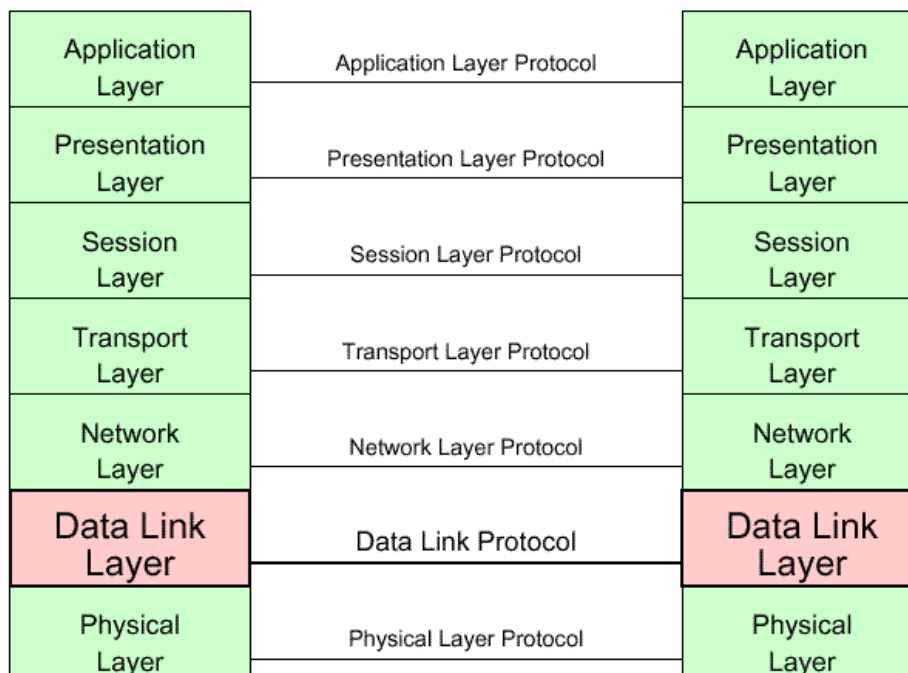

Controlo da Ligação de Dados

FEUP/DEEC/CDRC I – 2002/03

JAR/MPR

Camada de Ligação de Dados



Ligação de Dados - Conceito e Objectivos

- » Uma ligação física caracteriza-se por suportar uma transmissão não fiável de sinais que representam sequências de bits
- » O processo de transmissão não requer nem confere qualquer organização aos dados trocados na ligação, nem permite gerir e controlar a troca de dados
- » A comunicação de dados entre sistemas exige funções adicionais
 - estruturar e delimitar sequências de bits em unidades de comunicação (caracteres, blocos)
 - gerir e controlar a troca de dados (quando transmitir, identificar quem transmite / recebe)
 - detectar e eventualmente corrigir (recuperar) erros de transmissão
 - regular o fluxo de dados entre emissor e receptor
- » A execução destas funções (e possivelmente outras) requer um Protocolo de Ligação de Dados (ou de Ligação Lógica)
- » Uma Ligação de Dados consiste numa associação lógica entre entidades que utilizam uma ligação física não fiável para trocar dados de forma estruturada, fiável e controlada

Organização dos Dados - Caracteres

- ◆ A unidade de comunicação (estrutura) mais elementar presente ainda hoje em muitos sistemas de comunicação é o carácter
 - » Nesses casos são usados códigos (e.g., ASCII) que para além da representação de dados alfa-numéricos incluem caracteres que suportam funções associadas ao processo de comunicação e organização dos dados
- ◆ A forma mais elementar de comunicação consiste no envio de caracteres, de forma independente, competindo ao receptor o seu agrupamento para posterior interpretação
 - » Designa-se por **Transmissão Assíncrona** a técnica de transmissão correspondente

Organização dos Dados - Tramas

- ◆ Uma forma mais organizada e estruturada de comunicar consiste em agrupar caracteres em blocos (designados por tramas) e transmiti-los contiguamente de forma a constituírem uma unidade de comunicação
 - » Esta técnica designa-se por **Transmissão Síncrona** e os **Protocolos de Ligação de Dados** dizem-se neste caso **orientados ao carácter**

- ◆ Uma forma mais eficiente e flexível de comunicação organiza os dados em tramas constituídas por diversos campos com funções específicas, garantindo-se independência relativamente a códigos
 - » Baseia-se em **Transmissão Síncrona** e os **Protocolos de Ligação de Dados** dizem-se neste caso **orientados ao bit**

Funções Protocolares

Um protocolo de Ligação de Dados deve suportar as seguintes funções

- » Sincronismo de trama
 - Delimitação de início e fim de uma trama
- » Gestão da ligação
 - Estabelecimento, manutenção e terminação da ligação de dados, e controlo da comunicação
- » Endereçamento
 - Identificação de entidades físicas e lógicas que partilham a ligação física, permitindo em particular a multiplexagem de dados e informação de controlo
- » Controlo de erros
 - Detecção e eventualmente recuperação de erros (tipicamente por retransmissão)
- » Controlo de fluxo
 - Regulação do débito de dados do emissor
- » Transparência
 - Transferência de dados independente de códigos
- » Recuperação de erros de protocolo
 - Reposição do contexto da comunicação, que pode ser perdido devido a erros de protocolo

Transmissão Síncrona e Assíncrona

- ◆ Quer em Transmissão Assíncrona quer Síncrona um objectivo básico consiste em garantir sincronismo ao nível do bit, entre emissor e receptor, permitindo assim ao receptor recuperar a sequência de bits transmitidos
- ◆ O receptor deve utilizar um sinal de relógio “sincronizado” em frequência e fase com os dados transmitidos; o grau de precisão do relógio depende da técnica de transmissão utilizada, sendo crítico em Transmissão Síncrona
- ◆ É igualmente requerida sincronização a nível de carácter (nos casos em que os dados estejam organizados em caracteres) e a nível de trama (no caso da Transmissão Síncrona)

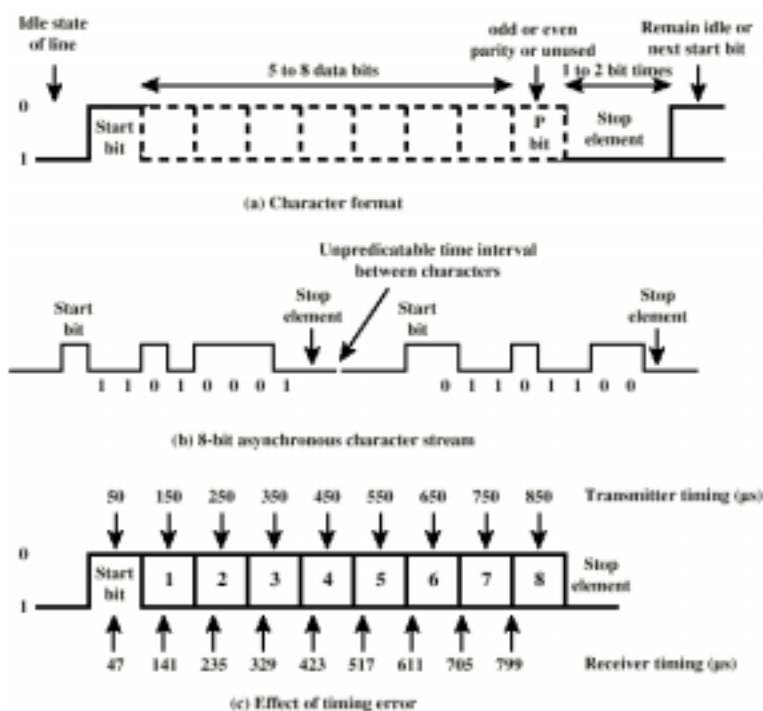
Transmissão Assíncrona

- ◆ A designação Transmissão Assíncrona traduz dois factos relacionados
 - » a ocorrência irregular (assíncrona) dos caracteres enviados de forma independente
 - » a inexistência de uma referência temporal comum ao emissor e ao receptor, isto é, o relógio do receptor é independente do do emissor (embora deva ter a mesma frequência nominal)
- ◆ O sincronismo de bit realiza-se com base no sincronismo de carácter
 - » Caracteres são enviados um a um com intervalos variáveis entre si
 - » Na ausência de transmissão (*idle*) o nível de sinal mantém-se constante (1 lógico)
 - » Cada carácter é precedido de um bit (*Start bit*) com o valor lógico 0
 - » A transição 1→0 produzida pelo *Start bit* indica o início do carácter (sincronismo de carácter) e fornece a referência de fase para o relógio do receptor
 - » Os bits subsequentes são recuperados por amostragem a partir do *Start bit*
 - » O carácter é seguido de *Stop bits* (valor lógico 1) para permitir distinguir o próximo *Start bit*, mesmo que não haja intervalo *idle* entre caracteres

Transmissão Assíncrona

- ◆ O método não requer elevada precisão na frequência de relógio do receptor
 - » Basta garantir o sincronismo de bit durante um carácter, pois o processo de sincronização recomeça com o início do carácter seguinte
 - » Para garantir o sincronismo de bit durante um carácter é necessário que o relógio do receptor esteja sincronizado em fase com o *Start* bit, o que não é fácil de garantir dado o facto de o relógio ser gerado independentemente dos dados
 - » O problema da sincronização de fase pode ser minimizado usando um relógio com uma frequência múltipla da frequência nominal (e.g., 8 ou 16 vezes)
- ◆ Caracterização
 - » Simples e económico
 - » Ineficiente
 - *Start* e *Stop* bits introduzem um *overhead* de 2 a 3 bits por carácter (5 a 8 bits de informação)
 - Transmissão *idle* entre caracteres impede a partilha da linha por vários terminais
 - » A precisão do método não é adequada para transmissão de blocos de dados (tramas) com número elevado de bits
 - » O método é apenas aceitável para transmissão a baixa velocidade e a pequena distância (devido à reduzida imunidade ao ruído e à elevada susceptibilidade a erros de amostragem)

