Inatel
Instituto Nacional de Telecomunicações

Unidade II – Camada de Enlace

TP308 – Introdução às Redes de Telecomunicações

Tópicos

- ✓ Serviços Providos pela Camada de Enlace
  - Delimitação de Quadros
  - Controle de Erros
  - Controle de Fluxo
- ✓ Protocolos de Retransmissão
- ✓ HDLC
- ✓ PPP

Serviços Providos pela Camada de Enlace

- ✓ O objetivo da camada de enlace é lapidar um enlace de transmissão ponto-a-ponto cru, transformando-o em um enlace confiável e eficiente para a camada de rede.
- ✓ Para isso os seguintes serviços são providos pela camada de enlace:
  - Delineamento de Quadros
  - Controle de Erros
  - Controle de Fluxo
  - Transmissão de Dados Ponto-a-Ponto
  - Controle de Acesso ao Meio
    - Será discutido mais adiante.

Delineamento de Quadros

- ✓ Para prover o serviço de transporte ponto a ponto a camada de enlace conta com os serviços providos pela camada física.
- ✓ A camada física por sua vez tenta enviar os *frames* da camada de enlace como um fluxo de *bits* através do meio físico.
- ✓ Este fluxo de *bits* pode sofrer erros durante a transmissão, de forma que:
  - O número de *bits* recebido pode ser igual, menor ou maior que o número de *bits* transmitidos.
  - Os *bits* recebidos podem estar corrompidos.

Delineamento de Quadros

- ✓ Assim, a camada física confia para a camada de enlace a detecção e, se necessário, a correção de erros.
- ✓ A solução mais usual para resolver este problema é calcular um *checksum* para cada *frame* transmitido e transmiti-lo juntamente com o *frame*.
- ✓ Quando o *frame* é recebido no destino, o *checksum* é recalculado e comparado com o *checksum* recebido.
- ✓ Se eles forem diferentes, a camada de enlace saberá que ocorreu um erro e poderá tentar recuperar a informação original danificada.

Delineamento de Quadros

- ✓ Contudo, para calcular este *checksum* é preciso conhecer o inicio e o fim de cada *frame*.
- ✓ Uma possibilidade seria deixar intervalos de tempo entre os *frames*, entretanto além de ineficiente esta solução teria que contar com uma rede totalmente síncrona, o que não existe na prática.
- ✓ Assim, alguns métodos alternativos foram desenvolvidos:
  - *Bytes* de *Flag* com Enchimento de *Bytes*
  - *Flags* de Inicio e Fim com Enchimento de *Bits*

Delineamento de Quadros

✓ *Bytes* de *Flag* com Enchimento de *Bytes*

- Resolve o problema da resincronização após um erro, através da utilização de *bytes* especiais no inicio e fim de cada *frame*.
- O valor do *byte* é o mesmo tanto no inicio quanto no final de cada *frame*.
- Este byte é conhecido por *byte* de *flag* ou *byte* bandeira.
- Desta forma, se o receptor perder um *frame*, ele só precisa procurar pelo próximo *byte* de *flag* para localizar o inicio do próximo *frame*.
- Dois *bytes* de *flag* consecutivos indicam o final de um *frame* e o inicio de outro.

Delineamento de Quadros

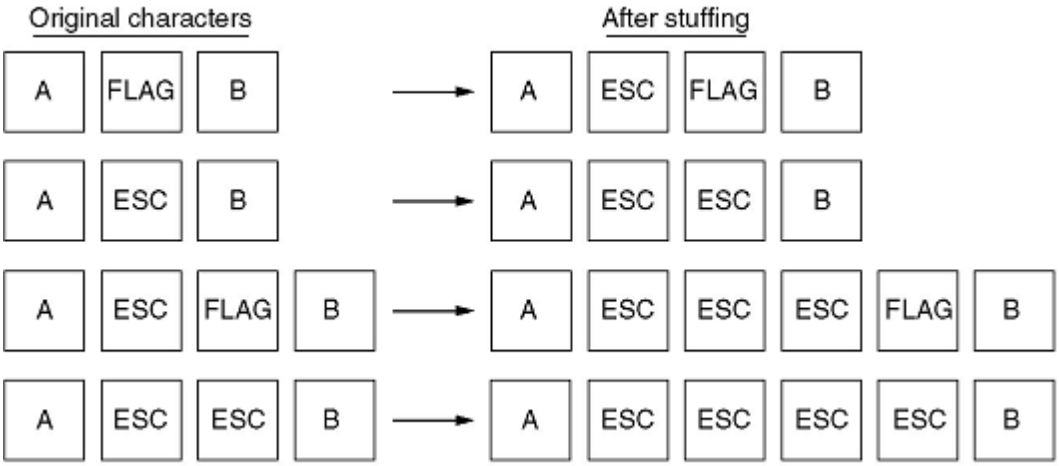
✓ *Bytes* de *Flag* com Enchimento de *Bytes* (cont.)

