transmitir

Tabela 4.6: Significado dos bits do registrador de status Rstat.

A Tabela 4.7 define o significado dos campos do registrador `Rparam`. O modo de operação do montador de protocolo é definido pelos valores gravados neste registrador. O campo cmd permite a programação do CISS para a operação de acordo com os modos definidos no protocolo HDLC, Tabelas 4.3, pág. 107, e 4.4, pág. 108 (*e.g.* Modo Normal de Resposta snrm, Modo Balanceado Assíncrono sabm ou Modo de Resposta Assíncrona sarm). As respostas aos comandos são mostradas pelo montador de protocolo no campo rsp. Se algum outro caracter de sincronismo que não o SYN for usado, este pode ser definido no campo carSyn. O endereço da estação no enlace deve ser gravado no campo endEst.

bits	campo	significado
0-7	cmd	comando para montador de protocolo
8-15	rsp	resposta do montador de protocolo
16-23	val	valor de parâmetro especificado em cmd
24-31	endEst	endereço da estação

Tabela 4.7: Significado dos campos do registrador de parâmetros Rparam.

Modos de Operação – E/S

Existem várias possibilidades para o acoplamento entre a funcionalidade oferecida pelo periférico e sua utilização nos programas de comunicação. Estas variam desde "polling" até acesso direto à memória. Os parágrafos que seguem discutem estas possibilidades. A Seção 4.4.3 contém um exemplo simplificado de código para uso com acesso direto à memória.

Consulta O controlador de interface serial síncrona descrito acima pode ser usado de três maneiras. O modo mais simples é a *consulta* ("polling") onde o CISS opera com as interrupções desabilitadas e o processador deve transferir os octetos individuais entre memória e periférico. Após provocar alguma alteração no estado de funcionamento do CISS, *e.g.* gravar um octeto no registrador `dadosTX`, o processador fica testando continuamente o estado do periférico (bit TXvazio) até que seja possível gravar novo octeto em `dadosTX`. Como discutido na Seção 3.5, este modo de operação é ineficiente. Além disso, o desempenho do processador impõe limites na taxa máxima de transmissão que pode ser atingida. O Exercício 4.3 explora a relação entre velocidade de processamento e taxa de transmissão.

Interrupções Um CISS também poderia ser implementado de forma a gerar uma *interrupção* a cada octeto transmitido ou recebido[2]. Ao atender uma interrupção, o processador transfere um novo octeto do/para o CISS. Este modo de operação é mais eficiente que consulta porque o processador pode executar outras tarefas entre o envio/recepção de dois octetos. A taxa máxima de transmissão ainda é limitada pelo desempenho do processador, como demonstra o Exercício 4.17. Um modo híbrido de utilização consiste em programar o CISS para gerar uma interrupção ao detectar o início de um quadro. Durante a recepção dos octetos do quadro, o processador faz consultas para transferir cada octeto do quadro.

Acesso direto à memória (ADM) Este é o modo de operação mais eficiente, do ponto de vista de utilização dos recursos do sistema. Ao transmitir um quadro, o processador escreve em um registrador do periférico o endereço em memória do início do bloco de dados e seu tamanho.

[2]Note que o CISS descrito aqui não permite este modo de operação.

O CISS monta e transmite o cabeçalho do quadro e então transfere os octetos de dados do armazenador em memória para o registrador de transmissão, sem qualquer interferência do processador. Ao esperar por um quadro, o processador escreve num registrador do periférico o endereço de um armazenador em memória onde os octetos recebidos devem ser depositados pelo CISS. Após transmitir ou receber um quadro completo, o CISS interrompe o processador para notificá-lo de que a operação está terminada ou da ocorrência de erros. Com ADM, o processador somente intervém para disparar uma transmissão ou para processar um quadro recebido.

4.4.3 Programação

Esta seção contém um exemplo simplificado do código necessário para a operação do CISS com ADM em um ambiente similar ao do sistema operacional Unix [Bac86]. A fim de limitar o tamanho dos segmentos de código as ações relacionadas ao tratamento de erros não são mostradas, embora aquelas sejam imprescindíveis. Em geral, o tratamento de erros é difícil de implementar e faz o código pouco compreensível.

Definições

Para tornar o código portável entre compiladores e/ou arquiteturas que definem o tipo `int` com tamanhos variados, as linhas abaixo definem inteiros com os tamanhos especificados. As versões com sinal destes inteiros também são úteis.

```
typedef int            int32;      /* inteiro de 32 bits, com sinal */
typedef unsigned int   uint32;     /* inteiro sem sinal de 32 bits */
typedef unsigned short uint16;     /* inteiro sem sinal de 16 bits */
typedef unsigned char  uint8;      /* inteiro sem sinal de 8 bits */
```

O tipo `CISScontrRegTipo` define as posições e os significados dos bits do registrador de controle. Note que há uma exata correspondência entre a Tabela 4.5 (pág. 118) e `CISScontrRegTipo`. O campo `nulo` é necessário para alinhar os bits definidos na estrutura com suas posições certas no registrador `Rcontr`.

```
typedef struct CISScontrRegBits {  /* bits individuais */
  uint32  dtr:1,                   /* ativa DTR (ETD operacional) */
          rts:1,                   /* ativa RTS (pede para transmitir) */
          abort:1,                 /* transmite comando ABORT */
          secund:1,                /* opera como secundária */
          RXint:1,                 /* habil. interr. de recepção */
          TXint:1,                 /* habil. inter. de transmissão */
          errTXint:1,              /* habil. interr. transm. muito lento */
          nulo:8,                  /* alinhamento com registrador */
          veloc:16;                /* velocidade de operação */
} CISScontrRegTipo;
```

O registrador de status `Rstat` é representado de três maneiras diferentes. Na primeira das representações, a cada um dos bits do registrador corresponde um campo da estrutura `CISSstatRegTipoBits`. Observe a correspondência entre os campos desta estrutura e os elementos da Tabela 4.6 (pág. 120). A estrutura `CISSstatRegTipoGrupos` contém uma forma resumida da Tabela 4.6 onde, cada campo da estrutura corresponde a um grupo de bits do registrador de status `Rstat`. O agrupamento por função facilita o exame do estado do periférico porque várias combinações de estado/erros podem ser examinadas com um único teste.

```
typedef struct CISSstatRegBits {   /* bits individuais */
  uint32  dsr:1,                   /* DSR ativo (ECD operacional) */
```

```
          cts:1,                  /* CTS ativo (pronto para transmitir) */
          dcd:1,                  /* DCD ativo (portadora detectada) */
          ECDmud:1,               /* algum sinal da interface mudou */
          fdq:1,                  /* fim de quadro */
          idq:1,                  /* início de quadro */
          abort:1,                /* comando ABORT recebido */
          procSyn:1,              /* procurando SYN */
          RXint:1,                /* interrupção de recepção */
          TXint:1,                /* interrupção de transmissão */
          errTXint:1,             /* transmissor muito lento */
          errRX:1,                /* erro: falta de sincronia de RX */
          errTX:1,                /* erro: falta de sincronia de TX */
          errTDP:1,               /* erro: paridade na recepção */
          errTemp:1,              /* erro: temporizador expirou */
          RXTXdisp:1,             /* octeto em dadosRX ou espaço em dadosTX */
          dadosRX:8,              /* octeto recebido */
          dadosTX:8;              /* octeto por transmitir */
} CISSstatRegTipoBits;

typedef struct CISSstatRegGrupos { /* agrupados por função */
  uint32  ECDstat:4,              /* estado do ECD */
          proto:4,                /* protocolo */
          interr:3,               /* interrupções */
          erros:4,                /* erros */
          rxtx:1,                 /* estado da recepção/transmissão */
          dadosRX:8,              /* dados recebidos */
          dadosTX:8;              /* dados a transmitir */
} CISSstatRegTipoGrupos;
```

A estrutura `CISSstatRegTipo` contém três representações para o registrador de status, *i.e.,* ela é a união de um inteiro de 32 bits com os tipos `CISSstatRegTipoBits` e `CISSstatRegTipoGrupos`. Dessa forma, o estado do periférico pode ser examinado bit a bit quando se necessita de informação detalhada sobre o resultado de uma certa operação. Quando se deseja verificar se alguma coisa mudou, o teste de um grupo de bits é o suficiente. Finalmente, pode ser necessário copiar todo o estado do periférico para a memória, para diagnóstico posterior, por exemplo.

```
typedef uint32 CISSstatRegTipoPal; /* 32 bits sem estrutura */

typedef union CISSstatReg {
  CISSstatRegTipoPal     pal;     /* palavra sem estrutura */
  CISSstatRegTipoBits    bit;     /* bits individuais */
  CISSstatRegTipoGrupos  grp;     /* agrupados por função */
} CISSstatRegTipo;
```

O tipo `CISSparamRegTipo` define os campos do registrador `Rparam` de acordo com a Tabela 4.7 (pág. 120). Os campos `cmd` e `rsp` permitem a escolha do modo de funcionamento do protocolo HDLC. O campo `val` permite a escolha de parâmetros de operação do CISS (*e.g.* caracter diferente de SYN para a sincronização entre transmissores e receptores num enlace, valor do temporizador). O endereço da estação no enlace deve ser gravado no campo `endEst`.

```
typedef struct CISSparamRegBits {
  uint32  cmd:8;                  /* comandos do protocolo HDLC */
          rsp:8,                  /* resposta do montador de protocolo */
          val:8,                  /* valor do parâmetro definido em cmd */
          endEst:8;               /* endereço da estação */
} CISSparamRegTipo;
```

A estrutura `CISStipo` é a representação na linguagem C dos registradores mostrados na Figura 4.28 (pág. 118). O primeiro elemento da estrutura (`CISScontrTipo`) é associado ao endereço base do CISS e corresponde ao registrador de controle (`Rcontr`). O segundo elemento corresponde ao registrador `Rstat` e é mapeado no endereço base+1 (uma palavra de 32 bits). O terceiro elemento da estrutura corresponde ao registrador de parâmetros de operação `Rparam`, mapeado no endereço base+2. Os dois elementos seguintes devem ser preenchidos com os endereços de armazenadores em memória de/para onde são feitas as transferências com acesso direto à memória. O registrador `RXtam` deve ser preenchido com o tamanho máximo do armazenador em memória onde mensagens recebidas devem ser armazenadas. Se uma mensagem excede o tamanho do armazenador, um erro de recepção é sinalizado. O registrador `TXtam` deve ser preenchido com o tamanho da próxima mensagem a ser transmitida com o auxílio do controlador de ADM.

```
typedef struct CISS {
  CISScontrTipo       rContr;      /* registrador de controle */
  CISSstatRegTipo     rStat;       /* registrador de status */
  CISSparamTipo       rParam;      /* parâmetros de operação */
  uint8               *rxEnder;    /* ADM: armazenador de recepção */
  uint8               *txEnder;    /* ADM: armazenador de transmissão */
  uint16              rxTam;       /* ADM: núm. de octetos recebidos */
  uint16              txTam;       /* ADM: núm. de octetos a transmitir */
} CISStipo;
```

Os armazenadores para mensagens recebidas e a transmitir são descritos pela estrutura `ArmazTipo`. Cada armazenador contém, além dos octetos da mensagem, uma cópia do registrador de status no momento em que a interrupção associada à mensagem foi atendida e o endereço da estação que enviou a mensagem ou da estação que vai receber a mensagem. O campo `emUso` indica se o armazenador contém um quadro que ainda não foi completamente transmitido, ou contém um quadro apenas parcialmente cheio.

```
typedef struct Armaz {
  uint8               emUso;       /* armazenador em uso/cheio */
  CISSstatRegTipoPal  stat;        /* cópia do registrador Rstat */
  uint8               quem;        /* destinatário/remetente da mensagem */
  uint16              tam;         /* tamanho da mensagem recebida */
  uint8               rsp;         /* resposta a comando HDLC */
  uint8               *msgm;       /* dados de usuário */
} ArmazTipo;
```

Declarações

Os armazenadores de transmissão e recepçãoe a estrutura que descreve o CISS devem ser alocados em memória. O endereço físico do CISS deve corresponder ao endereço da estrutura que o descreve. O código que segue parte da premissa de que endereços de periféricos são mapeados no mesmo espaço de endereçamento que memória (*i.e.,* E/S mapeado em memória).

```
/* variáveis globais */
ArmazTipo recArm;                    /* armazenador de recepção */
ArmazTipo envArm;                    /* armazenador de transmissão */
CISSstatRegTipo status;              /* cópia do registrador de status */

CISStipo ciss = CISSender;           /* endereço do periférico */
```

Inicialização

Os armazenadores e o registrador de controle do CISS são inicializados com os valores apropriados. O CISS vai operar em modo ADM, e inicialmente, com as interrupções desabilitadas. O sinal DTR, da interface com o ECD é ativado, informando ao ECD que o ETD está pronto para operar. A velocidade de operação é programada em 64 Kbps.

```
recArm = NULL;                          /* prepara armazenadores */
envArm = NULL;
envArm.quem = CISSmeuEnder;

ciss.rContr.dtr = ATIVO;                /* sinaliza ETD pronto */
ciss.rContr.RXint = INATIVO;            /* desabilita interrupções */
ciss.rContr.TXint = INATIVO;
ciss.rContr.errTXint = INATIVO;
ciss.rContr.adm = ATIVO;                /* opera em modo ADM */
ciss.rContr.veloc = CISSvel64K;
ciss.rParam.endEst = CISSmeuEnder;      /* endereço da estação */
```

O enlace deve funcionar em Modo Assíncrono Balanceado, entre duas estações combinadas. A Figura 4.24 (pág. 111) mostra um exemplo de enlace em Modo Assíncrono Balanceado. O intervalo de espera por respostas é estabelecido e o CISS é comandado a iniciar a fase de estabelecimento de enlace com o comando `CISSsabm`. O laço de espera testa continuamente o registrador de status até que a resposta ao comando sabm seja recebida ou que o temporizador expire.

```
ciss.rParam.cmd = CISSespera;           /* programa temporizador */
ciss.rParam.val = CISSintervalo;
ciss.rParam.cmd = CISSsabm;             /* Modo Assíncrono Balanceado */

do {                                    /* recebeu resposta ou esgotou tempo */
   status.pal = ciss.rStat.pal;
} until (status.bit.fdq || status.bit.errTemp);

if (ciss.rParam.rsp != CISSsabmACEITO)
   panico("nao estabeleceu enlace");

ciss.rContr.RXint = ATIVO;              /* habilita interrupções */
ciss.rContr.TXint = ATIVO;
ciss.rContr.errTXint = ATIVO;
```

Ao invés de ficar num laço de espera enquanto aguarda o estabelecimento do enlace, o programa poderia habilitar a interrupção de recepção e suspender sua própria execução. Quando o quadro com a resposta ao comando sabm for recebido, o registrador de status é então testado.

Transmissão

Uma vez estabelecido o enlace, o código contido na função `enviaMsgmCISS()` define as ações necessárias para o envio de uma mensagem. Note que os dados de usuário não podem ser alterados até que a transmissão complete. As funções `habInterr()` e `desInterr()` habilitam e desabilitam as interrupções, respectivamente.

```
void enviaMsgmCISS(uint16 tamMsgm, uint8 *msgm) {

   desInterr(CISS);                     /* desabilita interrupções */
```

```
   envArm.msgm = msgm;
   envArm.tam = tamMsgm;
   habInterr(CISS);                    /* habilita interrupções */
```

A função informa então ao ECD que as transmissões vão se iniciar e suspende sua própria execução enquanto espera que o CISS envie a mensagem. Após transmitir todos os octetos da mensagem, o CISS vai interromper o processador. O tratador de interrupções do CISS vai então reativar a execução desta função. Note que se o enlace é estabelecido em linha privativa, não é necessário esperar pelo sinal CTS do ECD, uma vez que a conexão é permanente.

```
   if (! ciss.rStat.dsr)               /* ECD deve estar pronto */
      panico("ECD inativo");

   ciss.txEnder = envArm→msgm;         /* canal ADM de transmissão */
   ciss.txTam = envArm→tam;

   ciss.rContr.rts = ATIVO;            /* avisa ECD da transmissão */
   ciss.rContr.idq = ATIVO;            /* CISS inicia a transmissão */

   suspende(CISSesperaTX);             /* espera transmissão completar */

   if (envArm→stat.grp.erros)          /* erros ?  */
      trataErrosTransmissao();
   return();
}
```

A função `suspende()` provoca uma troca de contexto. O processo que a invoca fica suspenso aguardando pela ocorrência do evento indicado no parâmetro, neste caso, uma interrupção de transmissão pelo CISS. Quando ocorrer o evento esperado pelo processo, o tratador da interrupção libera o processo para execução pelo processador, como mostrado abaixo.

Recepção

A função `recebeMsgmCISS()` implementa *recepção síncrona*: o processo que deseja receber fica bloqueado até que alguma mensagem lhe seja enviada. Com interrupções desabilitadas, a função programa o controlador de ADM com o endereço do armazenador que vai ser preenchido quando da recepção do quadro. Os octetos da mensagem são copiados para o armazenador do usuário pelo controlador de ADM.

```
uint16 recebeMsgmCISS(uint8 *msgm) {

   desInterr(CISS);                    /* desabilita interrupções */
   ciss.rxEnder = recArm.msgm = msgm;
   habInterr(CISS);                    /* habilita interrupções */

   suspende(CISSesperaRX);             /* espera por mensagem */

   if (recArm.stat.grp.erros)          /* erros ?  */
      trataErrosRecepcao();

   return(recArm.tam);                 /* tamanho da msgm recebida */
}
```

Tratador de Interrupções

O tratador de interrupções examina o registrador de status para determinar a causa da interrupção. Se um quadro foi *recebido*, o conteúdo do registrador de status é copiado para o armazenador, bem como o endereço do remetente, tamanho da mensagem e resposta da máquina de estados do protocolo. O processo destinatário é então marcado como pronto para continuar executando. Se a *transmissão* de um quadro completou, o conteúdo do registrador de status e a resposta da máquina de estados do protocolo são copiados para o armazenador. O processo remetente é então marcado como pronto para continuar executando.

```
void cissInterr(void) {
   CISSstatRegTipo stat;

   desInterr(CISS);
   stat.pal = ciss.rStat.pal;

   if (stat.bit.RXint) {              /* recepção completou ?  */
      recArm.stat.pal = stat.pal;     /* retorna status da recepccão */
      recArm.tam = ciss.rxTam;        /* tamanho da msgm recebida */
      recArm.quem = ciss.endEst;      /* remetente */
      recArm.rsp = ciss.rsp;          /* resposta do protocolo HDLC */
      reativa(CISSesperaRX);          /* libera processo receptor */

   } else if (stat.bit.TXint) {       /* transmissão completou ?  */
      envArm.stat.pal = stat.pal;     /* retorna status da transmissão */
      envArm.rsp = ciss.rsp;          /* resposta a comando HDLC */
      reativa(CISSesperaTX);          /* reativa processo transmissor */

   } else if (stat.bit.errTXint) {
      panico("erro de velocidade de transmissão");
   }
   habInterr(CISS);
   return();
}
```

O código mostrado acima, embora simples e eficiente, apresenta alguns problemas. Os armazenadores que contém as mensagens a transmitir não podem ser alterados até que a transmissão complete. Há pouco desacoplamento entre as velocidades do processador e da comunicação através do CISS. Se o processador estiver indisponível para desocupar o armazenador de recepção, o quadro mais antigo pode ser perdido. Uma solução para isso seria o uso de um armazenador circular capaz de conter oito quadros, por exemplo.

4.5 Bibliografia e Exercícios

Discussões sobre protocolos de controle de enlace podem ser encontradas em [PD96, Sta91]. Lam [Lam84a] compara os protocolos BSC e SDLC. Bertsekas [BG92] contém uma excelente e detalhada discussão sobre protocolos de controle de enlace, correção dos protocolos e inicialização e terminação de enlaces. Veja [Com95, PD96] sobre os protocolos da família TCP/IP. Veja [Hel92] para mais detalhes sobre compressão de dados. Os prós e contras do protocolo de compressão de dados V.42bis são discutidos em [Tho92].

Tanembaum [Tan89] contém exemplos codificados em Pascal de algumas das técnicas de controle de fluxo descritos neste capítulo. A implementação de um núcleo de sistema operacional multitarefa semelhante ao citado na Seção 4.4.3 pode ser encontrada em [Hex88]. Mais detalhes sobre programação de controladores de periféricos podem ser encontrados em [AT93].

Nas especificações dos fabricantes, o CISS pode ser descrito pelo nome "Universal Synchronous/Asynchronous Receiver-Transmitter" (USART). Como exemplo, veja o Intel 8273 [Int85].

Exercícios

Ex. 4.1 [§ 4.1] Em sistemas de comunicação de dados, pulsos de ruído duram alguns milissegundos e portanto tendem a distorcer sequências com muitos bits. Na detecção de erros com teste de paridade, o uso conjunto de TPV e TPL não garante a deteção de todos os erros que podem ocorrer. Identifique algumas das sequências de erros em um bloco de caracteres que não seriam detectadas por TPV, por TPL e nem por TPV+TPL.

Ex. 4.2 [§ 4.1] Calcule o TDP, pela divisão de polinômios e com o circuito da Figura 4.2, para a mensagem `1 0001 0001 0001 0001`. O bit menos significativo é o mais à direita.

Ex. 4.3 [§ 4.2.3] A *eficiência* de um protocolo pode ser definida como *o número de bits de usuário transferidos dividido pelo tempo necessário para transferir estes bits*, em uma transação típica. Derive uma expressão que indique de forma aproximada a eficiência de cada um dos protocolos PAR descritos na Seção 4.2.3 que inclua a influência da taxa de erros na eficiência de cada tipo de protocolo.

Ex. 4.4 [§ 4.2.3] No controle de fluxo por janela deslizante, o número máximo de mensagens na janela é dado por $(N-1)$. Isso evita um problema que pode ocorrer se as janelas tiverem tamanho N e mensagens de aceitação são perdidas. Descreva um cenário onde isso possa ocorrer.

Ex. 4.5 [§ 4.2.3] Qual a influência do tempo de propagação na eficiência de um enlace baseado em controle de fluxo do tipo (a) para-e-espera; (b) janela deslizante?

Ex. 4.6 [§ 4.2.3] Qual a influência do tempo de propagação na eficiência da recuperação de erros em um enlace baseado em controle de fluxo do tipo (a) para-e-espera; (b) janela deslizante com volta-N; (c) janela deslizante com repetição seletiva? Pista: em média, que fração de uma janela completa é re-transmitida em caso de erro?

Ex. 4.7 [§ 4.2.3] Desenhe dois diagramas similares à Figura 4.10 (pág. 91) com uma simulação da seqüência de eventos mostrada na Figura 4.29 (pág. 129) usando janela de transmissão de tamanho 4 (somente 3 quadros em viagem a cada instante). Num dos diagramas, use somente recuperação de erros com volta-N; no outro, use somente repetição seletiva. Note que no diagrama da Figura 4.29 as janelas tem tamanho 8. A seqüência de envio será diferente para janelas com tamanho 4.

Ex. 4.8 [§ 4.2.3] O tempo de propagação de ondas eletromagnéticas em meios guiados é da ordem de 20cm por nanosegundo (5ns/metro). Qual a influência do tempo de propagação p em um enlace urbano? E em um enlace entre Porto Alegre e Manaus (supondo que haja uma ligação direta de 5000Km entre as duas cidades)? Considere mensagens de 1000 bits. (a) Como o tempo de propagação influencia a escolha do tamanho de janela? (b) O *tamanho* do campo dos números de seqüência dos quadros limita a vazão máxima através do enlace. Por que? (c) Qual o mínimo tamanho da janela de transmissão para um enlace que transmite a 1 Mbps (microondas)? (d) E para um enlace que transmite a 1 Gbps (fibra óptica)?

Ex. 4.9 [§ 4.2.3] Usando a Equação 4.3 (pág. 94), faça um gráfico que relacione a eficiência (eixo y), o tamanho do quadro (eixo x) e a velocidade de transmissão [Mon90]. Desenhe um gráfico com quatro curvas para as velocidades de 2400, 4800, 9600 e 19200 bps. O tamanho dos blocos deve variar logaritmicamente de 10^2 a 10^5 caracteres.

Ex. 4.10 [§ 4.2.3] A Equação 4.3 (pág. 94) não considera a retransmissão de quadros rejeitados. Supondo uma taxa de erro do enlace de E bits em erro para cada 10^3 bits transmitidos, calcule o número médio de quadros rejeitados $N_r(E, Q)$ para quadros de tamanho Q e derive uma expressão para a eficiência que leve em conta a retransmissão de quadros. Repita o exercício anterior mas variando a taxa de retransmissão de quadros N_r. Use velocidade de transmissão $V = 9600$ bps.

Ex. 4.11 [§ 4.3.2] Faça um diagrama da máquina de estados do mecanismo de aceitação para mensagens com números de sequência pares e ímpares (ACK0 e ACK1) do protocolo BSC.

Ex. 4.12 [§ 4.3.1, 4.3.2, 4.3.3] Compare a eficiência, definida como o *número de bits de dados dividido pelo tamanho total do quadro* dos protocolos Kermit, BSC e HDLC, para o envio de 1, 10, 100 e 1000 caracteres.

Suponha que não haja limite de tamanho de quadro para o Kermit. Considere os caracteres de sincronismo quando necessários. Suponha 8 bits por caracter.

Ex. 4.13 [§ 4.3.1, 4.3.2, 4.3.3] Compare o número de quadros necessários para transferir um arquivo de texto contendo 2^{20} caracteres entre duas máquinas ligadas por uma linha duplex, usando os protocolos Kermit, BSC e HDLC. Use quadros com 90 caracteres e 512 caracteres. Suponha 8 bits por caracter e desconsidere erros de qualquer natureza.

Ex. 4.14 [§ 4.3.2] Considere a transmissão com o protocolo BSC em modo transparente de quadros com caracteres de 8 bits. Suponha que a probabilidade de ocorrência é a mesma para todos os caracteres. Qual é o número de DLEs inseridos para mascarar os caracteres de controle em quadros com 100 e com 1000 caracteres? Qual a perda de eficiência (Exercício 4.12)?

Ex. 4.15 [§ 4.3.3] Repita o exercício anterior com o protocolo HDLC, usando inserção de bits. Qual é o número de bits inseridos? E a perda de eficiência?

Ex. 4.16 [§ 4.3.3] Suponha um enlace com 1000 Km de comprimento e tempo de propagação de sinais de 1m em 5ns. Usando quadros HDLC de 1000 bits, calcule a eficiência da transmissão para janelas com capacidade para 7 e 127 quadros, e taxas de transmissão de 1, 10 e 100 Mbps.

Ex. 4.17 [§ 4.4] Repita os Exercícios 3.8, 3.9 e 3.10 no contexto do CISS descrito na Seção 4.4.

Proj. 4.1 [§ 4.3.6] Escreva um programa que implementa o algoritmo de compressão da Seção 4.3.6. Usando seu programa, faça medidas das taxas de compressão obtidas com textos em Português (CBII), código fonte C e código binário (executável) de um programa. Thomborson [Tho92] discute a implementação do algoritmo do padrão V.42bis.

Proj. 4.2 [§ 4.3.6] Modifique o programa do Projeto 4.1 para compressão usando dicionários de 2^{10} e 2^{12} elementos. O que acontece com as taxas de compressão? E com o tempo de processamento?

Proj. 4.3 [§ 4.4] Considere a transmissão de um quadro usando "polling". Escreva um fragmento de código com dois laços onde, no laço interno, o processador espera pela mudança de estado do CISS após a gravação de um octeto no registrador de transmissão. No laço externo, o processador transfere um bloco de octetos de um armazenador em memória para o CISS. Use a técnica do Exercício 3.9 (pág. 78) para determinar o número de instruções executadas por octeto transmitido. Supondo que o processador executa uma instrução a cada 10ns, qual a taxa máxima de transmissão?

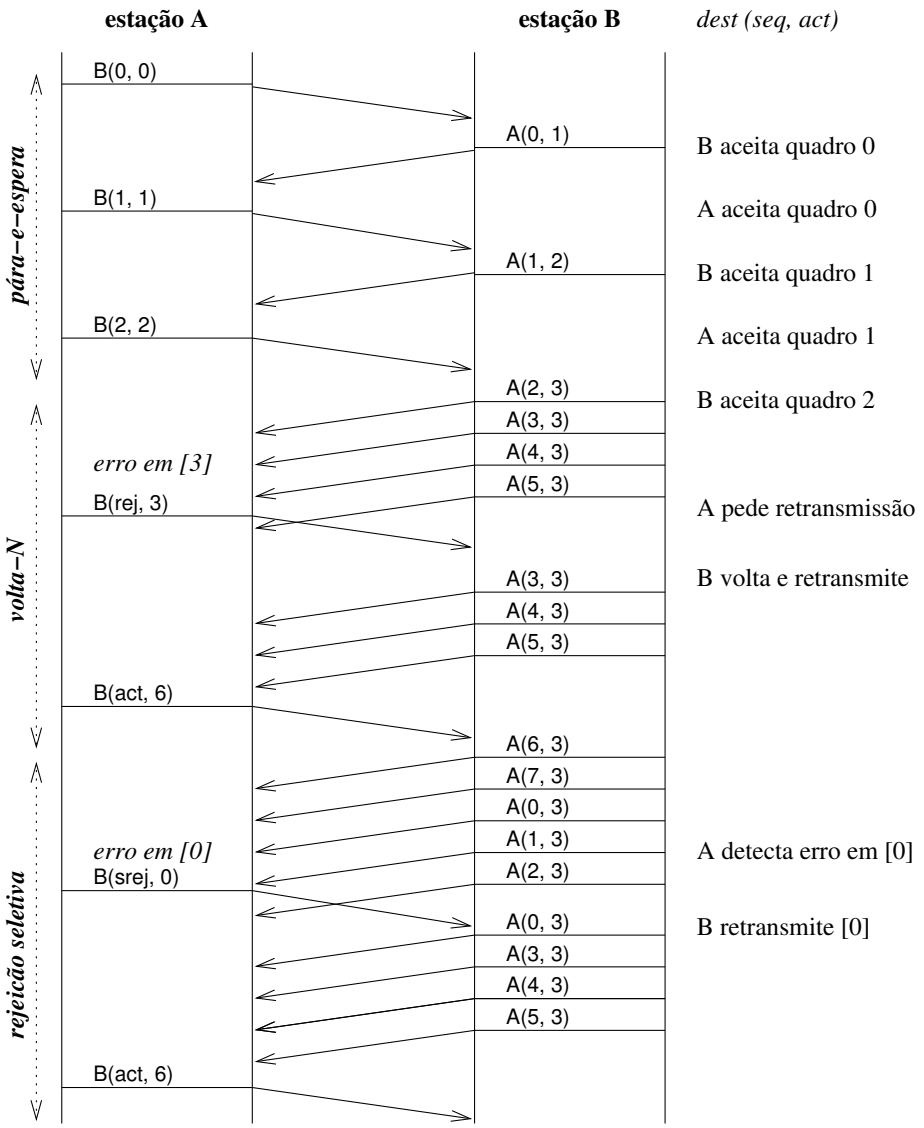


Figura 4.29: Exemplos de recuperação de erros com pedidos automáticos de repetição.

Capítulo 5

Redes Locais

Os capítulos anteriores discutem tópicos que são geralmente agrupados sob o título de *comunicação de dados*. Um *enlace de dados* consiste de duas ou mais estações interligadas por um meio de transmissão e operando segundo as regras definidas no protocolo HDLC em enlaces ponto-a-ponto ou multiponto. Enlaces de dados são suficientes para aplicações onde os volumes de dados sejam pequenos ou as transações são relativamente simples. Em ambientes onde os volumes de dados a transferir são grandes, as transações variadas e complexas e o número de estações é maior que uma dezena, uma outra forma de conexão deve ser usada.

Ao contrário dos enlaces multiponto descritos no Capítulo 4, as estações numa rede local interagem segundo um modelo *cliente-servidor*, ao invés de mestre-escravo. Num ambiente cliente-servidor, uma estação do tipo *servidor* oferece alguns serviços às demais estações na rede. Os *clientes* usam os serviços providos pelo servidor. Uma mesma estação pode ser o servidor de alguns, bem como cliente de outros, serviços. Do ponto de vista da implementação, um servidor é um processo que começa a executar quando a máquina é inicializada e que permanece num laço, esperando por solicitações e atendendo-as. Um processo cliente é iniciado pelo usuário, faz uma ou mais solicitações aos servidores na rede e então termina. Geralmente, processos servidores são parte do sistema operacional enquanto que processos clientes são parte de programas de usuário.

As *redes locais* foram desenvolvidas para permitir a comunicação do tipo um-para-muitos e um-para-todos entre estações de igual capacidade de processamento. Redes baseadas em enlaces ponto-a-ponto não suportam a difusão de mensagens eficientemente. Este tipo de mensagem é freqüente em ambientes cliente-servidor o que implica no uso de outro tipo de topologia. Geralmente, as redes locais empregam topologias que permitem o compartilhamento do meio físico por todas as estações. Isso traz dois benefícios importantes. O primeiro é o suporte eficiente de difusão propiciado pelo compartilhamento do meio. O segundo benefício é associado à possibilidade de interligação de um grande número de estações através da rede.

Existe uma grande variedade de arquiteturas de redes locais propostas e implementadas. Tipicamente, os meios físicos empregados em redes locais tem capacidades maiores que 10^6 bps e se extendem por distâncias da ordem de 10^2 a 10^4 metros. As taxas de erros são muito menores em redes locais do que em linhas telefônicas, com valores melhores que um erro em cada 10^8 bits transmitidos.

As redes de computadores são classificadas em função de sua abrangência geográfica. Uma *rede dispersa* ["wide area network" (WAN)] abrange um país ou um continente. Uma *rede metropolitana* ["metropolitan area network" (MAN)] abrange uma área de vários quilômetros de diâmetro. Uma *rede local* ["local area network" (LAN)] se limita a um prédio, fábrica ou campus. Finalmente, uma *rede confinada* ["desk area network" (DAN)] interliga equipamentos dentro de uma única sala ou componentes dentro de um equipamento. Os protocolos discutidos no Capítulo 4 tem uso na interligação de equipamentos em redes dispersas e metropolitanas.

O conteúdo deste capítulo se concentra em redes metropolitanas e locais.

Este capítulo é dividido como segue. A Seção 5.1 descreve alguns dos meios físicos comumente empregados em redes locais. A Seção 5.2 descreve alguns dos métodos de controle de acesso e de controle de enlace lógico. Para simplificar a discussão, a Seção 5.2 discute apenas o protocolo de controle de enlace definido pelos padrões IEEE 802.2 e ISO/IEC 8802.2 [IEE85b, ISO94]. A Seção 5.3 apresenta a família de protocolos padronizados originalmente pelo IEEE. Dois padrões de rede local são descritos em mais detalhe. A Seção 5.4 descreve a Ethernet, a mais difundida das tecnologias de redes locais, detentora de cerca de 80% do mercado mundial. A Seção 5.5 descreve a Fiber Distributed Data Interface, uma rede de alta velocidade projetada para interligar várias redes locais de baixa velocidade. A Seção 5.6 descreve o padrão Scalable Coherent Interface, que é uma rede confinada de alíssima velocidade, capaz de interligar até 65000 estações, com enlaces entre pares de estações operando a uma taxa de 1 Gigabyte por segundo.

5.1 Nível Físico

Como nos enlaces ponto-a-ponto, cabe ao Nível Físico da rede local implementar um infoduto que transporta os quadros inseridos pelas estações até seus destinatários. As formas de sinalização dependem dos meios físicos empregados e não diferem significativamente daquelas descritas nas Seções 2.4. A topologia da rede, *i.e.,* a forma de interligação entre es várias estações da rede, é determinada em grande parte pelas características do meio físico.

5.1.1 Topologias

A topologia de uma rede define a forma de interligação das estações na rede. As topologias de interesse no contexto de redes locais são aquelas baseadas em *difusão* e em *ligações ponto-a-ponto.* Numa rede baseada em difusão, os sinais injetados no meio por uma estação são difundidos para todas as estações ligadas ao meio físico. Numa rede baseada em ligações ponto-a-ponto, as estações são interligadas aos pares formando estrelas ou anéis.

Cada estação na rede possui um *endereço* que corresponder ao *ponto de conexão física* da estação à rede. O endereço também pode estar definido nos circuitos ou programas da estação propriamente dita. O *endereço físico* de uma estação geralmente é determinado por um número associado à interface de rede. A associação entre endereço físico e adaptador de interface de rede pode ser efetuada pelo fabricante do adaptador ou pelo administrador da rede. O *endereço lógico* de uma estação geralmente depende de sua localização na rede. A associação entre os endereços físico e lógico é determinada pelo administrador da rede quando esta é configurada.

Difusão

Os meios físicos que permitem a difusão são o par trançado, cabo coaxial e a radiodifusão em meio livre. A diferença entre difusão em cabos e em meio livre é que a geometria do cabo limita o espaço onde os sinais se propagam *i.e.,* os sinais seguem o percurso imposto pela geometria do cabo. Na difusão em meio livre não há limites para a propagação dos sinais. Redes baseadas em radiodifusão, rádio-enlaces via satélite ou microondas são discutidas em [Sta97].

Difusão num barramento Um barramento consiste de um segmento de cabo onde se ligam as estações da rede, como mostra a Figura 5.1. Quando uma estação injeta um quadro no barramento, todas as outras estações observam a passagem daquele quadro pelos seus pontos de conexão física. Os quadros contém o endereço do destinatário e do remetente. Quando uma

estação detecta um quadro endereçado para si, a estação copia aquele quadro para um armazenador em memória. As estações simplesmente ignoram os quadros que não são endereçados à elas. A Figura 5.1 mostra a estação $\mathcal{E}_1$ enviando um quadro para a estação $\mathcal{E}_3$. As estações $\mathcal{E}_0$ e $\mathcal{E}_2$ ignoram o quadro após verificar que seu destinatário é $\mathcal{E}_3$. Os pulsos correspondentes aos bits do quadro se propagam ao longo do cabo até os terminadores nas duas extremidades, onde são absorvidos.

Esta topologia é relativamente simples e barata, sendo muito popular em redes locais de baixo custo. Contudo, a limitação na aplicabilidade de barramentos advém de limites no comprimento de um segmento e no número de estações que podem se conectar a um segmento de cabo. A limitação no comprimento se relaciona à atenuação dos sinais com a distância. A limitação no número de estações advém da atenuação que os sinais sofrem ao atravessar cada tomada de conexão e à reflexão dos sinais nas descontinuidades do meio físico criadas pelas tomadas. Para a rede local Ethernet, um segmento de cabo coaxial fino pode se extender por 185 metros e conter até 30 tomadas –mais detalhes na Seção 5.4.

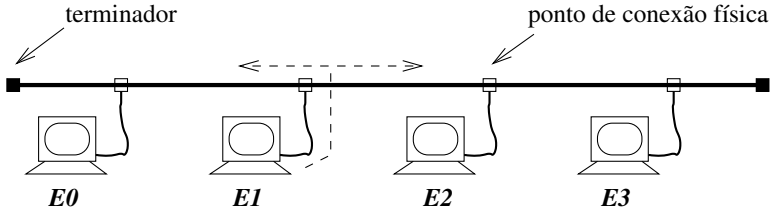


Figura 5.1: Difusão através de barramento.

Ligações Ponto-a-ponto

Uma alternativa ao uso de barramentos para interligar várias estações consiste em usar ligações ponto-a-ponto entre pares de estações. A saída de uma estação é ligada à entrada da seguinte de forma a que todas as estações na rede formem um circuito fechado. A vantagem das ligações ponto-a-ponto sobre barramentos se deve a maior simplicidade do projeto elétrico das interfaces. O circuito que implementa a interface é muito mais simples porque os sinais se propagam por um segmento de cabo somente e este segmento interliga apenas duas interfaces. Por estas razões, para uma mesma tecnologia de circuitos integrados, a taxa de sinalização pode ser mais alta numa rede com ligações ponto-a-ponto do que num barramento.

Anel A topologia mais simples com ligações ponto-a-ponto é o *anel unidirecional*. Num anel unidirecional, a saída de uma estação é ligada à entrada da próxima. A Figura 5.2 mostra cinco estações interligadas por um anel. A propagação dos sinais ao longo do anel é num único sentido (horário ou anti-horário). Quando a estação $\mathcal{E}_1$ envia um quadro para $\mathcal{E}_3$, a estação $\mathcal{E}_2$ deixa o quadro passar porque ele é destinado a outrem. $\mathcal{E}_3$ faz uma cópia do quadro enquanto este passa pela sua interface de rede. As estações $\mathcal{E}_4$ e $\mathcal{E}_0$ também ignoram o quadro. A estação que injetou o quadro, $\mathcal{E}_1$, remove o quadro do anel quando este retornar a si. Se a estação $\mathcal{E}_2$ envia um quadro para a estação $\mathcal{E}_1$, o quadro faz a volta completa pelo anel, passando pelas interfaces das estações $\mathcal{E}_3$, $\mathcal{E}_4$ e $\mathcal{E}_0$. Num *anel bidirecional*, as ligações são duplex e as estações podem transmitir para seus dois vizinhos.

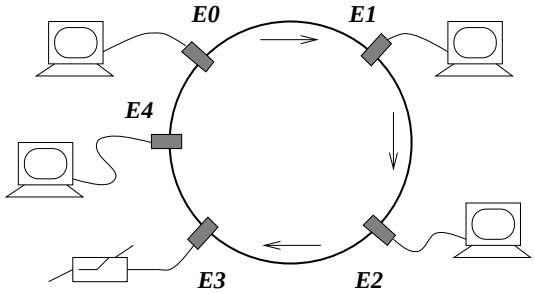


Figura 5.2: Rede em anel.

Estrela Uma rede em estrela pode ser encarada como um anel dobrado onde todas as ligações entre duas estações adjacentes passam através de um *concentrador* no centro da estrela. Esta topologia apresenta vantagens e desvantagens com relação ao anel. A maior desvantagem está em que o concentrador é o ponto mais crítico da rede: uma pane no concentrador e toda a rede fica fora de uso. A maior vantagem está na possibilidade da comutação dinâmica de circuitos internamente ao concentrador. Se a estação $\mathcal{E}_0$ envia um quadro para a estação $\mathcal{E}_3$, o concentrador pode estabelecer um circuito exclusivo entre aquelas estações enquanto durar a transmissão do quadro. As demais estações podem se comunicar porque o concentrador também é capaz de estabelecer caminhos entre aquelas, independentemente da comunicação entre $\mathcal{E}_0$ e $\mathcal{E}_3$. Outra vantagem importante desta topologia é a utilização de um sistema simples de cabeamento. Cada estação é ligada à rede por um único cabo duplex e isto simplifica a instalação do cabeamento. Outra vantagem decorre do aumento da confiabilidade da rede: se um dos cabos é seccionado, somente uma estação é desligada da rede.

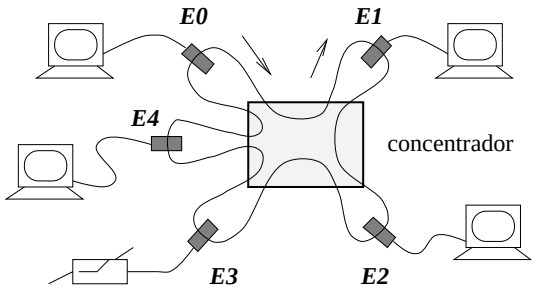


Figura 5.3: Rede em estrela.

5.2 Nível de Enlace

Em geral, o Nível de Enlace de redes locais é subdividido em dois subníveis, *Controle de Acesso ao Meio* (CAM) e *Controle de Enlace Lógico* (CEL). Esta divisão permite o uso de diferentes métodos de controle de acesso ao meio, cada um adequado às características particulares do meio físico. A interface funcional entre os vários métodos de acesso ao meio e o Controle de Enlace Lógico é uma só, independente do meio físico usado na rede local.

A Figura 5.4 mostra a subdivisão do Nível de Enlace em CEL e CAM e a funcionalidade implementada em cada subnível. O Nível Físico mantém as estações sincronizadas bit a bit (os ECDs devem identificar o valor e a duração de cada bit injetado no meio físico) e quadro a quadro (os ETDs devem identificar o início e o final de cada quadro). O Subnível Controle de Acesso ao Meio sincroniza e ordena os acessos ao meio físico de forma a resolver conflitos de acesso e a identificar as estações ligadas ao meio físico. O subnível de Controle de Enlace Lógico determina

a forma de detecção e recuperação de erros de transmissão, determina as interações e o diálogo entre estações bem como a forma de controle de fluxo através do enlace.

