

9

SISTEMAS DE TRANSMISSÃO

Neste capítulo, apresentarei os elementos de um sistema de teleprocessamento e como eles se interligam, dando ênfase ao sistema de transmissão de dados em termos de sua velocidade, capacidade e eficiência.

Quando um CPD (Centro de processamento de dados) se expande além das fronteiras de sua sala, digamos, a distâncias acima de 100 metros, passa a constituir um sistema de teleprocessamento, ou seja, processamento a distância. Normalmente, uma unidade de controle de comunicação (UCC) se incumba da tarefa de gerenciar os canais de comunicação e passar os dados, de forma mais disciplinada, à CPU. Quanto maior for a inteligência da UCC, mais livre fica a CPU para suas tarefas principais.

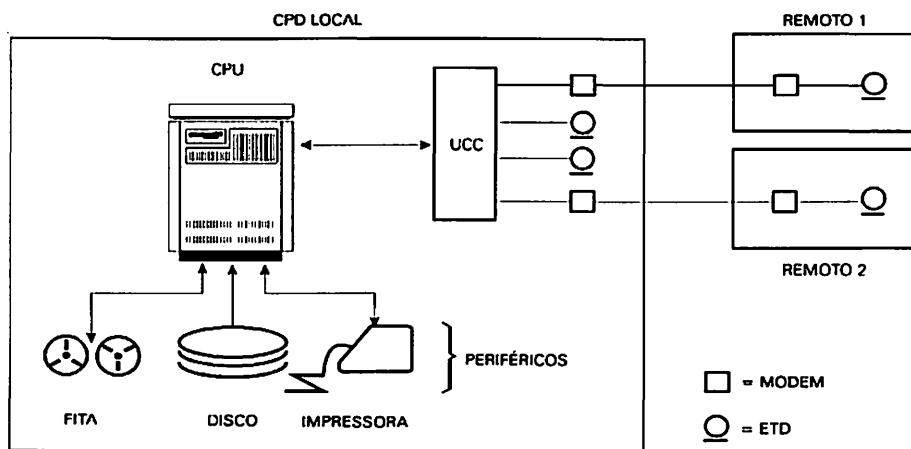


Fig.9.1: Sistema de teleprocessamento

A figura 9.1 ilustra um sistema de teleprocessamento, onde a UCC gerencia dois terminais locais e dois remotos. Em algumas figuras que já apresentei, e nas próximas, estarei omitindo a UCC. Estou supondo que a CPU, representada nessas figuras, possua infra-estrutura apropriada ao gerenciamento da comunicação remota, eventualmente fazendo uso da UCC.

Chamarei o conjunto de elementos, que se encontra entre os terminais e a UCC, de “sistema de transmissão”. Ele pode incluir, além dos modems e das linhas, alguns outros equipamentos, que descreverei no 9.1.

9.1 TOPOLOGIA

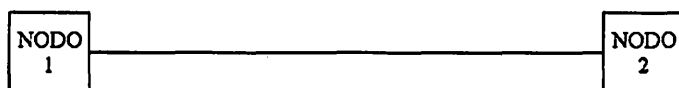
A distribuição geográfica dos terminais de um sistema de teleprocessamento e os circuitos utilizados para veicular os dados, definem a topologia do sistema.

De uma forma genérica, aos pontos extremos de uma topologia chama-se “nodo”, que pode ser um terminal de vídeo ou a CPU, por exemplo.

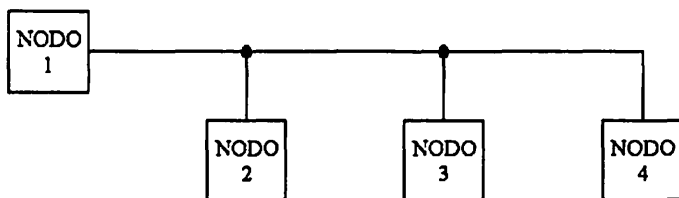
Existem, praticamente, dois tipos de circuitos que podem interligar os nodos:

Ponto-a-ponto = conecta somente dois nodos

Multiponto = um único circuito que conecta mais de dois nodos



PONTO-A-PONTO



MULTIPONTO

Fig. 9.2: Tipos de circuito

Um sistema completo de teleprocessamento pode conter, em sua topologia, circuitos ponto-a-ponto, circuitos multiponto ou ambos, interligados de formas variadas, como você pode ver na figura 9.3.

Vários equipamentos podem ser utilizados para configurar cada circuito, e, em termos de linha de transmissão, cada circuito pode ser efetivado a 2 ou 4 fios, utilizando modems.

Descreverei, a seguir, os principais equipamentos utilizados para configurar os diversos circuitos:

Equipamento	Utilização mais comum
UDA	multiponto
UDD	multiponto
MUX TDM	ponto-a-ponto
STAT MUX	ponto-a-ponto
MUX FDM	ponto-a-ponto

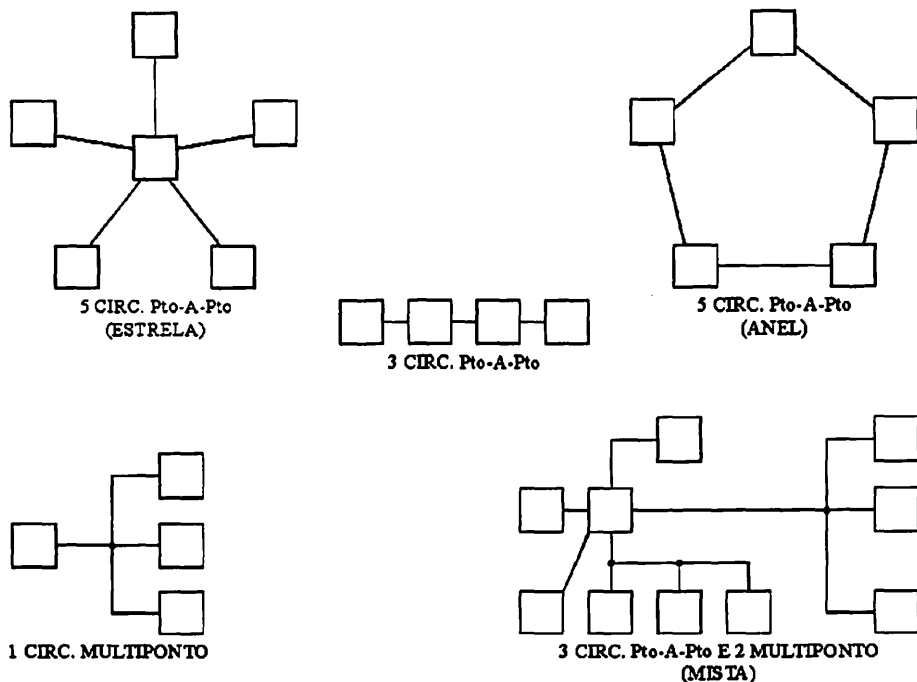


Fig. 9.3: Exemplos de topologia

9.1.1 UDA

Unidade de derivação analógica.

Este equipamento permite a um modem compartilhar mais de uma linha telefônica em um circuito multiponto.

A UDA é normalmente utilizada em circuitos multiponto a 4 fios.

O par de transmissão vindo do nodo "1" (estação mestre), normalmente funciona com portadora constante e seu sinal chega a todos os demais nodos, que devem identificar seu instante exato de transmitir para a estação mestre. Todos os pares de recepção chegam à UDA, que somente passa para a estação mestre um dos sinais de recepção por vez (a prioridade normalmente é dada ao primeiro que chegar).

Naturalmente, todos os demais nodos devem operar com portadora chaveada, ou seja, quando um está transmitindo, os demais estão em silêncio.

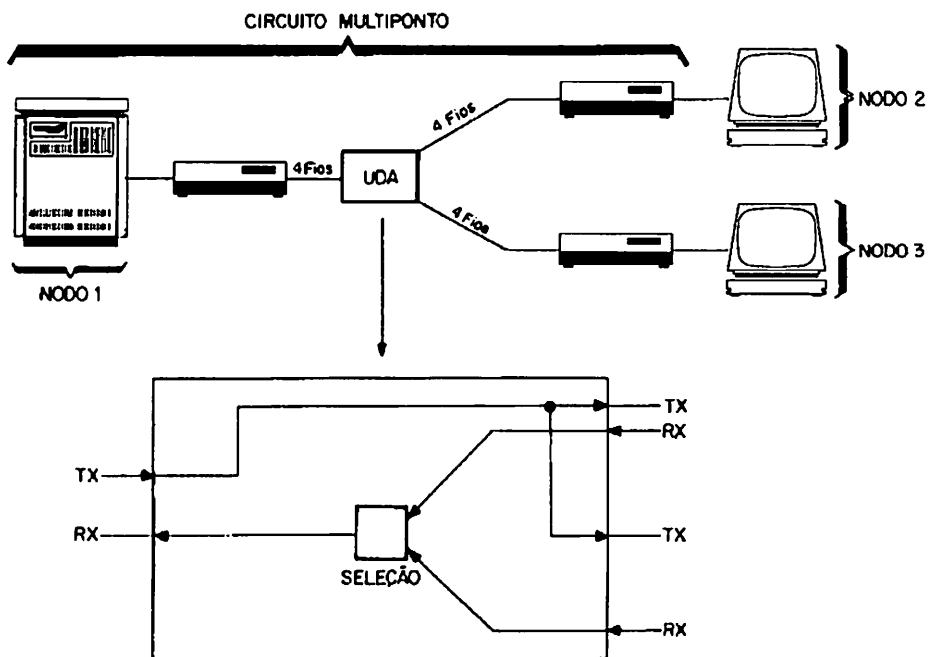


Fig.9.4: Multiponto com UDA

9.1.2 UDD

Unidade de derivação digital.

Este equipamento permite a um modem compartilhar mais de um terminal em um circuito multiponto.

A UDD normalmente é utilizada em circuitos multiponto a 4 fios.

Ela faz o compartilhamento a nível digital e o processo é semelhante àquele utilizado pela UDA.

Os dados recebidos do modem remoto (estação mestre) são encaminhados a todos os nodos que estão ligados à UDD, e, no sentido inverso, somente um terminal deve transmitir a cada vez.

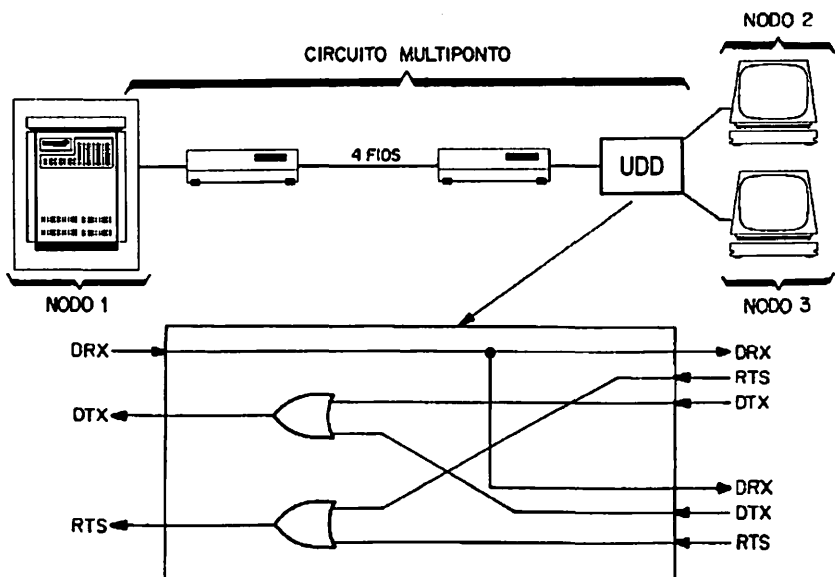


Fig. 9.5: Multiponto com UDD

9.1.3 MULTIPLEXADOR POR DIVISÃO DO TEMPO (MUX TDM)

O TDM normalmente é utilizado para otimizar o uso de linhas telefônicas em topologias que utilizem vários circuitos ponto-a-ponto.

A figura 9.6 mostra uma topologia com três circuitos ponto-a-ponto, ligados através de uma única linha telefônica.

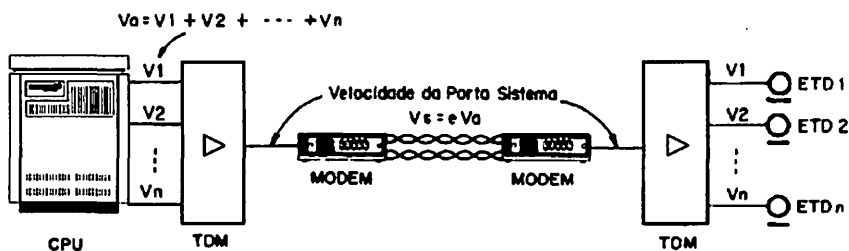


Fig.9.6: Ponto-a-ponto com TDM

A utilização do TDM permite economizar linhas telefônicas, o que pode ser significativo principalmente se a ligação for de longa distância. No entanto, o modem a ser utilizado deve ter uma velocidade maior do que os que seriam utilizados se as ligações fossem feitas sem o TDM.

O MUX TDM aloca uma parcela fixa de tempo para cada porta, sempre em uma sequência fixa.

A soma das velocidades das portas terminais é chamada de velocidade agregada (V_a).

No MUX TDM, portanto, as portas terminais são multiplexadas no tempo, cabendo a cada uma delas, uma parcela fixa de tempo, chamada de “envelope”. Cada envelope possui um número fixo de bits, que pode ser inclusive igual a um. Além dos envelopes de dados, correspondentes às portas terminais, o TDM pode ter a necessidade de transmitir envelopes de controle, cuja finalidade é endereçar os envelopes de dados a fim de garantir que a demultiplexação seja feita corretamente. Devido ao exposto, o TDM possui uma “relação de envelopamento”, que varia de equipamento para equipamento:

$$V_s = e \cdot (V_1 + V_2 + \dots + V_n)$$

$$V_s = e \cdot V_a$$

V_s = velocidade da porta sistema
 V_1, V_2, V_n = velocidades das portas terminais
 V_a = $V_1 + V_2 + \dots + V_n$ = velocidade agregada
 e = relação de envelopamento (maior ou igual a 1)

Naturalmente, é desejável que a relação de envelopamento seja a menor possível.

Em multiplexadores assíncronos ou específicos para um determinado protocolo, podemos encontrar $e=1$, mas em outros, síncronos e transparentes a códigos, podemos encontrar uma relação maior que 1. Algumas relações utilizadas são “4/3” e “10/9”.

9.1.3.1 TRANSDATA

A rede Transdata, da Embratel, suporta transmissões de dados digitais síncronos nas velocidades de 1.200, 2.400, 4.800 e 9.600 bps.

Existem, também, alguns circuitos a 300 e 1.200 bps assíncronos, porém, como a rede é essencialmente síncrona, estas modalidades, menos eficientes, têm pouca utilização.

A finalidade da rede é oferecer circuitos privativos ponto-a-ponto, interurbanos, que incluem também os modems.

O usuário paga uma tarifa mensal que depende da velocidade desejada e do degrau tarifário de transmissão de dados, que é uma função da distância entre as duas localidades.

A tabela a seguir mostra as tarifas interurbanas, mensais, a serem pagas pelo usuário do sistema, conforme o degrau tarifário e a velocidade, vigente em 25/11/86.

Degrau	[km]	1200bps	2400bps	4800bps	9600bps
D2	até 50	4.092,55	6.577,56	9.646,63	14.616,65
D3	50 a 100	4.780,10	7.682,59	11.267,27	17.072,25
D4	100 a 300	5.729,57	9.208,59	13.505,29	20.463,31
D5	300 a 700	11.459,15	18.417,18	27.010,57	40.926,63
D6	700 a 1.500	14.323,93	23.021,47	33.763,21	51.158,28
D7	acima 1.500	16.370,21	26.310,25	38.586,53	58.466,61

As tarifas desta tabela são obtidas da seguinte fórmula:

$$\text{Tarifa} = N.m.TCD$$

N	m	TCD
	0,25 (degrau 2)	Tarifa de comunicação de dados. Cz\$ 5,51
2.971 (1.200 bps)	0,292 (degrau 3)	
4.775 (2.400 bps)	0,35 (degrau 4)	
7.003 (4.800 bps)	0,7 (degrau 5)	
10.611 (9.600 bps)	0,875 (degrau 6)	
	1,0 (degrau 7)	

A figura seguinte mostra, de forma simplificada, a configuração do Transdata em Brasília.

Observe que a ligação do cliente até o centro Transdata é um acesso urbano e, em geral, é feito com modems banda-base, a menos que as condições de linha exijam modems analógicos.

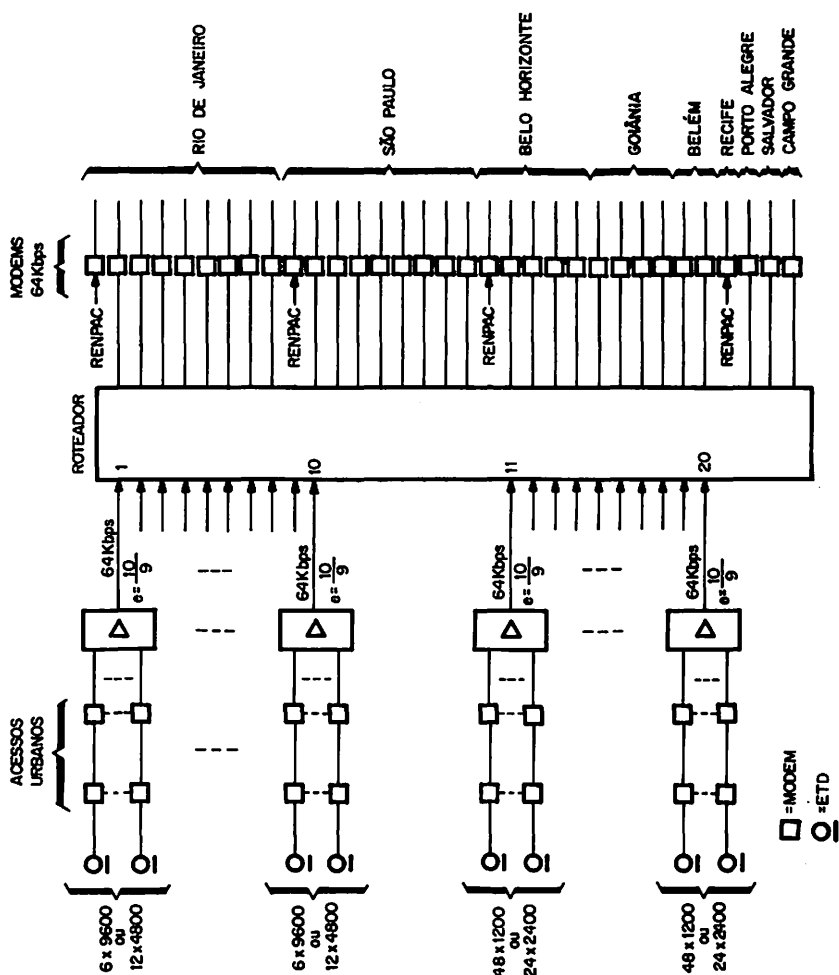


Fig.9.7: Transdata - Centro Brasília

Portanto, no acesso urbano, o modem da esquerda está no cliente e o da direita está na central Transdata.

Todos os acessos dos clientes entram em multiplexadores TDM com $e = 10/9$ e saída de 64 Kbps. Um roteador tem a tarefa de direcionar a transmissão para a localidade desejada, agrupando em canais de 64 Kbps.

As saídas do roteador entram em modems de 64 Kbps, cujas transmissões ocupam a faixa de grupo, e vão para outra sala onde se situam os multiplexadores FDM e os transmissores.