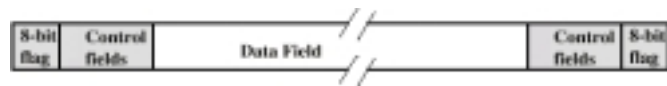
Transmissão Assíncrona



Transmissão Síncrona

- » A Transmissão Síncrona exige uma elevada precisão (em frequência e fase) do relógio do receptor relativamente aos dados, pois é utilizada em transmissão de unidades de dados de comprimento variável (tramas) que podem ser constituídas por um número muito elevado de bits
 - O relógio usado pelo receptor pode ser enviado numa linha separada, o que para além de consumir recursos está sujeito aos mesmos problemas do sinal (ruído, distorção de amplitude e de fase); justifica-se apenas para transmissão a pequenas distâncias
 - Mais usualmente o relógio do receptor é extraído do próprio sinal codificado ou a partir de uma portadora enviada com o sinal
- » Para se realizar sincronismo de trama são usadas sequências de bits com padrões conhecidos para delimitação de início e fim de trama; são adoptados métodos diferentes de organizar e delimitar tramas em protocolos orientados ao carácter e ao bit

- Exemplo: delimitação com *flags* em protocolos orientados ao bit



- » A Transmissão Síncrona é mais eficiente e flexível que a Transmissão Assíncrona podendo ser usada a muito alta velocidade e a grande distância

Mecanismos de Controlo

- ◆ Os mecanismos de Controlo de Erros e de Fluxo suportados em Protocolos de Ligação de Dados são normalmente usados em conjunto com um mecanismo de Confirmação de Tramas
- ◆ Estes três mecanismos baseiam-se na utilização de tramas de controlo (ou de supervisão) distintas das tramas de dados (ou de informação)
- ◆ Dada a relação entre estes mecanismos, as tramas de controlo podem realizar simultaneamente várias funções

Controlo de Fluxo

- ◆ Em protocolos de Ligação de Dados o Controlo de Fluxo é realizado por meio de tramas de controlo que:
 - » concedem ao emissor um crédito que pode ser explícito (valor especificado na trama) ou implícito (valor fixo conhecido), isto é, permitem-lhe transmitir um determinado número de tramas de dados (relativamente a uma trama de referência) sem necessidade de novas autorizações; o mais usual é o crédito ser implícito
 - » podem solicitar que o emissor suspenda temporariamente o envio de tramas de dados; normalmente usado como complemento do crédito implícito, visto que com crédito explícito pode ser indicado um crédito nulo, se necessário

Controlo de Erros

- ◆ Os mecanismos de Controlo de Erros têm por objectivo garantir a integridade da comunicação, isto é, assegurar a retransmissão de tramas de dados não correctamente recebidas
 - » As tramas são protegidas por um código detector de erros
 - » A estratégia básica consiste em realizar confirmações positivas de tramas de dados correctamente recebidas, por meio de tramas de controlo, genericamente designadas por ACK (*Acknowledgement*)
 - » A omissão de ACK após um intervalo de tempo pré-definido (*time-out*) é interpretada pelo emissor como indício de que uma trama de dados ainda não confirmada pode não ter sido correctamente recebida e que portanto pode ter de ser retransmitida
 - » É possível ainda usar tramas que realizam confirmação negativa (NAK), sinalizando explicitamente a necessidade de retransmissão
 - » O emissor mantém cópia de cada trama de dados transmitida, para eventual retransmissão; a recepção de confirmação permite-lhe descartar as tramas sinalizadas pelo receptor como correctamente recebidas

Necessidade de Numeração de Tramas

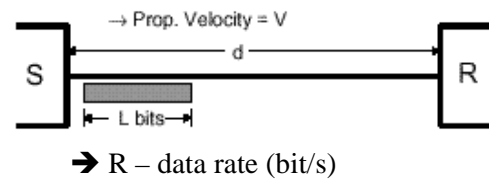
- ◆ No caso ideal de não ocorrerem erros nem ser necessário controlar o fluxo do emissor, este poderia enviar tramas a qualquer momento sem quaisquer restrições e sem necessidade de as identificar
- ◆ O mecanismo de Controlo de Erros exige que sejam identificadas as tramas de dados correctamente recebidas (e que assim são confirmadas) e as não recebidas ou recebidas com erros (e que portanto requerem retransmissão)
- ◆ O mecanismo de Controlo de Fluxo exige também um processo de referenciar tramas de dados para efeito da concessão de crédito
- ◆ Os protocolos de Ligação de Dados devem providenciar um mecanismo de identificação (numeração) de tramas de dados
 - » o número de sequência $N(s)$ de uma trama de dados deve ser transportado no respectivo cabeçalho
 - » tramas de controlo devem referenciar tramas de dados transmitidas em sentido oposto, por meio de um número de sequência $N(r)$ no seu cabeçalho

Método de Numeração de Tramas

- ◆ Com o objectivo de reduzir o *overhead*, procura-se limitar o número de bits usados para numerar as tramas de dados
 - » usando k bits para o efeito é possível dispor de $M = 2^k$ identificadores diferentes (0, 1, ..., $M-1$) - numeração módulo M
- ◆ Os identificadores têm de ser reutilizados, visto que sendo em número finito se repetem ciclicamente
 - » Não é possível que tramas de dados diferentes, transmitidas e ainda não confirmadas (pendentes), tenham o mesmo número de sequência
 - » O número máximo de tramas de dados pendentes depende do módulo de numeração e da estratégia de retransmissão
- ◆ A existência de um limite superior ao número de tramas de dados pendentes constitui uma restrição ao envio de tramas e portanto uma forma indirecta de Controlo de Fluxo

Modelo e Parâmetros de Comunicação

- ◆ As tramas são recebidas na mesma ordem com que são enviadas
 - » tramas de dados recebidas fora de sequência indicam ocorrência de perda(s)



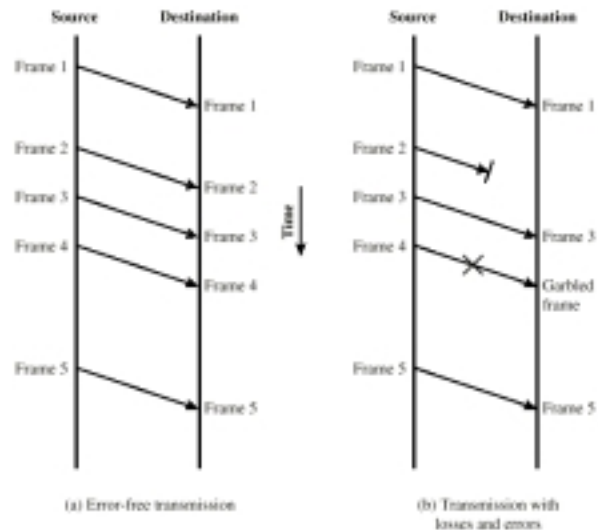
- ◆ Tempo de transmissão
 - » Tempo de envio de todos os bits da trama para o meio

$$T_f = L/R$$

- ◆ Tempo de propagação
 - » Tempo que um bit demora a propagar-se no meio entre o emissor e o receptor

$$T_p = d/V$$

- ◆ Técnicas a analisar
 - » *Stop and Wait*
 - » Janela deslizante (*Sliding Window*)



Stop and Wait

- ◆ Começemos por analisar um caso simples em que o emissor apenas pode transmitir uma nova trama de dados após ter recebido confirmação positiva (ACK) de recepção correcta da trama de dados anteriormente transmitida
- ◆ Este mecanismo designa-se por *Stop and Wait*
 - » Aparentemente não seria necessário numerar as tramas de dados (pois no máximo apenas uma poderá estar pendente)
 - » A numeração de tramas de dados é necessária devido à ocorrência de erros; em certos casos a retransmissão de tramas de dados pode dar origem a duplicados que só podem ser detectados e eliminados se as tramas forem numeradas (neste caso bastaria numeração módulo 2 - isto é, os números de sequência usados são 0 e 1)
 - » O *Stop and Wait* é em primeiro lugar um mecanismo muito elementar de confirmação de tramas de dados, mas que pode impor sérias restrições à transmissão continuada de tramas

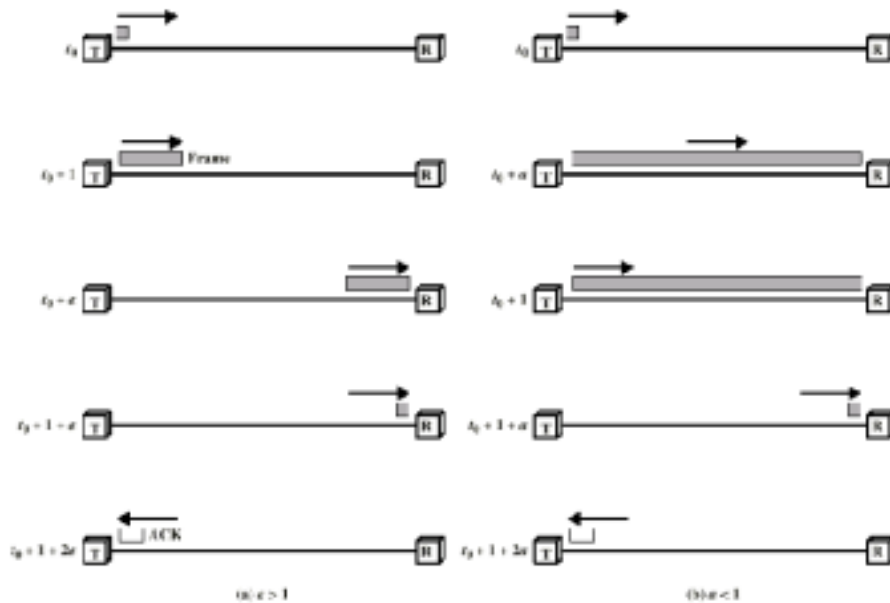
Stop and Wait e Controlo de Fluxo

- ◆ O *Stop and Wait* constitui um mecanismo de Controlo de Fluxo, pois ACK apenas autoriza o emissor a transmitir uma nova trama de dados (concede um crédito implícito de uma unidade)
 - » o emissor suspende a transmissão até receber ACK
 - » ACK concede crédito para transmitir uma nova trama de dados
- ◆ O processo de comunicação introduz um atraso inevitável entre o envio de uma trama de dados e a recepção do ACK respectivo; o receptor pode introduzir deliberadamente um atraso adicional, retardando (ou até cancelando) o envio de ACK
 - » no primeiro caso o Controlo de Fluxo é indirecto e certamente indesejável
 - » no segundo caso o Controlo de Fluxo é intencional, mas um atraso excessivo no envio de ACK pode ser confundido com ausência de confirmação (*time-out*) e despoletar uma retransmissão desnecessária, pelo que este mecanismo deve ser usado com prudência

