

# Capítulo 6

## A camada de enlace e LANs

Uma observação sobre o uso desses slides do PowerPoint:

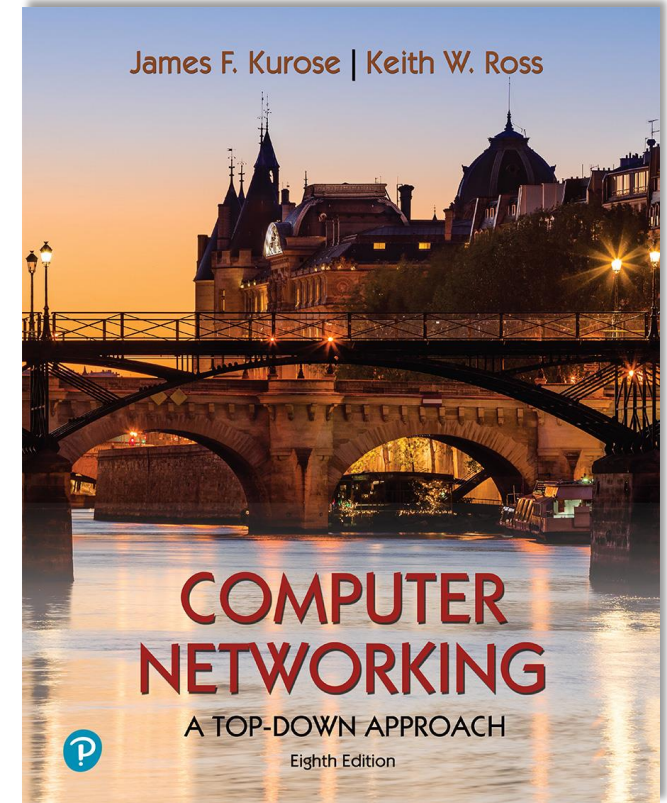
Estamos disponibilizando esses slides gratuitamente para todos (professores, alunos, leitores). Eles estão no formato PowerPoint para que você veja as animações e possa adicionar, modificar e excluir slides (inclusive este) e o conteúdo dos slides para atender às suas necessidades. Obviamente, eles representam *muito* trabalho de nossa parte. Em troca do uso, pedimos apenas o seguinte:

- Se você usar esses slides (por exemplo, em uma aula), mencione a fonte (afinal, gostaríamos que as pessoas usassem nosso livro!)
- Se você publicar algum slide em um site www, informe que ele foi adaptado de nossos slides (ou talvez idêntico a eles) e informe nossos direitos autorais sobre esse material.

Para obter um histórico de revisões, consulte a nota do slide desta página.

Obrigado e divirta-se! JFK/KWR

Todos os materiais têm direitos autorais de 1996 a 2023  
J.F. Kurose e K.W. Ross, Todos os direitos reservados



*Redes de computadores: A Top-Down Approach (Uma abordagem de cima para baixo)*

8<sup>th</sup> edition

Jim Kurose, Keith Ross  
Pearson, 2020

# Camada de enlace e LANs: nossos objetivos

- compreender os princípios por trás dos serviços de camada de link:
  - detecção e correção de erros
  - compartilhamento de um canal de transmissão: acesso múltiplo
  - endereçamento da camada de link
  - redes locais: Ethernet, VLANs
- redes de data center
- instanciação, implementação de várias tecnologias de camada de link



# Camada de enlace, LANs: roteiro

- **introdução**
- detecção e correção de erros
- protocolos de acesso múltiplo
- LANs
  - endereçamento, ARP
  - Ethernet
  - interruptores
  - VLANs
- virtualização de links: MPLS
- rede de data center



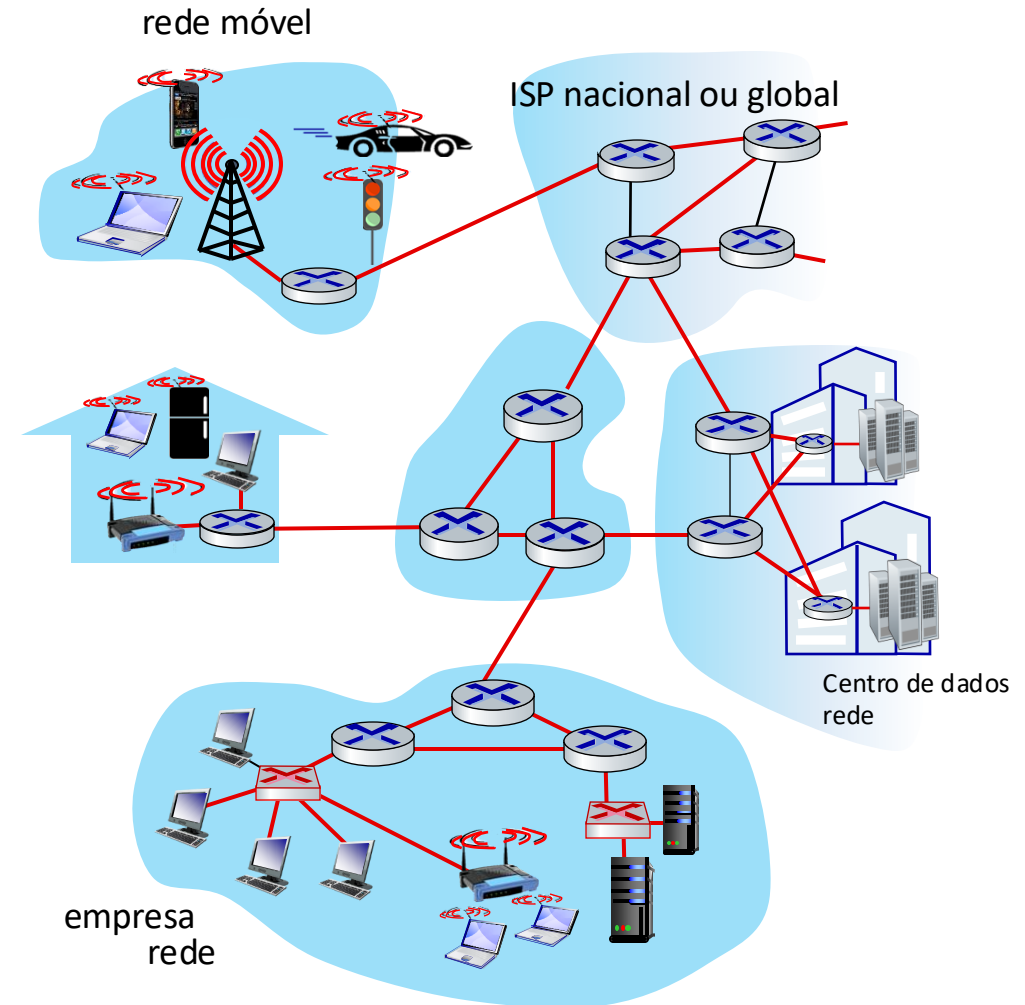
- Um dia na vida de uma solicitação da Web

# Camada de link: introdução

terminologia:

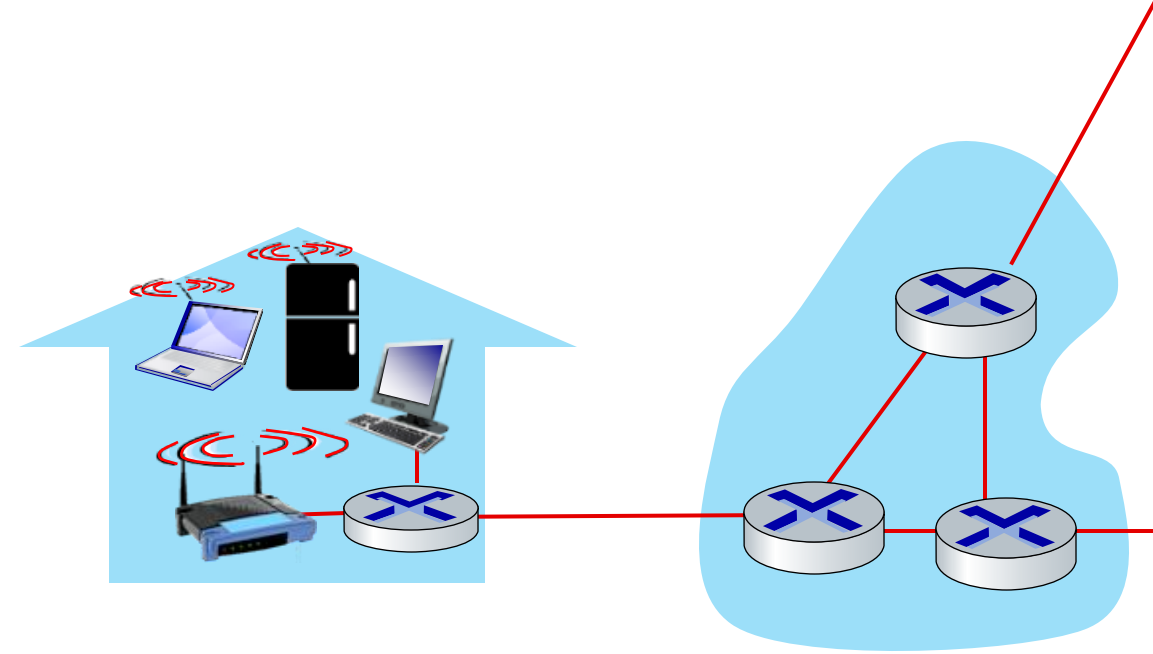
- hosts, roteadores: **nós**
- canais de comunicação que conectam nós **adjacentes** ao longo do caminho de comunicação: **links**
  - com fio , sem fio
  - LANs
- Pacote de camada 2: **quadro**, encapsula o datagrama

*A camada de link é responsável por transferência de datagrama de um nó para o nó fisicamente adjacente em um link*

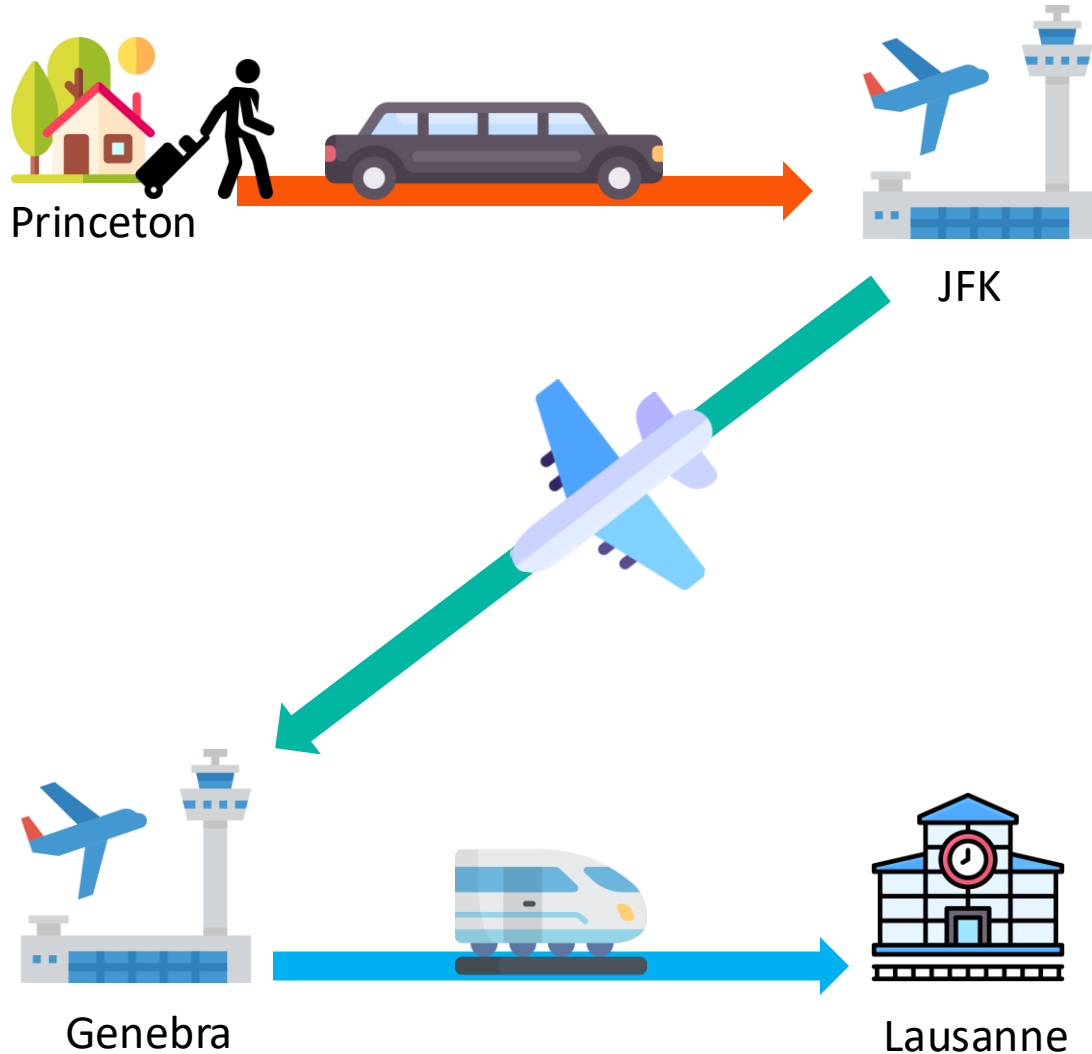


# Camada de link: contexto

- datagrama transferido por **diferentes protocolos de link** em diferentes links:
  - Por exemplo, WiFi no primeiro link, Ethernet no próximo link
- cada protocolo de link fornece serviços diferentes
  - Por exemplo, **pode ou não** fornecer transferência de dados confiável pelo link



# Analogia de transporte



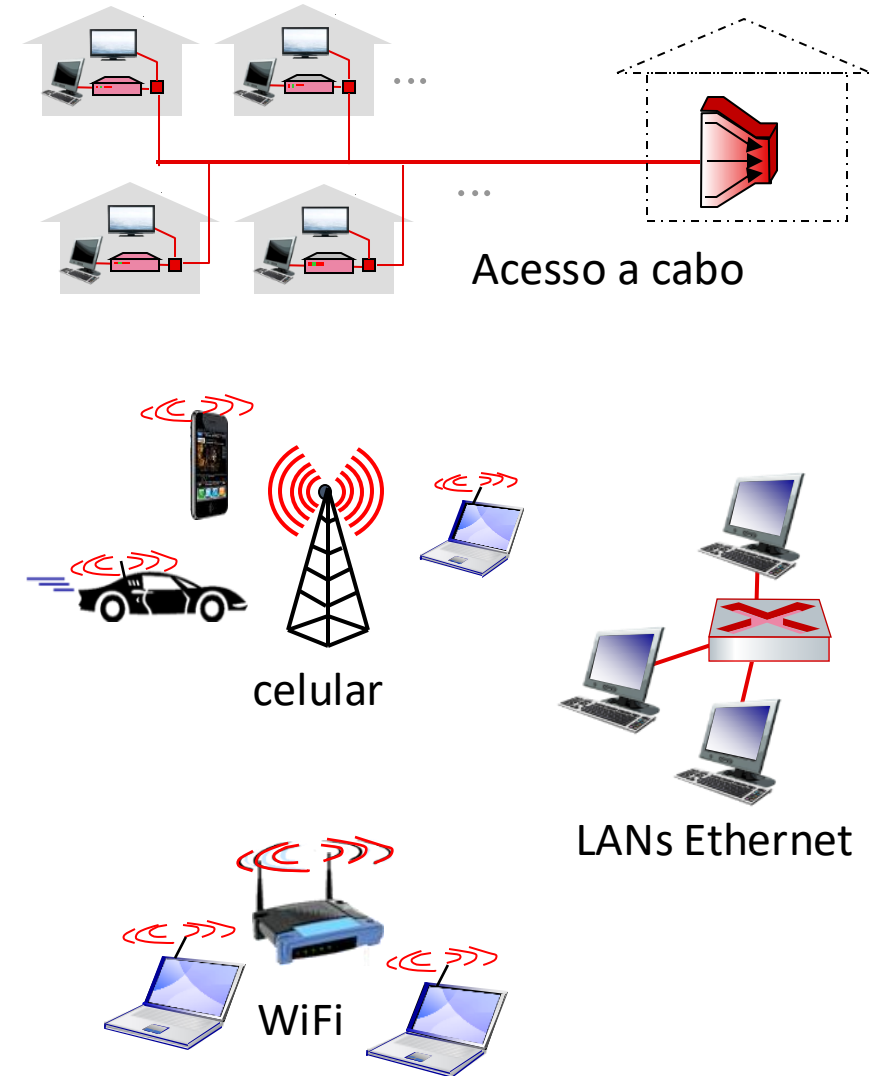
## analogia de transporte:

- viagem de Princeton para Lausanne
  - limusine: Princeton para JFK
  - avião: JFK para Genebra
  - trem: Genebra para Lausanne
- tourist = datagrama
- segmento de transporte = link de comunicação
- modo de transporte = protocolo de camada de link
- agente de viagens = algoritmo de roteamento



# Camada de link: serviços

- **enquadramento, acesso ao link:**
  - encapsular o datagrama em um quadro, adicionando cabeçalho, trailer
  - acesso ao canal se o meio for compartilhado
  - Os endereços "MAC" nos cabeçalhos dos quadros identificam a origem e o destino (diferente do endereço IP!)
- **fornecimento confiável entre nós adjacentes**
  - já sabemos como fazer isso!
  - raramente usado em links com baixo nível de erro de bit
  - links sem fio: altas taxas de erro
    - **P: Por que a confiabilidade no nível do link e no final?**



# Camada de link: serviços (mais)

## ■ controle de fluxo:

- ritmo entre nós adjacentes de envio e recebimento

## ■ detecção de erros:

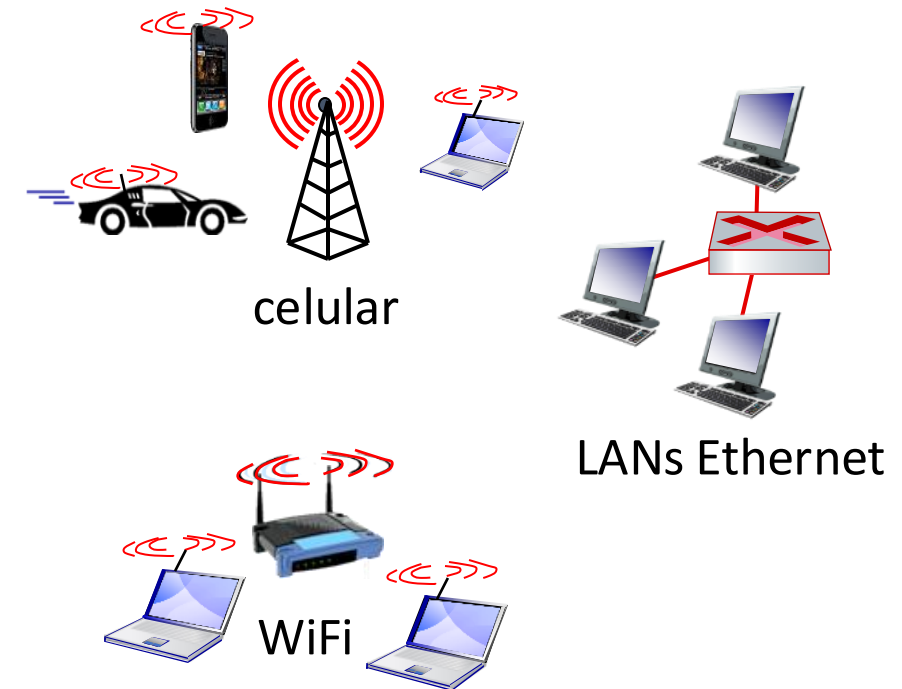
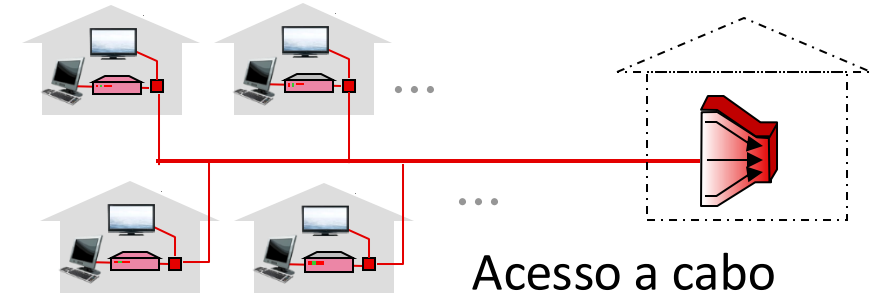
- erros causados por atenuação de sinal, ruído.
- o receptor detecta erros, sinaliza a retransmissão ou descarta o quadro

## ■ correção de erros:

- o receptor identifica *e corrige* os erros de bits sem retransmissão

## ■ half-duplex e full-duplex:

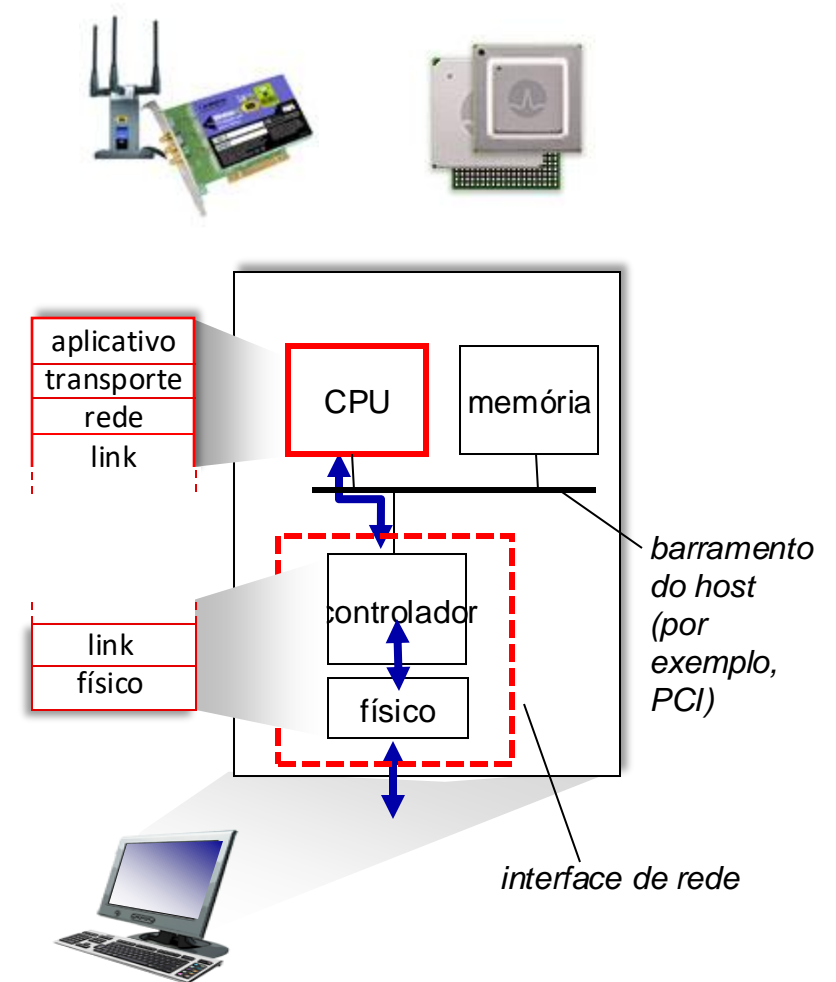
- com half duplex, os nós em ambas as extremidades do link podem transmitir, mas não ao mesmo tempo



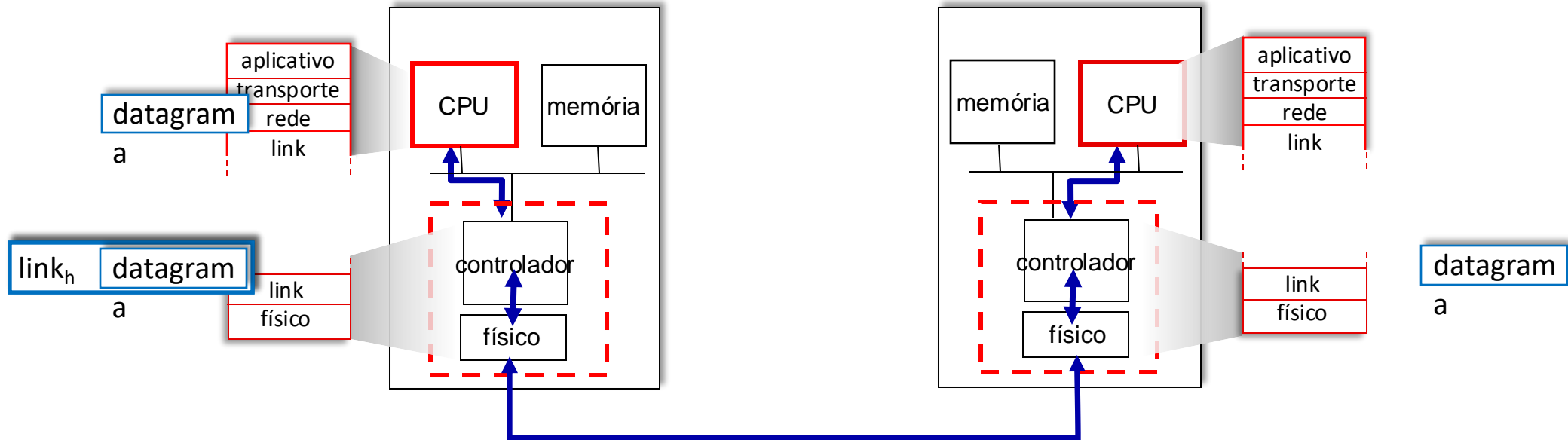


# Implementação da camada de link do host

- em todo e qualquer host
- camada de link implementada no chip ou na placa de interface de rede (NIC)
  - implementa o link, a camada física
- é conectado aos barramentos do sistema do host
- combinação de hardware, software, firmware



# Interfaces de comunicação



lado do envio:

- encapsula o datagrama em um quadro
- adiciona bits de verificação de erros, transferência confiável de dados, controle de fluxo, etc.

lado receptor:

- procura por erros, transferência confiável de dados, controle de fluxo, etc.
- extrai o datagrama, passa para a camada superior no lado receptor

# Camada de link, LANs: roteiro

- introdução
- detecção e correção de erros
- protocolos de acesso múltiplo
- LANs
  - endereçamento, ARP
  - Ethernet
  - interruptores
  - VLANs
- virtualização de links: MPLS
- rede de data center

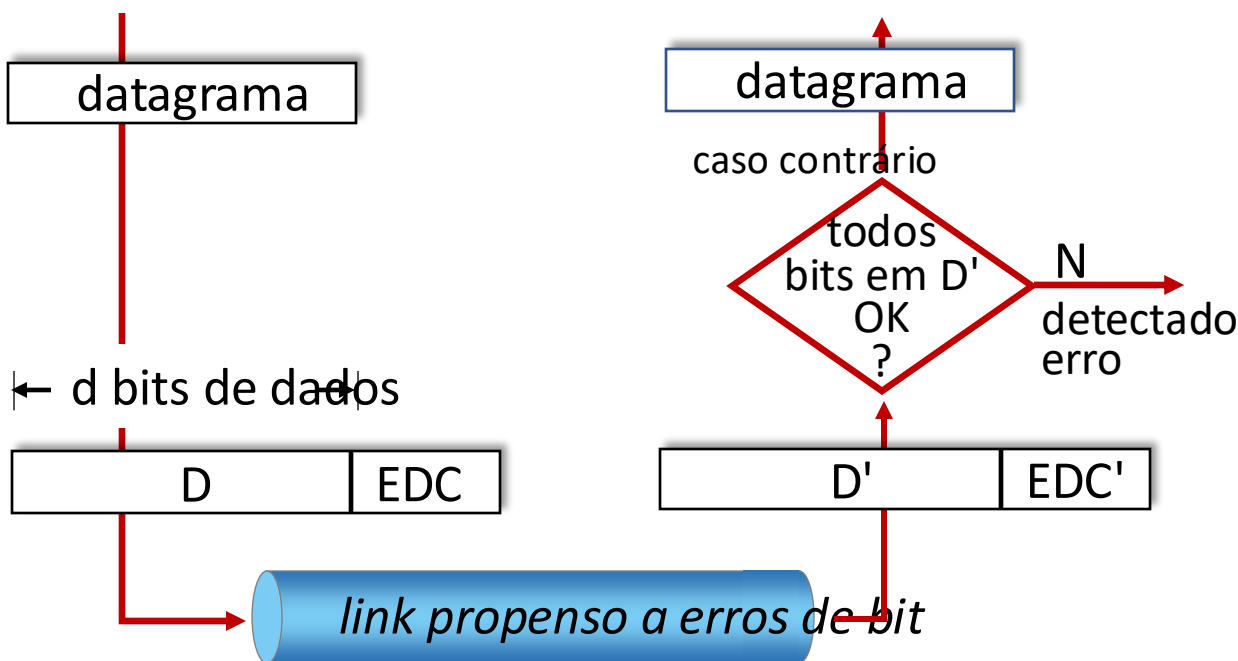


- Um dia na vida de uma solicitação da Web

# Detecção de erros

EDC: bits de detecção e correção de erros (por exemplo, redundância)

D: dados protegidos por verificação de erros, podem incluir campos de cabeçalho



A detecção de erros não é 100% confiável!