- Entretanto, uma situação critica pode acontecer neste método: o que aconteceria se no *frame* existisse um caractere de dados igual ao *byte* de *flag*?
- Para resolver este problema, a camada de enlace do transmissor deve inserir um caractere especial de escape (ESC) antes de qualquer *byte* de *flag* encontrado no *frame*.
- Esta técnica é chamada de enchimento de *bytes* (*byte stuffing*).
- E se um *byte* igual ao caractere especial de escape aparecer nos dados?

Delineamento de Quadros

✓ *Bytes* de *Flag* com Enchimento de *Bytes* (cont.)

- Também será inserido um caractere especial de escape a sua frente, indicando que este caractere apareceu naturalmente nos dados.



Fonte: Tanenbaum

Delineamento de Quadros

✓ *Flags* de Inicio e Fim com Enchimento de *Bits*

- Neste método é utilizado um padrão de *bits* para indicar o início e o fim de um quadro, inserindo-se *bits* de escape quando a seqüência de início e/ou fim aparece dentro dos dados transmitidos.

Dado a transmitir:	0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0
Sinalização:	0 1 1 1 1 1 1 0
Enquadramento:	01111110 | 011011111 **0** 11111 **0** 11111 **0** 10010 | 01111110

- A cada seqüência de 5 *bits* “1” consecutivos nos dados a serem transmitidos é inserido um *bit* de escape “0”.
- Quando o receptor recebe 5 bits “1” consecutivos, seguido por um bit “0”, ele automaticamente retira o *bit* “0”.

Controle de Erros

- Tendo resolvido o problema do delineamento de quadros, o próximo problema é:
  - Como ter certeza que todos os *frames* transmitidos foram recebidos sem erros e na ordem adequada pela camada de rede?
- A maneira usual de assegurar o envio confiável de informações é provendo o transmissor com algum *feedback* a respeito do que esta acontecendo na outra ponta do enlace.
- Tipicamente, isto é feito através do uso de um esquema de confirmações positivas/negativas.

Controle de Erros

- Se o transmissor recebe uma confirmação positiva (*positive acknowledgement*) sobre um *frame* transmitido, significa que este *frame* chegou no destino com sucesso.
- Por outro lado, se o transmissor receber uma confirmação negativa (*negative acknowledgement*) sobre um *frame* transmitido, significa que este *frame* não chegou no destino com sucesso.
- Agora, o que aconteceria se um receptor nunca chegasse a receber um *frame* transmitido devido a uma falha de *hardware*?

Controle de Erros

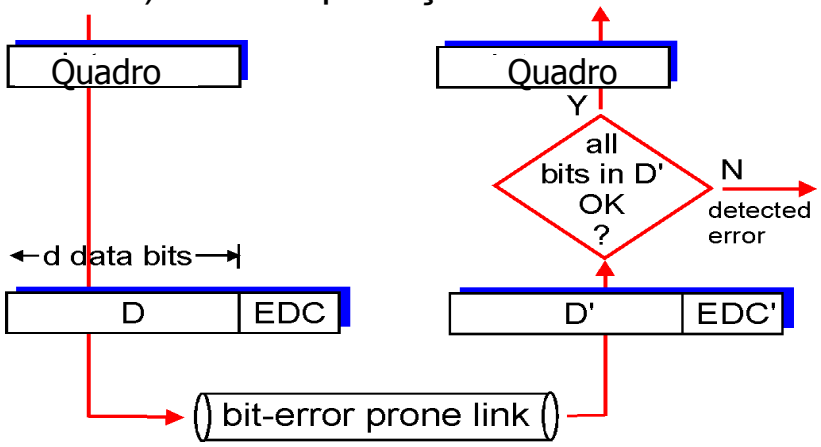
- ✓ O receptor não faria nada, pois ele não tem motivo nenhum para reagir.
- ✓ Já o transmissor poderia ficar esperando para sempre pela confirmação positiva/negativa do *frame* transmitido.
- ✓ Para resolver este problema a solução comumente adotada é a utilização de um relógio.
- ✓ Quando o transmissor envia um *frame*, ele inicializa um relógio, que é configurado para expirar depois de um certo intervalo de tempo.
- ✓ Este intervalo de tempo deve ser suficiente para que o *frame* seja transmitido, processado e confirmado.

Controle de Erros

- ✓ Entretanto, se o *frame* ou a confirmação forem perdidas, o relógio vai zerar alertando o transmissor de que algo deu errado.
- ✓ A solução óbvia neste caso é retransmitir o *frame*.
- ✓ Agora, o que aconteceria se o relógio zerasse, o *frame* fosse retransmitido, e por algum motivo, o *frame* anterior também chegasse ao ponto de destino?
- ✓ Para evitar este problema, geralmente são utilizados números de seqüência de tal forma que o receptor possa separar os *frames* originais de *frames* retransmitidos.

Controle de Erros

- ✓ Mas como é feita a detecção/correção dos erros?
  - São acrescentados *bits* de detecção e correção de erros (EDC) que oferecem redundância na transmissão.
  - Tanto os dados quanto o cabeçalho podem ser protegidos.
  - Quanto maior o número de *bits* EDC (*Error Detection and Correction*) maior a proteção fornecida.