Stop and Wait

- ◆ Funcionamento básico sem erros
 - » O Emissor transmite uma trama de dados e fica à espera de receber ACK antes de poder enviar a trama de dados seguinte
 - » O Receptor recebe a trama de dados e responde com confirmação positiva (ACK)
 - » O Receptor pode realizar controlo de fluxo não enviando ou atrasando ACK
- ◆ A eficiência pode ser aceitável com tramas grandes e transmissão a baixa velocidade e a pequena distância
- ◆ O sistema de comunicação pode requerer fragmentação em tramas pequenas
 - » Vantagens
 - Aconselhável se existirem limitações de *buffers* no receptor
 - Os erros são detectados mais rapidamente
 - Se ocorrerem erros a quantidade de dados a retransmitir é menor
 - Evita que uma estação ocupe o meio por longos períodos (pode ser importante se o meio for partilhado por várias ligações)
 - » A eficiência do *Stop and Wait* pode diminuir drasticamente se o tamanho das tramas for muito pequeno, o que é agravado com transmissão a alta velocidade e a grande distância

Stop and Wait

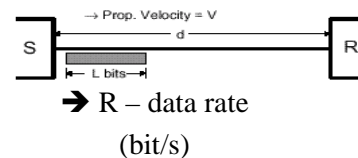


Stop-and-Wait Link Utilization (transmission time = 1; propagation time = a)

Utilização (Eficiência) do Stop and Wait

» WAN ATM

- $d = 1000$ km
- $L = 424$ bit, $R = 155.52$ Mbit/s
- $T_f = 2.7$ μ s
- Fibra óptica $\rightarrow 5$ μ s/km $\rightarrow T_p = 5$ ms
- $a = 1852$
- $U = 1 / 3705 = 0.0003$

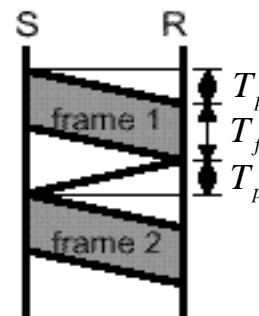


$$T_p = \frac{d}{V} \quad T_f = \frac{L}{R}$$

$$a = \frac{T_p}{T_f} = \frac{d}{V} \times \frac{R}{L}$$

» LAN

- $d = 0.1 \sim 10$ km
- $L = 1000$ bit, $R = 10$ Mbit/s
- $T_f = 100$ μ s
- Cabo coaxial $\rightarrow 4$ μ s/km $\rightarrow T_p = 0.4 \sim 40$ μ s
- $a = 0.004 \sim 0.4$
- $U = 0.55 \sim 0.99$ (e se $R = 100$ Mbit/s?)



» Modem sobre linha telefónica

- $d = 1000$ m
- $L = 1000$ bit, $R = 28.8$ kbit/s
- $T_f = 34.7$ ms
- UTP $\rightarrow 5$ μ s/km $\rightarrow T_p = 5$ μ s
- $a = 144 \times 10^{-6}$
- $U = 1.0$

$$U = \frac{T_f}{T_p + T_f + T_p} = \frac{1}{1 + 2a}$$

Sliding Window

- ◆ Podemos generalizar o mecanismo de confirmação e permitir que o emissor transmita várias tramas de dados antes de receber qualquer confirmação (ou autorização). Estes protocolos designam-se por Janela Deslizante (*Sliding Window*)
 - » As tramas de dados têm de ser numeradas para ser possível ao mecanismo de confirmação referenciar tramas correcta e incorrectamente recebidas
 - » São possíveis diferentes estratégias de retransmissão no caso de erros
 - » O número máximo de tramas de dados que o emissor pode ter pendentes designa-se por Janela de Transmissão
 - » O tamanho máximo possível da Janela de Transmissão está relacionado com o módulo de numeração de tramas e com a estratégia de retransmissão
 - » O mecanismo *Stop and Wait* pode ser visto como um caso particular de *Sliding Window* com janela unitária

Sliding Window e Controlo de Fluxo

- ◆ O *Sliding Window* constitui uma forma de Controlo de Fluxo por limitar o número de tramas que o emissor pode ter pendentes
- ◆ Permite ultrapassar as limitações de desempenho do *Stop and Wait* (devidas ao controlo de fluxo implícito). Na ausência de erros o objectivo será atingir uma utilização de 100% (caso o tráfego de dados seja persistente); nesta situação o emissor não chega a interromper a transmissão por falta de créditos (janela esgotada), não havendo portanto de facto Controlo de Fluxo
- ◆ Normalmente não existem nos protocolos de Ligação de Dados mecanismos de indicação explícita de crédito variável; a realização de Controlo de Fluxo pode então requerer o envio de uma trama de controlo para suspender a transmissão e mais tarde de outra para reactivar a transmissão e o mecanismo de janela; este método designa-se por *Stop and Go*

Janela Deslizante (Sliding Window)

◆ Funcionamento básico sem erros

- » O Emissor pode ter múltiplas tramas de dados pendentes (transmitidas e não confirmadas)
- » O Receptor dispõe de *buffers* para W tramas de dados (tamanho da janela)
- » O Emissor pode enviar até W tramas de dados sem receber ACK
- » As tramas de dados são numeradas sequencialmente (módulo $M = 2^k$), sendo k o número de bits usados para codificar os M números de sequência
- » Por convenção ACK referencia por meio de $N(r)$ o número de sequência $N(s)$ da próxima trama de dados esperada na sequência correcta

Janela Deslizante (Sliding Window)

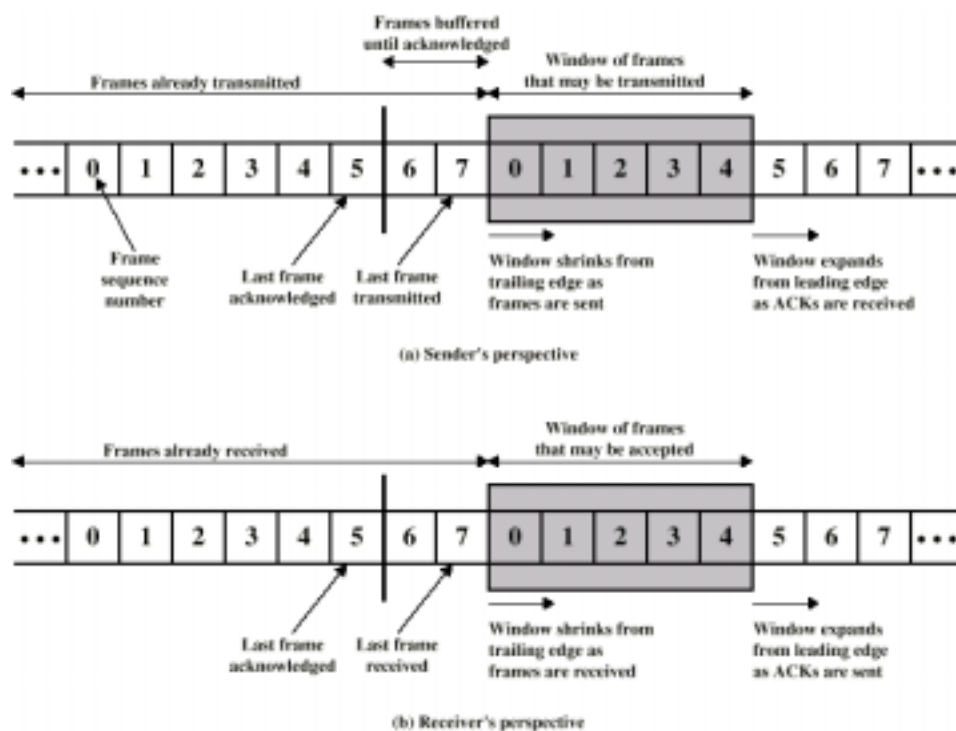
◆ Extensões ao funcionamento básico

- » Utilização de *piggybacking* com fluxo de dados bidireccional
 - Se o receptor também tiver dados para transmitir, pode evitar o envio de ACK realizando a confirmação nas tramas de dados em sentido oposto
 - Se o receptor não tiver dados para transmitir, usa ACK para realizar a confirmação
 - Se o receptor tiver dados para transmitir mas não tiver dados para confirmar, referencia o número da próxima trama de dados que espera receber