- O protocolo pode deixar passar alguns erros, mas raramente
- O campo maior de EDC produz melhor detecção e correção

# Verificação de paridade



## paridade de bit único:

- detectar erros de bit único

0111000110101011	1
------------------	---

←  $d$  bits de dados →  
 bit de paridade

Paridade par/ímpar: define o bit de paridade para que haja um número par/ímpar de 1's

## Na recepção:

- computar a paridade de  $d$  bits recebidos
- comparar com o bit de paridade recebido - se for diferente do erro detectado

Pode detectar **e** corrigir erros (sem retransmissão!)

- paridade bidimensional: detecta **e corrige** erros de bit único

				paridade de linha →
	$d_{1,1}$	...	$d_{1,j}$	$d_{1,j+1}$
	$d_{2,1}$	...	$d_{2,j}$	$d_{2,j+1}$
	...	...	...	...
	$d_{i,1}$	...	$d_{i,j}$	$d_{i,j+1}$
coluna paridade ↓	$d_{i+1,1}$	...	$d_{i+1,j}$	$d_{i+1,j+1}$

sem erros:

1	0	1	0	1	1
1	1	1	1	0	0
0	1	1	1	0	1
1	0	1	0	1	0

detectado e corrigível de um único bit erro:

1	0	1	0	1	1
1	0	1	1	0	0
0	1	1	1	0	1
1	0	1	0	1	0

paridade erro

# Checksum da Internet (revisão, consulte a seção 3.3)

**Objetivo:** detectar erros (ou seja, bits invertidos) no segmento transmitido

## remetente:

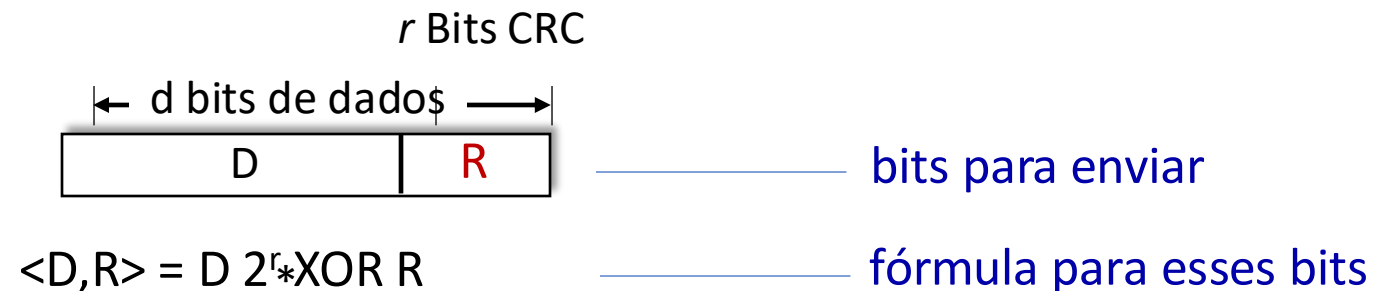
- tratar o conteúdo do segmento UDP (incluindo campos de cabeçalho UDP e endereços IP) como uma sequência de inteiros de 16 bits
- **soma de verificação:** adição (soma do complemento de um) do conteúdo do segmento
- valor de soma de verificação colocado no campo de soma de verificação UDP

## receptor:

- computar a soma de verificação do segmento recebido
- verifica se a soma de verificação computada é igual ao valor do campo de soma de verificação:
  - não é igual - erro detectado
  - igual - nenhum erro detectado. *Mas, mesmo assim, talvez haja erros?* Mais informações em ....

# Verificação de redundância cíclica (CRC)

- codificação de detecção de erros mais avançada
- **D**: bits de dados (dado, pense neles como um número binário)
- **G**: padrão de bits (gerador), de  $r+1$  bits (dado, especificado no padrão CRC)



**remetente:** calcula  $r$  bits CRC, **R**, de modo que  $\langle D, R \rangle$  *seja exatamente* divisível por  $G$  (mod 2)

- O receptor conhece  $G$ , divide  $\langle D, R \rangle$  por  $G$ . Se o resto for diferente de zero: erro detectado!
- pode detectar todos os erros de burst menores que  $r+1$  bits
- amplamente utilizados na prática (Ethernet, 802.11 WiFi)



# Verificação de redundância cíclica (CRC): exemplo

O remetente deseja calcular R  
de forma que:

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

... ou de forma equivalente (XOR R  
em ambos os lados):

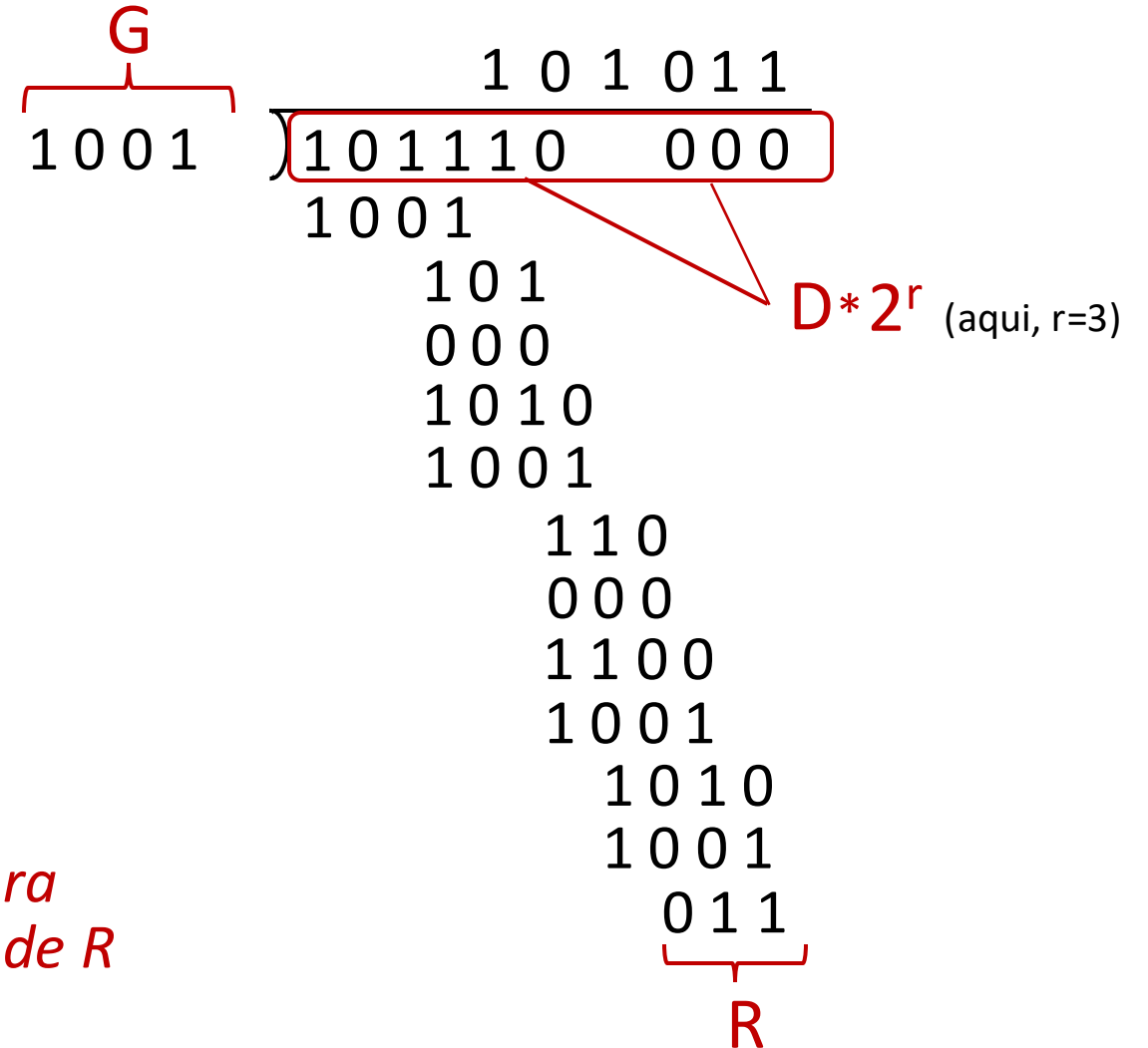
$$D \cdot 2^r = nG \text{ XOR } R$$

... que diz:

Se dividirmos  $D \cdot 2^r$  por G,  
queremos que o resto R seja  
satisfatório:

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

*algoritmo para  
computação de R*



# Camada de link, LANs: roteiro

- introdução
- detecção e correção de erros
- protocolos de acesso múltiplo
- LANs
  - endereçamento, ARP
  - Ethernet
  - interruptores
  - VLANs
- virtualização de links: MPLS
- rede de data center



- Um dia na vida de uma solicitação da Web

# Vários links de acesso, protocolos

dois tipos de "links":

- ponto a ponto
  - link ponto a ponto entre o switch Ethernet e o host
  - PPP para acesso discado
- **transmissão (fio ou meio compartilhado)**
  - Ethernet antiga
  - HFC upstream em rede de acesso baseada em cabo
  - LAN sem fio 802.11, satélite 4G/4G.



fio compartilhado (por exemplo,  
Ethernet com cabo)



Rádio compartilhado: 4G/5G



Rádio compartilhado: Wi-Fi



Rádio compartilhado: satélite



humanos em um coquetel  
(ar compartilhado, acústica)

# Protocolos de acesso múltiplo

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

## protocolo de acesso múltiplo

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

# Um protocolo de acesso múltiplo ideal

*dado:* canal de acesso múltiplo (MAC) de taxa  $R$  bps

*desejados:*

1. quando um nó deseja transmitir, ele pode enviar a uma taxa  $R$ .
2. quando  $M$  nós querem transmitir, cada um pode enviar a uma taxa média  $R/M$
3. totalmente descentralizado:
  - nenhum nó especial para coordenar as transmissões
  - sem sincronização de relógios, slots
4. simples

# Protocolos MAC: taxonomia

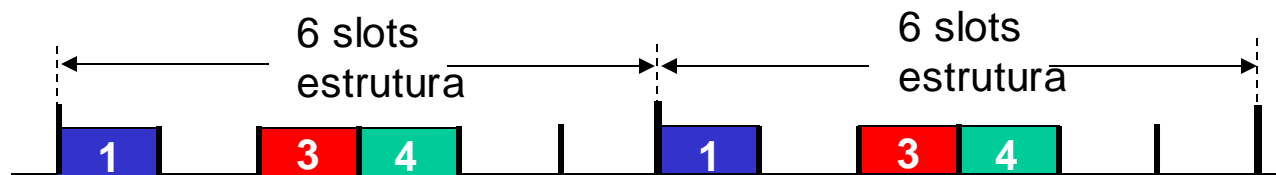
três classes amplas:

- **particionamento de canais**
  - dividir o canal em "pedaços" menores (intervalos de tempo, frequência, código)
  - alocar parte do nó para uso exclusivo
- **acesso aleatório**
  - canal não dividido, permite colisões
  - "recuperar-se" de colisões
- **"Taking turns" ( revezamento)**
  - os nós se revezam, mas os nós com mais para enviar podem se revezar por mais tempo

# Protocolos MAC de particionamento de canal: TDMA

## TDMA: acesso múltiplo por divisão de tempo

- acesso ao canal em "rodadas"
- cada estação recebe um slot de comprimento fixo (comprimento = tempo de transmissão do pacote) em cada rodada
- slots não utilizados ficam ociosos
- Exemplo: LAN de 6 estações, 1,3,4 têm pacotes para enviar, slots 2,5,6 ociosos

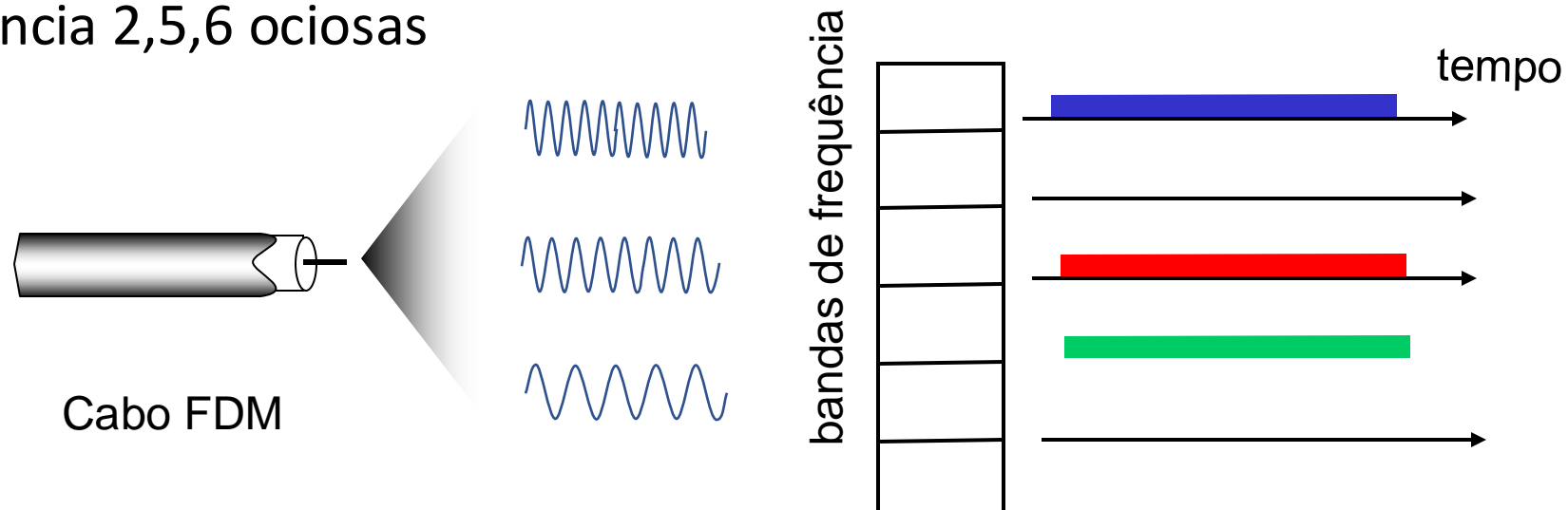




# Protocolos MAC de particionamento de canal: FDMA

## FDMA: acesso múltiplo por divisão de frequência

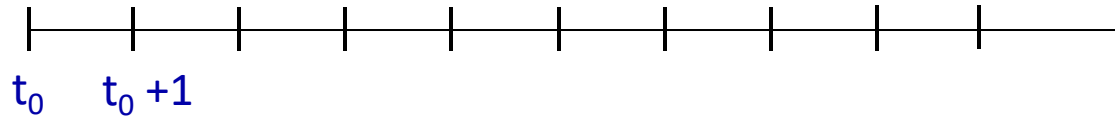
- espectro de canais dividido em bandas de frequência
- banda de frequência fixa atribuída a cada estação
- tempo de transmissão não utilizado em bandas de frequência ociosas
- Exemplo: LAN de 6 estações, 1,3,4 têm pacotes para enviar, bandas de frequência 2,5,6 ociosas



# Protocolos de acesso aleatório

- quando o nó tem um pacote para enviar
  - transmitir na taxa de dados total do canal R
  - nenhuma coordenação *a priori* entre os nós
- dois ou mais nós de transmissão:  
"colisão"
- O protocolo de acesso aleatório especifica:
  - como detectar colisões
  - como se recuperar de colisões (por exemplo, por meio de retransmissões atrasadas)
- exemplos de protocolos MAC de acesso aleatório:
  - ALOHA, ALOHA com slot
  - CSMA, CSMA/CD, CSMA/CA

# ALOHA com slot



## suposições:

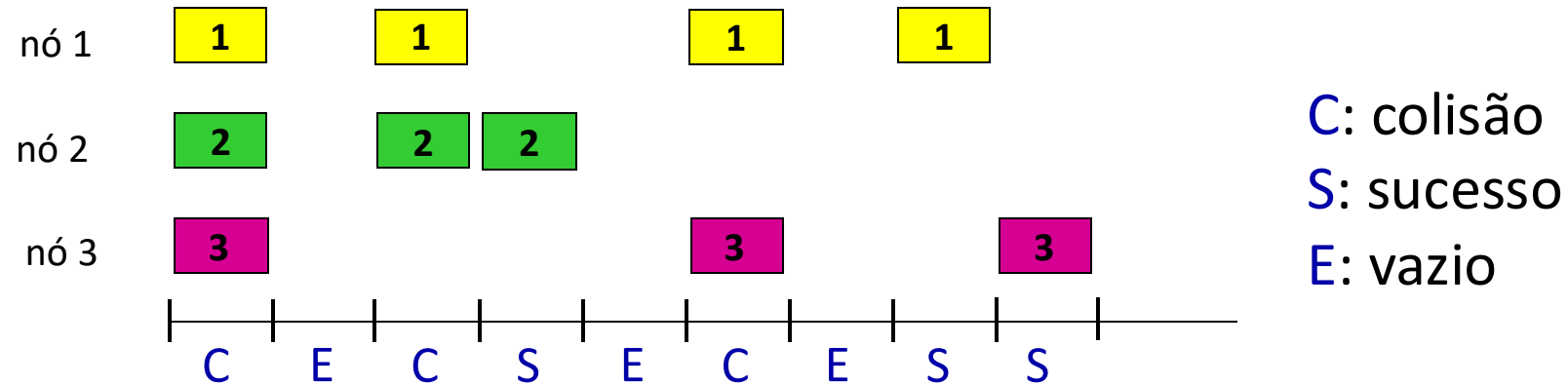
- todos os quadros do mesmo tamanho
- tempo dividido em slots de tamanho igual (tempo para transmitir 1 quadro)
- os nós começam a transmitir apenas no início do slot
- os nós são sincronizados
- se 2 ou mais nós transmitirem no slot, todos os nós detectarão a colisão

## operação:

- quando o nó obtém um novo quadro, transmite no próximo slot
  - *Se não houver colisão*: o nó pode enviar um novo quadro no próximo slot
  - *se houver colisão*: o nó retransmite o quadro em cada slot subsequente com probabilidade  $p$  até o sucesso

randomização - por quê?

# ALOHA com slot



## Prós:

- um único nó ativo pode transmitir continuamente na taxa total do canal
- altamente descentralizado: somente os slots nos nós precisam estar sincronizados
- simples

## Contras:

- colisões, desperdício de slots
- slots ociosos
- Os nós podem ser capazes de detectar a colisão em menos tempo do que o necessário para transmitir o pacote
- sincronização de relógio

# ALOHA com slot: eficiência

**eficiência:** fração de longo prazo de slots bem-sucedidos (muitos nós, todos com muitos quadros para enviar)

- *Suponha:*  $N$  nós com muitos quadros para enviar, cada um transmitindo em um slot com probabilidade  $p$ 
  - probabilidade de um determinado nó ter sucesso em um slot =  $p(1-p)^{N-1}$
  - probabilidade de que *qualquer* nó tenha sucesso =  $Np(1-p)^{N-1}$
  - eficiência máxima: encontrar  $p^*$  que maximize  $Np(1-p)^{N-1}$
  - para muitos nós, considere o limite de  $Np^*(1-p^*)^{N-1}$  à medida que  $N$  chega ao infinito:

*eficiência máxima =  $1/e = 0,37$*

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



# CSMA (acesso múltiplo com senso de portadora)

**CSMA simples:** escuta antes de transmitir:

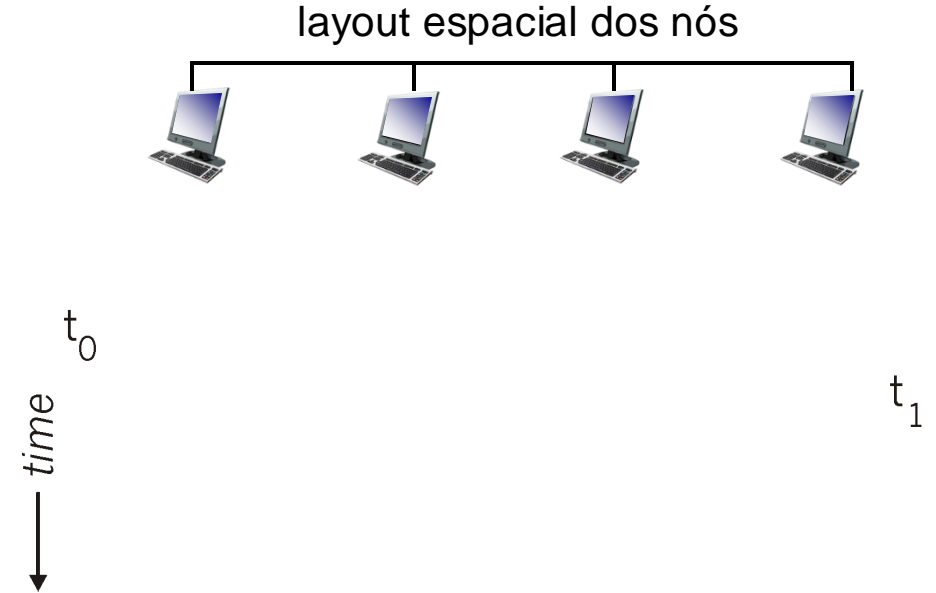
- se o canal for detectado como ocioso: transmitir o quadro inteiro
  - se o canal for detectado como ocupado: adiar a transmissão
- analogia humana: não interrompa os outros!

**CSMA/CD:** CSMA com *detecção de colisão*

- colisões *detectadas* em um curto espaço de tempo
  - transmissões colidentes abortadas, reduzindo o desperdício de canal
  - Detecção de colisão fácil com fio, difícil com o sistema sem fio
- analogia humana: o conversador educado

# CSMA: colisões

- as colisões *ainda* podem ocorrer com a detecção de portadora:
  - o atraso de propagação significa que dois nós podem não ouvir a transmissão recém-iniciada um do outro
- **colisão**: perda de todo o tempo de transmissão do pacote
  - a distância e o atraso de propagação desempenham um papel importante na determinação da probabilidade de colisão





# CSMA/CD:

- O CSMA/CD reduz a quantidade de tempo desperdiçado em colisões
  - transmissão interrompida por detecção de colisão



# Algoritmo Ethernet CSMA/CD

1. A Ethernet recebe um datagrama da camada de rede e cria um quadro
2. Se a Ethernet detectar o canal:
  - se **estiver ocioso**: iniciar a transmissão do quadro.
  - se estiver **ocupado**: aguarde até que o canal esteja ocioso e, em seguida, transmita
3. Se todo o quadro for transmitido sem colisão - pronto!
4. Se outra transmissão for detectada durante o envio: abortar, enviar sinal de congestionamento
5. Após o abortamento, insira *o backoff binário (exponencial)*:
  - após a *mésima* colisão, escolhe  $K$  aleatoriamente entre  $\{0, 1, 2, \dots, 2^m - 1\}$ . A Ethernet aguarda  $K \cdot 512$  bit vezes e retorna à Etapa 2
  - mais colisões: intervalo de backoff mais longo

# "Protocolos MAC de "revezamento

## protocolos MAC de particionamento de canal:

- compartilhar o canal de *forma eficiente e justa* em alta carga
- ineficiente com carga baixa: atraso no acesso ao canal, largura de banda  $1/N$  alocada mesmo se houver apenas um nó ativo!

## protocolos MAC de acesso aleatório

- eficiente em carga baixa: um único nó pode utilizar totalmente o canal
- alta carga: sobrecarga de colisão

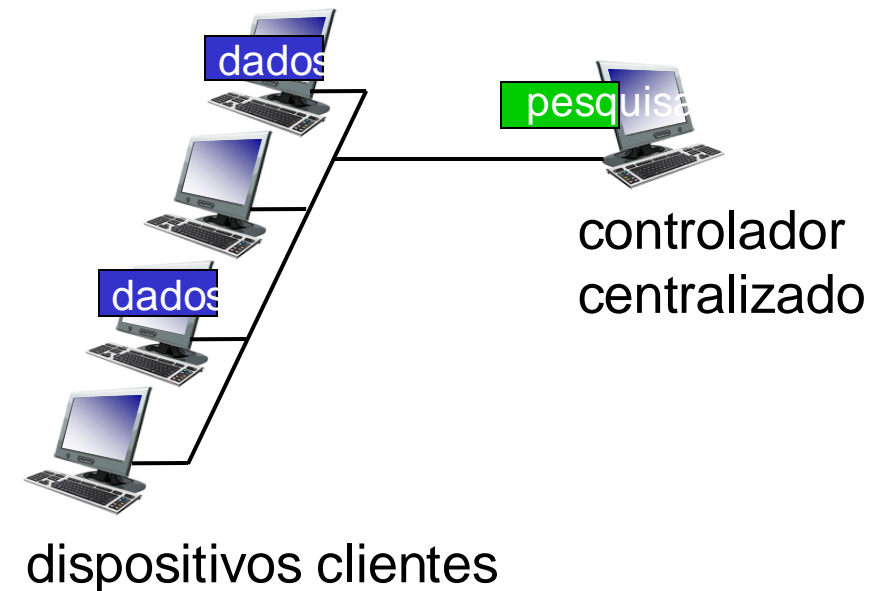
## "Protocolos de "revezamento

- procure o melhor dos dois mundos!