Observe que, como a rede transdata é síncrona, os acessos urbanos utilizam modems predispostos com sincronismo externo, no lado do Transdata, e sincronismo regenerado, no lado do cliente.

9.1.4 MULTIPLEXADOR ESTATÍSTICO (STAT MUX)

Ao contrário do MUX TDM, o multiplexador estatístico aloca parcelas de tempo conforme a necessidade de cada porta terminal, ou seja, portas que têm muitos dados a transmitir ocuparão mais tempo no MUX, do que as portas que têm poucos dados a transmitir.

Inicialmente, os dados de cada porta terminal são armazenados em uma fila e o MUX faz a transmissão de modo a otimizar o tempo total.

Dessa forma, o tempo praticamente só é alocado a portas que estão em atividade, permitindo que a velocidade agregada seja maior do que a velocidade da porta sistema. Devido a esta otimização do tempo, os STAT MUX normalmente possuem a capacidade de detecção e correção de erros.

Quanto a velocidade agregada pode ser maior que a velocidade da porta sistema?

Naturalmente, a resposta a esta pergunta vai depender da utilização de cada porta terminal, e a porcentagem de utilização exata nem sempre é conhecida. Para um primeiro cálculo, utilizam-se os seguintes valores práticos:

Terminal/computador = 25%

Impressora/computador = 100%

Exemplo:

Vamos supor que desejamos multiplexar 8 terminais, cada um operando a 2400 bps, utilizando um STAT MUX. Qual deve ser a velocidade do modem?

Velocidade agregada: $V_a = 8 \times 2.400 = 19.200 \text{ bps}$

Velocidade do modem: $V_s = 25\% \text{ de } 19.200 = 4.800 \text{ bps}$

Neste ponto, caberia mais uma pergunta: O que acontece se a utilização de algum terminal ultrapassar a capacidade da porta sistema?

Nesse caso, o STAT MUX se incumbirá de fazer um controle de fluxo, enviando um caractere de controle ao terminal, para que este interrompa a transmissão (X-OFF), ou abaixe o CTS.

Se, por algum motivo, o terminal não atender, os dados podem ser perdidos.

Como falei inicialmente, o STAT MUX possui uma fila de entrada ("buffer"), onde os dados são armazenados antes de serem transmitidos. Este fato traz uma consequência que deve ser levada em consideração, quando o sistema estiver sendo projetado: aumenta o tempo de resposta associado aos circuitos que passam pelo STAT MUX.

Quando a transmissão for via satélite, um cuidado maior deve ser tomado, pois alguns STAT MUX não possuem filas ("buffers") de comprimento suficiente para operar com os retardos extremamente longos, introduzidos pela ligação.

9.1.5 MULTIPLEXADOR POR DIVISÃO DE FREQUÊNCIA (FDM)

Você viu em 3.3.7 que a degeneração, correspondente à translação de frequência, é provocada pelos FDM.

Naquele item, descrevi o processo da multiplexação em frequência e retorno agora a este assunto, sob o ponto de vista de sistema.

De acordo com uma recomendação CCITT, 12 canais de voz podem ser multiplexados para formar um grupo básico, que ocupa a faixa de 60 a 108 KHz.

A cada canal de voz é reservada uma faixa de 4 KHz, para garantir que cada canal não sofra interferência de seus adjacentes.

Cinco grupos podem ser multiplexados para formar um “supergrupo”. Cinco “supergrupos” podem ser multiplexados para formar um “master-grupo”. Três “master-grupos” podem ser multiplexados para formar um “super-master-grupo” que, finalmente, carrega a informação de 900 canais de voz ($12 \cdot 5 \cdot 5 \cdot 3 = 900$).

A figura 6.8 ilustra a transmissão de dados, utilizando a faixa de grupo, com auxílio de FDM e TDM.

No caso do FDM, 12 canais de voz, cada um correspondendo a um ETD, ocupam a faixa de grupo, independente das velocidades dos terminais (v_t).

No caso do TDM, “n” ETD’s compartilham uma fatia da velocidade agregada de 48.000 bps, que gera uma velocidade de 64.000 bps na porta sistema do TDM, que, no caso deste exemplo, tem $c=4/3$.

Os dados a 64.000 bps entram em um modem V36, que gera, em sua saída, um sinal cujo espectro ocupa a faixa de grupo.

A quantidade de terminais na entrada do TDM depende da velocidade “ v_t ”, pois no caso da figura, deve-se ter:

$$V_a = N \cdot v_t = 48.000 \text{ bps}$$

V_a = velocidade agregada

N = nº de terminais

v_t = velocidade dos terminais

v_t	N
1.200	40
2.400	20
4.800	10
9.600	5

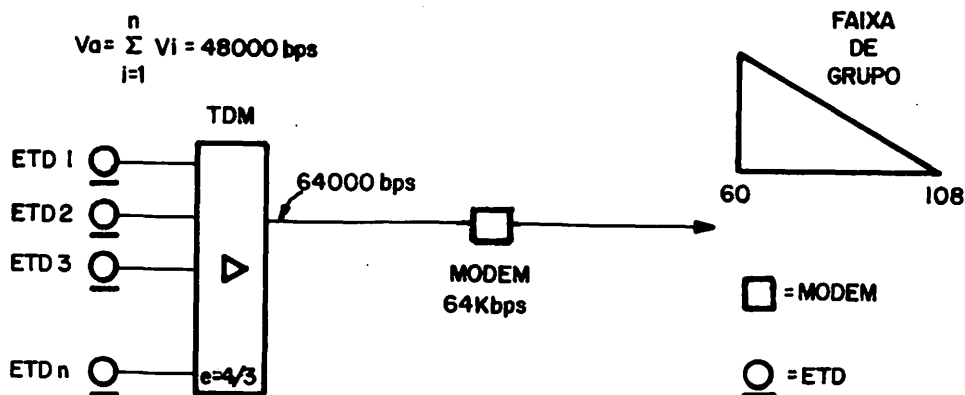
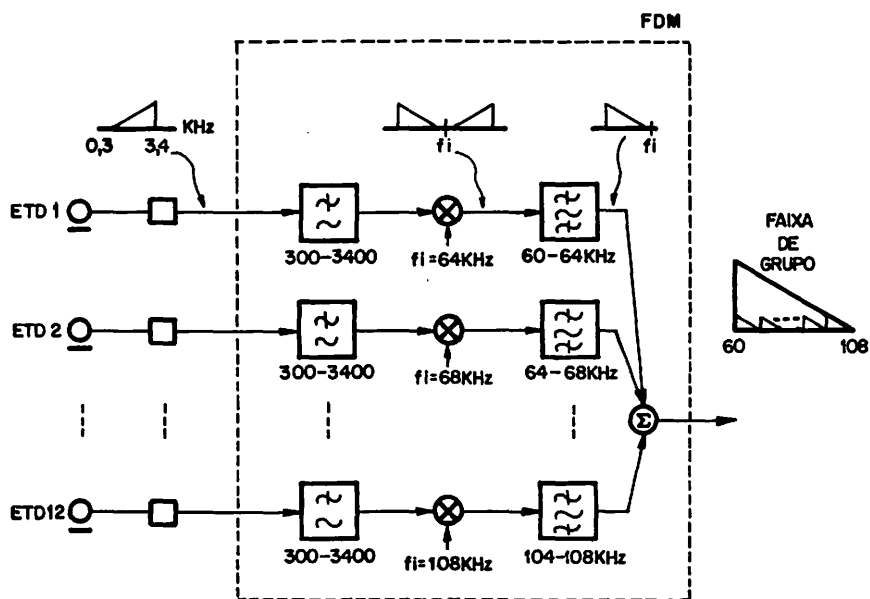


Fig.9.8: Utilização da faixa de grupo com FDM e TDM

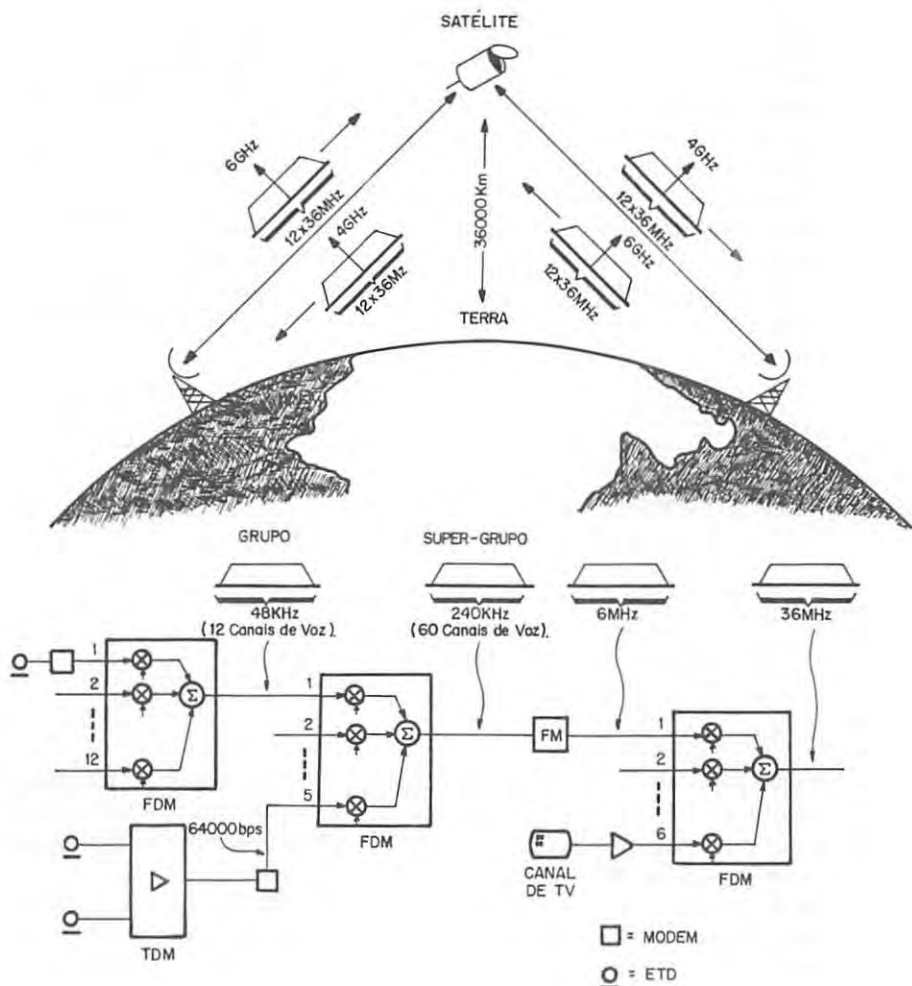


Fig.9.9: Transmissão via satélite - um exemplo

9.1.6 MODEM COM MUX

Alguns modems possuem um multiplexador TDM incorporado, e esta facilidade pode simplificar bastante alguns tipos de sistema.

Analisarei o modem V29, que pode oferecer um MUX configurável até 4 portas.

A tabela abaixo ilustra as possíveis configurações, para cada velocidade de transmissão (9.600, 7.200 e 4.800).

Na primeira, a porta “A” transmite a 9.600 bps, alocando os bits Q1 a Q4 do quadribit, que é o modo de operação normal do modem.

Na segunda, a porta “A” transmite a 7.200 bps, alocando os bits Q1 a Q3, e a porta “B” transmite a 2400 bps, alocando o bit Q4 do quadribit, e assim por diante.

	“A”	“B”	“C”	“D”
1	9.600(1,2,3,4)	—	—	—
2	7.200(1,2,3)	2.400 (4)	—	—
3	4.800(1,3)	4.800 (2,4)	—	—
4	4.800(1,3)	2.400 (2)	2.400 (4)	—
5	2.400 (1)	2.400 (2)	2.400 (3)	2.400 (4)
6	7.200 (2,3,4)	—	—	—
7	4.800(2,3)	2.400 (4)	—	—
8	2.400 (2)	2.400 (3)	2.400 (4)	—
9	4.800(2,3)	—	—	—
10	2.400 (2)	2.400 (3)	—	—

Como cada porta possui uma posição bem definida dentro do quadribit, não há necessidade de bits extras para endereçamento, fazendo com que este modem seja equivalente a um MUX TDM com “c=1” seguido de um modem convencional.

Uma potencial aplicação deste modem com MUX seria um sistema que utilize uma porta para entrada remota de trabalhos (“RJE”) e outra atendendo a vários terminais, operando em “polling”.

9.2 CÓDIGOS

Os códigos são definidos em função de uma necessidade específica.

Ainda no século 19, Morse idealizou seu código para transmissão telegráfica, tomando como base dois estados diferentes, conhecidos como “ponto” e “traço”. Este código binário (dois estados) possui quantidade variável de bits. Os caracteres mais presentes na linguagem, receberam menor número de bits para representá-los e isso faz sentido pois permite uma velocidade maior de transmissão, exigindo menos toques do operador.

O código Morse, no entanto, não se mostra apropriado a uma transmissão entre duas máquinas, pois sua tradução seria difícil, por possuir um número variável de bits por caractere.

Um outro código, idealizado por Emile Baudot, foi utilizado nos primeiros teletipos e ainda hoje é utilizado nas máquinas telex. Este possui um número fixo de 5 bits por caractere.

A combinação dos 5 bits permite 32 caracteres diferentes, mas usando o artifício de identificar cada um como “maiúsculo” ou “minúsculo” foi possível representar 58 caracteres ($2 \cdot 26 + 6$ especiais). Naturalmente, como numa máquina de datilografia, a posição “maiúsculo” ou “minúsculo”, deve ser definida antes, ou seja, o “status” deve ser observado ao se fazer a tradução do código.

Outros dois códigos, ASCII e EBCDIC são os mais utilizados atualmente para a transmissão de dados. O código ASCII possui 7 bits, enquanto o EBCDIC possui 8.

No Brasil, o código a ser padronizado é uma extensão do ASCII e está sendo chamado de BRASCII.

Nas próximas páginas apresento, a título de ilustração, os códigos citados. Internamente, os equipamentos digitais também utilizam códigos para representar os números decimais e os mais comuns são os códigos “binário”, “BCD” e “Gray”.

O código BCD (“binary coded decimal”) representa os algarismos decimais em binário.

O código Gray é chamado de código de distância unitária, ou seja, a diferença de um número para o imediatamente superior é somente em um bit. Esse código é utilizado na codificação dos símbolos durante a modulação nos modems, a fim de limitar a um bit errado quando houver decisão de fase adjacente na detecção.

Decimal	Binário	BCD	Gray
0	0000	0000	0000
1	0001	0001	0001
2	0010	0010	0011
3	0011	0011	0010
4	0100	0100	0110
5	0101	0101	0111
6	0110	0110	0101
7	0111	0111	0100
8	1000	1000	1100
9	1001	1001	1101
10	1010	—	1111
11	1011	—	1110
12	1100	—	1010
13	1101	—	1011
14	1110	—	1001
15	1111	—	1000

A	.-
B	-...
C	-.-.
D	-..
E	.
F
G	---
H
I	..
J	.----
K	-.-
L	.-..
M	--
N	-.
O	---
P	.-.-.
Q	---.-
R	.-.
S	...
T	-
U	...-
V	...--
W	.-.-
X	---.-
Y	---.-
Z	---..

1	.-.-.-
2	..-.-
3	...--
4-
5
6	-.....
7	---...
8	----..
9	-----.
0	-----
.	...-.-
:	---.-.-
?	..-.-..
ERRO
=	---.-
,	---....
;	-...-..
PAREN- TESIS	.-.-.-.-
/	---.-.
ESPERE	...-
FIM DE MENSAG.	...-..
CONVITE A TRANSM.	---
FIM	...-.-

Fig.9.10: Código Morse

CARACTERE		b ₅	b ₄	b ₃	b ₂	b ₁
MINUSCULO	MAIÚSCULO					
A	-	I	I			
B	?	I			I	I
C	:		I	I	I	
D	\$	I			I	
E	3	I				
F	!	I		I	I	
G	&		I		I	I
H	#			I		I
I	8		I	I		
J	.	I	I		I	
K	(I	I	I	I	
L)		I			I
M	.			I	I	I
N	,			I	I	
O	9				I	I
P	0		I	I		I
Q	1	I	I	I		I
R	4		I		I	
S	CAMPAINHA	I		I		
T	5					I
U	7	I	I	I		
V	;		I	I	I	I
W	2	I	I			I
X	/	I		I	I	I
Y	6	I		I		I
Z	''	I				I
LETRAS(muda para maiúsculo)		I	I	I	I	I
SÍMBOLOS (muda para maiúsculas)		I	I		I	I
ESPAÇO				I		
RETORNO DO CARRO					I	
ALIMENTA LINHA			I			
BRANCO						

Fig.9.11: Código Baudot

b7	b6	b5	b4	b3	b2	b1
----	----	----	----	----	----	----

b7=		0				1			
		00	01	10	11	00	01	10	11
b6, b5 =		00	01	10	11	00	01	10	11
b4, b2 b3, b1	HEXA	0	1	2	3	4	5	6	7
0000	0	NUL	DLE	SP	0	@	P	`	p
0001	1	SOH	DC1	!	1	A	Q	a	q
0010	2	STX	DC2	"	2	B	R	b	r
0011	3	EXT	DC3	#	3	C	S	c	s
0100	4	EOT	DC4	\$	4	D	T	d	t
0101	5	ENQ	NAK	%	5	E	U	e	u
0110	6	ACK	SYN	&	6	F	V	f	v
0111	7	BEL	ETB	'	7	G	W	g	w
1000	8	BS	CAN	(8	H	X	h	x
1001	9	HT	EM)	9	I	Y	i	y
1010	A	LF	SUB	*	:	J	Z	j	z
1011	B	VT	ESC	+	;	K	[k	{
1100	C	FF	FS	,	<	L	\	l	
1101	D	CR	GS	-	=	M]	m	}
1110	E	SO	RS	.	>	N	↑	n	~
1111	F	SI	US	/	?	O	—	o	DEL

Fig.9.12: Código ASCII

b8	b7	b6	b5	b4	b3	b2	b1
----	----	----	----	----	----	----	----

		0								1							
		0				1				0				1			
		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
b4 b3	b2 b1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	NUL	DLE			SP	a	-									0
0001	1	SOH	DC1					/		a	j			A	J		1
0010	2	STX	DC2	FS	SYN					b	k	s		B	K	S	2
0011	3	ETX	DC3							c	l	t		C	L	T	3
0100	4									d	m	u		D	M	U	4
0101	5	HT	NL	LF						e	n	v		E	N	V	5
0110	6	LC	BS	ETB	UC					f	o	w		F	O	W	6
0111	7	DEL	IL	ESC	EOT					g	p	x		G	P	X	7
1000	8									h	q	y		H	Q	Y	8
1001	9	RLF	EM							i	r	z		I	R	Z	9
1010	A	SMM	CC			¢	!	!	:								
1011	B	VT				.	\$,	✱								
1100	C	FF			DC4	<	✱	%	@								
1101	D	CR		Q	NAK	()	-	'								
1110	E	SO		ACK		+	:	>	=								
1111	F	SI		BEL	SUB			?	"								

Fig.9.13: Código EBCDIC

b8	b7	b6	b5	b4	b3	b2	b1
----	----	----	----	----	----	----	----

		0								1							
b8 =		0								1							
b7 =		0				1				0				1			
b6, b5 =		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
b4 b2 b3 b1	HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	NUL	DLE	SP	0	@	P	`	p			NSP	o	À	ø	à	ø
0001	1	SOH	DC1	!	1	A	Q	a	q			ì	±	Á	Ñ	á	ñ
0010	2	STX	DC2	"	2	B	R	b	r			í	²	Â	Ò	â	ò
0011	3	ETX	DC3	##	3	C	S	c	s			ê	³	Ã	Ó	ã	ó
0100	4	EOT	DC4	\$	4	D	T	d	t			ı	´	Ä	Ô	ä	ô
0101	5	ENQ	NAK	%	5	E	U	e	u			ı	µ	Å	Õ	å	õ
0110	6	ACK	SYN	&	6	F	V	f	v			ı	π	Æ	Ö	æ	ö
0111	7	BEL	ETB	'	7	G	W	g	w			š	·	Ç		ç	
1000	8	BS	CAN	(8	H	X	h	x			"	¸	È	Ø	è	ø
1001	9	HT	EM)	9	I	Y	i	y			©	ı	É	Ù	é	ù
1010	A	LF	SUB	*	:	J	Z	j	z				¸	Ê	Ú	ê	ú
1011	B	VT	ESC	+	;	K	[k	{			«	»	Ë	Û	ë	û
1100	C	FF	FS	,	/	L	\	l				¬	¼	Ì	Ü	ì	ü
1101	D	CR	GS	-	=	M]	m	}			-	½	Í	Ý	í	ý
1110	E	SO	RS	.	/	N	↑	n	~			®	¾	Î	Þ	î	þ
1111	F	SI	US	/	?	O	-	o	DEL			·	-	Ï	ß	ï	ÿ

Fig.9.14: Código BRASCI

9.2.1 DETECÇÃO DE ERROS

Toda transmissão de dados está sujeita a erros, provenientes de decisões erradas no demodulador do modem, devido a um sinal de recepção contaminado por uma ou várias degenerações.

