



CAMADA DE ENLACE

A camada de enlace

Serviços da camada de Enlace

A camada de enlace está situada imediatamente acima da camada física. Elas atuam juntas de forma direta.

Conforme estudado, apresentamos diversos fatores que podem afetar a transmissão do sinal no meio físico. Alguns desses fatores representam características do meio, não sendo possível eliminá-los (como o ruído térmico).

Os erros na recepção dos sinais são previstos e a camada física por si só não pode recuperá-los, cabendo à camada de enlace controlá-los.

Para que o serviço de transmissão de bits pelo meio físico seja confiável, a camada de enlace realiza as seguintes funções:

Delimitação de quadros

Controle de erros

Controle de fluxo

Controle de acesso ao meio

Observação importante é que o modelo OSI considera que todas as funções acima são desempenhadas pela camada de enlace, mas para facilitar o entendimento, iremos adotar a estrutura definida na arquitetura IEEE 802 que considera a camada de enlace dividida em duas subcamadas: LLC (Controle de Enlace Lógico) e MAC (Controle de Acesso ao Meio).

Delimitação de quadros (enquadramento)

Para o melhor desempenho de suas funções, **a camada de enlace utiliza o quadro como unidade de dados.**

O quadro é um conjunto de bytes de informação que pode variar de tamanho conforme o protocolo a ser utilizado.

Suponhamos que determinado transmissor tenha uma quantidade muito grande de dados para transmitir ao receptor. Ao final dessa transmissão, percebe-se que, em algum momento, houve um erro no sinal recebido por conta dos problemas do canal.

Dessa forma, o transmissor precisaria repetir toda a transmissão para garantir a informação correta ao receptor. No entanto, se dividíssemos essa grande quantidade de dados em conjuntos menores (quadros) e transmitíssemos

quadro após quadro, havendo um erro na transmissão, seria possível identificar qual quadro foi afetado.

Com isso, só repetiríamos a transmissão desse quadro, tornando o controle de erros muito mais eficiente.

Entendendo a importância da utilização de quadros no nível de enlace, verificaremos como criar os quadros, ou seja, como particionar os dados delimitando o início e o fim de cada quadro.

Existem basicamente quatro técnicas para realizar o enquadramento dos dados e, em alguns casos, as técnicas são combinadas. Vejamos!

Contagem de caractere

Enquadramento por caractere

Enquadramento por bit

Violação de códigos do nível físico

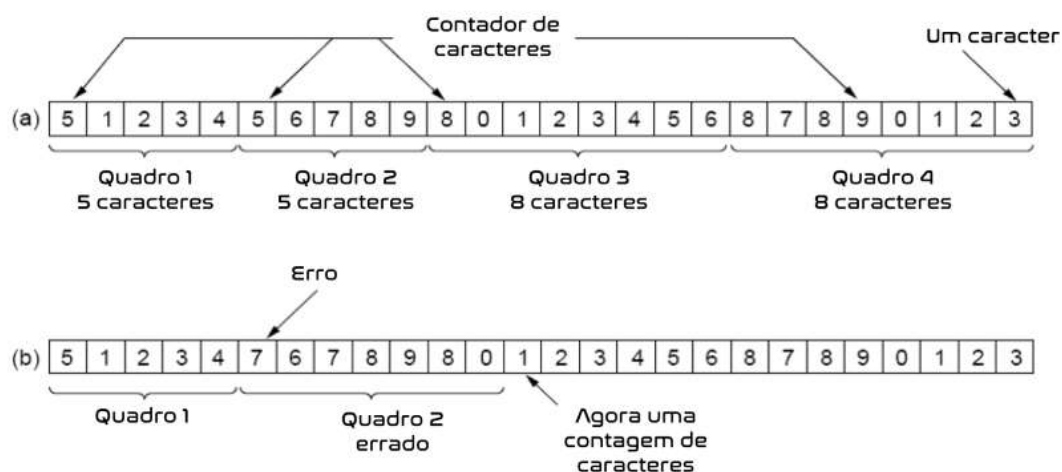
Enquadramento por caractere

Na técnica de contagem de caractere, a ideia é **adicionar um campo no início do quadro, informando o número total de caracteres presentes**.

Ao receber o quadro, o receptor (RX) lê o campo de contagem e, a partir de então, consegue determinar onde está o final do quadro.

O problema dessa técnica simples é que, se houver um erro justamente nesse campo de contagem, o transmissor (TX) e o receptor (RX) terão interpretações diferentes sobre os quadros e perderão completamente o sincronismo.