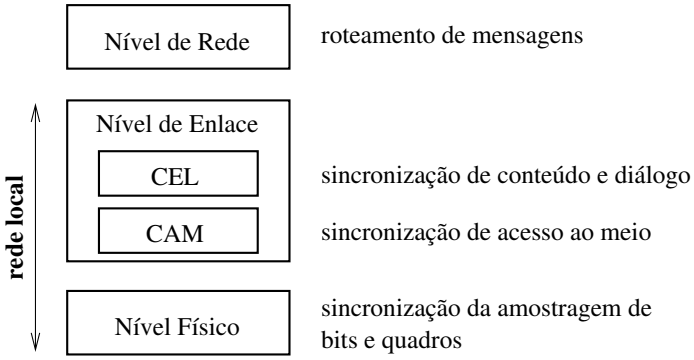


Figura 5.4: Subdivisão do Nível de Enlace.

Do ponto de vista de equipamentos, a funcionalidade do Nível Físico e dos Subníveis CEL e CAM é implementada por um ou mais circuitos integrados relativamente sofisticados. Estes, juntamente com os circuitos analógicos necessários para o acoplamento ao meio físico, formam um *adaptador* de rede. Geralmente, em computadores de baixo custo o adaptador deve ser comprado separadamente (*e.g.* cartões Ethernet para PCs). Em estações de trabalho de maior capacidade, o adaptador pode fazer parte da placa de circuito impresso onde se encontra o processador ou, deve ser adquirido separadamente para permitir maior flexibilidade de escolha pelo usuário *e.g.* adaptadores de interface para redes de alta velocidade em máquinas de alto desempenho.

Controle de Acesso ao Meio O subnível de *Controle de Acesso ao Meio* decide quando qual estação tem o direito de transmitir seus quadros. O *quando* está relacionado a eqüidade no acesso ao meio físico de forma a que todas as estações tenham vez na transmissão e que nenhuma estação monopolize o uso do meio em detrimento das demais. A escolha de *qual* estação pode transmitir num dado instante também está relacionada a eqüidade no uso do meio físico. Geralmente, se estabelece uma ordem lógica circular entre as estações na rede. O subnível CAM também define o método de endereçamento adequado ao meio físico da rede.

Controle de Enlace Lógico O subnível *Controle de Enlace Lógico* (CEL) permite a diferenciação das mensagens de controle das mensagens que transportam dados e garante que uma única cópia correta da cada quadro é entregue ao Nível de Rede. A sincronização de diálogo garante o estabelecimento, manutenção e término ordenado do diálogo entre duas estações. Isso implica no correto sequenciamento de mensagens de controle e de dados [Lam84b].

5.2.1 Controle de Acesso ao Meio

O método de acesso ao meio físico por consulta e seleção, apresentado na Seção 4.2, centraliza em uma estação as decisões de *quem* vai transmitir e por *quanto* tempo. As desvantagens deste método advém da existência de um ponto no sistema –a estação primária– cuja falha paraliza toda a rede. Além disso, se as distâncias entre estações forem muito grandes, o tempo perdido na propagação dos sinais entre mestre e escravos reduz drasticamente a utilização do meio físico.

A alternativa ao controle centralizado consiste em distribuir a responsabilidade pelo acesso ao meio entre todas as estações a ele ligadas. Existem basicamente três métodos de controle distribuído. O primeiro deles, consiste em permitir que todas as estações tenham a oportunidade de transmitir durante um certo intervalo. Um ordenamento circular é definido e cada estação

transmite quando chegar a sua vez. Este método é chamado de *controle de acesso por revezamento* ou *passagem de bastão*. Quando chega sua vez, uma estação pode transmitir durante o intervalo permitido ou, declinar do direito se não tiver dados e transmitir. Note que o método de revezamento com controle centralizado é equivalente a consulta e seleção.

No segundo método de controle distribuído, as estações fazem uma reserva antecipada da fração da capacidade do meio que elas esperam utilizar no futuro próximo. Este método é chamado de *controle de acesso ao meio por reserva*. O tempo é dividido em intervalos de tamanho fixo e uma estação transmite seus dados nos intervalos que reservou para si. Este método é uma solução intermediária entre multiplexação por divisão de tempo e multiplexação estatística.

Com o terceiro método de controle distribuído, as transmissões de uma estação iniciam assim que ela tiver dados a transmitir. Se duas estações transmitirem ao mesmo tempo, ocorre uma *colisão* entre as transmissões e os dois quadros são inutilizados e devem ser retransmitidos mais tarde. Por causa das colisões, este método é chamado de *controle de acesso com contenção*. O meio físico deve permitir a difusão dos quadros para todas as estações, inclusive para o transmissor. Quando um quadro é inserido no meio, o transmissor examina os sinais presentes no meio. Se o sinal recebido é diferente do transmitido então ocorreu uma colisão.

Quando o tráfego é esporádico e as mensagens tem tamanho pequeno, o controle de acesso com contenção permite uma melhor utilização do meio físico por todas as estações. Se o tráfego é constante e as mensagens são longas (*e.g.* transferência de arquivos), o método de reserva possibilita a melhor utilização da capacidade de meio físico. Para distâncias e mensagens relativamente curtas, o método do revezamento permite a utilização do canal de forma mais eficiente.

Acesso ao Meio – Difusão

Quando várias estações compartilham do meio físico, as transmissões devem ser coordenadas de forma a permitir a máxima utilização possível da rede. Se duas ou mais estações transmitem ao mesmo tempo, ocorre uma colisão entre os quadros e os sinais são distorcidos e detectados erradamente pelas demais estações. Os quadros envolvidas na colisão deverão ser retransmitidas novamente, o que reduz a utilização da rede.

Aloha O primeiro método de controle de acesso ao meio de sucesso foi desenvolvido em fins da década de 60 na Universidade do Havaí para interligar seus campi espalhados pelas ilhas do arquipélago do Havaí. O método, chamado de protocolo Aloha, é muito simples. As estações são interligadas por enlaces de rádio. Sempre que uma estação deseja transmitir ela pode fazê-lo. Sempre que ocorrer uma colisão, os quadros dela participantes são danificados. As estações verificam a correção de cada quadro recebido pelo seu bloco de paridade. Quadros com erros são descartados. A utilização do meio, como seria de se esperar, é baixa. Quando as estações transmitem muitos quadros, ocorrem muitas colisões e consequentemente, muitas re-transmissões, o que aumenta ainda mais o tráfego. Para quadros de mesmo tamanho, a utilização máxima do meio é de 18%, antes que as colisões reduzam a vazão efetiva das estações.

Aloha discreto O protocolo Aloha original permite que as estações transmitam a qualquer instante *i.e.,* o tempo é uma variável contínua no protocolo. Uma segunda versão do protocolo pressupõe que o tempo é dividido em intervalos e que as estações só podem transmitir no início de cada intervalo. A duração do intervalo é o tempo necessário para a transmissão de um quadro. Este protocolo é portanto chamado de *Aloha discreto*. A utilização máxima do meio físico com este protocolo é 37%, o dobro daquela do Aloha original. Isso decorre da redução da largura da janela onde as colisões podem ocorrer: se há uma colisão, ela envolve apenas quadros completos.

Detecção de portadora As distâncias envolvidas em redes como Aloha são tais que o tempo de propagação é maior que o tempo de transmissão dos quadros. Se o inverso for verdade, tornam-se viáveis os métodos baseados na *detecção de portadora* ["Carrier Sense Multiple Access" (CSMA)]. Quando o tempo de propagação é pequeno, as estações podem 'escutar' o meio de transmissão antes de iniciar suas transmissões. Caso haja uma transmissão em curso *i.e.,* alguma outra estação está inserindo um sinal portador de informação no meio, a estação não transmite. As estações com quadros a transmitir esperam até que haja silêncio no meio. Se ocorrer uma colisão, a estação espera um certo intervalo de tempo, escolhido aleatoriamente, e tenta nova transmissão. A utilização máxima neste caso chega a 50% porque as estações não interferem com transmissões em andamento, como é o caso das duas versões de Aloha.

Este protocolo é chamado de *1-persistente* porque uma estação transmite com probabilidade 1 sempre que o canal estiver em silêncio. Em outra versão deste protocolo, chamada *não-persistente*, se o canal está ocupado, a estação espera um intervalo de duração aleatória antes de tentar sua transmissão. Protocolos não-persistentes possibilitam melhor utilização que os 1-persistente embora, na média, o tempo de viagem das mensagens seja maior nos sistemas com protocolo não-persistente.

Uma melhoria sobre o método de detecção de portadora consiste em acrescentar às interfaces de rede a capacidade de detecção de colisões ["Carrier Sense Multiple Access with Collision Detection" (CSMA/CD)]. Quando uma estação inicia sua transmissão e percebe que ocorreu uma colisão, ela aborta sua transmissão imediatamente, transmitindo apenas um fragmento (relativamente) pequeno do quadro. As estações cujas transmissões colidiram esperam por intervalos escolhidos aleatoriamente antes de tentar retransmitir. A utilização do meio é maior que com CSMA porque há menos desperdício de tempo porque as estações não transmitem os quadros completos quando ocorrem colisões – elas mantém posse do meio apenas o tempo necessário para detectar a colisão.

Acesso ao Meio – Anéis

Passagem de bastão A Figura 5.5 mostra um dos tipos de interface normalmente usadas nas redes em anel. Cada interface consiste de um seletor de entrada, um registrador de deslocamento e um seletor de saída. Suponha que o seletor de entrada esteja na posição de recepção (RX) e que a estação não esteja inserindo um quadro no anel (seletor de saída na posição RX). Os bits transmitidos pela estação anterior no anel são deslocados um a um para dentro do reconhecedor de endereço e ao mesmo tempo, transferidos para a saída através do registrador e seletor de saída. Os bits recebidos são re-transmitidos após atraso correspondente a um ou mais intervalos de bit t_b.

Os quadros contém o endereço do destinatário. Quando uma interface reconhece o seu próprio endereço num quadro, este é copiado para um armazenador em memória. Quando uma estação reconhece o endereço de alguma outra estação, o conteúdo do reconhecedor de endereços é descartado. Quando uma estação vai transmitir, e portanto inserir um quadro no anel, o seletor de saída liga o transmissor à interface de saída (TX). Se a estação sofre uma pane, ela pode se desligar tanto lógica quanto fisicamente do anel fazendo com que os seletores redirecionem os sinais recebidos através do desvio (DSV). Assim, todos os bits recebidos são transferidos diretamente da entrada para a saída da interface, removendo a estação do anel.

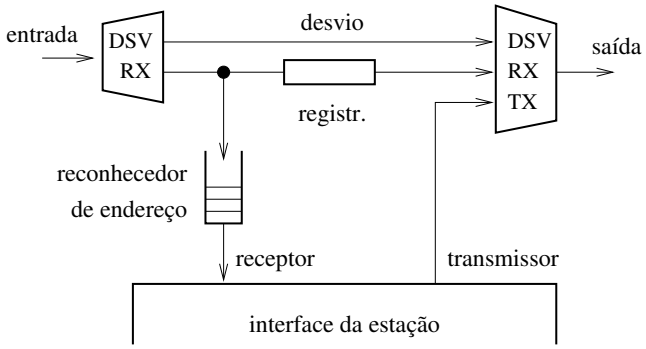


Figura 5.5: Interface para rede em anel.

O controle de acesso ao meio por *passagem de bastão* é similar a uma corrida com revezamento onde cada atleta carrega um bastão enquanto corre uma certa distância, finda a qual, um bastão é passado para o atleta que vai correr na próxima parte do trajeto. Quando este completa sua parte, o segundo atleta passa o bastão para o terceiro, e assim por diante. No método de CAM por passagem de bastão, o 'bastão' corresponde a uma mensagem ou quadro especial que dá ao seu detentor o direito a injetar quadros no meio físico. Quando a passagem de bastão é implementada num anel, cada estação que recebe o bastão transmite um ou mais quadros e passa o bastão para a estação vizinha no anel. Este método de controle de acesso ao meio também pode ser implementado num barramento. Neste caso, uma ordem entre as estações deve ser estabelecida para definir a sequência de passagem do bastão, de forma a que este circule em um anel lógico.

Um quadro consiste de preâmbulo, cabeçalho com os endereços de destinatário e remetente, informações de controle, dados, bloco de paridade e status, como mostra a Figura 5.6. O bastão, por sua vez, contém um preâmbulo, delimitador de início, campo de controle e delimitador de final. De posse do bastão, uma estação pode injetar seus quadros no anel. O destinatário copia o quadro para sua memória, alterando o campo de status para informar ao remetente de que o quadro foi recebido com sucesso. O quadro completa a volta pelo anel e retorna ao remetente, que o remove do anel, evitando assim que o quadro fique circulando eternamente. O remetente analisa o campo de status para descobrir se o quadro foi recebido ou se algum erro ocorreu. O protocolo de CAM deve impor um limite máximo no intervalo em que cada estação pode reter o bastão. Isso evita que uma estação retenha indefinidamente o bastão e portanto monopolize o uso do anel.

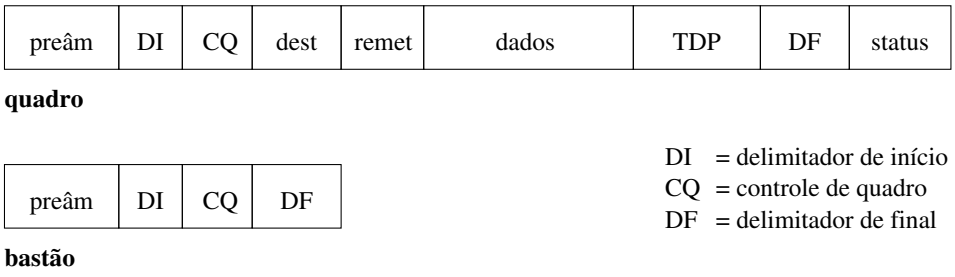


Figura 5.6: Formato de quadro de dados e bastão.

Anel segmentado A Figura 5.7 mostra uma rede cujo método de acesso ao meio é o de um *anel segmentado* Neste tipo de rede, ao invés do registrador de um bit, como aquele mostrado na Figura 5.5 (pág. 137), a interface de nível físico de um anel segmentado contém um registrador de deslocamento com vários bits. As interfaces de todas as estações no anel formam um registrador

de deslocamento que é composto por todos os registradores das interfaces de rede no anel. A concatenação dos registradores individuais forma um registrador de deslocamento que é longo o suficiente para acomodar vários quadros. O tamanho de um quadro determina o número de bits de cada segmento do anel. Os segmentos tem tamanho fixo igual ao de um quadro e o número de segmentos num anel depende da quantas estações compõem o anel. O número de segmentos é determinado dinamicamente quando a rede é inicializada. Note que um anel com um único segmento se comporta como um anel com passagem de bastão. Na Figura 5.7, cada interface de rede contribui com 8 elementos ao anel e os segmentos consistem de 2 elementos de cabeçalho (sombreados) e 8 elementos de informação.

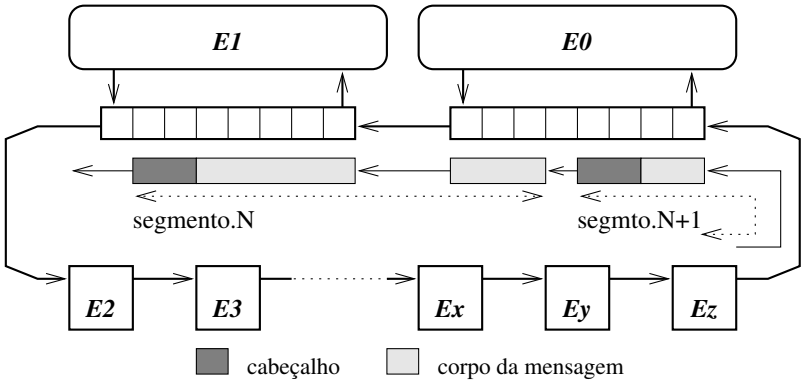


Figura 5.7: Anel segmentado.

Um segmento tem o tamanho de um quadro e se comporta como uma caixa que pode ou não conter uma mensagem. Cada segmento consiste de um campo de controle com bits de estado que indicam se o segmento está vazio ou contém uma mensagem, e se a mensagem transportada pelo segmento foi aceita pelo destinatário. Ao contrário do que ocorre num anel com passagem de bastão, num anel segmentado todas as mensagens inseridas no anel devem ter um tamanho igual ou menor que o de um segmento. Mensagens longas devem ser divididas e transmitidas em vários segmentos.

A Figura 5.8 mostra uma interface para anel segmentado. Os bits recebidos na entrada são deslocados em direção à saída na cadência de transmissão. Quando o reconhecedor de endereços reconhece o endereço da estação, ele copia o quadro e marca o segmento que o transporta como vazio. Quando o transmissor detecta a chegada de um segmento vazio, este transfere sua mensagem da fila de saída para o segmento e o marca como cheio.

Uma mensagem pode ser rejeitada pelo destinatário porque sua fila de recepção está cheia ou este detectou um erro na mensagem. Neste caso, quando o segmento que transporta a mensagem passa novamente pelo remetente, este preenche o segmento com a mensagem original e o marca como cheio. Se a mensagem foi aceita pelo destinatário, o remetente marca o segmento como vazio e este pode então ser usado para novas transmissões. Com este método de acesso ao meio, em média, um segmento transporta mensagens durante metade do comprimento do anel e aceitações durante a outra metade.

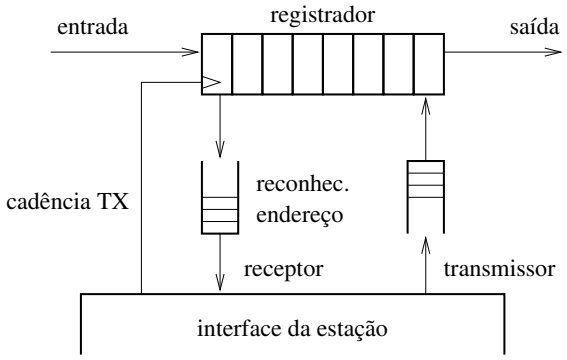


Figura 5.8: Interface de anel segmentado.

Para um certo número de estações e quadros de mesmo tamanho, comparando-se um anel com passagem de bastão com um anel segmentado, este último é mais eficiente porque permite que vários quadros estejam em viagem simultaneamente. No anel com passagem de bastão, somente uma estação pode inserir seus quadros a cada instante, enquanto que num anel segmentado tantas estações quantos forem os segmentos podem transmitir simultaneamente. Obviamente, esta comparação só é válida se o número de estações no anel for tal que existam pelo menos dois segmentos.

Inserção de registrador A Figura 5.9 mostra o diagrama de blocos da interface de um anel com inserção de registrador. As filas de desvio formam um registrador de deslocamento cujo tamanho varia com a intensidade de tráfego no anel. Um *símbolo* corresponde a uma unidade de transferência de informação do nível físico (*e.g.* 16 bits a cada pulso de relógio de transmissão). Um símbolo pode estar *vazio* quando não pertence a nenhum quadro e serve somente para sincronizar as estações ao longo do anel. Quando não há tráfego pelo anel (*i.e.,* nenhuma estação transmitindo), as filas de desvio contém somente *símbolos vazios*.

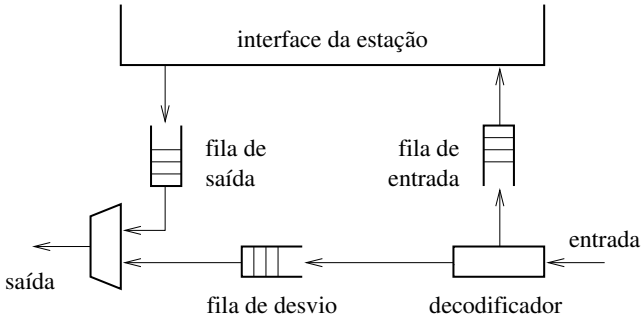


Figura 5.9: Interface de anel com inserção de registrador.

Cada quadro consiste de uma sequência contínua de símbolos, contém os endereços do remetente e do destinatário, informação de controle e estado, dados opcionais e uma palavra de paridade. A cada quadro transmitido corresponde um quadro de *eco* que é o quadro original sem o campo de dados e com os endereços de remetente e destinatário trocados. Ao receber o eco de um quadro, o remetente tem a confirmação de que o quadro foi aceito ou rejeitado. A Figura 5.10 mostra um quadro de dados e um quadro de eco.

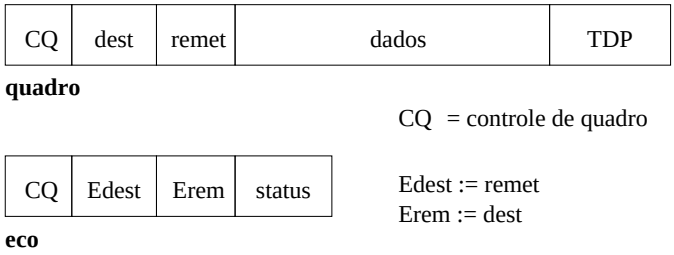


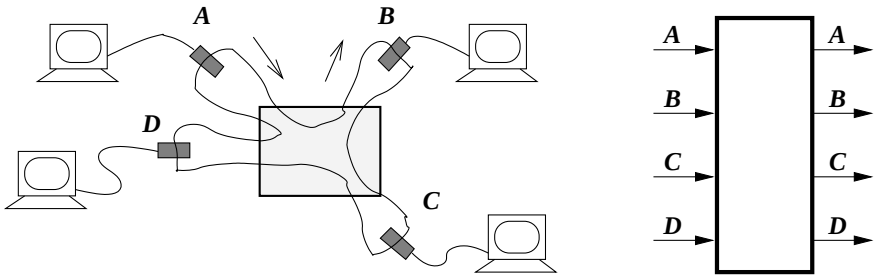
Figura 5.10: Quadro e eco de anel com inserção de registrador.

Cada quadro recebido à entrada da interface do anel é examinado pelo decodificador. Se o quadro é destinado à estação, ele é transferido para a fila de entrada e removido do anel. Em seu lugar, o decodificador insere na fila de desvio um quadro de eco endereçado ao remetente do quadro removido, informando-o da aceitação do quadro. Quando uma estação deseja transmitir um quadro, este é inserido na fila de saída. Se a fila de desvio estiver vazia, a transmissão pode iniciar imediatamente. Senão, a transmissão só se inicia quando a fila de desvio ficar vazia. Os quadros de eco são pequenos, consistindo de uns poucos símbolos. Quando um quadro é transferido para a fila de entrada, o eco que o substitui, por ser pequeno, produz uma redução líquida no tamanho da fila de desvio. Dessa forma, cada mensagem aceita abre espaço para novas transmissões.

A Scalable Coherent Interface 5.6 é o padrão IEEE 1956-92 que define uma rede confinada para a interconexão de até 65000 nodos de processamento. O sistema de comunicação da SCI é baseado num anel com inserção de registrador capaz de operar na velociade 1 Gbyte/s com ligações ponto-a-ponto unidirecionais de 17 bits de largura (16 de dados e um de sincronismo).

Meio Compartilhado e Comutação

Na maioria das redes locais projetadas há alguns anos, o meio físico é compartilhado entre todas as estações na rede. Devido aos avanços na microeletrônica, atualmente é possível e barato implementar redes locais com topologias em estrela, baseadas em ligações ponto-a-ponto entre várias estações e um concentrador. Os circuitos internos ao concentrador implementam mecanismos de controle de acesso ao meio que organizam a comunicação entre as estações. Por exemplo, concentradores para redes com tecnologia Ethernet se comportam como um segmento de cabo coaxial ao qual estão ligadas as estações. A Figura 5.11 mostra uma rede local com um concentrador de quatro portas e um diagrama do comutador com as quatro entradas no lado esquerdo e as quatro saídas no lado direito. O comutador da figura possui quatro pares <*entrada, saída*>, com entradas e saídas desenhadas em faces opostas para simplificar o diagrama.



(a) Rede com concentrador. (b) Concentrador.

Figura 5.11: Rede local baseada em concentrador.

O controle de acesso ao meio por contenção é relativamente ineficiente por causa das colisões e das conseqüentes re-transmissões. Existem várias alternativas para a implementação

das funções de um concentrador numa rede em estrela. Algumas destas alternativas consistem no estabelecimento de ligações entre as portas de entrada e de saída do concentrador de forma dinâmica. Neste caso, o concentrador é chamado de *comutador* porque sua função é a de comutar as ligações internas de forma a satisfazer a demanda de comunicação entre as estações. Um comutador é similar a uma central telefônica. Todos os telefones do bairro são ligados à central mas as ligações entre pares de terminais são estabelecidas dinamicamente de acordo com os números discados pelos usuários. Num comutador de redes de dados as ligações podem ser estabelecidas segundo os dois métodos básicos de comutação: comutação de circuitos e comutação de pacotes. Estes são apresentados nos próximos parágrafos.

Comutação de circuitos Uma forma de transferir quadros através de um comutador é a *comutação de circuitos*. Neste caso, a cada quadro transmitido através do comutador, este estabelece uma ligação ponto-a-ponto entre emissor e receptor enquanto durar a transmissão do quadro. Esta ligação ponto-a-ponto é interna ao comutador e é chamada de *circuito*. A Figura 5.12 mostra um comutador de circuitos com quatro portas, onde se ligam quatro estações. Enquanto $\mathcal{B}$ transmite um quadro para $\mathcal{D}$, $\mathcal{A}$ pode transmitir outro quadro para $\mathcal{C}$. Se, enquanto existir um circuito entre $\mathcal{B}$ e $\mathcal{D}$, outra estação tentar transmitir para $\mathcal{D}$, esta segunda transmissão fica bloqueada por causa do conflito de acesso à porta $\mathcal{D}$. Na ausência de conflitos, um comutador com N portas pode completar N circuitos e transferir N quadros simultaneamente. Isso corresponde a um aumento por um fator de N na faixa de passagem do meio físico. Na prática, o aumento é menor que N porque os quadros são de tamanho variável e a distribuição de tráfego entre as estações pode provocar conflitos freqüentes.

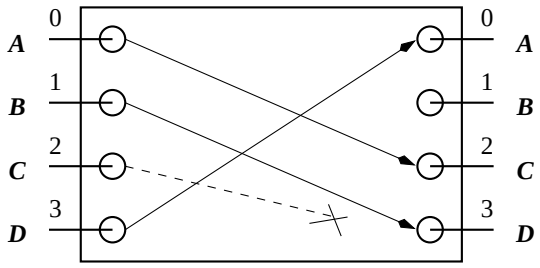


Figura 5.12: Comutador de circuitos.

Comutação de pacotes A *comutação de pacotes* é uma técnica de comutação geralmente empregada para interligar estações em redes de longa distância. Neste caso, um comutador de pacotes é um dispositivo de Nível de Rede e o comutador examina o campo de informação de cada quadro recebido para poder decidir por qual caminho a mensagem contida no quadro deve ser re-transmitida. O comutador remove o *pacote* de Nível de Rede contido no quadro e o armazena em sua memória enquanto o caminho de saída do pacote é escolhido. Os pacotes são de tamanho variável, com tamanhos máximo e mínimo definidos pelas tecnologias das redes ligadas ao comutador.

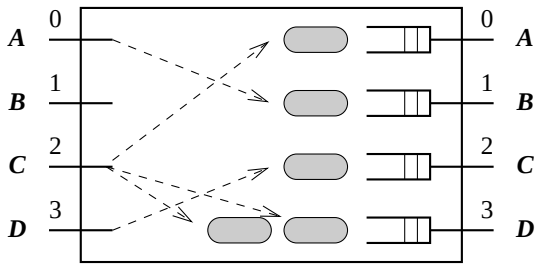


Figura 5.13: Comutação de pacotes.

A Figura 5.13 contém o diagrama de um comutador de pacotes com quatro portas. Um comutador pode ser construído com um computador dotado de várias interfaces de rede. Sob controle de um algoritmo de comutação, o computador transfere pacotes da entrada de uma interface de rede para sua memória, e de lá para a saída da interface de rede apropriada. Os pacotes, representados pelas elipses, são transferidos do terminal de entrada para a fila de saída apropriada. As filas são implementadas em memória interna ao comutador e armazenam os pacotes enquanto os terminais de saída transmitem pacotes depositados anteriormente nas filas.

Circuitos virtuais Quando uma rede é composta por várias sub-redes, onde cada sub-rede consiste de estações interligadas por comutadores de circuitos, a comunicação entre estações pode ser baseada na comutação de pacotes ou na comutação de circuitos. Caso seja empregada a comutação de circuitos, uma estação que deseja se comunicar com uma estação em outra sub-rede deve estabelecer um *circuito virtual* antes de iniciar suas transmissões. A Figura 5.14 mostra uma rede composta por três sub-redes e um circuito virtual entre as estações $\mathcal{A}$ e $\mathcal{B}$, em sub-redes distintas. Quando $\mathcal{A}$ deseja enviar uma mensagem para $\mathcal{B}$, a estação $\mathcal{A}$ solicita ao comutador a que está ligada o estabelecimento do circuito virtual com a estação $\mathcal{B}$. Este comutador, se possuir um canal disponível, repassa a solicitação ao próximo comutador no caminho entre $\mathcal{A}$ e $\mathcal{B}$. Se todos os comutadores concordarem, o circuito virtual é estabelecido e as estações podem trocar mensagens. Ao terminar a troca de mensagens, a estação $\mathcal{A}$ solicita a remoção do circuito virtual. Se um comutador estiver sobrecarregado, este se nega a estabelecer o circuito. Neste caso, a estação deve re-iniciar o processo de estabelecimento do circuito virtual, na esperança de que haja um canal disponível.

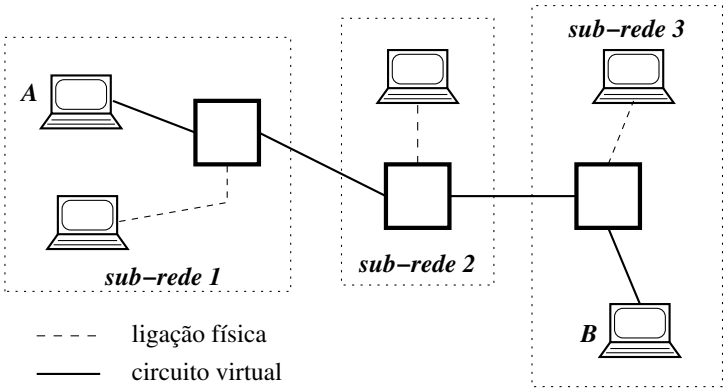


Figura 5.14: Comutação com circuitos virtuais.

Comutação de células Uma terceira forma de transferir dados entre as estações de uma rede é a *comutação de células*. Uma *célula* é a unidade de transmissão que contém endereços de emissor e desinatário, controle e uns poucos octetos de informação. As células tem tamanho fixo e o campo de informação comporta de 30 a 50 octetos, dependendo da implementação. Cada quadro é particionado e transportado em várias células. Por causa das células pequenas, os comutadores de células podem ser implementados para funcionamento extremamente rápido. Além disso, um comutador de células deve estabelecer circuitos virtuais entre pares de estações que desejam se comunicar. Os comutadores gerenciam estes circuitos de forma a maximizar a faixa de passagem efetiva entre as estações através de multiplexação estatística, acomodando assim flutuações no tráfego. Dessa forma, um circuito físico pode transportar vários fluxos de dados simultaneamente, cada um isolado em seu circuito virtual. Mais detalhes sobre comutação de células na Seção **??**.

A Figura 5.15 mostra as diferenças entre os dois tipos de comutação e circuitos virtuais. A figura mostra duas estações se comunicando através de três enlaces e dois comutadores, numa situação similar à da Figura 5.14. Na comutação de circuitos, a fonte do fluxo de dados deve solicitar o estabelecimento de um circuito até o destinatário (a). Os nós intermediários, se possuem recursos para satisfazer a solicitação, repassam a solicitação ao longo da rota até o destino. O nó destino informa à fonte de que o circuito está disponível para as transmissões (b) e estas se iniciam (c). Quando os dados são transferidos, o nó destino envia uma aceitação (d). O circuito poderia ser desfeito neste momento, caso não hajam mais dados a transmitir entre fonte e destino.

Na comunicação com circuitos virtuais, os passos (a) e (b) são similares à comutação de circuitos. Um circuito virtual é estabelecido entre fonte e destino e os dados são agrupados em pacotes ou células e então transmitidos através do circuito virtual. Na comutação de pacotes, os dados são agrupados em pacotes e remetidos através da rede. As decisões de roteamento são efetuadas para cada pacote individualmente. Dependendo das condições de tráfego na rede (enlaces ou nós intermediários congestionados por excesso de tráfego), pacotes adjacentes no fluxo podem seguir trajetórias diferentes através da rede. O protocolo de Nível de Rede ou Transporte deve re-seqüenciar os pacotes recebidos, entregando-os na ordem correta à entidade de aplicação.

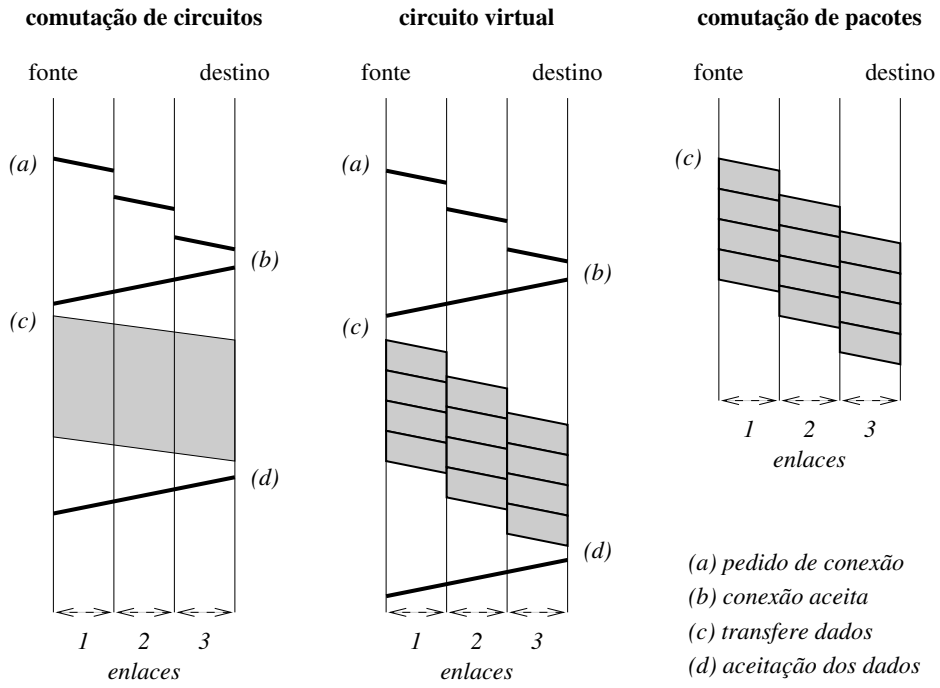


Figura 5.15: Comutação de pacotes, circuitos e células.

Estratégias de roteamento Existem basicamente duas estratégias de roteamento empregadas em comutadores. A estratégia de *redespacho* ("store and forward") consiste em transferir cada quadro recebido para um armazenador em memória para só então fazer a decisão de roteamento, escolhendo a porta de saída apropriada para transmitir o quadro. O comutador deve consultar uma tabela para determinar qual a porta de saída por onde transmitir o quadro. A estratégia de roteamento por *trespasse* consiste em examinar o campo de destinatário do quadro, fazer a decisão de roteamento e então re-transmitir o quadro imediatamente se o circuito de saída estiver disponível. Se o circuito de saída estiver bloqueado por outra transmissão, existem duas possibilidades de implementação. O quadro é depositado numa fila para transmissão posterior

ou então o quadro é descartado.

O redespacho é capaz de tolerar flutuações de tráfego através da rede porque as filas acomodam os quadros destinados a enlaces ou estações congestionadas. Por outro lado, a transferência da porta de entrada para memória e da memória para a saída é relativamente onerosa em termos do tempo de travessia da rede. O roteamento por trespasse é mais rápido porque não envolve armazenadores em memória. Por outro lado, em casos de conflito de destinatário, o descarte de quadros implica na sua retransmissão pelo remetente.

5.2.2 Controle de Enlace Lógico

As funções do Subnível de Controle de Enlace Lógico são similares às funções de controle de enlace do protocolo HDLC. A discussão que segue é baseada no padrão IEEE 802.2 – Logical Link Control. As funções de controle de enlace lógico são a detecção e recuperação de erros de transmissão, o seqüenciamento de quadros nos dois sentidos, o controle de fluxo e o controle de congestionamento. Numa rede local, existe a necessidade da diferenciação dos vários protocolos de Nível de Rede ou superiores que podem ser usados pelas estações na rede local. Esta diferenciação permite que as mensagens geradas pelos diferentes protocolos sejam transportadas por quadros do Nível de Enlace. Isso permite a multiplexação da comunicação entre diferentes protocolos através do meio físico.

A comunicação entre duas entidades de aplicação se estabelece através dos serviços oferecidos pelas interfaces dos protocolos de rede local. Quando uma entidade do Nível de Rede na estação $\mathcal{A}$ deseja se comunicar com sua equivalente na estação $\mathcal{B}$, a estação $\mathcal{A}$ invoca o serviço apropriado do Nível de Enlace. Um *Ponto de Acesso a Serviço* (PAS) ["Service Access Point" (SAP)] é a interface lógica para cada tipo de serviço suportado pelo Nível de Enlace. O par <*endereço–CAM, PAS*> determina de forma unívoca uma das duas extremidades da conexão lógica que interliga as entidades nas estações $\mathcal{A}$ e $\mathcal{B}$. Cada PAS corresponde a uma função ou subrotina do nível apropriado que é executada quando o serviço a ele associado é requisitado por uma entidade do nível imediatamente acima/abaixo.

O Subnível CEL pode suportar vários PAS diferentes na sua interface com o Nível de Rede, um para cada tipo de serviço oferecido às entidades de Rede. A interface com o Subnível CAM se dá através de um único PAS. A interface entre o Subnível CAM e o Nível Físico também se dá através de um único PAS.

A Figura 5.16 mostra a pilha de protocolos de duas estações de uma rede local. Naquela figura, cada Ponto de Acesso a Serviço é representado por um círculo. O PAS na estação $\mathcal{A}$ 'con(B, 2)' indica o endereço lógico da outra extremidade da conexão: a porta número 2 da estação $\mathcal{B}$. O PAS na estação $\mathcal{B}$ 'con(A,1)' indica que uma conexão existe com a porta 1 da estação $\mathcal{A}$. Nas estações, as portas 1 e 2 são associadas a uma classe de serviço; os endereços das estações são os endereços físicos empregados no Subnível CAM.

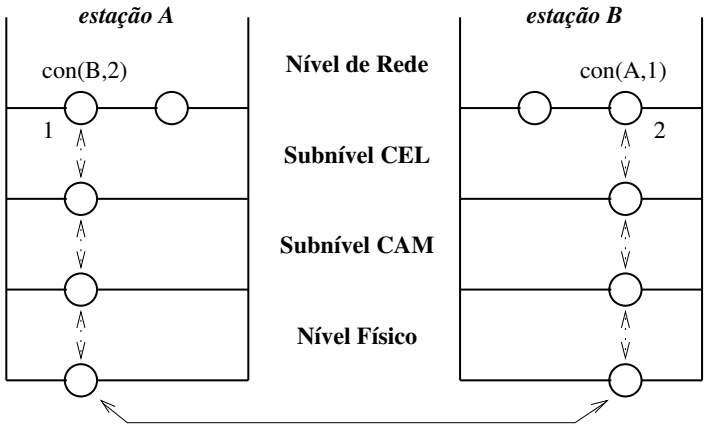


Figura 5.16: Conexão lógica via Pontos de Acesso a Serviço.

Tipos de Serviço

O modo de implementar a comunicação através de um enlace lógico é chamado de *tipo de serviço*. O Subnível CEL suporta três tipos de serviço, diferenciadas pela confiabilidade das transferências de dados e pela forma de intertravamento entre as estações nas pontas do enlace. O Tipo I provê comunicação precária entre estações, o Tipo II provê conexão confiável através de um circuito virtual e o Tipo III provê comunicação confiável sem estabelecimento de circuito virtual. As características dos três tipos de serviço são discutidas abaixo. Os protocolos de controle de enlace lógico 802.2 foram projetados para emprego em redes multiponto de acesso múltiplo. Num certo instante, uma estação pode estar engajada em comunicação com várias outras, usando o tipo de serviço apropriado em cada uma das conexões lógicas.

A Figura 5.17 mostra o relacionamento entre as funções usadas para implementar os três tipos de serviço suportados pelo Subnível CEL. Estas funções são chamadas de *primitivas* porque os três tipos de serviço de comunicação são implementados por diferentes combinações destas funções. O remetente (Nível de Rede) usa uma primitiva *comando* para enviar dados ou informação de controle ao destinatário. Uma *indicação* pelo sub-nível CEL informa da recepção de um comando. Uma *resposta* deve ser retornada nos serviços com aceitação. A recepção de uma resposta é sinalizada com uma *confirmação*.

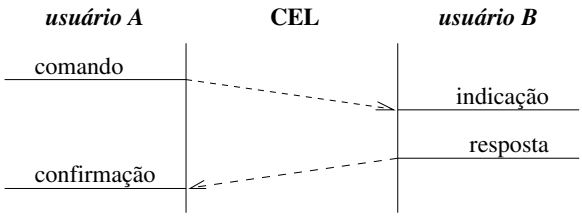


Figura 5.17: Primitivas de comunicação do Subnível CEL.

A Figura 5.17 mostra uma relação de causa e efeito entre, por exemplo, comando e indicação. As primitivas também podem ter causa ou efeito puramente locais, sem que sejam conseqüência da recepção, ou provoquem o envio, de uma mensagem através da rede. Por exemplo, um comando emitido pelo Nível de Rede provoca uma indicação no subnível de controle de enlace lógico da mesma estação. Da mesma forma, respostas e indicações podem ser geradas e consumidas localmente, apenas nas interfaces entre os níveis ou subníveis da pilha de protocolos de uma mesma estação.

Tipo I — Comunicação precária O primeiro tipo de serviço é a *comunicação precária sem aceitação* de mensagens ("best-effort"). Este serviço permite o envio de datagramas contendo dados sem nenhuma garantia de entrega porque o destinatário não informa da aceitação das mensagens. A *comunicação precária* admite perda ou duplicação de mensagens transferidas entre remetente e destinatário. O endereçamento pode ser ponto-a-ponto, um-para-muitos ou um-para-todos (difusão). Este tipo de serviço é empregado quando os protocolos de nível superior se encarregam de detectar e corrigir eventuais erros de seqüenciamento e de falta de entrega. Neste caso, não há necessidade de replicação destas funções no Subnível CEL. Esta classe de serviço é empregada em vários protocolos de redes locais, como TCP/IP e UDP/IP, por exemplo. A Figura 5.18 mostra as primitivas desta classe de serviço.

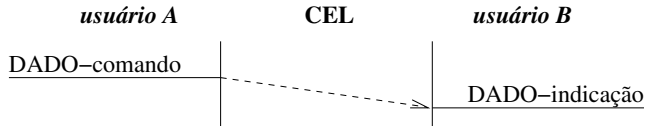


Figura 5.18: Comunicação Precária no Subnível CEL.

Tipo II — Conexão confiável O segundo tipo de serviço é a *conexão confiável* entre duas estações, similar àquela provida pelo protocolo HDLC em modo balanceado assíncrono. Este serviço garante que todos os bits enviados são recebidos corretamente e na ordem em que foram transmitidos. Uma conexão confiável é estabelecida entre duas entidades quando estas escolhem PAS de Tipo II para a conexão. Uma vez estabelecida a conexão confiável, dados são transferidos entre os dois PAS. Quando as estações completam as transferências, a conexão deve ser interrompida. Uma conexão confiável de Tipo II estabelece um circuito virtual entre dois PAS e portanto, este tipo de serviço suporta somente endereçamento ponto-a-ponto.

O Tipo II define o conjunto de serviços listado abaixo. Estes serviços permitem o estabelecimento, uso, re-inicialização e terminação de conexões de Nível de Enlace. Estas conexões são ligações ponto-a-ponto entre PAS. A Figura 5.19 mostra o estabelecimento, envio de uma mensagem e terminação de uma conexão.

serviço de estabelecimento de conexão permite a uma entidade solicitar ou ser notificada do estabelecimento de um enlace lógico;

serviço de transferência de dados via conexão permite a transferência de informação entre entidades através do enlace. Este serviço efetua seqüenciamento, controle de fluxo e recuperação de erros.

serviço de re-inicialização de conexão permite que entidades coloquem uma conexão já estabelecida num estado inicial bem-definido;

serviço de terminação de conexão permite a uma entidade solicitar ou ser notificada da terminação de um enlace lógico;

serviço de controle de fluxo permite o controle de fluxo de dados associado a uma conexão através da interface entre os Níveis de Enlace e de Rede.