Ao nível do modem, um erro significa um bit invertido, mas, a nível de sistema, esse erro pode significar um caractere errado, ou um bloco de caracteres não confiável.

É importante, em um sistema, que ele seja capaz de detectar erros, e quanto maior for essa capacidade maior será a sua confiabilidade.

Uma vez detectado um erro, o sistema pode agir de duas maneiras: pedir uma retransmissão da mensagem (caractere, bloco, etc.) ou corrigir o erro, se ele for capaz para isto.

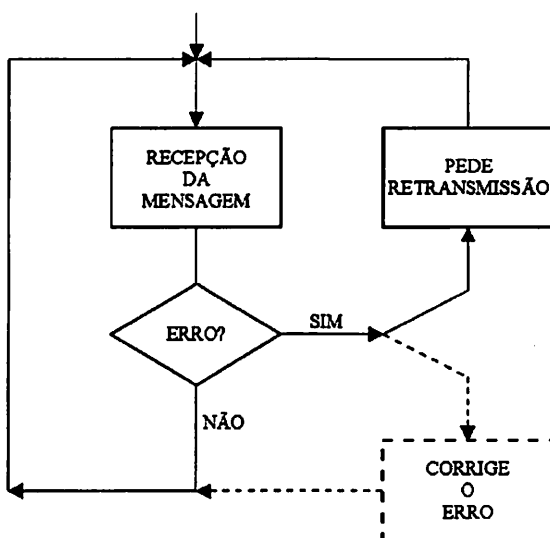


Fig.9.15: Detecção de um erro

Mas como o sistema pode detectar um erro ?

Suponha que um sistema esteja utilizando o código ASCII e na transmissão do caractere "j" houve um erro no 7º bit .

	Transmitido "j"	Recebido "*"
b1 =	0	0
b2 =	1	1
b2 =	0	0
b4 =	1	1
b5 =	0	0
b6 =	1	1
b7 =	1	Erro → 0

Simplemente, o sistema vai pensar que o caractere transmitido foi realmente o "*", e assim acontecerá com qualquer caractere errado, pois esse código é "não redundante", ou seja, todas as combinações possíveis são utilizadas.

Uma forma de detectar esse tipo de erro é acrescentar bits redundantes no código.

Um método de redundância utilizado é a adição de bits de paridade, resultado de uma operação do tipo "ou-exclusivo" sobre um conjunto de bits de dados.

Existem dois tipos de paridade: par e ímpar.

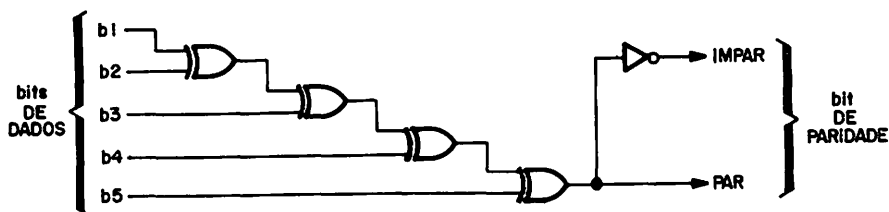


Fig.9.16: Geração do bit de paridade

O bit de paridade par (ímpar) tem um valor tal que, quando acrescentado aos bits de dados, o número total de bits "1" será sempre par (ímpar).

O procedimento consiste em acrescentar, na transmissão, um bit de paridade a cada conjunto fixo de bits de dados, e, na recepção, verificar se a paridade do conjunto está correta. Se o bit de paridade, na recepção, estiver invertido, houve erro. Observe que se houver um número par de erros, a ocorrência não será detectada.

Normalmente se utiliza a paridade por caractere e a paridade longitudinal. A paridade por caractere consiste em acrescentar um bit de paridade para cada caractere transmitido, e a paridade longitudinal consiste em acrescentar um caractere de paridade por bloco (BCC = "block check character") de forma que seu enésimo bit seja a paridade (normalmente é par) dos enésimos bits de todos os caracteres do bloco.

Caracteres de Dados								BCC
M	E	N	S	A	G	E	M	
1	1	0	1	1	1	1	1	1
0	0	1	1	0	1	0	0	1
1	1	1	0	0	1	1	1	0
1	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	0
0	1	0	0	0	0	1	0	0

: bits de paridade

Outro método de redundância é a cíclica (CRC = “cyclic redundancy check”), que consiste em acrescentar “r” bits redundantes, resto da divisão da sequência dos bits de dados por um polinômio gerador. A recomendação CCITT V41 apresenta o polinômio $x(16)+x(12)+x(5)+1$. Na recepção, a sequência é novamente dividida pelo mesmo polinômio gerador e o resto verificado deve ser zero, caso contrário houve erro. Uma descrição mais detalhada desse método foge ao objetivo deste trabalho, mas pode ser encontrada nas referências [4] e [5] da bibliografia.

9.3 PROTOCOLO

Um protocolo é um conjunto de regras e formalidades que rege uma comunicação. Existem vários tipos de protocolos. Abaixo, cito os quatro mais comuns em comunicação de dados:

- a - Protocolo de interface: estabelece as regras para se conectar dois equipamentos, a nível físico, ou seja, define a interface serial. Para equipamentos de transmissão de dados utiliza-se uma das interfaces seriais abordadas no capítulo 6, sendo EIA-232 a mais comum.
- b - Protocolo de controle da comunicação: estabelece as regras e procedimentos de linha entre duas estações, de forma a controlar o sincronismo e a detecção de erros.
É também chamado de DLC (“Data Link Control”)
- c - Protocolo de apresentação (“handshake”) entre modems: utilizado no início da comunicação a fim de estabelecer parâmetros entre os modems, como por exemplo, os protocolos definidos nas recomendações V22 e V22bis.
- d - Protocolo de modem esperto: estabelece as regras para que modem e ETD possam trocar parâmetros através da interface RS232. Um bem conhecido é o Hayes. Faço referência a tais protocolos na seção 6.17.2, e descrevo o protocolo Hayes no apêndice A5.

A seguir apresentarei, de forma simplificada, um protocolo de controle de comunicação (DLC), criado pela IBM em 1968 e ainda muito utilizado atualmente: BSC ou Bisync.

Suas principais características são:

- a - Permite comunicações semiduplex, ponto-a-ponto ou multiponto, de forma síncrona.
- b - Utiliza caracteres com significados especiais, ou seja, caracteres de controle.
- c - Transmite por blocos, que devem ser verificados e aceitos pelo receptor.

A figura 9.17 mostra seu formato básico.

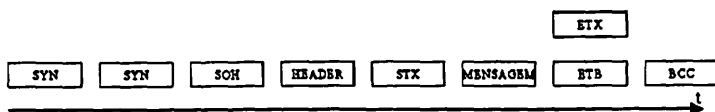


Fig.9.17: Formato do protocolo BSC

onde:

SYN = caractere de sincronismo

SOH = "Start of header" = informa ao receptor que a informação a seguir são caracteres de controle utilizados para endereçamento durante a operação do polling.

HEADER = Endereco da estação que deve receber a mensagem - esse campo é opcional e pode ser dispensado em circuitos ponto-a-ponto.

STX = "Start of text" = início do texto = indica ao receptor que a informação a seguir é a mensagem propriamente dita.

ETX = "End of text" = fim do texto = indica o fim da mensagem e que não há mais blocos a serem transmitidos.

ETB = "End of block" = fim do bloco = indica o término de um bloco e que o caractere BCC vem a seguir.

BCC = "Block check character" - utilizado para testar a integralidade do bloco. A verificação da paridade dos bits de mesma posição relativa, no caractere, é feita no caso de se utilizar o código ASCII, por exemplo.

Outros caracteres de controle utilizados são:

ACK = caractere enviado pelo receptor para informar que o bloco foi bem recebido.

NACK = caractere enviado pelo receptor para informar que o bloco recebido não foi aceito (bloco contendo erro).

ENQ = "enquiry" = caractere transmitido por uma estação, questionando se a outra estação está disposta a receber (ponto-a-ponto).

DLE = "data link escape" = caractere utilizado antes de um caractere de controle para que este não seja interpretado como tal. Com auxílio deste caractere pode-se construir textos transparentes ao protocolo.

EOT = "end of transmission" = caractere utilizado para indicar o fim de uma transmissão. Esse caractere também é utilizado para responder "nada a transmitir", durante uma inquirição de polling.

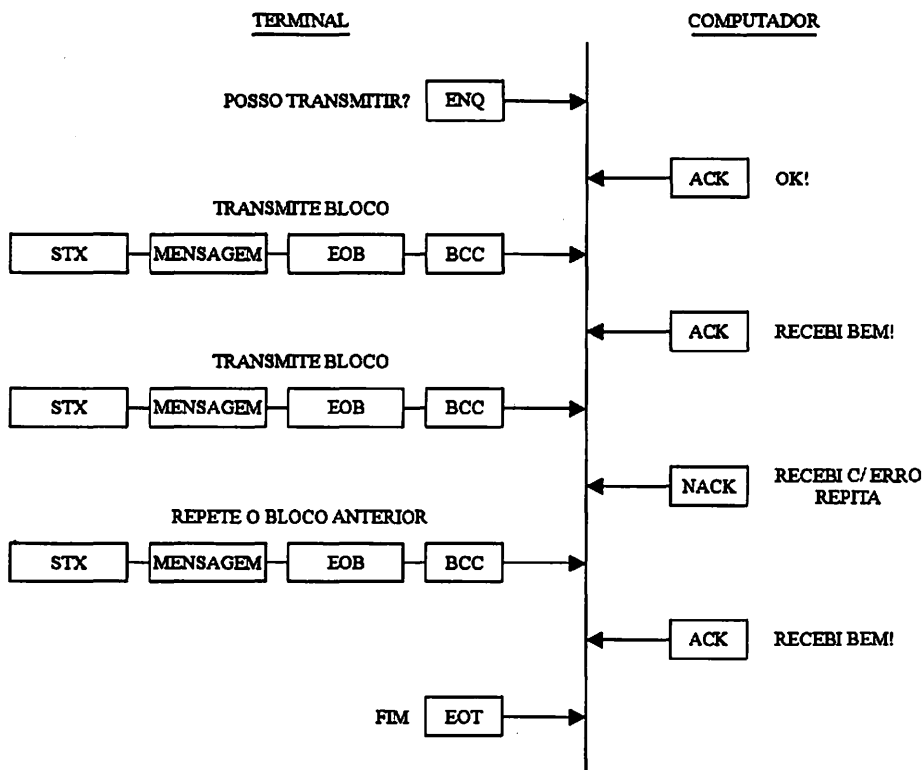


Fig.9.18: BSC em ponto-a-ponto

9.4 CIRCUITO PONTO-A-PONTO

Analisei, neste item, o circuito ponto-a-ponto, sob o ponto de vista de sua eficiência, ou seja, que parcela da velocidade do modem é efetivamente aproveitada para transmitir informação. Naturalmente, esta eficiência está intimamente ligada ao protocolo utilizado, e, como não faz parte deste trabalho avaliar os diversos protocolos, estarei considerando aqueles que precisam de uma confirmação “ACK” a cada bloco transmitido, antes de enviar o próximo. Caso o bloco seja recebido com erro, ele deve ser retransmitido.

Estarei considerando um circuito ponto-a-ponto em linha privativa, e, portanto, o terminal está permanentemente conectado ao computador.

Baseado no exposto acima, estabelecerei uma fórmula para determinar a eficiência do sistema, que será medida em porcentagem, a partir da definição de “taxa real de transferência de informação” = TRT , medida em bps.

$$TRT = \frac{\text{nº de bits de informação aceitos pelo computador}}{\text{tempo gasto para isso}}$$

Normalmente, os dados são transmitidos em blocos, seguindo algum protocolo. Os protocolos transmitem, além dos bits de informação (o que interessa), bits de controle da comunicação, bits de paridade para controle de erros, etc...

Esses bits de controle colaboram para que a taxa real de transferência de informação seja menor que a taxa de transmissão do modem (v_t).

Se um modem transmite a 2.400 bps, alguns desses bits são de controle, e certamente, a TRT será menor que 2.400 bps.

Outros fatores também colaboram para diminuir a TRT, conforme você verá adiante.

Sendo:

$$TRT = \frac{I}{T}$$

I = nº de bits de informação aceitos pelo computador

T = tempo necessário para aceitar esses bits.

Definirei eficiência:

$$f = \frac{TRT}{v_t}$$

f = eficiência

v_t = velocidade de transmissão do modem.

Podemos dizer que:

$$I = u \cdot (B - C)$$

u = bits de informação (úteis) por caractere

B = total de caracteres por bloco.

C = caracteres de controle por bloco.

$B - C$ = caracteres de informação

ainda:

$$T = N \left(\frac{B \cdot b}{v_t} + A_t \right)$$

N = nº de transmissões necessárias para aceitar um bloco

b = total de bits por caractere

v_t = velocidade do modem [bps]

A_t = tempo gasto entre a transmissão de 2 blocos.

$\frac{B \cdot b}{v_t}$ = tempo de transmissão de um bloco.

então:

$$TRT = \frac{u \cdot (B - c)}{N \left(\frac{B \cdot b}{v_t} + A_t \right)}$$

O parâmetro "N" pode ser colocado em termos da taxa de retransmissão do canal, que é, inclusive, uma informação de fácil medida.

$$N = \frac{\text{blocos transmitidos}}{\text{blocos certos}} = \frac{c + e}{c}$$

c = certos

e = errados

A taxa de retransmissão de bloco é definida por:

$$R = \frac{\text{blocos errados}}{\text{blocos transmitidos}} = \frac{e}{c + e}$$

Por substituição, você pode comprovar que:

$$N = \frac{1}{1-R}$$

Então, a fórmula da TRT finalmente fica:

$$TRT = \frac{(1-R) \cdot u \cdot (B-C)}{\frac{B \cdot b}{vt} + At}$$

A taxa de retransmissão de bloco, "R", é função da taxa de erro de bit, imposta pelo meio de comunicação (modem+ linha), e do comprimento do bloco. Esta relação, supondo que não haja correção de erro, é dada aproximadamente por:

$$R = 1 - (1 - em \cdot E)^{B \cdot b}$$

R = taxa de retransmissão

E = taxa de erro de bit

B = comprimento do bloco (caracteres)

b = bits por caractere

em = fator que desconta os erros múltiplos em um bloco, causados pelos randomizadores e equalizadores digitais. Podemos utilizar os valores:

em = 0,3 (V22, V22bis, V27 e V29)

em = 0,9 (V21,23,26)

Por exemplo, utilizando a equação acima, se uma ligação possui uma taxa de retransmissão de 1% e o comprimento do bloco é de 500 caracteres de 10 bits, então a taxa de erro de bit é da ordem de 7 ppm (em = 0,3).

O parâmetro At depende do modo de operação do modem:

a) semiduplex com portadora chaveada

b) duplex com portadora constante

a) Semiduplex:

A figura a seguir, esquematiza todos os retardos envolvidos entre a transmissão de dois blocos, numa ligação semiduplex.

No caso da ligação a 2 fios, os modems semiduplex introduzem os retardos RTS-CTS sempre que o sentido da transmissão é invertido.

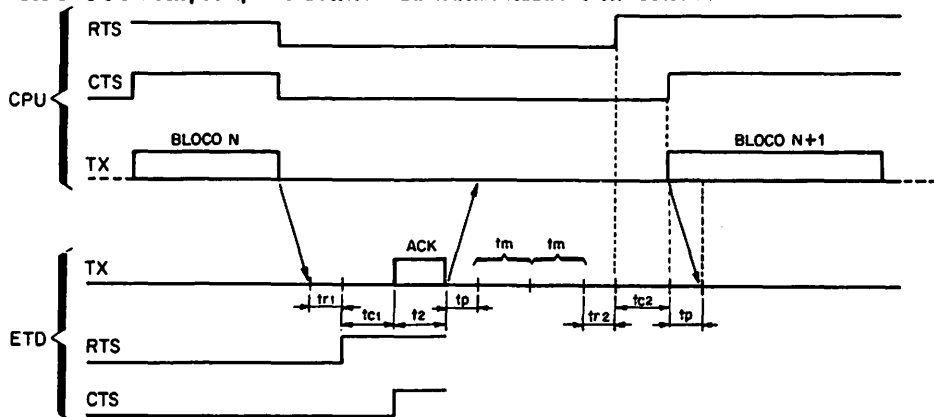


Fig.9.19: Temporizações ponto-a-ponto

Onde: $tr1$ = Tempo de reação do ETD para recepção do bloco anterior e levantar o RTS.
 $tc1$ = Retardo RTS-CTS do modem instalado no ETD.
 $t2$ = Tempo de envio da resposta ACK, que normalmente é formada por 7 a 10 bits
 tp = Tempo de propagação na linha de transmissão.
 Aparece duas vezes: tanto na transmissão do ETD quanto na da CPU.

Em termos práticos, pode-se considerar:

2,5ms para curtas distâncias
 5,0ms para médias distâncias
 15,0ms para longas distâncias
 300,0ms para transmissão via satélite.

tm = Tempo de propagação dentro do modem, considerando transmissão e recepção. Esse dado pode ser obtido com o fabricante do modem (varia de 3 a 60 ms).
 $tr2$ = Tempo de reação da CPU para avaliar a resposta ACK e levantar o RTS para o próximo bloco.
 $tc2$ = Retardo RTS-CTS do modem instalado na CPU

Podemos, então, dizer que:

$$At = (tr1+tr2) + (tc1+tc2) + t2 + 2.(tp+tm)$$

Na operação a dois fios, os tempos " $tr1$ " e " $tr2$ " são, no mínimo, iguais ao tempo necessário para que o modem reverta seu sentido de transmissão: (DCD-) + RCC. Veja 9.6.1.

b) Duplex

Este modo de operação engloba os modems que operam duplex a 4 fios (V26, V27 e V29), os modems que operam duplex a 2 fios (V22, V22bis) e os modems que operam duplex a 2 fios, com canal secundário (1.200/75).

Neste caso, o retardo At será menor pois não existirão os tempos gastos com RTS-CTS, já que os modems estarão operando com portadora constante. Então, aplica-se a mesma fórmula anterior com $(tc1+tc2) = 0$.

