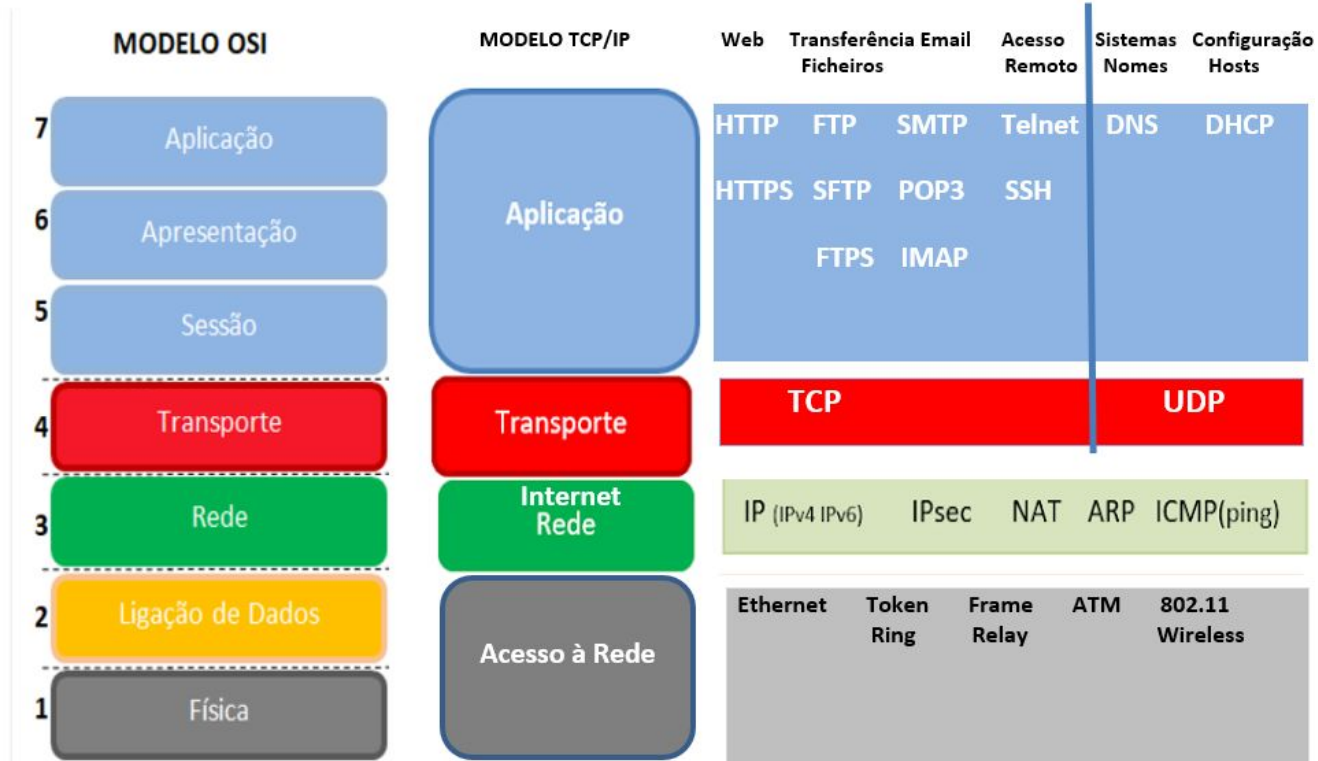


30-764

Redes de Computadores I

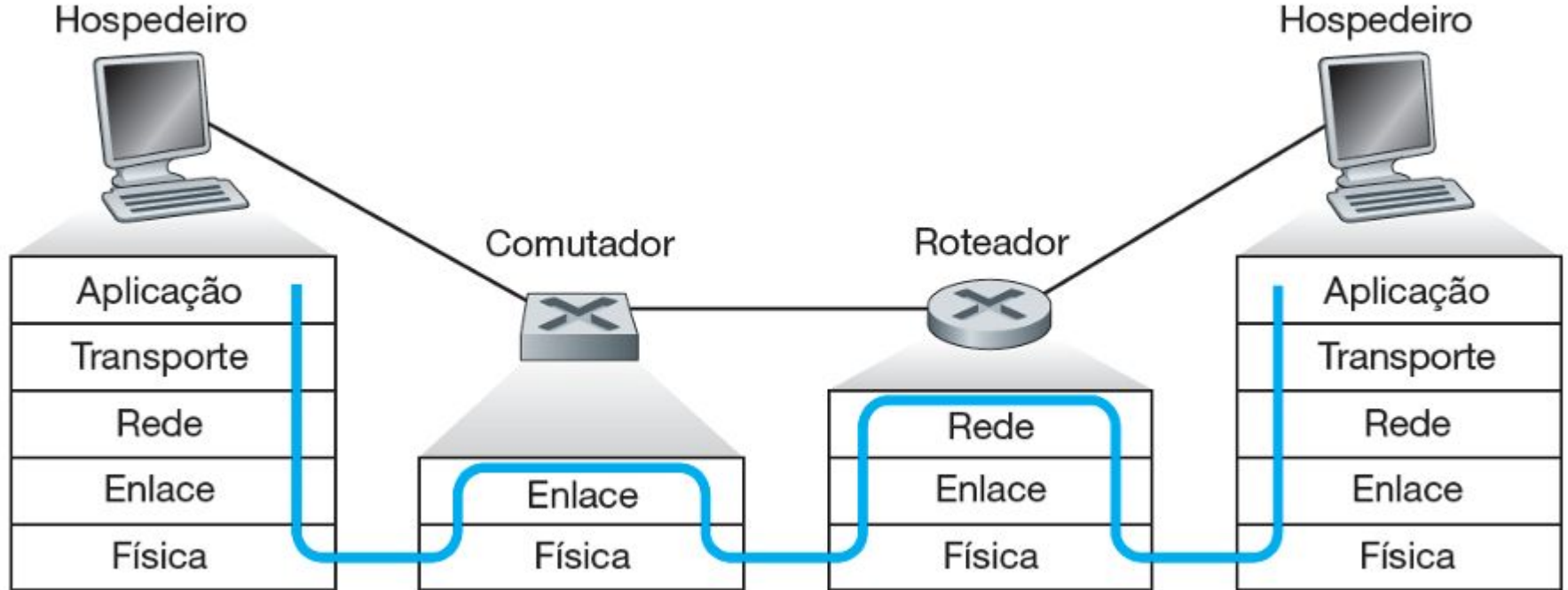
MSc. Fernando Schubert

CAMADA DE ENLACE



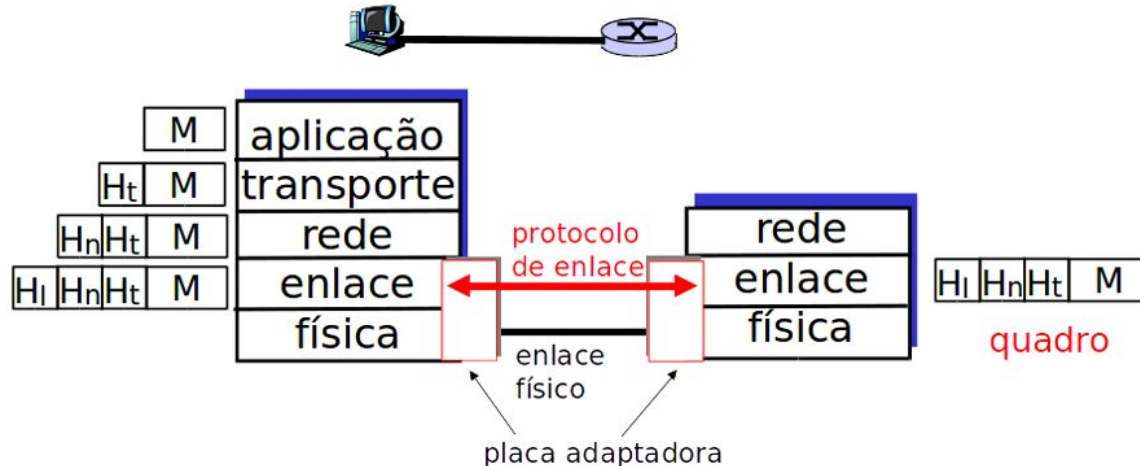
DEFININDO O CONTEXTO

- Algumas terminologias:
 - Hospedeiros e roteadores são nós
 - Canais de comunicação que conectam nós adjacentes ao longo do caminho de comunicação são enlaces
 - Enlaces com fio
 - Enlaces sem fio
 - LANs
