

Redes de Computadores

A Camada de Enlace e LANs

Camada de enlace e LANs: nossos objetivos

- entender os princípios por trás dos serviços da camada de enlace:
 - detecção e correção de erros
 - compartilhar um canal de difusão (*broadcast*): acesso múltiplo
 - endereçamento da camada de enlace
 - redes locais: Ethernet, VLANs
- redes de *datacenter*
- instanciação e implementação de várias tecnologias de camada de enlace



Camada de enlace, LANs: roteiro

- introdução
- detecção e correção de erros
- protocolos de acesso múltiplo
- LANs
 - endereçamento, ARP
 - Ethernet
 - comutadores (*switches*)
 - VLANs
- virtualização de enlaces: MPLS
- redes em *data centers*



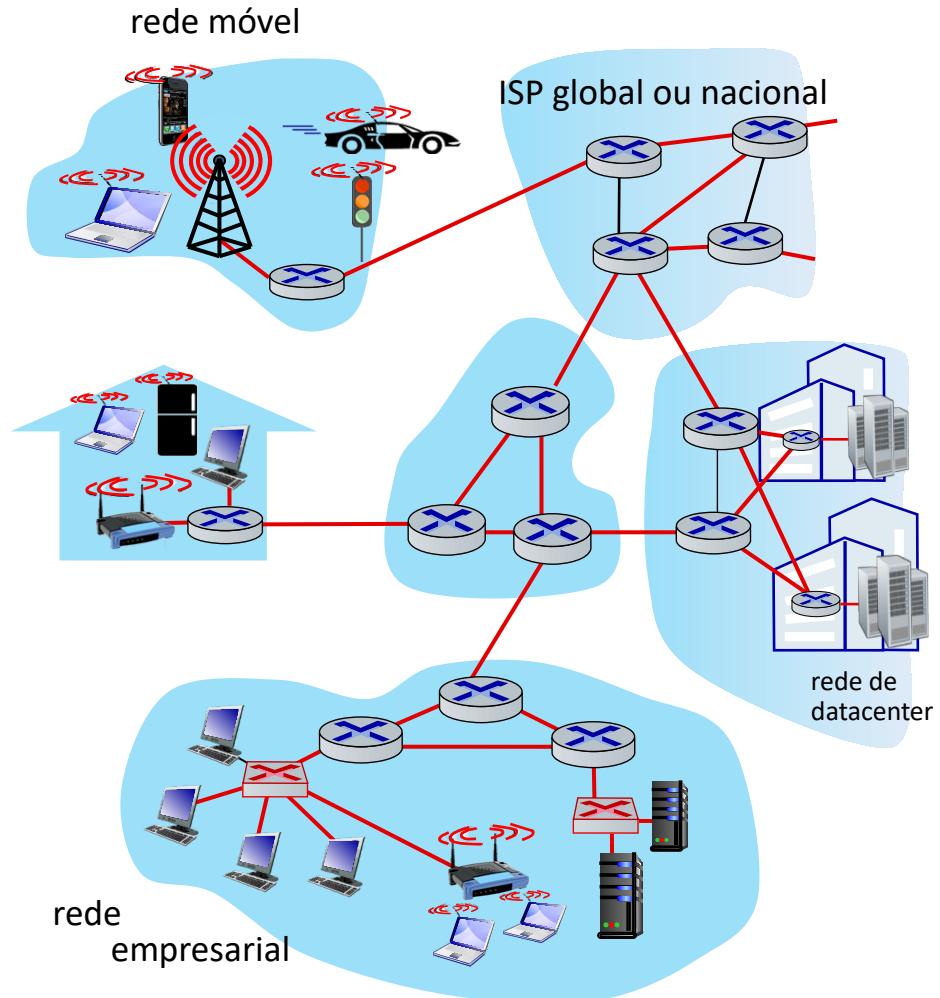
- um dia na vida de uma requisição da web

Camada de enlace: introdução

terminologia:

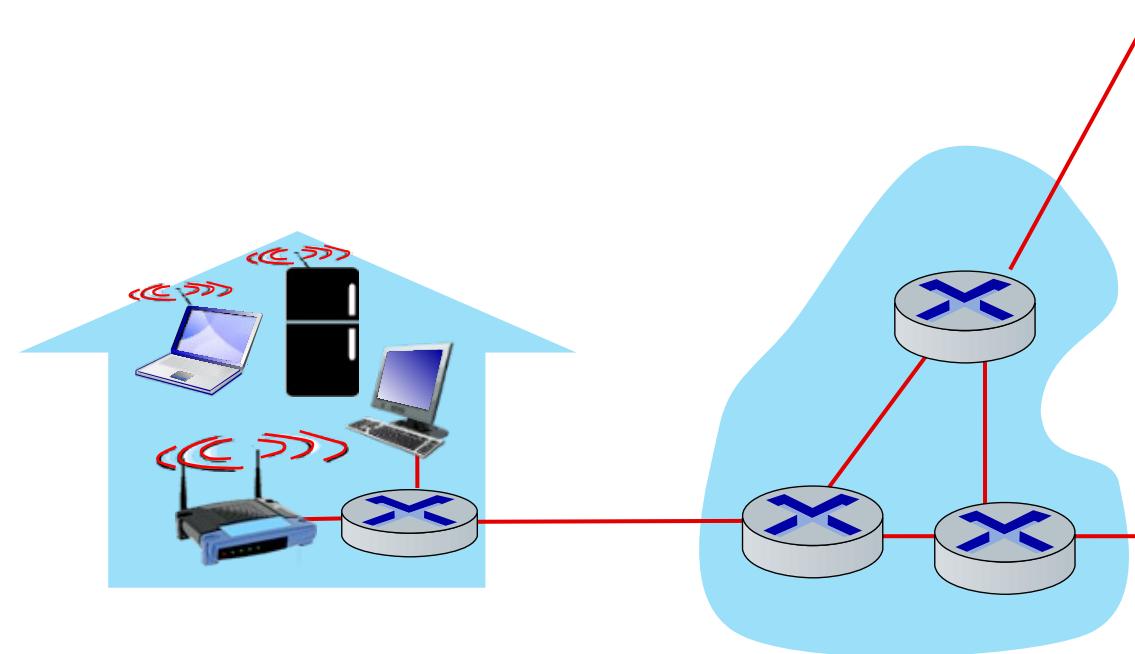
- hospedeiros e roteadores: **nós**
- canais de comunicação que conectam nós adjacentes ao longo do caminho de comunicação: **enlaces**
 - com fio, sem fio
 - LANs
- pacote de camada 2: **quadro**, encapsula datagrama

*camada de enlace tem a responsabilidade de transferir datagramas de um nó para outro nó **fisicamente adjacente** através de um enlace*

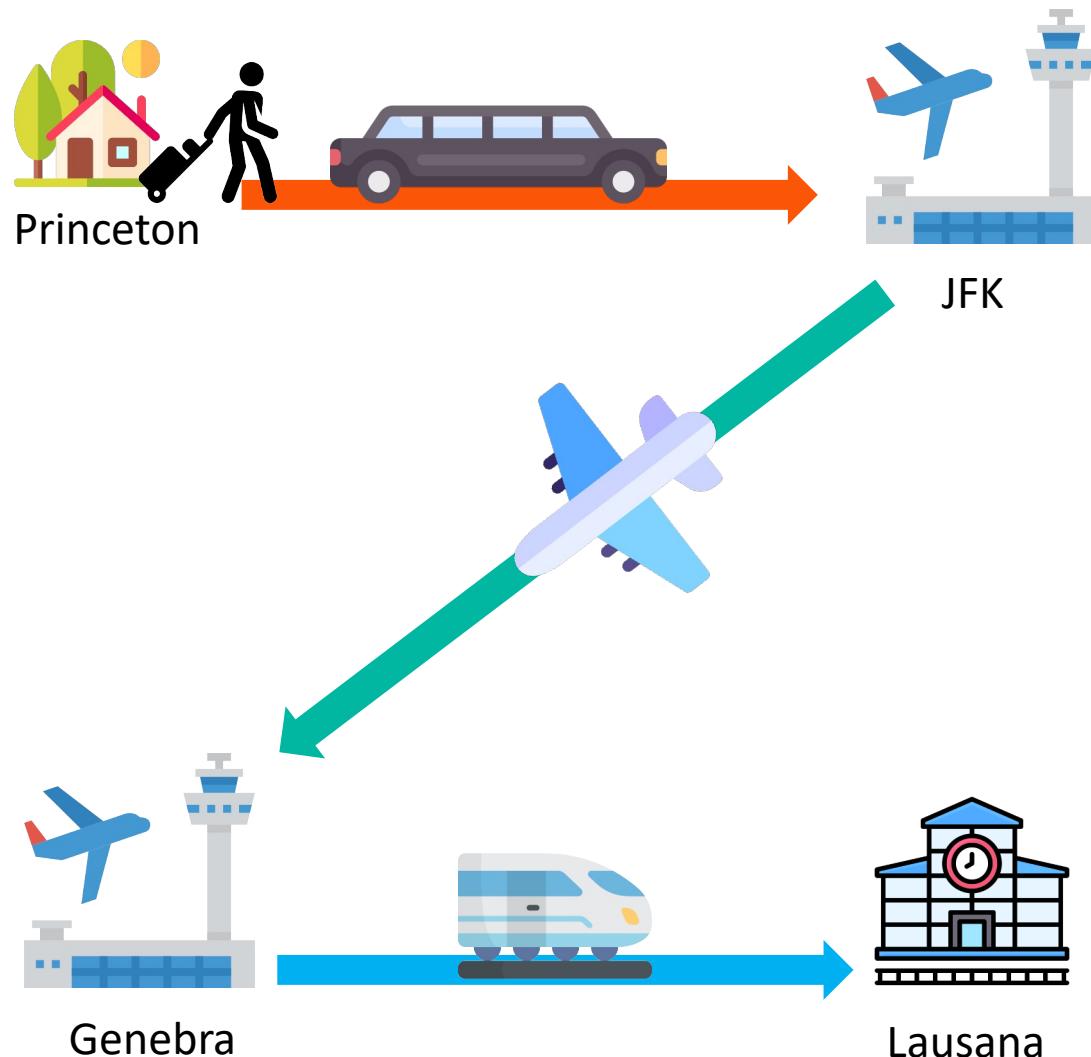


Camada de enlace: contexto

- datagrama transferido por diferentes protocolos de enlace sobre diferentes enlaces:
 - por exemplo, WiFi no primeiro enlace, Ethernet no próximo
- cada protocolo de enlace fornece serviços diferentes
 - por exemplo, pode ou não fornecer transferência de dados confiável sobre o enlace



Analogia de transporte

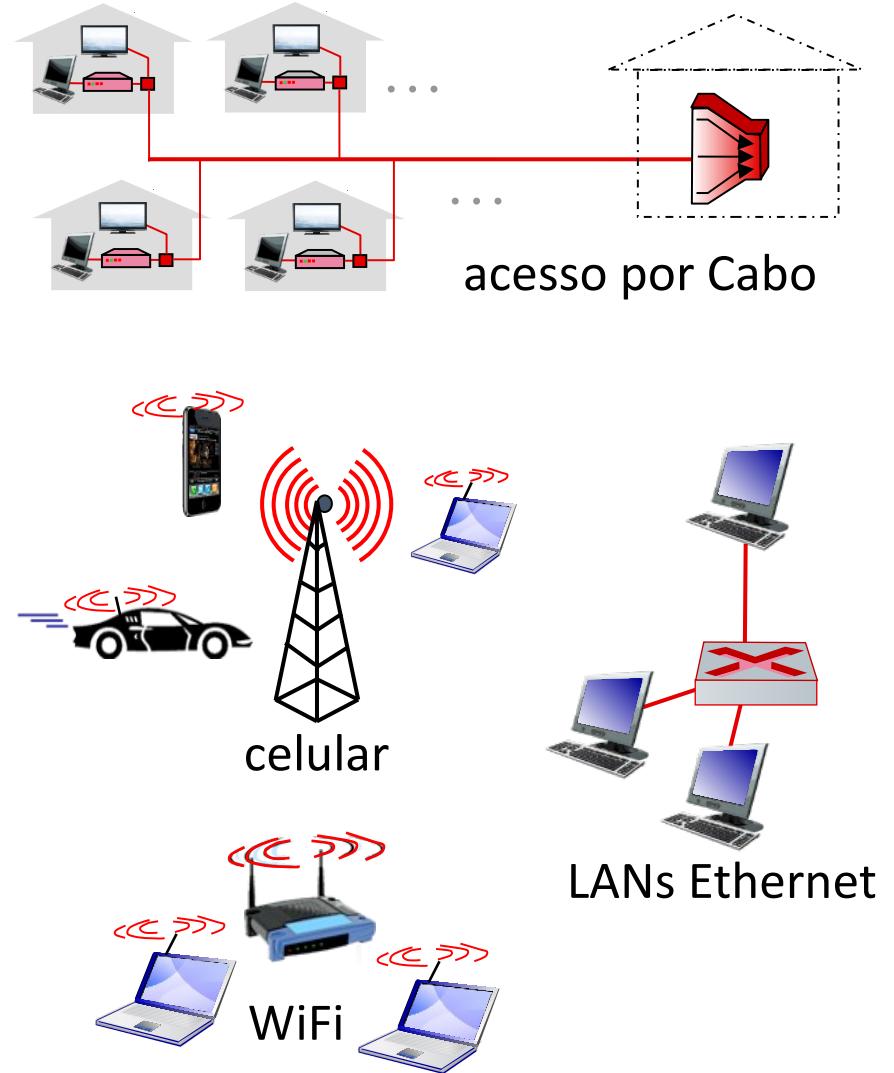


analogia de transporte:

- viagem de Princeton a Lausana
 - táxi: de Princeton para o JFK
 - avião: do JFK para Genebra
 - trem: de Genebra para Lausana
- turista = datagrama
- segmento de transporte = enlace de comunicação
- modo de transporte = protocolo da camada de enlace
- agente de viagens = algoritmo de roteamento

Camada de enlace: serviços

- **Enquadramento e acesso ao enlace:**
 - encapsulamento do datagrama no quadro, adicionando cabeçalho e trailer
 - acesso ao canal se o meio for compartilhado
 - os endereços “MAC” nos cabeçalhos dos quadros identificam a origem e o destino (diferente dos endereços IP!)
- **entrega confiável entre nós adjacentes**
 - raramente usado em enlaces com poucos erros de bits
 - enlaces sem fio: altas taxas de erro
 - Q: por que a confiabilidade tanto no nível de enlace quanto fim a fim?



Camada de enlace: serviços (mais)

- **controle de fluxo:**

- ritmo entre os nós de envio e recebimento adjacentes

- **deteção de erro:**

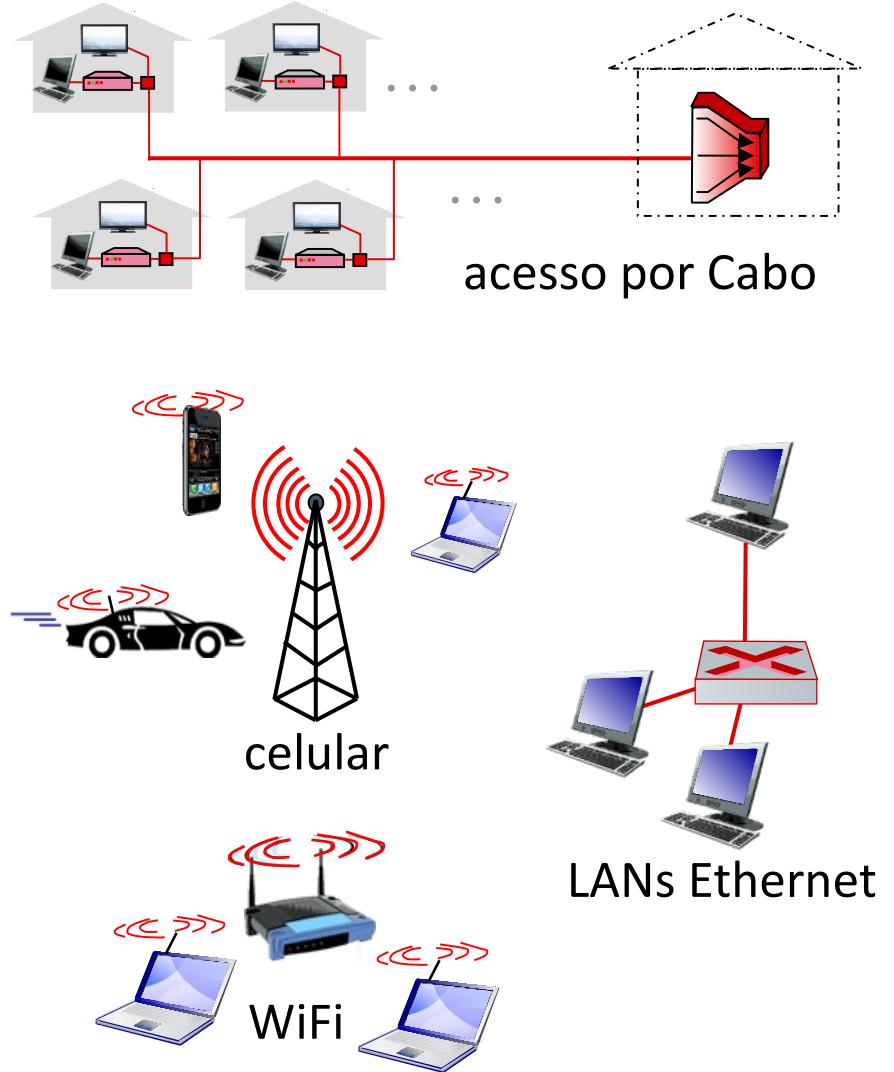
- erros causados pela atenuação do sinal e ruído.
- o receptor detecta erros, sinaliza para retransmissão ou descarta o quadro

- **correção de erro:**

- receptor detecta *e corrige* erro(s) de bit(s) sem retransmissão

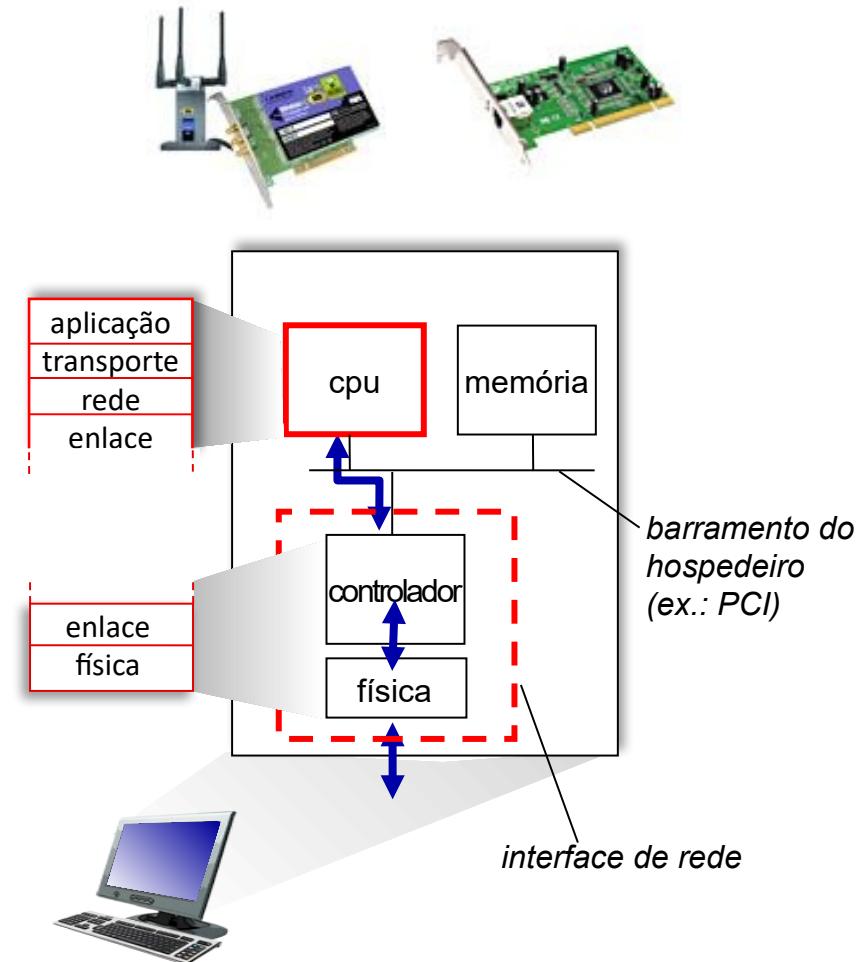
- **half-duplex e full-duplex:**

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

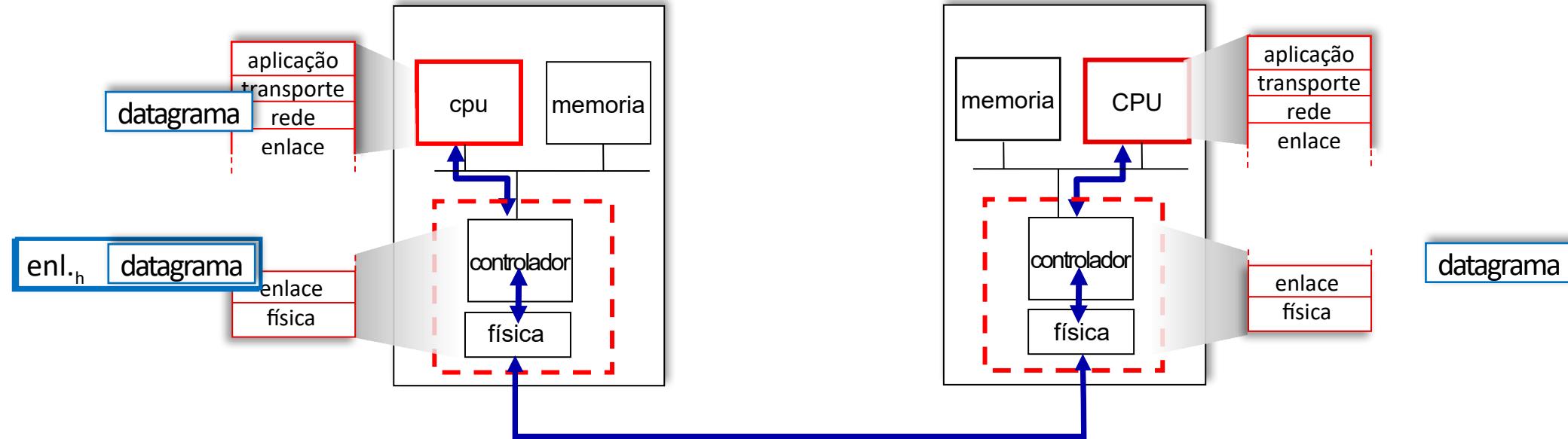


Onde a camada de enlace é implementada?

- em todo e cada hospedeiro
- camada de enlace implementada na *placa de interface de rede* (NIC – *network interface card*) ou em um chip
 - placa ou chip Ethernet ou WiFi
 - implementa as camadas de enlace e física
- se conecta aos barramentos do sistema do hospedeiro
- combinação de hardware, software, e firmware



Interfaces de comunicação



lado emissor:

- encapsula datagrama no 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 enlace, LANs: roteiro

- introdução
- detecção e correção de erros
- protocolos de acesso múltiplo
- LANs
 - endereçamento, ARP
 - Ethernet
 - comutadores (*switches*)
 - VLANs
- virtualização de enlaces: MPLS
- redes em *data centers*

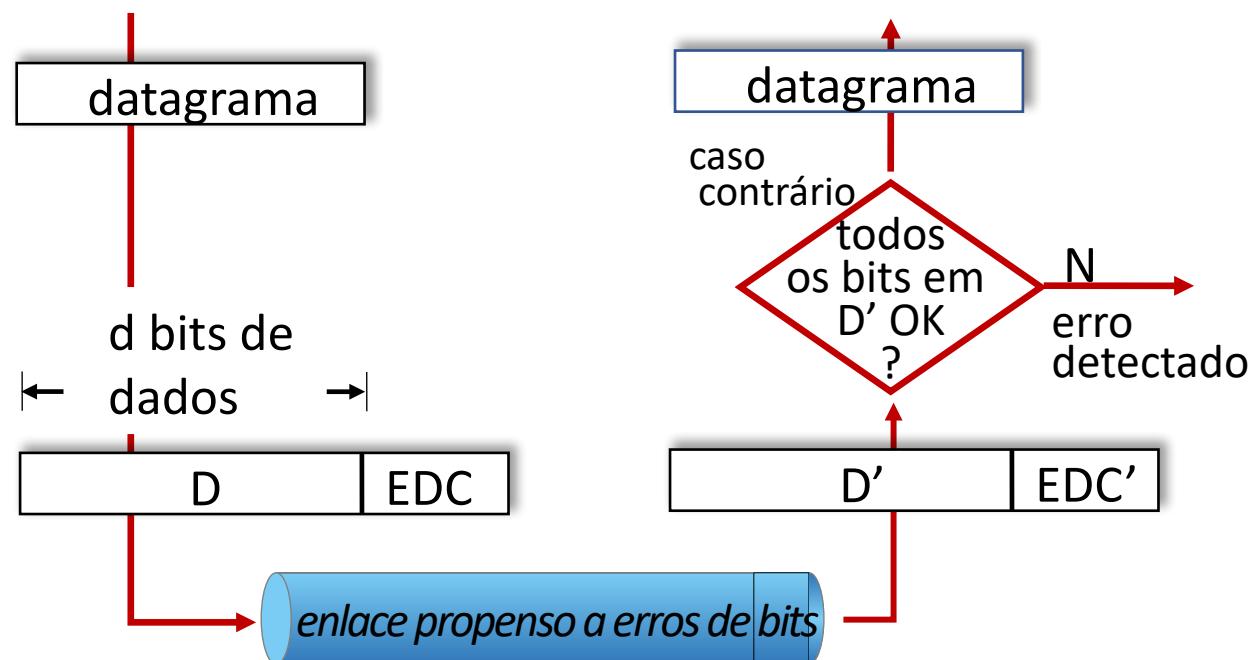


- um dia na vida de uma requisição web

Detecção de erro

EDC: *error detection and correction bits* (bits de detecção e correção de erros, por exemplo: redundância)

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



Detecção de erros não é 100% confiável!

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

Verificação de paridade

paridade de bit único:

- detecta erros de um único bit

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

← d bits de dados → |
bit de paridade

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

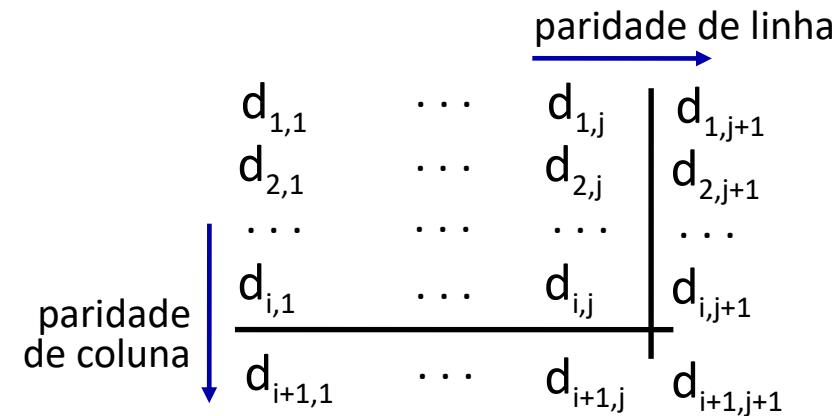
No receptor:

- compute paridade de d bits recebidos
- compare com o bit de paridade recebido – se for diferente então um erro foi detectado



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

- paridade bi-dimensional: detecta **e corrige** erros de um único bit



sem erros: 1 0 1 0 1 | 1
1 1 1 1 0 | 0
0 1 1 1 0 | 1

0 0 1 0 1 | 0

single-bit
erros de
único bit
detectados
e
corrigíveis

1 0 1 0 1 | 1
1 0 1 1 0 | 0 → erro de paridade
0 1 1 1 0 | 1

0 0 1 0 1 | 0
↓ erro de paridade

Soma de verificação da Internet

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

emissor:

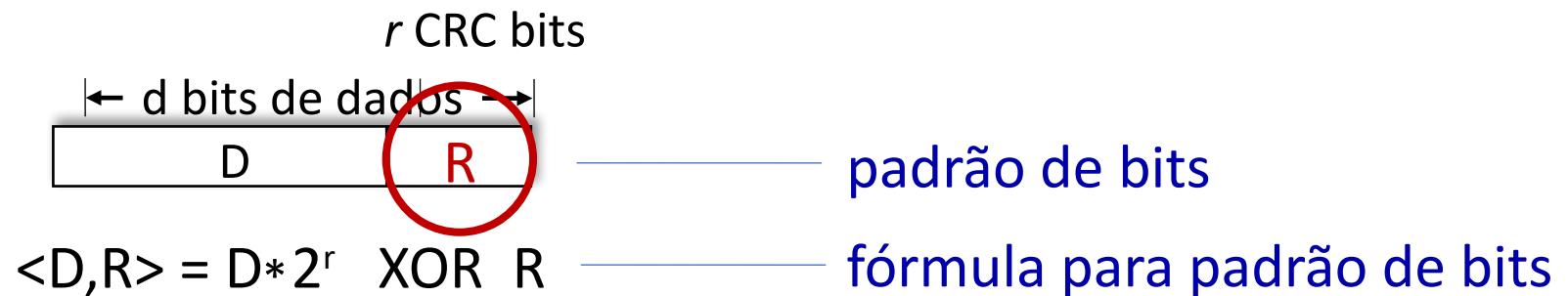
- trata o conteúdo do segmento UDP ou TCP (incluindo campos de cabeçalho UDP ou TCP e endereços IP) como sequências de inteiros de 16 bits
- **soma de verificação:** adição (soma do complemento de um) do conteúdo do segmento
- valor da soma de verificação colocado no campo de soma de verificação do UDP ou TCP

receptor:

- calcula a soma de verificação do segmento recebido
- verifica se a soma de verificação calculada é igual ao valor do campo da soma de verificação:
 - diferente - erro detectado
 - igual - nenhum erro detectado. *Mas talvez existam erros mesmo assim?* Já vimos que sim!