Deve ser salientado que, no caso do canal secundário, este sempre opera em velocidade mais baixa (geralmente 75 bps) e o tempo " $t2$ " será maior que no caso em que as duas velocidades são iguais.

Observe que o protocolo utilizado nesta análise é, por definição, semiduplex, ou seja, não existe transmissão e recepção simultaneamente.

Na figura 9.20, apresento um gráfico onde você pode ver como varia a eficiência de um circuito ponto-a-ponto em função do comprimento do bloco e da velocidade " vt " utilizada.

Observe que podemos tirar duas conclusões importantes:

- a) Existe um tamanho de bloco ótimo para cada velocidade.
 Para comprimentos de bloco abaixo do ponto ótimo, a eficiência cai devido ao atraso " At ", que independe do comprimento do bloco.

Para comprimentos de bloco acima do ponto ótimo, a eficiência cai devido a maior probabilidade de retransmissão.

- b) Quanto menor a velocidade “ v_t ”, maior é a eficiência do ponto ótimo e menor é o comprimento ótimo de bloco.

Na figura 9.21, você pode ver como a eficiência cai quando a taxa de erro aumenta. Observe que, quanto maior for a taxa de erro, menor será o comprimento ótimo de bloco.

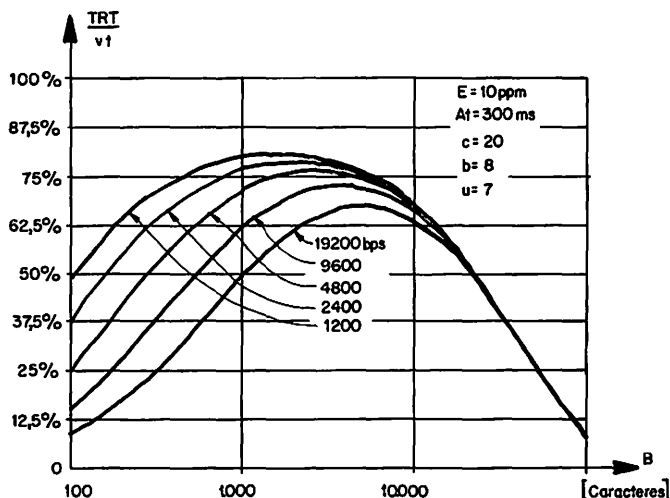


Fig.9.20: Eficiência x comprimento do bloco

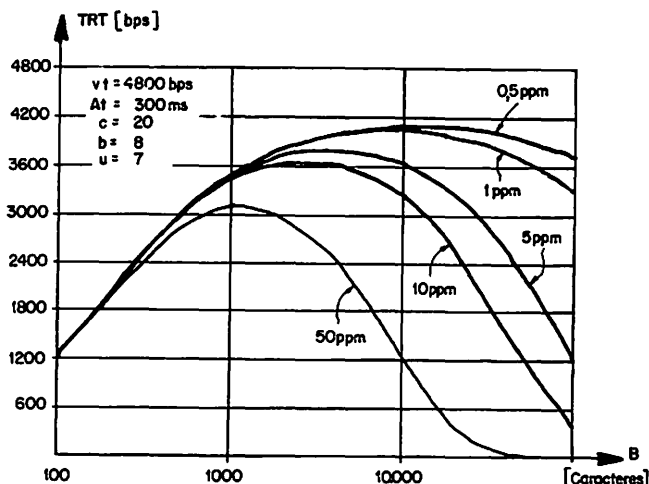


Fig.9.21: Queda da eficiência com a taxa de erro

9.4.1 EXEMPLO DE UM CIRCUITO PONTO-A-PONTO

Suponha que um terminal esteja ligado à CPU de um computador com um modem de 2.400 bps, operando semiduplex a dois fios, e que o código utilizado seja o ASCII (7 bits de informação e 1 de paridade).

Suponha que o tamanho do bloco seja de 1.000 caracteres e que cada bloco possua 15 caracteres de controle.

Então:

$$l = u.(B-C) = 7(1.000-15) = 6895 \text{ bits}$$

Suponha que a taxa de retransmissão seja de 1%. Então:

$$1 - R = 0,99$$

Suponha que esta ligação seja a média distância e que o tempo entre a transmissão de dois blocos seja formado por:

$$tr1 + tr2 = 20 \text{ ms}$$

$$tp + tm = 25 \text{ ms}$$

$$t2 = 20 \text{ ms}$$

$$tc1 + tc2 = 51 \text{ ms} \quad (\text{Retardo RTS-CTS de 25,5 ms em cada modem})$$

Então:

$$At = 141 \text{ ms} = 0,141 \text{ s}$$

$$TRT = \frac{(1 - R).l}{\frac{B \cdot b}{vt} + At} = \frac{(0,99).(6895)}{3,333 + 0,141} = 1965$$

$$TRT = 1.965 \text{ bps} ; f = 81,9 \%$$

Se a taxa de retransmissão fosse de 10%, por exemplo, obteríamos:

$$TRT = 1786 \text{ bps} ; f = 74,4 \%$$

Se essa ligação fosse a 4 fios, teríamos $tc1+tc2=0$ e obteríamos os seguintes valores:

Para R = 1%:	TRT = 1994 bps;	f = 83,1 %
Para R = 10%:	TRT = 1813 bps;	f = 75,5 %

Portanto, mudar a conexão para 4 fios(o que pode significar o dobro em gastos com linha), neste caso, aumentaria a eficiência em cerca de 1,2%.

9.4.2 MODEM DUPLEX OU SEMIDUPLEX ?

Considere um circuito ponto-a-ponto a dois fios.

Suponha que desejo efetivar uma transmissão de grande volume de dados, a 1.200 bps assíncrono.

Devo escolher o tipo de modem entre as três alternativas abaixo, que satisfaçam as condições estabelecidas:

- a) V23 1.200/1.200 bps semiduplex
- b) V23 1.200/75bps duplex
- c) V22 1.200/1.200 bps duplex

A fim de comparar as eficiências das alternativas citadas, devo estabelecer os parâmetros da transmissão a ser efetivada:

I	=	6895 bits; B=1.000 ; b=10
R	=	1 %
tr1+tr2	=	20 ms
2. (tp+tm)	=	50 ms
t2	=	8 ms para 1.200bps e 133 ms para 75bps
tc1+tc2	=	60 ms (semiduplex)

Finalmente, após o cálculo da TRT e da eficiência, obtenho:

Alternativa	At	TRT	f
V23 1.200/12.00	0,138	805,8 bps	67,1 %
V23 1.200/75	0,203	799,8 bps	66,6 %
V22 1.200/12.00	0,078	811,6 bps	67,6 %

Observe que a alternativa mais eficiente é a que utiliza modems V22, mas a diferença para aquela que utiliza modems V23 semiduplex é somente de 0,5 %.

Você deve observar que, o fato de se utilizar modems duplex, em geral, não leva a um aumento de eficiência como era de se esperar, pois os protocolos utilizados são do tipo que precisam de uma confirmação da boa recepção de cada bloco, antes de transmitir o próximo.

A mesma análise pode ser feita no caso de uma transmissão a 2.400 bps síncrona, onde a escolha vai recair entre modems V26 e V22bis.

9.4.3 LINHA COMUTADA OU TRANSDATA ?

Suponha que pretendo estabelecer uma ligação entre duas cidades, a fim de transmitir um certo volume de dados síncronos.

Tenho duas alternativas: alugar um circuito ponto-a-ponto Transdata ou comprar dois modems e utilizar a linha comutada via DDD.

Qual das duas será a mais vantajosa em termos de custo ?

A diferença básica é que, no Transdata irei pagar uma quantia fixa, mensal, pelo aluguel do circuito, independente do tráfego, e na linha comutada, irei pagar uma quantia proporcional ao tempo da conexão, ou seja, proporcional ao volume de dados transferido.

Vamos supor que:

Trecho =	Brasília/Rio (degrau D6)
Eficiência =	$f = \text{TRT}/v_t = 75\%$
CM =	Custo de 1 minuto de ligação DDD entre BSB e RIO, com tarifa normal. = Cz\$ 4,86
CCT=	Custo mensal do circuito Transdata, conforme tabela citada no 6.1.3.1.
TM =	(B-C)= Tráfego mensal desejado. [caracteres]
u =	bits úteis por caractere = 8
CLC=	Custo mensal em linha comutada
	$= \frac{cm}{60} = \frac{u.TM}{f.v_t}$

O gráfico abaixo foi traçado a partir destes dados e mostra como o custo mensal da linha comutada aumenta conforme a utilização, sendo menor que o custo do Transdata até um determinado tráfego.

Na velocidade de 4.800 bps, por exemplo, somente para um tráfego mensal acima de 187 milhões de caracteres (o que corresponderia a uma utilização da linha durante 5,2 horas em todos os dias úteis do mês) é que o circuito Transdata passa a ser mais barato.

Naturalmente, você não deve esquecer que, no caso da linha comutada, será preciso um investimento inicial para os modems.

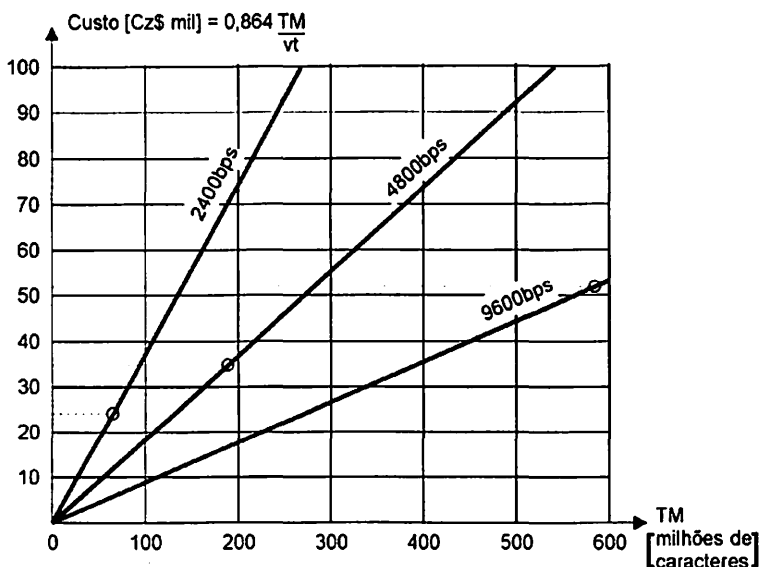


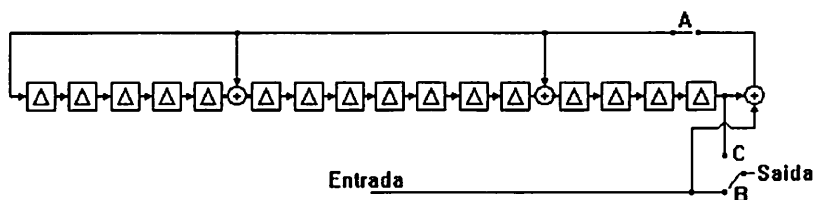
Fig.9.22: Custo Transdata x Linha comutada

A tarifa da ligação telefônica BSB-RIO, que considere, é aquela do horário normal (segunda a sexta, de 8:00 as 20:00). No caso da transmissão ser realizada em horário reduzido ou super-reduzido, aplique as tarifas Cz\$ 2,43 e Cz\$ 1,22 , respectivamente.

9.5 CORREÇÃO DE ERROS

Há muitos métodos de correção de erros e eles foram classificados em dois tipos: métodos blocado e convolucional.

A característica de um método blocado é que a sequência de bits a ser transmitida é seccionada em blocos de " k " bits e, a cada bloco, um codificador adiciona " $p = n - k$ " bits redundantes, algebricamente relacionados com os bits de dados, produzindo um bloco total de " n " bits, que serão efetivamente transmitidos pelo canal. Na recepção, os bits redundantes são verificados pelo decodificador, para detecção de erros e são corrigidos por retransmissão do bloco ou diretamente pelo decodificador. Dois tipos de codificadores são muito utilizados em comunicação de dados: o de paridade e o cíclico. Os de paridade geram um ou mais caracteres de paridade (chamados de BCC - "Block Check Character") que são adicionados no final do bloco. Os códigos cíclicos utilizados para detecção de erros são chamados de CRC ("Cyclic Redundancy Check"). Veja na próxima figura, o funcionamento do codificador CRC. Na primeira fase, as chaves A e B estão fechadas. Os dados recebidos do ETD passam para a saída e, ao mesmo tempo, vão entrando em um registrador de deslocamento, com realimentações em posições determinadas, que no caso da figura são 1, 5, 12 e 16. Essas realimentações definem o polinômio gerador. Após o último bit de dados do bloco, as chaves A e B se abrem e a chave C se fecha. Então os $p = 16$ bits armazenados no registrador, que correspondem à divisão dos bits de dados pelo polinômio gerador, são enviados, como bits de paridade cíclica, para a saída, completando os " n " bits do bloco. Na recepção, um outro codificador repete a operação e verifica a integridade dos dados. Propositamente, o polinômio do exemplo é aquele utilizado na V.42.



O método blocado pressupõe que o bloco transmitido será verificado e, se contiver erro, o receptor vai pedir uma retransmissão. Dos dois métodos o CRC está cada vez sendo mais utilizado.

O codificador convolucional opera sobre a mensagem de entrada de forma contínua, sem seccionamento em blocos, como foi visto anteriormente. O codificador convolucional executa a convolução discreta da sequência de entrada com a resposta impulsional do codificador. O codificador é uma máquina de estado finito que possui registradores com conexões através de portas "ou-exclusivo".

gerando uma saída redundante. Esse tipo de codificador é utilizado na modulação por treliça, TCM ("Trellis Coded Modulation") presente nos modems V.32, V.32bis e V.34. A codificação TCM efetua a correção direta de erros na recepção dos modems.

9.5.1 CCITT V.42

A recomendação V.42 especifica procedimentos de correção de erros para modems que utilizam conversores síncrono-assíncrono, baseados em método bloqueado com CRC. Foi pela primeira vez publicada no livro azul do CCITT em novembro de 1988.

A probabilidade de erros aumenta se a velocidade de transmissão do modem sobe, como você viu no capítulo sobre teste de modems analógicos. Cada vez mais as pessoas estão utilizando a rede comutada para transmitir dados e as velocidades subiram de 2.400 para 14.400 bps. Uma transmissão a 14.400 pode levar a uma TRT menor que uma transmissão a 9.600 bps, por exemplo, dependendo da taxa de erro e do comprimento do bloco. Um protocolo de detecção e correção de erros aumenta a confiabilidade da comunicação, permitindo que aplicações de maior volume de dados operem nas linhas telefônicas comuns.

A V.42 é um esquema de detecção e correção de erro que divide a sequência de dados a transmitir, em blocos, executando também a conversão de assíncrono para síncrono. Como a comunicação entre os modems é síncrona e a velocidade entre modem e ETD será maior que a velocidade de transmissão, há necessidade de registros temporários para gerenciar o controle de fluxo entre modem e ETD.

A detecção de erros é feita por uma verificação de redundância cíclica (CRC) na recepção do bloco. Se a verificação indicar a presença de erros, o receptor envia uma mensagem ao transmissor pedindo uma retransmissão do bloco contaminado. Essa forma de correção dos erros é chamada de ARQ ("Automatic Repeat reQuest").

Outra característica da V.42 é que prevê a interoperabilidade de um modem com correção e outro sem, por seleção automática do modo nativo (sem correção).

A V.42 possui dois algoritmos de correção de erros.

O primeiro é o LAPM ("Link Access Procedure for modems"). Baseado no protocolo HDLC, o LAPM é uma variante dos algoritmos LAPB e LAPD utilizados na RDSI. A maior razão da escolha desse algoritmo é a intenção de compatibilizar modems e RDSI no futuro.

O segundo é o MNP-4, desenvolvido pela Microcom, Norwood MA USA, que por ter sido largamente usado, é tratado como uma alternativa opcional para a conexão. O MNP-4 consta como anexo A da V.42. Após uma série de testes

comparativos dos dois algoritmos, o CCITT conclui que são equivalentes em performance. O MNP-4 foi incluído na recomendação devido à grande base instalada de modems que o utilizavam, mas todas as futuras melhorias desenvolvidas serão incorporadas ao LAPM. Um exemplo disso é a recomendação V.42bis, sobre compressão de dados, vista mais adiante, fortemente baseada no LAPM. Para que um modem seja compatível com a V.42 ele deve implementar o protocolo LAPM, mas o MNP-4 é opcional.

O protocolo V.42 possui uma série de funções de controle, como o procedimento de apresentação, no estabelecimento da chamada, onde os modems negociam os parâmetros da seção. Outra função é o controle de fluxo na interface serial para prevenir estouro do registro temporário (buffer) dos modems, nos dois lados da linha. Outra função é a própria conversão de assíncrono para síncrono. Outra função será a compressão de dados, conforme V.42bis.

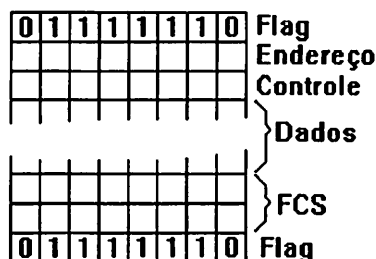
Vou descrever o protocolo LAPM, que é a base do V.42.

No LAPM, todas as mensagens são transmitidas em blocos com flags de abertura e fechamento. O bloco é dividido em campos, de um ou mais octetos (8 bits), com funções específicas. A flag de fechamento de um bloco pode ser utilizada como flag de abertura do próximo, para aumentar a eficiência da comunicação. Para que não ocorra uma flag em outra posição da comunicação, o transmissor insere um "0" se detectar cinco bits "1" seguidos e o receptor, ao examinar o bloco, descarta qualquer "0" que esteja após cinco "1" seguidos.

Logo após o flag de abertura vem um campo de endereço cuja função é identificar uma conexão com correção de erros e rotar os dados, por exemplo: o fluxo ETD para ETD, que é a comunicação normal, tem um endereço próprio.

O campo de controle indica o tipo de bloco: dados, comandos ou respostas a comandos. Por exemplo, rejeição de blocos por contaminação de erros, comandos de enlaces, comandos e respostas de conexão e desconexão.

Em seguida vem a parte reservada aos dados do ETD. O tamanho padrão desse campo é 128 octetos, nos dois sentidos da transmissão, mas esse parâmetro pode ser negociado independentemente para cada sentido.



Após os dados vem o campo FCS ("Frame Check Sequence") com dois octetos ou opcionalmente 4 octetos, que é o resultado da aplicação do cálculo do CRC. As seqüências de 16 ou 32 bits também são negociadas entre os modems, na fase de apresentação. A seqüência de 32 bits permite uma melhor verificação de erros, especialmente em blocos com dados, mas consome mais tempo da transmissão, diminuindo a TRT (Taxa Real de Transmissão).

O estabelecimento de uma conexão entre dois modems, com correção de erros e de acordo com a recomendação V.42, possui duas fases distintas: detecção e estabelecimento do protocolo. A fase de detecção se inicia logo após os procedimentos de resposta automática.

Na fase de detecção o modem origem envia uma seqüência de reconhecimento, chamada de ODP ("O Originator Detection Pattern"), composta de 32 bits (pode ter até 48 bits). Se o modem resposta não responder é porque ele não suporta V.42. Nesse caso o modem origem pode enviar uma seqüência de pedido de conexão MNP-4 ou abortar a tentativa de estabelecer uma ligação com correção de erros. Se o modem resposta for capaz de operar segundo a V.42, ele vai responder com uma seqüência chamada de ADP ("Answer Detection Pattern"), indicando que ele suporta V.42 e informando se ele deseja estabelecer a ligação com correção. Nesse caso os modems passarão para a próxima fase.

