

# Capítulo 5: A camada de enlace

## Objetivos do capítulo:

- entender os princípios por trás dos serviços da camada de enlace de dados:
  - Compartilhamento de um canal de broadcast: acesso múltiplo
  - endereçamento da camada de enlace
- instanciação e implementação de várias tecnologias da camada de enlace

# Resumo

| Camada     | Problema                           | Serviço de Comunicação | Protocolos                     |
|------------|------------------------------------|------------------------|--------------------------------|
| Aplicação  |                                    |                        | HTTP, DNS, openSVN<br>DHT, SSL |
| Transporte | Confiabilidade<br>Congestionamento | Entre processos        | TPC, UDP                       |
| Rede       | Endereçamento<br>Roteamento        | Entre sistemas         | IP, ICMP, RIP, OSPF            |
| Enlace     | Codificação<br>Compartilhamento    | Entre vizinhos         | Ethernet, Wi-fi                |

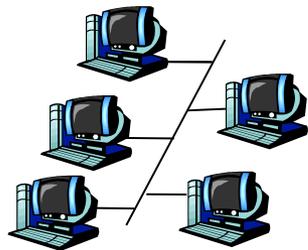
# Camada de enlace

- ❑ 5.1 Introdução e serviços
- ❑ 5.2 Detecção e correção de erros
- ❑ 5.3 Protocolos de acesso múltiplo
- ❑ 5.4 Endereçamento na camada de enlace
- ❑ 5.5 Ethernet
- ❑ 5.6 Comutadores de camada de enlace
- ❑ 5.7 PPP
- ❑ 5.8 Virtualização de enlace: MPLS
- ❑ 5.9 Um dia na vida de uma solicitação de página Web

# Enlaces e protocolos de acesso múltiplo

## Dois tipos de "enlaces":

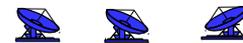
- ponto a ponto
  - PPP para acesso discado
  - enlace ponto a ponto entre computador Ethernet e hospedeiro
- **broadcast** (fio ou meio compartilhado)
  - Ethernet à moda antiga (cabo coaxial)
  - Ethernet conectado com Hub
  - LAN sem fio 802.11



fio compartilhado (p. e.,  
Ethernet cabeado)



RF compartilhada  
(p. e., WiFi 802.11)



RF compartilhada  
(satélite)

# Protocolo de acesso múltiplo ideal

## Canal de broadcast de velocidade $R$ bps

1. quando um nó quer transmitir, ele pode enviar na velocidade  $R$ .
2. quando  $M$  nós querem transmitir, cada um pode enviar na velocidade média de transmissão  $R/M$
3. totalmente descentralizado:
  - nenhum nó especial para coordenar transmissões
  - nenhuma sincronização de clocks, intervalos
4. simples

# Protocolos MAC: uma taxonomia

Três classes gerais:

## □ **Particionamento de canal**

- divide o canal em "pedaços menores" (intervalos de tempo, frequência, código)
- aloca pedaço ao nó para uso exclusivo

## □ **Acesso aleatório**

- canal não dividido, permite colisões
- "recupera" de colisões

## □ **"Revezando"**

- os nós se revezam, mas os nós com mais a enviar podem receber mais tempo

# Protocolos de acesso aleatório

- Quando o nó tem um pacote a enviar
  - transmite na velocidade de dados  $R$  total do canal.
  - sem coordenação *a priori* entre os nós
- dois ou mais nós transmitindo → "colisão",
- **protocolo MAC de acesso aleatório** especifica:
  - como detectar colisões
  - como recuperar-se de colisões (p. e., via retransmissões adiadas)
- Exemplos de protocolos MAC de acesso aleatório:
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

