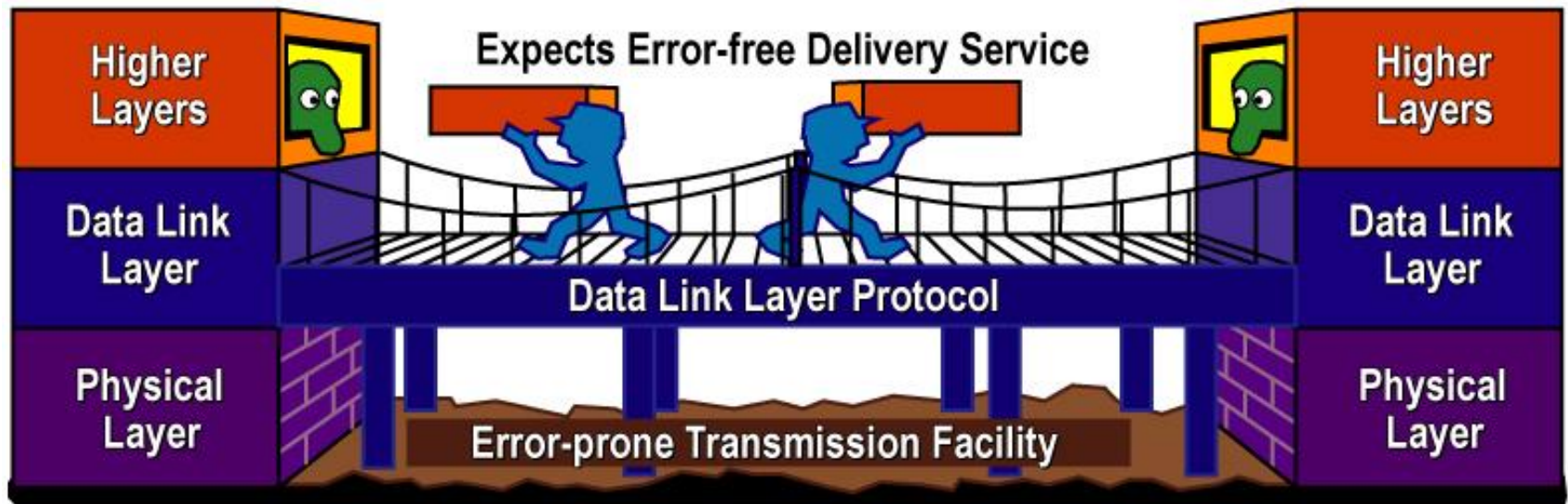


# O nível de Ligação de Dados



# O nível Físico

- ❑ 1. Modula o processo de conversão de um sinal de uma forma para outra, para que possa ser transmitido fisicamente através de um canal de comunicação.
- ❑ 2. Entrega bit-a-bit.
- ❑ 3. Codificação de linha, que permite que os dados sejam enviados por dispositivos de hardware otimizados para comunicações digitais, que podem ter sincronismo no link de transmissão.
- ❑ 4. Sincronização de Bit.
- ❑ 5. Sinalização Start-stop e controlo de fluxo em comunicações série assíncronas.
- ❑ 6. Comutação de circuitos e controlo de sinais digitais multiplexados.
- ❑ 7. Detecção da disponibilidade da portadora.
- ❑ 8. Equalização dos sinais para assegurar ligações fiáveis.
- ❑ 9. Forward error correction/codificação do canal tais como códigos de correcção de erros.
- ❑ 10. Auto-negociação.
- ❑ 11. Controlo de transmissão.

# O nível de Ligação de Dados

## Objectivos:

- ❑ Compreender os princípios dos serviços da camada de ligação de dados:
  - Detecção e correcção de erros
  - Partilha de um meio de difusão: acesso múltiplo
  - endereçamento de nível 2
  - Transferência fiável de dados, controlo de fluxo
- ❑ Instanciação e implementação de várias tecnologias de ligação de dados