Cyclic Redundancy Check (CRC- Verificação de Redundância Cíclica)

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



objetivo: computar r bits de CRC, **R**, tal que $\langle D, R \rangle$ seja exatamente divisível por G
(mod 2)

- 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 rajada menores que $r+1$ bits
- amplamente utilizado na prática (Ethernet, 802.11 WiFi)

Cyclic Redundancy Check (CRC): exemplo

Queremos computar R tal que:

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

...ou equivalentemente (XOR R em ambos os lados):

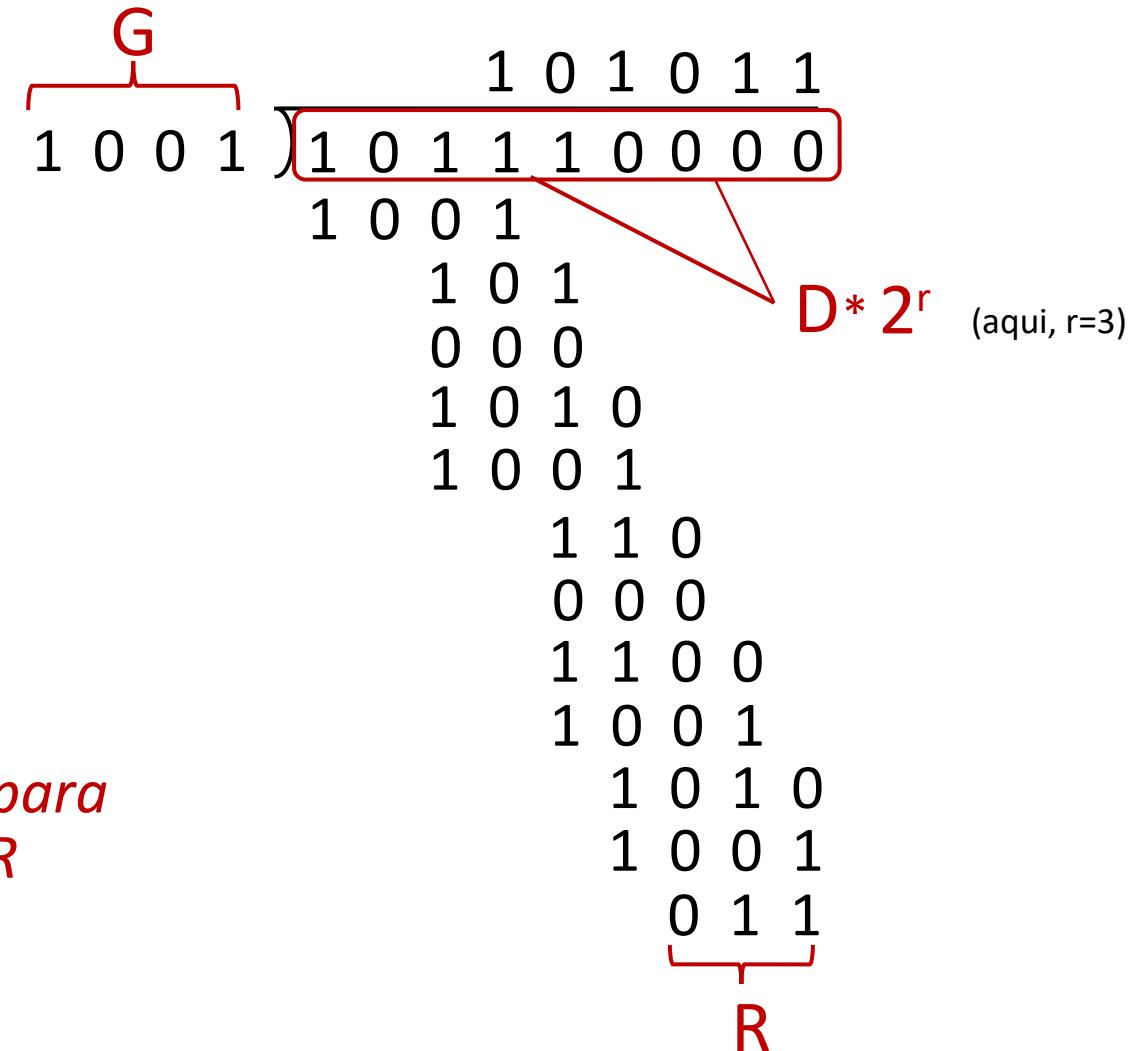
$$D \cdot 2^r = nG \text{ XOR } R$$

...o que significa:

se dividirmos $D \cdot 2^r$ por G,
queremos o resto R que
satisfaca:

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

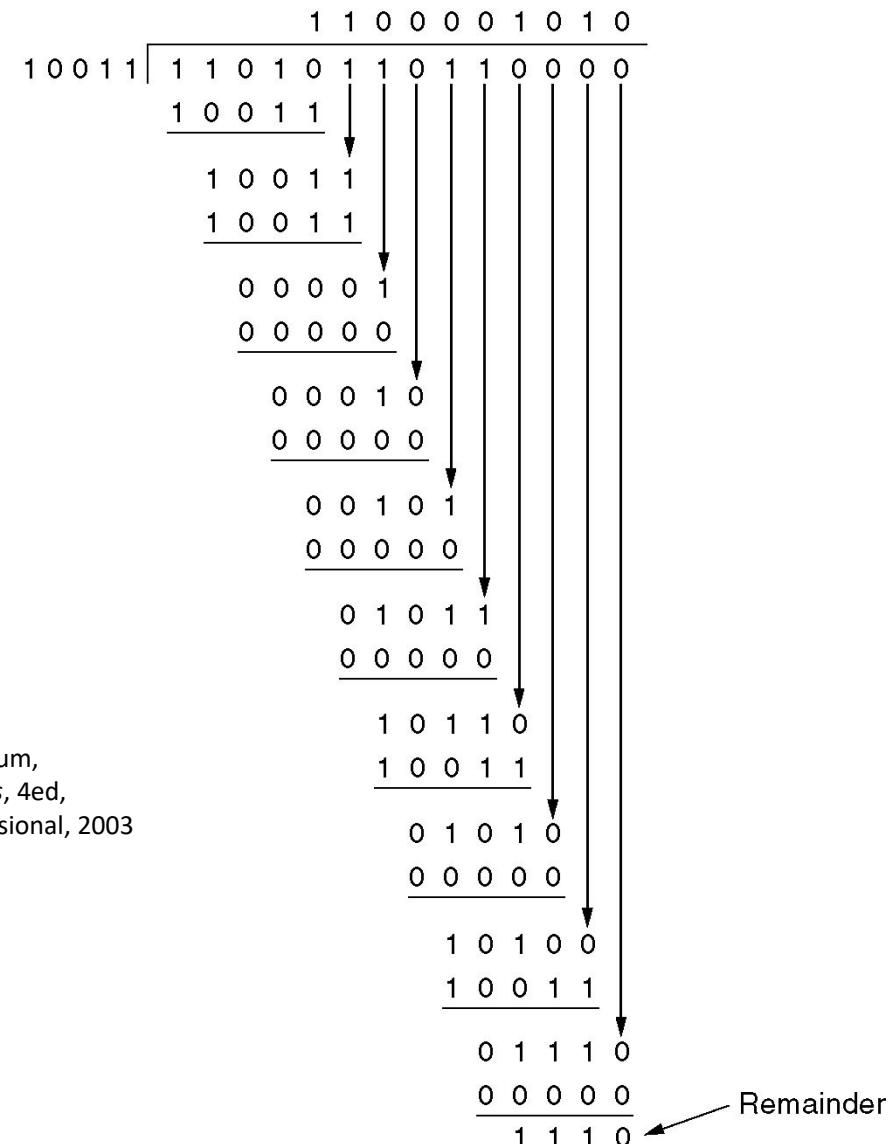
*algoritmo para
computar R*



* Confira os exercícios interativos online para mais exemplos: http://gaia.cs.umass.edu/kurose_ross/interactive/

CRC: outro exemplo

Frame : 1 1 0 1 0 1 1 0 1 1
Generator: 1 0 0 1 1
Message after 4 zero bits are appended: 1 1 0 1 0 1 1 0 1 1 0 0 0



Andrew S. Tanenbaum,
Computer Networks, 4ed,
Prentice Hall Professional, 2003

Transmitted frame: 1 1 0 1 0 1 1 0 1 1 1 1 0

Camada de enlace, LANs: roteiro

- introdução
- detecção e correção de erros
- **protocolos de acesso múltiplo**
- LANs
 - endereçamento, ARP
 - Ethernet
 - comutadores (*switches*)
 - VLANs
- virtualização de enlaces: MPLS
- redes em *data centers*



- um dia na vida de uma requisição web

Enlaces e protocolos de acesso múltiplo

dois tipos de “enlaces”:

- ponto a ponto
 - enlace ponto a ponto entre comutador Ethernet e hospedeiro
 - PPP para acesso discado
- difusão (*broadcast* - fio ou mídia compartilhados)
 - Ethernet à moda antiga
 - *upstream* de HFC em rede de acesso baseada em cabo
 - LAN sem fio 802.11, 4G/5G, satélite



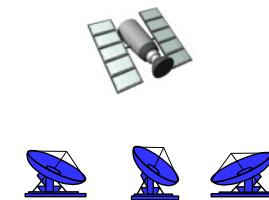
fio compartilhado
(por exemplo,
Ethernet cabeada)



rádio compartilhado:
4G/5G



rádio compartilhado:
WiFi



rádio compartilhado:
satélite



humanos em um coquetel (ar e
acústica compartilhados)

Protocolos de acesso múltiplo

- único canal de transmissão 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 o canal, ou seja, determina quando o nó pode transmitir
- comunicação sobre compartilhamento de canal 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

desejado:

1. quando um nó quer transmitir, ele pode enviar na 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 *clocks* ou intervalos (*slots*)
4. simples

Protocolos MAC: taxonomia

três grandes classes:

- **particionamento de canal**

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

- **acesso aleatório**

- canal não dividido, permite colisões
- se “recupera” de colisões

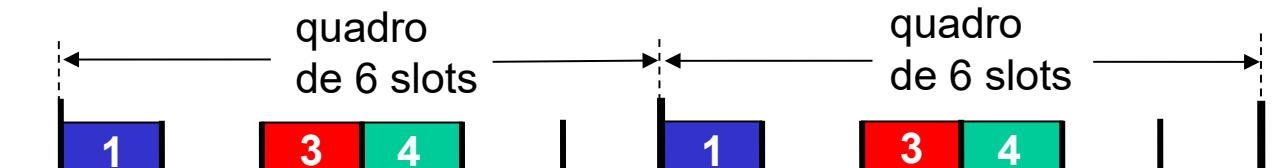
- **“revezamento”**

- os nós se revezam, mas os nós com mais para enviar podem ter turnos mais longos

Protocolos MAC de particionamento de canal: TDMA

TDMA: time division multiple access (acesso múltiplo por divisão de tempo)

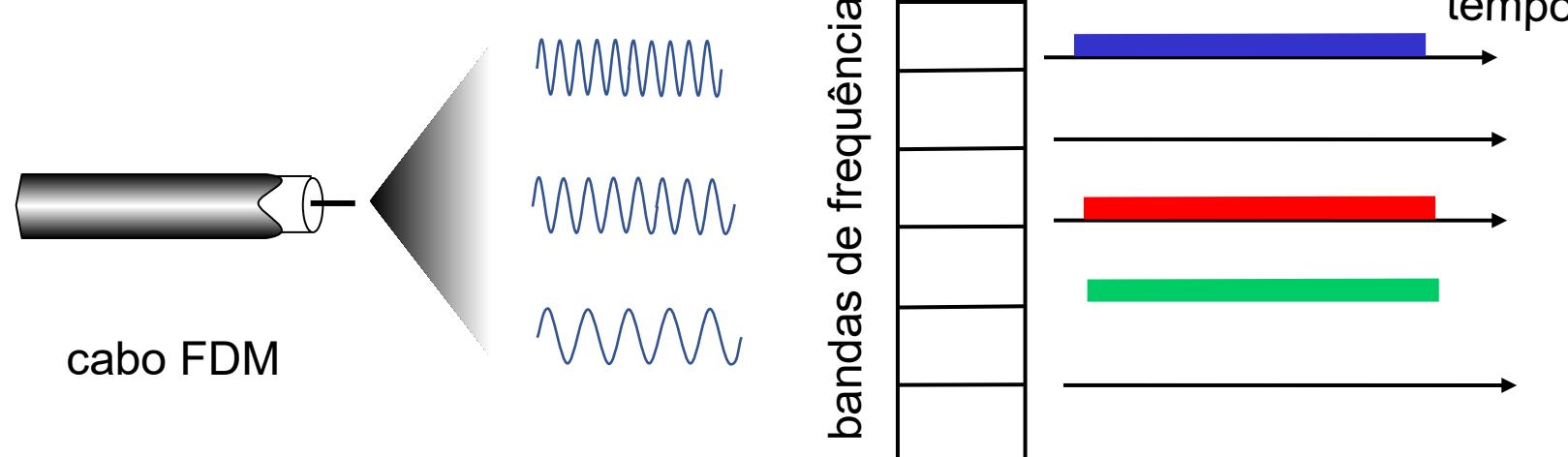
- acesso ao canal em “rodadas”
- cada estação recebe um slot de comprimento fixo (comprimento = tempo de transmissão do pacote) em cada rodada
- slots não utilizados ficam ociosos
- exemplo: LAN de 6 estações, 1,3,4 têm pacotes para enviar, slots 2,5,6 inativos



Protocolos MAC de particionamento de canal: FDMA

FDMA: frequency division multiple access (acesso múltiplo por divisão de frequência)

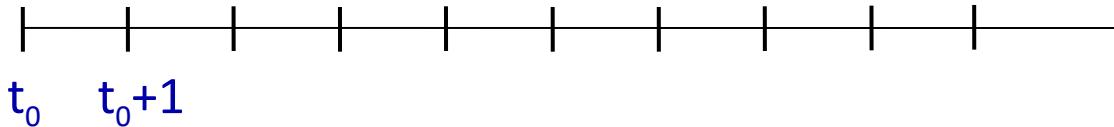
- espectro de canal dividido em bandas de frequência
- para cada estação é atribuída uma banda de frequência fixa
- o tempo de transmissão não utilizado nas bandas de frequência fica ocioso
- exemplo: LAN de 6 estações, 1,3,4 tem pacote para enviar, bandas de frequência 2,5,6 ociosas



Protocolos de acesso aleatório

- quando o nó tem pacote para enviar
 - transmite na taxa de dados total do canal R .
 - sem coordenação *a priori* entre nós
- dois ou mais nós transmitindo: “colisão”
- protocolos MAC de acesso aleatório especificam:
 - como detectar colisões
 - como se recuperar de colisões (por exemplo, por meio de retransmissões atrasadas)
- exemplos de protocolos MAC de acesso aleatório:
 - ALOHA, slotted ALOHA
 - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA



suposições:

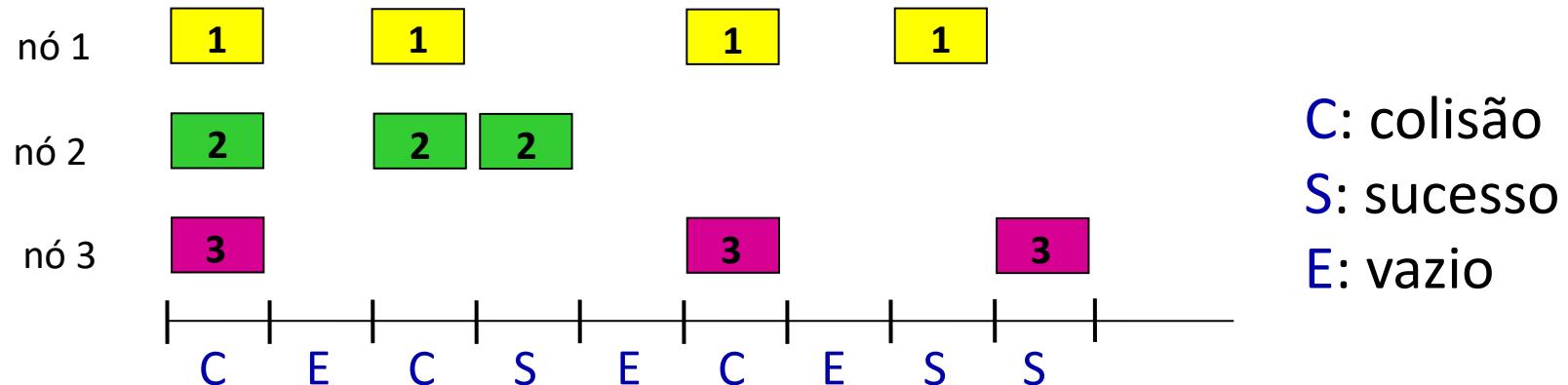
- todos os quadros do mesmo tamanho
- tempo dividido em slots de tamanhos iguais (tempo para transmitir 1 quadro)
- os nós começam a transmitir apenas o início do slot
- os nós são sincronizados
- se 2 ou mais nós transmitem no mesmo slot, todos os nós detectam a colisão

operação:

- quando um 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 quadro em cada slot subsequente com probabilidade p até o sucesso

aleatoriedade – por quê?

Slotted ALOHA



Prós:

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

Contras:

- colisões, desperdício de slots
- slots ociosos
- os nós podem ser capazes de detectar colisões em menos tempo do que o tempo necessário para transmitir um pacote
- sincronismo de clock

Slotted ALOHA: eficiência

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

- *suposição:* N nós com muitos quadros para enviar, cada um transmite em um slot com probabilidade p
 - probabilidade que um determinado nó tenha sucesso em um slot = $p(1-p)^{N-1}$
 - probabilidade de que *qualquer* nó tenha um sucesso = $Np(1-p)^{N-1}$
 - eficiência máxima: encontre p^* que maximiza $Np(1-p)^{N-1}$
 - para muitos nós, tomando o limite de $Np^*(1-p^*)^{N-1}$ conforme N tende ao infinito, temos:

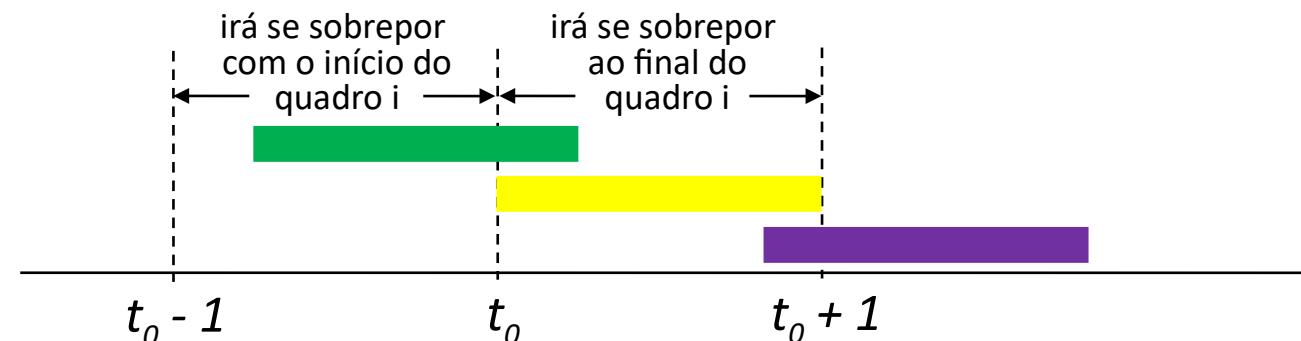
$$\text{eficiência máxima} = 1/e = .37$$

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



ALOHA puro

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



- eficiência do Aloha puro: 18% !

CSMA (carrier sense multiple access)

CSMA simples: ouça antes de transmitir:

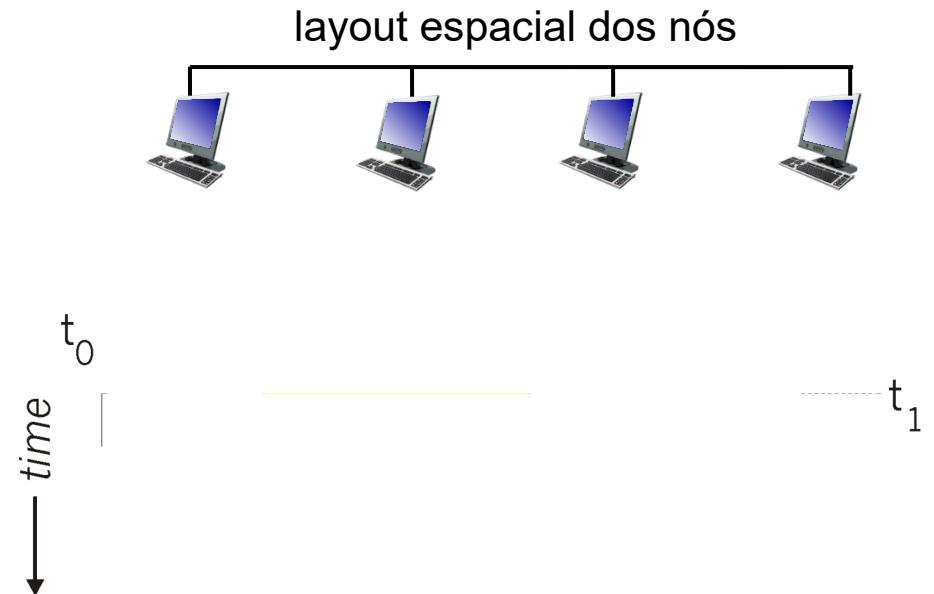
- se o canal estiver ocioso: transmita o quadro inteiro
- se o canal estiver ocupado: adie a transmissão
- analogia humana: não interrompa os outros!

CSMA/CD: CSMA com *detecção de colisão*

- colisões *detectadas* em pouco tempo
- transmissões colidindo são abortadas, reduzindo o desperdício de canal
- detecção de colisão é fácil com fio e difícil sem fio
- analogia humana: o conversador educado

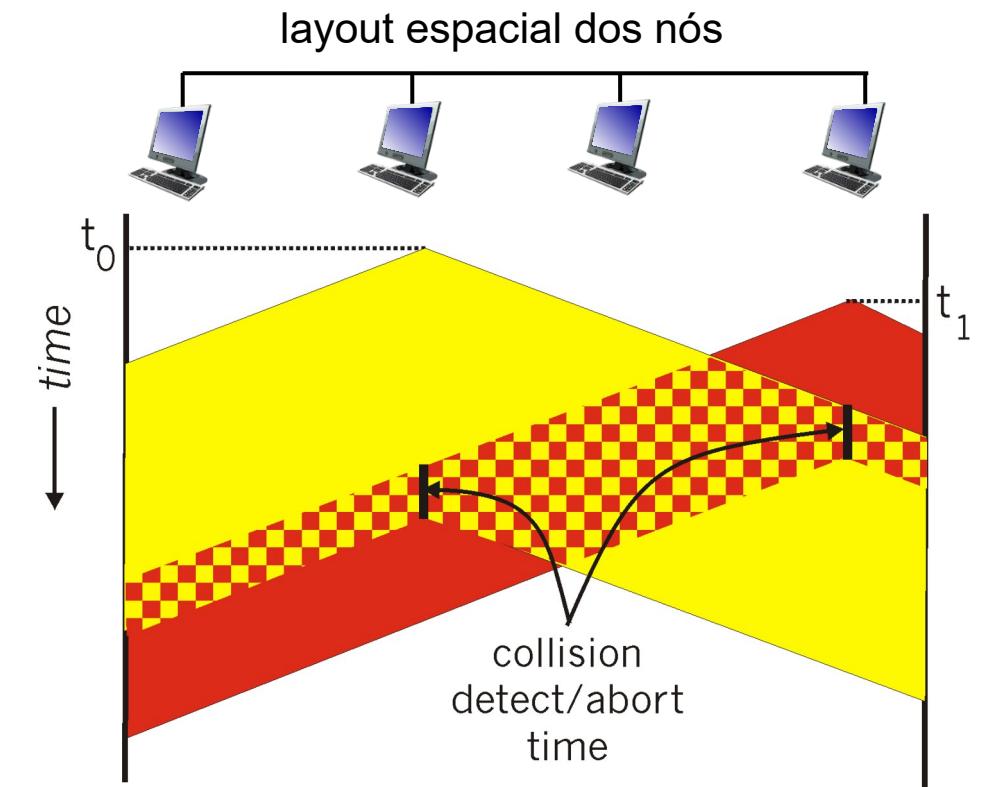
CSMA: colisões

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



CSMA/CD:

- CSMA/CD reduz a quantidade de tempo desperdiçado em colisões
 - transmissão abortada na detecção de colisão



Algoritmo do Ethernet CSMA/CD

1. Ethernet recebe datagrama da camada de rede e cria quadro
2. Se a Ethernet detectar o canal:
 - ocioso: inicia transmissão do quadro.
 - ocupado: espera até o canal ficar ocioso e então transmite
3. Se o quadro inteiro foi transmitido sem colisão – acabou!
4. Se outra transmissão foi detectada enquanto enviando: aborta e envia sinal de congestionamento
5. Após abortar, entra em *backoff binário (exponencial)*:
 - após a m -ésima colisão, escolhe um K aleatório de $\{0, 1, 2, \dots, 2^m - 1\}$. A Ethernet espera $K \cdot 512$ bits de tempo, e retorna ao Passo 2
 - mais colisões: intervalo de *backoff* mais longo

Eficiência do CSMA/CD

- t_{prop} = atraso máximo de propagação entre 2 nós na LAN
- t_{trans} = tempo para transmitir um quadro de tamanho máximo
$$= \frac{1}{1+5} /$$
- eficiência tende a 1
 - conforme t_{prop} tende a 0
 - conforme t_{trans} tende ao infinito
- melhor desempenho que ALOHA: e simples, barato, descentralizado!

Protocolos MAC de “revezamento”

protocolos MAC de particionamento de canal:

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

protocolos MAC de acesso aleatório

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

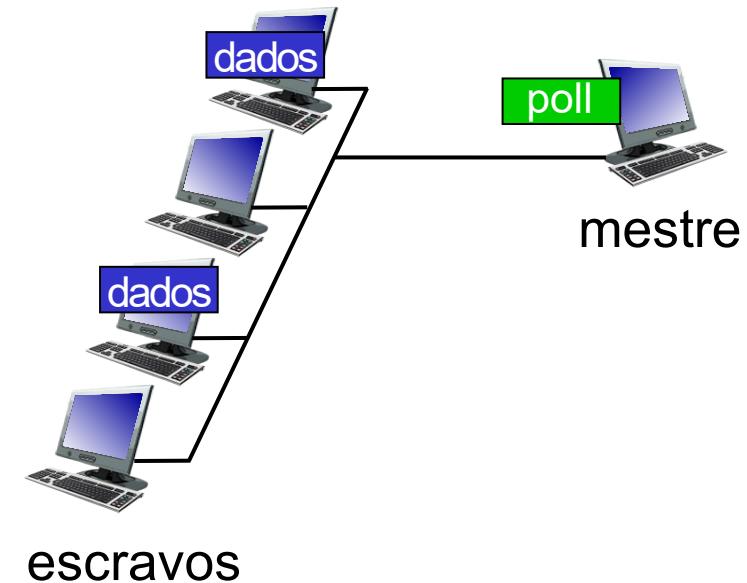
protocolos de “revezamento”

- procuram o melhor dos dois mundos!