 - Pacote de camada-2 é um quadro, encapsula o datagrama
 - Camada de enlace: camada de enlace tem a responsabilidade de transferir um datagrama de um nó para o nó adjacente sobre um enlace.

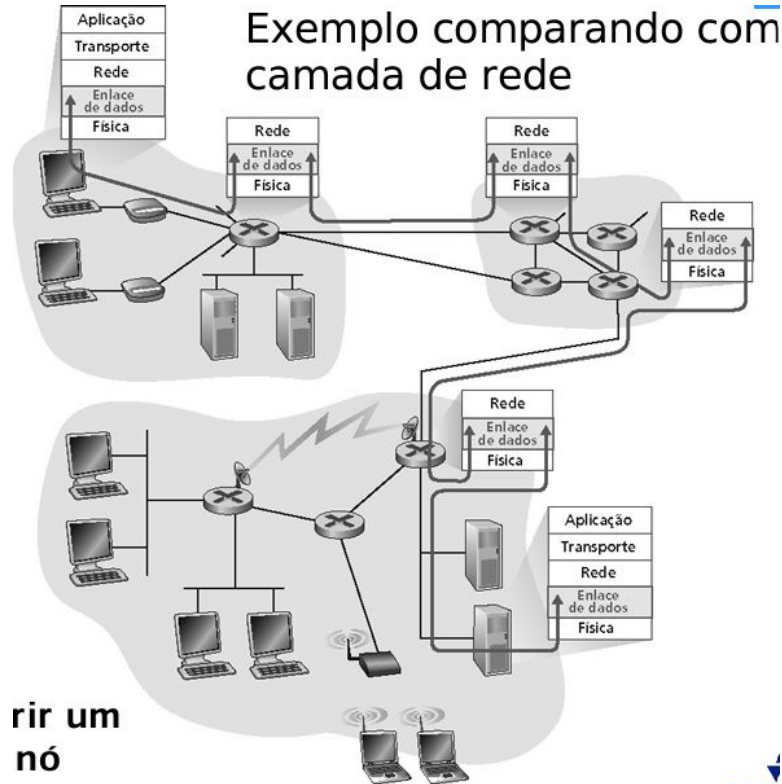


DEFININDO O CONTEXTO

- Dois elementos fisicamente conectados podem incluir:
 - Host-roteador
 - Roteador-roteador
 - Host-host
- A unidade de dados utilizada nesse contexto é chamada de quadro (frame).



