Anotações dos slides de InfraCom

‹ Módulo 5 ›

Objetivos :

- Compreender os princípios por trás dos serviços da camada enlace:
  - Detecção e correção de **erro**
  - Compartilhamento de canal broadcast : acesso múltiplo
  - Endereçamento na camada enlace
  - Transferência confiável de dados, controle de fluxo: **já estudados!**
- Instanciação e implementação de várias tecnologias da camada enlace

5.1 Introdução e Serviços

Algumas terminologias:

- Hosts e roteadores são **nós/nodos**.
- Canais de comunicação que interconectam nós adjacentes são **links ou enlaces**.
  - enlaces **cabeados**
  - enlaces **sem fio** (wireless)
  - **LANs**
- Um pacote da camada de enlace (camada 2) se chama **frame ou quadro** e encapsula um datagrama.

- A camada enlace tem como responsabilidade **transferir** um datagrama de um **nó** a outro **nó adjacente** através de um link.
- Um datagrama é transferido por diferentes protocolos da camada enlace sobre links diferentes.
  - exemplo: Ethernet no 1º link, frame relay em links intermediários, 802.11 no último link.
- Cada protocolo da camada enlace provê diferentes tipos de serviços.
  - exemplo: pode ou não prover rdt sob um link.

Analogia:

- Viagem de Recife para Lausana
  - limousine: Recife para REC
  - avião: REC para Genebra
  - trem: Genebra para Lausana
- Turista = datagrama
- Segmento de transporte = link de comunicação
- Modo de transporte = protocolo da camada enlace
- Agente de viagem = algoritmo de roteamento

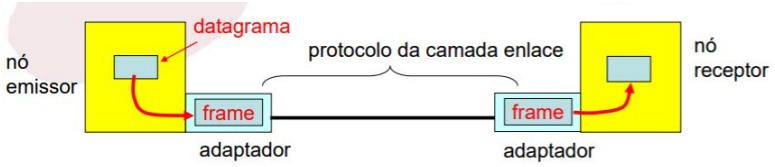
5.1.1 Serviços da Camada Enlace

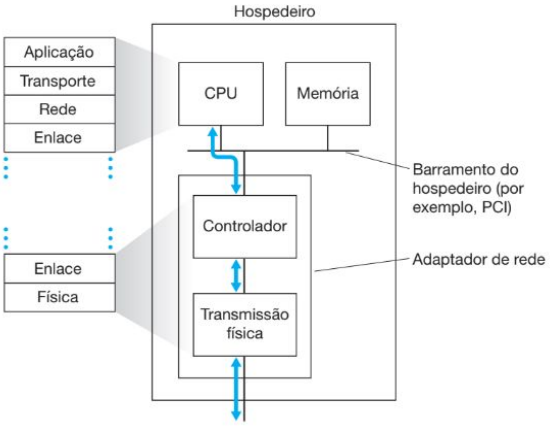
- **Framing**:
  - Encapsula datagrama em um quadro (frame), adicionando header, trailer.
- **Acesso ao link**:
  - Um protocolo de controle de acesso ao meio (medium access control — MAC) especifica as regras segundo as quais um quadro é transmitido pelo enlace.
  - Para enlaces ponto-a-ponto com um emissor e um remetente:
    - Protocolo MAC é simples ou inexistente .
    - Emissor pode enviar sempre que o enlace estiver ocioso.
  - Acesso múltiplo: vários nós compartilham um único enlace de difusão.
    - Endereços “MAC” usados no header para identificar fonte e destino.
    - Diferente de endereços IP!
- **Transferência confiável** entre nós adjacentes:
  - Já aprendemos isso (módulo 3)!
  - Raramente usada em links com baixa taxa de erro de bit (fibra, alguns pares trançados)
  - Links wireless: altas taxas de erro.
  - Mas por quê prover confiabilidade fim-a-fim e na camada enlace ao mesmo tempo? (Pois, com a confiabilidade na camada de enlace, o erro será detectado e corrigido antes do destino final)
- **Controle de Fluxo**:
  - Emissor não envia mais dados do que o receptor adjacente possa receber.
- **Detecção de Erros**:
  - Erros causados por ruído eletromagnético e atenuação de sinal.
  - Receptor detecta a presença de **erros**:
    - Sinaliza emissor para fazer retransmissão ou descarta o frame.