Protocolos MAC de “revezamento”

polling:

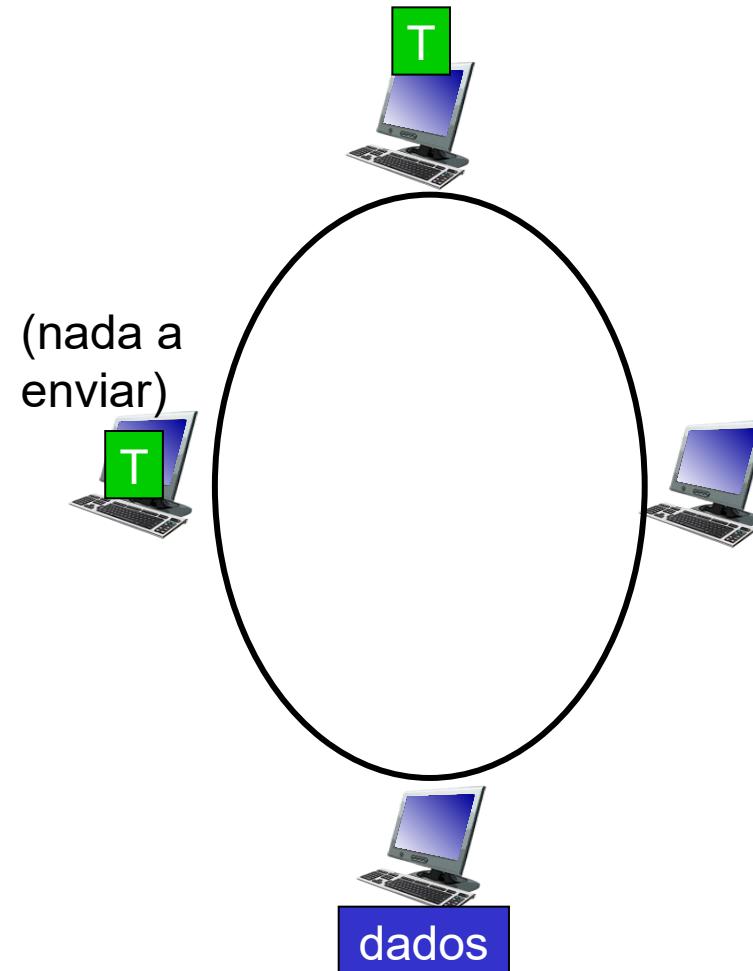
- nó mestre “convida” outros nós para transmitir em rodadas
- normalmente usado com dispositivos “burros”
- preocupações :
 - sobrecarga de polling
 - latência
 - ponto único de falha (mestre)
- Bluetooth usa polling



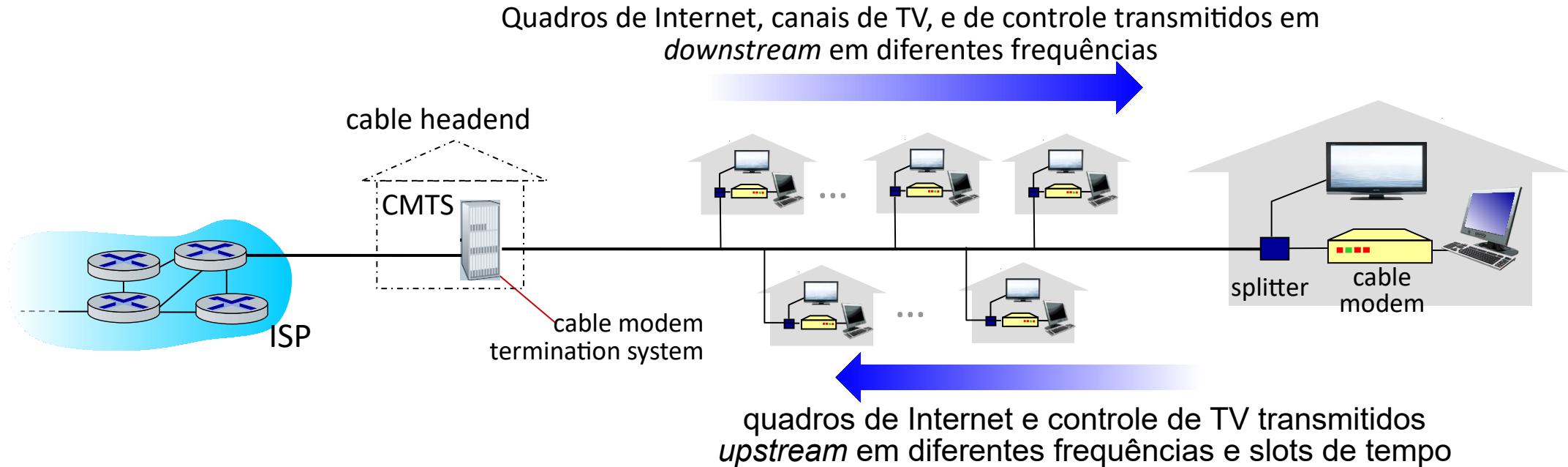
Protocolos MAC de “revezamento”

passagem de *token*:

- *token* de controle passado de um nó para o próximo, sequencialmente.
 - transmite enquanto estiver segurando o token
- preocupações:
 - sobrecarga do token
 - latência
 - ponto único de falha (token)

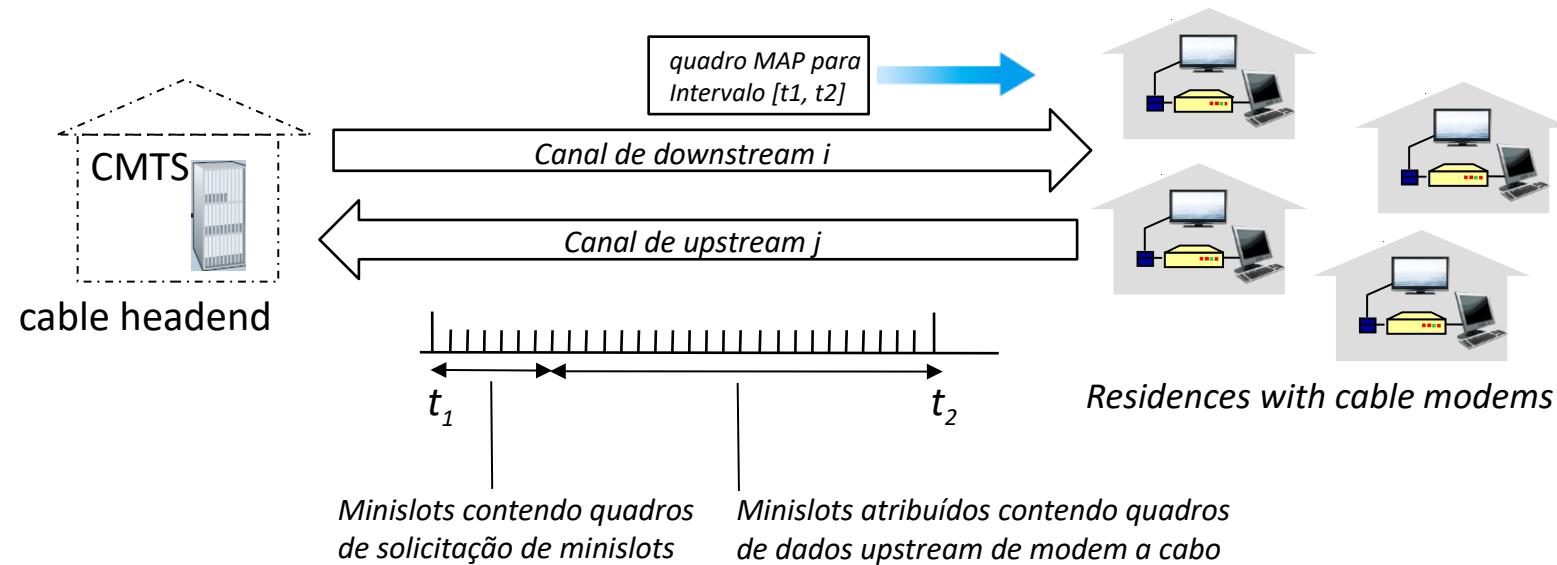


Rede de acesso a cabo: FDM, TDM e acesso aleatório!



- múltiplos canais FDM de *downstream* (broadcast): até 1,6 Gbps/canal
 - único CMTS transmite nos canais
- múltiplos canais *upstream* (até 1 Gbps/canal)
 - **acesso múltiplo:** todos os usuários disputam (acesso aleatório) por determinados intervalos de tempo do canal de *upstream*; outros são atribuídos por TDM

Rede de acesso a cabo:



DOCSIS: data over cable service interface specification
(especificação de interface para serviço de dados sobre cabo)

- FDM sobre canais de frequência *upstream* e *downstream*
- TDM no *upstream*: alguns slots atribuídos, alguns têm contenção
 - quadro MAP de *downstream*: atribui slots *upstream*
 - solicitação de slots *upstream* (e dados) transmitidos com acesso aleatório (recurso binário) em slots selecionados

Camada de enlace, LANs: roteiro

- introdução
- detecção e correção de erros
- protocolos de acesso múltiplo
- **LANs**
 - endereçamento, ARP
 - Ethernet
 - comutadores (*switches*)
 - VLANs
- virtualização de enlaces: MPLS
- redes de *data center*



- um dia na vida de uma requisição web

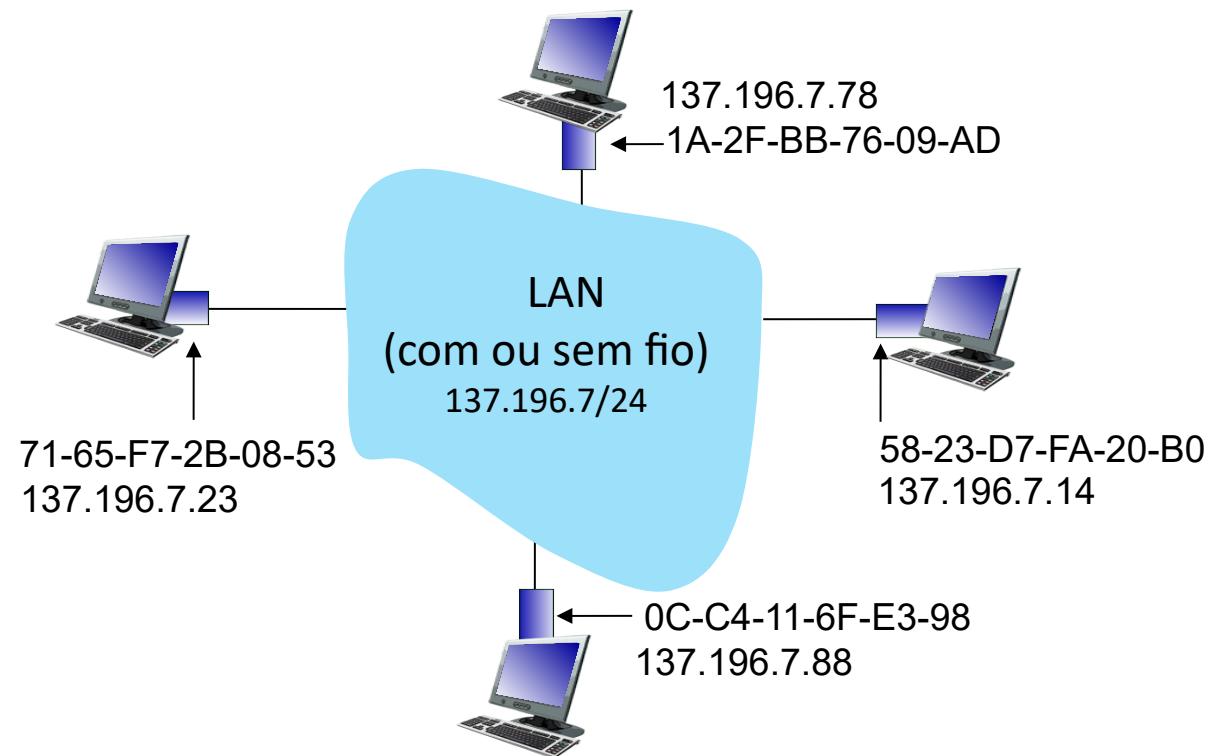
Endereços MAC

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

Endereços MAC

cada interface na LAN

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



Endereços MAC

- alocação de endereços MAC administrada pelo IEEE
- fabricante compra parte do espaço de endereço MAC (para garantir exclusividade)
- analogia:
 - endereço MAC: como o CPF
 - endereço IP: como o endereço postal
- endereço MAC fixo: portabilidade
 - pode-se mover a interface de uma LAN para outra
 - lembre-se de que os endereços IP *não* são portáveis: dependem da sub-rede IP à qual o nó está conectado



Administrador: Windows Pow X + | v

- □ X

PS C:\Users\fbrev> ipconfig /all

Configuração de IP do Windows

Nome do host : DONALD
Sufixo DNS primário :
Tipo de nó : híbrido
Roteamento de IP ativado. : não
Proxy WINS ativado. : não

Adaptador Ethernet Ethernet:

Sufixo DNS específico de conexão. :
Descrição : Intel(R) I211 Gigabit Network Connection
Endereço Físico : 04-D9-F5-85-20-A2 
DHCP Habilitado : Sim
Configuração Automática Habilitada. : Sim
Endereço IPv6 : 2804:431:c7d2:b4c4:a659:f84e:c017:73bf(Preferencial)
Endereço IPv6 Temporário. : 2804:431:c7d2:b4c4:102a:cc6f:9a20:1911(Preterido)
Endereço IPv6 Temporário. : 2804:431:c7d2:b4c4:88a6:5d47:870a:6b01(Preferencial)
Endereço IPv6 de link local : fe80::8e35:86c4:f4d4:b43f%10(Preferencial)
Endereço IPv4. : 192.168.50.5(Preferencial)
Máscara de Sub-rede : 255.255.255.0
Concessão Obtida. : segunda-feira, 24 de outubro de 2022 17:07:54
Concessão Expira. : quinta-feira, 27 de outubro de 2022 02:42:50
Gateway Padrão. : fe80::1a31:ffff:fe35:4d48%10
192.168.50.1
Servidor DHCP : 192.168.50.1
IAID de DHCPv6. : 889510389
DUID de Cliente DHCPv6. : 00-01-00-01-2A-D1-FC-10-04-D9-F5-85-20-A2

ARP: address resolution protocol (protocolo de resolução de endereço)

Questão: como determinar o endereço MAC da interface, conhecendo seu endereço IP?

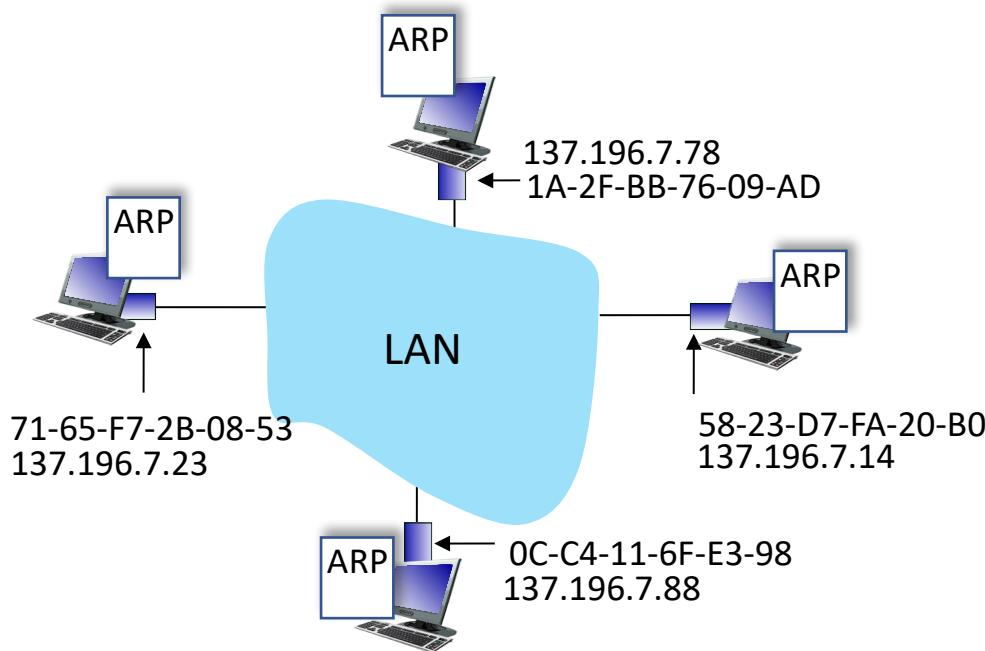


Tabela ARP: cada nó IP (hospedeiro, roteador) na LAN tem uma tabela

- Mapeamentos de endereços 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

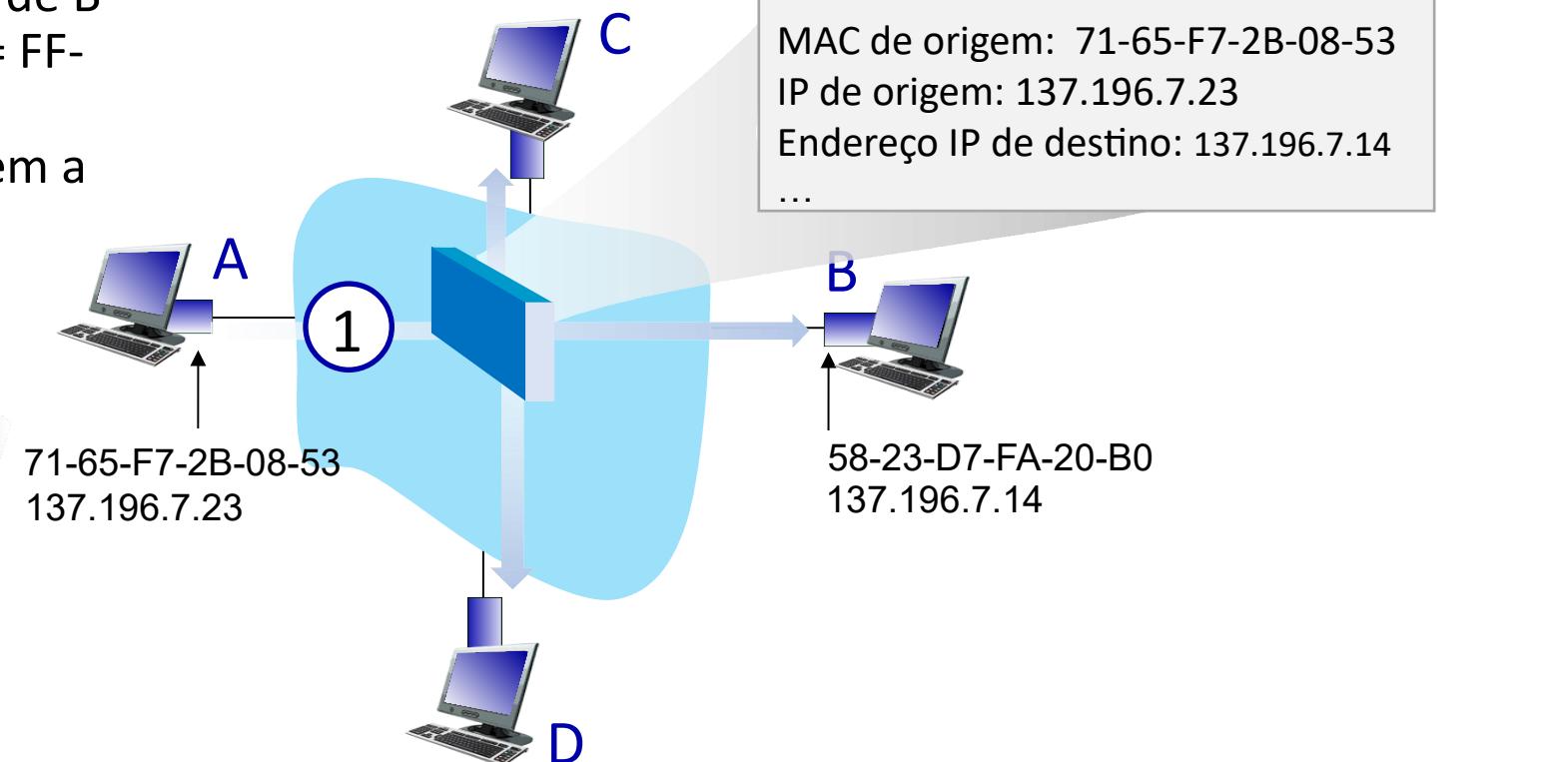
exemplo: A quer enviar datagrama para B

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

- 1 A transmite (*broadcast*) a consulta ARP, contendo o endereço IP de B
- endereço MAC de destino = FF-FF-FF-FF-FF-FF
 - todos os nós na LAN recebem a consulta ARP

Tabela ARP em A

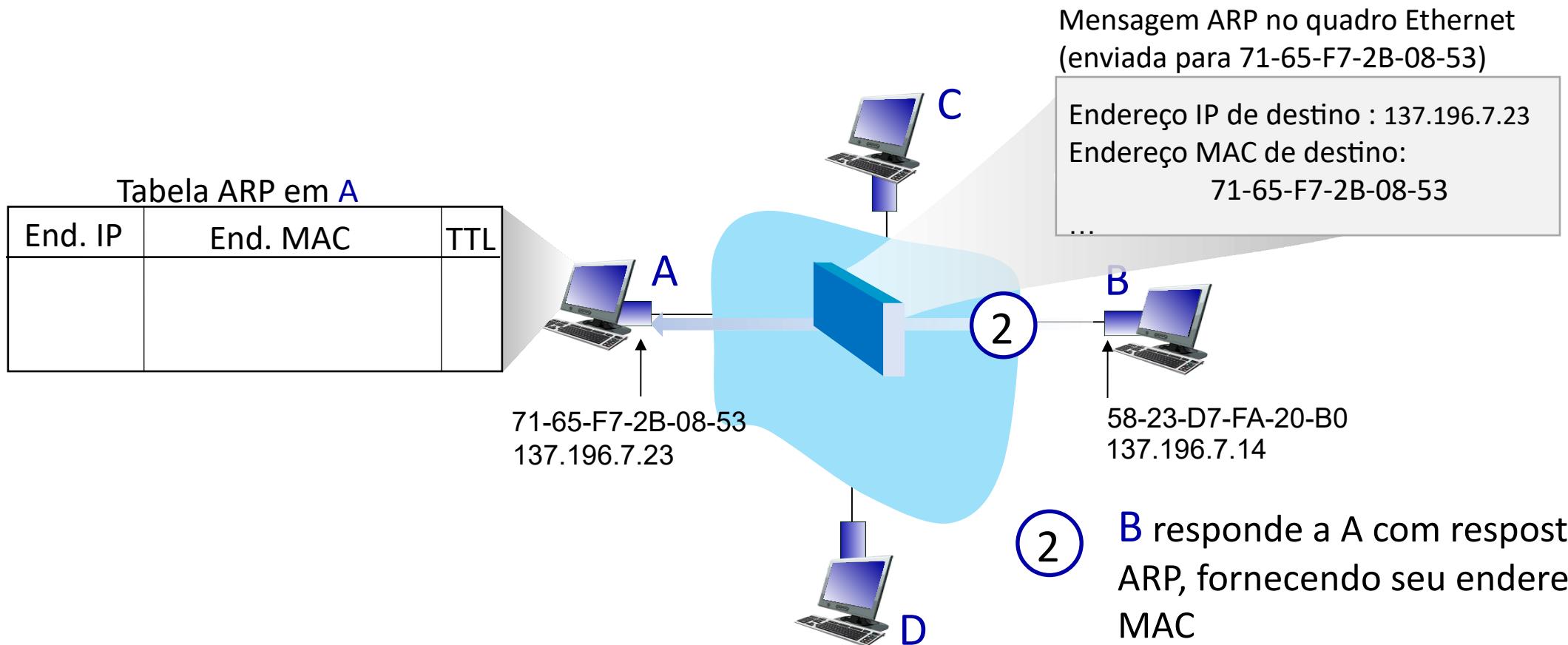
End. IP	End. MAC	TTL



Protocolo ARP em ação

exemplo: A quer enviar datagrama para B

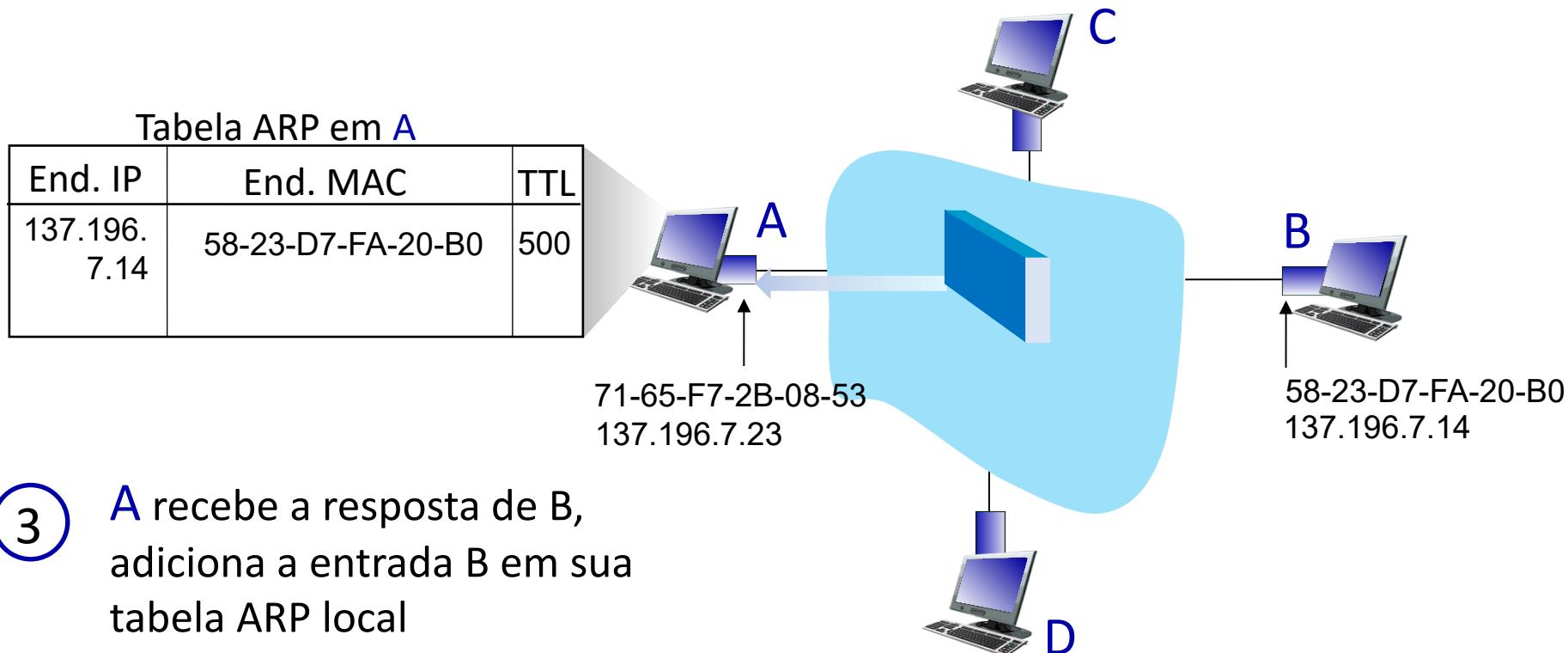
- O endereço MAC de B não está na tabela ARP de A, então A usa o ARP para encontrar o endereço MAC de B



Protocolo ARP em ação

exemplo: A quer enviar datagrama para B

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



```
PS C:\Users\fbrev> arp -a

Interface: 192.168.50.74 --- 0x6
Endereço IP      Endereço físico      Tipo
192.168.50.1    18-31-bf-35-4d-48  dinâmico
192.168.50.2    d8-50-e6-d0-f6-40  dinâmico
192.168.50.5    04-d9-f5-85-20-a2  dinâmico
192.168.50.17   00-11-32-2a-1d-3c  dinâmico
192.168.50.45   54-60-09-ad-cf-8c  dinâmico
192.168.50.58   00-11-32-8b-8d-47  dinâmico
192.168.50.61   f0-5c-77-03-ec-ca  dinâmico
192.168.50.89   70-77-81-fe-b4-9b  dinâmico
192.168.50.100  74-d0-2b-2c-8a-9d  dinâmico
192.168.50.107  70-74-14-01-84-71  dinâmico
192.168.50.115  48-b0-2d-65-84-6d  dinâmico
192.168.50.132  8c-fa-ba-89-9e-31  dinâmico
192.168.50.158  3c-07-71-01-9b-20  dinâmico
192.168.50.159  a8-13-74-7b-d8-a9  dinâmico
192.168.50.160  44-09-b8-1e-b2-95  dinâmico
192.168.50.163  04-5d-4b-f0-55-bc  dinâmico
192.168.50.165  a8-9c-ed-8a-34-8a  dinâmico
192.168.50.178  a4-77-33-38-14-b0  dinâmico
192.168.50.245  b0-4f-13-f0-9b-76  dinâmico
192.168.50.249  00-a0-de-fd-a2-2e  dinâmico
192.168.50.255  ff-ff-ff-ff-ff-ff  estático
224.0.0.22      01-00-5e-00-00-16  estático
224.0.0.251     01-00-5e-00-00-fb  estático
224.0.0.252     01-00-5e-00-00-fc  estático
239.255.255.250 01-00-5e-7f-ff-fa  estático
239.255.255.253 01-00-5e-7f-ff-fd  estático
255.255.255.255 ff-ff-ff-ff-ff-ff  estático

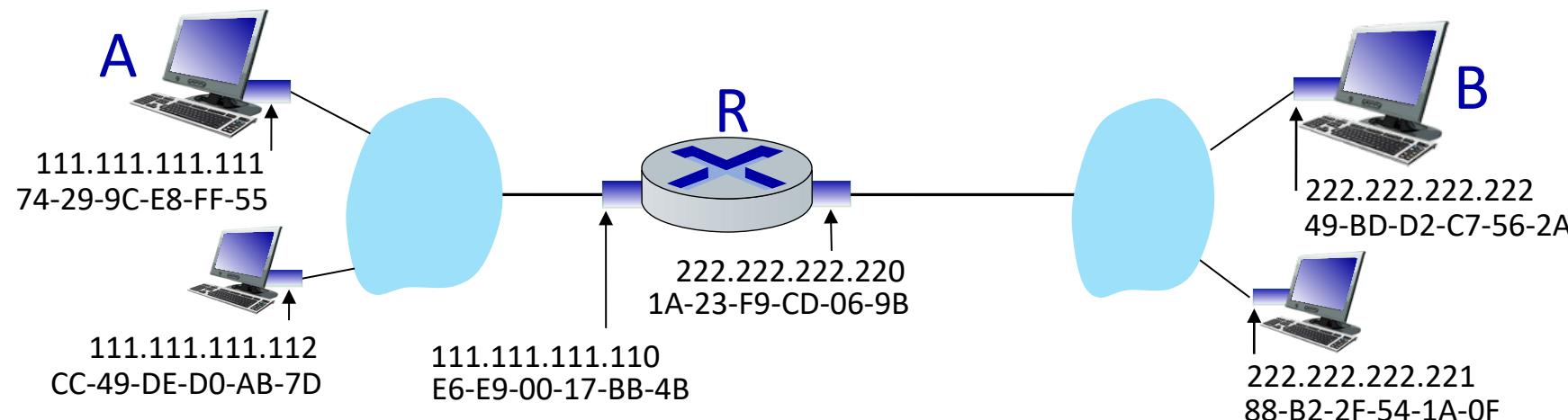
Interface: 192.168.50.5 --- 0xa
Endereço IP      Endereço físico      Tipo
192.168.50.1    18-31-bf-35-4d-48  dinâmico
192.168.50.17   00-11-32-2a-1d-3c  dinâmico
192.168.50.45   54-60-09-ad-cf-8c  dinâmico
192.168.50.58   00-11-32-8b-8d-47  dinâmico
192.168.50.61   f0-5c-77-03-ec-ca  dinâmico
192.168.50.74   04-d9-f5-85-20-79  dinâmico
192.168.50.89   70-77-81-fe-b4-9b  dinâmico
192.168.50.100  74-d0-2b-2c-8a-9d  dinâmico
192.168.50.115  48-b0-2d-65-84-6d  dinâmico
192.168.50.132  8c-fa-ba-89-9e-31  dinâmico
192.168.50.160  44-09-b8-1e-b2-95  dinâmico
192.168.50.163  04-5d-4b-f0-55-bc  dinâmico
192.168.50.178  a4-77-33-38-14-b0  dinâmico
192.168.50.245  b0-4f-13-f0-9b-76  dinâmico
192.168.50.249  00-a0-de-fd-a2-2e  dinâmico
192.168.50.255  ff-ff-ff-ff-ff-ff  estático
224.0.0.22      01-00-5e-00-00-16  estático
224.0.0.251     01-00-5e-00-00-fb  estático
224.0.0.252     01-00-5e-00-00-fc  estático
239.255.255.250 01-00-5e-7f-ff-fa  estático
239.255.255.253 01-00-5e-7f-ff-fd  estático
255.255.255.255 ff-ff-ff-ff-ff-ff  estático
```

