

# Capítulo 5

## Camada de enlace e redes locais

### Nota sobre o uso destes slides ppt:

Estamos disponibilizando estes slides gratuitamente a todos (professores, alunos, leitores). Eles estão em formato do PowerPoint para que você possa incluir, modificar e excluir slides (incluindo este) e o conteúdo do slide, de acordo com suas necessidades. Eles obviamente representam *muito* trabalho da nossa parte. Em retorno pelo uso, pedimos apenas o seguinte:

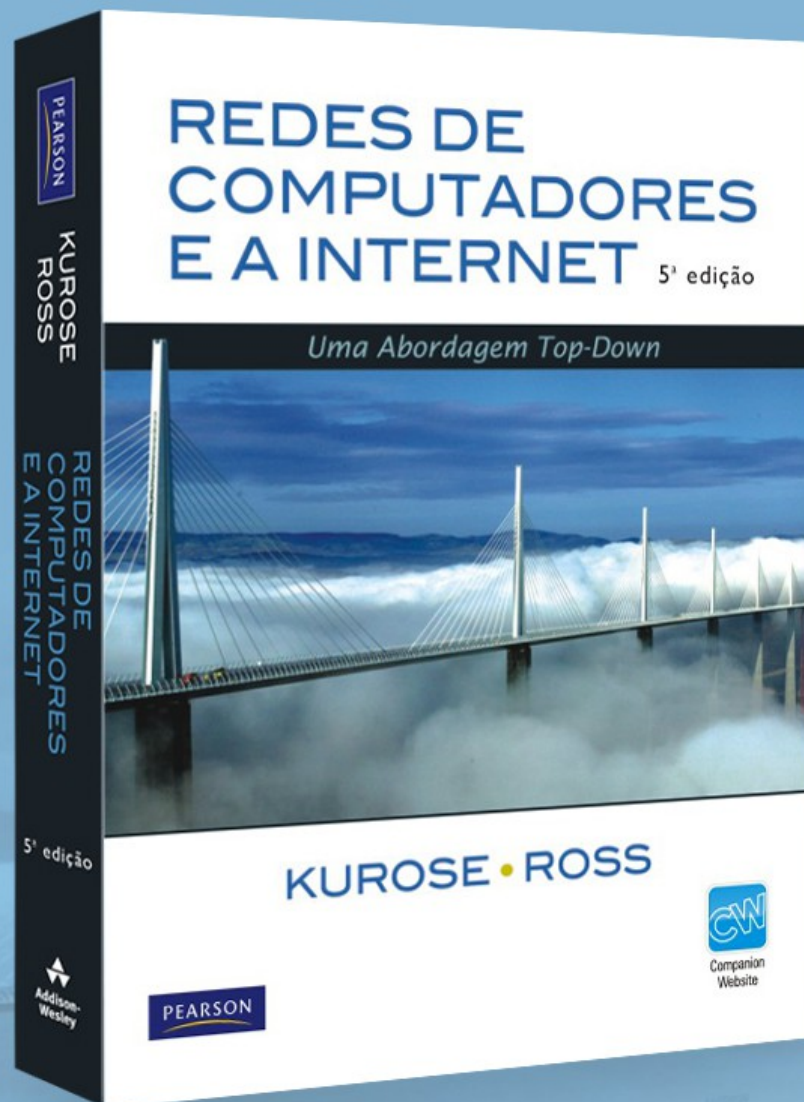
Se você usar estes slides (por exemplo, em sala de aula) sem muita alteração, que mencione sua fonte (afinal, gostamos que as pessoas usem nosso livro!).

Se você postar quaisquer slides sem muita alteração em um site Web, que informe que eles foram adaptados dos (ou talvez idênticos aos) nossos slides, e inclua nossa nota de direito autoral desse material.

Obrigado e divirta-se! JFK/KWR

Todo o material copyright 1996-2009

J. F. Kurose e K. W. Ross, Todos os direitos reservados



# Capítulo 5: A camada de enlace

## Objetivos do capítulo:

- ρ entender os princípios por trás dos serviços da camada de enlace de dados:
  - μ detecção e correção de erro
  - μ Compartilhamento de um canal de broadcast: acesso múltiplo
  - μ endereçamento da camada de enlace
  - μ transferência de dados confiável, controle de fluxo: *feito!*
- ρ instanciação e implementação de várias tecnologias da camada de enlace

## Camada de enlace

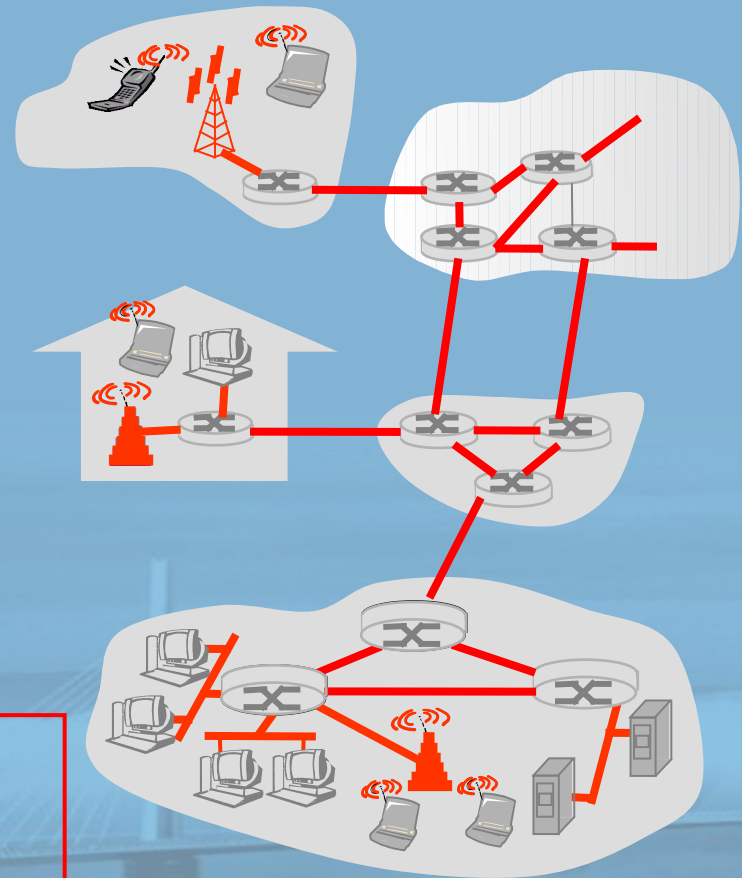
- ρ 5.1 Introdução e serviços
- ρ 5.2 Detecção e correção de erros
- ρ 5.3 Protocolos de acesso múltiplo
- ρ 5.4 Endereçamento na camada de enlace
- ρ 5.5 Ethernet
- ρ 5.6 Comutadores de camada de enlace
- ρ 5.7 PPP
- ρ 5.8 Virtualização de enlace: MPLS
- ρ 5.9 Um dia na vida de uma solicitação de página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

# Camada de enlace: introdução

## Alguna terminologia:

- ρ hospedeiros e roteadores são **nós**
- ρ canais de comunicação que se conectam a nós adjacentes pelo caminho de comunicação são **enlaces**
  - μ enlaces com fio
  - μ enlaces sem fio
  - μ LANs
- ρ 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

## ρ *enquadramento, acesso ao enlace:*

- μ encapsula datagrama no quadro, incluindo cabeçalho, trailer
- μ acesso ao canal de meio compartilhado
- μ endereços "MAC" usados nos cabeçalhos de quadro para identificar origem, destino
  - diferente do endereço IP!

## ρ *entrega confiável entre nós adjacentes*

- μ já aprendemos a fazer isso (Capítulo 3)!
- μ raramente usado em enlace com pouco erro de bit (fibra, alguns pares trançados)
- μ enlaces sem fio: altas taxas de erro
  - P: Por que confiabilidade em nível de enlace e fim a fim?

## ρ *controle de fluxo:*

- μ controle entre nós de emissão e recepção adjacentes

## ρ *detecção de erro:*

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

## ρ *correção de erro:*

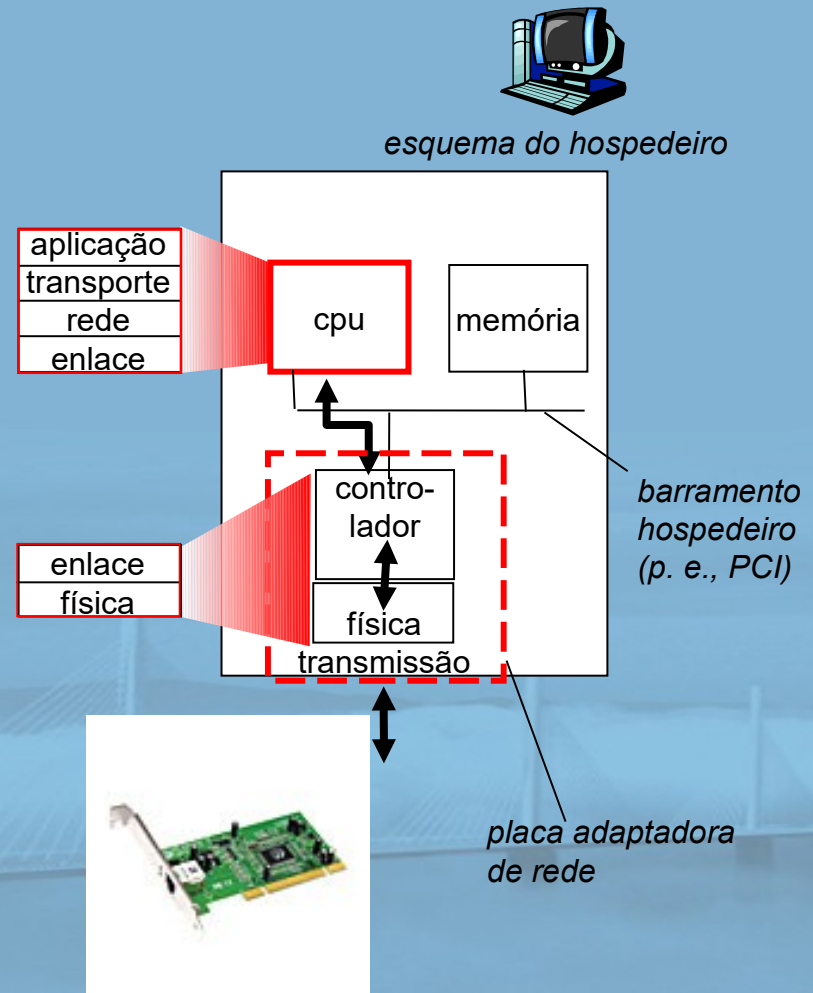
- μ receptor identifica *e corrige* erro(s) de bit sem lançar mão da retransmissão

## ρ *half-duplex e full-duplex*

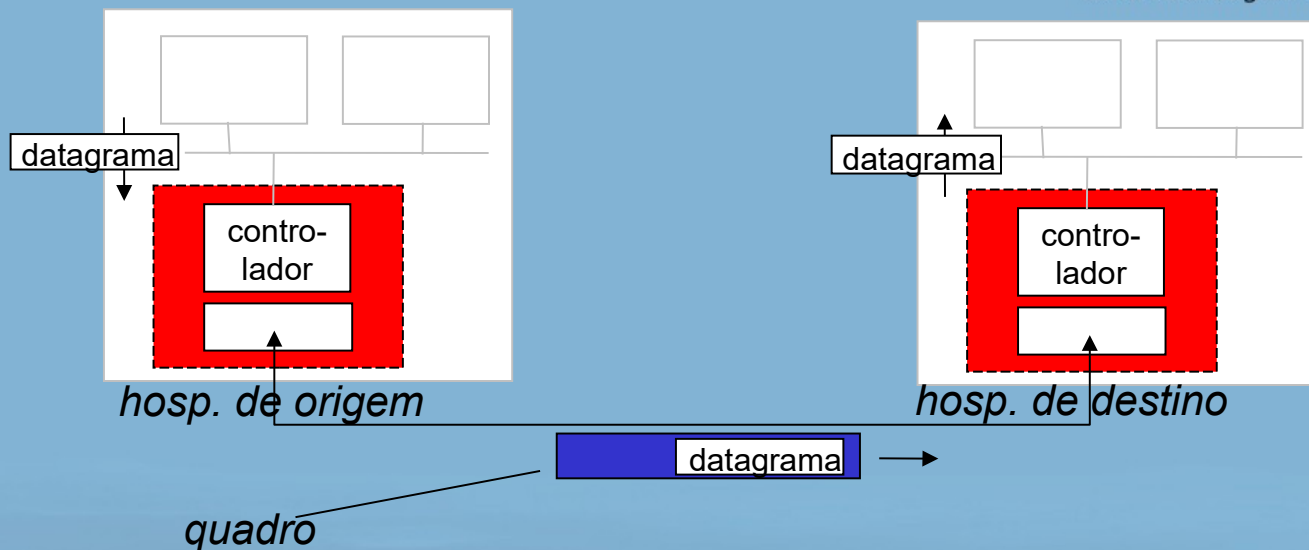
- μ com *half-duplex*, os nós nas duas extremidades do enlace podem transmitir, mas não ao mesmo tempo

# Onde é implementada a camada de enlace?

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



# Comunicação entre adaptadores



## $\rho$ lado emissor:

- $\mu$  encapsula datagrama no quadro
- $\mu$  inclui bits de verificação de erro, rdt, controle de fluxo etc.