# "Protocolos MAC de "revezamento

## votação:

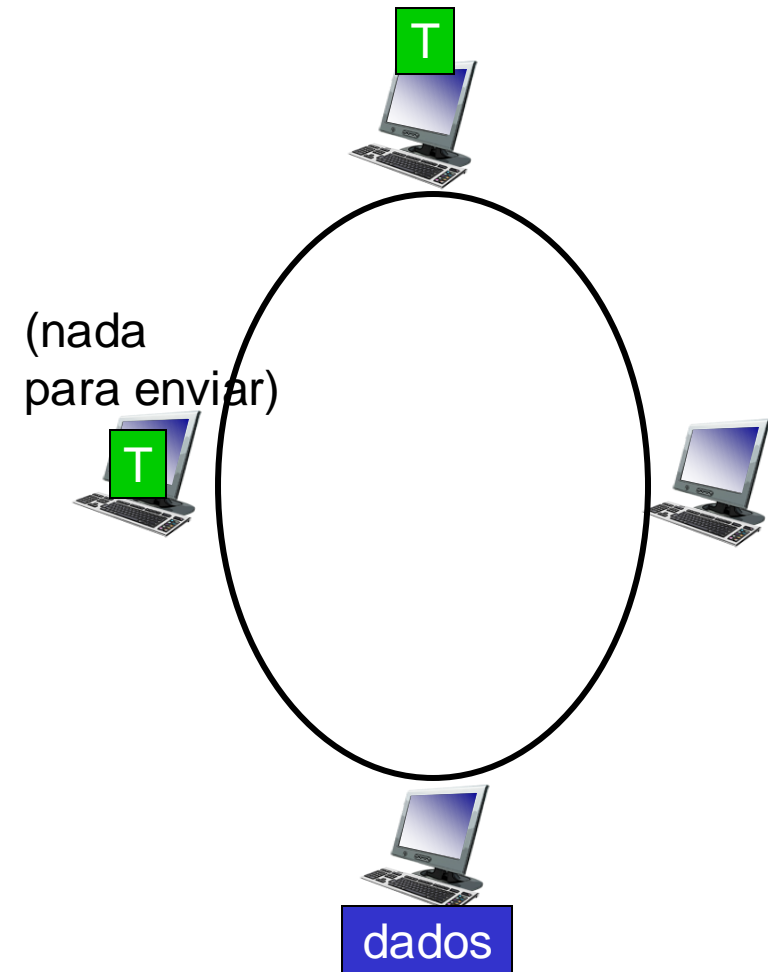
- o controlador centralizado "convida" os outros nós a transmitir, por sua vez
- normalmente usado com dispositivos "burros"
- preocupações:
  - sobrecarga de sondagem
  - latência
  - ponto único de falha (mestre)
- O Bluetooth usa polling



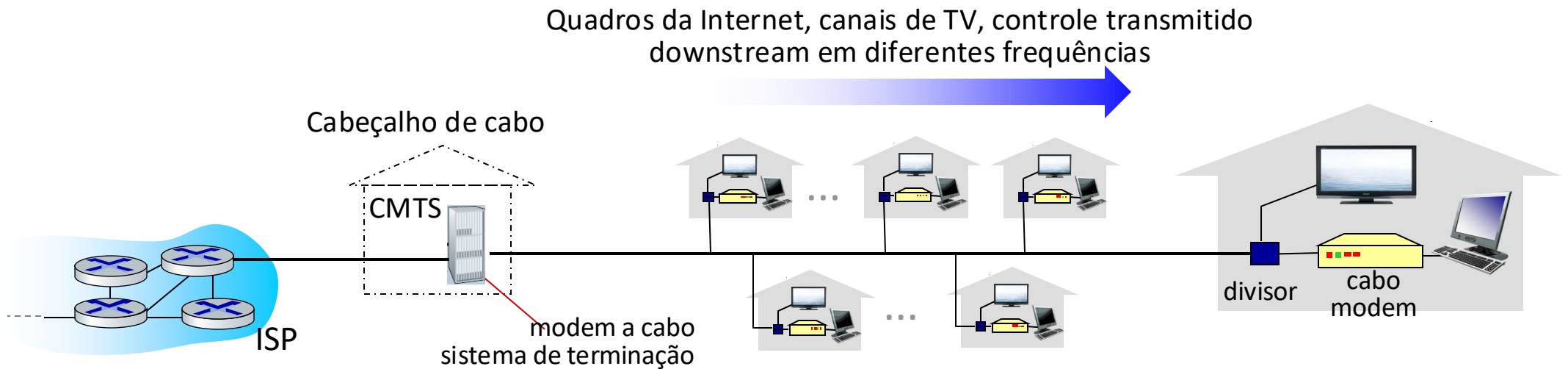
# "Protocolos MAC de "revezamento

## passagem de token:

- mensagem *de token* de controle explicitamente passada de um nó para o outro, sequencialmente
  - transmitir enquanto estiver segurando o token
- preocupações:
  - sobrecarga de token
  - latência
  - ponto único de falha (token)

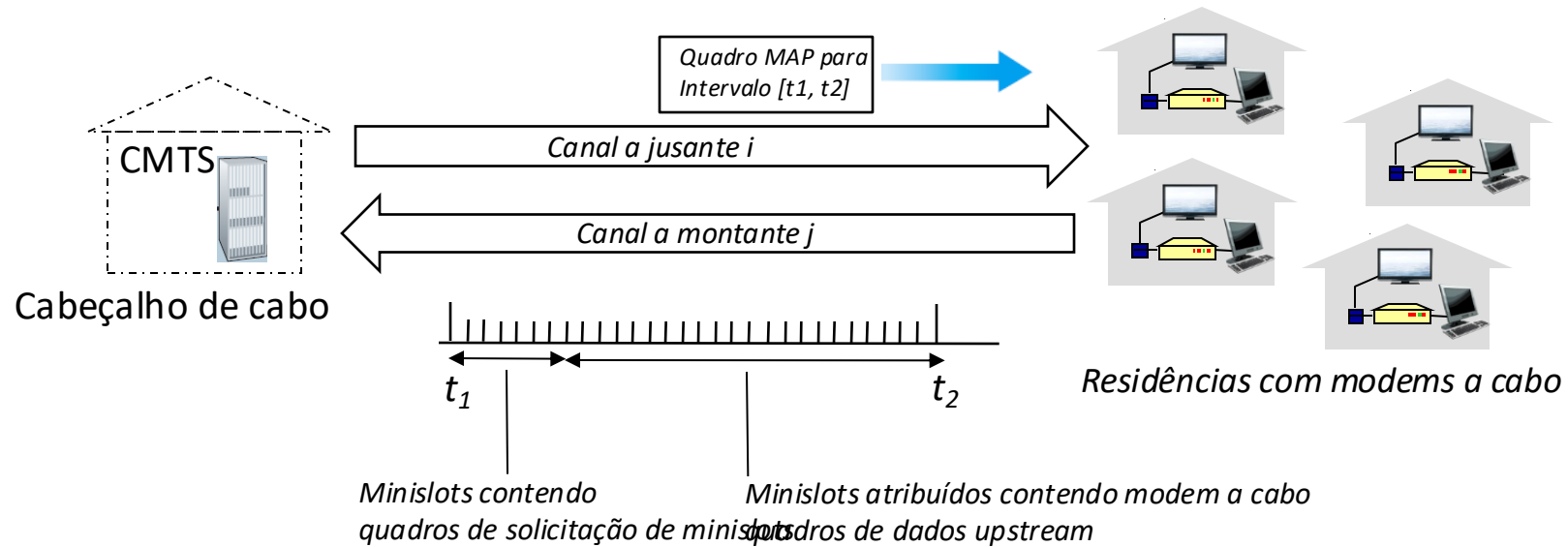


# Rede de acesso a cabo: FDM, TDM e acesso randômico!



- **vários** canais FDM downstream (broadcast): até 1,6 Gbps/canal
  - um único CMTS transmite em canais
- **vários** canais upstream (até 1 Gbps/canal)
  - **acesso múltiplo**: todos os usuários disputam (acesso aleatório) determinados intervalos de tempo do canal upstream; outros recebem TDM

# Rede de acesso a cabo:



**DOCSIS:** especificação de interface de serviço de dados por cabo

- FDM em canais de frequência upstream e downstream
- TDM upstream: alguns slots atribuídos, alguns têm contenção
  - Quadro MAP de downstream: atribui slots de upstream
  - solicitação de slots (e dados) upstream transmitidos acesso aleatório (backoff binário) em slots selecionados



# Resumo dos protocolos MAC

- **particionamento de canais**, por tempo, frequência ou código
  - Divisão de tempo, divisão de frequência
- **acesso aleatório** (dinâmico),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - detecção de portadora: fácil em algumas tecnologias (com fio), difícil em outras (sem fio)
  - CSMA/CD usado na Ethernet
  - CSMA/CA usado no 802.11
- **revezamento**
  - polling do site central, passagem de token
  - Bluetooth, FDDI, token ring

# Camada de link, LANs: roteiro

- introdução
- detecção e correção de erros
- protocolos de acesso múltiplo
- **LANs**
  - endereçamento, ARP
  - Ethernet
  - interruptores
  - VLANs
- virtualização de links: MPLS
- rede de data center



- Um dia na vida de uma solicitação da Web

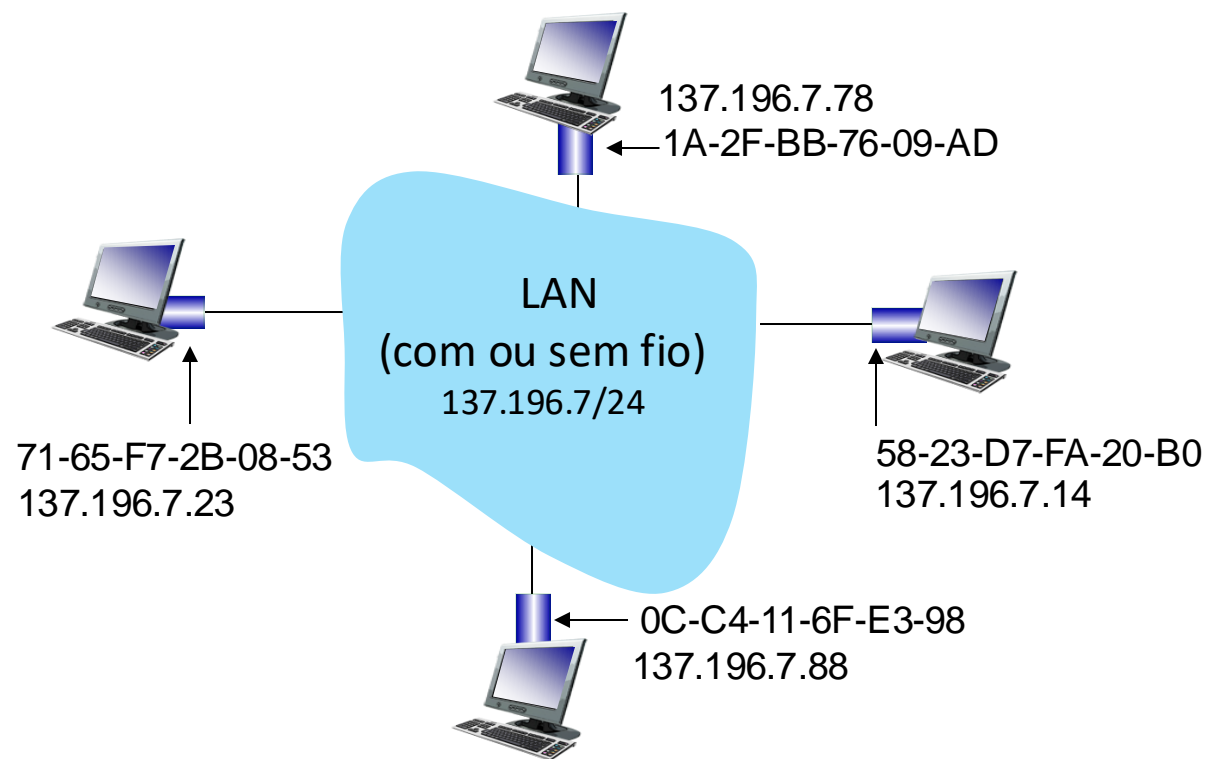
# Endereços MAC

- Endereço IP de 32 bits:
  - endereço da *camada de rede* para a interface
  - usado para encaminhamento na camada 3 (camada de rede)
  - Por exemplo: 128.119.40.136
- Endereço MAC (ou LAN ou físico ou Ethernet):
  - função: usado "localmente" para levar o quadro de uma interface para outra interface fisicamente conectada (mesma sub-rede, no sentido de endereçamento IP)
  - Endereço MAC de 48 bits (para a maioria das LANs) gravado na ROM da placa de rede, às vezes também configurável por software
  - Por exemplo: 1A-2F-BB-76-09-AD
    - notação hexadecimal (base 16)  
(cada "numeral" representa 4 bits)

# Endereços MAC

cada interface na LAN

- tem um endereço **MAC** exclusivo de 48 bits
- tem um endereço IP de 32 bits localmente exclusivo (como vimos)

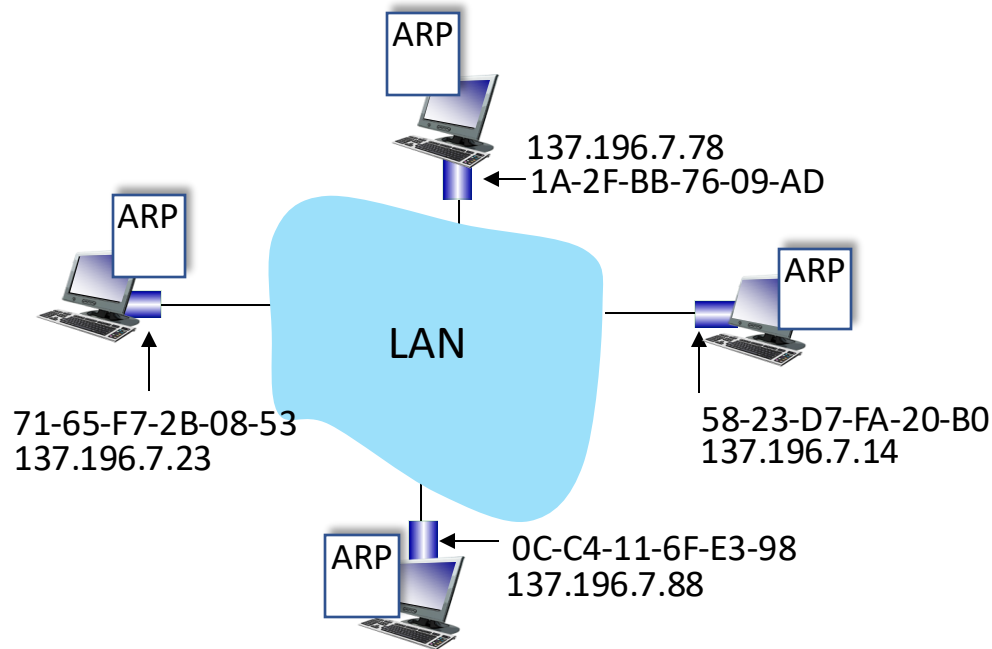


# Endereços MAC

- Alocação de endereços MAC administrada pelo IEEE
- O fabricante compra parte do espaço de endereço MAC (para garantir a exclusividade)
- analogia:
  - Endereço MAC: como o número do seguro social
  - Endereço IP: como o endereço postal
- Endereço plano MAC: portabilidade
  - pode mover a interface de uma LAN para outra
  - endereço IP de recall *não* portátil: depende da sub-rede IP à qual o nó está conectado

# ARP: protocolo de resolução de endereços

*Pergunta:* como determinar o endereço MAC da interface, sabendo seu endereço IP?



**Tabela ARP:** cada nó IP (host, roteador) na LAN tem uma tabela

- 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 endereços será esquecido (normalmente 20 minutos)

# Protocolo ARP em ação

Exemplo: A deseja enviar um datagrama para B

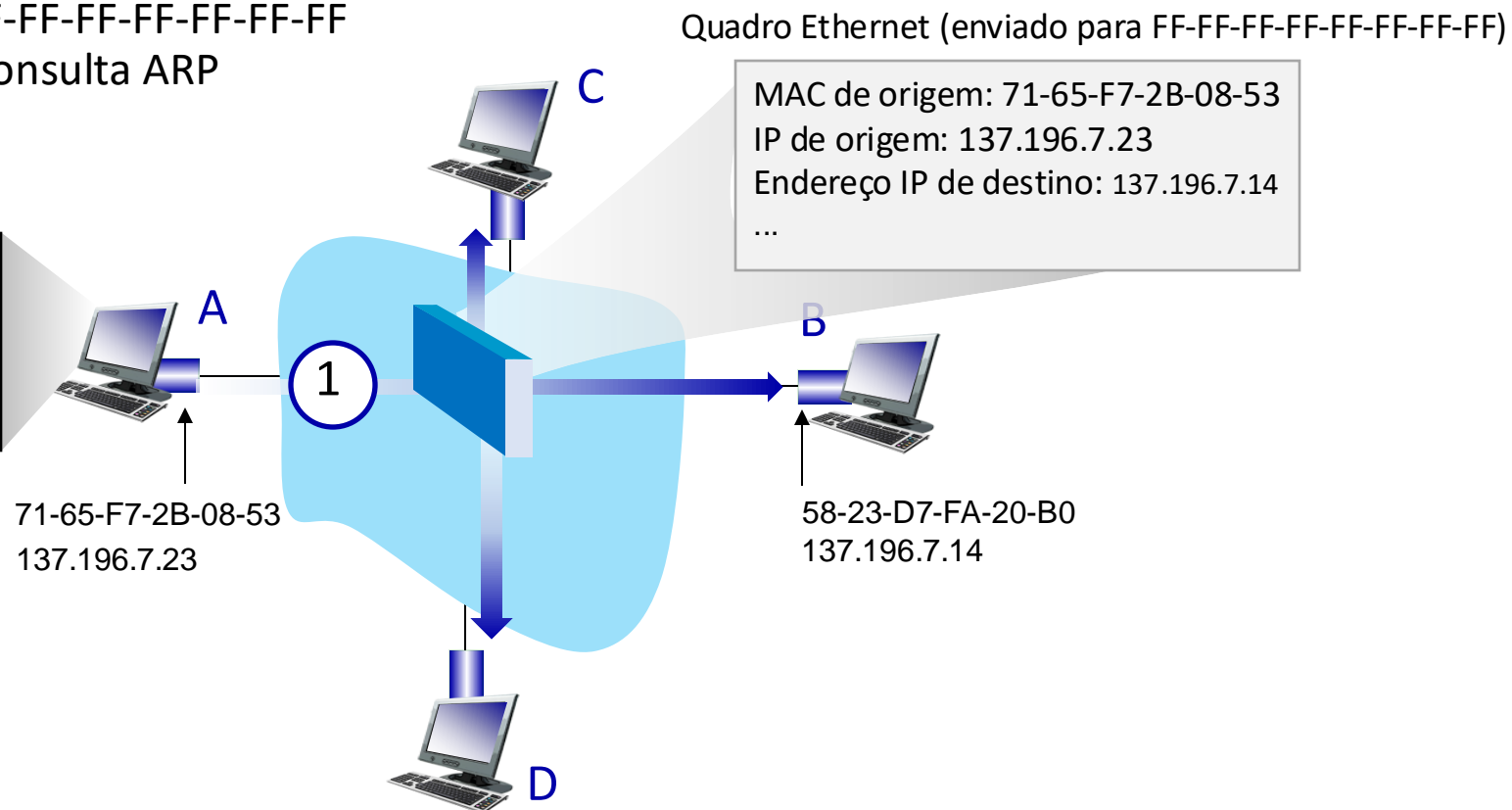
- O endereço MAC de B não está na tabela ARP de A, portanto A usa ARP para encontrar o endereço MAC de B

A transmite uma consulta ARP, contendo o endereço IP de B

- 1 endereço MAC de destino = FF-FF-FF-FF-FF-FF-FF-FF  
todos os nós da LAN recebem a consulta ARP

Tabela ARP em A

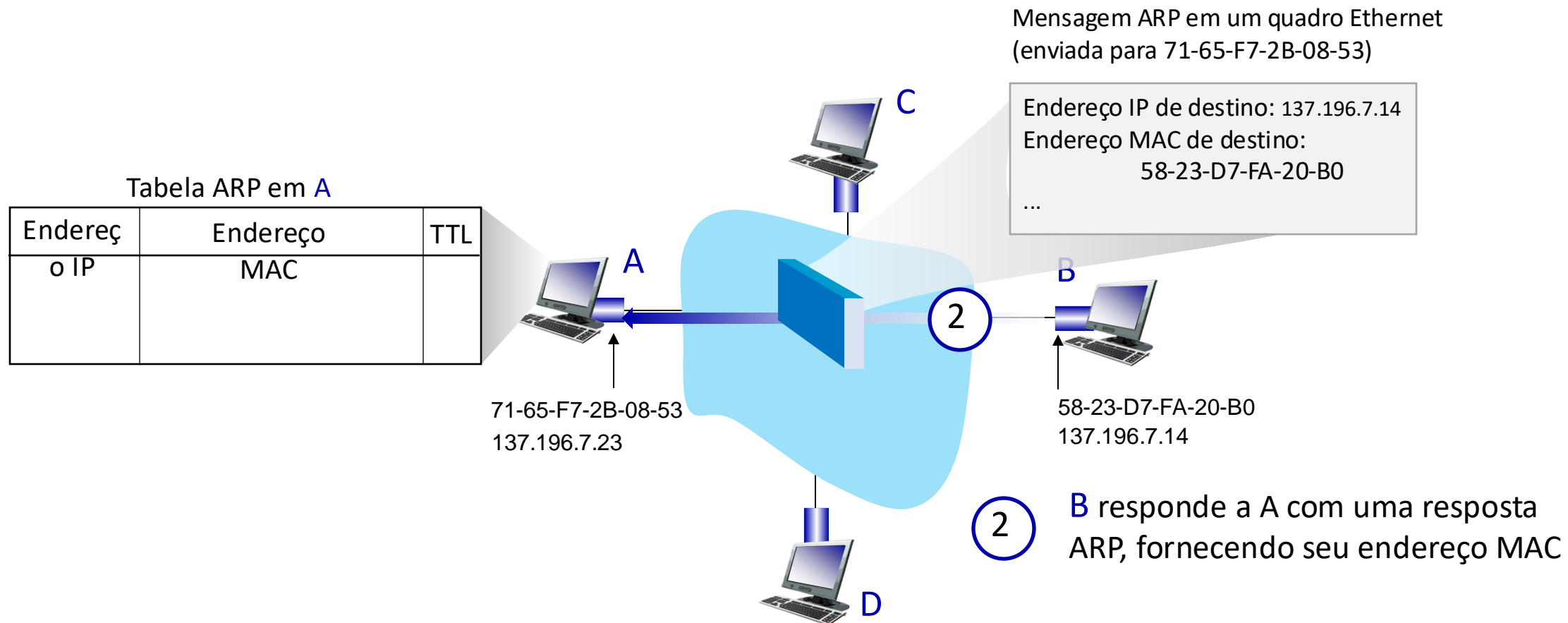
Endereç o IP	Endereço MAC	TTL



# Protocolo ARP em ação

exemplo: A deseja enviar um datagrama para B

- O endereço MAC de B não está na tabela ARP de A, portanto A usa ARP para encontrar o endereço MAC de B

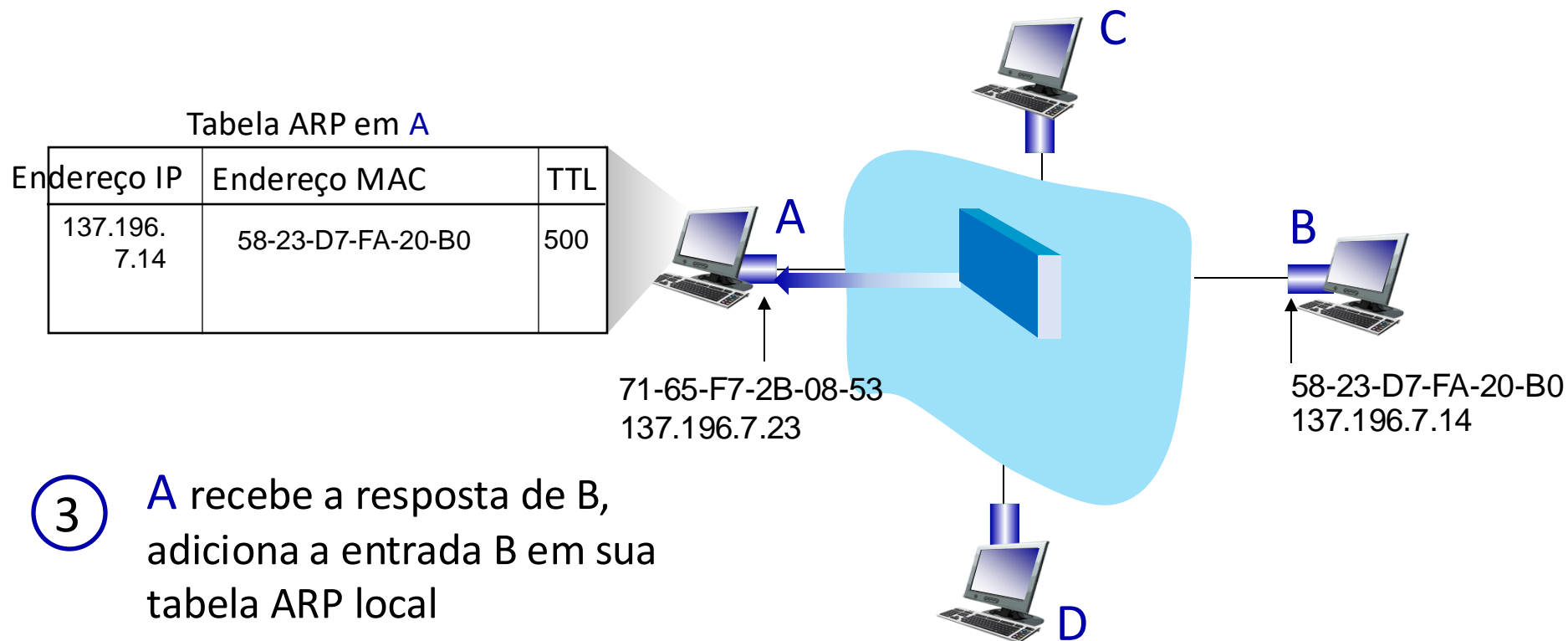




# Protocolo ARP em ação

exemplo: A deseja enviar um datagrama para B

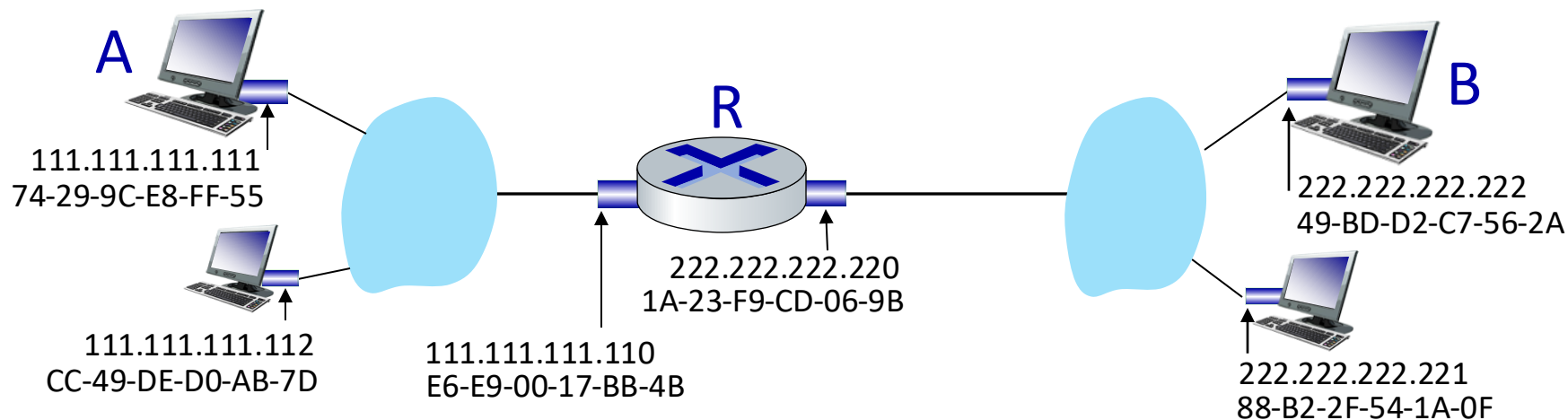
- O endereço MAC de B não está na tabela ARP de A, portanto A usa ARP para encontrar o endereço MAC de B