Interface: 192.168.50.245 --- 0xe		
Endereço IP	Endereço físico	Tipo
192.168.50.1	18-31-bf-35-4d-48	dinâmico
192.168.50.2	d8-50-e6-d0-f6-40	dinâmico
192.168.50.5	04-d9-f5-85-20-a2	dinâmico
192.168.50.17	00-11-32-2a-1d-3c	dinâmico
192.168.50.45	54-60-09-ad-cf-8c	dinâmico
192.168.50.58	00-11-32-8b-8d-47	dinâmico
192.168.50.61	f0-5c-77-03-ec-ca	dinâmico
192.168.50.89	70-77-81-fe-b4-9b	dinâmico
192.168.50.100	74-d0-2b-2c-8a-9d	dinâmico
192.168.50.107	70-74-14-01-84-71	dinâmico
192.168.50.115	48-b0-2d-65-84-6d	dinâmico
192.168.50.132	8c-fa-ba-89-9e-31	dinâmico
192.168.50.158	3c-07-71-01-9b-20	dinâmico
192.168.50.159	a8-13-74-7b-d8-a9	dinâmico
192.168.50.160	44-09-b8-1e-b2-95	dinâmico
192.168.50.163	04-5d-4b-f0-55-bc	dinâmico
192.168.50.165	a8-9c-ed-8a-34-8a	dinâmico
192.168.50.178	a4-77-33-38-14-b0	dinâmico
192.168.50.245	b0-4f-13-f0-9b-76	dinâmico
192.168.50.249	00-a0-de-fd-a2-2e	dinâmico
192.168.50.255	ff-ff-ff-ff-ff-ff	estático
224.0.0.22	01-00-5e-00-00-16	estático
224.0.0.251	01-00-5e-00-00-fb	estático
224.0.0.252	01-00-5e-00-00-fc	estático
239.255.255.250	01-00-5e-7f-ff-fa	estático
239.255.255.253	01-00-5e-7f-ff-fd	estático
255.255.255.255	ff-ff-ff-ff-ff-ff	estático
Interface: 169.254.161.193 --- 0x14		
Endereço IP	Endereço físico	Tipo
169.254.255.255	ff-ff-ff-ff-ff-ff	estático
224.0.0.22	01-00-5e-00-00-16	estático
224.0.0.251	01-00-5e-00-00-fb	estático
224.0.0.252	01-00-5e-00-00-fc	estático
239.255.255.250	01-00-5e-7f-ff-fa	estático
239.255.255.253	01-00-5e-7f-ff-fd	estático
Interface: 192.168.50.5 --- 0xa		
Endereço IP	Endereço físico	Tipo
192.168.50.1	18-31-bf-35-4d-48	dinâmico
192.168.50.17	00-11-32-2a-1d-3c	dinâmico
192.168.50.45	54-60-09-ad-cf-8c	dinâmico
192.168.50.58	00-11-32-8b-8d-47	dinâmico
192.168.50.61	f0-5c-77-03-ec-ca	dinâmico
192.168.50.74	04-d9-f5-85-20-79	dinâmico
192.168.50.89	70-77-81-fe-b4-9b	dinâmico
192.168.50.100	74-d0-2b-2c-8a-9d	dinâmico
192.168.50.115	48-b0-2d-65-84-6d	dinâmico
192.168.50.132	8c-fa-ba-89-9e-31	dinâmico
192.168.50.160	44-09-b8-1e-b2-95	dinâmico
192.168.50.163	04-5d-4b-f0-55-bc	dinâmico
192.168.50.178	a4-77-33-38-14-b0	dinâmico
192.168.50.245	b0-4f-13-f0-9b-76	dinâmico
192.168.50.249	00-a0-de-fd-a2-2e	dinâmico
192.168.50.255	ff-ff-ff-ff-ff-ff	estático
224.0.0.22	01-00-5e-00-00-16	estático
224.0.0.251	01-00-5e-00-00-fb	estático
224.0.0.252	01-00-5e-00-00-fc	estático
239.255.255.250	01-00-5e-7f-ff-fa	estático

Roteamento para outra sub-rede: endereçamento

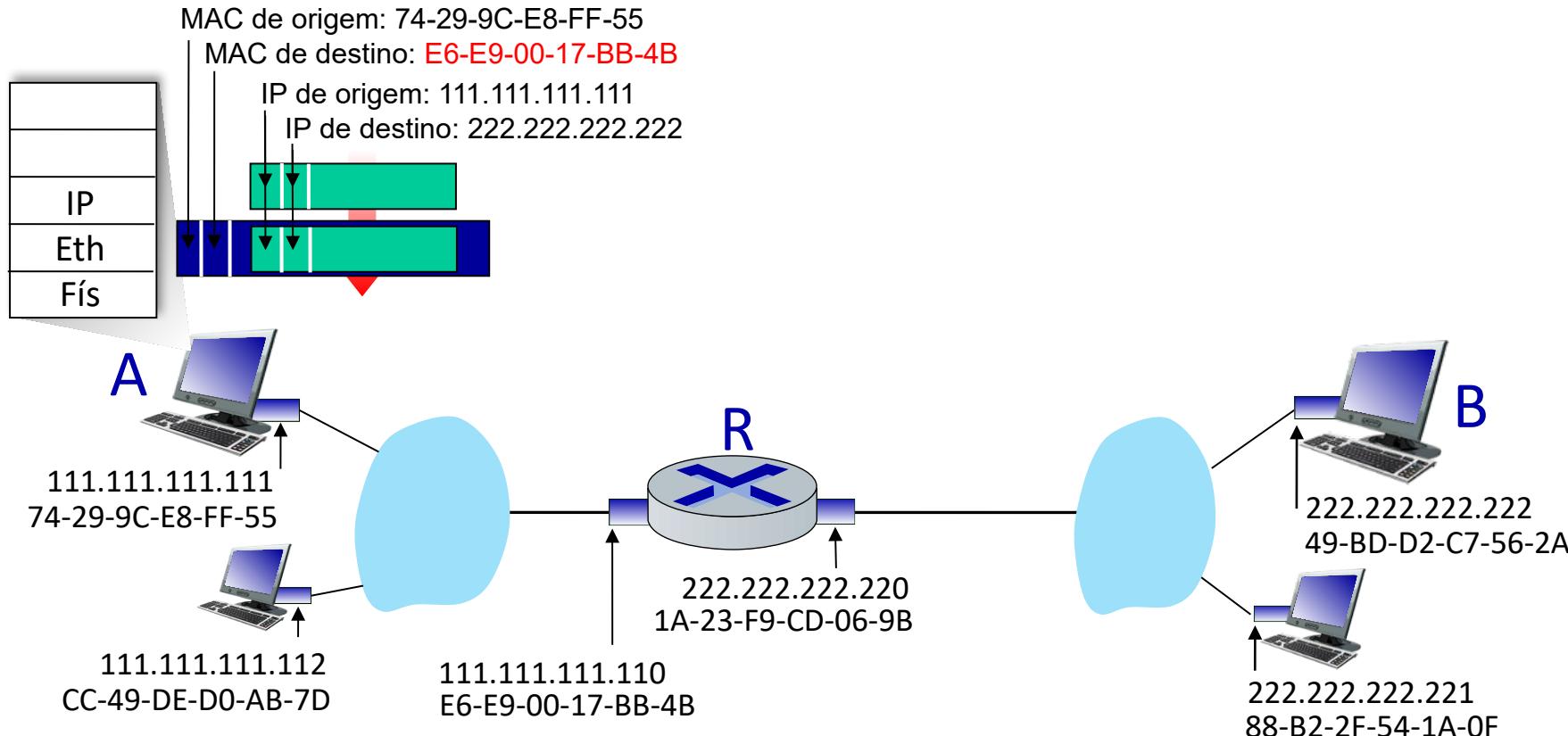
passo a passo: enviando um datagrama de *A* para *B* via *R*

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



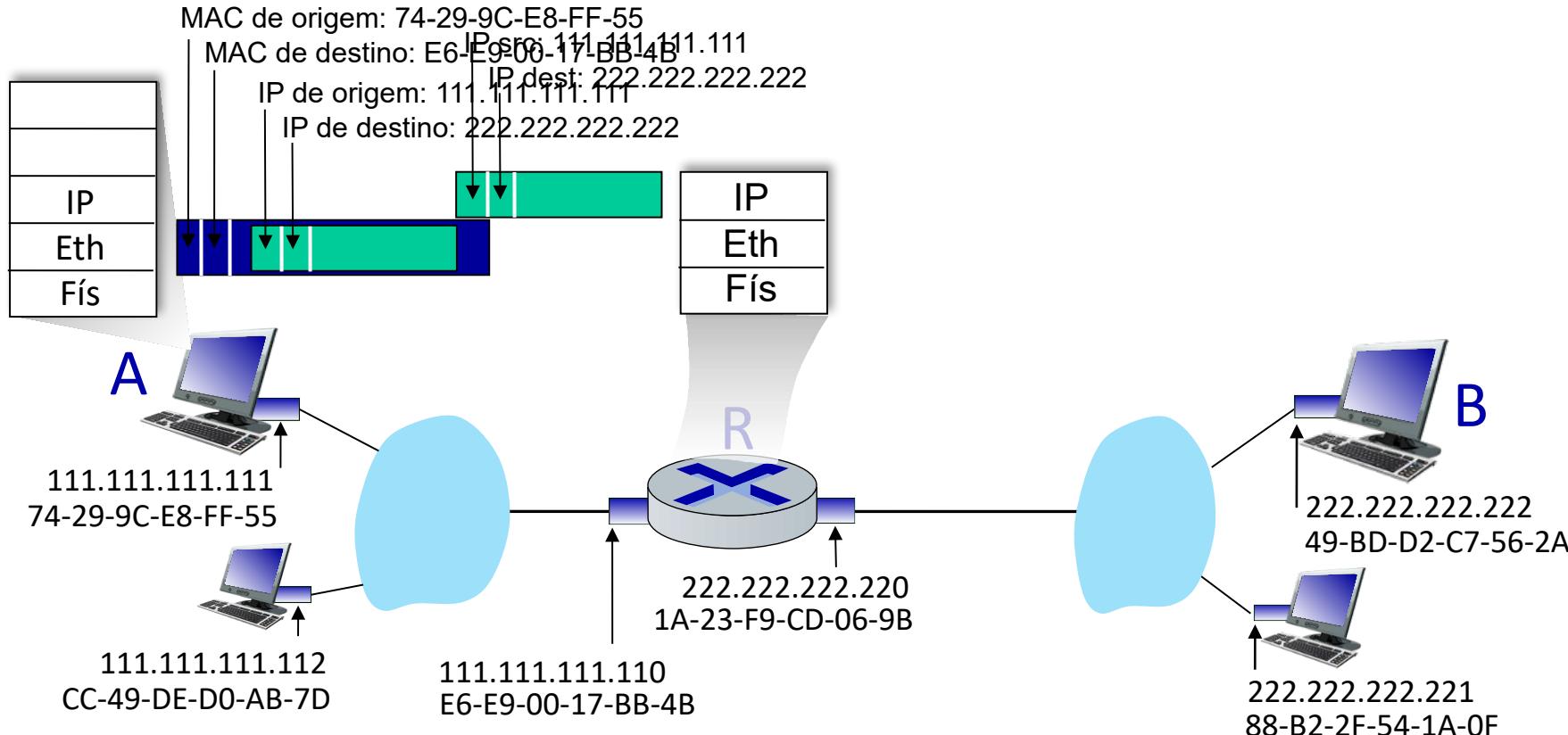
Roteamento para outra sub-rede: endereçamento

- A cria datagrama IP com IP de origem A e IP de destino B
- A cria um quadro de camada de enlace contendo datagrama IP A-para-B
 - o endereço MAC de **R** é o destino do quadro



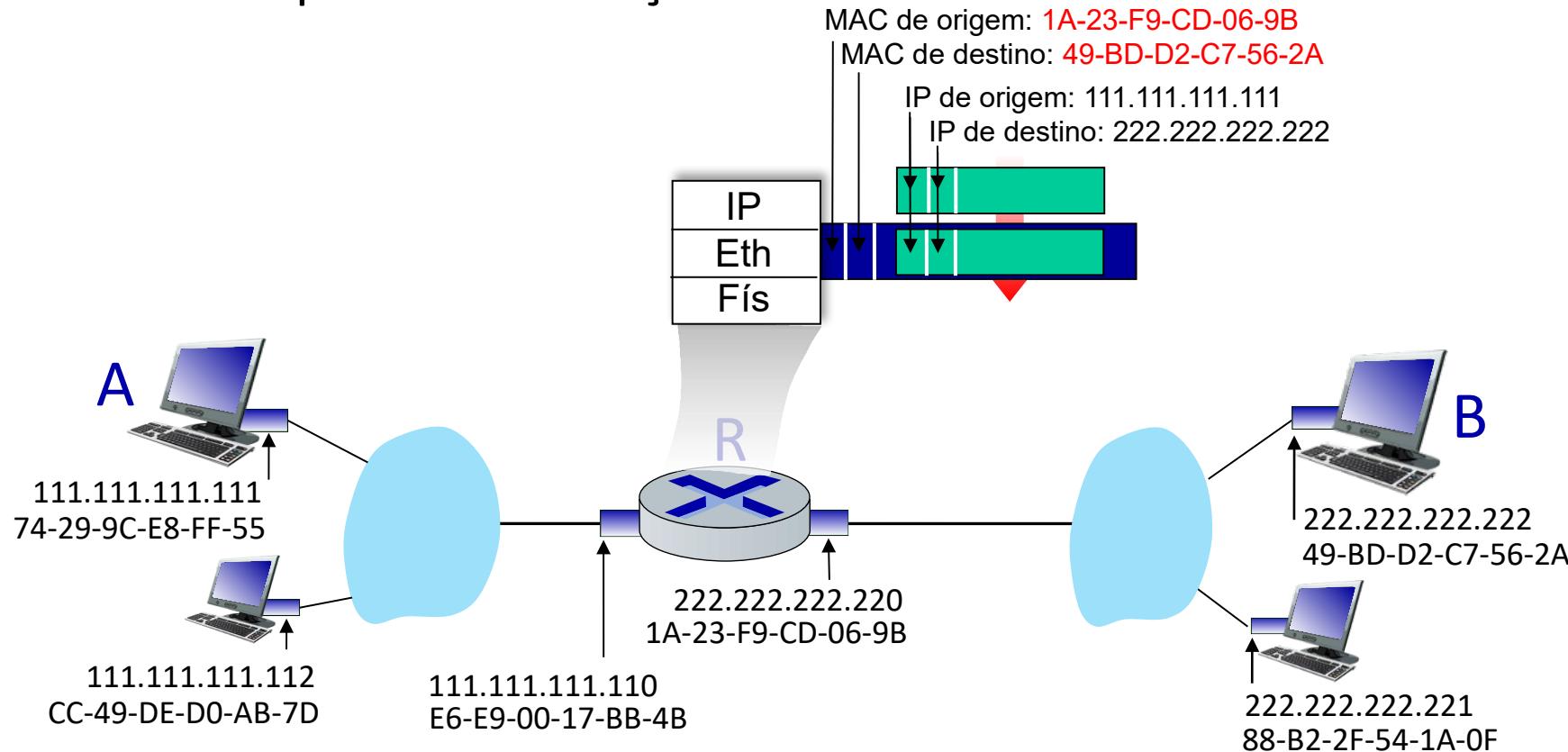
Roteamento para outra sub-rede: endereçamento

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



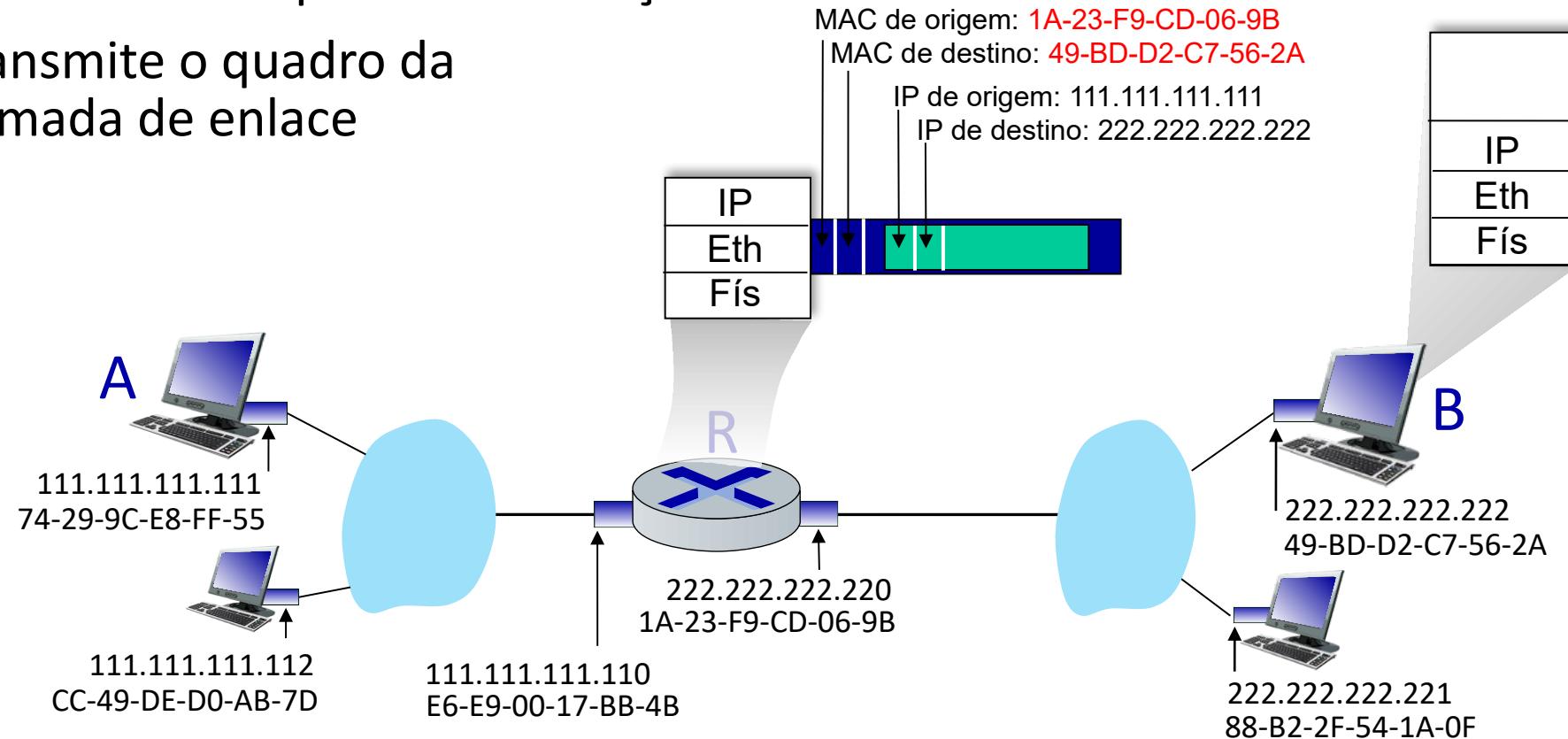
Roteamento para outra sub-rede: endereçamento

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



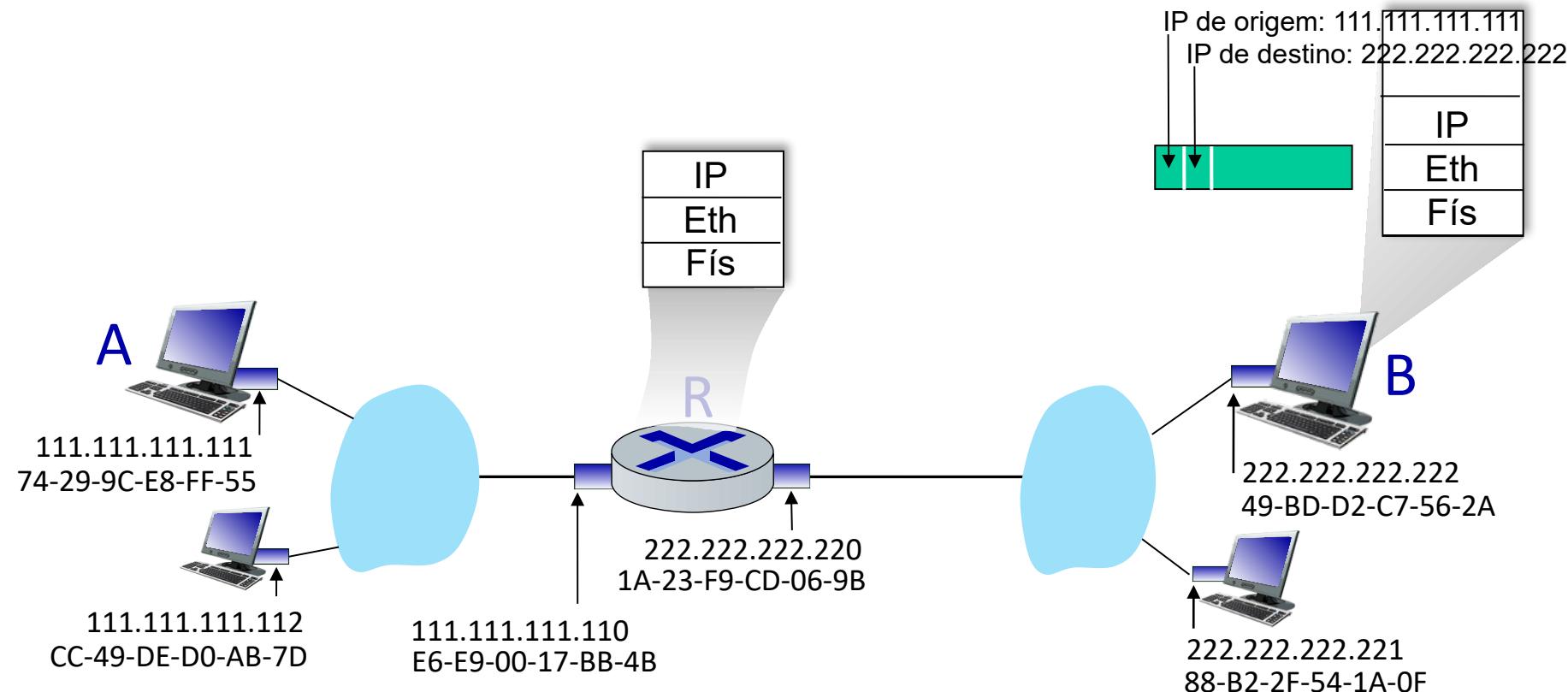
Roteamento para outra sub-rede: endereçamento

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



Roteamento para outra sub-rede: endereçamento

- B recebe quadro, extrai datagrama IP com destino a B
- B passa datagrama para cima na pilha de protocolos, para IP



Camada de enlace, LANs: roteiro

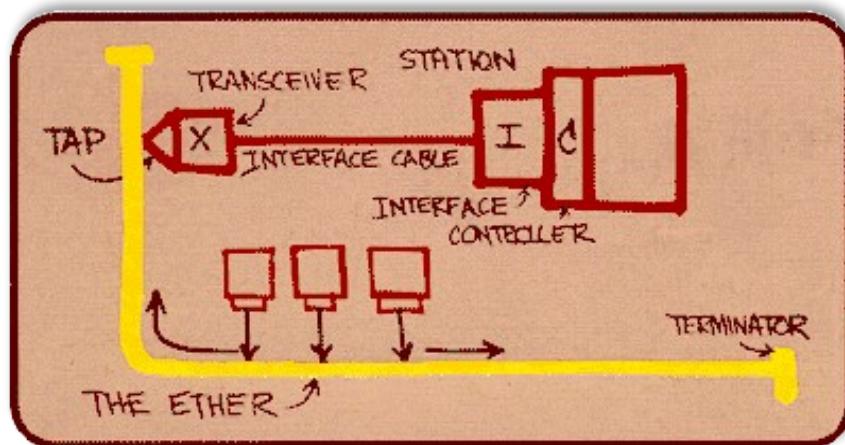
- introdução
 - detecção e correção de erros
 - protocolos de acesso múltiplo
 - **LANs**
 - endereçamento, ARP
 - **Ethernet**
 - comutadores (*switches*)
 - VLANs
 - virtualização de enlaces: MPLS
 - redes de *data center*
- 
- um dia na vida de uma requisição web

Ethernet

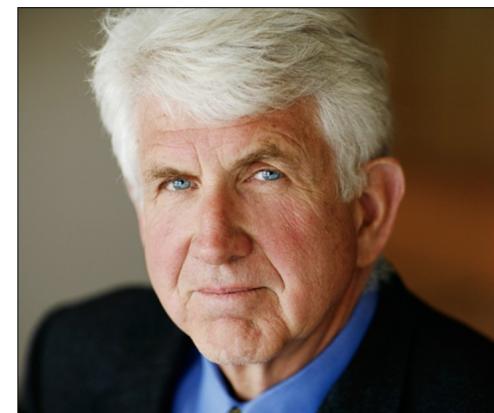
tecnologia LAN com fio “dominante”:

- primeira tecnologia LAN amplamente utilizada
- mais simples e barata
- acompanhar a corrida de velocidade: 10 Mbps – 800 Gbps
- chip único, várias velocidades (por exemplo, Broadcom BCM5761)

Esboço da Ethernet de Metcalfe



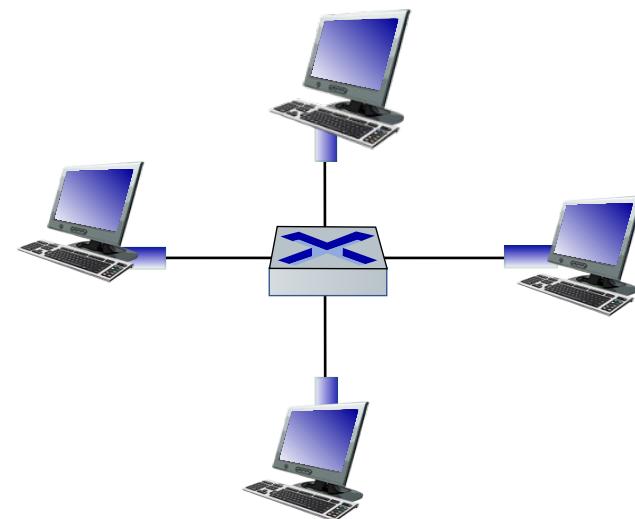
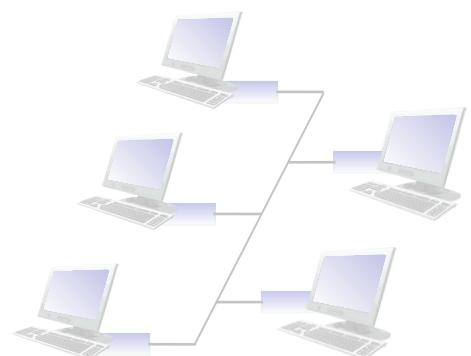
Bob Metcalfe: co-inventor da Ethernet,
Vencedor do Prêmio ACM Turing 2022