- **Correção de Erro**:
  - **Receptor** identifica e corrige erro(s) de bit sem ter que pedir retransmissões.
- **Half-duplex e Full-duplex**
  - Half-duplex: nós na extremidade do link podem transmitir mas não ao mesmo tempo.

5.1.2 Adaptadores

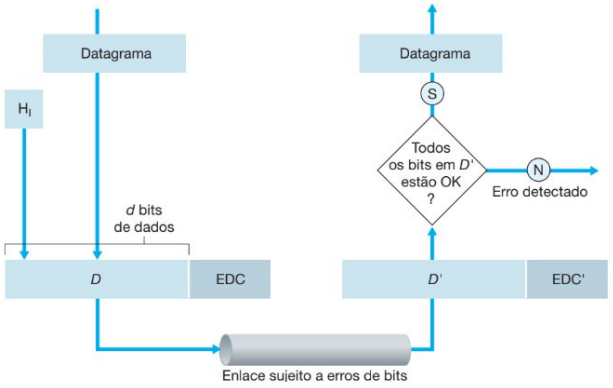
- A camada de enlace é implementada em um **adaptador de rede**, às vezes também conhecido como placa de interface de rede (NIC).
- Lado emissor:
  - Encapsula datagrama em um quadro (frame).
  - Adiciona bits para verificação de erro, rdt, controle de fluxo, etc.
- Lado receptor:
  - Procura por erros, rdt, controle de fluxo, etc.
  - Extrai datagrama, passa-o para a camada de rede do nó receptor
- Adaptador é **semi-autônomo**
- Implementa camada física e enlace.





5.2 Detecção e correção de erro

- Detecção de Erro não é 100% confiável!
  - Protocolo pode não achar alguns erros, mas é raro
  - Campo EDC grande permite melhor desempenho de detecção e correção

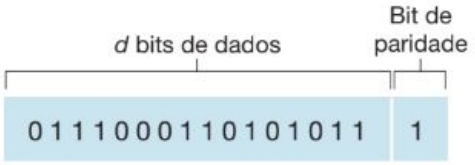


← EDC = bits de Detecção e Correção de erro (redundância)

← D = Dado protegido pela verificação de erro, pode incluir campos de cabeçalho (header)

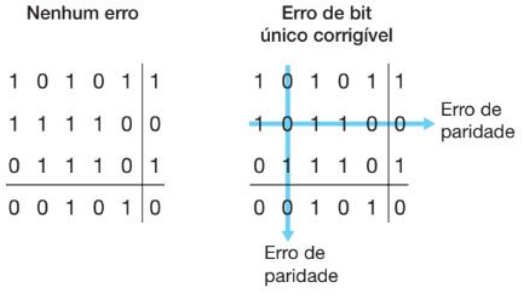
5.2.1 Verificação de Paridade

- Bit de paridade único
  - Detecta um único bit errado
  - Esquema de paridade **par**, o remetente apenas inclui um bit adicional e escolhe o valor desse bit de modo que o número total de "1" nos d + 1 bits (a informação original mais um bit de paridade) seja par.
  - Esquemas de paridade **ímpar**, o valor do bit de paridade é escolhido para que haja um número ímpar de "1".



Paridade par

- Paridade Bidimensional
  - Detecta e **corrige** erro em um único bit



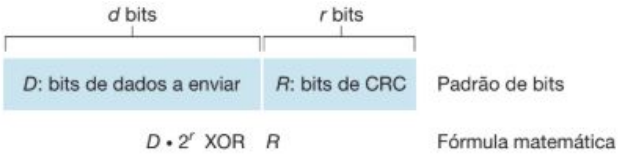
5.2.2 Internet checksum

Objetivo: detectar "erros" (exemplo: bits trocados) em um segmento transmitido (nota: usado na camada transporte somente)

- Emissor:
  - Trata conteúdo de segmentos como sequência de números inteiros de 16 bits
  - Checksum: adição (soma complemento 1) do conteúdo do segmento
  - Emissor coloca o valor do checksum no campo checksum do UDP
- Receptor:
  - Computa checksum do segmento recebido
  - Verifica se checksum computado é igual ao informado no campo checksum:
    - ✗ NÃO - erro detectado
    - ✓ SIM - nenhum erro detectado. Mas erros podem não ter sido notados ...

