

## Nível Físico – Transmissão Banda Base

### 1 Objetivos

Entender a transmissão física em redes de computadores através da análise numa interface serial (abstraindo para diversos outros tipos de meios físicos e seus protocolos). Observar características da interface V.24/V.28 e buscar entender a importância de estabelecer padrões / protocolos.

### 2 Protocolos de transmissão física

Em termos físicos, o sinal é transmitido de um ponto ao outro, e necessita ter bem especificados seus parâmetros elétricos, mecânicos e funcionais, de forma que um bit seja transmitido e recebido corretamente.

Um protocolo de nível físico bastante simples é o NRZ (Non Return to Zero), onde existem dois níveis de tensão (um para “zero” e um para “um”). Ver exemplo na Fig 1. [http://pt.wikibooks.org/wiki/Introdu%C3%A7%C3%A3o\\_%C3%A0\\_comunica%C3%A7%C3%A3o\\_entre\\_computadores\\_e\\_tecnologias\\_de\\_rede/Como\\_representar\\_dados\\_em\\_sinais\\_digitais](http://pt.wikibooks.org/wiki/Introdu%C3%A7%C3%A3o_%C3%A0_comunica%C3%A7%C3%A3o_entre_computadores_e_tecnologias_de_rede/Como_representar_dados_em_sinais_digitais)

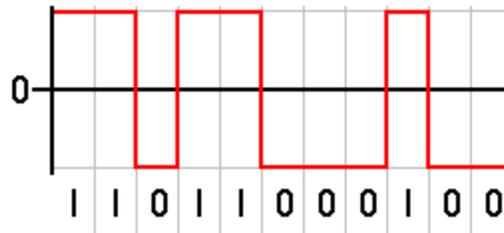
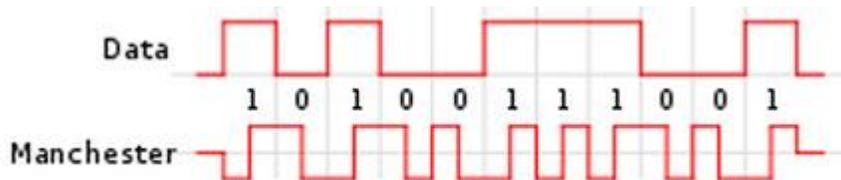


Fig. 1 – Codificação NRZ

O problema inerente a esse código é o dessincronismo de relógio entre transmissor e receptor para sequências grandes de zeros “000000000...” ou uns “111111111...”. Isso acontece pois ambos trabalham com relógio de quartzo, com imprecisão de algumas partes por milhão. Em altas taxas isso é bastante.

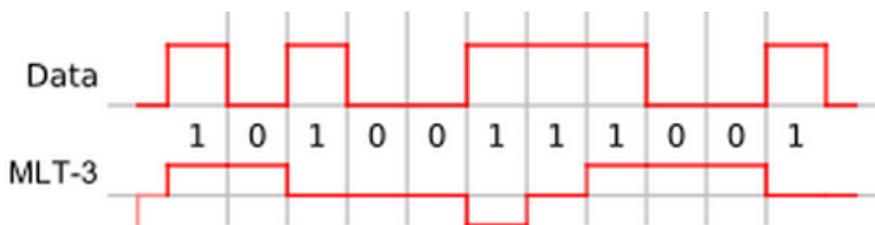
**Prática intermediária:** Calcule o tempo de bit na rede do laboratório.

Um código que resolve o problema sem necessitar clock externo é a codificação Manchester (Fig. 2) e seus derivados. Segue o exemplo. Note que a cada bit o receptor se resincroniza.



**Fig. 2 – Codificação Manchester**

Existem diversos outros exemplos, como AMI (*Alternate Mark Inversion*), HDB3 (*High Density Bipolar 3*). Numa rede Fast Ethernet com meio físico de par trançado, o sincronismo é feito através do código MLT-3 (*Multi-Level Transmit*) associado com o 4B-5B para eliminar sequências de zeros. Com fibra ótica já se usa em alguns casos o 8B-10B com NRZi.