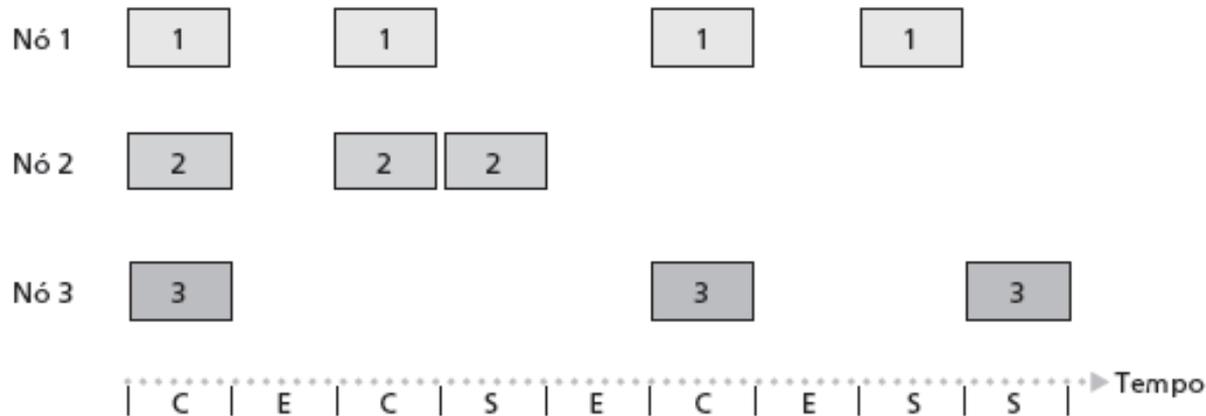
# Slotted ALOHA

## Suposições:

- ❑ todos os quadros do mesmo tamanho
- ❑ tempo dividido em intervalos de mesmo tamanho (tempo para transmitir 1 quadro)
- ❑ nós começam a transmitir somente no início dos intervalos
- ❑ nós são sincronizados
- ❑ se 2 ou mais nós transmitem no intervalo, todos os nós detectam colisão

## Operação:

- ❑ quando nó obtém quadro novo, transmite no próximo intervalo
  - *se não há colisão*: nó pode enviar novo quadro no próximo intervalo
  - *se há colisão*: nó retransmite quadro em cada intervalo subsequente com probabilidade  $p$  até que haja sucesso (backoff aleatório)



## Prós

- ❑ único nó ativo pode transmitir continuamente na velocidade plena do canal
- ❑ altamente descentralizado: somente intervalos nos nós precisam estar em sincronismo
- ❑ simples

## Contras

- ❑ colisões, intervalos desperdiçados
- ❑ intervalos ociosos
- ❑ nós podem ser capazes de detectar colisão em menos tempo do que para transmitir pacote
- ❑ sincronismo de clock

# Eficiência do Slotted Aloha

**Eficiência:** fração de intervalos bem sucedidos durante longo tempo (muitos nós, todos com muitos quadros para enviar)

- *suponha:* N nós com muitos quadros a enviar, cada um transmitindo no intervalo com probabilidade  $p$
- prob de um nó ter sucesso em um intervalo =  $p(1-p)^{N-1}$
- prob de *qualquer* nó ter sucesso =  $Np(1-p)^{N-1}$

- eficiência máxima: ache  $p^*$  que maximiza  $Np(1-p)^{N-1}$
- para muitos nós, com limite de  $Np^*(1-p^*)^{N-1}$  enquanto N tende a infinito, temos:

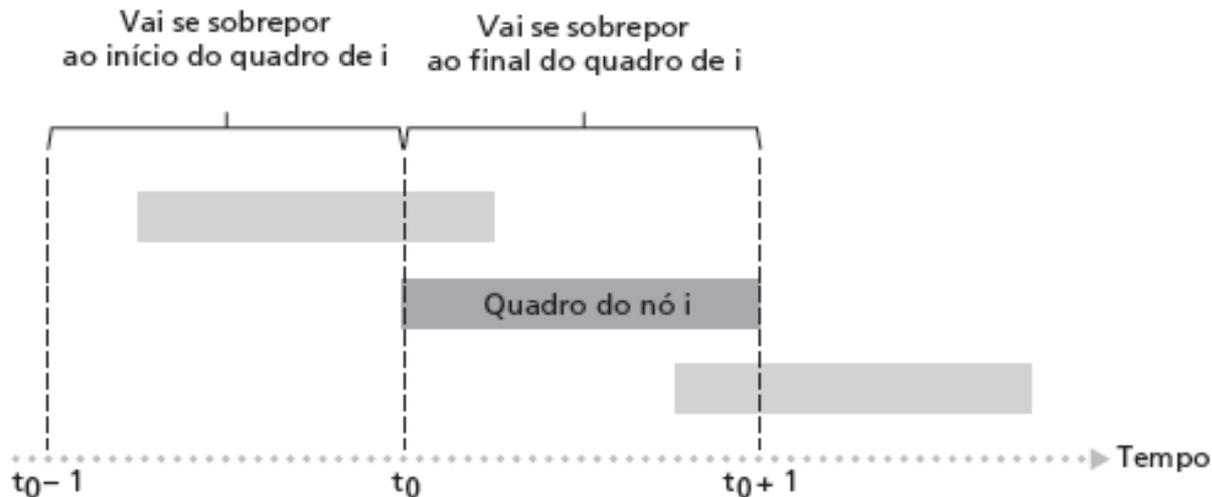
Eficiência máxima =  $1/e = 0,37$

**Na melhor das hipóteses:** canal usado para transmissões úteis 37% do tempo!



# ALOHA puro (não slotted)

- ❑ Aloha não slotted: mais simples, sem sincronismo
- ❑ quando quadro chega primeiro
  - transmite imediatamente
- ❑ probabilidade de colisão aumenta:
  - quadro enviado em  $t_0$  colide com outros quadros enviados em  $[t_0-1, t_0+1]$



# Eficiência do Aloha puro

$P(\text{sucesso por determinado nó}) = P(\text{nó transmite})$

$P(\text{nenhum outro nó transmite em } [p_0-1, p_0])$

$P(\text{nenhum outro nó transmite em } [p_0-1, p_0])$

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

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

... escolhendo  $p$  ideal e depois considerando  $n \rightarrow$  infinito ...

$$= 1/(2e) = 0,18$$

***ainda pior que slotted Aloha!***

# CSMA (Carrier Sense Multiple Access)

**CSMA**: ouça antes de falar:

se perceber canal ocioso: transmite quadro inteiro

- se perceber canal ocupado, adia transmissão
- analogia humana: não interrompa os outros!