## $\rho$ lado receptor

- $\mu$  procura erros, rdt, controle de fluxo etc.
- $\mu$  extrai datagrama, passa para camada superior no lado receptor



## Camada de enlace

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

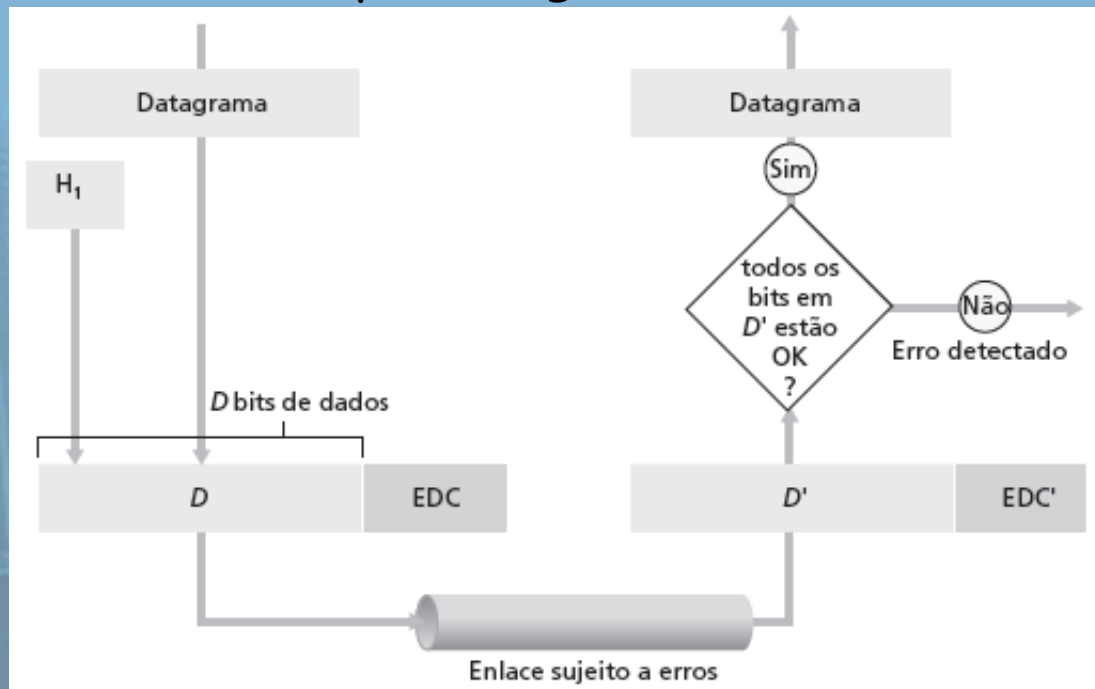
# Deteccção de erros

EDC = Bits de deteccção e correccção de erros (redundância)

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

Deteccção de erro não 100% confiável!

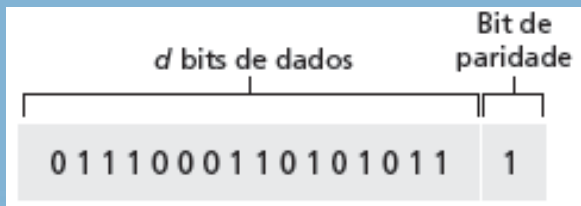
- protocolo pode perder alguns erros, mas raramente
- maior campo EDC gera melhor deteccção e correccção



# Verificação de paridade

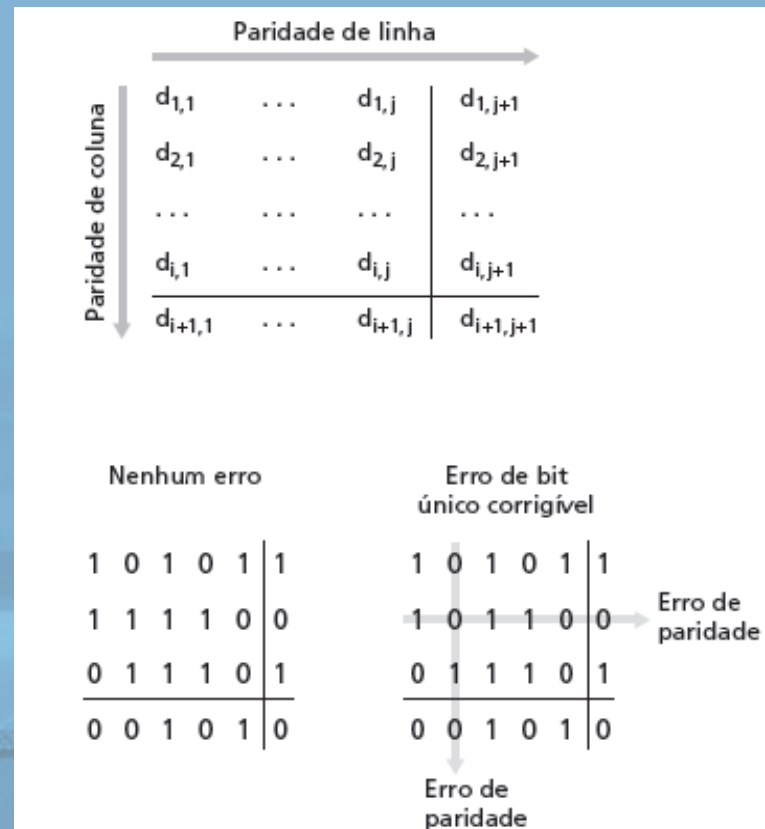
## Paridade de único bit:

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 (nota: usada *somente* na camada de transporte)

## **Emissor:**

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

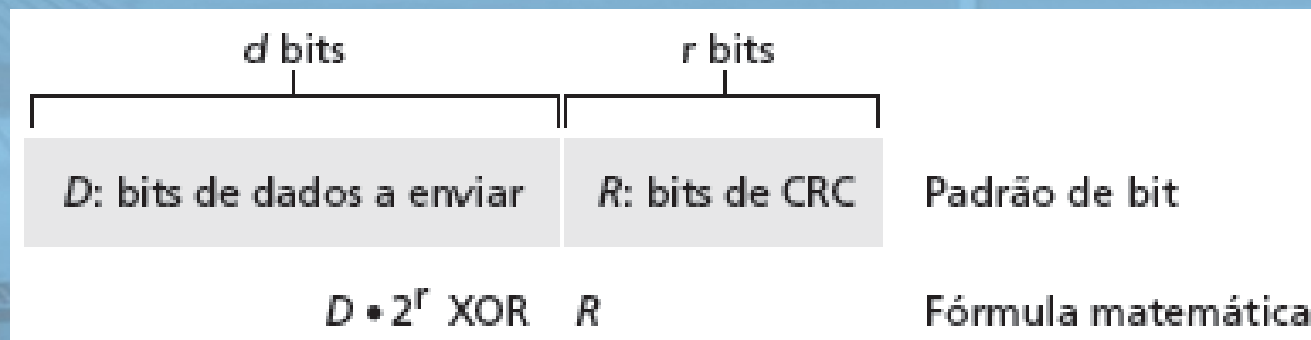
## **Receptor:**

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



# Soma de verificação: verificação de redundância cíclica

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



## Camada de enlace

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

# Enlaces e protocolos de acesso múltiplo

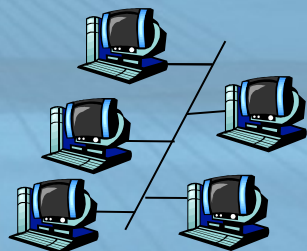
## Dois tipos de "enlaces":

### ρ ponto a ponto

- μ PPP para acesso discado
- μ enlace ponto a ponto entre comutador Ethernet e hospedeiro

### ρ **broadcast** (fio ou meio compartilhado)

- μ Ethernet à moda antiga
- μ HFC anterior
- μ LAN sem fio 802.11



fio compartilhado (p. e.,  
Ethernet cabeado)



RF compartilhada  
(p. e., WiFi 802.11)



RF compartilhada  
(satélite)



humanos em uma festa  
(ar e acústica  
compartilhados)

# Protocolos de acesso múltiplo

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

## protocolo de acesso múltiplo

- ρ algoritmo distribuído que determina como os nós compartilham canal, ou seja, determinam quando o nó pode transmitir
- ρ comunicação sobre compartilhamento de canal deve usar o próprio canal!
  - μ 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:
  - μ nenhum nó especial para coordenar transmissões
  - μ nenhuma sincronização de clocks, intervalos
4. simples

# Protocolos MAC: uma taxonomia

Três classes gerais:

ρ **Particionamento de canal**

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

ρ **Acesso aleatório**

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

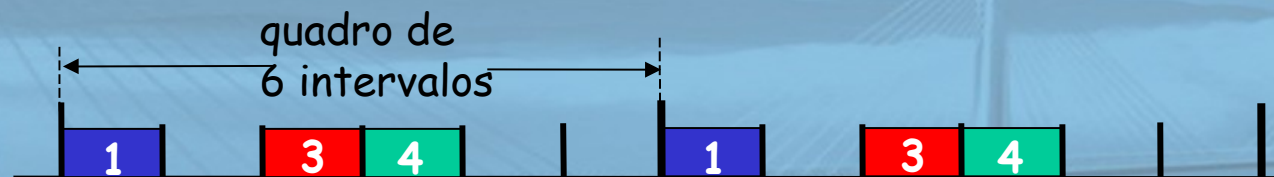
ρ **"Revezando"**

- μ 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

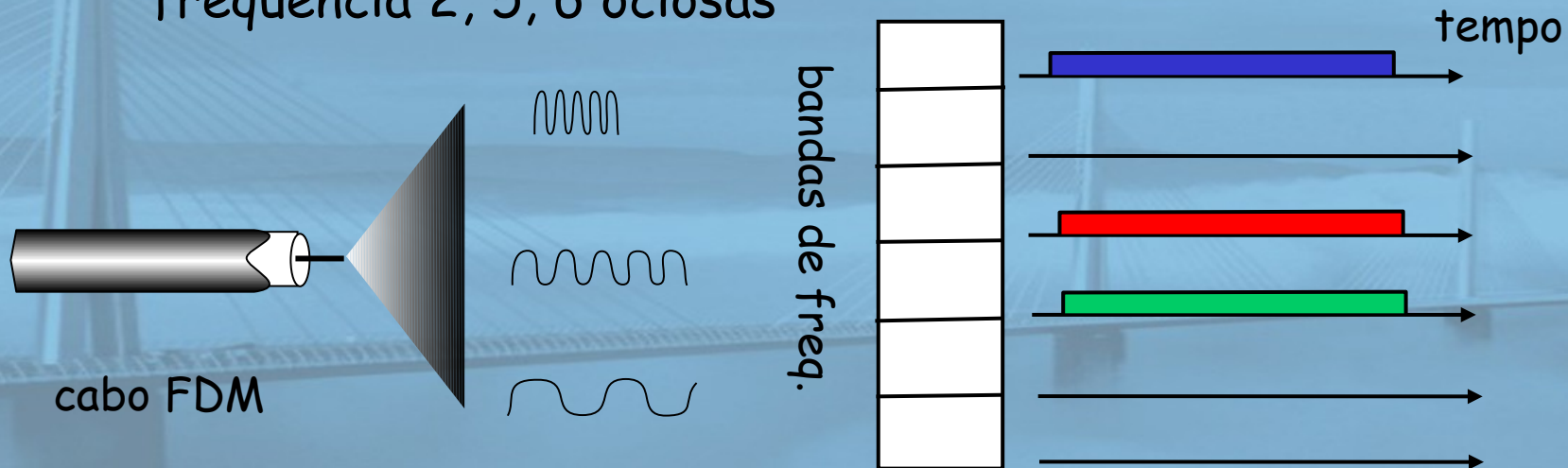
- ρ acesso ao canal em “rodadas”
- ρ cada estação recebe intervalo de tamanho fixo (tamanho = tempo transm. pacote) a cada rodada
- ρ intervalos não usados ficam ociosos
- ρ exemplo: LAN de 6 estações, 1, 3, 4 têm pacote, intervalos 2, 5, 6 ociosos



# Protocolos MAC de particionamento de canal: FDMA

## FDMA: Frequency Division Multiple Access

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





# Protocolos de acesso aleatório

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

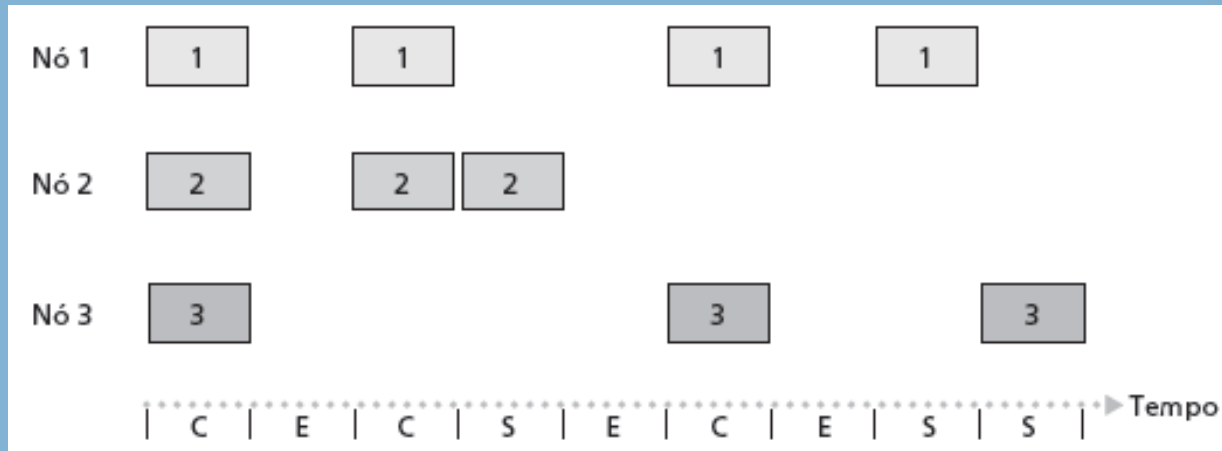
# Slotted ALOHA

## Suposições:

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

## Operação:

- ρ quando nó obtém quadro novo, transmite no próximo intervalo
  - μ *se não há colisão*: nó pode enviar novo quadro no próximo intervalo
  - μ *se há colisão*: nó retransmite quadro em cada intervalo subsequente com prob. até que haja sucesso