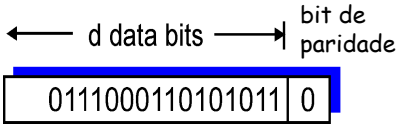


Controle de Erros

- ✓ Verificação de Paridade

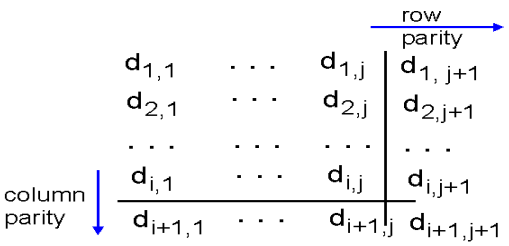
Paridade com Bit único:

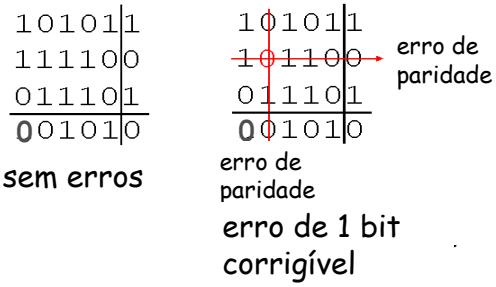
Detecta erro de um único bit



Paridade Bi-dimensional:

Detecta *e corrige* erros de um único bit





Controle de Erros

✓ CRC – *Cyclic Redundancy Checking*

- Dado um bloco de dados M(x), o transmissor gera um bloco T(x), tal que T(x) seja exatamente divisível por um polinômio gerador P(x) com grau r.
- Para se obter T(x), deve-se multiplicar M(x) por x^r, e dividir o resultado por P(x). O resto desta divisão, ou seja R(x), é somada a x^r . M(x), produzindo:
- $T(x) = x^r . M(x) + R(x)$
- No receptor, T'(x) é dividido por P(x). Se não houver resto, T'(x) é considerado igual a T(x).

Controle de Erros

✓ CRC – *Cyclic Redundancy Checking*

- Mensagem = 1010001101 => $M(x) = x^9 + x^7 + x^3 + x^2 + 1$
  Polinômio gerador: $P(x) = x^5 + x^4 + x^2 + 1$

Transmissão

1. $M(x).x^r = M(x).x^5 = x^{14} + x^{12} + x^8 + x^7 + x^5$
2. $x^r.M(x)/P(x) = Q(x) + R(x)/P(x)$
   $Q(x) = x^9 + x^8 + x^6 + x^4 + x^2 + x$
   $R(x) = x^3 + x^2 + x$
3. $T(x) = x^r.M(x) + R(x)$
   $T(x) = x^{14} + x^{12} + x^8 + x^7 + x^5 + x^3 + x^2 + x$

Controle de Erros

- ✓ CRC – *Cyclic Redundancy Checking*
  - Mensagem = 1010001101 => $M(x) = x^9 + x^7 + x^3 + x^2 + 1$
    Polinômio gerador: $P(x) = x^5 + x^4 + x^2 + 1$

Recepção

1. Divide-se $T'(x)/P(x) = Q(x) + R(x)/P(x)$
   $T'(x) = x^{14} + x^{12} + x^8 + x^7 + x^5 + x^3 + x^2 + x$
   $P(x) = x^5 + x^4 + x^2 + 1$
   $R(x) = 0$
2. Se $R(x) = 0$, $T'(x) = T(x)$.

Controle de Fluxo

- ✓ Outro importante serviço provido pela camada de enlace é o controle de fluxo.
- ✓ O controle de fluxo visa resolver o seguinte problema: o que fazer com um transmissor que transmite mais informação do que o receptor pode aceitar.
- ✓ Esta situação pode acontecer quando o transmissor está rodando em uma máquina rápida (ou sem carga) e o receptor em uma máquina lenta (ou sobrecarregada).
- ✓ O transmissor continua enviando *frames* até que o receptor fique completamente sobrecarregado e não tenha outra alternativa a não ser descartar *frames*.

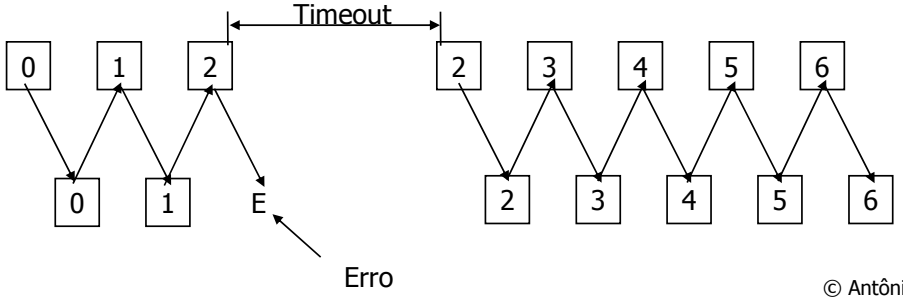
Protocolos de Retransmissão

- Dentre os principais protocolos de retransmissão temos:
  - Protocolo ***Stop-and-Wait***
  - Protocolos de **Janela Deslizante**
    - ***Go-Back N***
    - ***Selective Repeat***

- Estes protocolos também são utilizados para controle de fluxo e de congestionamento.