# Roteamento para outra sub-rede: endereçamento

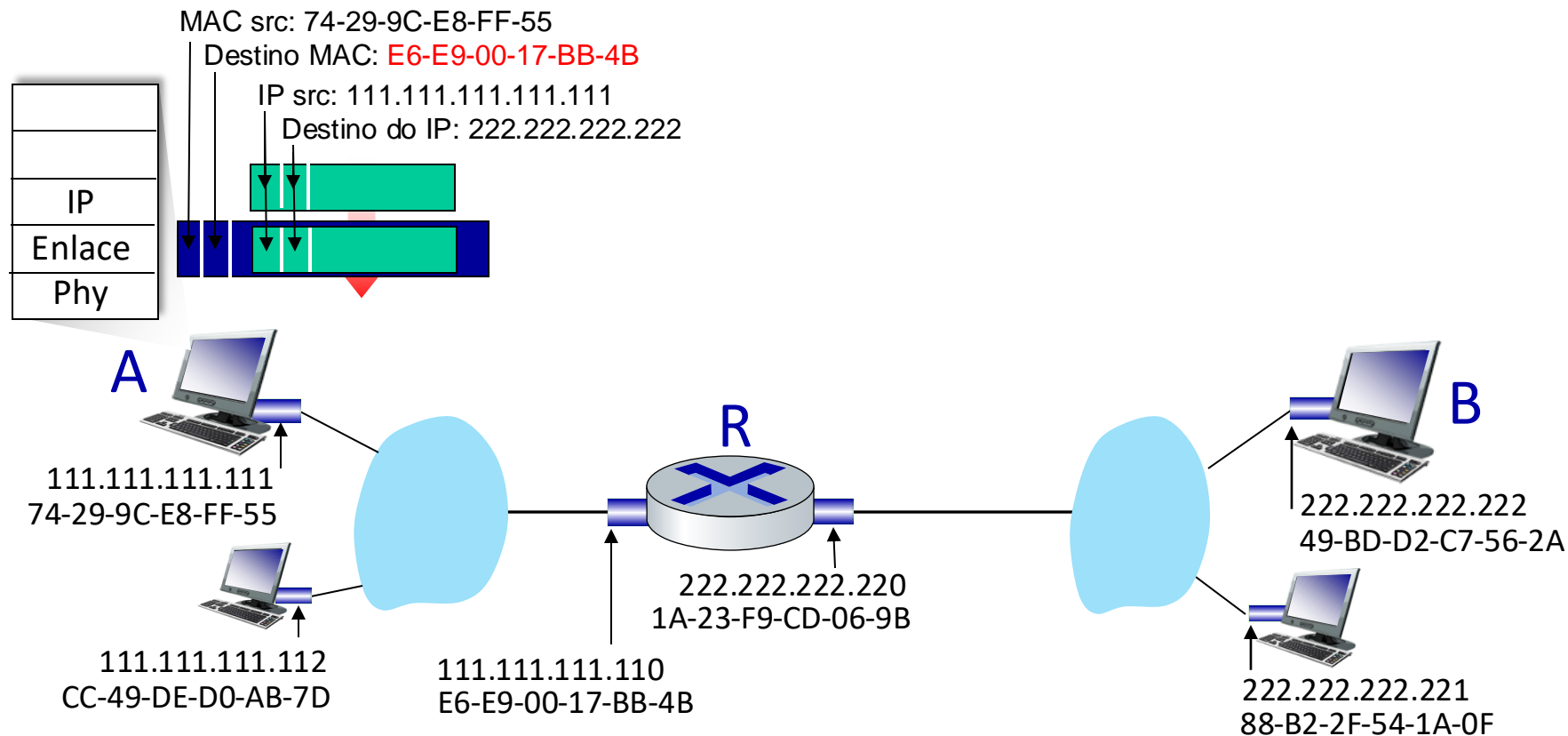
Passo a passo: envio de um datagrama de *A* para *B* via *R*

- foco no endereçamento - nos níveis de IP (datagrama) e camada MAC (quadro)
- suponha que:
  - A sabe o endereço IP de B
  - A sabe o endereço IP do roteador de primeiro salto, R (como?)
  - A sabe o endereço MAC de R (como?)



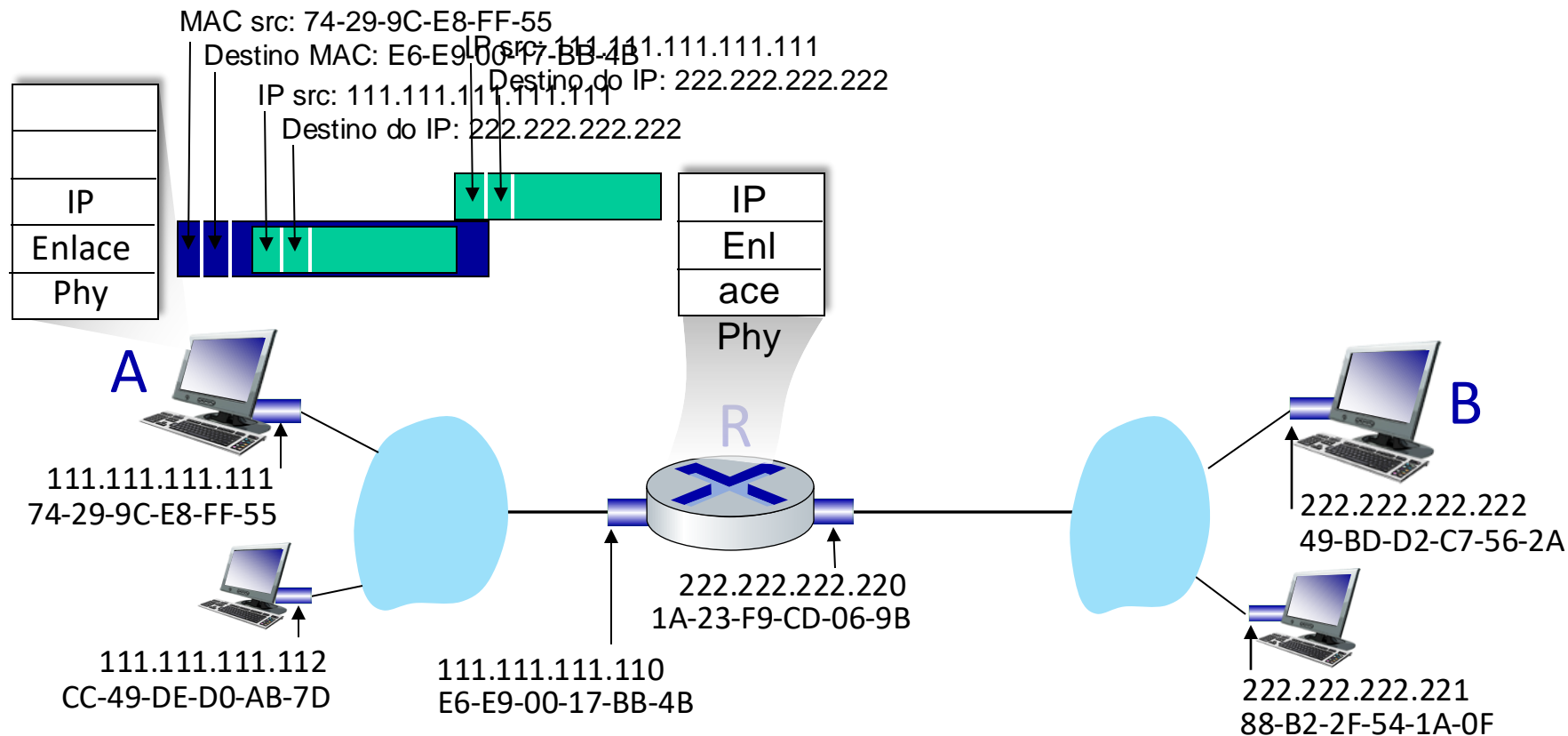
# Roteamento para outra sub-rede: endereçamento

- A cria um datagrama IP com origem IP A, destino B
- Cria um quadro da camada de enlace que contém um datagrama IP de A para B
  - O endereço MAC **de R** é o destino do quadro



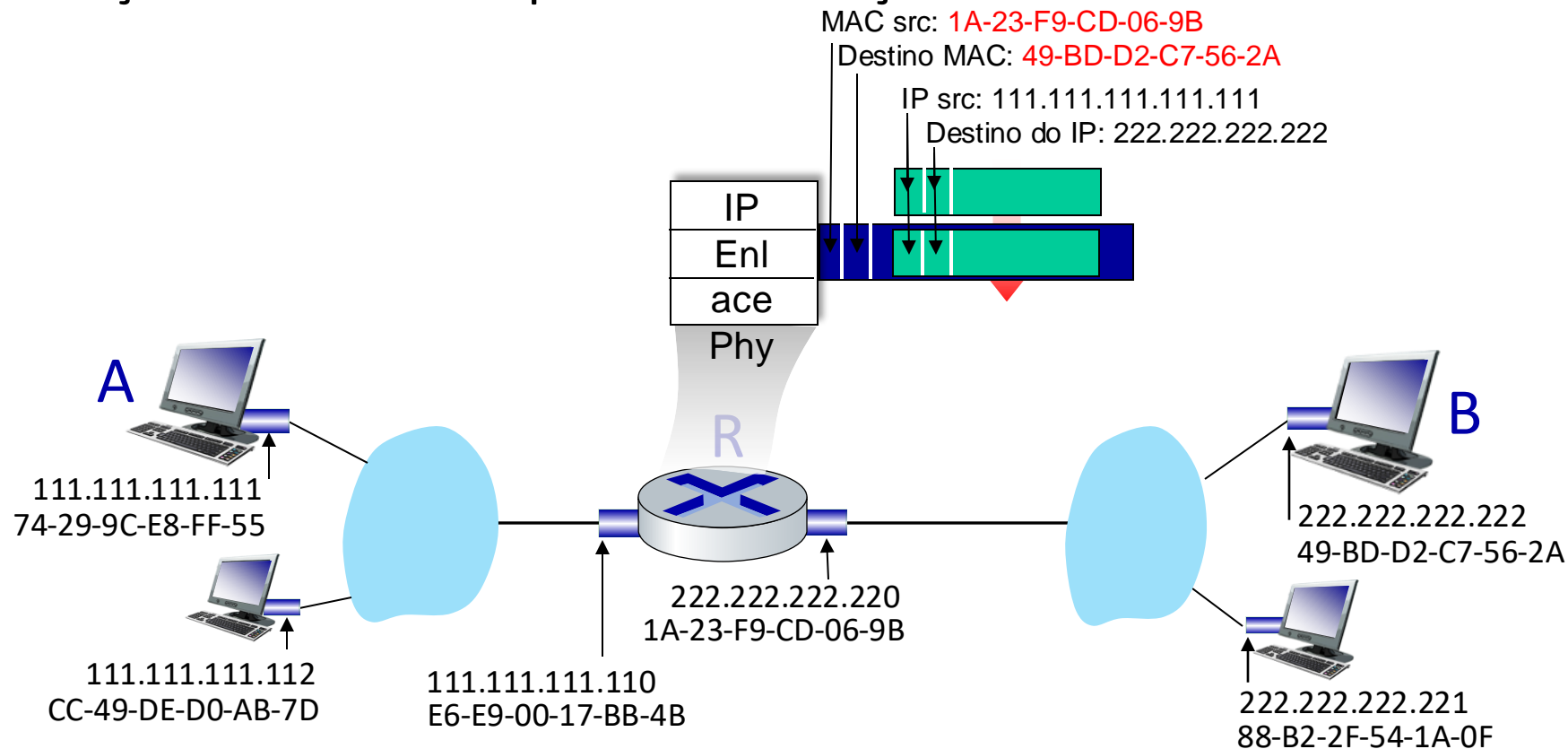
# Roteamento para outra sub-rede: endereçamento

- quadro enviado de A para R
- frame recebido em R, datagrama removido, passado para o IP



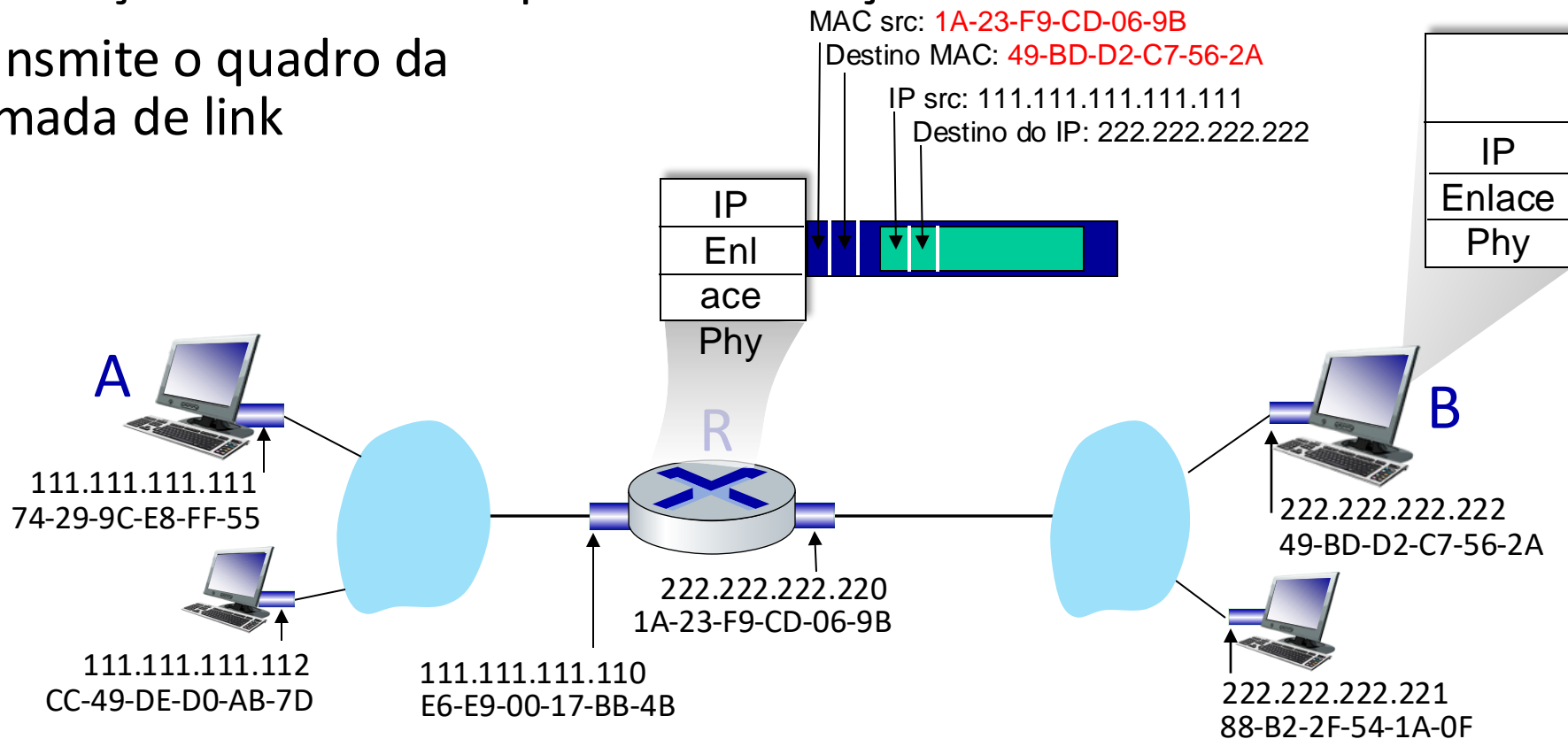
# Roteamento para outra sub-rede: endereçamento

- R determina a interface de saída, passa o datagrama com a fonte IP A e o destino B para a camada de link
- R cria um quadro de camada de enlace contendo um datagrama IP de A para B. Endereço de destino do quadro: Endereço MAC de B



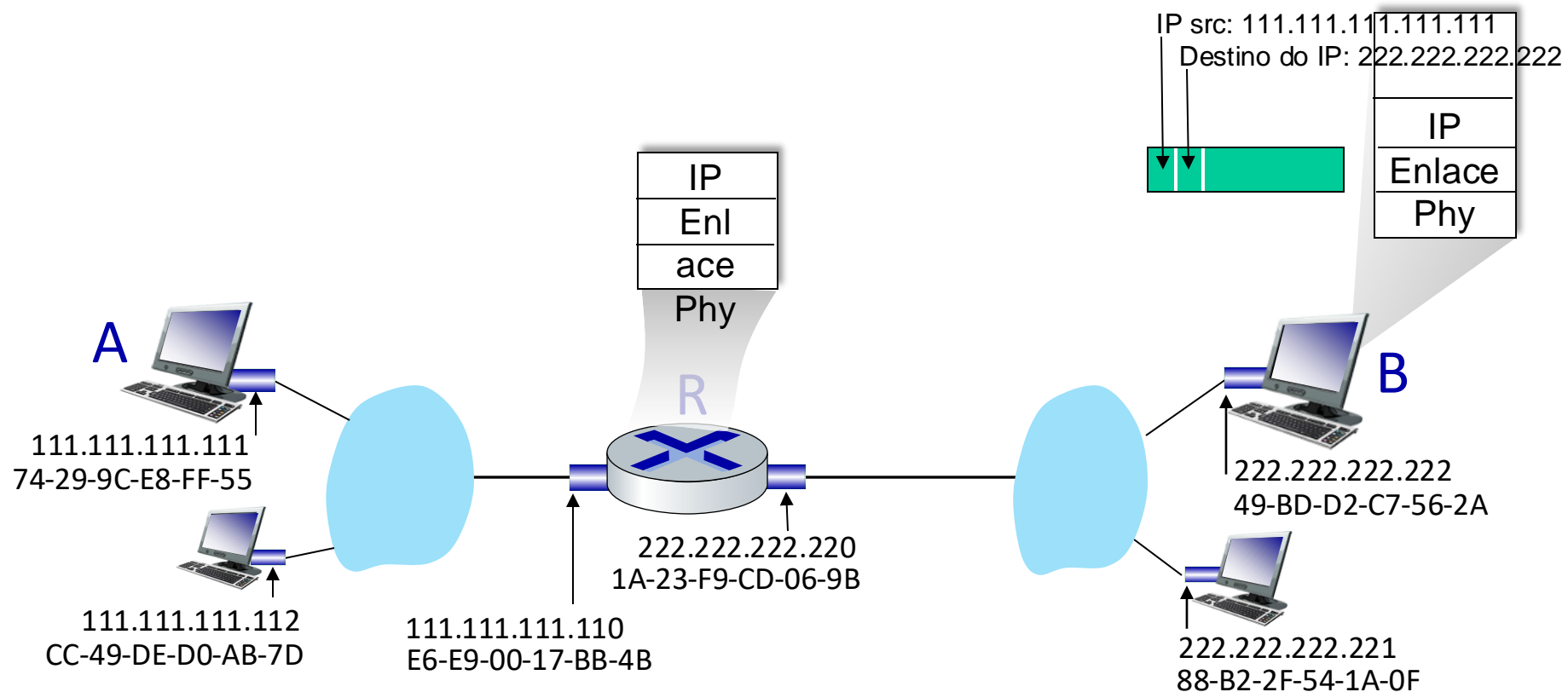
# Roteamento para outra sub-rede: endereçamento

- R determina a interface de saída, passa o datagrama com a fonte IP A e o destino B para a camada de link
- R cria um quadro de camada de enlace contendo um datagrama IP de A para B. Endereço de destino do quadro: Endereço MAC de B
- transmite o quadro da camada de link



# Roteamento para outra sub-rede: endereçamento

- B recebe o quadro, extrai o datagrama IP de destino B
- B passa o datagrama pela pilha de protocolos até o IP



# Camada de link, LANs: roteiro

- introdução
- detecção e correção de erros
- protocolos de acesso múltiplo
- **LANs**
  - endereçamento, ARP
  - **Ethernet**
  - interruptores
  - VLANs
- virtualização de links: MPLS
- rede de data center



- Um dia na vida de uma solicitação da Web

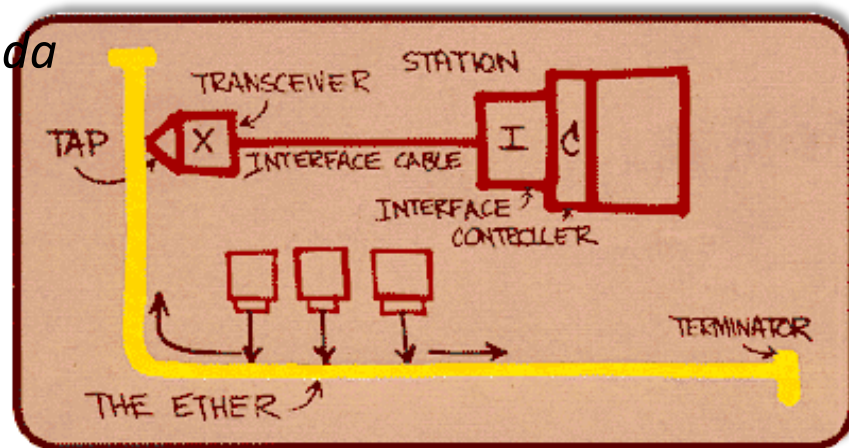


# Ethernet

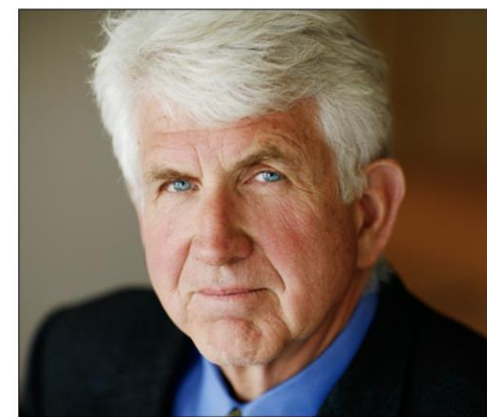
tecnologia LAN com fio "dominante":

- primeira tecnologia LAN amplamente utilizada
- mais simples, barato
- acompanhou a corrida pela velocidade: 10 Mbps - 400 Gbps
- chip único, várias velocidades (por exemplo, Broadcom BCM5761)

*Esboço de Ethernet da Metcalfe*

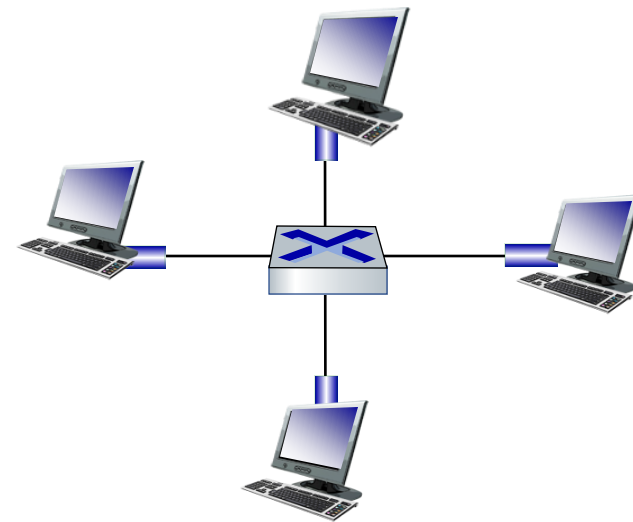


Bob Metcalfe: co-inventor da Ethernet,  
Recebeu o Prêmio Turing da ACM em 2022



# Ethernet: topologia física

- **barramento:** popular até meados dos anos 90
  - todos os nós no mesmo domínio de colisão (podem colidir uns com os outros)
- **chaveamento:** prevalece hoje
  - *switch* ativo de camada de link 2 no centro
  - cada "spoke" executa um protocolo Ethernet (separado) (os nós não colidem uns com os outros)



# Estrutura do quadro Ethernet

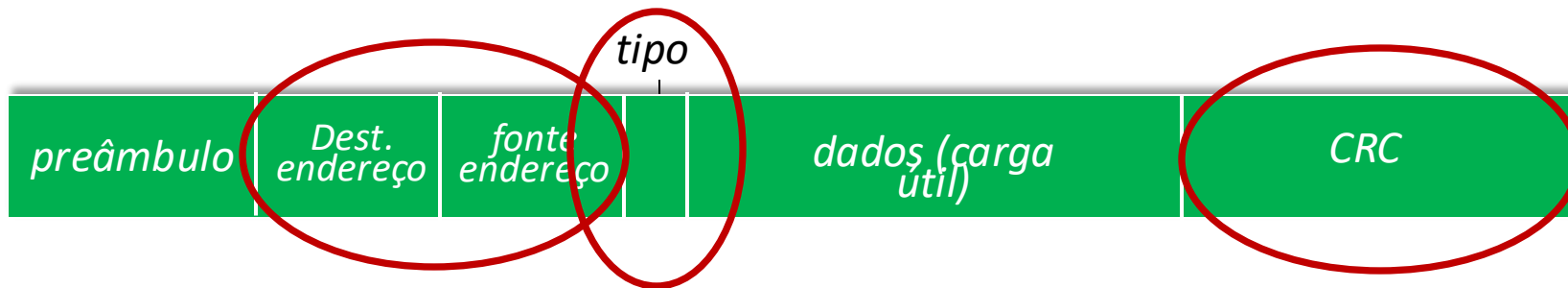
a interface de envio encapsula o datagrama IP (ou outro pacote de protocolo de camada de rede) em um **quadro Ethernet**



## *preâmbulo:*

- usado para sincronizar as taxas de relógio do receptor e do remetente
- 7 bytes de 10101010 seguidos por um byte de 10101011