## Prós

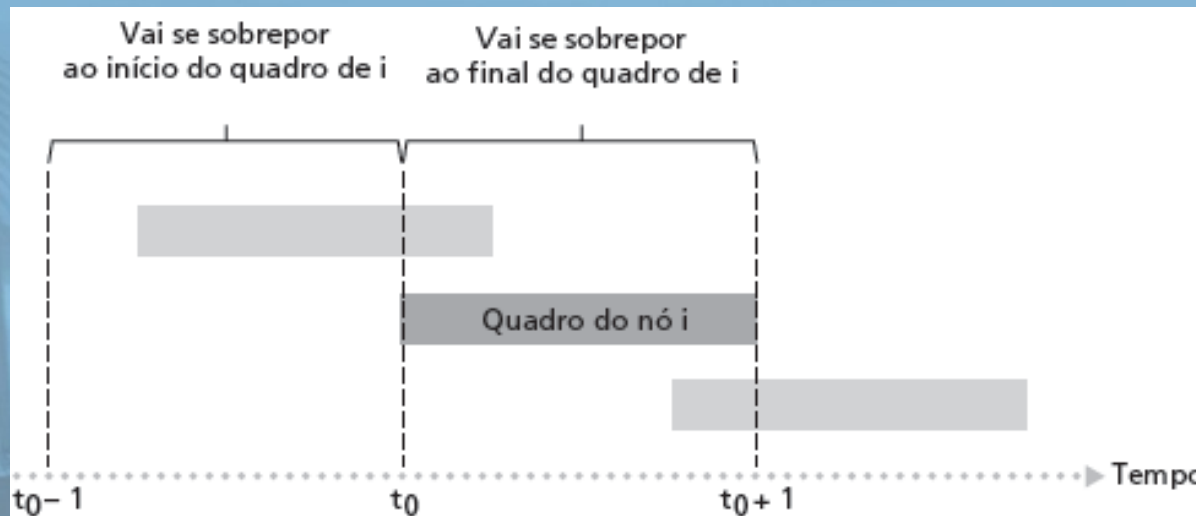
- ρ único nó ativo pode transmitir continuamente na velocidade plena do canal
- ρ altamente descentralizado: somente intervalos nos nós precisam estar em sincronismo
- ρ simples

## Contras

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

## ALOHA puro (não slotted)

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





# CSMA (Carrier Sense Multiple Access)

**CSMA**: ouça antes de falar:

se perceber canal ocioso: transmite quadro inteiro

ρ se perceber canal ocupado, adia transmissão

ρ 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

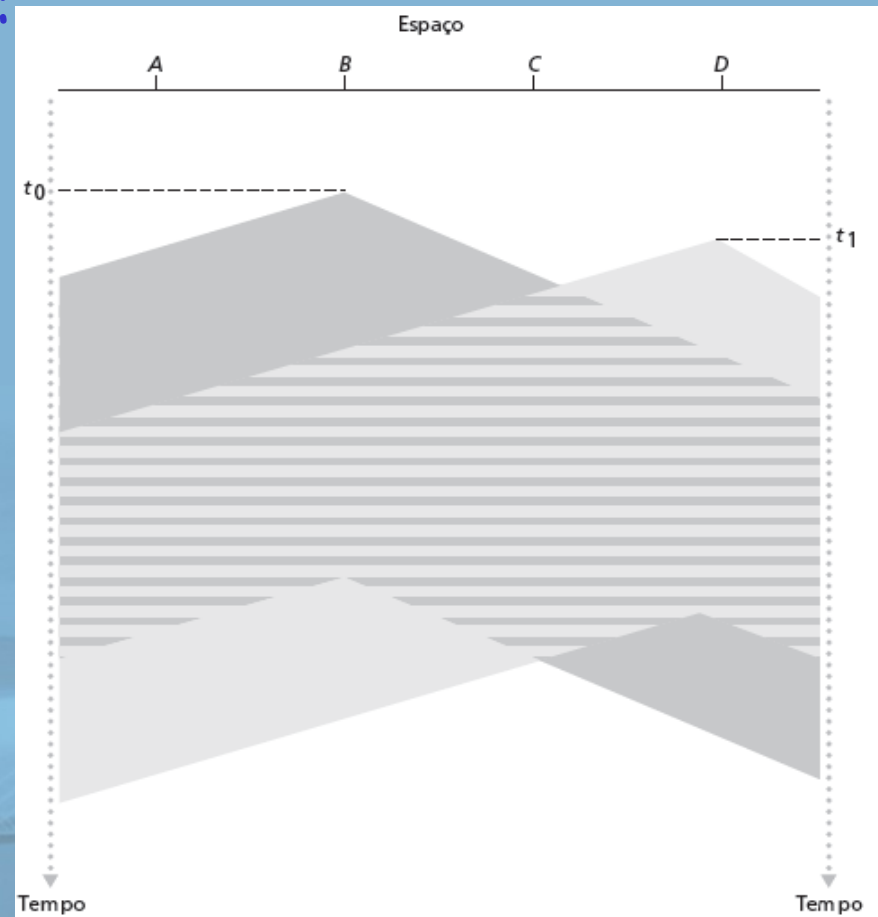
colisão:

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)

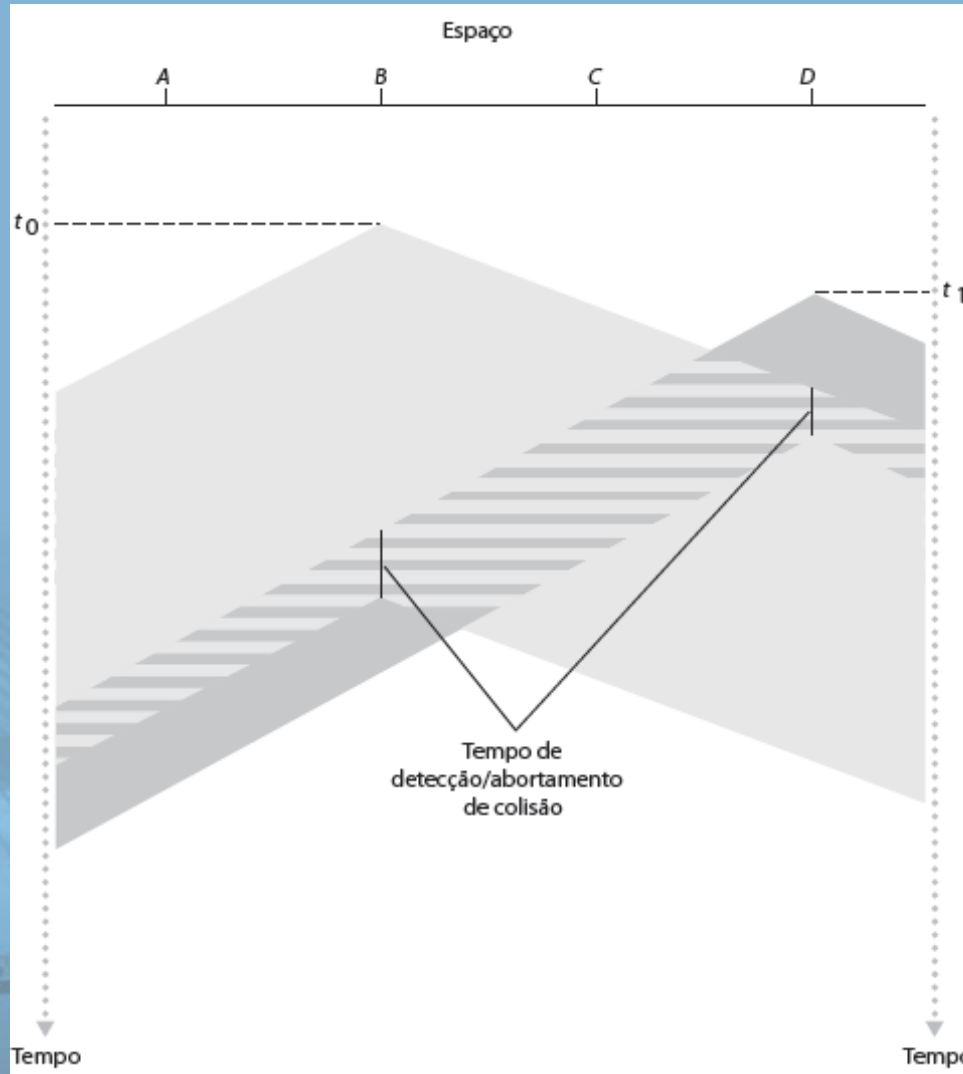
**CSMA/CD:** detecção de portadora, adiada como no CSMA

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

# Deteção de colisão CSMA/CD

REDES DE  
COMPUTADORES  
E A INTERNET 5ª edição

*Uma Abordagem Top-Down*





# "Revezando" protocolos MAC

## protocolos MAC de particionamento de canal:

- μ compartilham canal de modo *eficaz e justo* com alta carga
- μ 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

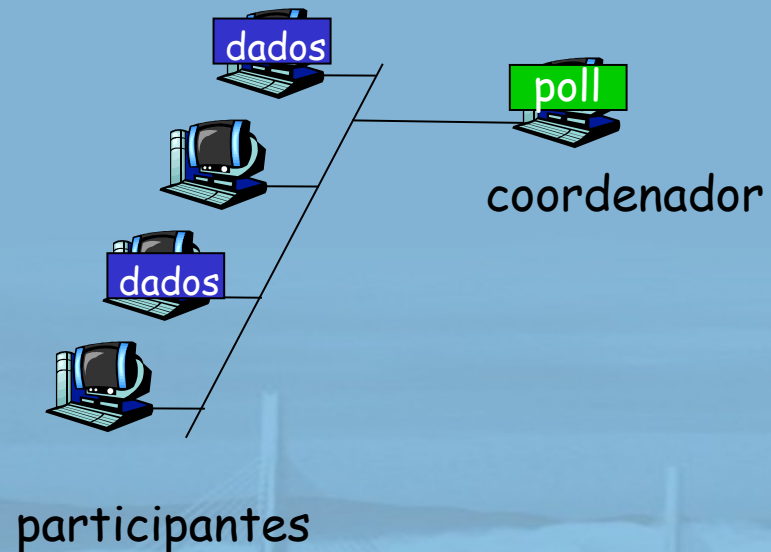
- μ eficaz com baixa carga: único nó pode utilizar o canal totalmente
- μ alta carga: sobrecarga de colisão

## "revezando" protocolos

procure o melhor dos dois mundos!

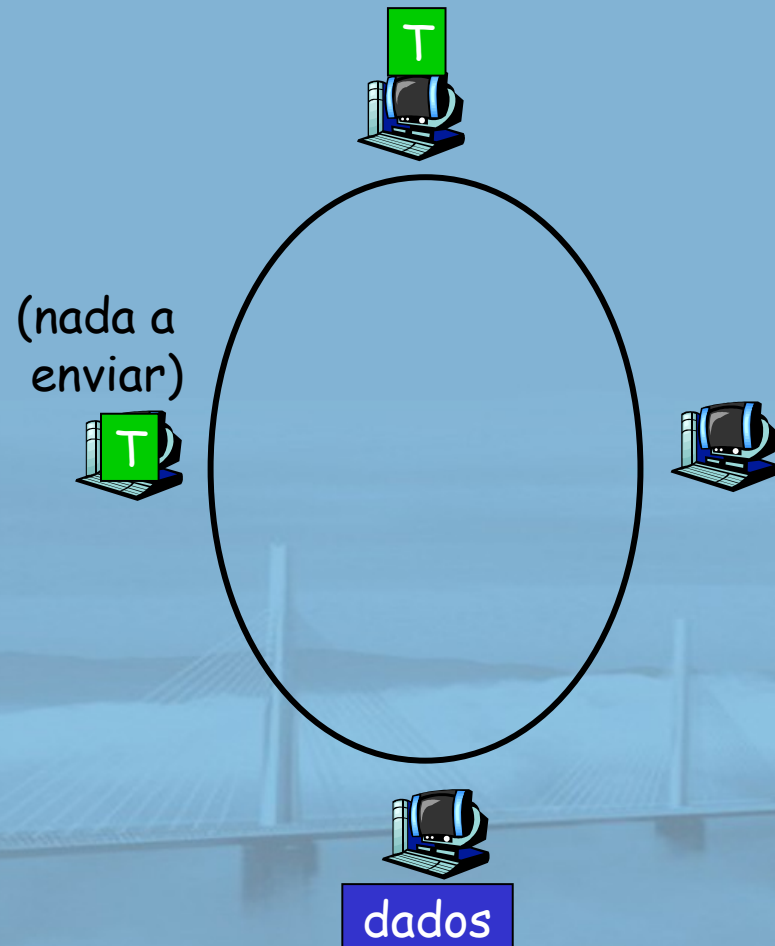
## Polling (seleção):

- ρ nó coordenador  
"convida" nós  
participantes a  
alterarem a  
transmissão
- ρ normalmente usado  
com dispositivos  
participantes "burros"
- ρ preocupações:
  - μ sobrecarga da seleção
  - μ latência
  - μ único ponto de falha  
(coordenador)