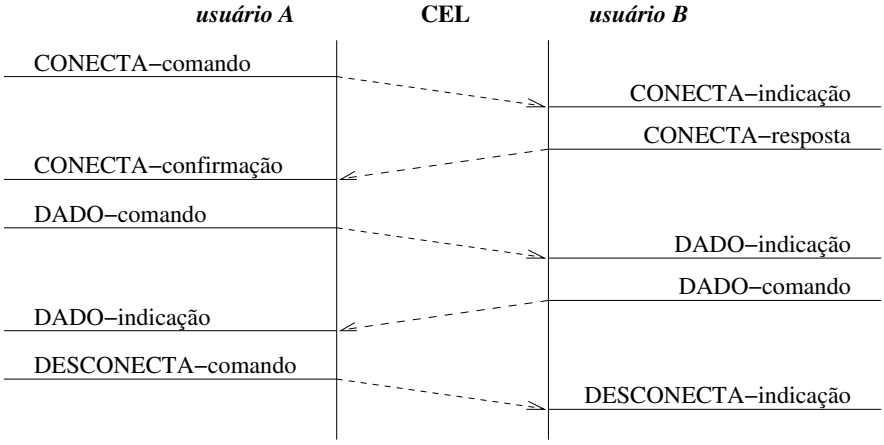


Figura 5.19: Conexão Confiável no Subnível CEL.

Tipo III — Comunicação confiável O terceiro tipo de serviço é a *comunicação confiável com aceitação* de mensagens. Este serviço permite a transferência confiável de dados entre duas estações sem necessidade do estabelecimento prévio de uma conexão. A confiabilidade advém da aceitação de cada mensagem pelo destinatário.

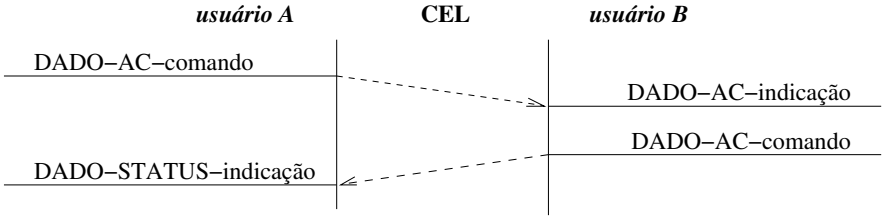


Figura 5.20: Comunicação confiável no Subnível CEL.

Classes de Estações As implementações do Controle de Enlace Lógico 802.2 podem suportar diferentes combinações de tipos de serviço. Uma estação de *Classe I* suporta somente serviços de Tipo I. Uma estação de *Classe II* suporta serviços de Tipo I e Tipo II. Estações da *Classe III* suportam serviços de Tipo I e Tipo III. Estações de *Classe IV* suportam os três tipos de serviço.

Encapsulamento A unidade de dados transmitida entre duas entidades do nível n é chamada de *unidade de dados do protocolo* (UDP) de nível n. A Figura 5.21 mostra as UDPs dos Subníveis CEL e CAM. Os endereços de origem e destino na UDP do Subnível CAM são endereços físicos das estações na subrede onde a UDP-CAM trafega. O campo de informação da UDP-CAM transporta uma UDP do Subnível CEL. Na UDP do Subnível CEL, os campos de destino e origem se referem a endereços de enlace lógico nas estações das duas pontas do enlace *i.e.,* aos PAS nas duas estações. O campo de informação da UDP-CEL transporta uma UDP do Nível de Rede.

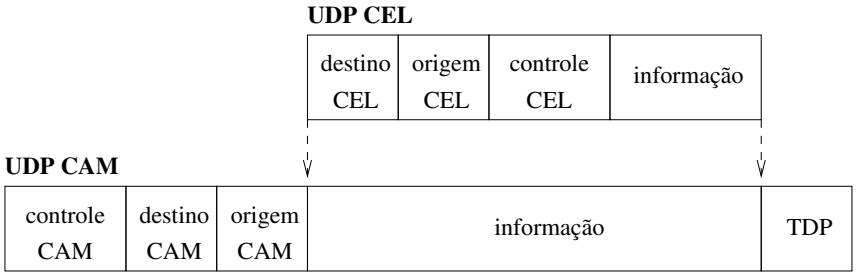


Figura 5.21: Encapsulamento de UDPs CAM e CEL.

A Figura 5.22 mostra a ligação entre as estações $\mathcal{E}_1$ e $\mathcal{E}_2$ através de uma rede local composta de duas subredes *sr-1* e *sr-2* interligadas por uma *ponte*. Estas subredes podem ou não empregar o mesmo tipo de meio físico e topologia. Se as subredes forem diferentes, os métodos de controle de acesso ao meio poderão ser diferentes também. Neste caso, cada subrede usa um protocolo do Subnível CAM diferente. A ligação lógica entre duas estações através de uma única subrede é chamada de *enlace de dados*. Em condições normais esta é uma ligação permanente. A ligação lógica entre duas estações na mesma rede local é chamada de *enlace lógico*. Um enlace lógico é estabelecido dinamicamente em função das necessidades das estações ou seus usuários. Uma estação pode manter enlaces lógicos com várias estações num dado instante e o enlace lógico com cada uma das estações é controlado independentemente dos demais.

Quando uma rede local consiste de subredes de tipos diferentes, uma *ponte* deve ser usada para interligar pares de subredes diferentes. Este dispositivo faz o mapeamento dos endereços empregados nos protocolos CAM das subredes que interliga e, se necessário, traduz o código de transmissão usado em uma subrede para o código da(s) outra(s). A funcionalidade do subnível CEL transforma várias sub-redes, possivelmente empregando tecnologias distintas, em uma única rede local.

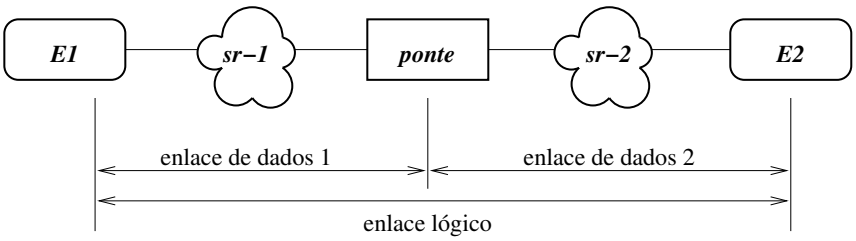


Figura 5.22: Enlace de dados e enlace lógico.

A interface entre os Subníveis CEL e CAM permite a combinação das várias tecnologias de uma forma transparente para os protocolos dos níveis superiores. A Figura 5.23 mostra em mais detalhe as ligações lógicas e físicas entre as estações $\mathcal{A}$ na subrede 1, e $\mathcal{B}$ na subrede 2 da Figura 5.22. As implementações dos Níveis Físico e CAM nas duas subredes são independentes uma da outra. As pontes do Subnível CEL efetuam todas as traduções necessárias para permitir a comunicação entre estações nas subredes que interligam. Os endereços usados pelos Subníveis CAM das subredes devem ser traduzidos se estes forem diferentes nas subredes. Como a linha tracejada indica, a ponte possui dois adaptadores de rede separados, cada um adequado para a tecnologia da subrede a que se conecta.

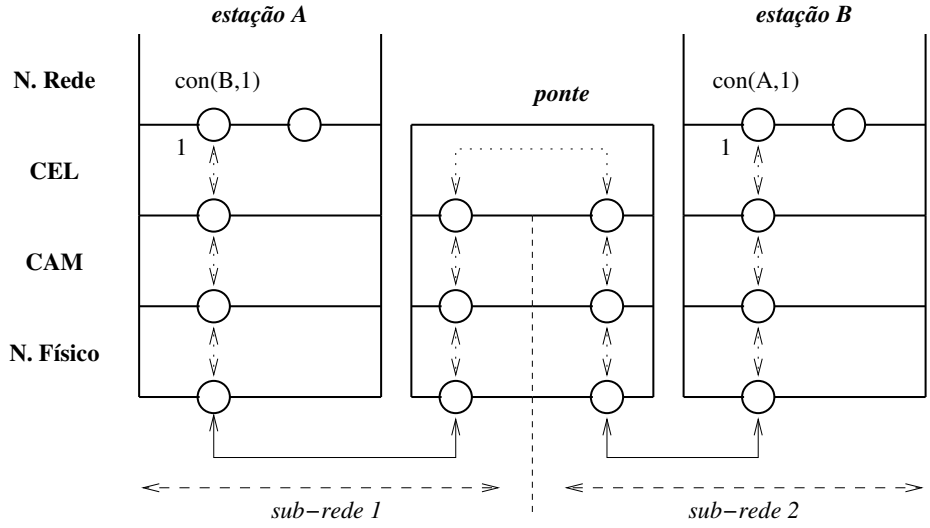


Figura 5.23: Conexões através de uma ponte entre duas subredes.

Uma ponte 802.2 deve ser transparente à difusão de mensagens. Quando uma estação numa das sub-redes interligadas pela ponte emite uma mensagem que deve ser difundida, a ponte restransmite esta mensagem para todas as outras sub-redes. Juntamente com a tradução de endereços e transcodificação dos alfabetos de transmissão, o suporte à difusão é o último requisito para que uma ponte transforme várias sub-redes físicas em uma única rede local.

5.3 Os Padrões IEEE 802 para Redes Locais

O Institute of Electrical and Electronics Engineers (IEEE) publica várias recomendações de padrões para redes locais e metropolitanas. Essas recomendações são, em geral, amplamente acatadas pelos fabricantes porque a venda de equipamentos que obedecem ao padrão, e portanto interoperam com os demais existentes no mercado, garante aumento nas vendas. Os padrões do IEEE para redes locais tem sido aceitos e publicados pela International Standards Organization.

O trabalho de normatização pelo IEEE é distribuído entre vários comitês e cada comitê publica padrões para a tecnologia sob sua responsabilidade. O padrão 802.1 especifica interfaces de alto nível e a arquitetura das redes, gerenciamento de redes, controle de acesso ao meio físico ["Medium Access Control", (MAC)] e pontes entre as redes da família 802. O padrão 802.2 define o Controle de Enlace Lógico (CEL), ["Logical Link Control" (LLC)], de forma independente do controle de acesso ao meio para as redes 802. O CEL é um subconjunto do protocolo HDLC (Seção 4.3.3). Os padrões 802.3, 802.4, 802.5 e 802.6 definem redes locais tipo CSMA/CD (Seção 5.4), barramento com passagem de bastão, anel com passagem de bastão e redes metropolitanas, respectivamente. Os padrões 802.7 e 802.8 tratam das camadas físicas de redes baseadas em transmissão em banda larga e óptica, respectivamente. O padrão 802.9 trata de redes integradas de voz e dados. O padrão 802.10 trata de aspectos de segurança em redes locais enquanto que o padrão 802.11 define redes com comunicação via sinais de rádio. A Figura 5.24 mostra o relacionamento entre os padrões 802.1, 802.2, 802.3, 802.4, 802.5 e FDDI (Seção 5.5).

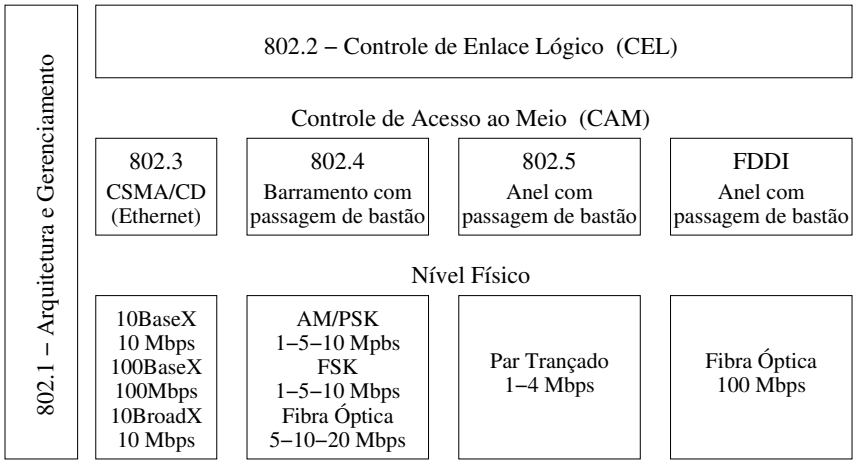


Figura 5.24: Padrões da família IEEE 802.

5.4 Ethernet

A rede local Ethernet é a mais popular dentre as redes locais disponíveis, detendo cerca de 80% do mercado mundial. Tal popularidade se deve à relativa facilidade de instalação e ao baixo custo por ponto de conexão. Na sua versão mais barata, uma Ethernet permite a conexão em modo semi-duplex, com taxa de transmissão de 10 Mbits por segundo, de até trinta máquinas, interligadas por um cabo com até 185 metros de comprimento. Estas características são suficientes para atender as necessidades de uma parcela considerável das instalações de pequeno e médio porte como escritórios e laboratórios, por exemplo.

A rede Ethernet foi concebida nos laboratórios do Xerox Palo Alto Research Center no início da década de 70 [MB76]. O projeto original foi revisado e melhorado nos laboratórios da Digital (DEC), Intel e Xerox, que publicaram a definição do Padrão Ethernet em 1980 [DIX80, SDRC84]. Em 1985 o IEEE publicou o Padrão IEEE 802.3 que passou então a ser o padrão "de fato" para redes locais de baixo custo [IEE85c]. A versão mais recente do padrão IEEE 802.3 é de 1996, publicada conjuntamente com ISO/IEC [ISO96]. Estritamente falando, a tecnologia de rede local descrita nesta seção se chama "IEEE 802.3 CSMA/CD". O nome original Ethernet tem uso tão difundido que é o nome usado aqui.

O IEEE designou um grupo de trabalho para definir um padrão para transmissões a 1 Giga bits por segundo. Este padrão, IEEE 802.3z, é conhecido como Gigabit Ethernet e deverá estar concluído até o final de 1998.

As próximas seções descrevem os Níveis Físico e de Enlace da rede Ethernet. O Nível de Enlace é dividido em CEL –formato e conteúdo dos quadros– e CAM –controle de acesso ao meio físico compartilhado. As camadas superiores dos protocolos de comunicação não são definidas pelo padrão Ethernet. Note que os valores para número de tomadas por segmento de cabo, comprimento do cabo, *etc* são definidos pelo padrão IEEE 802.3 mas fabricantes podem adotar valores diferentes para aqueles parâmetros de projeto.

5.4.1 Arquitetura

A Figura 5.25 mostra uma configuração típica com um segmento de cabo coaxial interligando várias estações de trabalho e uma impressora. O meio físico é um barramento serial ao qual as estações se conectam e o controle de acesso ao meio físico é distribuído entre as estações pois seu uso é compartilhado entre elas.

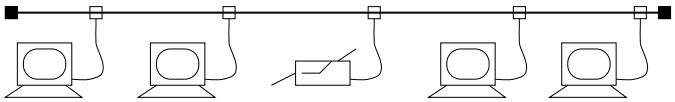


Figura 5.25: Rede local Ethernet.

A Figura 5.26 mostra a arquitetura de um adaptador de interface Ethernet. O transceptor acopla a estação ao meio físico, transmitindo os quadros gerados pela estação e recebendo os quadros injetados no cabo por todas as demais estações. Na transmissão, os dados NRZ-L internos a interface são transcodificados para Manchester (Seção 2.4). O acesso ao meio físico obedece ao método de acesso múltiplo por detecção de portadora e detecção de colisão ["Carrier Sense Multiple Access with Collision Detection" (CSMA/CD)]. Os dados a transmitir são concatenados ao cabeçalho do quadro e ao bloco de paridade e então injetados no meio físico.

Na recepção, o trajeto inverso é seguido. Os sinais recebidos são transcodificados para NRZ-L e o endereço do destinatário é examinado. Se o quadro é destinado à estação, o cabeçalho é removido e os dados são transferidos para a estação, se não foi detectado nenhum erro de transmissão. Tipicamente, a funcionalidade dos Níveis Físico e de Enlace (exceto transmissão e recepção) é implementada em um único circuito integrado, como as famílias LANCE e PC-net [AMD94].

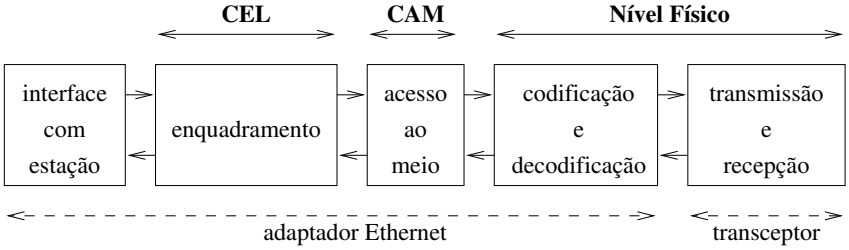


Figura 5.26: Arquitetura de um adaptador de interface Ethernet.

5.4.2 Nível Físico

A descrição que segue descreve os componentes do Nível Físico da Ethernet e é baseada em na versão para cabo coaxial fino, 10Base-2. A Figura 5.27 mostra a configuração básica dos dois tipos de interface Ethernet padrão 10Base-2. O meio de comunicação é um cabo coaxial tipo RG58. As estações se conectam a um segmento de cabo através de tomadas implementadas com conectores "T". Este conector é chamado de *interface dependente do meio* ("medium dependent interface"–MDI). O conector T é ligado ao transceptor ("physical medium attachement"–PMA).

Em algumas implementações, o transceptor é separado da estação, ligando-se a ela pela *interface de transceptor* ("attachment unit interface"–AUI) que consiste de um cabo com dois conectores de 15 pinos. O transceptor possui um conector macho e o DTE um conector fêmea; isso permite a ligação do transceptor diretamente ao ETD, sem um cabo AUI.

O transceptor se liga ao *circuito de sinalização* ("physical signalling"–PLS), responsável pela codificação e decodificação Manchester dos sinais. O conjunto interface dependente do meio e transceptor é chamado de *unidade de acoplamento ao meio* ("medium attachment unit"–MAU). Um tipo diferente de unidade de acoplamento ao meio é definido para cada tipo de meio físico. A interface de transceptor é a mesma para todos os meios físicos. Nas maioria das implementações mais recentes, o transceptor é diretamente acoplado ao DTE, sem que a interface de transceptor fique exposta.

O segmento de cabo deve ser cortado em dois pedaços para a ligação de uma tomada. Dois conectores BNC fêmea devem ser ligados às duas extremidades e o conector T inserido entre

os dois conectores BNC, de forma a re-ligar os dois segmentos resultantes do corte no cabo. As tomadas devem ser espaçadas de 0.5m, no mínimo. A instalação de tomadas para Ethernet com cabo grosso não implica no corte do cabo em dois pedaços. Para a instalação de uma tomada, o isolamento do cabo é perfurado e uma ponta de prova é inserida até o condutor no centro do cabo. O transceptor deve ser firmemente acoplado ao cabo para evitar que a ligação elétrica seja interrompida pela movimentação do cabo do transceptor. Com cabo grosso, as tomadas devem ser espaçadas de 2.5m, no mínimo.

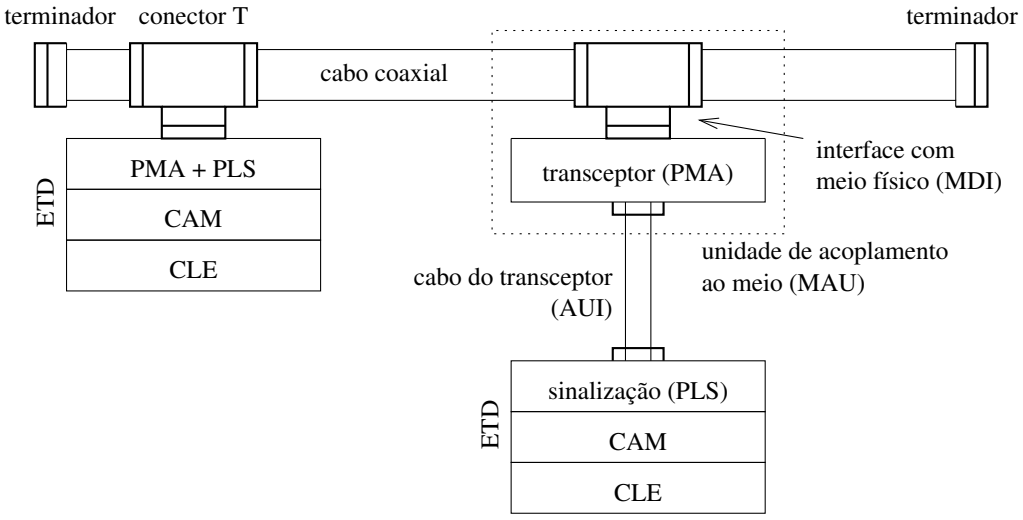


Figura 5.27: Componentes do Nível Físico da Ethernet.

Ethernet de 10, 100 e 1000 Mbps

O padrão IEEE 802.3u define os parâmetros para a taxa de transmissão de 100 Mbps, conhecido por 100Base-T Fast Ethernet. O método de acesso ao meio é o mesmo que na Ethernet de 10 Mbps mas o Nível Físico foi modificado para acomodar a velocidade 10 vezes maior. Outro padrão para transmissão a 100 Mbps, chamado 100VG-AnyLAN, utiliza o mesmo formato de quadros que a Ethernet mas o método de acesso ao meio é um mecanismo de *prioridade por demanda*. Esta rede local é definida pelo padrão IEEE 802.12. Dada a maior popularidade das redes 100Base-T, somente estas serão abordadas no que segue.

Transmissão a 10 Mbps O padrão IEEE 802.3 define quatro tipos básicos de meio físico: cabo coaxial fino, cabo coaxial grosso, par trançado e fibra óptica. A Tabela 5.1 mostra, para transmissões a 10Mbps, os parâmetros máximos e mínimos de distância e número de tomadas para redes com cabo grosso (10Base-5 = **10**Mbps, banda **base**, segmento de **5**00m), cabo fino (10Base-2, segmento de 185m), par trançado (10Base-T, "**T**wisted pair") e fibra óptica (10Base-F, "**F**iber").

Parâmetro	10Base-5	10Base-2	10Base-T	10Base-F
Comprimento do segmento (máx.)	500m	185m	100m	≤2Km
Separação entre estações (máx.)	1700m	755m	100m	≤2Km
Estações por segmento (máx.)	100	30	2	2
Espaçamento entre tomadas (mín.)	2.5m	0.5m	—	—
Cabo	Belden$^{\alpha}$	RG58	par tr.$^{\beta}$	FOIRL$^{\gamma}$

α Cabo Belden tipo 9880 (PVC amarelo) ou 89880 (Teflon laranjado).

β Cabo com dois pares trançados e dois conectores macho tipo RJ-45.

γ "Fiber-Optic Inter-Repeater Link".

Tabela 5.1: Parâmetros para transmissões a 10Mbps, padrão IEEE 802.3.

Um *segmento* consiste de um certo comprimento de cabo limitado por terminadores, repetidores, pontes ou roteadores (veja abaixo). Um *segmento multiponto* conecta duas ou mais estações através de cabo coaxial (10Base-5 e 10Base-2). Um *segmento ponto-a-ponto* conecta somente as duas estações nas extremidades do par trançado ou fibra óptica (10Base-T e 10Base-F). Um segmento de cabo coaxial deve conter um *terminador* em cada extremidade do cabo. A função do terminador é absorver os sinais que se propagam até as extremidades do cabo, evitando reflexões indesejáveis que podem interferir com os sinais injetados pelas estações. Um terminador é um resistor com valor igual ao da impedância característica do cabo (50Ω) ligado entre conector central e malha.

Transmissão a 100 Mbps O padrão Ethernet da 100 Mbps define três tipos de Fast Ethernet. O primeiro é 100Base-TX, para uso com par-trançado sem blindagem, 100Base-FX para uso com fibra óptica e 100Base-T4 para uso com par trançado de baixa qualidade, que usa quatro pares de fios. A Tabela 5.2 mostra os parâmetros físicos para transmissões a 100 Mbps.

Parâmetro	100Base-TX	100Base-FX	100Base-T4
Comprimento do segmento (máx.)	100m	2Km	100m
Estações por segmento (máx.)	2	2	2
Cabo	UTP-5$^{\alpha}$	F.O.	UTP-3$^{\beta}$

α Categoria 5, dois pares e conectores macho tipo RJ-45.
β Categoria 3, quatro pares e conectores macho tipo RJ-45.

Tabela 5.2: Parâmetros para transmissões a 100Mbps, padrão IEEE 802.3.

Transmissão a 1000 Mbps A quarta geração da Ethernet possibilitará transmissões a taxas de 1 Giga-bps, 100 vezes maior que a segunda geração, a 10Base-$*$. Os componentes da Gigabit Ethernet ainda não estão sacramentados em um padrão. Contudo, é muito provável que o padrão seja aprovado com a arquitetura descrita aqui. O Nível Físico toma emprestado do padrão *Fibre Channel* ANSI X3T11 os componentes ópticos (conectores e fibras –FC0) e codificação 8B/10B (FC1). O Nível de Enlace permanece basicamente aquele definido pelos padrões 802.3 –controle de acesso ao meio é por detecção de colisões (CSMA/CD)– e controle de enlace lógico 802.2 –veja [MKK97] para detalhes. A Tabela 5.3 mostra os parâmetros para transmissão a 1 Gbps.

Parâmetro	F.O.mono	F.O.multi	par tr.
Comprimento do segmento (máx.)	2-10Km	200-500m	100m
Codificação	8B/10B	8B/10B	?
Estações por segmento (máx.)	2	2	2
Cabo	780nm$^{\alpha}$	1300nm$^{\beta}$	par tr.

α Par de fibras monomodo com núcleo de 780nm.
β Par de fibras multimodo com núcleo de 1300nm.

Tabela 5.3: Parâmetros para transmissões a 1 Gbps, padrão IEEE 802.3z.

A interface entre o Subnível CAM e o Nível Físico se dá através da *Interface Independente do Meio*. A codificação 8B/10B consiste em transmitir 8 bits de informação para cada 10 bits inseridos no meio físico. Os dois bits adicionais são usados para transferir comandos e status e

sincronizar os equipamentos nas pontas do enlace. O Subnível CAM prevê operação nos modos duplex e semi-duplex.

O aumento de capacidade propiciado pela Gigabit Ethernet deve proporcionar alívio numa série de situações onde redes com menor capacidade enfrentam congestionamentos. Contudo, se esta tecnologia for empregada com adaptadores de rede "comuns", os pontos de estrangulamento migram da rede para o sistema de memória dos computadores. Mais detalhes no Capítulo 6.

Cabeamento – Par trançado Os esquemas de cabeamento para redes Ethernet mais difundidos são baseados em pares trançados sem blindagem ["unshielded twisted pair" (UTP)]. Além da facilidade de instalação e manutenção, este tipo de cabeamento apresenta vantagens com relação a vazão da rede, discutidas abaixo. Estes cabos são similares a cabos telefônicos e são classificados de acordo com sua faixa de passagem. O custo por metro é proporcional à categoria. A Tabela 5.4 mostra as categorias e suas faixas de passagem. Cabos de categoria 1 e 2 não são usados em redes do tipo 10Base-T.

Categoria	f. pass.
5	100 Mbps
4	20 Mbps
3	16 Mbps

Tabela 5.4: Categorização de pares trançados.

Os pares de fios nos cabos de categoria 5 são torcidos ao longo de todo o seu comprimento e tem impedância característica de 100Ω. Os cabos são terminados por dois conectores de oito pinos, tipo RJ45 macho, enquanto que os equipamentos possuem tomadas para estes conectores. A unidade de acoplamento ao meio (MAU) neste caso é o conector RJ45. Os dois fios em cada par são polarizados. Os dois fios que transportam dados a transmitir devem ser ligados aos pinos 1 e 2: DT+ ao pino 1 e DT− ao pino 2. Os fios que transportam dados a receber são ligados aos pinos 3 e 6: RD+ ao pino 3 e RD− ao pino 6. Os cabos usados em redes 100Base-TX tem comprimento limitado em 100m para atender às restrições de tempo de propagação. Em redes 10Base-T, o comprimento dos segmentos é limitado em 100m por causa da atenuação dos sinais.

Sinalização

A sinalização empregada na rede Ethernet é na banda básica e a codificação usada é Manchester, descrita na Seção 2.4.1. Na codificação Manchester, a primeira metade do intervalo de bit tem o valor do complemento do bit e a segunda metade do intervalo tem o valor do bit. Isso garante a ocorrência de uma transição no meio de cada intervalo de bit e facilita a sincronização entre os relógios de transmissor e receptor(es). Por outro lado, a taxa de sinalização é o dobro da taxa de transmissão.

Repetidores, Concentradores, Pontes, Comutadores e Roteadores

O padrão define o tamanho máximo de cada segmento de cabo, dependendo do meio físico. O padrão também define equipamentos que podem ser usados para ultrapassar os limites impostos para um único segmento. Um *repetidor* é um dispositivo de Nível Físico que contém dois pontos de conexão e interliga dois segmentos, fazendo o que seu nome sugere: todos os sinais recebidos por uma das entradas são transmitidos pela outra e vice-versa. Os sinais recebidos de um segmento são amplificados, os pulsos re-formatados, e possivelmente re-sincronizados, antes de serem transmitidos para o outro segmento. Por exemplo, para uma rede do tipo 10Base-2

(cabo fino), o limite de 185m pode ser ultrapassado com o uso de quatro repetidores para interligar cinco segmentos 10Base-2, conectando 150 estações, em uma extensão de 925m entre as estações topologicamente mais distantes. Usando fibra óptica, repetidores podem ser conectados por enlaces de até 2Km. Estes enlaces são chamados de 10Base-FL ou "Fiber-Optic Inter-repeater Link" (FOIRL). Note que a Ethernet só funciona em barramentos –uma rede que contiver um ciclo por onde os sinais possam circular não funciona, a não ser que os segmentos sejam interligados por pontes.

Um segmento ponto-a-ponto com par trançado ou fibra óptica conecta somente as duas estações ligadas nas suas extremidades. Estes segmentos são usados nas topologias em estrela onde várias estações se comunicam através do "centro" da estrela, chamado de *concentrador* ("hub"). Um concentrador é um dispositivo de Subnível CAM que se comporta como um segmento multiponto: cada quadro gerado por uma estação é propagado para todas as estações ligadas ao concentrador. A Figura 5.28 mostra uma rede local com duas sub-redes, três segmentos, um repetidor e uma ponte.

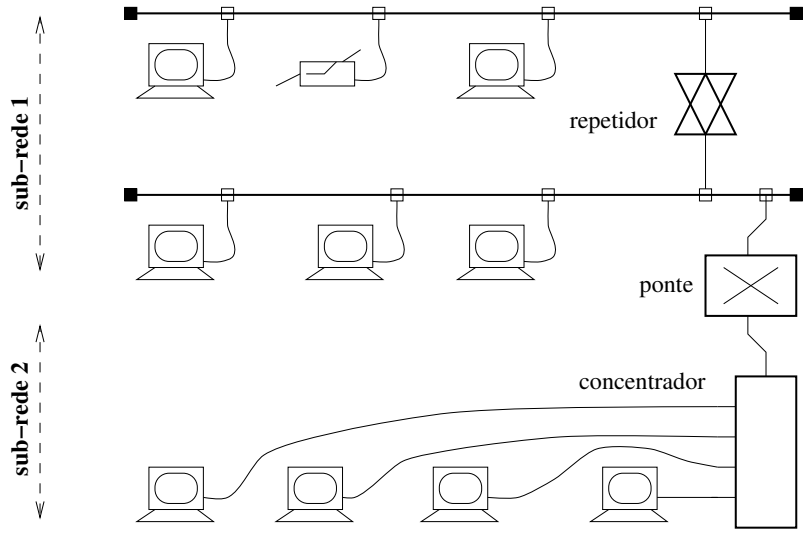


Figura 5.28: Rede local com três segmentos, repetidor e ponte.

Os equipamentos descritos acima servem para aumentar o diâmetro de uma rede local, aumentando a distância máxima possível entre as duas estações topologicamente mais distantes. O diâmetro não pode exceder os limites determinados no padrão para garantir que o tempo de propagação dos sinais através de toda a rede não exceda o mínimo necessário para a detecção de colisões (veja abaixo). As estações interligadas por segmentos compartilhados, concentradores, repetidores e pontes estão num mesmo *domínio de colisão* porque todas as suas transmissões estão sujeitas às regras de acesso ao meio físico.

Uma *ponte* é um dispositivo de Subnível CAM que permite a interligação de duas ou mais redes Ethernet, transformando-as em uma única rede local. Cada Ethernet que se conecta à ponte é um domínio de colisão diferente. A ponte permite a extensão do diâmetro da rede local porque, a cada uma de suas portas se liga uma rede independente cujo diâmetro e população podem estar próximos dos limites determinados pelo padrão. Uma ponte faz comutação de pacotes com redespacho, isto é, a ponte armazena em sua memória os quadros recebidas de cada uma de suas portas. Após verificar a integridade de um quadro a ponte retransmite-o pela(s) porta(s) adequada(s). As pontes também removem quadros danificados, evitando sua propagação para as sub-redes.

Uma ponte observa o endereço do remetente de cada quadro que recebe em suas portas para montar uma tabela que associa estações à sub-redes *i.e.,* a tabela contém pares <estação, porta>.

Quando a ponte recebe um quadro, se fonte e destinatário residem no mesmo segmento, o quadro não necessita "cruzar a ponte". Se a ponte recebe um quadro com endereço desconhecido, este sempre é transferido para o(s) outro(s) segmento(s) e um novo par <estação, porta> é adicionado à tabela. O uso de pontes para interligar as sub-redes de uma rede local tende a reduzir o tráfego entre sub-redes. Isso ocorre porque a ponte aumenta a localidade da comunicação quando evita a propagação desnecessária dos quadros com fonte e destino em um mesmo segmento. Enquanto reduz o tráfego entre as sub-redes, a ponte também permite a utilização concorrente dos segmentos que interliga. Isso aumenta a capacidade agregada da rede local porque, quando não há quadros "cruzando" a ponte, os segmentos operam de forma independente e em paralelo, como mostra a Figura 5.29.

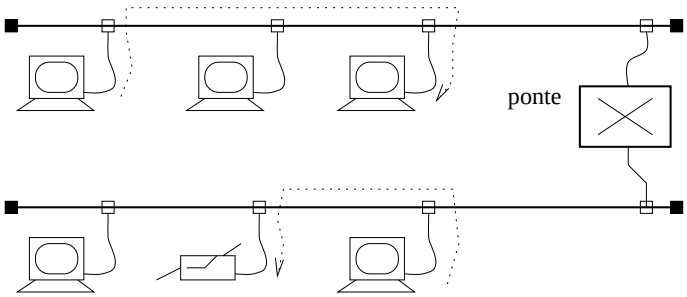


Figura 5.29: Aumento da capacidade agregada pelo paralelismo.

Um *comutador* é um dispositivo funcionalmente similar a uma ponte e é usado para interligar vários segmentos Ethernet, um segmento em cada uma de suas portas. A diferença entre um comutador e uma ponte é que comutadores podem interligar *simultaneamente* várias sub-redes. Comutadores podem fazer a comutação de pacotes por redespacho ou trespasse. No caso de trespasse, o comutador examina o quadro e assim que o endereço do destinatário for lido pelo comutador, este faz a decisão de roteamento e transfere o quadro para a porta adequada. O comutador também mantém uma tabela de endereços que relaciona o número de uma porta com todos os endereços das estações ligadas àquela sub-rede.

Roteadores são equipamentos mais sofisticados que pontes porque estes determinam a transferência de quadros de um segmento para outro em função dos endereços empregados pelos protocolos do Nível de Rede. O roteador examina o conteúdo do campo de informação de cada quadro Ethernet para então tomar a decisão de roteamento. Devido ao processamento adicional necessário para determinar os endereços de Nível de Rede, roteadores são mais lentos e mais caros do que comutadores. Em geral, roteadores são usados para ligar uma rede Ethernet a outra(s) rede(s) através de linhas seriais, redes locais ou dispersas.

A Figura 5.30 mostra a arquitetura destes dispositivos. Repetidores, concentradores e alguns comutadores contém somente funcionalidade de Nível Físico. Pontes e comutadores contém funcionalidade de Sub-nível CAM enquanto que roteadores são equipamentos que devem implementar funcionalidade de Nível de Rede.

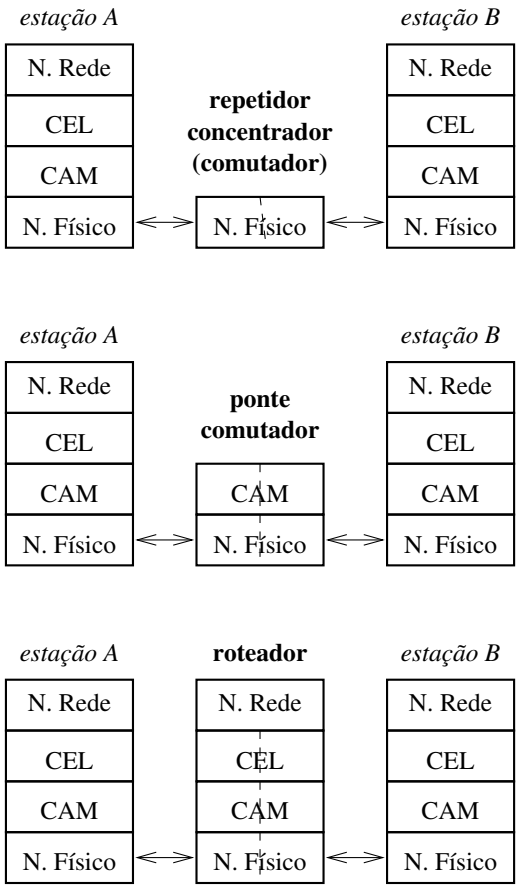


Figura 5.30: Repetidores, pontes, comutadores e roteadores.

5.4.3 Nível de Enlace – Controle de Acesso ao Meio

As estações interligadas por uma Ethernet devem cooperar no controle de acesso ao meio físico porque não existe uma única entidade que decide quem pode transmitir quando. O protocolo distribuído que é empregado para controlar o acesso ao meio é chamado de *Método de Acesso Múltiplo por Detecção de Portadora com Detecção de Colisões* ["Carrier Sense Multiple Access with Collision Detection" (CSMA/CD)]. O "acesso múltiplo" se refere ao fato de que qualquer estação pode tomar a iniciativa de transmitir. "Detecção de portadora" significa que, antes de transmitir, a estação deve sensorear o meio para "ouvir" se há portadora, e portanto, alguma outra estação está transmitindo. Se nenhuma estação está transmitindo, a linha está em *silêncio*.

Existe a possibilidade de que, após detectar a ausência de portadora, duas estações iniciem a transmissão de seus quadros ao mesmo tempo. O transceptor deve ser capaz de *detectar a colisão* e abortar a transmissão do quadro. Uma colisão entre os sinais de dois quadros no meio é similar ao choque de duas ondas que viajam em direções opostas na água. Ao contrário do que a intuição possa indicar, a ocorrência de colisões é absolutamente normal e esperada. Quanto maior o tráfego na rede, maior o número de colisões; o controle de acesso ao meio é projetado para operar corretamente nesta circunstância.

As estações devem operar segundo as regras listadas abaixo para garantir o correto funcionamento do protocolo CSMA/CD. Estas regras definem o comportamento de uma estação antes e durante transmissões e garantem justiça na resolução dos conflitos de acesso ao meio.

Escuta Antes de acessar o meio, cada estação deve "escutar" o cabo.

Deferência Se a estação detecta alguma transmissão em andamento, ela deve esperar até que haja silêncio novamente.

Colisão Ao detectar a colisão entre o quadro que está inserindo no meio com um quadro inserido por alguma outra estação, a interface deve abortar a transmissão do quadro e retransmiti-lo mais tarde.

Consenso Ao detectar colisão durante uma transmissão, a interface deve inserir quatro octetos no meio antes de abortar a transmissão para garantir que todas as outras estações detectem a colisão.

Transmissão O procedimento de transmissão de um bloco de dados se inicia com o recebimento de um comando originado pelo Subnível de Controle de Enlace Lógico. Este comando contém um apontador para os dados e o endereço do destinatário. A interface calcula o TDP, monta o quadro e escuta o cabo. Caso este esteja em silêncio, o quadro é codificado e transmitido. Enquanto injeta seu quadro, o transceptor compara os bits que recebe com os que transmite. Se estes forem idênticos não há colisão e o quadro é transmitido até seu último octeto. Se os bits recebidos forem diferentes dos transmitidos, ocorreu uma colisão e o transceptor transmite mais alguns octetos para garantir que as outras estações detectem a colisão. A transmissão é abortada e a interface deve esperar um intervalo adequado antes de tentar a retransmissão do quadro.

Recepção Ao detectar portadora, após um período de silêncio, o transceptor sincroniza seu circuito de recepção com o do transmissor com base nos octetos do preâmbulo. Se o campo destino do quadro recebido contiver o endereço da estação, a interface copia os octetos do quadro para um armazenador enquanto calcula o TDP. Se o quadro foi recebido sem erro, o conteúdo dos campos de informação, origem e tamanho são passados para o CEL. Se foram detectados erros, subnível CEL é notificado.

Transporte precário O protocolo de enlace da Ethernet *não garante* a entrega de quadros aos seus destinatários. A única garantia é de que o protocolo de acesso ao meio vai fazer *o melhor possível* para entregar os quadros, isto é, existe uma alta probabilidade de que quadros serão entregues aos destinatários. Os protocolos do Subnível CEL e níveis superiores devem garantir a entrega do conteúdo dos quadros aos seus destinatários. Os projetistas da Ethernet tomaram esta decisão para reduzir a complexidade do protocolo e de suas implementações.

Resolução de Colisões

A detecção de colisões pelo transceptor decorre da comparação do sinal transmitido com o sinal presente no meio *i.e.,* o transceptor escuta o meio físico ao mesmo tempo em que nele injeta sinal. Se o sinal proveniente do meio é diferente daquele sendo injetado ocorreu uma colisão. A Figura 5.31 mostra duas estações transmitindo ao mesmo tempo e a consequente colisão. As duas estações detectam silêncio no cabo e iniciam suas transmissões. Devido a separação entre ambas, a estação $\mathcal{B}$ não detecta a transmissão iniciada por $\mathcal{A}$ e inicia a inserção de um quadro no meio físico. Quando o sinal gerado por $\mathcal{A}$ se propagar até o transceptor de $\mathcal{B}$, este vai detectar a colisão, transmitir mais alguns octetos e abortar sua transmissão. Quando o sinal gerado por $\mathcal{B}$ se propagar até a estação $\mathcal{A}$, aquela vai detectar a colisão e também abortar sua transmissão.

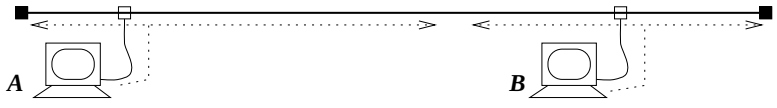


Figura 5.31: Colisão entre duas transmissões.

As características de transmissão dos cabos coaxiais usados nas versões 10Base-[5,2] são tais que, numa rede com comprimento máximo, os sinais injetados num extremo da rede se propagam até o outro extremo em tempo menor que 9.6μs. Este intervalo de 9.6μs é chamado de *hiato entre quadros* e garante que, uma vez concluída a transmissão de um quadro e decorrido o hiato, todas as outras estações na rede devem ter detectado silêncio. O hiato entre quadros é abreviado por t_{heq}. Quando uma estação inicia a transmissão de um quadro, se não ocorrer uma colisão durante um intervalo igual ao hiato, a estação *capturou* o meio físico e poderá transmitir o quadro até o último bit. A ocorrência de uma colisão após a captura do meio indica que há alguma estação defeituosa na rede.

O padrão Ethernet permite o uso de endereços com 16 ou 48 bits. Se forem usados endereços com 16 bits, um quadro de tamanho mínimo tem 12 octetos no cabeçalho, 48 octetos de dados e 4 octetos de TDP, perfazendo 64 octetos. Nesse caso, a uma velocidade de transmissão de 10 Mbps, cada bit dura 100ns e o tempo de transmissão de um quadro de 64 octetos é 51.2μs. Caso o campo de dados contenha menos de 48 octetos, a estação deve enxertar octetos suficientes até perfazer os 48 prescritos. O intervalo de 51.2μs é a unidade de tempo E do algoritmo de resolução de colisões, descrito abaixo. Em redes 100Base-[TX,FX,T4], cada bit dura 10ns e o tempo de transmissão do quadro de tamanho mínimo é $E = 5.12\mu$s.