# Estrutura do quadro Ethernet (mais)



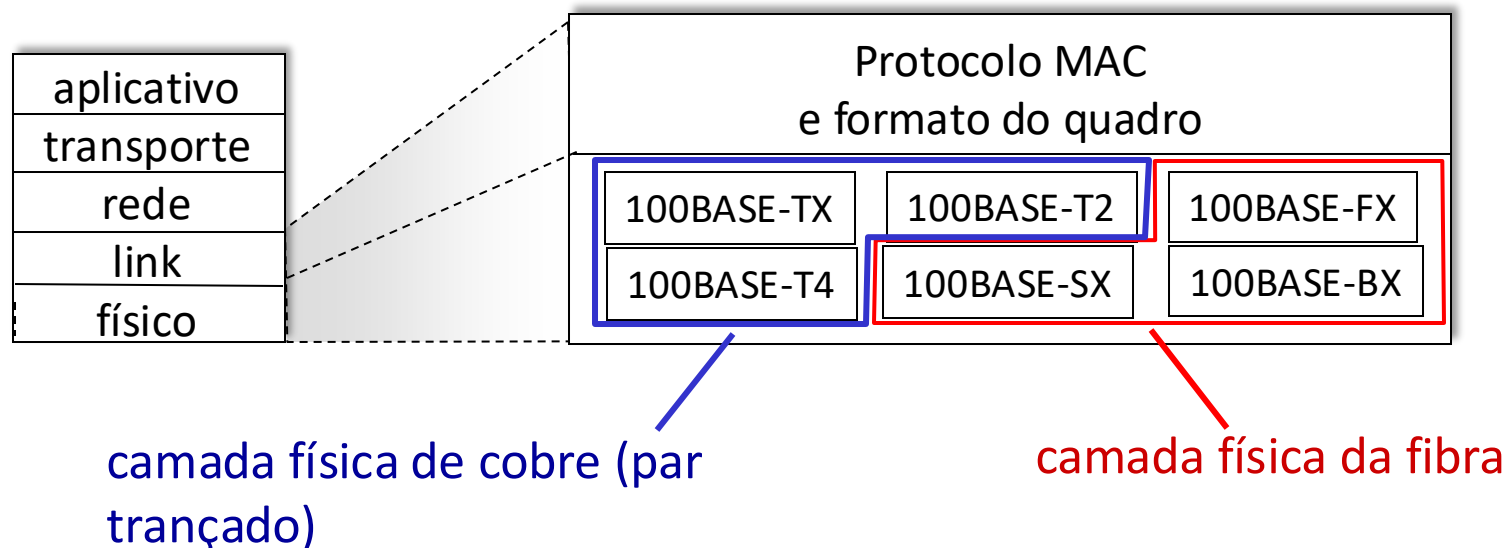
- **endereços:** Endereços MAC de origem e destino de 6 bytes
  - Se o adaptador receber um quadro com endereço de destino correspondente ou com endereço de difusão (por exemplo, pacote ARP), ele passará os dados do quadro para o protocolo da camada de rede
  - Caso contrário, o adaptador descarta o quadro
- **type:** indica o protocolo da camada superior
  - principalmente IP, mas outros são possíveis, por exemplo, Novell IPX, AppleTalk
  - usado para demultiplexar no receptor
- **CRC:** verificação de redundância cíclica no receptor
  - erro detectado: o quadro é descartado

# Ethernet: não confiável, sem conexão

- **sem conexão:** sem handshaking entre as NICs de envio e recebimento
- **não confiável:** a NIC receptora não envia ACKs ou NAKs para a NIC remetente
  - os dados em quadros descartados são recuperados somente se o remetente inicial usar rdt de camada superior (por exemplo, TCP); caso contrário, os dados descartados são perdidos
- Protocolo MAC da Ethernet: **CSMA/CD** sem ranhuras **com backoff binário**

# Padrões Ethernet 802.3: camadas física e de link

- *vários* padrões diferentes de Ethernet
  - protocolo MAC comum e formato de quadro
  - velocidades diferentes: 2 Mbps, ... 100 Mbps, 1Gbps, 10 Gbps, 40 Gbps, 80 Gbps
    - diferentes mídias de camada física: fibra, cabo



# Camada de link, LANs: roteiro

- introdução
- detecção e correção de erros
- protocolos de acesso múltiplo
- **LANs**
  - endereçamento, ARP
  - Ethernet
  - **interruptores**
  - VLANs
- virtualização de links: MPLS
- rede de data center



- Um dia na vida de uma solicitação da Web

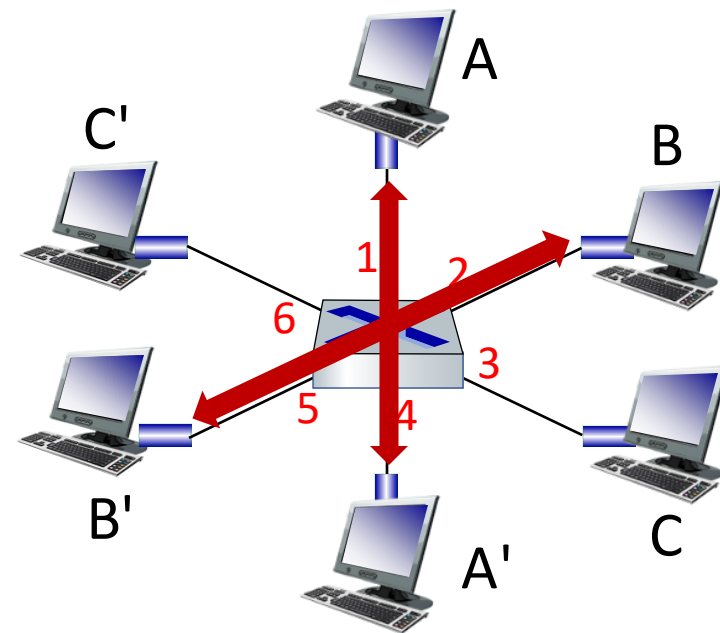
# Switch Ethernet

- O switch é um dispositivo **de camada de link**: assume uma função *ativa*
  - armazenar e encaminhar quadros Ethernet (ou outro tipo de quadro)
  - examinar o endereço MAC do quadro de entrada, encaminhar *seletivamente* o quadro para um ou mais links de saída quando o quadro tiver que ser encaminhado no segmento, usar CSMA/CD para acessar o segmento
- **transparente**: os hosts *não sabem* da presença de switches
- **plug-and-play, autoaprendizagem**
  - os switches não precisam ser configurados



# Switch: várias transmissões simultâneas

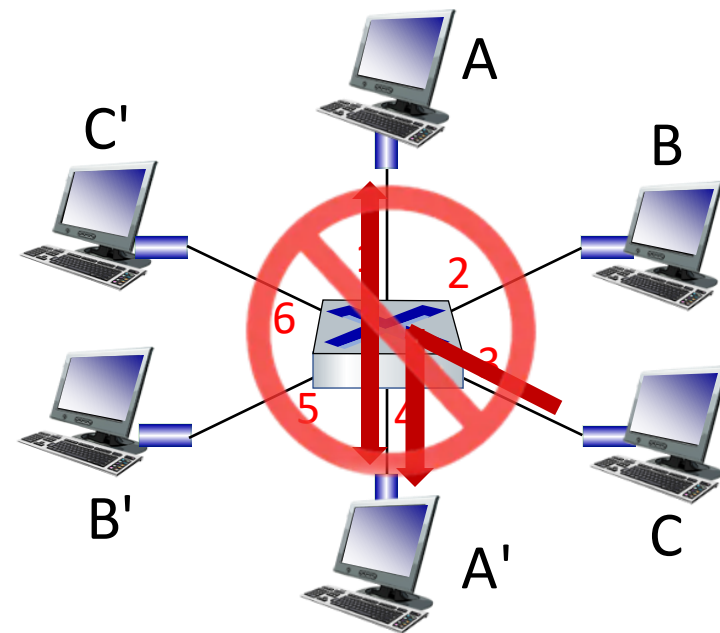
- Os hosts têm conexão dedicada e direta com o switch
- pacotes de buffer de switches
- Protocolo Ethernet usado em *cada* link de entrada, portanto:
  - sem colisões; full duplex
  - cada link é seu próprio domínio de colisão
- **comutação:** A-to-A' e B-to-B' podem transmitir simultaneamente, sem colisões



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

# Switch: várias transmissões simultâneas

- Os hosts têm conexão dedicada e direta com o switch
- pacotes de buffer de switches
- Protocolo Ethernet usado em *cada* link de entrada, portanto:
  - sem colisões; full duplex
  - cada link é seu próprio domínio de colisão
- **comutação:** A-to-A' e B-to-B' podem transmitir simultaneamente, sem colisões
  - mas A para A' e C para A' *não* podem ocorrer simultaneamente



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

# Tabela de encaminhamento do switch

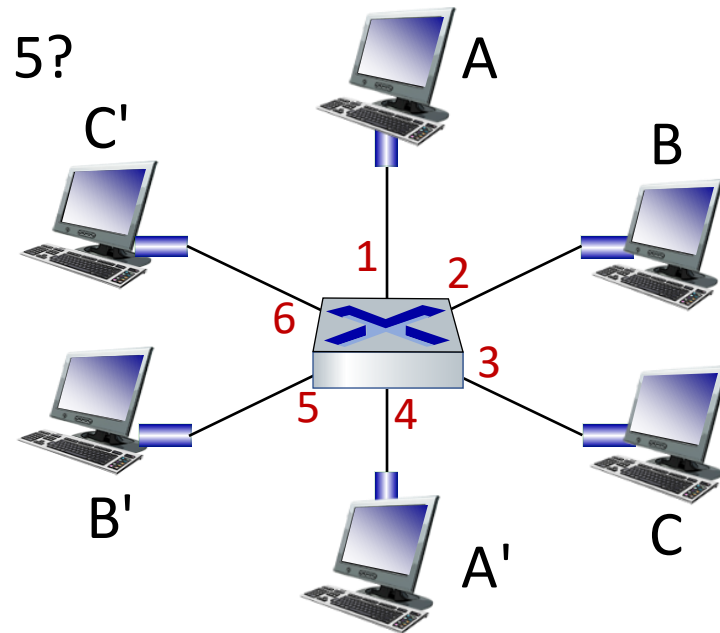
P: Como o switch sabe que A' pode ser acessado pela interface 4 e B' pode ser acessado pela interface 5?

R: Cada switch tem uma **tabela de switch**, cada entrada:

- (endereço MAC do host, interface para acessar o host, registro de data e hora)
- parece uma tabela de roteamento!

P: Como as entradas são criadas e mantidas na tabela de comutação?

- Algo como um protocolo de roteamento?



# Interruptor: autoaprendizagem

- O switch *descobre* quais hosts podem ser acessados por meio de quais interfaces
  - Quando o quadro é recebido, o switch "aprende" a localização do remetente: segmento de LAN de entrada
  - registra o par remetente/localização na tabela de comutação

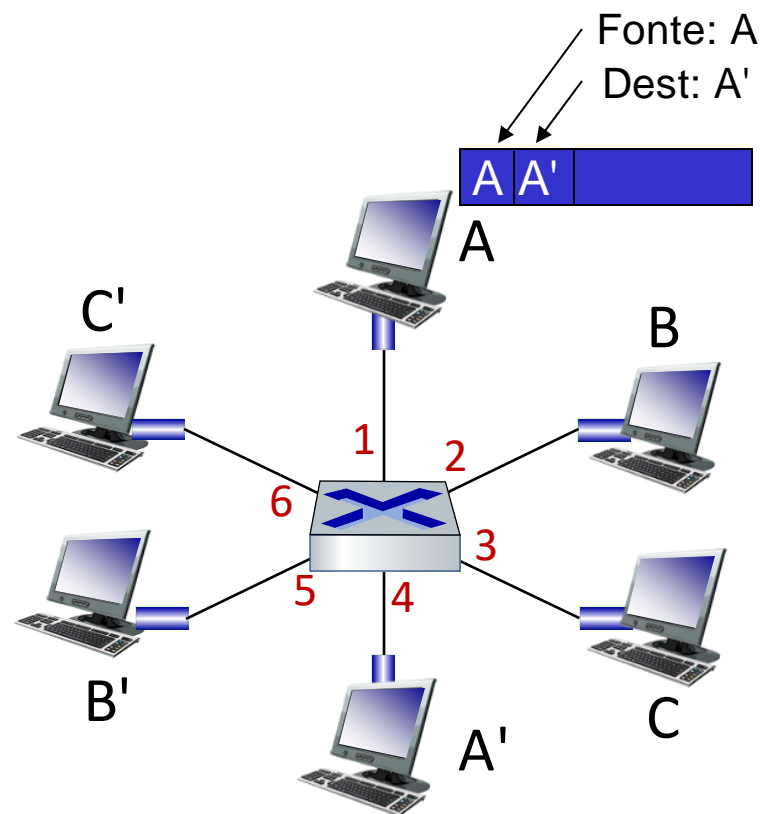


Tabela de interruptores (inicialmente vazio)		
Endereço MAC	TTL da interface	
A	1	60

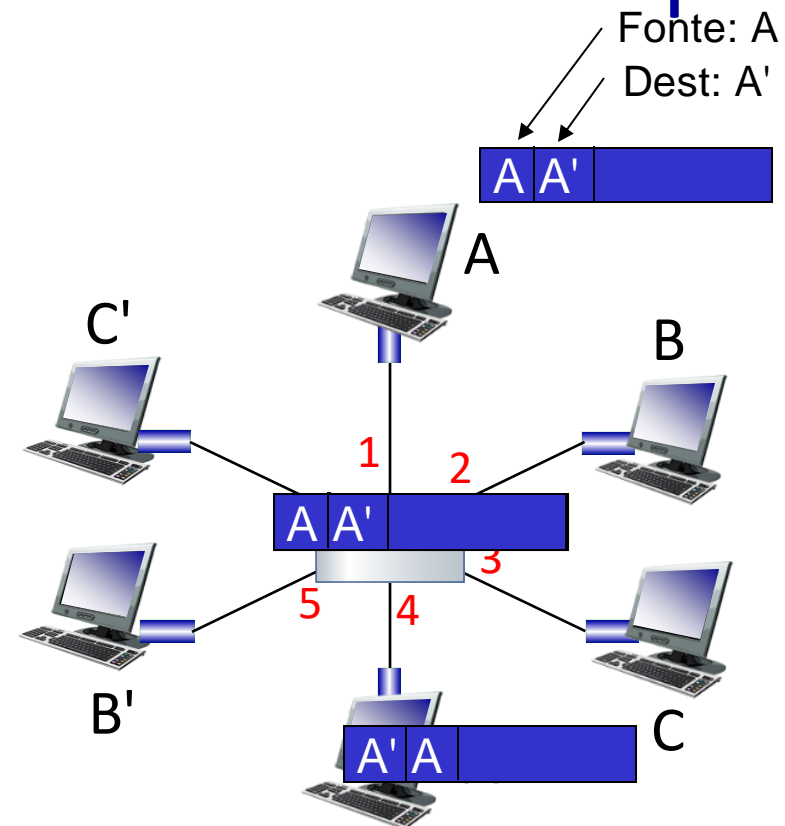
# Switch: filtragem/encaminhamento de quadros

quando o quadro é recebido no switch:

1. registrar o link de entrada, endereço MAC do host de envio
2. indexar a tabela do switch usando o endereço de destino MAC
3. **se** for encontrada uma entrada para o destino  
    **then** {  
        **se** o destino estiver no segmento de onde o quadro chegou  
            **então**, descartar o quadro  
        **senão** encaminhar o quadro na interface indicada pela entrada  
    }  
    **else** flood /\* encaminhar em todas as interfaces, exceto na interface de chegada \*/

# Autoaprendizagem, encaminhamento: exemplo

- destino do quadro, A', local desconhecido: **inundação**
- destino Um local conhecido: **enviar seletivamente em apenas um link**

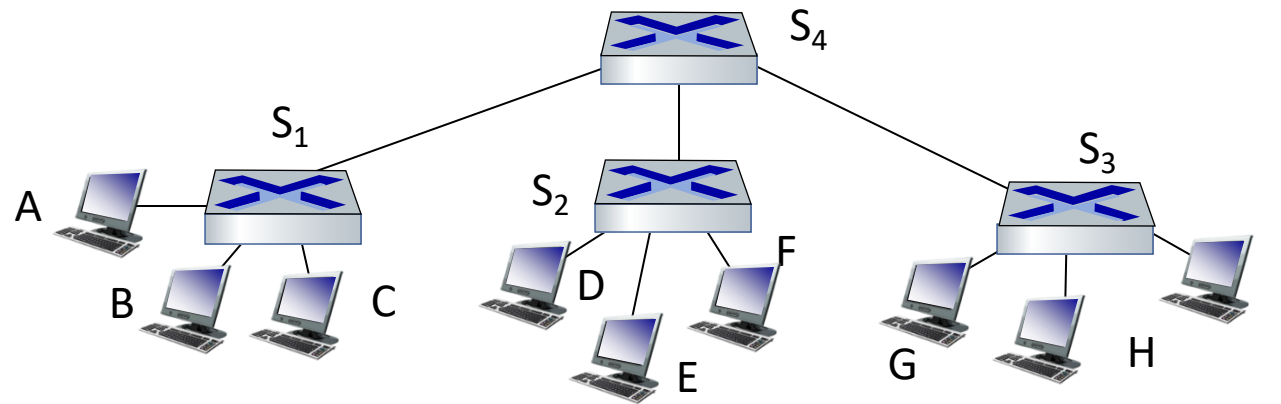


Endereço	MAC	TTL da interface
A	1	60
A'	4	60

*tabela de interruptores (inicialmente vazio)*

# Chaves de interconexão

podem ser conectados juntos:

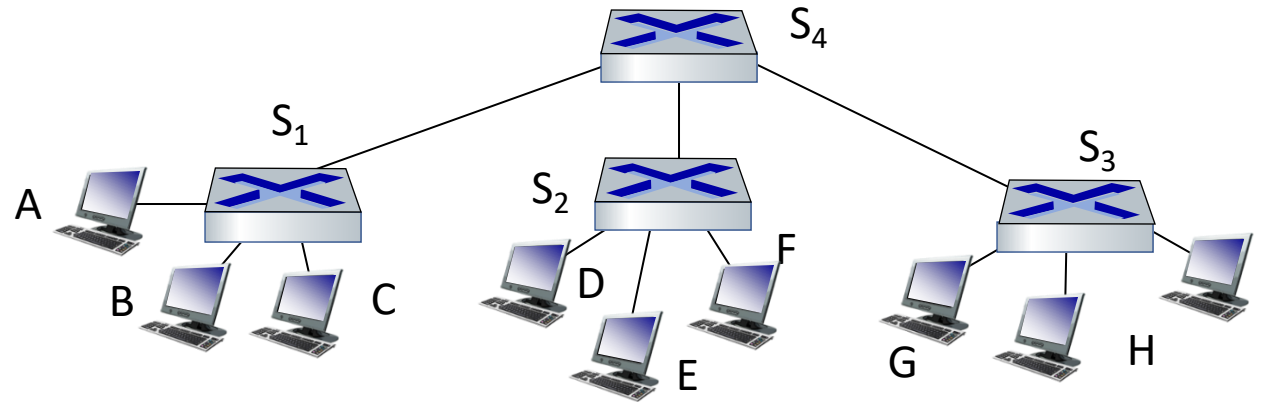


P: enviando de A para G - como S<sub>1</sub> sabe que deve encaminhar o quadro destinado a G por meio de S<sub>4</sub> e S<sub>3</sub>?

- R: autoaprendizado! (funciona exatamente da mesma forma que no caso do interruptor único!)

# Exemplo de comutador múltiplo de autoaprendizagem

Suponha que C envie um quadro para I e que I responda a C



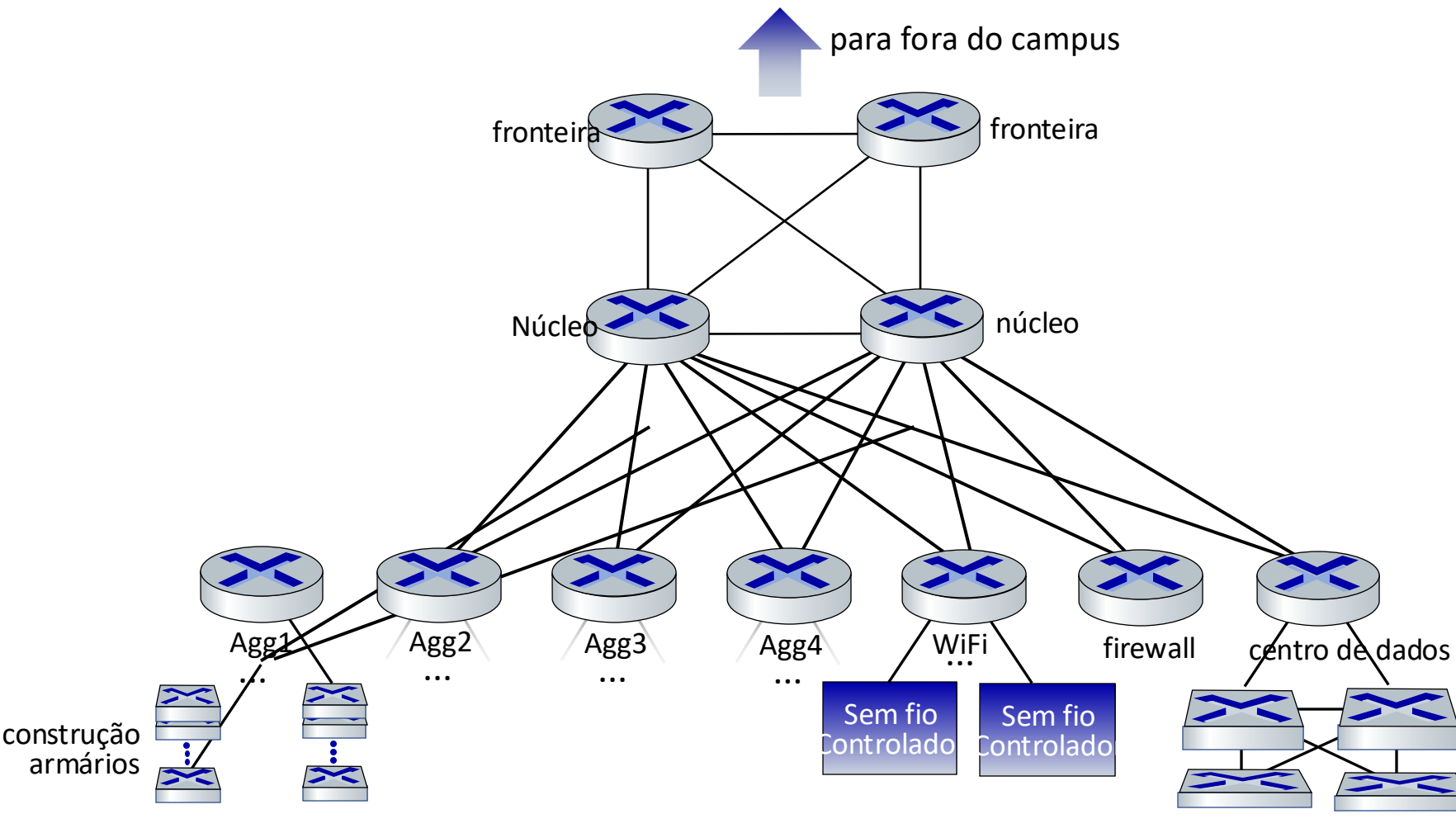
P: mostrar tabelas de switch e encaminhamento de pacotes em S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, S<sub>4</sub>



# Rede do campus da UMass - Detalhes

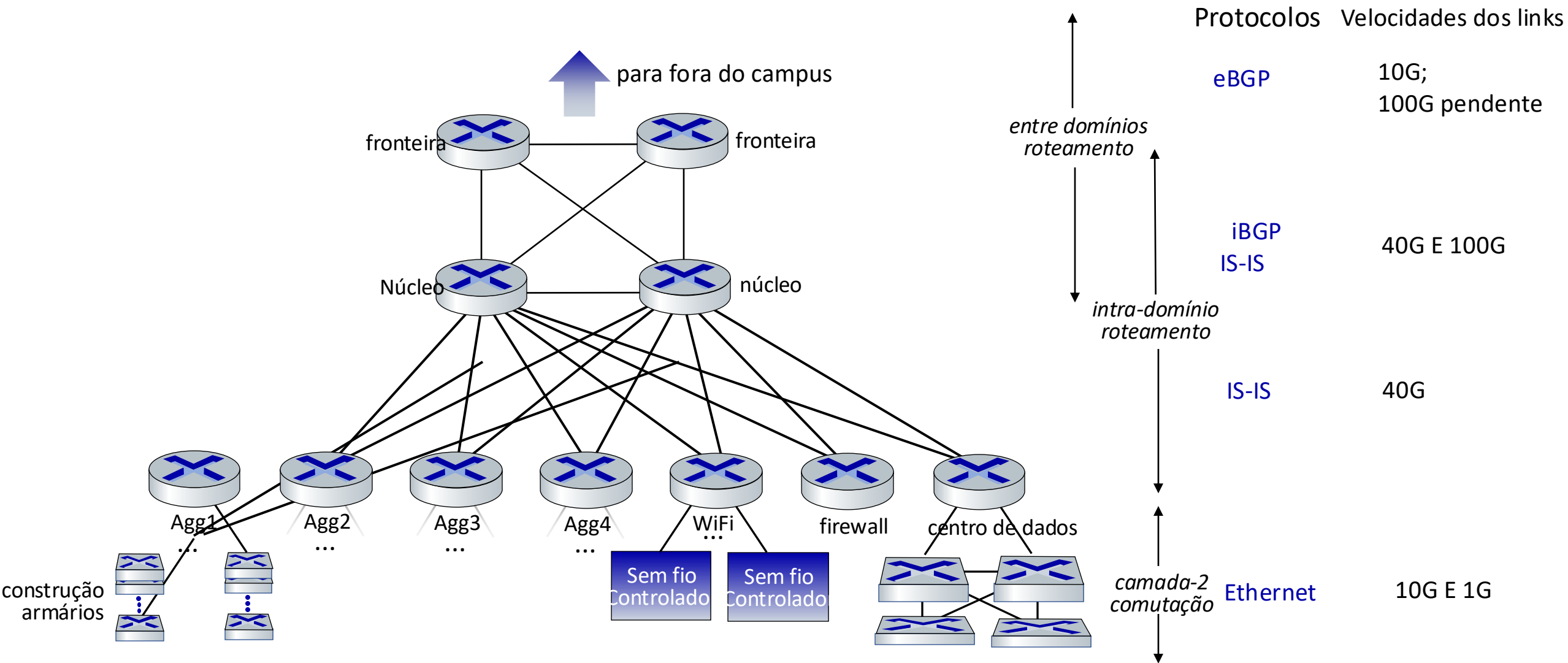
## Rede UMass:

- 4 firewalls
- 10 roteadores
- Mais de 2000 switches de rede
- 6000 pontos de acesso sem fio
- 30000 conectores de rede com fio ativos
- 55.000 dispositivos sem fio ativos para usuários finais