## Passagem de permissão:

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



## Resumo de protocolos MAC

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



## Camada de enlace

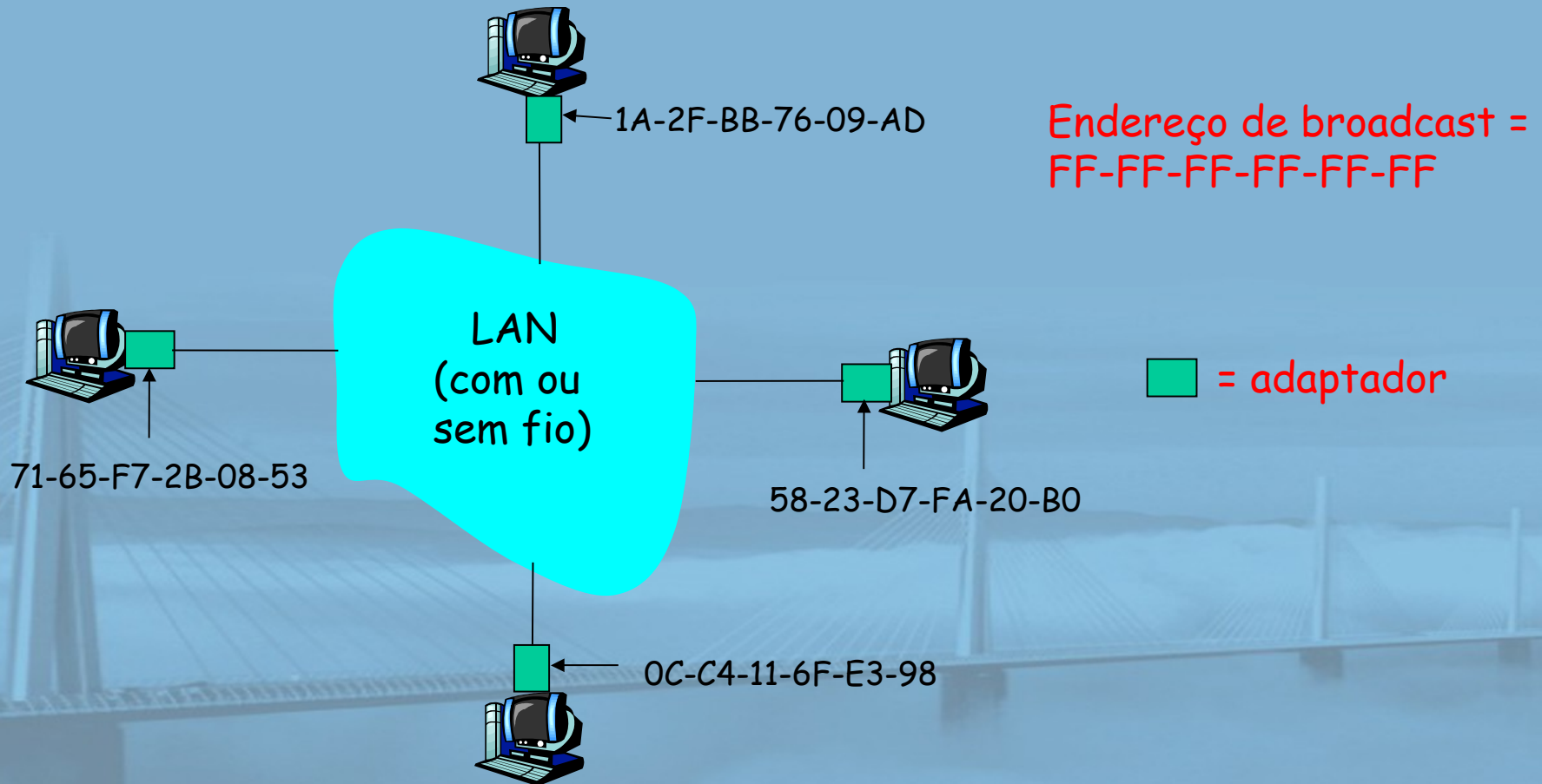
- ρ 5.1 Introdução e serviços
- ρ 5.2 Detecção e correção de erros
- ρ 5.3 Protocolos de acesso múltiplo
- ρ 5.4 Endereçamento na camada de enlace
- ρ 5.5 Ethernet
- ρ 5.6 Comutadores de camada de enlace
- ρ 5.7 PPP
- ρ 5.8 Virtualização de enlace: MPLS
- ρ 5.9 Um dia na vida de uma solicitação de página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

# Endereçamento MAC e ARP

- ρ Endereço IP de 32 bits:
  - μ endereço da camada de rede
  - μ usado para obter datagrama até sub-rede IP de destino
- ρ Endereço MAC (ou LAN ou físico ou Ethernet) :
  - μ função: *levar quadro de uma interface para outra interface conectada fisicamente (na mesma rede)*
  - μ 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)

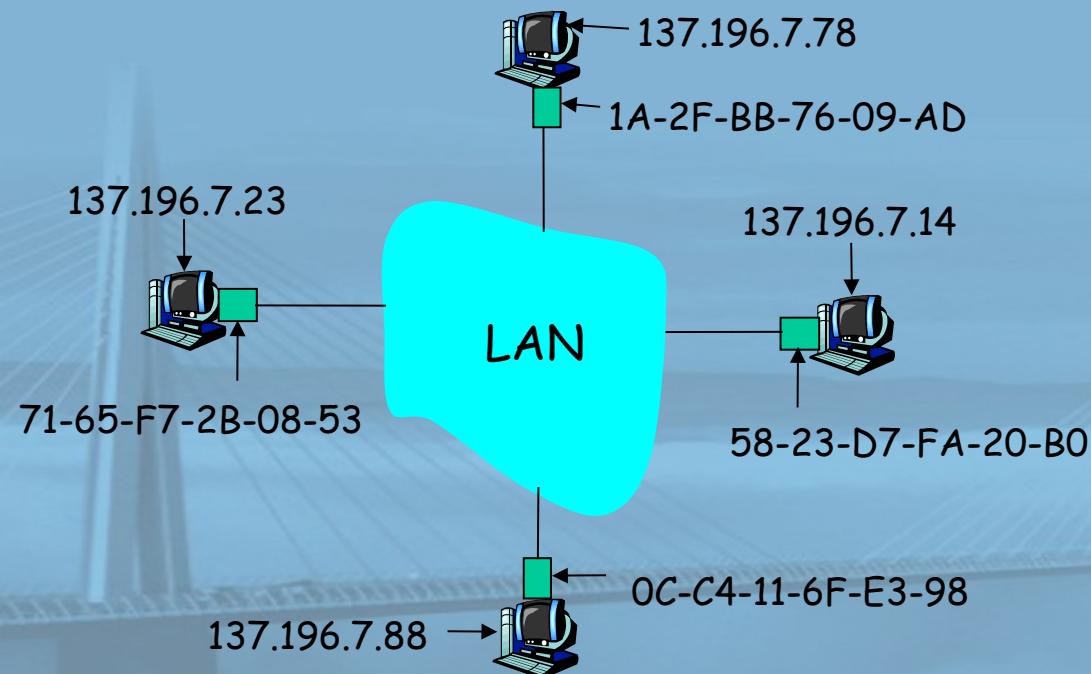
- ρ alocação de endereço MAC administrada pelo IEEE
- ρ fabricante compra parte do espaço de endereços MAC (para garantir exclusividade)
- ρ analogia:
  - (a) Endereço MAC: como o CPF
  - (b) Endereço IP: como o endereço postal
- ρ endereço MAC plano → portabilidade
  - μ pode mover placa de LAN de uma LAN para outra
- ρ endereço IP hierárquico NÃO portátil
  - μ 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?

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



# Protocolo ARP: mesma LAN (rede)

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

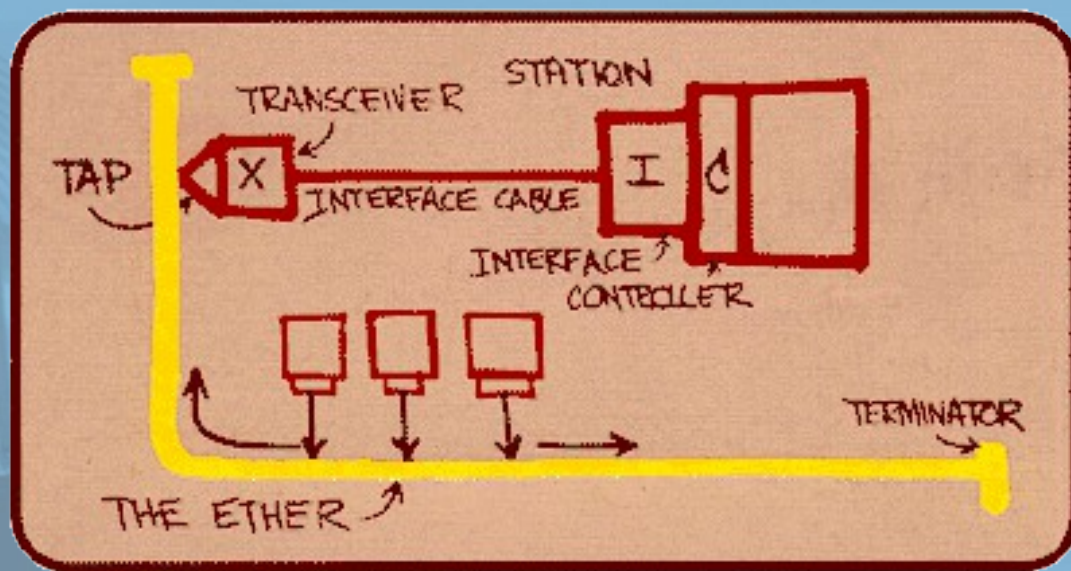
## Camada de enlace

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

# Ethernet

## Tecnologia de LAN com fio "dominante":

- ρ barata: US\$ 20 para NIC
- ρ primeira tecnologia de LAN utilizada em larga escala
- ρ mais simples e mais barata que as LANs de permissão e ATM
- ρ acompanhou corrida da velocidade: 10 Mbps - 10 Gbps

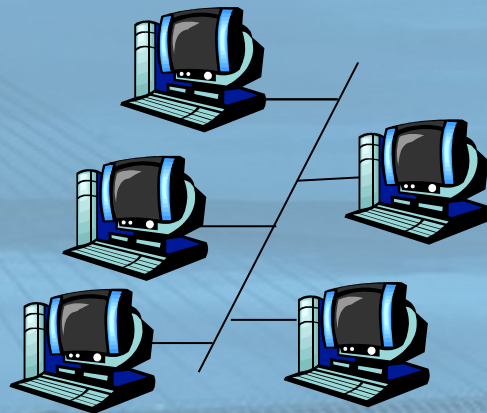


Projeto original da  
Ethernet de Metcalfe

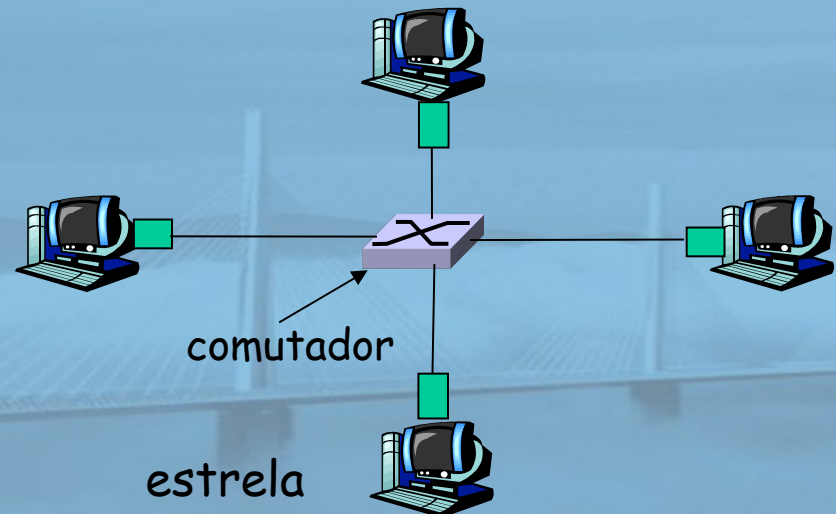


# Topologia de estrela

- ρ topologia de barramento popular até meados dos anos 90
  - μ todos os nós no mesmo domínio de colisão (podem colidir uns com os outros)
- ρ hoje: topologia de estrela prevalece
  - μ **comutador** ativo no centro
  - μ 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:

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

## ρ Endereços: 6 bytes

- μ 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
- μ caso contrário, adaptador descarta quadro

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

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



# Ethernet: não confiável, sem conexão

- ρ **sem conexão:** sem apresentação entre NICs de origem e destino
- ρ **não confiável:** NIC de destino não envia confirmações ou não confirmações à NIC de origem
  - μ fluxo de datagramas passados à camada de rede pode ter lacunas (datagramas faltando)
  - μ lacunas serão preenchidas se aplicação estiver usando TCP
  - μ caso contrário, aplicação verá lacunas
- ρ 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, 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

Veja/interaja com applet Java no site Web da AWL: altamente recomendado!

## Backoff exponencial:

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



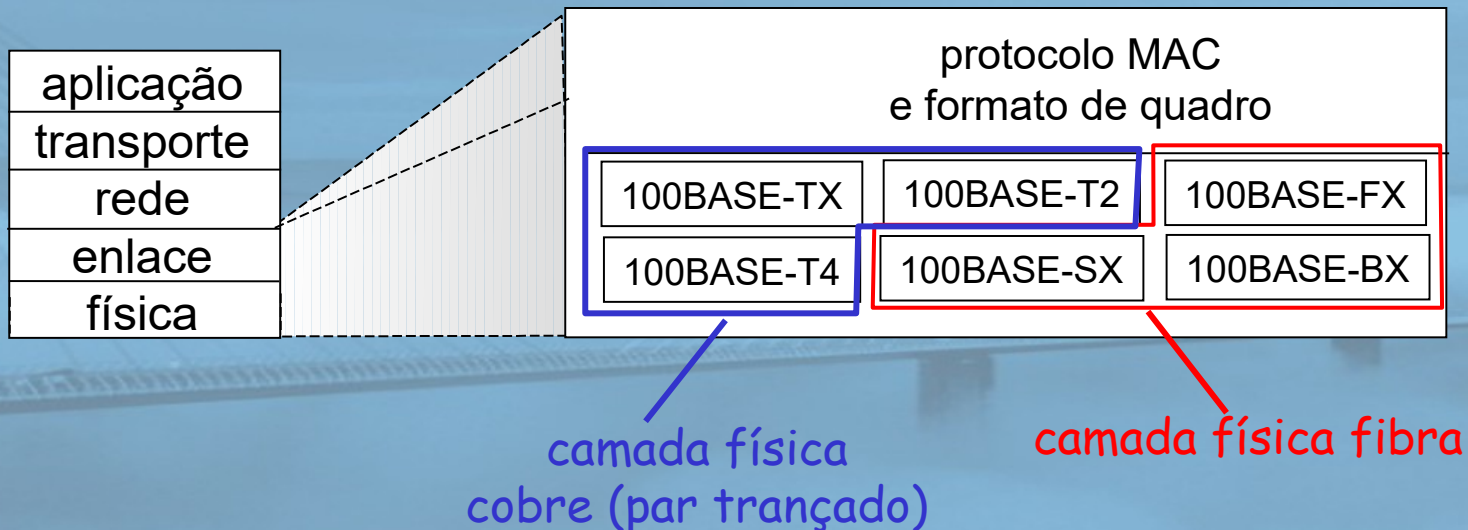
# Padrões Ethernet 802.3: camadas de enlace e física