DEFININDO O CONTEXTO

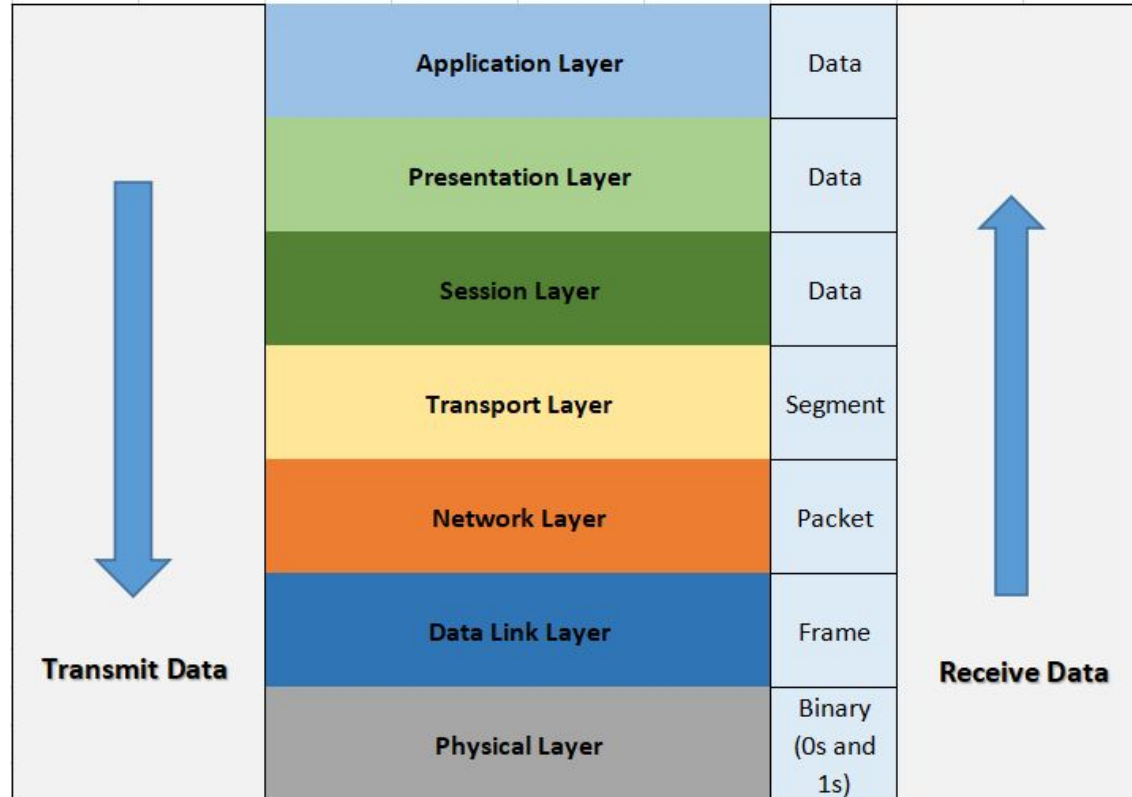


DEFININDO O CONTEXTO

A Camada Física desempenha um papel-chave na comunicação entre computadores, mas seus esforços, sozinhos, não são suficientes. Cada uma de suas funções tem suas limitações. A Camada de Enlace trata dessas limitações.

Camada 1	Camada 2
não pode se comunicar com as camadas de nível superior	Logical Link Control (LLC)
não pode nomear ou identificar computadores	usa um processo de endereçamento
descreve apenas os fluxos de bits	usa o enquadramento para organizar ou agrupar os bits
não pode decidir que computador irá transmitir os dados binários de um grupo onde todos tentam transmitir ao mesmo tempo	Media Access Control (MAC)

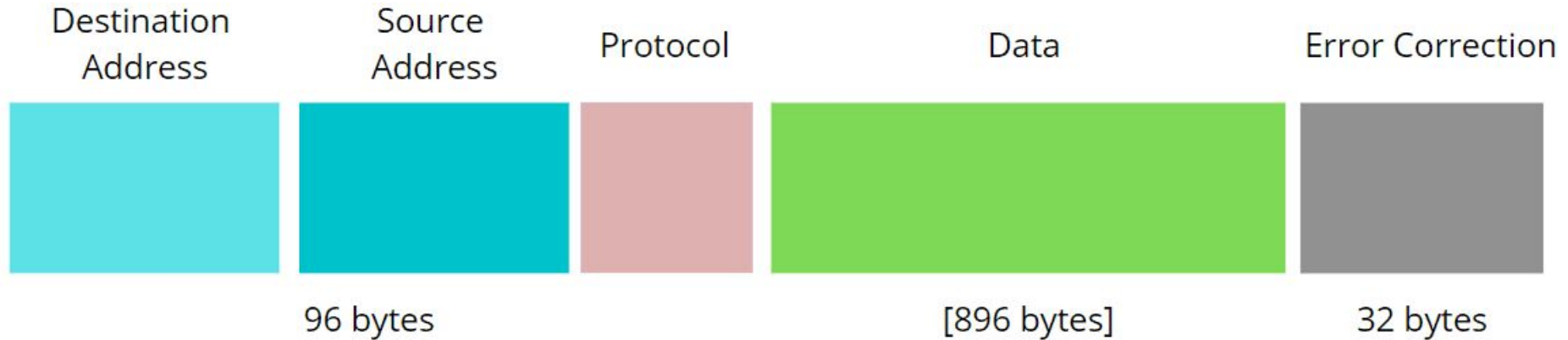
UNIDADES DE TRANSPORTE EM REDES



UNIDADES DE TRANSPORTE EM REDES - PACOTES

- Ao comunicar através de redes, é importante enviar e receber arquivos e informações. A unidade básica de comunicação entre uma origem e um destino em uma rede é um pacote.
- Os dados enviados pela rede são divididos em pacotes, que são recombinaados pelos dispositivos de destino. Dividir os dados em pacotes permite que a rede gerencie diferentes larguras de banda, rotas e múltiplos dispositivos conectados que compartilham dados e recebem pacotes independentemente uns dos outros.
- Isso facilita a retransmissão de pedaços perdidos de dados ou interrompidos. Os pacotes são unidades de dados dentro da camada de rede no modelo OSI.
- Cada pacote contém um cabeçalho com os endereços IP de origem e destino, um campo para a especificação do protocolo, os dados, um trailer, a versão do protocolo, etc. O campo do trailer contém informações sobre correções de erro e outras bandeiras para a identificação.

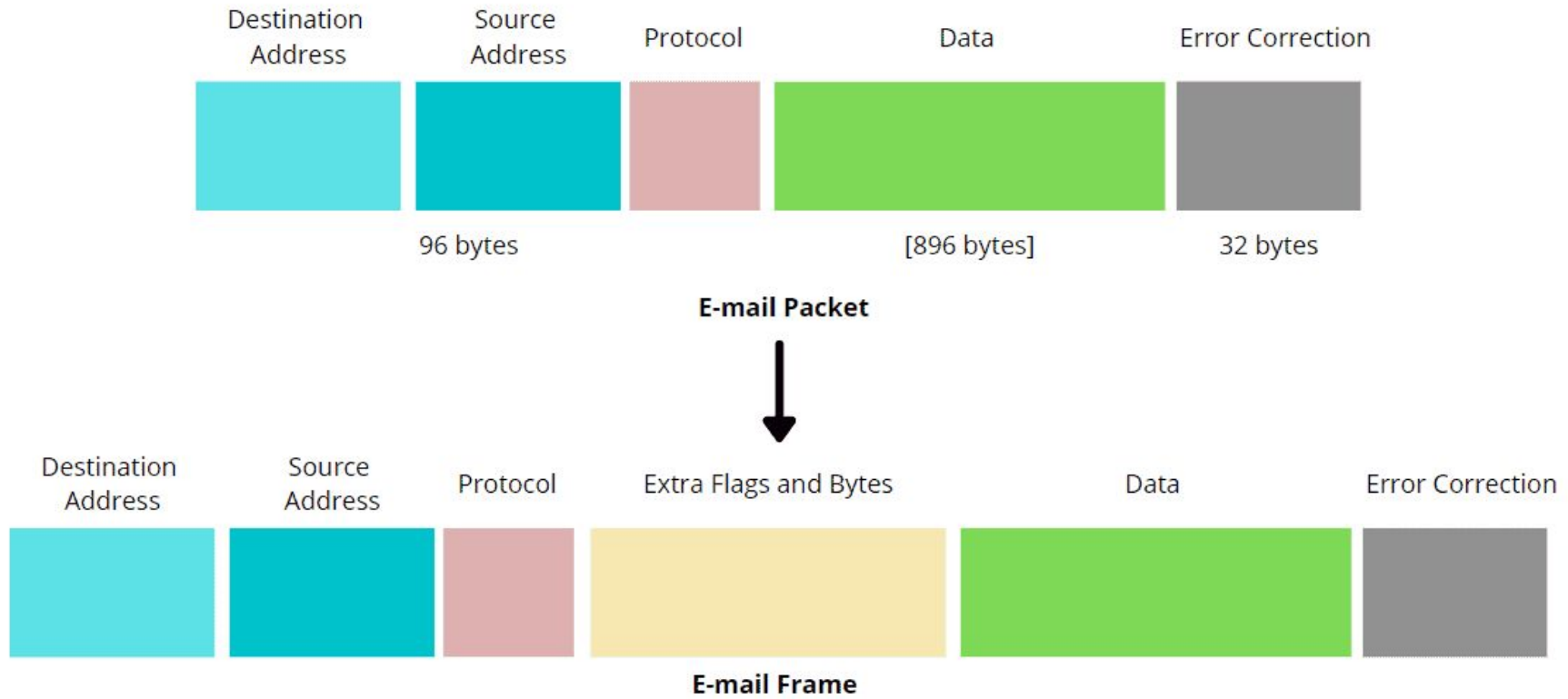
UNIDADES DE TRANSPORTE EM REDES - PACOTES



