

# Capítulo 5: A camada de enlace

## Objetivos do capítulo:

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

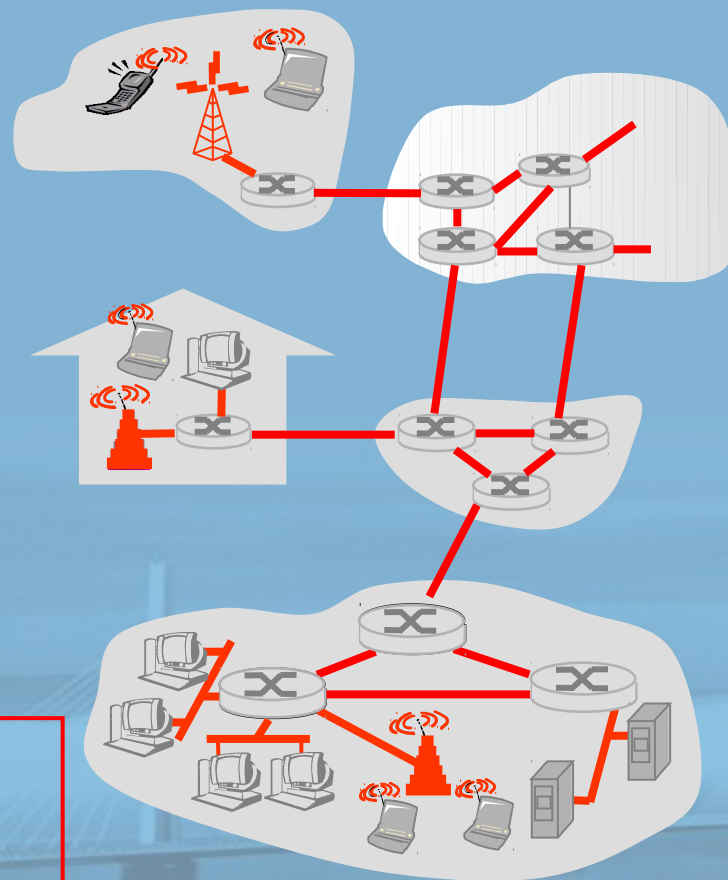
# Camada de enlace

- ❑ 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

# Camada de enlace: introdução

## Alguma terminologia:

- ❑ hospedeiros e roteadores são **nós**
- ❑ canais de comunicação que se conectam a nós adjacentes pelo caminho de comunicação são **enlaces**
  - enlaces com fio
  - enlaces sem fio
- ❑ pacote na camada-2 é um **quadro**, encapsula datagrama



**Camada de enlace de dados** tem a responsabilidade de transferir um datagrama de um nó ao nó adjacente por um enlace.

# Serviços da camada de enlace

- ❑ *enquadramento, acesso ao enlace:*
  - encapsula datagrama no quadro, incluindo cabeçalho, trailer
  - acesso ao canal de meio compartilhado
  - endereços “MAC” usados nos cabeçalhos de quadro para identificar origem, destino
    - diferente do endereço IP!
- ❑ *entrega confiável entre nós adjacentes*
  - raramente usado em enlace com pouco erro de bit (fibra, alguns pares trançados)
  - enlaces sem fio: altas taxas de erro
    - P: Por que confiabilidade em nível de enlace e fim a fim?

## ❑ *controle de fluxo:*

- controle entre nós de emissão e recepção adjacentes

## ❑ *detecção de erro:*

- erros causados por atenuação de sinal, ruído.
- receptor detecta presença de erros:
  - pede ao remetente para retransmitir ou descarta quadro

## ❑ *correção de erro:*

- receptor *identifica e corrige* erro(s) de bit sem lançar mão da retransmissão

## ❑ *half-duplex e full-duplex*

- com *half-duplex*, os nós nas duas extremidades do enlace podem transmitir, mas não ao mesmo tempo

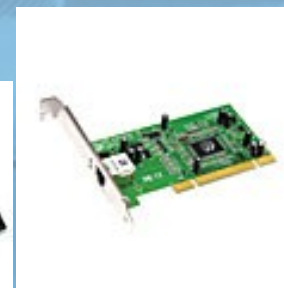
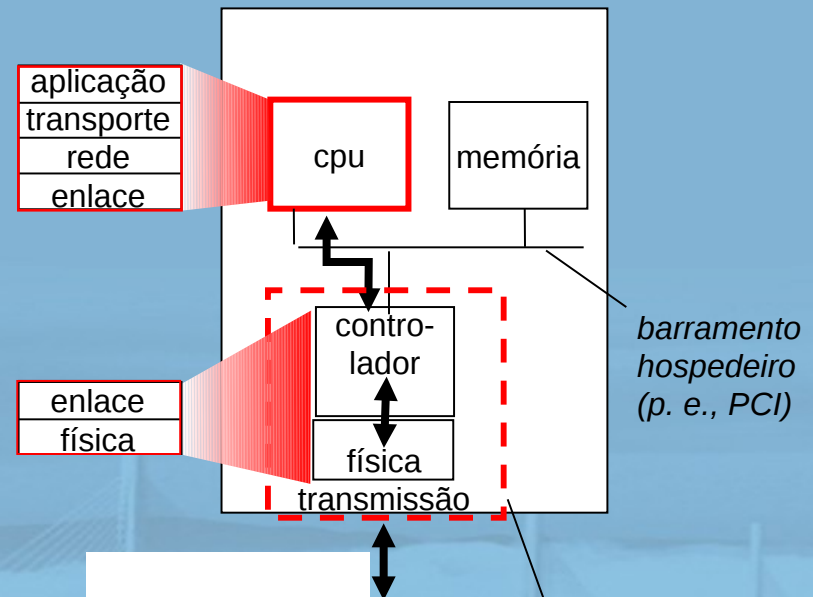


# Onde é implementada a camada de enlace?

- ❑ em todo e qualquer hosp.
- ❑ camada de enlace implementada no “adaptador” (ou *placa de interface de rede*, NIC)
  - placa Ethernet, placa PCMCIA, placa 802.11
  - implementa camada de enlace, física
- ❑ conecta aos barramentos de sistema do hospedeiro
- ❑ combinação de hardware e software

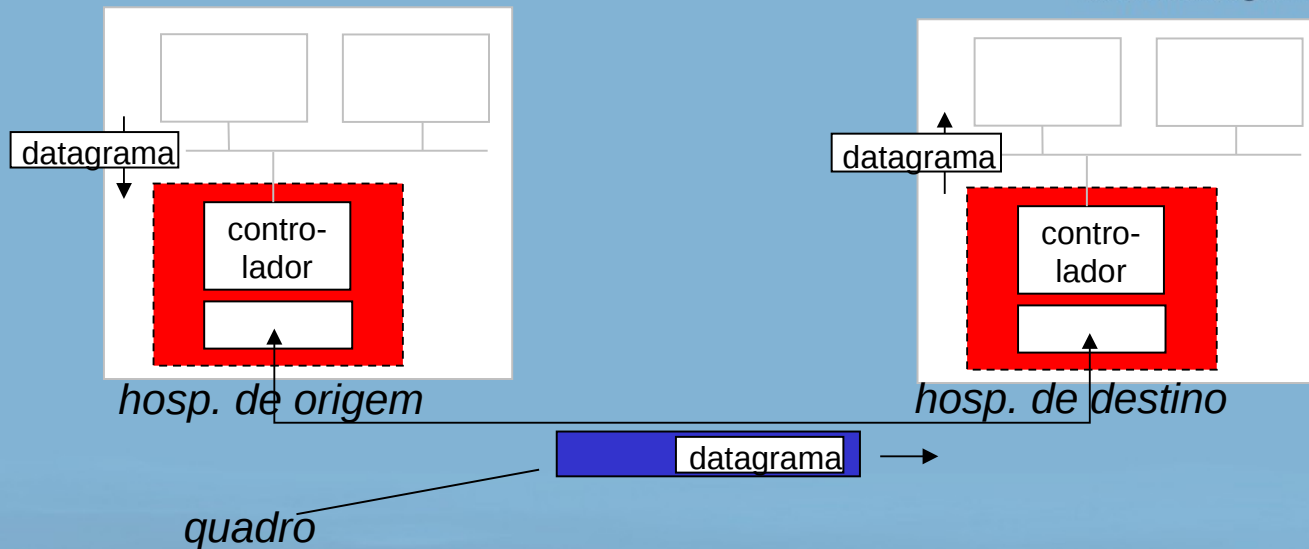


esquema do hospedeiro



placa adaptadora de rede

# Comunicação entre adaptadores



## ❑ lado emissor:

- encapsula datagrama no quadro
- inclui bits de verificação de erro, controle de fluxo, endereços, etc.

## ❑ lado receptor

- procura erros, controle de fluxo, verifica endereço, etc.
- extrai datagrama, passa para camada superior no lado receptor

## 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



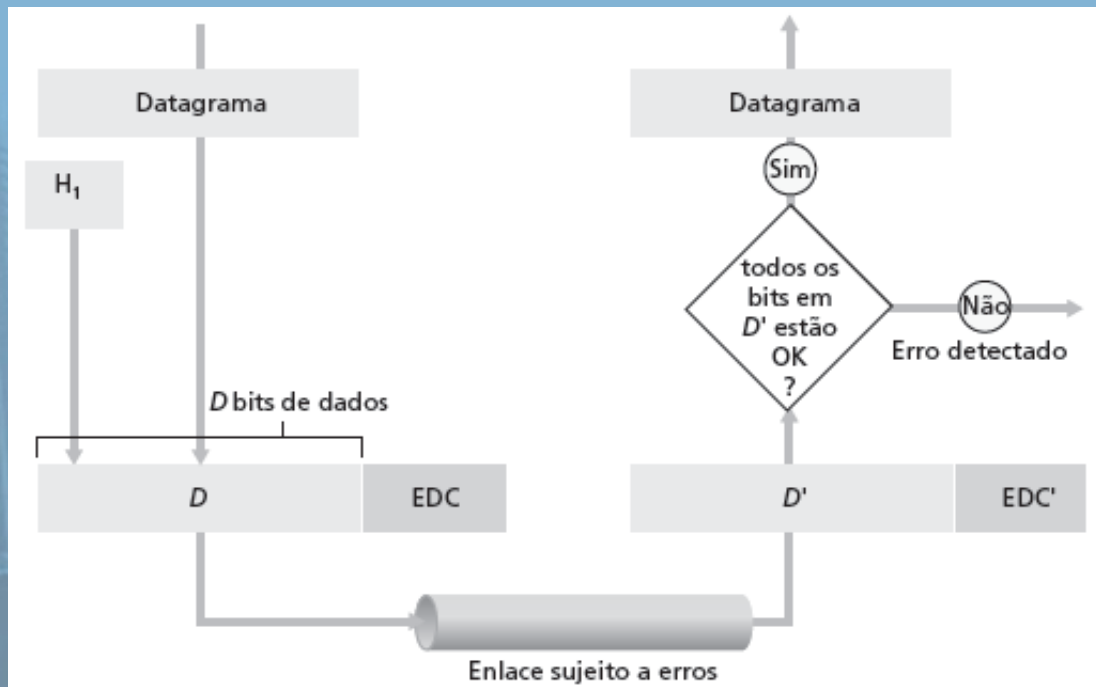
# Detecção de erros

EDC = Bits de detecção e correção de erros (redundância)

D = Dados protegidos por verificação de erro, podem incluir campos de cabeçalho

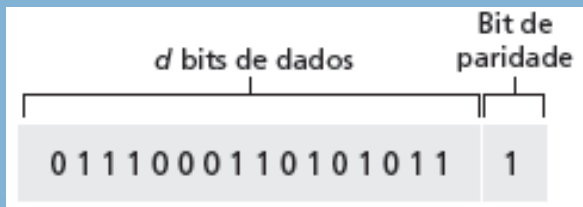
Detecção de erro não 100% confiável!

- protocolo pode perder alguns erros, mas raramente
- maior campo EDC gera melhor detecção e correção

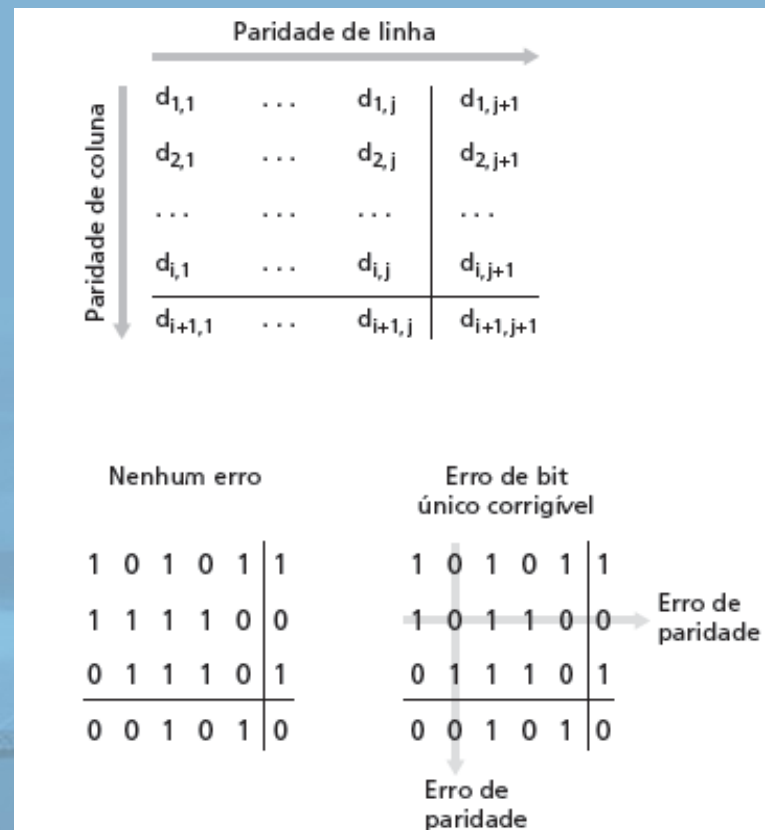


# Verificação de paridade

## Paridade de único bit: Detecta erros de único bit



## Paridade bidimensional: Detecta e corrige erros de único bit



# Soma de verificação da Internet (análise)

**Objetivo:** detectar “erros” (p. e., bits invertidos) no pacote transmitido

## **Emissor:**

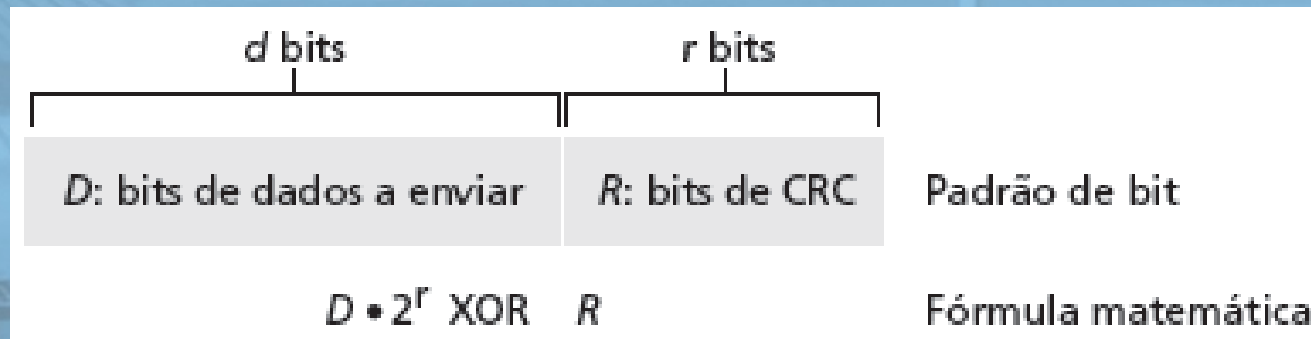
- ❑ trata conteúdo do segmento como sequência de inteiros de 16 bits
- ❑ soma de verificação: adição (soma no complemento de 1) do conteúdo do segmento
- ❑ emissor colocar valor da soma de verificação no campo de soma de verificação UDP

## **Receptor:**

- ❑ calcula soma de verificação do segmento recebido
- ❑ verifica se soma de verificação calculada é igual ao valor do campo de soma de verificação:
  - NÃO – erro detectado
  - SIM – nenhum erro detectado. *Mas pode haver erros, apesar disso?*

# Soma de verificação: verificação de redundância cíclica

- veja bits de dados, **D**, como um número binário
- escolha padrão de bits  $r + 1$  (gerador), **G**
- objetivo: escolher  $r$  bits de CRC, **R**, tal que
  - $\langle D, R \rangle$  exatamente divisível por  $G$  (módulo 2)
  - receptor sabe  $G$ , divide  $\langle D, R \rangle$  por  $G$ . Se resto diferente de zero: erro detectado!
  - pode detectar todos os erros em rajada menores que  $r + 1$  bits
- muito usada na prática (Ethernet, 802.11 WiFi, ATM), p.e.,  $G$  com 33 bits