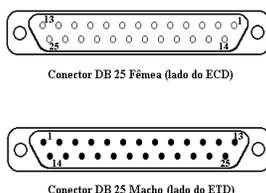


**Fig. 3 – Codificação MLT-3**

### 3 A Interface Serial V.24/V.28 do ITU-T ou RS 232 da EIA

A interface V.24/V.28 do ITU-T define *funcionalmente* (recomendação V.24) e *eletricamente* (recomendação V.28) os diversos sinais entre dois pontos. Apesar de possuir ao todo 25 pinos, cada qual com sua função específica, a comunicação é **serial**, pois existe apenas uma linha para transmissão de dados, e uma para recepção de dados.

As características físicas do conector também são objeto de especificação da recomendação V.28. Foi definido para a interface um conector de 25 pinos. No lado do ETD (Equipamento Terminal de Dados, com o computador) este conector deve ser do tipo macho, enquanto no lado do ECD (Equipamento de Comunicação de Dados, como o Modem) deverá ser do tipo fêmea conforme mostra a Figura 4. Atualmente também é aceita a implementação simplificada da interface RS 232 segundo um conector trapezoidal de 9 pinos.

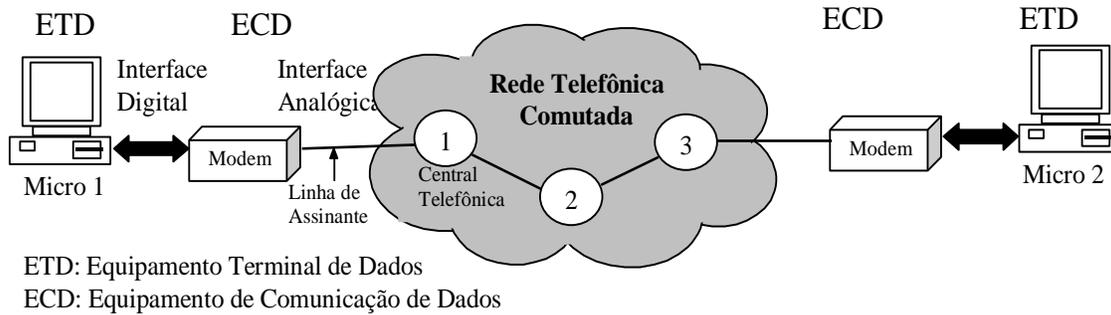


**Fig. 4 – Especificação do conector DB25**

Em termos elétricos, os NÍVEIS DE TENSÃO dessa interface são +3V a +25V para representar o "0" lógico, e -3V a -25V para representar o "1" lógico.

**Prática intermediária:** Desenhe a transmissão dos bits “10111000” utilizando codificação NRZ e os níveis de tensão do V.24 (adequados para o laboratório). Taxa de transmissão = 9600bit/s

