Instituto de computação

Departamento de Ciência da Computação
Instituto de Computação
Universidade Federal Fluminense

Aulas 1 e 2

Camada de Enlace

Serviços, endereçamento e elementos de interconexão

Igor Monteiro Moraes

Redes de Computadores II

ATENÇÃO!

Instituto de computação

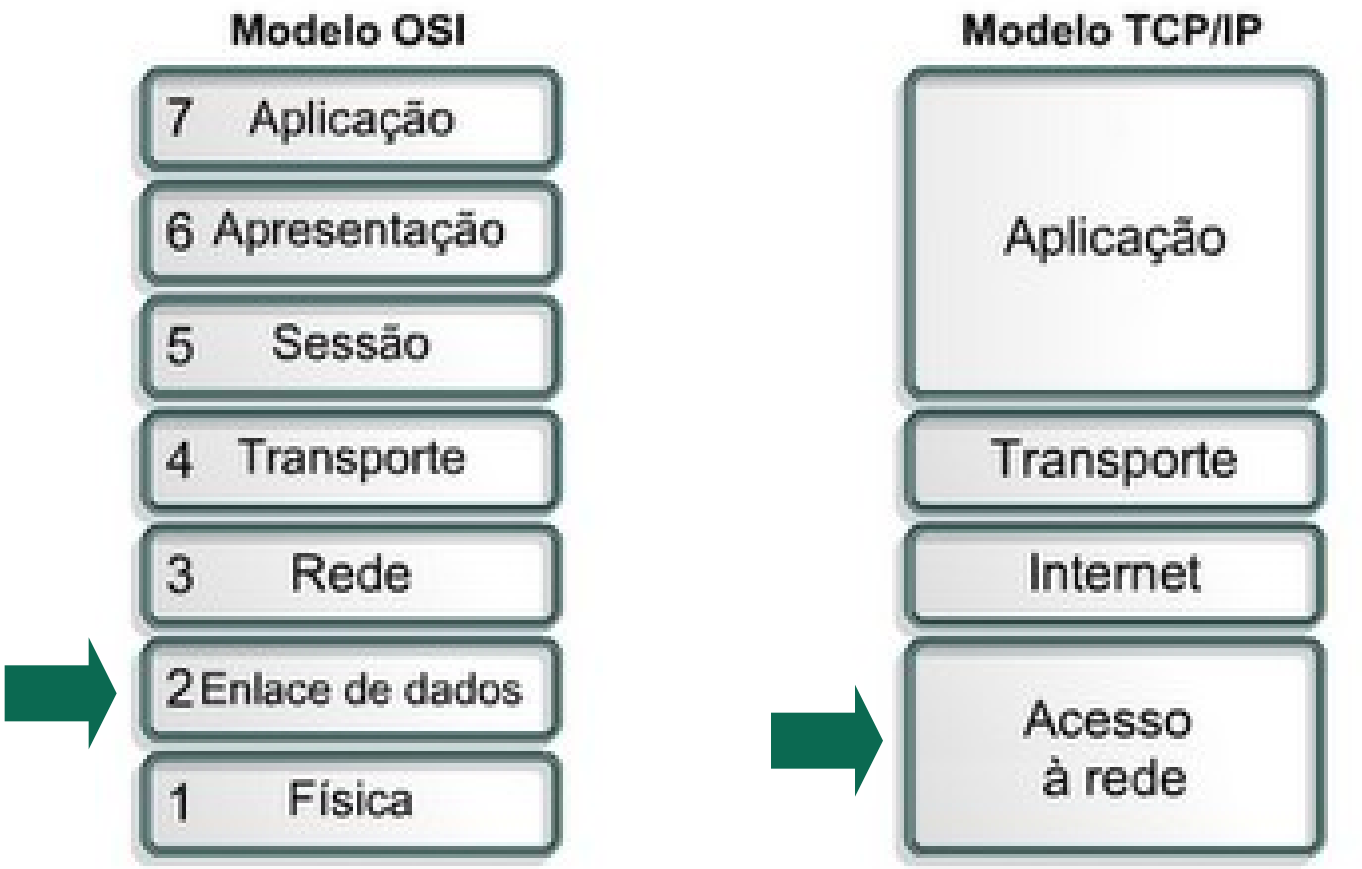
- Este apresentação é contém partes baseadas nos seguintes trabalhos
  - Notas de aula do Prof. Marcelo Rubinstein, disponíveis em http://www.lee.eng.uerj.br/~rubi
  - Notas de aula do Prof. José Augusto Suruagy Monteiro, disponíveis em http://www.nuperc.unifacs.br/Members/jose.suruagy/cursos
  - Material complementar do livro Computer Networking: A Top Down Approach, 5th edition, Jim Kurose and Keith Ross, Addison-Wesley, abril de 2009
  - Computer Networks, Andrew S. Tanenbaum, 4a. Edição, Editora Prentice Hall

Revisão

Instituto de computação

- Quais são as camadas do modelo TCP/IP?
- Qual a função de cada camada?

Camada de Enlace



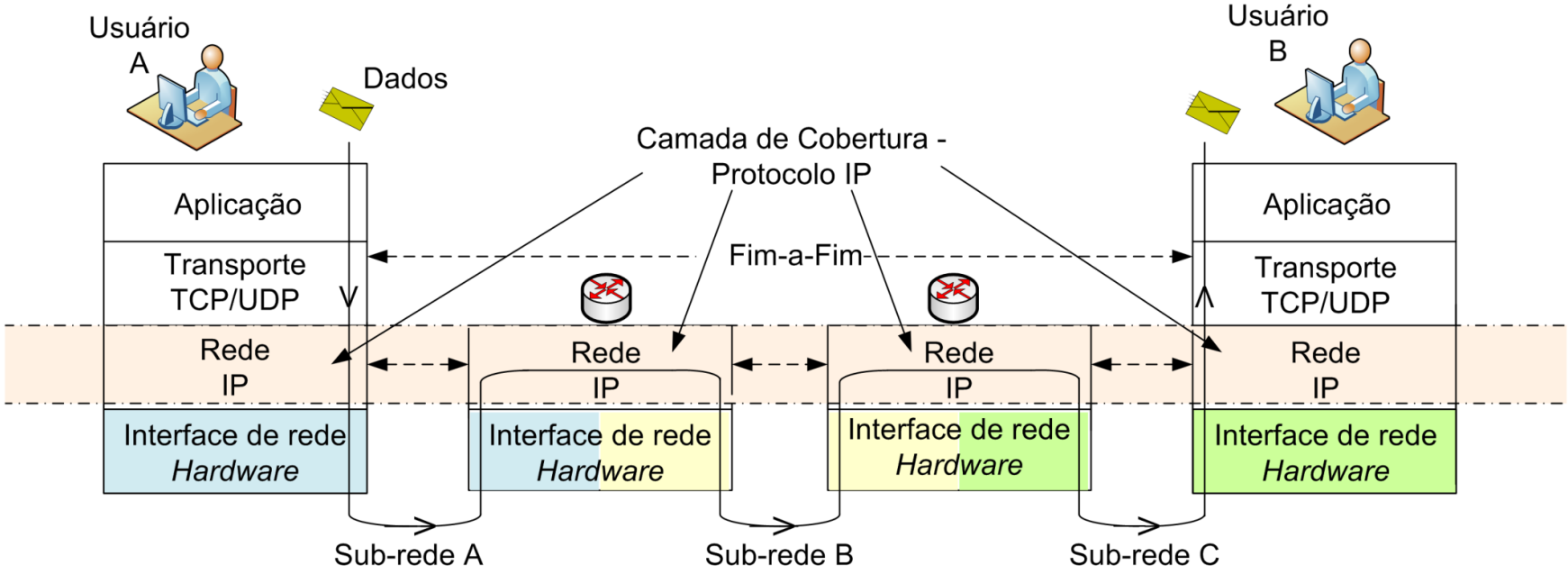
Camada de Rede

Instituto de computação

- Responsável por
  - **Determinar o melhor caminho** para o envio dos pacotes
    - É função dos protocolos de roteamento
  - **Encaminhar** os pacotes até o destino
    - É função do protocolo IP
  - **Interconectar** redes de diferentes tecnologias
    - É função do protocolo IP

Rede x Enlace

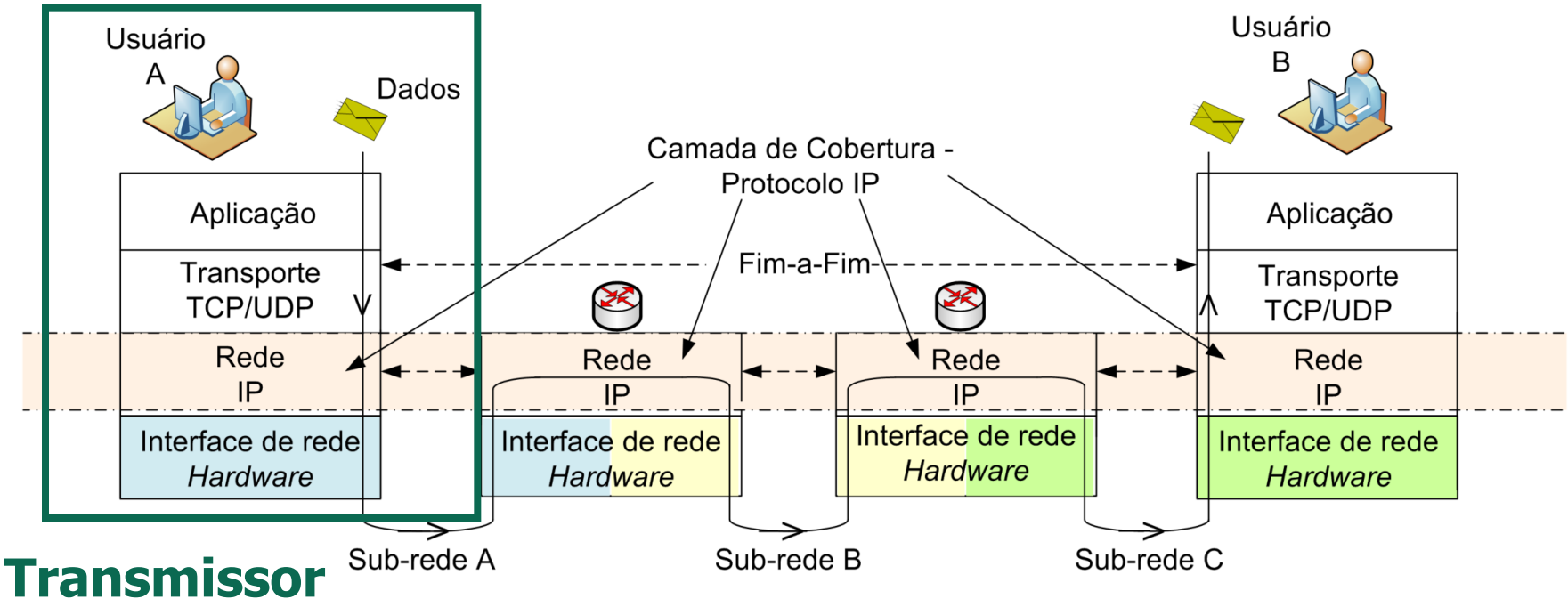
- Protocolos da camada de rede
  - Executados nos **sistemas finais** e nos **roteadores**



Transporta segmentos da estação remetente à receptora

Rede x Enlace

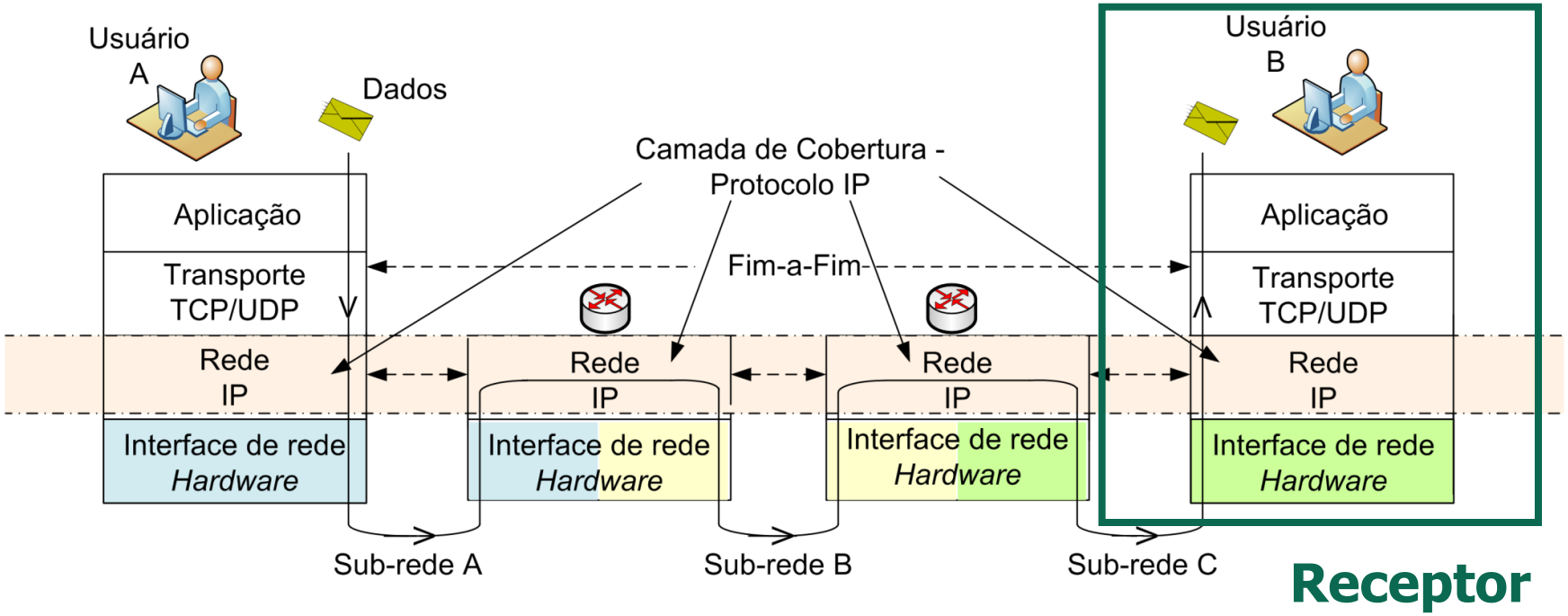
- Protocolos da camada de rede
  - Executados nos **sistemas finais** e nos **roteadores**



No lado transmissor, encapsula segmentos dentro de datagramas

Rede x Enlace

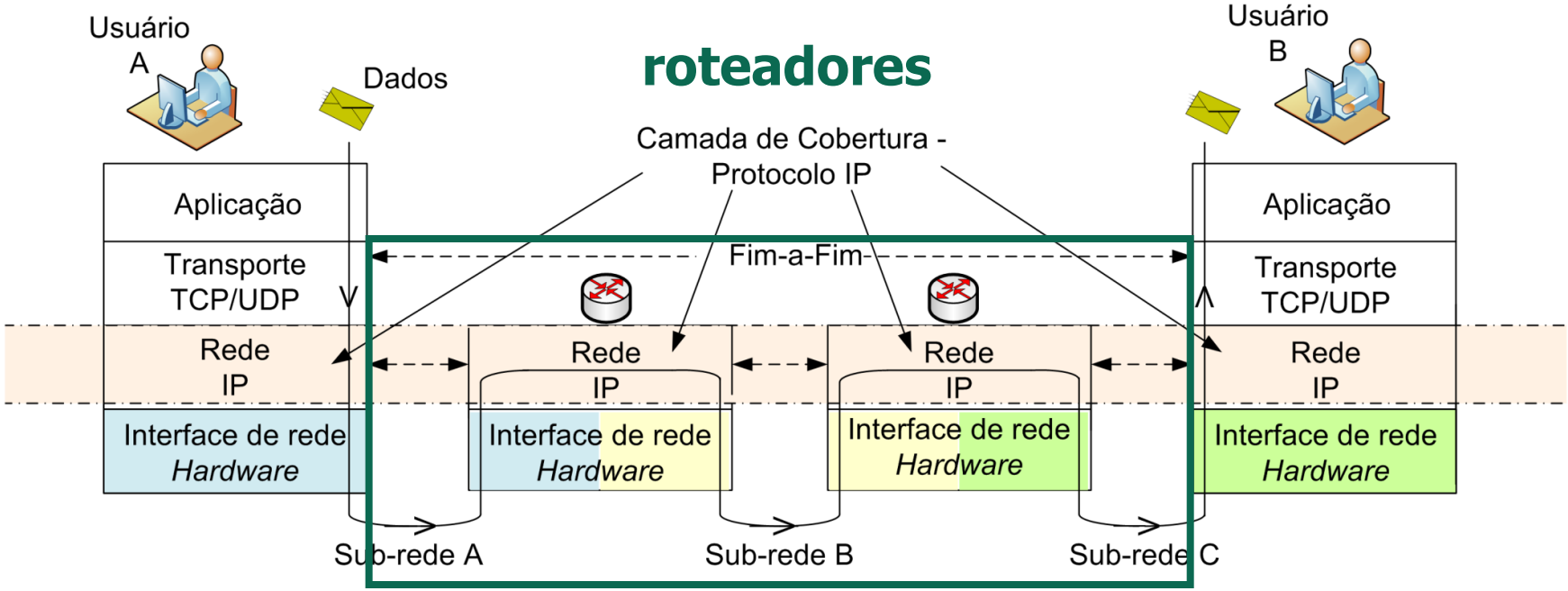
- Protocolos da camada de rede
  - Executados nos **sistemas finais** e nos **roteadores**



No lado receptor, entrega os segmentos para a camada de transporte

Rede x Enlace

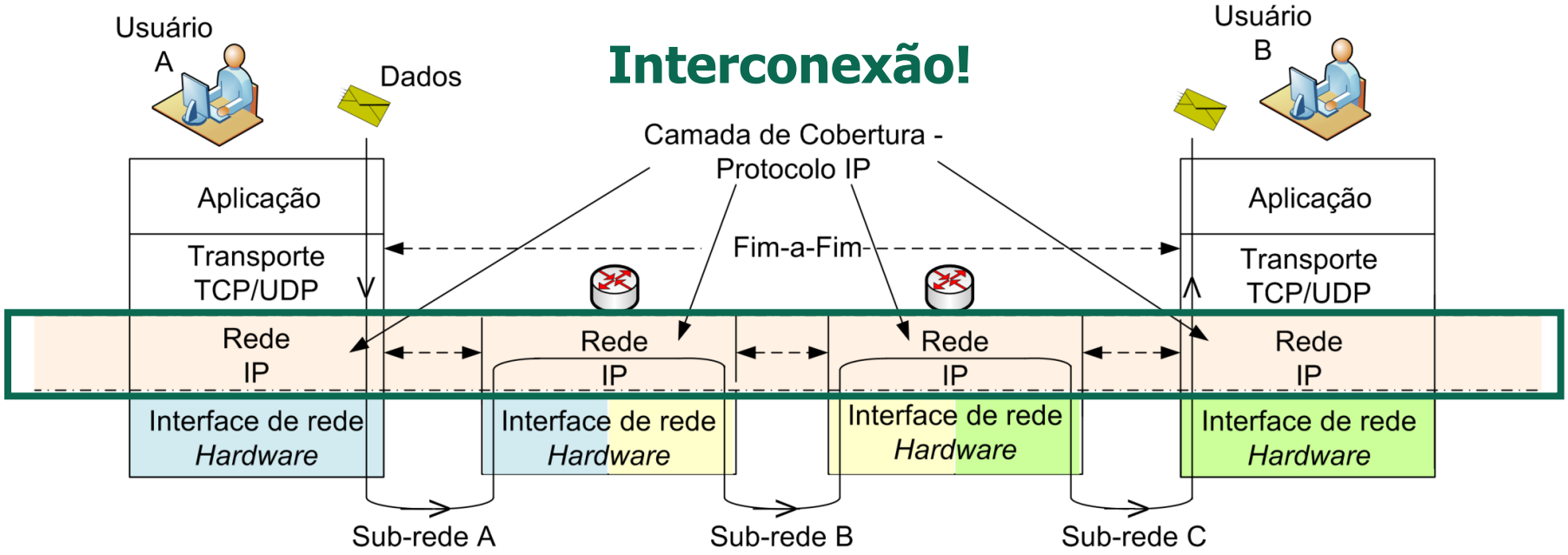
- Protocolos da camada de rede
  - Executados nos **sistemas finais** e nos **roteadores**