A Figura 5.32 mostra a relação entre o tempo de propagação pelo meio físico e o tamanho mínimo de um quadro, para o pior caso onde as duas estações estão nos extremos opostos de uma Ethernet [Sta91]. A estação $\mathcal{A}$ inicia sua transmissão no instante t. O tempo de propagação dos sinais entre as estações $\mathcal{A}$ e $\mathcal{B}$ é p. Em $t + p - \mu$, para μ pequeno, a estação $\mathcal{B}$ inicia sua transmissão. Quando o primeiro bit do quadro de $\mathcal{A}$ chega ao transceptor de $\mathcal{B}$, em $t + p$, $\mathcal{B}$ detecta a colisão e aborta sua transmissão. A estação $\mathcal{A}$ continua com sua transmissão até que a interferência resultante da colisão se propague de volta, em $t + 2p$. Se o tempo de transmissão de um quadro for menor que $2p$, a interferência causada por uma colisão somente vai ser detectada pela estação transmissora *após* a inserção do último bit do quadro. Assim, a estação que transmitiu o quadro não toma conhecimento da colisão até que seja muito tarde para abortar sua transmissão. Durante uma transmissão, se o transceptor detectar uma colisão, o transmissor emite mais 32 bits de *interferência* para garantir que todas as estações percebam a colisão. A unidade de tempo de espera para retransmissões E deve ser maior que a soma do tempo de propagação no meio de ida-e-volta, e a duração dos 32 bits de interferência.

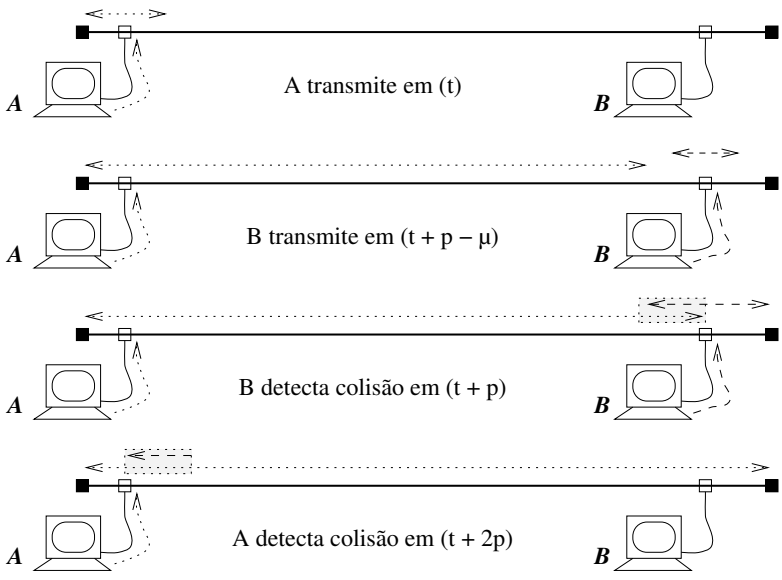


Figura 5.32: Relação entre tempo de propagação t_p e tamanho dos quadros.

Intervalo de Retração Exponencial Binário

Quando uma estação detecta uma colisão no quadro que está transmitindo, a transmissão é abortada e nova tentativa é feita "mais tarde". Se o tempo de espera para a retransmissão for fixo, há grande probabilidade de as colisões se repetirem. Para evitar que colisões se repitam indefinidamente, as estações esperam por um certo tempo, que é escolhido a esmo dentre uma faixa de valores, na esperança de que as outras estações que participam da colisão escolham intervalos de espera distintos. Se as colisões se repetem, o tamanho do conjunto de valores aumenta a cada nova colisão. Os tempos de espera para retransmissão são calculados usando o algoritmo *Retração em Intervalo Exponencial Binário* ("Truncated Binary Exponential Backoff"). Este algoritmo resolve conflitos entre até 1024 estações de uma forma relativamente justa.

O tempo de espera E é um múltiplo do tempo de duração de um quadro de tamanho mínimo com 64 octetos $t_{64} = 51.2\mu$s ($t_{64} = 5.12\mu$s em redes 100Base-[TX,FX,T4]) e para a n-ésima tentativa, E é escolhido dentre o intervalo $I = [0, (2^n - 1)]$ para $0 \leq n \leq 10$. A primeira tentativa de transmissão corresponde $n = 0$. Para $11 \leq n \leq 15$, o intervalo é truncado e permanece [0, 1024]. Quando $n = 16$, a estação é informada da impossibilidade de transmissão.

Ao detectar silêncio, a estação inicia a transmissão. Se ocorre uma colisão, cada uma das estações envolvidas escolhe a esmo um tempo de espera dentre $I = [0, 1]$, isto é, cada estação espera um tempo de 0 ou t_{64} segundos. Se ocorrer nova colisão, o intervalo de escolha passa a ser $I = [0, 2]$ e as estações esperam 0, t_{64} ou $2t_{64}$ segundos antes da nova tentativa. Se ainda ocorrerem outras colisões, a cada vez I aumenta para [0, 4], [0, 8], [0, 16], e assim por diante. Após um número pequeno de colisões repetidas, os valores de tempo de espera escolhidos a esmo se espalham e as chances de novas colisões diminuem. A Tabela 5.5 mostra os intervalos de escolha para colisões repetidas. A estação que escolher o menor tempo transmite seu quadro e reinicializa o intervalo de escolha em [0, 0]. As demais estações expandem seus intervalos de acordo com o número de tentativas frustradas.

tentativa	colisões	intervalo
1^a	–	$[0, 0]$
2^a	1^a	$[0, 1]$
3^a	2^a	$[0, 2]$
4^a	3^a	$[0, 4]$
5^a	4^a	$[0, 8]$
...	...	...
9^a	8^a	$[0, 512]$
10^a	9^a	$[0, 1024]$
11^a	10^a	$[0, 1024]$
...	...	...
15^a	14^a	$[0, 1024]$
16^a	15^a	pânico

Tabela 5.5: Retração em Intervalo Exponencial Binário.

O método CSMA/CD possibilita a ocorrência do *efeito de sequestro* de meio físico por uma estação. Suponha que a estação $\mathcal{E}$ é a vencedora da rodada de contenção, transmite seu quadro e reinicializa seu contador de tentativas em zero ([0, 0]). Se $\mathcal{E}$ tiver outro quadro esperando para ser transmitido, ela vai entrar novamente na competição pelo acesso ao meio físico. Como seu contador de tentativas vai ser menor que o das outras estações, a estação $\mathcal{E}$ tem uma chance

maior de transmitir seu quadro. Enquanto tiver quadros a transmitir, $\mathcal{E}$ pode capturar o meio físico e (com probabilidade maior que zero) impedir outras estações de transmitir seus quadros.

A utilização de uma Ethernet deve ser baixa para que o número de colisões seja relativamente pequeno. Em condições de pouco tráfego, o lapso de tempo entre a requisição de envio de uma mensagem pela estação e a transmissão da mensagem é pequeno. Veja [BMK88] para uma discussão sobre a eficiência de Ethernets. A distribuição espacial das estações influi na capacidade efetiva de canal disponível para cada estação individual. Uma análise deste efeito e alterações no protocolo acima para atenuá-lo estão em [GT88].

Detecção de Colisões em Concentradores

O método de detecção de colisões em redes com concentradores e cabeamento em par trançado é mais simples que em redes com cabo coaxial. O concentrador gera um sinal de *presença de colisão* em todas as suas saídas sempre que detectar a presença de sinal em mais de uma entrada. Considere a rede da Figura 5.33. Quando a estação $\mathcal{B}$ transmite, o concentrador 1 repete os sinais na entrada 2 em todas as saídas; o concentrador 2 também repete os sinais recebidos do concentrador 1 através da sua entrada 5. Quando as estações $\mathcal{B}$ e $\mathcal{C}$ transmitem simultaneamente, o concentrador 1 gera o sinal de presença de colisão em todas as saídas. Este sinal também é repetido pelo concentrador 2. Se as estações $\mathcal{B}$ e $\mathcal{H}$ transmitirem ao mesmo tempo, ao detectar a colisão, os dois concentradores emitem o sinal de presença de colisão.

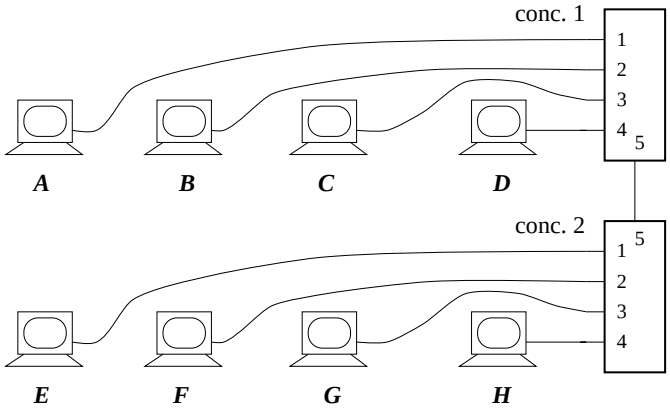


Figura 5.33: Colisões em redes com par trançado.

5.4.4 Nível de Enlace – Controle de Enlace Lógico

O Subnível de Controle de Enlace Lógico da rede Ethernet é responsável pelo empacotamento dos mensagens recebidas dos PAS do Nível de Rede e Subnível CAM. A Figura 5.34 mostra o formato dos quadros usados na Ethernet. O preâmbulo, com 64 bits, consiste da sequência $(10101010)^7 10101011$ e serve para sincronizar transmissor e receptores. O par de bits 11, no oitavo octeto, demarca o início do quadro propriamente dito. O endereço do destino é especificado por um número de 48 bits. O campo origem contém o endereço da estação que emitiu o quadro. O conteúdo do campo tamanho indica o número de octetos no campo de informação; se o conteúdo deste campo for maior que 1500, então o número é usado para identificar o protocolo de alto nível transportado pelo quadro. O campo de informação contém no mínimo 46 e no máximo 1500 octetos. O tamanho mínimo (18+46) é necessário para distinguir um quadro válido de fragmentos resultantes de colisões. Os quatro últimos octetos contém o TDP-32, que é usado para a detecção de erros.

O endereço de cada adaptador de interface Ethernet é único e absoluto, chamado de *endereço físico* ou *endereço CAM*. Em princípio, cada adaptador de interface é fabricado com um endereço

pré-estabelecido e gravado em ROM. O IEEE controla o espaço de endereçamento Ethernet e distribui faixas de endereços aos fabricantes. Cada faixa consiste de um identificador de 24 bits, chamado "Organizationally Unique Identifier" (OUI). Cada fabricante adquire um ou mais OUIs e produz interfaces de rede cujos endereços são compostos do seu OUI concatenado com um número de 24 bits que identifica a interface.

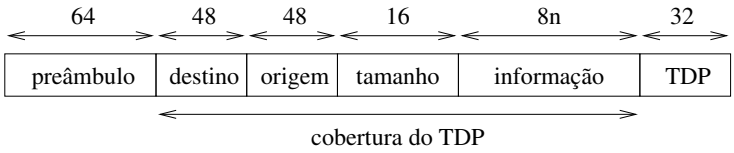


Figura 5.34: Formato dos quadros IEEE 802.3.

Duas classes de endereços são usadas, uma para estações individuais e a outra para grupos de estações. Se o primeiro bit do campo destino é 0, o número é o endereço de uma estação; se o primeiro bit é 1, o número especifica o endereço de um grupo lógico de estações. Se todos os bits forem 1, o quadro é destinado a todas as estações na rede. Os 47 bits destinados a endereços individuais permitem a indentificação de $2^{47} \approx 7 \cdot 10^{13}$ estações.

5.4.5 Limites no Diâmetro de uma Rede Ethernet

O padrão IEEE 802.3 estabelece limites para as possíveis combinações de tipos de meio físico e de repetidores, pontes, *etc* que podem ser usados para interligar vários segmentos. A finalidade destes limites é garantir que o tempo de propagação de um quadro (ida-e-volta) seja menor que o tempo de inserção do menor quadro (64 octetos).

Um *domínio de colisão* é definido formalmente como *uma única rede CSMA/CD onde ocorre uma colisão se duas estações transmitirem simultaneamente.* Se o diâmetro da rede for menor ou igual ao máximo especificado para os diversos tipos de meios físicos e equipamentos de interconexão, não existe a possibilidade de duas estações transmitirem quadros pequenos e uma não perceber a transmissão da outra, bem como a colisão entre os dois quadros.

Além do tempo de propagação, os limites no número de dispositivos repetidores servem para garantir o espaçamento mínimo entre dois quadros consecutivos. Este espaçamento é necessário para que os circuitos das interfaces consigam absorver sem perdas todos os quadros transmitidos. Os circuitos de entrada dos repetidores re-condicionam os sinais recebidos antes da retransmissão. Dependendo das características de carga elétrica no meio físico, os primeiros bits do preâmbulo podem ser distorcidos. Isso pode causar a inserção ou remoção de alguns bits no quadro a ser retransmitido pelo repetidor. Os bits removidos alteram o espaçamento entre os quadros. Portanto, a fim de manter o espaçamento mínimo, o número de dispositivos repetidores deve ser limitado.

Os limites podem ser resumidos da seguinte forma. Pode haver, no máximo, cinco segmentos entre quaisquer duas estações numa rede Ethernet. Pode haver, no máximo, quatro repetidores entre quaisquer duas estações. Dos cinco segmentos, somente três podem ser multiponto. Esta regra é chamada de regra "5-4-3" – 5 segmentos, 4 repetidores e 3 segmentos multiponto. Além disso, os limites no número de estações por segmento e comprimento dos segmentos devem ser observados. Se os segmentos forem de tipos diferentes, estes limites são mais estritos.

5.4.6 Comutadores Ethernet

Um cabo coaxial é compartilhado por todas as estações por ele interligadas. Isso significa que só pode haver um quadro em trânsito pelo cabo a cada instante. O modo normal de transmissão na Ethernet é semi-duplex porque uma estação deve adquirir o direito de transmitir segundo o

método CSMA/CD. Isso, aliado à perda de tempo causada pelas colisões, reduz a vazão de dados através da rede. Por outro lado, uma topologia em estrela, onde estações com enlaces ponto-a-ponto são interligados por um comutador, permite que vários quadros estejam atravessando o comutador a cada instante, propiciando transmissões duplex simultâneas. Um desenvolvimento recente consiste em usar *comutadores* no lugar de concentradores. Um comutador cria, dinamicamente, ligações físicas entre pares de portas (estações ou segmentos) permitindo várias transmissões simultâneas entre estações ou segmentos. As estações em todas as portas do comutador podem ainda transmitir e receber quadros simultaneamente, dessa forma multiplicando a vazão agregada da rede.

A Figura 5.35 mostra um comutador com 4 entradas e 4 saídas. Para simplificar a descrição, a cada porta está ligada uma estação; na prática, cada porta do comutador fica ligada a um segmento com uma ou mais estações. No comutador da Figura 5.35, cada uma das quatro entradas pode se conectar a cada uma das quatro saídas. Num dado instante, o comutador pode interligar quatro pares de estações. Ocorre um *conflito* quando as estações em duas entradas solicitam conexão com a mesma saída. Neste caso, o comutador completa uma das duas conexões e envia o sinal de presença de colisão para a estação que não foi conectada. As linhas espessas na Figura 5.35 indicam conexões simultâneas onde as estações $\mathcal{C}$ e $\mathcal{D}$ estão transmitindo para as estações $\mathcal{D}$ e $\mathcal{B}$, respectivamente. Como a comunicação é duplex, a estação $\mathcal{D}$ transmite um quadro para $\mathcal{B}$ enquanto recebe outro quadro de $\mathcal{C}$. O comutador cria um segmento virtual entre cada par de estações que são interligadas, cada segmento virtual com uma capacidade de 10 (ou 100) Mbps entre as duas estações enquanto o segmento virtual existir. Na ausência de conflitos, o comutador da Figura 5.35 possui uma capacidade agregada de 40 (ou 400) Mbps.

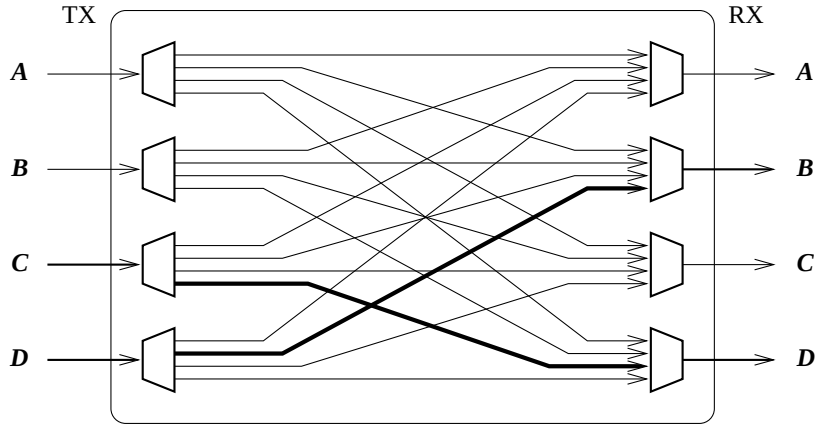


Figura 5.35: Comutador 4x4.

O comutador deve ser capaz de escolher uma rota de saída para cada quadro apresentado às suas entradas. Como a rota é uma função do endereço do destinatário, o comutador só necessita examinar os primeiros 48 bits de um quadro para determinar a posição dos seletores de entrada e saída. Este método de roteamento é chamado de *trespasse* porque somente os primeiros bits de um quadro são necessários para se escolher o caminho pelo qual o quadro vai "atravessar" o comutador [Dal90]. Uma vez que o caminho foi determinado, o quadro é transferido diretamente da entrada para a saída. A alternativa consiste em copiar todo o quadro para um armazenador e só então escolher a rota de saída. Este método é chamado de *redespacho* e é ineficiente tanto em termos de tempo quanto de espaço.

Além do aumento da vazão, possibilitado por várias conexões simultâneas, pontes e comutadores estão sendo usados com maior freqüência por razões que tem mais a ver com facilidade de instalação e manutenção do que capacidade de transporte de bits. Quando um cabo coaxial é rompido acidentalmente, todas as máquinas do segmento ficam impedidas de acessar a

rede. Se ocorre uma pane no acoplamento de uma estação ao cabo, é muito provável que todas estações ligadas pelo cabo percam acesso à rede. Ao contrário de um cabo coaxial, cada estação se liga à ponte ou comutador por um cabo que interliga somente estação e ponte/comutador. Caso ocorra alguma pane ou interrupção física do cabo, somente a estação ligada ao cabo é afetada. Todas as demais podem continuar operando normalmente, a menos da funcionalidade perdida na estação que foi desconectada da ponte/comutador.

A grande vantagem do uso de comutadores advém do particionamento dinâmico da rede em várias sub-redes em função do tráfego entre as sub-redes. A Figura 5.36 mostra uma rede composta de três subredes, interligadas em árvore por um comutador. Cinco servidores estão ligadas diretamente ao comutador porque eles são acessadas freqüentemente por todas as máquinas na rede. Quando um cliente numa sub-rede envia um quadro para um servidor, duas outras ligações entre clientes nas outras sub-redes e servidores podem estar ocorrendo simultaneamente. Neste exemplo, a vazão agregada da rede na direção dos servidores é o triplo do que seria numa rede sem o comutador.

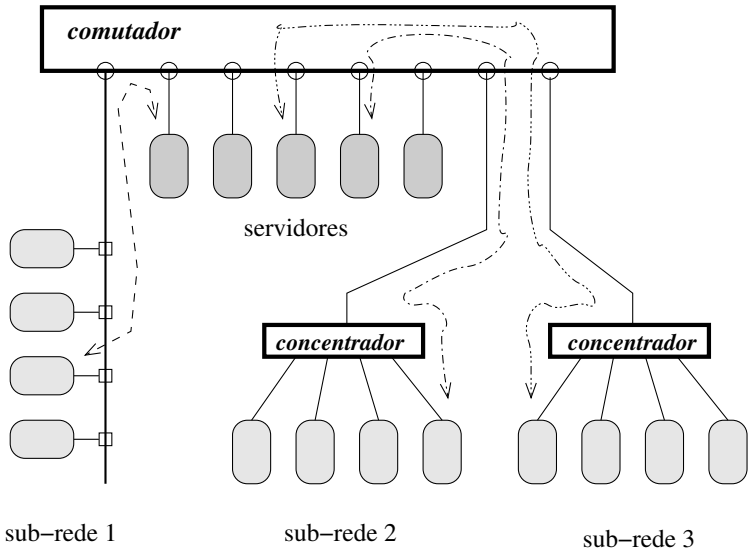


Figura 5.36: Estruturação em árvore de uma Ethernet.

5.5 Fiber Distributed Data Interface

A rede local "Fiber Distributed Data Interface" (FDDI) permite a transmissão de dados a uma taxa de 100Mbps entre estações ligadas em anel. Esta velocidade corresponde aproximandamente a transmissão de 2500 páginas de texto por segundo. Algumas aplicações demandam maior faixa de passagem do que os 10Mbps oferecidos por Ethernet 10Base∗, como a transmissão de voz e imagem em tempo real, além de garantias de entrega de quadros dentro de limites estritos de tempo. Além disso, fica evidente a necessidade de redes locais de alta velocidade, quando se considera a disseminação do uso de sistemas distribuídos e as taxas de crescimento de capacidade de processamento dos microprocessadores (25–50% ao ano) e estações de trabalho (20–30% ao ano) [HP96].

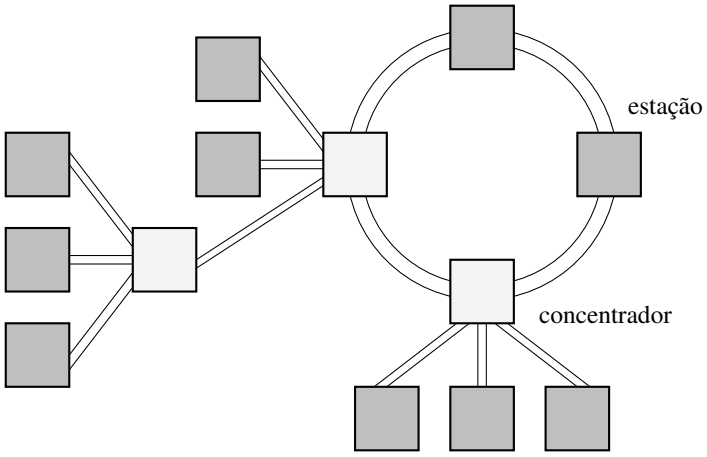


Figura 5.37: Topologia típica de redes FDDI.

O padrão FDDI foi publicado originalmente pela American National Standards Institute (ANSI), Accredited Standards Committee Task Group X3T9 [ANS87, ANS88, ANS90] e adotado pela ISO [ISO89a, ISO89b, ISO90]. As estações em uma rede FDDI são interligadas em uma topologia de anel de árvores, como mostra a Figura 5.37. Os dois anéis transportam sinais em direções opostas *i.e.,* sentidos horário e anti-horário. O meio físico empregado é, geralmente, fibra óptica, com um ou dois pares de fibras ligando as estações à rede.

Como o 'Distributed' do nome indica, os algoritmos de inicialização do anel e controle de acesso ao meio são distribuídos, e foram projetados de forma a aumentar a confiabilidade da rede. Uma rede FDDI pode interligar até 500 estações; as estações mais distantes na rede podem estar a 100 Km uma da outra e dependendo do tipo de fibra usado, estações adjacentes podem ser ligadas por cabos de até 2 Km ou até 60 Km. Cada quadro FDDI pode transportar pouco menos de 4500 octetos de dados. As três aplicações de redes FDDI são (1) como espinha dorsal de uma rede intra-campus onde o anel FDDI conecta redes locais em vários prédios (2) como uma rede local de alta velocidade interligando estações de trabalho diretamente, e (3) como uma interface de alta velocidade ligando periféricos rápidos a um computador de grande porte, como controladores de discos ou terminais gráficos de alta resolução.

O método de acesso ao meio físico é a passagem de bastão temporizada ("timed token passing"). O FDDI suporta duas classes de serviço de transporte de dados. Em aplicações que dependem de respostas dentro de limites estritos de tempo, como o transporte de voz e vídeo em tempo real, a comunicação entre as estações deve usar a *classe de serviço síncrono*, que garante uma certa fração da faixa de passagem para as estações que estão engajadas em comunicação síncrona. A *classe de serviço assíncrono* é usada para transporte de dados sem restrições temporais, tais como correio eletrônico ou transferência de arquivos. Nestas aplicações, não há limites rígidos no prazo para a entrega dos quadros. Note que uma mesma estação pode transmitir quadros através dos serviços síncrono e assíncrono concomitantemente, dependendo dos aplicativos que estão executando num dado momento.

Tolerância a falhas Na Ethernet, o meio físico pode ser considerado como um componente 'isolado' da rede porque as interfaces de rede não fazem parte do meio propriamente dito. Se uma estação sofre uma pane, há uma grande probabilidade de que a rede continue operando normalmente. Numa rede em anel, as interfaces de rede são parte integrante do anel pois os sinais no meio físico atravessam os circuitos da interface. Portanto, um ponto fraco da topologia em anel é a baixa tolerância a falhas nos equipamentos e enlaces que fazem parte do anel. A saída de uma estação se conecta à entrada da seguinte através de um cabo de cobre ou fibra óptica. Se uma das estações sofre uma pane, todo o tráfego pelo anel é interrompido. Da mesma forma,

se um segmento de fio/fibra é seccionado, o tráfego também se interrompe.

Para minimizar estes problemas, o padrão FDDI determina que a ligação das estações ao meio físico possa ser desviada de modo a desconectar fisica- e/ou logicamente uma estação da rede. Além disso, o uso de um anel duplo permite que um segmento seccionado possa ser desviado de forma a que o anel se reestabeleça. Contudo, se um segundo segmento for seccionado, dependendo da configuração, a rede pode ficar dividida em duas subredes. A Figura 5.38 mostra as duas situações onde as ligações entre estações são alteradas para reestabelecer o funcionamento da rede. As técnicas usadas na FDDI permitem que uma pane seja detectada e automaticamente contornada em poucos milisegundos.

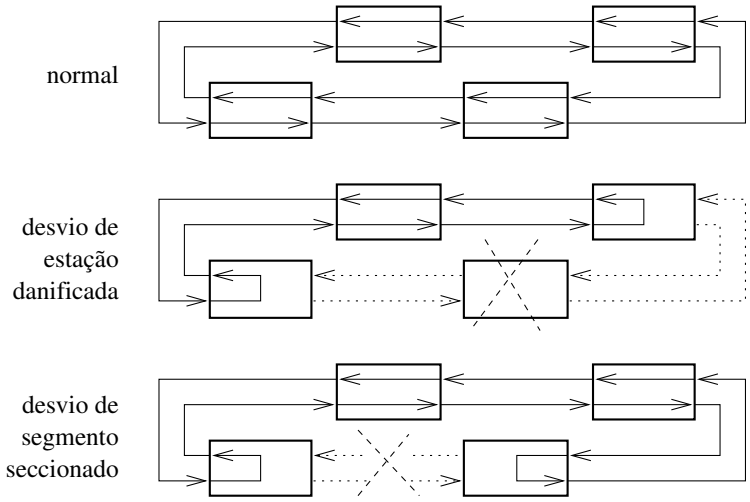


Figura 5.38: Reconfiguração em presença de falhas.

Quando um anel é reconfigurado dinamicamente por causa de uma pane, as estações no anel que resulta da reconfiguração passam a fazer parte do caminho crítico que interliga todas as estações. Isso significa que qualquer outra pane necessariamente vai interromper o tráfego através da rede. Enquanto que o uso de anéis duplos melhora a tolerância a *uma* falha no anel, uma vez que a primeira falha tenha ocorrido, a confiabilidade do anel cai drasticamente.

5.5.1 Arquitetura

A Figura 5.39 mostra os protocolos FDDI de Nível de Enlace e Nível Físico e sua relação com modelo estratificado ISO/OSI. Na FDDI, o Nível de Enlace é dividido nos subníveis de *Controle de Acesso ao Meio* (CAM) e *Controle de Enlace Lógico* (CEL). O subnível CAM define os procedimentos de controle de acesso ao meio físico determinando qual estação tem direito de transmitir, quando e por quanto tempo. O subnível CEL define o controle de enlace lógico e obedece o padrão IEEE 802.2.

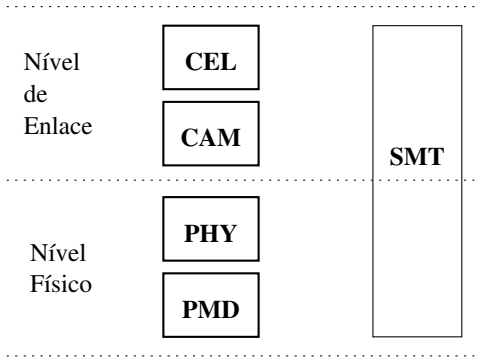


Figura 5.39: Relação entre modelo OSI e FDDI.

O Nível Físico da FDDI também é dividido em dois subníveis. O nível PHY ("PHYsical") define os componentes de nível físico que são independentes do meio físico empregado *i.e.,* no nível PHY são definidos a codificação dos bits inseridos no meio físico, detecção de erros e sincronização de relógio entre estações vizinhas, por exemplo. O nível PMD ("Physical Medium Dependent") especifica os componentes ópticos, características dos pulsos transmitidos pelas fibras, e os diferentes tipos de meio físico empregado, se fibra óptica ou cabos de cobre. O subnível PMD é o único que muda com os diferentes tipos de meio físico que podem ser usados em redes FDDI. Os protocolos SMT ("STation Management") especificam os procedimentos de *gerenciamento das estações* tais como as seqüências de quadros usadas para a localização automática de problemas e a conseqüente reconfiguração da rede.

5.5.2 Nível Físico

O padrão FDDI divide o Nível Físico em dois subníveis, PHY e PMD. Os protocolos do subnível PHY definem os elementos que são independentes da implementação do meio físico, como por exemplo a codificação dos sinais *i.e.,* o mapeamento entre os bits transportados pelos quadros e os símbolos que os representam na fibra óptica. O subnível PMD define os componentes do meio físico como o tipo de fibra óptica, conectores, transmissores e receptores.

Nível Físico – PHY

Os protocolos do subnível PHY definem a codificação dos sinais injetados no meio físico de forma a transferir dados transparentemente entre estações vizinhas bem como sinais de controle e status. A codificação ocorre em duas fases. Na primeira fase, o conteúdo do campo de dados é dividido em quartetos e cada quarteto é representado por 5 bits. Os bits dos quartetos são chamados de *bits da dados* e os bits dos quintetos de *bits de código*. A Tabela 5.6 mostra a codificação dos bits de dados em bits de código definidos pelo subnível PHY. Esta codificação é chamada de 4b/5b ou *quatro em cinco* porque quatro bits de dados produzem cinco bits de código. Os símbolos de dados foram escolhidos de forma a que não ocorram mais de um `0` no início de um símbolo nem mais de dois `0`s no final de um símbolo. Isso garante que o campo de informação dos quadros FDDI nunca contém seqüências com mais de três `0`s.

Das 32 seqüências possíveis de se representar em 5 bits, somente 16 são usadas para representar dados. Das outras 16, 9 são usadas para representar informação de controle. As outras 7 são inválidas e não devem nunca ser produzidas pelos transmissores porque elas violam as regras de seqüenciamento de `1`s e `0`s. As seqüências de bits são escolhidas de forma a facilitar a detecção de erros e a explicitar o posicionamento dos campos nos quadros.

dados			símbolos de controle				
código		dado	código		valor	nome	significado
30	11110	0x0	00	00000	Q	quiet	estado: silêncio
09	01001	0x1	31	11111	I	idle	estado: inativo
20	10100	0x2	04	00100	H	halt	estado: suspenso
21	10101	0x3	24	11000	J	DI	delimitador de início
10	01010	0x4	17	10001	K	DI	delimitador de início
11	01011	0x5	05	00101	L	DI	delimitador de início
14	01110	0x6	13	01101	T	DF	delimitador de final
15	01111	0x7	07	00111	R	reset	resultado: falso
18	10010	0x8	25	11001	S	set	resultado: verdadeiro
19	10011	0x9	01	00001	V		violação de código
22	10110	0xA	02	00010	V		violação de código
23	10111	0xB	03	00011	V		violação de código
26	11010	0xC	06	00110	V		violação de código
27	11011	0xD	08	01000	V		violação de código
28	11100	0xE	12	01100	V		violação de código
29	11101	0xF	16	10000	V		violação de código

Tabela 5.6: Codificação dos símbolos no subnível PHY.

Na segunda fase da codificação, os bits de código são codificados em NRZ-I ("Non Return to Zero, Invert on Ones", Seção 2.4.1). Nesta codificação, a forma de onda resultante apresenta uma inverão de nível sempre que ocorre um bit de código em 1. O nível permanece constante para bits de código em 0. Com a combinação da codificação 4b/5b e NRZ-I, o sinal transmitido pelo meio físico apresenta um número grande de transições, o que facilita a recuperação da informação de sincronismo embutida no sinal transmitido. A escolha da codificação 4b/5b, ao invés da codificação Manchester, se deve à menor taxa de sinalização possível com aquela. Com codificação 4b/5b, a taxa de sinalização da FDDI é 125Mbaud. Se a codificação fosse Manchester, a taxa de sinalização seria 200Mbaud.

Além da codificação, o subnível PHY ainda contém dispositivos para evitar que pequenas diferenças na frequência de relógio de duas estações adjacentes provoquem a remoção ou criação de bits. Um *registrador de deslocamento elástico* em cada estação acomoda diferenças da ordem de alguns bits nos relógios das estações transmissoras e receptoras.

Nível Físico – PMD

O padrão FDDI define três protocolos do subnível físico PMD para fibra óptica e um para par trançado. No caso de fibra óptica, o padrão define as características opto-elétricas de sistemas baseados em fibras multimodal com comprimento de onda de 1300nm e distância máxima entre dois nodos menor ou igual a 2Km. A versão para fibra monomodal permite a ligação de nodos através de fibras com até 60Km de comprimento. A versão para fibras de baixo custo permite ligações a distâncias de até 500m. A versão para par trançado permite ligações com até 100m de comprimento a um custo muito mais baixo do que ligações com fibra. Para detalhes veja [Jai94, Gre91].

5.5.3 Nível de Enlace – Controle de Acesso ao Meio

O método de acesso ao meio físico empregado na rede FDDI é *passagem de bastão temporizada*. Esse método de acesso é similar a uma corrida com revezamento 4 por 100. A cada estágio da corrida, um corredor leva o bastão de sua equipe e no final do estágio, o bastão troca de mãos

e outro atleta corre no próximo estágio. De forma similar, uma estação somente transmite seus quadros quando ela está de posse do bastão. O *bastão* é um quadro de formato especial cuja recepção indica permissão para transmitir. Ao terminar suas transmissões, a estação entrega o bastão para sua sucessora no anel. Isso se repete em todas as estações ao longo do anel. Para evitar que uma estação monopolize o uso do anel, o tempo de transmissão é limitado: uma estação pode transmitir enquanto o seu intervalo de transmissão não expirar. Quando o intervalo expira, a estação deve entregar o bastão para sua sucessora.

Para garantir que as estações com enlace(s) baseados em serviço síncrono recebam o bastão dentro de intervalos previsíveis, o *intervalo de rotação do bastão* (IRB) deve ser determinado pelo administrador da rede. O lapso de tempo em que cada uma das N estações na rede pode reter o bastão é de aproximadamente IRB/N segundos. Esta questão é discutida com mais detalhe adiante.

A responsabilidade pela remoção dos quadros inseridos na rede é da estação que os inseriu. Ao detectar um quadro cujo campo fonte é o seu próprio endereço, a estação remove o quadro do anel e insere um número suficiente de símbolos idle no lugar dos símbolos do quadro removido. Quadros que não foram inseridos por uma estação são transferidos para a estação sucessora no anel. Quadros cujo campo destino é o endereço da estação devem ser copiados pela destinatária e passados adiante. Após transmitir o último símbolo do último quadro, a estação transmite imediatamente o bastão. A próxima estação no anel pode inserir seus quadros assim que a anterior encerrar suas transmissões. A liberação do bastão imediatamente após o último quadro permite a máxima utilização do anel.

O controle de acesso ao meio garante, ao longo do tempo, oportunidades iguais de transmissão para todas as estações no anel. Para garantir que todas estações possam transmitir seus quadros dentro de intervalos razoáveis, um crédito de transmissão é alocado a cada estação. Se a estação $\mathcal{E}_\mathrm{i}$ não tem o que transmitir, o bastão é passado adiante para $\mathcal{E}_\mathrm{j}$. O tempo não utilizado pela estação $\mathcal{E}_\mathrm{i}$ fica como crédito adicional para $\mathcal{E}_\mathrm{j}$. É possível que o crédito da estação $\mathcal{E}_\mathrm{i}$ se esgote logo após o início da transmissão de um quadro longo. Para garantir o tempo de rotação do bastão ao longo do anel, o tempo adicional usado por $\mathcal{E}_\mathrm{i}$ é debitado de $\mathcal{E}_\mathrm{j}$, resultando num intervalo de transmissão menor para $\mathcal{E}_\mathrm{j}$. O algoritmo de alocação de créditos garante que, na próxima passagem do bastão, $\mathcal{E}_\mathrm{j}$ terá o seu crédito integral de transmissão. Este algoritmo é descrito nos próximos parágrafos. Note que, se o crédito de transmissão se esgota após o início da inserção de um quadro, a inserção não é interrompida e o quadro é transmitido integralmente.

O *tempo de rotação do bastão* (TRB) é o intervalo que decorre entre duas chegadas sucessivas do bastão a uma mesma estação. O TRB é proporcional ao número de estações ao longo do anel e do comprimento do caminho ao longo do anel. Cada estação pode transmitir seus quadros e reter o bastão durante um certo intervalo (o crédito de transmissão) chamado de *intervalo de retenção do bastão* (IRT). O IRT é um parâmetro operacional fixado pelo administrador da rede. O TRB depende do número de estações ativas (NEA) *i.e.,* estações que estão tentando transmitir, do IRT e do tempo de propagação do bastão ao longo do anel (TP):

$$\mathrm{TRB} = \mathrm{NEA} \cdot \mathrm{IRT} + \mathrm{TP} \tag{5.1}$$

Cada estação tem chance de transmitir durante o IRT. Se uma estação estiver inativa e não tem nada a transmitir, ela simplesmente passa o bastão para sua sucessora. Se uma estação tiver um número grande de quadros a transmitir, ela transmite somente enquanto o IRT não expirar.

O *tempo desejado de rotação do bastão* (TDRB) é um valor escolhido de comum acordo por todas as estações no anel de forma a garantir que a utilização do meio físico seja a máxima possível mas permita oportunidades de transmissão satisfatórias para todas as estações. Cada estação mantém uma medida do tempo de rotação do bastão, $\mathrm{TRB}_\mathrm{med}$, obtida pela diferença entre o instante da chegada do bastão e sua última passagem pela estação.

Uma vez escolhido o TDRB, as estações comparam o tempo de rotação do bastão medido TRB_{med} com o tempo desejado. Se o bastão chegar no momento esperado, ou se o bastão está adiantado, a estação pode transmitir até esgotar o intervalo de retenção do bastão. Se o bastão está atrasado ($\text{IRB} \leq 0$), a estação não transmite.

$$\text{IRB} = \text{TDRB} - \text{TRB}_{\text{med}} \tag{5.2}$$

Este esquema de alocação de crédito de transmissão só se aplica ao transporte de dados da classe de serviço assíncrono. As transmissões da classe síncrona devem ser pré-alocadas e os quadros síncronos devem ser pequenos. Existe a possibilidade de que o bastão esteja atrasado e uma estação necessite transmitir quadros síncronos. Neste caso, a estação transmite os quadros síncronos mas isso faz com que o TDRB seja excedido. As alocações para serviços síncronos devem ser tais que o TRB seja no máximo o dobro do TDRB.

Por ocasião da inicialização da rede, o TDRB deve ser escolhido de comum acordo pelas estações. Além disso, para evitar a criação de vários bastões, a escolha do TDRB durante a inicialização também resulta da criação de *um único* bastão e da entrega deste para a estação que solicitou o menor valor para TDRB. O algoritmo distribuído para a criação do bastão e escolha do TDRB é similar a um leilão onde todas as estações solicitam aos seus pares um valor para o TDRB. Cada estação transmite um quadro de solicitação de TDRB continuamente até que um único valor para o TDRB resulte do processo. A transmissão destes quadros se dá sempre que uma estação suspeitar que o anel sofreu alguma pane, ou a rede não está funcionando corretamente porque o bastão foi destruído ou duplicado. O algoritmo consiste dos seguintes passos:

1. Cada estação transmite um quadro de solicitação com seu valor desejado de TDRB;
2. Ao receber um quadro de solicitação, a estação compara o valor de TDRB recebido com o seu próprio. Se o valor recebido for menor que o desejado, a estação passa adiante o quadro recebido e pára de transmitir o seu quadro de solicitação;
3. Se o valor do TDRB recebido for maior que a sua própria solicitação, a estação bloqueia os quadros de solicitação recebidos e transmite quadros com o seu valor (menor) de TDRB;
4. Se o valor do TDRB for igual ao desejado, o fator de desempate é o endereço das estações: se o endereço fonte dos quadros de solicitação for um número maior que o endereço da estação, os quadros são copiados sem modificação. Senão, a estação bloqueia os quadros recebidos e transmite seus próprios;
5. Se uma estação recebe de volta o seu próprio quadro, aquela estação venceu o processo de escolha e é a dona do bastão. O seu quadro de solicitação deu uma volta completa no anel sem ter sido bloqueado por nenhuma outra estação.

A estação vencedora transmite um bastão, que não é usado para transmissão de dados mas apenas para inicialização do anel. Todas as demais estações ajustam seus valores de TDRB para o menor valor recebido durante o processo de escolha. A duração do processo de escolha é aproximadamente um tempo de propagação ao longo do anel mais uma volta do bastão de inicialização.

5.5.4 Nível de Enlace – Controle de Enlace Lógico

Os quadros definidos pelo padrão FDDI podem ter no máximo 4500 octetos. O tamanho dos quadros deve ser limitado porque ele influi diretamente no desempenho do anel. Para um certo

tempo de rotação do bastão, a transmissão de quadros excessivamente longos reduz as chances de transmissão de todas estações no anel. Por outro lado, quanto maior o quadro, maior a utilização do meio físico pelas transmissões.

A Figura 5.40 mostra os campos de um quadro e do bastão. Cada símbolo de dados representa 4 bits de informação codificados como mostra a Tabela 5.6 (pág. 168). Cada quadro deve ser precedido de um preâmbulo de 16 símbolos idle. Estes símbolos permitem que as estações se preparem para a recepção do quadro que segue o preâmbulo. O delimitador de início marca o início do quadro propriamente dito e consiste do par de símbolos de controle JK. A seqüência de bits contida no par JK (100010001) não ocorre no campo de dados, sendo portanto um marcador facilmente identificado. O campo controle de quadro identifica o tipo de quadro. Os campos de destino e fonte identificam o remetente e o destinatário do quadro. Os endereços podem ter 48 bits (mesmo formato que os endereços Ethernet) ou 16 bits. Para simplificar o uso de pontes, o uso de endereços de 16 bits é desaconselhado. O campo TDP é o TDP-32 descrito na Seção 4.1. O delimitador de final indica o final do quadro e consiste do símbolo T. Os três símbolos no campo de status indicam o estado do quadro. Estes símbolos podem ser set (=verdadeiro) ou reset (=falso) e indicam três condições: (1) erro detectado, (2) endereço reconhecido, e (3) se o quadro foi copiado pelo destinatário. Dado o limite de 4500 octetos por quadro, o campo de informação fica limitado a 4478 octetos –o cálculo do tamanho do campo de informação considera somente 4 símbolos do preâmbulo, ao invés dos 16 símbolos, como mostrado na Figura 5.40. O quadro bastão também é precedido de 16 símbolos idle e do par JK. O controle de quadro indica o tipo de bastão e é seguido pelo delimitador de final.

Os três símbolos de status são chamados de indicadores E, A e C, para Erro, 'Address' e Cópia, respectivamente. Ao transmitir um quadro, a estação emite o indicador E=reset, indicando a ausência de erros. Cada estação no anel verifica a validade do quadro e se um erro é detectado, o indicador E é marcado set. O indicador A é marcado set se uma estação reconhece o campo destino do quadro como sendo seu próprio endereço. O indicador C indica que o quadro foi copiado por alguma estação, que tanto pode ter sido o destinatário quanto uma ponte que vai transferir o quadro para uma outra rede.