UNIDADES DE TRANSPORTE EM REDES - QUADROS (FRAMES)

- Assim como os pacotes, os quadros são partes pequenas de uma mensagem na rede. Eles ajudam a identificar dados e determinar a forma como devem ser decodificados e interpretados. A principal diferença entre um pacote e um quadro é a associação com as camadas OSI.
- Enquanto um pacote é a unidade de dados usada na camada de rede, um quadro é a unidade de dados usada na camada de enlace de dados do modelo OSI. Um quadro contém mais informações sobre a mensagem transmitida do que um pacote.
- Na rede, existem dois tipos de quadros: quadros de comprimento fixo e quadros de comprimento variável. No enquadramento de comprimento fixo, o tamanho do quadro funciona como delimitador. Portanto, não é necessário definir um limite para o quadro. No enquadramento de comprimento variável, é difícil determinar o início e o fim de um quadro. Portanto, é essencial definir o início e o fim de um quadro.

UNIDADES DE TRANSPORTE EM REDES - QUADROS (FRAMES)



UNIDADES DE TRANSPORTE EM REDES - DATAGRAMAS

- O datagrama representa uma unidade de transferência de dados em redes. Os dados transmitidos em uma rede são divididos em partes menores chamadas datagramas. Em um datagrama, dividimos os dados frequentemente e os transmitimos de origem para destino sem uma rota predefinida. Também não podemos garantir a ordem de entrega no destino.
- Enquanto o TCP usa pacotes em protocolos orientados à conexão, datagramas são usados no UDP, o que os torna carregarem menos informações, já que não precisam ter uma mensagem de resposta do destino. A camada de transporte usa datagrama como unidade de transferência de dados. Um datagrama é composto por um cabeçalho, os endereços IP de destino e origem, e os dados.
- O problema com datagramas é que eles não podem gerenciar comunicações subsequentes ou anteriores de dados. Além disso, para uma única transferência, um datagrama pode acomodar até 65535 bytes de dados, o que é muito pequeno.
- No caso de enviarmos um e-mail usando o protocolo UDP, não haverá pacotes, mas datagramas. A informação transmitida seria conforme a seguinte figura:

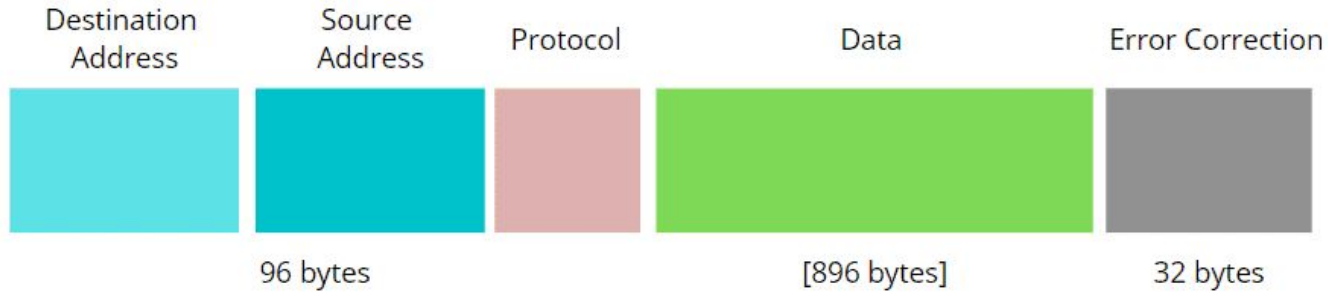
UNIDADES DE TRANSPORTE EM REDES - DATAGRAMAS



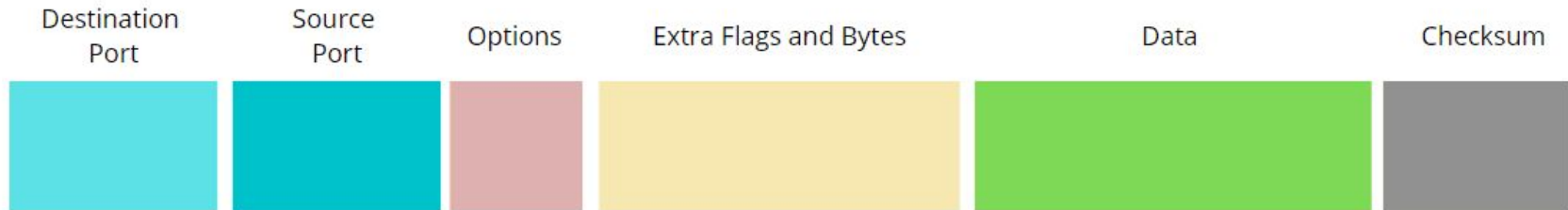
UNIDADES DE TRANSPORTE EM REDES - SEGMENTOS (Segment)

- Um segmento é um pedaço quebrado de um pacote com um cabeçalho TCP em cada um deles. Junto com as portas de origem e destino, ele contém o campo de checksum que garante a correção dos dados transmitidos pela rede. Os segmentos aumentam a eficiência do desempenho da rede e melhoram a segurança.
- Se enviássemos um e-mail com uma conexão TCP, então os dados que existem na camada de transporte para concluir a operação são um exemplo de um segmento. O pacote de e-mail então se tornaria:

UNIDADES DE TRANSPORTE EM REDES - SEGMENTOS (Segment)