Ethernet: topologia física

- **barramento:** popular até meados dos anos 90
 - todos os nós no mesmo domínio de colisão (podem colidir uns com os outros)
- **Comutada (estrela):** prevalece hoje
 - *comutador (switch)* de camada de enlace (camada 2) ativo no centro
 - cada “raio” executa um protocolo Ethernet (separado) (os nós não colidem uns com os outros)

barramento:
cabô coaxial



comutada

Estrutura do quadro Ethernet

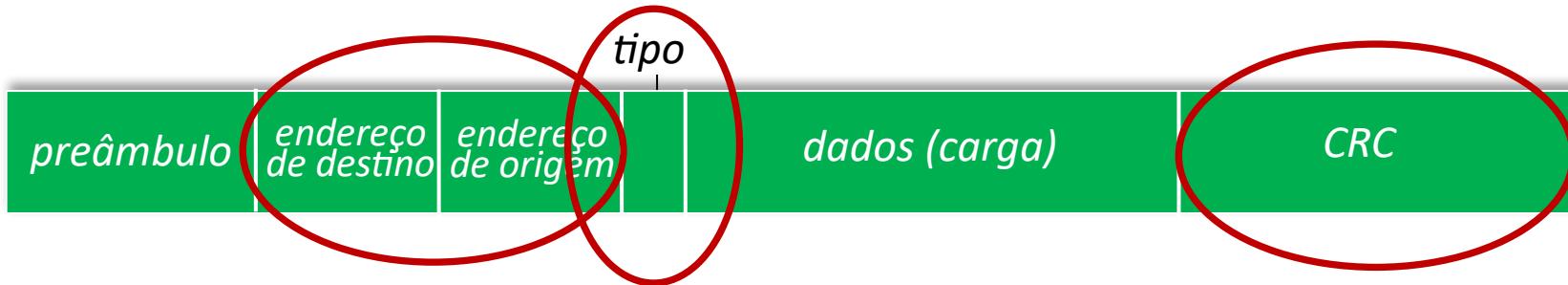
interface de envio encapsula datagrama IP (ou outro pacote de protocolo da camada de rede) em um **quadro Ethernet**



preâmbulo:

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

Estrutura do quadro Ethernet (mais)



- **endereços:** endereços MAC de origem e destino de 6 bytes
 - se o adaptador recebe o quadro com endereço de destino correspondente ao seu ou com endereço de difusão (*broadcast*) (por exemplo, pacote ARP), ele passa os dados do quadro para o protocolo da camada de rede
 - caso contrário, o adaptador descarta o quadro
- **tipo:** indica 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 (**cyclic redundancy check**) no receptor
 - erro detectado: o quadro é descartado

Quadro Ethernet II vs. Quadro IEEE 802.3

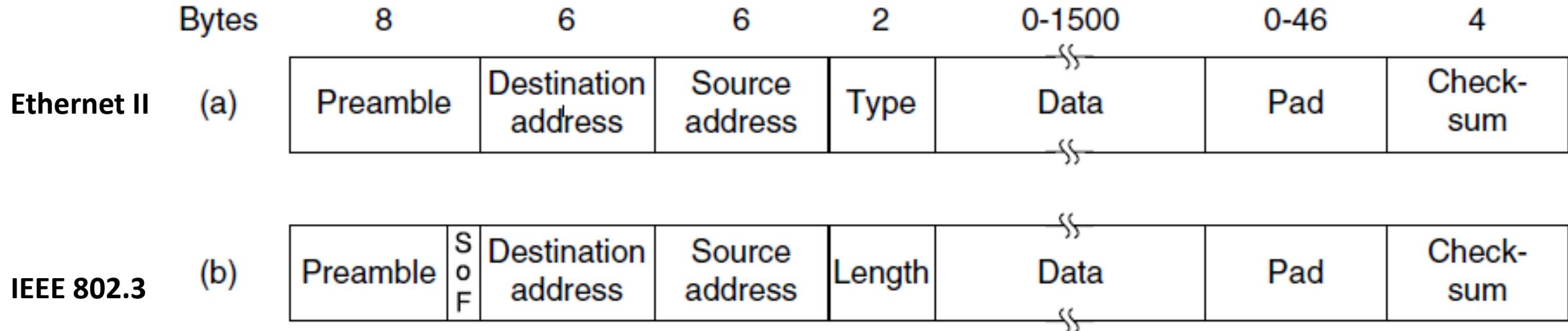
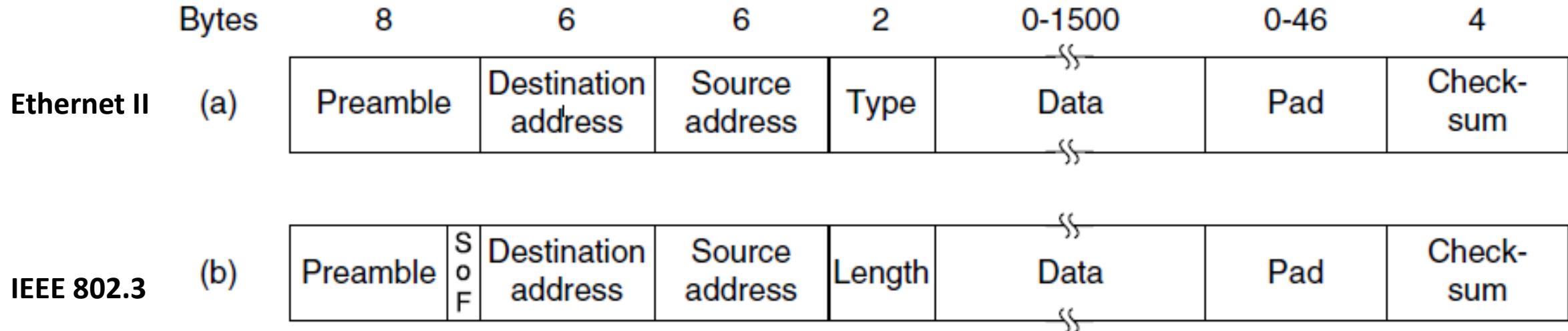


Imagen extraída de Tanenbaum, Andrew S., 1944-
Computer networks / Andrew S. Tanenbaum, David J. Wetherall. -- 5th ed.

- IEEE 802.3x-1997 unificou os padrões:
 - Valores ≤ 1500 representam tamanho
 - Valores ≥ 1536 representam tipo
 - Tamanhos em Bytes

Quadro Ethernet II vs. Quadro IEEE 802.3



- **Preamble:** 7 bytes (56 bits) – usado para sincronização.
- **SFD (Start Frame Delimiter):** 1 byte (8 bits).
- **Endereços MAC (Destino + Origem):** $6 + 6 = 12$ bytes (96 bits).
- **Length/Type:** 2 bytes (16 bits).
- **Payload (dados):** entre 46 e 1500 bytes (mínimo e máximo permitidos).
- **Pad:** Campo de preenchimento para tamanho mínimo (se necessário)
- **FCS (Frame Check Sequence):** 4 bytes (32 bits).

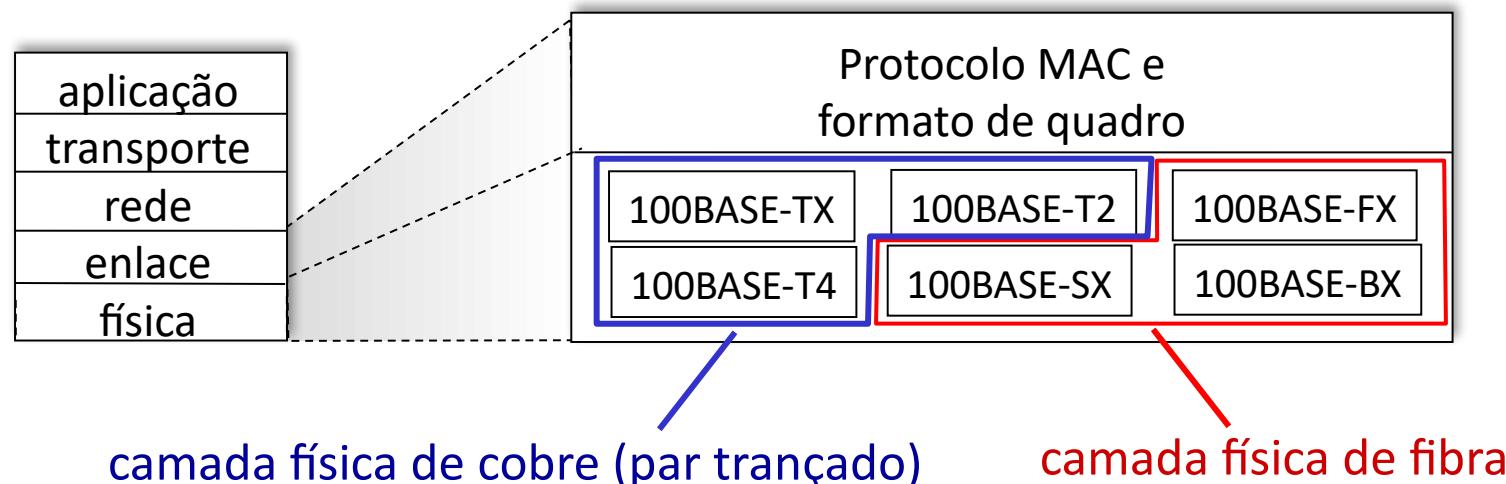
Imagen extraída de Tanenbaum, Andrew S., 1944-
Computer networks / Andrew S. Tanenbaum, David J. Wetherall. -- 5th ed.

Ethernet: não confiável, sem conexão

- **sem conexão:** sem *handshaking* entre NICs emissor e receptor
- **não confiável:** NIC receptor não envia ACKs ou NAKs para NIC emissor
 - dados em quadros descartados são recuperados somente se o emissor inicial usar transferência confiável de dados em camada superior (por exemplo, TCP), caso contrário, dados descartados serão perdidos
- Protocolo MAC da Ethernet : **CSMA/CD com backoff binário e sem slots**

Padrões Ethernet 802.3: camadas de enlace e física

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



Camada de enlace, LANs: roteiro

- introdução
- detecção e correção de erros
- protocolos de acesso múltiplo
- **LANs**
 - endereçamento, ARP
 - Ethernet
 - **comutadores (*switches*)**
 - VLANs
- virtualização de enlaces: MPLS
- redes de *data center*



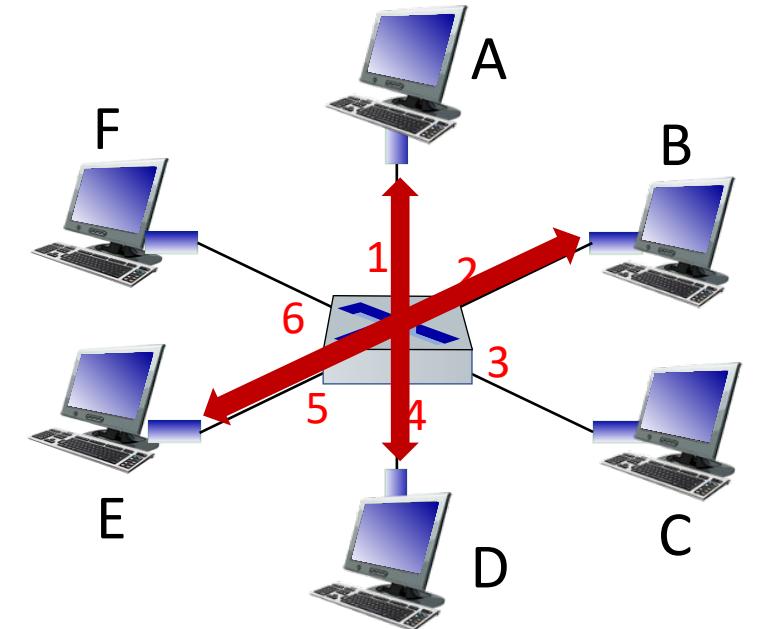
- um dia na vida de uma requisição web

Comutador Ethernet

- Comutador é um dispositivo de **camada de enlace**: assume um papel *ativo*
 - armazena e encaminha quadros Ethernet
 - examina o endereço MAC dos quadros de entrada, encaminha *seletivamente* um quadro para um ou mais enlaces de saída quando o quadro deve ser encaminhado no segmento, usa CSMA/CD para acessar o segmento
- **transparente**: hospedeiros *não sabem* da presença de comutadores
- **plug-and-play, self-learning**
 - comutadores não precisam ser configurados

Comutador: várias transmissões simultâneas

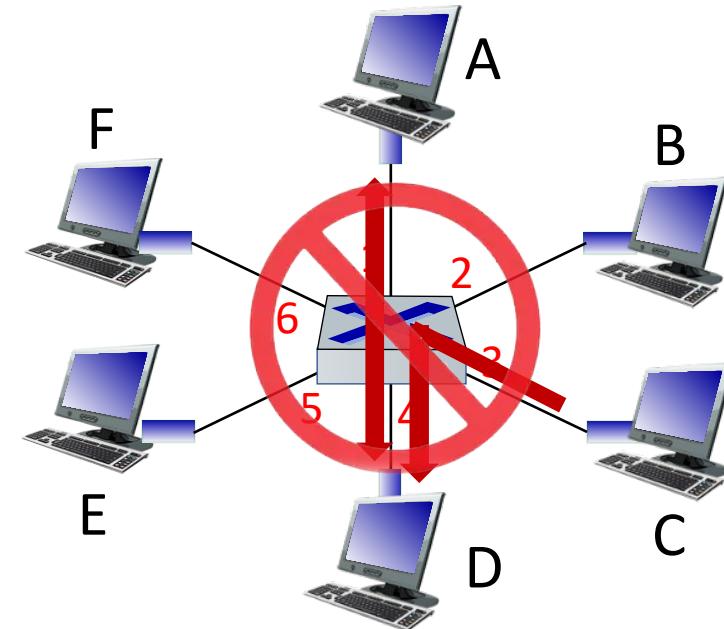
- hospedeiros têm conexão direta e dedicada com o comutador
- comutador faz *buffer* de pacotes
- protocolo Ethernet usado em cada enlace de entrada, então:
 - sem colisões; full duplex
 - cada enlace é seu próprio domínio de colisão
- **comutação:** A-para-D e B-para-E podem transmitir simultaneamente, sem colisões



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

Comutador: várias transmissões simultâneas

- hospedeiros têm conexão direta e dedicada com o comutador
- comutador faz *buffer* de pacotes
- protocolo Ethernet usado em cada enlace de entrada, então:
 - sem colisões; full duplex
 - cada enlace é seu próprio domínio de colisão
- **comutação:** A-para-D e B-para-E podem transmitir simultaneamente, sem colisões
 - mas transmissões A-para-D e C-para-D não podem ocorrer simultaneamente



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

Tabela de encaminhamento do comutador

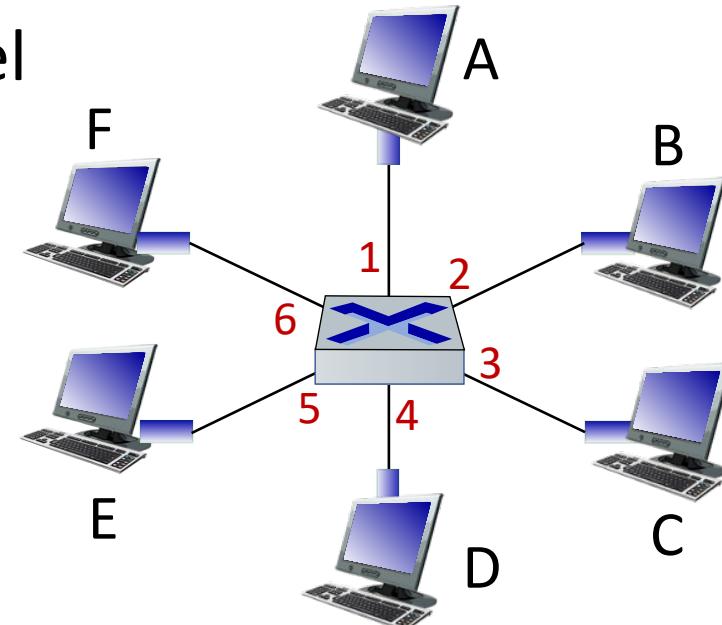
Q: como o comutador sabe que D pode ser alcançado pela interface 4 e E é alcançável pela interface 5?

R: cada comutador tem uma **tabela de comutação**, onde cada entrada é:

- (endereço MAC do hospedeiro, interface para alcançar o hospedeiro, carimbo de data/hora)
- parece uma tabela de roteamento!

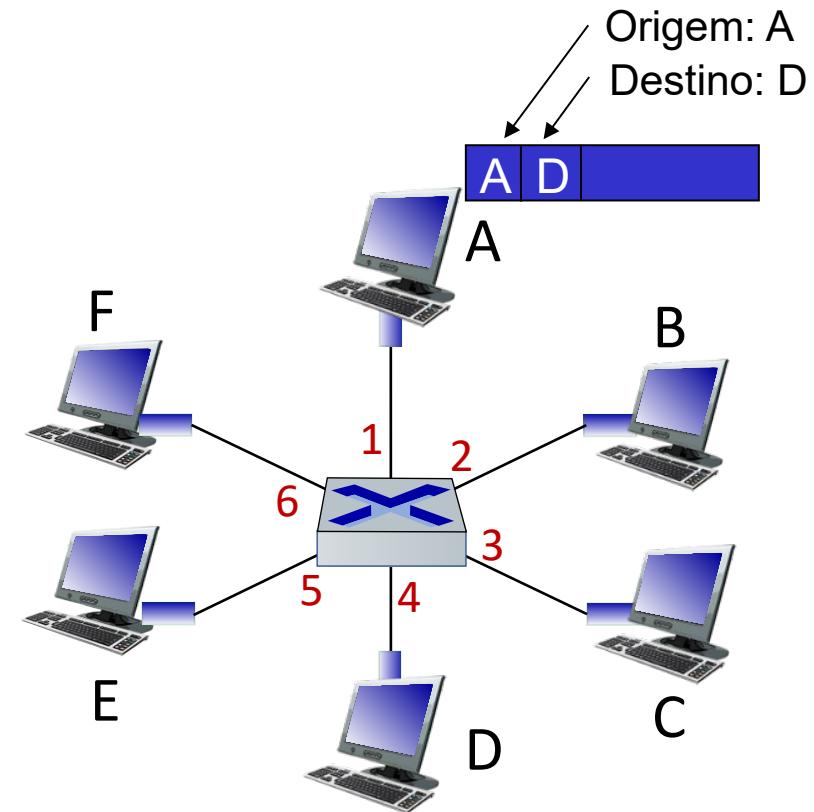
Q: como as entradas são criadas e mantidas na tabela de comutação?

- algo como um protocolo de roteamento?



Comutador: auto aprendizado

- comutador *aprende* quais hospedeiros podem ser alcançados através de quais interfaces
 - quando o quadro é recebido, o comutador “aprende” a localização do remetente: segmento de LAN de entrada
 - registra o par remetente/local na tabela de comutação



end. MAC	interface	TTL
A	1	60

Tabela de Comutação (inicialmente vazia)

Comutador: filtragem/encaminhamento de quadros

quando um quadro é recebido no comutador:

1. registra enlace de entrada e endereço MAC do hospedeiro emissor
2. procura um registro com o endereço MAC de destino na tabela de comutação

3. se um registro é encontrado para o destino
então {

 se o destino está no segmento pelo qual o quadro chegou
 então descarta quadro

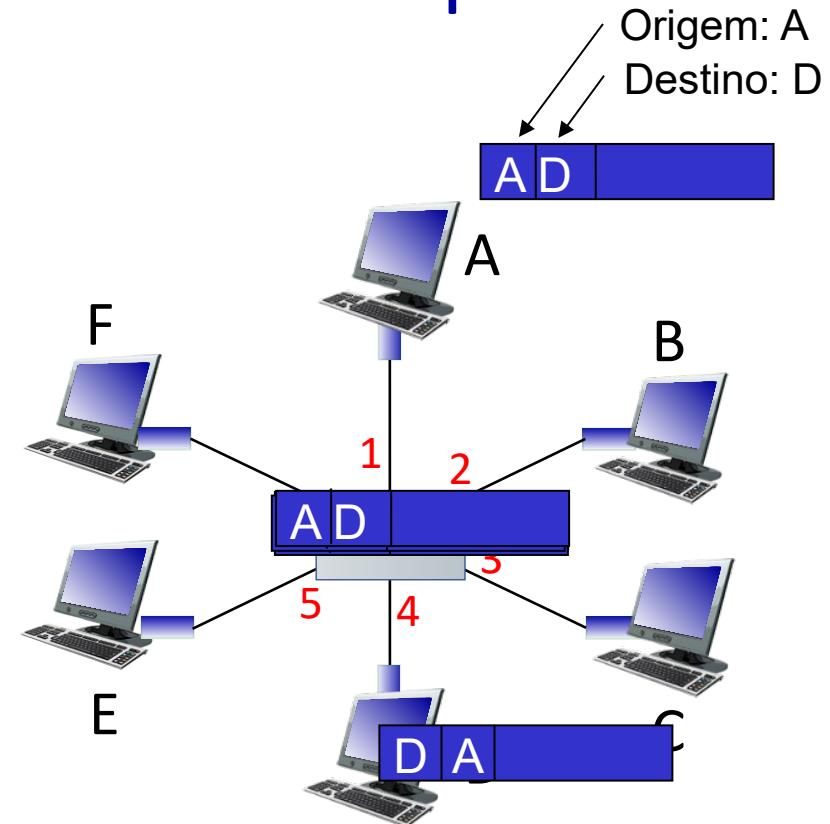
 senão encaminha quadro na interface indicada pelo registro

}

 senão inunda /* encaminha em todas as interfaces, exceto na interface de chegada */

Auto aprendizado e encaminhamento: exemplo

- destino do quadro, D, local desconhecido: **inunda**
- localização do destino A conhecida: **envia seletivamente em apenas um enlace**

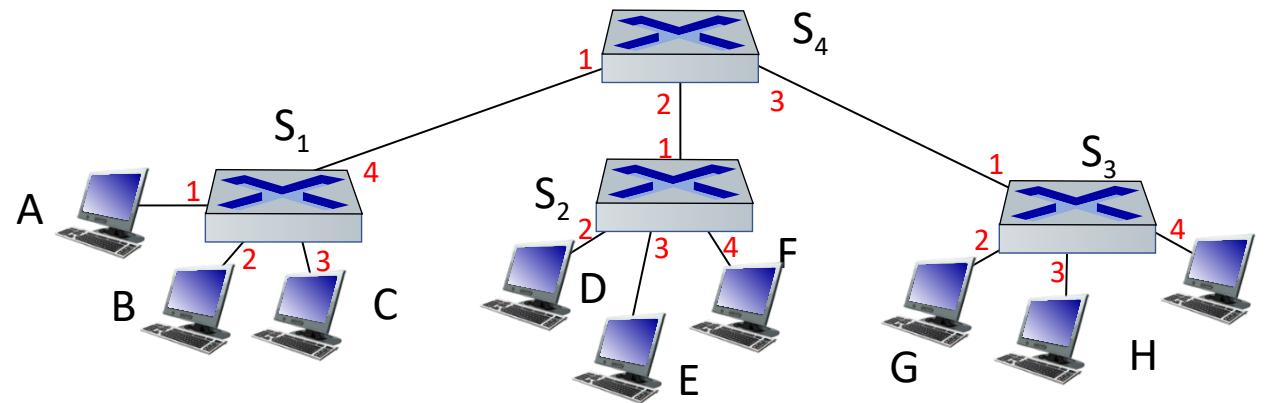


end. MAC	interface	TTL
A	1	60
D	4	60

*tabela de comutação
(inicialmente vazia)*

Interconectando comutadores

comutadores com auto aprendizado podem ser conectados:

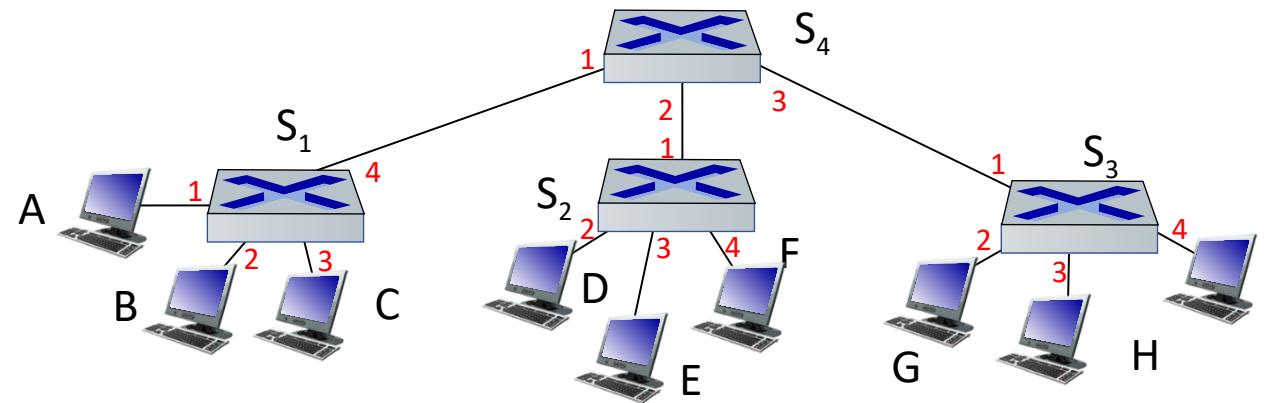


Q: enviando de A para G - como S_1 sabe que deve encaminhar o quadro destinado a G via S_4 e S_3 ?

- **R:** auto aprendizado! (funciona exatamente igual ao caso com um único comutador!)