... tudo construído, operado e mantido por aproximadamente 15 pessoas

# Rede do campus da UMass - Detalhes



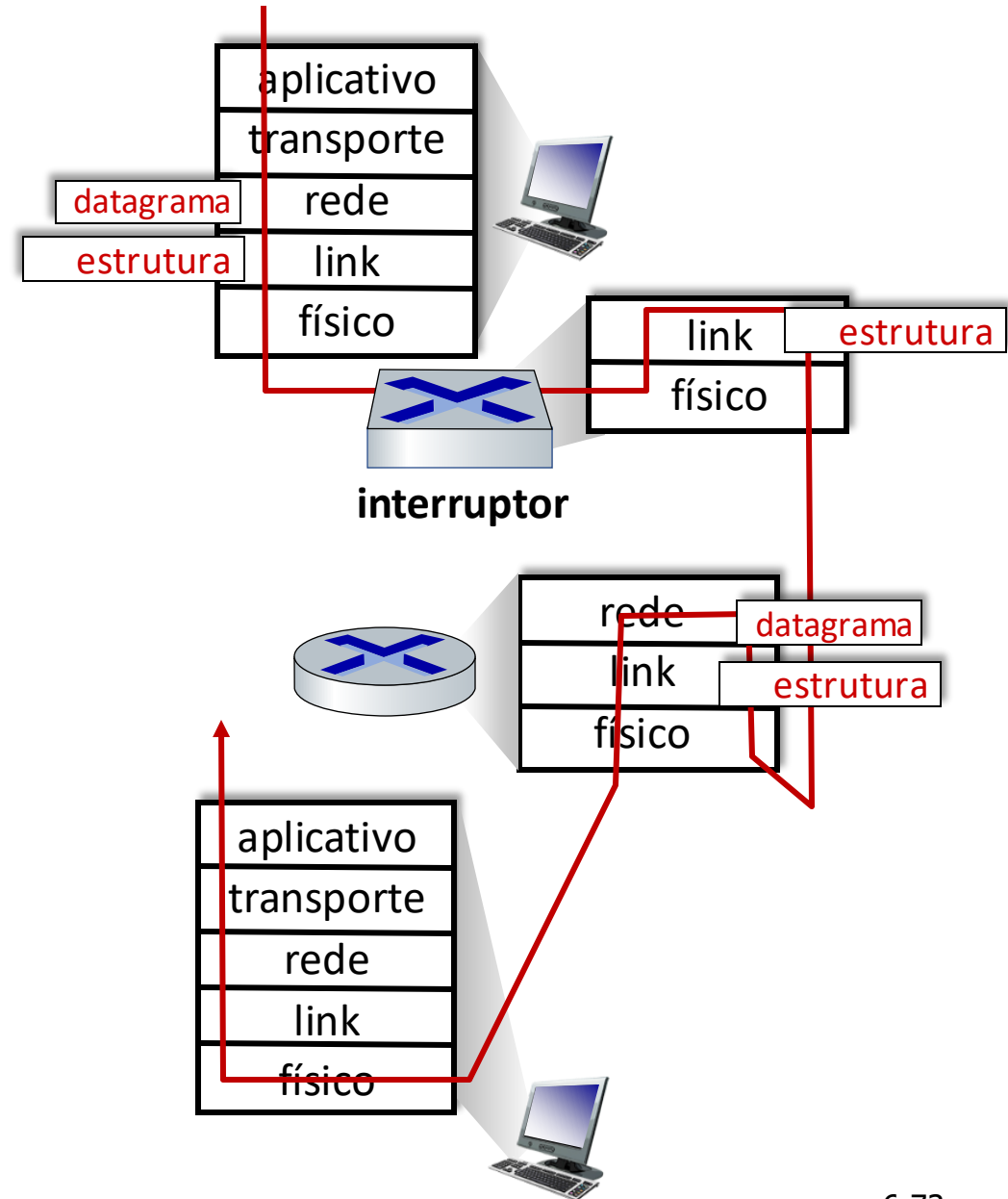
# Switches vs. roteadores

ambos são de armazenamento e encaminhamento:

- **roteadores**: dispositivos da camada de rede (examinam os cabeçalhos da camada de rede)
- **switches**: dispositivos da camada de link (examinam os cabeçalhos da camada de link)

ambos têm tabelas de encaminhamento:

- **roteadores**: calculam tabelas usando algoritmos de roteamento, endereços IP
- **Switches**: aprender a tabela de encaminhamento usando flooding, aprendizagem, endereços MAC



# Camada de enlace, LANs: roteiro

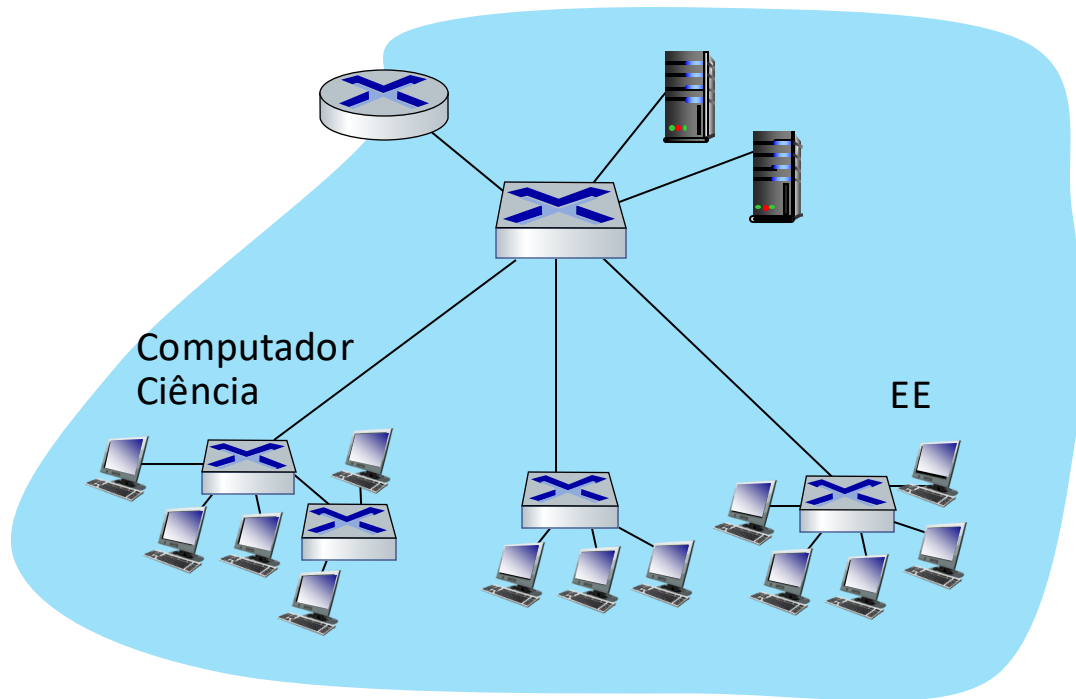
- introdução
- detecção e correção de erros
- protocolos de acesso múltiplo
- **LANs**
  - endereçamento, ARP
  - Ethernet
  - interruptores
  - **VLANs**
- virtualização de links: MPLS
- rede de data center



- Um dia na vida de uma solicitação da Web

# LANs virtuais (VLANs): motivação

*P:* O que acontece quando o tamanho da LAN aumenta e os usuários mudam o ponto de conexão?

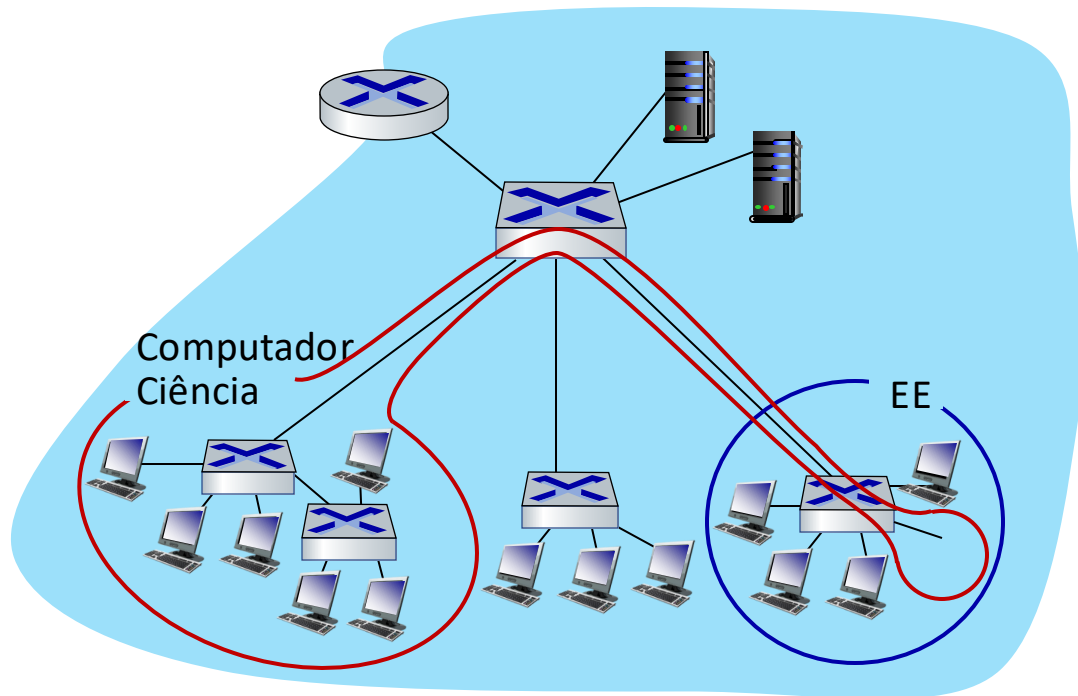


domínio de transmissão único:

- *Dimensionamento:* todo o tráfego de difusão da camada 2 (ARP, DHCP, MAC desconhecido) deve atravessar toda a LAN
- questões de eficiência, segurança e privacidade

# LANs virtuais (VLANs): motivação

*P:* O que acontece quando o tamanho da LAN aumenta e os usuários mudam o ponto de conexão?



domínio de transmissão único:

- *escalonamento*: todo o tráfego de difusão de camada 2 (ARP, DHCP, MAC desconhecido) deve atravessar toda a LAN
- eficiência, segurança, privacidade, problemas de eficiência

questões administrativas:

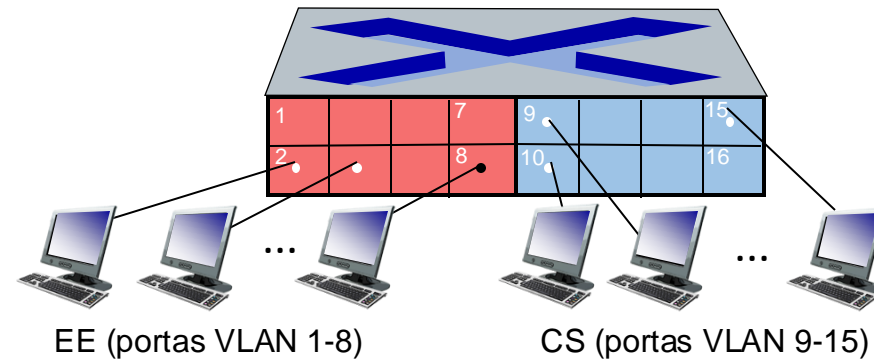
- O usuário CS muda o escritório para EE - *fisicamente* conectado ao switch EE, mas deseja permanecer *logicamente* conectado ao switch CS

# VLANs baseadas em portas

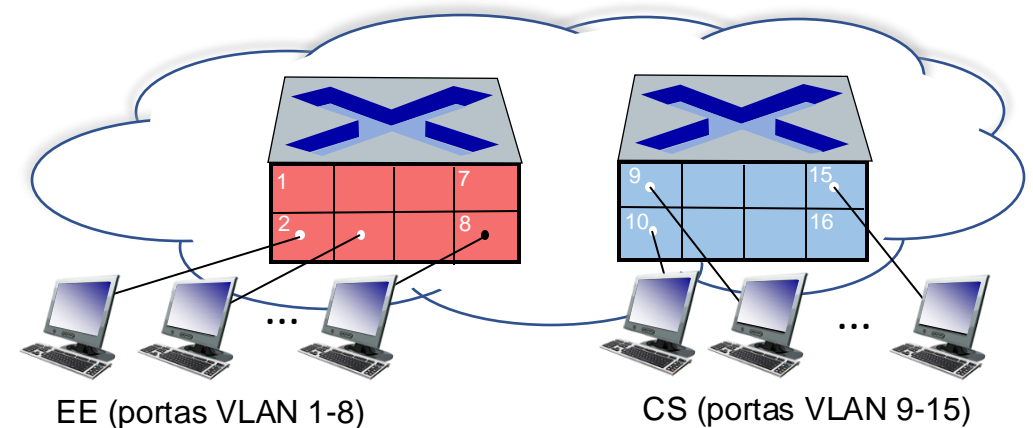
## Rede local virtual (VLAN)

O(s) switch(s) com suporte a recursos de VLAN pode(m) ser configurado(s) para definir várias LANS *virtuais* em uma única infraestrutura de LAN física.

**VLAN baseada em porta:** portas de switch agrupadas (pelo software de gerenciamento de switch) de modo que *um único* switch físico .....

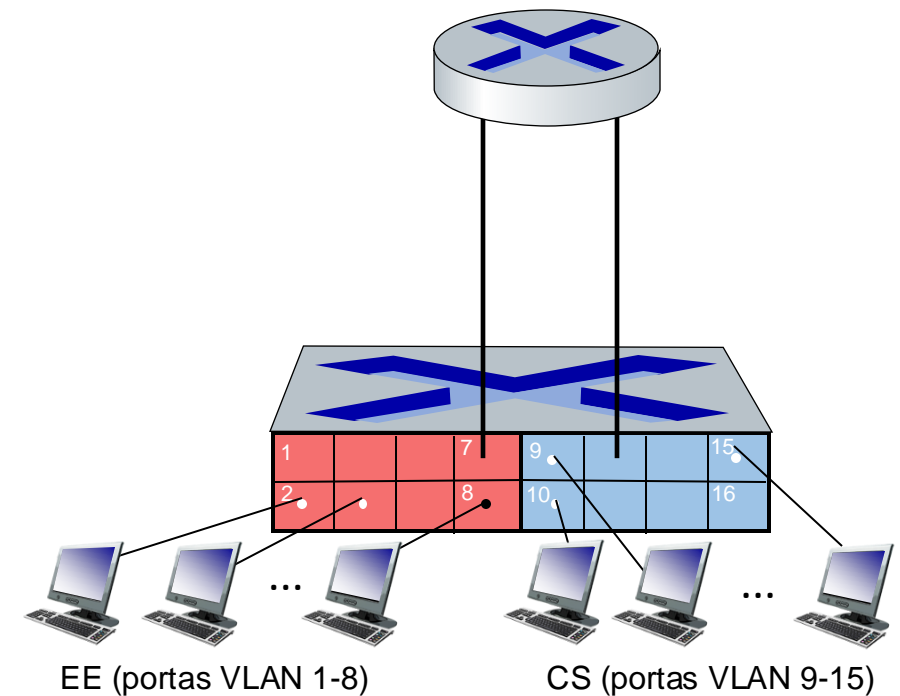


... opera como **vários** switches virtuais



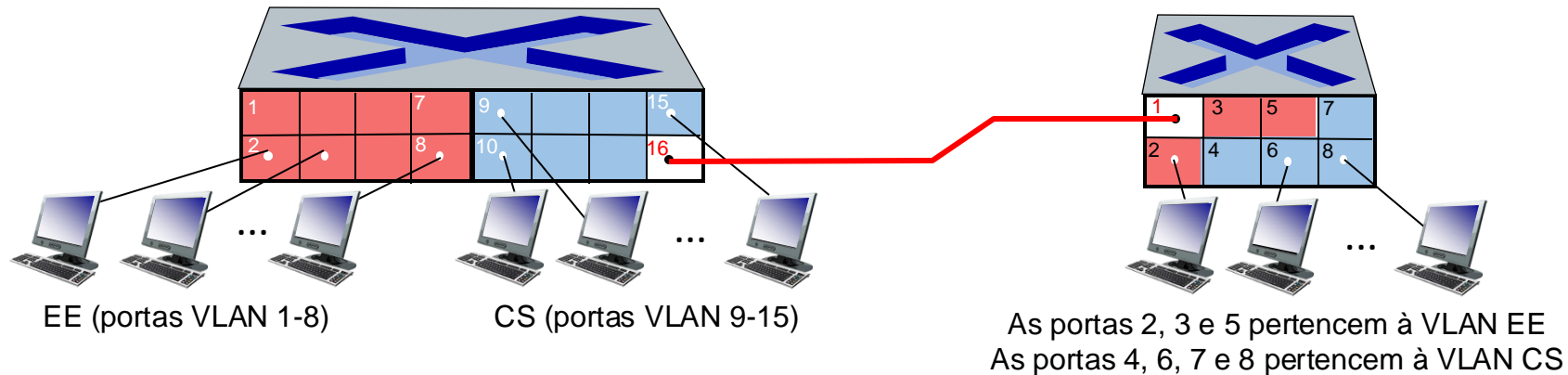
# VLANs baseadas em portas

- **isolamento de tráfego:** os quadros de/para as portas 1-8 só podem alcançar as portas 1-8
  - também pode definir a VLAN com base nos endereços MAC dos pontos de extremidade, em vez da porta do switch
- **associação dinâmica:** as portas podem ser atribuídas dinamicamente entre VLANs
- **encaminhamento entre VLANs:** feito por meio de roteamento (assim como em switches separados)
  - na prática, os fornecedores vendem switches e roteadores combinados





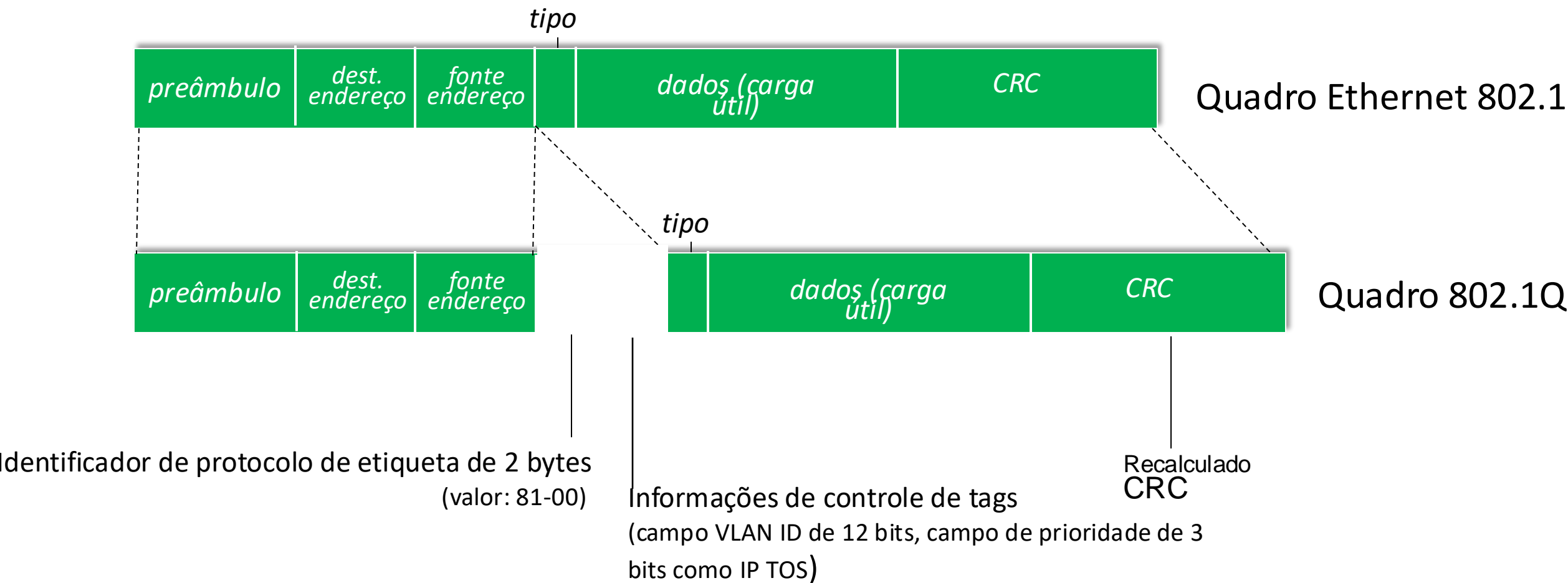
# VLANs que abrangem vários switches



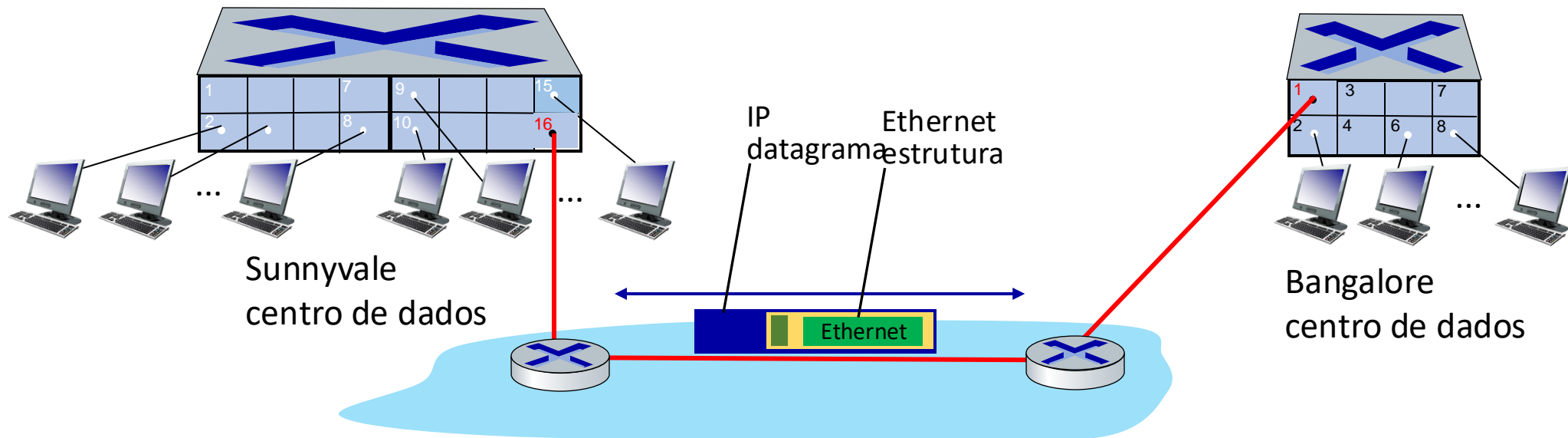
**porta tronco:** transporta quadros entre VLANs definidas em vários switches físicos

- os quadros encaminhados dentro da VLAN entre switches não podem ser quadros vanilla 802.1 (devem conter informações de VLAN ID)
- O protocolo 802.1q adiciona/remove campos de cabeçalho adicionais para quadros encaminhados entre portas tronco

# Formato de quadro de VLAN 802.1Q



# EVPN: VPNs Ethernet (também conhecidas como VXLANs)



Switches Ethernet de camada 2 conectados *logicamente* uns aos outros (por exemplo, usando IP como *subcamada*)

- Quadros Ethernet transportados *em* datagramas IP entre sites
- "esquema *de tunelamento* para *sobrepôr redes de Camada 2 sobre redes de Camada 3* ... é executado sobre a infraestrutura de rede existente e fornece um meio de "esticar" uma rede de Camada 2." [RFC 7348]

# Camada de link, LANs: roteiro

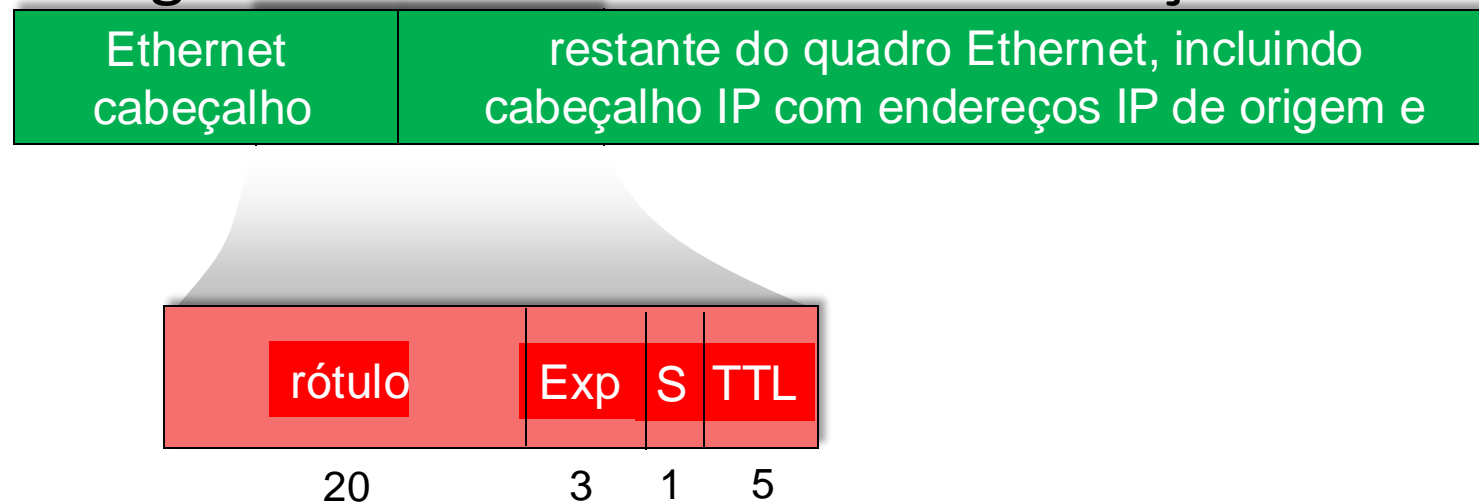
- introdução
- detecção e correção de erros
- protocolos de acesso múltiplo
- LANs
  - endereçamento, ARP
  - Ethernet
  - interruptores
  - VLANs
- **virtualização de links: MPLS**
- rede de data center



- Um dia na vida de uma solicitação da Web

# Comutação de rótulos multiprotocolo (MPLS)

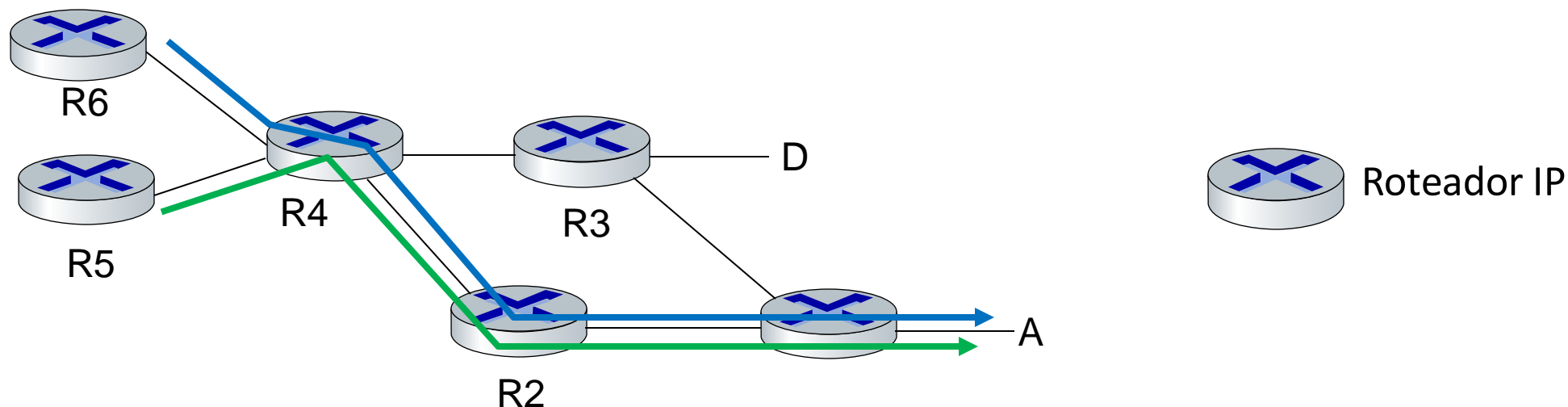
- **Objetivo:** encaminhamento de IP de alta velocidade entre a rede de roteadores compatíveis com MPLS, usando rótulo de comprimento fixo (em vez de correspondência de prefixo mais curto)
  - pesquisa mais rápida usando identificador de comprimento fixo
  - emprestando ideias da abordagem de Circuito Virtual (VC)
  - mas o datagrama IP ainda mantém o endereço IP!