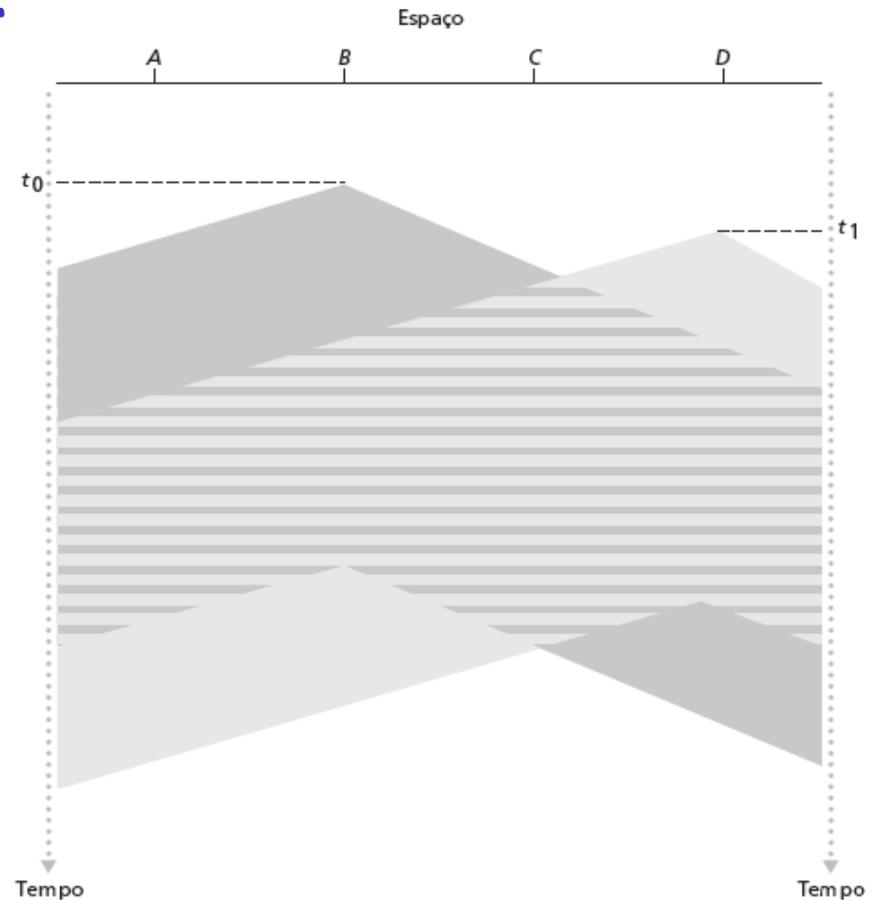
# Colisões CSMA

**colisões ainda podem ocorrer:**  
atraso de propagação significa que dois nós podem não ouvir a transmissão um do outro

**colisão:**  
tempo de transmissão de pacote inteiro desperdiçado

**nota:**  
papel da distância & atraso de propagação determinando probabilidade de colisão

layout espacial dos nós

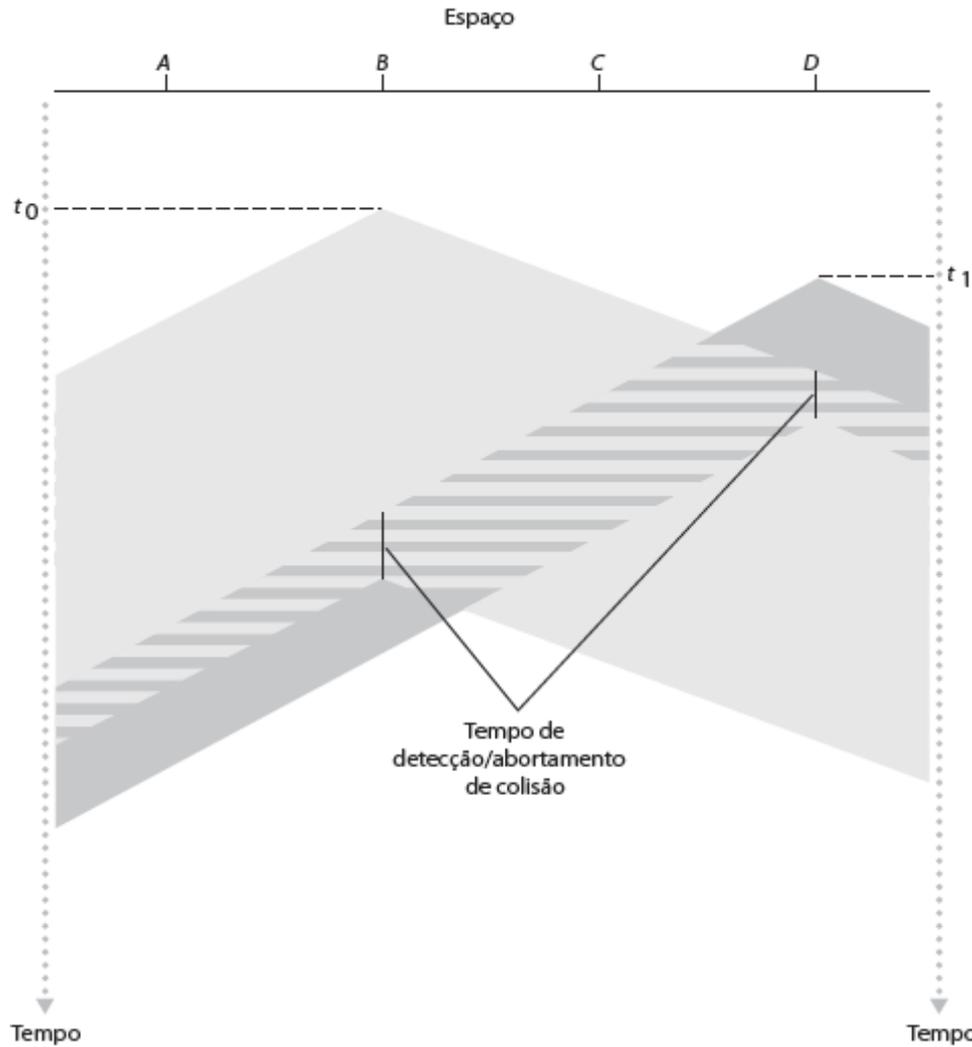


# CSMA/CD (Collision Detection)

**CSMA/CD:** detecção de portadora, adiada como no CSMA

- colisões *detectadas* dentro de pouco tempo
- transmissões colidindo abortadas, reduzindo desperdício do canal
- retransmissão com Back Off aleatório
- detecção de colisão:
  - fácil em LANs com fio: mede intensidades de sinal, compara sinais transmitidos, recebidos
  - difícil nas LANs sem fio: intensidade do sinal recebido abafada pela intensidade da transmissão local
- analogia humana: o interlocutor educado