ρ **muitos** padrões Ethernet diferentes

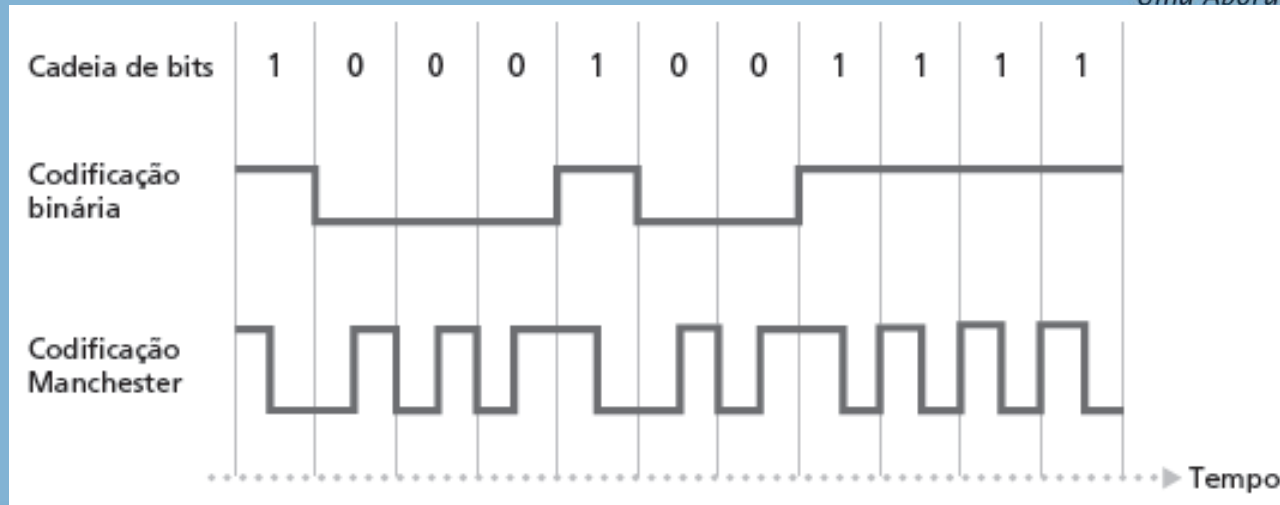
μ protocolo MAC e formato de quadro comuns

μ diferentes velocidades: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10G bps

μ diferentes meios da camada física: fibra, cabo



# Codificação Manchester



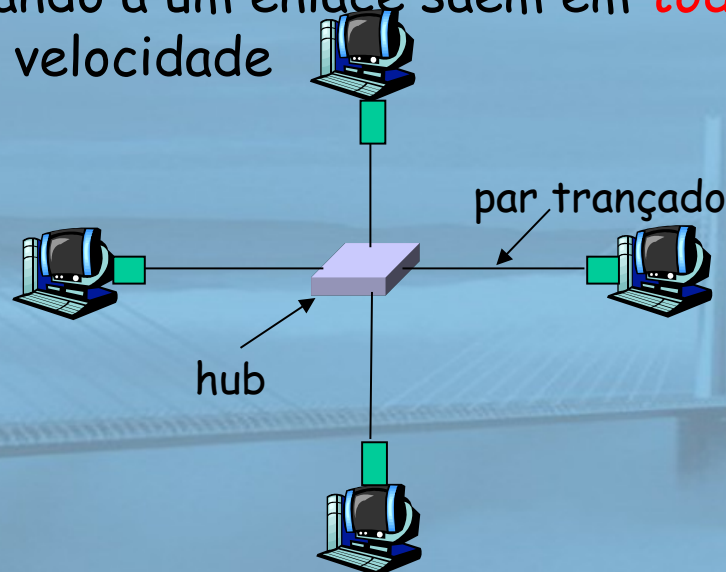
- ρ usado no 10BaseT
- ρ cada bit tem uma transição
- ρ permite que clocks nos nós emissor e receptor sejam sincronizados entre si
  - μ desnecessário para um clock centralizado, global entre os nós!
- ρ Ei, isso é coisa da camada física!

## Camada de enlace

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

... repetidores da camada física ("burros") :

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



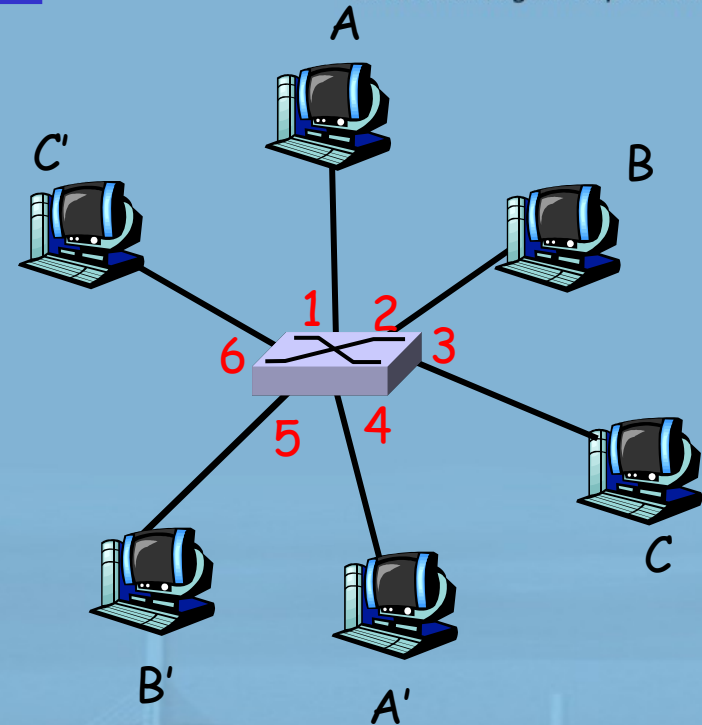
## Comutador (switch)

- ρ dispositivo da camada de enlace: mais inteligente que os hubs, têm papel ativo
  - μ armazenam e repassam quadros Ethernet
  - μ 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
- ρ **transparente**
  - μ hosps. não sabem da presença de comutadores
- ρ **plug-and-play, autodidata**
  - μ comutadores não precisam ser configurados



# Comutador: permite múltiplas transmissões simultâneas

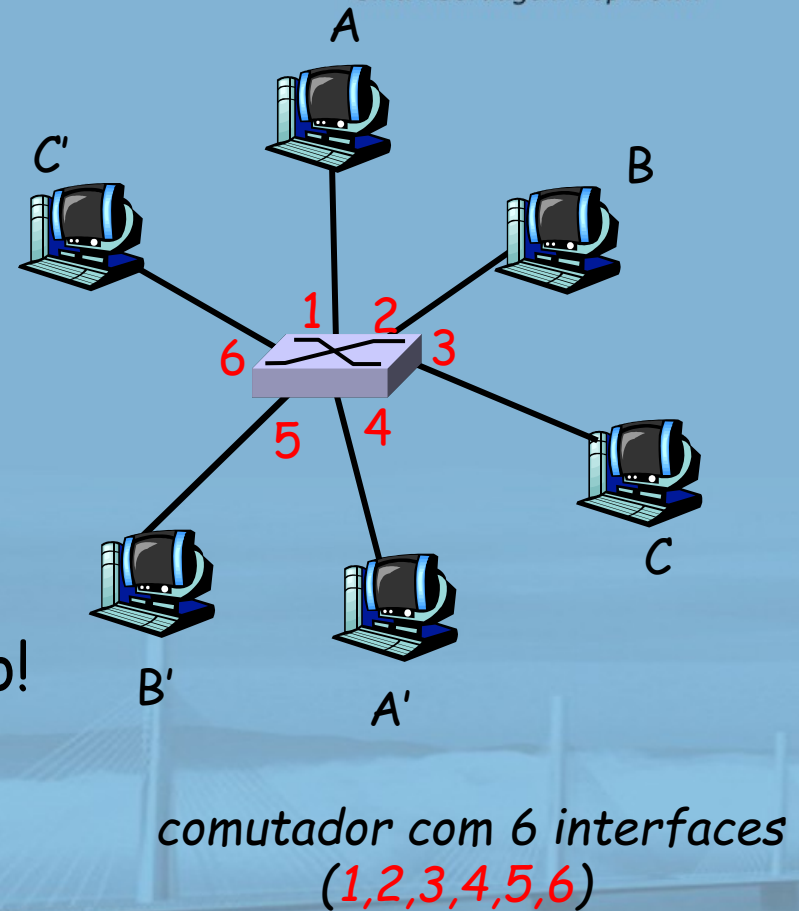
- ρ hosps. têm conexão dedicada, direta com comutador
- ρ comutadores mantêm pacotes
- ρ Protocolo Ethernet usado em cada enlace de chegada, mas sem colisões; full duplex
  - μ cada enlace é seu próprio domínio de colisão
- ρ **comutação:** A-para-A' e B-para-B' simultaneamente, sem colisões
  - μ não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



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

## Tabela de comutação

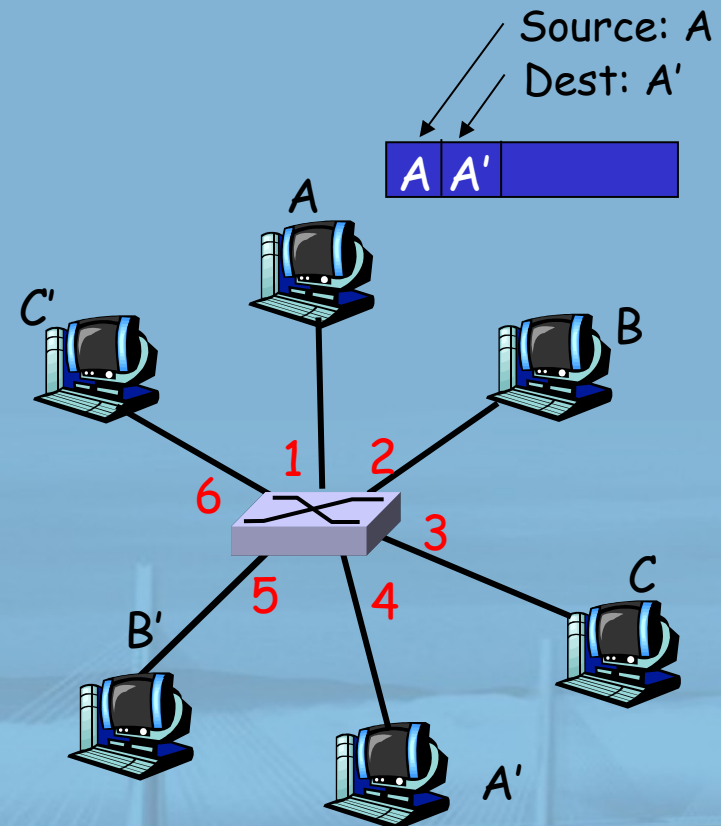
- ρ P: Como o comutador sabe que A' se encontra na interface 4, B' se encontra na interface 5?
- ρ R: Cada comutador tem uma **tabela de comutação**, cada entrada:
  - μ (endereço MAC do nó, interface para alcançar nó, horário)
- ρ parece com tab. de roteamento!
- ρ P: Como as entradas são criadas, mantidas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



## Comutador: autodidata

- ρ comutador *descobre* quais nós podem ser alcançados por quais interfaces
- μ quando quadro recebido, comutador "aprende" local do emissor: segmento de LAN de chegada
- μ 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

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

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

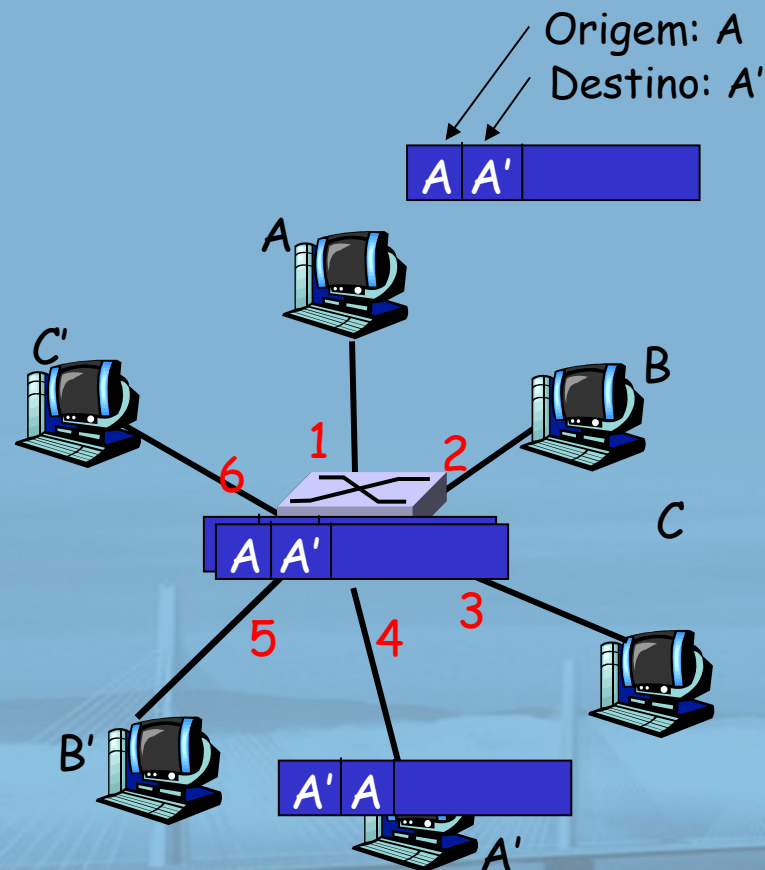
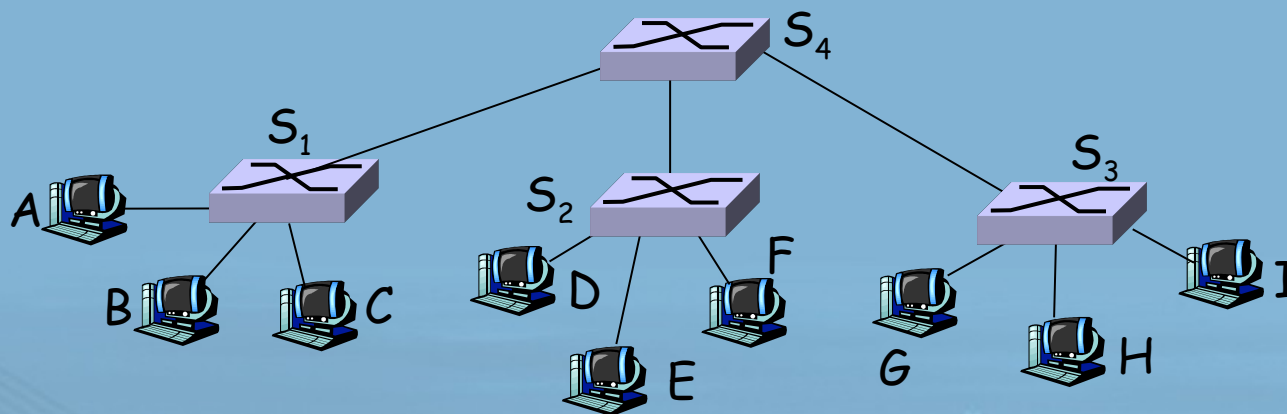