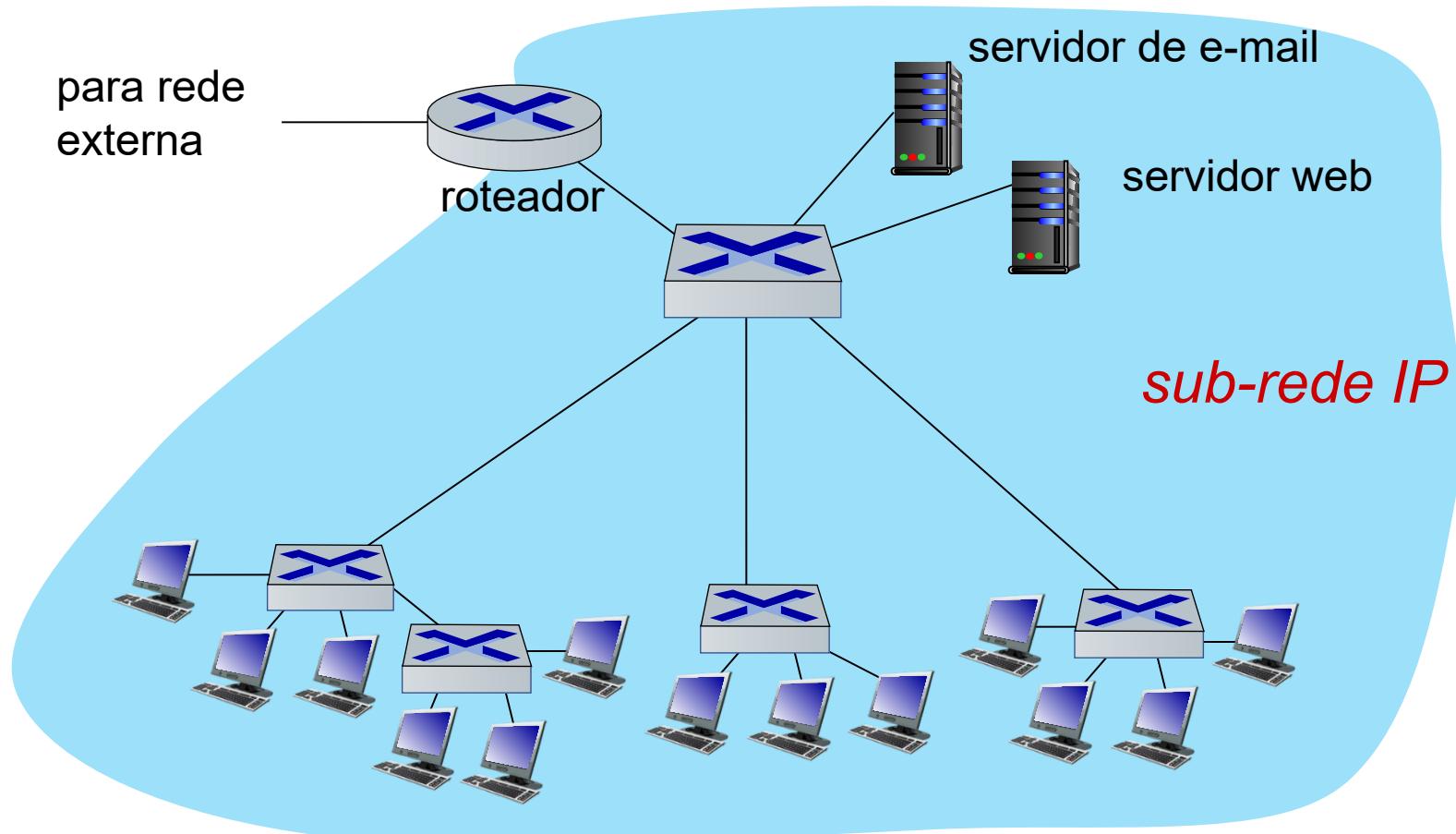
Exemplo de auto aprendizado com múltiplos comutadores

Suponha que C envia um quadro para I, I responde para C



Q: mostre as tabelas de comutação e encaminhamento de pacotes em S_1 , S_2 , S_3 , e S_4

Pequena rede institucional



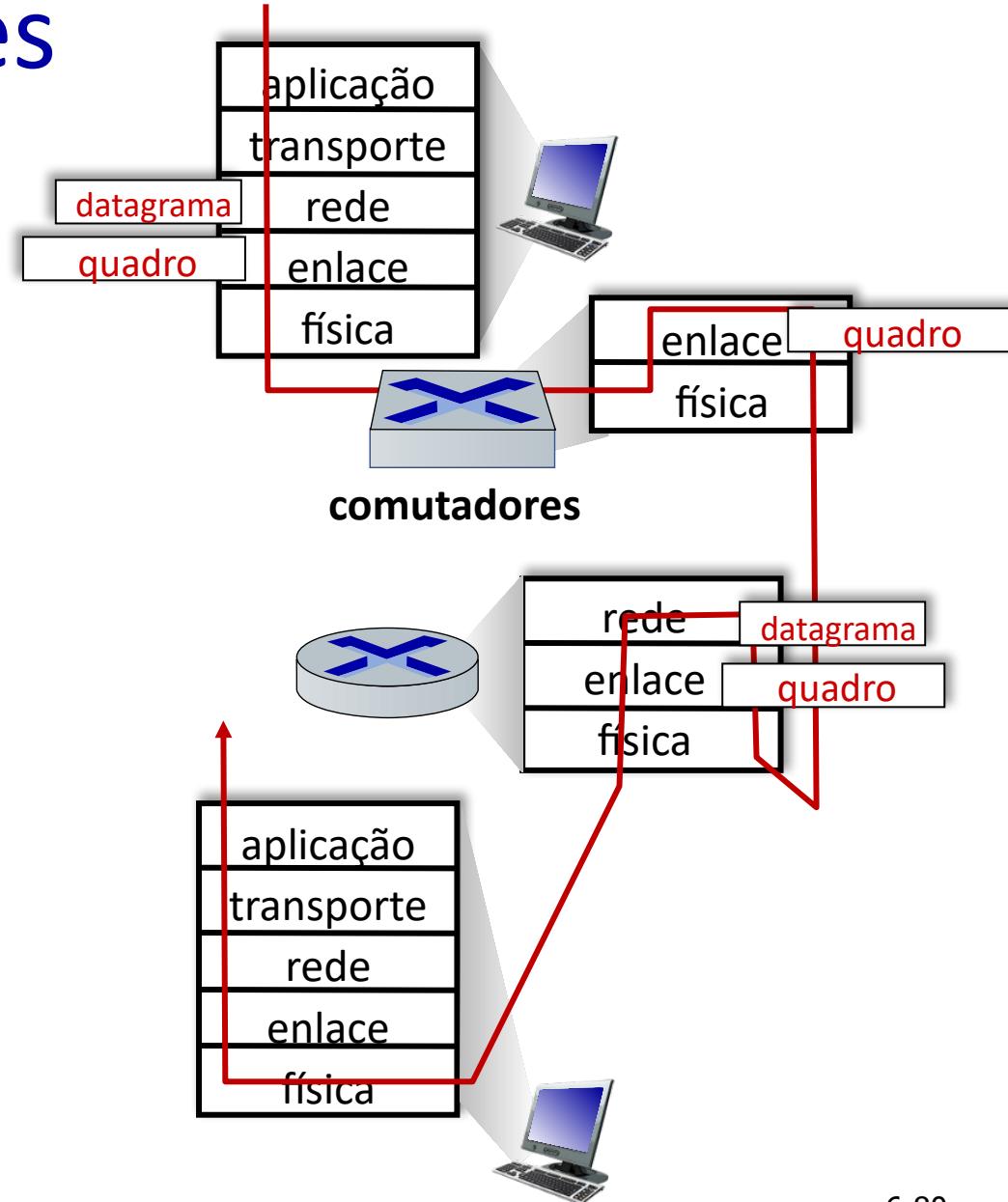
Comutadores vs. roteadores

ambos armazenam e encaminham:

- *roteadores*: dispositivos de camada de rede (examinam cabeçalhos de camada de rede)
- *comutadores*: dispositivos de camada de enlace (examinam cabeçalhos de camada de enlace)

ambos têm tabelas de encaminhamento:

- *roteadores*: calculam tabelas usando algoritmos de roteamento e endereços IP
- *comutadores*: aprendem a tabela de encaminhamento usando inundação, auto aprendizado e endereços MAC

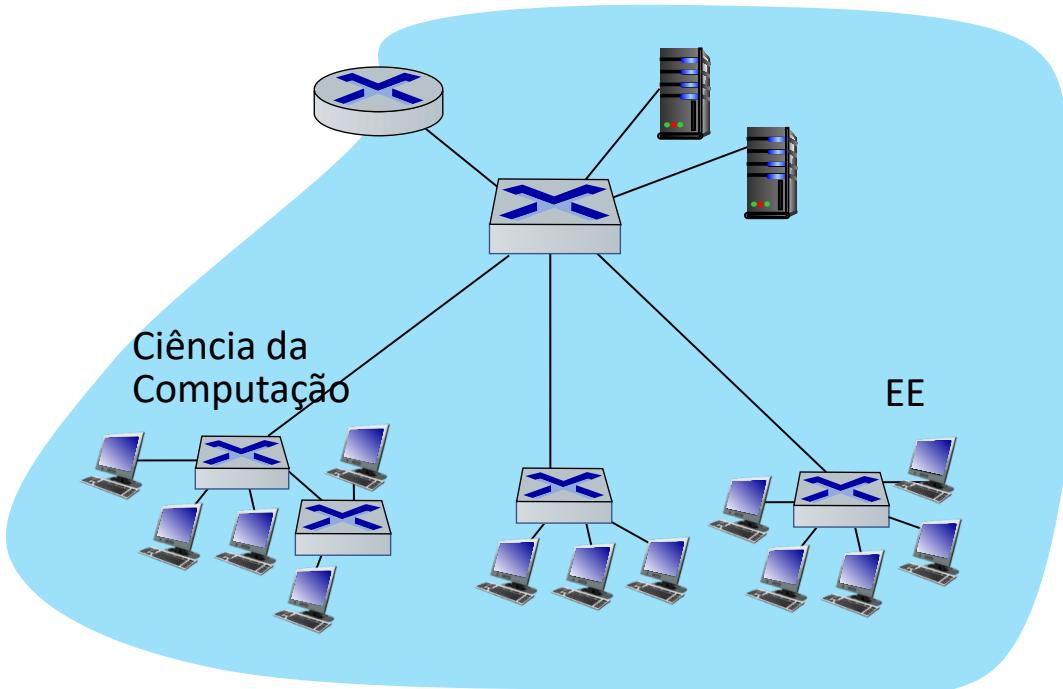


Camada de enlace, LANs: roteiro

- introdução
 - detecção e correção de erros
 - protocolos de acesso múltiplo
 - **LANs**
 - endereçamento, ARP
 - Ethernet
 - comutadores (*switches*)
 - **VLANs**
 - virtualização de enlaces: MPLS
 - redes de *data center*
- 
- um dia na vida de uma requisição web

Virtual LANs (VLANs): motivação

Q: o que acontece à medida que os tamanhos da LAN aumentam e os usuários mudam de ponto de conexão?

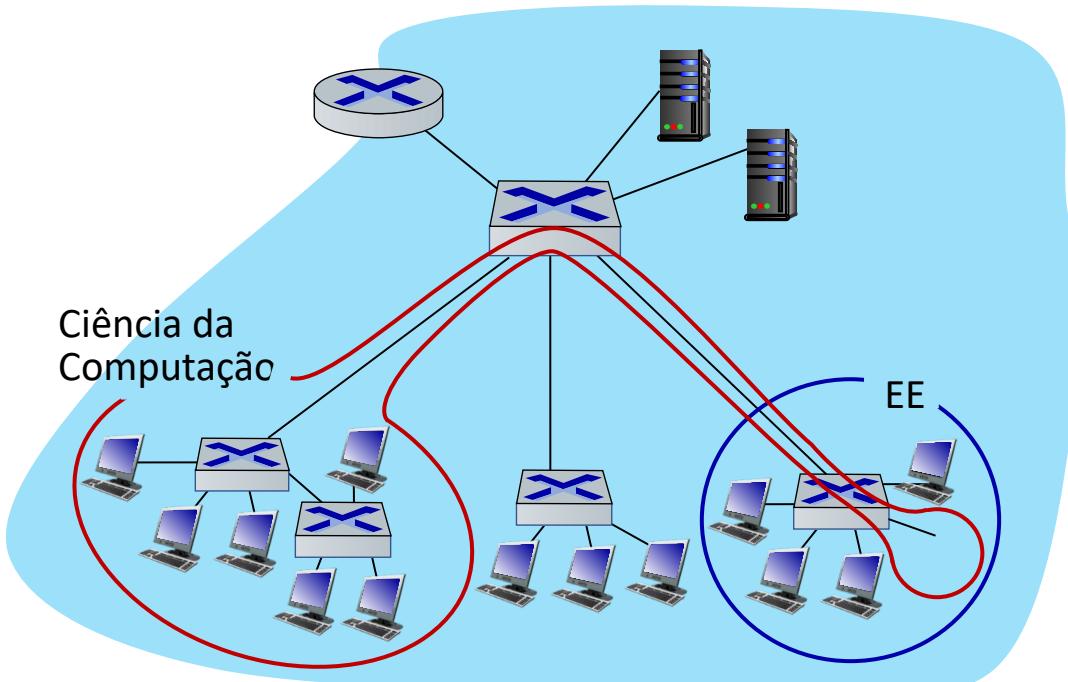


domínio de difusão (*broadcast*) único:

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

Virtual LANs (VLANs): motivação

Q: o que acontece à medida que os tamanhos da LAN aumentam e os usuários mudam de ponto de conexão?



domínio de difusão (*broadcast*) único:

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

problemas administrativos :

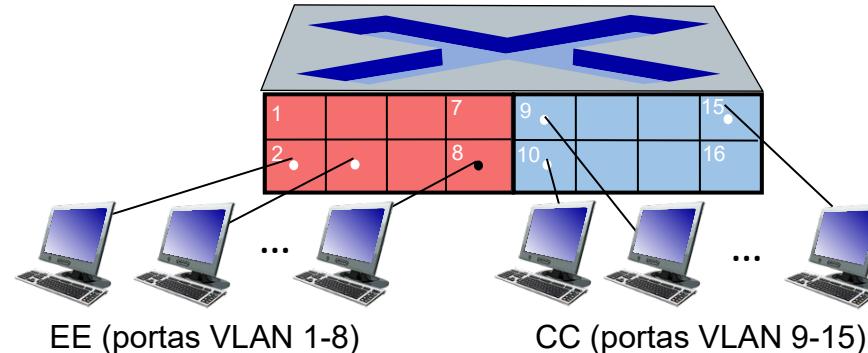
- usuário da CC muda sua sala para a EE
 - *fisicamente* conectado ao comutador da EE, mas quer permanecer *logicamente* conectado ao comutador da CC

VLANs baseadas em porta

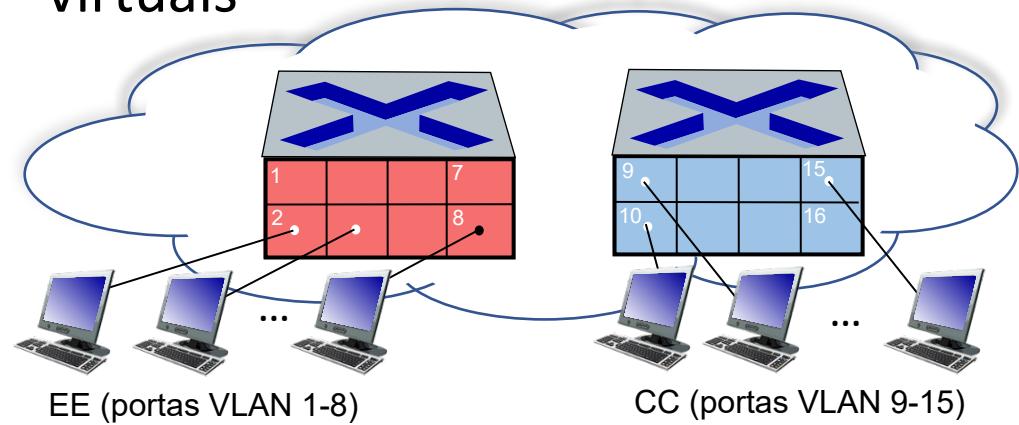
Virtual Local Area Network (VLAN)

comutador(es) que suportam recursos de VLAN podem ser configurados para definir várias LANs *virtuais* em uma única infraestrutura de LAN física.

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

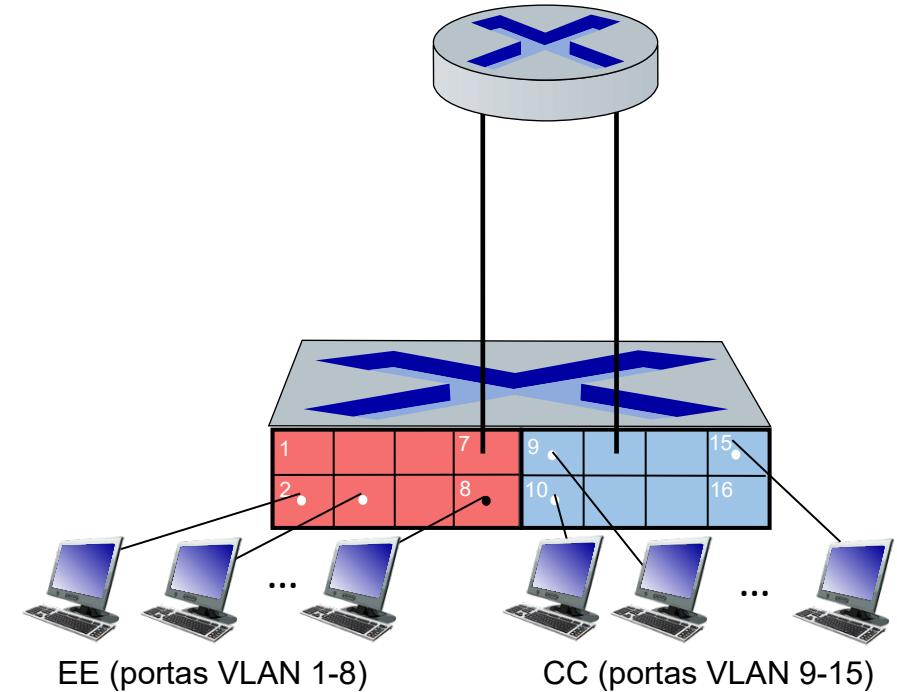


... opere como **múltiplos** comutadores virtuais

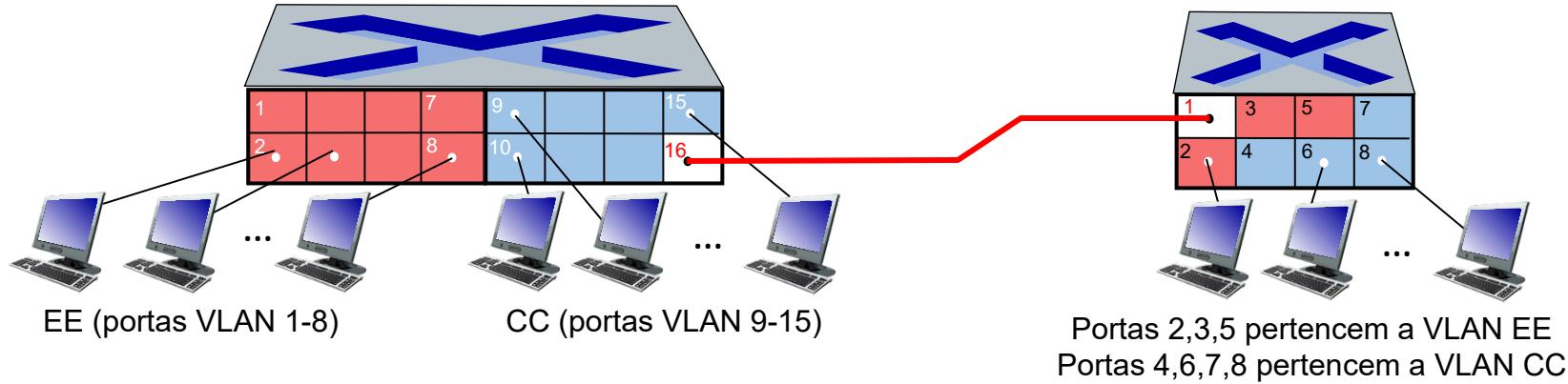


VLANs baseadas em porta

- **isolamento de tráfego:** quadros de/para as portas 1-8 só podem alcançar as portas 1-8
 - também pode-se definir VLANs com base em endereços MAC de sistemas finais, em vez de portas do comutador
- **associação dinâmica:** as portas podem ser atribuídas dinamicamente entre VLANs
- **encaminhamento entre VLANs:** feito via roteamento (assim como com comutadores separados)
 - na prática, os fornecedores vendem comutadores com roteadores combinados



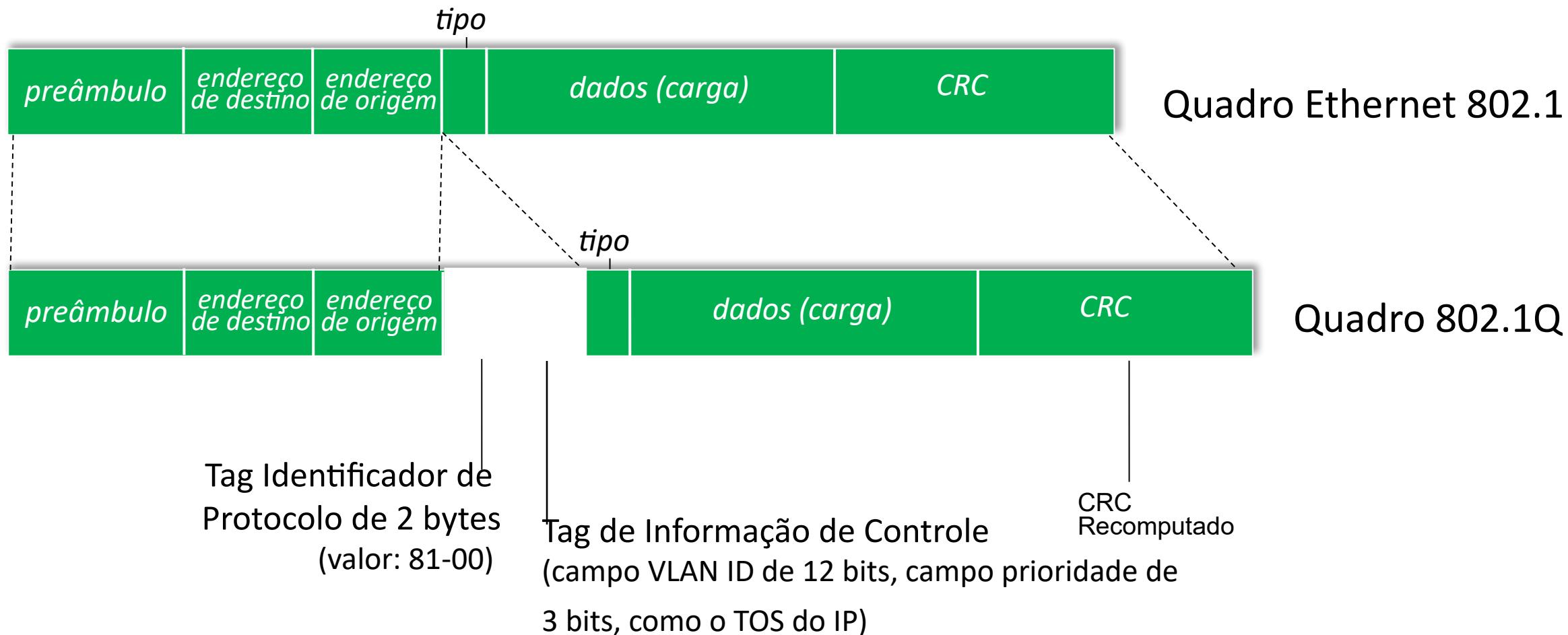
VLANs abrangendo vários comutadores



porta de tronco: transporta quadros entre VLANs definidas em vários comutadores físicos

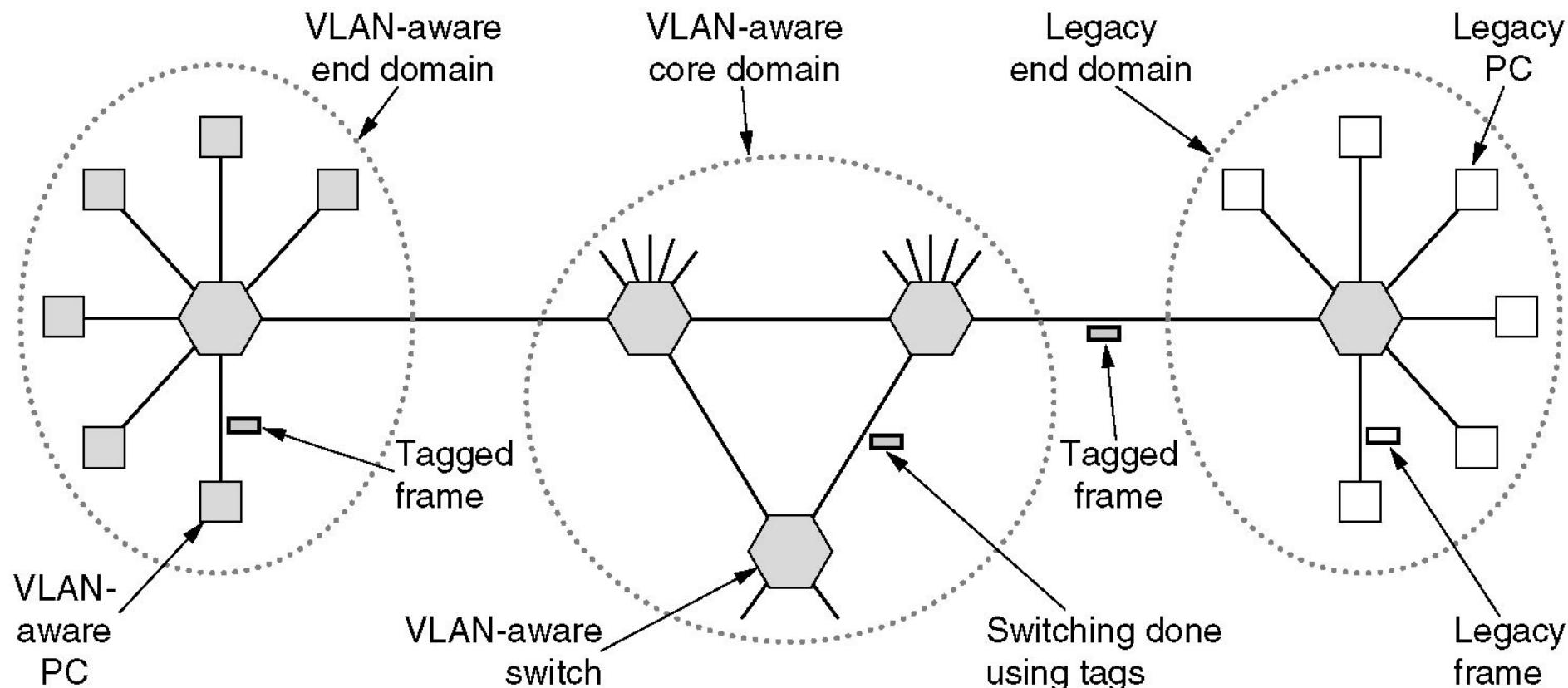
- os quadros encaminhados dentro da VLAN entre os comutadores não podem ser quadros 802.1 comuns (devem levar informações de ID da VLAN)
- o protocolo 802.1q adiciona/remove campos de cabeçalho adicionais para quadros encaminhados entre portas de tronco

Formato de quadro VLAN 802.1Q



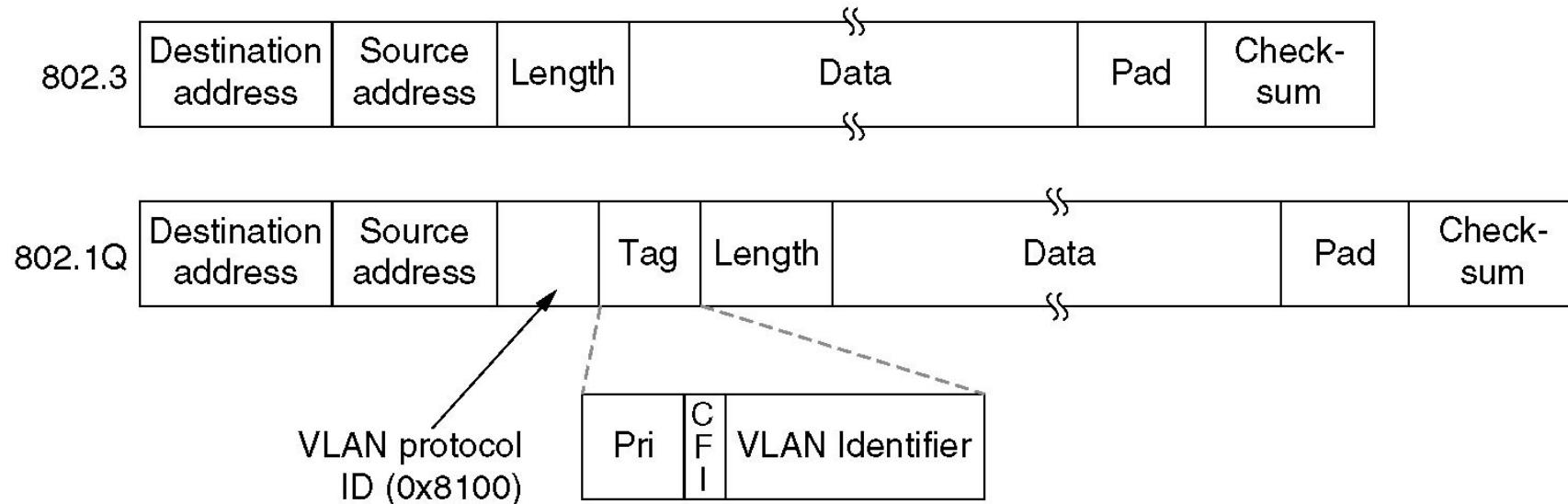
O Padrão IEEE 802.1Q

- Transição da Ethernet legada para a Ethernet com VLAN. Símbolos sombreados reconhecem VLAN. Os vazios não reconhecem.



O Padrão IEEE 802.1Q (2)

- The 802.3 (legacy) and 802.1Q Ethernet frame formats.