## Exemplo de CRC

Queremos:

$$D \cdot 2^r \text{ XOR } R = nG$$

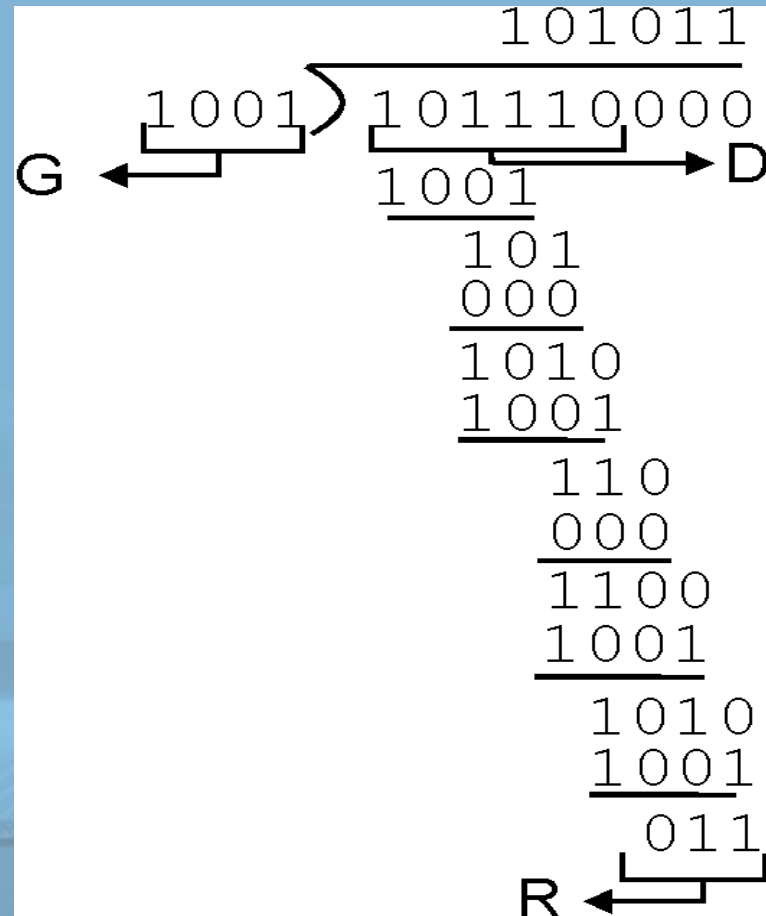
*de modo equivalente:*

$$D \cdot 2^r = nG \text{ XOR } R$$

*de modo equivalente:*

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

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





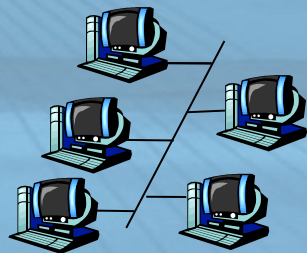
## Camada de enlace

- ❑ 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 comutador Ethernet e hospedeiro
- ❑ **broadcast** (fio ou meio compartilhado)
  - Ethernet à moda antiga
  - HFC anterior
  - LAN sem fio 802.11



fio compartilhado (p. e.,  
Ethernet cabeado)



RF compartilhada  
(p. e., WiFi 802.11)



RF compartilhada  
(satélite)

# Protocolos de acesso múltiplo

- ❑ único canal de broadcast compartilhado
- ❑ duas ou mais transmissões simultâneas por nós:  
interferência
  - **colisão** se o nó recebe dois ou mais sinais ao mesmo tempo

## protocolo de acesso múltiplo

- ❑ algoritmo distribuído que determina como os nós compartilham canal, ou seja, determinam quando o nó pode transmitir
- ❑ comunicação sobre compartilhamento de canal deve usar o próprio canal!
  - nenhum canal fora-de-banda para coordenação

# 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”**

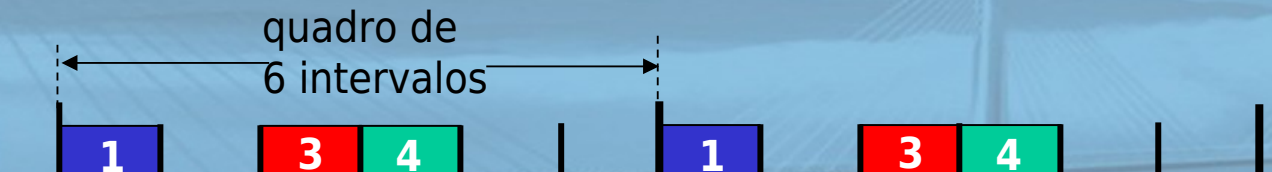
- os nós se revezam, mas os nós com mais a enviar podem receber mais tempo



# Protocolos MAC de particionamento de canal: TDMA

## TDMA: Time Division Multiple Access

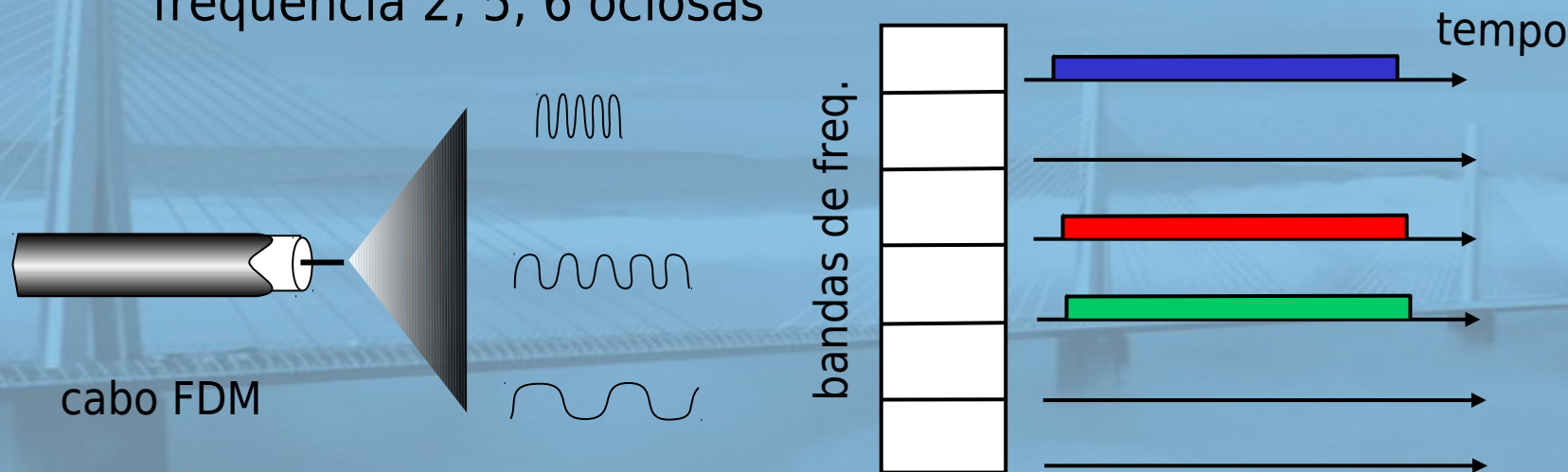
- ❑ acesso ao canal em “rodadas”
- ❑ cada estação recebe intervalo de tamanho fixo (tamanho = tempo transm. pacote) a cada rodada
- ❑ intervalos não usados ficam ociosos
- ❑ exemplo: LAN de 6 estações, 1, 3, 4 têm pacote, intervalos 2, 5, 6 ociosos



# Protocolos MAC de particionamento de canal: FDMA

## FDMA: Frequency Division Multiple Access

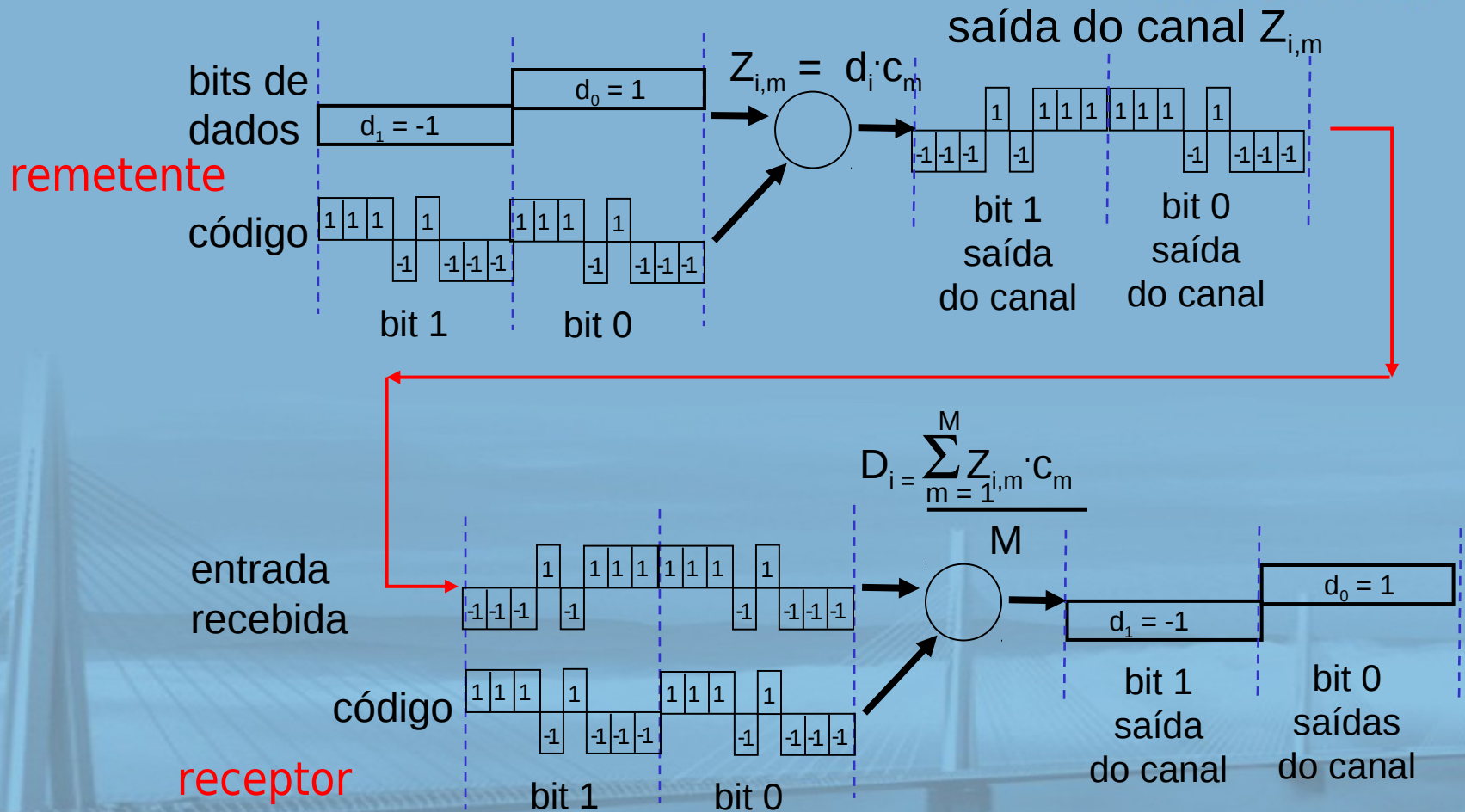
- espectro do canal dividido em bandas de frequência
- cada estação recebe banda de frequência fixa
- tempo de transmissão não usado nas bandas de frequência fica ocioso
- exemplo: LAN de 6 estações, 1, 3, 4 têm pacote, bandas de frequência 2, 5, 6 ociosas



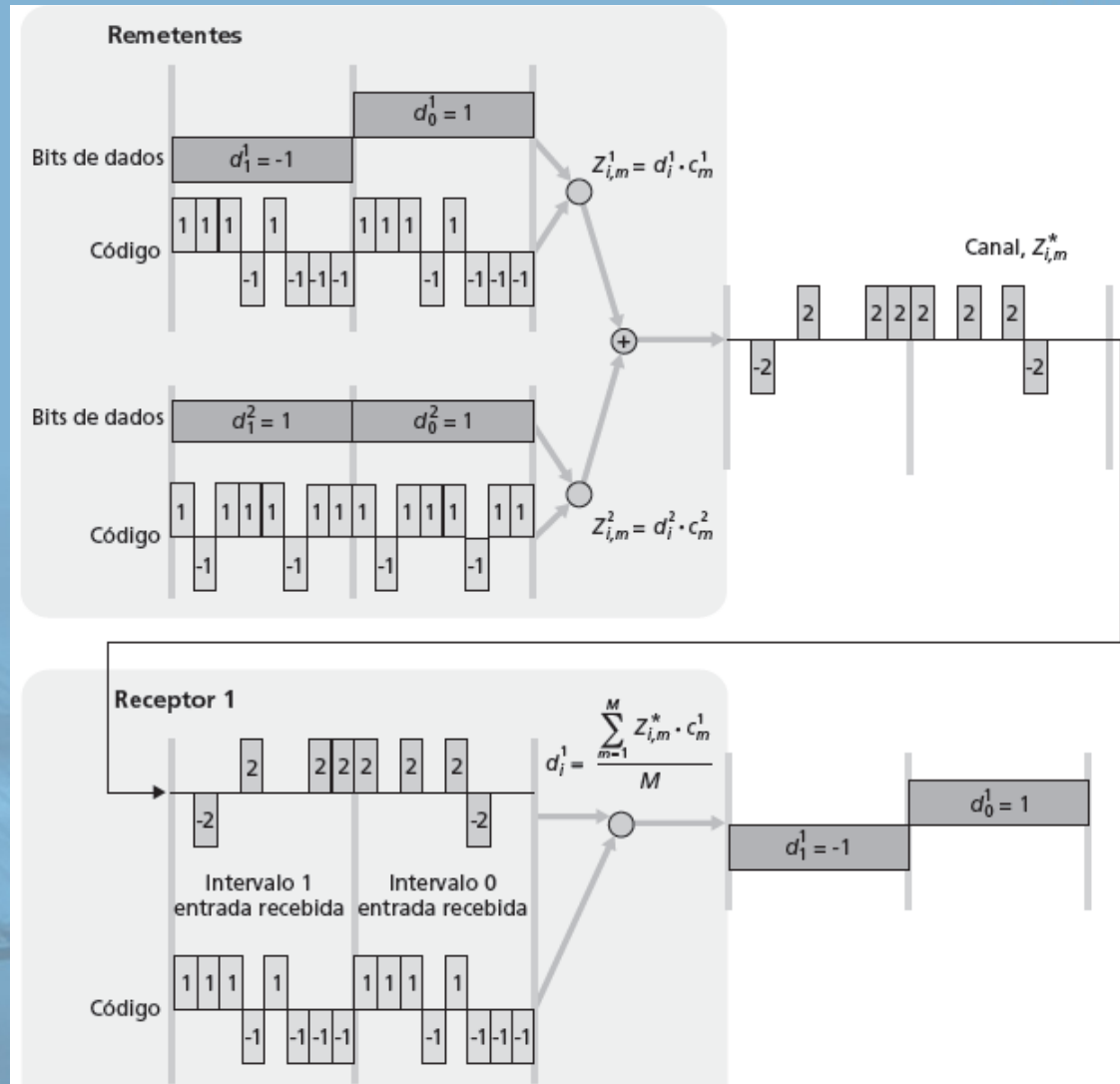
# Protocolos MAC de particio- namento de canal: Code Division Multiple Access (CDMA)

- ❑ usado em vários padrões de canais de broadcast sem fio (celular, satélite etc.)
- ❑ “código” exclusivo atribuído a cada usuário; ou seja, particionamento de conjunto de código
- ❑ todos usuários compartilham mesma frequência, mas cada usuário tem a própria sequência de “chipping” (ou seja, código) para codificar dados
- ❑ **senal codificado** = (dados originais) X (sequência de chipping)
- ❑ **decodificação**: produto interno entre sinal codificado e sequência de chipping
- ❑ permite que múltiplos usuários “coexistam” e transmitam simultaneamente com o mínimo de interferência (se os códigos forem “ortogonais”)

# Codificação/decodificação CDMA



# CDMA: interferência de dois remetentes





# 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

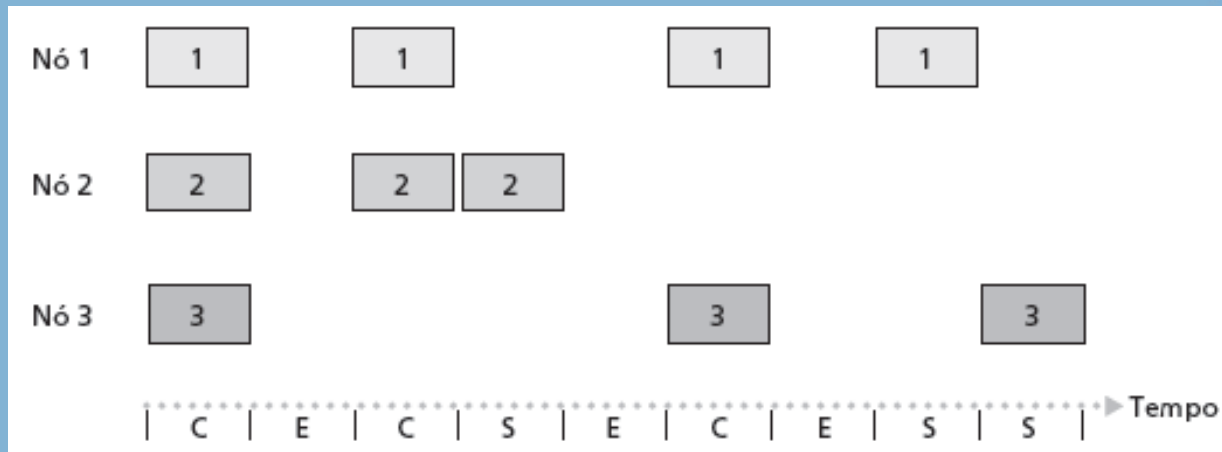
# Slotted ALOHA

## 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



## 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 durante longo tempo de intervalos bem sucedidos (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:

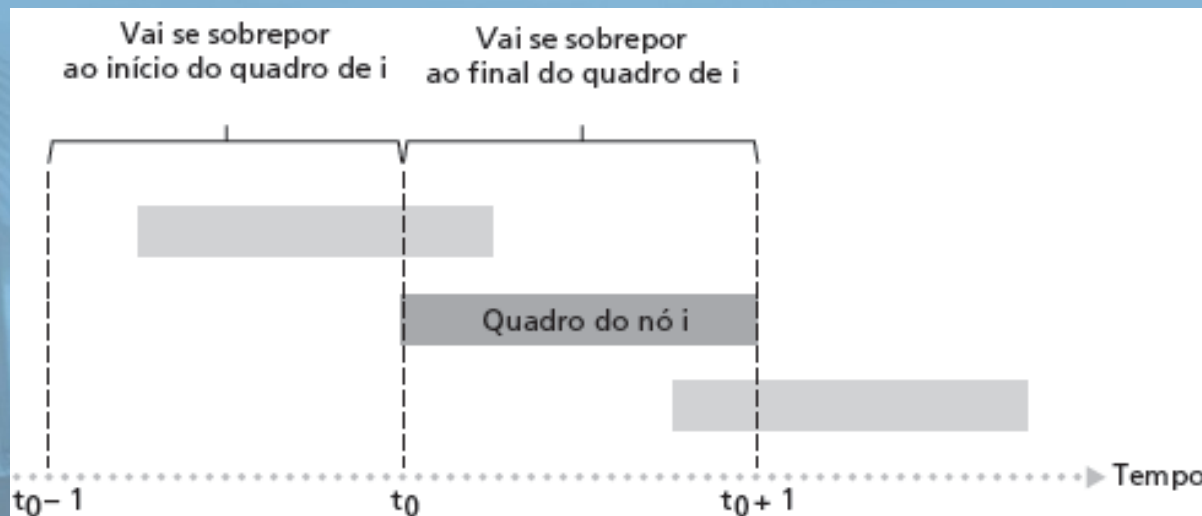
Eficiência máxima =  $1/e = 0,37$

***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  $n \rightarrow \infty$  ...

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

**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

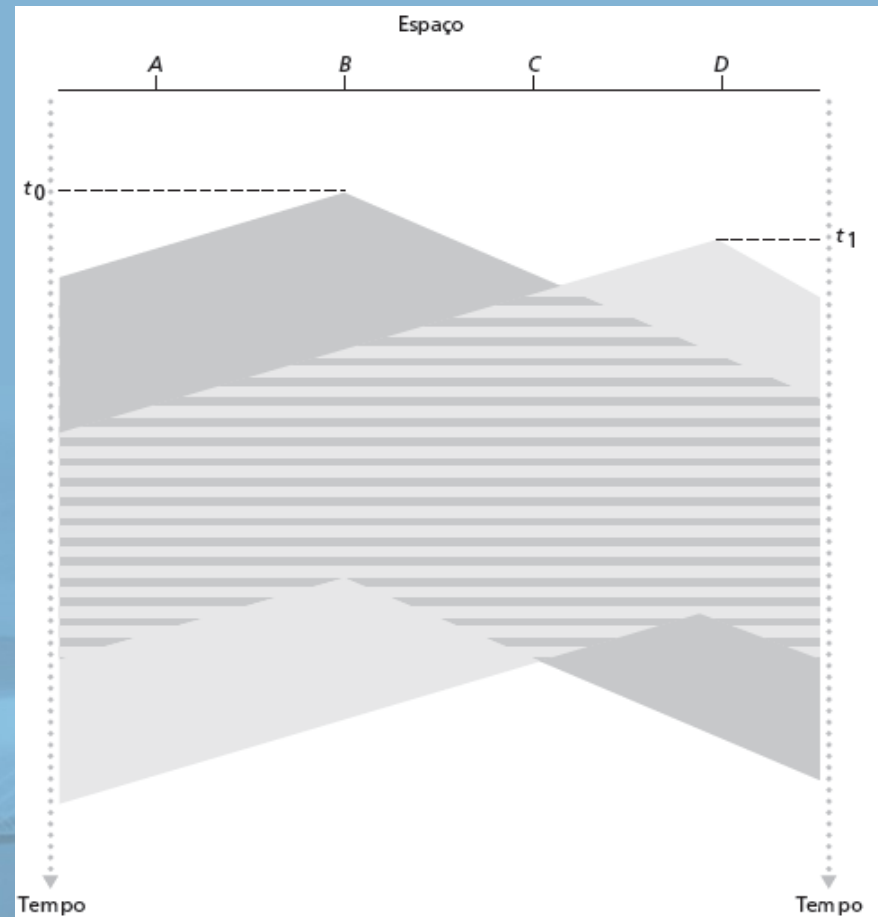
colisão:

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

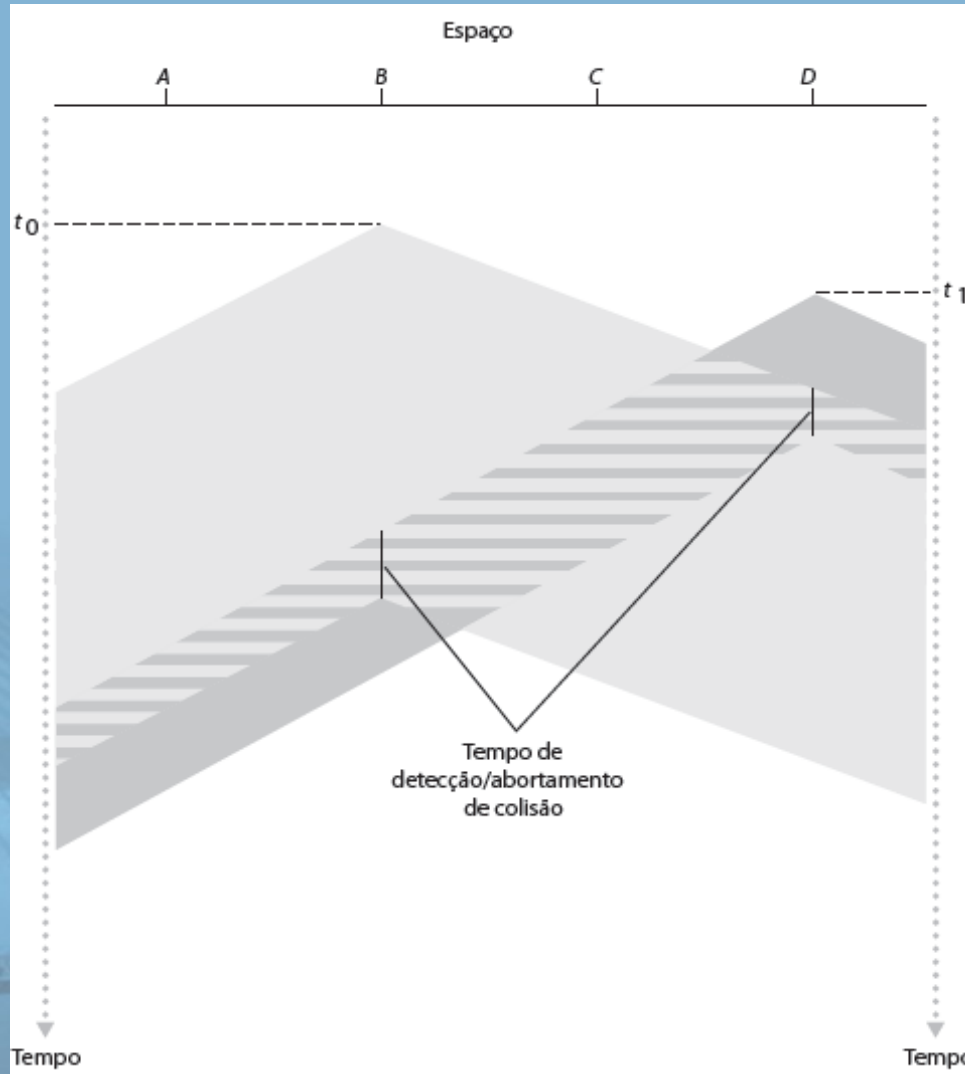
- colisões *detectadas* dentro de pouco tempo
- transmissões colidindo abortadas, reduzindo desperdício do canal

□ 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





# 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

## ❑ Funcionamento

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

# “Revezando” protocolos MAC

## protocolos MAC de particionamento de canal:

- compartilham canal de modo *eficaz* e *justo* com alta carga
- ineficaz com baixa carga: atraso no acesso ao canal,  $1/N$  largura de banda alocada mesmo que apenas 1 nó ativo!

## Protocolos MAC de acesso aleatório

- eficaz com baixa carga: único nó pode utilizar o canal totalmente
- alta carga: sobrecarga de colisão

## “revezando” protocolos

procure o melhor dos dois mundos!

## Polling (seleção):

- ❑ nó mestre “convida” nós escravos a alterarem a transmissão
- ❑ preocupações:
  - sobrecarga da seleção
  - latência
  - único ponto de falha (mestre)



## Passagem de permissão (token):

- ❑ **permissão** de controle passada de um nó para o próximo sequencialmente.
- ❑ mensagem de permissão
- ❑ preocupações:
  - sobrecarga da permissão
  - latência
  - único ponto de falha (permissão)



## Resumo de protocolos MAC

- ❑ *particionamento de canal*, por tempo, frequência ou código
  - Time Division, Frequency Division
- ❑ *acesso aleatório* (dinâmico),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - percepção de portadora: fácil em algumas tecnologias (com fio), difícil em outras (sem fio)
  - CSMA/CD usado na Ethernet
  - CSMA/CA usado na 802.11
- ❑ *revezamento*
  - polling do servidor central, passagem de permissão
  - Bluetooth, FDDI, Token Ring



## Camada de enlace

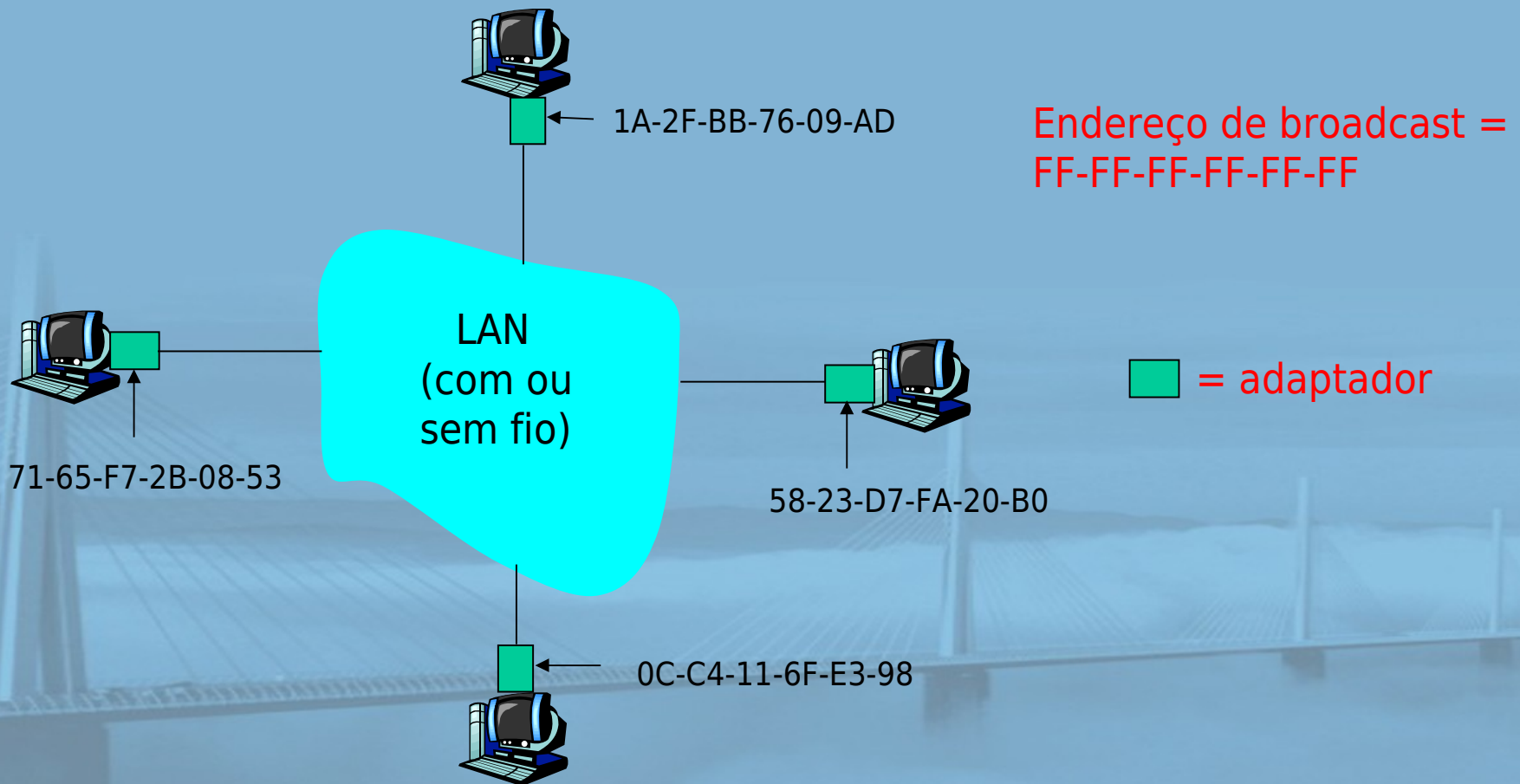
- ❑ 5.1 Introdução e serviços
- ❑ 5.2 Detecção e correção de erros
- ❑ 5.3 Protocolos de acesso múltiplo
- ❑ 5.4 Endereçamento na camada de enlace
- ❑ 5.5 Ethernet
- ❑ 5.6 Comutadores de camada de enlace
- ❑ 5.7 PPP
- ❑ 5.8 Virtualização de enlace: MPLS
- ❑ 5.9 Um dia na vida de uma solicitação de página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

# Endereçamento MAC e ARP

- ❑ Endereço IP de 32 bits:
  - endereço da *camada de rede*
  - usado para obter datagrama até sub-rede IP de destino
- ❑ Endereço MAC (ou LAN ou físico ou Ethernet) :
  - função: *levar quadro de uma interface para outra interface conectada fisicamente (na mesma rede)*
  - Endereço MAC de 48 bits (para maioria das LANs)
    - queimado na ROM da NIC, às vezes também configurável por software

# Endereços de LAN e ARP

Cada adaptador na LAN tem endereço de LAN exclusivo

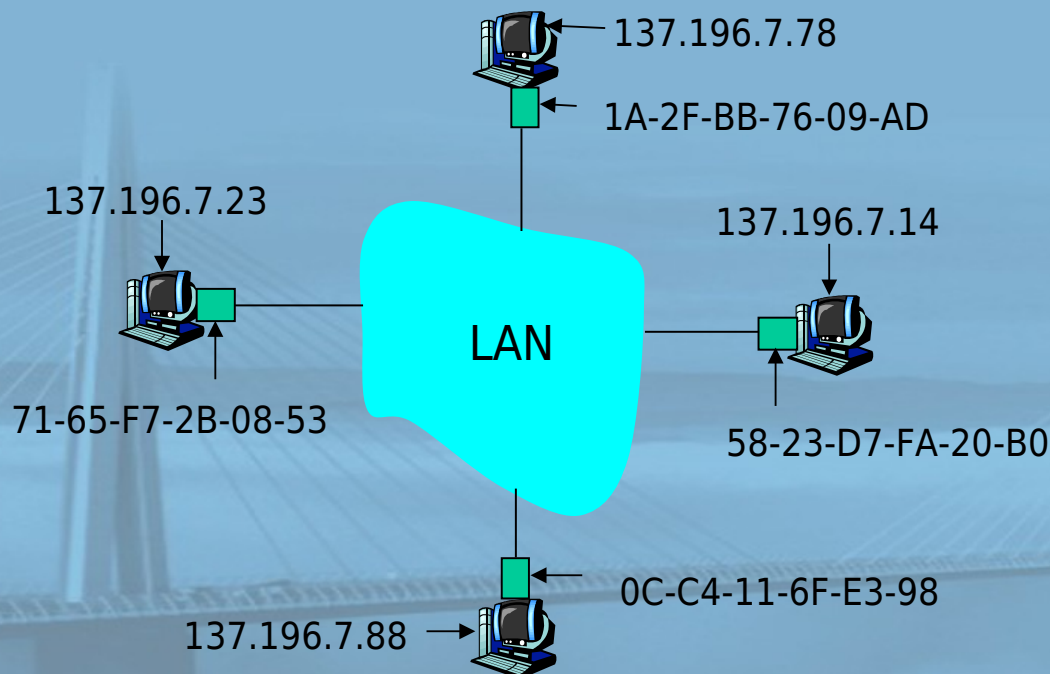


## Endereços de LAN (mais)

- ❑ alocação de endereço MAC administrada pelo IEEE
- ❑ fabricante compra parte do espaço de endereços MAC (para garantir exclusividade)
- ❑ analogia:
  - (a) Endereço MAC: como o CPF
  - (b) Endereço IP: como o endereço postal
- ❑ endereço MAC plano → portabilidade
  - pode mover placa de LAN de uma LAN para outra
- ❑ endereço IP hierárquico NÃO portátil
  - endereço depende da sub-rede IP à qual o nó está conectado

# ARP: Address Resolution Protocol

Pergunta: Como determinar endereço MAC de B sabendo o endereço IP de B?