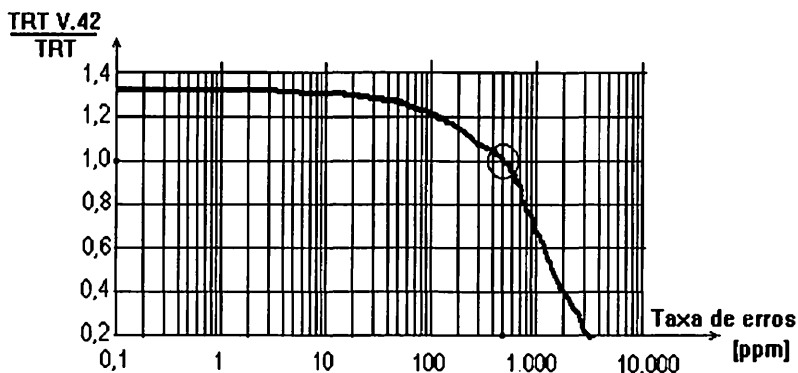
Na fase de estabelecimento do protocolo os modems vão negociar os parâmetros da conexão. A primeira providência tomada pelo modem origem é enviar uma seqüência de pedido de estabelecimento da conexão V.42. Após confirmação do modem resposta, o modem origem envia uma seqüência com sua sugestão para os parâmetros. Finalmente o modem resposta envia a confirmação dos parâmetros a serem utilizados, encerrando a negociação. Estabelecido o protocolo e seus parâmetros, os dados do usuário são encapsulados, segundo o protocolo LAPM, e transmitidos entre os dois modems.

A função do V.42 é melhorar a qualidade da comunicação, garantindo um fluxo isento de erros, e isso é feito pela divisão da informação do ETD em blocos, que serão retransmitidos na ocorrência de um erro. O preço a se pagar por isso é a queda na TRT, como você estudou em 9.4 (Circuito ponto a ponto). Você viu que, para uma determinada taxa de erro de bit do canal, quanto maior o comprimento do bloco melhor a TRT, até um certo ponto em que a TRT começa a cair novamente devido às retransmissões. Há, portanto, para cada conexão, um tamanho ótimo de bloco a ser determinado.

O assunto da eficiência foi tratado com detalhes em 9.4, considerando a degeneração somente do ruído branco. Em um canal real, todas as degenerações contaminam o sinal transmitido, agravando a situação. Por essa razão, pelo fato da distribuição estatística dos dados também influenciar na taxa de erro e porque a V.42 elimina os bits de start e stop, é difícil quantificar a eficiência do V.42. Na verdade, a eliminação dos bits de start e stop pode levar a uma

eficiência maior que 100% em situações ideais pois não deixa de ser uma compressão de dados da ordem de 25%.

A próxima figura ilustra a TRT normalizada em relação a um modem síncrono comum, sem correção de erros, do protocolo V.42 em função da taxa de erro de bit, considerando blocos de 128 octetos (1.024 bits). Observe que, para uma taxa de erro igual a 500 ppm a TRT normalizada é 1, ou seja, se um modem sem correção opera gerando 500 ppm de erro, com a V.42 ele vai operar sem erro e sem alterar a TRT!



Observe também, pela figura anterior, que para taxas de erros normalmente encontradas nas linhas telefônicas, o protocolo V.42 oferece um ganho na TRT de aproximadamente 30%. Para linhas altamente contaminadas por ruído (ou outras degenerações), a TRT cai drasticamente. Por exemplo, para uma taxa de erro de 3.000 ppm a TRT cai para 20% do valor que seria sem a V.42, mas, em compensação, não há erros.

É importante notar que os algoritmos MNP-4 e LAPM são incompatíveis. No entanto, os dois podem ser suportados por um modem, através do protocolo V.42. Quando um modem com V.42 fala com outro MNP-4, é porque aquele está usando o algoritmo MNP-4 residente no seu V.42. Qualquer modem com conversor assíncrono-síncrono pode ter V.42, como por exemplo V.22, V.22bis, V32, V32bis ou V.34.

Outro ponto que deve ser notado é que os modems podem funcionar com dados síncronos e assíncronos, mas o protocolo V.42 somente se aplica quando o modem está transmitindo dados assíncronos.

O CCITT continua estudando melhorias para o protocolo V.42 e, dentre outras, podemos citar a multiplexação estatística, a correção direta de erros e a criptografia dos dados.

A correção direta consegue, até um certo grau de contaminação do sinal recebido, corrigir os erros sem necessidade de retransmissão, mas pela análise da sequência recebida, que possui redundâncias para esse fim. A grande vantagem da correção direta é aumentar drasticamente a TRT, já que diminui as retransmissões.

A criptografia de dados vai permitir segurança nas comunicações via modems com V.42 incorporados, uma característica cada vez mais solicitada pelas instituições financeiras e governamentais.

9.6 COMPRESSÃO DE DADOS

A compressão de dados é utilizada há muito tempo para aproveitar melhor o espaço disponível em um dispositivo de armazenamento. Por exemplo, vamos supor que você digitou um texto em um microcomputador que ocupa 2 Mbyte de memória e deseja guardar esse arquivo em um disquete de apenas 1,4 Mbyte. Esse problema é normalmente resolvido com o uso de um software compactador de arquivos, ou seja, um programa que faz a compressão de um ou vários arquivos, juntando tudo em um único, que ocupa menos espaço de memória. Dois programas são bem conhecidos dos que já se depararam com esse problema: ARC e PKZIP, das empresas Systems Enhancement Associates e PKware. Esses programas são executados pelo micro, sob comando do usuário que indica os arquivos a serem compactados, e o arquivo resultante é armazenado no próprio micro. Não há um compromisso com o tempo de processamento, ou seja, você está bem na frente do micro, esperando que ele termine seu trabalho e um segundo a mais ou a menos não muda muito para você.

O objetivo, portanto, da compressão de dados no exemplo citado, é economizar espaço de memória.

Outra aplicação da compressão de dados é quando ela é executada em tempo real, ou seja, há um compromisso com o tempo de processamento. Nesse caso, os dados vão chegando ao compressor e, após um tempo fixo, vão saindo comprimidos do outro lado. Essa aplicação é a comunicação de dados. Nessa aplicação, deseja-se economizar tempo de transmissão. Quanto mais compactados estiverem os dados, menos bits eles representarão e, portanto, menos tempo levará para ser transmitido através de um canal com velocidade fixa, em bits por segundo.

Vamos falar sobre compressão para comunicação de dados.

Uma compressão de dados em tempo real pressupõe a ação de um método de compressão que pode ter um ou mais algoritmos de compressão. O método é o conjunto de regras para gerar o resultado e o algoritmo é o procedimento específico de uma transformação.

Mais especificamente, vamos falar sobre compressão de dados assíncronos.

O método de compressão MNP-5 ("Microcom Networking Protocol") da Microcom Inc, USA, se tornou um padrão de fato nos últimos anos, cujo objetivo foi diminuir o tempo de transmissão dos modems com conversão assíncrono-síncrono, como o V.22 e o V.22bis. A idéia do MNP começou com a eliminação dos bits de start e stop. Realmente, só de extrair os bits de start e stop, antes da transmissão, consegue-se uma compressão razoável. Os algoritmos desse método foram incorporados aos circuitos do modem, de forma que ficava transparente para o usuário a transmissão com compressão.

O método MNP-5 consegue uma compressão média da ordem de 2,4:1, permitindo aumentar a TRT nessa proporção. É comum se ouvir dizer que a compressão aumenta a velocidade do modem. Na verdade, o modem transmite sempre na mesma velocidade mas os dados são comprimidos antes de chegarem ao modulador.

9.6.1 CCITT V.42BIS

O CCITT publicou a recomendação V.42bis, sobre compressão de dados, no início de 1990, após um longo estudo dos métodos e algoritmos disponíveis. Logo em seguida vários fabricantes de modem começaram a implementar esse método em seus produtos, e com isso a tão desejada compatibilidade em compressão começou a surgir inicialmente para os modems V.22bis e logo em seguida para os V.32.

O CCITT escolheu o melhor algoritmo dentre os conhecidos, para incorporar ao V.42bis, o que foi um grande avanço em relação ao padrão de fato existente na época, que era o MNP-5. Hoje, praticamente todos os modems seguem as recomendações CCITT. A compatibilidade é fundamental e com a compressão de dados não é diferente. A V.42bis está se tornando o padrão absoluto.

A V.42bis foi concebida como uma continuação da V.42. Apesar de serem duas recomendações independentes, um modem com V.42bis sempre estará atendendo a V.42. Na implementação dos algoritmos de compressão, deve-se considerar, além da taxa de compressão, fatores como a demanda de processamento e a quantidade de memória RAM necessários. Esses fatores influenciam diretamente no custo do produto final.

Em seus estudos, o CCITT analisou, dentre outros, o MNP-5, o MNP-7 e uma série de algoritmos desenvolvidos e publicados em 1977, por Lempel e Ziv.

A escolha recaiu sobre os algoritmos de Abraham Lempel e Jacob Ziv que, após terem sido modificados e adaptados para modems pela BT ("British Telecom"), ofereciam a melhor performance global. O algoritmo analisa a cadeia de caracteres de entrada e substitui conjuntos de caracteres por uma palavra-chave (código) obtida de um dicionário dinâmico, residente no codificador e no decodificador. A implementação da BT para os algoritmos de Lempel-Ziv possui um dicionário de comprimento variável. Quanto maior o dicionário

melhor será a compressão. O tamanho mínimo do dicionário é de 512 bytes, podendo, opcionalmente, ter 1.024, 2.048, 4.096 ou mais kbytes.

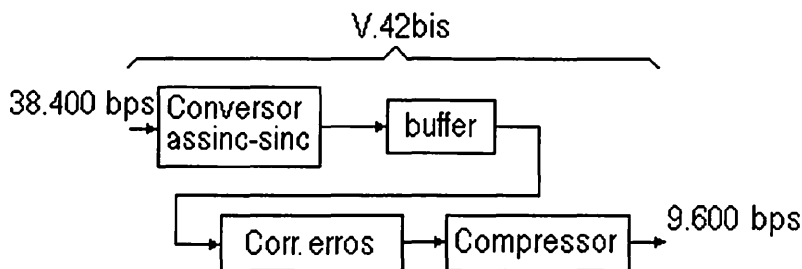
A tabela abaixo, mostra um comparativo entre os algoritmos citados, com relação à taxa de compressão média e requisitos de memória e processamento. Note que os algoritmos possuem taxas médias de compressão semelhantes, em torno de 3,2, a menos do MNP-5 que possui uma taxa de 2,45, bem abaixo da média. Com relação ao requisito memória, há grande diferença entre eles. A quantidade de processamento mencionada é em relação ao algoritmo Lempel-Ziv, ou seja, o MNP-5 precisa de 2,6 mais processamento que o Lempel-Ziv, para executar a compressão assinalada. Obviamente, o requisito processamento depende da implementação, mas os dados apresentados consideram implementações equivalentes.

Algoritmo	compressão média	RAM [kByte]	Processamento
Lempel-Ziv 512	2,96	3	1
Lempel-Ziv 1024	3,19	12	1
Lempel-Ziv 2048	3,29	28	1
Lempel-Ziv 4096	3,31	41	1
MNP-5	2,45	1	2.6
MNP-7	3,14	36	2.8

Outros dois parâmetros devem ser mencionados quando se fala de compressão de dados em tempo real para modems: capacidade de comprimir dados randômicos e tamanho do buffer de entrada.

O V.42bis, como qualquer algoritmo de compressão moderno, procura encontrar, na sequência de entrada, cadeias repetitivas de caracteres para substituí-las por palavra-chaves, que no caso do V.42bis são obtidas de um dicionário dinâmico. Se a sequência de entrada representa dados randômicos, o compressor não vai encontrar repetições e, portanto, não conseguirá comprimir. Mas, como o processo de compressão exige também a colocação de informações de controle, o que vai ocorrer se os dados não podem ser comprimidos? Haverá uma expansão ou invés de compressão. A taxa de compressão do MNP-5 para dados aleatórios é 0,75, ou seja, se o modem sem compressão apresenta uma TRT de 2.000 bps, com compressão a TRT cairá para 1.500 bps! O V.42bis possui um mecanismo de detecção automática de dados aleatórios e desativa o compressor. Então, na V.42bis, a taxa de compressão de dados randômicos é 1, contra 0,75 do MNP-5.

O tamanho do buffer de entrada é também importante, pois, apesar dos modems serem duplex, na verdade os protocolos de comunicação são semiduplex, ou seja, após a transmissão de um bloco o receptor deve informar se recebeu bem ou se quer uma retransmissão. Veja a próxima figura. Imagine que o ETD passa dados a 38.400 bps a um modem V.32 que transmite a 9.600 bps. Mesmo com a compressão total do V.42bis, digamos de 3,2, o modem não vai conseguir dar vazão aos dados que chegam do ETD. Os dados vão sendo colocados em um buffer e daí seguem para os algoritmos de correção e compressão. Se o compressor não consegue dar vazão, o buffer vai enchendo e quando chega em um determinado ponto (por exemplo 80%), emite um controle de fluxo para o ETD parar de enviar dados até o buffer se esvaziar novamente (por exemplo 20%). Então, o fato do ETD estar transmitindo teoricamente a 38.400 e o modem a 9.600 bps, não significa que a compressão total seja de 4,0. Como existe o controle de fluxo, a TRT será função da eficiência do compressor.



A taxa de compressão total é a medida do ganho de eficiência em relação à transmissão sem compressão. É a divisão da TRT com compressão pela TRT sem compressão.

A taxa de compressão depende totalmente da distribuição estatística dos dados, ou seja, o quanto eles são randômicos. Vimos que para dados totalmente aleatórios não há o que ser comprimido e, portanto a taxa de compressão será 1,0 (se o compressor puder detectar isso e desativar seu algoritmo, como faz o V.42bis). Por outro lado, se os dados são altamente repetitivos, pode-se chegar a taxas de compressão altas. A tabela a seguir, mostra a taxa de compressão média para alguns tipos de dados, obtida com o V.42bis com dicionário de 2048 bytes.

Tipo de dados	Taxa de compressão
Programa assembler	4,15
Arquivo texto	3,92
Programa fonte	3,84
Imagens	3,40
Planilha de dados	3,36
Arquivo pre-comprimido com ARC	1,47
Dados aleatórios	1,00

Nas comunicações micro a micro para transferência de arquivo, a TRT vai depender do software de comunicação que está instalado e do protocolo de comunicação que ele está usando, além dos outros fatores discutidos. A tabela a seguir, mostra a TRT da transferência de um arquivo texto de 18.944 bytes editado no WordStar, usando vários protocolos de comunicação, com e sem a aplicação de correção de erros em um modem operando a 2.400 bps. Cada byte possui 8 bits e foi transmitido com start e stop, totalizando 10 bits por caractere (8N1), sem paridade. O protocolo Xmodem transmite em blocos de 128 bytes enquanto o Ymodem transmite em blocos de 1.024 bytes e faz a verificação de erros por CRC. O protocolo Ymodem é mais eficiente que o Xmodem pois possui bem menos bits de controle. O protocolo YmodemG não possui correção de erros, ou seja, não pede nem espera por retransmissões e, por isso, é o melhor para ser utilizado com modems que têm correção de erros. Esses protocolos são encontrados nos programas de comunicação Crosstalk Mk4, Procom Plus, Smartcom III, dentre outros.

V_t = velocidade do modem = 2.400 bps

T = tempo de transmissão

I = bits de informação = $18.944 \times 8 = 151.552$

TRT = Taxa Real de Transmissão = I/T

f = eficiência = TRT/V_t

T_c = Taxa de compressão total em relação ao Xmodem = $TRT/1579$

Protocolo	Correção no modem	TRT	T	f	T _c
Xmodem	-	1.579	96 s	66 %	1
Xmodem	MNP-5	1.722	88 s	72 %	1,09
Ymodem	-	1.722	88 s	72 %	1,09
Ymodem	MNP-5	3.031	50 s	126 %	1,92
YmodemG	MNP-5	3.789	40 s	158 %	2,40
YmodemG	V.42bis	5.396	28 s	225 %	3,42

FUNCIONAMENTO DO V.42BIS

O algoritmo básico do V.42bis é ir montando cadeias de caracteres, a partir dos dados vindos do ETD, e colocando em uma tabela, chamada de dicionário, com uma palavra-chave equivalente. Portanto, cada palavra-chave no dicionário, equivale a um cadeia de caracteres. A cada nova cadeia de entrada o algoritmo verifica se ela já consta do dicionário, caso contrário acrescenta no dicionário. Se a cadeia é encontrada no dicionário apenas a palavra-chave é transmitida. A montagem do dicionário é dinâmica. Não há um dicionário predeterminado na V.42bis. O receptor, que fica sincronizado com o transmissor (compressão-descompressão), possui o mesmo dicionário. Ao receber a palavra-chave o receptor verifica seu dicionário e a substitui pela seqüência de caracteres. Conforme o tempo vai passando, a tendência do dicionário é ir agregando as seqüências freqüentes e cada vez mais longas, melhorando assim, a taxa de compressão. Obviamente, para um dicionário maior, mais seqüências podem ser armazenadas e, portanto, melhor será a compressão.

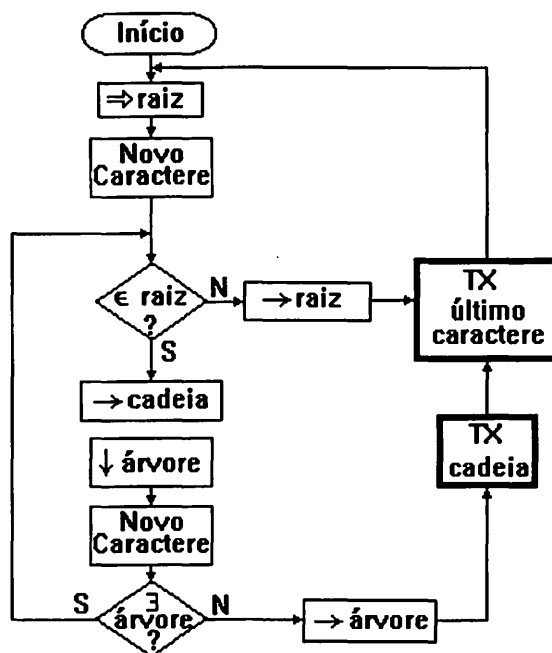
A tarefa do codificador (compressor) não é fácil! Ele deve procurar, no dicionário, a existência de cada seqüência observada, e existe seqüência que faz parte de seqüência maior. Uma das formas de identificar seqüências é montar árvores de caracteres. O primeiro caractere de cada árvore é chamado de raiz.

Da mesma forma que no V.42, o V42bis prevê uma negociação inicial de certos parâmetros que vão nortear a comunicação.

Um dos parâmetros é o tamanho do dicionário; o valor mínimo é 512 palavras, mas pode ser negociado um valor maior. Outro parâmetro é o tamanho máximo de cadeia de caracteres; o valor mínimo é 6 mas os dois modems podem estabelecer um valor de 6 a 250, durante a negociação. Outro parâmetro é a quantidade de bits da palavra-chave.

Siga o fluxograma do algoritmo V.42bis que se encontra a seguir. Após início do algoritmo, um ponteiro fica posicionado na raiz, aguardando a entrada de um caractere. Ao entrar um caractere verifica-se sua presença como raiz - se não há, é incluído na raiz e transmitido. Quando entra um caractere que existe na raiz, o ponteiro desce a árvore e coloca o caractere em um registro que vai

formar a cadeia a ser transmitida. Novo caractere entra e verifica-se sua presença nesse nível da árvore: se há, ele é adicionada à cadeia e o ponteiro desce mais um degrau na árvore. E continua assim até chegar um caractere que não consta no nível atual, então, ele é incluído na árvore e a cadeia formada até o caractere anterior é transmitida, bem como esse último caractere, e o ponteiro volta para a raiz.