Veja o exemplo a seguir:

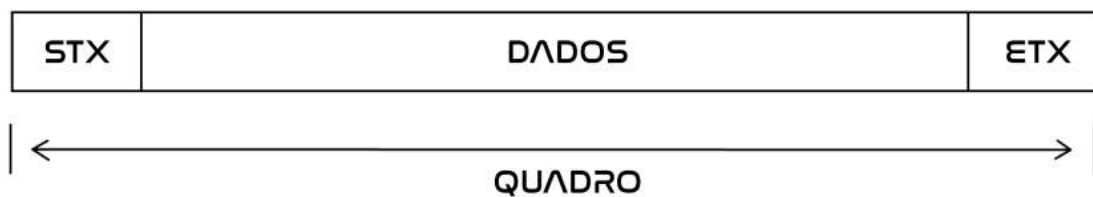


Problema de sincronização entre o TX e o RX quando há um erro no campo de contagem.

Devido ao problema, ilustrado na imagem acima, a técnica de contagem de caractere só é utilizada em conjunto com outras técnicas.

Enquadramento por caractere

Na técnica de enquadramento por caractere, a ideia é utilizar caracteres especiais para indicar o início (STX) e o fim do quadro (ETX). O transmissor insere essas marcas. Com isso, o receptor fica sabendo exatamente onde começa e termina cada quadro.



Enquadramento por caractere.

Além disso, outros campos são incluídos no quadro, como os campos de sincronização (SYN), cabeçalho (HEADER) e códigos para verificação de erros (CRC). A imagem abaixo ilustra o quadro com todos esses campos.



Enquadramento por caractere.

Uma dificuldade que pode ocorrer com essa técnica é que o campo de dados representa as informações do usuário, e a camada de enlace não tem controle sobre elas. Assim, pode estar presente no campo de dados o padrão idêntico ao do caractere ETX. Ao receber o quadro e percorrê-lo, o receptor interpretaria esse ETX como fim do quadro, o que seria um erro de interpretação.

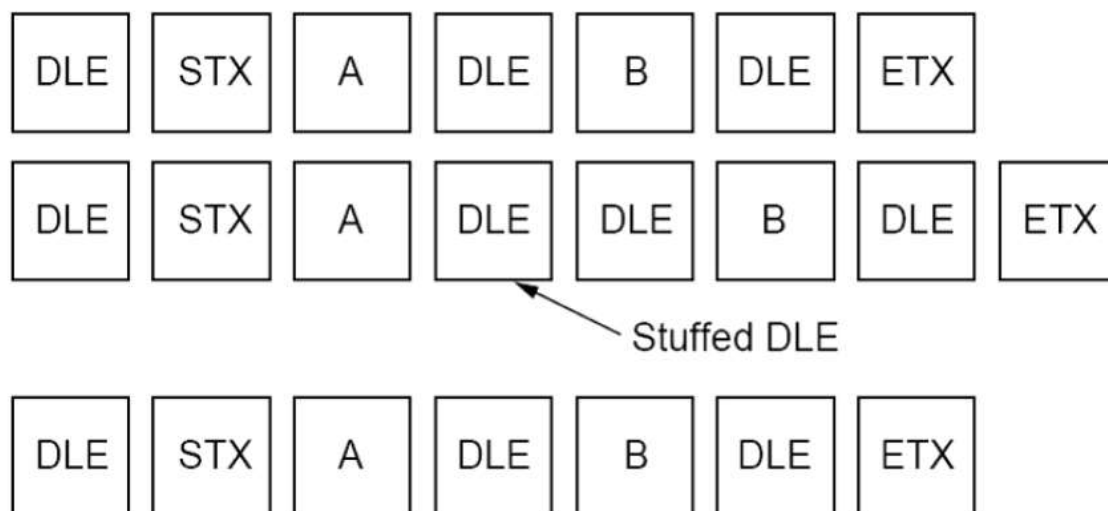
Qual é a solução para esse problema?

A solução é, em primeiro lugar, a utilização de outro caractere especial (DLE) para indicar que, imediatamente após esse caractere, aparecerá o caractere delimitador (STX ou ETX). Dessa forma, um início de quadro seria marcado

como DLE STX e um fim de quadro por DLE ETX. Isso ainda não resolve o problema, pois o DLE também pode estar presente no campo de DADOS do usuário.

A solução completa é implementada da seguinte forma: ao gerar o quadro, o transmissor percorre o campo de DADOS do usuário em busca do padrão DLE. Encontrando esse padrão, ele insere no campo de dados outro DLE (operação conhecida como *character stuffing*), e segue normalmente com a construção do quadro.