# Detecção de colisão CSMA/CD



# Algoritmo CSMA/CD da Ethernet

1. NIC recebe datagrama da camada de rede e cria quadro
2. Se NIC sentir canal ocioso (96 tempos de bits), inicia transmissão do quadro; canal ocupado, espera até estar ocioso, depois transmite
3. Se NIC transmitir quadro inteiro sem detectar outra transmissão, NIC terminou com o quadro!
4. Se NIC detectar outra transmissão enquanto transmite, aborta e envia sinal de congestionamento
5. Depois de abortar, NIC entra em **backoff exponencial**: após  $m$  colisões, NIC escolhe  $K$  aleatoriamente dentre  $\{0,1,2,\dots,2^m-1\}$ . NIC espera  $K \cdot 512$  tempos de bit, retorna à Etapa 2

# CSMA/CD da Ethernet (mais)

## Sinal de congestionamento:

cuide para que todos os outros transmissores saibam da colisão; 48 bits

**Tempo de bit:**  $0,1 \mu\text{s}$  para Ethernet de 10 Mbps; para  $K = 1023$ , tempo de espera cerca de 50 ms

## Backoff exponencial:

- *Objetivo:* adaptar tentativas de retransmissão à carga estimada
  - carga pesada: espera aleatória será maior
- primeira colisão: escolha  $K$  a partir de  $\{0,1\}$ ; atraso é  $K \cdot 512$  tempos de transmissão de bit
- após segunda colisão: escolha  $K$  dentre  $\{0,1,2,3\}$ ...
- após dez colisões, escolha  $K$  dentre  $\{0,1,2,3,4,\dots,1023\}$

# Eficiência do CSMA/CD

- $t_{prop}$  = atraso máx. propag. entre 2 nós na LAN
- $t_{trans}$  = tempo para transmitir quadro de tamanho máximo

$$eficiência = \frac{1}{1 + 5t_{prop}/t_{trans}}$$

- eficiência vai para 1
  - quando  $t_{prop}$  vai para 0
  - quando  $t_{trans}$  vai para infinito
- melhor desempenho que ALOHA: é simples, barato, descentralizado!

# CSMA/CA (Collision Avoidance)

**CSMA/CA:** detecção de portadora, adiada como no CSMA

## □ Utilizado em Rede Wireless

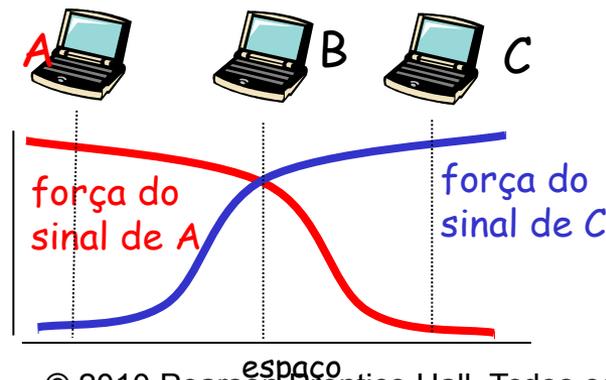
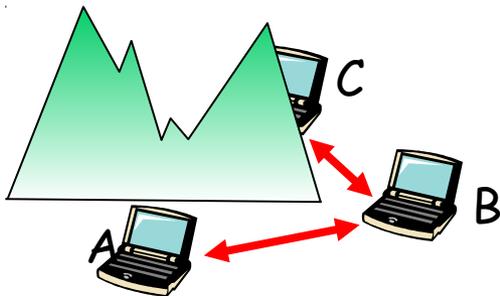
- Dificuldade em detectar colisões (sinal enviado muito mais forte)
- Nó pode não receber por estar em zona de sombra
- Necessidade de ACK (acknowledgement)