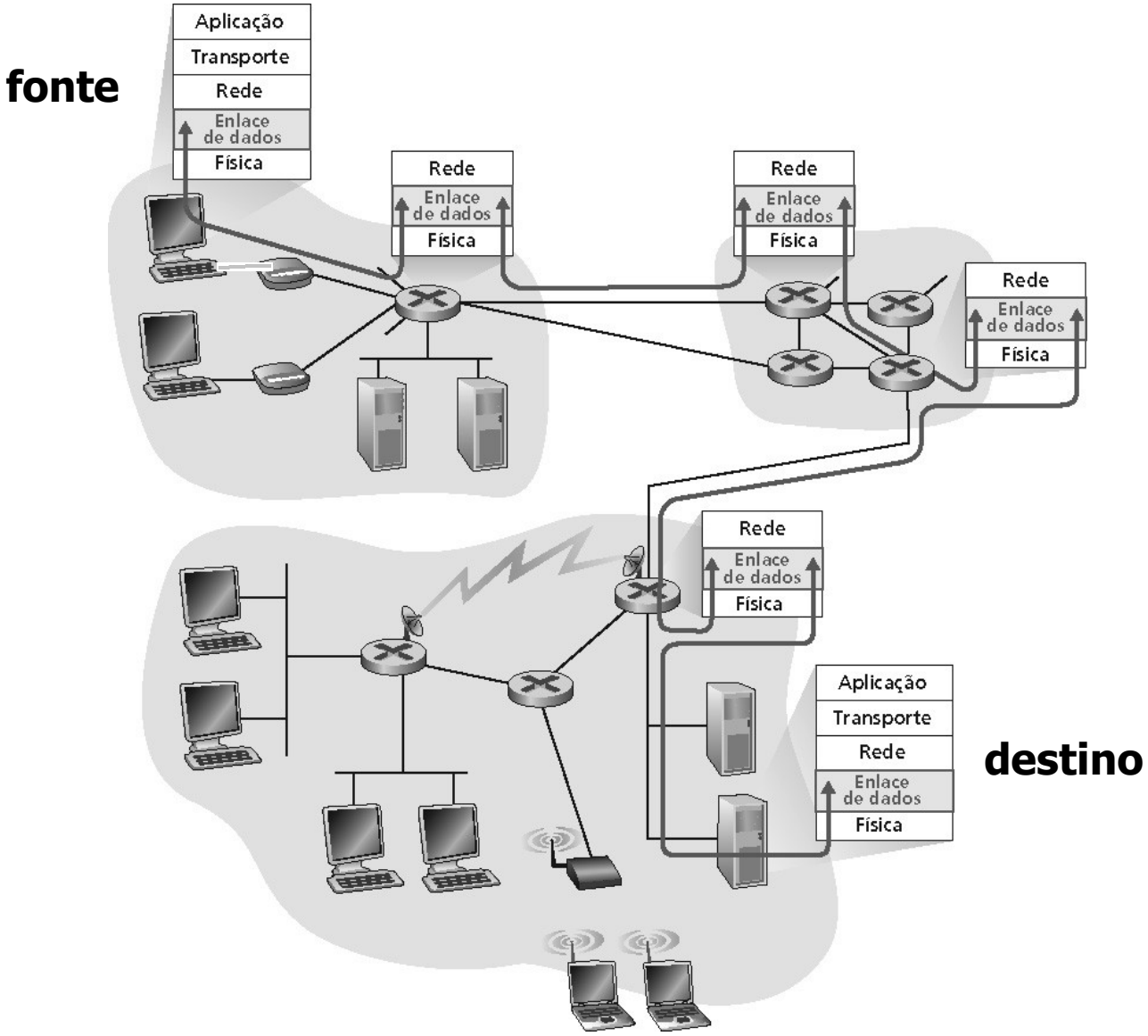
Roteadores examinam campos de cabeçalho de **todos** os datagramas IP que passam por eles

Rede x Enlace

- Protocolos da camada de rede
  - Executados nos **sistemas finais** e nos **roteadores**



Transmissão dos dados pelos enlaces (fonte: Kurose)



Camada de Rede

Instituto de computação

- Responsável por
  - **Determinar o melhor caminho** para o envio dos pacotes
    - É função dos protocolos de roteamento
  - **Encaminhar** os pacotes até o destino
    - É função do protocolo IP
  - **Interconectar** redes de diferentes tecnologias
    - É função do protocolo IP

Um caminho é composto por um ou mais **enlaces** (*links*)

Camada de Rede

Instituto de computação

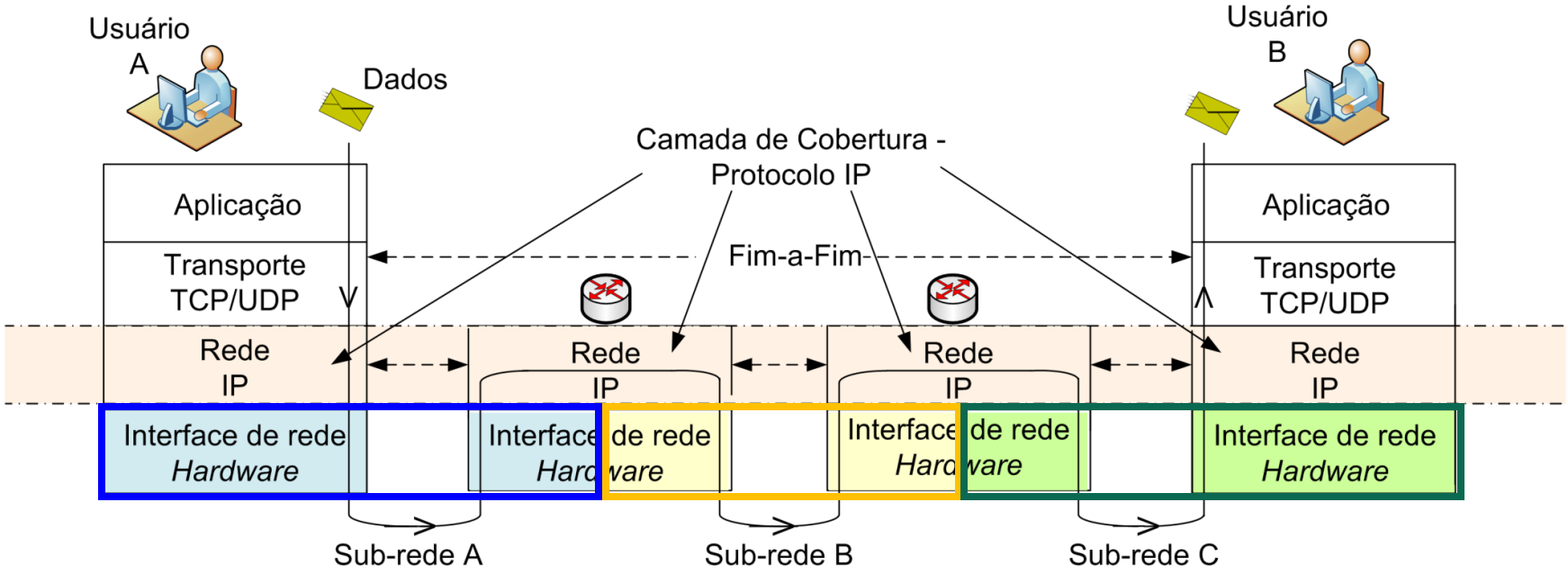
- Responsável por
  - **Determinar o melhor caminho** para o envio dos pacotes
    - É função dos protocolos de roteamento
  - **Encaminhar** os pacotes até o destino
    - É função do protocolo IP
  - **Interconectar** redes de diferentes tecnologias
    - É função do protocolo IP

Um caminho é composto por um ou mais **enlaces** (*links*)

Como os dados são transmitidos em cada enlace?

Rede x Enlace

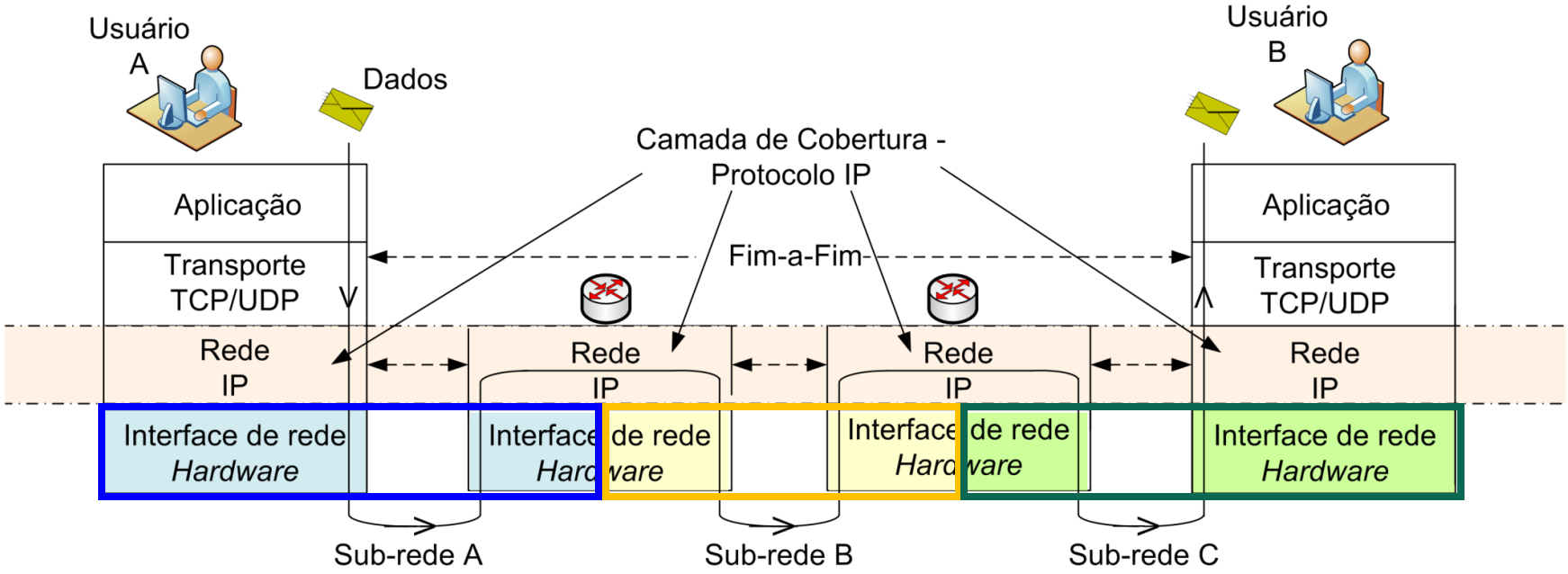
- Protocolos da camada de enlace
  - Executados em **todos os nós**



Cada **enlace** pode ter um **protocolo diferente**

Rede x Enlace

- Protocolos da camada de enlace
  - Executados em **todos os nós**



Serviço **salto-a-salto**

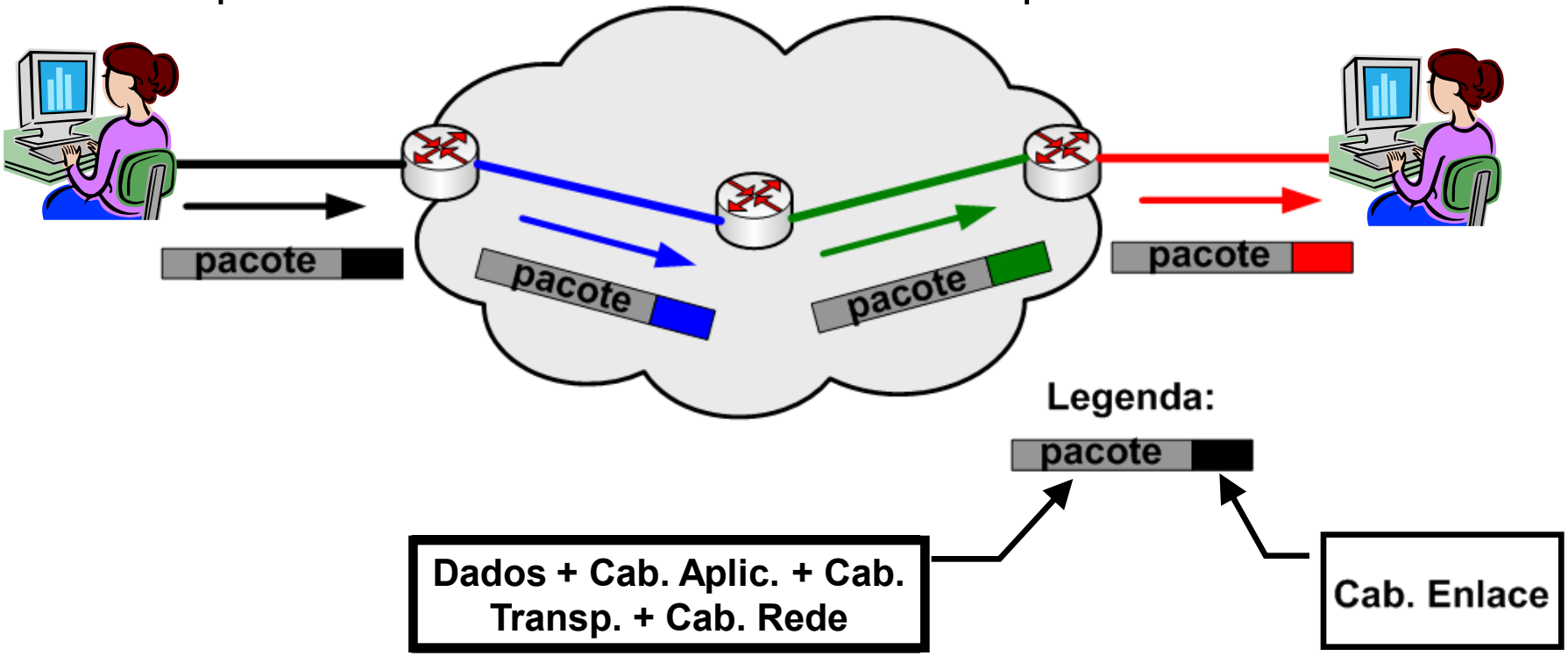
Camada de Enlace

Instituto de computação

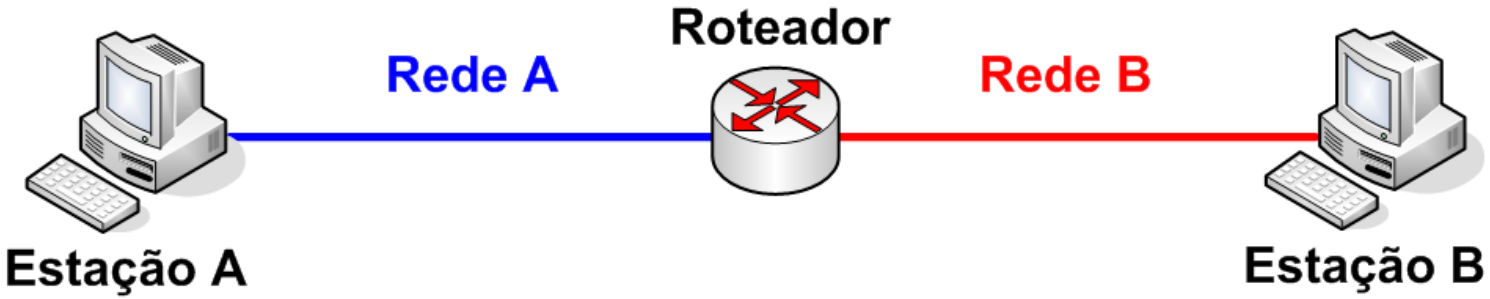
- Responsável por
  - **Transmitir sobre o meio físico** os datagramas provenientes da camada de rede **salto-a-salto**
    - Serviço básico
    - Definem
      - O formato dos quadros trocados entre os nós adjacentes
      - Ações de cada nó ao enviar e receber um quadro
- Um enlace é um canal de comunicação entre nós adjacentes
  - Pacote da camada 2 é um quadro (*frame*)
    - Encapsula o datagrama

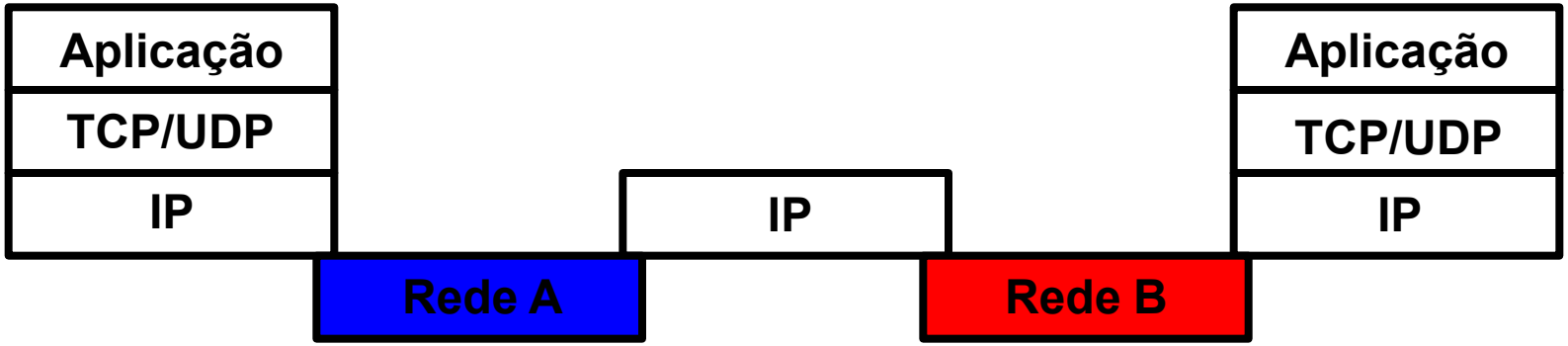
Operação do IP

- Transparência sintática
  - Pacotes são transferido da origem ao destino sem que a rede modifique os dados
    - Apenas erros de transmissão modificam pacotes

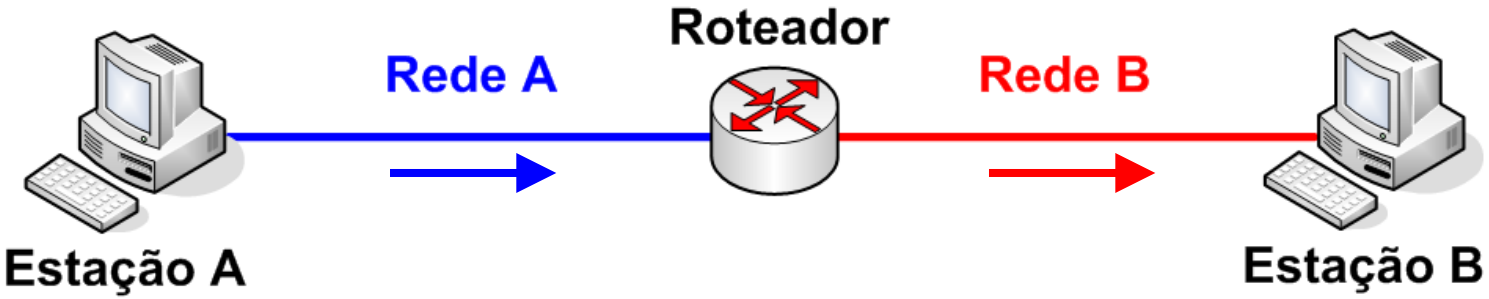


Operação do IP





Transmissão de um Pacote IP



A1 → C1, IP	A → B, TCP	cabeçalho TCP + dados
Cabeçalho Ethernet	Cabeçalho IP	

C2 → B2, IP	A → B, TCP	cabeçalho TCP + dados
Cabeçalho Ethernet	Cabeçalho IP	

Camada de Enlace

Instituto de computação

- Protocolos podem prover serviços além do básico
  - Enquadramento (*framing*)
  - Entrega confiável
  - Controle de fluxo
  - Detecção de erros
  - Correção de erros
  - Transmissão *half-duplex* ou *full-duplex*
  - Controle de acesso ao meio

Camada de Enlace

Instituto de computação

- Dois tipos de canais de comunicação
  - Canal **ponto-a-ponto**
    - Uma estação em cada extremidade
      - Redes de acesso domiciliares
      - Entre roteadores
  - Canal **difusão** (*broadcast*)
    - Várias estações conectadas ao mesmo canal
    - É necessário um protocolo para **coordenar as transmissões**
      - Protocolos de controle de acesso ao meio

Instituto de computação

Serviços da Camada de Enlace

Serviços da Camada de Enlace

Instituto de computação

- Enquadramento
  - Delimitar onde começa e onde termina um quadro
  - Encapsular um datagrama em um quadro
    - Adicionar cabeçalho e fim de quadro (*trailer*)
- Entrega confiável entre nós adjacentes
  - Similar aos mecanismos da camada de transporte
  - Pouco usada em canais com baixas taxas de erro
    - Ex.: fibra óptica, alguns tipos de pares trançados, etc.
  - Necessária em canais com altas taxas de erros
    - Ex.: canais sem-fio

Serviços da Camada de Enlace

Instituto de computação

- Controle de fluxo
  - Compatibilizar taxas de produção e consumo de quadros entre remetentes e receptores
- Detecção de erros
  - Erros são causados por atenuação do sinal e por ruído
  - Receptor detecta presença de erros
    - Receptor sinaliza ao remetente para retransmissão, ou simplesmente descarta o quadro em erro
- Correção de erros
  - Mecanismo que permite que o receptor localize e **corrija** o(s) erro(s) sem precisar da retransmissão

Serviços da Camada de Enlace

Instituto de computação

- *Half-duplex* e *full-duplex*
  - Com *half-duplex* um nó não pode transmitir e receber pacotes ao mesmo tempo

- Controle de acesso ao meio
  - Implementa o controle de acesso ao canal se meio for compartilhado
  - 'Endereços físicos (MAC)' são usados nos cabeçalhos dos quadros para identificar origem e destino de quadros em enlaces multiponto
    - Diferentes do endereço IP

Serviços da Camada de Enlace

Instituto de computação

- Funções similares: enlace e transporte
  - Para que confiabilidade nas duas camadas?
  - Para que controle de fluxo nas duas camadas?