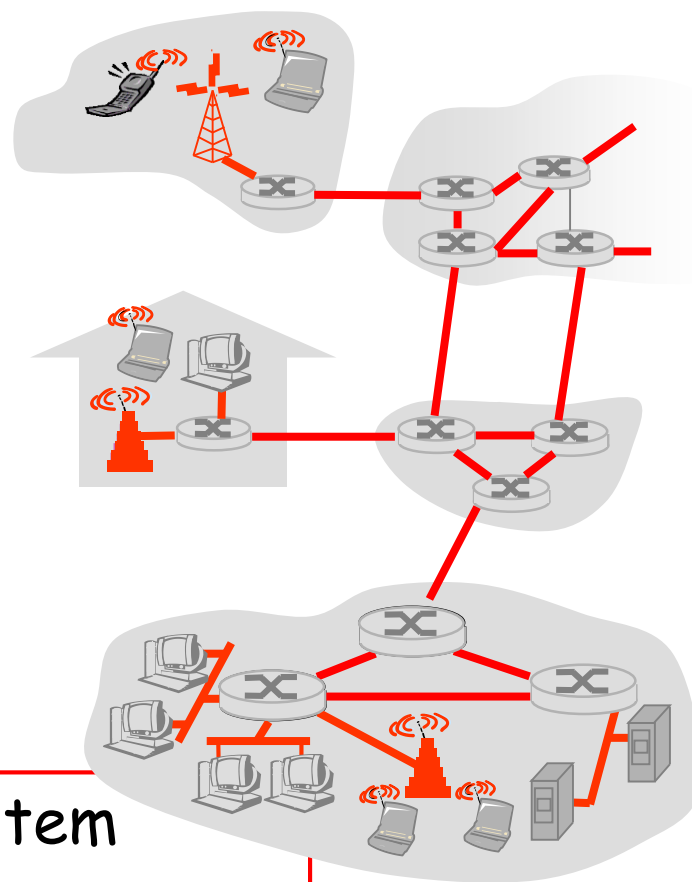
# Sumário

- ❑ 5.1 Introdução e serviços
- ❑ 5.2 Detecção e correcção de erros
- ❑ 5.3 Protocolos de acesso múltiplo
- ❑ 5.4 Endereçamento de Nível 2 e ARP
- ❑ 5.5 Ethernet
- ❑ 5.6 Concentradores, pontes e comutadores
- ❑ 5.7 Ligações sem fios e LAN's
- ❑ 5.8 PPP
- ❑ 5.9 Link virtualização: ATM, MPLS
- ❑ 5.10 Frame Relay

# Ligação de dados: Introdução

## Terminologia:

- ❑ Sistemas terminais e routers são **nós**
- ❑ Canais de comunicação que ligam nós adjacentes são **ligações**
  - Ligações com fios
  - Ligações sem fios
  - LANs
- ❑ O pacote de nível 2 é uma **trama**, que encapsula um datagrama



**A camada de ligação de dados** tem a responsabilidade de transferir datagramas de um nó para o nó adjacente através de uma ligação

# Ligação de dados: contexto

- ❑ Os datagramas são transferidos por diferentes protocolos em várias ligações:
  - e.g., Ethernet na 1ª ligação, frame relay nas ligações intermédias, 802.11 na última ligação
- ❑ Cada protocolo de ligação de dados oferece serviços diferentes
  - e.g., pode oferecer transferência fiável de dados ou não

## Analogia dos transportes

- ❑ Viagem de Lisboa para Lausanne
  - metro: Lisboa para Aeroporto
  - avião: Lisboa para Geneva
  - comboio: Geneva para Lausanne
- ❑ turista = **datagrama**
- ❑ Fase do transporte = **ligação da comunicação**
- ❑ Modo do transporte = **protocolo de ligação de dados**
- ❑ agência de viagem = **algoritmo de encaminhamento**

# Serviços do Nível de Ligação de Dados

## ❑ *Delimitação e acesso à linha:*

- Encapsula os datagramas em tramas, adição de cabeçalho e caudas nas tramas
- Realiza o acesso ao meio físico, no caso de canais partilhados
- Endereços físicos usados nos cabeçalhos das tramas para identificar a origem e destino das tramas
  - Diferentes dos endereços IP !!!

## ❑ *Transferência fiável entre dois nós adjacentes*

- Raramente utilizado em ligações com taxas de erros baixa (fibra, pares entrançados)
- Ligações sem fios: taxa de erros elevada
  - Q: porquê fiabilidade no nível de ligação de dados e ao nível extremo a extremo?

# Serviços do Nível de Ligação de Dados

## ❑ *Controlo de fluxo:*

- Adaptar o passo dos emissores e receptores

## ❑ *Detecção de erros:*

- Erros causados por atenuação do sinal ou ruído.
- Receptores detectam a presença de ruído:
  - Sinaliza emissor para retransmitir ou descarta trama

## ❑ *Correcção de erros:*

- Receptor identifica *e corrige* os bits errados sem recorrer à retransmissão

## ❑ *half-duplex e full-duplex*

- Em half duplex, os nós em ambos os extremos da linha podem transmitir, mas não em simultâneo