5.2.3 Verificação de redundância cíclica (CRC)

- Enxerga bits de dados D como um número binário



- Escolhe padrão de r+1 bits (gerador) G
- Objetivo: escolha r CRC bits R tal que
  - é exatamente divisível por G (módulo 2)
  - Receptor conhece G, divide por G.
    - Se resto **diferente** de **zero**: **erro detectado**!
  - Pode detectar qualquer rajada de erro menor que r+1 bits
- Amplamente utilizado na prática (ATM, HDLC)

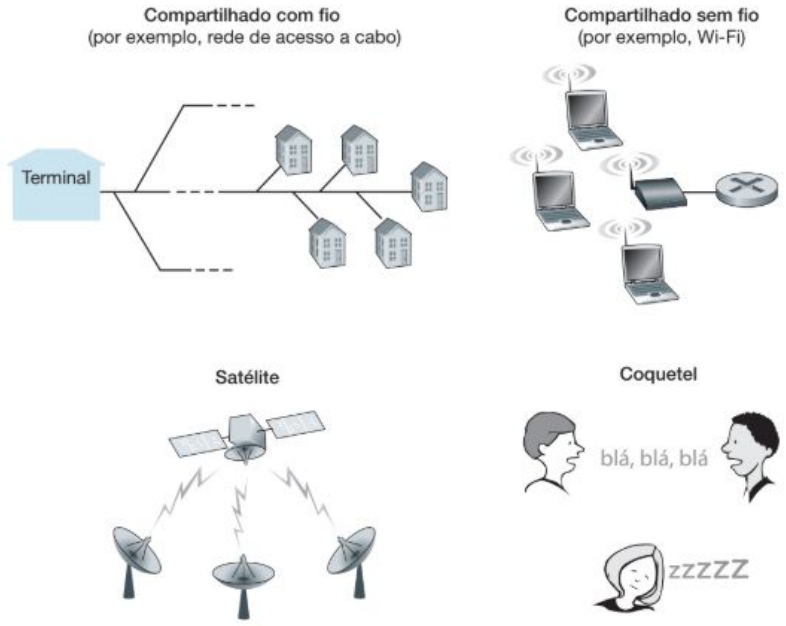
- Exemplo
  - Cálculo de R

$$R = \text{resto}\,\frac{D \cdot 2^r}{G}$$

5.3 Protocolos de Acesso Múltiplo

Há basicamente dois tipos de enlaces: os **ponto-a-ponto**, onde o enlace é exclusivo aos dois hosts que interliga, e os **broadcast**, que são compartilhados por vários usuários.

- Como **coordenar o acesso** de vários nós remetentes e receptores a um canal de difusão compartilhado?



exemplos de canais de acesso múltiplo

✘ Problema do acesso múltiplo

- 2 ou mais transmissões simultâneas: **interferência**
  - Há colisão se nó recebe 2 ou mais sinais ao mesmo tempo

5.3.1 Protocolos de Acesso Múltiplo

- Algoritmo distribuído que determina como os nós compartilham o canal, por exemplo, determina quando o nó pode transmitir
- Comunicação sobre compartilhamento de canal deve usar o próprio canal!
  - Não há canal fora de banda (out-of-band) para coordenação de transmissões.

⇨ Protocolo de acesso múltiplo **ideal**

- Simples para que sua implementação seja barata.
- Completamente **descentralizado**:
  - **Não** há nó especial para coordenar transmissões
  - **Não** há sincronização de relógios, slots
  - **Não** representa um único ponto de falha para a rede.
- Quando um só nó estiver ativo, ele deve obter toda a banda passante *R*.
- Quando *n* nós estiverem ativos, cada um pode transmitir a uma taxa média *R/n*.

Há basicamente três abordagens para se construir um protocolo de acesso múltiplo: partição de canal, acesso aleatório e taking turns.

☆ Particionamento de Canal

- Divide canal em pequenas “partes” (slots de tempo, frequência, código).
- Aloca “parte” para uso exclusivo do nó.

☆ Acesso Randômico

- Canal não é dividido e podem ocorrer colisões.
- Precisa tratar colisões .

☆ “Taking turns”

- Cada nó aguarda sua vez para transmitir. Um passa a vez para o outro.
- Nós que desejam enviar mais experimentarão maiores atrasos.