Ao receber o quadro, o receptor analisa os caracteres do quadro e, ao se deparar com um DLE, ele verifica qual é o próximo caractere. Caso seja outro DLE, ele sabe que este foi inserido pelo transmissor. Assim, ele exclui esse DLE e continua varrendo o quadro. Caso o caractere seguinte não seja outro DLE, mas um marcador (por exemplo, ETX), ele sabe que a marcação está correta. A imagem abaixo exemplifica esse processo de *character stuffing*.



Enquadramento por caractere, técnica conhecida como *character stuffing*.

Uma desvantagem da técnica de enquadramento por caractere é ser orientada completamente pela existência e pelo reconhecimento de caracteres. Uma alternativa similar a essa é o enquadramento por bit, onde não há a necessidade de se trabalhar com caracteres, mas, sim, com bits de dados.

Enquadramento por bit

Nessa técnica, o delimitador de quadros é o *flag* – sequência padrão de bits, geralmente 01111110. Cada quadro começa e termina com uma marca *flag*. Havendo falta de sincronismo por algum motivo, tudo

o que o receptor tem a fazer é procurar por um *flag* para ficar sincronizado com o transmissor novamente.

De forma análoga à técnica anterior, aqui também ocorre o *stuffing*. O transmissor percorre o campo de DADOS todo e, ao perceber uma sequência de 5 bits “1”, ele insere um bit “0”, para quebrar o padrão de *flag*. Ao percorrer o quadro e identificar 5 bits “1” seguidos, o receptor fica alerta; se o próximo bit for “0”, ele sabe que esse bit foi inserido pelo transmissor, caso contrário (o próximo bit for “1”) ele sabe que se trata de um delimitador de quadro, *flag*.

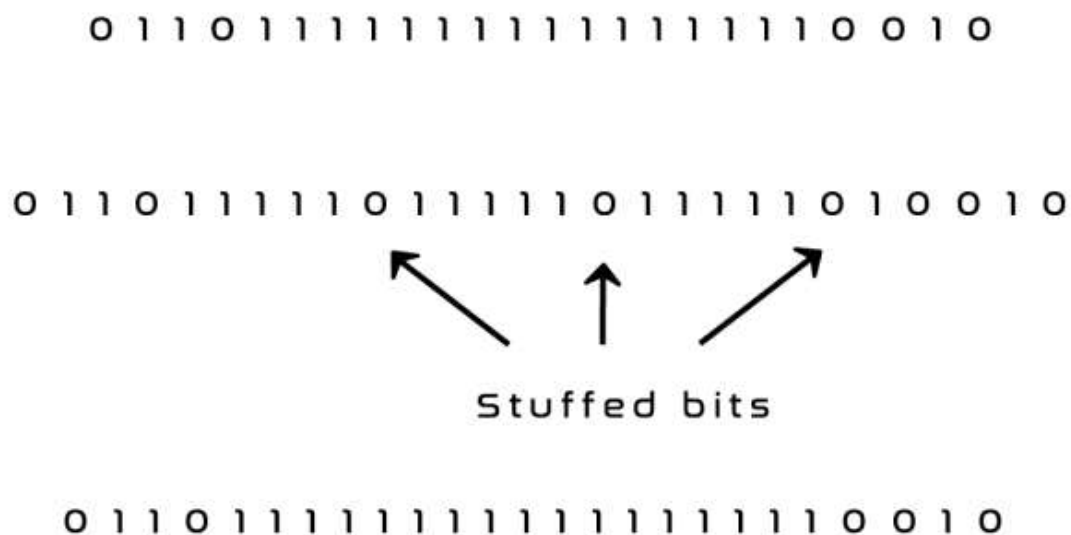


Ilustração da técnica bit *stuffing*.

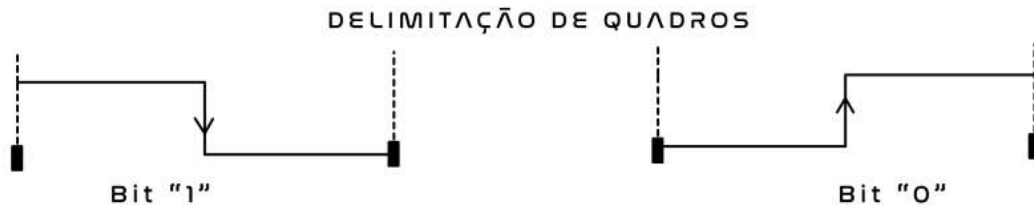
Violação de códigos do nível físico

A última técnica de enquadramento estudada é conhecida como violação de códigos do nível físico. A ideia é bastante simples: na transmissão do sinal no meio físico, o bit “1” é representado por alguma característica ou variação do sinal, e o bit “0”, por outra.