Protocolo Stop-and-Wait

- O transmissor envia um *frame* e aguarda por uma confirmação antes de enviar o próximo *frame.*
- Se o *frame* recebido estiver corrompido, o receptor irá descartá-lo.
- Se o *frame* não for recebido, o transmissor irá retransmiti-lo após um *timeout*.
- A transmissão de quadros é *half-duplex*.



Protocolo Stop-and-Wait

- ✓ É importante observar que tanto os *frames* de dados quanto os de confirmação podem ser danificados ou perdidos completamente.
- ✓ Em geral, assume-se que se um *frame* for danificado em trânsito, o *hardware* do receptor detectará este erro quando o *checksum* for computado.
- ✓ Para contornar a ocorrência de erros de transmissão um relógio geralmente é utilizado no transmissor.
- ✓ As confirmações são utilizadas não só para controle de fluxo mas também para efetuar retransmissões de *frames* perdidos.

Inatel
Instituto Nacional de Telecomunicações

Camada de Enlace
Protocolos de Retransmissão
Unidade 2

Protocolo Stop-and-Wait

- ✓ Números de seqüência são utilizados para evitar que um mesmo *frame* seja enviado duas ou mais vezes para a camada de rede no receptor.
- ✓ Isto pode acontecer se uma confirmação positiva fosse perdida.
- ✓ Protocolos na qual o transmissor aguarda por uma confirmação positiva antes de avançar para o próximo *frame* são chamados de PAR (*Positive Acknowledgment with Retransmission*).

© Antônio M. Alberti 2007

Piggybacking

- ✓ No protocolo anterior, os *frames* são transmitidos em uma direção apenas (*simplex* ou *half-duplex*).
- ✓ Na prática, existe a necessidade de transmissão nas duas direções (*full-duplex*).
- ✓ Uma maneira de se fazer a transmissão *full-duplex* é alocar dois canais separados de comunicação, um em cada direção.
- ✓ Fazendo-se isto, teremos dois canais físicos separados, um canal direto (*forward*) para os dados e um canal reverso (*backward*) para as confirmações.

Inatel
Instituto Nacional de Telecomunicações

Camada de Enlace
Protocolos de Retransmissão
Unidade 2

Piggybacking

- ✓ Entretanto, esta estratégia desperdiça muitos recursos, uma vez que o tráfego no canal é reverso é muito inferior ao tráfego no canal direto.
- ✓ Uma solução melhor consiste em utilizar o mesmo canal físico em ambas as direções.
- ✓ Neste caso, quando um *frame* chega, ao invés de se enviar imediatamente um *frame* de confirmação, o receptor aguarda até que a camada de rede deseje enviar outro pacote na direção reversa.

© Antônio M. Alberti 2007

Piggybacking

- ✓ A confirmação é feita, configurando-se, o cabeçalho deste *frame,* de forma a indicar que trata-se de um *frame* de *acknowledgment*.
- ✓ Esta técnica é conhecida por *piggybacking*.
- ✓ Para que ela funcione corretamente, é necessário ajustar o tempo que a camada de enlace pode esperar por uma pacote da camada de rede, antes de enviar a confirmação.
- ✓ Se o receptor esperar mais do que o *timeout* do relógio do transmissor, o *frame* será retransmitido desnecessariamente.

Inatel
Instituto Nacional de Telecomunicações

Camada de Enlace
Protocolos de Retransmissão
Unidade 2

Piggybacking

- ✓ Tipicamente, são utilizados *timeouts* da ordem de alguns milisegundos.
- ✓ Assim, se um novo pacote chegar dentro deste intervalo de tempo, a confirmação é enviada no *frame* que transportará este pacote na direção reversa, até o transmissor da direção direta.
- ✓ Caso contrário, a camada de enlace envia um *frame* avulso para realizar a confirmação.

© Antônio M. Alberti 2007

Protocolos de Janela Deslizante

- ✓ Um outro avanço no aumento da eficiência de transmissão são os chamados protocolos de Janela Deslizante (*Sliding Window*).

- ✓ Estes protocolos, ao invés de transmitirem um único *frame* e aguardarem pela sua confirmação para que um próximo *frame* possa ser transmitido, iniciam transmitindo vários *frames* e avançam a janela de transmissão à medida que confirmações vão sendo recebidas.

Protocolos de Janela Deslizante

- ✓ Ou seja, o transmissor mantém um conjunto de números de seqüência (igual ao tamanho da janela de transmissão) correspondente aos *frames* que ele está autorizado a transmitir.