E-mail Packet



CAMADA DE ENLACE - CONTEXTO

- Datagrama transferido por protocolos de enlace diferentes sobre enlaces diferentes:
 - ex.: Ethernet no primeiro enlace, frame relay nos enlaces intermediários, 802.11 no último enlace.
- Cada protocolo de enlace provê serviços diferentes
 - ex.: pode ou não prover transferência confiável sobre o enlace
- Analogia do transporte
 - Viagem de Princeton até Lausanne
 - Carro: Princeton até JFK
 - Avião: JFK até Geneva
 - Trem: Geneva até Lausanne
 - Turista = datagrama
 - Segmento de transporte = enlace de comunicação
 - Modo de transporte = protocolo da camada de enlace
 - Agente de viagem = algoritmo de roteamento

CAMADA DE ENLACE - SERVIÇOS

- Enquadramento:
 - Encapsula datagramas em quadros acrescentando cabeçalhos e trailer
- Acesso ao enlace:
 - Um único remetente e um destinatário (ponto- a- ponto)
 - Pode-se enviar sempre que enlace ocioso
 - Caso mais interessante - > enlace compartilhado
 - Conhecido como enlace de broadcast
- Entrega confiável entre dois equipamentos fisicamente conectados:
 - camada de enlace pode garantir que um datagrama seja transportado sem erros - > semelhante ao serviço oferecido pela camada de transporte •
 - Usado para enlaces com altas taxas de erros -> wireless
 - Raramente usado em enlaces com baixa taxa de erro (fibra, alguns tipos de par de fios trançados de cobre)

CAMADA DE ENLACE - SERVIÇOS

- Controle de fluxo:
 - Os adaptadores têm capacidade limitada para armazenar quadros-
 - Buffer do receptor pode transbordar
 - Limitação da transmissão entre transmissor e receptor
- Detecção de erros:
 - Erros causados pela atenuação do sinal e por ruídos
 - Quem envia pacote deve adicionar bits para a detecção
 - Serviço geralmente implementado em hardware
 - O receptor detecta a presença de erros:
 - Avisa o transmissor para reenviar o quadro perdido
- Correção de erros:
 - O receptor identifica e corrige o bit com erro(s) sem recorrer à retransmissão
- Half- duplex e full- duplex
 - Com half-duplex, os nós em ambas as extremidades do enlace podem transmitir, mas não ao mesmo tempo

COMUNICAÇÃO DE ADAPTADORES

- Camada de enlace implementada no “adaptador” (isto é, NIC)
 - Cartão Ethernet, cartão PCMCIA, cartão 802.11
- Lado transmissor:
 - Encapsula o datagrama em um quadro
 - Adiciona bits de verificação de erro, pode existir um mecanismo de entrega confiável (num seqs, temporizadores, acks), controle de fluxo etc.
- Lado receptor
 - Procura erros, entrega confiável, controle de fluxo etc
 - Extrai o datagrama, passa para o lado receptor
- Adaptador toma decisão se uma quadro está OK e encaminha para remetente - > camada de rede

TIPOS DE SERVIÇOS

- Os serviços fornecidos nas seguintes combinações são:
 - Serviços em conexão não confirmada;
 - Serviços em conexão confirmada;
 - Serviço orientado a conexão confirmada.

O serviço sem conexão não confirmada.

- Conexão não é previamente estabelecida;
- A máquina emissora envia frames sem receber confirmação de recebimento da máquina receptora;
- Quadros perdidos são ignorados e tratados pelas camadas superiores;
- Adequado para:
 - Aplicações onde a taxa de erro é muito baixa;
 - Aplicações de tempo real onde dados atrasados são piores que dados ruins, como streaming de áudio.
- Serviço normalmente usado em LANs.

O serviço sem conexão confirmada.

- Conexão é estabelecida previamente;
- Cada frame enviado é individualmente confirmado. Dessa forma, o emissor sabe se o frame foi recebido ou não e poderá enviá-lo novamente;
- A origem usa um mecanismo de temporização para reenviar quadros não confirmados;
- Útil para canais não confiáveis, como Wireless.

Serviço orientado a conexão confirmado

Serviço mais sofisticado:

- Emissor e receptor estabelecem conexão antes do envio dos dados;
- Cada frame enviado é numerado;
- Cada frame é recebido exatamente uma vez e todos os frames chegam em ordem;
- O serviço oferecido para a camada de rede é uma sequência de bits corretos.

Considerações sobre os diferentes tipos de serviço:

Considerações sobre os diferentes tipos de serviço:

1. ****Serviço sem conexão não confirmada****:

- Não requer estabelecimento prévio de conexão.
- Os frames são enviados sem confirmação de recebimento.
- Adequado para aplicações onde a taxa de erro é baixa e tolerável.

2. ****Serviço em conexão confirmada****:

- Requer estabelecimento prévio de conexão.
- Cada frame é confirmado individualmente.
- Utilizado em canais não confiáveis, como em conexões wireless.

Considerações sobre os diferentes tipos de serviço:

3. **Serviço orientado a conexão confirmada**:

- Mais sofisticado, requer estabelecimento prévio de conexão.
- Cada frame é numerado e recebido exatamente uma vez, garantindo ordem.
- Oferece uma sequência de bits corretos para a camada de rede.

Essas considerações destacam a importância de escolher o tipo de serviço adequado dependendo das necessidades específicas da aplicação e das características da rede utilizada.

Delimitação de quadros

- A camada física transmite uma sequência de bits (bitstream), que pode ser grande e conter erros;
- A camada de enlace deve detectar e, se necessário, corrigir os erros de transmissão;
- Inicialmente, a sequência de bits é quebrada em frames.