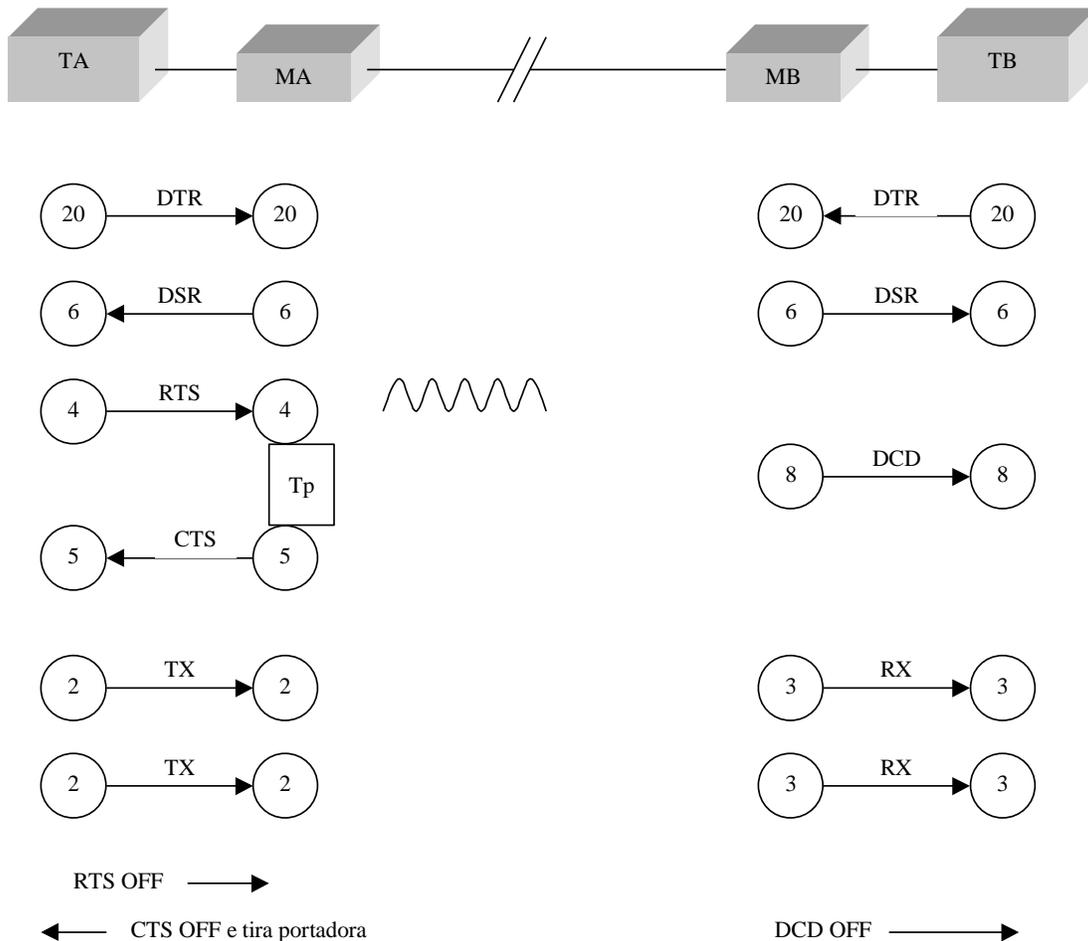
Na Figura 5 apresenta-se, como exemplo, a topologia de um sistema de comunicação para transmitir dados através da rede telefônica. A fonte de dados (Micro 1) e o receptor de dados (Micro 2) são designados genericamente como ETD (Equipamento Terminal de Dados). O Equipamento de Comunicação de Dados ou ECD (ex: modem) possibilita a conexão à rede telefônica. A interação entre ETD e ECD é feita através da interface V.24/V.28 do ITU-T.



**Fig. 5 - Ligação ponto-a-ponto entre dois computadores via rede telefônica**

### 3.1 Visão funcional da Interface

A Figura 6 mostra a sequência de sinalização na interface V.24 de forma funcional, de acordo com a função dos seus pinos.



**Fig. 6 – Visão funcional da interface V.24.**

### 3.2 Tipos de Sinais da Interface

A seguir serão analisados os principais pinos para dar uma noção mais genérica da interface:

- **TX:** pino 2 - é a linha utilizada para transmissão de dados;
- **RX:** pino 3 - é a linha utilizada para recepção dos dados;
- **GND:** pino 7 - é o terra lógico do conector. Todos os níveis de tensão utilizam como referência esta linha;
- **RTS:** pino 4 - Request To Send: pedido do DTE para o DCE para transmitir dados. Utilizado quando existe a necessidade de se efetuar uma transmissão;
- **CTS:** pino 5 - Clear To Send: permissão do DCE para o início da transmissão;
- **DSR:** pino 6 - Data Set Ready: indica que o DCE está pronto e energizado. Após ligar o equipamento, esta linha é ativada e assim permanece;
- **DTR:** pino 20 - Data Terminal Ready: indica que o DTE está pronto e energizado. Após ligar o equipamento, esta linha é ativada e assim permanece;
- **DCD:** pino 8 - Data Carrier Detect: indica que o DCE detectou portadora e deve começar a receber dados;
- **RI:** pino 22 - Ring Indicator: indica que está chegando uma ligação telefônica destinada àquele DCE, e se ele estiver em modo de resposta automática, poderá atender a ligação automaticamente após determinado número de rings.

Outra forma de ver os principais pinos (associados ao conector) é mostrada na figura a seguir, de [http://www.camiresearch.com/Data\\_Com\\_Basics/RS232\\_standard.html](http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html).



Fig. 7 – Pinos do DB25 e sinais associados.

No DB9, a pinagem é a seguinte:

<http://files.informatec-sp.com/pinagens.html#ydb9db9>.