Tabela comutação  
(inicialmente  
vazia)



# Interconectando comutadores

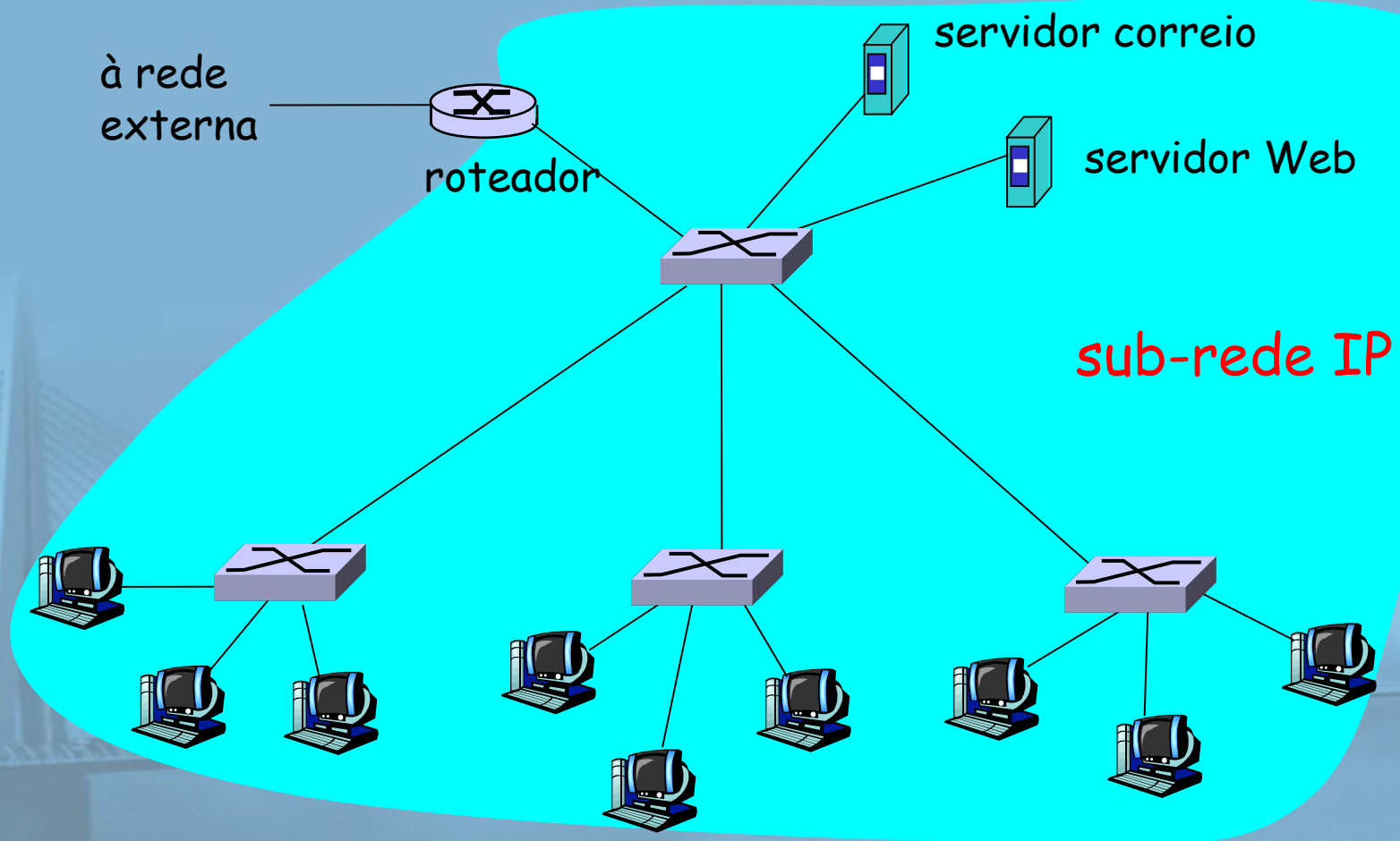
ρ comutadores podem ser conectados



ρ P: Enviando de A p/G - como  $S_1$  sabe repassar quadro destinado a F por  $S_4$  e  $S_3$ ?

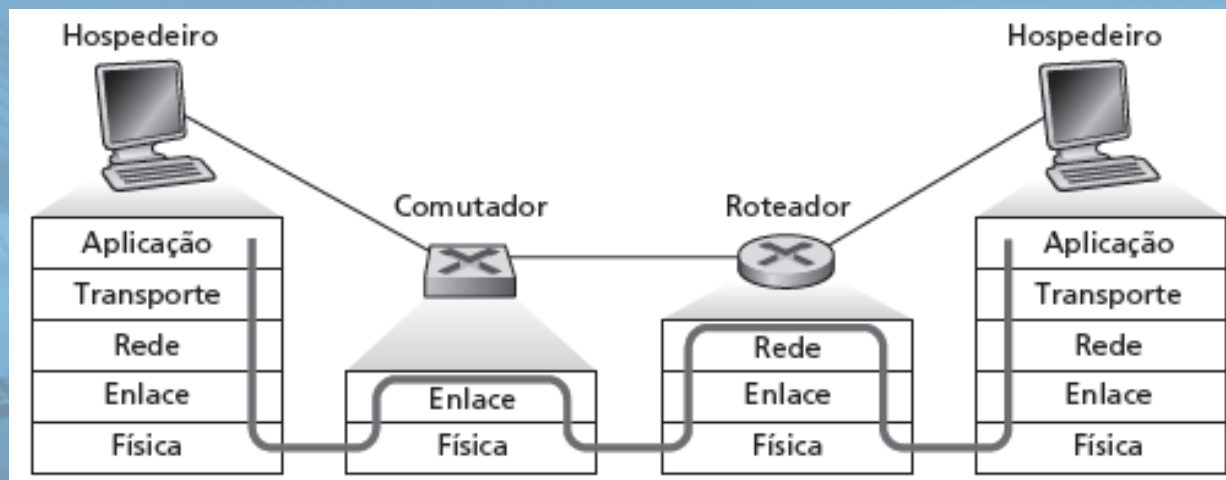
ρ R: Autoaprendizagem! (funciona da mesma forma que no caso do único comutador!)

# Rede institucional



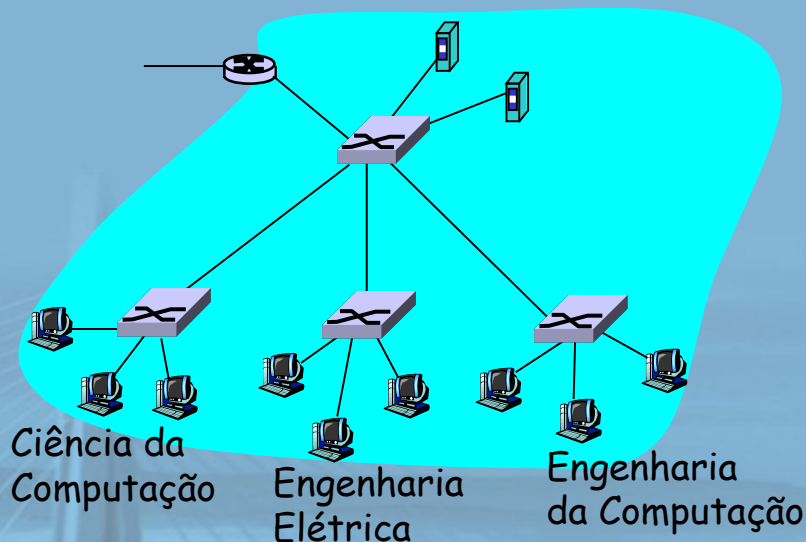
# Comutadores versus roteadores

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



## VLANs: motivação

*O que há de errado nesta figura?*



O que acontece se:

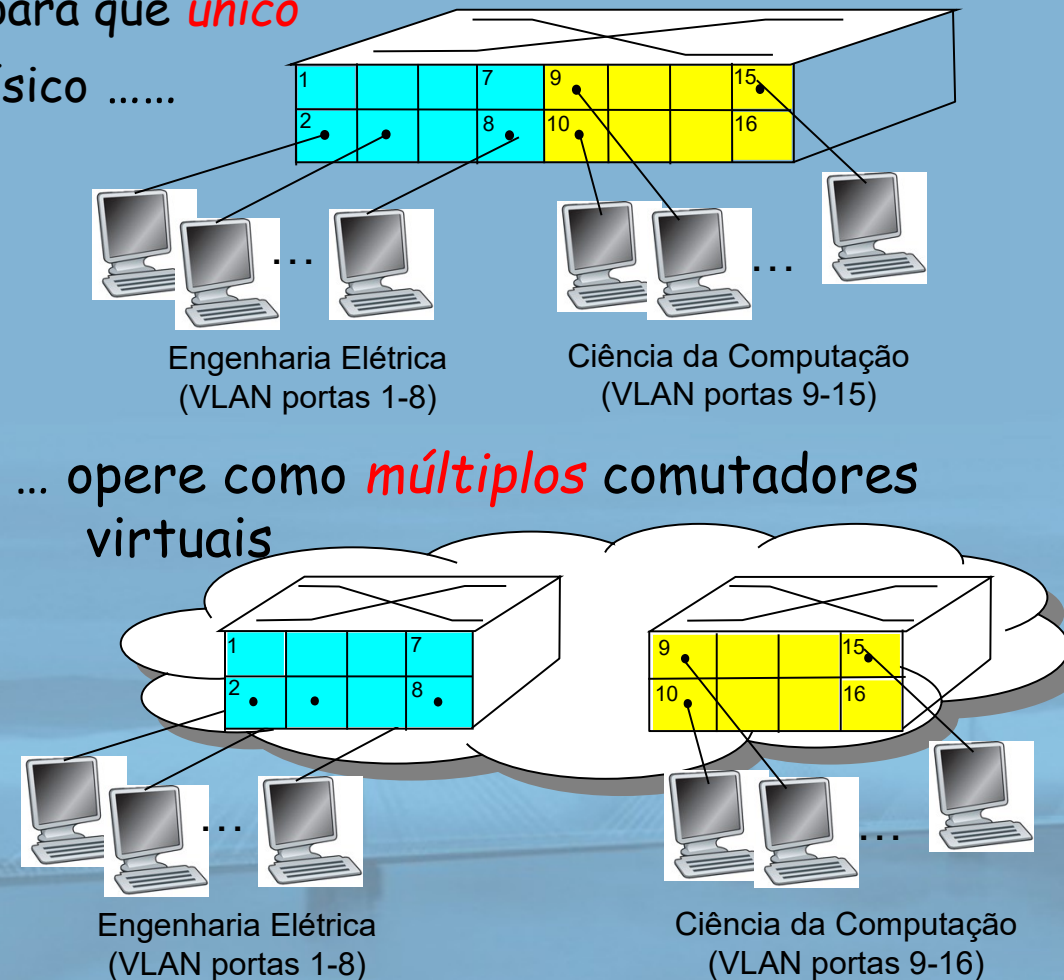
- ρ usuário da CC muda para EE, mas quer se conectar ao comutador CC?
- ρ único domínio de broadcast:
  - μ todo tráfego de broadcast da camada 2 (ARP, DHCP) cruza a LAN inteira (questões de eficiência, segurança/privacidade)
- ρ cada comutador de nível mais baixo tem apenas algumas portas em uso

# VLANs

**VLAN baseada em porta:**  
portas de comutador agrupadas  
(por software de gerenciamento de  
comutador) para que **único**  
comutador físico .....

## Virtual Local Area Network

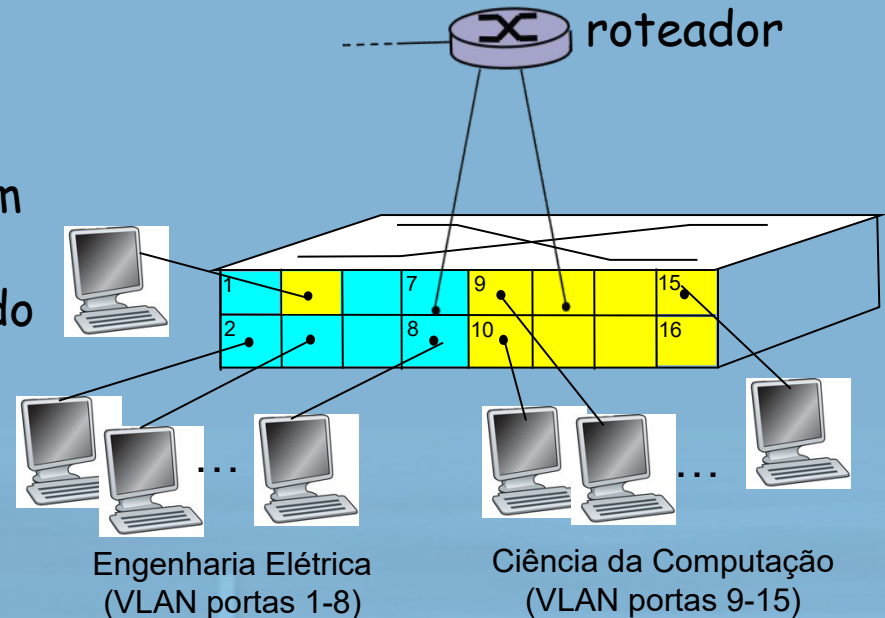
Comutador(es)  
admitindo capacidades  
de VLAN podem ser  
configurados para  
definir múltiplas LANs  
**virtuais** por única  
infraestrutura de LAN  
física.