Serviços da Camada de Enlace

Instituto de computação

- Funções similares: enlace e transporte
  - Para que confiabilidade nas duas camadas?
  - Para que controle de fluxo nas duas camadas?

Camada de transporte: fim-a-fim
x
Camada de enlace: salto-a-salto

Classificação dos Serviços

Instituto de computação

- Em geral, há três tipos de serviços providos
  - Não orientado a conexões e sem confirmação
  - Não orientado a conexões e com confirmação
  - Orientado a conexões e com confirmação

Classificação dos Serviços

Instituto de computação

- Não orientado a conexões e sem confirmação
  - Apropriado quando a **taxa de erros é baixa**
  - Recuperação de perdas a cargo das camadas superiores
  - Apropriado para tráfego de tempo real
  - Maior parte das redes locais usa um serviço desse tipo

Classificação dos Serviços

Instituto de computação

- Não orientado a conexões e com confirmação
  - Quadros são numerados
  - Usa temporizadores para implementar a confiabilidade
  - Usado atualmente em redes sem-fio
    - Essas redes possuem canais não confiáveis

Classificação dos Serviços

Instituto de computação

- Orientado a conexões e com confirmação
  - Quadros são numerados
  - Usa temporizadores para implementar a confiabilidade

- Confirmação na camada enlace
  - Questão de otimização
    - Pode estar em camadas superiores
      - Problema é a fragmentação dos pacotes em quadros
        - » Pode fazer com que se leve muito tempo para transmitir um pacote

Enquadramento

Instituto de computação

- Por que é preciso?
  - Serviço provido pela camada física não garante que o fluxo de bits seja livre de erros
    - Número de bits pode ser maior ou menor do que o número de bits transmitidos
    - Bits podem ter valores diferentes dos bits transmitidos

Enquadramento

Instituto de computação

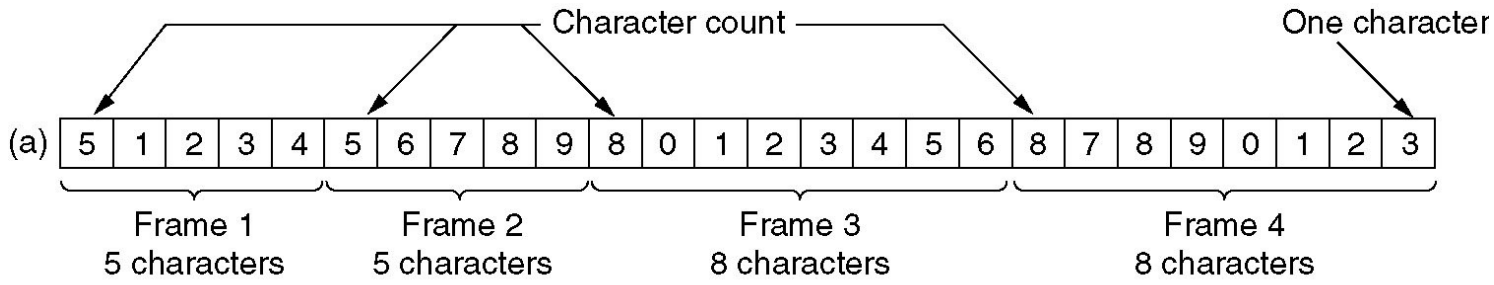
- Camada enlace divide o fluxo de bits em quadros e faz uma verificação em cada quadro (detecção e correção de erros)

- Vários métodos para marcar o início e o fim dos quadros
  - Contagem de caracteres
  - Bytes de *flags*, com inserção de bytes
  - *Flags* iniciais e finais, com inserção de bits
  - Entre outros

Enquadramento: contagem de caracteres

- Usa um campo no cabeçalho para especificar o número de caracteres do quadro

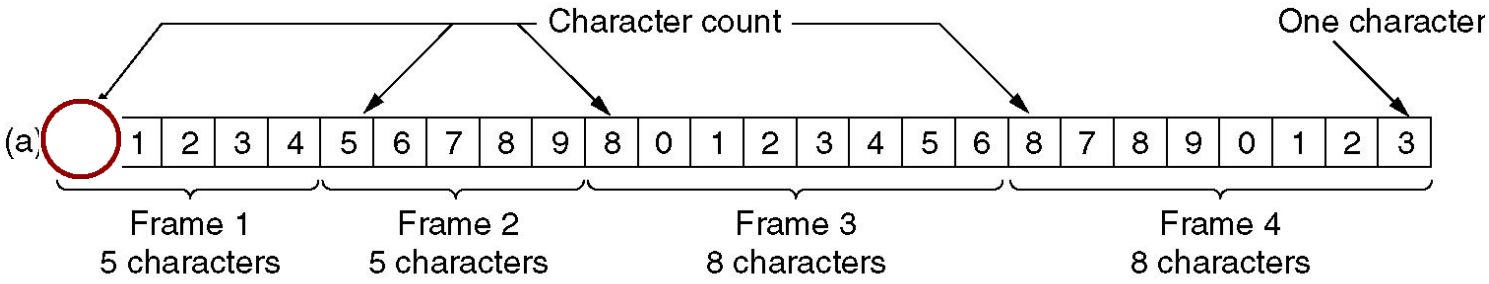
Exemplo de contagem de caracteres (fonte: Tanenbaum)



Enquadramento: contagem de caracteres

- Usa um campo no cabeçalho para especificar o número de caracteres do quadro

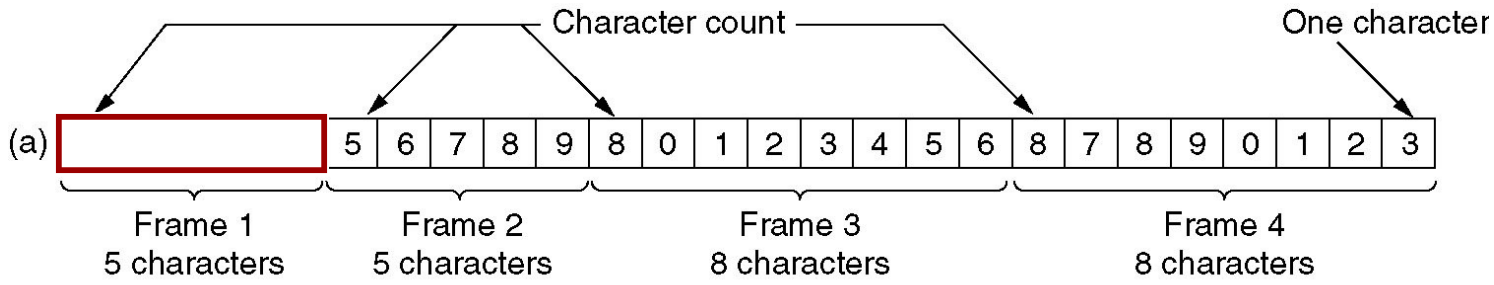
Exemplo de contagem de caracteres (fonte: Tanenbaum)



Enquadramento: contagem de caracteres

- Usa um campo no cabeçalho para especificar o número de caracteres do quadro

Exemplo de contagem de caracteres (fonte: Tanenbaum)



Enquadramento: contagem de caracteres

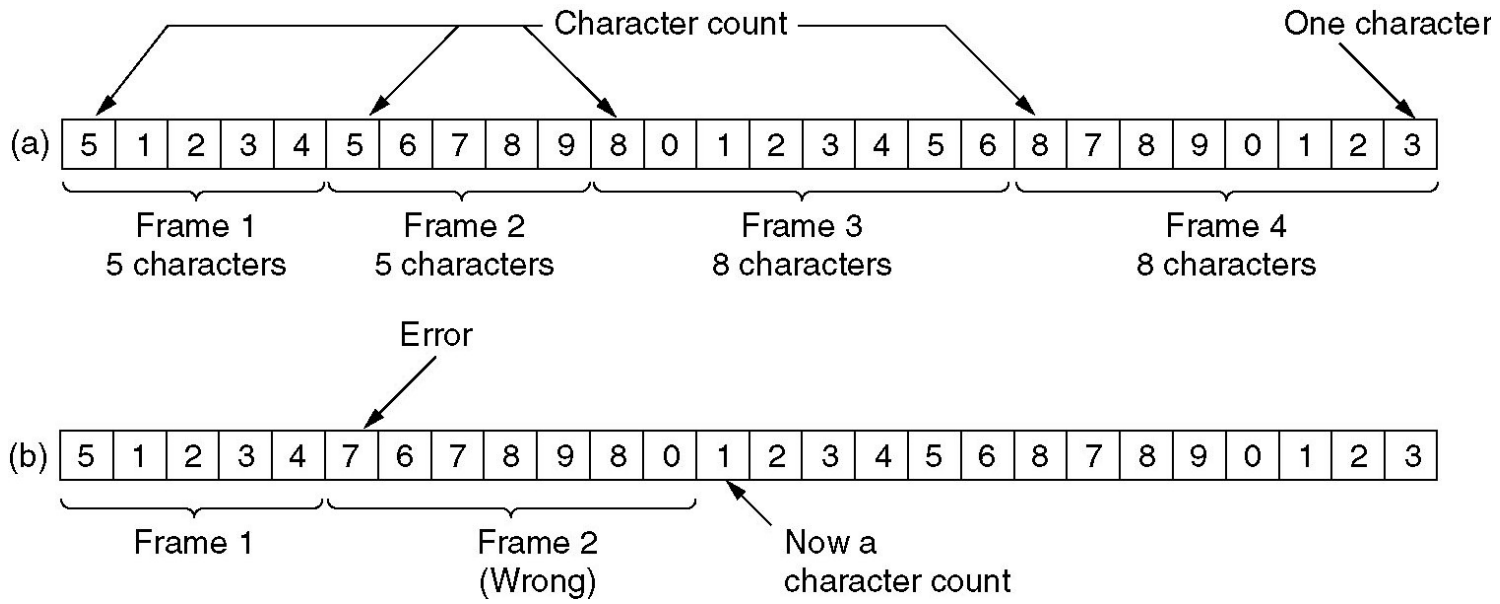
Instituto de computação

- Problema
  - Contagem pode ser adulterada por um erro de transmissão
    - Mesmo com a verificação incorreta, destino não sabe onde começa o próximo quadro
    - Solicitação de retransmissão também não adianta
      - Destino não sabe quantos caracteres devem ser ignorados para chegar ao início da retransmissão

Enquadramento: contagem de caracteres

Instituto de computação

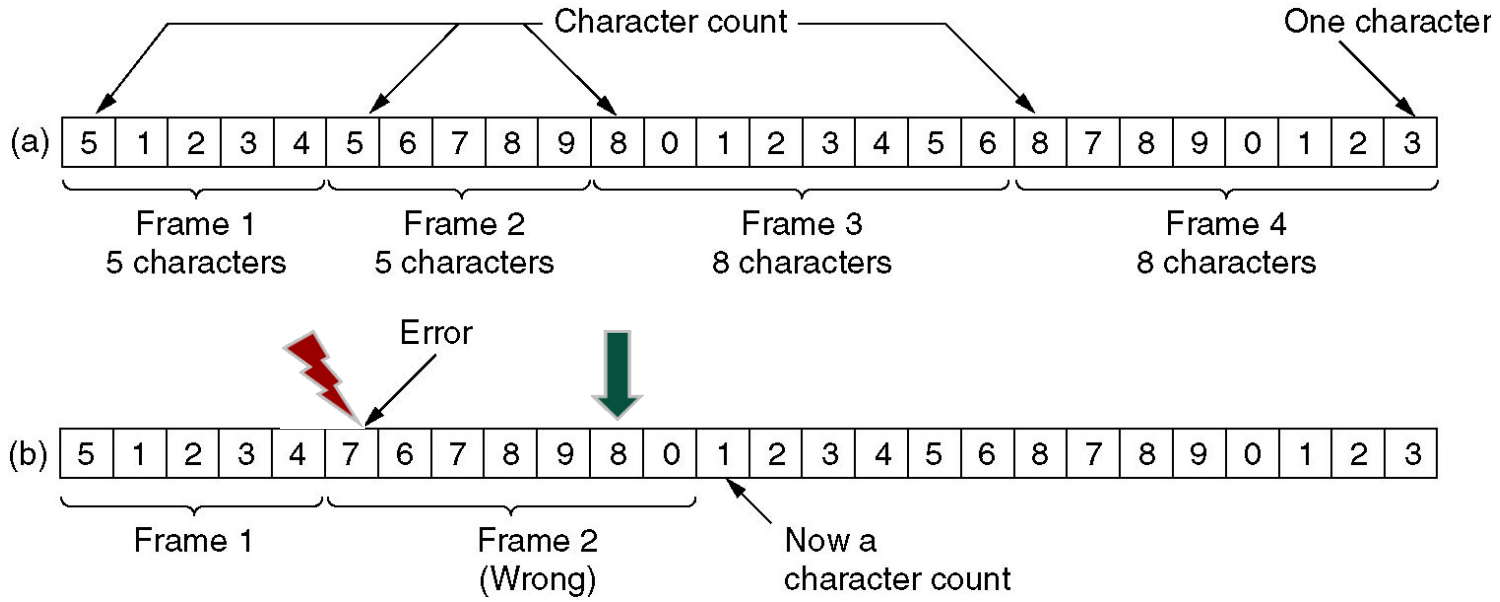
Exemplo de contagem de caracteres com um erro (fonte: Tanenbaum)



Enquadramento: contagem de caracteres

Instituto de computação

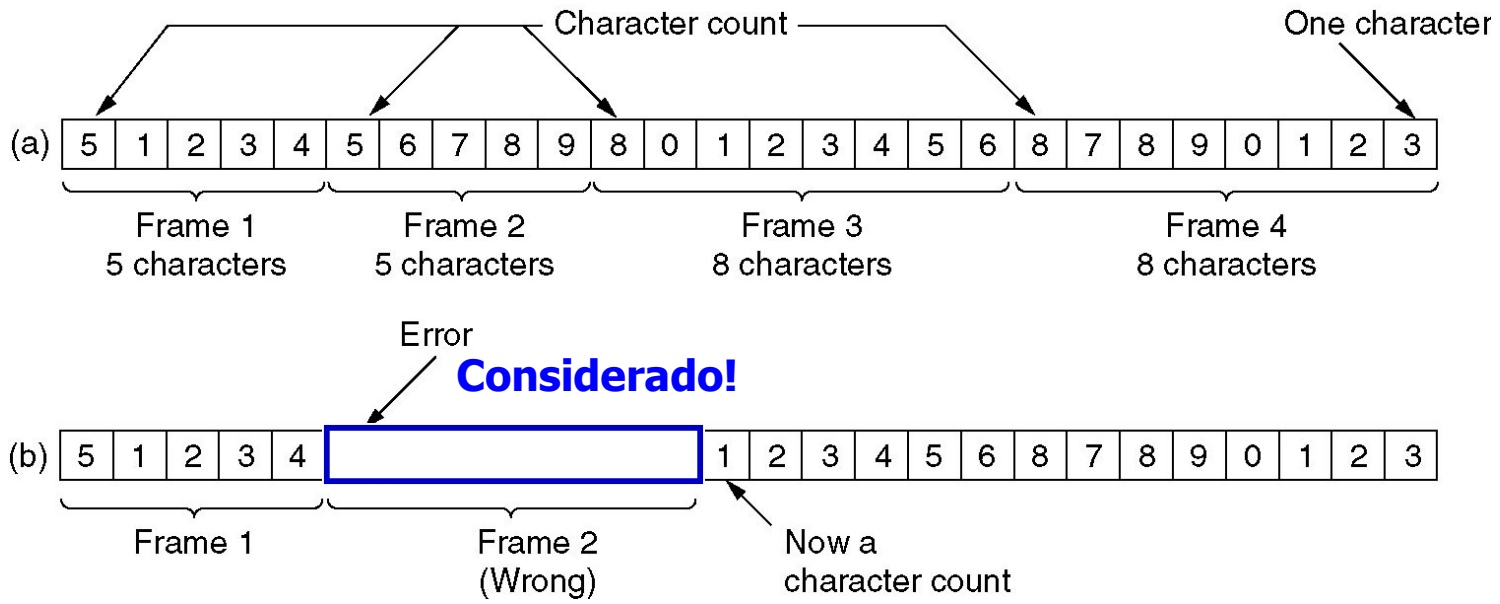
Exemplo de contagem de caracteres com um erro (fonte: Tanenbaum)



Enquadramento: contagem de caracteres

Instituto de computação

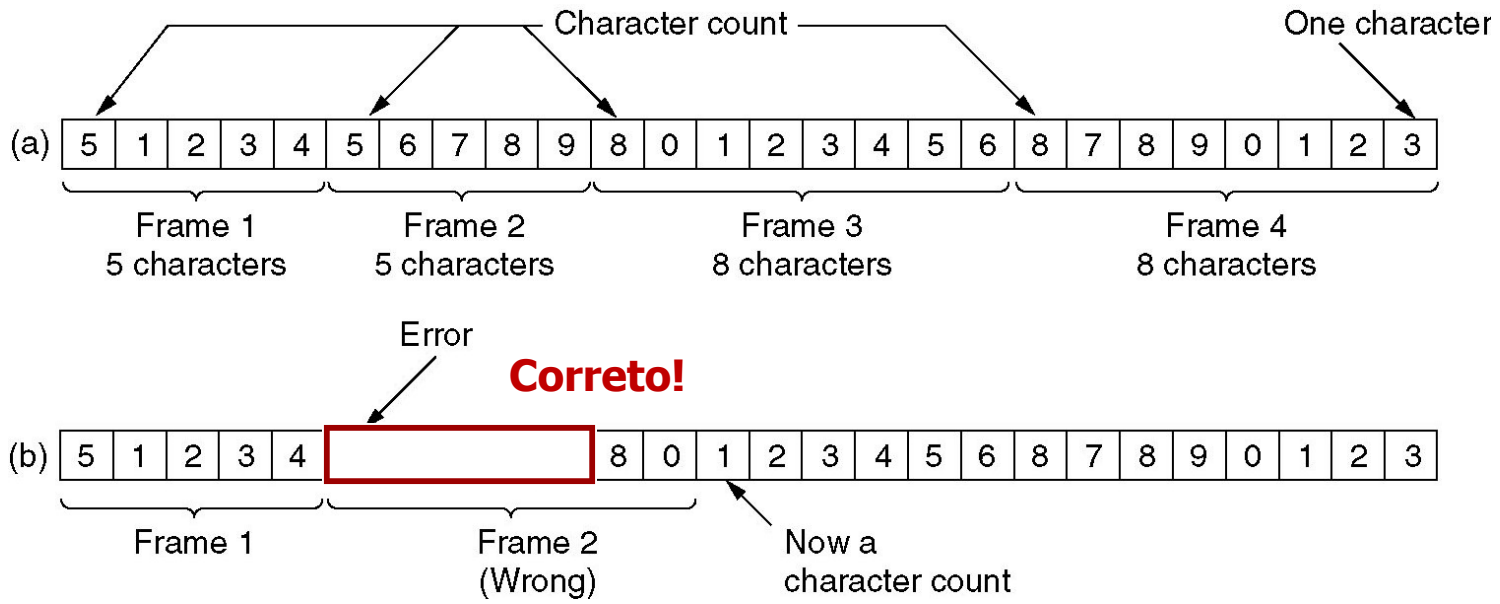
Exemplo de contagem de caracteres com um erro (fonte: Tanenbaum)



Enquadramento: contagem de caracteres

Instituto de computação

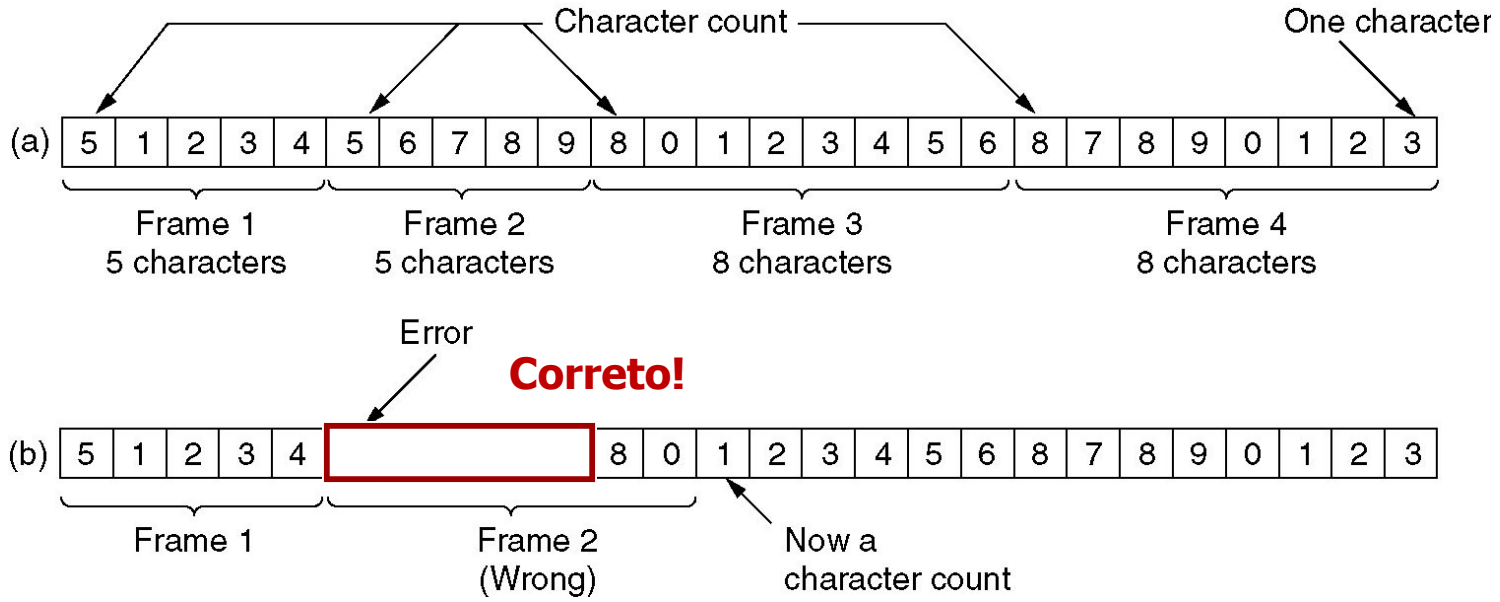
Exemplo de contagem de caracteres com um erro (fonte: Tanenbaum)



Enquadramento: contagem de caracteres

Instituto de computação

Exemplo de contagem de caracteres com um erro (fonte: Tanenbaum)



Contagem de caracteres é pouco usada!