- ❑ Cada nó IP (hosp., roteador) na LAN tem tabela **ARP**
- ❑ Tabela ARP: mapeamentos de endereço IP/MAC para alguns nós da LAN  
<endereço IP; endereço MAC; TTL>
  - TTL (Time To Live): tempo após o qual o mapeamento de endereço será esquecido (normalmente, 20 min)

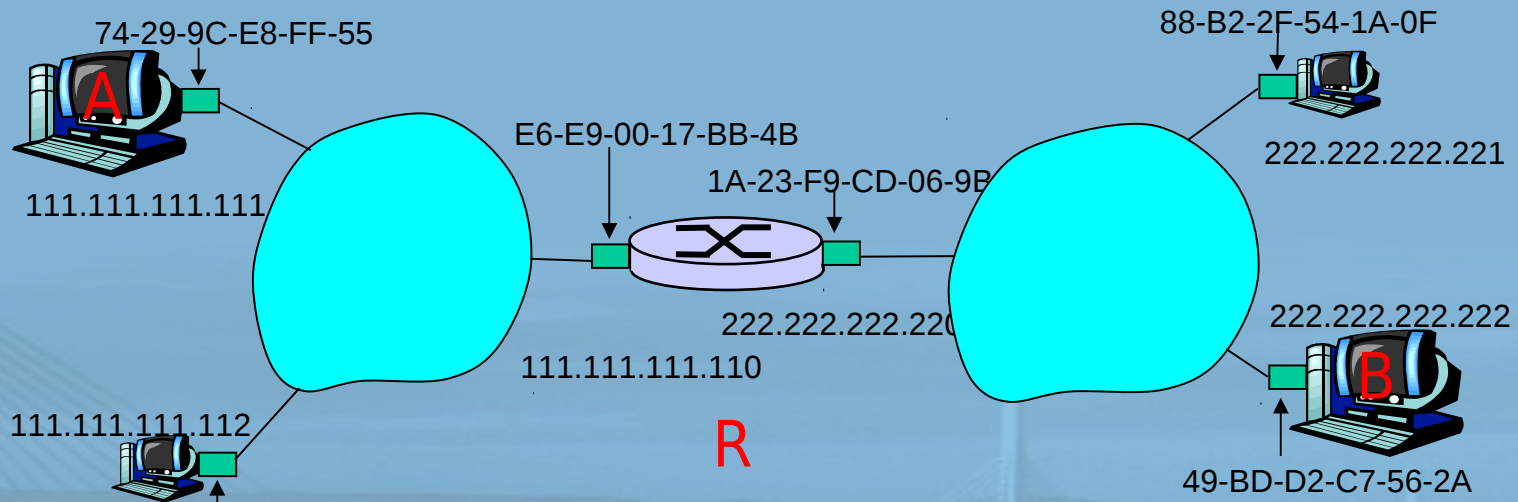


# Protocolo ARP: mesma LAN (rede)

- ❑ A quer enviar datagrama a B, e endereço MAC de B não está na tabela ARP de A.
- ❑ A envia por **broadcast** pacote de consulta ARP, contendo endereço IP de B
  - endereço MAC de destino = FF-FF-FF-FF-FF-FF
  - todas as máquinas na LAN recebem consulta ARP
- ❑ B recebe pacote ARP, responde para A com seu endereço MAC (de B)
  - quadro enviado ao endereço MAC de A (unicast)
- ❑ A salva em cache par de endereços IP-para-MAC em sua tabela ARP até a informação expirar
  - estado soft: informação que expira (desaparece) se não for renovada
- ❑ ARP é “plug-and-play”:
  - nós criam suas tabelas ARP *sem intervenção do administrador de rede*

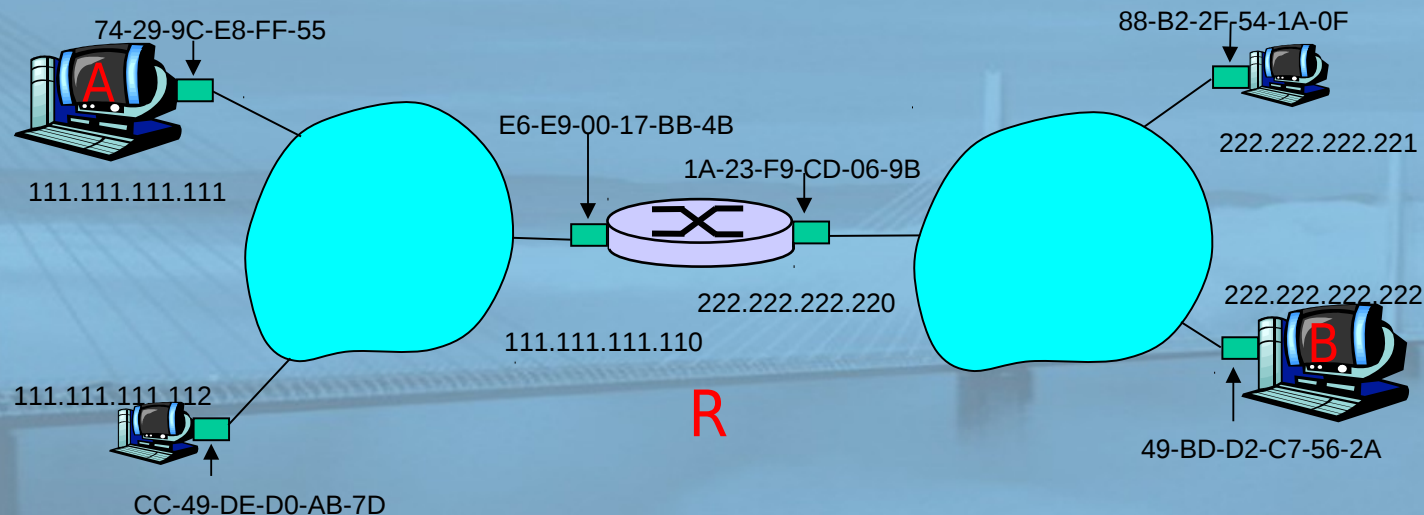
# Endereçamento: roteando para outra LAN

acompanhamento: **enviar datagrama de A para B via R**  
suponha que A saiba o endereço IP de B



- duas tabelas ARP no roteador R, uma para cada rede IP (LAN)

- ❑ A cria datagrama IP com origem A, destino B
- ❑ A usa ARP para obter endereço MAC de R para 111.111.111.110
- ❑ A cria quadro da camada de enlace com endereço MAC de R como destino, quadro contém datagrama IP A-para-B
- ❑ NIC de A envia quadro
- ❑ NIC de R recebe quadro
- ❑ R remove datagrama IP do quadro Ethernet, vê o seu destino a B
- ❑ R usa ARP para obter endereço MAC de B
- ❑ R cria quadro contendo datagrama IP A-para-B e envia para B



## Camada de enlace

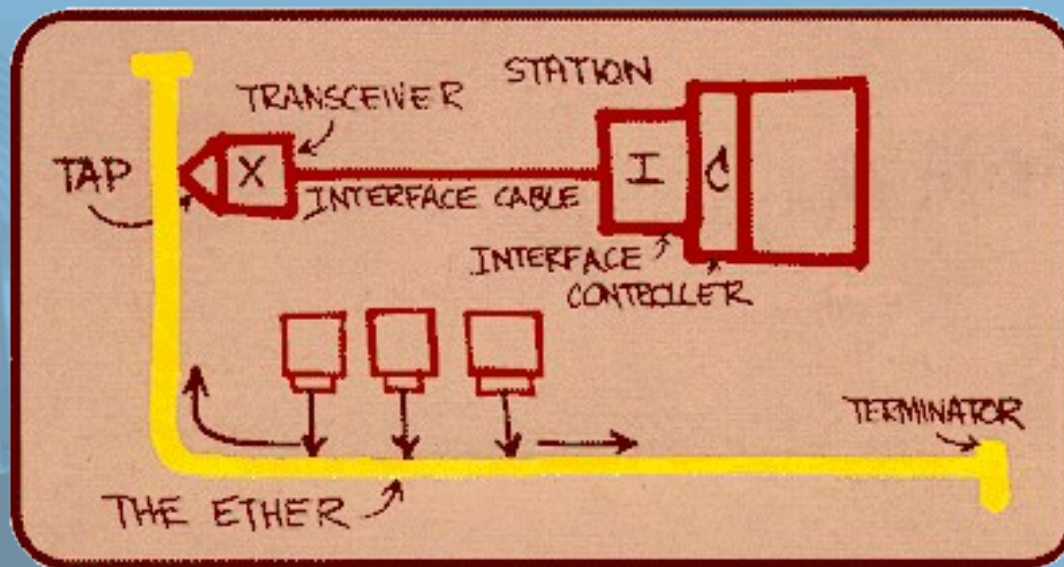
- ❑ 5.1 Introdução e serviços
- ❑ 5.2 Detecção e correção de erros
- ❑ 5.3 Protocolos de acesso múltiplo
- ❑ 5.4 Endereçamento na camada de enlace
- ❑ 5.5 Ethernet
- ❑ 5.6 Comutadores de camada de enlace
- ❑ 5.7 PPP
- ❑ 5.8 Virtualização de enlace: MPLS
- ❑ 5.9 Um dia na vida de uma solicitação de página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



# Ethernet

Tecnologia de LAN com fio “dominante”:

- ❑ barata
- ❑ primeira tecnologia de LAN utilizada em larga escala
- ❑ mais simples e mais barata que as LANs de permissão e ATM
- ❑ acompanhou corrida da velocidade: 10 Mbps - 10 Gbps

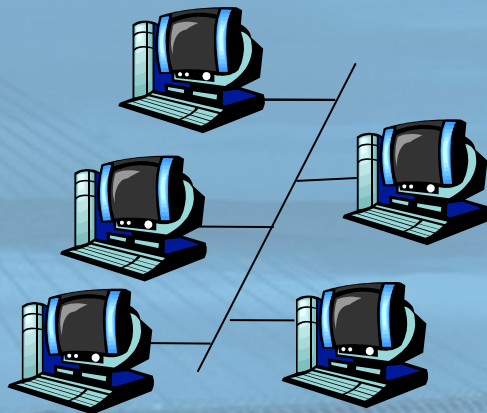


Projeto original da  
Ethernet de Metcalfe

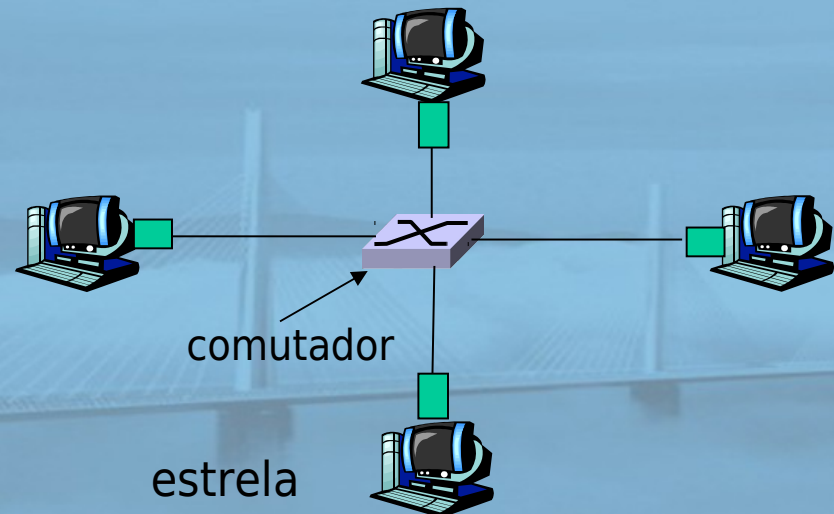


# Topologia de estrela

- ❑ topologia de barramento popular até meados dos anos 90
  - todos os nós no mesmo domínio de colisão (podem colidir uns com os outros)
- ❑ hoje: topologia de estrela prevalece
  - **comutador** ativo no centro
  - cada “ponta” roda um protocolo Ethernet (separado) - nós não colidem uns com os outros



barramento: cabo coaxial



estrela

# Estrutura do quadro Ethernet

Adaptador enviando encapsula datagrama IP (ou outro pacote de protocolo da camada de rede) no **quadro Ethernet**



## Preâmbulo:

- ❑ 7 bytes com padrão 10101010 seguido por um byte com padrão 10101011
- ❑ usado para sincronizar taxas de clock do receptor e emissor

## ❑ Endereços: 6 bytes

- se adaptador recebe quadro com endereço de destino combinando, ou com endereço de broadcast (p. e., pacote ARP), passa dados do quadro ao protocolo da camada de rede
- caso contrário, adaptador descarta quadro

## ❑ Tipo: indica protocolo da camada mais alta (principalmente IP, mas outros são possíveis, p. e., Novell IPX, AppleTalk)

## ❑ Dados: mínimo 64 bytes, máximo 1518

## ❑ CRC: verificado no receptor; se detectar erro, quadro é descartado



# Ethernet: não confiável, sem conexão

- ❑ **sem conexão:** sem apresentação entre NICs de origem e destino
- ❑ **não confiável:** NIC de destino não envia confirmações ou não confirmações à NIC de origem
  - fluxo de datagramas passados à camada de rede pode ter lacunas (datagramas faltando)
  - lacunas serão preenchidas se aplicação estiver usando TCP
  - caso contrário, aplicação verá lacunas
- ❑ Protocolo MAC da Ethernet: **CSMA/CD** não slotted

# 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  $\mu$ 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  $\{0,1,2,3,4,...,1023\}$

## Eficiência do CSMA/CD

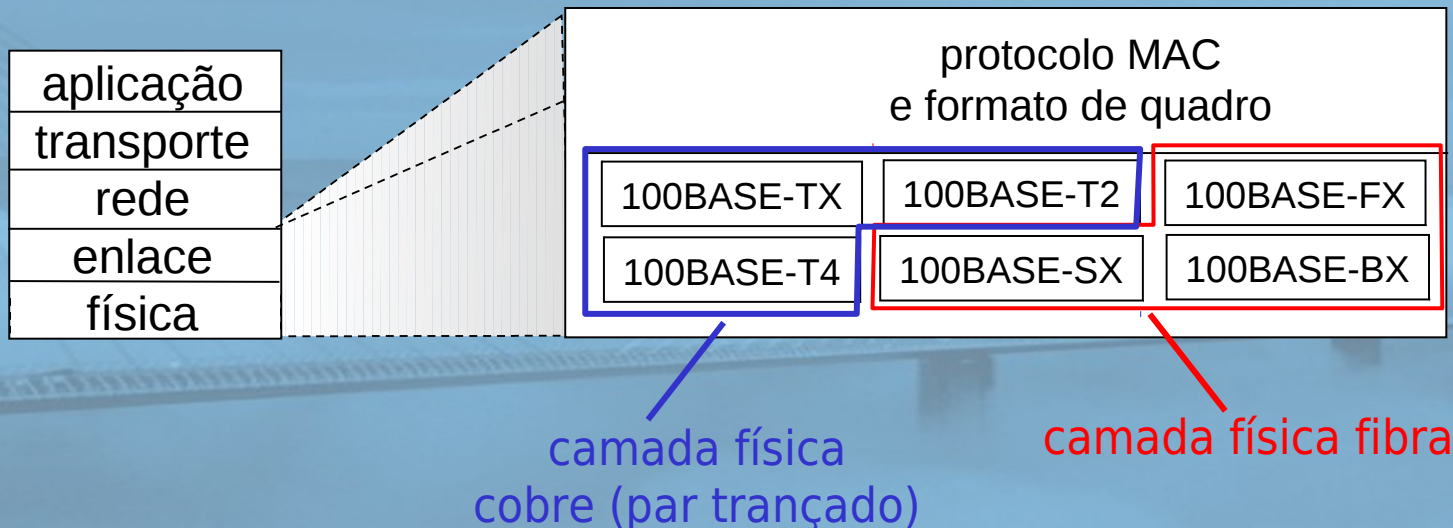
- ❑  $t_{\text{prop}}$  = atraso máx. propag. entre 2 nós na LAN
- ❑  $t_{\text{trans}}$  = tempo para transmitir quadro de tamanho máximo

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

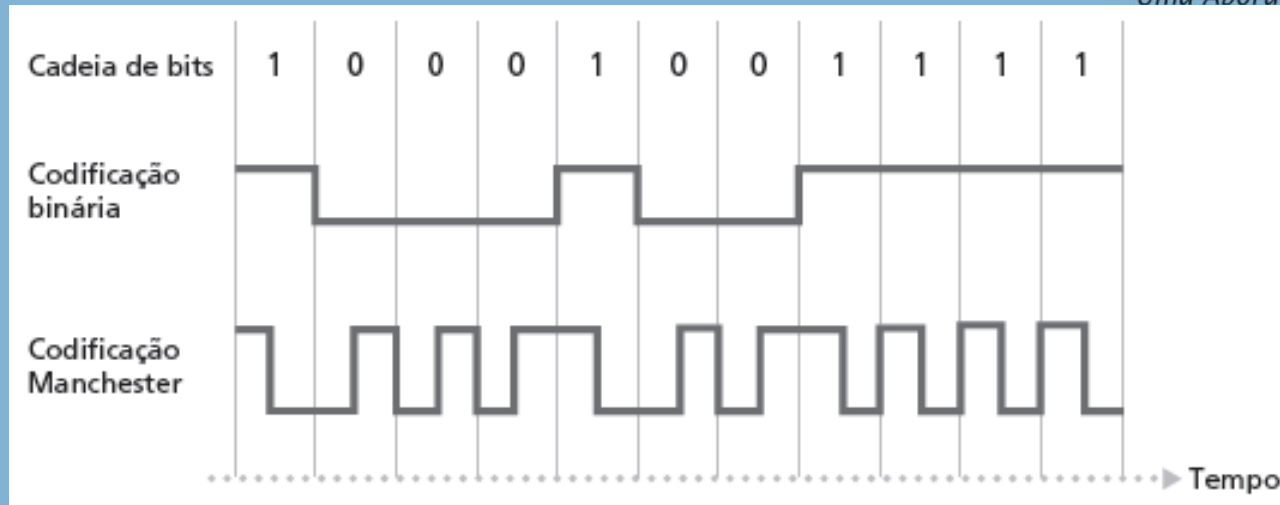
- ❑ eficiência vai para 1
  - quando  $t_{\text{prop}}$  vai para 0
  - quando  $t_{\text{trans}}$  vai para infinito
- ❑ melhor desempenho que ALOHA: é simples, barato, descentralizado!