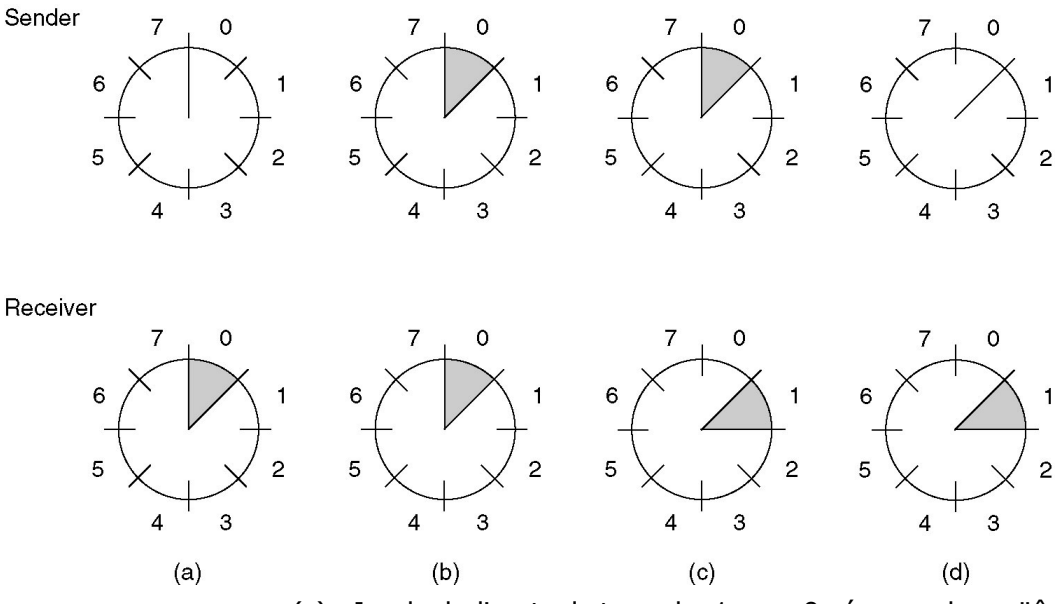
- ✓ Já o receptor, mantém um conjunto de números de seqüência, correspondente aos *frames* que estão autorizados a serem recebidos.

- ✓ Tipicamente, o tamanho da janela de transmissão é igual ao tamanho da janela de recepção, podendo permanecer fixo em um determinado valor ou variar com o tempo.

Protocolos de Janela Deslizante

- ✓ Sempre que um pacote chega da camada de rede para ser transmitido, o transmissor aloca para o *frame* que transportará este pacote o número de seqüência mais alto disponível.
- ✓ À medida que uma confirmação chega, a janela avança eliminado o número de seqüência mais baixo.
- ✓ A janela do transmissor mantém continuamente uma lista de *frames* transmitidos, mas não confirmados.

Protocolos de Janela Deslizante



Fonte: Tanenbaum

(a) Janela deslizante de tamanho 1, com 8 números de seqüência.
(b) Após o envio do primeiro *frame*.
(c) Após a recepção do primeiro *frame*.
(d) Após a primeira confirmação ter sido recebida.

Protocolos de Janela Deslizante

- ✓ O transmissor deve manter em um *buffer* os *frames* transmitidos, a fim de retransmiti-los, em caso de perda ou dano.
- ✓ Assim, a capacidade de armazenamento deve ser igual ao tamanho da janela utilizada.
- ✓ Qualquer *frame* recebido que esteja fora da janela, será descartado, sem qualquer comunicação ao transmissor.
- ✓ Caso contrário, o *frame* é passado para a camada de rede, uma confirmação é gerada, e a janela avança em uma unidade.

Inatel
Instituto Nacional de Telecomunicações

Camada de Enlace
Protocolos de Retransmissão

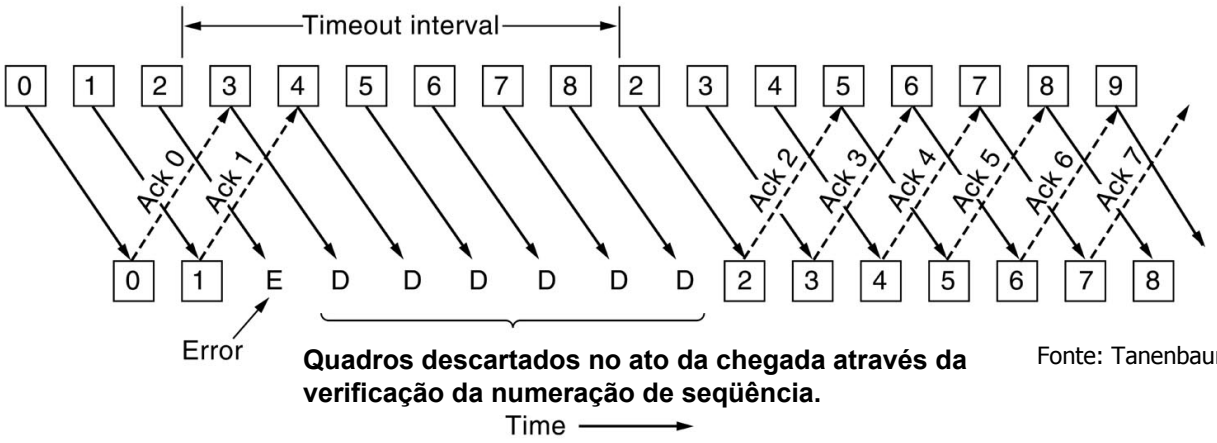
Protocolos de Janela Deslizante

- ✓ O envio de *frames* em paralelo através de um canal não confiável possui uma complicação adicional: o que acontece se um *frame* entre vários é danificado ou perdido? O que o receptor deve fazer com os demais *frames* que chegaram após o *frame* danificado?
- ✓ Existem duas soluções possíveis:
  - Retransmitir todos os *frames*: *Go-Back N*
  - Retransmitir somente o *frame* perdido: *Selective Repeat*