Enquadramento: bytes de *flags*

Instituto de computação

- Soluciona o problema de ressincronização após um erro
- Quadro começa e termina com bytes especiais
  - Chamados de bytes de *flags*
  - **Delimitadores** de início e de fim de quadro
- Receptor identifica dois bytes de flag consecutivos
  - Recuperar a sincronização

Enquadramento: bytes de *flags*

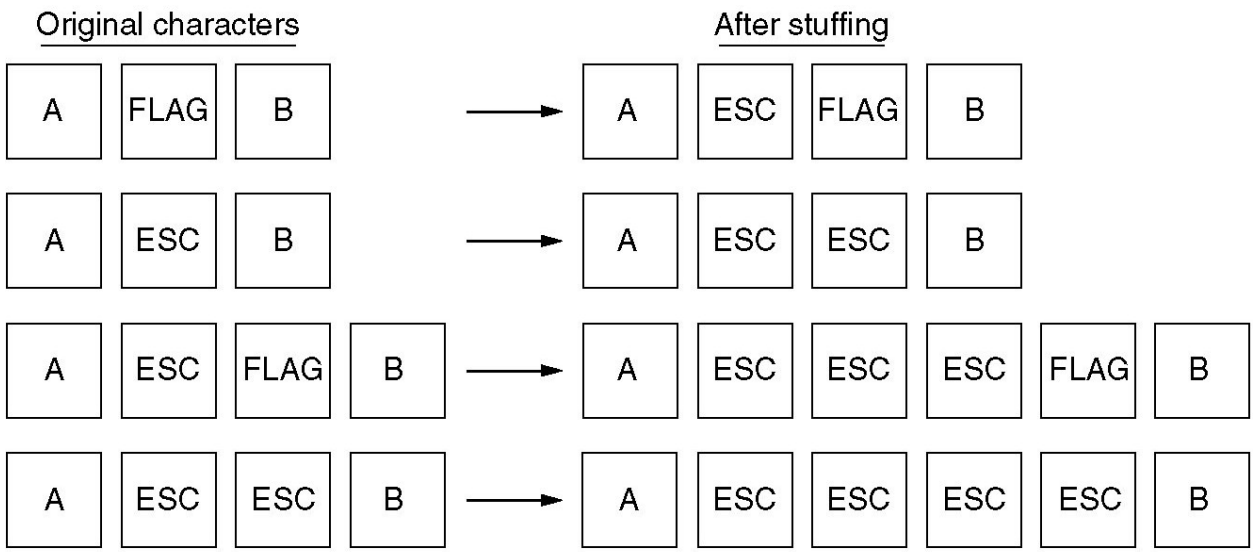
Instituto de computação

- Problema
  - *Payload* pode conter os bytes de *flags*
    - Se for composto por dados binários
- Solução
  - Transmissor da camada de enlace introduz um caractere de escape especial (ESC) antes de cada byte de *flag* "acidental" nos dados
    - Técnica chamada inserção de octetos ou inserção de caracteres
    - Usada no protocolo PPP

Enquadramento: bytes de *flags*

Instituto de computação

Sequências de quadros com octetos de flags (fonte: Tanenbaum)

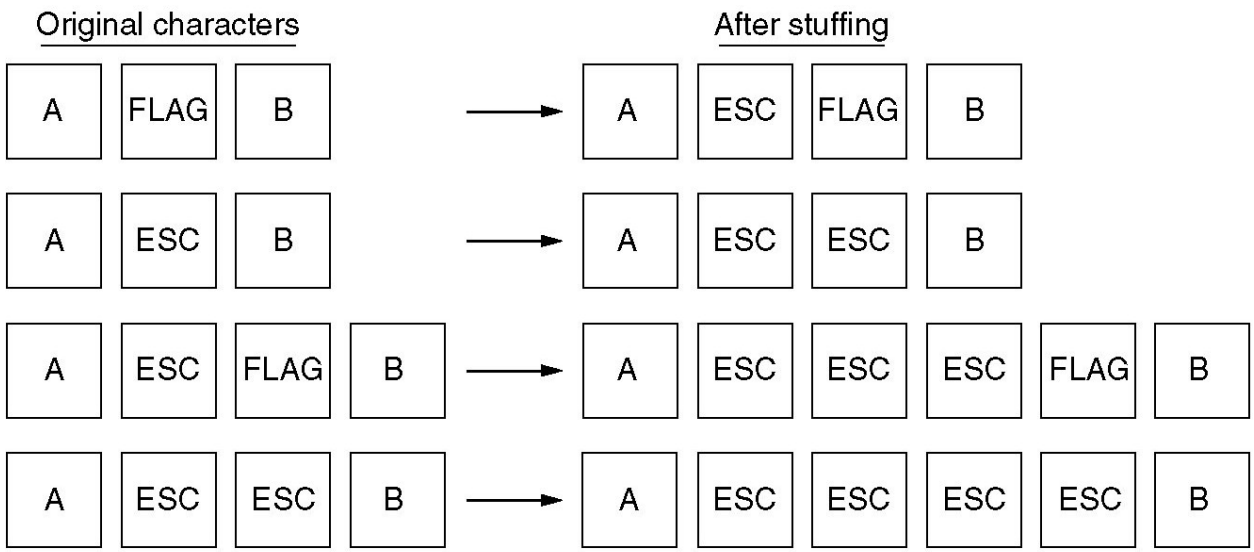


Não há dois *flags* seguidos na carga útil, somente no início e fim de um quadro

Enquadramento: bytes de *flags*

Instituto de computação

Sequências de quadros com octetos de flags (fonte: Tanenbaum)



Problema: depende do uso de caracteres de 8 bits

Enquadramento: *flags* iniciais e finais

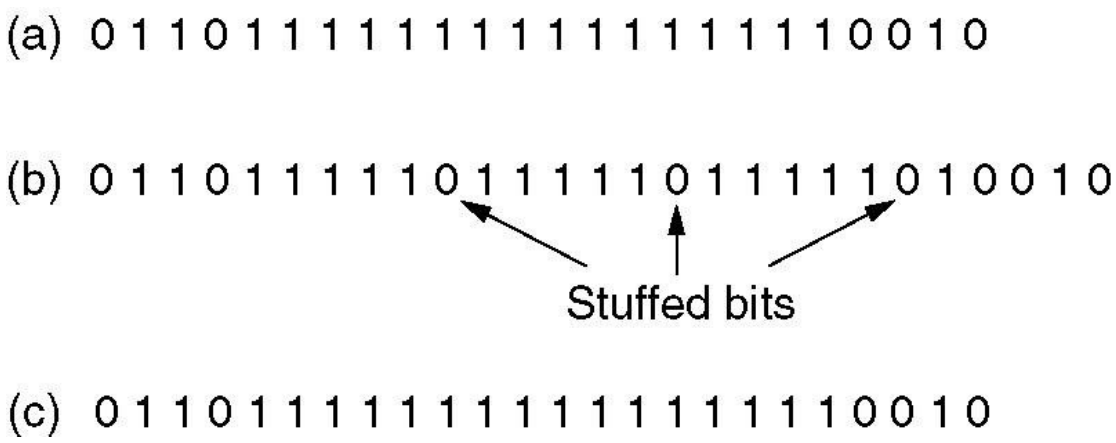
Instituto de computação

- Vantagem
  - Dados podem ter um número arbitrário de bits
- Cada quadro começa e termina com um padrão de bits
- Exemplo:
  - Delimitador → 01111110
  - Transmissor
    - Quando encontra cinco bits 1 consecutivos nos dados insere um bit 0, após a sequência
      - Inserção de bits → mesma finalidade do ESC
  - Receptor
    - Ao receber cinco bits 1 seguidos por um bit 0, remove o bit 0

Enquadramento: *flags* iniciais e finais

Instituto de computação

Exemplo de inserção de bits (fonte: Tanenbaum)



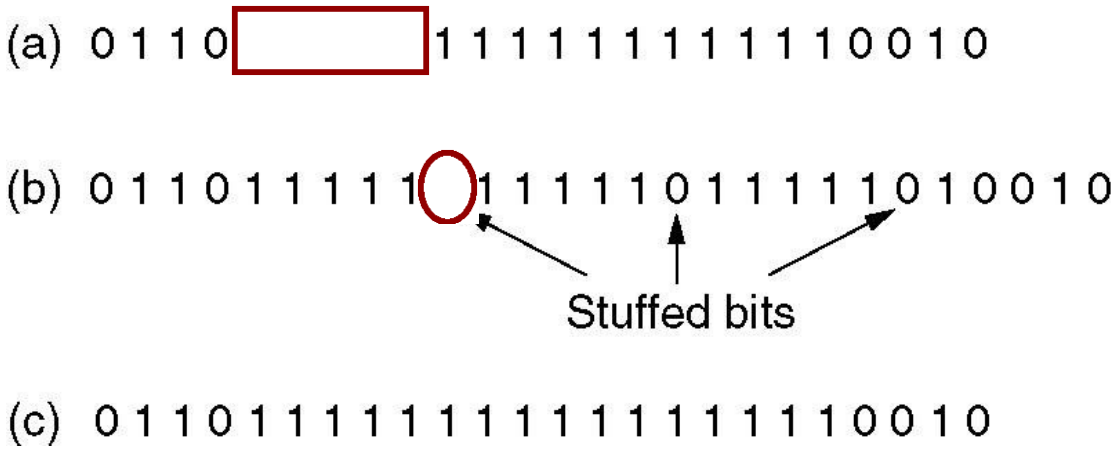
(a) Dados originais (b) Dados transmitidos

(c) Dados recebidos após a remoção dos bits

Enquadramento: *flags* iniciais e finais

Instituto de computação

Exemplo de inserção de bits (fonte: Tanenbaum)



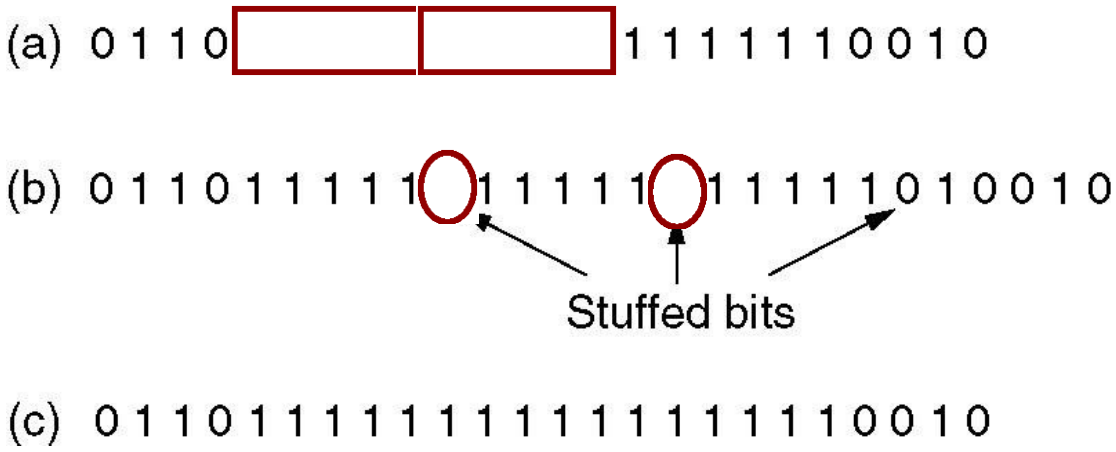
(a) Dados originais (b) Dados transmitidos

(c) Dados recebidos após a remoção dos bits

Enquadramento: *flags* iniciais e finais

Instituto de computação

Exemplo de inserção de bits (fonte: Tanenbaum)



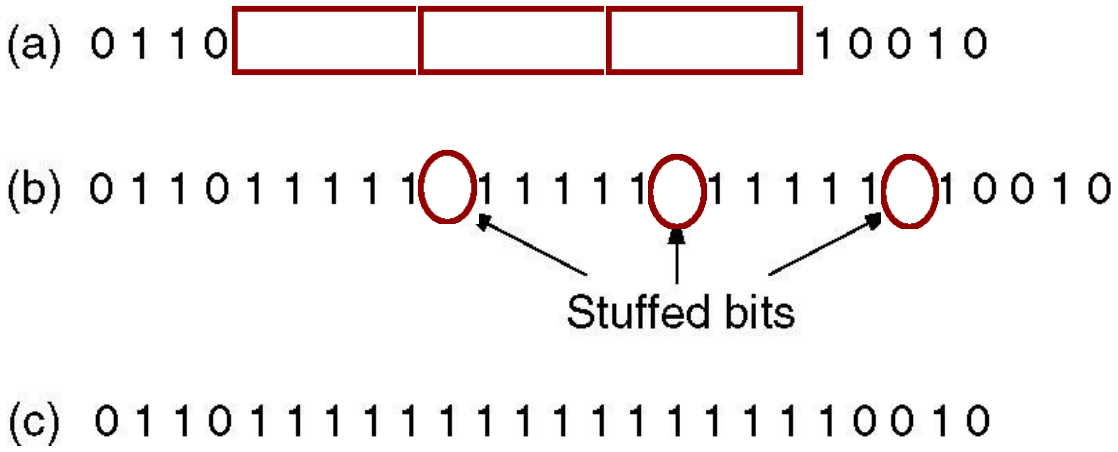
(a) Dados originais (b) Dados transmitidos

(c) Dados recebidos após a remoção dos bits

Enquadramento: *flags* iniciais e finais

Instituto de computação

Exemplo de inserção de bits (fonte: Tanenbaum)



(a) Dados originais (b) Dados transmitidos

(c) Dados recebidos após a remoção dos bits

Enquadramento: *flags* iniciais e finais

Instituto de computação

- Se o receptor perder a sincronização
  - Basta procurar pelo padrão de bits

Controle de Fluxo

Instituto de computação

- Um nó quer transmitir a uma taxa maior do que a taxa que o outro pode receber

- Duas abordagens mais comuns
  - Controle de fluxo baseado em realimentação
  - Controle de fluxo baseado na velocidade
    - Mecanismo interno limita a velocidade com que os transmissores podem enviar os dados
    - Não usa realimentação do receptor
    - Não utilizado na camada de enlace

Controle de Erros

Instituto de computação

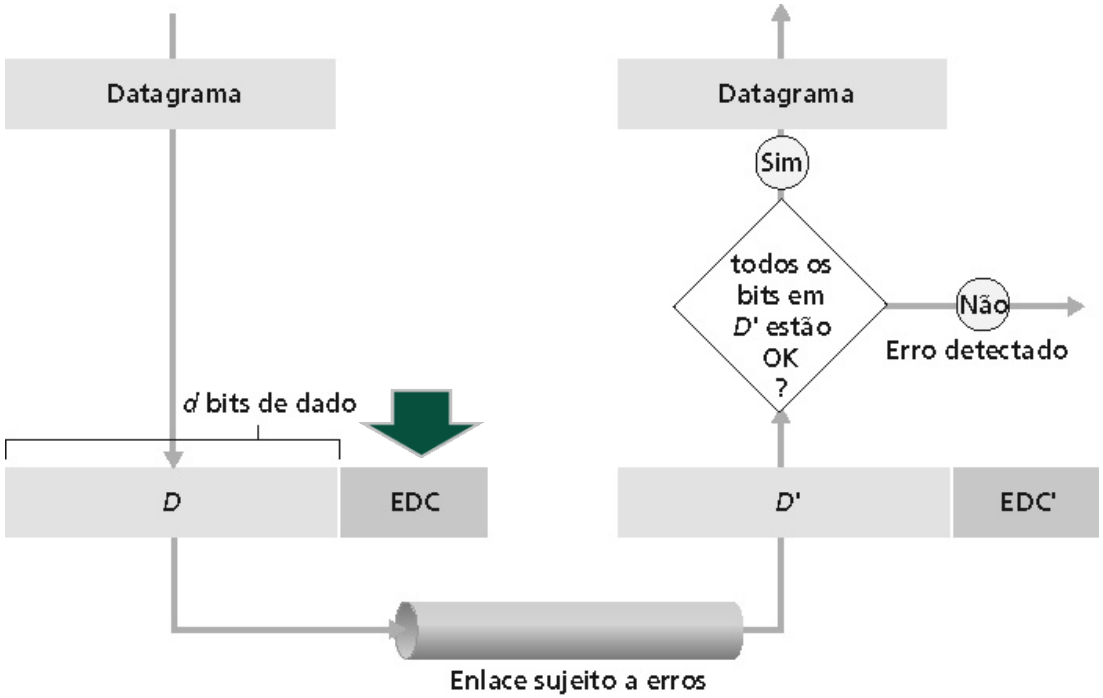
- Como garantir que os quadros enviados foram recebidos ordenadamente?
  - Mais comum é dar ao transmissor alguma realimentação sobre o que está se passando do outro lado
    - Reconhecimentos positivos e negativos
  - Além disso usam-se temporizadores
    - Espera pela confirmação durante um tempo
  - Números de sequência também são usados
    - Várias cópias do mesmo quadro podem ser recebidas
      - Ex.: Reconhecimentos perdidos

Detecção e Correção de Erros

Instituto de computação

- **Nível de bits**
- Erros de transmissão frequentes
  - *Loops* locais
  - Enlaces sem-fio
- Erros tendem a ocorrer em rajadas
  - Vantagem
    - Podem danificar poucos quadros
  - Desvantagem
    - Dificultam a correção dos erros
- Ponto-chave: enviar **informações redundantes** para detectar e corrigir erros

Detecção de Erros

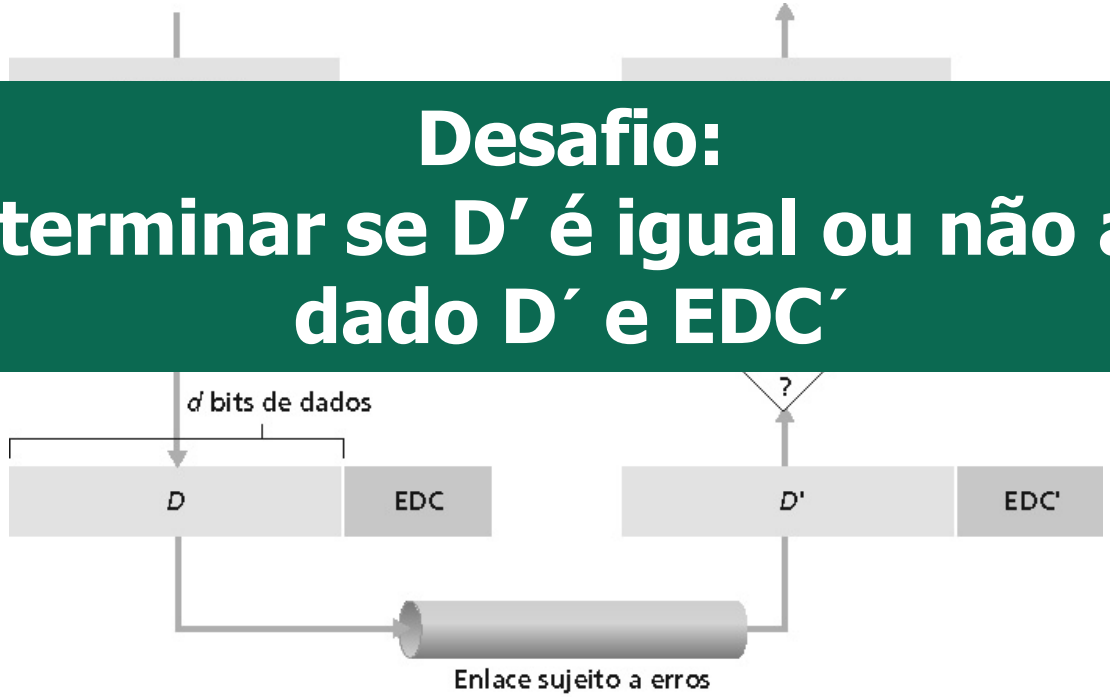


EDC= bits de Detecção e Correção de Erros (redundância)
D = dados protegidos por verificação de erros, podem incluir alguns campos do cabeçalho