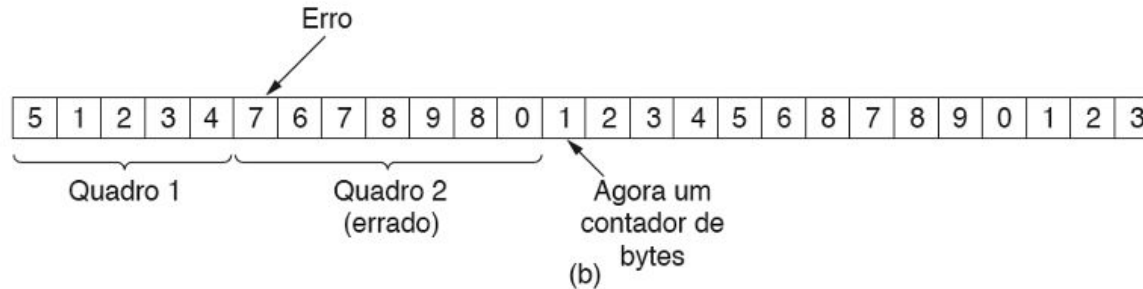
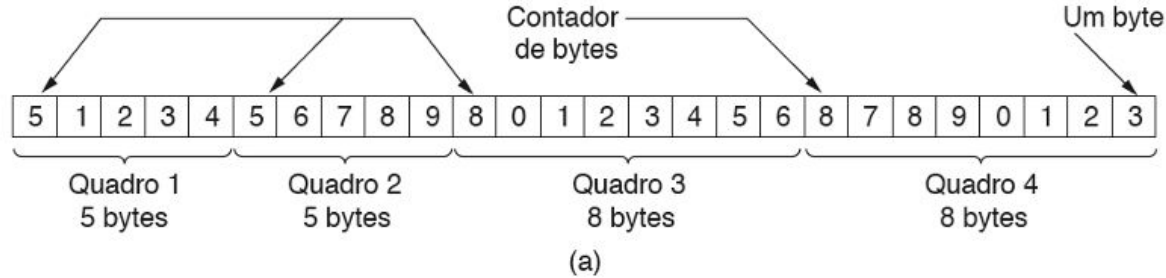
Delimitação de quadros

- Os principais métodos para delimitação de quadros são:
 - Contagem de caracteres;
 - Caracteres de início e fim, com caractere de preenchimento;
 - Flags de início e fim, com caractere de preenchimento; e
 - Violação de código da camada física.

Contagem de Caracteres

- Usa um campo no cabeçalho para indicar o número de caracteres no quadro;
- Problema: o caractere de contagem pode sofrer erro de transmissão, impossibilitando o reconhecimento do início do próximo quadro;
- Não é usado na prática para protocolos da camada de enlace.

Contagem de Caracteres



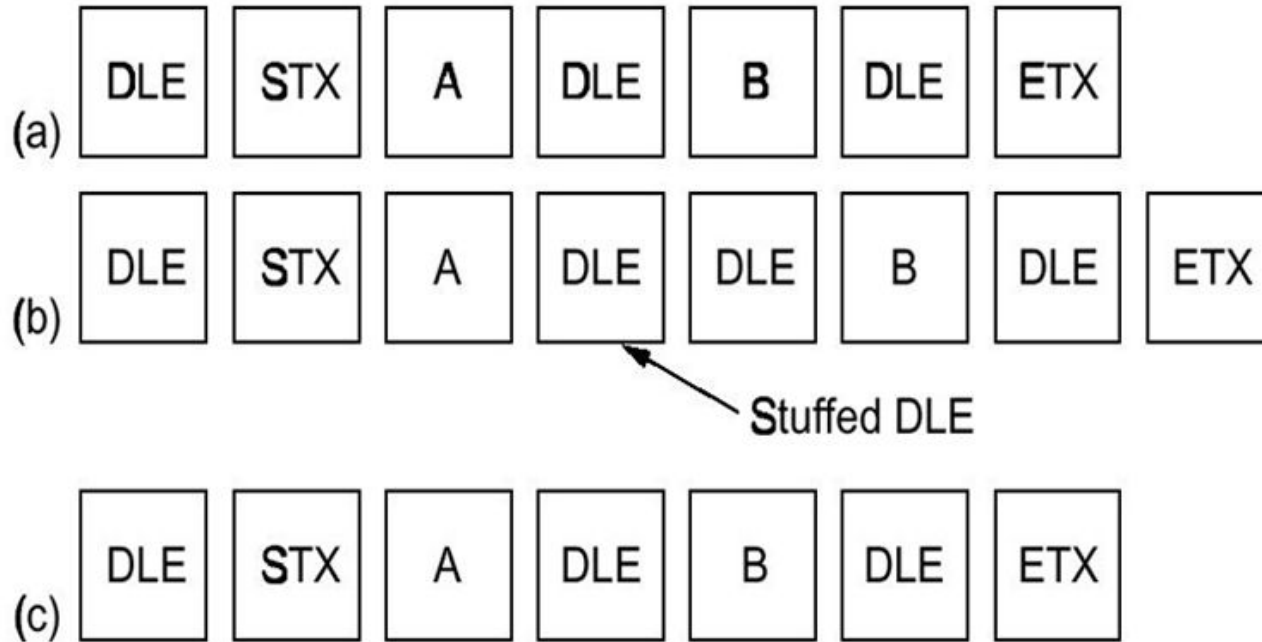
Fluxo de bytes, (a) sem erro, (b) com erro.

Caracteres de início e fim

- Reconhecimento do início e do fim de um quadro através dos caracteres ASCII:
 - Início: DLE STX (Data Link Escape, Start of Text);
 - Fim: DLE ETX (Data Link Escape, End of Text);
 - Método usado em protocolos orientados a caracteres.

Caracteres de início e fim

- E se os caracteres para DLESTX e DLEETX ocorrem nos dados?
- Inserir um caractere DLE adicional antes de cada DLE nos dados (bytestuffing);
- Técnica presa ao código ASCII e a caracteres de 8 bits (caracteres de tamanho arbitrário).



- (a) Dado enviado pela camada de rede.
- (b) Dado após a inserção do DLE pela camada de enlace.
- (c) Dado repassado para a camada de rede no receptor.

Flags de início e fim

- Permite codificar caracteres com um número arbitrário de bits por caractere;
- Usa padrão especial de bits (flag) para sinalizar início e fim do quadro;
- Sempre que 5 "uns" consecutivos são encontrados nos dados, o emissor insere um zero (bit stuffing);
- Quando o receptor encontra cinco "uns" seguidos por um zero, o stuff é retirado.