# Padrões Ethernet 802.3: camadas de enlace e física

- ❑ *muitos* padrões Ethernet diferentes
  - protocolo MAC e formato de quadro comuns
  - diferentes velocidades: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10G bps
  - diferentes meios da camada física: fibra, cabo



# Codificação Manchester



- ❑ usado no 10BaseT
- ❑ cada bit tem uma transição
- ❑ permite que clocks nos nós emissor e receptor sejam sincronizados entre si
  - desnecessário para um clock centralizado, global entre os nós!

# Capítulo 6: Esboço

## ❑ 6.1 Introdução

### Redes sem fio

## ❑ 6.2 Características de enlaces e redes sem fio

- CDMA

## ❑ 6.3 LANs sem fio 802.11 (“wi-fi”)

## ❑ 6.4 Acesso celular à Internet

- arquitetura
- padrões (p. e., GSM)

## Mobilidade

## ❑ 6.5 Gerenciamento da mobilidade: princípios

## ❑ 6.6 IP móvel

## ❑ 6.7 Gerenciamento de mobilidade em redes celulares

## ❑ 6.8 Mobilidade e protocolos de camadas superiores

## ❑ 6.9 Resumo



# Características do enlace sem fio

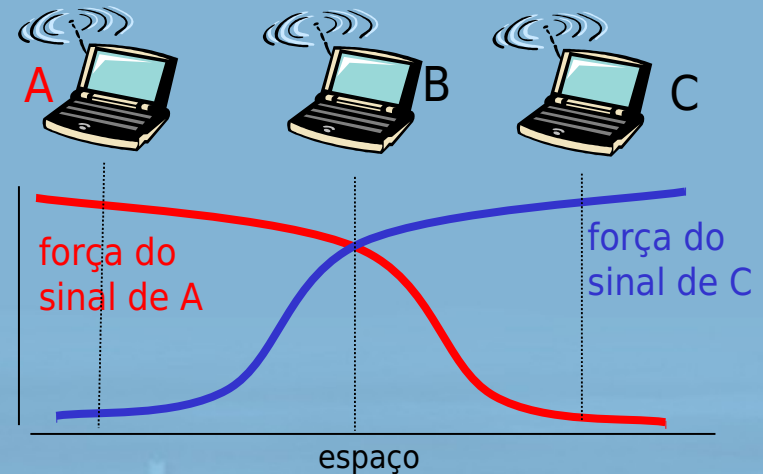
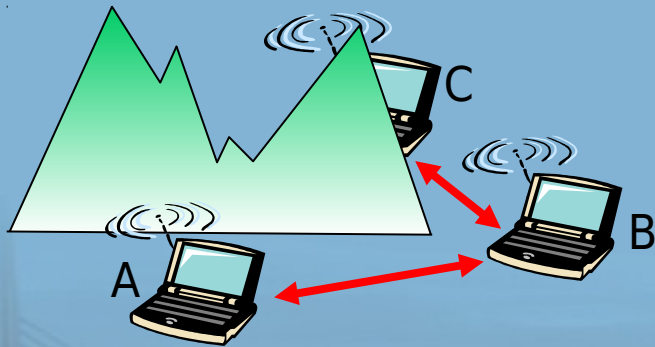
Diferenças do enlace com fio...

- **Redução fora do sinal:** sinal de rádio se atenua enquanto se propaga pela matéria (perda do caminho)
- **interferência de outras fontes:** frequências padrão de rede sem fio (p. e., 2,4 GHz) compartilhadas por outros dispositivos (p. e., telefone); dispositivos (motores) também interferem
- **propagação multivias:** sinal de rádio reflete-se em objetos e no solo, chegando ao destino em momentos ligeiramente diferentes

... tornam a comunicação por enlace sem fio muito mais “difícil”

# Características da rede sem fio

Múltiplos remetentes e receptores sem fio criam problemas adicionais (além do acesso múltiplo):



## Problema do terminal oculto

- ☐ B, A escutam um ao outro
- ☐ B, C escutam um ao outro
- ☐ A, C não podem ouvir um ao outro

significa que A, C não sabem de sua interferência em B

## Atenuação do sinal:

- ☐ B, A escutam um ao outro
- ☐ B, C escutam um ao outro
- ☐ A, C não podem escutar um ao outro interferindo em B

# LAN sem fio IEEE 802.11

## ❑ 802.11b

- espectro não licenciado de 2,4-5 GHz
- até 11 Mbps
- Direct Sequence Spread Spectrum (DSSS) na camada física
  - todos os hospedeiros usam o mesmo código de chipping

## ❑ 802.11a

- intervalo 5-6 GHz
- até 54 Mbps

## ❑ 802.11g

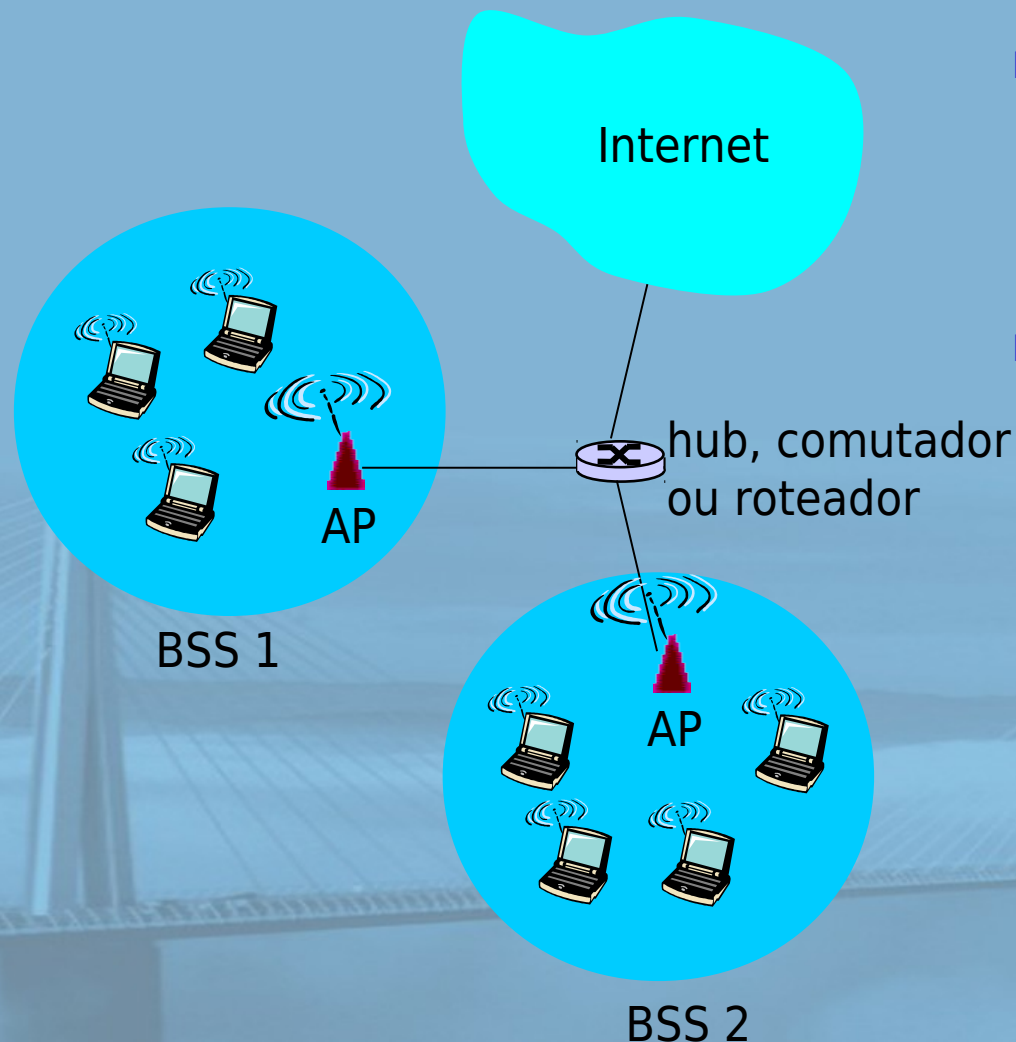
- intervalo 2,4-5 GHz
- até 54 Mbps

## ❑ 802.11n: múltiplas antenas

- intervalo 2,4-5 GHz
- até 200 Mbps

- 
- ❑ todos usam CSMA/CA para acesso múltiplo
  - ❑ todos têm versões de estação-base e rede ad-hoc

# Arquitetura de LAN 802.11



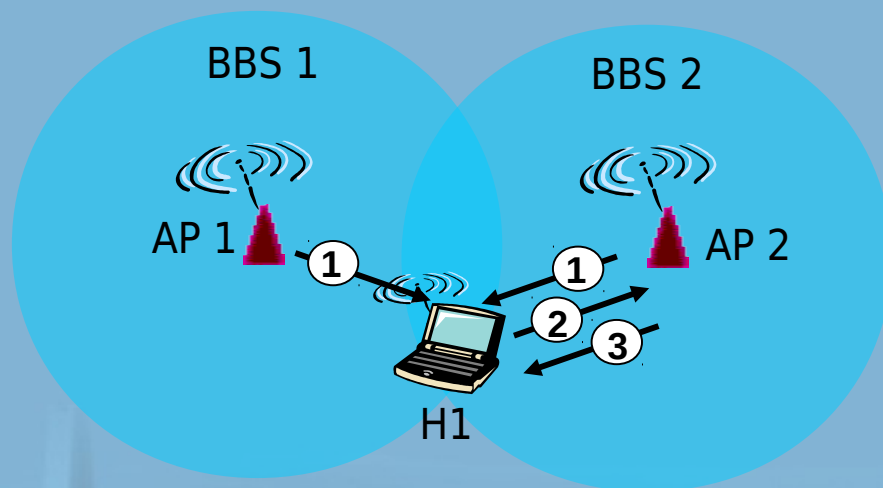
- hospedeiro sem fio se comunica com estação-base
  - estação-base = ponto de acesso (AP)
- **Basic Service Set (BSS)** (ou “célula”) no modo de infraestrutura contém:
  - hospedeiros sem fio
  - ponto de acesso (AP): estação-base
  - modo ad hoc: apenas hosts

## 802.11: Canais, associação

- ❑ 802.11b: espectro de 2,4 GHz-2,485 GHz dividido em 11 canais em diferentes frequências
  - Admin. do AP escolhe frequência para AP
  - possível interferência: canal pode ser o mesmo daquele escolhido pelo AP vizinho!
- ❑ hospedeiro: precisa *associar-se* a um AP
  - varre canais, escutando *quadros de sinalização* contendo nome do AP (SSID) e endereço MAC
  - seleciona AP para associar-se
  - pode realizar autenticação [Capítulo 8]
  - normalmente rodará DHCP para obter endereço IP na sub-rede do AP

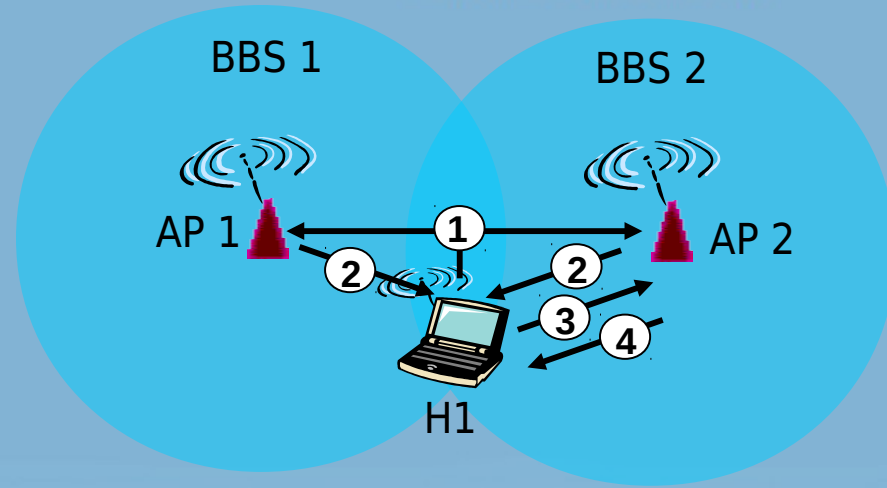


# 802.11: varredura passiva/ativa



## Varredura passiva:

- (1) quadros de sinalização enviados dos APs
- (2) quadro de solicitação de associação enviado: H1 para AP selecionado
- (3) quadro de resposta de associação enviado: H1 para AP selecionado

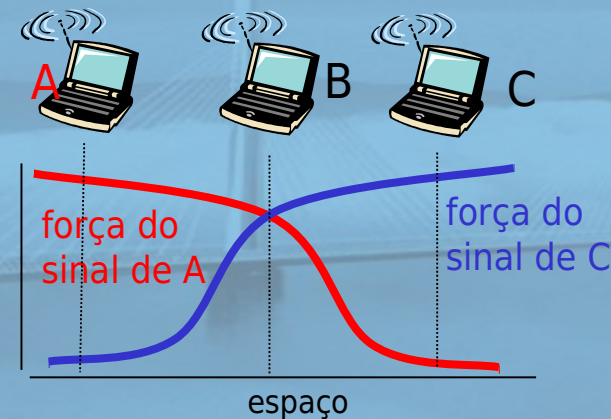
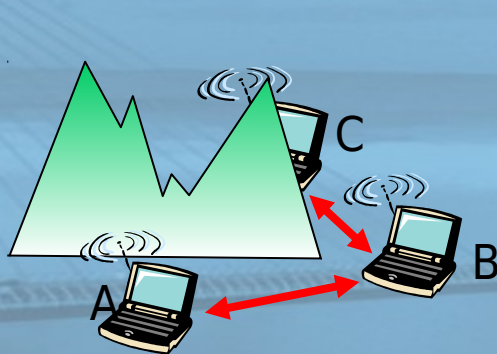


## Varredura ativa:

- (1) Broadcast de quadro de solicitação de investigação de H1
- (2) Quadro de resposta de investigações enviado de APs
- (3) Quadro de resposta de associação enviado: H1 para AP selecionado
- (4) Quadro de resposta de associação enviado: AP selecionado para H1

# 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

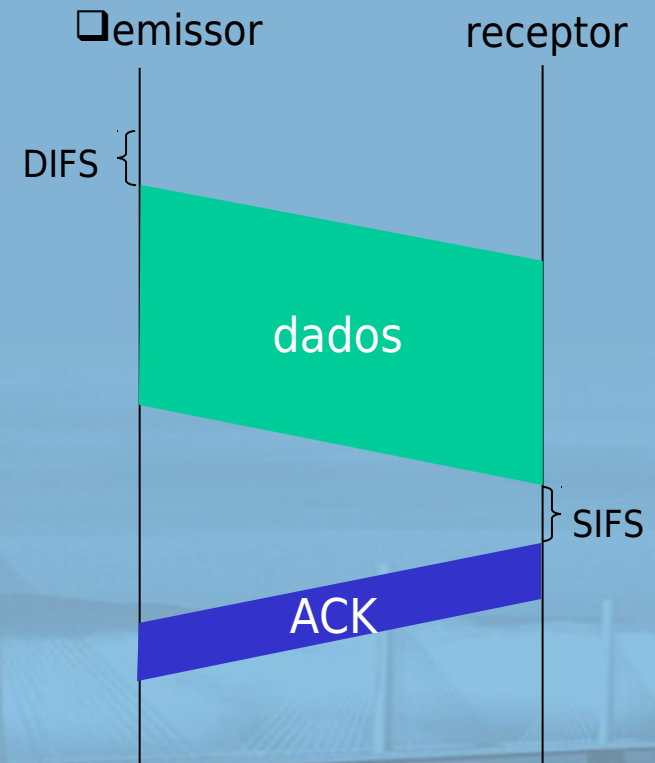
## Distribution Coordination Function remetente