• Quanto maior o campo EDC melhor será a capacidade de detecção e correção de erros

Detecção de Erros

Desafio:
Determinar se D' é igual ou não a D, dado D´ e EDC´



EDC= bits de Detecção e Correção de Erros (redundância)
D = dados protegidos por verificação de erros, podem incluir alguns campos do cabeçalho

• Quanto maior o campo EDC melhor será a capacidade de detecção e correção de erros

Detecção e Correção de Erros

Instituto de computação

- Códigos de **detecção** de erros
  - Menos redundância para deduzir um erro
  - Bons para enlaces confiáveis
    - Ex.: enlaces de fibra
    - Retransmissão "mais confiável"

Detecção e Correção de Erros

Instituto de computação

- Códigos de **correção** de erros
  - Mais redundância para deduzir quais dados foram transmitidos
  - Bons para enlaces pouco confiáveis
    - Ex: enlaces sem-fio
    - Retransmissão pode conter erros
  - Em geral, chamada de Correção Antecipada de Erros (*Forward Error Correction* – FEC)
    - Vantagem: antecipar a decisão

Detecção e Correção de Erros

Instituto de computação

- Quadro com d bits de dados e r bits de redundância
  - Tamanho total n bits → $n = d + r$
- A unidade de n bits é chamada **palavra** de código de n bits
- Número de posições de bits que duas palavras diferem entre si é chamado de **distância** (de Hamming)
  - 0011 e 0000 tem distância igual a 2
    - 0011 XOR 0000 = 11 → 2 bits em 1 → distância = 2

Detecção e Correção de Erros

Instituto de computação

- Se duas palavras de código estiverem a uma distância l uma da outra
  - É necessário corrigir l erros para converter uma na outra
- Em geral todas as 2^d mensagens de dados são válidas
- Mas nem todas as 2^n palavras de código possíveis são usadas
- Pode-se elaborar uma lista contendo todas as palavras válidas e localizar duas palavras de código cuja distância é mínima
  - Distância de Hamming do código completo

Detecção e Correção de Erros

Instituto de computação

- Detecção e correção de erros dependem da distância de Hamming do código completo
  - Para detectar x erros → código de distância $l = x + 1$
    - Não há como x erros de bits transformarem uma palavra de código válida em outra válida
  - Para corrigir x erros → código de distância $l = 2x + 1$
    - Palavras de código válidas estarão tão distantes que, mesmo com l alterações, a palavra de código original continuará mais próxima do que qualquer outra

Detecção e Correção de Erros

Instituto de computação

- Exemplo
  - Código contendo as seguintes palavras:
    - 0000000000, 0000011111, 1111100000 e 1111111111
  - Distância igual a 5 ($5 = 2x + 1 \rightarrow x = 2$)
    - Pode corrigir erros duplos
  - Se detecta 0000000111 (é um erro duplo)
    - Original deve ser 0000011111
  - Se detecta 0000000111 e foi transmitido 0000000000 (é um erro triplo) $\rightarrow$ erro não corrigido de maneira correta

Verificação de Paridade

Instituto de computação

- Código simples de detecção de erros
- **Bit de paridade acrescentado aos dados**
  - Escolhido de forma que o número de bits 1 da palavra de código seja par (paridade par) ou ímpar (paridade ímpar)
- Receptor conta quantos bits 1 a palavra possui
  - Se é usada a paridade par e contou um número ímpar de 1s $\rightarrow$ Ocorreu um número ímpar de erros
  - Número par de erros $\rightarrow$ não são detectados
- Código com um único bit de paridade tem uma distância igual a 2
  - Pode detectar erros isolados

Verificação de Paridade

Instituto de computação

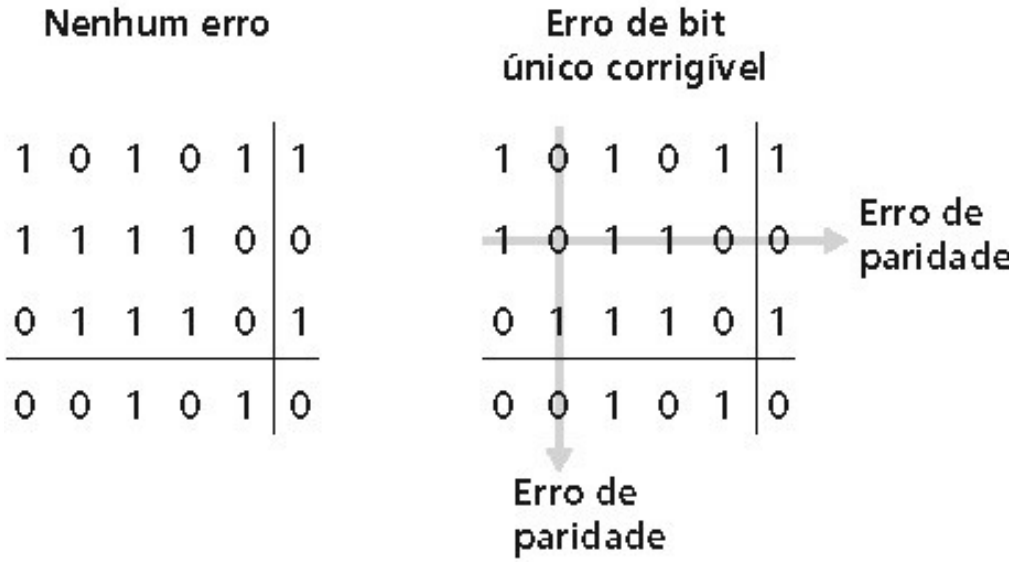
- Exemplo
  - 1011010 enviado com paridade par $\rightarrow$ 10110100
  - 1011010 enviado com paridade ímpar $\rightarrow$ 10110101
- Problema: erros ocorrem geralmente em rajada
  - Paridade com um bit não é suficiente
  - Solução $\rightarrow$ aumenta-se o número de bits de paridade

Verificação de Paridade

Instituto de computação

- Paridade bidimensional
  - Paridade de linha
  - Paridade de coluna
  - Paridade dos bits de paridade
- Pode detectar e corrigir erros isolados
- Pode detectar erros duplos

Exemplo de paridade bidimensional (fonte: Kurose)



Detecção e correção de erros: Hamming

Instituto de computação

- Bits da palavra de código são numerados consecutivamente
  - Bit 1 na extremidade esquerda
- Bits que são potências de 2 são bits de verificação
- Outros bits são preenchidos com os *d* bits de dados

Detecção e correção de erros: Hamming

Instituto de computação

- Bits de verificação calculados usando a paridade par ou ímpar dos bits que verificam
  - Exemplo com código de mais de 12 bits
    - Bit 1 $\rightarrow$ paridade dos bits 3, 5, 7, 9, 11, ...
    - Bit 2 $\rightarrow$ paridade dos bits 3, 6, 7, 10, 11, ...
    - Bit 4 $\rightarrow$ paridade dos bits 5, 6, 7, 12, ...
    - Bit 8 $\rightarrow$ paridade dos bits 9, 10, 11, 12, ...
  - Decomposição
    - 3 = 1 + 2; 5 = 1 + 4; 6 = 2 + 4; 7 = 1 + 2 + 4;
    - 9 = 1 + 8; 10 = 2 + 8; 11 = 1 + 2 + 8
  - Exemplo de Código (11,7): 1000001 $\rightarrow$ 00100001000

Detecção e correção de erros: Hamming

- Exemplo: Código com palavra de 11 bits
  - Mensagem de 7 bits (# bits - # potências de 2): 1001000

			1		0	0	1		0	0	0
posição	1	2	3	4	5	6	7	8	9	10	11

  - Calculando a paridade
    - Bit 1 → paridade de 3,5,7,9,11
    - Bit 2 → paridade de 3,6,7,10,11
    - Bit 4 → paridade de 5,6,7
    - Bit 8 → paridade de 9,10,11

posição	1	2	3	4	5	6	7	8	9	10	11

Detecção e correção de erros: Hamming

Instituto de computação

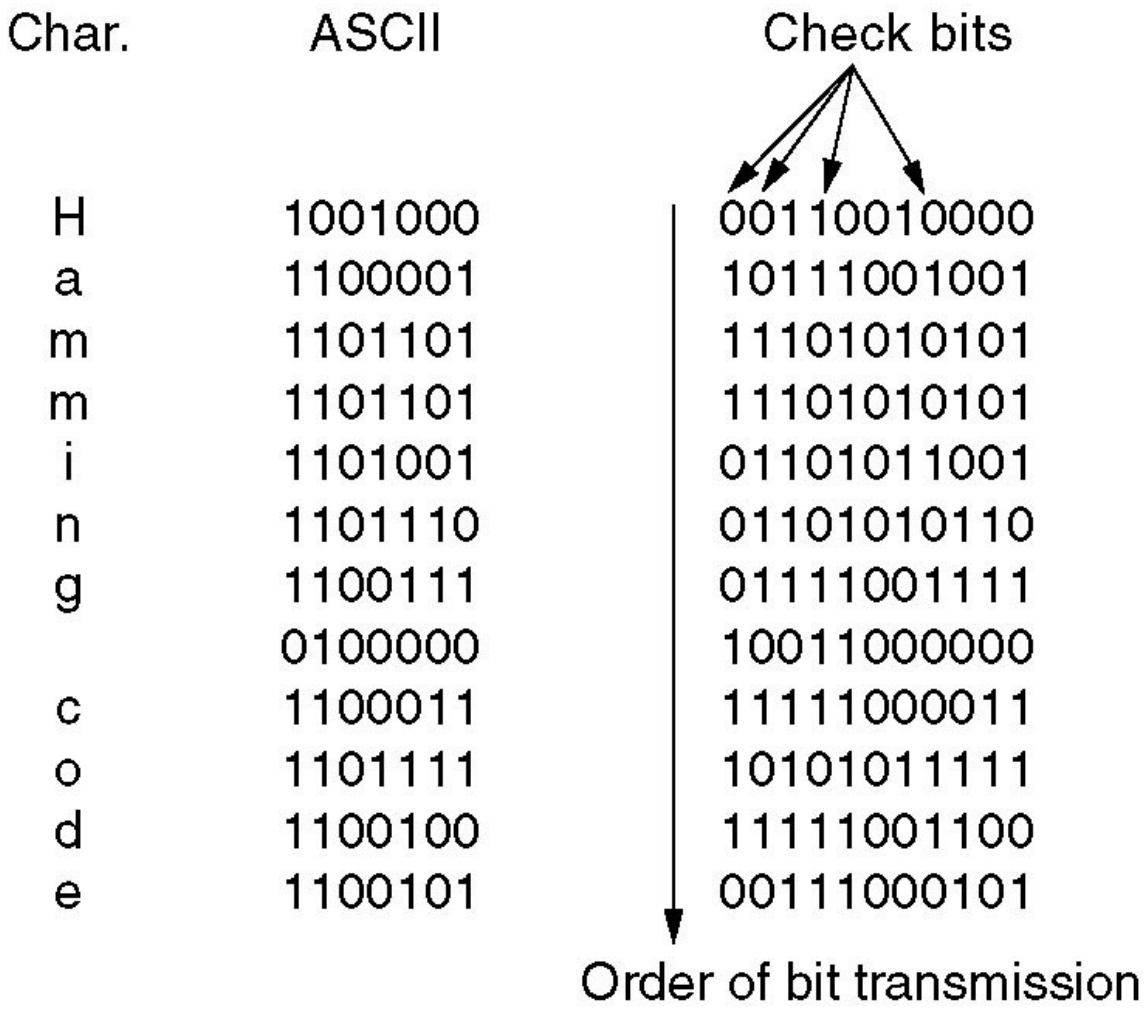
- Receptor
  - Inicializa um contador com zero
  - Examina cada bit de verificação k (k = 1, 2, 4, 8, ...) para confirmar se a paridade está correta
    - Caso não esteja, k é somado ao valor do contador
  - Contador indica zero após o exame de todos os bits de verificação
    - Palavra aceita como válida
  - Se o contador não é zero
    - Ele contém o número do bit errado
    - Ex.: Se os bits de verificação 1, 2 e 8 estiverem incorretos, o bit invertido será igual a 11 (o único verificado por 1, 2 e 8)

Detecção e correção de erros: Hamming

Instituto de computação

- Só pode corrigir erros simples
- Para corrigir erros em rajada
  - Sequência de *k* palavras consecutivas é organizada como uma matriz, com uma palavra de código por linha
  - Em vez de transmitir os dados uma palavra de código por vez, da esquerda para a direita, transmite-se uma coluna por vez, começando pela coluna mais à esquerda
  - Receptor reconstrói a matriz, uma coluna por vez
  - Se ocorrer um erro em rajada com a extensão *k*, no máximo um bit de cada uma das *k* palavras de código será afetado → bloco restaurado

Exemplo para corrigir erros em rajada (fonte: Tanenbaum)



Detecção e Correção de Erros: Soma de Verificação (*checksum*)

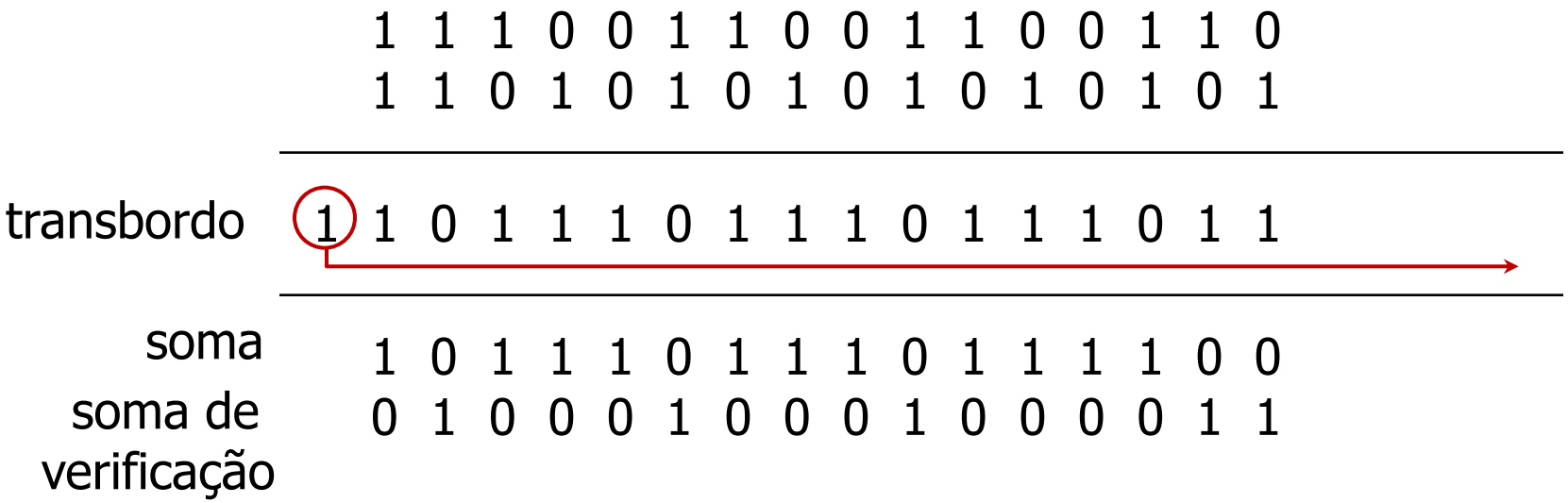
Instituto de computação

- Método simples
  - Normalmente implementado em *software*
- Bits de dados tratados como uma sequência de números inteiros de *k* bits
- Soma-se esses números inteiros (em complemento a 1) e usa-se o total como bits de detecção de erros

Detecção e Correção de Erros: Soma de Verificação (*checksum*)

Instituto de computação

- Observação
  - Ao adicionar números, o transbordo (vai um) do bit mais significativo deve ser adicionado ao resultado
- Exemplo: adição de dois inteiros de 16-bits



Detecção e Correção de Erros: Soma de Verificação (*checksum*)

Instituto de computação

- Método simples
  - Normalmente implementado em *software*
- Bits de dados tratados como uma sequência de números inteiros de *k* bits
- Soma-se esses números inteiros (em complemento a 1) e usa-se o total como bits de detecção de erros
- Receptor recalcula o *checksum* dos dados recebidos
  - Se o resultado contém algum bit 0 $\rightarrow$ erro
- Usado no TCP, no UDP e no IP

Detecção e Correção de Erros: CRC

Instituto de computação

- Código de redundância cíclica (*Cyclic Redundancy Check*) ou código polinomial
- Mais complexo
  - Geralmente implementado em hardware
- Trata uma sequência de bits como representações de polinômios com coeficientes 0 e 1
- Quadro de k bits $\rightarrow$ k termos, de x^{k-1} até x^0
  - Polinômio de grau k-1
  - Ex.: 11001 → $1x^4 + 1x^3 + 0x^2 + 0x^1 + 1x^0 = x^4 + x^3 + 1$

Detecção e Correção de Erros: CRC

- Aritmética polinomial feita em **módulo 2**, sem transportes para adição nem empréstimos para subtração
  - Adição e subtração são **idênticas** à operação ou-exclusivo

```
  10011011      11110000
+ 11001010    - 10100110
----------    ----------
  01010001      01010110
```

- Divisão similar à binária
  - Um divisor "cabe em" um dividendo se o dividendo tem a mesma quantidade de bits do divisor

Detecção e correção de erros: CRC

Instituto de computação

- Transmissor e receptor devem concordar em relação ao uso de um polinômio gerador G(x)
  - Tanto o bit de mais alta ordem quanto o bit de mais baixa ordem devem ser iguais a 1
  - Polinômio gerador pode ser escolhido de acordo com a probabilidade de ocorrerem erros
- Quadro de *d* bits corresponde a D(x)
- Grau D(x) > Grau G(x)

Detecção e correção de erros: CRC

Instituto de computação

- CRC acrescentado ao final do quadro de forma que o quadro verificado seja divisível por G(x)
  - Sequência de verificação de quadro (*Frame Check Sequence* – FCS)
- Ao receber o quadro verificado, o receptor tentará dividi-lo por G(x)
  - Se o resto é diferente de zero $\rightarrow$ erro

Detecção e correção de erros: CRC

Instituto de computação

- Algoritmo
  - Seja *r* o grau de G(x)
  - Acrescentar *r* bits à extremidade de mais baixa ordem de D(x)
    - Polinômio $x^r D(x)$
  - Dividir a sequência de bits correspondente a $x^r D(x)$ pela sequência correspondente por G(x)
  - Subtrair o resto da sequência correspondente a $x^r D(x)$
  - Resultado é o quadro verificado que deverá ser transmitido
    - Polinômio T(x)

Detecção e correção de erros: CRC

- Idéia com dividendo A = 7 e divisor B = 3
- Transmissor
  1. Dividir A por B

$$\begin{array}{rr|l} 7 & & 3 \\ \hline \text{resto } 1 & & 2 \end{array}$$

  2. **Subtrair** o resto de A → 7 – 1 =6
  3. O resultado é a mensagem C a ser transmitida → 6

Detecção e correção de erros: CRC

- Idéia com dividendo A = 7 e divisor B = 3
- Receptor
  1. Recebe C’ (canal pode ter erros)
  2. Divide C’ por B
  3. Caso resto=0 → não há erros

	6	3
resto	0	2

Detecção e correção de erros: CRC

- Exemplo: D(x) = 111 → $x^2 + x + 1$, G(x) = 11 → $x + 1$
  - Grau de G(x) = ***r* = 1**
  - Acrescentar **1 bit** à extremidade de mais baixa ordem de D(x)
    - $x^r D(x)$ → 2.D(x) = 1110
  - Dividir a sequência de bits correspondente a $x^r D(x)$ pela sequência correspondente por G(x)

```
     101
   ------
11 | 1110
     11
     --
      01
      00
      --
       10
       11
       --
resto  1
```