#### Pinagem serial - RS-232C - conector DB9

Pino	Função	Abreviação
1	Data Carrier Detect	DCD
2	Receive Data	RX
3	Transmitted Data	TX
4	Data Terminal Ready	DTR
5	Signal Ground	GND
6	Data Set Ready	DSR
7	Request To Send	RTS
8	Clear To Send	CTS

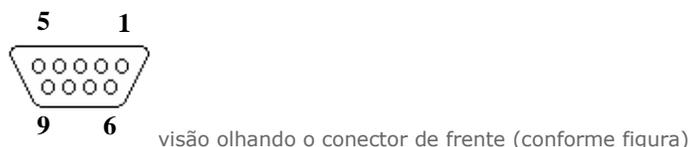


Fig. 8 – Pinagem e tipos de sinais do conector DB9.

### 3.3 Transmissão de Dados pela Interface V.24/V.28

Na transmissão assíncrona o terminal não fornece a cadência ou o sincronismo dos dados (relógio de transmissão). Neste caso, cada ETD utiliza o seu próprio relógio interno, regulado por técnicas de sincronismo de linha, como visto no início desta aula.

Em transmissão assíncrona V.24, quando queremos transmitir um bloco de dados binários, este é segmentado em conjuntos de oito bits de informação, que podem ser acompanhados ou não de um bit de paridade. A eles são acrescentados um cabeçalho, constituído de um bit de início (*start bit*), e um finalizador, constituído de um ou dois bits de parada (*stop bits*), formando um conjunto de dez a doze bits (como ilustrado na figura 9).

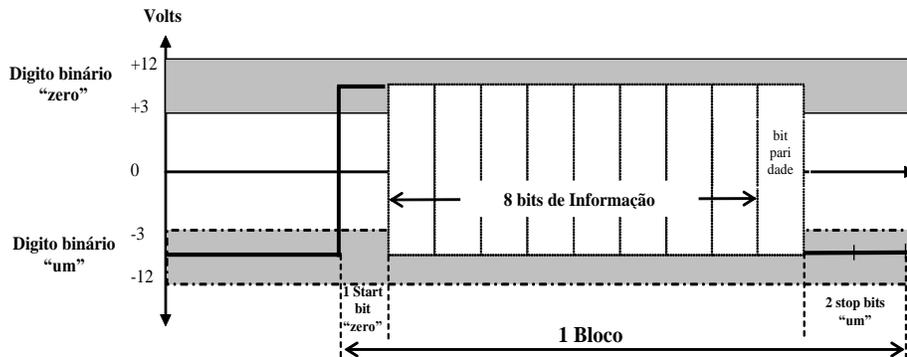


Fig. 9 - Transmissão assíncrona de caracteres ou bytes

O *start bit* ou cabeçalho é formado por um dígito binário zero, pois sinaliza o início do bloco de oito bits de informação. O finalizador, constituído de um a dois dígitos binários em *um* (também chamados de *stop bits*), indica o fim dos bits de informação. Os bits de *start* e *stop*, além de localizarem temporalmente os oito bits de informação, também têm a função de sincronizar a fase do relógio de recepção local no terminal de recepção.

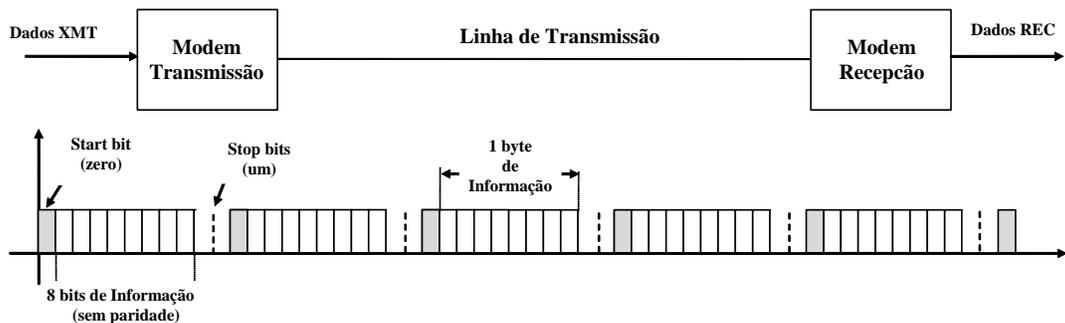


Fig. 10 - Transmissão assíncrona de um bloco de dados

Fica claro que a transmissão assíncrona, para enviar oito bits de informação, necessita um total de 11 bits, ou seja, uma eficiência  $\eta = 8/11 = 0,72$  (72%). Na transmissão de caracteres alfanuméricos, o nono bit às vezes ainda é utilizado para implementar um mecanismo de detecção de erros, baseado na paridade dos restantes oito bits de informação. Neste caso, a eficiência baixa para  $\eta = 8/12 = 0,66$  (66%).