- 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)

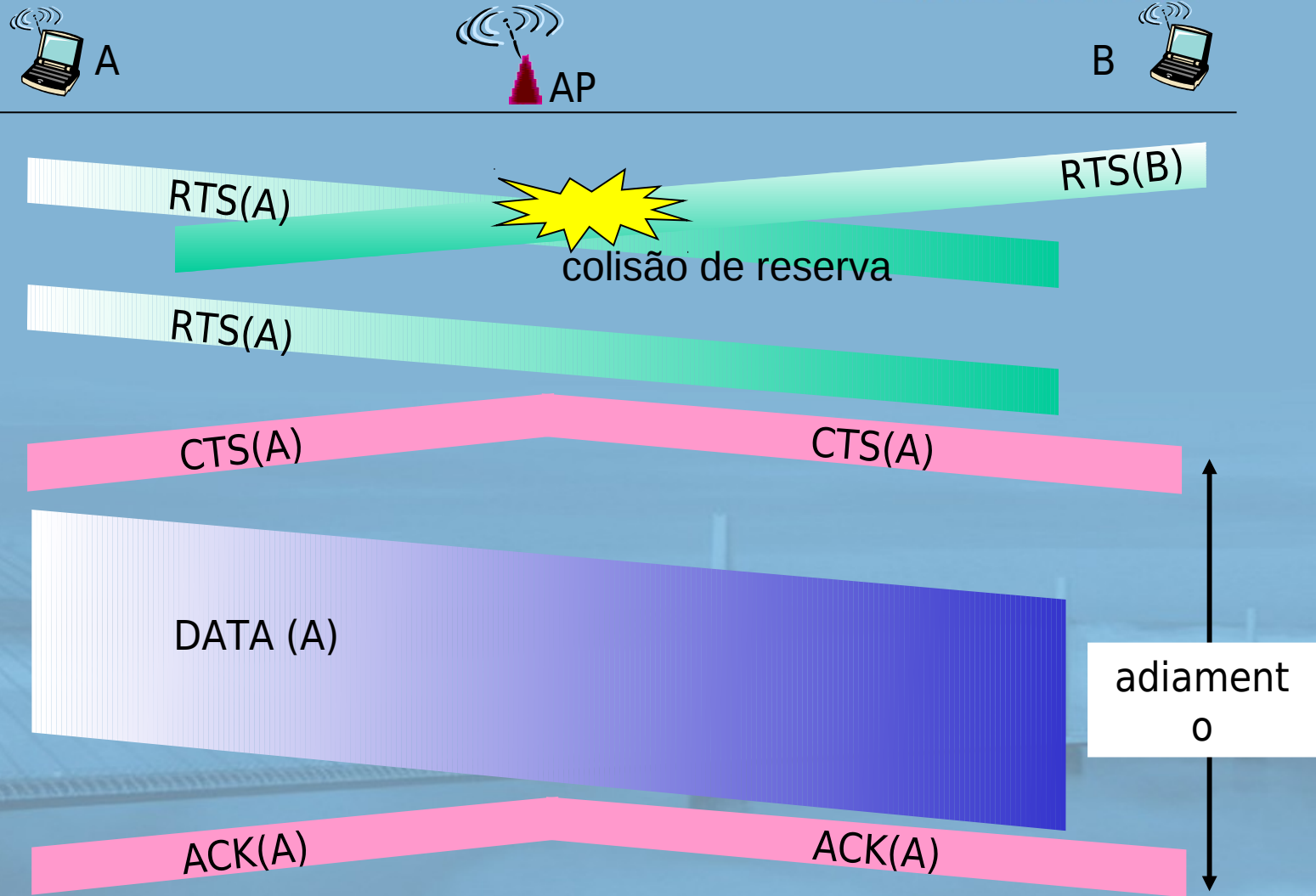
### *Point Coordination Function*

*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 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

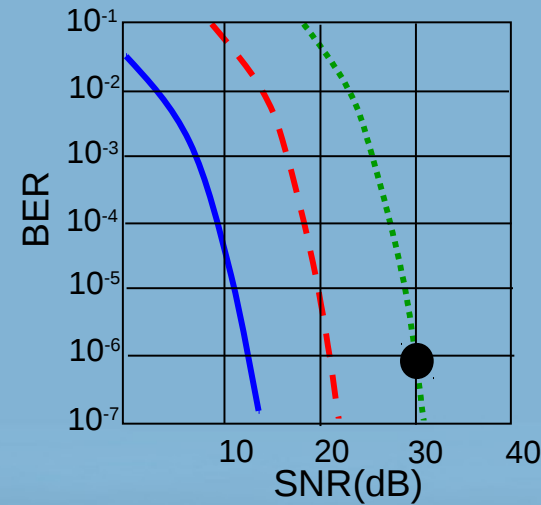
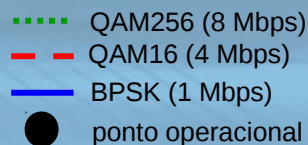




# 802.11: capacidades avançadas

## *Adaptação de taxa*

- estação-base, disp. móvel muda taxa de transmissão dinamicamente (técnica de modulação da camada física) enquanto móvel se move, SNR varia



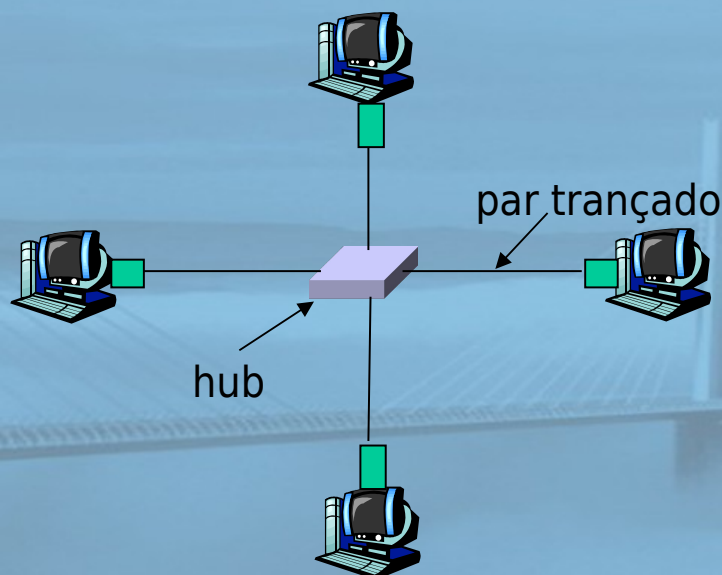
1. SNR diminui, BER aumenta quando nó se afasta da estação-base
2. Quando BER se torna muito alto, passa para taxa de transmissão inferior, mas com BER mais baixo

## 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

... repetidores da camada física (“burros”) :

- todos os nós conectados ao hub podem colidir uns com os outros
- sem buffering de quadros
- sem CSMA/CD no hub: NICs do hospedeiro detectam colisões
- bits chegando a um enlace saem em *todos* os outros enlaces na mesma velocidade

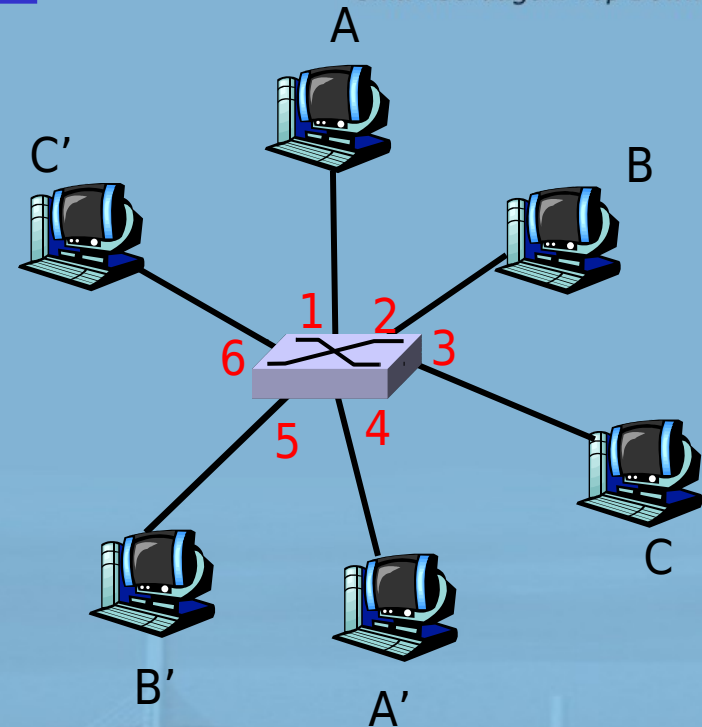


## Comutador (switch)

- ❑ dispositivo da camada de enlace: mais inteligente que os hubs, têm papel *ativo*
  - armazenam e repassam quadros Ethernet
  - examinam endereço MAC do quadro que chega, repassam **seletivamente** o quadro para um ou mais enlaces de saída quando o quadro deve ser repassado no segmento, usa CSMA/CD para acessar segmento
- ❑ *transparente*
  - hosps. não sabem da presença de comutadores
- ❑ *plug-and-play, autodidata*
  - comutadores não precisam ser configurados

# Comutador: permite *múltiplas* transmissões simultâneas

- ❑ hosps. têm conexão dedicada, direta com comutador
- ❑ comutadores mantêm pacotes
- ❑ Protocolo Ethernet usado em *cada* enlace de chegada, mas sem colisões; full duplex
  - cada enlace é seu próprio domínio de colisão
- ❑ **comutação:** A-para-A' e B-para-B' simultaneamente, sem colisões
  - não é possível com hub burro

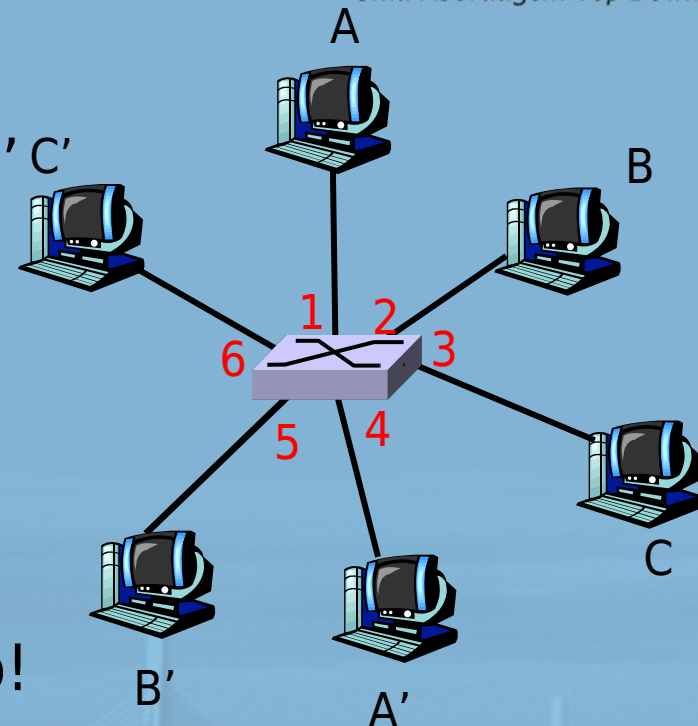


comutador com seis interfaces  
(1,2,3,4,5,6)



## Tabela de comutação

- ❑ P: Como o comutador sabe que A' se encontra na interface 4, B' C' se encontra na interface 5?
- ❑ R: Cada comutador tem uma **tabela de comutação**, cada entrada:
  - (endereço MAC do nó, interface para alcançar nó, horário)
- ❑ parece com tab. de roteamento!
- ❑ P: Como as entradas são criadas, mantidas na tabela comutação?
  - algo como um prot. de roteamento?

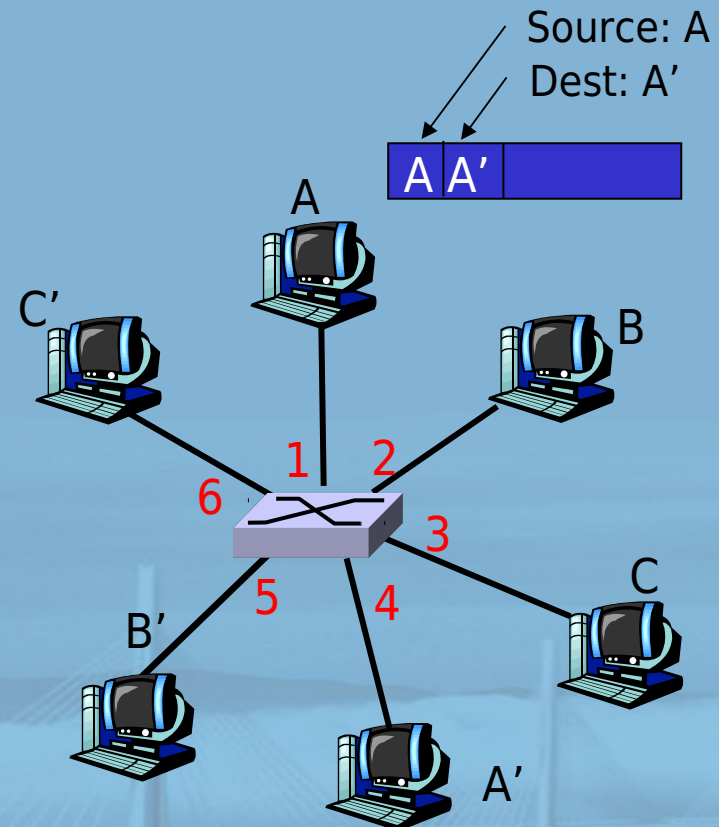


*comutador com 6 interfaces  
(1,2,3,4,5,6)*

## Comutador: autodidata

- ❑ comutador *descobre* quais nós podem ser alcançados por quais interfaces
  - quando quadro recebido, comutador “aprende” local do emissor: segmento de LAN de chegada
  - registra par emissor/local na tabela de comutação

end. MAC	interface	TTL
A	1	60



*Tabela comutação  
(inicialmente vazia)*

# Switch: filtragem/repasse de quadros

## Quando quadro recebido:

1. Registra enlace associado ao host emissor
2. Indexa tabela de comutação usando endereço MAC de destino
3. **if** entrada encontrada para o destino  
    **then {**  
        **if** dest no segmento do qual o quadro chegou  
            **then** remove o quadro  
            **else** repassa o quadro na interface indicada  
    **}**  
    **else** inunda

*repassa para todas as interfaces,  
menos aquela em que o quadro chegou*

## Autoaprendizagem, repasse: exemplo

- destino do quadro desconhecido: *inunda*
- local de destino A conhecido: *envio seletivo*

end. MAC	interface	TTL
A	1	60
A'	4	60

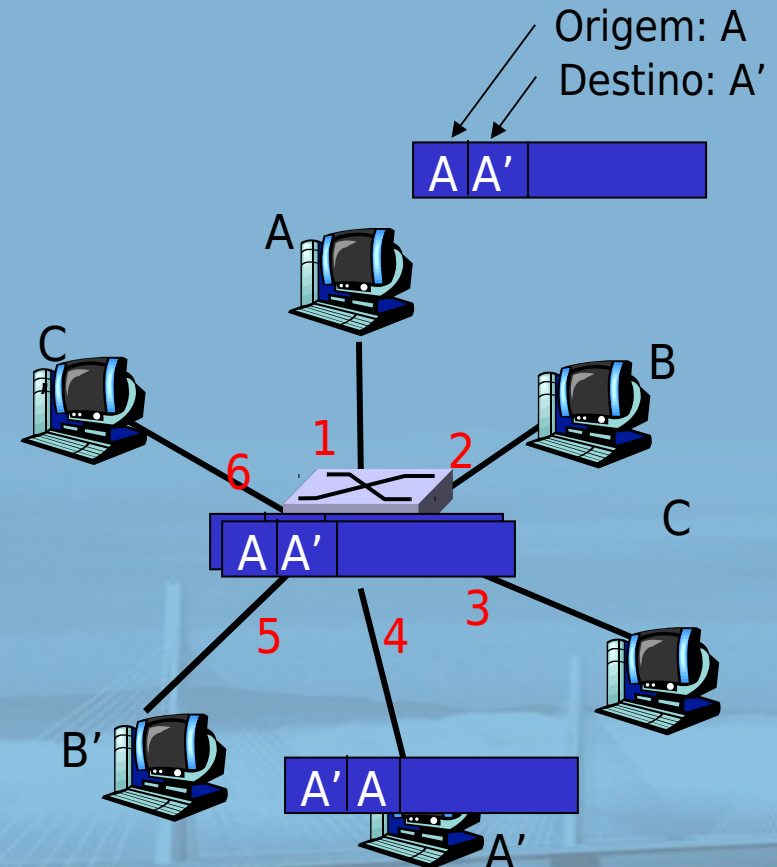
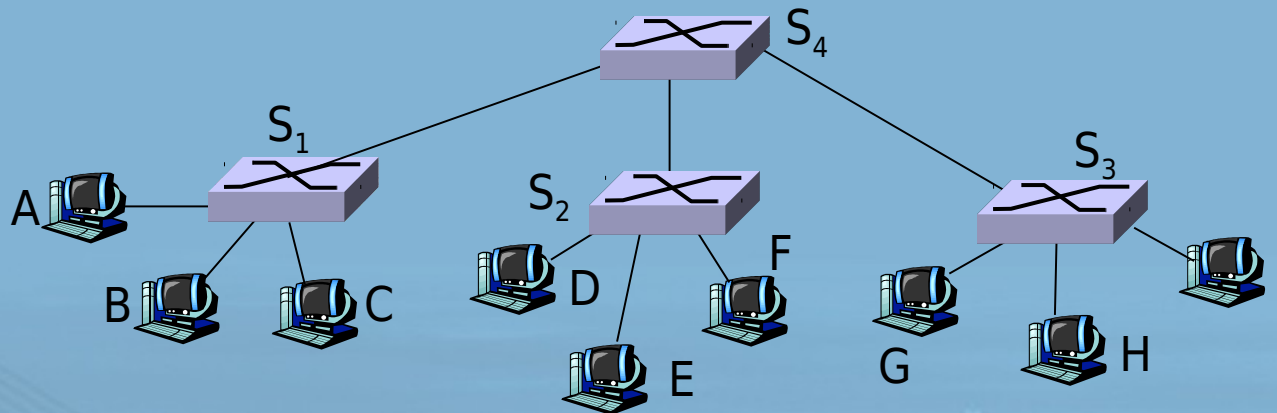