Flags de início e fim

Quadro Original

01111110	011111101010110	01111110
----------	-----------------	----------

Quadro Transmitido

01111110	011111 0 1010110	01111110
----------	-------------------------	----------

Quadro Recebido

01111110	011111 0 1010110	01111110
----------	-------------------------	----------

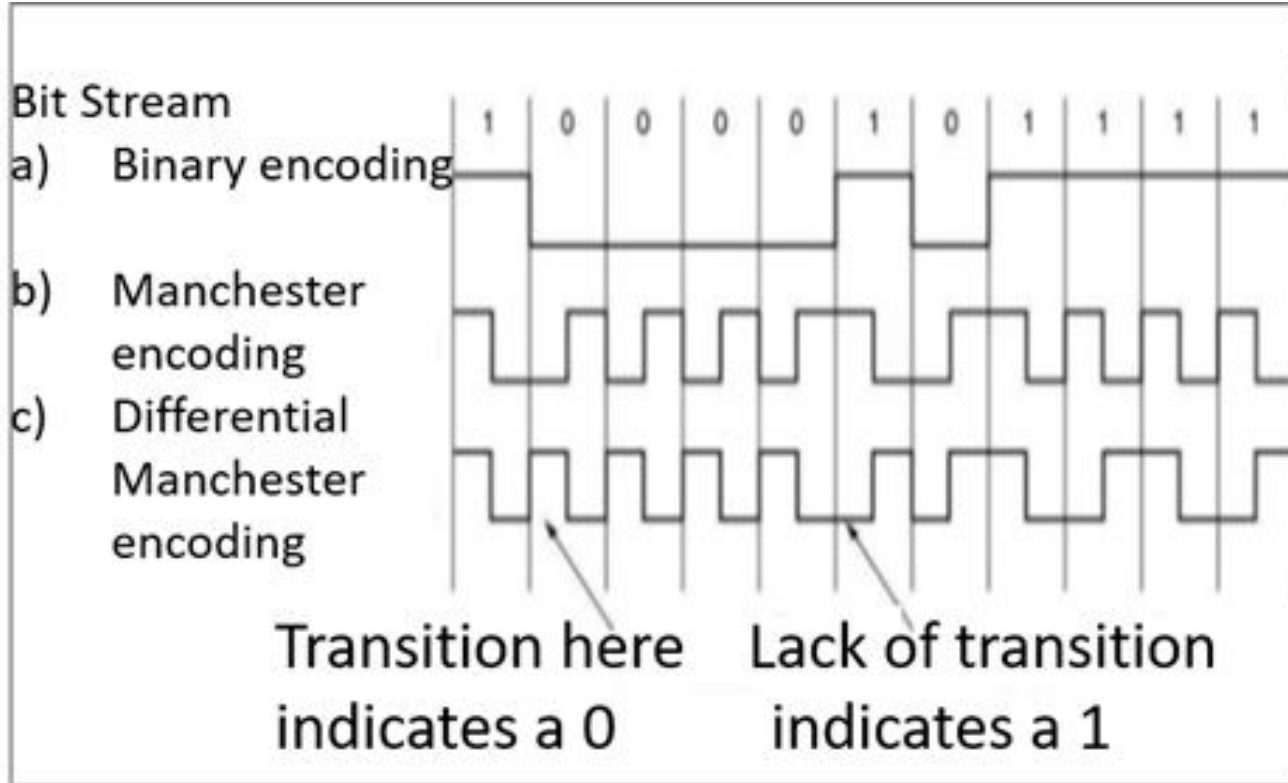
Quadro Original

01111110	011111101010110	01111110
----------	-----------------	----------

Violação de Código

- Método baseado em uma característica da camada inferior;
- Existem códigos de transmissão que possuem uma transição no meio do período de transmissão de um bit (Manchester);
- O início e fim de quadros são determinados por um código de transmissão inválido;
- Algumas LANs codificam um bit de dados usando dois bits físicos:
 - Um bit 1 é codificado como um par alto-baixo;
 - Um bit 0 é codificado como um par baixo-alto;
- Os pares alto-alto e baixo-baixo não são utilizados na codificação, então podem ser usados para a delimitação de quadros.
- Usado no padrão IEEE 802.

Violação de Código



EXERCÍCIO

slido.com with #2088733

Exercício

A codificação de caracteres a seguir é usada em um protocolo de enlace de dados:

- A: 01000111;
- B: 11100011;
- FLAG: 01111110;
- DLE: 00010000;
- STX: 00000010;
- ETX: 00000011.

Mostre a sequência de bits transmitida (em binário) para o quadro de quatro caracteres: A B DLE FLAG quando é utilizado cada um dos métodos de enquadramento a seguir:

- (a) Contagem de caracteres.
- (b) Caracteres de início e de fim.
- (c) *Flags* de início e de fim.

CONTROLE DE ERROS

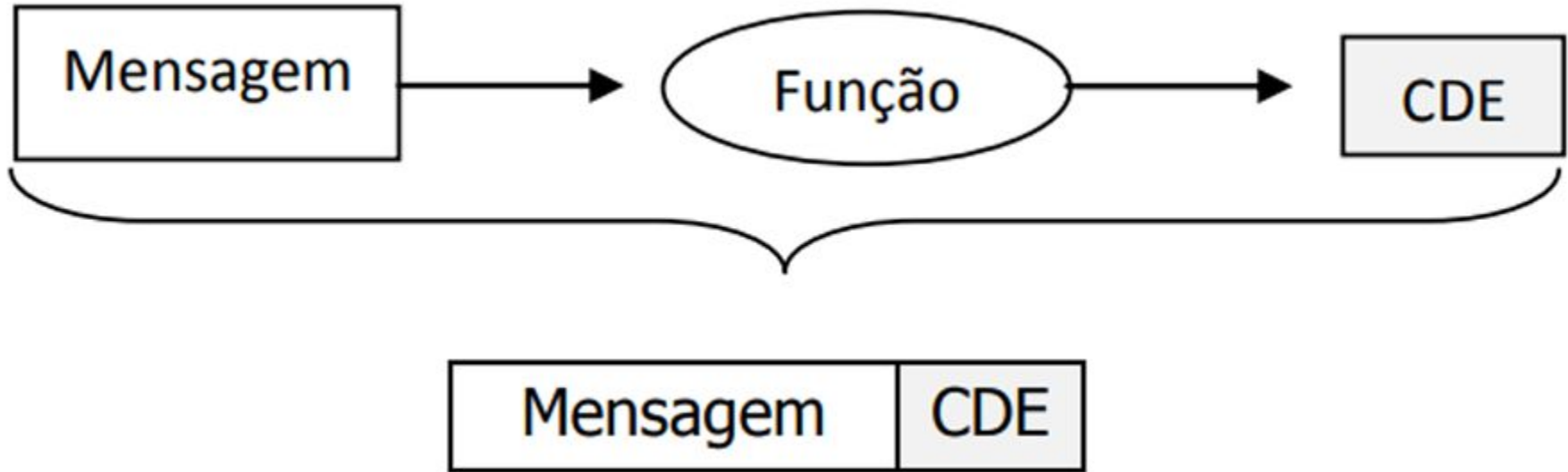
- Como ter certeza de que todos os quadros transmitidos foram entregues à camada de rede do destino e na ordem correta?
- Qualquer transmissão está sujeita a problemas, como ruídos e atenuação;
- A camada de enlace de dados tem a função de realizar o tratamento dos possíveis erros;
- O controle de erro envolve duas etapas:
 - Detecção dos possíveis erros nos dados transmitidos;
 - Correção dos erros encontrados.