Unidade 2

© Antônio M. Alberti 2007

Protocolo Go-Back N



Fonte: Tanenbaum

- ✓ Os *frames* 0 e 1 são corretamente recebidos.
- ✓ O *frame* 2 é perdido.
- ✓ O transmissor, sem saber do ocorrido, continua a enviar *frames,* até que o relógio do *frame* 2 expire.

Inatel
Instituto Nacional de Telecomunicações

Camada de Enlace
Protocolos de Retransmissão

Protocolo Go-Back N

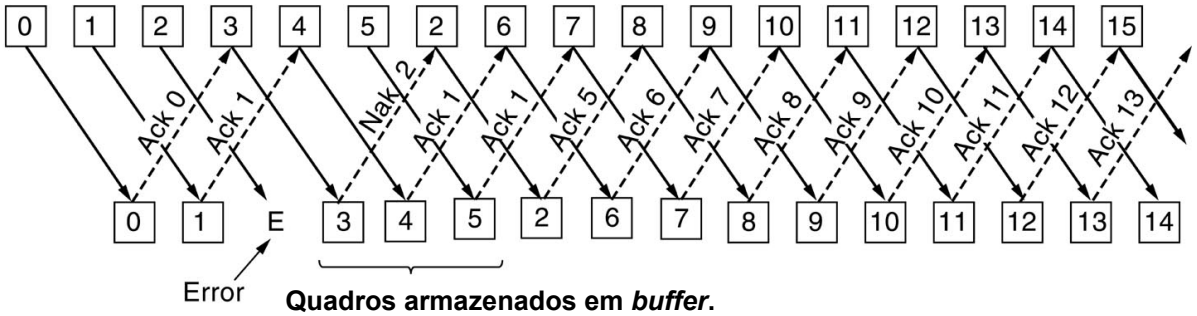
- ✓ Então, o transmissor retransmite o *frame* 2 e todos os demais *frames* na seqüência.
- ✓ Neste caso, todos os *frames* recebidos após o *frame* 2 ter sido perdido são descartados.
- ✓ Embora a janela de recepção seja maior que 1, na ocorrência de um erro, o receptor age como se tivesse uma janela de tamanho 1.

Unidade 2

© Antônio M. Alberti 2007

Protocolo Selective Repeat

Fonte: Tanenbaum



Quadros armazenados em *buffer*.
Nak 2 – Não recebi o quadro 2.
Ack 1 – Confirma somente a recepção do quadro 1.
Ack 5 – Confirma a recepção de todos os quadro até 5.

- ✓ Os *frames* 0 e 1 são corretamente recebidos.
- ✓ O *frame* 2 é perdido.
- ✓ Quando o *frame* 3 chega ao receptor, este percebe que está faltando o *frame* 2 e envia uma confirmação negativa para o transmissor.

Protocolo Selective Repeat

- ✓ O *frame* 3, ao invés de ser descartado ou enviado à camada de rede, é armazenado em um *buffer*.
- ✓ Quando os *frames* 4 e 5 chegam, eles também são armazenados.
- ✓ Finalmente, a confirmação negativa do *frame* (NAK – *Negative Acknowledgment*) 2 chega ao transmissor, que retransmite este *frame,* imediatamente.
- ✓ Quando o *frame* 2 retransmitido chega, todos os *frames* são entregues à camada de rede em ordem.

Protocolo Selective Repeat

- Se o NAK for perdido, o relógio do transmissor cairá a zero para o *frame* 2 e ele será retransmitido, mas com um atraso maior do que se o NAK tivesse sido recebido.
- Portanto, o uso de confirmações negativas acelera a retransmissão de *frames* em caso de falha.
- *Selective repeat* corresponde a um receptor com uma janela maior que 1.
- Esta solução requer o uso de *buffers* com grande capacidade de armazenamento na camada de enlace.

Comparação Go-Back N x Selective Repeat

- *Go-Back* N
  - Ignora toda a seqüência de *frames* a partir do errado.
  - Não confirma a recepção.
  - Aguarda a retransmissão de todos os quadros a partir do errado.
  - É um procedimento ruim para canais de comunicação com muito erro.
- *Selective Repeat*
  - Guarda os quadros da seqüência após o quadro errado.
  - Não confirma o quadro errado.
  - Aguarda a retransmissão do mesmo.
  - É um procedimento bem mais eficiente em termos de aproveitamento de banda, mas requer mais memória no nível de enlace do receptor.