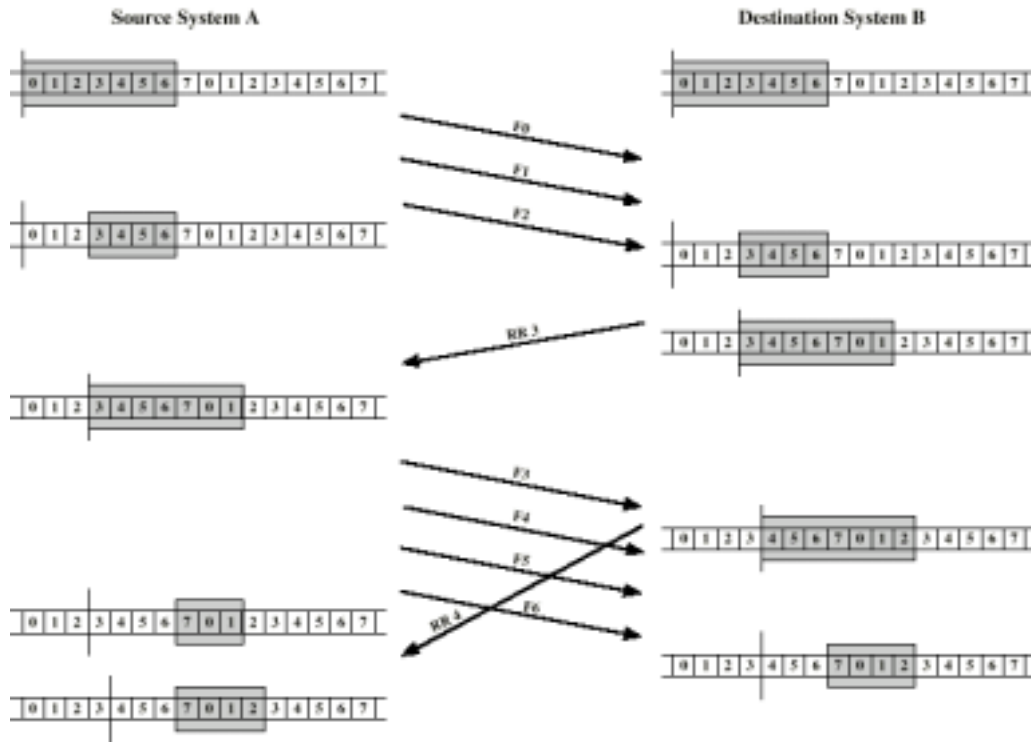
Controlo de Fluxo Stop and Go

- ◆ O receptor pode pretender confirmar tramas de dados e proibir novas transmissões - *Stop and Go*
 - » É necessário uma trama de controlo para suspender temporariamente a transmissão (designada habitualmente por *Receiver Not Ready* - RNR)
 - » A reactivação da transmissão será feita recorrendo a uma trama de controlo, designada normalmente por *Receiver Ready* - RR, caso o receptor não tenha entretanto descartado qualquer trama de dados; no caso de descarte, procede como se tivesse ocorrido perda de trama(s) de dados
 - » RR e RNR realizam igualmente a função de confirmação (ACK); aliás, em condições normais, RR é a trama de controlo usada para realizar confirmações

Janela Deslizante (Sliding Window)

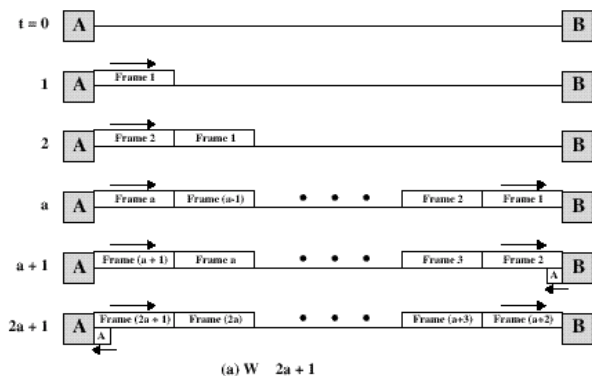


Exemplo de funcionamento de Sliding Window

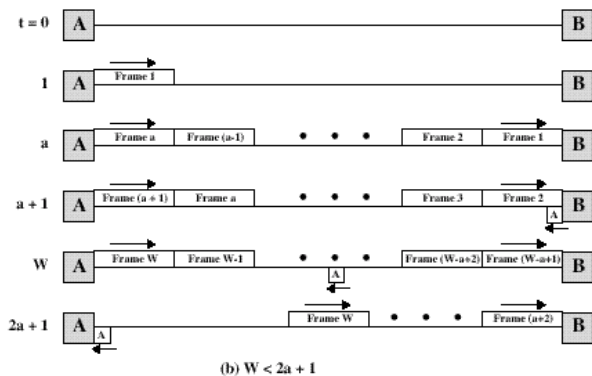


Utilização (Eficiência) da Janela Deslizante

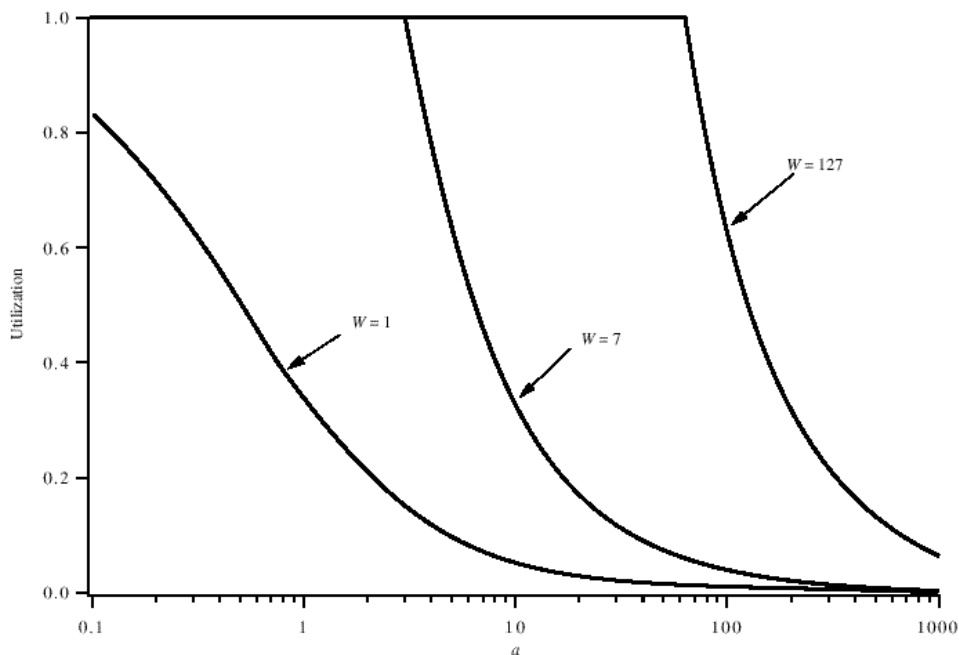
♦ Se $W \geq 1+2a \rightarrow U = 1$



♦ Se $W < 1+2a \rightarrow U = W/(1+2a)$

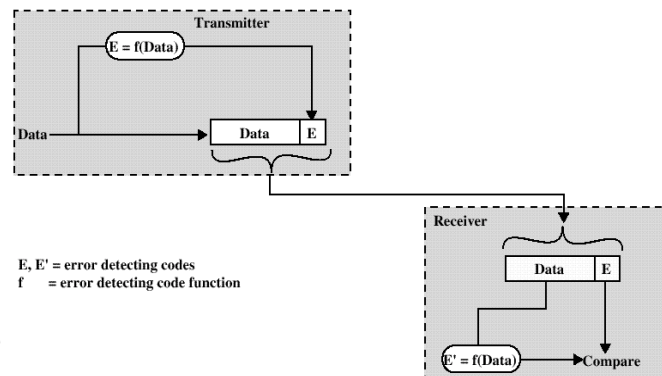


Janela Deslizante – $U = f(a, W)$



Detecção de Erros

- ◆ Códigos detectores de erros baseiam-se na adição de bits de controlo no final da trama (e ao nível de carácter, se aplicável)
- ◆ Paridade
 - » Valor do bit de paridade tal que o número de 1's seja
 - Par → paridade par
 - Ímpar → paridade ímpar
 - » Erro não detectado se o número de bits errados for par
- ◆ *Cyclic Redundancy Check (CRC)*
 - » Para um bloco de **k** bits gera **n** bits de controlo
 - » Transmite **k+n** bits divisíveis por um número (polinómio gerador do código)
 - » Receptor divide a trama por esse número
 - Se resto = 0, assume que não há erro
 - » Códigos mais usados
 - CRC-16, CRC-CCITT
 - CRC-32



Controlo de Erros

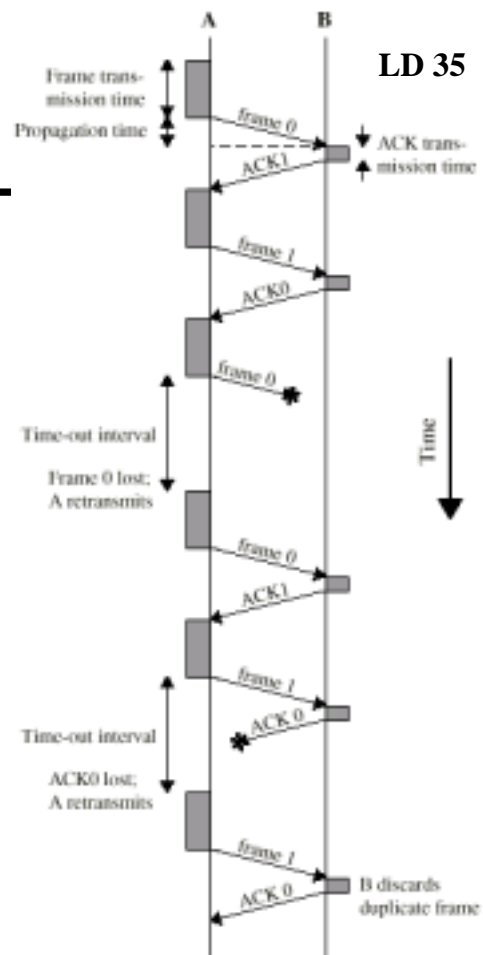
- » Objectivo: detecção e correcção de erros (por meio de retransmissão)
- » Numa Ligação de Dados as tramas são recebidas pela mesma ordem com que são enviadas, mas podem estar sujeitas a erros
 - Tramas perdidas (interrupção do meio, corrupção severa de tramas durante a transmissão, *overflow* nos *buffers* do receptor, controlo de fluxo *Stop and Go*, etc.)
 - Tramas com estrutura intacta (reconhecida) mas afectadas por erros de transmissão
- » Técnica usada → *Automatic Repeat Request (ARQ)*
 - Baseada em confirmações positivas e negativas
 - Tramas de dados não aceites pelo receptor são descartadas e terão de ser retransmitidas
 - Recuperação de erros e perdas
 - ◆ Iniciada pelo emissor após *time-out* (ausência de confirmação positiva - ACK)
 - ◆ Iniciada pelo receptor mediante envio de confirmação negativa (NAK); não estritamente necessário, mas vantajoso
- » Versões ARQ mais usadas
 - *Stop and Wait ARQ*
 - *Go-back-N ARQ / Continuous Reject ARQ* (retransmissão contínua)
 - *Selective Reject ARQ* (retransmissão selectiva)