Se o sinal puder apresentar ainda outras variações que não sejam utilizadas para codificar os bits, essas variações podem ser utilizadas para marcar o início e o fim do quadro, tendo em vista que não serão confundidas com os bits propriamente ditos.

Um exemplo é a codificação Manchester padronizada pelo IEEE para redes locais. Nesta codificação, o bit “1” é representado por uma transição do sinal de alto para baixo, e o bit “0”, pela transição contrária do sinal de baixo para alto.

Assim, as outras duas transições (ou ausência de transições), de baixo para baixo e de alto para alto, estão livres para serem usadas como marcadores de quadro.



Codificação Manchester. Transições alto-alto e baixo-baixo não são usadas.

Controle de erros (codificação)

Existem duas estratégias básicas para o controle de erro no nível de enlace: *open loop* (malha aberta, sem canal de retorno) e *feedback* (malha fechada, com canal de retorno).

Open loop

Na estratégia de *open loop*, a **detecção** e **correção** de erros são feitas completamente pelo receptor. São empregados códigos especiais (FEC: *Forward Error Correction*) para inserir informação redundante no quadro. Tudo isso para que, ao receber um quadro, o receptor:

- Possa usar a codificação para verificar a integridade do quadro;
- Havendo algum problema, possa, por si só, alterar o quadro para a forma correta.

A desvantagem dessa técnica é a necessidade de se inserir grande quantidade de informação redundante no quadro, de forma que o receptor possa executar as duas tarefas listadas acima sozinho. No entanto, pode ser a única solução, caso a transmissão não tenha canal de retorno.

Exemplo

O código hamming é o exemplo mais simples de código de correção de erros.

Feedback

A estratégia *feedback* emprega apenas códigos de detecção de erros, isto é, insere informação redundante que seja suficiente apenas para o receptor testar a integridade do quadro. Havendo um problema, o receptor solicita ao transmissor que retransmita aquele quadro. Assim, é necessário haver um

canal de retorno do receptor ao transmissor, situação comum em redes de dados, pois as estações geralmente precisam transmitir e receber dados.

Um exemplo simples de código de detecção de erros é o bit de paridade, que é inserido ao final do quadro. Assumindo a escolha da paridade par, o transmissor, ao transmitir o quadro, verifica a quantidade de bits “1” presentes nele. Se houver um número par de números 1, a paridade estará correta, e o bit de paridade receberá o valor “0”. Caso haja um número ímpar de bits “1” a ser transmitido no quadro, o transmissor fechará a paridade par inserindo o bit “1” no campo do bit de paridade. Ao receber o quadro, o receptor deve checar a paridade; se não for par, certamente houve algum problema com a recepção do quadro, e ele deve solicitar uma nova transmissão deste mesmo quadro.



O bit de paridade, apesar de ser simples e de fácil implementação, não é eficaz em muitos casos, como, por exemplo, na situação em que houve um problema no sinal, fazendo com que o receptor interprete erradamente dois bits do quadro. Assim, quando o receptor fizer o teste, a paridade estará correta, e o receptor não perceberá o erro na recepção do quadro.

Dessa forma, outros códigos de detecção mais poderosos foram desenvolvidos e padronizados para uso em redes de computadores, como, por exemplo, o CRC (Verificação Cíclica de Redundância), conforme ilustrado no quadro sobre técnica de enquadramento por caractere. (STALLINGS, 2004)



Enquadramento por caractere.

Depois de termos estudado as técnicas de enquadramento e codificação, entendemos como os quadros podem ser formados e como o receptor é capaz de verificar se houve ou não erro na recepção do quadro. Falta agora estudarmos como as retransmissões são realizadas automaticamente por meio de protocolos de enlace.

Controle de fluxo

Outra operação que pode ser implementada aproveitando-se dos protocolos ARQ é o **controle de fluxo**.

O objetivo desse controle é evitar que um transmissor mais rápido acabe sobrecarregando um receptor mais lento com o envio de quadros a uma velocidade mais rápida que o receptor é capaz de suportar, causando um “afogamento” no receptor.

Esse descompasso não é desejável, pois o receptor vai acabar descartando os quadros novos e o transmissor teria que retransmiti-los em um outro momento.

Uma forma do receptor dosar a velocidade de transmissão de quadros quando estiver empregando os protocolos da família ARQ é simplesmente retardando o envio dos ACKs.

Dessa forma, o receptor consegue reduzir a velocidade com que novos quadros são inseridos no canal por parte do transmissor. O receptor pode inclusive reter todos os ACKs em determinado momento o que causaria timeouts no transmissor e a pausa na transmissão de novos quadros.

Assim que o receptor estivesse pronto, bastaria enviar os ACKs para que a comunicação fosse reestabelecida e seguisse com os novos quadros.