# Roteadores compatíveis com MPLS

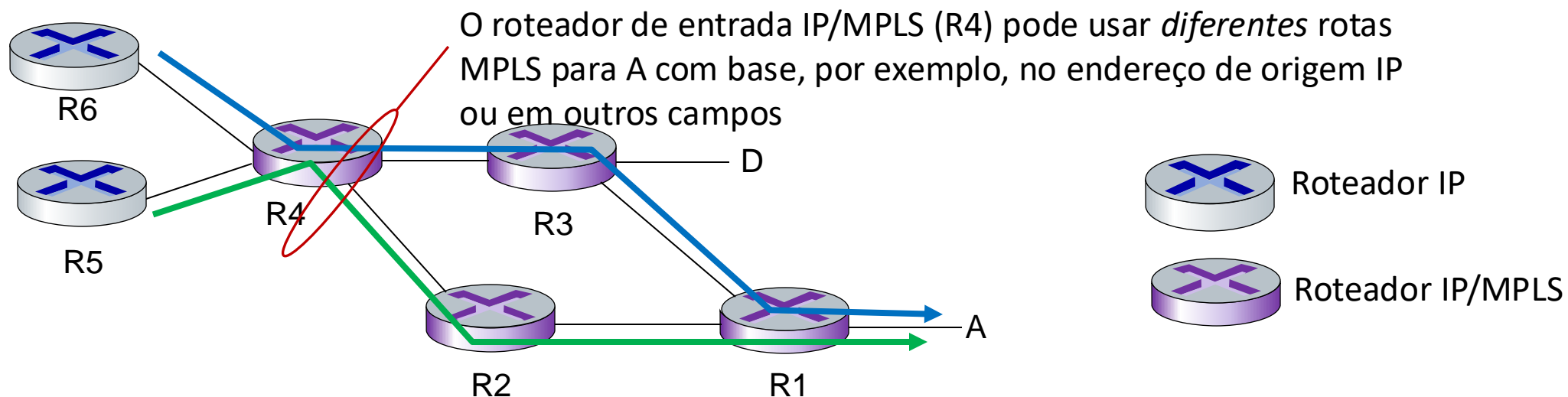
- também conhecido como roteador comutado por rótulo
- encaminhar pacotes para a interface de saída com base apenas no valor do rótulo (*não inspecionar o endereço IP*)
  - Tabela de encaminhamento MPLS diferente das tabelas de encaminhamento IP
- *flexibilidade*: As decisões de encaminhamento do MPLS podem ser *diferentes* das do IP
  - usar endereços de destino e de origem para rotear fluxos para o mesmo destino de forma diferente (engenharia de tráfego)
  - redirecionar fluxos rapidamente se o link falhar: caminhos de backup pré-computados

# Caminhos MPLS versus IP



- **Roteamento IP:** caminho para o destino determinado apenas pelo endereço de destino

# Caminhos MPLS versus IP

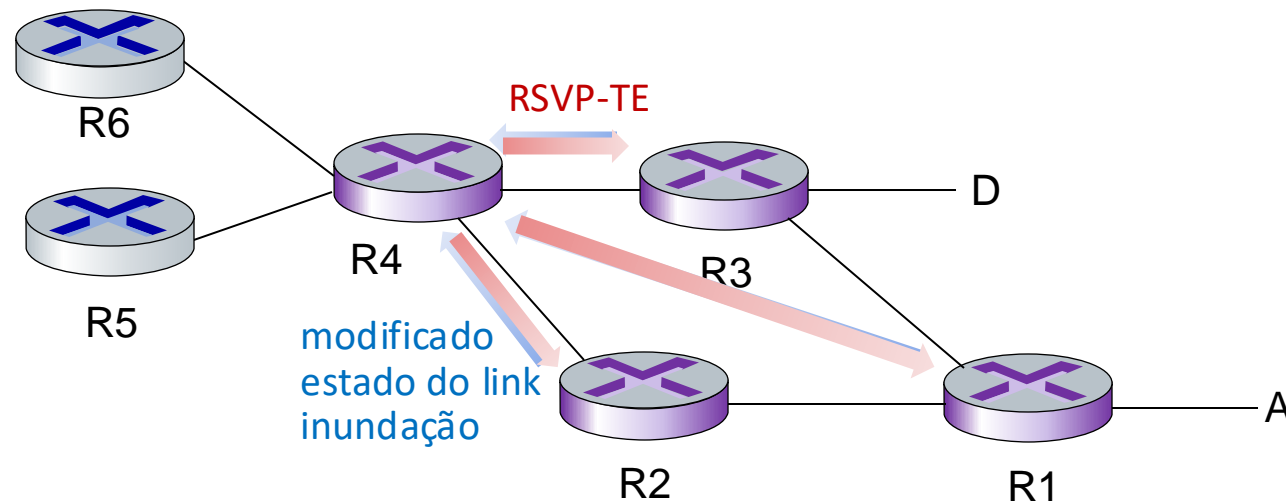


- **Roteamento IP:** caminho para o destino determinado apenas pelo endereço de destino
- **Roteamento MPLS:** o caminho para o destino pode ser baseado no endereço de origem e de destino
  - sabor do encaminhamento generalizado (MPLS 10 anos antes)
  - *redirecionamento rápido:* pré-calcular rotas de backup em caso de falha de link

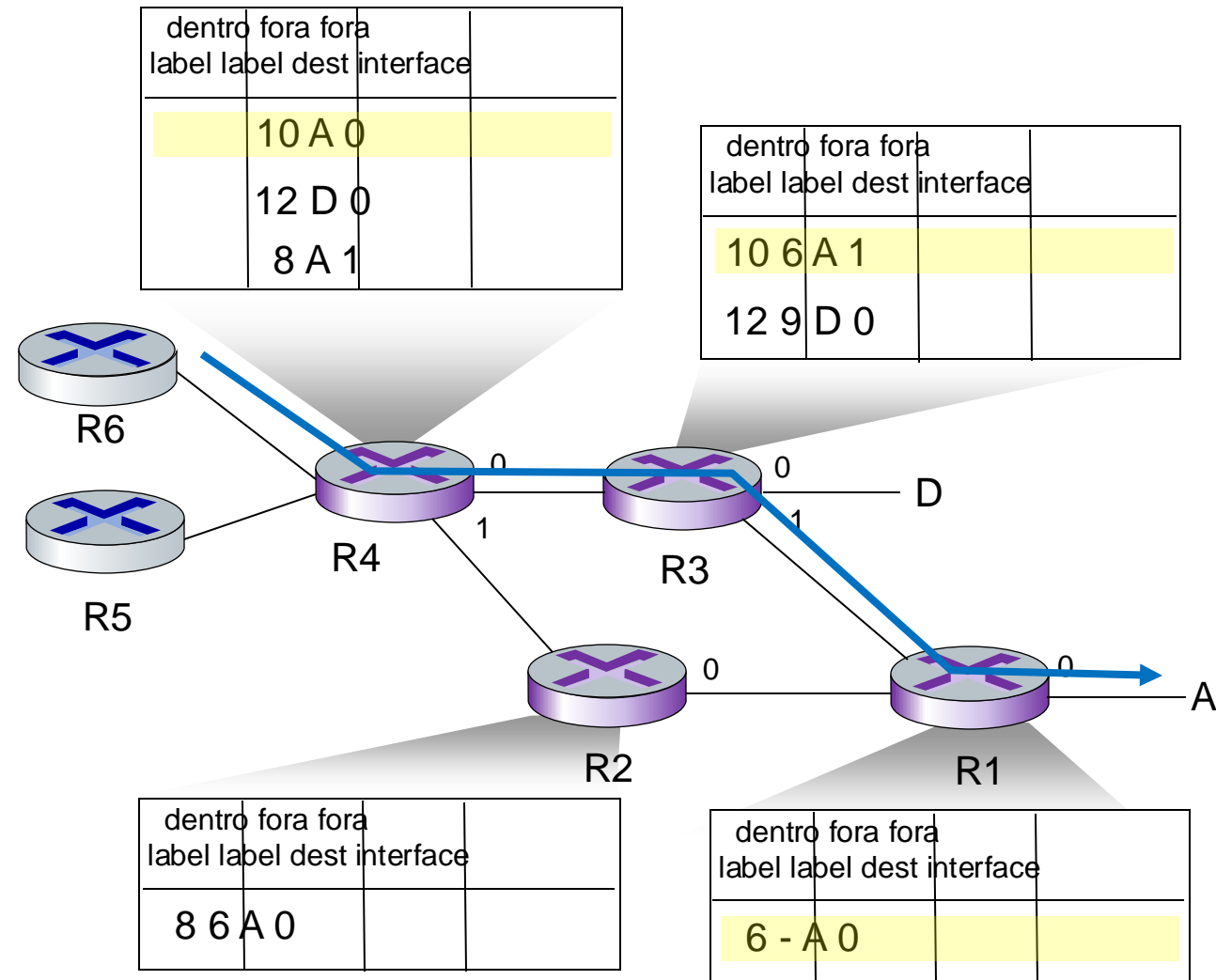


# Sinalização MPLS

- modificar os protocolos de inundação de estado de link OSPF e IS-IS para transportar informações usadas pelo roteamento MPLS:
  - Por exemplo, largura de banda do link, quantidade de largura de banda "reservada" do link
- O roteador MPLS de entrada usa o protocolo de sinalização RSVP-TE para configurar o encaminhamento MPLS nos roteadores downstream



# Tabelas de encaminhamento MPLS



# Camada de link, LANs: roteiro

- introdução
- detecção e correção de erros
- protocolos de acesso múltiplo
- LANs
  - endereçamento, ARP
  - Ethernet
  - interruptores
  - VLANs
- virtualização de links: MPLS
- rede de data center



- Um dia na vida de uma solicitação da Web

# Redes de data center

De 10 a 100 milhares de hosts, muitas vezes intimamente acoplados e próximos uns dos outros:

- comércio eletrônico (por exemplo, Amazon)
- servidores de conteúdo (por exemplo, YouTube, Akamai, Apple, Microsoft)
- mecanismos de pesquisa, mineração de dados (por exemplo, Google)

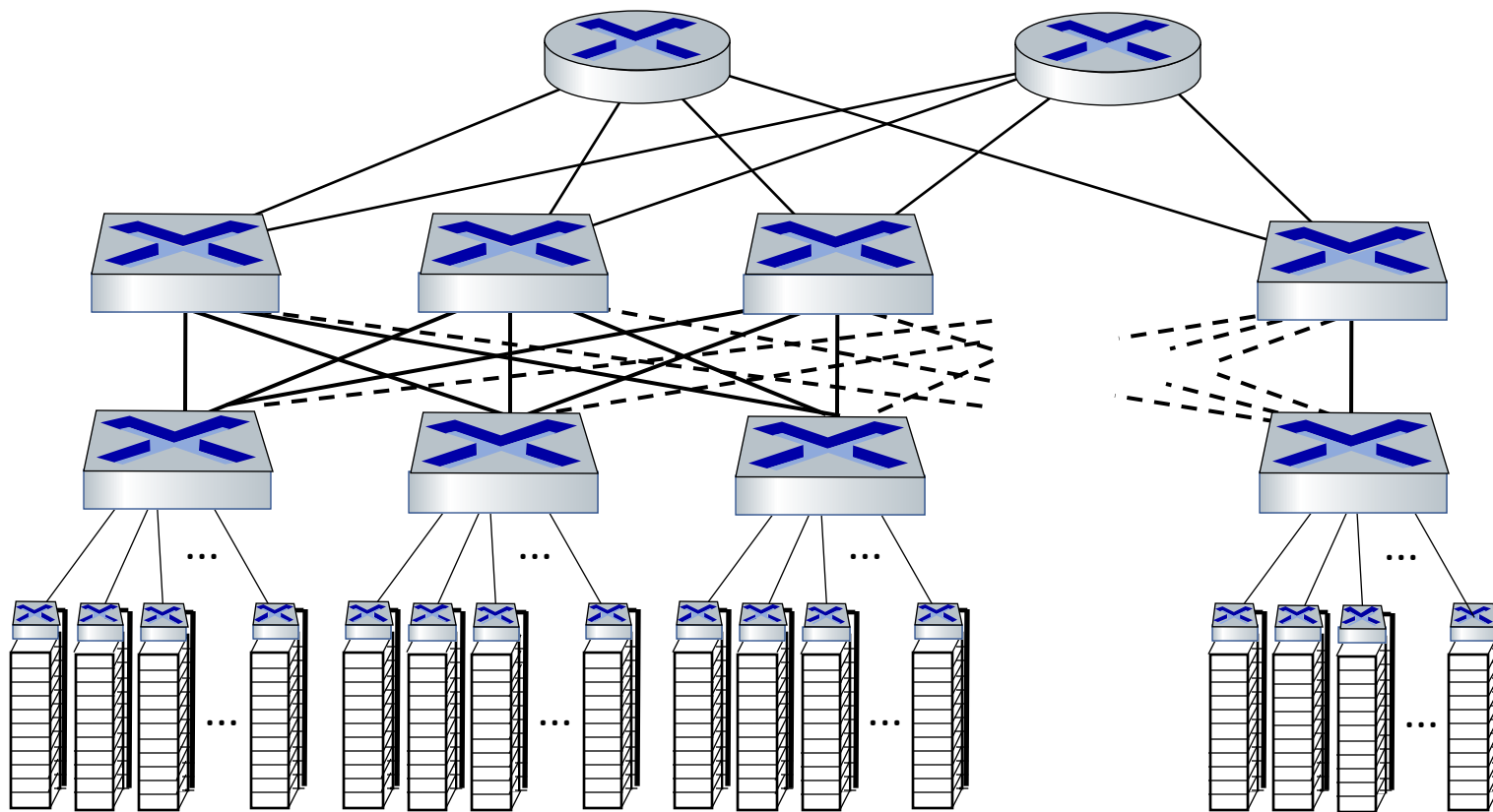
desafios:

- vários aplicativos, cada um atendendo a um grande número de clientes
- confiabilidade
- gerenciamento/equilíbrio de carga, evitando processamento, rede, gargalos de dados



Dentro de um contêiner de 40 pés da Microsoft, o data center de Chicago

# Redes de data center: elementos de rede



## Roteadores de borda

- conexões fora do data center

## Switches de nível 1

- conectando-se a ~16 T-2s abaixo

## Switches de nível 2

- conexão com ~16 TORs abaixo

## Chave do topo do rack (TOR)

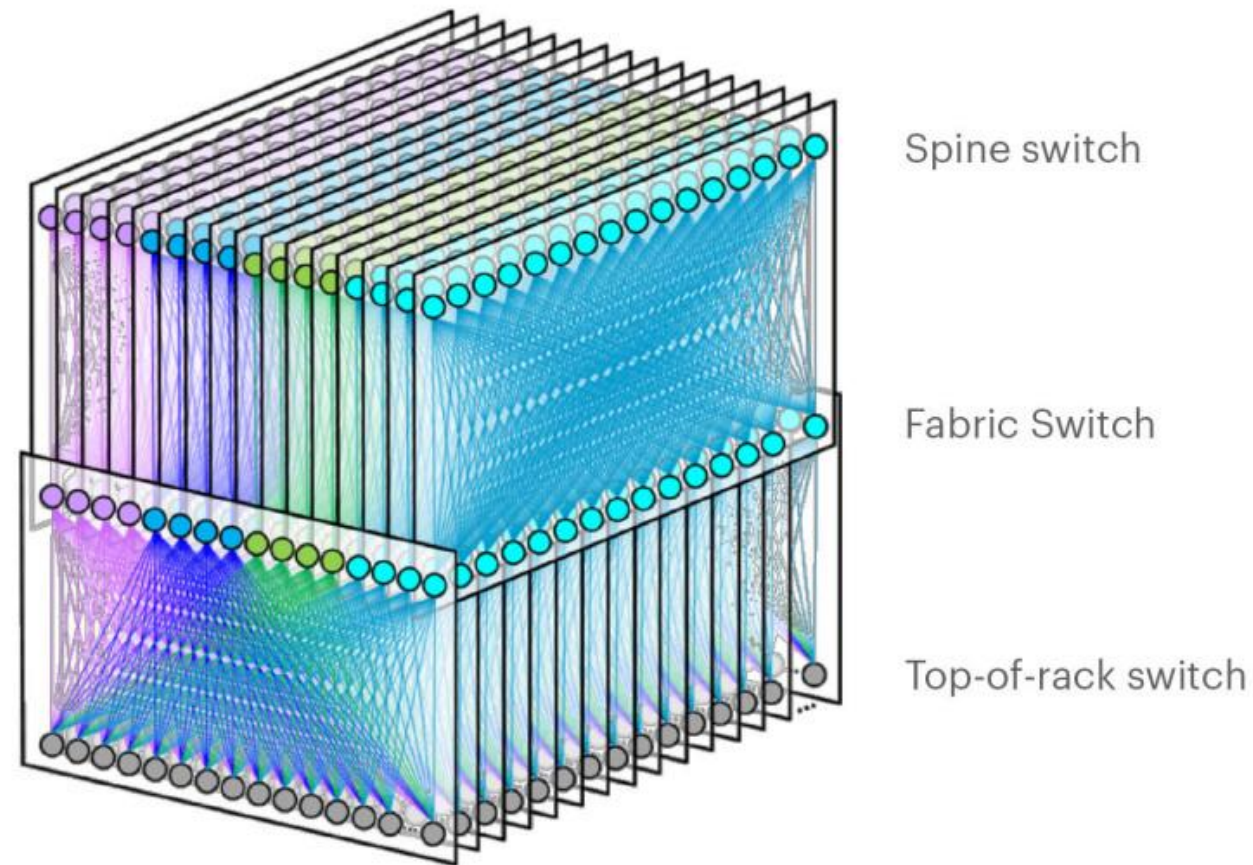
- um por rack
- Ethernet 100G-400G para blades

## Racks para servidores

- 20- 40 blades de servidor: hosts

# Redes de data center: elementos de rede

Topologia de rede do data center F16 do Facebook:

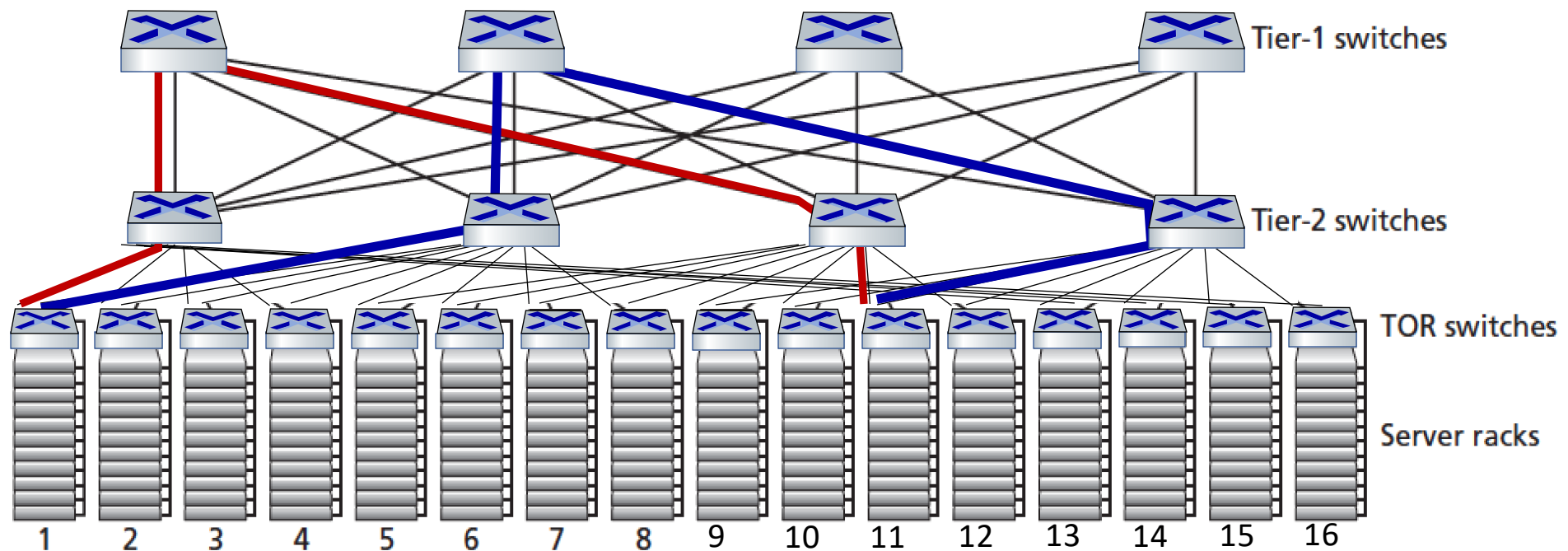


<https://engineering.fb.com/data-center-engineering/f16-minipack/> (publicado em 3/2019)



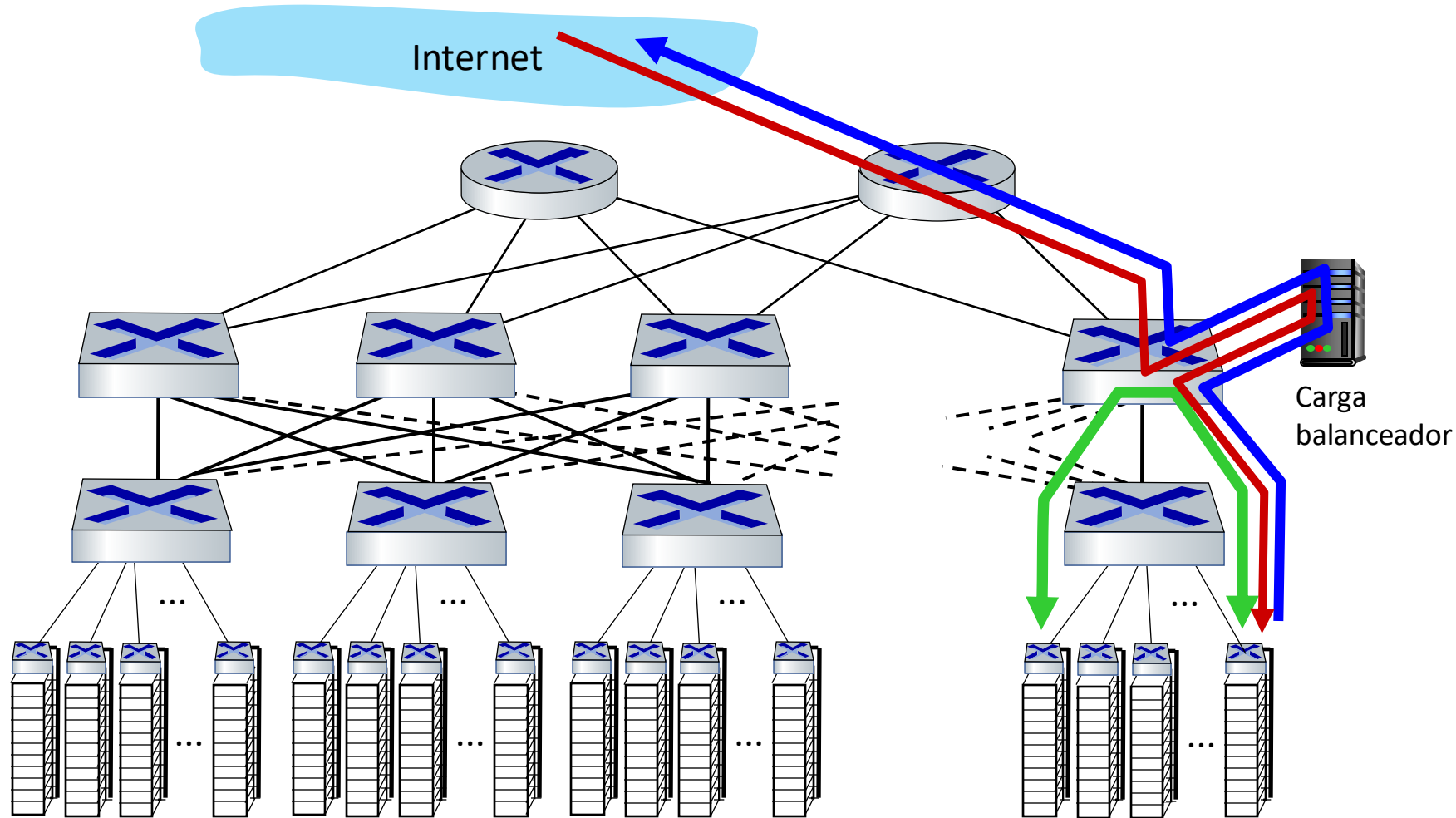
# Redes de data center: multipath

- rica interconexão entre switches, racks:
  - maior rendimento entre racks (possibilidade de vários caminhos de roteamento)
  - maior confiabilidade por meio de redundância



dois caminhos **disjuntos** destacados entre os racks 1 e 11

# Redes de data center: roteamento na camada de aplicativos



balanceador de carga: roteamento na camada de aplicativos

- recebe solicitações de clientes externos
- direciona a carga de trabalho ao data center
- retorna os resultados para o cliente externo (ocultando os dados internos do data center do cliente)



# Redes de data center: inovações de protocolo

- camada de link:

- RoCE: DMA remoto (RDMA) sobre Ethernet convergente

- camada de transporte:

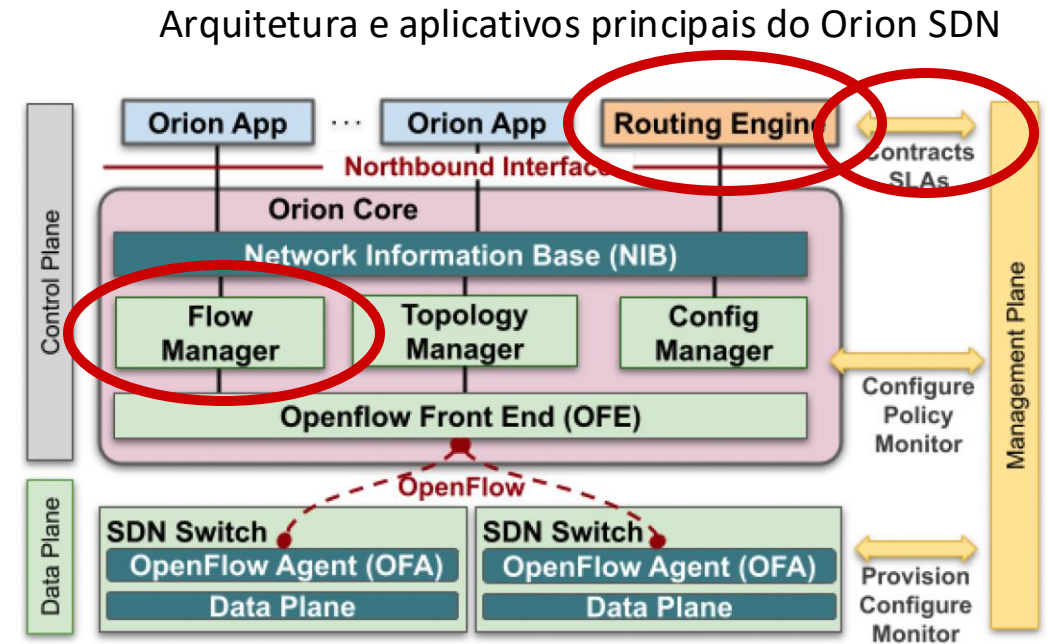
- ECN (notificação explícita de congestionamento) usada no controle de congestionamento da camada de transporte (DCTCP, DCQCN)
- experimentação com controle de congestionamento hop-by-hop (backpressure)

- roteamento, gerenciamento:

- SDN amplamente utilizado dentro/entre os data centers das organizações
- Coloque os serviços relacionados e os dados o mais próximo possível (por exemplo, no mesmo rack ou em um rack próximo) para minimizar a comunicação entre as camadas 2 e 1.