Variáveis de Estado e Numeração de Tramas

- » Para facilitar a operação dos protocolos ARQ, cada estação deve manter duas variáveis de estado, com o objectivo de reter o valor corrente dos números de sequência $N(s)$ e $N(r)$
 - $V(s)$ - número de sequência da próxima trama de dados a ser transmitida pelo emissor
 - $V(r)$ - número de sequência da próxima trama de dados esperada pelo receptor
- » O valor das variáveis de estado é colocado a 0 após o estabelecimento da ligação de dados
- » A actualização das variáveis de estado deve ser feita de modo que
 - O campo $N(s)$ possa ser preenchido com o valor corrente de $V(s)$
 - O campo $N(r)$ possa ser preenchido com o valor corrente de $V(r)$

Stop and Wait ARQ

- ◆ Baseado no *Stop and Wait*
 - » O emissor transmite uma única trama e espera até receber ACK (ou *time-out*)
- ◆ Erros possíveis
 - » Trama de dados com erros
 - O Receptor elimina a trama (e não envia ACK)
 - O Emissor retransmite a trama depois de *time-out*
 - Em protocolos orientados ao carácter a recepção duma trama com erros pode ser sinalizada com NAK, o que desencadeia a retransmissão
 - » ACK com erros (ou com atraso excessivo)
 - O Emissor não recebe ACK
 - O Emissor retransmite a trama depois de *time-out*
 - O Receptor recebe dados em duplicado
 - O Receptor só pode detectar e eliminar duplicados se as tramas de dados forem numeradas módulo 2
 - ◆ Receptor elimina duplicado e envia ACK
 - ◆ ACK0 e ACK1 usados em alternância
- ◆ Simples mas ineficiente



LD 36

Utilização do Stop and Wait ARQ

- » P_e - probabilidade de transmissão de uma trama com erros
- » Prob [k] – Probabilidade de serem necessárias k tentativas para transmitir uma trama

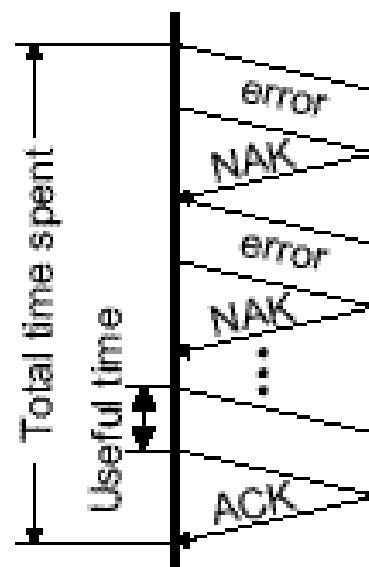
$$\text{Prob}[k] = P^{k-1}(1 - P_e)$$

- » N_r – número médio de tentativas para transmitir uma trama com sucesso

$$N_r = \sum_{k=1}^{+\infty} k * \text{Prob}[k] = \frac{1}{1 - P_e}$$

- » Utilização do *Stop and Wait ARQ*

$$U = \frac{\text{tempo}_{\text{útil}}}{\text{tempo}_{\text{total}}} = \frac{T_f}{N_r(T_f + 2 * T_p)} = \frac{1}{N_r(1 + 2a)} = \frac{1 - P_e}{1 + 2a}$$



Go-Back-N ARQ

- ◆ Baseado em *Sliding Window*
- ◆ O emissor transmite tramas de dados numeradas sequencialmente, dentro da janela
- ◆ O receptor apenas aceita tramas de dados na sequência esperada
- ◆ Tramas de dados aceites pelo receptor são confirmadas com *Receiver Ready* ou por *piggybacking* (ou excepcionalmente com *Receiver Not Ready* no caso de estar a realizar controlo de fluxo *Stop and Go*)
- ◆ Várias tramas de dados podem ser confirmadas em bloco
- ◆ Tamanho máximo da Janela $W = 2^k - 1$

Go-Back-N ARQ

- ◆ Em protocolos orientados ao bit tramas de dados com erro são simplesmente descartadas pelo receptor; apenas as tramas de dados subsequentes recebidas fora da sequência esperada desencadeiam acções de recuperação
 - » O Receptor
 - Envia confirmação negativa (*Reject* - REJ) após recepção da primeira trama recebida fora de sequência; REJ (re)confirma tramas anteriores correctamente recebidas
 - Elimina todas as restantes tramas de dados fora de sequência até receber a trama de dados em falta, mas não repete o envio de REJ (apenas pode existir um REJ pendente)
 - » O Emissor volta atrás na sequência de transmissão (*Go-Back-N*)
 - Retransmite a trama de dados indicada em REJ e (re)transmite as tramas de dados subsequentes
- ◆ Em protocolos orientados ao carácter o receptor pode enviar REJ (NAK) se uma trama de dados com erros for recebida na sequência correcta e tiver estrutura reconhecida (não consideramos este caso na análise)
- ◆ Os procedimentos de recuperação dependem do tipo de trama afectada pelo(s) erro(s) (dados, RR, REJ)

Transmissão de Tramas de Dados

- ◆ Quando o emissor tiver uma trama de dados pronta a transmitir e não existirem condições restritivas, adopta os seguintes procedimentos
 - » Coloca o valor corrente de $V(s)$ no campo $N(s)$ da trama
 - » Incrementa (módulo M) o valor de $V(s)$, apontando assim para a próxima trama de dados a transmitir
 - » Activa um temporizador (se inactivo)

- ◆ O emissor suspende a transmissão de tramas de dados se ocorrer uma das seguintes condições
 - » Número de tramas de dados pendentes igual ao tamanho da janela
 - » Recepção de RNR (pode eventualmente admitir-se o envio apenas da trama referenciada por $N(r)$)
 - » Ocorrência de *time-out* (inicia processo especial de recuperação)

Time-out

- ◆ A ocorrência de *time-out* pode dever-se a
 - » Perda da última trama de dados de uma sequência (impede o envio de REJ)
 - » Perda da confirmação (RR) da última trama de dados de uma sequência
 - A perda de uma confirmação RR não terá consequências se se referir a uma trama de dados intermédia e entretanto forem geradas novas confirmações para tramas de dados subsequentes, antes que ocorra *time-out*
 - » Perda da trama REJ enviada após o receptor detectar a perda de uma trama de dados (não é possível o envio de outro REJ)
 - » Perda da trama de dados retransmitida após recepção de REJ (não é possível o envio de outro REJ)

- ◆ Recuperação após *time-out*
 - » O emissor envia uma trama de controlo (RR) como comando, forçando o receptor a responder (indicando assim o seu estado)
 - O emissor poderia em alternativa retransmitir a trama de dados não confirmada, o que poderia originar um duplicado e tornar o processo de recuperação complexo
 - » A trama de resposta indica em $N(r)$ qual a trama de dados a partir da qual o emissor deve (re)transmitir

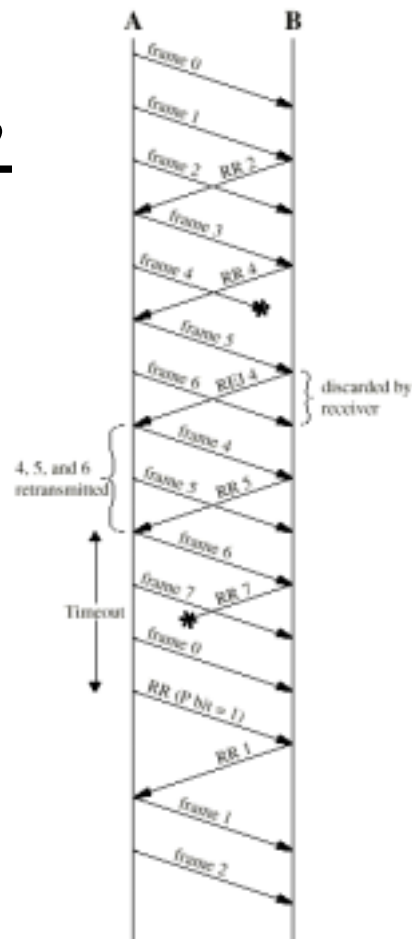
Recepção de Tramas de Controlo

- » O emissor de tramas de dados analisa o valor $N(r)$ recebido em tramas de controlo (ou em tramas de dados no caso de *piggyback*); se representar uma confirmação efectiva de tramas de dados pendentes suspende o temporizador e reactiva-o se subsistir(em) trama(s) de dados não confirmada(s)
- » Se a trama recebida for
 - **RR** - a transmissão de tramas de dados pode continuar ou recomeçar, caso estivesse suspensa (e.g., RNR anteriormente recebido ou janela esgotada)
 - **RNR** - deve suspender-se o envio de tramas de dados (podendo eventualmente transmitir-se a trama de dados referenciada por $N(r)$, para evitar *deadlocks* resultantes duma eventual perda do RR ou REJ subsequente; a alternativa seria activar um temporizador após a recepção de RNR)
 - **REJ** - deve(m) retransmitir-se a(s) trama(s) de dados a partir da referenciada por $N(r)$, começando por fazer-se $V(s) = N(r)$ (*Go-Back-N*)
- » A recepção de uma trama de controlo em resposta ao comando accionado na sequência de *time-out* termina a situação de excepção e permite retomar os procedimentos normais