HDLC – High-level Data Link Control

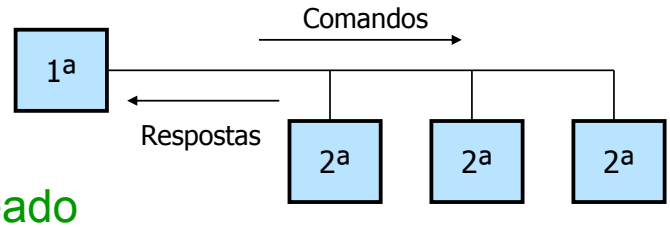
- ✓ O HDLC é um protocolo de transmissão de dados voltado para a camada de enlace do modelo OSI.
- ✓ O protocolo HDLC é um padrão da ISO (3309-1979) desenvolvido a partir do SDLC – *Synchronous Data Link Control* proposto pela IBM nos anos 70.
- ✓ As principais funções do protocolo HDLC são:
  1. Delineamento – Provê o delineamento de *frames* para a transmissão sobre um enlace de rede síncrona.
  2. Correção de Erros – Provê a detecção e correção de erros.
  3. Controle de Fluxo – Provê o controle de fluxo dos *frames*.

HDLC - Tipos de estações

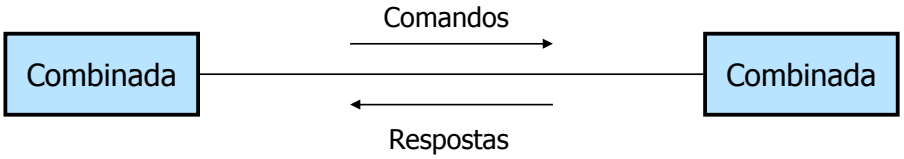
- ✓ Estação Primária (Mestre)
  - Controla todas as outras estações do enlace.
  - Envia quadros de comando.
  - Mantém um enlace lógico separado para cada estação secundária.
- ✓ Estação Secundária (Escrava)
  - Sob controle da estação primária.
  - Envia quadros de resposta.
- ✓ Estação Combinada
  - Pode enviar comandos e respostas.

HDLC - Configurações do enlace

- ✓ Não-Balanceado
  - Uma estação primária e uma ou mais estações secundárias. Suporta operação *full duplex* e *half duplex*.

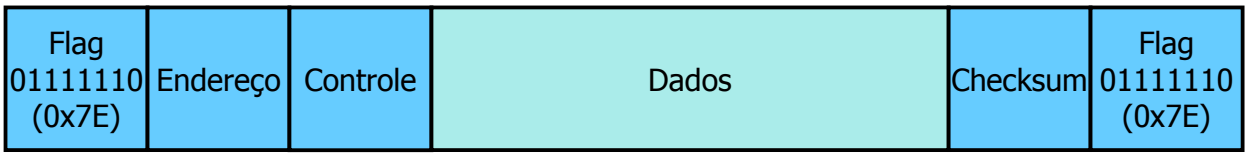


- ✓ Balanceado
  - Duas estações combinadas. Também suporta operação *full duplex* e *half duplex*.



HDLC – High-level Data Link Control

- ✓ O *frame* HDLC possui os seguintes campos:



  - Flag
    - São dois *flags* de delineamento 01111110 (0x7E), um no inicio e outro no final do *frame* HDLC (***Flags* de Inicio e Fim com Enchimento de *Bits***).
    - Pode encerrar um quadro e iniciar outro.
    - Receptor busca pela seqüência de *flag* para iniciar o sincronismo.
    - *Bit stuffing* usado para evitar confusão com dados contendo a seqüência 01111110.
    - 0 inserido após seqüência de cinco 1s.

HDLC – High-level Data Link Control

- Endereço
  - Identifica a estação secundária que está enviando ou que irá receber um determinado quadro.
  - Usualmente tem 8 *bits*.
  - LSB de cada octeto indica se o octeto é o último (1) ou não (0).
  - Todos os *bits* iguais a 1 indicam *broadcast.*
- Controle
  - O campo de controle é usado para **numeração de seqüência** com janela deslizante de 3 *bits*, **confirmações** e outros propósitos.
- *Checksum*
  - Este campo é utilizado para detecção/correção de erros sobre parte do *frame* HDLC.

PPP – Point-to-Point Protocol

- ✓ O PPP é um protocolo desenvolvido para conectar computadores a Internet através de enlaces ponto a ponto.
- ✓ É descrito na RFC 1661 do IETF e consistente com as Recomendações X.25 do ITU-T.
- ✓ O PPP foi projetado para o transporte simplificado de pacotes através de um enlace full-duplex com operação bidirecional que assegura que os pacotes serão entregues no destino em ordem.

PPP – Point-to-Point Protocol

- ✓ As principais funções do protocolo PPP são:
  1. Delineamento – Fornece um método de enquadramento que delimita o início/fim de cada quadro com um padrão de *bits*.
  2. Controle de Enlace – Provê um controle de enlace para ativar a linha de comunicação, testá-la, negociar opções e desativá-la quando não mais necessária. Isso é feito pelo sub-protocolo LCP (*Link Control Protocol*).
  3. Controle de Rede – Provê um mecanismo de negociação de opções de rede, de modo independente do protocolo de camada de rede adotado. Isso é implementado por um NCP (*Network Control Protocol*) para cada camada de rede suportada.