Tabela comutação  
(inicialmente vazia)

# Interconectando comutadores

- comutadores podem ser conectados

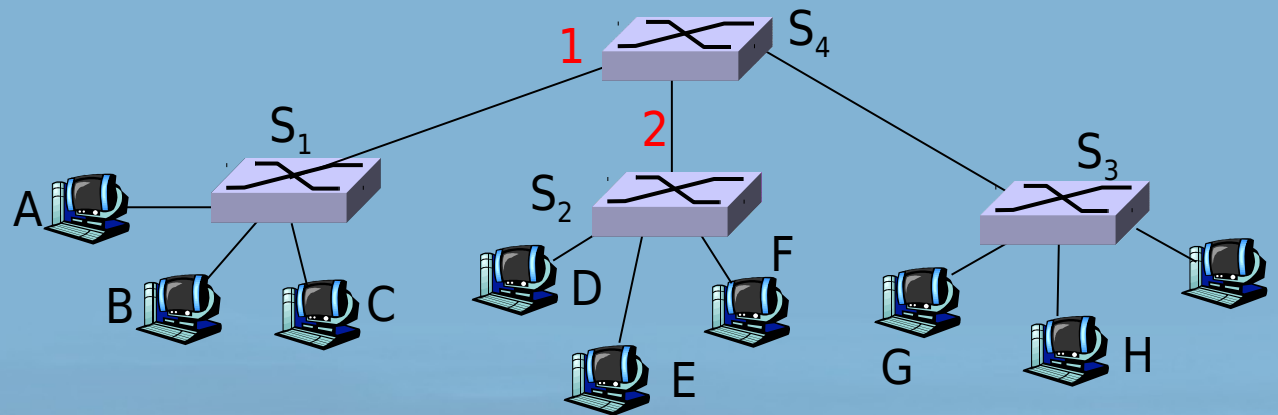


- P: Enviando de A p/ G - como S<sub>1</sub> sabe repassar quadro destinado a G por S<sub>4</sub> e S<sub>3</sub>?
- R: Autoaprendizagem! (funciona da mesma forma que no caso do único comutador!)



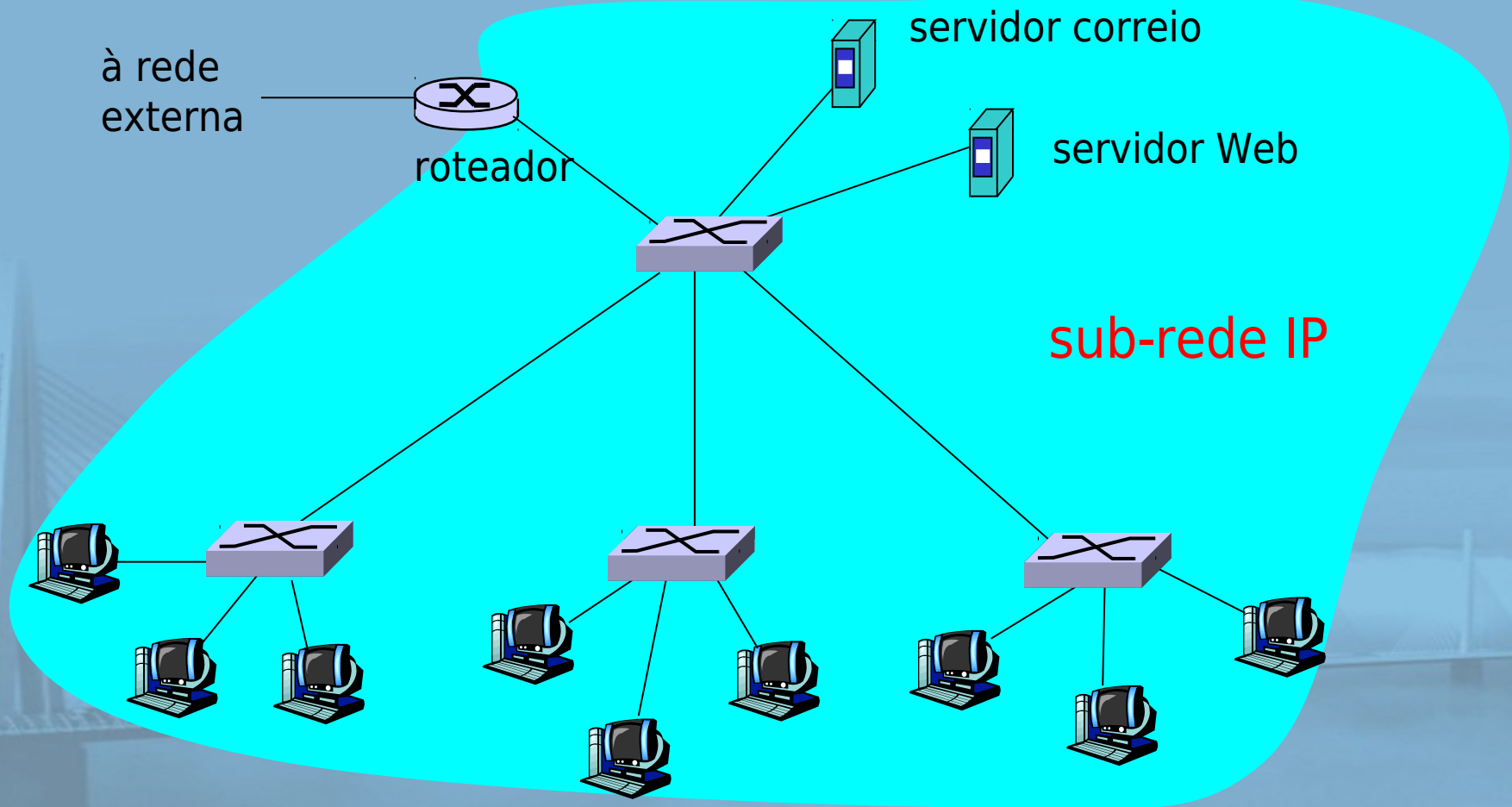
# Multicomutação com autoaprendizagem

Suponha que C envie quadro para I, I responde a C



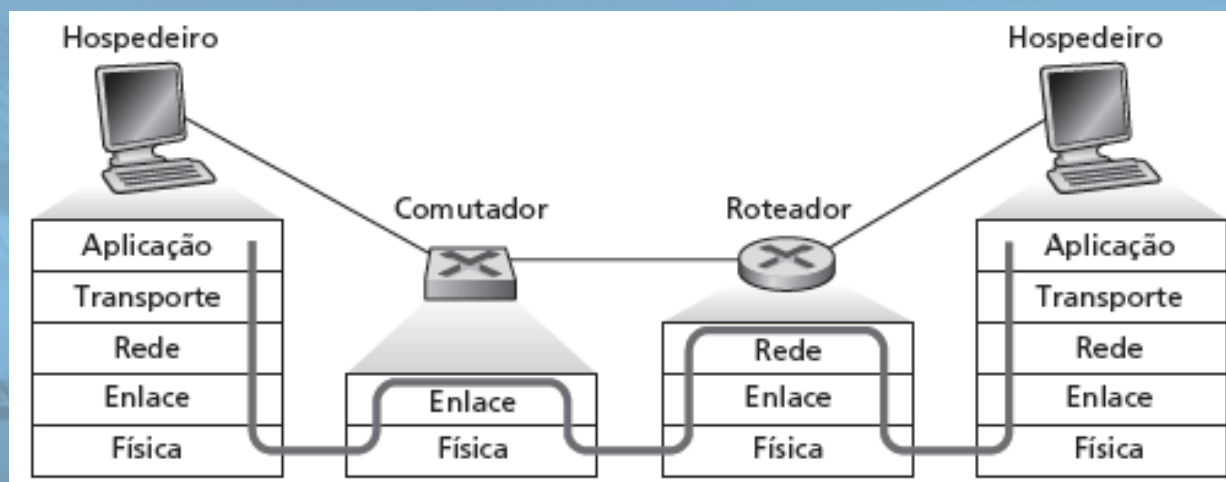
- P: Mostre tabelas de comutação e repasse de pacotes em  $S_1$ ,  $S_2$ ,  $S_3$ ,  $S_4$

## Rede institucional



# Comutadores versus roteadores

- ❑ ambos dispositivos de armazenamento e repasse
  - roteadores: dispositivos da camada de rede (examinam cabeçalhos da camada de rede)
  - comutadores são dispositivos da camada de enlace
- ❑ roteadores mantêm tabelas de roteamento, implementam algoritmos de roteamento
- ❑ switches mantêm tabelas de comutação, implementam filtragem, algoritmos de aprendizagem



## Camada de enlace

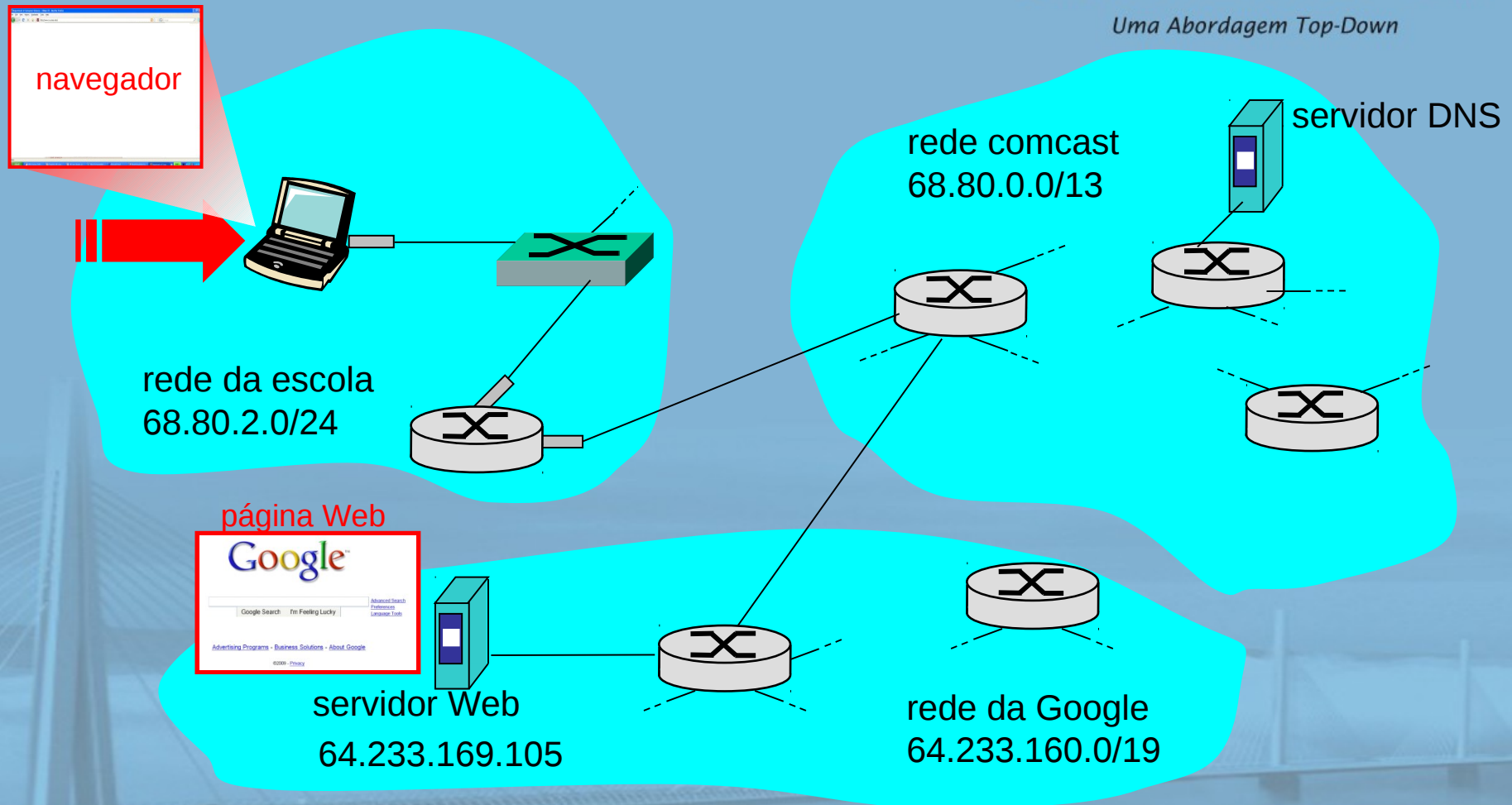
- ❑ 5.1 Introdução e serviços
- ❑ 5.2 Detecção e correção de erros
- ❑ 5.3 Protocolos de acesso múltiplo
- ❑ 5.4 Endereçamento na camada de enlace
- ❑ 5.5 Ethernet
- ❑ 5.6 Comutadores de camada de enlace
- ❑ 5.7 PPP
- ❑ 5.8 Virtualização de enlace: MPLS
- ❑ 5.9 Um dia na vida de uma solicitação de página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

# Síntese: um dia na vida de uma solicitação Web

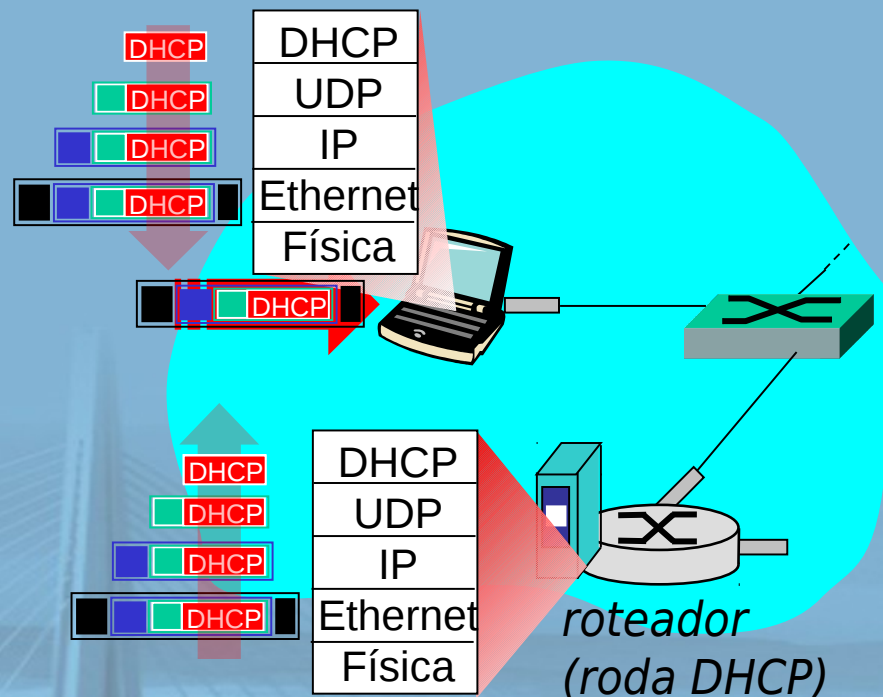
- ❑ viagem pela pilha de protocolos completa!
  - aplicação, transporte, rede, enlace
- ❑ juntando tudo: síntese!
  - *objetivo*: identificar, analisar, entender os protocolos (em todas as camadas) envolvidos no cenário aparentemente simples: solicitar página WWW
  - *cenário*: aluno conecta laptop à rede do campus, solicita/recebe [www.google.com](http://www.google.com)