Recepção de Tramas de Dados

- » Em *Go-Back-N* o receptor apenas aceita tramas de dados em sequência, pelo que deve comparar o valor do campo $N(s)$ com $V(r)$
- » No caso de a estação não estar a realizar controlo de fluxo
 - Se $N(s) = V(r)$
 - ◆ Aceita a trama (eliminando uma eventual condição de rejeição estabelecida por REJ anterior)
 - ◆ Incrementa (módulo M) o valor de $V(r)$, apontando assim para a próxima trama de dados que espera receber
 - ◆ Na primeira oportunidade envia uma confirmação, por meio de RR ou *piggyback*, colocando o valor corrente de $V(r)$ no campo $N(r)$ da trama respectiva
 - Se $N(s) \neq V(r)$
 - ◆ Descarta a trama
 - ◆ Envia REJ, com o valor corrente de $V(r)$ no campo $N(r)$, se não houver já REJ pendente
- » No caso de a estação se encontrar a realizar controlo de fluxo
 - Envia RNR, se ainda o não fez, confirmando a(s) trama(s) de dados realmente aceite(s)
 - Pode aceitar ou não a trama de dados recebida, mantendo a informação correspondente para posteriormente accionar RR ou REJ, conforme o caso

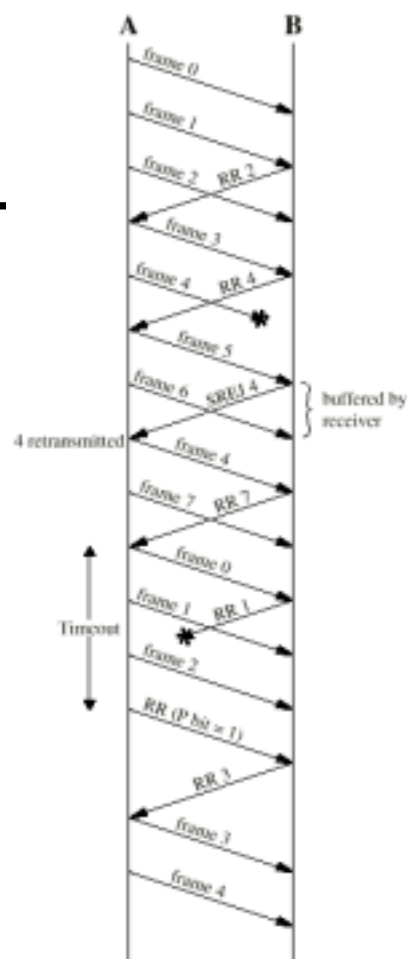
Go-Back-N ARQ - Exemplo



LD 43

Selective Reject ARQ

- ◆ Baseado em *Sliding Window*
- ◆ O receptor aceita tramas fora de sequência
 - » Confirma negativamente com SREJ tramas de dados em falta
 - » Confirma positivamente com RR apenas blocos de tramas de dados consecutivas
- ◆ O emissor apenas retransmite as tramas sinalizadas por SREJ, o que minimiza o número de retransmissões
 - » Vantajoso se W muito grande (e.g., satélite)
- ◆ Requer
 - » No receptor
 - Buffers maiores
 - Lógica para reinserção de tramas
 - » No emissor
 - Lógica mais complexa para enviar tramas fora da sequência natural
- ◆ Tamanho máximo da Janela $W = 2^{k-1}$



LD 44

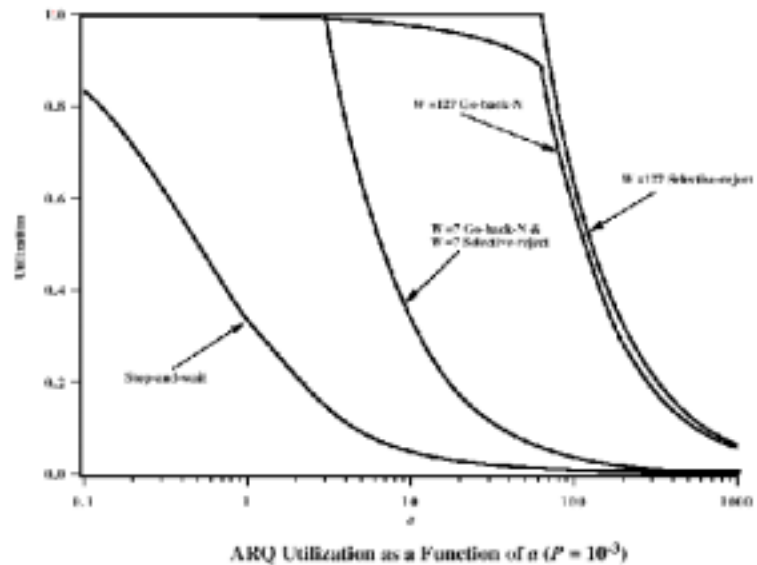
Utilização de Go-Back-N e Selective Reject ARQ

» Go-Back-N ARQ

$$U = \begin{cases} \frac{1 - P_e}{1 + 2aP_e} & , W \geq 1 + 2a \\ \frac{W(1 - P_e)}{(1 + 2a)(1 - P_e + WP_e)} & , W < 1 + 2a \end{cases}$$

» Selective Reject ARQ

$$U = \begin{cases} \frac{1 - P_e}{1 + 2a} & , W \geq 1 + 2a \\ \frac{W(1 - P_e)}{1 + 2a} & , W < 1 + 2a \end{cases}$$



High Level Data Link Control (HDLC)

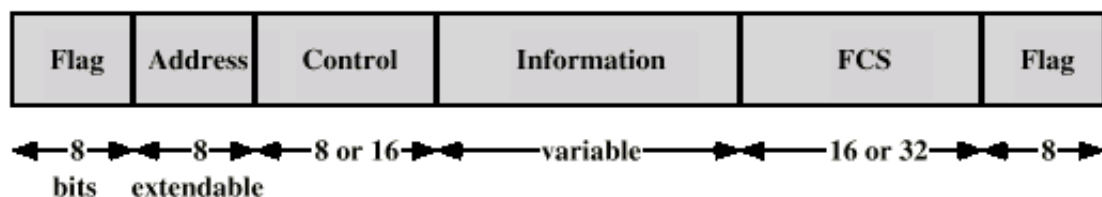
- » HDLC (ISO 33009, ISO 4335)
- » Protocolo orientado ao bit (independente de códigos)
- » Tipos de estações
 - Estação Primária
 - ◆ Controla a operação da ligação (uma ligação por cada estação Secundária)
 - ◆ Tramas enviadas designam-se por **comandos**
 - Estação Secundária
 - ◆ Controlada pela estação Primária
 - ◆ Tramas enviadas designam-se por **respostas**
 - Estação Combinada
 - ◆ Combina funções Primárias e Secundárias; pode enviar **comandos** e **respostas**
- » Configurações da ligação
 - Não balanceada (não equilibrada)
 - ◆ Uma Primária e uma ou mais Secundárias
 - ◆ *Full-duplex* ou *half-duplex*
 - Balanceada (equilibrada)
 - ◆ Duas estações Combinadas
 - ◆ *Full-duplex* ou *half-duplex*

Modos de Transferência de Dados em HDLC

- » **Normal Response Mode (NRM)**
 - Configuração não balanceada
 - A estação Primária é responsável pelo estabelecimento e gestão da ligação
 - Uma estação Secundária só responde a comandos da estação Primária
 - Usado em configurações *multidrop* (e.g., Primário: Computador, Secundárias: terminais)
- » **Asynchronous Response Mode (ARM)**
 - Configuração não balanceada
 - A estação Primária é responsável pelo estabelecimento e gestão da ligação
 - A estação Secundária pode transmitir sem permissão da estação Primária, após estabelecida a ligação
 - Raramente usado
 - ♦ Foi proposta na primeira versão de X.25 (LAP) para garantir operação simétrica do terminal e do nó de comutação (actualmente sem interesse); requeria uma Estação Primária e uma Estação Secundária em cada sistema e portanto duas ligações ARM
- » **Asynchronous Balanced Mode (ABM)**
 - Configuração balanceada
 - Qualquer das estações pode estabelecer a ligação e transmitir sem permissão da outra
 - Usado em X.25 (LAPB), RDIS (LAPD) e *Frame Relay* (LAPF)

Estrutura de Trama HDLC

- » Em HDLC a transmissão é síncrona e organizada em tramas
- » Uma trama HDLC é constituída por vários campos
 - Os campos têm tamanho fixo ou variável (existindo mecanismos que permitem estabelecer e reconhecer as fronteiras entre campos)
 - As funções associadas a cada campo dependem da respectiva estrutura e de valores dos bits que o constituem, de acordo com a posição que ocupam no campo
- » Formato único para todas as tramas de controlo e dados



Trama HDLC - Flag

- ◆ Tramas HDLC são delimitadas por *flags* - seqüências 01111110
- ◆ Uma *flag* pode terminar uma trama e começar outra
- ◆ *Bit stuffing* é usado para evitar o falso reconhecimento de *flags* dentro da trama
 - » O Emissor insere um 0 após uma seqüência de cinco 1's
 - » Quando o Receptor detecta cinco 1's consecutivos verifica o 6º e 7º bits
 - Se 6º bit = 0 → elimina bit
 - Se 6º bit = 1 e 7º bit = 0 → aceita *flag*
 - Se 6º bit = 1 e 7º bit = 1 → indica que o emissor abortou a transmissão