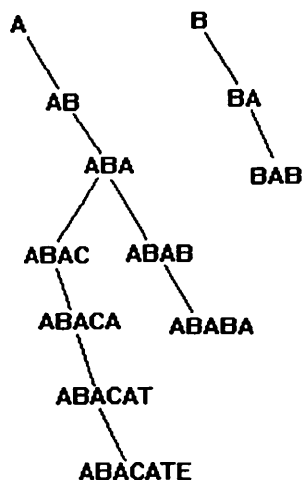


No início, como o dicionário está vazio, um caractere é uma cadeia e será armazenado com seu próprio código ASCII. A primeira cadeia de dois caracteres terá palavra-chave igual a 256. Todos os caracteres de 8 bits ocupam de 0 a 255.

Vamos imaginar, como exemplo, a transmissão seguida dos caracteres "A" e "B".

Passo	ponteiro	Entra o caractere	Montagem da árvore	Montagem da cadeia	Palavra-chave transmitida
1	raiz	A	A		A
2	raiz	B	B		B
3	raiz	A		A	
4	A	B	AB		A, B
5	raiz	A		A	
6	A	B		AB	
7	AB	A	ABA		AB, A
8	raiz	B		B	
9	B	A	BA		B, A
10	raiz	B		B	
11	B	A		BA	
12	BA	B	BAB		BA, B
13	raiz	A		A	
14	A	B		AB	
15	AB	A		ABA	
16	ABA	B	ABAB		ABA, B
17	raiz	A		A	
18	A	B		AB	
19	AB	A		ABA	
20	ABA	B		ABAB	
21	ABAB	A	ABABA		ABAB, A
...

Observe que, conforme o codificador V.42bis vai "aprendendo" como a sequência de entrada se repete, as transmissões vão ocorrendo cada vez menos. Observe, também, que o receptor tem condições de montar o mesmo dicionário. Nos 21 passos do algoritmo, a taxa de compressão média foi de 1,5:1, mas se considerarmos do passo 11 ao 21 (segunda metade do tempo em que o processo ocorreu), a taxa de compressão média cresceu para 2:1. A próxima figura mostra a árvore.



A título de exemplo, a árvore contém a palavra "ABACATE", que poderia se formar, caso houvesse repetição suficiente, pegando um ramo da árvore formada pelos "A"s e "B"s.

9.6.2 COMPRESSÃO DE DADOS SÍNCRONOS

Alguns modems atualmente já estão efetuando compressão de dados no modo síncrono. A compressão no modo assíncrono fica facilitada pelo fato do modem tratar os dados que chegam do ETD, caractere a caractere e incorporar a função de transformação em uma transmissão síncrona. Já no caso do modo síncrono, que é uma transmissão mais eficiente e bastante dependente do protocolo, os modems incorporam compressores em geral dependentes de protocolos bloqueados.

Protocolos bloqueados podem ser facilmente identificados pelos seu delimitadores. O protocolo BSC, por exemplo, possui blocos delimitados por caracteres de controle, conforme você viu em 9.3. Os protocolos HDLC, SDLC, X.25, dentre outros, possuem "Flags" para delimitar início e fim de bloco. Normalmente, o modem está preparado para comprimir apenas alguns protocolos. Se o modem souber qual protocolo está circulando em sua interface, conseguirá identificar exatamente onde é o início e o fim do bloco e poderá fazer a compressão das informações contidas em cada bloco. Uma vez conhecido o protocolo, o algoritmo de compressão utilizado pode ser baseado em um dicionário dinâmico, como o V.42bis.

9.7 CIRCUITO MULTIPONTO

Circuito multiponto é aquele onde vários terminais compartilham uma mesma linha de acesso a uma porta do computador, ou seja, cada terminal terá a sua vez para receber e transmitir mensagens.

O protocolo utilizado vai disciplinar a utilização da linha, de forma a viabilizar as transações.

Gostaria de salientar que não pretendo, neste item, analisar os possíveis protocolos a serem utilizados, mas sim mostrar o comportamento de tal circuito, abordando aspectos de eficiência, de forma semelhante à que fiz para o circuito ponto-a-ponto.

Para esta abordagem, utilizarei novamente o protocolo BSC e estarei supondo que o circuito está implementado a 4 fios, semiduplex, utilizando a técnica "poll-select".

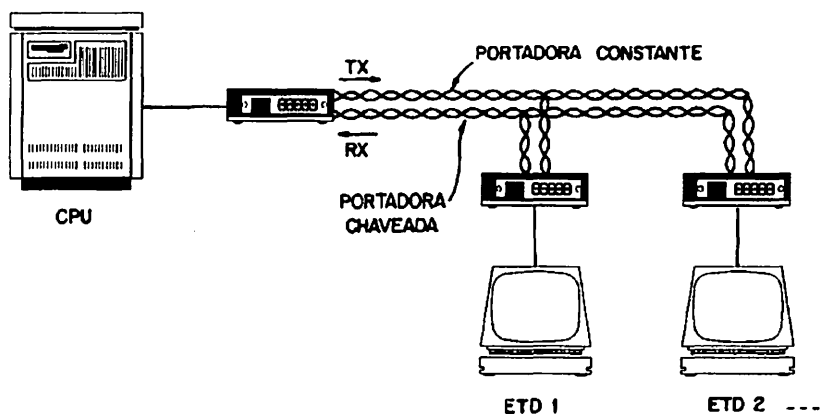


Fig.9.23: Circuito multiponto

9.7.1 POLL-SELECT

Esta técnica consiste em deixar o controle da linha por conta da CPU, da seguinte forma:

Entradas =	A CPU faz um "poll" de todos os terminais que estão conectados na linha, solicitando que eles transmitam suas mensagens de entrada, ou seja, cada terminal, em um instante diferente é inquirido a transmitir
Saídas=	A CPU seleciona o terminal e transmite a mensagem de saída.

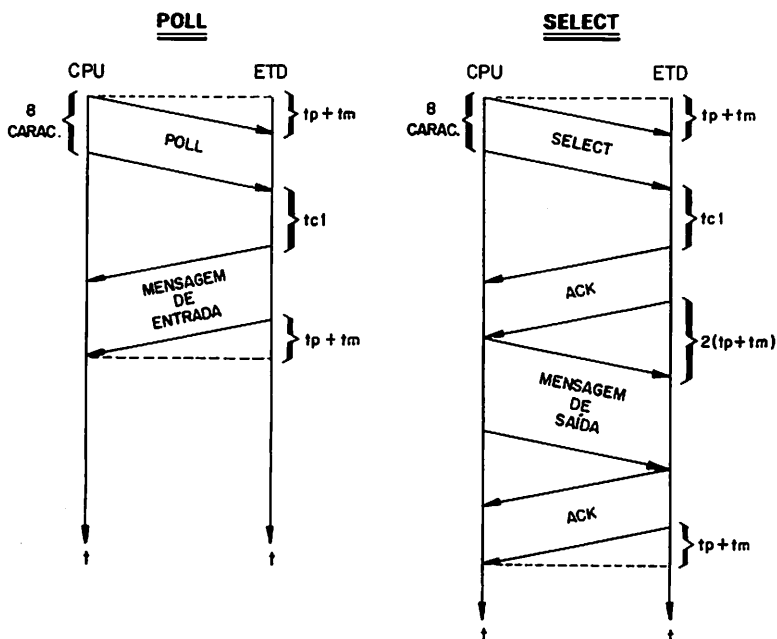


Fig.9.24: Poll-select a 4 fios

O modem instalado na CPU opera com portadora constante, pois a linha "TX" é exclusiva para sua transmissão, evitando os retardos RTS-CTS, e os modems instalados nos ETD operam com portadora chaveada, pois a linha "RX" deve ser compartilhada, no tempo, por todos.

9.7.2 TEORIA DAS FILAS

Uma das ferramentas utilizadas na análise de sistemas de teleprocessamento é a teoria das filas, e por este motivo, gostaria de apresentá-la agora, de maneira simplificada, antes de utilizá-la especificamente na análise dos circuitos multiponto.

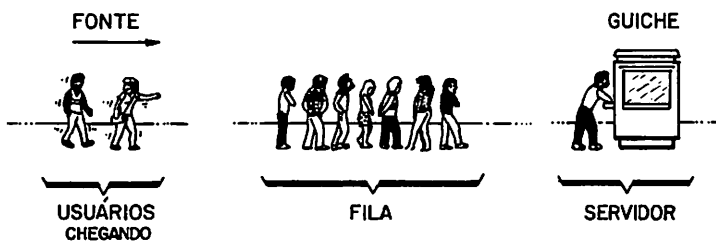


Fig.9.25: Fila de atendimento

A teoria das filas é utilizada para representar um modelo de sistema onde as transações chegam, esperam sua vez e são atendidas.
 Num sistema desse tipo, podemos distinguir:

- M = nº de servidores (nº de transações servidas simultaneamente)
- N = nº de fontes de transação
- L = taxa média de chegada das transações, por fonte
- N.L = taxa média de chegada de todas as transações
- ts = tempo médio de serviço

Tanto “L” quanto “ts” são variáveis aleatórias e a disciplina na fila é do tipo FIFO (“first in, first out”), ou seja, o primeiro a chegar será o primeiro a ser atendido.

A utilização da linha é definida como sendo:

$$p = \frac{ts \cdot N \cdot L}{M}$$

9.7.2.1 FILA UNISSERVIDOR

Suponha, por exemplo, que em um banco que possua apenas um caixa, cheguem 90 clientes por hora e fiquem em uma fila, aguardando o atendimento. Logo que o caixa desocupe, o primeiro da fila é chamado para ser atendido e o tempo médio de serviço é de 36 segundos por cliente.

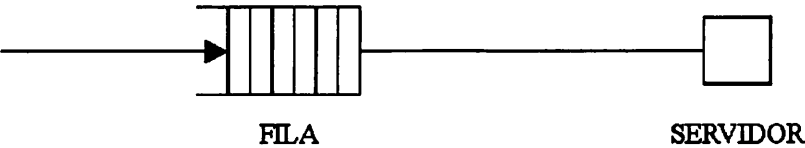


Fig.9.26: Fila unisservidor

Você pode, então, identificar:

M	=	1
N	=	1
L	=	90/3600 = 0,025[1/s]
ts	=	36 [s]
p	=	0,9

Observe que, se “p” for maior ou igual a “1”, o sistema estará saturado, ou seja, a fila crescerá indefinidamente.

Qual é o tempo médio que os clientes esperam na fila ?

Esta é uma informação importante fornecida pela teoria das filas:

$$tw = \frac{p \cdot ts}{2 \cdot (1-p)}$$

O tempo médio de espera na fila, “tw”, é o espaço de tempo decorrido desde o instante em que o cliente entra na fila, até o instante em que ele vai ao caixa para ser atendido.

k = Constante de variação do tempo de serviço.

Mostrarei com mais detalhes, adiante, a forma de determinar o valor de “k”. No momento, adianto que se o tempo de serviço “ts” for constante, então $k=1$.

Suponha, então, que o caixa leve, sempre, exatamente 36 segundos para atender cada cliente. Neste caso, $k=1$, e:

$$tw = \frac{0,9 \cdot 36 \cdot 1}{2 \cdot (1 - 0,9)} = 162 [s]$$

Cada cliente, em média, ficará 162 segundos (2,7 minutos) na fila. O tempo total que ele fica no banco será:

$$tr = tw + ts$$

tr = tempo médio de resposta
tw = tempo médio de espera
ts = tempo médio de serviço

No caso do banco, $tr = 3,3$ minutos

Se o tempo de serviço não for constante, ou seja, variar de transação para transação, devemos calcular o tempo médio de serviço:

$$ts = \frac{\sum_{i=1}^m ti}{m}$$

ts = tempo médio de serviço
ti = tempo de serviço observado em uma das “m” medidas
m = nº total de medidas realizadas

A variância do tempo de serviço é definida como sendo:

$$ds^2 = \frac{\sum_{i=1}^m (ti - ts)^2}{m}$$

A raiz quadrada da variância, ou seja, “ds”, é chamada de “desvio padrão” ou “desvio médio quadrático”.

O desvio padrão é a média (quadrática) dos desvios. É uma informação importante pois traduz o grau de variação do tempo de serviço.

Suponha que, naquele exemplo do banco, o tempo médio de serviço tenha sido fruto de uma série de 5 medidas: $t1=0,9$; $t2=1,2$; $t3=0,4$; $t4=0,2$ e $t5=0,3$ minutos, ou seja:

$$ts = \frac{0,9 + 1,2 + 0,4 + 0,2 + 0,3}{5} = 0,6 \text{ minutos}$$

Você pode facilmente confirmar que a variância e o desvio padrão são:

$$ds^2 = 0,148 \text{ minutos} \quad ds = 0,385 \text{ minutos}$$

A fórmula geral para obtenção da constante de variação do tempo de serviço é:

$$k = 1 + \left(\frac{ds}{ts} \right)^2$$

Observe que, se "ts" for constante, seu desvio padrão será nulo e $k = 1$, conforme adiantei no início do exemplo.

A fórmula geral do tempo de espera, então, fica:

$$tw = \frac{\rho \cdot ts}{2(1-\rho)} \left[1 + \left(\frac{ds}{ts} \right)^2 \right]$$

Esta fórmula foi desenvolvida por Pollaczek-Khintchine e é válida para qualquer distribuição probabilística do tempo de serviço.

Qual será, agora, o tempo médio de espera na fila do banco ?

Confira:

$$k = 1,41$$

$$tw = 228,4[s]$$

$$tr = 264,4[s]$$

Observe que, quanto maior for o desvio padrão do tempo de serviço, maior será o tempo médio de espera na fila !

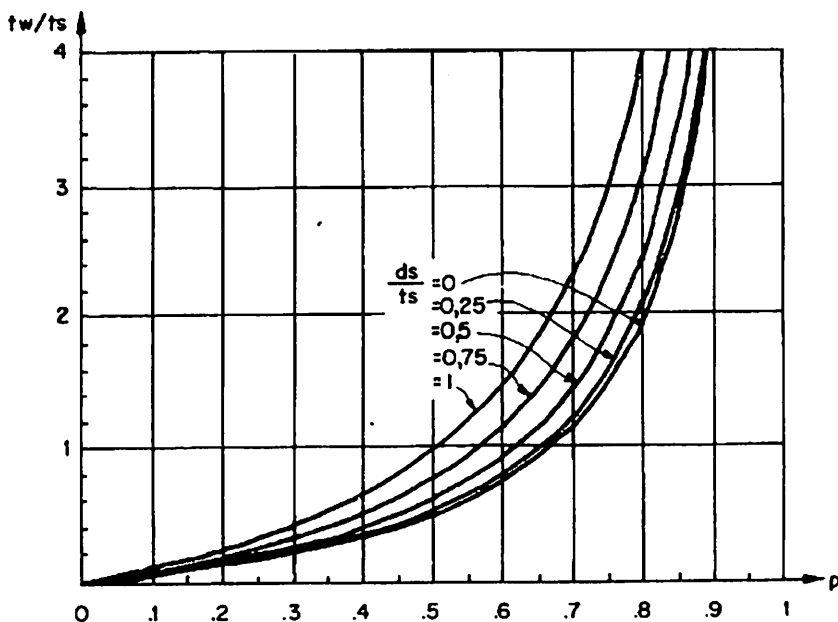


Fig.9.27: Tempo de espera em fila unisservidor

Outra informação importante é o comprimento médio da fila, dado pela equação de Little:

$$nw = N.L.tw$$

nw = nº de transações aguardando na fila
N.L = taxa total de chegada das transações
tw = tempo médio de espera

No exemplo do banco, a fila terá, em média, cerca de 4 pessoas se o desvio padrão for zero e 6 pessoas se o desvio padrão for 0,385 minutos.

Detalhes sobre as equações apresentadas:

Nas equações que apresentei, considere que a distribuição de "L" é do tipo exponencial.

A distribuição exponencial é dada por:

$$p(x) = \frac{e^{-x/ts}}{ts} ; ts = ds$$

$$P(x < X) = 1 - e^{-x/ts}$$

p(x) = função densidade de probabilidade do tempo de serviço.
ts = tempo médio de serviço.
ts = $\int_0^{\infty} x.p(x).dx$
e = 2,71828...
P(x < X) = probabilidade do tempo de serviço ser menor que um valor "X".

A distribuição exponencial, para o tempo de serviço, corresponde a k=2, pois:

$$k = 1 + \left(\frac{ds^2}{ts^2} \right) = 1 + 1 = 2$$

Utilizando a distribuição exponencial para o tempo de serviço (o que pode ser feito na prática quando $1,8 < k < 2,2$), podemos determinar facilmente a probabilidade do tempo de resposta (ou serviço) ser menor que um determinado valor:

$$P(x < X) = 1 - e^{-x/t} ; t = ts \text{ ou } tr$$

Podemos também determinar qual tempo de resposta satisfaz uma determinada probabilidade preestabelecida:

$$e^{-x/t} = 1 - P(x < X)$$

$$X = -t \cdot \{\log_e [1 - P(x < X)]\}$$

Confira:

$P(x < X)$	X
63%	1r
90%	2,3.tr
95%	3,0.tr
99%	4,6.tr

Por exemplo, considerando que a distribuição de “ts”, naquele caso do banco, é exponencial, qual é o número máximo de clientes que o banco pode receber para garantir que 90% deles ficará na fila no máximo 3 minutos ?

Confira: $2,3.tr - ts = 180$

então: $tr = 94[s]$

calculamos: $p = 1 - \frac{ts}{tr} = 0,61$

calculamos: $L = \frac{p}{ts} = 0,017 [1/s]$

Concluimos que o banco pode receber até 61 clientes por hora.

RESUMO DA TEORIA DAS FILAS UNISSERVIDOR:

$$\begin{aligned} p &= ts.N.L & n &= tr.N.L \\ tr &= tw + ts & nw &= tw.N.L \\ n &= nw + na & na &= p \end{aligned}$$

	Distribuição de “ts”		
	qualquer	constante	exponencial
tw	$\frac{p.ts.k}{2.(1-p)}$	$\frac{p.ts}{2.(1-p)}$	$\frac{p.ts}{1-p}$
tr	tw + ts	tw + ts	$\frac{ts}{1-p}$
n	$p + \frac{p^2.k}{2.(1-p)}$	$p + \frac{p^2}{2.(1-p)}$	$\frac{p}{1-p}$
nw	$\frac{p^2.k}{2.(1-p)}$	$\frac{p^2}{2.(1-p)}$	$\frac{p}{1-p}$

9.7.2.2 FILA MULTISSERVIDOR

Neste caso, vários servidores estarão atendendo a uma mesma fila, ou seja, “M” será maior que “1”.

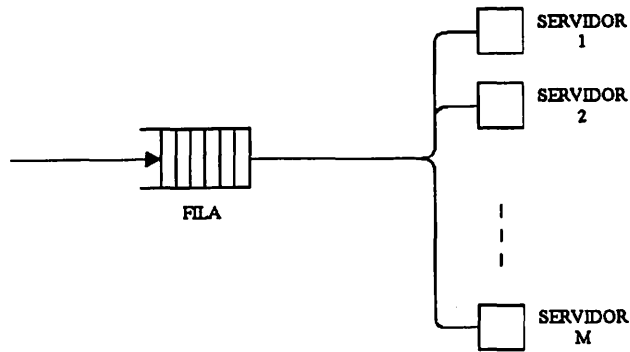


Fig.9.28: Fila multisservidor

A disciplina é tal que, logo que um servidor desocupa, a primeira transação da fila é chamada para atendimento (FIFO).