## 4 Atividades

*OBS: sempre utilizar unidades que se utilizam na prática. Ex: utilize 3h20min em vez de 12.014,93 segundos.*

O arranjo experimental que será utilizado nesta experiência é apresentado na figura a seguir. O sistema de comunicações é constituído por dois computadores interconectados através de um cabo do tipo eliminador de modem ou cross-over, conectado nas portas de comunicação serial (COMn) dos computadores. Para um melhor acesso aos diversos circuitos da interface será utilizado um testador de dados (Data Test) configurado para a função de breakout box. Nesta função, o testador funciona unicamente como um monitor dos diversos sinais da interface V.24/V.28 que estão presentes no cabo cross-over.

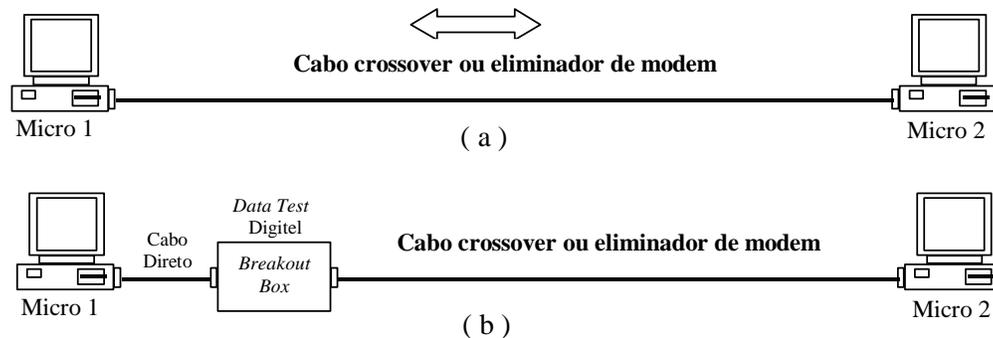


Fig 11. Interligação de dois computadores através de um cabo *cross-over* entre as portas seriais V.24/V.28 dos computadores: (a) sem *breakout box* (b) com *breakout box*

1. Utilizando o aplicativo Putty (disponível na máquina) ou Hyperterminal (download na página da disciplina), faça os seguintes itens (OBS: sugere-se utilizar a função de memória do osciloscópio para congelar o caractere recebido):
  - a. Obtenha a forma de onda **da primeira letra do seu nome** com as seguintes configurações: (9600 bit/s, 1 stop, 8 bits de dados, sem paridade).
  - b. Mostre a foto do osciloscópio. Justifique a forma de onda observada, bit a bit, comparando a parte de dados com a tabela ASCII (editando a foto do osciloscópio, preferencialmente, identificando start, stop e paridade).
  - c. Justifique a duração do caractere, comparando a taxa de transmissão configurada e a largura de cada bit no osciloscópio.
2. Repita os itens da questão acima, porém utilizando o caractere a **segunda letra do seu nome**, e a configuração 19200 bit/s, 2 stop bits, 7 bits de dados, paridade ímpar.
3. Utilize a **opção “transferir”+“enviar arquivo de texto”** ou “send”+“text file” no aplicativo HYPERTERMINAL (página da disciplina). Pode-se ainda utilizar o Putty. Usando o Putty, deve-se colocar os caracteres na memória (CTRL-C) e dar um clique da direita. Efetue a transferência de um arquivo de texto contendo caracteres **da sua escolha, com um mínimo de 4 caracteres**. Mostre a foto da forma de onda gerada, explicando onde está cada caractere e todos sinais de controle. Utilize 9600bit/s, 1 stop, 8 bits de dados, paridade par. **Sugestão:** editar a foto para facilitar a interpretação.  
SEQUÊNCIA DEFINIDA: \_\_\_\_\_
4. Se o transmissor enviar o caractere "u", HEXA 0x75, com a serial configurada para: mode com1: 96,n,6,1 (9600 bit/s, sem paridade, 6 bits de dados e 1 stop). Qual o caractere em hexa que chegará no receptor? Fotografe a onda. Explique.