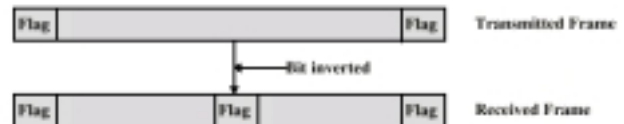
Original Pattern:

1111111111110111111101111110

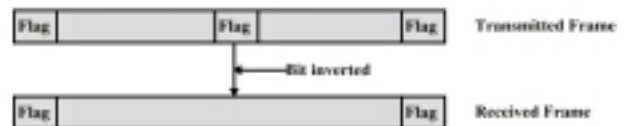
After bit-stuffing

111110111111011011111101011111010

(a) Example



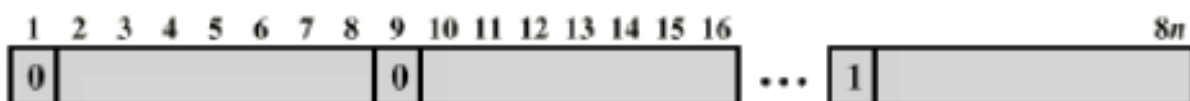
(b) An inverted bit splits a frame in two



(c) An inverted bit merges two frames

Trama HDLC - Campo de Endereço

- » Por convenção o endereço identifica a estação secundária que enviou ou vai receber a trama (resposta e comando, respectivamente); um bit (C/R) neste campo permite estabelecer a diferença entre comandos e respostas
- » Formato básico - 8 bits
- » Formato expandido
 - Múltiplo de 8 bits
 - LSB de cada octeto indica se este é o último octeto (1) ou não (0)
- » Endereço de *broadcast* - 11111111

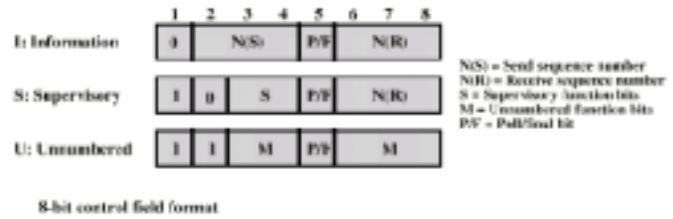


Extended Address Field

Trama HDLC - Campo de Controle

» 3 tipos de tramas (funções)

- Informação (I)
 - ◆ Contém dados de camadas superiores
 - ◆ Confirmação de tramas (*piggyback*)
- Supervisão (S)
 - ◆ Confirmação de tramas, Controlo de Erro e de Fluxo (RR, RNR, REJ, SREJ)
- Não numeradas (U)
 - ◆ Controlo da ligação (estabelecimento, reinicialização, terminação)

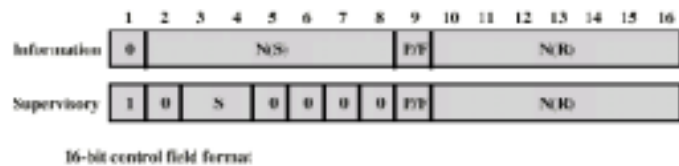


» Números de sequência

- N(S) em tramas I
- N(R) em tramas I e S

» Bit P/F (Poll / Final)

- Permite sincronizar um comando com uma resposta (o funcionamento normal do protocolo não garante uma relação um para um entre comandos e respostas)
- Em comandos, P=1 solicita resposta
- Em respostas F=1 indica resposta solicitada por um comando com P=1



HDLC Commands and responses

Name	Command/ Response	Description
Information (I)	C/R	Exchange user data
Supervisory (S)		
Receive ready (RR)	C/R	Positive acknowledgment; ready to receive I-frame
Receive not ready (RNR)	C/R	Positive acknowledgment; not ready to receive
Reject (REJ)	C/R	Negative acknowledgment; go back N
Selective reject (SREJ)	C/R	Negative acknowledgment; selective reject
Unnumbered (U)		
Set normal response/extended mode (SNRM/SNRMIE)	C	Set mode; extended = 7-bit sequence numbers
Set asynchronous response/extended mode (SARM/SARME)	C	Set mode; extended = 7-bit sequence numbers
Set asynchronous balanced/extended mode (SABM/SABME)	C	Set mode; extended = 7-bit sequence numbers
Set initialization mode (SIM)	C	Initialize link control functions in addressed station
Disconnect (DISC)	C	Terminate logical link connection
Unnumbered Acknowledgment (UA)	R	Acknowledge acceptance of one of the set-mode commands
Disconnected mode (DM)	C	Terminate logical link connection
Request disconnect (RD)	R	Request for DISC command
Request initialization mode (RIM)	R	Initialization needed; request for SIM command
Unnumbered information (UI)	C/R	Used to exchange control information
Unnumbered poll (UP)	C	Used to solicit control information
Reset (RSFET)	C	Used for recovery; resets N(R), N(S)
Exchange identification (XID)	C/R	Used to request/report status
Test (TEST)	C/R	Exchange identical information fields for testing
Frame reject (FRMR)	R	Reports receipt of unacceptable frame

Trama HDLC – Campos de Informação e FCS

◆ Campo de Informação

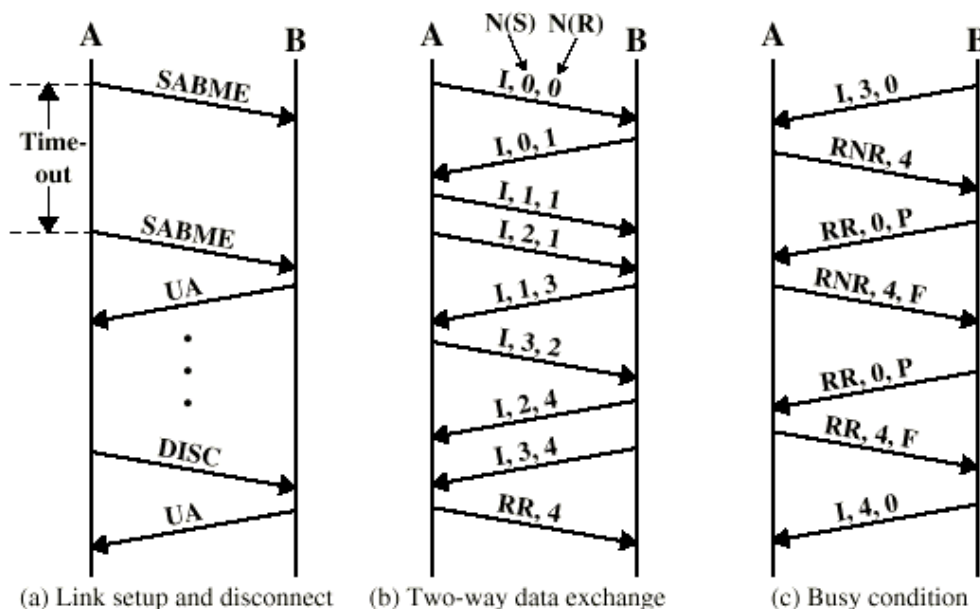
- » Apenas presente em tramas I e em algumas tramas U
- » Contém tipicamente um número inteiro de octetos
- » Comprimento variável (valor máximo fixado previamente)

◆ FCS (*Frame Check Sequence*)

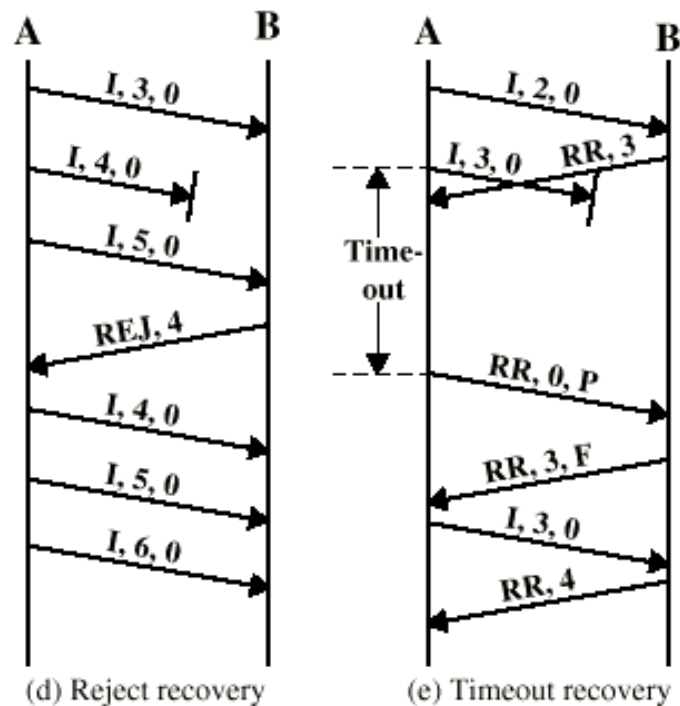
- » Baseado num código polinomial
- » Normalmente CRC-CCITT (16 bits)
- » Por vezes CRC-32 (32 bits)

Exemplos de Funcionamento do HDLC

- » Três fases: estabelecimento, transferência de dados, terminação



Exemplos de Funcionamento do HDLC



LAPB (X.25) e LAPD (RDIS)