Calculadora polinomial
http://www.ee.unb.ca/cgi-bin/tervo/calc.pl

Detecção e correção de erros: CRC

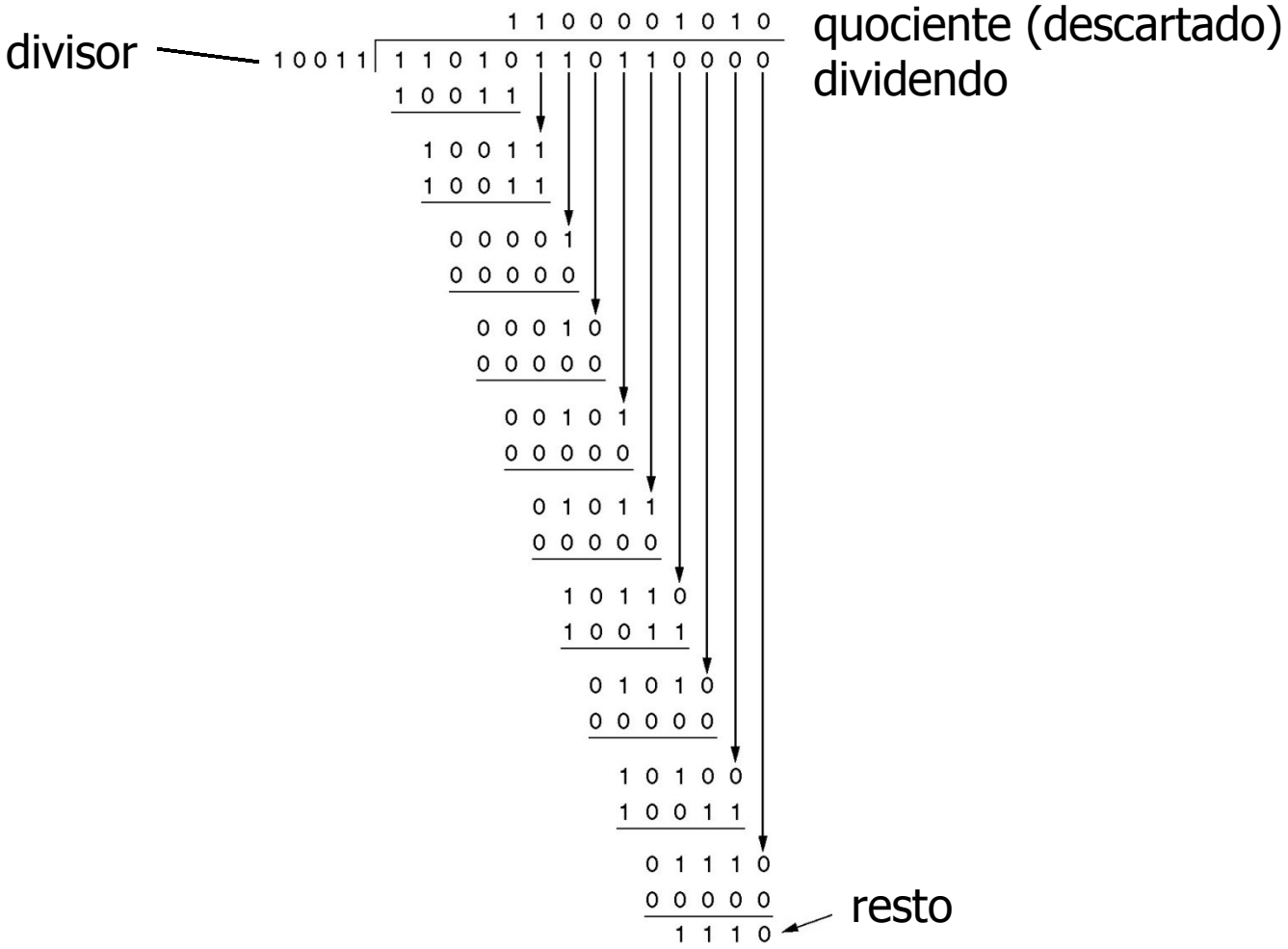
- Exemplo: D(x) = 111 → $x^2 + x + 1$, G(x) = 11 → $x + 1$
  - Subtrair o resto da sequência correspondente a $x^r D(x)$

$$\begin{array}{r} 1110 \\ -\ 0001 \\ \hline 1111 \end{array}$$

  - Resultado é o quadro verificado que deverá ser transmitido
    - T(x) = 1111

Outro exemplo de cálculo de CRC (fonte: Tanenbaum)

Frame : 1 1 0 1 0 1 1 0 1 1
Generator: 1 0 0 1 1
Message after 4 zero bits are appended: 1 1 0 1 0 1 1 0 1 1 0 0 0 0

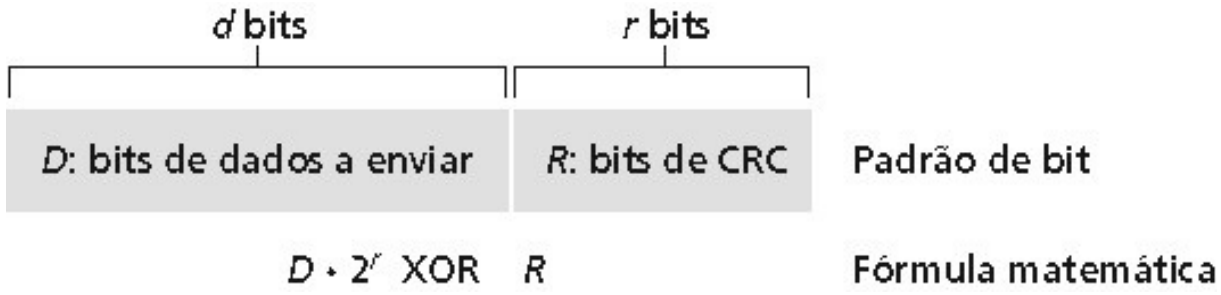


$d = 10$
$r = 4$

Transmitted frame: 1 1 0 1 0 1 1 0 1 1 1 1 1 0

Detecção e Correção de Erros: CRC – Outra Interpretação

- Dados (D) → seqüência de coeficientes de um polinômio (D)
- É escolhido um polinômio *Gerador*, (G), (grau(G) => r+1 bits)
  - Divide-se (módulo 2) o polinômio $D*2^r$ por G
  - Acrescenta-se o resto (R) a D
  - Observa-se que, por construção, a nova seqüência <D,R> agora é **exatamente divisível por G**
  - Receptor conhece G, divide <D,R> por G
    - **Caso o resto seja diferente de zero: detectado erro!**
  - Pode detectar os erros em rajadas menores do que r+1 bits
- Largamente usado na prática (ATM, HDLC)



Mais um Exemplo de CRC

Queremos:

$D \cdot 2^r$ XOR R = nG

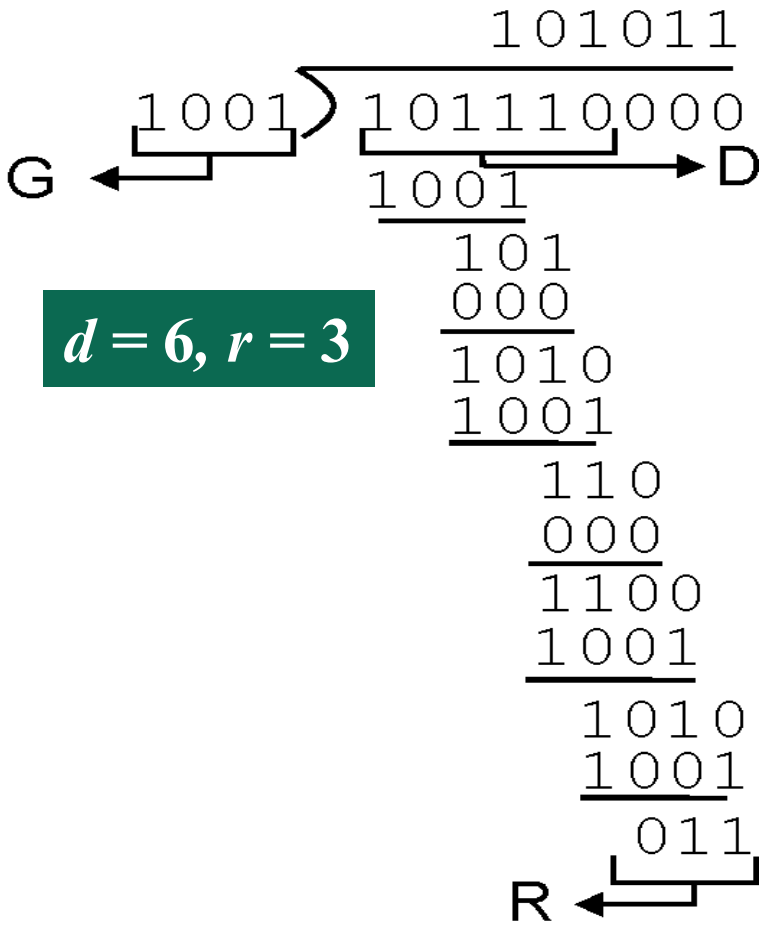
De forma equivalente:

$D \cdot 2^r$ = nG XOR R

de forma equivalente :

se dividirmos $D \cdot 2^r$ por G, teremos o resto R

$$R = \text{resto}\left(\frac{D \cdot 2^r}{G}\right)$$



$d = 6, r = 3$

Bits transmitidos 101110011

Detecção e correção de erros: CRC

Instituto de computação

- Usado em diversos padrões de redes locais e metropolitanas
- Exemplo de G(x) do IEEE 802
  - $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 +$ $+ x^4 + x^2 + x^1 + 1$

Instituto de computação

Implementação

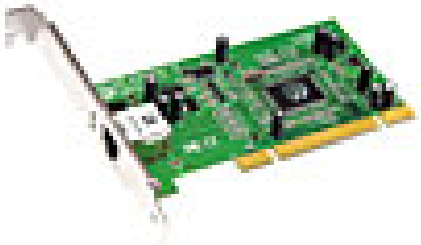
Implementação

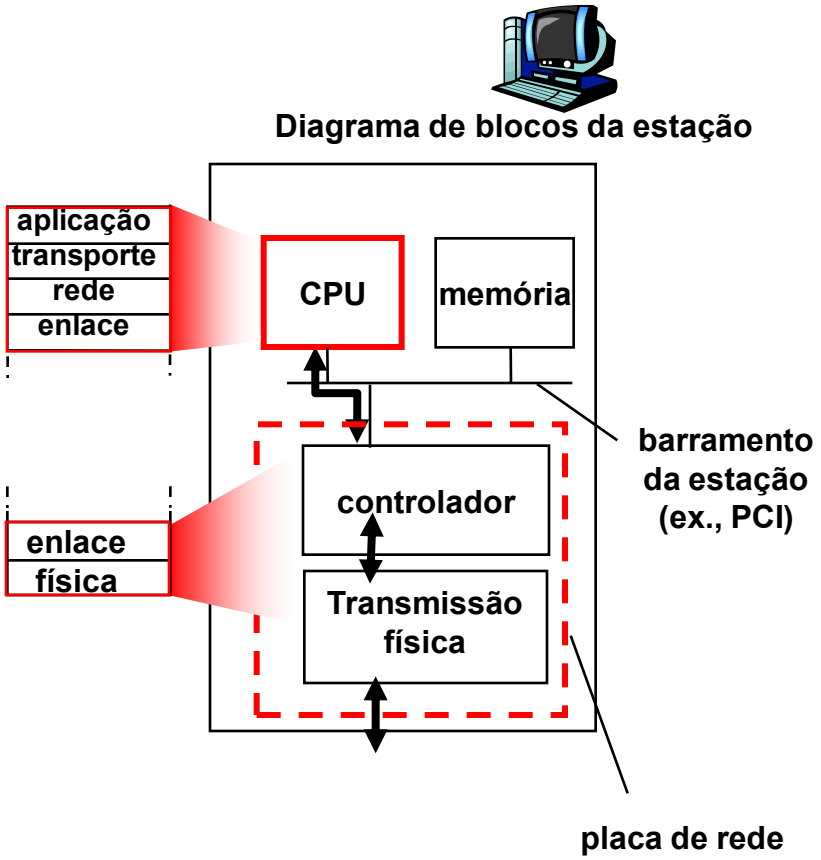
Instituto de computação

- A camada de enlace é implementada por cada um dos nós da rede
  - Cada um pode implementar uma tecnologia

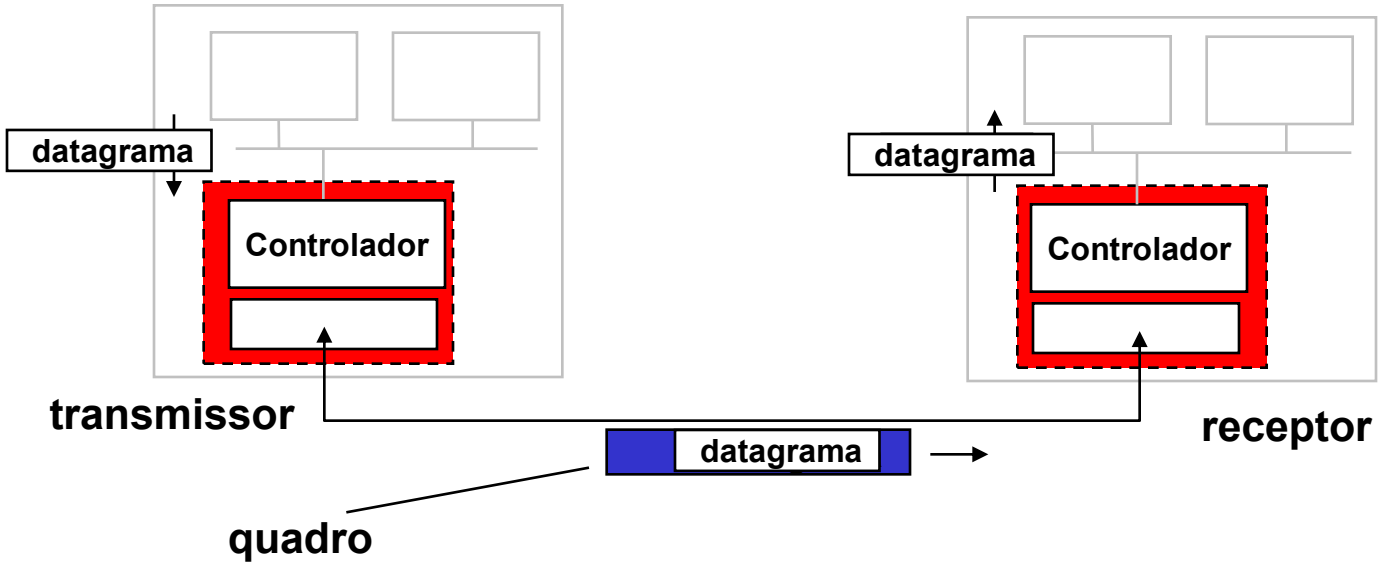
- É implementada no "adaptador" (NIC – *Network Interface Card*)
  - Exs: placa Ethernet, cartão PCMCIA, cartão 802.11
  - Também implementa a camada física
  - Está conectado ao barramento de sistema do nó
    - Ou integrada na placa mãe
  - É uma combinação de hardware, software e firmware

Implementação





Comunicação entre Adaptadores



- Lado transmissor
  - Encapsula o datagrama em um quadro
  - Adiciona bits de verificação de erro, transferência confiável de dados, controle de fluxo, etc.
- Lado receptor
  - Verifica erros, transporte confiável, controle de fluxo, etc.
  - Extrai o datagrama, passa-o para o nó receptor

Instituto de computação

Endereçamento

Endereços MAC

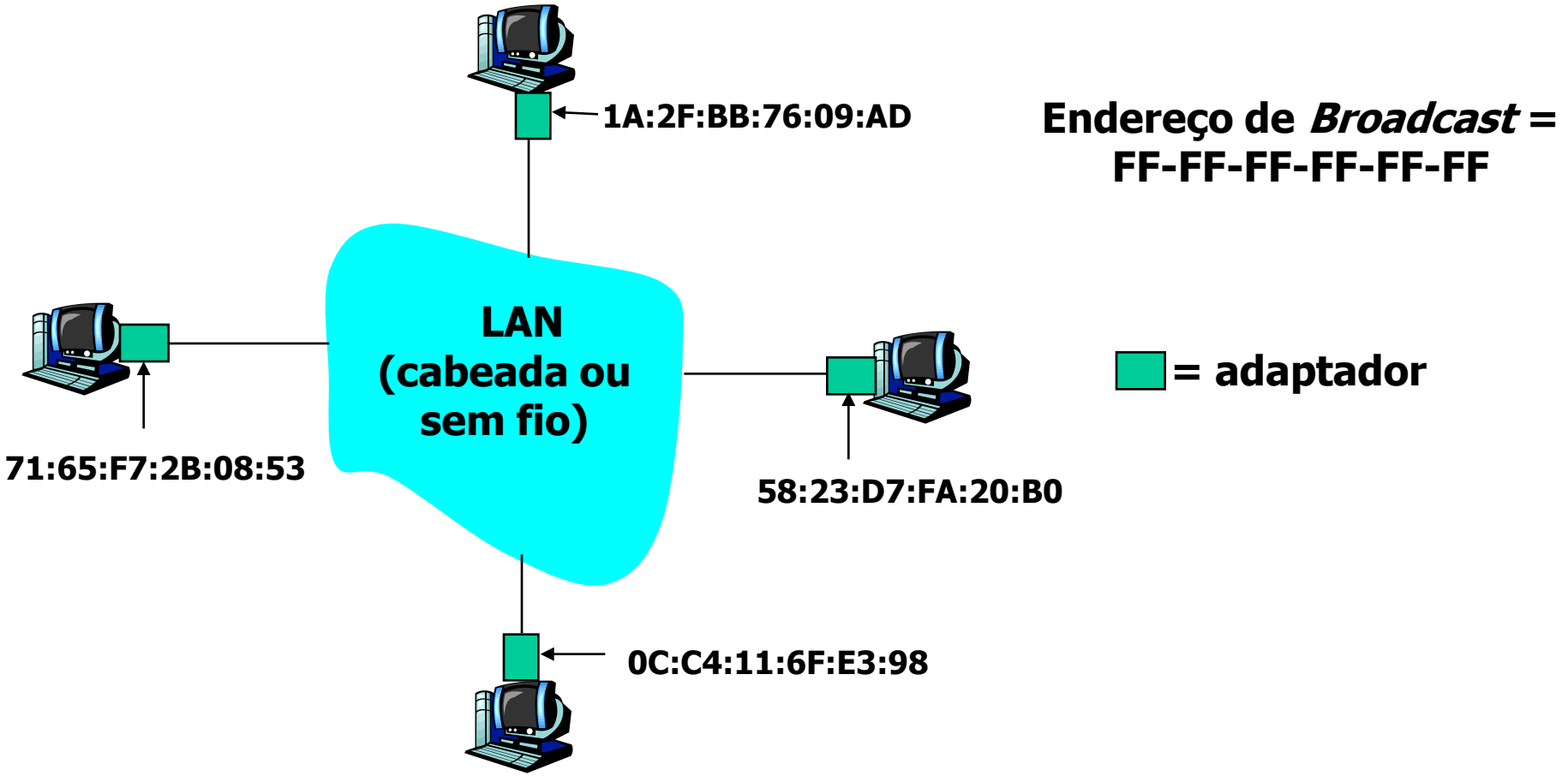
Instituto de computação

- Endereço IP de 32 bits
  - Endereços da camada de rede
  - Usado para levar o datagrama à sub-rede IP destino

- Endereço MAC (ou LAN, ou físico, ou Ethernet)
  - Levar o datagrama de **uma interface até outra interface** conectada fisicamente (na mesma rede)
  - 48 bits (para a maioria das redes)
    - Representados por 12 dígitos hexadecimais agrupados 2 a 2
      - Ex.: 1A:2F:BB:76:09:AD
  - Gravado na ROM do adaptador, ou configurado por software

Endereços MAC

Cada adaptador na LAN possui um **endereço MAC único**



Endereços MAC

Instituto de computação

- Alocação de endereços MAC gerenciada pelo IEEE
- Um fabricante compra uma parte do espaço de endereços
  - Garantir a unicidade
- Analogia
  - Endereço MAC
    - Como número do CPF
  - Endereço IP
    - Como endereço postal (CEP)

Endereços MAC

Instituto de computação

- Endereço MAC tem estrutura linear
  - Portabilidade
    - É possível mover um cartão LAN de uma LAN para outra
- Endereço IP hierárquico NÃO é portátil
  - Requer IP móvel, por exemplo
  - Depende da sub-rede IP à qual o nó está conectado

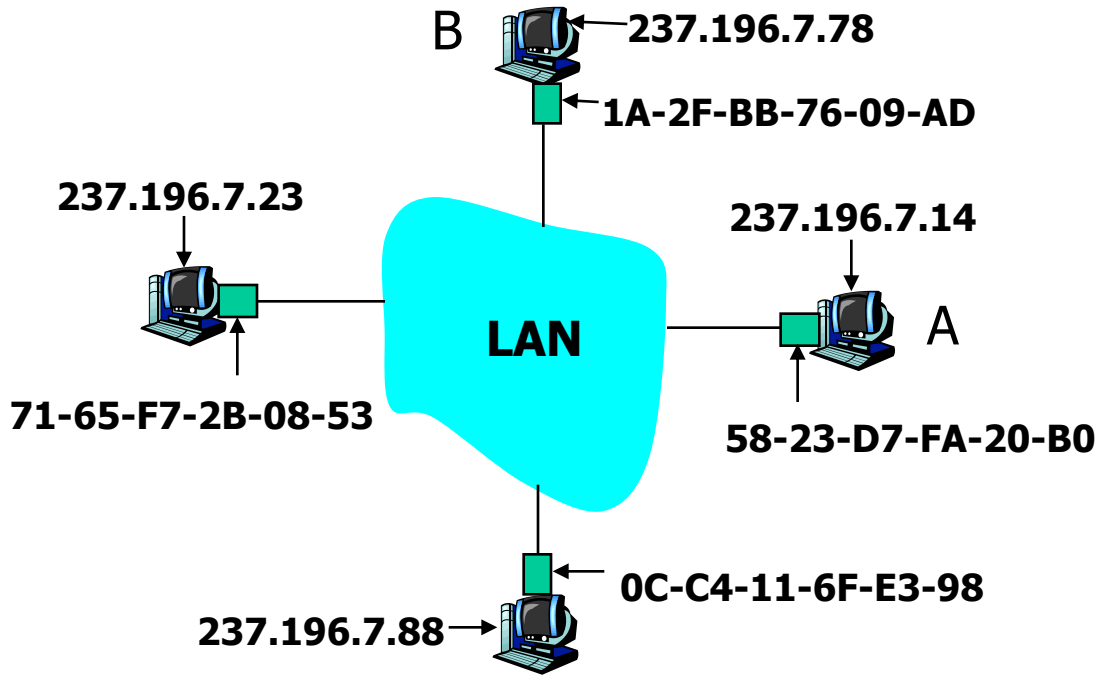
ARP

Instituto de computação

- Protocolo de resolução de endereços (*Address Resolution Protocol*)
- Descrito na RFC 826
- Faz a tradução de endereços IP para endereços MAC da maioria das redes IEEE 802
- Executado dentro da sub-rede
- Cada nó (estação ou roteador) possui uma tabela ARP
  - Contém endereço IP, endereço MAC e TTL
- Tabela ARP construída automaticamente

ARP

Como obter o endereço MAC de B a partir do endereço IP de B?