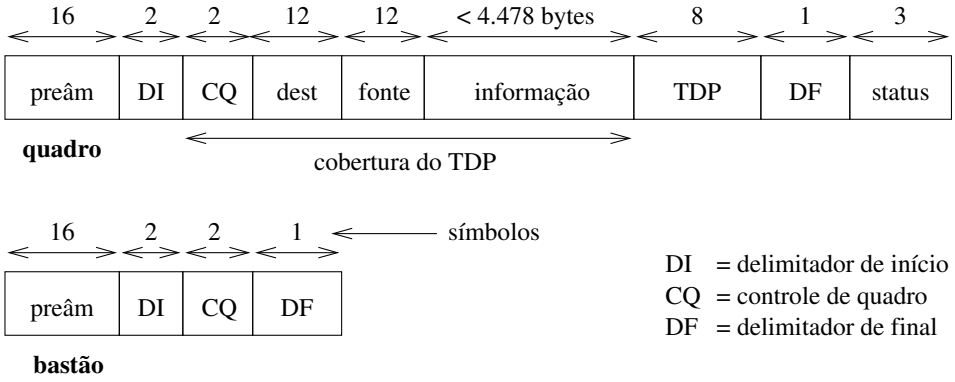


Figura 5.40: Formato dos quadros e do bastão.

5.5.5 Topologia, Estações e Concentradores

A FDDI define dois tipos de estação, em função do número de portas de ligação à rede. Uma *porta de ligação* à rede consiste de um par de fibras pois cada fibra transporta informação em somente uma direção. Assim, das duas fibras, uma traz os sinais da estação anterior no anel e a outra leva os sinais para a estação posterior no anel. Uma estação com uma única porta (duas fibras) é chamada de *estação mono-porta* ("Single-Attachment Station" ou SAS). Uma estação mono-porta se conecta a somente um dos dois anéis lógicos. Estas estações contém os protocolos

dos subníveis PMD, PHY, MAC e SMT. A Figura 5.41 mostra os componentes lógicos de uma estação mono-porta.

Uma *estação dual-porta* se conecta aos dois anéis lógicos com quatro fibras, duas em cada uma das suas duas portas ("Dual-Attachment Station" ou DAS). Assim, uma estação dual-porta contém dois pares de protocolos PMD+PHY e pode conter uma ou duas instâncias do protocolo CAM (no mínimo uma). Opcionalmente, as estações dual-porta podem conter um 'desvio' que permite a reconfiguração dinâmica dos anéis. Dependendo das ligações internas da unidade de desvio, a estação pode estar conectada aos dois anéis lógicos, a um anel lógico ou desconectada da rede.

Obviamente, a interface de rede em estações mono-porta é mais barata que aquela de estações dual-porta. Em princípio, as interfaces mono-porta são usadas em estações de trabalho e periféricos. As interfaces dual-porta são usadas em servidores ou outros equipamentos cuja operação necessita da confiabilidade propiciada pela ligação aos dois anéis. Além disso, equipamentos conectados aos dois anéis não podem sofrer cortes frequentes de energia por causa da interferência que isso causa na operação do anel –interrupção do anel, reconfiguração e reinicialização. Dependendo da cultura institucional, as estações de trabalho podem ser desligadas ao final do expediente. Para evitar os transtornos decorrentes da interrupção do anel, as estações de trabalho tem interfaces mono-porta e se ligam à rede através de concentradores. O concentrador desativa automaticamente a porta de uma estação se ela é desligada. Um concentrador se liga aos dois anéis e as estações se ligam ao concentrador. Vários concentradores podem ser conectados formando uma árvore de concentradores e estações.

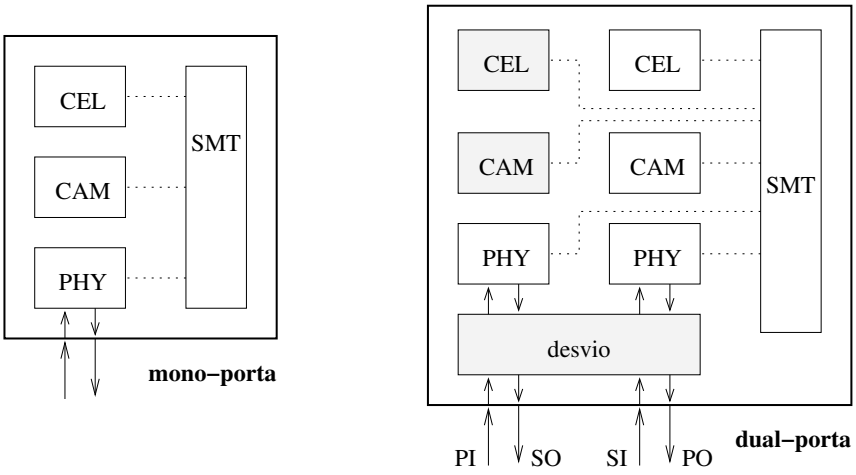


Figura 5.41: Estações dual-porta e mono-porta.

Assim como as estações mono- e dual-porta, os *concentradores* também podem ser implementados com uma ou duas portas ou, com nenhuma porta e somente ligações para estações mono-porta. As conexões para estações mono-porta são chamadas de *portas do tipo M*. A Figura 5.42 mostra os três tipos de concentradores e os componentes lógicos de um *concentrador dual-porta* ("Double-Attachment Concentrator" ou DAC). Naquela figura, os componentes sombreados são opcionais. As duas portas permitem a conexão de concentradores dual-porta aos dois anéis lógicos. Os concentradores com somente portas para estações mono-porta são usados tipicamente para interligar algumas estações em redes pequenas e de baixo custo. Estes são chamados de *concentradores zero-porta* ("Null-Attachment Concentrator" ou NAC) e devem ser a raiz da árvore de estações. Os *concentradores mono-porta* ("Single-Attachment Concentrator" ou SAC) interligam grupos de estações a somente um dos anéis lógicos.

A única porta das estações mono-porta (SAS ou SAC) é chamada de *porta S* ("Slave Port") e

os componentes PHY+PMD são ditos de tipo S. Nos concentradores, as portas de ligação com as portas S das estações SAS são chamadas de *portas M* ("Master Ports"). A porta que conecta um concentrador mono-porta ao anel lógico também é chamada de porta S. As duas fibras ligadas a uma porta S pertencem ao mesmo anel; uma das fibras traz sinais da estação $\mathcal{E}_{i-1}$ e a outra leva sinais para a estação $\mathcal{E}_{i+1}$. As duas portas das estações dual-porta (DAS ou DAC) são chamadas de *porta A* e *porta B*. A fibra ligada à entrada na porta A transporta sinais provenientes do anel primário ("primary-in" ou PI); a fibra ligada á saída da porta A transporta sinais destinados ao anel B ("secondary-out" ou SO). A porta B se liga as fibras SI e PO ("secondary-in" e "primary out"), como mostra a Figura 5.42.

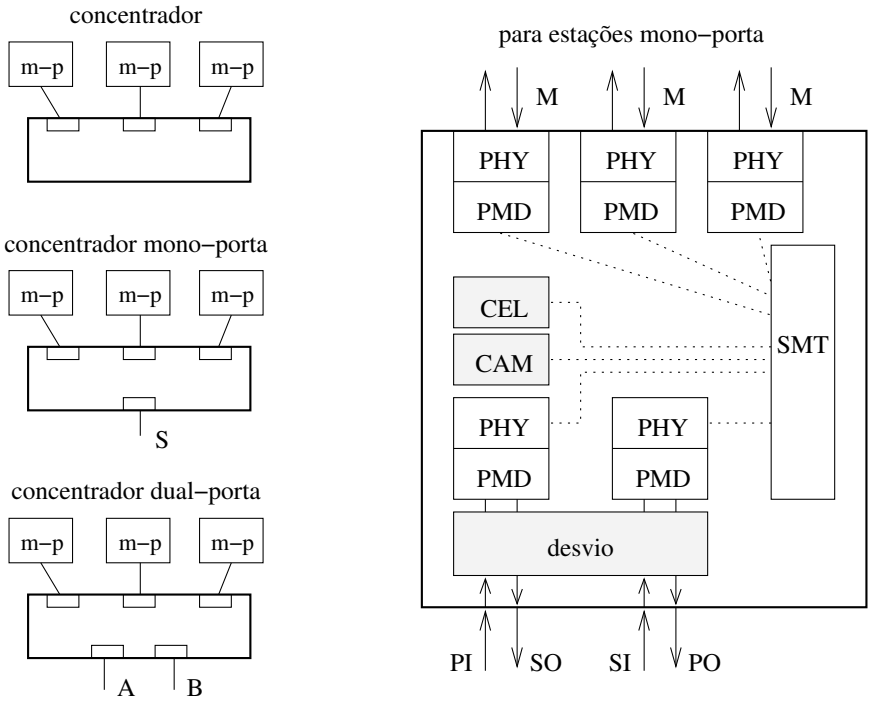


Figura 5.42: Tipos de concentradores.

A Figura 5.43 mostra a representação lógica de uma rede FDDI em uma configuração em anel e árvore. A raiz da árvore é um concentrador dual-porta conectado aos dois anéis lógicos e a três estações mono-porta. Duas estações dual-porta também se conectam aos dois anéis lógicos. O anel representado pela linha contínua é chamado de *anel primário* e, em operação normal, transporta todos os quadros inseridos no meio físico pelas estações. O *anel secundário*, representado pela linha tracejada, geralmente não é usado para o transporte de quadros. Em caso de problemas em uma estação ou um segmento de fibra, o anel secundário é usado para desviar o tráfego do segmento ou estação inoperantes. O segundo componente CAM nas estações dual-porta, responsável pelo controle de acesso ao meio no anel secundário, é opcional porque o anel secundário é usado principalmente para desviar o tráfego ao invés de para transportar quadros.

Um dos requisitos de projeto da rede FDDI foi a possibilidade de interligação de várias redes locais em uma rede intra-campus. A alta capacidade da FDDI, quando comparada a Ethernet 10Base[T,2,5], faz dela um veículo apropriado para o transporte de dados entre várias Ethernets. A Figura 5.44 mostra uma rede FDDI interligando quatro redes Ethernet e uma rede FDDI. A transferência de quadros de uma rede para a outra se dá através de pontes 802.2, numeradas de 1 a 5 na figura. Neste exemplo, cada ponte se comporta como uma estação dual-porta na rede FDDI-0. As pontes possuem memória interna para armazenar quadros durante a transferência de uma rede de alta velocidade para uma rede de baixa velocidade (*e.g.* FDDI

para 10Base2). Como a forma de endereçamento na Ethernet e na FDDI é idêntica, as pontes devem traduzir a codificação de Nível Físico dos dados que seguem de uma rede para a outra. Cada ponte ainda mantém tabelas com os endereços das estações em cada uma das redes que interliga.

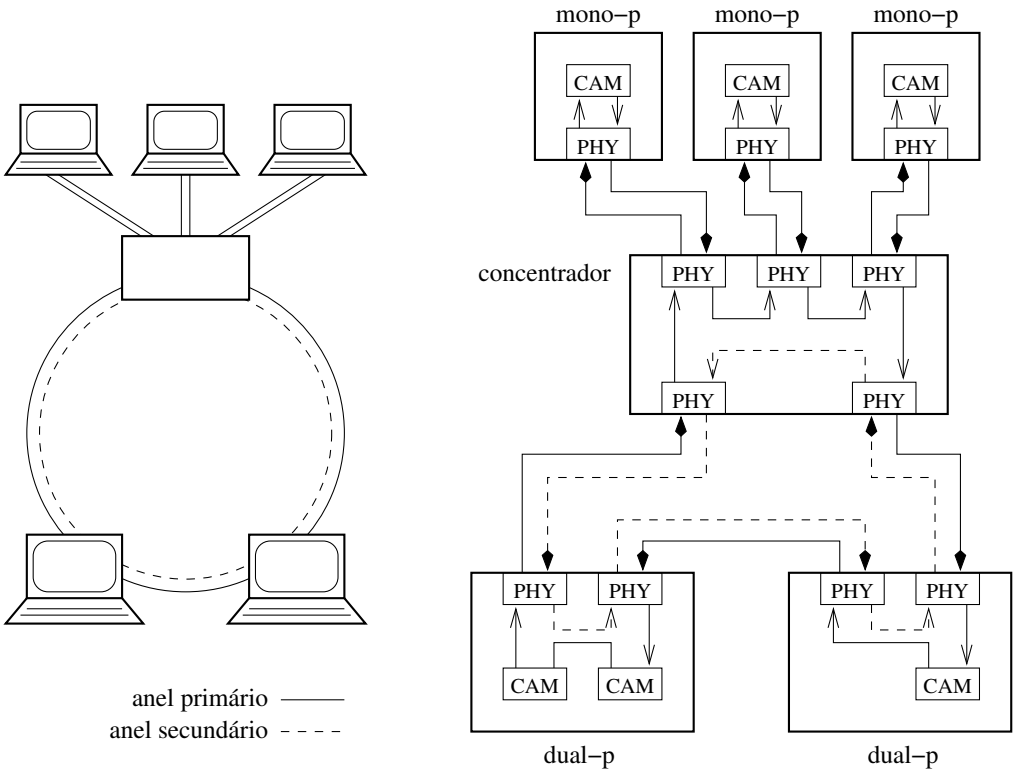


Figura 5.43: Exemplo de topologia FDDI.

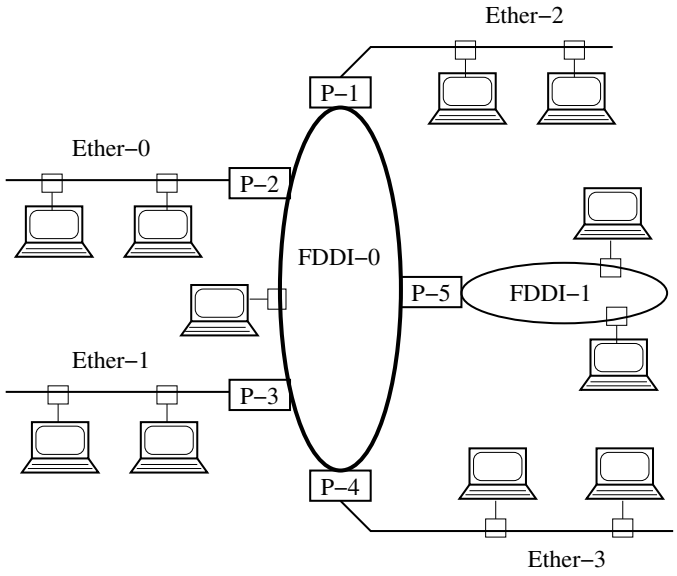


Figura 5.44: Rede FDDI como espinha dorsal.

5.5.6 FDDI-II

Uma extensão da FDDI, chamada FDDI-II, permite o transporte de canais de voz através de uma rede FDDI. Isso permite a integração dos serviços de transporte de dados assíncrono com o transporte síncrono de voz numa única infra-estrutura. Dessa forma, uma instituição pode combinar seu sistema interno de telefonia com sua rede local. Na FDDI-II, o Subnível CAM da FDDI é dividido em duas camadas. A camada inferior é chamada de *multiplexador híbrido* e a camada superior consiste de dois protocolos CAM, o *CAM de processamento de pacotes* e o *CAM isócrono*. O multiplexador híbrido examina cada quadro recebido. Se um quadro é destinado à estação, o multiplexador remove o cabeçalho do subnível CAM e repassa o quadro para a camada de processamento de pacotes ou para o CAM isócrono. Esta organização é mostrada na Figura 5.45.

O tráfego assíncrono e síncrono de pacotes de dados é tratado como na FDDI. O transporte isócrono tem restrições sérias quanto aos intervalos de entrega dos dados aos destinatários porque esta classe de tráfego destina-se ao transporte de sinais de voz. Na telefonia digital, os sinais de voz são discretizados e transformados em uma seqüência de amostras digitais. Cada amostra consiste de um número de 8 bits, proporcional à intensidade do som no instante da amostra. As amostras são tomadas a cada 125 μs e um canal de voz digitalizado gera um fluxo de 64.000 bits por segundo. Para evitar deterioração na qualidade do som recebido, estas amostras devem ser entregues ao receptor numa cadência muito próxima à cadência de amostragem. O transporte isócrono deve, portanto, garantir que os fluxos de voz tenham vazão livre através da rede. Isso é obtido pela criação de quadros chamados *ciclos*, que circulam pelo anel a intervalos de 125 μs. Cada ciclo pode transmportar até 16 *canais* e cada canal pode transportar um fluxo de até 6.144 Mbps. Um fluxo de 6.144 Mbps transporta vários canais de voz e informação de controle e gerenciamento para o sistema de telefonia.

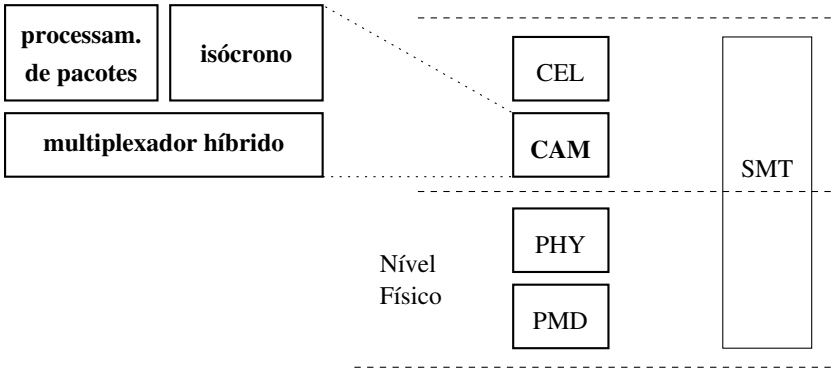


Figura 5.45: Subnível CAM na FDDI-II.

5.6 Scalable Coherent Interface

A Scalable Coherent Interface (SCI) é definida no padrão IEEE 1596-92 e consiste de uma rede confinada para a interconexão de até 64K nodos de processamento. O sistema de comunicação da SCI é baseado em anel com inserção de registrador capaz de operar na velocidade 1 Gbytes/s com ligações ponto-a-ponto unidirecionais. Além da interconexão, a SCI contém um protocolo de coerência de caches para multiprocessadores com memória logicamente compartilhada e fisicamente distribuída. Esta seção descreve os protocolos de Nível Físico e de Transporte definidos no padrão SCI. O protocolo de coerência de caches é descrito em [IEE92] e seu desempenho avaliado em [HT95, HT96].

A SCI define um meio de transporte de dados versátil, de alto desempenho e que pode

ser usado em multicomputadores, para a interligação de computadores completos segundo o modelo de memória logicamente distribuída. A SCI pode também ser usada em multiprocessadores, para interligar vários processadores a uma memória logicamente compartilhada entre eles. Neste último caso, a SCI suporta uma hierarquia de memória com caches coerentes. A SCI pode ainda ser usada como uma rede confinada de alta velocidade para interligar computadores a seus periféricos. Todos estes usos são possíveis porque os protocolos definidos pela SCI são simples e portanto podem ser implementados de forma eficiente.

Alguns dos serviços de transporte suportados pela SCI também o são por redes de projeto recente, como a comunicação em Modo de Transferência Assíncrono [MS95]. Ao contrário das redes ATM, SCI foi projetada exclusivamente para satisfazer as demandas por altas taxas de transferência (maiores que 10^9 bytes por segundo) com intervalos pequenos entre uma requisição de dados e sua satisfação (menor que 1/2 microsegundo). Atualmente, não existe outra rede capaz de satisfazer estes requisitos de projeto, basicamente porque aquelas são projetadas para resolver um outro tipo de problema – a interligação de um número variável de usuários ativos, cada um com demanda altamente variável por serviços de transporte de dados.

5.6.1 Topologias

Um *nó de processamento* pode ser desde um módulo com um processador e sua memória cache até uma estação de trabalho ou periférico veloz. Um *aglomerado* de computadores é um conjunto de nós de processamento interligados por uma rede e normalmente usados de forma concorrente na solução de problemas computacionais.

Existem inúmeras maneiras de se interligar um certo número de nós de processamento para que estes possam cooperar na solução de problemas computacionais. A "melhor" delas é escolhida em função da aplicação do aglomerado, do custo final do aglomerado e do custo relativo entre seus componentes (rede *versus* nós). As topologias mais adequadas a uma rede do tipo da SCI são da família do toro multidimensional com ligações unidirecionais. Os três "caçulas" da família são o anel, a malha e o cubo.

O *anel* é um toro unidimensional onde cada nó tem duas ligações, uma de entrada e outra de saída, como mostra a Figura 5.46. O tráfego pela rede tende a ser proporcional ao número de nós e esta característica impõe um limite no número de nós que podem ser interligados num anel. Para aplicações numéricas, anéis com mais de 8-10 nós ficam sujeitos a congestionamentos por excesso de tráfego [HT95].

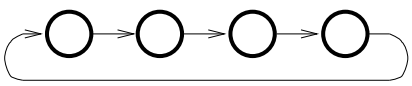


Figura 5.46: Toro unidimensional – anel.

Se o número de nós exceder 8-10, um toro com mais de uma dimensão deve ser usado para interligar os nós. Os nós de uma *malha* possuem quatro ligações, com um par pertencendo a um anel na dimensão X e o outro par pertencendo a um anel na dimensão Y, como mostra a Figura 5.47. O *cubo* mostrado na Figura 5.48 possui 64 nós e pode ser construído com quatro planos iguais ao da Figura 5.47, interligados como 4x4x4 nós. Algumas das ligações foram omitidas para simplificar o diagrama. Redes com maior dimensionalidade podem ser implementadas aumentando-se o número de ligações entre os nós.

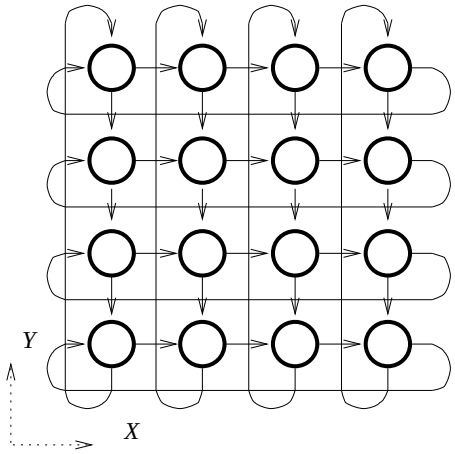


Figura 5.47: Toro bidimensional – malha.

Quando a rede é organizada como um n-cubo k-ário que interliga N nós, anéis com k nós e cubo com dimensão n, estas três grandezas são relacionadas por

$$N = k^n \ , \quad k = \sqrt[n]{N} \ , \quad n = \log_k N \ .$$

Para um certo N, aumentando-se o número de dimensões aumenta-se também a *capacidade* C da rede de transportar pacotes ($C \propto n$), bem como o número de caminhos entre qualquer par de nós. A distância máxima entre dois nós, medida pelo número de ligações é

$$d_{\text{máx}} = n(k-1) \ .$$

Em interconexões do tipo da SCI, para uma dado número de nós, o desempenho da rede melhora com o aumento na dimensionalidade por causa do comportamento de "pipeline" das interfaces de rede [SG91].

Nas redes com dimensão maior que 1 existem duas ou mais ligações entre dois nós, existindo portanto mais de um caminho entre qualquer par de nós. A estratégia para o roteamento de pacotes adotada na SCI é similar ao "wormhole routing" [DS87]. Quando recebe um pacote, o nó necessita examinar o início do pacote para efetuar uma decisão de roteamento. Esta decisão é muito simples, bastando examinar o campo de endereço da cada pacote. A Figura 5.49 mostra uma malha com 16 nós e a associação entre posição no anel e endereço do nó. A estratégia de roteamento consiste, por exemplo, em enviar cada pacote pelo anel da dimensão menor (X), por onde ele viaja até o nó mais distante naquela dimensão. Este nó transfere então o pacote para o anel da dimensão maior (Y), por onde ele vai viajar até chegar ao seu destinatário.

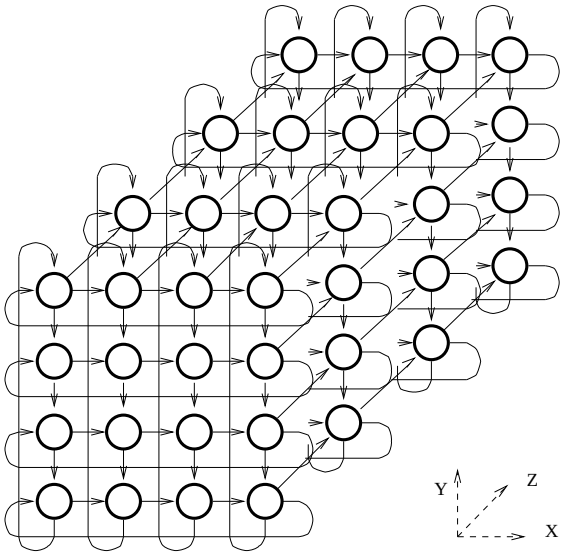


Figura 5.48: Toro tridimensional – cubo.

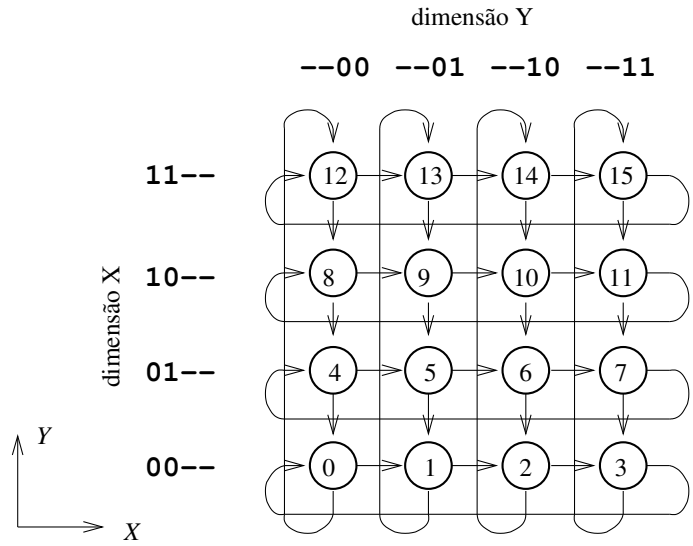


Figura 5.49: Endereçamento e roteamento de pacotes numa malha.

O volume de tráfego tende a ser alto e a se concentrar em certas regiões da rede, próximo aos nós que contém as estruturas de dados do sistema operacional compartilhadas por todos os nós, por exemplo. As transações na SCI consistem de pares pedido–resposta. Um pedido é o pacote que transporta uma requisição de cópia de uma variável compartilhada, e a resposta transporta a cópia solicitada. Para espalhar o tráfego mais uniformemente pela rede, pedidos e respostas são roteados diferentemente. Os pedidos iniciam sua viagem sempre pela dimensão menor enquanto que as respostas iniciam viagem sempre pela dimensão maior. Além de espalhar o tráfego, esta estratégia de roteamento ajuda a garantir a ausência de bloqueios ("deadlocks") entre pedidos e respostas. A Figura 5.49 mostra os caminhos seguidos por um pedido emitido pelo nó número 9, endereçado ao nó 11, e pela resposta de 11 a 9.

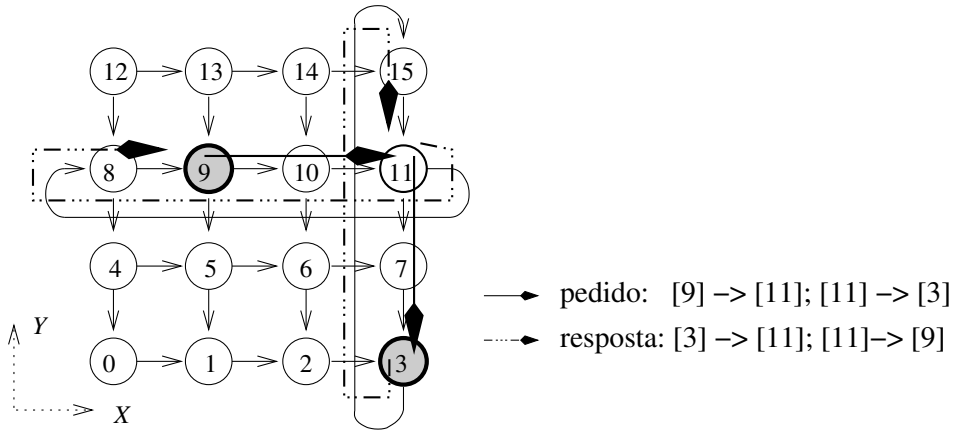


Figura 5.50: Roteamento numa malha.

Cada nó de uma rede com dimensionalidade maior que 1 deve possuir um *comutador* que faz a interface entre o nó e os dois ou mais anéis da rede. Num cubo $k \times k \times k$, cada comutador interliga os três anéis que passam pelo nó e transfere pacotes de uma dimensão para outra de acordo com a estratégia de roteamento. Uma avaliação de desempenho de multiprocessadores baseados em SCI indica que, para um certo conjunto de programas, 64 nós podem ser interligados por uma malha de 8x8 nós sem sofrer congestionamento significativo quando comparado com um cubo de 4x4x4 nós [HT96].

O comutador da Figura 5.51 interliga 4 nós, dois dos quais se ligam a anéis com 4 nós. Este tipo de rede seria mais barato que uma malha de 4x4 nós ou cubo de 4x4x2 nós porque nela, a maioria dos nós possui somente uma interface de anel enquanto que na malha os nós possuem duas e no cubo, três. O custo do comutador é menor que aquele das interfaces adicionais em cada um dos nós. O desempenho desta "estrela de anéis" poderia ser inferior ao da malha ou cubo devido ao tráfego elevado em cada um dos anéis e através do comutador. O desempenho das diferentes classes de aplicação depende de maneiras distintas da vazão e da latência da rede. Em multiprocessadores, a latência deve ser, no máximo, da ordem de 50 μs, e a vazão ão muito grande, da ordem de 10^8 MBytes/s por nó.

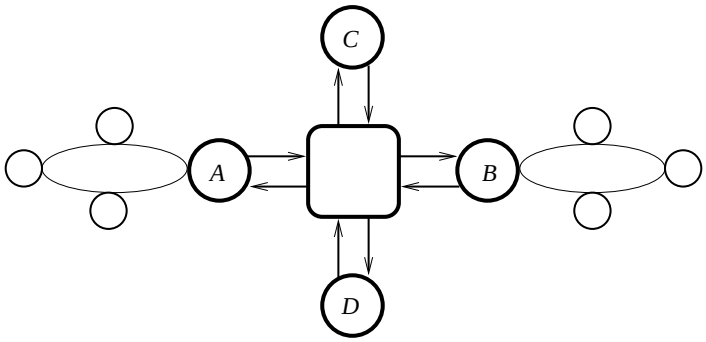


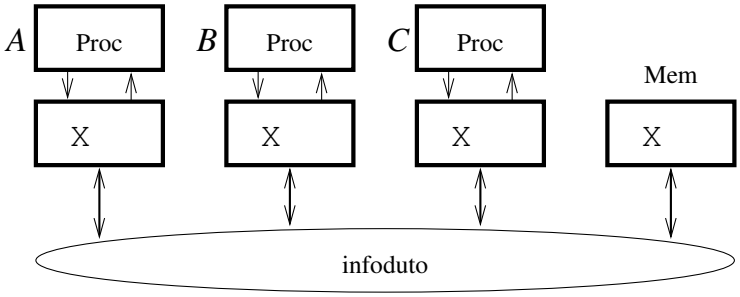
Figura 5.51: Rede com comutador.

5.6.2 Caches em Multiprocessadores

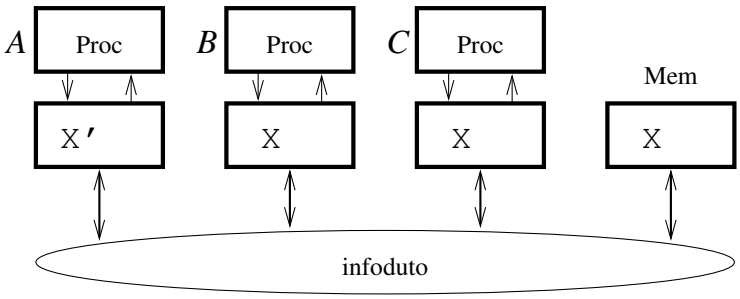
Existe grande diferença nas velocidades de operação de processadores e memória dinâmica (DRAM). A enorme maioria dos computadores de projeto recente possuem uma pequena quantidade de memória estática (SRAM) próxima do processador. Esta memória é capaz de operar na velocidade do processador e contém cópias de dados e instruções que estão sendo acessados pelo processador. Como opera na velocidade do processador, a memória estática reduz o tempo

médio de acesso à memória pelo processador. Esta memória é chamada de *cache* porque, por princípio, ela é invisível ao programador. Mais detalhes sobre caches na Seção 6.3.

A memória cache é efetiva na redução do tempo médio de acesso à memória pelo processador. A função das caches em multiprocessadores com memória compartilhada (MMCs) também é reduzir o tempo médio de acesso à memória. Isso decorre da replicação de variáveis compartilhadas nas caches dos processadores e da migração de dados para a cache do processador que os está usando, como mostrado na Figura 5.52. A discussão abaixo contempla sistemas com uma hierarquia de duas ou mais caches ligadas a um processador.



(a) Inicialmente, três cópias idênticas de X



(b) Processador A atualiza sua cópia para X′

Figura 5.52: Multiprocessador com memória compartilhada.

Quando todos os processadores de um multiprocessador operam sobre um mesmo espaço de endereçamento, estes processadores tem acesso a todas as posições de memória, que é logicamente compartilhada por eles. Quando caches são usadas em MMCs, há a possibilidade de que existam várias cópias de variáveis compartilhadas, espalhadas pelas caches do sistema. Se vários processadores podem atualizar uma dada palavra, o multiprocessador deve conter mecanismos tais que todos os processadores tenham uma visão coerente da memória, isto é, todos os processadores devem ter acesso a apenas a versão "mais recente" daquela palavra. De outro modo, caches não seriam transparentes para o programador.

Protocolos de coerência Num MMC, o infoduto permite a interligação dos processadores, caches e memórias. A Figura 5.52 mostra um multiprocessador com três processadores que se comunicam entre si e com memória através do infoduto. A figura mostra o estado do sistema de memória após os três processadores terem referenciado a variável X, havendo portanto três cópias de X, uma em cada cache. Quando o processador $\mathcal{A}$ atualiza a sua cópia de X para X', os outros dois processadores continuam usando a cópia desatualizada X que está em suas caches. Na maioria dos casos, tal comportamento seria considerado errôneo. Um multiprocessador que contenha algum mecanismo para evitar que esta situação ocorra possui *caches coerentes*.

Dois tipos de *protocolos de coerência* são comumente usados em MMCs. Protocolos baseados em *espionagem e atualização* são implementados em máquinas cujo infoduto permita a difusão de informação e um custo muito baixo (*i.e.,* um barramento, ou seu equivalente topológico, o anel) enquanto que protocolos baseados em *fofoca e invalidação* ou são empregados em máquinas onde a difusão de informação tem um custo proibitivo.

Protocolos com espionagem e atualização Estes protocolos são implementados em multiprocessadores construídos com um barramento, onde se conectam todos os nós de processamento. Quando um processador necessita de uma cópia da variável X, este faz um acesso à memória através do barramento. Se outro processador possui uma versão X', mais recente que a versão em memória, este processador impede que a memória entregue sua versão e propaga X' no barramento. Todos os nós que possuem uma cópia de X em suas caches também atualizam suas versões para X'. Este protocolo se baseia na capacidade de todos os nós observarem todos os acessos ao barramento. Quando um nó observa uma transação no barramento que lhe interessa, este atualiza sua cache com o valor que é propagado no barramento.

Protocolos com fofoca e invalidação Num MMC com protocolo de coerência de caches baseado na invalidação de cópias, quando um processador atualiza uma variável compartilhada, seu controlador de cache deve enviar comandos de invalidação para todas as caches que possuam cópia do bloco que contém a variável a ser atualizada.

Quando um processador lê a variável X pela primeira vez e uma cópia dela não se encontra em sua cache, o processador fica bloqueado até que a cópia de X seja carregada em sua cache. Antes de o processador poder atualizar X, este deve consultar o controlador de memória e enviar comandos de invalidação para as outras caches com cópia de X. Somente quando todas as cópias estiverem invalidadas, o processador pode completar o ciclo de escrita. Quando o processador deseja atualizar um objeto que não se encontra em sua cache, o controlador da cache consulta o controlador de memória. Se houverem cópias, estas devem ser invalidadas; senão, o objeto é copiado para a cache e o processador completa o ciclo de escrita. Quando um processador referencia um objeto cuja cópia em sua cache foi invalidada, uma cópia nova (e mais recente) deve ser obtida pelo controlador de cache.

5.6.3 Arquitetura

O padrão IEEE 1596-1992 define três subsistemas da Scalable Coherent Interface: as interfaces de nível físico, o protocolo de comunicação e o protocolo distribuído de coerência de caches. O nível físico e o protocolo de comunicação são baseados num *anel por inserção de registrador*. Todos os nós de um anel são interconectados por ligações ponto-a-ponto. Todos os nós têm igual direito de transmitir seus pacotes através da rede e um algoritmo distribuído é usado para decidir quais nós podem transmitir em cada instante.

Um *símbolo* corresponde a uma unidade de transferência de informação através do infoduto (*e.g.* 16 bits a cada pulso de relógio de transmissão). Um símbolo é *vazio* se ele não pertence a um pacote. *Símbolos vazios* são transmitidos entre pacotes com a função de sincronizar os relógios de transmissão e recepção de nós vizinhos. Um *pacote* consiste de uma seqüência contínua de símbolos não vazios e contém os endereços dos nós remetente e destinatário, informação de controle e estado, dados opcionais e uma palavra de paridade. A cada pacote transmitido corresponde um *eco*, que informa ao remetente do pacote se aquele foi recebido com sucesso pelo destinatário. Um eco é basicamente o pacote original sem o campo de dados e com os endereços de remetente e destinatário trocados. A Figura 5.53 mostra um pacote com dados e seu eco.

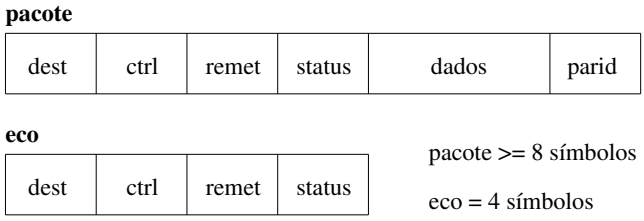


Figura 5.53: Pacote e seu eco.

A Figura 5.54 mostra o diagrama de blocos da interface do anel com inserção de registrador usado na SCI. As filas de desvio formam um registrador de deslocamento cujo comprimento varia com a intensidade de tráfego no anel. Quando não há tráfego de pacotes pelo anel (*i.e.,* nenhum nó transmitindo), as interfaces emitem símbolos vazios continuamente para manter a sincronia entre as interfaces de nós adjacentes.

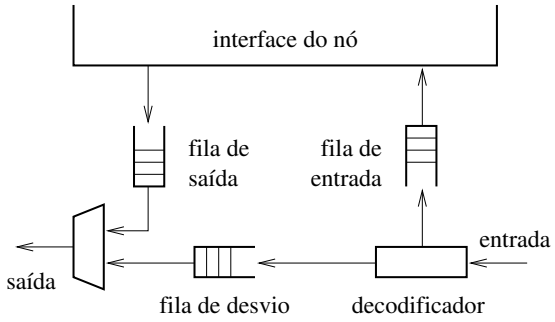


Figura 5.54: Interface com inserção de registrador.

Cada pacote recebido à entrada da interface do anel é examinado pelo decodificador. Se o pacote é destinado ao nó, ele é transferido para a fila de entrada e removido do anel. Em seu lugar, o decodificador insere na fila de desvio um pacote de eco endereçado ao remetente do pacote removido, informando-o da aceitação do pacote. Quando um nó deseja transmitir um pacote, este é inserido na fila de saída. Se a fila de desvio estiver vazia, a transmissão pode iniciar imediatamente. Senão, a transmissão só se inicia quando a fila de desvio ficar vazia. Os pacotes de eco são pequenos, consistindo de 4 símbolos. Quando um pacote é transferido para a fila de entrada, o eco que o substitui, por ser pequeno, produz uma redução líquida no conteúdo da fila de desvio. Dessa forma, cada mensagem aceita abre espaço para novas transmissões. O decodificador substitui os ecos de pacotes emitidos pelo próprio nó por 4 símbolos vazios.

5.6.4 Nível Físico

As interfaces de Nível Físico consistem de ligações unidirecionais ponto-a-ponto de alta velocidade. A versão mais rápida da interface prescreve uma conexão paralela de 16 bits que pode transferir dados a uma taxa de 1 Gbytes/s. O padrão define uma rede genérica que pode interligar até 64K nós. Um nó SCI pode conter um módulo de memória, um processador com suas caches, processadores de E/S ou uma combinação destes. A topologia mais simples consiste de um anel unidirecional com 2 a 20 nós. Em sistemas de maior porte, vários anéis são interligados por comutadores SCI.

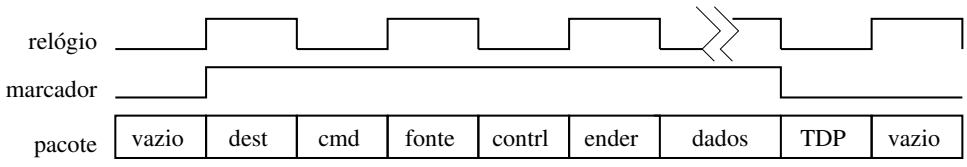
A escolha de ligações unidirecionais facilita a implementação dos circuitos de transmissão e recepção e permite a operação com altas taxas de sinalização. Não há realimentação direta entre transmissor e receptor *i.e.,* não há um sinal do receptor para o transmissor indicando a aceitação de um pacote; quem se encarrega disto é o protocolo de comunicação. Caso contrário,

a velocidade de sinalização ficaria limitada pelo tempo de propagação dos sinais ao longo do infoduto, por causa da viagem de ida e volta dos sinais de controle (transmissor $\rightarrow$ receptor e receptor $\rightarrow$ transmissor).

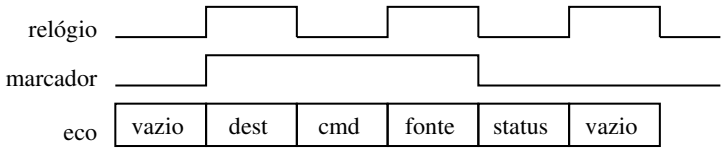
Os infodutos podem ser implementados com cabos de cobre e sinalização diferencial. Nesta forma de sinalização, os sinais são conduzidos por um *par* de fios e o valor de cada bit depende da *diferença* de tensão entre os dois fios (*e.g.* `1` se a tensão no fio A é maior que a do fio B; `0` caso contrário). Cada infoduto consiste de 18 pares de fios. Destes, 16 pares transportam `dados`, um par transporta o sinal de `relógio` e o último par é o `marcador` de início e fim de pacote. As duas bordas do sinal de relógio são usadas para demarcar `símbolos` (*i.e.,* o intervalo de duração de um bit). Para um relógio com freqüência de 250 MHz, cada símbolo dura 2ns e transporta 16 bits com informação de controle ou dados.

A Figura 5.55 mostra o relacionamento entre os sinais de `relógio`, `marcador` e `dados`. Os pacotes com pedidos e respostas são diferenciados dos pacotes de eco pela duração do intervalo em que o `marcador` fica ativo. Os pacotes de eco tem um tamanho fixo de 4 símbolos enquanto que os demais pacotes tem 8 ou mais símbolos. Se o `marcador` ficar ativo por somente três ciclos do relógio, o pacote é um eco; senão, o pacote contém um pedido ou resposta e seu tamanho é um símbolo a mais que a duração do `marcador`.

A limitação mais importante no uso de cabos paralelos de cobre é a pequena distância máxima possível entre dois nós. O relógio de transmissão da SCI opera a 250 MHz, amostrando dados nas duas bordas. A esta velocidade, dificilmente se consegue alcance maior que 10 metros. O padrão define uma interface física com transmissão através de uma fibra óptica. Neste caso, os dados são serializados antes da transmissão, o que resulta em velocidades mais baixas do que na transmissão paralela. Um desenvolvimento recente permite a transmissão paralela através de um feixe com 18 fibras, possibilitando alcances da ordem de 100 metros com a mesma velocidade de transmissão de cabos de cobre [EKB+96].



(a) Enquadramento de pedido ou resposta



(b) Enquadramento do pacote de eco

Figura 5.55: Relação temporal entre sinais no infoduto.