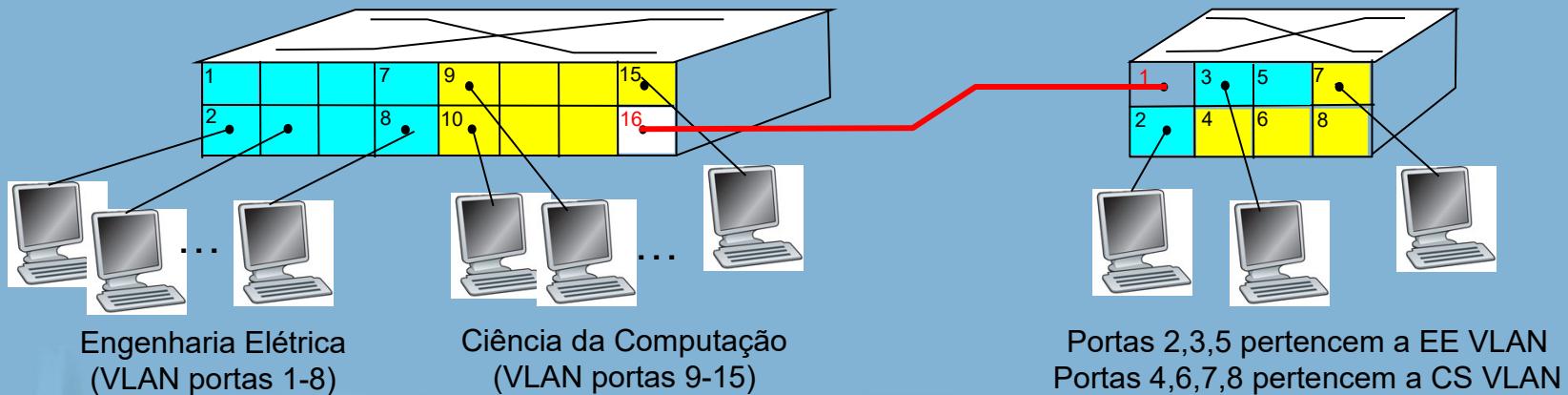


## VLAN baseada em porta

- ρ **isolamento de tráfego:** quadros de/para portas 1-8 só podem alcançar portas 1-8
  - μ também podem definir VLAN com base em endereços MAC das extremidades, em vez de porta do comutador
- ρ **inclusão dinâmica:** portas podem ser atribuídas dinamicamente entre VLANs
- ρ **repassse entre VLANs:** feito por roteamento (assim como em comutadores separados)
  - μ na prática, fornecedores vendem uma combinação de comutador e roteador



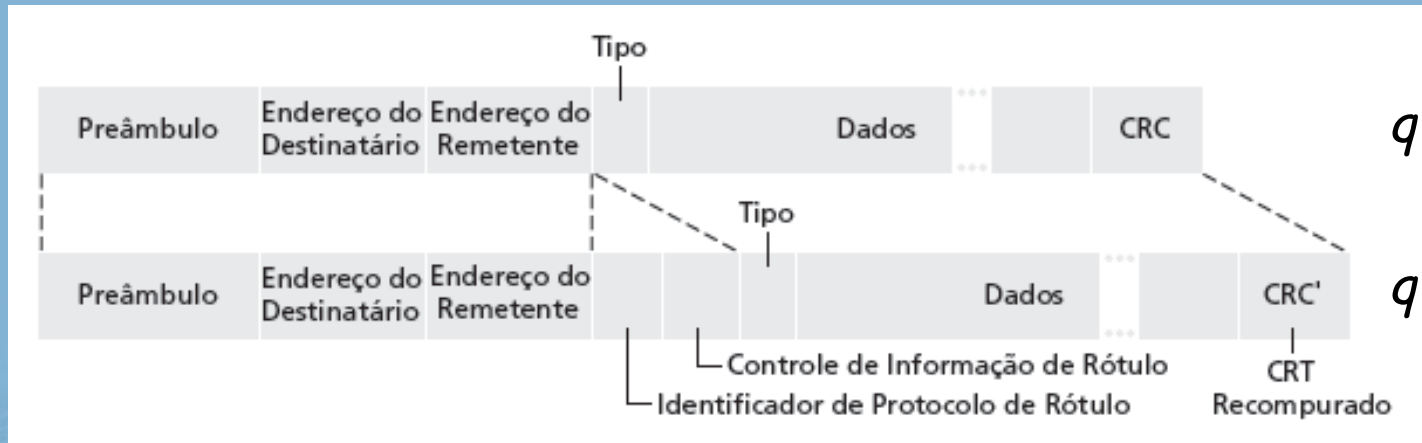
# VLANs spanning multiple switches



ρ **porta de tronco:** carrega quadros entre VLANs definidas sobre vários comutadores físicos

- μ quadros repassados dentro da VLAN entre comutadores não podem ser quadros 802.1 comuns (devem ter informação de VLAN ID)
- μ protocolo 802.1q inclui campos de cabeçalho adicionais para quadros repassados entre portas de tronco

# Formato de quadro 802.1Q VLAN



*quadro 802.1*

*quadro 802.1Q*

## Camada de enlace

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



# Síntese: um dia na vida de uma solicitação Web

ρ viagem pela pilha de protocolos completa!

μ aplicação, transporte, rede, enlace

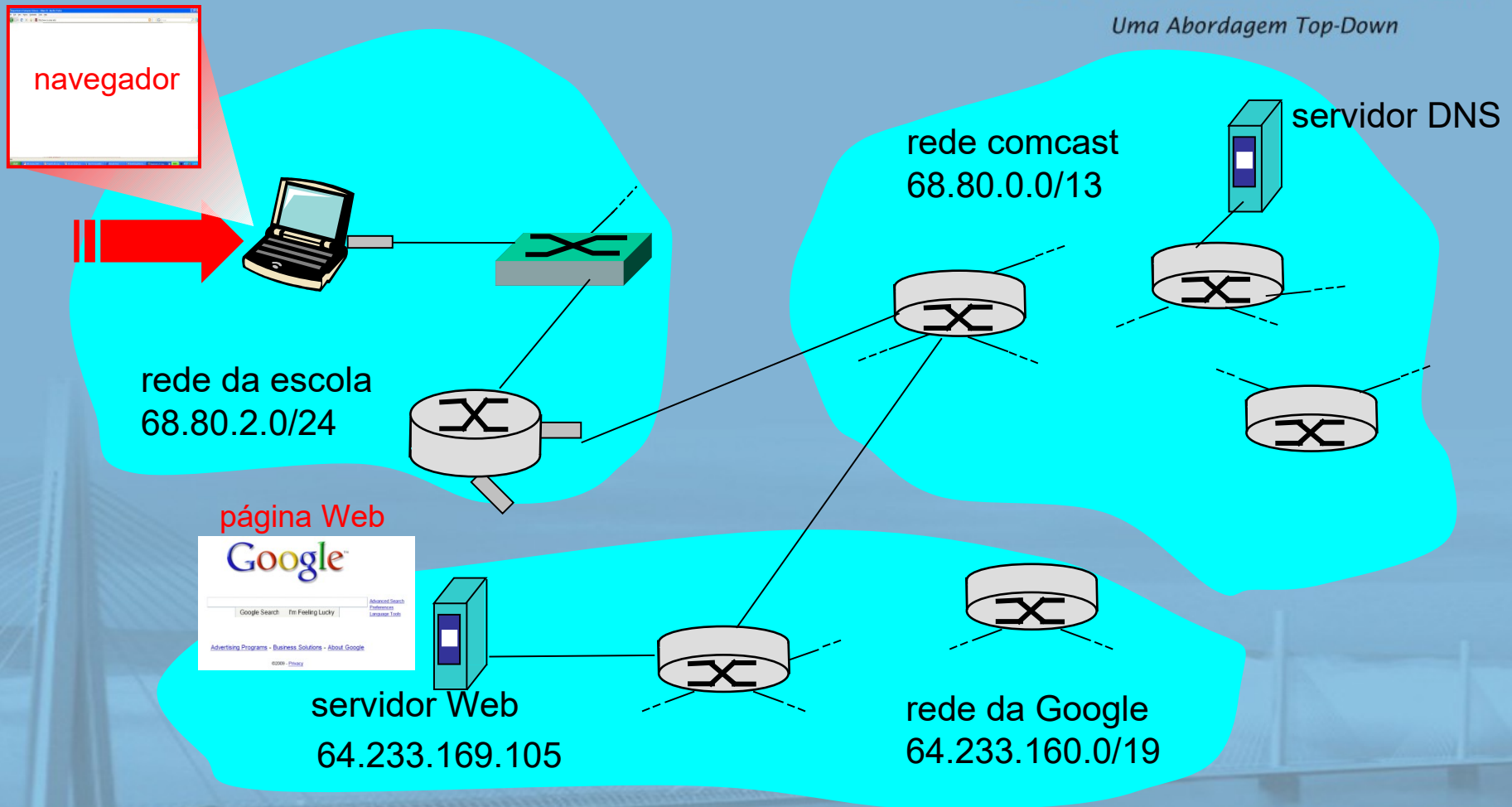
ρ juntando tudo: síntese!

μ **objetivo:** identificar, analisar, entender os protocolos (em todas as camadas) envolvidos no cenário aparentemente simples: solicitar página WWW

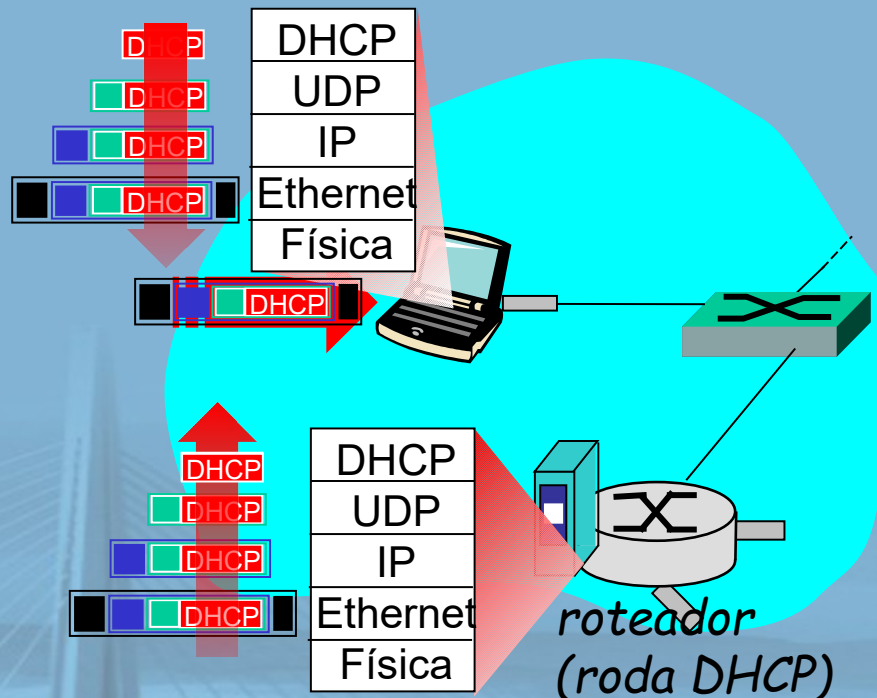
μ **cenário:** aluno conecta laptop à rede do campus, solicita/recebe [www.google.com](http://www.google.com)



## Um dia na vida: cenário



# Um dia na vida... conectando à Internet

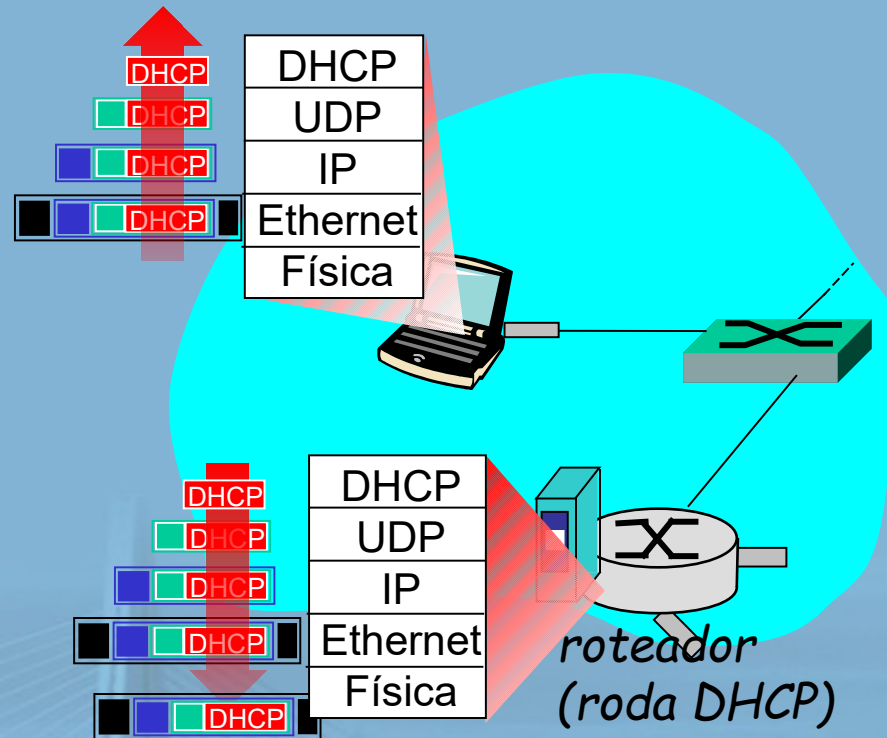


- ρ o laptop conectando precisa obter seu próprio endereço IP, end. do roteador do 1º salto e do servidor DNS: use **DHCP**
- ρ Solicitação DHCP **encapsulada** no **UDP**, encapsulada no **IP**, encapsulada na Ethernet **802.1**
- ρ Quadro Ethernet enviado por **broadcast** (dest.: FFFFFFFFFFFFFFFF)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