- Cada nó de uma LAN possui uma tabela ARP
- Tabela ARP: mapeamento de endereços IP/MAC para alguns nós da LAN

< endereço IP; endereço MAC; TTL>

  - TTL (*Time To Live*): tempo a partir do qual o mapeamento de endereços será esquecido (valor típico de 20 min)

Funcionamento na Mesma Rede

Instituto de computação

- **A** deseja enviar datagrama para **B**, e o endereço MAC de **B** não está na tabela ARP
- A **difunde** o pacote de solicitação ARP, que contém o endereço IP de B
  - Endereço MAC destino = FF-FF-FF-FF-FF-FF
  - Todas as máquinas na LAN recebem a consulta do ARP
- **B** recebe o pacote ARP, responde a **A** com o seu endereço MAC
- Quadro enviado para o endereço MAC (*unicast*) de **A**

Funcionamento na Mesma Rede

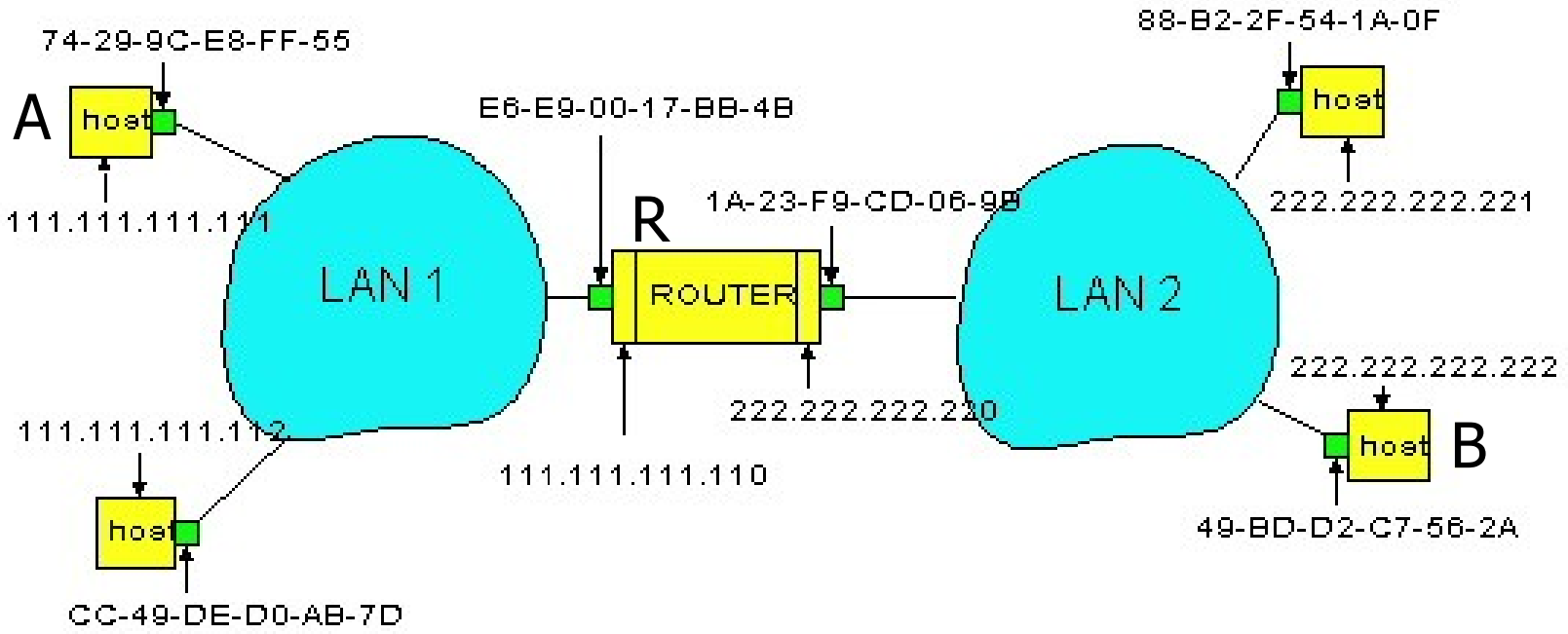
Instituto de computação

- Um *cache* (salva) o par de endereços IP-para-MAC na sua tabela ARP até que a informação expire
  - É "*soft state*"
    - Informação que expira (vai embora) a menos que seja renovada
  - Um nó pode responder a uma requisição com um endereço MAC que conhece
    - Não necessariamente o nó destino

- ARP é "*plug-and-play*"
  - Os nós criam suas tabelas ARP **sem a intervenção** do administrador da rede

Funcionamento entre Redes Diferentes

- Envio de datagrama de A para B através de R
- O Roteador R possui duas tabelas ARP
  - Uma para cada rede local



Funcionamento entre Redes Diferentes

Instituto de computação

- *A* cria o datagrama com endereço IP fonte *A* e destino *B*
- *A* consulta a tabela de roteamento e obtém *R* como próximo salto
- *A* usa o ARP para obter o endereço MAC de *R*
- *A* cria um quadro com endereço MAC de destino *R* e o datagrama de *A* para *B* na carga útil
- Adaptador de *A* envia o quadro para *R*
- Adaptador de *R* recebe o quadro

Funcionamento entre Redes Diferentes

Instituto de computação

- *R* remove o datagrama IP do quadro Ethernet e verifica que é destinado a *B*
- *R* consulta a tabela de roteamento
- *R* usa o ARP para obter o endereço MAC de *B*
- *R* cria o quadro contendo o datagrama de *A* para *B*
- Adaptador de *R* envia o quadro para *B*
- Adaptador de *B* recebe o quadro

Ferramentas ARP

Instituto de computação

- Saída do **tcpdump**

```
[root@masq-gw]# tcpdump -i eth0 \( arp \)
tcpdump: listening on eth0
0:80:c8:f8:4a:51 ff:ff:ff:ff:ff:ff 42: arp who-has 192.168.99.254 tell 192.168.99.35
0:80:c8:f8:5c:73 0:80:c8:f8:4a:51 60: arp reply 192.168.99.254 is-at 0:80:c8:f8:5c:73
```

Ferramentas ARP

- Ferramenta **`arp`**
  - Mostra a tabela ARP de uma estação

```
[igor@flechas ~]#arp
Address                  HWtype  HWaddress           Flags Mask  Iface
imbuquinha.midiacom.uff  ether   00:22:6b:7c:9c:df   C           br0
xara.midiacom.uff.br     ether   00:0f:3d:79:d8:04   C           br0
jardimicarai.midiacom.u  ether   00:1d:09:ff:20:b4   C           br0
```

Ferramentas ARP

Instituto de computação

- Arping
  - "Ping" da camada de enlace

```
ARPING 192.168.0.1 from 192.168.0.10 eth0
Unicast reply from 192.168.0.1 [00:01:80:38:F7:4C]  0.510ms
Unicast reply from 192.168.0.1 [00:01:80:38:F7:4C]  0.601ms
Unicast reply from 192.168.0.1 [00:01:80:38:F7:4C]  0.610ms
Unicast reply from 192.168.0.1 [00:01:80:38:F7:4C]  0.605ms
Sent 4 probes (1 broadcast(s))
Received 4 response(s)
```

Instituto de computação

Elementos de Interconexão

Repetidor

Instituto de computação

- Nível físico
- Tem um número pequeno de interfaces
- Recebe, conforma (recupera a forma do sinal original), amplifica e retransmite os bits de uma interface para todas as outras

Hub

Instituto de computação

- Nível físico
- É um repetidor
- **Repete** os bits de uma porta para **todas** as outras
- Segmentos da rede formam um **único domínio de colisão**
  - Domínio de colisão é uma única rede com CSMA/CD na qual haverá colisão se duas estações da rede transmitirem ao mesmo tempo
- Geralmente não pode conectar segmentos da rede operando em diferentes taxas
  - Esse caso poderia ser implementado usando dois hubs que operam em velocidades diferentes conectados internamente por um comutador de duas portas

Ponte (*bridge*)

Instituto de computação

- Nível enlace
- Tem um pequeno número de interfaces
- Usam o endereço MAC de destino para encaminhar e filtrar quadros
- Cada **segmento** de rede é um **domínio de colisão separado**
- Pode conectar segmentos de rede operando em diferentes taxas
- Pode conectar segmentos de rede diferentes

Ponte (*bridge*)

Instituto de computação

- Conceitos
  - Filtragem
    - Capacidade da ponte decidir se um quadro será repassado para alguma interface ou descartado
  - Repasse
    - Capacidade de determinar as interfaces para as quais um quadro deve ser repassado e fazê-lo

Ponte (*bridge*)

Instituto de computação

- Tabela de comutação usada no repasse
  - Se o endereço de destino está na tabela e a interface não é a mesma de onde veio, transmite para a interface correspondente
  - Se o endereço de destino está na tabela e a interface é a mesma de onde veio, descarta
  - Se o endereço de destino não está na tabela, transmite em todas as interfaces exceto a interface de onde veio

Ponte (*bridge*)

Instituto de computação

- Possui a característica de aprendizagem automática
  - Construção automática da tabela de comutação
    - Cada quadro que passa pela ponte é examinado e são colocados na tabela o endereço fonte, a interface de onde veio o quadro e o tempo do registro na tabela
    - Registros expiram
      - Ex.: Pode-se trocar uma estação de lugar

Comutador (*switch*)

Instituto de computação

- Nível enlace
- Pontes de alto desempenho e múltiplas interfaces
- Tem um maior número de interfaces
- Atualmente a maioria é utilizada para acesso dedicado
  - **Uma única estação por domínio de colisão**
- Usam o endereço MAC de destino para encaminhar e filtrar quadros
- Cada segmento de rede é um domínio de colisão separado
- Pode conectar segmentos de rede operando em diferentes taxas
- Pode conectar segmentos de rede diferentes

Comutador (*switch*)

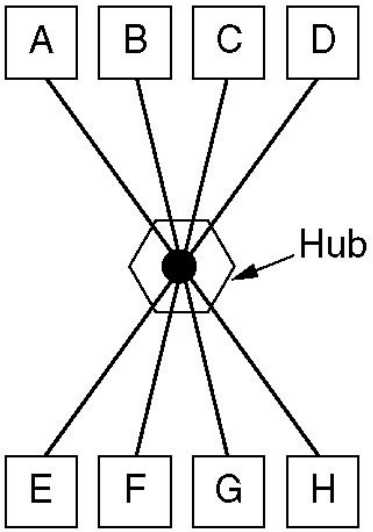
Instituto de computação

- Pode trabalhar em *full-duplex*
- Comutação de quadros
  - Quadro sempre é expedido pela mesma saída, decidida uma vez por todas quando da aceitação de trocar dados

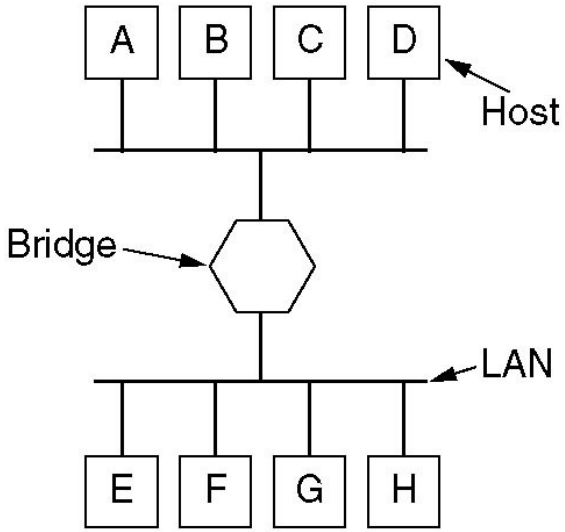
Elementos de Interconexão

Instituto de computação

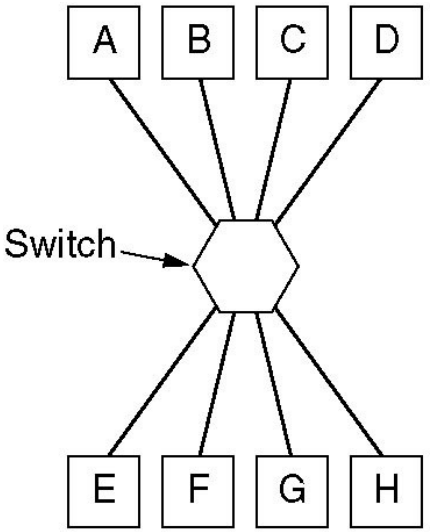
Hub, ponte e comutador (fonte: Tanenbaum)



(a)



(b)

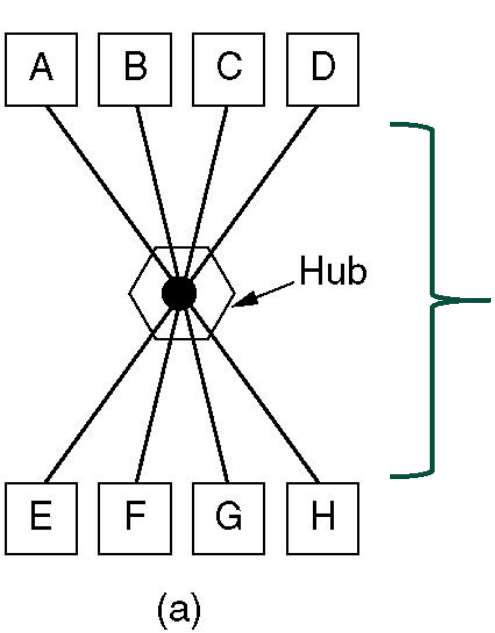


(c)

Elementos de Interconexão

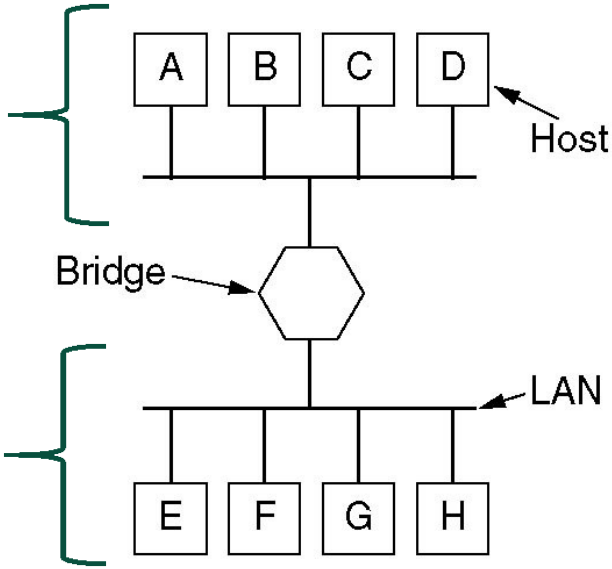
Instituto de computação

Hub, ponte e comutador (fonte: Tanenbaum)



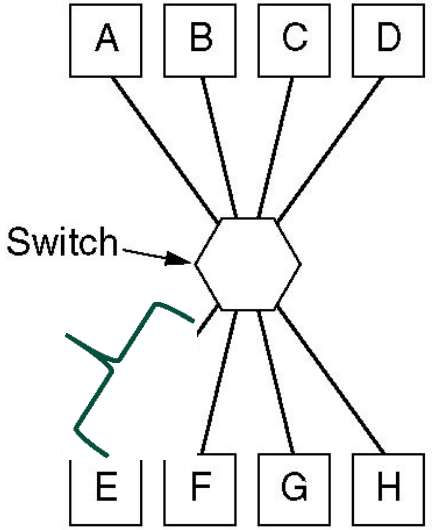
(a)

1 domínio de colisão



(b)

2 domínios de colisão

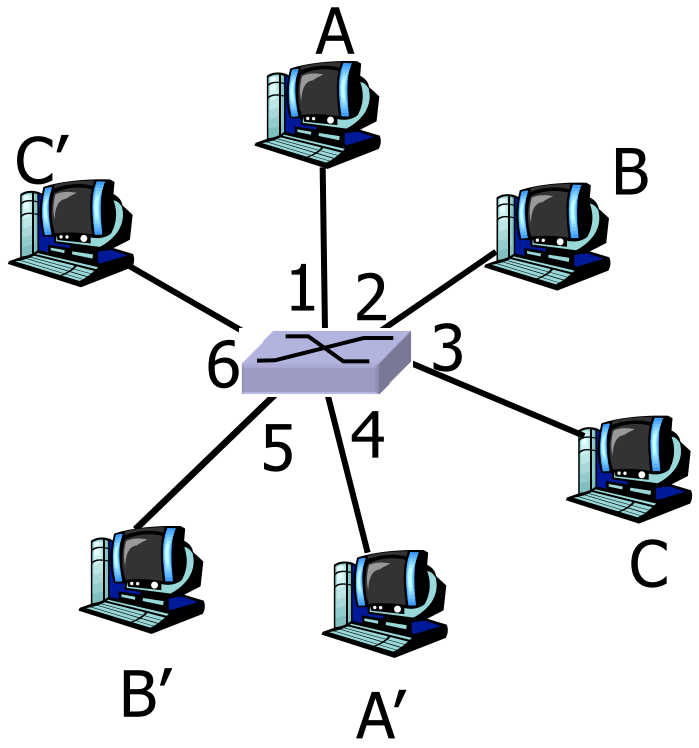


(c)

8 domínios de colisão

Comutador: Múltiplas Transmissões Simultâneas

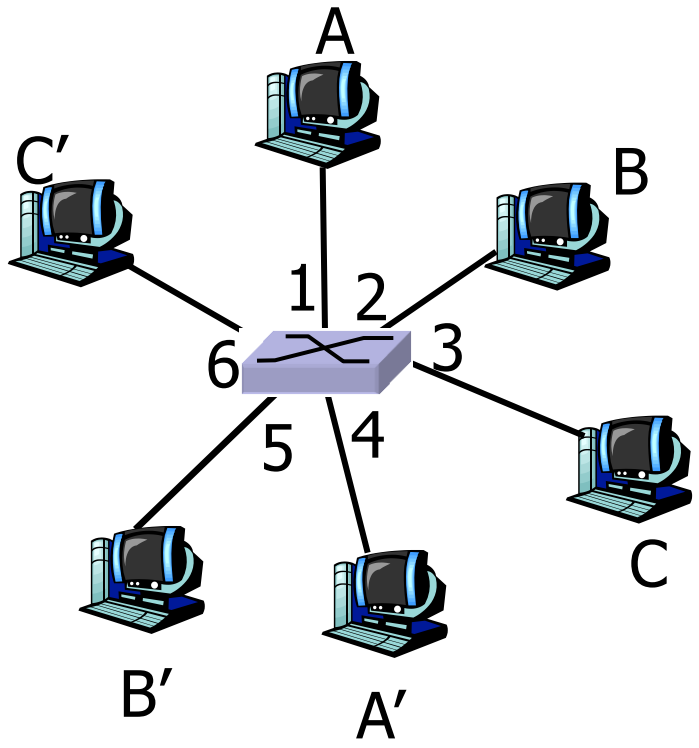
- Estações têm conexão direta e dedicada para o comutador
- Os comutadores armazenam quadros
- O protocolo Ethernet é usado em *cada* enlace de entrada, mas não há colisões; *full duplex*
  - Cada enlace é o seu próprio domínio de colisão
- Comutação: A-para-A′ e B-para-B′ simultaneamente, sem colisões
  - isto não é possível com hubs



comutador com seis interfaces (1,2,3,4,5,6)