5.6.5 Nível de Enlace – Controle de Acesso ao Meio

O protocolo de enlace é baseado na comutação de pacotes e define os tipos e tamanhos dos pacotes, além das ações necessárias para a transferência de dados entre os nós. As transações suportadas pelo protocolo de comunicação são *bifásicas*. Na primeira fase, o nó *cliente* emite um pedido. Quando o pedido pode ser satisfeito pelo *servidor*, este envia um pacote com a resposta ao pedido. Cada nó deve ser capaz de se comportar tanto como cliente como servidor porque, após completar uma transação, os papéis dos dois nós podem se inverter.

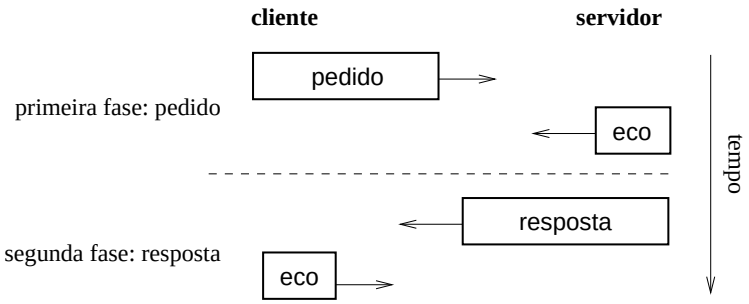


Figura 5.56: Transação com pedido e resposta.

A Figura 5.56 mostra as mensagens trocadas entre cliente e servidor para executar uma transferência de dados entre dois nós. Uma transação completa, *e.g.* busca de um bloco de dados em outro nó, inicia com o cliente enviando um pacote do tipo pedido para o servidor. A aceitação do pacote pelo servidor é sinalizada com um pacote eco-pedido. Quando o servidor executa a ação solicitada, ele envia um pacote resposta contendo informação de estado e possivelmente, dados. O cliente, ao receber este pacote, completa a transação enviando um pacote eco-resposta ao servidor. Um pacote que transporta um comando de movimentação de dados tem 8 símbolos. Um pacote com 64 octetos de dados tem 40 símbolos e um pacote de eco tem 4 símbolos. O protocolo garante progresso contínuo e contém mecanismos para evitar bloqueios ("deadlock" e "livelock") [WM87, IEE92].

O método de controle de acesso ao meio físico usado na SCI é a inserção de registrador. A Figura 5.57 mostra um diagrama de blocos da interface do anel com inserção de registrador. O processador do nó cliente solicita serviços do nó servidor (*e.g.* cópia ou atualização de um bloco de memória) ao inserir um pedido na sua fila de saída. Um nó remove os pacotes de sua fila de entrada, processa as respostas recebidas e atende aos pedidos de outros nós. Ao contrário do anel com passagem de bastão, todos os nós podem transmitir simultaneamente, o que permite uma melhor utilização do anel.

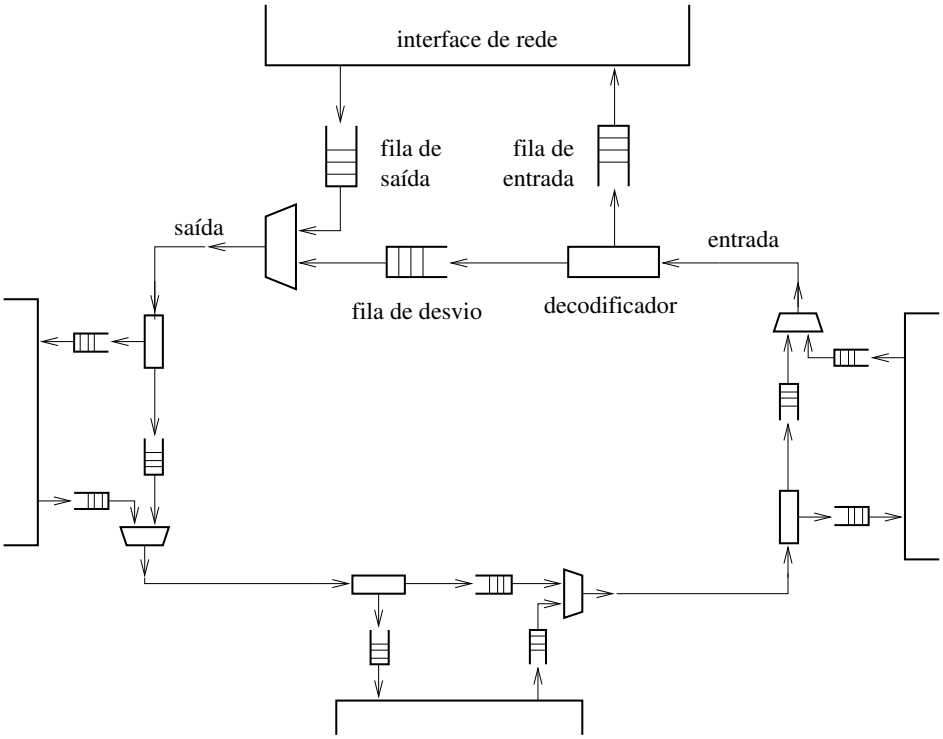


Figura 5.57: Anel SCI.

Inicialmente, o anel contém somente símbolos vazios, usados para o controle de fluxo, discutido adiante. Uma transação inicia quando o cliente insere um pacote pedido, endereçado ao servidor, na fila de saída. A transmissão deste pacote pode iniciar imediatamente se a fila de desvio estiver vazia; senão, a transmissão fica bloqueada até que a fila de desvio se esvazie. Na interface do servidor, o decodificador examina os pacotes recebidos e, se destinados a si, os transfere para a fila de entrada. Quando um pacote é aceito, o decodificador gera um eco-pedido endereçado ao cliente e o insere na fila de desvio, no lugar do pacote transferido para a fila de entrada. Se a fila de entrada tiver espaço para o pacote recebido, o eco-pedido contém a aceitação do pacote; senão, o eco-pedido informa que o pacote foi descartado. Neste caso, quando receber o eco-pedido, o cliente deverá retransmitir o pedido.

A inserção de um eco (4 símbolos) no lugar do pacote transferido para a fila de entrada (8 a 96 símbolos) libera espaço na fila de desvio, possivelmente permitindo o início da transmissão de um pacote retido na fila de saída. Uma vez inserido o eco, o decodificador insere símbolos vazios na fila de desvio até que o eco mais os vazios igualem o tamanho do pacote transferido para a fila de entrada. Note que pacotes destinados a outros nós, a ser retransmitidos pela interface, ultrapassam os símbolos vazios na fila de desvio.

A fila de desvio fica vazia quando ela contiver somente vazios. Os $T-4$ vazios criados pela aceitação de um pacote de tamanho T são agregados num único símbolo. Enquanto a fila de desvio não estiver vazia, qualquer pacote por transmitir fica bloqueado na fila de saída. Uma vez que a fila de desvio só contenha vazios, o nó insere o seu pacote no anel e imediatamente transmite o vazio da cabeça da fila de desvio. Durante uma transmissão, os pacotes recebidos, que são destinados a outros nós, ficam armazenados na fila de desvio até que a inserção do pacote a transmitir se complete. Ao concluir a inserção de um pacote, o nó examina a fila de desvio. Se esta não estiver vazia, o nó transfere quaisquer pacotes nela retidos para a saída, até que a fila de desvio se esvazie novamente. O esvaziamento da fila de desvio é chamado de *fase de recuperação*. Somente após a recuperação o nó poderá tentar a inserção de novo pacote.

A Figura 5.58 mostra em mais detalhe a interface do anel SCI. As filas de entrada e de saída são divididas em duas, uma para pacotes com pedidos (Ped) e outra para pacotes com respostas (Rsp). Isto é necessário para evitar bloqueios que podem ocorrer se respostas ficam enfileiradas atrás de pedidos na fila de transmissão. Quando da transmissão, pacotes com respostas tem precedência sobre pacotes com pedidos. Estas filas devem ter capacidade para conter, no mínimo, um pacote com o maior tamanho dentre pedidos e respostas.

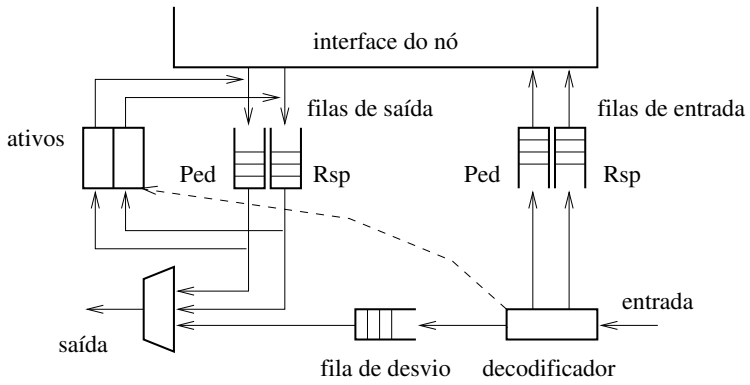


Figura 5.58: Interface SCI.

A Figura 5.58 mostra ainda dois *armazenadores ativos* junto às filas de saída. A função destes armazenadores é permitir que o nó insira mais de um pacote no anel, permitindo assim

que o nó participe de várias transações simultâneas. Uma cópia de cada pacote inserido no anel é mantida no armazenador de pacotes ativos. Quando o eco correspondente a um pacote transmitido anteriormente indica sua aceitação pelo destinatário, este pacote é removido do armazenador de pacotes ativos, liberando espaço no armazenador para a transmissão de um novo pacote. Se o eco indica que o pacote foi rejeitado por falta de espaço na fila de entrada do destinatário, o pacote é copiado novamente do armazenador ativo para a fila de saída. A linha tracejada na figura indica o relacionamento entre os pacotes ativos e os ecos recebidos pelo decodificador. Cada `eco` contém um número de seqüência de 6 bits, que identifica a transação a que ele está associado, permitindo assim que cada nó efetue até 64 transações simultâneas.

Quando um pacote é rejeitado pelo destinatário porque sua fila de entrada está cheia, este reserva espaço para o pacote rejeitado. Quando o pacote for retransmitido, haverá espaço suficiente para ele na fila de entrada do destinatário. Este mecanismo evita as situações em que pacotes de um nó seriam rejeitados eternamente pelo destinatário. O algoritmo usado marca como "velhos" os pacotes rejeitados por falta de espaço. Pacotes velhos tem preferência sobre pacotes "novos" quando da inserção na fila de entrada porque o decodificador sempre reserva espaço para os pacotes rejeitados.

Um dos nós em cada anel SCI é eleito o *zelador* do anel. Este nó verifica a validade dos pacotes que circulam no anel, sendo responsável pela eliminação de pacotes que já circularam pelo anel mais de uma vez. Quando um pacote passa pelo decodificador do nó zelador, este marca o pacote como "visto". Se o zelador receber um pacote já marcado como sendo visto, este pacote é eliminado do anel porque, ou sofreu um erro de transmissão, ou seu destinatário está incapacitado a reconhecer seu próprio endereço nos pacotes. No segundo caso, o remetente dos pacotes deverá efetuar algum procedimento de recuperação de erros fatais ou panes.

5.6.6 Nível de Enlace – Controle de Enlace Lógico

Os pacotes usados na SCI têm os formatos mostrados na Figura 5.59. A cada campo do pacote corresponde um ou mais símbolos de 16 bits. Os campos `idDest` e `idFnte` são os endereços físicos dos nós destinatário e remetente e permitem a identificação de 65520 nós (os endereços acima de 0xFFF0 são reservados). O campo `cmdo` identifica o tipo de transação e indica se o pacote já circulou mais de uma vez pelo anel (*cmdo.visto*). Se o bit *cmdo.visto* estiver ativo, o nó zelador elimina o pacote. O campo *cmdo.fase* é usado no controle do uso das filas de recepção. Se um pacote foi rejeitado pelo destinatário porque a fila de entrada estava cheia, este aloca espaço para o pacote rejeitado de forma a garantir progresso contínuo para o remetente do pacote.

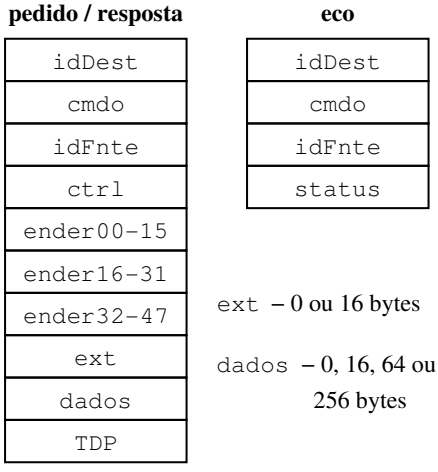


Figura 5.59: Formato dos pacotes SCI.

O campo `ctrl` contém um número de 6 bits para identificar transações (*ctrl.idTrans*) e um indicador de longevidade (*ctrl.expira*). O protocolo de comunicação permite que uma interface de anel transmita vários pacotes em seqüência, sem ter que esperar pelo eco de um pacote antes de enviar o próximo. Devido às diferenças nas distâncias entre os nós na rede, os ecos podem ser recebidos numa ordem diferente daquela em que os pacotes foram enviados. O identificador de transação no campo `ctrl` (*ctrl.idTrans*) serve para "casar" os ecos recebidos com seus respectivos pacotes. Uma vez recebido o eco indicando a aceitação do pacote, este pode ser removido do armazenador de pacotes ativos (Figura 5.58).

Existe a possibilidade de erros no tratamento de pacotes recebidos por causa de erros de transmissão (*e.g.* erro no endereço do destinatário) ou de roteamento (*e.g.* um eco é duplicado no comutador). Para evitar que pacotes defeituosos fiquem circulando eternamente pela rede, o campo *ctrl.expira* indica o instante em que o pacote deixa de ser válido, devendo então ser descartado pelo nó que detectou o final da validade do pacote. O campo *ctrl.expira* contém um número de ponto flutuante que indica o horário, do ponto de vista do sistema SCI, em que o pacote perde sua validade. O padrão SCI prescreve um algoritmo para a sincronização dos relógios de tempo-real de todos os nós na rede.

Em pacotes do tipo pedido, os três campos `ender00-15`, `ender16-31` e `ender32-47` indicam a posição em memória sobre a qual o comando especificado em `cmdo` deve atuar. Em pacotes com respostas, estes campos transportam os apontadores usados no protocolo de coerência de caches e indicação de status da operação indicada no `cmdo` do pedido associado a esta resposta. O campo `ext` é usado em algumas das transações do protocolo de coerência de caches. O campo de `dados` pode transportar 0, 16, 64 ou 256 octetos. O campo `TDP` contém o bloco de paridade do pacote, computado com o polinômio CCITT TDP-16.

A cada pacote com pedido ou resposta corresponde um eco. Os endereços de fonte e destinatário do eco são os campos `idDest` e `idFnte` do pacote que está sendo ecoado, isto é, o destinatário do eco é a fonte do pedido ou resposta. O campo `cmdo` do eco contém o mesmo identificador de transação do pacote que lhe deu origem (*ctrl.idTrans*). Há um bit que indica se o pacote foi aceito ou rejeitado pelo seu destinatário (*ctrl.rej*). Outro bit indica o tipo de eco, se eco-pedido ou eco-resposta (*ctrl.resp*).

Controle de fluxo O controle de fluxo é efetuado de forma distribuída pelos nós no anel. Cada símbolo vazio contém um *go-bit* que é usado pelos nós para controlar o fluxo dos pacotes que recebe. Um nó pode inserir seus pacotes somente após a fila de desvio estar vazia e o símbolo vazio na cabeça da fila estar com seu *go-bit* ativo. Um nó que está tentando inserir seus pacotes no anel pode impedir os demais de transmitir pela emissão de símbolos vazios com o *go-bit* inativo. Quando um nó passa a emitir vazios com o *go-bit* inativo, os pacotes em trânsito no anel serão aceitos pelos seus destinatários e os ecos correspondentes criarão espaço nas filas de desvio, permitindo assim novas transmissões. Quando um nó não tem pacotes a transmitir, o nó retransmite os vazios sem alterar o estado de seus *go-bit*s.

Quando a fila de desvio está vazia, a interface acumula inclusivamente o valor do *go-bit* dos vazios recebidos. Quando um nó tem um pacote por transmitir, este espera até que a fila de desvio esteja vazia para inserir o pacote. Isto só é possível se o símbolo vazio na cabeça da fila estiver com seu *go-bit* ativo. Após inserir um pacote, a interface deve emitir um vazio com o *go-bit* ativo, permitindo assim que um dos outros nós tenha a oportunidade de transmitir. Quando o anel é inicializado, todos os vazios tem o *go-bit* ativo. O número de *go-bit*s no anel varia de forma inversamente proporcional à intensidade do tráfego.

5.7 Bibliografia e Exercícios

Mais detalhes sobre os padrões de redes locais da família IEEE 802 podem ser encontrados em [IEE85a, IEE85b, IEE85c], `http://www.ieee.org` e `http://www.iso.ch` . Uma quantidade enorme de informações sobre redes locais pode ser encontrada em
`http://web.syr.edu/~jmwobus/lans` .

[Lam84b] discute protocolos de acesso a meios físicos compartilhados. [Bux84] faz uma comparação do desempenho de redes locais com as topologias de barramento e anel. [SCP84] discute a adequação da topologia em anel para uso em redes locais. [BGY87] compara o desempenho dos protocolos de acesso ao meio em anéis. Modelos matemáticos do comportamento dos métodos de controle de acesso ao meio são encontrados em [Tan89, BG92]. Os artigos originais sobre Ethernet são [MB76, SDRC84]. Informações sobre a instalação e configuração de redes IEEE 802.3 podem ser encontradas em [Spu94b, Spu94a]. Veja [MKK97] para detalhes sobre o método de acesso ao meio da Ethernet 1000Base∗. Veja [KL90] para uma análise de desempenho de barramentos de alta velocidade. [AD89] contém extenso material sobre técnicas de comutação usadas em redes de dados. Arquiteturas de comutadores são discutidas em [Lee90].

O Cambridge Ring é um exemplo de rede local com anel segmentado [HN88]. O Express Ring usa um anel segmentado como o equivalente topológico de um barramento de multiprocessador, além de um protocolo de acesso mais eficiente que o descrito aqui [BD91]. O MetaRing usa dois anéis, cada um com transmissões numa direção. As mensagens são transmitidas sempre no anel com menor distância entre remetente e destinatário. Isso reduz drasticamente o tempo médio de viagem das mensagens. Para uma descrição completa veja [CO93]. [ZNV88] analisa o desempenho de redes baseadas em anéis segmentados.

Conceitos relacionados a FDDI e discussão sobre seu processo de padronização são apresentados em [Ros86, Ros89, DB88]. [Jai94] contém descrições detalhadas de todos os componentes do padrão FDDI e discussões sobre as razões para a escolha das técnicas empregadas. [AK91, LCLY91] contém estudos comparativos sobre redes de alta velocidade. [GCW91] contém extensa discussão sobre sistemas de fibras ópticas.

A Seção 5.6 é baseada em [Hex97]. Para outros exemplos de multiprocessadores e multicomputadores, bem como comparações entre eles, veja [Hwa93, AG94, LW95, HP96]. Para uma discussão sobre aglomerados de computadores ("clusters") veja [Pfi95]. Informações sobre sistemas e projetos baseados em SCI podem ser obtidas na página WWW de Associação de Usuários da SCI – `http://sunrise.scu.edu` . Para uma introdução aos problemas associados a hierarquias de memória com caches coerentes veja [Hex96]. Mais detalhes sobre o protocolo de coerência de caches da SCI podem ser encontrados em [IEE92]. Estudos detalhados sobre o desempenho do protocolo de comunicação da SCI, multiprocessadores baseados em anéis, malhas e cubos SCI são encontrados em [SGV92, SG94, BG95, KABG95, HT95, HT96].

Exercícios

Ex. 5.1 [§ 5.1] Considere uma estação transmitindo continuamente nas taxas de 1, 10 e 100Mbps. Qual é o intervalo médio entre dois erros consecutivos para meios físicos com taxas de erro de um bit errado em cada 10^8 e em 10^{12} transitidos?

Ex. 5.2 [§ 5.2.1] Modifique o protocolo de acesso ao meio do anel segmentado descrito na página 137 para aumentar a utilização dos segmentos, evitando que eles estejam vazios durante meio ciclo, mas sem deixar de informar ao remetente sobre a aceitação das mensagens.

Ex. 5.3 [§ 5.4] Considerando um hiato entre quadros de $9.6\mu s$, um preâmbulo de 64 bits e os campos do cabeçalho e TDP, determine a taxa de transmissão efetiva máxima em uma rede Ethernet 10Base2, para quadros que transportam de um a 1500 octetos. Suponha que somente uma estação está transmitindo em toda a rede. Trace um gráfico que relaciona tamanho do campo de informação à taxa de transmissão.

Ex. 5.4 [§ 5.4] Sob as mesmas ciscunstâncias do exercício acima, qual a taxa de transmissão efetiva máxima em uma rede Ethernet 10BaseT com o comutador da Figura 5.35, pág. 163. Considere dois casos: (1) somente uma estação transmitindo e (2) todas as estações transmitindo sem que ocorram conflitos de destinatário.

Ex. 5.5 [§ 5.4] Com base nos dados dos Exercícios 5.3 e 5.4, qual é a máxima eficiência da comunicação, considerando o campo de dados com o tamanho máximo permitido?

Ex. 5.6 [§ 5.4] Considere uma rede 10Base2 com 5 segmentos e tempo de propagação de um extremo ao outro de $9.6\mu s$. Calcule o tamanho dos fragmentos de quadro resultantes de colisões na situação mostrada na Figura 5.32 (pág. 160) para uma estação próxima às estações $\mathcal{A}$, $\mathcal{B}$ e entre as duas. Pista: determine a duração da colisão em cada um destes pontos.

Ex. 5.7 [§ 5.4, 5.5] Quanto tempo uma mensagem de tamanho mínimo pode ficar bloqueada enquanto espera pela oportunidade de transmitir? No caso da Ethernet, considere somente a espera pela transmissão de um quadro de tamanho máximo (*i.e.,* ignore colisões). Para a rede FDDI, considere os casos em que (1) nenhuma das N estações no anel tem algo a transmitir, e (2) todas as N estações usam todo o tempo a elas disponível.

Ex. 5.8 [§ 5.4, 5.5] Numa Ethernet, se muitas estações injetam quadros na rede, o tráfego total tende a se reduzir por causa das colisões. Nas mesmas condições, o que ocorre num anel FDDI?

Ex. 5.9 [§ 5.4, 5.5] Compare a utilização do meio físico *pelas estações* em redes Ethernet e FDDI. Como a utilização varia com a intensidade do tráfego e com o tamanho dos quadros?

Ex. 5.10 [§ 5.4, 5.5] Estime o número máximo de quadros por segundo que o circuito receptor de uma estação deve tratar nos seguintes casos: (1) Ethernet 10Base2; (2) Ethernet 100BaseTX; (3) FDDI anel pequeno (4 estações); (4) FDDI anel grande (500 estações).

Ex. 5.11 [§ 5.4, 5.5] Com base na resposta do Exercício 5.10, determine as características necessárias ao tratamento das mensagens pelos programas controladores de interface de rede e pelo sistema operacional.

Ex. 5.12 [§ 5.5] A liberação imediata do bastão (pág. 169) numa rede FDDI com muitas estações e enlaces longos pode fazer com que uma estação repasse o bastão antes de receber a informação contida nos símbolos de status do(s) último(s) quadro(s) inserido(s) no anel. Como isso influi no projeto do programa controlador de periférico da interface de rede?

Proj. 5.1 [§ 5.2.1, 5.4, pág 140] Projete uma ponte capaz de interligar redes Ethernet 10Base∗ e 100Base∗. Quanto de memória interna a ponte deve possuir?

Proj. 5.2 [§ 5.2.1, 5.4, 5.5, pág 140] Projete uma ponte capaz de interligar redes Ethernet 10Base∗, 100Base∗ e FDDI. O que fazer quando quadros FDDI com mais de 1500 bytes são destinados a uma Ethernet? O que fazer com os três símbolos de status de um quadro FDDI (E, A, C) destinado a uma estação numa Ethernet?

Capítulo 6

A Interface Rede–Computador

Os princípios básicos de projeto de Sistemas Operacionais foram desenvolvidos há mais de duas décadas, quando os periféricos eram lentos se comparados com os processadores de então. As técnicas de gerenciamento de periféricos em uso corrente ainda são baseadas naqueles princípios, o que limita o desempenho dos sistemas baseados em redes de alta velocidade. Quando uma rede de alta velocidade interliga dois computadores, boa parte da vazão disponível nos terminais do cabo é desperdiçada porque o processamento das operações de Entrada e Saída é muito complexo e ineficiente, limitando o desempenho de todo o sistema.

O desempenho de aplicações baseadas em redes de alta velocidade depende de três fatores. (1) alta vazão para mensagens longas; (2) baixa latência para mensagens curtas; (3) sincronia para transmissões com mídia contínua como voz e vídeo. Estes fatores são, por sua vez, influenciados por três características de projeto dos sistemas de transmissão de alta velocidade. São eles: (a) a taxa de transmissão na rede; (b) o processamento dos protocolos nas duas pontas do infoduto; (c) a interface entre o Sistema Operacional e a aplicação.

O problema fundamental enfrentado pelos projetistas de interfaces de redes de alta velocidade pode ser formulado como a *preservação da vazão entre os aplicativos em máquinas distintas.* Este problema é chamado de *preservação de vazão* ("throughput preservation") e sua solução depende de um conjunto de técnicas de projeto de sistemas de Entrada e Saída (E/S) e de Sistemas Operacionais (SO). Algumas destas técnicas são examinadas nas próximas seções. Este capítulo discute a implementação de sistemas de comunicação de alta velocidade em ambiente Unix e estações de trabalho de alto desempenho porque estes são os sistemas onde a necessidade de serviços de comunicação eficientes é maior. As observações e conclusões se também se aplicam a outros sistemas operacionais e classes de máquinas.

6.1 Vazão e Latência

A *vazão* através de um infoduto é definida como *o número de bits transferidos por unidade de tempo através do enlace.* Esta grandeza é uma métrica importante porque descreve o desempenho de um sistema na transferência de grandes quantidades de dados. Os fatores que limitam a vazão de um infoduto são a sua taxa de sinalização (número de bits transferidos por unidade de tempo), sua largura (número de vias) e a eficiência do protocolo de controle de enlace (informação de endereçamento e controle).

As maneiras de aumentar a vazão através de um infoduto são:

1. aumentar a taxa de sinalização — existem limites físicos e tecnológicos que impedem aumentos arbitrários na taxa de sinalização. Fibras ópticas permitem taxas de sinalização de até $100 \cdot 10^9$ bps, mas a eletrônica necessária para funcionamento a estas taxas pode ser extremamente cara;
2. aumentar a largura do enlace — se as distâncias são da ordem de uns cem metros, enlaces com muitas vias são econômica e tecnologicamente factíveis. Para distâncias maiores, uma única via é normalmente usada;
3. aumentar a eficiência do protocolo — simplificar o protocolo ou melhorar sua implementação. Dependendo da aplicação, o protocolo deve possuir certas características que impossibilitam maior eficiência (*e.g.* detecção e recuperação de erros);
4. interligar os nós através de comutadores porque estes podem efetuar várias conexões diretas entre suas portas simultaneamente.

A *latência* de um infoduto é o tempo consumido entre a emissão de um pacote pelo transmissor e sua absorção pelo receptor. Esta métrica é importante porque indica o custo da transferência de mensagens curtas entre dois nós de processamento. Um dos fenômenos que contribuem para a latência é a velocidade de propagação das ondas eletromagnéticas, que determina o tempo necessário para um bit viajar entre as duas pontas do infoduto. A taxa de transmissão determina a duração de cada bit ou palavra. Quanto maior a taxa de transmissão, menor o tempo dispendido na emissão de um quadro pelo transmissor. Outro componente da latência é o tempo de processamento no produtor do pacote para montá-lo e inserí-lo no infoduto, e o tempo de processamento dispendido pelo consumidor na sua validação. O tempo de processamento nas pontas do infoduto depende do protocolo empregado. A taxa de transmissão depende da taxa de sinalização e da largura do infoduto.

Maneiras de reduzir a latência de um infoduto são:

1. aumentar a velocidade de propagação — até o momento, a velocidade da luz parece ser um limite intransponível... Para distâncias pequenas, sua importância é desprezível. Para taxas de transmissão maiores que 10^8 bps, não pode ser desconsiderada;
2. aumentar a taxa de transmissão — se viável, a largura do infoduto e/ou a taxa de sinalização podem ser aumentadas;
3. reduzir o tempo de processamento — vale a mesma observação que na melhora do desempenho e eficiência dos protocolos. Existe um mínimo de processamento que deve ser efetuado nas pontas do infoduto.

6.2 Que Interfaces?

A Figura 6.1 mostra uma pilha de protocolos como a da família TCP/IP e as interfaces entre o sistema de transporte de bits –a rede– e o computador propriamente dito. A interface entre o programa de usuário e Sistema Operacional se dá através das funções de estabelecimento de conexão, envio e recepção de mensagens. A interface entre o protocolo de transporte (TCP) e o de rede (IP) é interna ao SO e (geralmente) invisível ao programador. A interface entre o protocolo de rede e o protocolo de enlace lógico da rede local é implementado parte em hardware e parte em software. A parte em hardware consiste das funções de controle de acesso ao meio e transferência de octetos entre o adaptador de rede e a memória do computador. A parte em software consiste das rotinas para o tratamento das interrupções geradas pelo adaptador de rede

quando da recepção ou transmissão de quadros ou pacotes. A interface entre o Nível Físico e o de Enlace geralmente consiste da adaptação de sinais elétricos e/ou ópticos e as operações de serialização de dados.

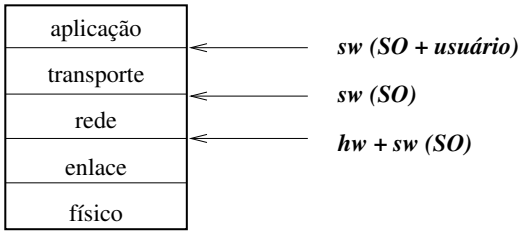


Figura 6.1: Interfaces funcionais entre rede e computador.

A velocidade de processadores tem crescido a taxas de 50% ao ano enquanto que a velocidade de memória dinâmica (DRAM) tem crescido a taxas de 7% ao ano. A velocidade de transmissão tem, pelo menos, acompanhado aquela dos processadores [HP96]. Há uma diferença enorme nas velocidades de operação dos processadores com relação à memória. Um microprocessador de alto desempenho executa uma ou mais instruções por ciclo de relógio, a uma freqüência de operação de 100 a 500 MHz. Uma referência à memória DRAM demora em torno de 100 ns (de 10 a 50 ciclos de relógio do processador). Existem portanto dois pontos de estrangulamento no desempenho de sistemas de transmissão de alta velocidade. O primeiro deles é o sistema de memória dos computadores: qualquer função que efetue muitas operações de memória tem baixo desempenho por causa do custo elevado dos acessos à memória. O segundo ponto de estrangulamento está associado ao tratamento das operações de E/S pelo Sistema Operacional. Este tópico é discutido na Seção 6.4.

Há vinte ou trinta anos atrás, quando a tecnologia de SO em uso corrente foi desenvolvida, os periféricos eram muito lentos quando comparados aos processadores de então. Isso levou a uma série de decisões de projeto que são inadequadas quando aplicadas a interfaces para redes de alta velocidade. A Tabela 6.1 mostra os custos relativos no processamento envolvido na transmissão de mensagens de 1 octeto com os protocolos da família TCP/IP, entre duas estações interligadas pela rede de alta velocidade Nectar [Ste94]. Com os dois protocolos de transporte, TCP e UDP, mais de 60% do tempo é consumido em duas classes de operações efetuadas pelo SO: (1) gerenciamento de armazenadores para as mensagens e (2) trocas de contexto e tratamento de interrupções. A Seções 6.3 e 6.4 discutem técnicas para reduzir os custos associados a estas operações.

Considere uma rede com taxa de transmissão de 1 Gbit/s. Descontando cabeçalhos etc, a taxa de transferência de dados entre adaptador de rede e memória é de $\approx$ 100 Mbytes/s. Para um processador com palavra de 32 bits, isso corresponde a 25 Mpalavras/s ou $\approx$ 40 ns/palavra. Um processador com relógio de 100 MHz que executa uma instrução por ciclo é capaz de executar 4 instruções por palavra transferida. Um processador com relógio de 500 MHz pode executar 20 instruções por palavra transferida. Estes limites implicam em restrições sérias à aplicabilidade de redes de alta velocidade. A Lei de Amdahl garante que é contra-producente usar uma rede de alta velocidade se o sistema de memória é o gargalo no desempenho [HP96].

Protocolo	TCP/IP	UDP/IP	quem
enlace de dados	33%	16%	hw+protocolo
transporte	6%	21%	protocolo
armazenadores	31%	26%	SO †
trocas de contexto	30%	37%	SO ‡

† *cópia de dados.*
‡ *tecnologia de SO para periféricos lentos.*

Tabela 6.1: Custos relativos na transmissão TCP de uma mensagem com 1 octeto.

6.3 Interação entre Caches e E/S

Os fabricantes de computadores interpõem uma memória de alta vlocidade entre o processador e a memória DRAM para reduzir o tempo médio de acesso à memória. A *memória cache* é implementada com tecnologia de circuitos integrados similar a dos processadores, o que a torna mais rápida que DRAM porém mais cara e com menor tamanho. A memória cache faz com que o sistema de memória tenha um tempo de acesso à memória similar ao da cache a um custo por bit próximo ao da memória DRAM. O tempo de acesso à memória DRAM é da ordem de 10 a 50 ciclos do relógio do processador enquanto que o acesso à cache custa de 1 a 10 ciclos. Tamanhos típicos de caches variam de 16 Kbytes a 4 Mbytes enquanto que a memória principal (DRAM) tem de 16 Mbyte a 1 Gbyte.

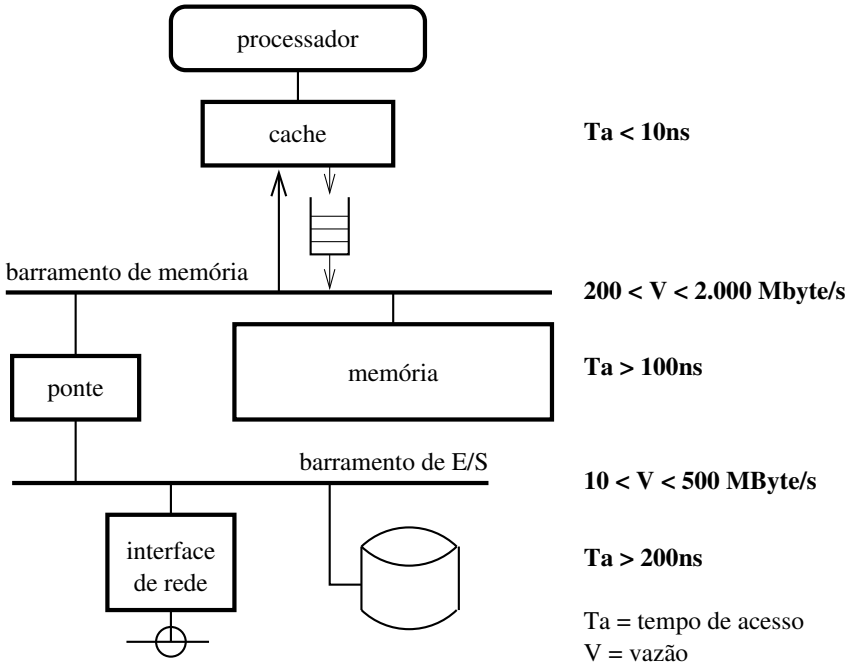


Figura 6.2: Modelo de computador.

A Figura 6.2 contém o modelo de computador que será discutido no que segue. O barramento que interliga processador, cache e memória está no caminho crítico de desempenho e é portanto projetado para possibilitar altas taxas de transferência com baixa latência. O barramento de E/S deve ser capaz de acomodar uma grande variedade de periféricos, alguns rápidos (rede e discos), outros lentos (teclado e apontador). Devido aos requisitos de desempenho, o projeto elétrico do barramento de memória é altamente complexo, o que impede que os periféricos sejam conectados a ele. Uma *ponte* separa eletricamente o barramento de memória do

barramento de E/S. Os parâmetros de projeto do barramento de E/S, como o número e a variedade de periféricos, impedem a operação em altas velocidades. Enquanto que o barramento de memória pode operar com relógio de 50 a 80 MHz, barramentos de E/S operam com relógios de 10 a 50 MHz. A discussão que segue supõe a existência de uma única cache entre processador e memória. Os problemas discutidos abaixo também ocorrem em hierarquias de caches, em formas mais ou menos graves que a tratada aqui. Veja [Hwa93, HP96, Hex96].

O ganho em desempenho advindo do uso de memória cache depende dos dois Princípios da Localidade. A *Localidade Espacial* implica em que os endereços próximos ao que está sendo acessado pelo processador serão, com grande probabilidade, também acessados no futuro próximo. A *Localidade Temporal* implica em que um objeto referenciado pelo processador será referenciado novamente, com grande probabilidade, num futuro próximo. No fragmento de código abaixo, os elementos de `vetor[]` são referenciados de tal forma que elementos alocados contiguamente em memória são também acessados seqüencialmente. Estas referências exibem localidade espacial. O índice `i` e as instruções que implementam o código do comando `for()` são referenciados a cada iteração. Estas referências exibem localidade temporal.

```
for (i = 0; i < TAM; i++)
   vetor[i] = ...  ;
```

Na primeira vez em que um objeto é referenciado pelo processador, este é copiado para a cache com o custo de um acesso à memória. As próximas referências ao objeto vão encontrá-lo na cache, ao custo de um acesso à cache. Quando o processador encontra uma cópia de um objeto na cache, ocorre um *acerto*. Quando o processador não encontra uma cópia de um objeto na cache ocorre uma *falta*. A *taxa de acerto* $\mathcal{A}$ na cache é definida como o *número total de acertos dividido pelo total de referências à cache*. O complemento da taxa de acertos é a *taxa de faltas* $\mathcal{F}$, definida como $\mathcal{F} = 1 - \mathcal{A}$. O *tempo médio de acesso à memória* T_M, em ciclos de relógio do processador, é

$$T_M = T_c \cdot [\mathcal{A} + T_m \cdot (1 - \mathcal{A})], \qquad 1 \leq T_c \leq 10,\ 10 \leq T_m \leq 50$$

onde T_c é o tempo de acesso à cache e T_m é o tempo de acesso à memória DRAM, em ciclos do processador. Quanto maior for a taxa de acertos, mais o tempo médio de acesso à memória se aproxima do tempo de acesso da cache. Quanto maior a localidade exibida por referências a código e dados, maior a taxa de acertos.

As caches são organizadas em vários *blocos*, cada bloco contendo de 1 a 16 palavras de 32 ou 64 bits. Quando o processador referencia uma palavra e ela não se encontra na cache, a palavra faltante e as demais que pertencem ao mesmo bloco são copiadas da memória para a cache. Blocos de mais de uma palavra são vantajosos porque o princípio da localidade espacial indica que palavras vizinhas à palavra que falta na cache serão também referenciadas no futuro próximo. Quando isso ocorrer, as palavras vizinhas já terão sido copiadas para a cache.

As operações de E/S envolvem principalmente a movimentação de dados entre periféricos e memória e a modificação e/ou inspeção simples dos dados (*e.g.* cálculo de paridade). Estas operações de E/S exibem pouca localidade porque a movimentação de dados envolve somente uma leitura seguida de uma escrita de cada octeto/palavra movido/a. No tratamento de E/S associado à comunicação via rede, a taxa de transferência entre periféricos e memória é limitada pelo mínimo dentre (1) a velocidade com que o processador executa o código que faz a transferência e (2) o tempo de acesso à memória.

As operações de E/S necessitam do envolvimento do SO e causam o re-escalonamento de processos por causa de interrupções e atendimento das requisições dos processos. A Figura 6.3 mostra um cenário onde o processo $\mathcal{P}_1$ solicita uma operação `tx` de E/S, representada por 'ES(tx)'. Esta operação será executada por um programa controlador de periférico (PCP).

Quando o PCP completa o tratamento da operação solicitada, ele suspende sua própria execução e o SO escolhe o processo $\mathcal{P}_2$ para executar. Quando o periférico completar o serviço solicitado por $\mathcal{P}_1$, ele vai interromper o processador. O tratador da interrupção interr( ) atende ao periférico, possivelmente transferindo dados, e ao concluir, o processador retoma a computação do processo $\mathcal{P}_2$. Após um certo tempo, o processador volta a executar o processo $\mathcal{P}_1$ a partir do ponto onde o PCP suspendeu sua execução, e o PCP, após completar sua tarefa, retorna ao proceso $\mathcal{P}_1$.

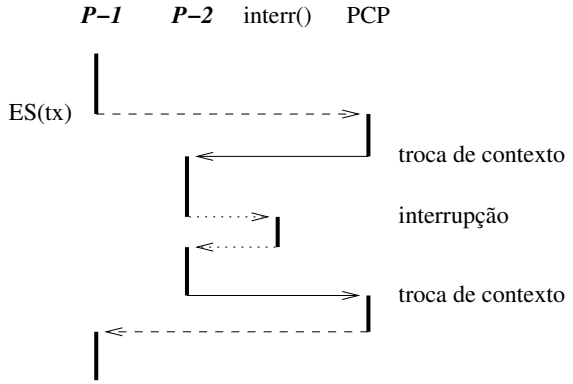


Figura 6.3: Trocas de contexto causadas por operações de E/S.

Ocorrem duas trocas de contexto entre os processos $\mathcal{P}_1$ e $\mathcal{P}_2$. Tanto a execução do PCP quanto a do tratador de interrupção provocam uma mudança na seqüência das instruções que estavam sendo executadas. Dependendo da implementação, uma troca de contexto invalida uma fração considerável do conteúdo da cache. Uma interrupção causa poluição na cache porque o código do tratador e seus dados ejetam da cache objetos úteis ao processo que foi interrompido. O processo que re-inicia sofre uma elevada taxa de faltas porque parte de seu conjunto de trabalho foi ejetado pelo tratador de interrupção.

Como mostra a Figura 6.2, a maioria dos microprocessadores de alto desempenho possui uma fila de escrita entre a cache e a memória. Esta fila tem a finalidade de esconder parte do custo das escritas em memória. Quando o processador atualiza uma palavra na cache, ao invés de ficar bloqueado esperando que o novo valor seja propagado ao longo da hierarquia de memória, o controlador da cache insere o valor atualizado na fila de escrita, juntamente com seu endereço em memória. Quando chegar à cabeça da fila, a atualização será efetuada na memória DRAM. O ganho em desempenho provido pela fila de escrita advém da superposição que ocorre entre o progresso da atualização na fila de escrita e as referências de leitura que ocorrem depois dela. Na maioria das vezes, a atualização é inserida na fila sem que o processador fique bloqueado esperando pela escrita em memória. Quando ocorrerem muitas atualizações num curto intervalo de tempo, a fila fica cheia, e uma nova atualização bloqueia o processador até que a atualização da cabeça da fila complete na memória DRAM. As filas são normalmente implementadas com capacidade para 1 a 16 registros de atualização (pares <endereço, valor>). Note que operações de E/S que envolvem cópias ou movimentação de muitos octetos enchem fila de escrita e fazem com que o processador passe a operar na velocidade da memória DRAM.

Dada a disparidade nas velocidades de operação entre processadores e memória DRAM, a memória cache é um componente indispensável de qualquer computador de uso geral. Contudo, como exposto acima, caches são ineficazes em operações de E/S.

6.3.1 Entrada e Saída Programada

A técnica de tratamento de periféricos por *entrada e saída programada* é eficiente para dispositivos de baixa velocidade como teclado e apontadores. Quando esta técnica é empregada, o processador efetua as transferências entre memória e periférico. Enquanto espera pela ocorrência de uma operação de E/S, o processador deve testar o estado do periférico, como mostra o fragmento de código abaixo. A Seção 3.5.2 contém um exemplo mais completo de código baseado nesta técnica.

```
for (i=0; i < TAM; i++) {
   while (Interf.status != PRONTO)
      ;
   M[i] = Interf.dados;      /* lê do periférico e escreve em memória */
}

# linguagem de máquina equivalente a M[i] = Interf.dados;
load  reg, dados(idx)       # reg <- Interf[dados] --lê periférico
store reg, 0(ptr)           # M[ptr] <- reg --escreve em memória
```

A Figura 6.4 mostra o percurso dos dados quando da execução do código mostrado acima. Quando o adaptador de interface de rede detecta a chegada de uma mensagem, este interrompe o processador (1). O tratador da interrupção vai interagir com o periférico e transferir a mensagem recebida do periférico para a memória (2) e (3). As cópias efetuadas pelo processador expulsam blocos da cache que estavam sendo utilizados pelo processo que foi interrompido.