- ◆ *Link Access Procedures, Balanced (LAPB)*
 - » Parte do X.25 (ITU-T)
 - » Subconjunto do HDLC (ABM)
 - » Ligação ponto-a-ponto entre sistema e nó da rede de comutação de pacotes
- ◆ *Link Access Procedures, D-Channel (LAPD)*
 - » RDIS / ISDN (ITU-T)
 - » ABM
 - » Números de sequência com 7 bits (numeração estendida)
 - » Campo de endereço com 16 bits
 - Permite multiplexagem de ligações de dados na interface utilizador-rede, o que requer dois níveis de identificação (SAPI, TEI) para suportar:
 - ◆ Múltiplas entidades em cada sistema (sinalização, comunicação em modo pacote ou trama, etc.) - *Service Access Point Identifier (SAPI)*
 - ◆ Múltiplos terminais na mesma interface com a rede - *Terminal Endpoint Identifier (TEI)*; cada terminal necessita de ter pelo menos um TEI atribuído

LAPF (Frame Relay)

- ◆ *Link Access Procedures for Frame-Mode Bearer Services (LAPF)*
 - » Serviços de suporte baseados na comutação de tramas transportadas em circuitos virtuais estabelecidos por meio de procedimentos de sinalização, que usam canais lógicos distintos dos usados para transferência de dados
 - » Dois subníveis
 - *DL-core* - funções básicas (apenas campo de endereço)
 - *DL-control* - confirmação, controlo de erros e controlo de fluxo
 - » ABM
 - » Números de sequência com 7 bits (*DL-control*)
 - » CRC de 16 bits
- ◆ Campo de endereços com 2, 3 ou 4 octetos
 - » Identificação da Ligação de Dados
 - *Data Link Connection Identifier (DLCI)*
 - » Funções diversas
 - Notificação de congestionamento e indicação para descarte prioritário de tramas pela rede

PPP (Point-to-Point Protocol)

- ◆ Protocolo de Ligação de Dados usado em ligações físicas ponto a ponto na Internet - definido nos RFCs 1661, 1662 e 1663
- ◆ Características
 - » Suporta detecção de erros
 - » Suporta múltiplos protocolos
 - » Permite negociação de endereços IP
 - » Permite autenticação
- ◆ Mecanismos
 - » Delineação de tramas e detecção de erros
 - » Protocolo de controlo de ligação (*LCP - Link Control Protocol*) para estabelecimento, teste e terminação de ligações e negociação de opções
 - » Protocolo de controlo de rede (*NCP - Network Control Protocol*) específico de cada protocolo de rede para negociação de opções da camada de Rede

Tramas PPP

- ◆ Formato baseado em HDLC
 - » Número inteiro de octetos (*character stuffing* em vez de *bit stuffing*)
 - » Delineação por *flags* (com *character stuffing* se ocorrer uma *flag* na trama)
 - » Campo de Endereço (1 octeto)
 - 11111111 (indica que trama deve ser aceite por qualquer estação)
 - » Campo de Controlo (1 octeto)
 - valor *default* 00000011 (trama não numerada) - transmissão não fiável (erros são detectados mas não recuperados por retransmissão)
 - RFC 1663 define um modo de transmissão fiável para usar em meios adversos (*wireless*)
 - » Campo de Protocolo (1 ou 2 octetos)
 - Identifica o protocolo transportado no campo de dados (LCP, NCP, IP, etc.)
 - » Campo de dados (tamanho variável - número inteiro de octetos)
 - Valor máximo negociado; *default*: 1500 octetos
 - » Campo de verificação de paridade (2 ou 4 octetos)
 - Tipicamente 2 octetos, podendo ser negociados 4 octetos
 - » LCP permite negociar a omissão dos campos de Endereço e de Controlo

LCP - Link Control Protocol

- ◆ Após estabelecimento da ligação física (por exemplo, por *dial-up*) inicia-se a fase de negociação de opções
- ◆ Se esta fase se completar com sucesso passa-se à fase de autenticação - as duas partes verificam as respectivas identidades
- ◆ Na fase seguinte é invocado o protocolo NCP apropriado para configurar a camada de Rede (de acordo com o protocolo de Rede acordado na fase de negociação)
- ◆ Completada a fase de configuração, pode dar-se início à troca de dados, de acordo com o protocolo seleccionado (por exemplo IP)
- ◆ Concluída a troca de dados a ligação PPP é terminada e a ligação física abandonada

Logical Link Control (LANs IEEE 802)

- » Parte da arquitectura das LANs IEEE 802
 - Controlo da ligação dividido em
 - ◆ *Medium Access Control (MAC)*
 - ◆ *Logical Link Control (LLC)*
- » Subcamada MAC
 - Protocolo de acesso a um meio partilhado
 - Endereçamento físico das estações
 - Detecção de erros (sem recuperação)
- » Subcamada LLC
 - Endereçamento lógico permite identificação (multiplexagem) de protocolos de camadas superiores
 - ◆ *DSAP - Destination Service Access Point*
 - ◆ *SSAP - Source Service Access Point*
 - Três tipos de serviços
 - ◆ confirmado, com conexão (fiável) - procedimentos semelhantes a HDLC
 - ◆ confirmado, sem conexão
 - ◆ não confirmado, sem conexão (não fiável) - o mais comum (usado, por exemplo, com TCP/IP)

Protocolos orientados ao caracter

- » Os protocolos orientados ao caracter utilizam caracteres de controlo de um código (por exemplo o código ASCII) para delimitar os blocos de dados (tramas) e para supervisionar a troca de dados
- » Os protocolos orientados ao caracter apresentam limitações importantes que os tornam pouco eficientes e mal adaptados para aplicações interactivas
 - caracterizam-se por ligação rígida a um código, o que requer versões diferentes para códigos diferentes, tornando-os pouco flexíveis para futuras expansões (difícil modificação) e inerentemente não transparentes (a transparência consegue-se recorrendo a caracteres de escape, o que aumenta a complexidade)
 - utilizam procedimentos do tipo bidireccional alternado, o que exige um número elevado de *turnarounds* da ligação lógica (em particular para a confirmação individual de tramas de dados)
 - em geral apenas uma função é realizada por cada trama enviada
 - apenas as tramas de dados estão protegidas contra erros
 - a necessidade de distinguir dados e caracteres de controlo cria *overheads* adicionais
- » Um exemplo de um protocolo deste tipo é o BISYNC (*Binary Synchronous*)

Caracteres de Controlo (ASCII)

- » **SYN** *Synchronous Idle*; caracter de sincronismo de trama
- » **SOH** *Start of Heading*; delimitador de início de cabeçalho
- » **STX** *Start of Text*; delimitador de início de texto (dados)
- » **ETB** *End of Transmission Block*; delimitador de fim de dados numa trama intermédia de uma mensagem
- » **ETX** *End of Text*; delimitador de fim de dados na última trama de uma mensagem
- » **ENQ** *Enquiry*; solicita uma resposta (usado em *Polling / Selecting*)
- » **ACK** *Acknowledgement*; confirmação (positiva) de recepção de trama
- » **NAK** *Negative Acknowledgement*; solicita retransmissão em virtude de ter sido recebida uma trama com erro (confirmação negativa)
- » **EOT** *End of Transmission*; indica fim de transmissão (terminação da ligação)
- » **DLE** *Data Link Escape*; modifica o significado do caracter seguinte
 - **DLE STX** inicia um bloco de dados transparentes numa mensagem
 - **DLE ETB** termina um bloco intermédio de dados transparentes numa mensagem
 - **DLE ETX** termina o último bloco de dados transparentes numa mensagem
 - **DLE DLE** transmissão de DLE num bloco de dados transparentes
 - **DLE 0 / DLE 1** realiza **ACK0 / ACK1**

Formatos de Tramas

Exemplos de formatos de tramas de dados e controlo

- » SYN SYN SOH Cabeçalho ETB BCC
- » SYN SYN SYN STX Dados ETB/ETX BCC
- » SYN SYN SOH Cabeçalho STX Dados ETB/ETX BCC
- » SYN SYN DLE STX Dados transparentes DLE ETB/ETX BCC
- » SYN SYN Endereço ENQ
- » SYN SYN ACK (ou SYN SYN DLE 0 / SYN SYN DLE 1)
- » SYN SYN NAK
- » SYN SYN EOT

As tramas que contêm cabeçalho e/ou dados são protegidas por um código detector de erros baseado em caracteres de controlo de paridade no fim da trama (BCC - *Binary Check Character*), do tipo LRC (*Longitudinal Redundancy Check*) ou CRC (*Cyclic Redundancy Check*)

Polling

- » **P:** O Primário inicia *Polling* ao Secundário designado (início da ligação)
SYN SYN Endereço (Poll) ENQ
- » **S:** Se o Secundário não pretender transmitir, envia
SYN SYN EOT
Se estiver em condições de transmitir, envia dados de acordo com formato apropriado; antes de transmitir a trama de dados seguinte espera confirmação do Primário
- » **P:** O Primário confirma individualmente cada trama recebida
SYN SYN ACK (ou ACK0/ACK1)
- » **S:** No final da troca de dados, para devolver controlo ao Primário (fim da ligação), envia
SYN SYN EOT
- » **P:** Pode iniciar *Polling* (ou *Selecting*) a outro terminal

Selecting

- » **P:** O Primário inicia *Selecting* ao Secundário designado (início da ligação)
SYN SYN Endereço (Select) ENQ
- » **S:** Se o Secundário estiver pronto a receber dados, envia
SYN SYN ACK (ou ACK0)
Se não estiver em condições de receber, envia
SYN SYN NAK
- » **P:** No primeiro caso o Primário inicia a transferência de dados, esperando uma confirmação do Secundário antes de enviar uma nova trama
No segundo caso o Primário pode tentar novamente (repetindo o processo um número de vezes pré-determinado) ou terminar com
SYN SYN EOT
- » **S:** No primeiro caso o Secundário confirma uma a uma cada trama recebida
SYN SYN ACK (ou ACK0/ACK1)
- » **P:** No final da troca de dados, para terminar a ligação envia
SYN SYN EOT
e pode iniciar novo *Polling* ou *Selecting*