Comutador: Tabela de Comutação

- Como é que o comutador sabe que A' é alcançável através da interface 4, e que B' é alcançável a partir da interface 5?
  - Cada comutador possui uma tabela de comutação, cada entrada contém:
    - (endereço MAC da estação, interface para alcançar a estação, carimbo de tempo)
  - Similar a uma tabela de roteamento
- Como são criadas e mantidas as entradas na tabela de comutação?
  - Há algo como um protocolo de roteamento?



comutador com seis interfaces (1,2,3,4,5,6)

Comutador: autoaprendizagem

- Comutador **aprende** quais estações podem ser alcançados através de quais interfaces
  - Quando um quadro é recebido, o comutador "aprende" a localização do transmissor: segmento LAN de entrada
  - registra o par transmissor/localização na tabela de comutação

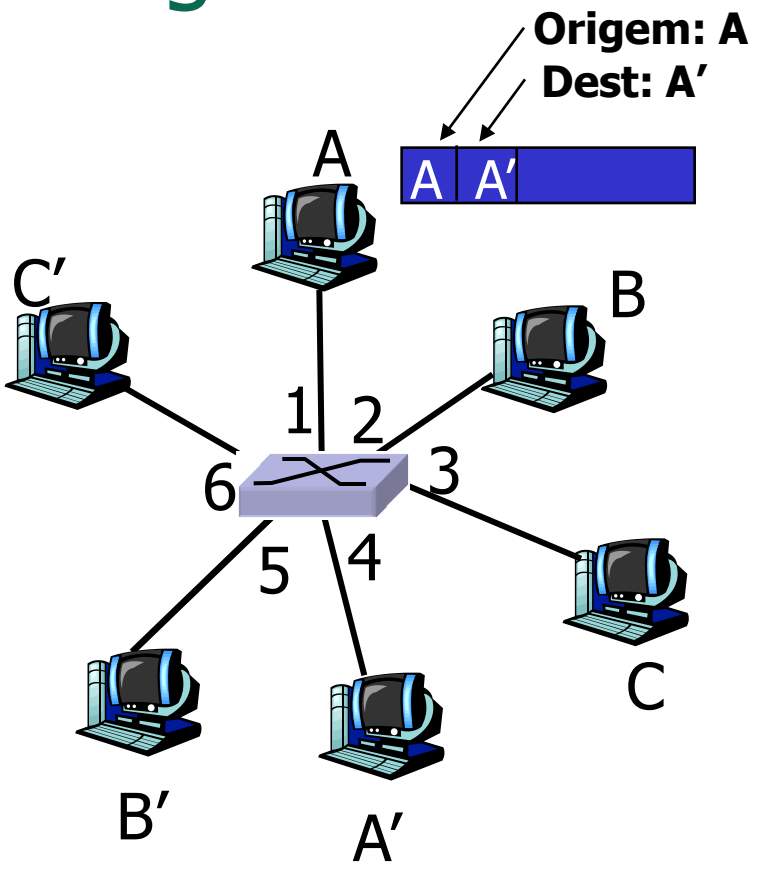


Tabela de comutação (inicialmente vazia)

end MAC	interface	TTL
A	1	60

Filtragem e Encaminhamento

Quando um comutador recebe um quadro:

registra o enlace associado com a estação transmissora

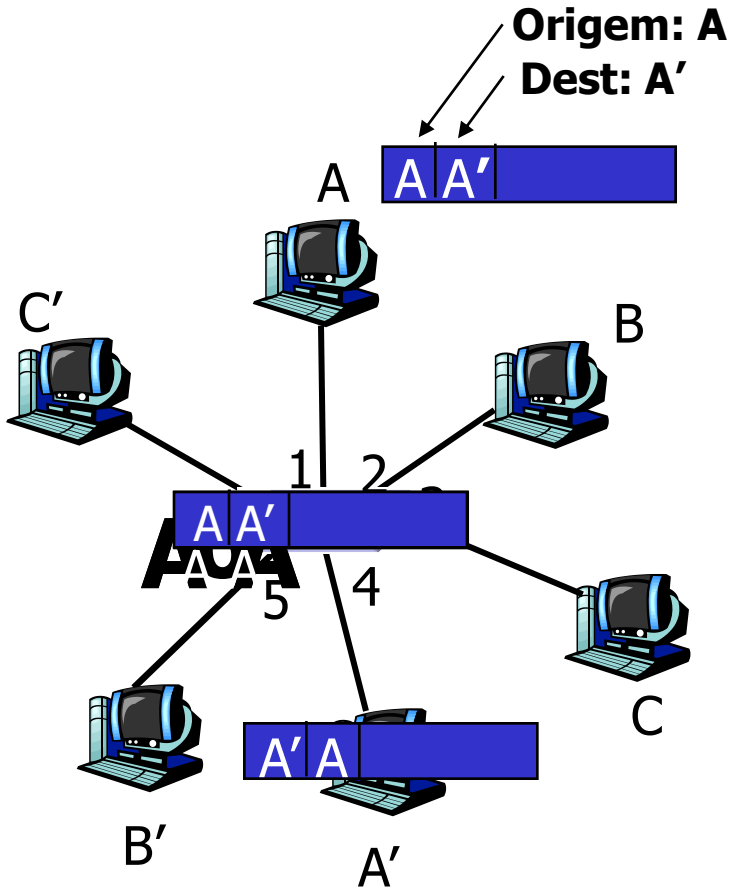
indexa a tabela de comutação usando o endereço MAC do destino

if entrada encontrada para o destino
then{
if dest estiver no segmento de onde veio o quadro
then descarta o quadro
else repassa o quadro na interface indicada
}
else usa inundação

Repassa o quadro para todas as demais interfaces exceto aquela em que o quadro foi recebido

Autoaprendizagem

- Destino do quadro desconhecido:
  inundação
- Local do destino A conhecido:
  transmissão seletiva

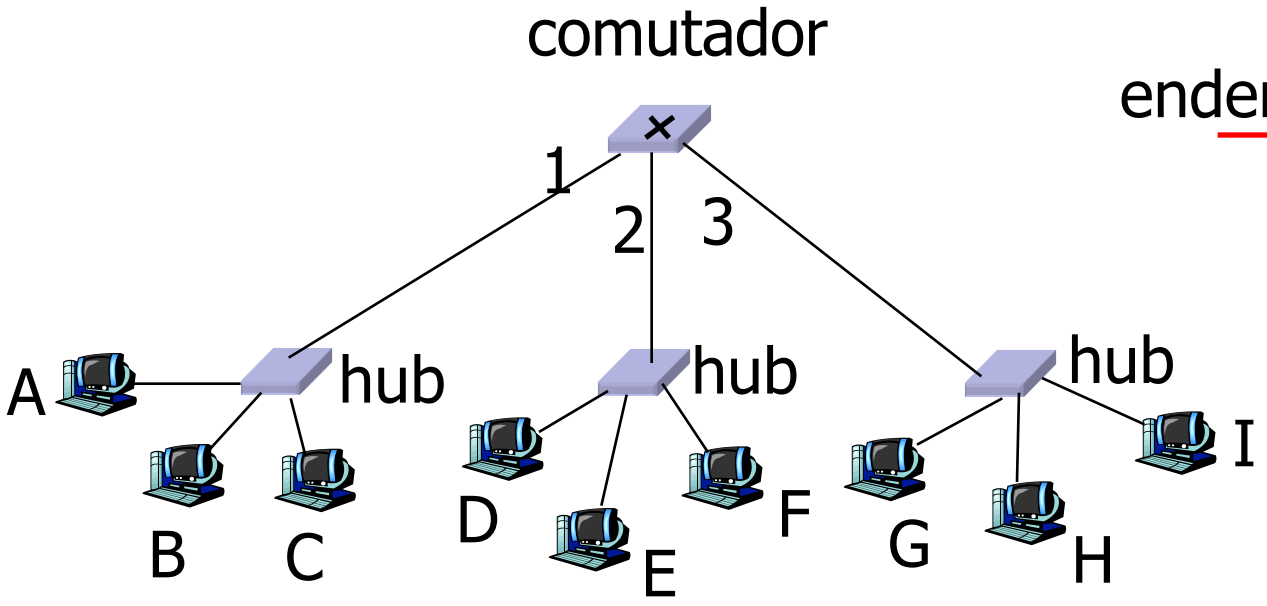


end MAC	interface	TTL
A A′	1 4	60 60

Tabela de comutação (inicialmente vazia)

Exemplo com comutador

Suponha que C envia quadro para D

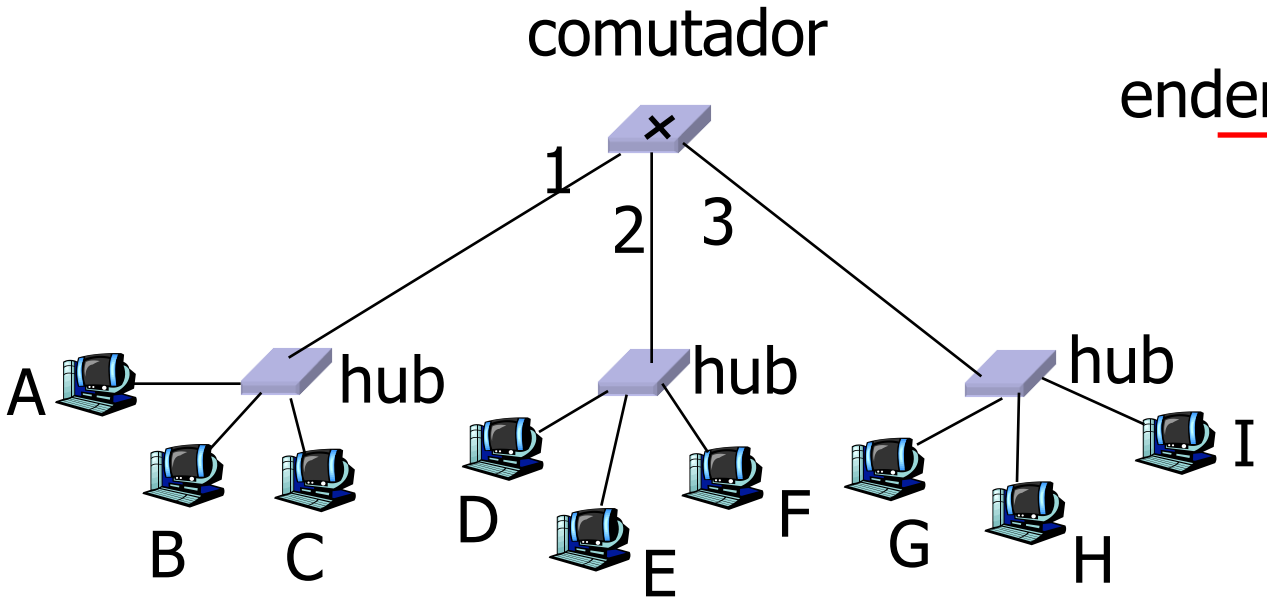


endereço	interface
A	1
B	1
E	2
G	3

- Comutador recebe o quadro vindo de C
  - Anota na tabela de comutação que C está na interface 1
  - Dado que D não se encontra na tabela, encaminha o quadro para as demais interfaces: 2 e 3
- Quadro é recebido por D

Exemplo com comutador

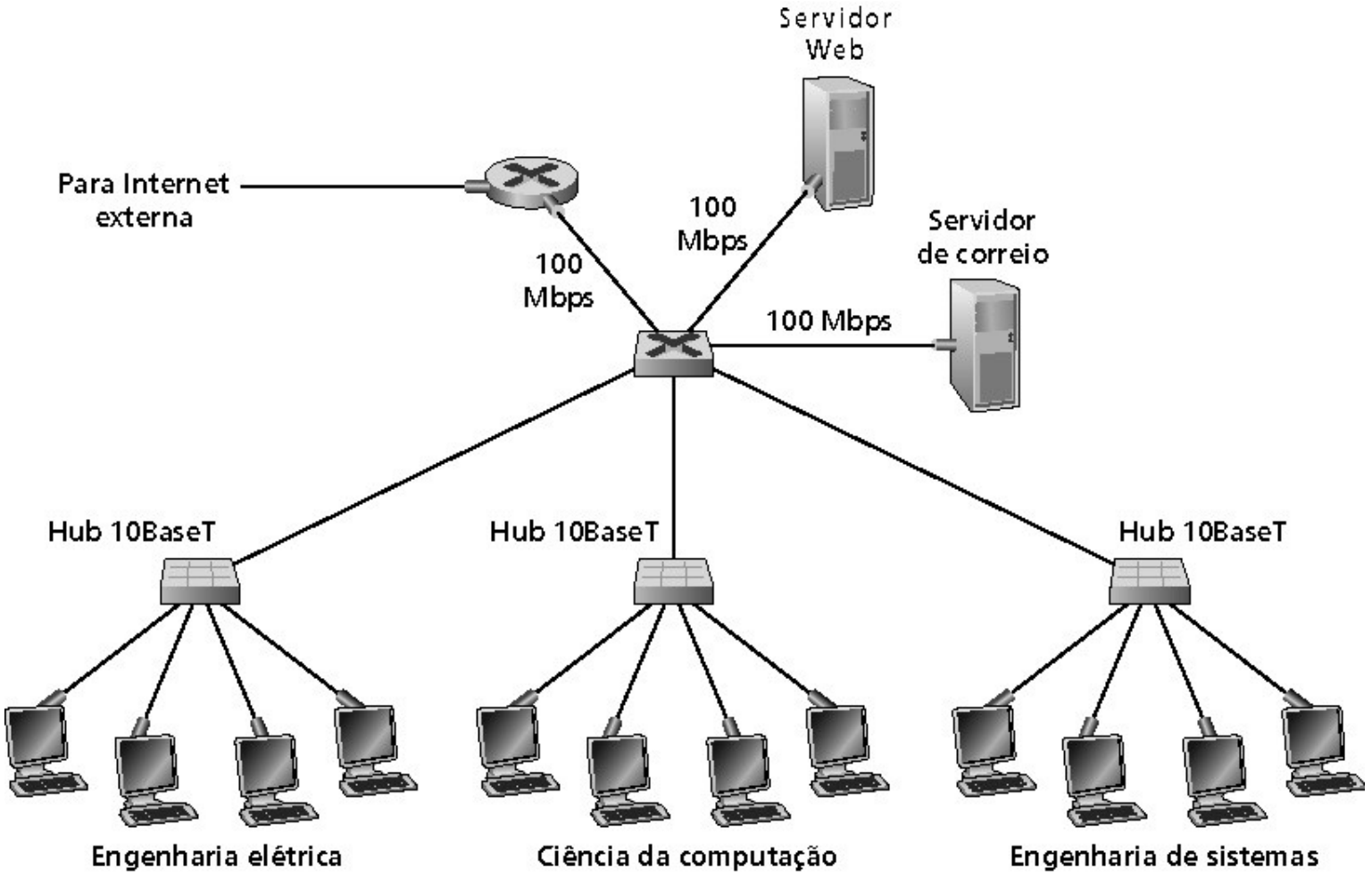
Suponha que D responda com um quadro para C



endereço	interface
A	1
B	1
E	2
G	3
C	1

- Comutador recebe o quadro vindo de D
  - Anota na tabela de comutação que D está na interface 2
- Dado que C está na tabela, encaminha o quadro apenas na interface 1
  - Quadro é recebido por C

Rede Institucional/Corporativa



Roteador

Instituto de computação

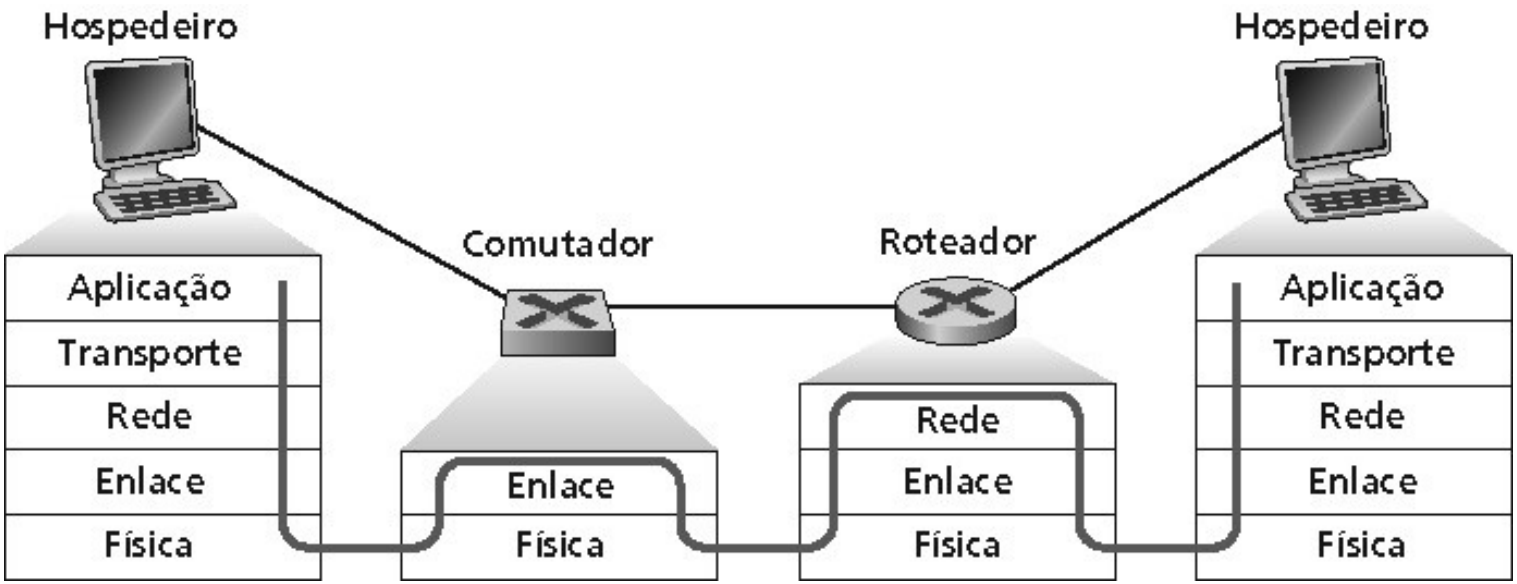
- Nível rede
- Roteamento de pacote
  - Endereço do destinatário
  - Escolha da melhor saída no momento da decisão

Comutadores x Roteadores

Instituto de computação

- Ambos são dispositivos do tipo armazena-e-repassa
  - Roteadores → dispositivos da camada de rede
    - Examinam os cabeçalhos dessa camada
  - Comutadores → dispositivos da camada de enlace
- Roteadores
  - Mantêm tabelas de roteamento, implementam algoritmos de roteamento
- Comutadores
  - Mantêm tabelas de comutação, implementam filtragem, algoritmos de aprendizagem

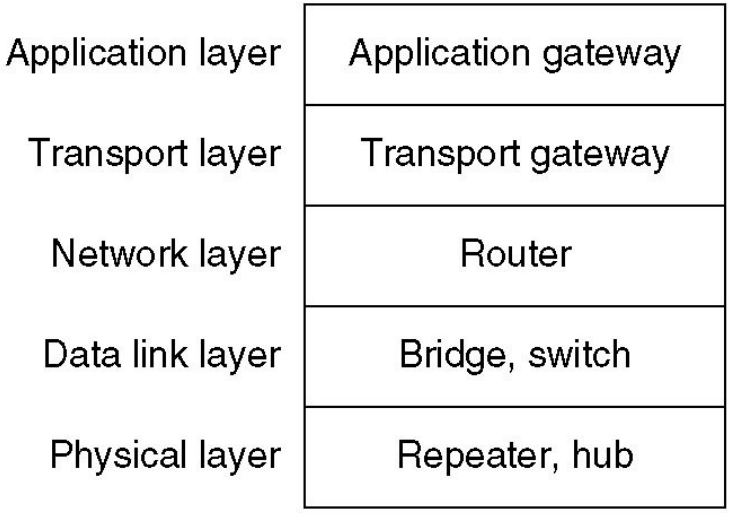
Comutadores x Roteadores



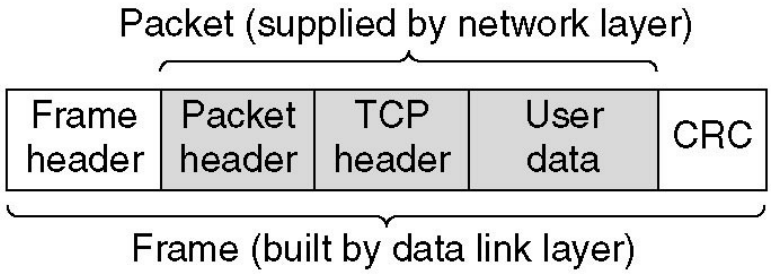
Elementos de Interconexão

Instituto de computação

Elementos (fonte: Tanenbaum)



(a)



(b)

Instituto de computação

Departamento de Ciência da Computação
Instituto de Computação
Universidade Federal Fluminense

Aulas 1 e 2

Camada de Enlace

Serviços, endereçamento e elementos de interconexão

Igor Monteiro Moraes

Redes de Computadores II