A fórmula para cálculo do tempo médio de espera é bem mais complexa que no caso da fila unisservidor e considera $k=2$, que corresponde a uma distribuição exponencial de “ts”.

$$tw = ts \cdot \frac{1-A}{M \cdot (1-p) \cdot (1-p \cdot A)}$$

$$A = \frac{\sum_{j=0}^{M-1} \frac{(M \cdot p)^j}{j!}}{\sum_{j=0}^M \frac{(M \cdot p)^j}{j!}}$$

Você pode consultar a figura 9.29 a fim de obter a relação tw/ts , desde que conheça “p” e “M”.

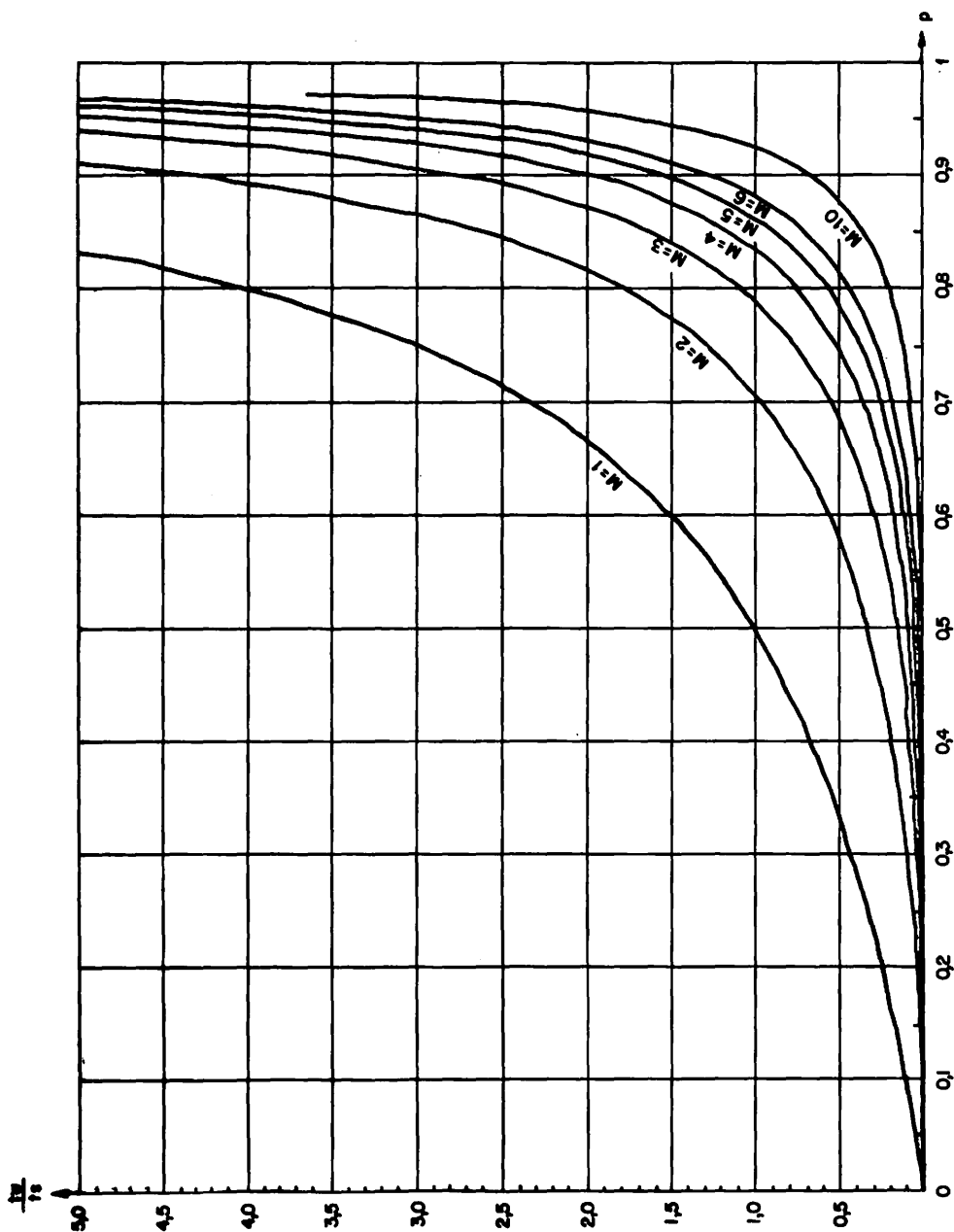


Fig.9.29: Tempo de espera em fila multisservidor

Verifique que, no caso de $M=1$, o tempo de espera é exatamente igual ao da fila unisservidor, considerando $k=2$.

Se você fizer $M=1$, encontrará:

$$A = \frac{1}{1+p} \quad e \quad tw = ts \cdot \frac{p}{1-p} ,$$

o que demonstra o fato de que a equação de “tw” para fila unisservidor é um caso particular da equação de “tw” para fila multisservidor, considerando $k=2$.

9.7.3 MULTIPONTO UNISSERVIDOR

Um circuito multiponto na modalidade “poll-select” pode ser visto como uma fila unisservidor, onde a CPU atende às transações dos “N” terminais.

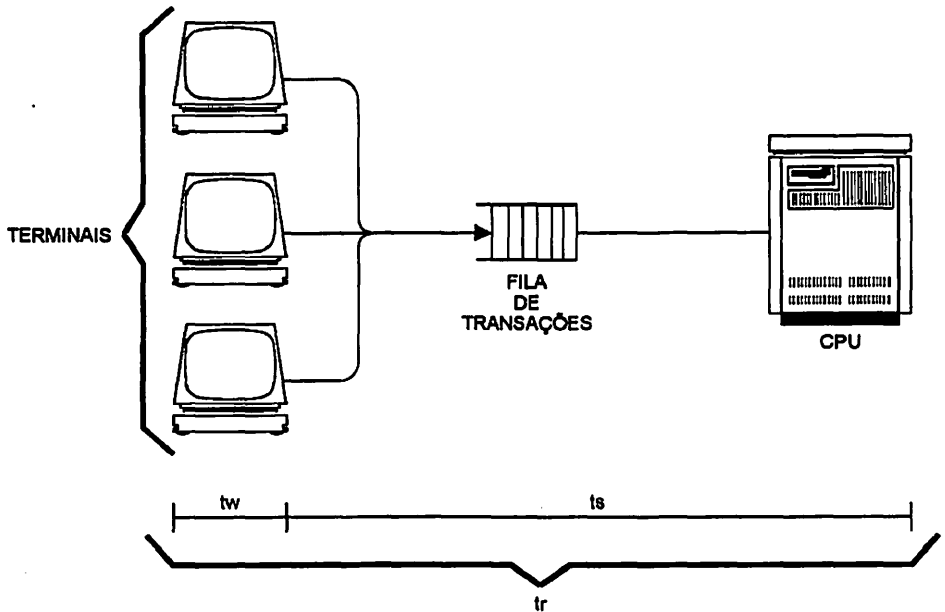


Fig.9.30: Multiponto unisservidor

O tempo médio de resposta, verificado pelos terminais é, como você já viu, a soma do tempo médio de espera na fila, com o tempo médio de serviço:

$$tr = \frac{p \cdot ts \cdot k}{2 \cdot (1-p)} + ts$$

O tempo médio de serviço pode ser expresso como:

$$ts = \frac{tg \cdot At}{1-R} + tc_{cpu}$$

onde:

- tg = (tgi + tgo) = tempo médio para transmissão das mensagens de entrada e saída.
- At = soma dos retardos envolvidos: tempos de propagação na linha e retardos RTS-CTS dos modems.
- tc_{cpu} = tempo de processamento na CPU
- R = taxa de retransmissão.

O tempo médio das mensagens pode ser expresso em função de seus comprimentos, em número de caracteres, e da velocidade do modem utilizado:

$$tg = tgi + tgo = \frac{b.(Mi + Ci + Mo + Co)}{vt} =$$

$$tg = \frac{b.(Mm + Cm)}{vt}$$

onde:

- b = bits por caractere
- vt = velocidade do modem [bps]
- Mi, Mo = nº médio de caracteres das mensagens de entrada e saída, respectivamente, excluindo os caracteres de controle
- Ci, Co = nº de caracteres de controle associados às mensagens de entrada e saída, respectivamente
- Mm, Cm = nº médio geral de caracteres de mensagem e controle, respectivamente.

O protocolo utilizado é responsável diretamente pelo parâmetro “Cm” e também influi em “At”, pois ele define quantas vezes os modems remotos devem passar do estado de recepção para o de transmissão.

Considerarei o protocolo BSC, com:

$$Cm = 48$$

$$At = 2.tci + 6.(tp + tm),$$

a fim de chegar a uma fórmula mais próxima de situações reais.

Então, para o protocolo BSC, o tempo de serviço será:

$$ts = \frac{1}{1-R} \left[\frac{b.(Mm + 48)}{vt} + 2.tci + 6.(tp + tm) \right] + tc_{pu}$$

Finalmente, para calcular o tempo médio de resposta, em sistemas multiponto que utilizem o protocolo BSC, basta calcularmos “ts”, “p” e “tr”. Algumas figuras, apresentadas a seguir, ilustram o assunto discutido e dão uma boa idéia do comportamento de sistemas multiponto. Nessas figuras, utilizei os seguintes parâmetros:

- k = 2
- tc_{pu} = 1 segundo
- R = 0,005 (0,5% de retransmissão)
- b = 9 bits por caractere
- 6.(tp + tm) = 0,075 (75 ms)

A figura 9.31 apresenta a variação do tempo de resposta, em função do número de terminais ligados à linha, para as velocidades de 2.400, 4.800, 9.600 e 19.200 bps. Note que a velocidade de 19.200 bps normalmente se refere a modems banda-base, o que significaria uma rede limitada em distância.

As figuras 9.32 e 9.33 mostram o número máximo de terminais que podem compartilhar uma linha multiponto, sem que se ultrapasse um determinado tempo médio de resposta.

Finalmente, a figura 9.34 ilustra a influência do retardo RTS-CTS dos modems no tempo de resposta do sistema.

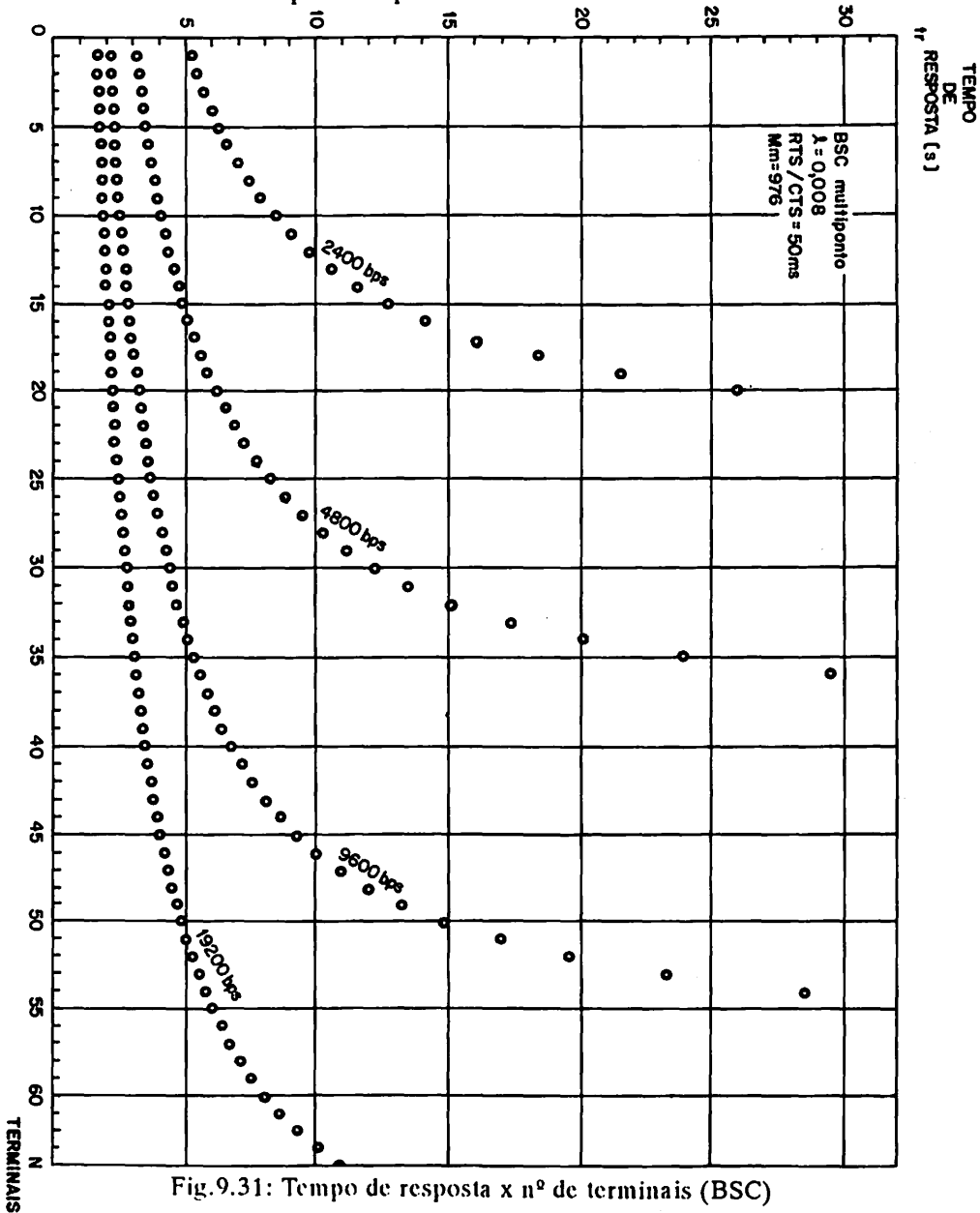


Fig.9.31: Tempo de resposta x nº de terminais (BSC)

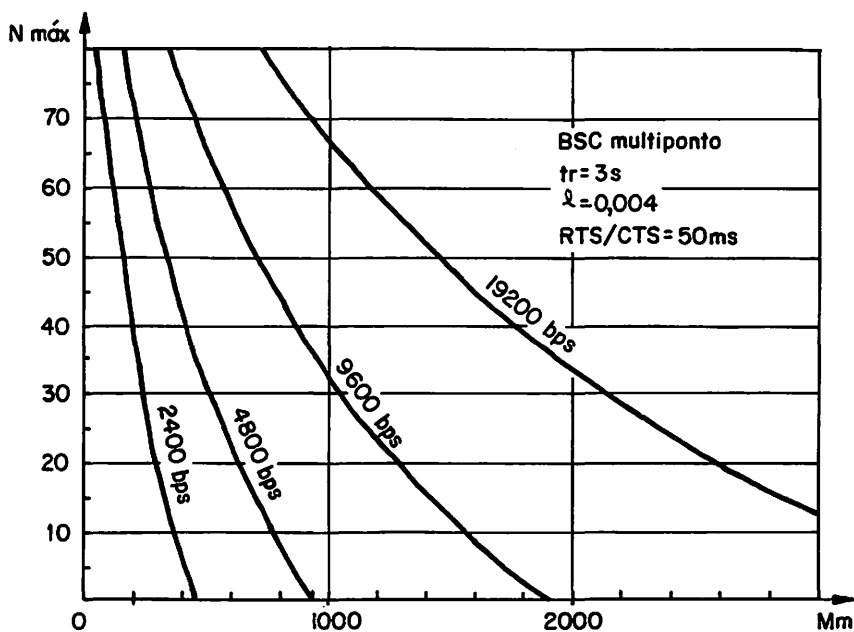


Fig.9.32: N° de terminais x M_m para $t_r=3s$ (BSC)

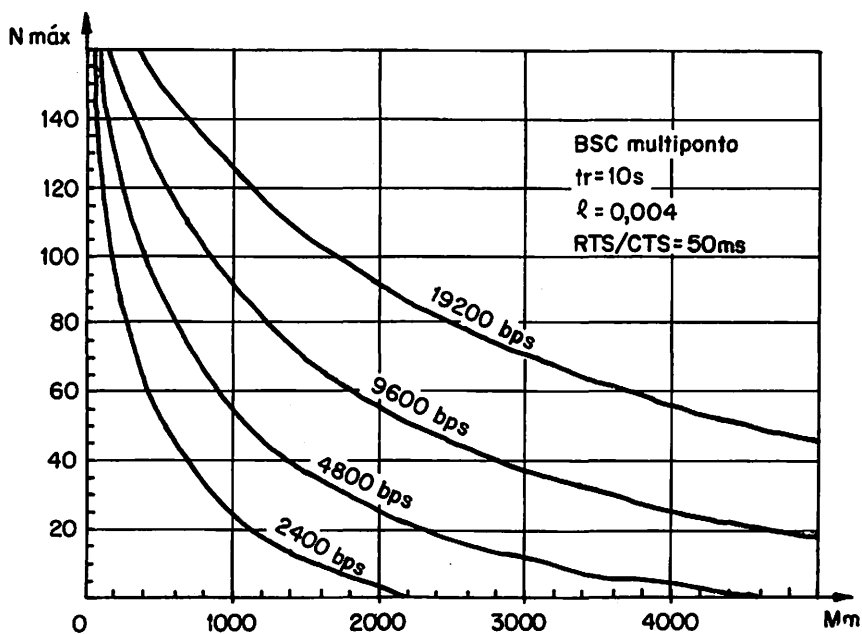


Fig.9.33: N° de terminais x M_m para $t_r=10s$ (BSC)

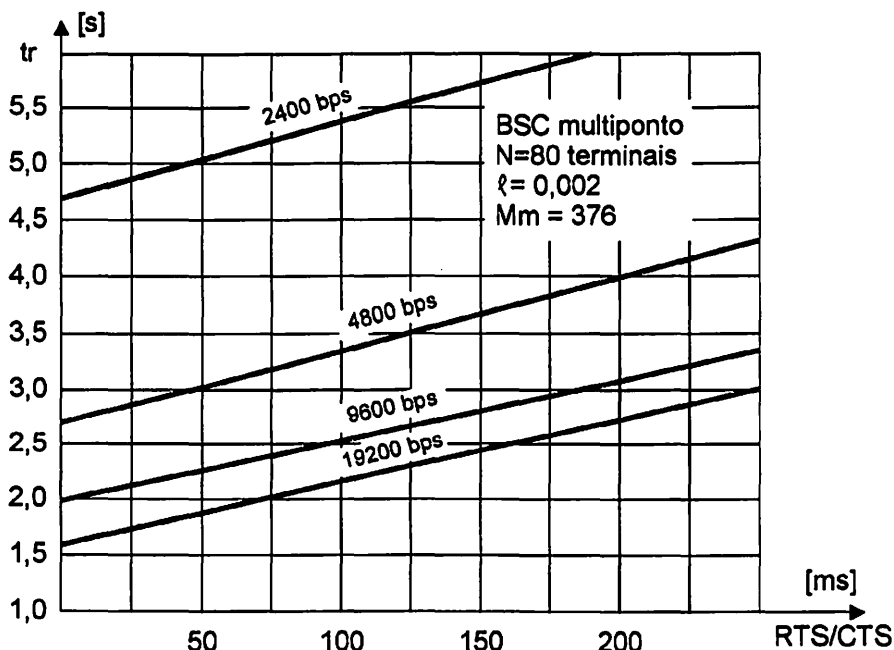


Fig.9.34: tempo de resposta x RTS-CTS (BSC)

9.7.4 EXEMPLO DE UM CIRCUITO MULTIPONTO

Suponha um sistema multiponto, utilizando o protocolo BSC com caracteres de 8 bits mais 1 de paridade, a 4 fios, sendo que a CPU oferece apenas uma porta servidora com um modem em portadora constante, e gasta 0,5 segundos para processar cada transação. O sistema possui 10 terminais remotos com modems 2.400 bps, operando com portadora chaveada e retardos RTS-CTS de 50 ms.