# ORION: o novo plano de controle SDN do Google para datacenter interno (Jupiter) + rede de área ampla (B4)

- **Roteamento** (intradomínio, iBGP), engenharia de tráfego: implementado em *aplicativos* sobre o núcleo do ORION
- controles **baseados em fluxo de borda a borda** (por exemplo, agendamento CoFlow) para atender aos SLAs de contrato
- **gerenciamento**: microsserviços distribuídos pub-sub no núcleo do Orion, OpenFlow para sinalização/monitoramento de switches



## Observação:

- sem protocolos de roteamento, controle de congestionamento (parcialmente) também gerenciado por SDN em vez de por protocolo
- os protocolos estão morrendo?

# Camada de link, LANs: roteiro

- introdução
- detecção e correção de erros
- protocolos de acesso múltiplo
- LANs
  - endereçamento, ARP
  - Ethernet
  - interruptores
  - VLANs
- virtualização de links: MPLS
- rede de data center

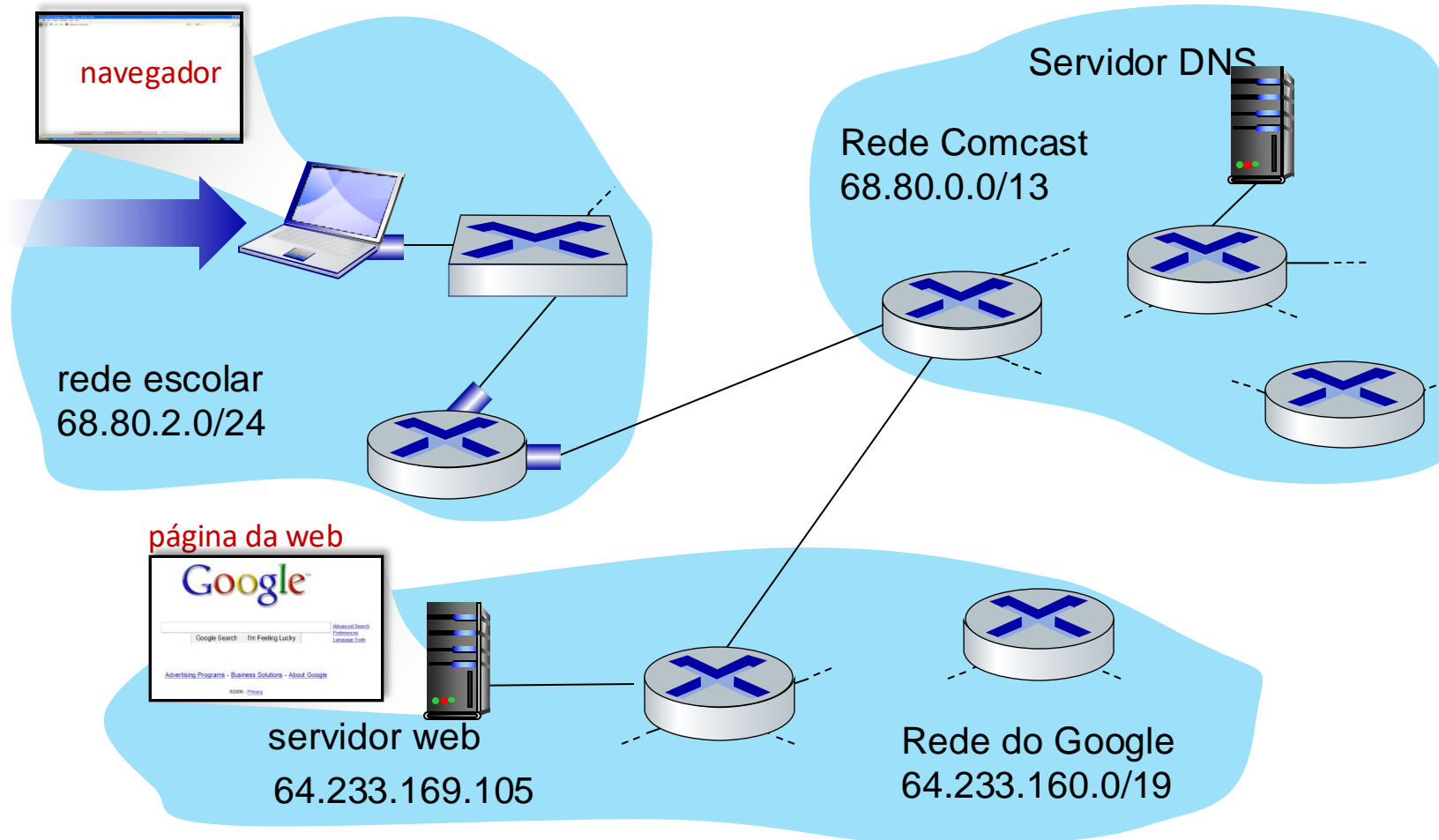


- Um dia na vida de uma solicitação da Web

# Síntese: um dia na vida de uma solicitação da Web

- nossa jornada pela pilha de protocolos está concluída!
  - aplicativo, transporte, rede, link
- Juntando tudo: síntese!
  - *Objetivo*: identificar, revisar e compreender os protocolos (em todas as camadas) envolvidos em um cenário aparentemente simples: solicitação de página www
  - *Cenário*: o aluno conecta o laptop à rede do campus, solicita/recebe `www.google.com`

# Um dia na vida: cenário

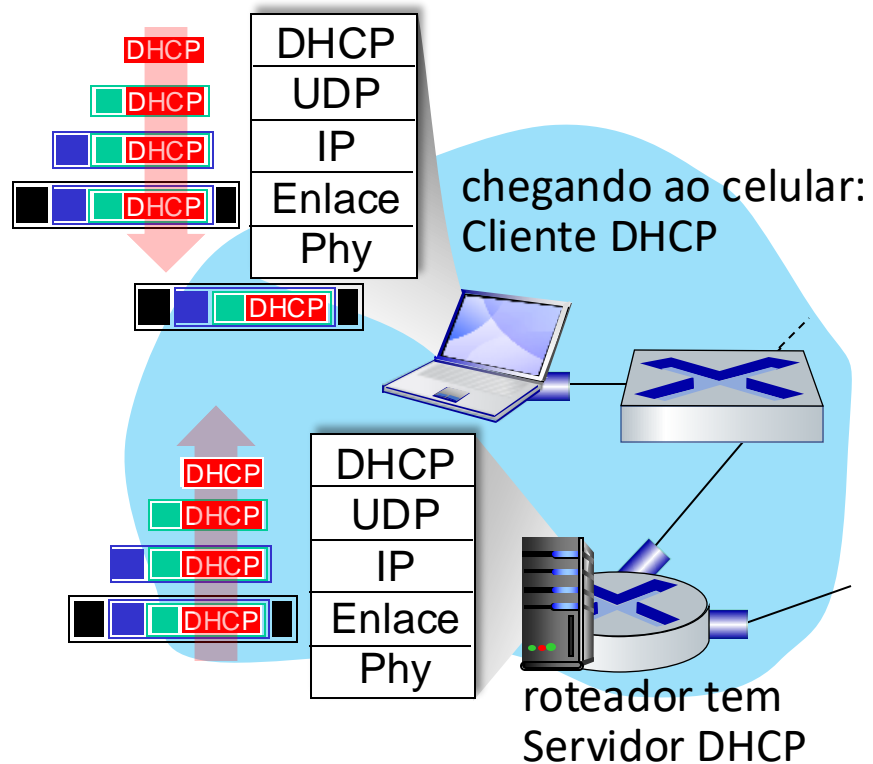


cenário:

- O cliente móvel que chega se conecta à rede
- solicita página da web: [www.google.com](http://www.google.com)

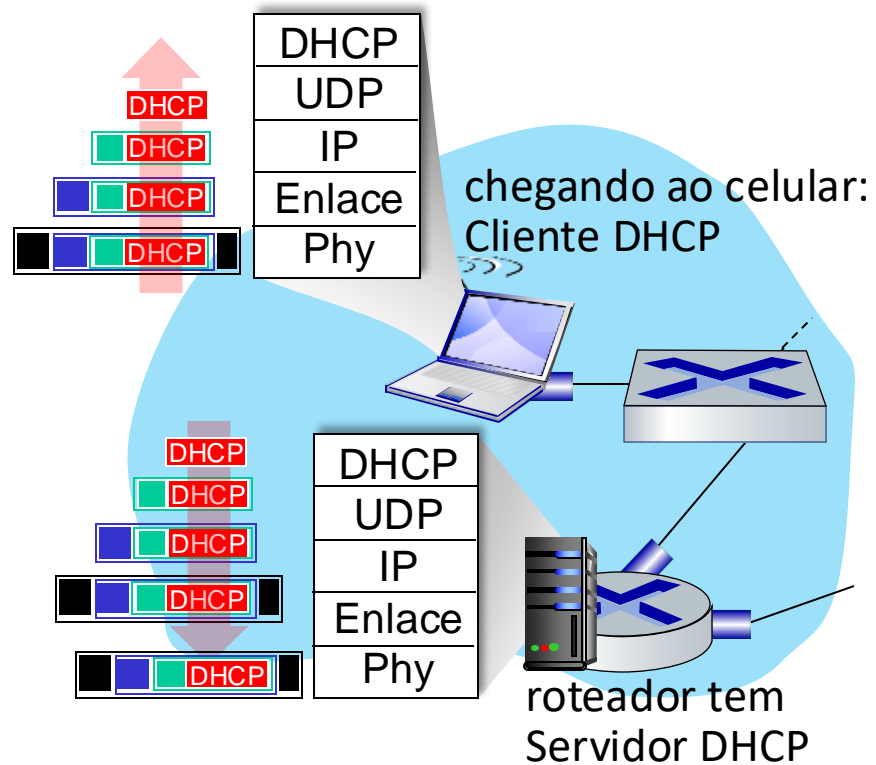
*Sons  
simples!* 

# Um dia na vida: conectando-se à Internet



- O laptop conectado precisa obter seu próprio endereço IP, o endereço do roteador de primeiro salto e o endereço do servidor DNS: use o **DHCP**
- Solicitação DHCP **encapsulada** em **UDP**, encapsulada em **IP**, encapsulada em Ethernet **802.3**
- **Broadcast** de quadro Ethernet (destino: FFFFFFFFFFFFFFFF) na LAN, recebido no roteador que está executando o servidor **DHCP**
- Ethernet **sem fusão** para IP sem fusão, UDP sem fusão para DHCP

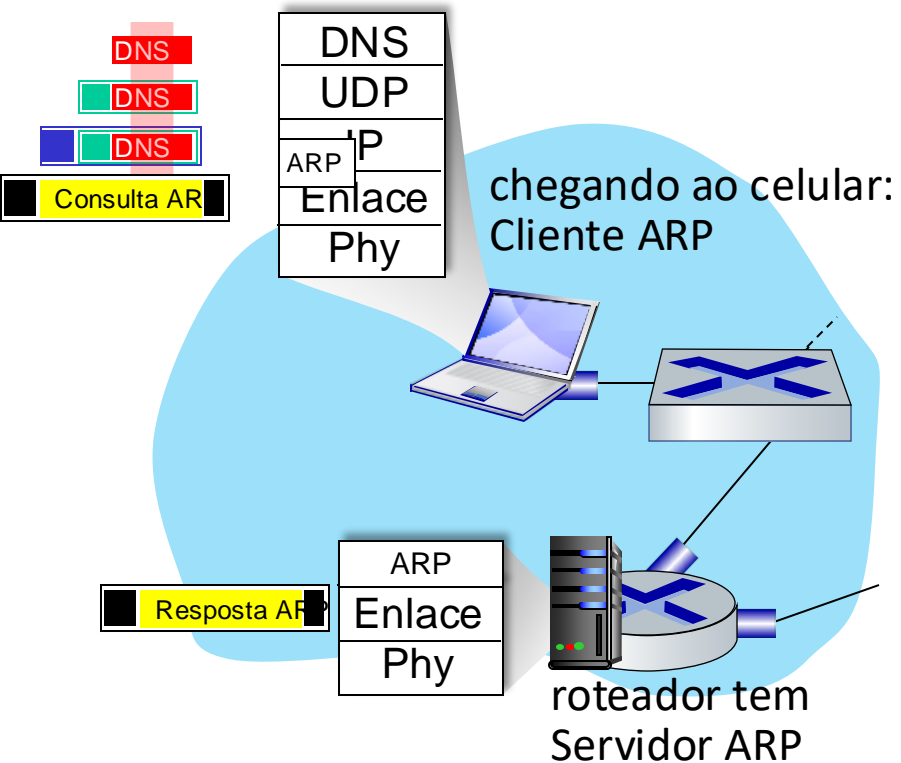
# Um dia na vida: conectando-se à Internet



- O servidor DHCP formula o **DHCP ACK** contendo o endereço IP do cliente, o endereço IP do roteador de primeiro salto para o cliente, o nome e o endereço IP do servidor DNS
- encapsulamento no servidor DHCP, quadro encaminhado (**aprendizagem do switch**) pela LAN, demultiplexação no cliente
- O cliente DHCP recebe a resposta DHCP ACK

*O cliente agora tem endereço IP, sabe o nome e o endereço do DNS servidor, endereço IP de seu roteador de primeiro salto*

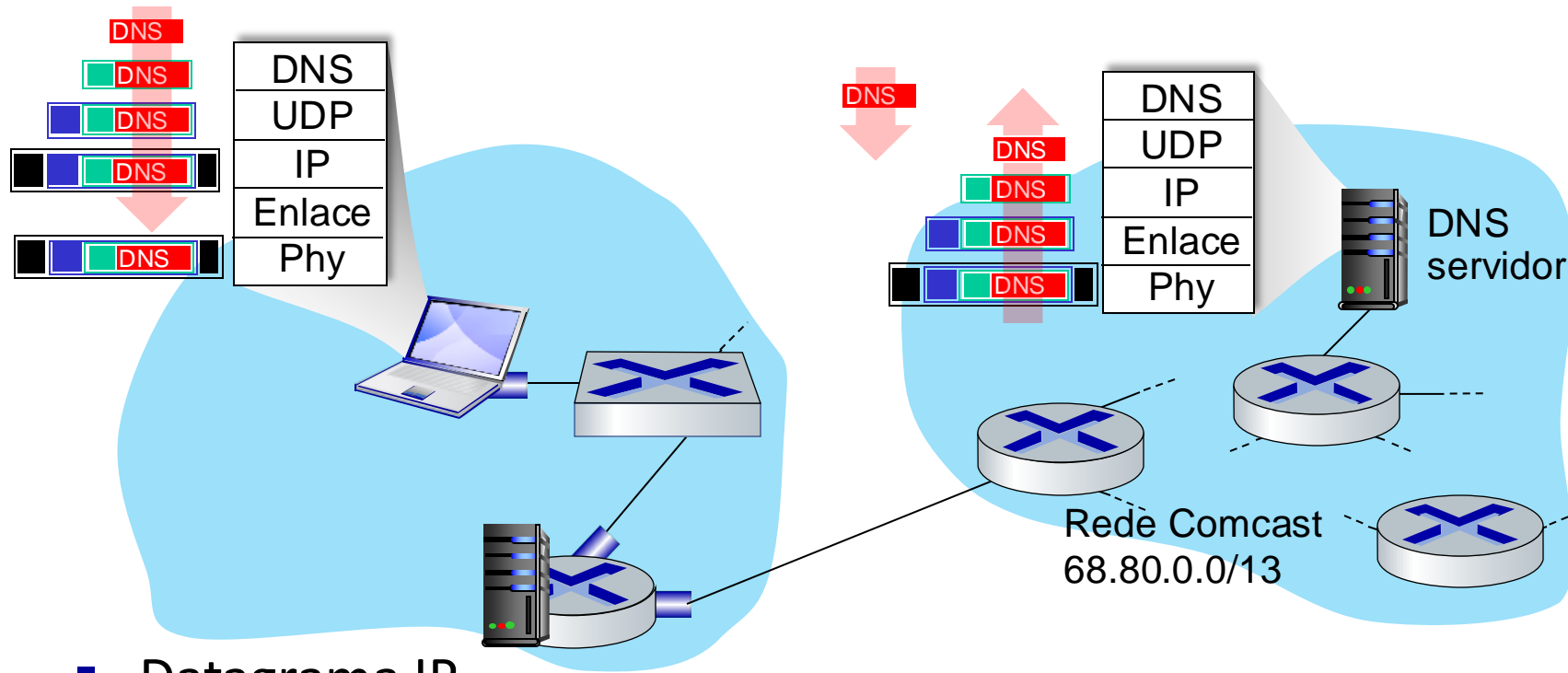
# Um dia na vida... ARP (antes do DNS, antes do HTTP)



- antes de enviar a solicitação **HTTP**, precisa do endereço IP de `www.google.com`: **DNS**
- Consulta de DNS criada, encapsulada em UDP, encapsulada em IP, encapsulada em Eth. Para enviar o quadro ao roteador, é necessário o endereço MAC da interface do roteador: **ARP**
- **Consulta ARP** transmitida, recebida pelo roteador, que responde com uma **resposta ARP** fornecendo o endereço MAC da interface do roteador
- o cliente agora sabe o endereço MAC do roteador de primeiro salto, portanto, pode enviar um quadro contendo uma consulta de DNS



# Um dia na vida... usando DNS

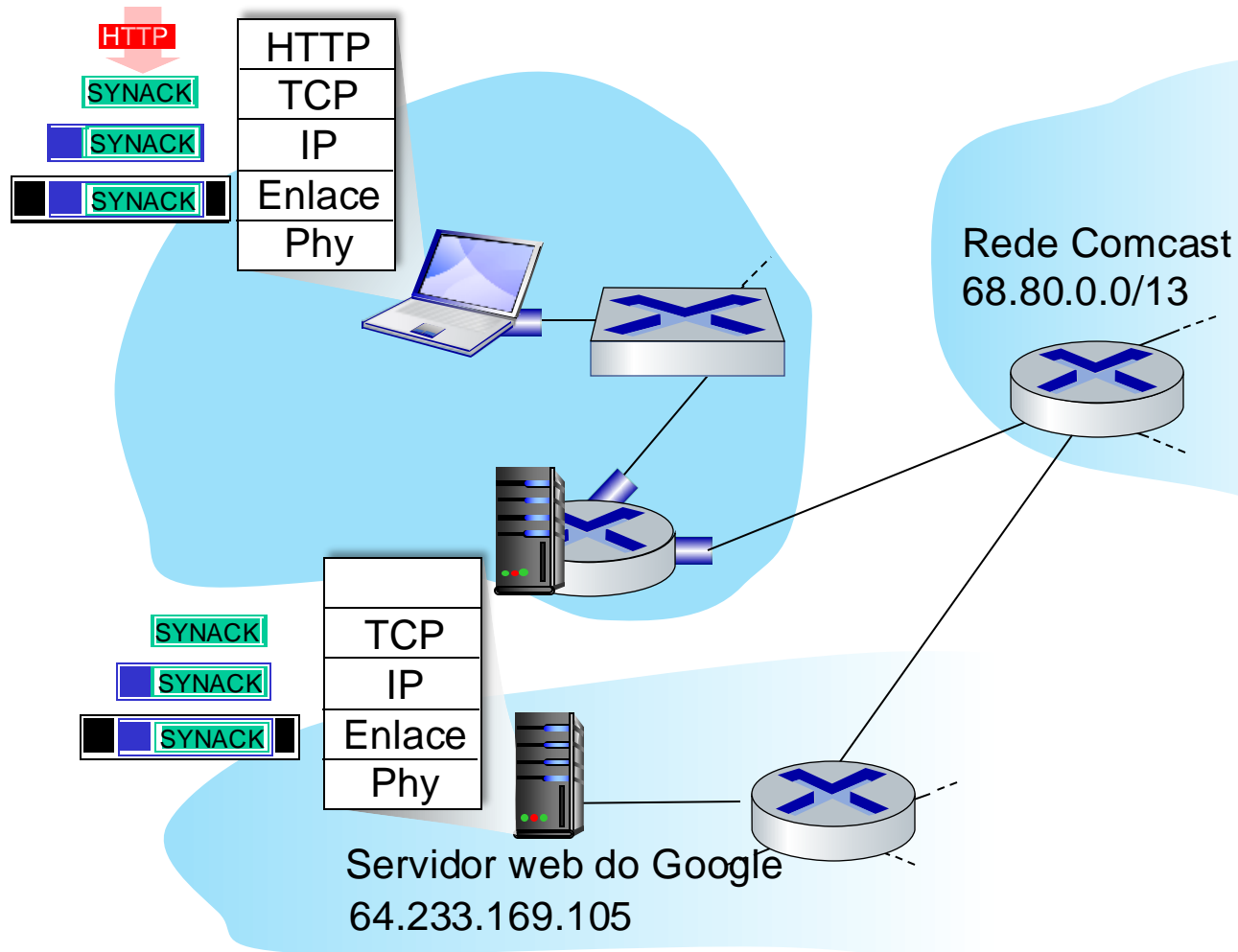


- Datagrama IP contendo consulta de DNS encaminhado via switch de LAN do cliente para 1<sup>st</sup> roteador de salto

- Datagrama IP encaminhado da rede do campus para a rede da Comcast, roteado (tabelas criadas pelos protocolos de roteamento **RIP**, **OSPF**, **IS-IS** e/ou **BGP**) para o servidor DNS

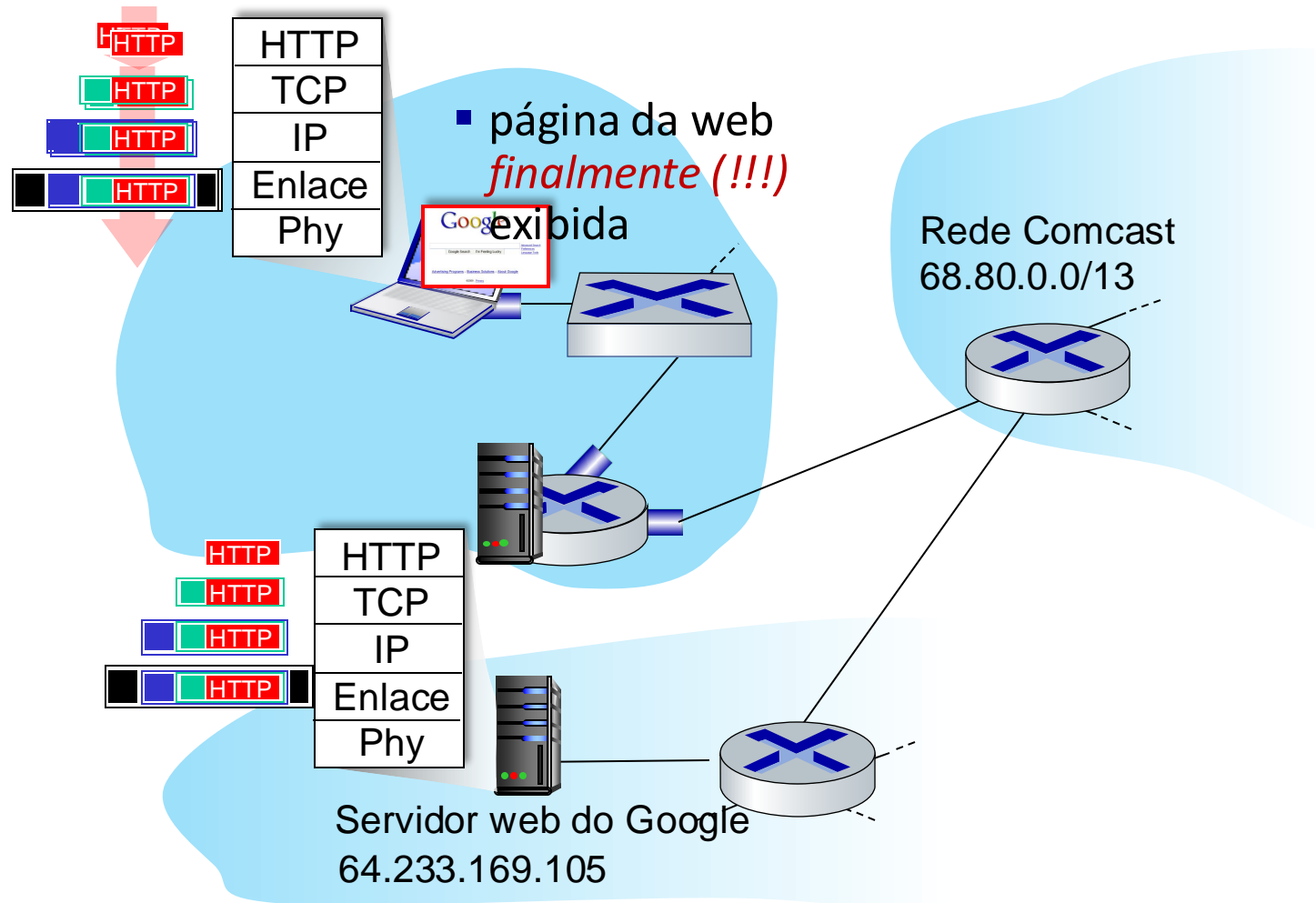
- descompactado para DNS
- O DNS responde ao cliente com o endereço IP de [www.google.com](http://www.google.com)

# Um dia na vida... Conexão TCP com HTTP



- Para enviar uma solicitação HTTP, o cliente primeiro abre **um socket TCP** para o servidor da Web
- **Segmento TCP SYN** (etapa 1 no handshake de 3 vias do TCP) interdomínio roteado para o servidor da Web
- o servidor da Web responde com **TCP SYNACK** (etapa 2 do handshake de 3 vias do TCP)
- **Conexão TCP estabelecida!**

# Um dia na vida... Solicitação/resposta HTTP



- **Solicitação HTTP** enviada para o socket TCP
- Datagrama IP contendo solicitação HTTP encaminhada para `www.google.com`
- O servidor da Web responde com uma **resposta HTTP** (contendo a página da Web)
- Datagrama IP contendo resposta HTTP encaminhada de volta ao cliente

# Capítulo 6: Resumo

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

# Capítulo 6: Vamos respirar um pouco

- Pilha de protocolo de jornada descendente *completa* (exceto PHY)
- sólido entendimento dos princípios de rede, prática!
- ..... poderia parar por aqui .... mas há tópicos *mais* interessantes!
  - sem fio
  - segurança

# Slides adicionais do Capítulo 6

# Pura eficiência da ALOHA

$$P(\text{sucesso de um determinado nó}) = P(\text{nó transmite}) *$$

$$P(\text{nenhum outro nó transmite em } [t_0 - 1, t_0]) *$$

$$P(\text{nenhum outro nó transmite em } [t_0 - 1, t_0]) *$$

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

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

... escolhendo o melhor  $p$  e, em seguida, deixando  $n$

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

→ ∞

ainda pior do que o Aloha com slot!