DETECÇÃO DE ERROS

- O mecanismo de detecção de erro é semelhante ao esquema do dígito verificador largamente utilizado em códigos de barras, no CPF, etc.;
- O dígito verificador é gerado a partir dos números que compõem os números antecessores, utilizando-se de uma função previamente definida;

DETECÇÃO DE ERROS

- Antes de enviar uma mensagem, o transmissor utiliza uma função para gerar um código de detecção de erro (CDE) a partir da mensagem a ser enviada, de forma a gerar uma espécie de dígito verificador. Esse código é adicionado ao quadro que será enviado.



DETECÇÃO DE ERROS

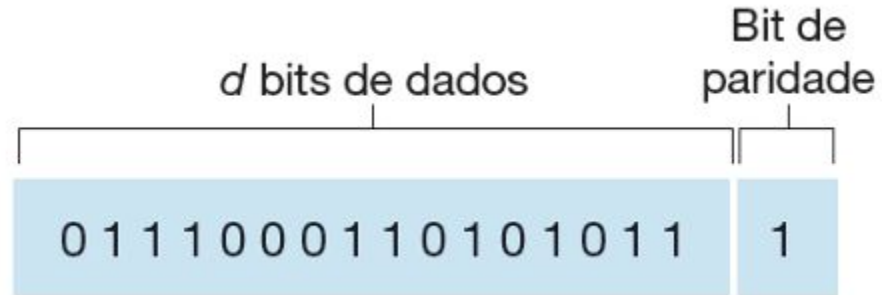
- Bit de paridade;
- Checksum;
- Redundância Cíclica (CRC).

BIT DE PARIDADE

A detecção de erros com bit de paridade funciona da seguinte maneira:

1. Adição do bit de paridade: Um bit adicional, chamado de bit de paridade, é adicionado aos dados. Esse bit é calculado de forma que o número total de bits 1 nos dados, incluindo o bit de paridade, seja sempre par (paridade par) ou sempre ímpar (paridade ímpar).
2. Verificação: Após a transmissão dos dados, o receptor verifica se o número de bits 1, incluindo o bit de paridade, é par ou ímpar, dependendo do tipo de paridade utilizado.
3. Detecção de erro: Se o número de bits 1 não corresponder ao tipo de paridade esperado, isso indica que ocorreu um erro na transmissão dos dados.
4. ****Limitações****: A detecção de erro com bit de paridade só pode detectar a ocorrência de um número ímpar de erros, pois um erro par não alteraria a paridade dos dados. Além disso, não é capaz de corrigir erros, apenas detectá-los.

BIT DE PARIDADE



BIT DE PARIDADE

Paridade bidimensional: os bits são divididos em i linhas e j colunas, e para cada linha e coluna é calculado um bit de paridade. Exemplo para um número par:

Nenhum erro								Erro de bit único corrigível							
1	0	1	0	1	1			1	0	1	0	1	1		
1	1	1	1	0	0			1	0	1	1	0	0		
0	1	1	1	0	1			0	1	1	1	0	1		
0	0	1	0	1	0			0	0	1	0	1	0		

Diagram illustrating bidimensional parity for error correction. The left table shows a correct data matrix with parity bits. The right table shows a single bit error (highlighted in blue) at row 2, column 2. The parity bits for the affected row and column are also highlighted in blue, indicating the error location.

Possível fazer correção no caso de um erro, mas não em dois erros em linhas ou colunas consecutivas.

BIT DE PARIDADE

Paridade bidimensional: os bits são divididos em i linhas e j colunas, e para cada linha e coluna é calculado um bit de paridade. Exemplo para um número par:

Nenhum erro								Erro de bit único corrigível							
1	0	1	0	1	1			1	0	1	0	1	1		
1	1	1	1	0	0			1	0	1	1	0	0		
0	1	1	1	0	1			0	1	1	1	0	1		
0	0	1	0	1	0			0	0	1	0	1	0		

Diagram illustrating bidimensional parity for a 4x6 data matrix. The left table shows a correct state with no errors. The right table shows a single bit error (the second bit in the second row, which is 0 instead of 1). Blue arrows indicate the error detection process: a vertical arrow points to the second column, labeled 'Erro de paridade' (Parity Error), and a horizontal arrow points to the second row, also labeled 'Erro de paridade'.

Possível fazer correção no caso de um erro, mas não em dois erros em linhas ou colunas consecutivas.

BIT DE PARIDADE

Paridade bidimensional: os bits são divididos em i linhas e j colunas, e para cada linha e coluna é calculado um bit de paridade. Exemplo para um número par:

Nenhum erro								Erro de bit único corrigível							
1	0	1	0	1	1			1	0	1	0	1	1		
1	1	1	1	0	0			1	0	1	1	0	0		
0	1	1	1	0	1			0	1	1	1	0	1		
0	0	1	0	1	0			0	0	1	0	1	0		

Diagram illustrating bidimensional parity for a 4x6 data matrix. The left table shows a correct state with no errors. The right table shows a single bit error (the bit at row 2, column 2 is 0 instead of 1). Blue arrows indicate the error detection process: a vertical arrow points to the second column, labeled 'Erro de paridade' (Parity Error), and a horizontal arrow points to the second row, also labeled 'Erro de paridade'.

Possível fazer correção no caso de um erro, mas não em dois erros em linhas ou colunas consecutivas.

CHECKSUM

- Checksum: é usado para indicar um grupo de bits de verificação, independentemente de como são calculados.
- Um grupo de bits de paridade pode ser um exemplo de checksum.
- Porém, existem checksums mais robustos que os bits de paridade.
- Opera sobre palavras e não bits.
- Erros que passaram pelos bits de paridade podem ser encontrados.
- Exemplo: Protocolo IP - soma de verificação de 16 bits.

CHECKSUM

- Transmissor:
 - Divide a mensagem em k segmentos de n bits;
 - Soma os k segmentos;
 - Forma o Checksum com o complemento da soma;
 - Envia a mensagem junto com o checksum;

CHECKSUM

Receptor:

- Divide a mensagem em k segmentos de bits;
- Soma os k segmentos;
- Forma o Checksum com o complemento da soma;
- Se o checksum for igual a zero, dados aceitos!

CHECKSUM

Exemplo:

- Mensagem recebida: 101010010011100100011101;
- Segmentos: 10101001 - 00111001 - 0011101;
- Soma: $10101001 + 00111001 + 0011101 = 11111111$;
- Checksum: 00000000 (checksum é zero, mensagem aceita!);
- Mensagem decodificada: 1010100100111001 (o checksum enviado é descartado).