# REDES DE COMPUTADORES E A INTERNET

5ª edição

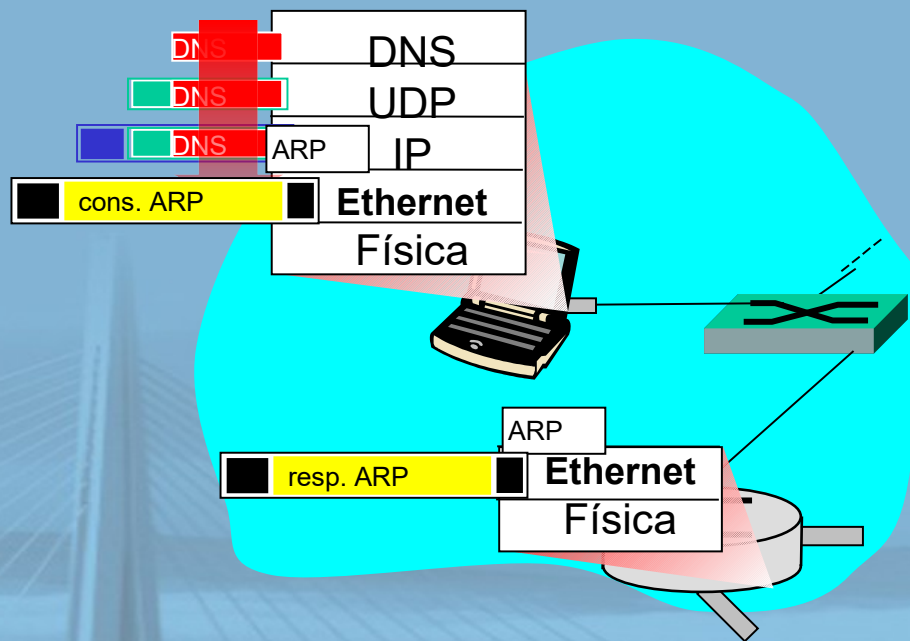
Uma Abordagem Top-Down



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

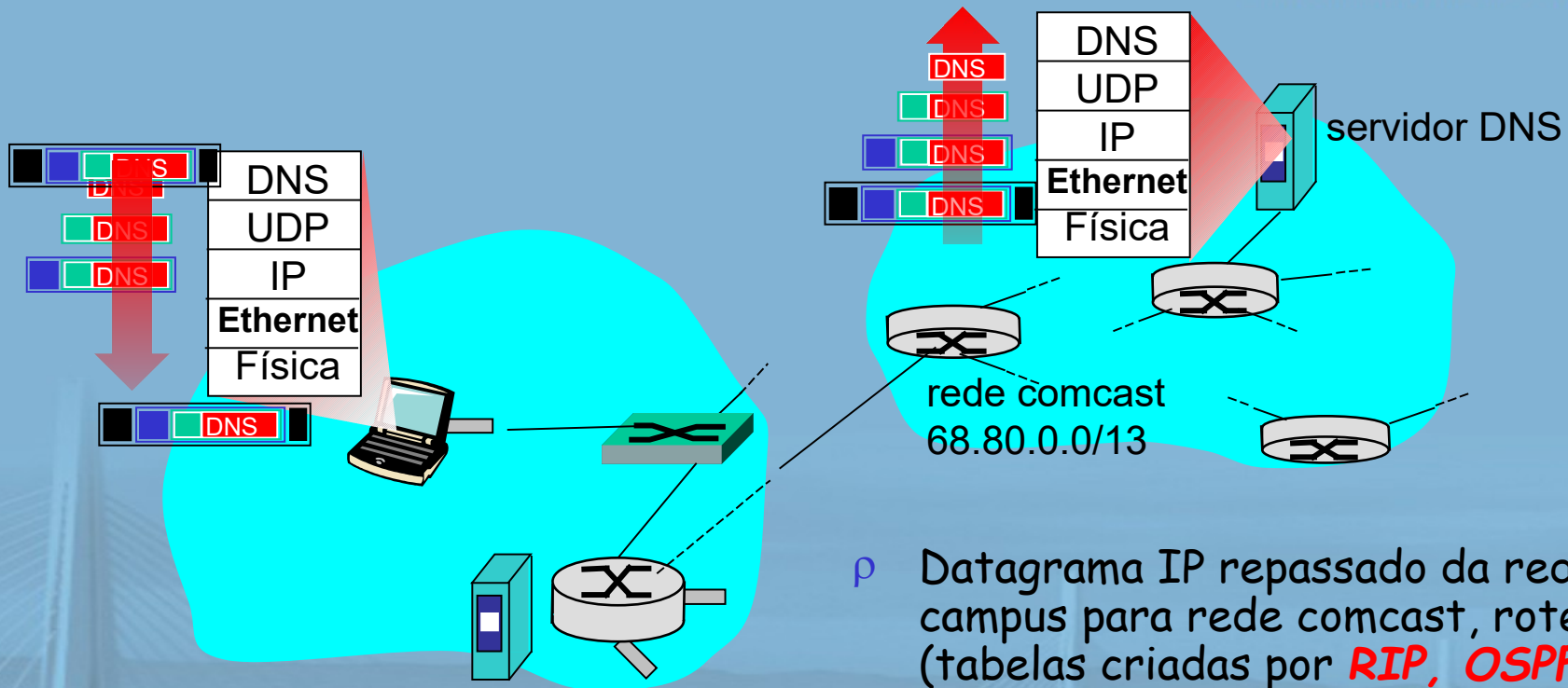
# Um dia na vida... ARP (antes do DNS, antes do HTTP)



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



# Um dia na vida... usando DNS



- ρ Datagrama IP contendo consulta DNS repassada via comutador da LAN do cliente ao roteador do 1º salto

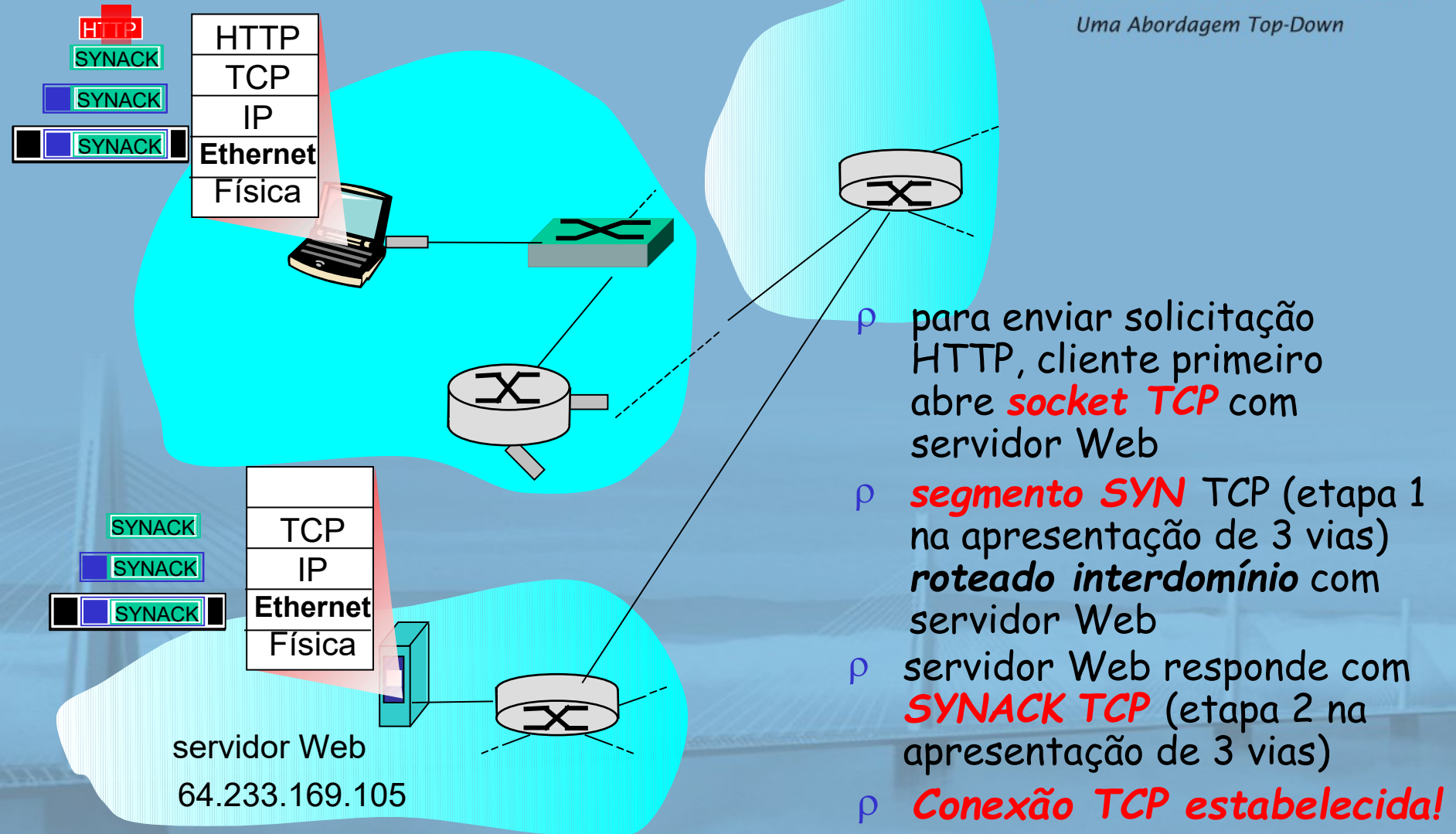
- ρ Datagrama IP repassado da rede do campus para rede comcast, roteado (tabelas criadas por **RIP**, **OSPF**, **IS-IS** e/ou protocolos de roteamento **BGP**) ao servidor DNS
- ρ demultiplexado ao servidor DNS
- ρ Servidor DNS responde ao cliente com endereço IP de [www.google.com](http://www.google.com)



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

REDES DE  
COMPUTADORES  
E A INTERNET 5ª edição

Uma Abordagem Top-Down



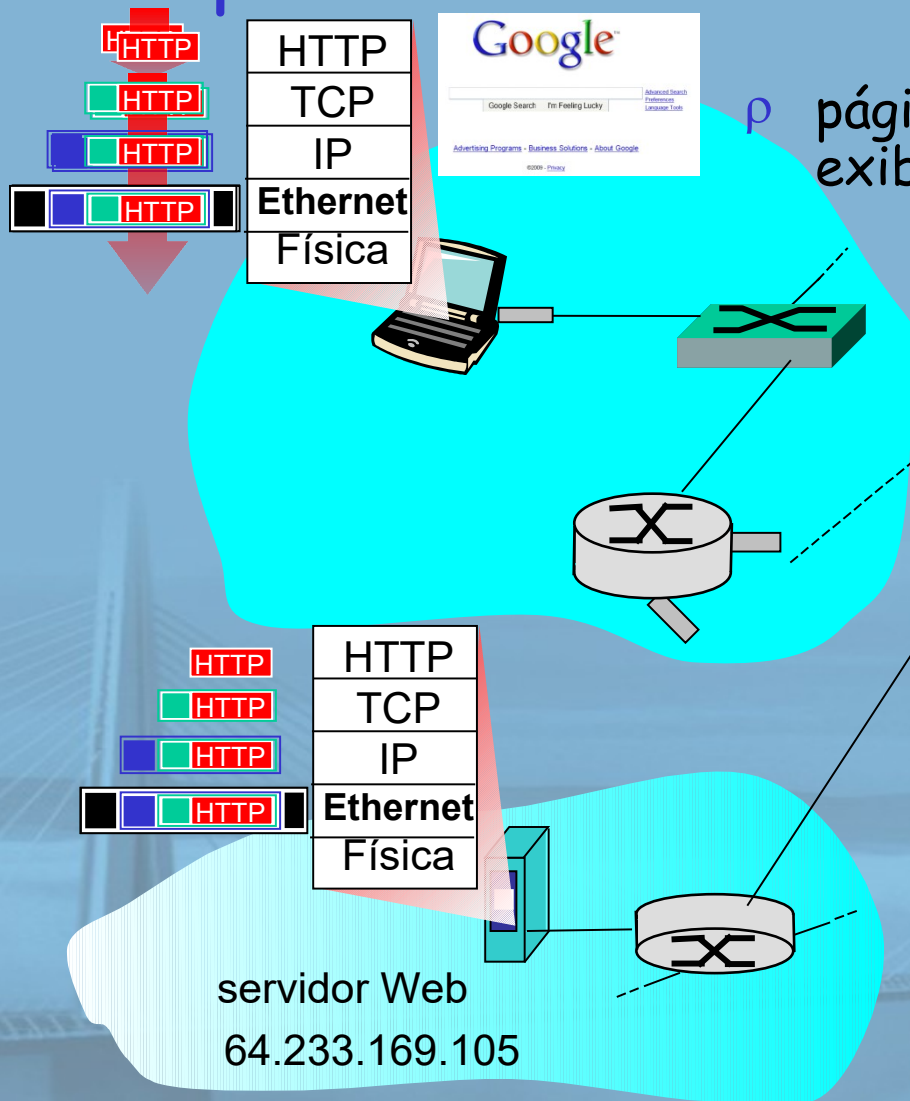
# Um dia na via... solicitação/ resposta HTTP

REDES DE  
COMPUTADORES  
E A INTERNET 5ª edição

Uma Abordagem Top-Down

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

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



## Capítulo 5: Resumo

- ρ princípios por trás dos serviços da camada de enlace de dados:
  - μ detecção e correção de erro
  - μ compartilhamento de canal de broadcast: acesso múltiplo
  - μ endereçamento da camada de enlace
- ρ instanciação e implementação de várias tecnologias da camada de enlace
  - μ Ethernet
  - μ LANS, VLANs comutadas
  - μ PPP
  - μ redes virtualizadas como camada de enlace: MPLS
- ρ síntese: um dia na vida de uma solicitação Web

# Capítulo 5: vamos fazer uma pausa

- ρ viagem pela pilha de protocolos **completa** (exceto a camada física)
- ρ conhecimento sólido dos princípios de rede e da prática
- ρ ..... poderíamos parar aqui... mas há **muitos** tópicos interessantes!
  - μ redes sem fio
  - μ multimídia
  - μ segurança
  - μ gerenciamento de rede