A arquitetura 80x86 possui instruções especiais para referenciar periféricos. Estas instruções usam um espaço de endereçamento reservado para operações de E/S, logicamente disjunto do espaço de endereçamento de programas e dados. Nesta arquitetura, os periféricos são *mapeados como E/S*. Isso traz algumas dificuldades na programação destes sistemas, como discutido na Seção 3.5.2.

Na maioria das arquiteturas de microprocessadores recentes, os periféricos são mapeados em uma porção do espaço de endereçamento, isto é, os periféricos são *mapeados como memória*. Assim, não há diferença lógica entre uma leitura de um registrador de periférico e de uma posição de memória. Embora não haja diferença do ponto de vista lógico, os periféricos ainda residem no barramento de E/S onde as referências são mais custosas que referências à memória.

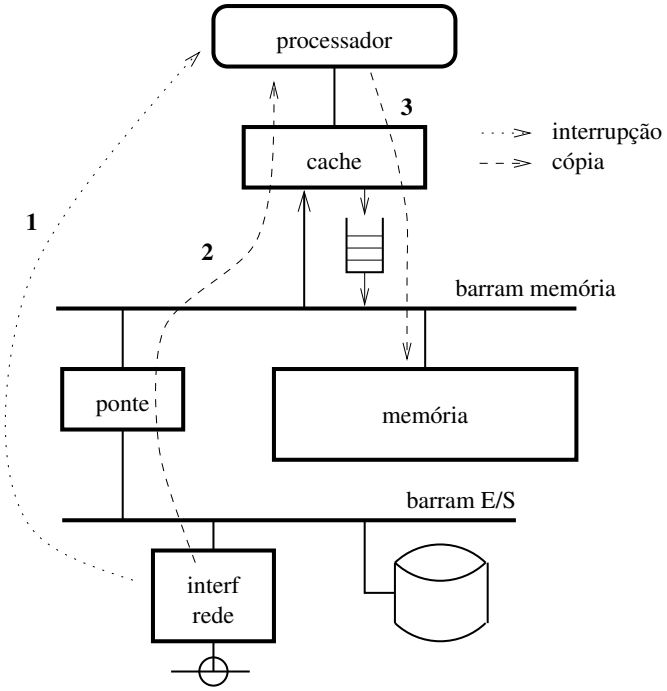


Figura 6.4: Transferência por E/S programada.

Do ponto de vista de redes de alta velocidade, esta arquitetura introduz os seguintes problemas: (1) nas operações de entrada (recepção), as leituras ocorrem na velocidade do barramento de E/S enquanto que as escritas ocorrem na velocidade da memória se fila de escrita ficar cheia. (2) nas operações de saída (transmissão) o caminho é revertido: as leituras ocorrem na velocidade da memória, por causa de faltas na cache, e as escritas ocorrem na velocidade do barramento de E/S. Os acessos a periféricos podem desviar a cache para evitar inconsistências. Se as referências são através da cache, após uma leitura ou escrita em registradores de periférico, os blocos apropriados da cache devem ser invalidados para manter a consistência entre a cópia na cache e o registrador do periférico.

Embora ocorram os problemas mencionados acima, as transferências por E/S programada são muito eficientes para mensagens pequenas. Os custos envolvidos são relativamente baixos e facilmente amortizados com mensagens de uns poucos octetos.

6.3.2 Acesso Direto à Memória

Quando as transferências entre periférico e memória envolvem volumes relativamente grandes de dados, a técnica de E/S programada se torna ineficiente por ocupar muito do tempo do processador com transferências de dados. A alternativa consiste na implementação de um circuito especializado para executar o laço de programa que faz as transferências. Esta técnica é chamada de *acesso direto à memória* (ADM) porque as transferências ocorrem quase que sem interferência do processador. A Figura 6.5 mostra um controlador de ADM implementado na ponte entre os barramentos de memória e de E/S. Quando uma transferência deve ser efetuada, o processador programa o circuito controlador de ADM para mover os dados entre periférico e memória (1). O controlador move os dados para os endereços especificados sem a interferência do processador (2). Quanto completa a transferência, o controlador de ADM interrompe o processador (3).

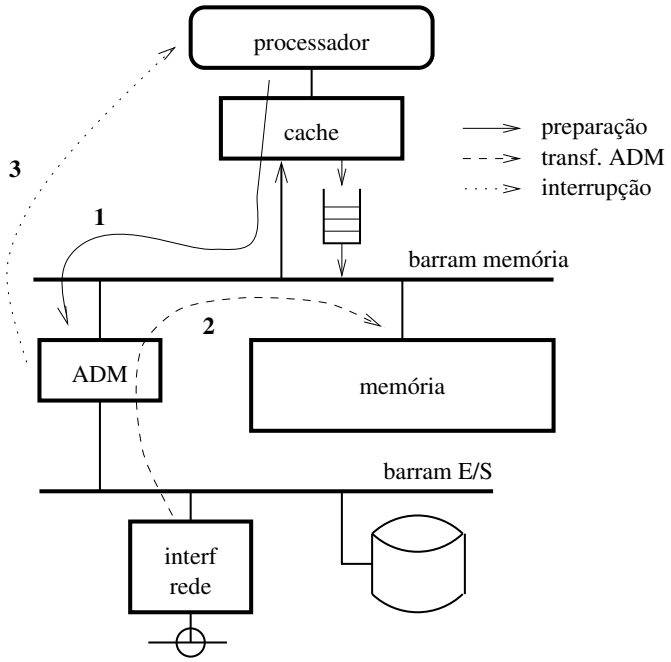
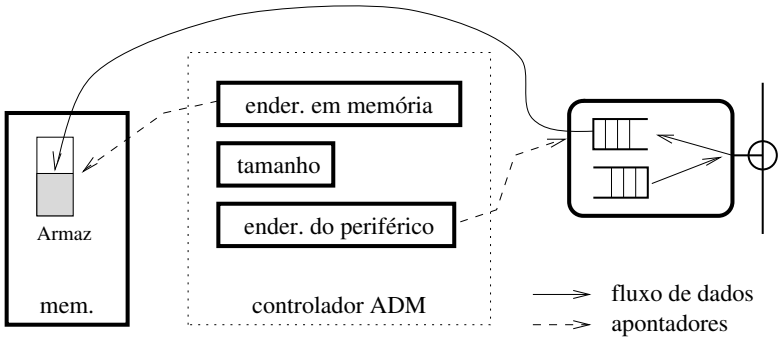


Figura 6.5: Transferência por ADM.

A Figura 6.6 mostra uma versão simplificada de um controlador de ADM, efetuando uma transferência entre periférico e memória. O controlador contém três registradores. O primeiro aponta o endereço do armazenador em memória onde os dados devem ser depositados. O segundo registrador é programado com o número de unidades a ser transferidas (*i.e.,* o tamanho da transferência). O terceiro registrador aponta para a fonte dos dados, neste caso, o adaptador de interface de rede. O processador deve carregar os registradores do controlaor de ADM com os endereços apropriados e comandar o início da transferência. O controlador de ADM executa o laço que move os dados e interrompe o processador quando completa sua tarefa.

Transferências por ADM são muito eficientes para mensagens grandes. O processador só interfere na programação do controlador e quando da interrupção que sinaliza o final da transferência. Enquanto aquela está ocorrendo, o processador pode continuar executando instruções, possivelmente de um outro processo. Como mostra a Figura 6.5, enquanto está havendo uma transferência por ADM, o barramento de memória transporta dados entre memória e periférico. Se o processador tentar um acesso à memória, este pode ficar bloqueado esperando pela oportunidade de usar o barramento até que a transferência por ADM complete. Por esta razão, a maioria dos controladores de ADM (e barramentos) implementa transferências em várias séries de 4 a 32 ciclos de barramento. Assim, o processador fica bloqueado apenas durante uma série e não durante toda a transferência. Se a taxa de acertos na cache é alta, processador e controlador de ADM podem operar em paralelo sem que um interfira com as atividades do outro.




```
/* inicialização pelo processador */
ADM->mem = &(Armaz[0]);
ADM->perif = &(Interf[RX_REDE]);
ADM->tam = Interf.tam;
/* transferência ADM */
for ( ; ADM->tam > 0; ADM->tam--)
   (ADM->mem)++ = ADM->perif;
/* interrompe processador */
```

Figura 6.6: Controlador de ADM.

Os protocolos de rede são baseados no encapsulamento de mensagens onde cada nível da pilha de protocolos acrescenta um cabeçalho a cada unidade de dados de protocolo recebida do nível imediatamente acima. Se o controlador de ADM suporta *coleta* e *dispersão* de dados, os cabeçalhos dos vários níveis podem ser preparados em armazenadores separados. Quando a quadro completo vai ser transmitido pelo adaptador de interface, o processador passa ao controlador de ADM uma lista encadeada com os endereços e tamanhos dos vários armazenadores em memória que compoẽm o quadro, como mostra a Figura 6.7. Na recepção, o desmembramento do quadro recebido deve ser efetuado pelo processamento dos vários níveis e protocolos, a não ser que o adaptador de rede possa identificar e separar os vários componentes das mensagens.

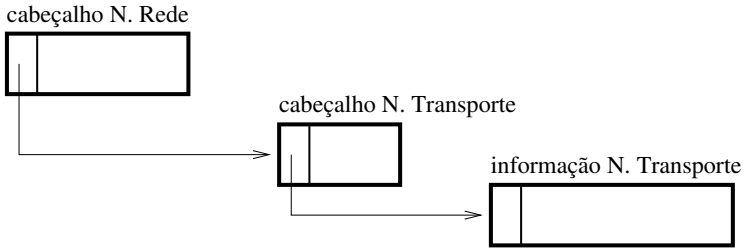


Figura 6.7: ADM com coleta e dispersão.

Infelizmente, o desempenho de controladores de ADM também sofre com problemas relacionados ao sistema de memória. Como o controlador de ADM fica localizado na ponte entre os barramentos, ou nos próprios periféricos, sua programação se dá, no melhor caso, na velocidade da memória, e no pior caso, na velocidade do barramento de E/S. Além disso, podem ser necessários muitos parâmetros de operação (*e.g.* listas de pares <endereço, tamanho>). Como citado acima, os ciclos de barramento de acesso direto à memória impedem que processador e controlador da cache acessem a memória, bloqueando o processador. Finalmente, há o custo associado ao tratamento de interrupções, discutido em mais detalhe abaixo.

Consistência entre caches e memória Existe a possibilidade de que a versão de um objeto na cache possa ficar inconsistente com aquela em memória. Isso pode ocorrer quando um registrador de estado de um periférico é lido pelo processador, ficando uma cópia na cache. Muitos periféricos atualizam seus registradores após uma leitura pelo processador. A menos que sejam tomados cuidados especiais, a versão na cache pode estar diferente daquela no registrador do periférico. Esta situação é mostrada na Figura 6.8, onde Y' é mais recente que Y.

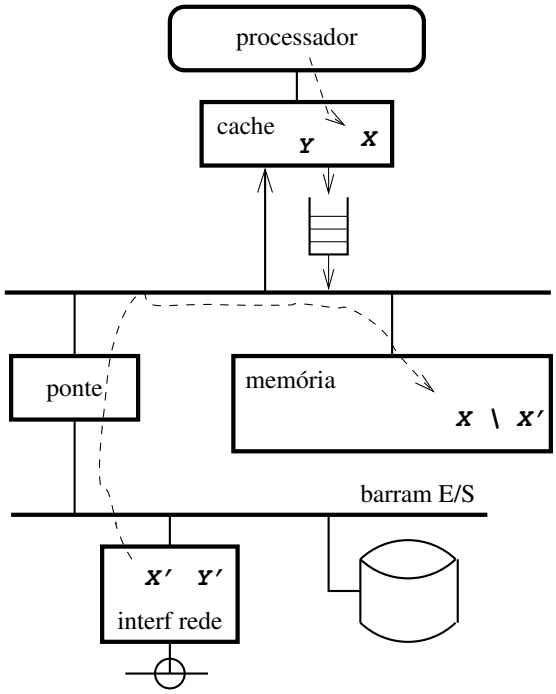


Figura 6.8: Inconsistência entre cache e memória.

Inconsistências podem ainda ser causadas quando as operações de E/S envolvem acesso direto à memória. Como ainda mostra a Figura 6.8, a cache contém uma versão da variável X, lida recentemente pelo processador. Ocorre uma transferência por ADM e o valor em memória é alterado para X'. A fim de impedir que inconsistências ocorram, as operações de E/S devem forçar a consistência entre cache e memória. Em operações de saída (transmissão), as versões mais recentes, na cache, devem ser empurradas para memória. Em operações de entrada (recepção), as cópias na cache devem ser atualizadas com os novos valores em memória. Outra solução consiste em o SO invalidar as cópias na cache que possam ter ficado caducas após uma operação de E/S. Isso pode envolver um número grande de instruções para escolher quais blocos da cache devem ser invalidados. Uma terceira alternativa consiste em projetar a cache para que ela observe as transações no barramento de memória e invalide ou atualize automaticamente quaisquer blocos que contenham cópia de objetos que atravessam o barramento de memória. Processadores de projeto recente como Alpha, Pentium e PowerPC contém os mecanismos necessários para manter cache e memória consistentes.

6.3.3 Interrupções

Interrupções servem para avisar o processador da ocorrência de eventos externos a si próprio e associados à periféricos. Geralmente, a interrupção é detectada pelo processador antes do início da execução de uma nova instrução. Antes de iniciar o tratamento de uma interrupção, o processador deve drenar seu "pipeline" para evitar interferências indevidas entre o processo interrompido e o tratador de interrupções; os registradores usados pelo tratador de interrupção

devem então ser salvos na pilha. O tratador atende ao periférico que causou a interrupção, talvez efetuando uma transferência de dados. Feito isso, os valores salvos na pilha são re-carregados nos registradores e o processo interrompido volta a executar.

O salvamento dos registradores causa muitas escritas/leituras na/da pilha, a maioria delas ocorrendo na velocidade da memória porque são faltas na cache, ou porque a fila de escrita fica cheia. O código e os dados do tratador de interrupção causam poluição na cache expulsando blocos que pertencem ao conjunto de trabalho do processo interrompido. Após uma interrupção, a recuperação do conjunto de trabalho do processo interrompido na cache pode se extender por até 10.000 ciclos do processador [MB91, ALBL91].

Dado seu elevado custo, interrupções devem ser usadas com critério. Com o fim de minimizar os custos do atendimento da interrupção, o tratador só deve terminar, e retornar ao programa interrompido, se não houver mais trabalho a fazer. Por exemplo, o tratador deve examinar o estado de todos os periféricos de uma mesma classe e atender a todos que necessitem de atenção. Quando o tratador atende a periféricos relativamente lentos, ao invés de cada periférico gerar uma interrupção, o tratador pode ser invocado periodicamente para examinar o estado de vários periféricos, atendendo àqueles que necessitem de atenção.

Quando um processo solicita ao SO o envio uma mensagem, algumas implementações provocam uma interrupção após completar-se a transmissão. Isso deve ser evitado porque o trabalho do atendimento desta interrupção será desperdiçado, a menos que o processo já esteja em condições de enviar nova mensagem. Antes de enviar uma mensagem, o SO pode testar a disponibilidade do adaptador de interface para nova transmissão a um custo mínimo quando comparado ao custo da interrupção. No caso da recepção, devem ser evitadas interrupções enquanto a fila de recepção estiver não-vazia. O tratador da interrupção de recepção deve transferir todas as mensagens recebidas e com tratamento pendente para armazenadores em memória antes de terminar sua execução.

6.4 Interações entre Memória Virtual e E/S

Em processadores de 32 bits, o espaço de endereçamento virtual permite referências a qualquer um dentre 2^{32} bytes de memória (ponteiros de 32 bits). Geralmente, os computadores contém de 16 Mbytes a 1 Gbytes de memória física (DRAM). A função do sistema de memória virtual (SMV) é prover ao programador a abstração de um espaço de endereçamento físico do mesmo tamanho do espaço de endereçamento virtual. Para implementar esta abstração, a memória virtual é dividida em *páginas* de 1 a 64 Kbytes e a memória física é dividida em *quadros* com o mesmo tamanho que as páginas. O mapeamento entre páginas do espaço virtual e quadros do espaço físico é feito pela *tabela de páginas*. Através da tabela de páginas, um processo pode localizar uma página virtual em um quadro na memória física.

Uma referência à memória passa por duas fases. Na primeira, o processador descobre onde o objeto referenciado reside –em que quadro na memória física. Na segunda fase, o processador efetua a referência ao objeto no quadro apropriado da memória física. Para evitar a necessidade de uma referência adicional à tabela de páginas para cada acesso à memória (*i.e.,* dois acessos à memória para cada referência), os processadores empregam uma cache que contém somente mapeamentos de endereços virtuais em físicos, ou seja, esta cache só contém cópias de elementos da tabela de páginas. Esta cache é chamada de *cache de mapeamento de endereços* (CME) ("translation lookaside buffer", TLB). Como numa cache para dados ou instruções, quando o processador emite uma referência, se o mapeamento se encontra na CME ocorre um *acerto na CME*. Caso contrário, ocorre uma *falta na CME* e o mapeamento faltante deve ser copiado da tabela de páginas para a CME. Esta cópia tem um custo de 20 a 100 ciclos do processador. Um acerto na CME resolve um mapeamento em 1/2 ciclo do processador. A outra metade do ciclo é usada para o acesso à cache.

A Figura 6.9 mostra a organização de um sistema de memória virtual. Os endereços emitidos pelo processador são traduzidos pela CME antes de aplicados à cache e memória física. A memória física contém os espaços de endereçamento de três processos, dois de usuário (Us-1 e Us-2) e um do SO. Cada espaço de endereçamento contém a sua própria tabela de páginas (TP). Apesar de a figura mostrar os quadros de cada processo agrupados num retângulo, na prática os quadros ficam espalhados a esmo pela memória física.

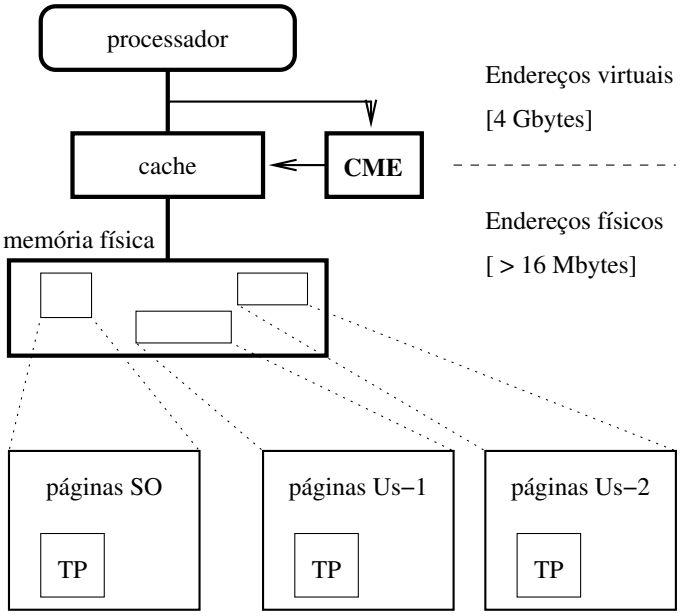


Figura 6.9: Modelo de sistema de memória virtual.

6.4.1 Gerenciamento de Memória

Muitos dos problemas e ineficiências associados a transferências de dados internas aos computadores são causadas pelos mecanismos de gerenciamento de memória. As causas destes problemas são de duas naturezas. Uma é associada ao baixo nível de abstração das primitivas de transmissão de dados. O segundo grupo de causas é associado à implementação do sistema de gerenciamento de memória, que é ineficiente porque foi projetado para periféricos lentos ao invés de redes rápidas.

Num ambiente de programação como o Sistema Operacional Linux, o programador deve alocar espaço em armazenador(es) de tamanho suficiente para as mensagens a ser trocadas através da rede. Os aplicativos montam as mensagens por transmitir nos armazenadores e invocam funções do SO para enviar as mensagens aos seus destinatários. Na recepção, o SO monta as mensagens recebidas nos armazenadores disponibilizados pelo aplicativo antes de entregá-las. Se as transmissões usam a interface de soquete, os aplicativos podem escrever nos, e ler dos, soquetes como se estes fossem arquivos. A primitiva para o envio (escrita) de uma mensagem é mostrada abaixo [Bac86, CS93, CS94].

```
size_t write(int fd, const char *armaz, size_t count)
       /* fd: identificador da conexão */
       /* armaz: armazenador */
       /* count: tamanho da mensagem */
```

Os parâmetros de `write()` são (a) um identificador de conexão lógica, (b) um armazenador alocado pela aplicação (`armaz[]`), e (c) a capacidade do armazenador. O processo da aplicação pode escrever em `armaz[]` imediatamente após o retorno de `write()` porque `armaz[]` é um

armazenador comum. O tamanho deste armazenador, `count`, não pode ser arbitrariamente grande porque o processo pode bloquear por falta de espaço nos armazenadores internos ao soquete. Isso ocorre porque, como o usuário pode escrever em `armaz[]`, o SO deve ser cauteloso e copiar o conteúdo de `armaz[]` para outro armazenador, este de acesso restrito ao SO. Veja as páginas de manual de `write()` e `read()` para mais detalhes.

A primitiva que efetua o envio de uma mensagem para uma máquina é `send()`, cuja síntaxe é mostrada abaixo. Os parâmetros são (a) o identificador do destinatário, (b) um armazenador que contém a mensagem (`msg[]`), (c) seu tamanho (`tam`), e (d) um indicador do tipo da mensagem. O armazenador pertence ao processo do aplicativo e pode, portanto, ser atualizado a qualquer instante por ele. Da mesma forma que com `write()`, o SO deve copiar o conteúdo de `msg[]` para um armazenador em seu espaço de endereçamento. O processo que invoca `send()` pode ficar bloqueado se o armazenador do SO não for grande o suficiente para conter a mensagem. Além disso, as mensagens não podem ser arbitrariamente longas porque elas devem ser transmitidas atomicamente pelo protocolo de transporte. `send()` pode bloquear por falta de espaço no armazenador ou porque a rede por onde o pacote vai ser transmitido está congestionada. Veja as páginas de manual de `send()` e `recv()` para mais detalhes.

```
int send(int s, const void *msg, int tam, unsigned int flag)
    /* s: identificador do destinatário */
    /* msg: armazenador */
    /* tam: tamanho da mensagem */
    /* flag: tipo da mensagem */
```

Esta implementação das primitivas de comunicação é pouco abstrata e permite que o usuário enxergue muitos detalhes da sua implementação. Idealmente, a interface de programação deveria ser mais abstrata e regular. Do ponto de vista da implementação de protocolos de comunicação, a interface deve permitir a agregação de armazenadores para facilitar o encapsulamento de unidades de dados dos protocolos (*e.g.* cabeçalhos e informação) e a concatenação de fragmentos de mensagens recebidas. Ainda, a implementação deve evitar cópias desnecessárias e permitir o compartilhamento de mensagens entre vários processos. Estas características são discutidas em mais detalhe no que segue.

6.4.2 Custos Associados ao Sistema de Memória

A transmissão de uma mensagem (ou segmento TCP) através dos serviços implementadas pelos protocolos da família TCP/IP envolve os seguintes passos: (1) o remetente estabelece uma conexão tipo soquete com o destinatário; (2) o remetente envia uma ou mais mensagens invocando a função `send()`; (3) a conexão é terminada pelo remetente quando não existem mais mensagens a transmitir. Pelo lado do destinatário, este (a) aceita o pedido de conexão e fica a espera de mensagens; (b) recebe as mensagens através da função `receive()`; (c) deixa de esperar por mensagens quando a conexão é terminada.

O envio de uma mensagem pelo programa de aplicação envolve os seguintes passos, mostrados na Figura 6.10.

1. a aplicação preenche um armazenador com a mensagem;
2. a aplicação executa `send()` e passa controle ao SO;
3. o SO copia a mensagem para seu espaço de endereçamento;
4. o SO ativa o adaptador de rede, preparando-o para transmissão;
5. o SO copia a mensagem para o adaptador de rede, que a transmite.

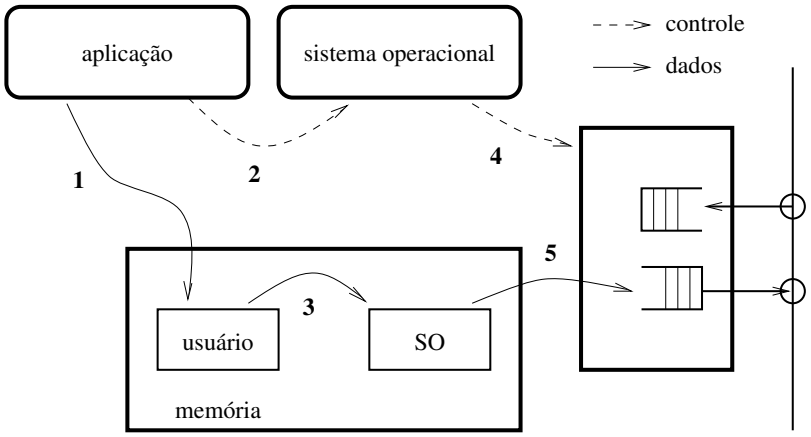


Figura 6.10: Eventos na transmissão de uma mensagem.

Como discutido na Seção 6.3, o barramento de memória é o gargalo para as operações de E/S. Dada a seqüência de ações necessárias para o envio de uma mensagem, a Figura 6.11 mostra as ações que resultam em travessias do barramento de memória.

1. a aplicação prepara o armazenador com a mensagem por transmitir, copiando de uma ou mais variáveis para um armazenador contínuo em memória [1];
2. o SO lê a mensagem para copiá-la em seu armazenador [2];
3. o SO escreve a mensagem em seu armazenador [3];
4. o SO lê todo o armazenador para computar a paridade da mensagem [4];
5. o SO lê a mensagem para movê-la para o adaptador de interface [5];
6. o SO escreve a mensagem no registrador de transmissão do adaptador de rede [6].

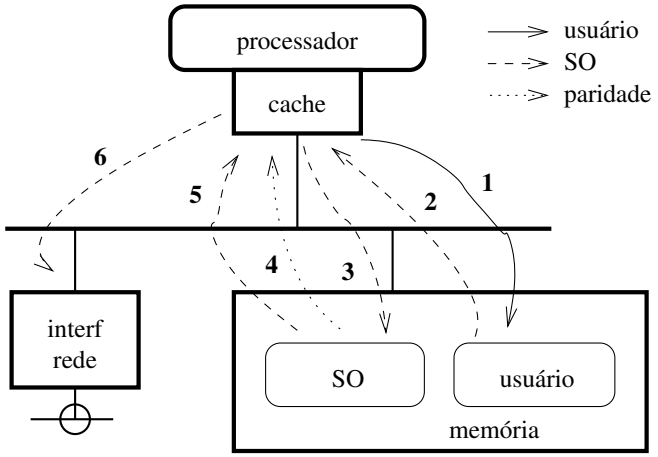


Figura 6.11: Percurso dos dados na transmissão.

Esta implementação exibe baixo desempenho porque os processadores são mais rápidos que a memória DRAM e portanto, as cópias são muito caras. Algumas das travessias do barramento de memória podem ser evitadas se houver uma melhor integração entre o projeto dos adaptadores de interface de rede, o sistema de gerenciamento de memória pelo SO e o sistema de proteção do SO.

Várias otimizações são possíveis e elas se dividem em dois grupos. As otimizações do primeiro grupo objetivam eliminar cópias desnecessárias através do uso de ADM, do cômputo de paridade pelo adaptador de interface e de um gerenciamento mais eficiente dos armazenadores.

O segundo grupo de otimizações objetiva reduzir as interações entre processos de usuário e o SO. Estas otimizações são discutidas no que segue.

Otimização I Como mostra a Figura 6.11, uma das travessias [5] ou [6] pode ser eliminada se um controlador de ADM for empregado para transferir mensagens entre o armazenador do SO e o adaptador de interface de rede.

Otimização II O número de travessias pode ser reduzido para cinco se a paridade da mensagem for computada enquanto o SO copia a mensagem para seu espaço de endereçamento, eliminando a travessia [4]. Isso também pode ser obtido se o controlador de ADM computar a paridade, eliminando as travessias [4], e [5] ou [6].

Otimização III O número de travessias pode ser reduzido para quatro se o armazenador for compartilhado entre o programa de aplicação e SO, eliminando cópia entre armazenadores e as travessias [2] e [3].

Otimização IV Se o armazenador do SO for localizado no adaptador de interface, as travessias [3] e [5] podem ser eliminadas. Se a transferência entre armazenador de usuário e adaptador de rede usar E/S por programa, as travesias [2] e [6] são necessárias. Se a transferência for com ADM, são necessárias as travessias [1] e a da transferência por ADM, equivalente a [2]+[6].

Otimização V Finalmente, somente uma travessia é necessária [1] se o armazenador é localizado no adaptador de interface e é compartilhado entre aplicação e SO.

O armazenador do SO é um recurso limitado que deve ser gerenciado com parcimônia. Se o armazenador fica no adaptador de interface, o gerenciamento do uso do armazenador pode se tornar o ponto de estrangulamento do sistema de comunicação. Neste caso, o adaptador de interface deve ser capaz de demultiplexar as conexões lógicas estabelecidas pelo Nível de Transporte, entregando as mensagens recebidas aos processos que são seus destinatários. O adaptador deve ser capaz de examinar os campos de cada quadro recebido, validar o conteúdo do quadro e determinar seu destinatário.

Um Adaptador de Interface Eficiente: Afterburner

O Afterburner é um adaptador de interface que permite uma única travessia de barramento, entre aplicativo e armazenador do SO [DWB+93]. Este adaptador foi desenvolvido para estações de trabalho da série HP 700 e permite a ligação eficiente entre aplicativo e o Nível de Enlace de uma rede de alta velocidade. A Figura 6.12 mostra um diagrama em blocos do Afterburner. A interface com o processador se dá pelo barramento de vídeo da estação, onde o adaptador é mapeado como memória. A interface com o Nível de Enlace é independente da tecnologia da rede, com implementações para modo de transferência assíncrono, HIPPI e Jetstream.

O armazenador é implementado com 1 Mbyte de memória de vídeo (VRAM). Este armazenador é dividido em N blocos e o tamanho dos blocos pode ser configurado entre 2 e 64 Kbytes. A figura mostra 256 blocos de 4 Kbytes cada. A memória VRAM possui uma porta paralela, acoplada ao barramento de vídeo, e duas portas seriais, uma ligada ao circuito de transmissão e a outra ao circuito de recepção do enlace de dados. Os três módulos de controle efetuam as ações necessárias para transmitir e receber mensagens.

Transmissão Antes de transmitir uma mensagem, o processo remetente deve reservar um bloco de VRAM para a mensagem. Para tanto, um descritor de mensagem é removido da fila TXfree pelo processo e preenchido com tamanho da mensagem, endereço do destinatário, tipo

de protocolo e outras informações. O remetente preenche o bloco de VRAM com a mensagem propriamente dita e insere o descritor na fila TXrdy. Quando este descritor chegar à cabeça da fila, o controlador de enlace serializa o conteúdo do bloco e o insere no circuito de transmissão. Quanto a mensagem é transmitida, o controlador de transmissão insere o descritor na fila TXfree para que o bloco possa ser usado novamente.

Recepção Quando o controlador de enlace detecta uma mensagem no circuito de recepção, este remove um descritor de mensagem da fila RXfree e preenche o bloco com a mensagem recém chegada. Após examinar o cabeçalho da mensagem, o controlador de enlace preenche o descritor com o nome do processo destinatário insere o descritor na fila RXrdy. Quando o descritor chegar à cabeça da fila, o controlador de recepção interrompe o processo destinatário. Este copia a mensagem do bloco para um armazenador do aplicativo e insere o descritor na fila RXfree.

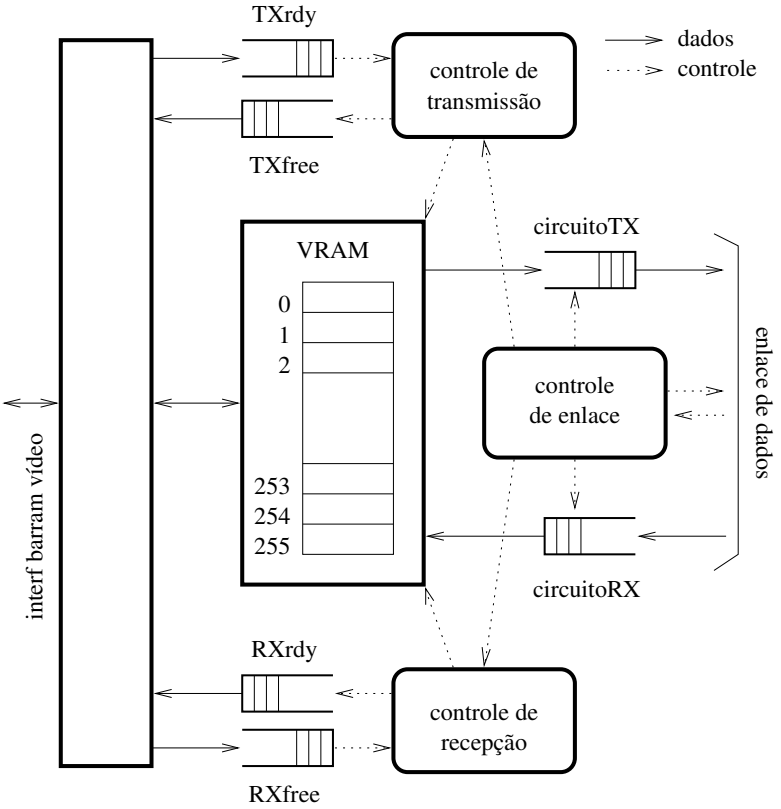


Figura 6.12: Afterburner.

O tamanho dos blocos pode ser configurado para conter até 64 Kbytes de dados. Para permitir o tratamento de mensagens maiores que isso, os descritores possuem um bit de controle chamado mais. Se a mensagem é maior que um bloco, o remetente remove tantos descritores quanto necessário, preenche os blocos correspondentes e insere os descritores na ordem correta na fila TXrdy. O remetente ativa o bit mais em todos os descritores exceto o último. Mesmo que os blocos não sejam adjacentes em VRAM, o controlador de transmissão remete o conteúdo dos blocos na ordem em que eles foram inseridos na fila TXrdy. O tratamento de mensagens longas pelo controlador de recepção é similar.

A quantidade de blocos alocada para transmissão ou recepção é determinada quando o adaptador é inicializado. Dado o elevado custo do tratamento de interrupções, o adaptador pode ser programado para interromper o processador de forma a minimizar o desperdício de

tempo. O Exercício 6.3 examina as possibilidades.

6.4.3 Eliminação de Cópias Através do Sistema de Memória Virtual

Em sistemas operacionais da família Unix, a abstração de processo engloba as noções de domínio de proteção e espaço de endereçamento. Um *domínio de proteção* identifica os recursos que podem ser usados por um processo e os recursos deste proceso que *não* podem ser acessados pelos demais processos no sistema. Um *espaço de endereçamento* contém todas as posições de memória alcançáveis através de mecanismos como apontadores de C ou Pascal, contidas no domínio de proteção do processo.

As cópias efetuadas durante operações de E/S cruzam as fronteiras entre os domínios de proteção do SO e aplicativos. Estes cruzamentos de fronteira são caros porque envolvem traduções de endereçamento físico–virtual e verificação das permissões de acesso. As operações mostradas na Figura 6.10 (pág. 204) são geralmente implementadas por serviços do sistema operacional. Cada vez que um destes serviços é invocado, há uma mudança no nível de proteção com que o serviço é executado (de usuário para sistema), e o retorno ao nível de proteção mais baixo quando o serviço é concluído. Estas mudanças de nível de proteção são custosas e reduzem a eficiência do sistema de comunicação.

Como discutido na Seção 6.4.2, a cópia do armazenador de usuário para o armazenador do SO pode ser evitada se houver compartilhamento do armazenador entre aplicativo e SO. Este compartilhamento é implementado através de mecanismos do sistema de memória virtual. As três técnicas usadas para permitir o compartilhamento de armazenadores são descritas nos próximos parágrafos.

Técnica I: Re-mapeamento de páginas de memória virtual Se os aplicativos alocam seus armazenadores de forma a que eles sejam alinhados em páginas virtuais, uma cópia pode ser evitada se o SO relocar as páginas de memória virtual com os armazenadores do espaço de endereçamento do aplicativo para o espaço do SO. Na transmissão, o SO transfere as páginas do espaço de usuário para o espaço do SO enquanto que na recepção o SO transfere páginas de seu espaço para o espaço do usuário [PAM94]. Uma característica importante do re-mapeamento de páginas virtuais é que o remetente perde o acesso às páginas porque elas são removidas do seu espaço de endereçamento. Portanto, o re-mapeamento de páginas virtuais tem a semântica da operação `move()` e não da operação `copia()`.

Técnica II: Cópia virtual A técnica de cópia virtual também é chamada de *copy-on-write* [FR86]. Nela, as páginas com armazenadores são compartilhadas entre o processo que produz a mensagem e o processo que a transfere ao destinatário através da interface de rede. Quando o processo produtor invoca a função `send()`, o SO cede algumas de suas páginas ao produtor e marca estas páginas como compartilhadas. Se o produtor não escreve nas páginas do armazenador compartilhado, o SO retoma estas páginas quando a mensagem é aceita pelo destinatário. Se o usuário escreve nas páginas do armazenador compartilhado, o SO faz uma cópia da mensagem em páginas de seu uso exclusivo. Ao contrário do re-mapeamento de páginas, a cópia virtual tem a semântica de `copia()`. Contudo, cópia virtual evita cópias se, e somente se, as páginas compartilhadas não são atualizadas.

Re-mapeamento de páginas e cópia virtual Estas duas técnicas para evitar cópias sofrem de algumas ineficiências. As operações necessárias para o re-mapeamento de uma página são: (1) atualizar os espaços de endereçamento dos dois processos; (2) atualizar as tabelas de páginas com mapeamento virtual-físico nos dois processos; (3) atualizar as tabelas de páginas com mapeamento físico-virtual nos dois processos; (4) atualizar a cache de mapeamentos de

endereços (CME); (5) dependendo da implementação, invalidar os blocos na cache que correspondem às páginas compartilhadas. Para mensagens pequenas, o custo dos acessos às tabelas de páginas, CME e caches pode ser maior que simplesmente copiar a mensagem entre os dois espaços de endereçamento.

Outra fonte de ineficiências advém do tamanho da unidade de transferência que é um múltiplo da página. Se a mensagem é um pouco maior ou menor que uma página ocorre fragmentação com páginas sub-utilizadas. Os fragmentos "vazios" das páginas devem ser preenchidos com zeros para garantir a privacidade entre os processos –esta operação tem, em média, um custo que equivale à metade de uma cópia. Outro problema decorre da maior competição entre os processos por páginas de memória física e o conseqüente aumento de transferências entre disco e memória ("swapping"). Por sua vez, esta competição aumenta o custo da alocação de páginas e do mapeamento entre memória virtual e física, bem como a competição por blocos da CME que também se intensifica.

Técnica III: Memória Virtual Compartilhada Esta técnica para eliminar cópias desnecessárias é baseada no compartilhamento estático de páginas de memória virtual entre dois ou mais processos. Um dos processos que desejam compartilhar armazenadores de mensagens aloca um número razoável de páginas para o compartilhamento e avisa aos demais processos de sua localização. Estas páginas são então adicionadas aos espaços de endereçamento de todos os "sócios" e passam a ser compartilhadas livremente por eles.

Esta técnica apresenta dois problemas. Todos os sócios das páginas devem ser conhecidos a priori, o que impede o compartilhamento dinâmico de armazenadores por um grupo variável de processos. O outro problema está associado à seguraça e privacidade. Todos os sócios tem acesso a todas as páginas durante todo o tempo, o que pode ser indesejável em certas circunstâncias.

Um Sistema de Gerenciamento de Memória Eficiente: Fbufs

Uma tendência recente é o uso de ambientes interativos distribuídos tais como teleconferência e vídeo sob demanda. Nestes ambientes ocorre o compartilhamento do tráfego recebido da rede entre vários processos: o processo–"browser" conversa com processo–gerenciador-de-janelas, que conversa com processo–gerenciador-de-áudio/vídeo, que conversa com servidor-de-interface-gráfica, que conversa com ...? As técnicas de implementação de gerenciamento de memória descritas acima são inadequadas para este padrão de uso.

Uma técnica que resolve o problema de compartilhamento múltiplo de uma forma eficiente é aquela empregada no sistema de "Fast Buffers" (fbufs) [DAPP93, Dru96]. Num sistema com fbufs, os armazenadores são implementados por tipo de dados abstrato e as operações sobre os armazenadores escondem a implementação do tipo abstrato. O usuário manipula *agregados* de fbufs e não os armazenadores diretamente. A Figura 6.13 mostra um agregado com três fbufs. O agregado contém uma árvore de apontadores de fbufs e novas folhas podem ser agregadas dinamicamente. Em especial, folhas mais à esquerda podem ser adicionadas de forma eficiente para facilitar encapsulamento de mensagens. Note que nem todo o espaço alocado a um fbuf contém informação. Isso permite edições bastante eficientes de mensagens de forma a que concatenações e cortes tenham um custo mínimo, e evitam cópias desnecessárias.

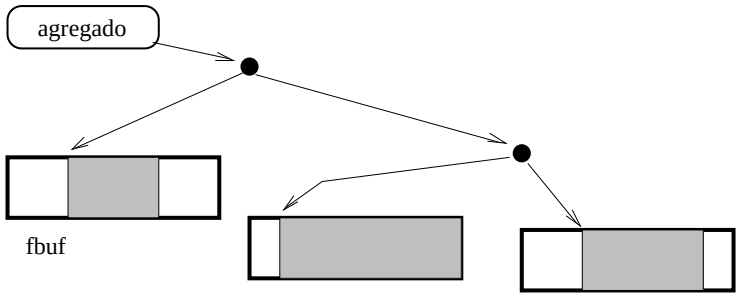


Figura 6.13: Agregado de fbufs.

Quando cópias de mensagens são necessárias, elas ocorrem através de re-mapeamento e compartilhamento de páginas virtuais. O único processo com direito de escrita num agregado é aquele que criou o agregado. Quando outro processo necessita alterar o conteúdo de um fbuf, a parte afetada é copiada para outro fbuf, de propriedade do processo que efetua a alteração. Os fbufs são alocados em uma mesma região do espaço de endereçamento de todos os processos. Os agregados são sempre alocados em espaço de endereçamento privativo a cada processo.

Numa situação onde vários processos compartilham de um fluxo de mensagens, o primeiro processo na seqüência aloca uma certa quantidade de fbufs e avisa aos demais sócios da sua localização. Este primeiro processo é chamado de *produdor* de mensagens e os demais são os *consumidores* de mensagens. Quando uma mensagem deve ser compartilhada, o produtor cria um agregado para contê-la e passa aos consumidores os endereços dos fbufs associados a esta mensagem. Cada consumidor aloca um agregado para a mensagem e liga o seu agregado aos fbufs. Uma vez que a mensagem é consumida, os agregados são destruídos em cada processo mas os fbufs permanecem associados ao grupo composto pelo produtor e consumidores. Dessa forma, nas próximas mensagens do fluxo, os fbufs já estarão disponíveis para uso imediato. Os fbufs ficam portanto numa cache, associada ao fluxo de mensagens. Esta seqüência de eventos é mostrada na Figura 6.14, onde um produtor e dois consumidores compartilham de um fluxo de mensagens.

Esta técnica de implementação do sistema de gerenciamento de memória é eficiente porque somente ocorrem cópias quando um processo consumidor altera um fbuf compartilhado. Senão, o compartilhamento ocorre através de uma área de memória especialmente reservada para compartilhamento múltiplo. Isso evita as operações custosas de re-mapeamento de páginas a cada mensagem transferida. Outra vantagem é a cache de fbufs. Uma vez estabelecido o trajeto do fluxo de mensagens, um certo número de fbufs é alocado dinamicamente para conter as mensagens do fluxo. Enquanto o fluxo perdurar, não são necessárias as operações de alocação e de-alocação de fbufs.