- Prioridade (3 bits)
 - Não tem nada a ver com VLAN, mas aproveitaram a modificação para incluir essa possibilidade de controle de qualidade de serviço
 - Distingue entre tráfego em tempo real permanente, tráfego em tempo real provisório e tráfego não relacionado a tempo
- CFI (Indicador de Formato Canônico): nada a ver com VLAN também, incluído por motivos políticos (“vote na minha modificação que voto na sua”). Serve para indicar que carga útil contém quadro IEEE 802.5 (Token Ring) encapsulado
- Identificador de VLAN (12 bits)

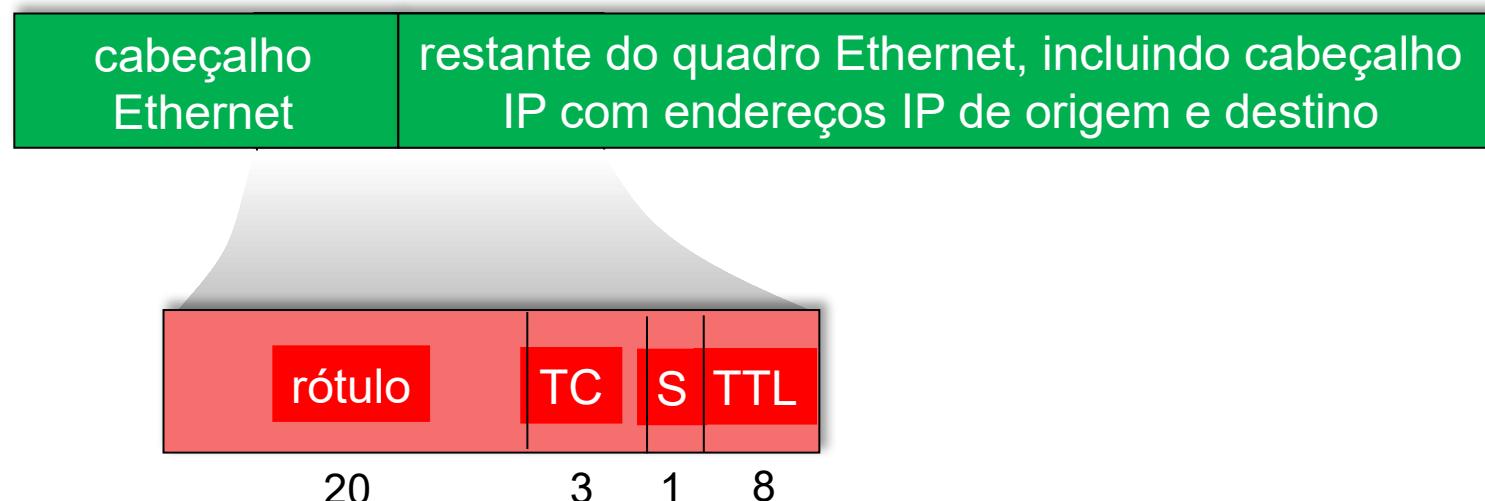
Camada de enlace, LANs: roteiro

- introdução
- detecção e correção de erros
- protocolos de acesso múltiplo
- LANs
 - endereçamento, ARP
 - Ethernet
 - comutadores (*switches*)
 - VLANs
- virtualização de enlaces: MPLS
- redes de *data center*
- um dia na vida de uma requisição web



Multiprotocolo label switching (MPLS – Comutação de Rótulo Multiprotocolo)

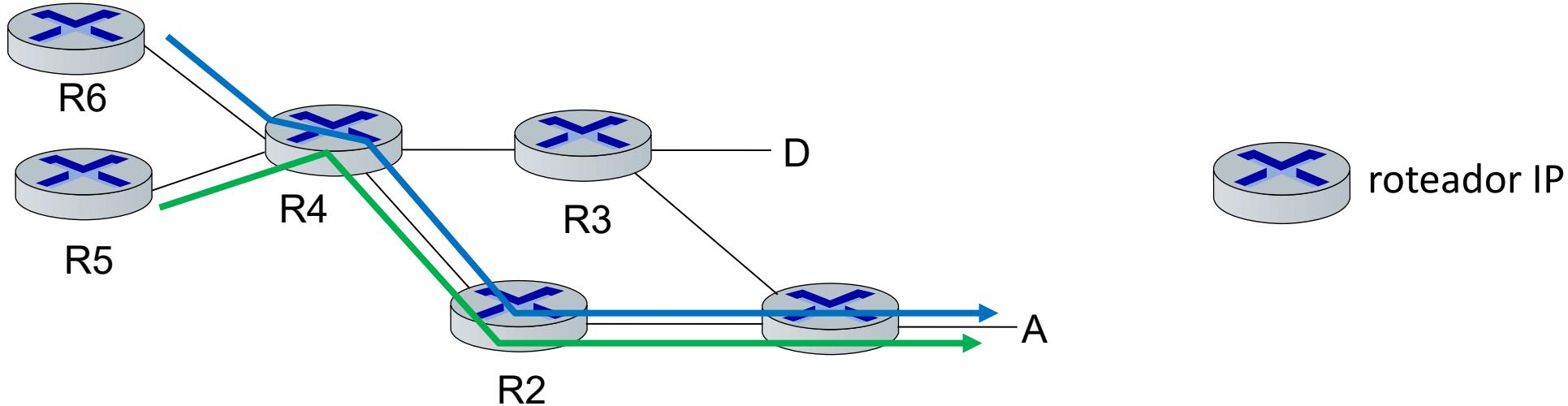
- **objetivo:** encaminhamento IP de alta velocidade entre a rede de roteadores compatíveis com MPLS, usando rótulo de comprimento fixo (em vez de correspondência de prefixo mais curto)
 - pesquisa mais rápida usando identificador de comprimento fixo
 - emprestando ideias da abordagem de Circuito Virtual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



Roteadores compatíveis com MPLS

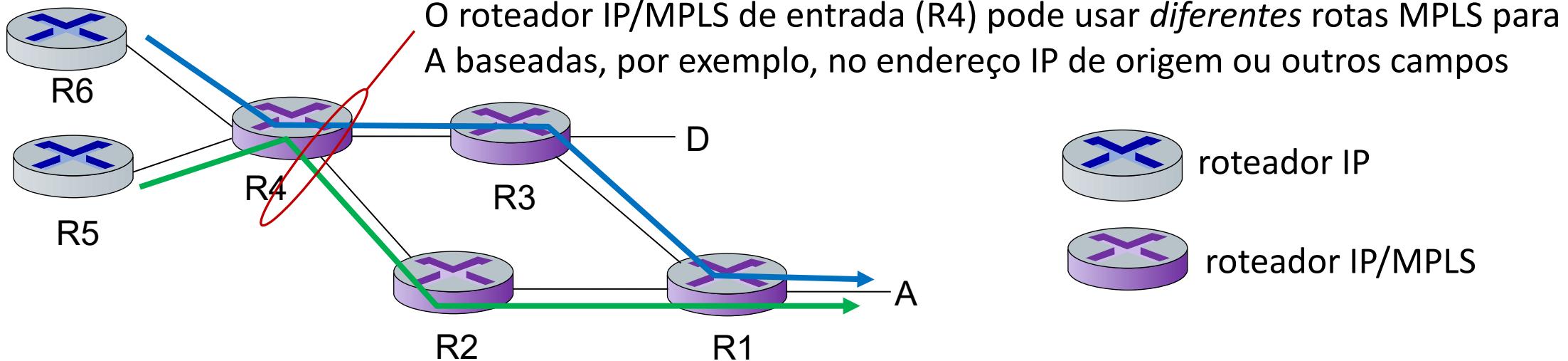
- também conhecido como roteador comutado por rótulo
- encaminha pacotes para a interface de saída com base apenas no valor do rótulo (*não inspeciona o endereço IP*)
 - Tabela de encaminhamento MPLS distinta das tabelas de encaminhamento IP
- ***flexibilidade***: Decisões de encaminhamento MPLS podem *diferir* daquelas do IP
 - usa endereços de destino e origem para rotear fluxos para o mesmo destino de forma diferente (engenharia de tráfego)
 - redireciona fluxos rapidamente se o enlace falhar: caminhos de backup pré-computados

Caminhos MPLS versus caminhos IP



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

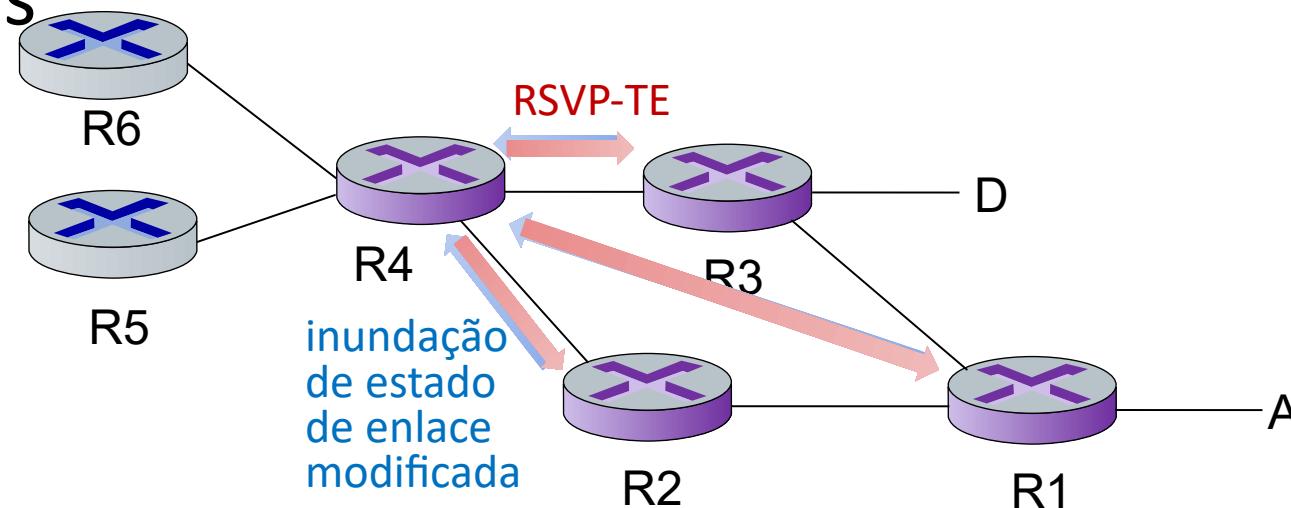
Caminhos MPLS versus caminhos IP



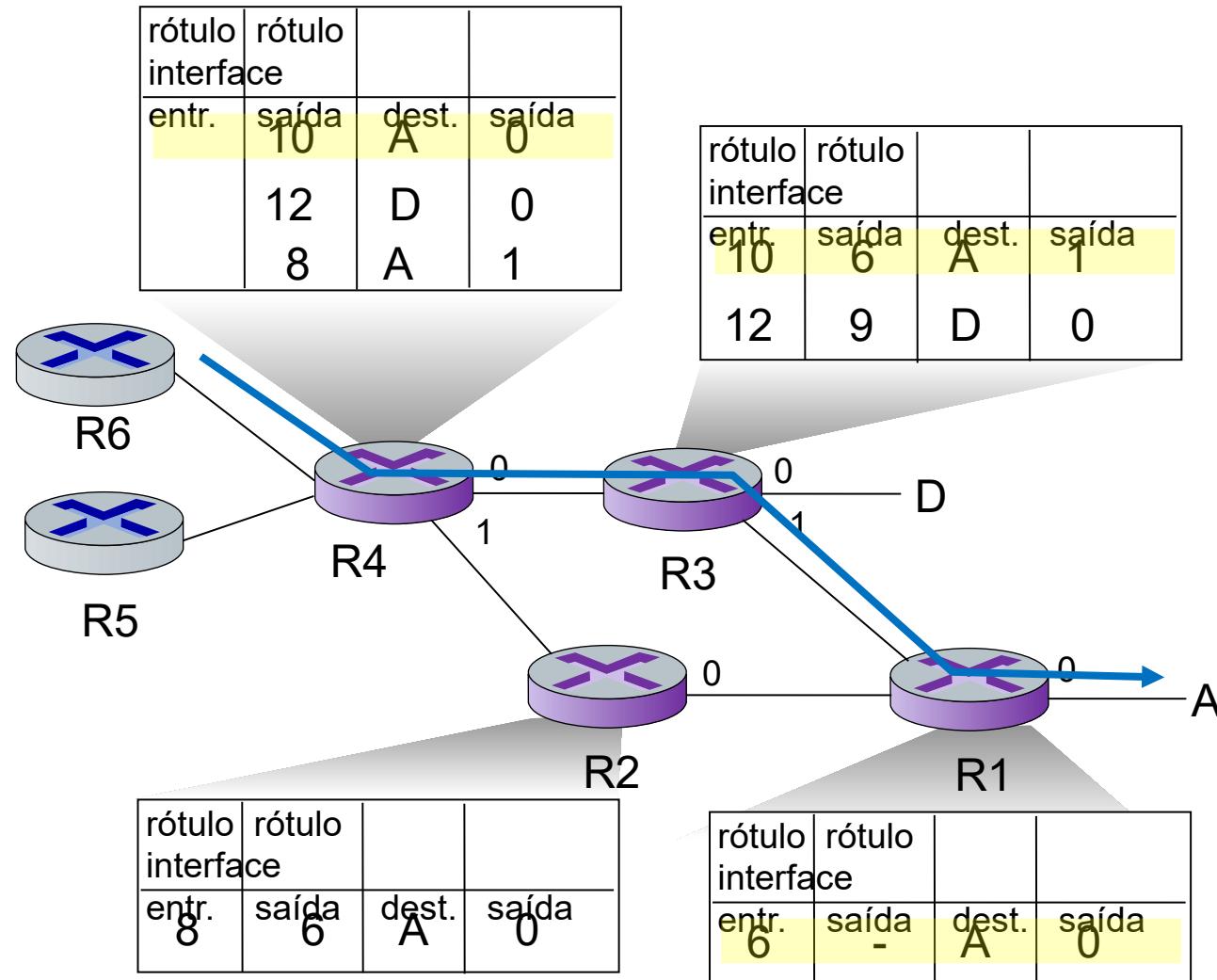
- **roteamento IP:** caminho para o destino determinado apenas pelo endereço de destino
- **roteamento MPLS:** caminho para o destino pode ser baseado no endereço de origem e no endereço de destino
 - um tipo de encaminhamento generalizado (MPLS chegou 10 anos antes)
 - *re-roteamento rápido:* pré-computa rotas de backup para o caso de falha de enlace

Sinalização MPLS

- modifica protocolos de inundação de estado de enlace do OSPF e do IS-IS para transportar informações usadas pelo roteamento MPLS:
 - por exemplo, largura de banda do enlace, quantidade “reservada” de largura de banda do enlace
- roteador MPLS de entrada usa o protocolo de sinalização RSVP-TE para configurar o encaminhamento MPLS nos próximos roteadores



Tabelas de encaminhamento MPLS



MPLS

- Fica entre o IP (camada de rede) e o PPP ou Ethernet (camada de enlace)
 - Não é da camada de rede, pois depende do IP ou outros endereços da camada de rede
 - Não é da camada de enlace, pois encaminha pacotes por vários saltos
 - As vezes é descrito como protocolo da “Camada 2,5”

Camada de enlace, LANs: roteiro

- introdução
- detecção e correção de erros
- protocolos de acesso múltiplo
- LANs
 - endereçamento, ARP
 - Ethernet
 - comutadores (*switches*)
 - VLANs
- virtualização de enlaces: MPLS
- redes em *data centers*



- um dia na vida de uma requisição web

Redes em *data centers*

De dezenas a centenas de milhares de hospedeiros, muitas vezes fortemente acoplados e próximos:

- e-business (ex.: Amazon)
- servidores de conteúdo (ex.: YouTube, Akamai, Apple, Microsoft)
- motores de busca, mineração de dados (ex.: Google)

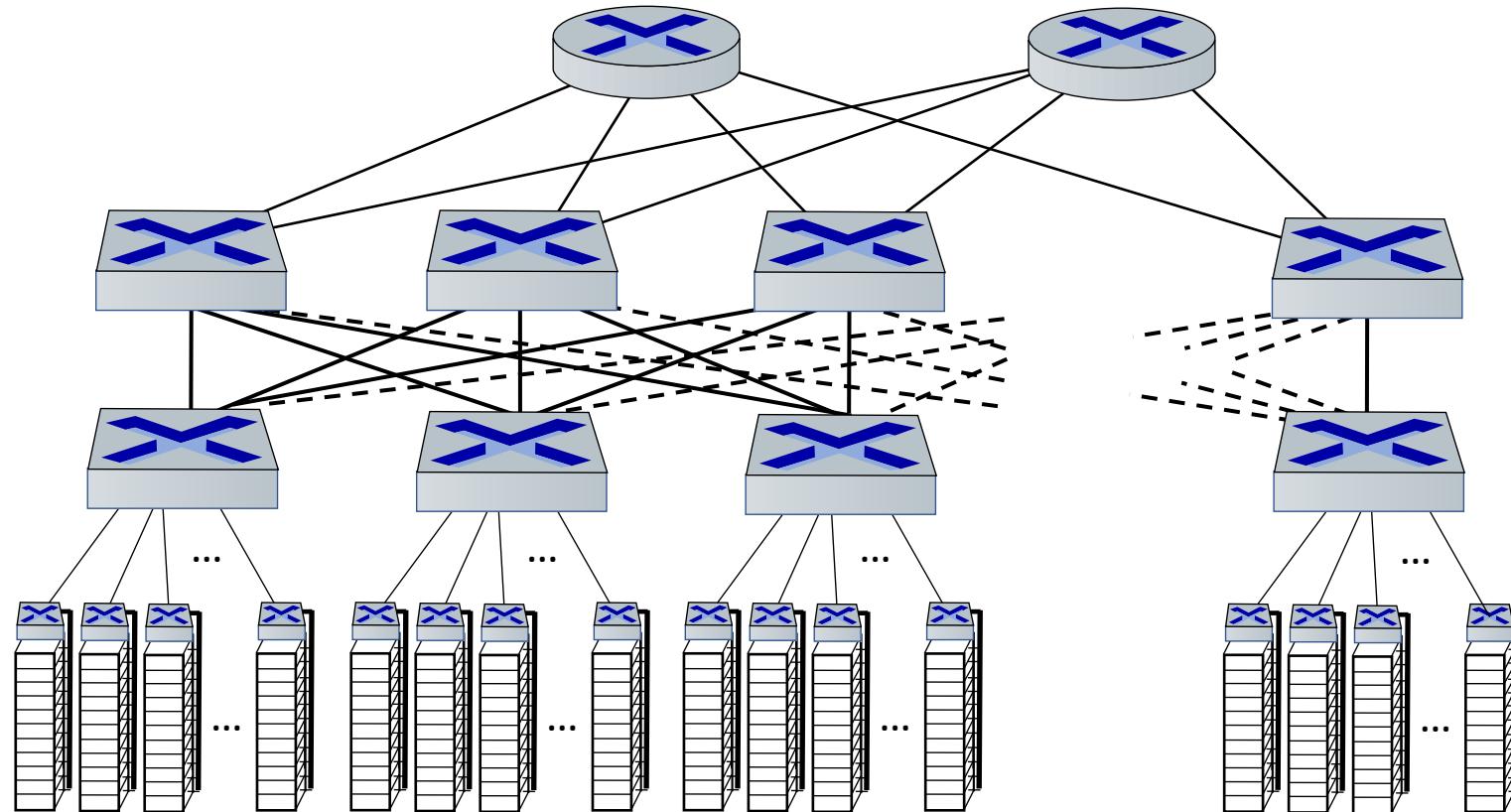
desafios:

- múltiplas aplicações, cada uma atendendo a um grande número de clientes
- confiabilidade
- gerenciamento/balanceamento de carga, evitando gargalos de processamento, de rede e de dados



Dentro de um contêiner da Microsoft de 40 pés (12,192 metros), *data center* de Chicago

Redes de *datacenter*: elementos de rede



Roteadores de borda

- conexões para fora do *datacenter*

Comutadores de nível 1

- conectado a ~16 nível 2 abaixo

Comutadores de nível 2

- conectado a ~16 TORs abaixo

Comutadores Top of Rack (TOR)

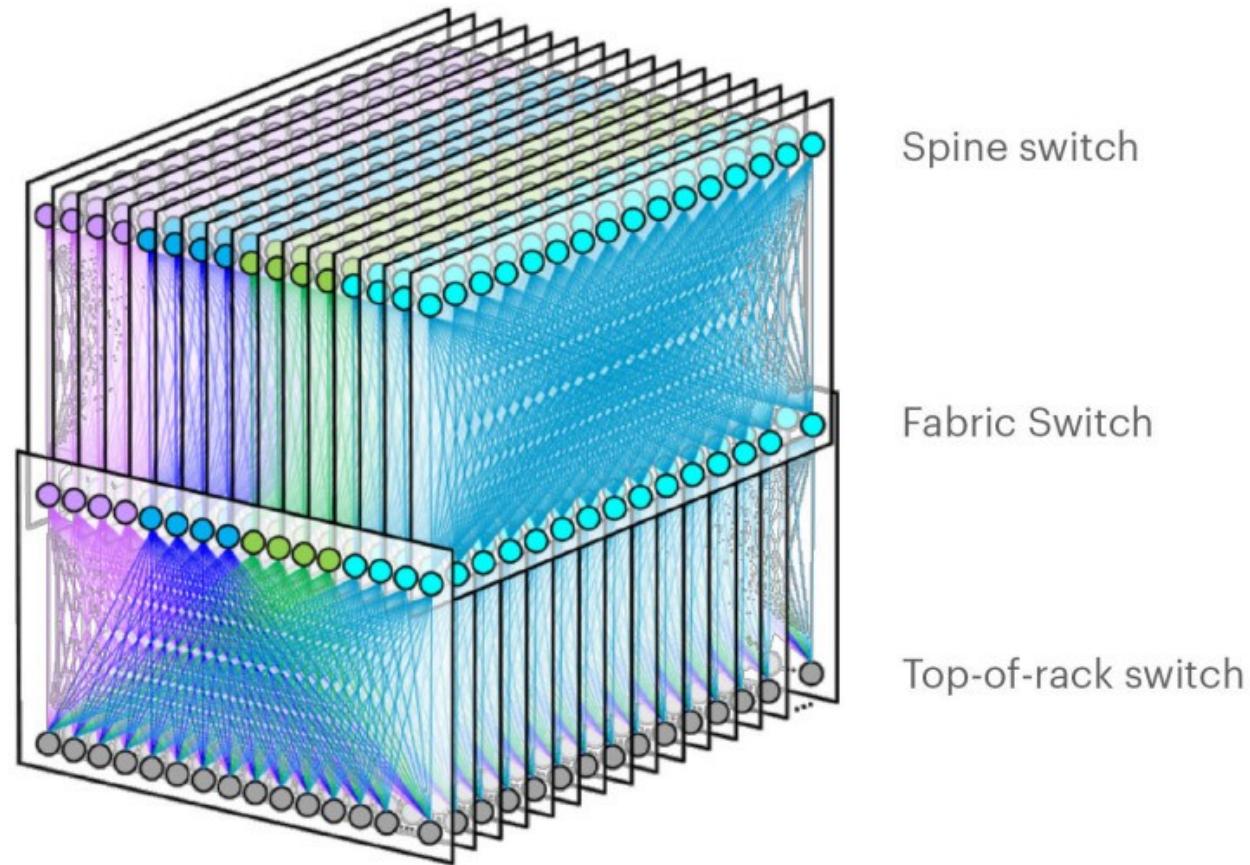
- um por rack
- 40-100Gbps Ethernet para lâminas

Racks de servidores

- 20- 40 servidores *blade*: hospedeiros

Redes de *datacenter*: elementos de rede

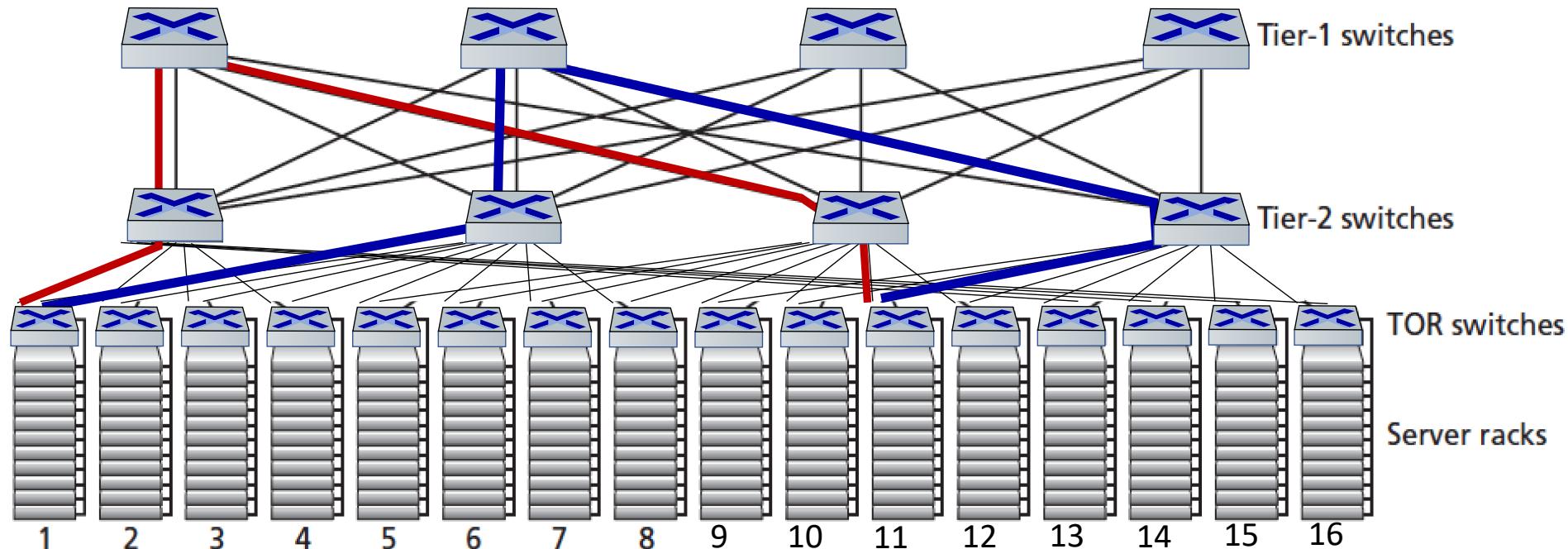
Topologia de rede do data center F16 do Facebook:



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

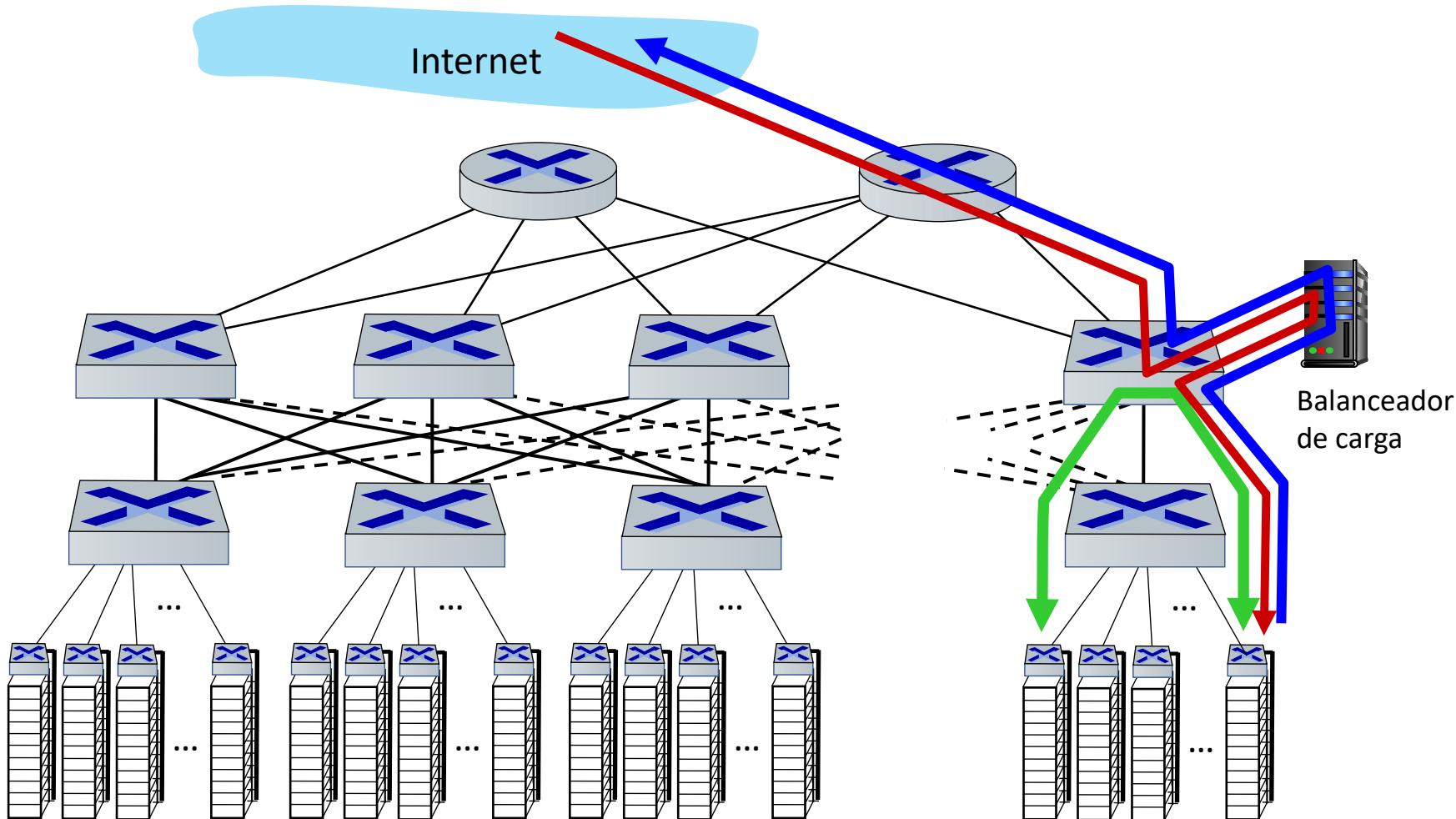
Redes de *datacenters*: multicaminho

- rica interconexão entre comutadores e racks:
 - aumento da taxa de transferência entre racks (vários caminhos de roteamento possíveis)
 - maior confiabilidade por meio de redundância



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

Redes de *datacenters*: roteamento de camada de aplicação



balanceador de carga: roteamento de camada de aplicação

- recebe solicitações de clientes externos
- direciona a carga de trabalho dentro do *data center*
- retorna resultados ao cliente externo (ocultando os componentes internos do *data center* do cliente)

Redes de *datacenters*: inovações de protocolos

- **camada de enlace:**
 - RoCE: DMA remoto (RDMA) sobre Ethernet Convergente
- **camada de transporte:**
 - ECN (*explicit congestion notification* - notificação explícita de congestionamento) usado 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 dos/entre os datacenters das organizações
 - coloque serviços e dados relacionados o mais próximo possível (por exemplo, no mesmo rack ou rack próximo) para minimizar a comunicação de nível 2 e nível 1

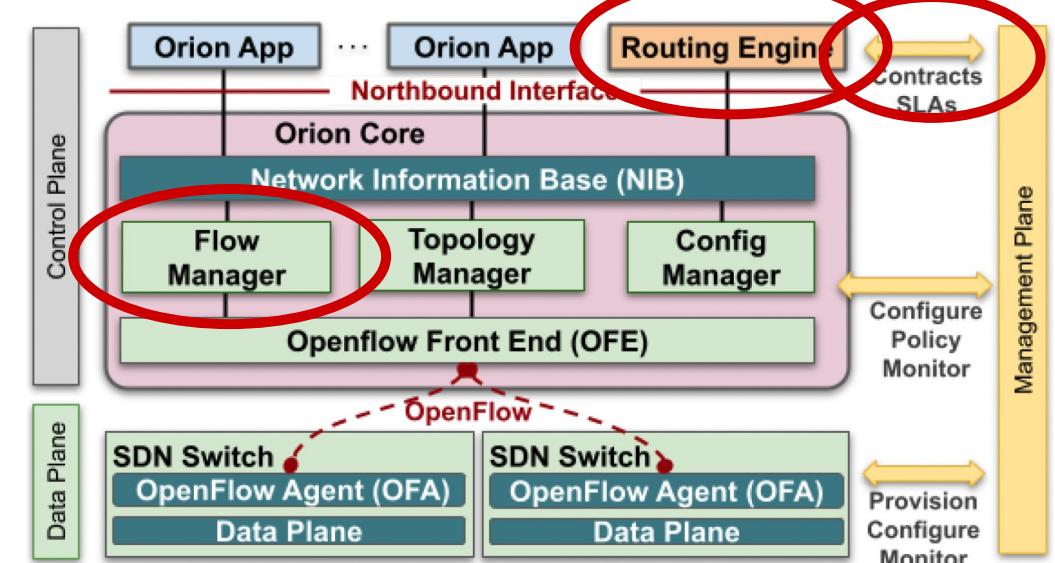
Google Networking: Infrastructure and Selected Challenges (Slides:

<https://networkingchannel.eu/google-networking-infrastructure-and-selected-challenges/>

ORION: o novo plano de controle SDN do Google para data centers internos (Jupiter) + rede de longa distância (B4)

- roteamento (intradomínio, iBGP), engenharia de tráfego: implementada em aplicações sobre o núcleo do ORION
- controles de **fluxo baseados em edge-edge** (por exemplo, agendamento CoFlow) para cumprir SLAs de contrato
- **gerenciamento**: microserviços distribuídos de publicação-subscrição no núcleo do Orion, OpenFlow para sinalização/monitoramento de switches
- **Nota:**
 - sem protocolos de roteamento, controle de congestionamento (parcialmente) também gerenciado por SDN em vez de por protocolo
 - os protocolos estão morrendo?