5.3.2 Particionamento de Canal

⇨ TDMA: time division multiple access

- Acesso ao canal ocorre em "rounds"
- Cada host recebe um slot de tamanho fixo em cada round
- Slots não utilizados são desperdiçados (ficam idles ou vazios)

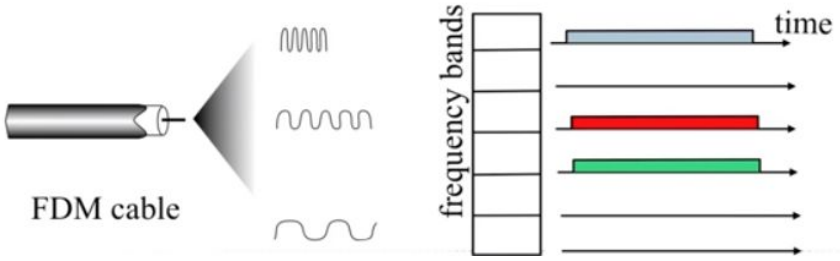
- exemplo: 6 hosts em um LAN, 1,3,4 possuem pacote, slots 2,5,6 idle (vazios)

- TDM (Time Division Multiplexing):
  - Canal dividido em N slots de tempo, 1 por usuário.
  - Ineficiente com usuários com baixo ciclo de trabalho e a altas cargas.

⇾ FDMA: frequency division multiple access

- Espectro do canal dividido em bandas de frequência.
- Cada host recebe uma banda de frequência.
- Se não há transmissões na banda, ela é desperdiçada.

- Exemplo: 6 hosts em um LAN, 1, 3, 4 possuem pacote, bandas 2, 5, 6 sem uso



- FDM (Frequency Division Multiplexing): a frequência é subdivida

5.3.3 Protocolos de Acesso Randômico

- Quando nó possui pacote para enviar
  - Transmite a **taxa máxima** R do canal.
  - Não há **coordenação prévia** entre nós
- 2 ou mais nós transmitindo ao mesmo tempo ➜ "colisão"
- Protocolos MAC de acesso randômico especificam:
  - Como **detectar** colisões
  - Como **tratar** colisões (por exemplo, atrasando retransmissões)

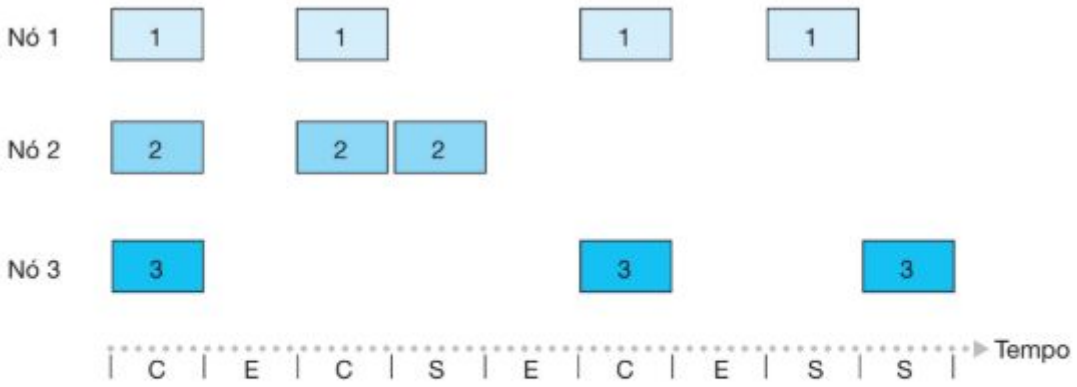
⇾ Slotted ALOHA

Hipóteses

- Frames de mesmo tamanho
- Tempo é dividido em slots de mesmo tamanho, tempo para transmitir um quadro da camada enlace
- Nós **iniciam transmissão** somente no **início** de um slot
- Nós estão sincronizados
- Se 2 ou mais nós transmitem em um mesmo slot, todos detectam a colisão.