## Um dia na vida: cenário



# Um dia na vida... conectando à Internet

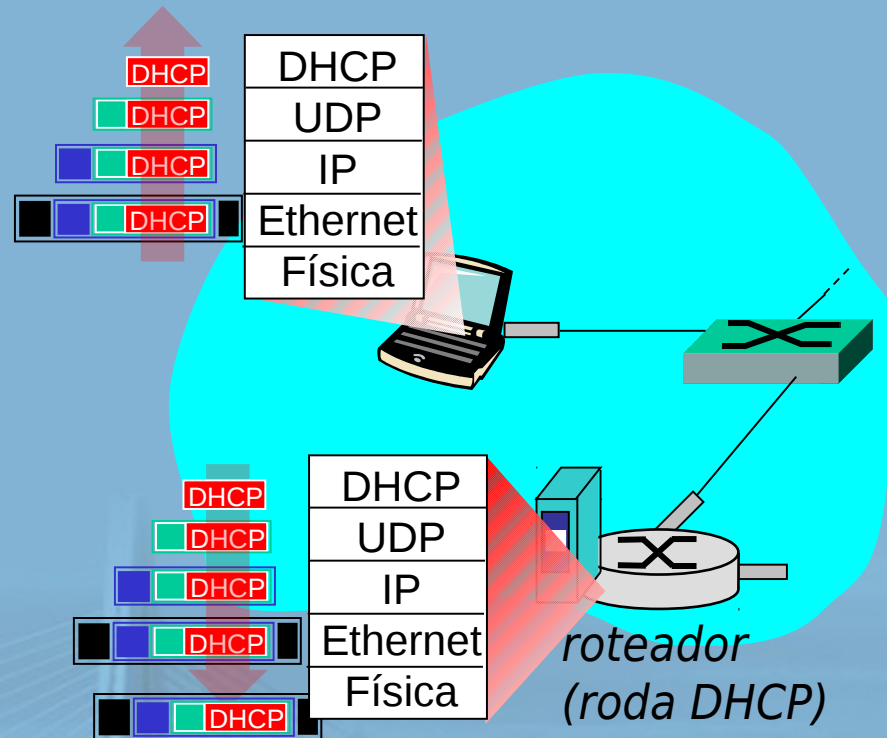


- o laptop conectando precisa obter seu próprio endereço IP, end. do roteador do 1º salto e do servidor DNS: use **DHCP**
- Solicitação DHCP **encapsulada** no **UDP**, encapsulada no **IP**, encapsulada na Ethernet **802.1**
- Quadro Ethernet enviado por **broadcast** (dest.: FFFFFFFFFFFFFFFF) na LAN, recebido no roteador rodando servidor **DHCP**
- Ethernet **demultiplexado** para IP demultiplexado, UDP demultiplexado para DHCP

# REDES DE COMPUTADORES E A INTERNET

5ª edição

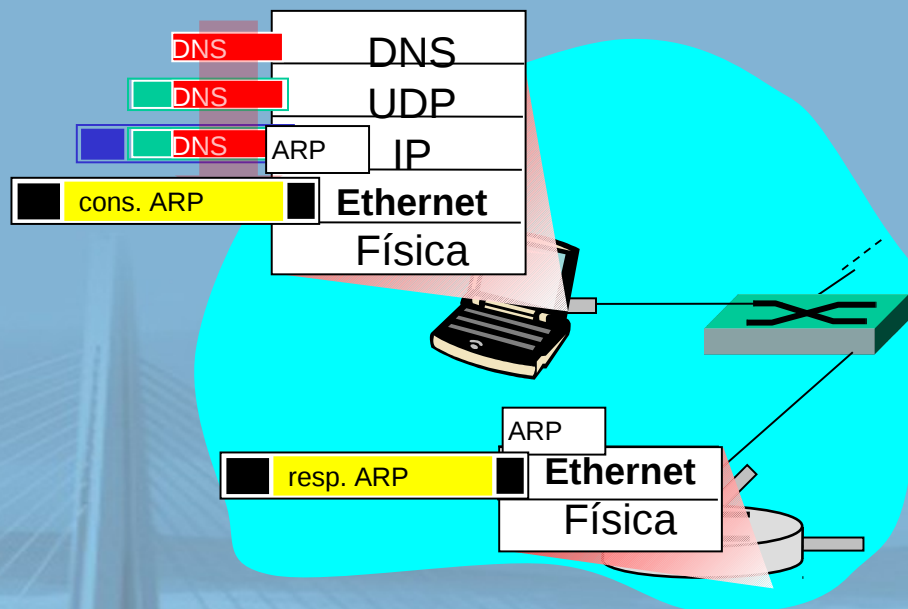
Uma Abordagem Top-Down



- ❑ Servidor DHCP formula **ACK DHCP** contendo endereço IP do cliente, IP do roteador no 1º salto para cliente, nome & endereço IP do servidor DNS
- ❑ Encapsulamento no servidor DHCP, quadro repassado (**aprendizagem do comutador**) através da LAN, demultiplexando no cliente
- ❑ Cliente DHCP recebe resposta ACK do DHCP

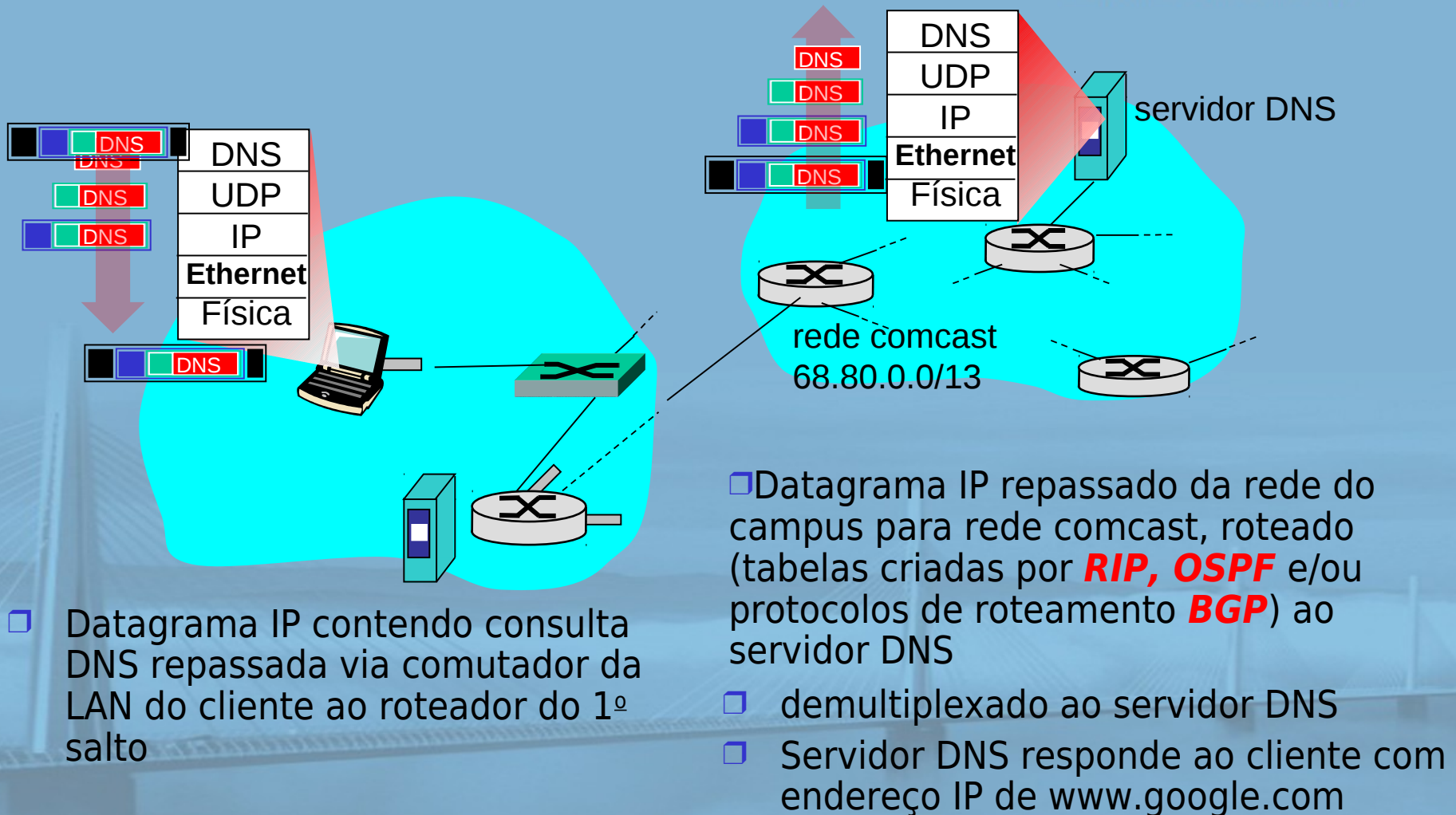
*Cliente agora tem endereço IP, sabe nome e endereço do servidor DNS, endereço IP do seu roteador no primeiro salto*

# Um dia na vida... ARP (antes do DNS, antes do HTTP)



- ❑ Antes de enviar solicitação **HTTP**, precisa de endereço IP de `www.google.com`: **DNS**
- ❑ Consulta DNS criada, encap. no UDP, no IP, na Ethernet. Para enviar quadro ao roteador, precisa de endereço MAC da interface do roteador: **ARP**
- ❑ Broadcast da **consulta ARP**, recebido pelo roteador, que responde com **resposta ARP** dando endereço MAC da interface do roteador
- ❑ cliente agora sabe endereço MAC do roteador no 1º salto, e agora pode enviar quadro contendo consulta DNS

# Um dia na vida... usando DNS

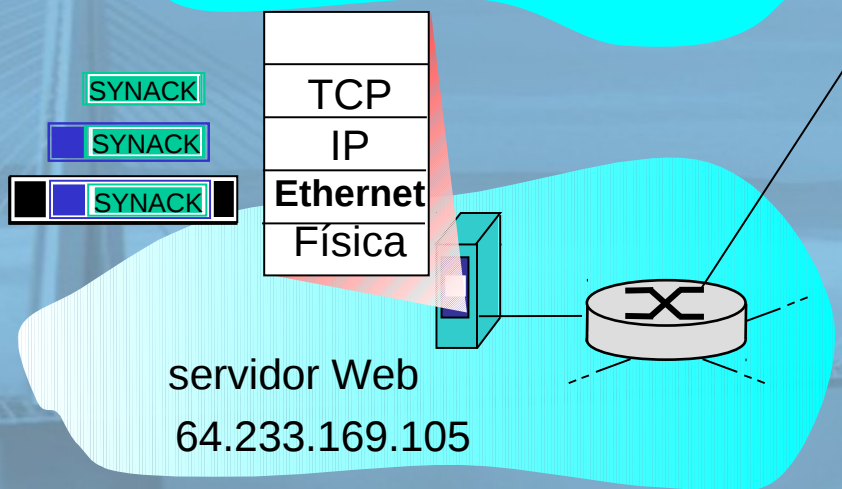
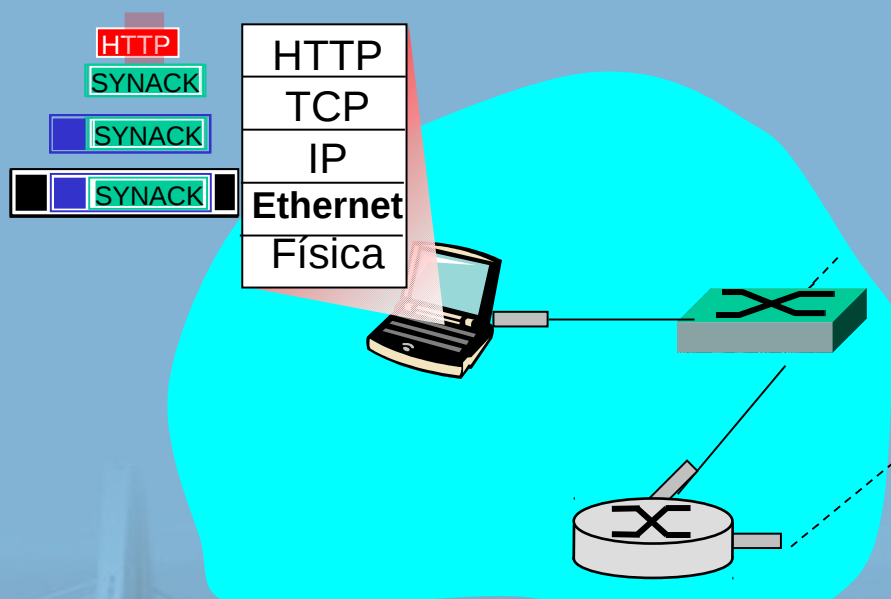




# Um dia na vida... conexão TCP transportando HTTP

REDES DE  
COMPUTADORES  
E A INTERNET 5ª edição

*Uma Abordagem Top-Down*

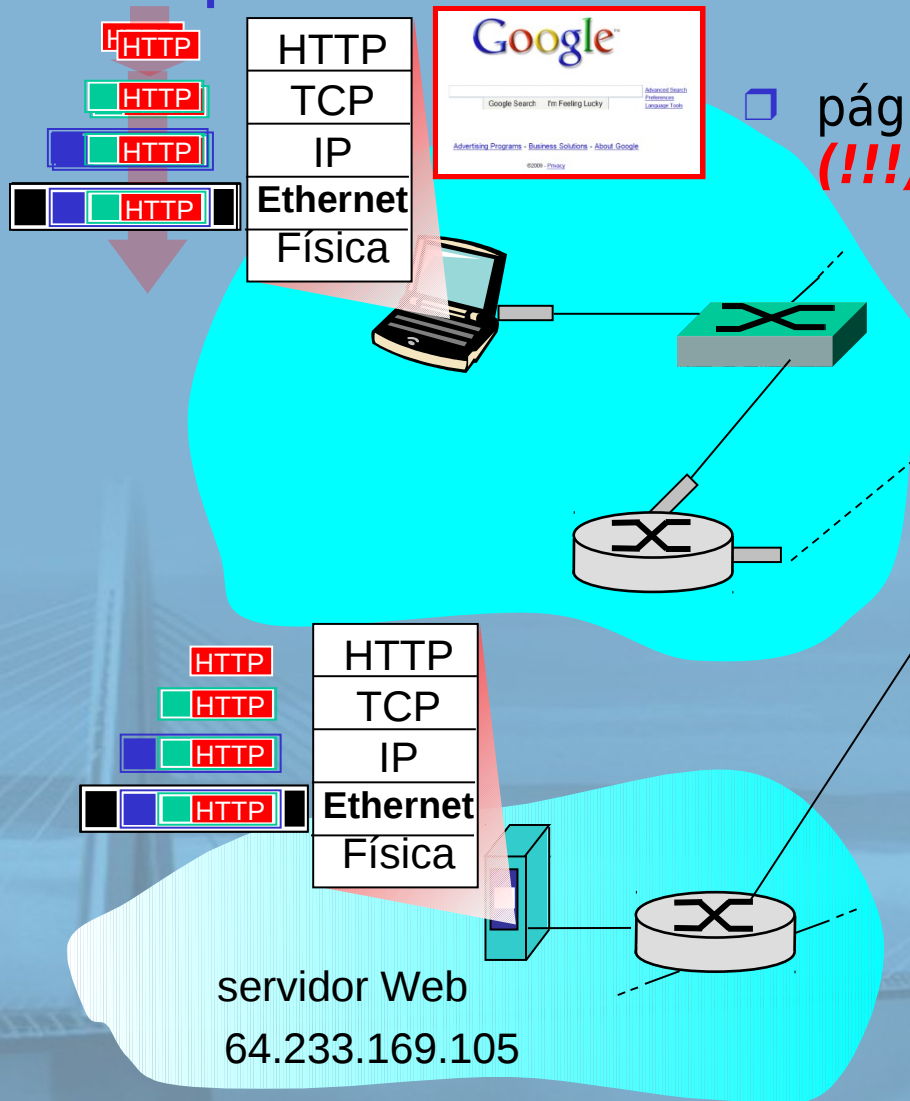


- para enviar solicitação HTTP, cliente primeiro abre **socket TCP** com servidor Web
- **segmento SYN** TCP (etapa 1 na apresentação de 3 vias) **roteado interdomínio** com servidor Web
- servidor Web responde com **SYNACK TCP** (etapa 2 na apresentação de 3 vias)
- **Conexão TCP estabelecida!**

# Um dia na via... solicitação/ resposta HTTP

□ página Web **finalmente**  
**(!!!)** exibida

- **solicitação HTTP** enviada ao socket TCP
- datagrama IP contendo solicitação HTTP roteado para `www.google.com`
- servidor Web responde com **resposta HTTP** (contendo página Web)
- datagrama IP contendo resposta HTTP roteada de volta ao cliente



## Capítulo 5: Resumo

- ❑ princípios por trás dos serviços da camada de enlace de dados:
  - detecção e correção de erro
  - compartilhamento de canal de broadcast: acesso múltiplo
  - endereçamento da camada de enlace
- ❑ instanciação e implementação de várias tecnologias da camada de enlace
  - Ethernet
  - LANS comutadas
  - WiFi
- ❑ síntese: um dia na vida de uma solicitação Web

# Capítulo 5: vamos fazer uma pausa

- ❑ viagem pela pilha de protocolos *completa* (exceto a camada física)
- ❑ conhecimento sólido dos princípios de rede e da prática
- ❑ Outros tópicos interessantes!
  - redes sem fio
  - multimídia
  - *segurança*
  - gerenciamento de rede