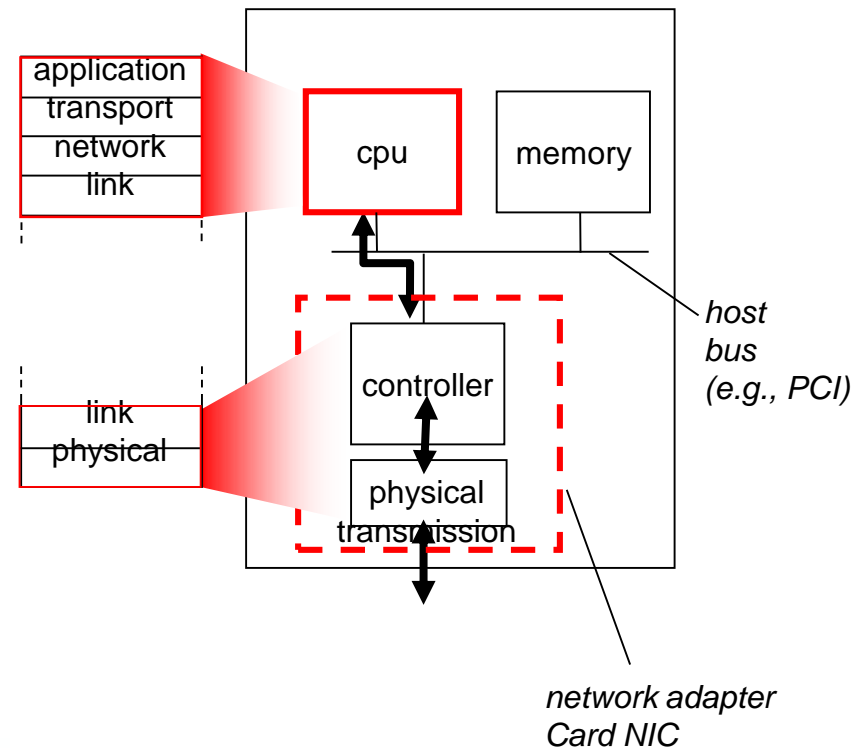


# Onde é realizado o nível de ligação de dados?

- ❑ Em cada posto
- ❑ Realizado no adaptador de rede (aka **network interface card NIC**)
  - Ethernet card, PCMCIA card, 802.11 card
  - Realiza os níveis de ligação de dados e físico
- ❑ ligado ao bus dos sistemas terminais
- ❑ Combinação de hardware, software, firmware



Esquema de um posto



# Sumário

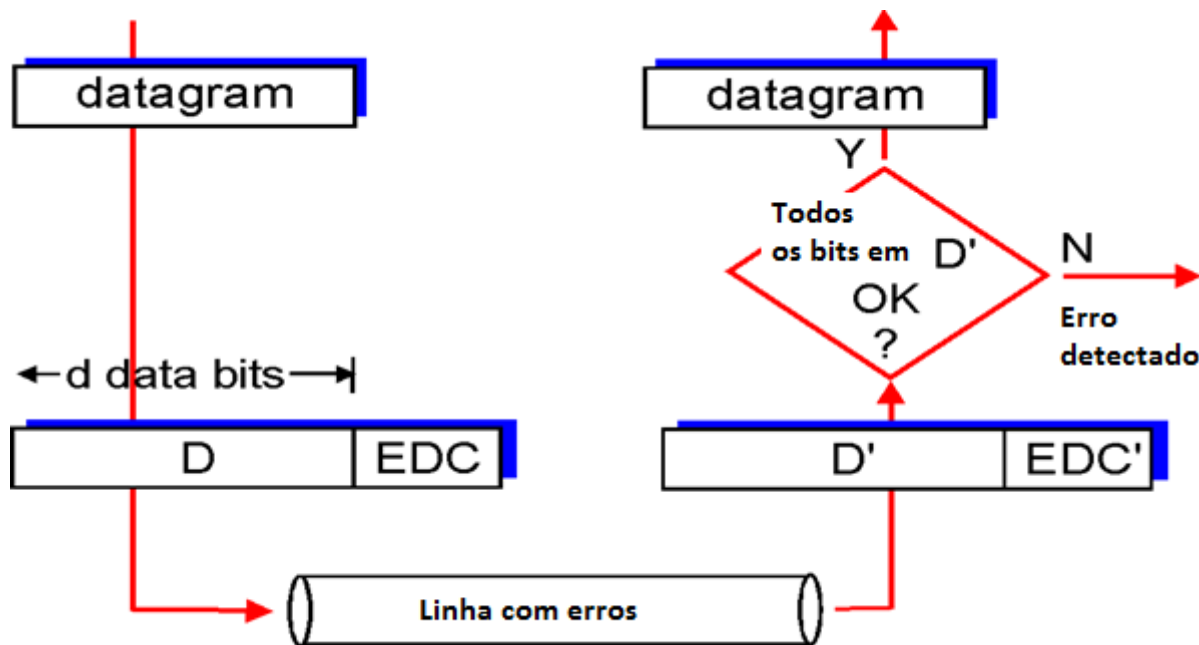
- ❑ 5.1 Introdução e serviços
- ❑ 5.2 Detecção e correcção de erros
- ❑ 5.3 Protocolos de acesso múltiplo
- ❑ 5.4 Endereçamento de Nível 2 e ARP
- ❑ 5.5 Ethernet
- ❑ 5.6 Concentradores, pontes e comutadores
- ❑ 5.7 Ligações sem fios e LAN's
- ❑ 5.8 PPP
- ❑ 5.9 Link virtualização: ATM, MPLS
- ❑ 5.10 Frame Relay