Funcionamento

- Quando nó obtém um novo quadro da camada enlace para transmitir, ele o **transmite** no próximo slot.
- Se não há colisão, nó pode enviar novo quadro da camada enlace no slot seguinte
- Se há colisão, nó retransmite quadro da camada enlace em cada slot subsequente com **probabilidade *p*** até o sucesso ou espera (não faz nada) com **probabilidade *1-p***



Legenda:

C = Intervalo de colisão
E = Intervalo vazio
S = Intervalo bem-sucedido

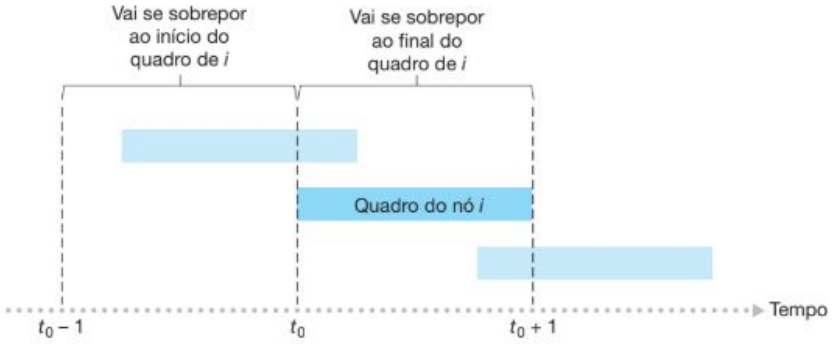
- **Prós**
  - Um único nó ativo pode transmitir continuamente na taxa máxima do canal.
  - Altamente descentralizado: mas não totalmente, somente slots precisam estar sincronizados.
  - Simples
- Contras
  - Colisões, desperdício de slots
  - Slots livres
  - Nós precisam ter a habilidade de detectar colisões em um tempo menor que o necessário para transmitir um pacote
  - Sincronização (todo nó precisa saber quando o slot inicia)

- **Eficiência** é a fração de slots utilizados com **sucesso** a longo-prazo quando há muitos nós, cada qual com muitos quadros a serem transmitidos.
  - Suponha N nós com muitos quadros a serem enviados, cada nó transmite em um slot com *probabilidade p*
  - Probabilidade de que 1 nó tenha sucesso em um slot $= p\,(1-p)^{N-1}$
  - Probabilidade de que qualquer nó tenha sucesso $= Np(1-p)^{N-1}$
  - Para obter a eficiência máxima com N nós, encontre p' que maximiza $Np(1-p)^{N-1}$
  - Quando há muitos nós, o limite de $Np'(1-p')^{N-1}$ quando N vai a infinito, indica uma **eficiência máxima de 1/e = 0,37**.

No melhor dos casos: Canal usado em transmissões *úteis* **37% do tempo**!

☞ ALOHA Puro (unslotted)

- Unslotted Aloha: mais simples, **sem sincronização**
- Quando camada enlace possui quadros para transmitir
  - **transmite-os imediatamente**
- **Probabilidade de colisão aumenta**:
  - Quadro enviado em t_0 colide com outros quadros enviados em $[t_0 -1, t_0 +1]$



- **Eficiência**
  - P(sucesso de um dado nó) = P(nó transmitir)

P(nenhum outro nó transmitir em $[t_0 -1, t_0]$) × P(nenhum outro nó transmitir em $[t_0, t_0 +1]$)

$= p(1-p)^{N-1} \times (1-p)^{N-1} = p(1-p)^{2(N-1)}$

... escolhendo *p* ótimo e fazendo N → infinito ...

= 1/(2e) = 0.18

No melhor dos casos: Canal usado em transmissões *úteis* **18% do tempo**!

Metade da eficiência do Slotted ALOHA.

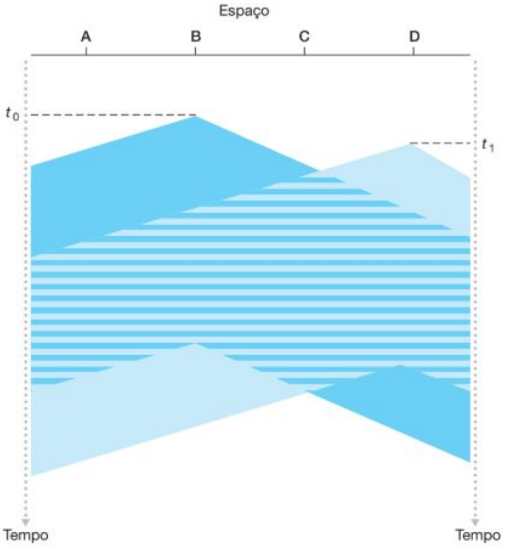
☞ CSMA (Carrier Sense Multiple Access)