Arquitetura SDN do Orion e aplicativos de núcleo



Camada de enlace, LANs: roteiro

- introdução
- detecção e correção de erros
- protocolos de acesso múltiplo
- LANs
 - endereçamento, ARP
 - Ethernet
 - comutadores (*switches*)
 - VLANs
- virtualização de enlaces: MPLS
- redes em *data centers*

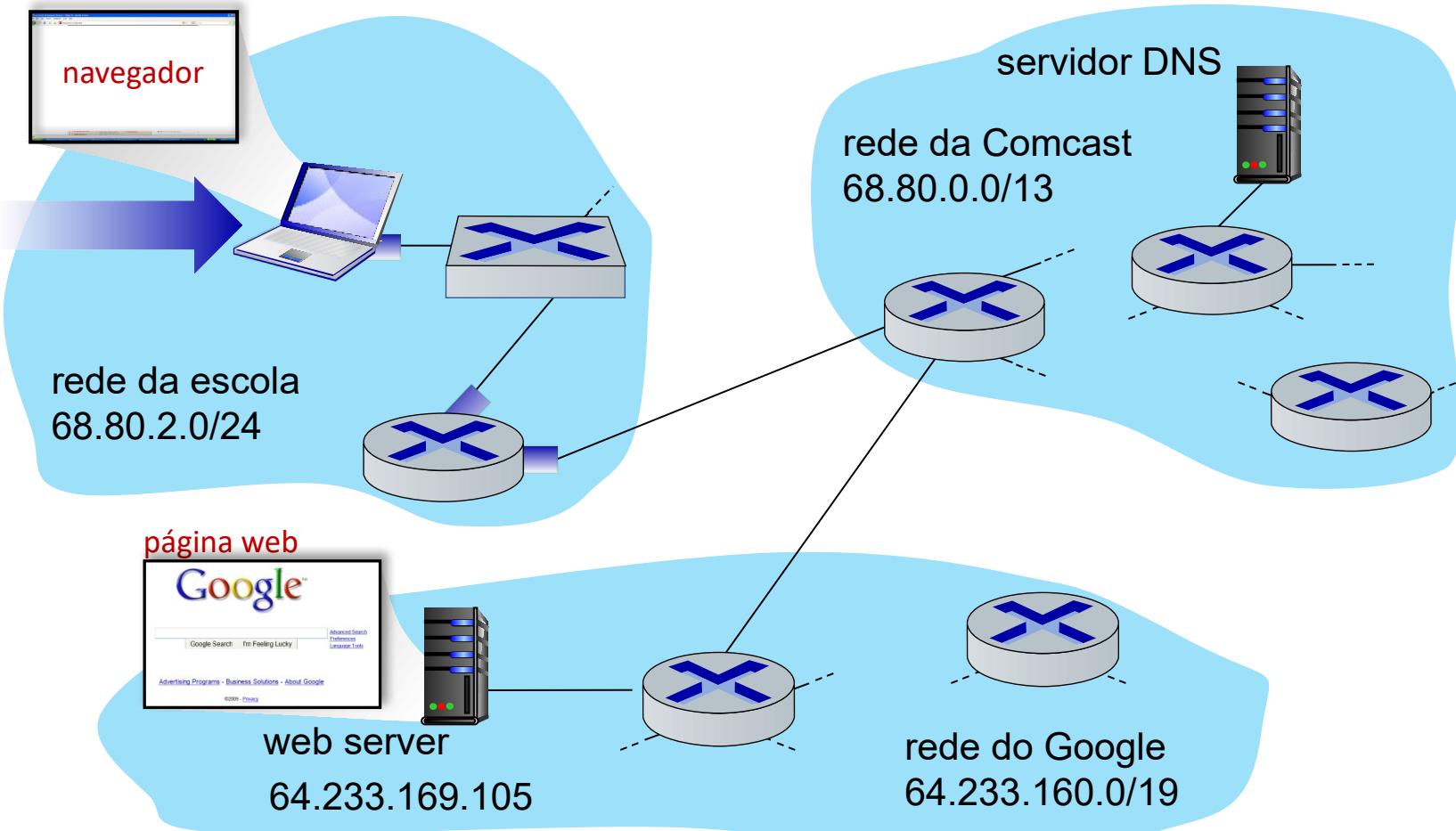


- um dia na vida de uma requisição web

Síntese: um dia na vida de uma solicitação da web

- nossa jornada pela pilha de protocolos agora está completa!
 - aplicação, transporte, rede, enlace
- juntando tudo: síntese!
 - *objetivo*: identificar, revisar, entender os protocolos (em todas as camadas) envolvidos em um cenário aparentemente simples: requisitando uma página www
 - *cenário*: aluno conecta laptop à rede do campus, solicita/recebe www.google.com

Um dia na vida: cenário

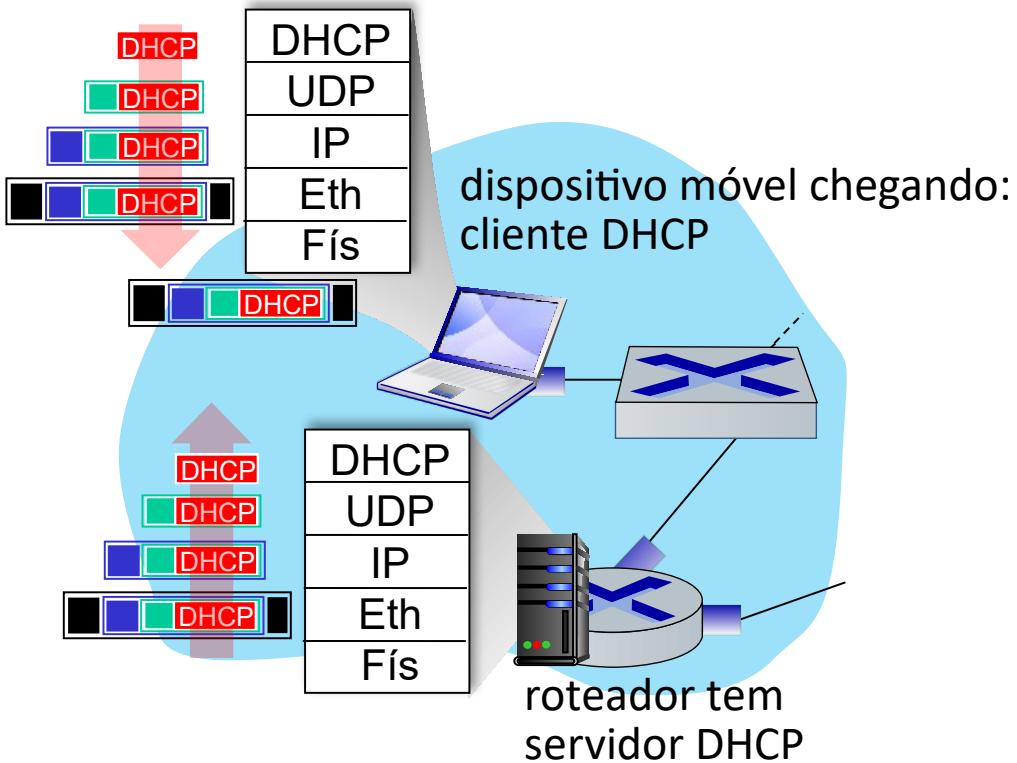


cenário:

- cliente móvel que chega se conecta à rede ...
- requisita página web: www.google.com

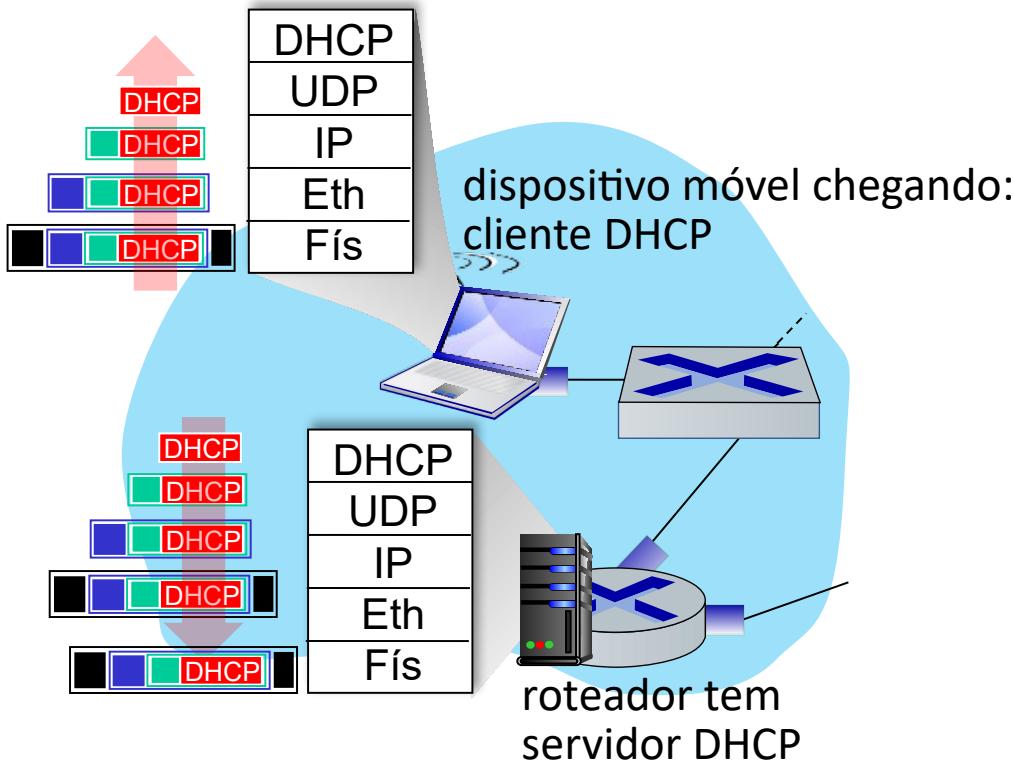
Parece simples! !

Um dia na vida: conectando-se à Internet



- laptop conectando precisa obter seu próprio endereço IP, endereço do roteador do primeiro salto, e endereço do servidor DNS: usa **DHCP**
- requisição DHCP **encapsulada em UDP**, encapsulada em **IP**, encapsulada em **Ethernet 802.3**
- quadro Ethernet enviado por **broadcast** (dest: FFFFFFFFFFFF) na LAN, recebido no roteador executando servidor **DHCP**
- Ethernet **demultiplexado para IP**, demultiplexado para **UDP**, demultiplexado para **DHCP**

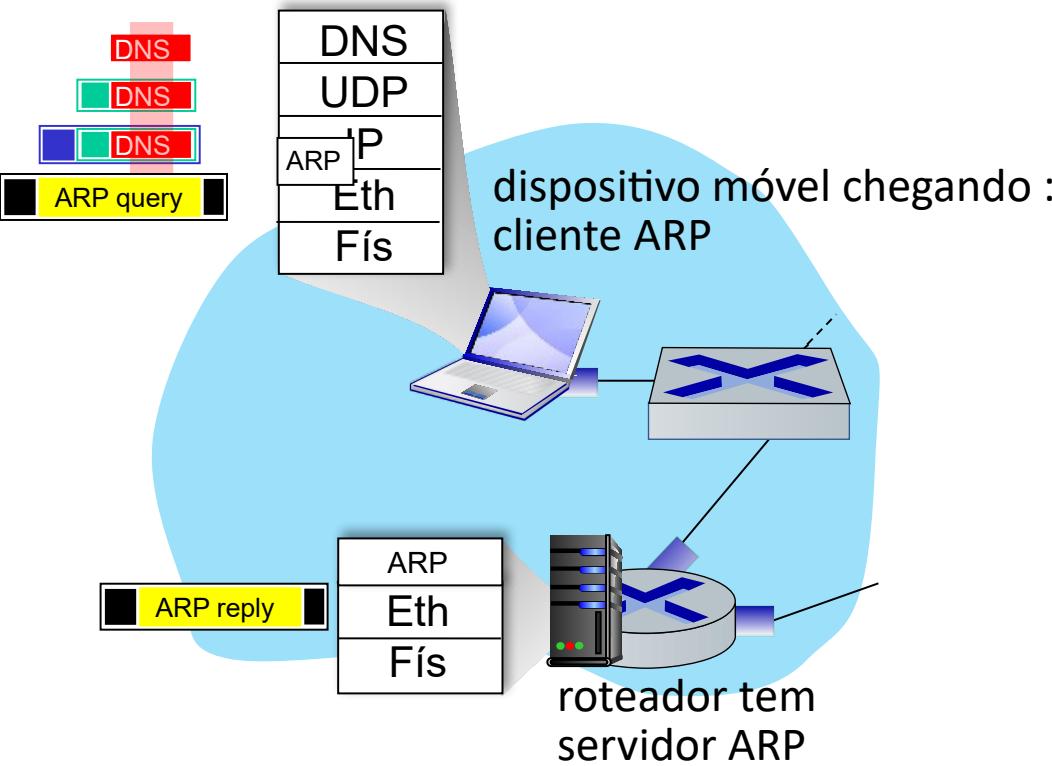
Um dia na vida: conectando-se à Internet



- servidor DHCP formula **DHCP ACK** contendo endereço IP do cliente, endereço IP do roteador de primeiro salto, e nome e endereço IP do servidor DNS
- encapsulamento no servidor DHCP, quadro encaminhado (**aprendizado do comutador**) através da LAN, demultiplexando no cliente
- cliente DHCP recebe resposta DHCP ACK

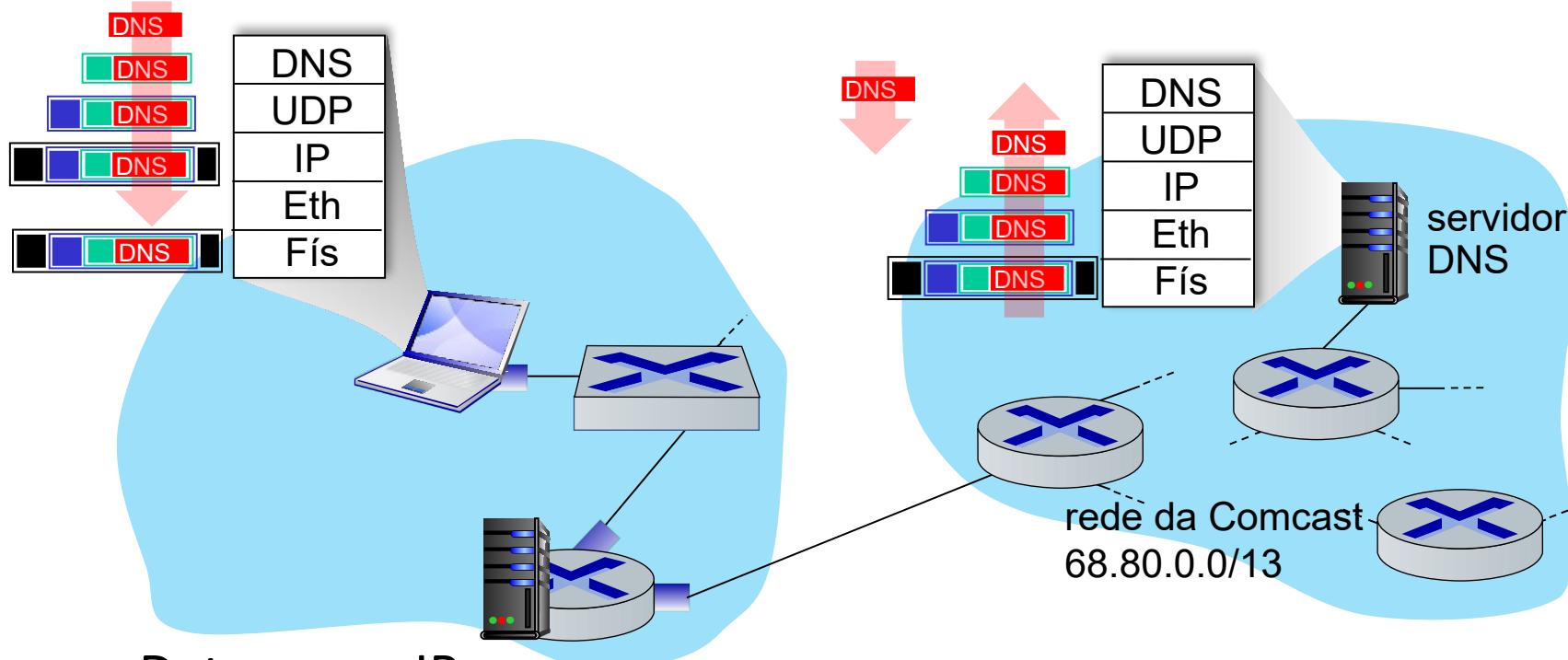
O cliente agora tem endereço IP, sabe o nome e endereço do servidor DNS e endereço IP de seu roteador de primeiro salto

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



- antes de enviar requisição **HTTP**, precisa do endereço IP de www.google.com: **DNS**
- Consulta DNS criada, encapsulada em UDP, encapsulada em IP, encapsulada em Ethernet. Para enviar o quadro para o roteador, é necessário o endereço MAC da interface do roteador: **ARP**
- broadcast da **requisição ARP**, recebida pelo roteador, que responde com uma **resposta ARP** fornecendo o endereço MAC da interface do roteador
- o cliente agora conhece o endereço MAC do roteador do primeiro salto, então agora pode enviar o quadro contendo a consulta **DNS**

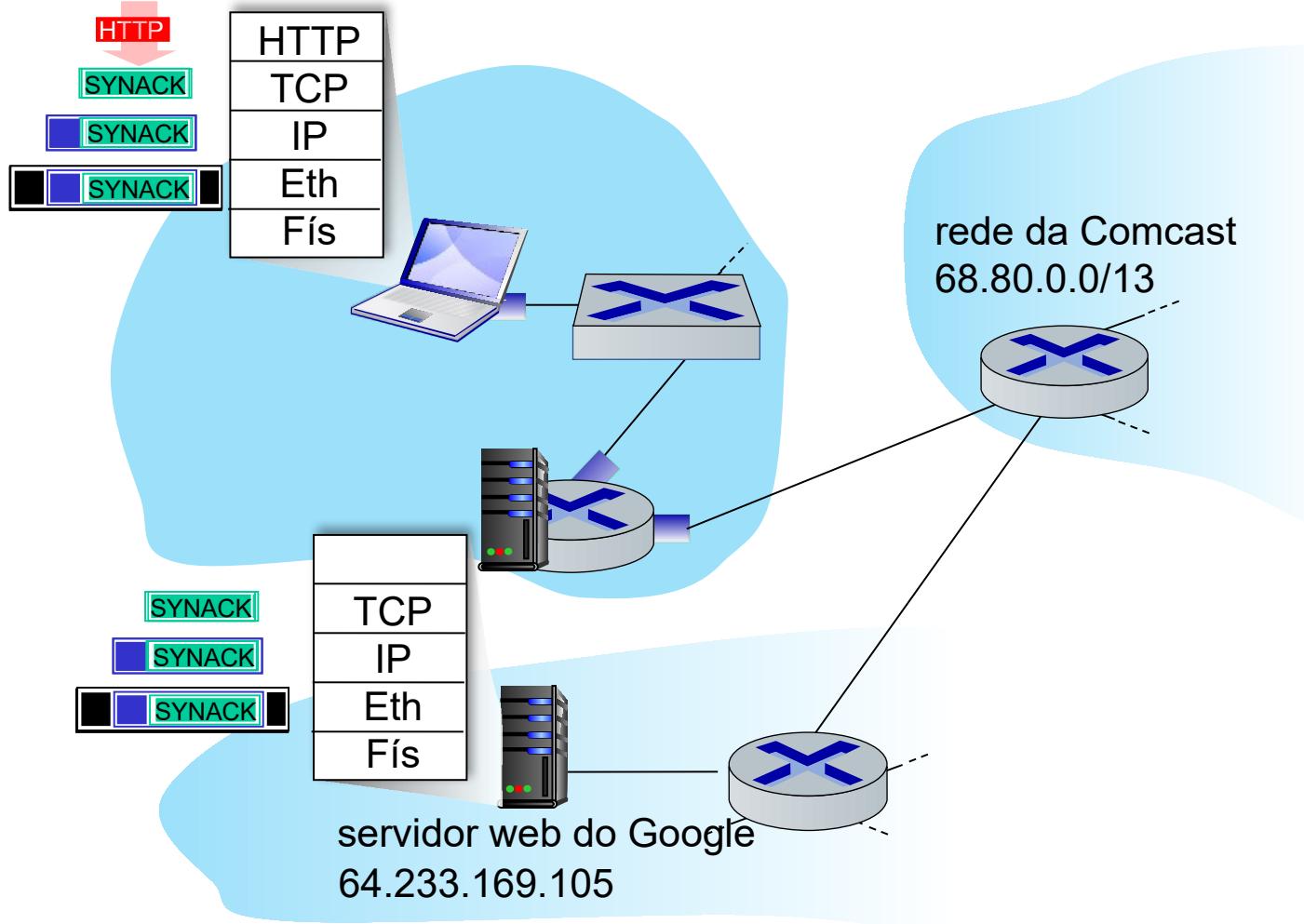
Um dia na vida... usando DNS



- Datagrama IP contendo consulta DNS encaminhada via comutador LAN do cliente para o roteador de 1º salto
- Datagrama IP encaminhado da rede do campus para a rede da Comcast, roteado (tabelas criadas por protocolos de roteamento **RIP, OSPF, IS-IS e/ou BGP**) para o servidor DNS

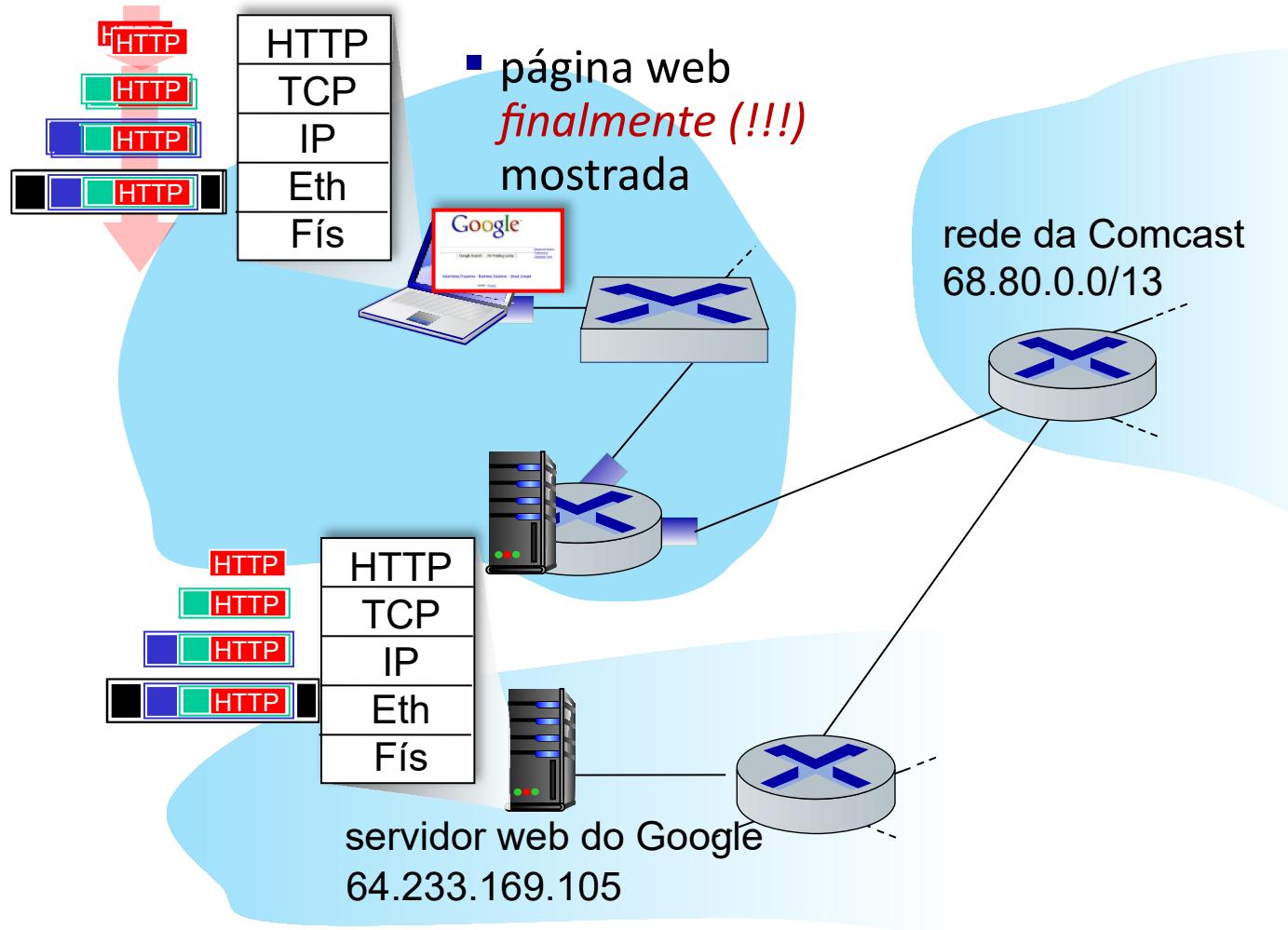
- demultiplexado para DNS
- DNS responde ao cliente com endereço IP de www.google.com

Um dia na vida... conexão TCP carregando HTTP



- para enviar solicitação HTTP, o cliente primeiro abre **socket TCP** para o servidor web
- **segmento TCP SYN** (passo 1 no *handshake de 3 vias* do TCP) roteado entre domínios para o servidor web
- servidor web responde com **TCP SYNACK** (passo 2 no *handshake de 3 vias* do TCP)
- **conexão TCP estabelecida!**

Um dia na vida... requisição/resposta HTTP



- **requisição HTTP** enviada em 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 roteado 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 enlace
- instanciação e implementação de várias tecnologias de camada de enlace
 - Ethernet
 - LANs comutadas, VLANs
 - redes virtualizadas como uma camada de enlace: MPLS
- síntese: um dia na vida de uma requisição web

Capítulo 6: vamos respirar

- jornada pela pilha de protocolos *completa* (exceto camada física)
- compreensão sólida dos princípios e prática de rede!
- poderia parar por aqui mas ainda há *mais* tópicos interessantes!
 - sem fio
 - segurança

Leitura Recomendada e Complementar

● Leitura Recomendada:

- [KUROSE, James F. e ROSS, Keith W. Redes de computadores e a Internet: Uma abordagem top-down. 8ª Edição. Bookman, 2021.](#)
 - Capítulo 6 – A Camada de Enlace e Redes Locais
- [TANENBAUM, Andrew S., FEAMSTER, Nick e WETHERALL, David. Redes de Computadores. 6ª Edição. São Paulo: Bookman, 2021.](#)
 - Capítulo 3 – A Camada de Enlace de Dados
 - Capítulo 4 – A Subcamada de Controle de Acesso ao Meio

● Leitura Complementar:

- [FOUROUZAN, Behrouz A. e FIROUZ, Mosharraf. Redes de Computadores: uma abordagem top-down. Porto Alegre: AMGH, 2013.](#)
 - Capítulo 5 – Camada de Enlace de Dados: Redes com Fios.
- [TORRES, Gabriel. Redes de Computadores: Curso Completo. Axcel Books, 2001.](#)
 - Capítulo 3 – Capítulos 13 a 17.
- [COMER, Douglas E. Interligação de Redes com TCP/IP. Volume 1: Princípios, protocolos e arquitetura. 6ª Edição. Rio de Janeiro: Elsevier, 2015.](#)
 - Capítulos 2 – Revisão das tecnologias de rede básicas.
- [MAIA, Luiz Paulo. Arquitetura de Redes de Computadores. 2ª edição. LTC, 2013.](#)
 - Capítulos 4 e 5



Slides adicionais do Capítulo 6

Eficiência do ALOHA puro

$$\begin{aligned} P(\text{sucesso de determinado nó}) &= P(\text{nó transmitir}) * \\ &\quad P(\text{nenhum outro nó transmitir em } [t_0-1, t_0]) * \\ &\quad P(\text{nenhum outro nó transmitir em } [t_0-1, t_0]) * \\ &= p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1} \\ &= p \cdot (1-p)^{2(N-1)} \end{aligned}$$

... escolhendo p ótimo e considerando \bar{n}^∞

$$= 1/(2e) = .18$$

ainda pior do que o slot Aloha!