A estatística das transações é a seguinte:

10% com	100 caracteres
20% com	200 caracteres
40% com	600 caracteres
20% com	800 caracteres
10% com	1.000 caracteres

Cada terminal faz, em média, 10 consultas por hora.

Desejo saber qual é o tempo médio de resposta deste sistema.

COMPRIMENTO MÉDIO DAS MENSAGENS

$$Mm = \frac{\sum_{i=1}^m M_i}{m} = \sum M_i P_i$$

M_i = comprimento da i ésima mensagem
 P_i = probabilidade de ocorrer o comprimento M_i .

Então:

$$Mm = (0,1).100 + (0,2).200 + (0,4).600 + (0,2).800 + (0,1).1000$$

$$Mm = 550 \text{ caracteres}$$

DESVIO PADRÃO DO COMPRIMENTO

$$dm^2 = \frac{\sum_{i=1}^m (M_i - Mm)^2}{m} = \frac{\sum_{i=1}^m M_i^2}{m} - Mm^2$$

$$dm^2 = \sum M_i^2 \cdot P_i - Mm^2$$

$$dm^2 = 38,1 \cdot (10)^4 - 30,25 \cdot (10)^4 \quad dm = 280 \text{ caracteres}$$

VARIAÇÃO DO TEMPO DE SERVIÇO

$$k = 1 + \left[\frac{ds}{ts} \right]^2 = 1 + \left[\frac{dm}{Mm} \right]^2$$

A igualdade acima pode ser utilizada pois os retardos de modem e linha podem ser desprezados quando comparados aos tempos de transmissão das mensagens.

Então:

$$k = 1 + \left[\frac{280}{550} \right]^2 = 1,26$$

TEMPO MÉDIO DE SERVIÇO

Como a taxa de retransmissão e o retardo ($tp+tm$) não foram mencionados, considerarei:

$$R = 0$$

$$tp + tm = 0,02$$

Então:

$$ts = \frac{9 \cdot (550 + 48)}{2400} + 2 \cdot (0,05) + 6 \cdot (0,02) + 0,5 \quad ts = 2,9625 \text{ segundos}$$

UTILIZAÇÃO DA LINHA

$$p = \frac{ts.N.L}{M} = (2,9625).(10).\left(\frac{10}{3600}\right) \quad p = 0,082$$

TEMPO DE RESPOSTA

$$tr = \frac{p.ts.k}{2(1-p)} + ts = 0,167 + 2,9625 \quad tr = 3,13 \text{ segundos}$$

9.7.5 MULTISSERVIDOR COM PERDA DE CHAMADA

A teoria das filas se aplica a sistemas onde as transações ficam aguardando em uma fila, quando todos os servidores estão ocupados.

O modelo multisservidor com perda de chamada se aplica a sistemas onde as transações são recusadas, quando todos os servidores estão ocupados. Um exemplo prático é o PABX: se você tenta ligar para um local que possui PABX e recebe o sinal de ocupado, você deve desligar e tentar novamente, pois sua ligação foi simplesmente recusada por não ser permitida a fila.

A probabilidade de uma chamada receber o sinal de ocupado é:

$$P[\text{ocupado}] = \frac{\frac{(M.p)^M}{M}}{\sum_{j=0}^M \frac{(M.p)^j}{j!}}$$

Onde:

M = nº de servidores

$p = \frac{ts.L}{M}$ = utilização da linha ($N=1$)

L = taxa média de chegada das chamadas

A equação acima considera que a distribuição do tempo de serviço é exponencial.

A figura 9.35 mostra a probabilidade de uma chamada encontrar o sinal de ocupado (probabilidade de perda), em função da utilização da linha (" p ") e do número de servidores (" M ").

A partir destas considerações, você pode determinar a quantidade de portas de um sistema com atendimento via linha comutada, desde que seja fornecida a probabilidade de perda e a ocupação da linha.

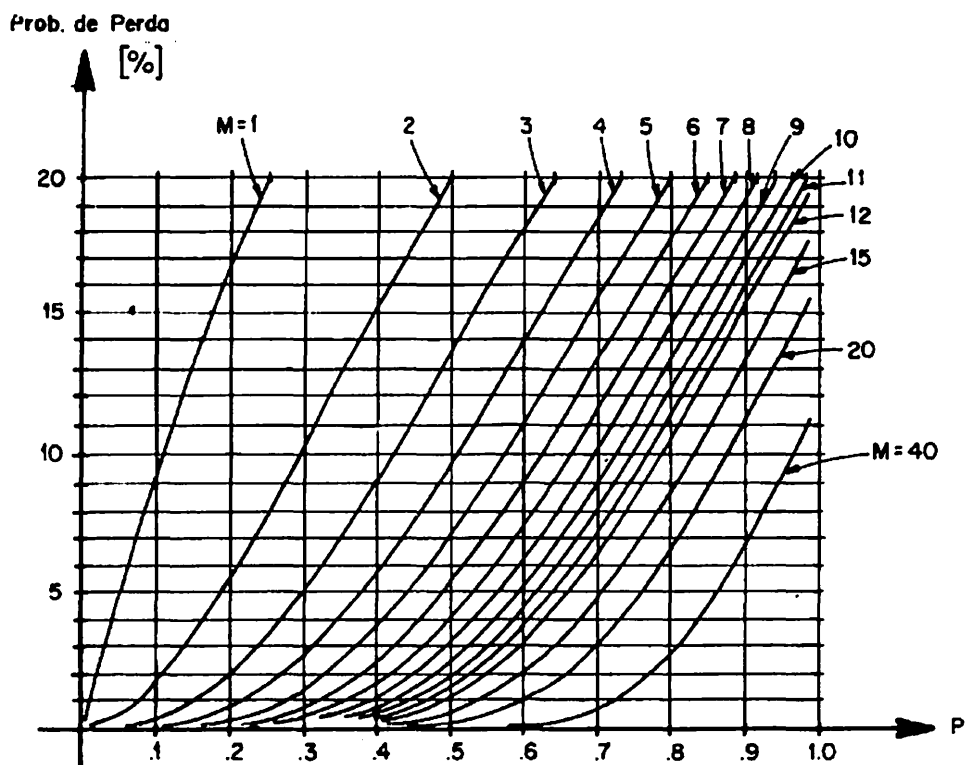


Fig.9.35: Probabilidade de perda de chamada

Por exemplo, suponha que um sistema (semelhante ao Videotexto) atenda a chamadas externas, utilizando modems com resposta automática através de um PABX, conforme ilustra a figura 9.36.

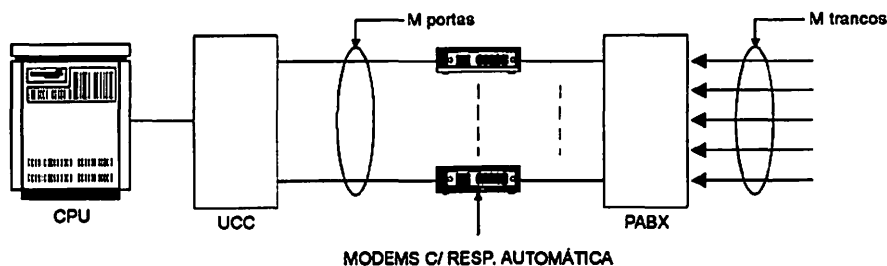


Fig.9.36: Consulta por linha comutada

Suponha, ainda, que o sistema tenha as seguintes características:

- a) Recebe, em média, uma consulta por minuto.
- b) Cada consulta dura, em média, 3 minutos.
- c) Possui 6 portas.

Qual é a probabilidade de um usuário receber o sinal de ocupado ao discar o número deste sistema?

Confira:

$$p = (3.1)/6 = 0,5$$

Consultando a figura 9.35 para $M= 6$ e $p= 0,5$, encontramos uma probabilidade de, aproximadamente, 5,3%

Observe que, se a quantidade de consultas dobrar, o sistema deverá ser expandido de 6 para 10 portas, se for desejável manter o nível de perda abaixo dos 5%.

9.8 ESTUDO DE CASOS

9.8.1 PONTO-A-PONTO A 2 FIOS COM MODEMS REPETIDORES

Quando uma linha física é muito extensa, ela pode impedir que se faça uma comunicação de dados utilizando modems banda-base, pois estes possuem alcances limitados.

Por exemplo: suponha que desejamos efetivar uma transmissão de dados, a 4.800 bps, por uma linha com bitola AWG 26 e extensão de 25 km.

Você viu no 5.2.1, que o modem banda-base S192 tem um alcance típico (50 ppm) de 15 km nessas condições (veja figura 5.65). Você pode ver que este alcance não é suficiente para efetivar a ligação de 25 km.

Nessas situações, podemos utilizar um par de modems repetidores, no meio do caminho, a fim de regenerar o sinal e assim conseguirmos um alcance maior.

Os modems repetidores devem estar predispostos com sincronismo de transmissão externo, pois os dados a serem transmitidos, provenientes do outro modem repetidor, já possuem seu próprio sincronismo (sincronismo de recepção).

Observe que, neste exemplo, estamos considerando uma ligação a 2 fios e isto exige um cuidado especial nos ajustes dos retardos RTS-CTS.

Veja, pela figura 9.37, que devemos sempre garantir que:

$$\min[tc1]_a > \max[tc2]_r + \max[DCD+]_r$$

Onde:

$\min[tc1]_a$	=	retardo RTS-CTS, mínimo, do modem situado na extremidade.
$\max[tc2]_r$	=	retardo RTS-CTS, máximo, do modem repetidor.
$\max[DCD+]_r$	=	retardo, máximo, entre a presença de sinal na linha e a ativação do DCD (pino 8 da interface RS232).

No caso do RHEDE S192, temos condição de seleccionar:

$$\begin{array}{lcl} \text{RTS-CTS} & = & \begin{array}{l} 8,5 \pm 1\text{ms} \\ 15,0 \pm 1\text{ms} \\ 23,5 \pm 1\text{ms} \\ 113,5 \pm 1\text{ms} \end{array} \\ \text{DCD+} & = & 6,0 \pm 2\text{ms (fixo)} \end{array}$$

Então, podemos escolher 23,5ms como retardo RTS-CTS dos modems situados nas extremidades e 8,5ms como retardo RTS-CTS dos modems repetidores, o que garante a condição estabelecida:

$$\begin{array}{l} 22,5 > 9,5 + 8 \\ 22,5 > 17,5 \end{array}$$

Outra condição que deve ser satisfeita para que o sistema opere é:

$$\min[tc1] > 2.\max[DCD+]$$

Observe que $22,5 > 2.(8)$ satisfaz esta condição.

No § 6.4, quando falei sobre a TRT no circuito ponto-a-ponto, defini "At" como sendo o tempo gasto entre a transmissão de dois blocos.

Se considerarmos que os tempos de reacção do terminal e da CPU, e os tempos de propagação(tp) são nulos, podemos dizer, observando a figura da próxima página, que:

$$At = 2.[2.(DCD-)+RCC] + [tc1+tc4] + 4.tm + t2$$

Então:

$$At = 2.[2.(15)+20] + 2.(23,5) + 4.(0,8) + t2$$

$$At = t2 + 150 \text{ [ms]}$$

Confira que, se não houvessem os modems repetidores, poderíamos reduzir At em 60ms e, ainda, se a ligação fosse a quatro fios, reduziríamos At em 147ms!

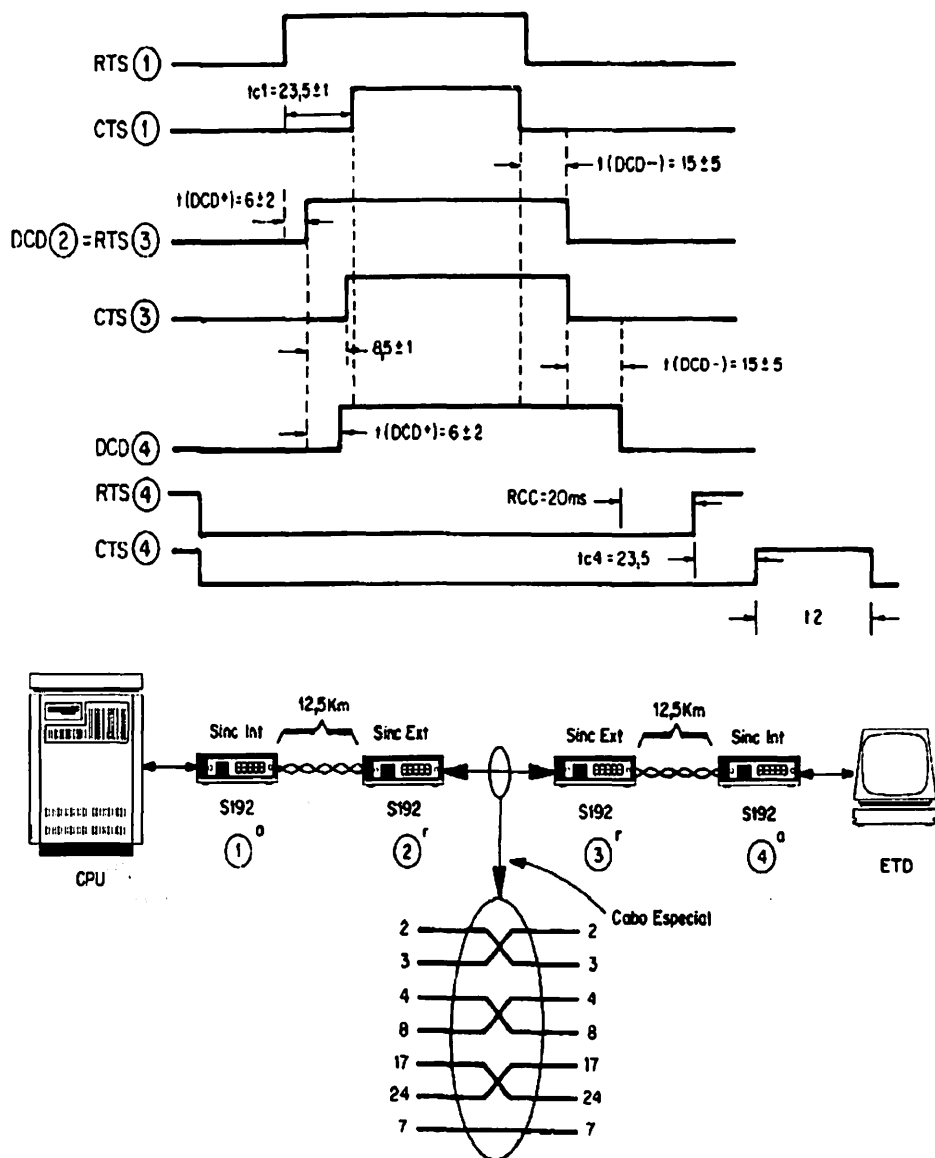


Fig.9.37: Ponto-a-ponto a 2 fios com modems repetidores

9.8.2 CIRCUITO FLORÊNCIO

Quando a concessionária instala uma linha privativa (LP) a 4 fios, existem os pares de transmissão e de recepção, que podem ser unidirecionais, caso exista algum equipamento de repetição ou PCM, por exemplo, no trajeto.

Suponha que você solicitou uma LP a 4 fios, entre duas localidades, e instalou um terminal operando a 1.200 bps assíncrono.

O circuito Florêncio consiste em ligar dois terminais nesta LP, utilizando modems tipo V22.

Os modems "1" e "4" devem operar no modo origem e os modems "2" e "3" no modo resposta.

Desta forma você pode instalar mais um terminal sem custos adicionais de linha telefônica, graças à característica do modem V22 em ocupar faixas de frequência distintas para transmissão e recepção.

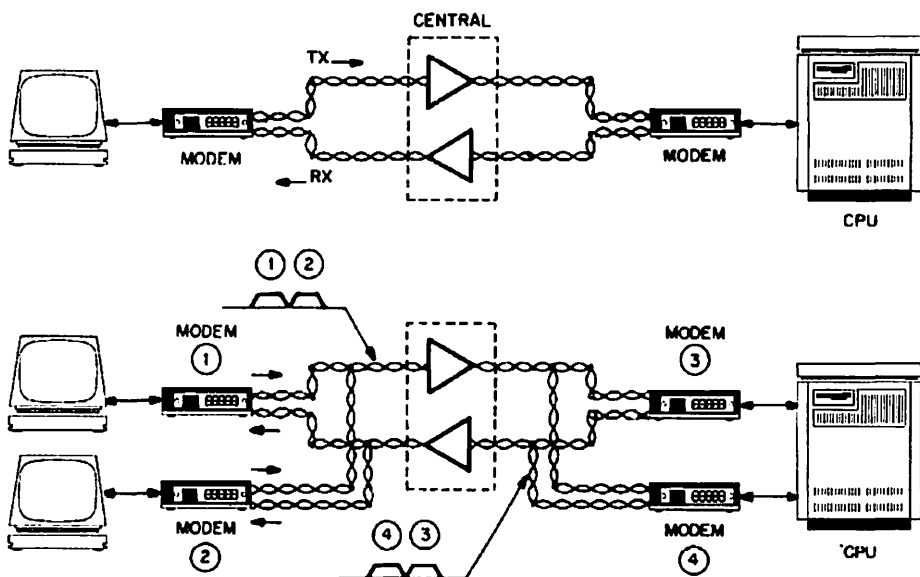


Fig.9.38: Circuito Florêncio

9.8.3 AUTOMAÇÃO BANCÁRIA

A figura 9.39 mostra, de forma simplificada, um sistema de automação bancária, atendendo duas cidades (Praças "1" e "2").

Este esquema se refere a um sistema "on-line", ou seja, as agências possuem terminais conectados ao computador central da praça, de forma que qualquer movimentação em conta-corrente, por exemplo, é uma transação onde a agência utiliza sempre os dados atualizados da central. Cada praça possui um computador central (CPU) com uma unidade de

controle de comunicação (UCC) de 512 portas.

Cada agência da praça possui um concentrador com capacidade para 16 (ou 32) terminais que podem ser: terminal de caixa (utilizado para a movimentação de boca de caixa), terminal de extrato (fica em local onde o próprio cliente pode consultar seu saldo) e terminal administrativo (utilizado internamente, inclusive para inicialização da agência, normalmente fica desvinculado do atendimento ao público).

Neste exemplo, todo o sistema "on-line" opera a 2.400 bps, síncrono, utiliza LP's a 4 fios e protocolo BSC.

O concentrador da agência reúne as transações dos terminais pelo processo de "polling" e se liga a uma porta da CPU através de um modem de 2.400 bps, que pode ser analógico ou banda-base, dependendo da distância e condições da linha telefônica. Naturalmente, dependendo do movimento da agência, ela pode ter mais que um concentrador.

A interconexão entre as duas praças é feita de CPU a CPU, normalmente a velocidades maiores. O exemplo da figura 9.39 utiliza 15 portas de 4.800 bps.

Os dois aspectos mais importantes de um sistema deste tipo são: a disponibilidade e o tempo de resposta.

A disponibilidade normalmente é assegurada pelo uso de circuitos e equipamentos reserva, e, o tempo de resposta, normalmente especificado em torno de 3 segundos, é sempre acompanhado por meio de estatísticas periodicamente efetuadas por um "software" de gerência".

Fig.9.39: Automação bancária: sistema "on-line"