Inatel
Instituto Nacional de Telecomunicações

Camada de Enlace
Unidade 2

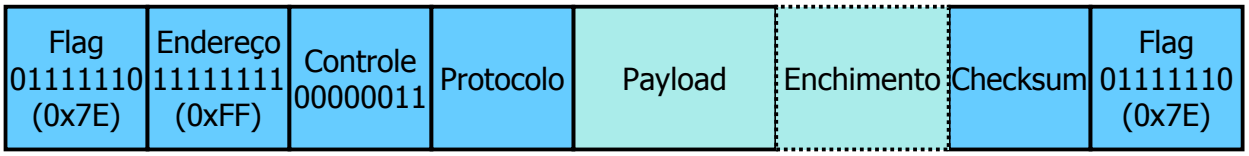
PPP – Point-to-Point Protocol

- ✓ Cont.:
  4. Encapsulamento – Permite multiplexar diferentes protocolos de camada de rede simultaneamente em um mesmo enlace.

© Antônio M. Alberti 2007

PPP – Point-to-Point Protocol

- ✓ O *frame* PPP tem um formato parecido com o formato do *frame* HDLC.
- ✓ A maior diferença é que o PPP é orientado a caracteres (enchimento de *bytes*), enquanto o HDLC é orientado a *bits* (enchimento de *bits*).
- ✓ O *frame* PPP possui os seguintes campos:



PPP – Point-to-Point Protocol

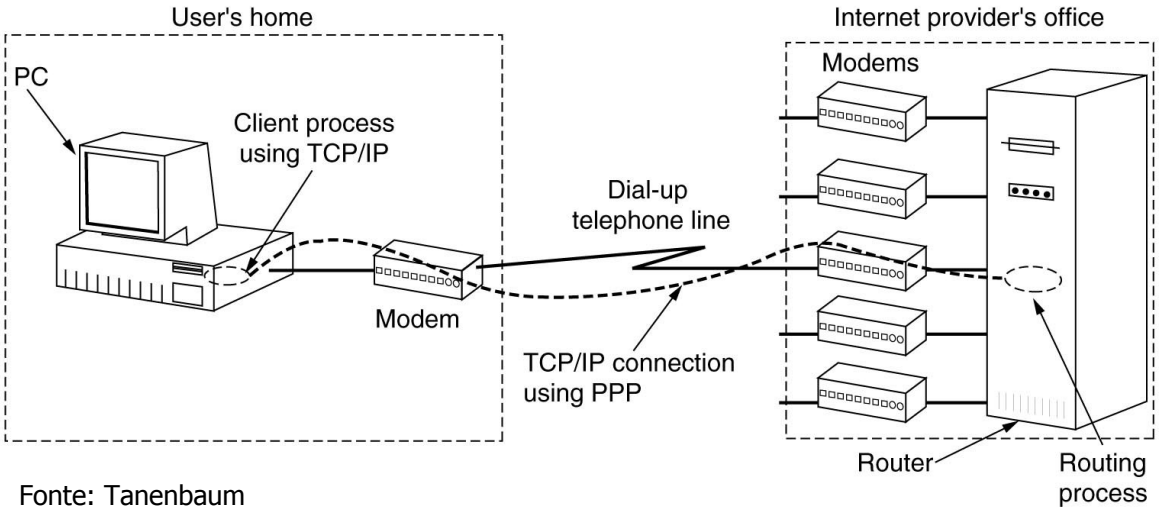
- *Flag* – Todos os *frames* PPP iniciam com o mesmo *byte* de *flag* utilizado no HDLC.
- Endereço – É sempre configurado para 0xFF para indicar que todas as estações podem aceitar o *frame*.
- Controle – O valor *default* é 00000011. Este valor indica um *frame* não numerado. Ou seja, o PPP por default não provê a transmissão confiável usando números de seqüência e confirmações. Os detalhes sobre a transmissão confiável utilizando PPP são discutidos na RFC 1663. Na prática, é pouco usada.

PPP – Point-to-Point Protocol

- Protocolo – Identifica a qual protocolo pertence a informação armazenada no *payload*. Foram definidos códigos para os protocolos LCP, NCP, IP, IPX, *AppleTalk*, etc.
- Payload – O tamanho máximo do campo de informações, excluindo o campo de enchimento, é determinado pelo MRU – *Maximum Receive Unit*, o qual possui o valor *default* de 1500 *bytes*. Outros valores podem ser negociados.
- Enchimento – Na transmissão o campo de informações pode ser preenchido por enchimento até que o valor do MRU seja atingido. É de responsabilidade de cada protocolo de rede distinguir entre informação e enchimento.

PPP – Point-to-Point Protocol

✓ Como o PPP permite que um PC atue com uma estação Internet utilizando uma linha telefônica e um modem?



Fonte: Tanenbaum