CSMA: escuta antes de transmitir:

- Se canal está "idle" (vazio): transmite o quadro completo
- Se canal está ocupado, adiar transmissão

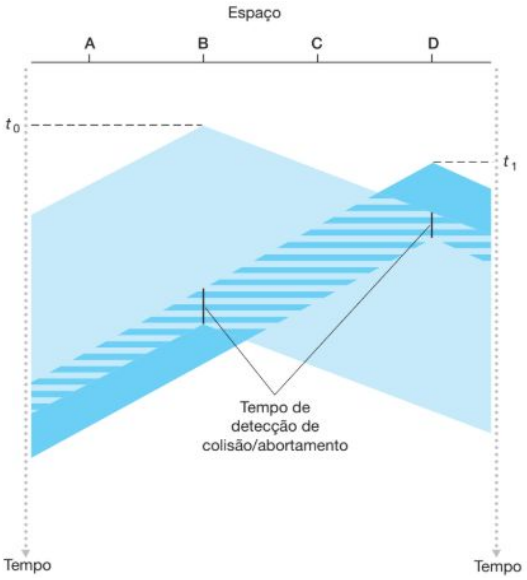
Analogia humana: *Ouça antes de falar. Se alguém começar a falar ao mesmo tempo que você, pare de falar.*

- **Colisões ainda podem ocorrer**:
  - Ter atraso de propagação significa que dois nós podem não ouvir a transmissão um do outro
- Colisão:
  - Tempo total de transmissão do pacote é desperdiçado distribuição espacial dos nós
- Nota: distância e atraso de propagação são importantes para se determinar a **probabilidade de colisão**



⇾ CSMA/CD (Collision Detection)

- CSMA/CD: escuta portadora, adiamento como no CSMA
  - Colisões **detectadas** em um intervalo curto de tempo.
  - Transmissões colidindo são abortadas, reduzindo o desperdício do canal.
- Detecção de colisão:
  - **Fácil** em LANs cabeadas: mede força de sinais, compara força do sinal transmitido com o recebido.
  - Difícil em LANs sem fio: receptor desliga enquanto se transmite. Só receptor sabe se houve colisão ou não.



Analogia humana: uma pessoa educada com grande habilidade de conversação.

5.3.4 Protocolos de revezamento

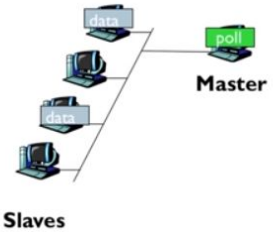
- Protocolos MAC de Particionamento de Canal
  - Compartilhamento eficiente do canal e justo com alta carga
  - Ineficiente com baixa carga: atraso de acesso ao canal, 1/N da banda alocada mesmo se somente 1 nó estiver ativo!
- Protocolos MAC de Acesso Randômico
  - Eficiente com baixa carga: um único nó pode usar toda a capacidade do canal
  - Alta carga: sobrecarga (overhead) de colisões
- Protocolos "taking turns"
  - Olham para o melhor de ambos os mundos!

⇾ Polling

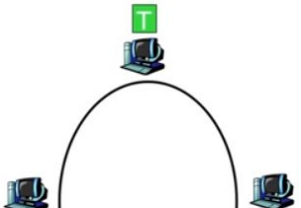
Nó mestre "convida" nós escravos a transmitirem e convite segue uma ordem ou sequência.

Fraquezas:

- Polling overhead (Gasta banda perguntando se há pacote)
- Latência (independente da quantidade de nós que realmente precisam transmitir quadros, o mestre é obrigado a perguntar a todos os nós se eles querem transmitir.)
- Ponto único de falha (mestre)



⇾ Passagem de Token



Token de controle passado de um nó a outro sequencialmente
✉ Mensagem token

Fraquezas:

- Token overhead (Gasta banda passando Token)
- Latência (Token deve passar por todos, mesmo que não queiram transmitir)
- Ponto único de falha (token)

Sem tempo pra resumo irmão

5.4 Endereçamento na camada de enlace

5.5 Ethernet

5.6 Hubs e switches