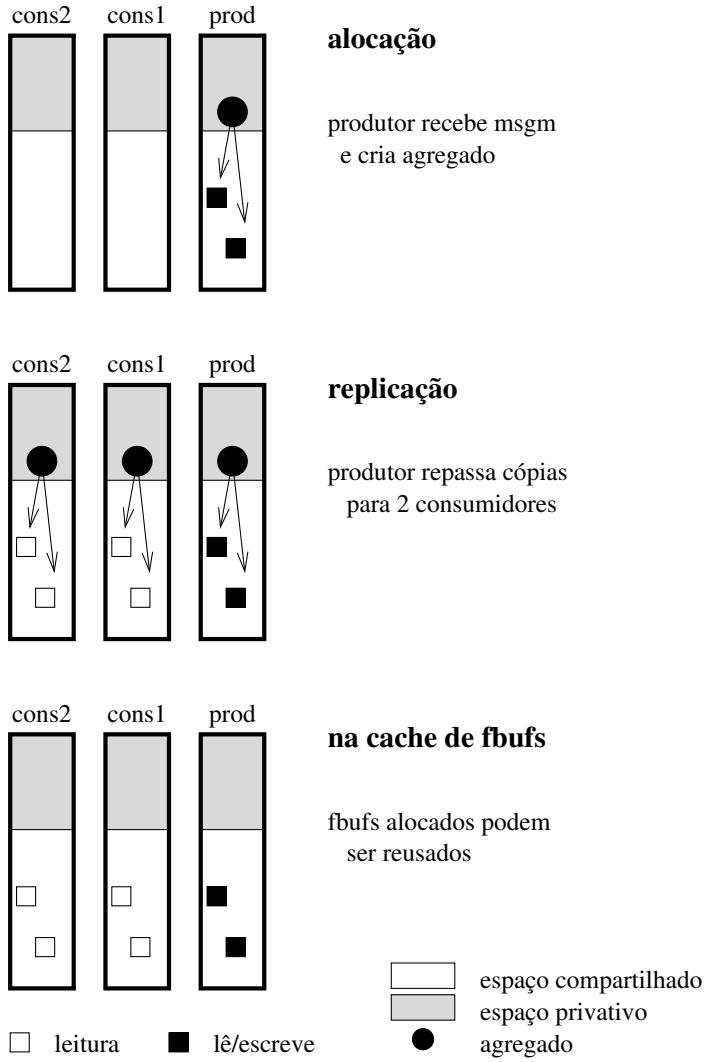


Figura 6.14: Compartilhamento com fbufs.

6.5 Interações com o Sistema Operacional

As operações de E/S necessitam do envolvimento do SO para que a funcionalidade associada à abstração de processo seja mantida de forma consistente. Do ponto de vista de dispositivos de alta velocidade, a participação do SO tem um custo elevado e geralmente incompatível com velocidades elevadas. O envolvimento do SO implica em trocas de contexto freqüentes e cópias desnecesárias entre espaços de endereçamento. Estas atividades tem conseqüências sérias no desempenho da hierarquia de memória e no sistema de memória virtual.

A maioria das implementações de protocolos de transmissão de dados consiste de código do sistema operacional que deve implementar as primitivas de envio e recepção de mensagens. Este código deve ser executado em modo previlegiado para garantir a integridade do sistema como um todo. As trocas de contexto entre modo usuário e modo previlegiado podem ser evitadas se o código das primitivas usadas mais freqüentemente for executado em modo usuário. Para tanto, o código de algumas funções é disponibilizado em bibliotecas que são ligadas aos aplicativos. Estas bibliotecas são parte do SO e portanto sua integridade e confiabilidade podem ser garantidas. O código completo dos protocolos deve necessariamente estar disponível através das funções usuais [DWB+93, Dru96].

Num sistema como o descrito no parágrafo acima, a seqüência de operações necessárias para o estabelecimento da comunicação entre processos executando em máquinas distintas é:

1. o processo de usuário invoca `open()` com os parâmetros usuais;
2. o código do SO estabelece a conexão e aloca um "canal" ao usuário;
3. o processo do usuário invoca as funções `send()` e `receive()` e as executa sem interferência do SO, exceto no tratamento das interrupções;
4. quando conclui a troca de mensagens, o usuário invoca a função `close()`;
5. o SO executa o código que encerra a conexão.

Para que estas operações sejam eficientes, os aplicativos devem poder controlar parcialmente a operação dos adaptadores de interface de rede. Na recepção de mensagens, os adaptadores devem ser capazes de de-multiplexar as várias conexões e direcionar o tráfego para os canais corretos. Na transmissão, o adaptador deve associar os descritores dos canais com as informações necessárias à operação correta de cada protocolo e canal. Do ponto de vista dos aplicativos, cada processo enxerga um adaptador virtual que implementa o "seu" canal de comunicação com o proceso remoto.

Esta idéia tem sido aplicada em agregados de estações de trabalho e multiprocessadores com memória logicamente compartilhada [BLA+94, Gil96]. Nestes sistemas, a latência das operações de comunicação deve ser a mínima possível porque seu impacto no desempenho é muito severo. Para melhorar a eficiência das transmissões, a interface de rede é mapeada no espaço de endereçamento de um processo de usuário. Dessa forma, o SO somente interfere na criação do processo de usuário; os custos da alocação da interface e sua inicialização são amortizados ao longo de todas as operações de comunicação efetuadas pelo processo.

Em resumo, *todas* as técnicas expostas neste capítulo devem ser integradas para garantir alta vazão entre aplicativos: as interfaces em hardware, as interfaces em software e a implementação destas interfaces no sistema operacional.

6.6 Bibliografia e Exercícios

Dois excelentes artigos, de leitura obrigatória, sobre projeto de sistemas são [Lam84c, Met93]. Vários livros contém excelente material sobre redes de alta velocidade: [Par93, PD96, WV96]. O projeto de interfaces e adaptadores de interface de rede é discutido em [Par93, Ste94, SLS94, HP96]. Os programas de controle de adaptadores de rede e interfaces em software são discutidos em [MS92, WV93, Ste94, MBH95, BDF+95]. Sobre avaliação de desempenho de redes para agregados de estações, veja [CLMY96]. A edição de novembro de 1998 da revista Computer contém vários artigos sobre desenvolvimentos recentes [CHM98, BRB98, vEV98, LDK+98]. Veja ainda [DRM+98, WBHvE98].

Exercícios

Ex. 6.1 [§ 6.3] Estime a taxa máxima de transferência entre um periférico e memória num sistema com o processador mais lento da família 80x86 e barramento ISA. Escreva um programa para validar sua estimativa. Repita com um controlador de ADM.

Ex. 6.2 [§ 6.3] Repita o Exercício 6.1 com o processador mais veloz da família 80x86 e barramento PCI com 32 ou 64 bits de largura.

Ex. 6.3 [§ 6.4.2] Quais são as ocasiões em que os controladores de tranmissão e recepção do Afterburner podem gerar interrupções? Quais interrupções são imprescindíveis e quais são dispensáveis? O que fazer com mensagens longas?

Ex. 6.4 [§ 6.4.3] Compare a implementação de gerenciamento de memória com `mbufs`, como usado na versão BSD do Unix, com o gerenciamento com `fbufs`. Em quais circunstâncias uma é melhor que outra?

Proj. 6.1 [§ 6.3] Escreva um programa para medir a taxa máxima de transferência do sistema de memória. O programa deve mover blocos de dados de uma posição de memória para outra.

Proj. 6.2 [§ 6.3] Use o programa do Projeto 6.1 para determinar o tamanho das caches em seu computador. Para isso, efetue medições da taxa máxima de transferência variando o tamanho dos blocos em todas as potências de 2 entre 1 Kbytes e 2 Mbytes. Não esqueça que 1 K=1024.

Proj. 6.3 [§ 6.3] Use o programa do Projeto 6.1 para determinar o tamanho das páginas de memória virtual em seu computador. A estratégia de medição deve considerar e existência da cache de mapeamento de endereços.

Bibliografia

[AD89] Hamid Ahmadi and Wolfgang E Denzel. A survey of modern high-performance switching techniques. *IEEE J. Selected Areas in Communications*, 7(7):1091–1103, September 1989.

[AG94] George S Almasi and Allan Gottlieb. *Highly Parallel Computing*. Benjamin-Cummings, 1994. ISBN 0-8053-0443-6.

[AK91] Bandula W Abeysundara and Ahmed E Kamal. High-speed local area networks and their performance: A survey. *ACM Computing Surveys*, 23(2):221–264, June 1991.

[ALBL91] T E Anderson, H M Levy, B N Bershad, and E D Lazowska. The interaction of architecture and operating system design. In *4th Intl Conf on Arch'l Support for Progr Lang and Oper Sys*, pages 108–120. ACM Comp Arch News 19(2), April 1991.

[AMD94] AMD. *Data Communications Handbook*. Advanced Micro Devices, Inc., 1994.

[ANS87] ANSI. Information systems–fiber distributed data interface (FDDI)–token ring media access control (MAC). ANSI Standard X3.139, American National Standards Institute, 1987.

[ANS88] ANSI. Information systems–fiber distributed data interface (FDDI)–token ring physical layer protocol (PHY). ANSI Standard X3.148, American National Standards Institute, 1988.

[ANS90] ANSI. Information systems–fiber distributed data interface (FDDI)–token ring physical layer medium dependent (PMD). ANSI Standard X3.166, American National Standards Institute, 1990.

[AT93] Philip M Adams and Clovis L Tondo. *Writing UNIX Device Drivers in C*. Prentice-Hall, 1993. ISBN 013963869-5.

[Bac86] Maurice J Bach. *The Design of the UNIX Operating System*. Prentice-Hall, 1986. ISBN 013201799-7.

[BBP88] R Braden, D Borman, and C Partridge. Computing the Internet checksum. Request For Comments 1071, Internet Architecture Board, September 1988.

[BD91] Luiz A Barroso and Michel Dubois. Cache coherence on a slotted ring. In *Proc 1991 Intl Conf Parallel Processing*, volume 1, pages 230–237, August 1991.

[BDF+95] M A Blumrich, C Dubnicki, E W Felten, K Li, and M R Mesarina. Virtual-memory-mapped network interfaces. *IEEE Micro*, 15(1):21–28, February 1995.

[BG92] Dimitri Bertsekas and Robert Gallager. *Data Networks*. Prentice-Hall, 2nd edition, 1992. ISBN 013201674-5.

[BG95] Douglas C Burger and James R Goodman. Simulation of the SCI transport layer on the Wisconsin Wind Tunnel. Tech Report 1265, Computer Sciences Dept, Univ of Wisconsin–Madison, 1995.

[BGY87] L N Bhuyan, D Ghosal, and Q Yang. Approximate analysis of single and multiple ring networks. *IEEE Trans on Computers*, C-38(7):1027–1040, July 1987.

[Bla93] Uyless D Black. *Data Communications and Distributed Networks*. Prentice-Hall, 3rd edition, 1993. ISBN 013203464-6.

[BLA+94] M A Blumrich, K Li, R Alpert, C Dubnicki, E W Felten, and J Sandberg. A virtual memory mapped network interface for the Shrimp multicomputer. In *Proc 21st Intl Symp on Computer Arch*, pages 142–153. ACM Comp Arch News 22(2), April 1994.

[BMK88] D R Boggs, J C Mogul, and C A Kent. Measured capacity of an Ethernet: Myths and reality. Tech report, Digital, Western Research Laboratory, September 1988.

[Bob87] Leonard S Bobrow. *Elementary Linear Circuit Analysis*. Holt, Rinehart & Winston, 1987. ISBN 003007298-0.

[BRB98] R A F Bhoedjang, T Rühl, and H E Bal. User-level network interfaces. *IEEE Computer*, 31(11):53–60, November 1998.

[Bux84] Werner Bux. Local-area subnetworks: A performance comparison. In S S Lam, editor, *Principles of Communication and Networking Protocols*, pages 149–157. IEEE Computer Society Press, 1984. ISBN 081860582-0.

[CHM98] A A Chien, M D Hill, and S S Mukherjee. Design challenges for high-performance network interfaces. *IEEE Computer*, 31(11):42–44, November 1998.

[CLMY96] D E Culler, Lok Tin Liu, R P Martin, and C O Yoshikawa. Assessing fast network interfaces. *IEEE Micro*, 16(1):35–46, February 1996.

[CO93] Israel Cidon and Yoram Ofek. MetaRing – a full-duplex ring with fairness and spatial reuse. *IEEE Trans on Communications*, 41(1):110–120, January 1993.

[Com84] Douglas E Comer. *Operating System Design – The XINU Approach*. Prentice-Hall, 1984.

[Com87] Douglas E Comer. *Operating System Design Vol. II – Internetworking with XINU*. Prentice-Hall, 1987.

[Com95] Douglas E Comer. *Internetworking with TCP/IP – Principles, Protocols and Architecture*, volume 1. Prentice-Hall, 3rd edition, 1995. ISBN 013216987-8.

[CS93] Douglas E Comer and David L Stevens. *Internetworking with TCP/IP – Client-Server Programming and Applications*, volume 3. Prentice-Hall, 1993. ISBN 013474222-2.

[CS94] Douglas E Comer and David L Stevens. *Internetworking with TCP/IP – Design, Implementation and Internals*, volume 2. Prentice-Hall, 2nd edition, 1994. ISBN 013125527-4.

[Dal90] William J Dally. Performance analysis of k-ary n-cube interconnection networks. *IEEE Trans on Computers*, C-39(6):775–785, June 1990.

[DAPP93] P Druschel, M B Abbot, M A Pagels, and L L Peterson. Network subsystem design. *IEEE Network*, 7(4):8–17, July 1993.

[DB88] Doug Dykeman and Werner Bux. Analysis and tuning of the FDDI media access control protocol. *IEEE J. Selected Areas in Communications*, 6(6):997–1010, September 1988.

[dC87] Frank da Cruz. *Kermit, a File Transfer Protocol*. Digital Press, 1987.

[DIX80] Digital, Intel, and Xerox. *The Ethernet, A Local Area Network: Data Link Layer and Physical Layer Specifications – Version 1.0*. DIX, 1980.

[Dou90] Jack Douglas. *V.32* modems are breaking through the echo barrier. In *Connectivity and Standards*, McGraw-Hill Data Comm. Book Series, pages 40–45. McGraw-Hill, 1990. ISBN 007607020-4.

[DRM+98] D Dunning, G Regnier, G McAlpine, D Cameron, B Shubert, F Berry, A M Merrit, E Gronke, and C Dodd. The virtual interface architecture. *IEEE Micro*, 18(2):66–76, March 1998.

[Dru96] Peter Druschel. Operating system support for high-speed computations. *Comm. of the ACM*, 39(9):41–51, September 1996.

[DS87] William J Dally and Charles L Seitz. Deadlock-Free message routing in multiprocessor interconnection networks. *IEEE Trans on Computers*, C-36(5):547–553, May 1987.

[DWB+93] C Dalton, G Watson, D Banks, C Calamvokis, A Edwards, and J Lumley. Afterburner. *IEEE Network*, 7(4):36–43, July 1993.

[EKB+96] D R Engebretsen, D K Kutchta, R C Booth, J D Crow, and W G Nation. Parallel fiber-optic SCI links. *IEEE Micro*, 16(1):20–26, February 1996.

[FR86] R Fitzgerald and R F Rashid. The integration of virtual memory management and interprocess communication in Accent. *ACM Trans on Computer Systems*, 4(2):147–177, May 1986.

[GCW91] W F Giozza, E Conforti, and H Waldman. *Fibras Ópticas: Tecnologia e Projeto de Sistemas*. Makron McGraw-Hill, 1991.

[Gil96] Richard B Gillett. Memory channel network for PCI. *IEEE Micro*, 16(1):12–18, February 1996.

[Gre91] Paul E Green. The future of fiber-optic computer networks. *IEEE Computer*, 24(??):78–87, September 1991.

[GT88] Timothy A Gonsalves and Fouad A Tobagi. On the performance effects of station locations and access protocol parameters in ethernet networks. *IEEE Trans on Communications*, 36(4):441–449, April 1988.

[Hel92] Gilbert Held. *Compressão de Dados: Técnicas e Aplicações, Considerações de Hardware e Software*. Érica Editora, 1992. ISBN 047192941-7, edição original.

[Hex88] Roberto A Hexsel. *Núcleo Multiprocessador para Aplicações de Tempo-real*. Dissertação de mestrado, Depto de Ciência da Computação, Univ Estadual de Campinas, Agosto 1988.

[Hex96] Roberto A Hexsel. Sistemas de memória para multiprocessadores. Tutorial apresentado no VIII Simp Bras de Arquit de Computadores – Proc de Alto Desempenho, agosto 1996.

[Hex97] Roberto A Hexsel. Estado da arte em arquitetura de computadores: Scalable Coherent Interface. In *I Escola Regional de Informática, Regional Minas e Centro-Oeste da SBC*, May 1997. Campo Grande, MS e Cuiabá, MT.

[HN88] A Hooper and R Needham. The cambridge fast ring networking system. *IEEE Trans on Computers*, C-37(10):1214–1224, October 1988.

[HP96] John L Hennessy and David A Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann, 2nd edition, 1996. ISBN 1-55860-329-8.

[HT95] Roberto A Hexsel and Nigel P Topham. The performance of SCI multiprocessor rings. *Journal of the Brazilian Computer Society*, 1(2):24–37, July 1995.

[HT96] Roberto A Hexsel and Nigel P Topham. The performance of cache coherency in SCI-based multiprocessors. In *VIII Simp Bras de Arquit de Computadores – Proc de Alto Desempenho*, pages 47–56, Recife, agosto 1996.

[Hwa93] Kai Hwang. *Advanced Computer Architecture: Parallelism, Scalability, Programmability*. McGraw-Hill, 1993. ISBN 0-07-031622-8.

[IBM70] IBM. *General Information — Binary Synchronous Communication*. IBM Corporation, 3rd edition, October 1970. Manual GA27-3004-2.

[IBM92] IBM. *Synchronous Data Link Control — Concepts*. IBM Corporation, 5th edition, May 1992. Manual GA27-3093-04.

[IEE85a] IEEE. *IEEE Standard 802-1 – Local Area Networks: Overview and Architecture*. IEEE, 1985.

[IEE85b] IEEE. *IEEE Standard 802-2 – Logical Link Control*. IEEE, 1985.

[IEE85c] IEEE. *IEEE Standard 802-3 – Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*. IEEE, 1985.

[IEE92] IEEE. *IEEE Std 1596-1992 – Standard for Scalable Coherent Interface*. IEEE, 1992.

[Int79] Intel. *8250: Assynchronous Communications Controller*. Intel Corporation, 1979.

[Int85] Intel. *8273, 8273-4, 8273-8: Programmable HDLC/SDLC Protocol Controller*. Intel Corporation, 1985. Doc. AFN-00743B.

[ISO84a] ISO. Information technology—Open systems interconnection—Basic reference model. International Standard 7498, ISO, 1984.

[ISO84b] ISO/IEC. Information technology—Telecommunications and information exchange between systems–High-level data link control (HDLC) procedures—Frame structure. International Standard 3309, ISO and IEC, 1984.

[ISO89a] ISO. Information processing systems–fiber distributed data interface (FDDI) part 1: Token ring physical layer protocol (PHY). International Standard 9314-1, ISO, 1989.

[ISO89b] ISO. Information processing systems–fiber distributed data interface (FDDI) part 2: Token ring media access control (MAC). International Standard 9314-2, ISO, 1989.

[ISO90] ISO. Information processing systems–fiber distributed data interface (FDDI) part 3: Token ring physical layer medium dependent (PMD). International Standard 9314-3, ISO, 1990.

[ISO93a] ISO/IEC. Information technology—Telecommunications and information exchange between systems–High-level data link control (HDLC) procedures—Classes of procedures. International Standard 7809, ISO and IEC, 1993.

[ISO93b] ISO/IEC. Information technology—Telecommunications and information exchange between systems–High-level data link control (HDLC) procedures—Elements of procedures. International Standard 4335, ISO and IEC, 1993.

[ISO94] ISO/IEC. Information technology—Telecommunications and information exchange between systems–Local and metropolitan area networks—Specific requirements—Part 2: Logical link control. International Standard 8802-2, ISO and IEC, 1994. ANSI/IEEE Std 802.2, 1994 edition.

[ISO96] ISO/IEC. Information technology—Telecommunications and information exchange between systems–Local and metropolitan area networks—Specific requirements—Part 2: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. International Standard 8802-3, ISO and IEC, 1996. ANSI/IEEE Std 802.3, 1996 edition.

[Jai94] Raj Jain. *FDDI Handbook: High Speed Networking Using Fiber and Other Media*. Addison-Wesley, 1994. ISBN 020156376-2.

[KABG95] A Kägi, N Aboulenein, D C Burger, and J Goodman. An analysis of the interactions of overhead-reducing techniques for shared-memory multiprocessors. In *Proc of the Intl Conf on Supercomputing (ICS95)*, pages 11–20, Barcelona, July 1995. ACM Press.

[Kie94] Tim Kientzle. Intelligent XYmodem. *Dr Dobb's Journal*, 224:36–42+90–91, December 1994.

[KL90] Leonard Kleinrock and Hanoch Levy. On the behaviour of a very fast bidirectional bus network. *IEEE Trans on Communications*, 38(10):1854–1862, October 1990.

[Lam84a] Simon S Lam. Data link control procedures. In S S Lam, editor, *Principles of Communication and Networking Protocols*, pages 70–95. IEEE Computer Society Press, 1984. ISBN 081860582-0.

[Lam84b] Simon S Lam. Multiple access protocols. In S S Lam, editor, *Principles of Communication and Networking Protocols*, pages 117–148. IEEE Computer Society Press, 1984. ISBN 081860582-0.

[Lam84c] Butler W Lampson. Hints for computer system design. *IEEE Software*, pages 11–28, January 1984.

[LCLY91] V O K Li, J-F Chang, K-C Lee, and T-S Yang. A survey of research and standards in high-speed networks. *???*, ??(??):269–309, 1991.

[LDK+98] W S Lee, W J Dally, S W Keckelr, N P Carter, and A Chang. An efficient, protected message interface. *IEEE Computer*, 31(11):69–75, November 1998.

[Lee90] Tony T Lee. A modular architecture for very large packet switches. *IEEE Trans on Communications*, 38(7):1097–1106, July 1990.

[LW95] Daniel E Lenoski and Wolf-Dietrich Weber. *Scalable Shared-Memory Multiprocessing*. Morgan Kaufmann, 1995. ISBN 1-55860-315-8.

[MB76] R M Metcalfe and D R Boggs. Ethernet: Distributed packet switching for local area networks. *Comm. of the ACM*, 19(7):395–404, July 1976.

[MB91] Jeffrey C Mogul and Anita Borg. The effect of context switches on cache performance. In *4th Intl Conf on Arch'l Support for Progr Lang and Oper Sys*, pages 75–84. ACM Comp Arch News 19(2), April 1991.

[MBH95] R Minnich, D Burns, and F Hady. The memory-integrated network interface. *IEEE Micro*, 15(1):11–20, February 1995.

[McN88] John E McNamara. *Technical Aspects of Data Communication*. Digital Press, 3rd edition, 1988. ISBN 155558111-0.

[Met93] Robert M Metcalfe. Computer/network interface design: Lessons from Arpanet and Ethernet. *IEEE J. Selected Areas in Communications*, 11(2):173–180, February 1993.

[MK90] T Mallory and A Kullberg. Incremental updating of the Internet checksum. Request For Comments 1141, Internet Architecture Board, January 1990.

[MKK97] M Molle, M Kalkunte, and J Kadambi. Frame bursting: a technique for scaling CSMA/CD to gigabit speeds. *IEEE Network*, 11(4):6–15, July 1997.

[Mon90] Fábio de Azevedo Montoro. *Transmissão de Dados e Modem*. Érica Editora, 1990. ISBN 0857194049-5.

[Mon94] José Augusto Suruagy Monteiro. *Rede Digital de Serviços Integrados de Faixa Larga RDSI-FL*. IX Escola de Computação, UFPE, Recife, PE, 1994. ISBN 0857194049-5.

[MS92] Hanafy E Meleis and Dimitrios N Serpanos. Designing communication subsystems for high-speed networks. *IEEE Network*, pages 40–46, July 1992.

[MS95] David E McDysan and Darren L Spohn. *ATM Theory and Applications*. McGraw-Hill, 1995. ISBN 007113537-5.

[NSC95] NSC. *PC16550D Universal Asynchronous Receiver/Transmitter with FIFOs*. National Semiconductor Corporation, June 1995.

[PAM94] J Pasquale, E Anderson, and P K Muller. Container shipping: Operating system suport for I/O-intensive applications. *IEEE Computer*, 27(3):84–93, March 1994.

[Par93] Craig Partridge. *Gigabit Networking*. Addison-Wesley, 1993. ISBN 020156333-9.

[PD96] Larry L Peterson and Bruce S Davie. *Computer Networks: A Systems Approach*. Morgan Kaufmann, 1996. ISBN 155860368-9.

[Pfi95] Gregory F Pfister. *In Search of Clusters*. Prentice-Hall, 1995. ISBN 0-13-437625-0.

[Pos80] Jon Postel. User Datagram Protocol. Internet Standard 6, Internet Architecture Board, August 1980. RFC0768.

[Pos81a] Jon Postel. The Internet Protocol. Internet Standard 5, Internet Architecture Board, September 1981. RFC0791, RFC0950, RFC0919, RFC0922, RFC792, RFC1112.

[Pos81b] Jon Postel. Transmission Control Protocol. Internet Standard 7, Internet Architecture Board, September 1981. RFC0793.

[Rij94] A Rijsinghani. Computation of the Internet checksum via incremental update. Request For Comments 1624, Internet Architecture Board, May 1994.

[Rom93] J L Romkey. Nonstandard for transmission of IP datagrams over serial lines: SLIP. Internet Standard 47, Internet Architecture Board, August 1993. RFC1055.

[Ros86] Floyd E Ross. FDDI — a tutorial. *IEEE Communications Magazine*, 24(5):10–17, May 1986.

[Ros89] Floyd E Ross. An overview of FDDI: The fiber distributed data interface. *IEEE J. Selected Areas in Communications*, 7(7):1043–1051, September 1989.

[SCP84] J H Saltzer, D D Clark, and K T Pogran. Why a ring? In S S Lam, editor, *Principles of Communication and Networking Protocols*, pages 232–248. IEEE Computer Society Press, 1984. ISBN 081860582-0.

[SDRC84] J F Shoch, Y K Dalal, D D Redell, and R C Crane. Evolution of the Ethernet local computer network. In S S Lam, editor, *Principles of Communication and Networking Protocols*, pages 215–231. IEEE Computer Society Press, 1984. ISBN 081860582-0.

[SG91] Steven L Scott and James R Goodman. Performance of pipelined K-ary N-cube networks. Tech Report 1010, Computer Sciences Dept, Univ of Wisconsin–Madison, February 1991.

[SG94] Steven L Scott and James R Goodman. The impact of pipelined channels on k-ary n-Cube networks. *IEEE Trans on Parallel and Distributed Systems*, 5(1):2–16, January 1994.

[SGV92] S L Scott, J R Goodman, and M K Vernon. Performance of the SCI ring. In *Proc 19th Intl Symp on Computer Arch*, pages 403–414. ACM Comp Arch News 20(2), May 1992.

[Sim94] W Simpson. The Point-to-Point Protocol (PPP). Internet Standard 51, Internet Architecture Board, July 1994. RFC1661, RFC1662.

[SLS94] M W Sachs, A Leff, and D Sevigny. LAN and I/O convergence: a survey of the issues. *IEEE Computer*, 27(12):24–32, December 1994.

[Spu94a] Charles Spurgeon. Guide to Ethernet configuration (v-2.3). Tech report, Networking Services, Univ of Texas at Austin, January 1994. ftp.utexas.edu.

[Spu94b] Charles Spurgeon. Guide to Ethernet (v-2.5). Tech report, Networking Services, Univ of Texas at Austin, January 1994. ftp.utexas.edu.

[Sta91] William Stallings. *Data and Computer Communications*. Macmillan Publ Co, 3rd edition, 1991. ISBN 002415454-7.

[Sta92] William Stallings. *ISDN and Broadband ISDN*. Macmillan Publ Co, 2nd edition, 1992. ISBN 002415475-X.

[Sta97] William Stallings. *Data and Computer Communications*. Prentice-Hall, 5th edition, 1997. ISBN 002415425-3.

[Ste94] Peter A Steenkiste. A systematic approach to host interface design for high speed networks. *IEEE Computer*, 27(3):47–57, March 1994.

[Tan89] Andrew S Tanenbaum. *Computer Networks*. Prentice-Hall, 2nd edition, 1989. ISBN 013166836-6.

[Tan92] Andrew S Tanenbaum. *Modern Operating Systems*. Prentice-Hall, 1992. ISBN 013588187-0.

[Tar86] Liane M R Tarouco. *Redes de Computadores Locais e de Longa Distância*. Makron McGraw-Hill, 1986.

[Tho92] Clark Thomborson. The V.42bis standard for data compressing modems. *IEEE Micro*, 12(5), October 1992.

[TS86] Herbert Taub and Donald L Schilling. *Principles of Communications Systems*. McGraw-Hill, 2nd edition, 1986.

[vEV98] Thorsten von Eicken and Werner Vogels. Evolution of the virtual interface architecture. *IEEE Computer*, 31(11):61–68, November 1998.

[WBHvE98] M Welsh, A Basu, X W Huang, and T von Eicken. Memory management for user-level network interfaces. *IEEE Micro*, 18(2):77–82, March 1998.

[Wel94] Bill Wells. Writing serial drivers for Unix. *Dr Dobb's Journal*, 224:68–71+97–99, December 1994.

[WM87] Richard W Watson and Sandy A Mamrak. Gaining efficiency in transport services by appropriate design and implementation choices. *ACM Trans on Computer Systems*, 5(2):97–120, May 1987.

[WV93] Larry Wakeman and John Von Voros. Architectural choices for network performance. Application Note 873, National Semiconductor, February 1993.

[WV96] Jean Walrand and Pravin Varaiya. *High-Performance Communication Networks*. Morgan Kaufmann, 1996. ISBN 155860341-7.

[ZL78] J Ziv and A Lempel. Compression of individual sequences via variable-rate coding. *IEEE Trans Information Theory*, 24(5):530–536, 1978.

[ZNV88] M Zafirovic-Vukotic, I G Niemegeers, and D S Valk. Performance analysis of slotted ring protocols in HSLAN's. *IEEE J. Selected Areas in Communications*, 6(6):1011–1024, September 1988.

Índice

Símbolos

$P(f)$, 26
V, 87
$\Omega(x)$, 26
ω, 19
ω_p, 42
t_b, 6, 29, 37, 59, 136
t_p, 6, 159, 169
t_{heq}, 159

A

aceitação,
  de carona, 91, 104, 109
  inclusiva, 89, 104, 107
acesso direto à memória, *veja* ADM
acoplamento eletromagnético, 28
adaptador,
  de interface serial, 60
  de rede, 134, 191, 205
ADM, 117, 120, 197, 204
  coleta e dispersão, 199
  controlador, 197
Afterburner, 205
aglomerado, 176
alfabeto, 4
Aloha, 135
  discreto, 135
anel, 132
  inserção de registrador, 139, 181, 182
  passagem de bastão, 136, 165
  segmentado, 137
ANSI, 165
  X3T11, 153
  X3T9, 165
ARQ, *veja* PAR
ASCII, 5
ASK, 42
atenuação,
  em frequência, 30
  resistiva, 29
atraso de fase, 31, 32

B

barramento, 181
  de E/S, 60, 193
  de memória, 193
baud, 35, 37, 44, 45, 168
bit,
  de final, 59
  de início, 59
  de paridade, 60, 80
bit pipe, *veja* infoduto
bit stuffing, *veja* enxerto de bits
bloqueio, 185
bps, 6
breakout box, 70
broadcast, *veja* difusão
BSC, 99, 116
  paridade, 80, 100
  quadro, 99

C

célula, 142
código, 4
  ASCII, 5
  bipolar, 39
  Brasileiro de Intercâmbio de Informações, 5
  CBII, 5
cópia virtual, 207
cabo,
  blindado, 34
  coaxial, 16
  par trançado, 16, 154
cache, 180, 193
  consistência com memória, 200, 208
  em multiprocessador, 180
  fila de escrita, 195
  localidade, 194
  poluição, 195
  protocolos de coerência, 180
  taxa de acerto, 194
  tempo médio de acesso, 194
cache de mapeamento de endereços, *veja* CME
CAM, 134
  anel segmentado, 137
  contenção, 135
  inserção de registrador, 139, 183
  passagem de bastão, 135, 136, 168
  reserva, 135
  revezamento, 135
Cambridge Ring, 188
canal de E/S, 53
canal de voz, 6, 26
cancelamento de eco, 48, 49
capacidade de canal, 6, 34, 47
capacitância, 28
Carrier Sense Multiple Access, 136
Carrier Sense Multiple Access with Collision Detection, 136
CCITT, 13
  Série V, 41, 49
  V.14, 112
  V.21, 49
  V.22, 26, 49

V.22bis, 45, 49
V.23, 43
V.24, 62–64
V.28, 62, 63
V.29, 45
V.32, 47, 49
V.35, 42, 117
V.41, 82
V.42, 112
V.42bis, 113
X.25, 112
CEL, 134, 144
802.2, 149
tipos de serviço, 145
ponte, 148, 173
circuito,
virtual, 142
CISA, 71
cliente-servidor, 130
CME, 201, 208
taxa de acerto, 201
codificação,
4b/5b, 167
8b/10b, 153
AMI, 39
diferencial, 38
HDB3, 39
Manchester, 38, 151, 154, 168
multinível, 36
NRZ, 38
NRZ-I, 168
NRZ-L, 151
componente DC, 23
compressão de dados, 113
comunicação,
confiável, 147
precária, 146
comutação, 141
de células, 142
de circuitos, 141
de pacotes, 141, 183
comutador, 163
concentrador, 133, 155, 171
de terminais, 54
conexão confiável, 146
constelação, 45
controlador,
de ADM, 197
de terminais, 53
Controlador de Interface,
Serial Assíncrona, *veja* CISA
Serial Síncrona, *veja* CISS
Controle de Acesso ao Meio, *veja* CAM
Controle de Enlace Lógico, *veja* CEL
controle de fluxo, 87, 187
janela deslizante, 89
para-e-espera, 88
copy-on-write, 207
CRC, *veja* TDP
CSMA, 136
CSMA/CD, 136, 149, 157
cubo k-ário, 177

D

DAN, 130
datagrama, 10
dB, 27
Decibel, 27
detecção de erros, 79
detecção de portadora, 136
1-persistente, 136
não-persistente, 136
diâmetro de rede local, 155
dibit, 35
difusão, 4, 131
disciplina de linha, 85
consulta e seleção, 86, 101, 106, 134
multiponto, 86, 101
ponto-a-ponto, 86, 101
distorção de fase, 31
domínio,
freqüência, 22
tempo, 22
DPSK, 44
DRAM, 193
duplex, 7, 48, 49, 67, 85, 105, 154, 163
duplexidade, 85
duplex, 7
semi-duplex, 7
simplex, 7

E

E/S,
ADM, 120, 197
consulta, 77, 120
interrupção, 77, 120
mapeada como E/S, 76, 196
mapeada como memória, 76, 123, 196
programada, 196
ECD, 3, 40, 41
eficiência de código, 40
EIA, 14
232, 62, 71–73, 112, 117
conexão sem modem, 67
diagnóstico e teste, 69
enlace analógico local, 69
enlace digital remoto, 69
interface elétrica, 63
interface funcional, 64, 66
interface mecânica, 63
NRZ-L, 38
reversão de linha, 67
testador de interface, 70
422, 42, 70, 117
423, 70
449, 70
encapsulamento, 10, 83, 199, 208
enlace, 3
multiponto, 85, 86, 99
ponto-a-ponto, 85, 96, 99
enlace de dados, 130
enlace lógico, 11, 134
enquadramento, 8, 59, 73, 84, 115, 183
entidade, 9
envelhecimento de pacotes, 186

enxerto de bits, 104
equalizador, 24, 30
equipamento de comunicação de dados, *veja* ECD
equipamento terminal de dados, *veja* ETD
erro,
  assincronia na recepção, 73
  de amostragem, 59
  de enquadramento, 60, 73
  de paridade, 74
  de transmissão, 118
espaço, 59
espectro de freqüência, 19, 39
  envoltória, 22
estação, 3
ETD, 3
Ethernet, 6, 38, 82, 150
  100Base-T, 152, 153
  100VG-AnyLAN, 152
  10Base-2, 151, 152
  10Base-T, 152
  colisão, 157
  comutador, 156, 163
  concentrador, 155
  domínio de colisão, 155, 162
  endereços, 161
  limites no diâmetro, 162
  Organizationally Unique Identifier, 162
  quadro, 161
  regra 5-4-3, 162
  repetidor, 154
  sequestro do meio, 160
  sinalização, 151
Express Ring, 188

F

faixa de passagem, 26, 32, 35, 39, 47, 141
Fast Ethernet, 152
fbufs, 208
FDDI, 6, 82, 164
  anel primário, 173
  anel secundário, 173
  estações, 171
  inicialização, 170
  liberação do bastão, 169
  quadro, 171
  serviço assíncrono, 165
  serviço síncrono, 165
  sinalização, 167
  tempo de rotação do bastão, 169
  tolerância a falhas, 165
FDDI-II, 175
Fiber Distributed Data Interface, *veja* FDDI
fibra óptica, 17, 168
  monomodal, 32
  multimodal, 32
Fibre Channel, 153
filtro, 24, 25, 43
flag, *veja* marcador
FSK, 43

G

gerador de quadratura, 46
Gigabit Ethernet, 150

H

harmônicos, 21
HDLC, 103, 116, 146, 149
  operação balanceada, 110
  operação desbalanceada, 108
  quadro, 103

I

IEC, 13
IEEE, 14
  802.*, 149
  802.2, 112, 144, 161, 166
    tipos de serviço, 145
  802.3, 150
  1956, 140
  Organizationally Unique Identifier, 162
indutância, 28
infoduto, 9, 10, 190
interface,
  de alta velocidade, 190
  ETD-ECD, 53, 60, 117
  serial, 58
    assíncrona, 71
    síncrona, 116
interferência entre símbolos, 32
Internet, 12
internet, 11
Internet Protocol, 112
interrupção, 117, 120, 194, 200
  custo de, 201
  tratador de, 126, 200
intervalo de bit, 6, 37, 58
ISDN, *veja* RDSI
ISO, 13, 165
  8859-1, 5
  3309, 112
  4335, 112
  7809, 112
ITU, 13

J

janela deslizante, 89, 104

K

Kermit, 96

L

LAN, 130
LAPM, 112
largura de faixa, 26
latência, 14, 87, 179, 191
Lei de Amdahl, 192
linha,
  comutada, 8
  condicionada, 8, 31
  discada, 8, 49
  LPCD, 8
  privada, 8, 49
LLC, *veja* CEL
Logical Link Control, *veja* CEL

M
MA, 42
MAC, *veja* CAM
MAN, 130
MAQF, 45, 48, 49
marca, 59
marcador, 103, 116
MCT, 47, 49
MDDF, 44, 49
MDF, 44
Medium Access Control, *veja* CAM
meio compartilhado, 162
memória,
  cache, 180, 193
  dinâmica, 193
  DRAM, 179, 192
  virtual, 201, 207
mensagem, 2, 10
MetaRing, 188
MF, 43, 49
modelo ISO/OSI, 8, 9, 12, 83
modem, 7, 10, 18
  analógico, 41
  banda básica, 40
  constelação, 45
modulação,
  codificação em treliça, 46, 47
  deslocamento de fase, 44
  deslocamento diferencial de fase, 44
  em amplitude, 42
  em amplitude e fase, 45
  em fase, 44
  em freqüência, 43
montador de protocolo, 117
multiplexação,
  em freqüência, 48
  estatística, 55, 142
  por divisão de freqüência, 57
  por divisão de tempo, 54
multiplexador,
  estatístico, 55
  por divisão de tempo, 54
multiprocessador, 176, 180

N
Nível de Aplicação, 9, 12
Nível de Apresentação, 9, 11
Nível de Enlace, 11, 83, 112, 133, 205
Nível de Rede, 9, 11, 112
Nível de Sessão, 9, 11
Nível de Transporte, 9, 11
Nível Físico, 9, 10, 112, 131

O
Open Sistems Interconnection, 8
OSI, 8

P
pacote, 10, 83, 141
PAR, 91, 103, 104
  para-e-espera, 92
  repetição seletiva, 93, 107, 110, 112
  volta-N, 93, 107, 110, 111
par trançado, 16, 154
paridade,
  longitudinal, 80, 100
  vertical, 80
PAS, 144
passagem de bastão, 135
pedido automático de repetição, *veja* PAR
Point-to-Point Protocol, 112
polling, 77, 119, 120
ponte, 148, 155
  802.2, 173
Ponto de Acesso a Serviço, *veja* PAS
portadora, 36, 41
preservação de vazão, 190
produto alcance-vazão, 16
produto retardo-vazão, 87
produto vazão latência, 14
programa controlador de periférico, 194
Protocol Data Unit, *veja* UDP
protocolo, 3, 84
  BSC, 99
  HDLC, 103
  IP, 112
  Kermit, 96
  LAP, 112
  LAPM, 112
  LLC, 112
  PPP, 112
  SDLC, 112
  SLIP, 112
  TCP, 192
  TCP/IP, 191
  UDP, 192
PSK, 44

Q
QAM, 45
quadro, 8, 10, 83, 84, 99, 103, 161, 171, 186

R
RDSI, 112
recuperação de erros, *veja* PAR
rede,
  confinada, 130, 140, 175
  dispersa, 8, 130
  local, 8, 130
  metropolitana, 8, 130
  topologia, 131
Rede Digital de Serviços Integrados, 112
redespacho, 143, 155, 163
relação sinal-ruído, 33, 36, 47, 48
remontagem, 84
resistência, 28
resposta em freqüência, 24
reversão de linha, 67, 88, 99
roteador, 2, 156
roteamento,
  redespacho, 163
  trespasse, 163
  wormhole, 177
ruído, 33

S
símbolo, 4, 32, 37, 139, 167, 181
Série de Fourier, 19
SAP, *veja* PAS
Scalable Coherent Interface, *veja* SCI
SCI, 140, 175
  pacote, 186
SDLC, 112
segmentação, 84
semi-duplex, 7, 49, 67, 85, 93, 96, 99, 105, 150, 154, 163
send(), 203
Serial Line IP, 112
simplex, 7
sinalização diferencial, 183
sistema operacional, *veja* SO
SO, 190, 191, 210
  gerenciamento de memória, 202
  Linux, 202
  memória virtual, 201, 207
  troca de contextos, 194, 195
  Unix, 207
soquete, 203
statmux, 55

T
taxa,
  de erro, 79
  de modulação, 40
  de sinalização, 35, 37, 40, 154, 168, 190
  de transmissão, 6, 35, 37
TCM, 47
TCP, 13
TCP/IP, 146, 191, 203
TDP, 81
  circuito, 82
TDP-12, 100
TDP-16, 82, 100, 187
TDP-32, 82, 161, 171
TDP-CCITT, 82, 103, 113
Telebrás,
  225-540-730, 62
tempo de propagação, 6, 87, 159, 169
temporizador, 86, 92, 102, 109
Teorema de Nyquist, 35, 47
Teorema de Shannon, 36, 47
terminal, 53
testador de interface, 70
teste,
  da divisão polinomial, 81
  de paridade longitudinal, 80
  de paridade vertical, 80
topologia, 3
  anel, 3, 132
  barramento, 4, 131
  estrela, 3, 133
  malha, 4
  multiponto, 85
  ponto-a-ponto, 3, 85
  toro, 176
toro,
  anel, 176
  cubo, 176
  malha, 176
transação bifásica, 183
transceptor, 3, 18, 151
transmissão,
  a dois fios, 48
  a quatro fios, 48, 49
  assíncrona, 7, 58, 59, 112
  banda básica, 36
  guiada, 4
  livre, 4
  síncrona, 7, 112, 115
    baseada em bit, 116
    baseada em caracter, 115
  serial, 5
transparência de protocolo, 95
  BSC, 101
  enxerto de bits, 104
  HDLC, 104
  Kermit, 97
trespasse, 143, 156, 163

U
UART, *veja* CISA
UDP, 10, 13, 83, 146, 147, 199
Unidade de Dados de Protocolo, *veja* UDP
USART, *veja* CISS
USRT, *veja* CISS
utilização, 7, 56, 134, 135, 184
UTP, *veja* par trançado

V
vazão, 6, 14, 164, 190
velocidade de transmissão, 6
violação de código, 39, 167

W
WAN, 130
write(), 202

X
Xon-Xoff, 57