## □ Funcionamento

- Escuta o canal por um tempo pré-determinado
- Se o canal está vazio, envia um pacote RTS (Request to Send) com tamanho do pacote a ser enviado
- Se recebe um CTS (Clear to Send), então envia o pacote de dados e outros hosts esperam o tempo especificado em CTS
- Se não recebe CTS, espera por Back Off aleatório

# IEEE 802.11: acesso múltiplo

- ❑ evita colisões: 2 ou + nós transmitindo ao mesmo tempo
- ❑ CSMA/CD - detecta antes de transmitir
  - não colide com transmissão contínua de outro nó
- ❑ 802.11: *sem* detecção de colisão!
  - difícil de receber (sentir colisões) na transmissão devido a sinais recebidos fracos (desvanecimento)
  - não pode sentir todas as colisões em qualquer caso: terminal oculto, desvanecimento
  - objetivo: *evitar colisões*: CSMA/C(ollision)A(voidance)



# Protocolo MAC IEEE 802.11: CSMA/CA

## Distribution Coordination Function remetente

1 se sentir canal ocioso por **DIFS** então transmite quadro inteiro (sem CD)

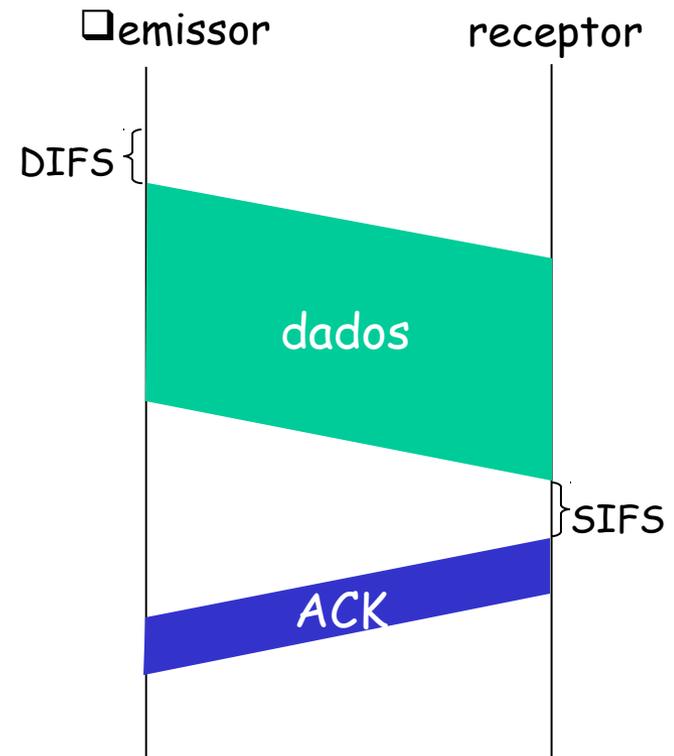
2 se sentir canal ocupado então

- inicia tempo aleatório de backoff
- temporizador conta regressivamente enquanto canal está ocioso
- transmite quando temporizador expira
- se não há ACK, aumenta intervalo de backoff aleatório, repete 2

## receptor

- se quadro recebido OK

retorna ACK após **SIFS** (ACK necessário devido ao problema de terminal oculto)



# Evitando colisões (mais)

## *Point Coordination Function*

*ideia:* permite que remetente "reserve" canal em vez de acesso aleatório aos quadros de dados: evitar colisões de quadros de dados longos

- ❑ remetente primeiro transmite *pequenos* pacotes request-to-send (RTS) ao AP (access point) usando CSMA
  - RTSs ainda podem colidir uns com os outros (mas são curtos)
- ❑ AP envia por broadcast clear-to-send (CTS) em resposta a RTS
- ❑ CTS escutado por todos os nós
  - remetente transmite quadro de dados
  - outras estações adiam transmissões

Evite colisões de quadro de dados completamente usando pequenos pacotes de reserva!

# Prevenção de colisão: troca RTS-CTS