# Detecção de erros

**EDC** = **E**rror **D**etection and **C**orrection - bits de detecção e correcção de erros  
**D** = dados protegidos por verificação de erros, pode incluir campos de cabeçalho

A detecção de erros **não é 100% fiável!**

- o protocolo pode deixar passar erros, mas raramente
- um campo EDC maior resulta numa melhor detecção e correcção, mas com mais overhead

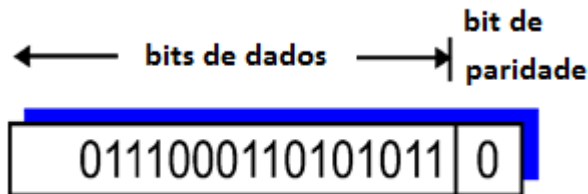


# Verificação de paridade

Combinação à priori  
do tipo de paridade  
(paridade par ou ímpar)

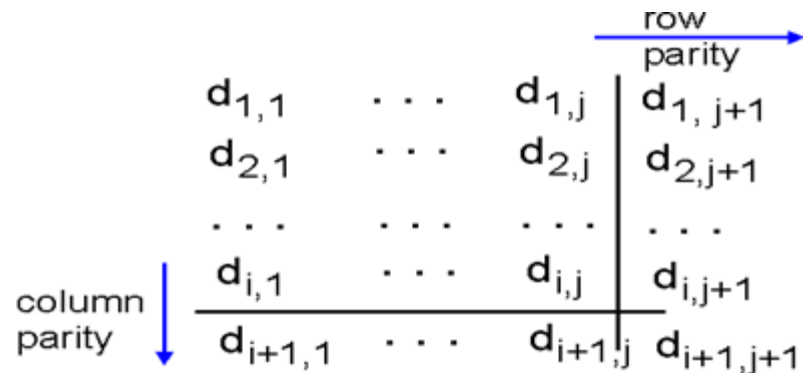
Bit de Paridade:

Detecta erros singulares  
(paridade ímpar)



Paridade bidimensional:

Detecta e corrige erros singulares  
(paridade par)



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

sem erros

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

Erro de  
paridade  
erro singular  
corrigível

Erro de  
paridade

# Checksum na Internet

objectivo: detectar "erros" (e.g., bits trocados) no segmento transmitido (utilizado apenas na camada de transporte)

## Emissor:

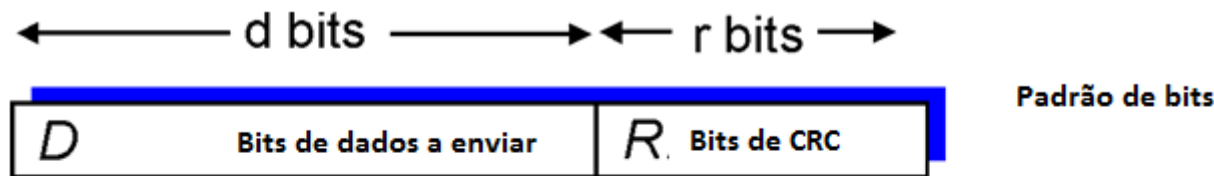
- ❑ Trata o conteúdo do segmento como uma sequência de inteiros de 16 bits
- ❑ checksum: soma do conteúdo do segmento em complemento para 1
- ❑ Emissor coloca o valor do checksum no campo respectivo do segmento UDP ou TCP

## Receptor:

- ❑ Calcula o checksum do segmento recebido
- ❑ Verifica se o valor calculado é igual ao enviado :
  - Não - erro detectado
  - SIM - não houve erros detectados

# Códigos cíclicos: Cyclic Redundancy Check

- ❑ Considera os bits de dados,  $D$ , como um número binário ou um polinómio
- ❑ Escolhe um padrão de  $r+1$  bits (polinómio gerador),  $G$
- ❑ objectivo: obter  $r$  bits de CRC,  $R$ , tais que
  - $\langle D, R \rangle$  sejam exactamente divisíveis por  $G$  (módulo 2)
  - O receptor conhece  $G$ , divide  $\langle D, R \rangle$  por  $G$ . Se o resto não for zero: erro detectado!
  - Pode detectar todas as rajadas (burst) de erros com menos de  $r+1$  bits
- ❑ Muito utilizado na prática (Ethernet, 802.11 WiFi, ATM)



$$D * 2^r \text{ XOR } R$$

Fórmula matemática

# CRC Exemplo

Quere-se:

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

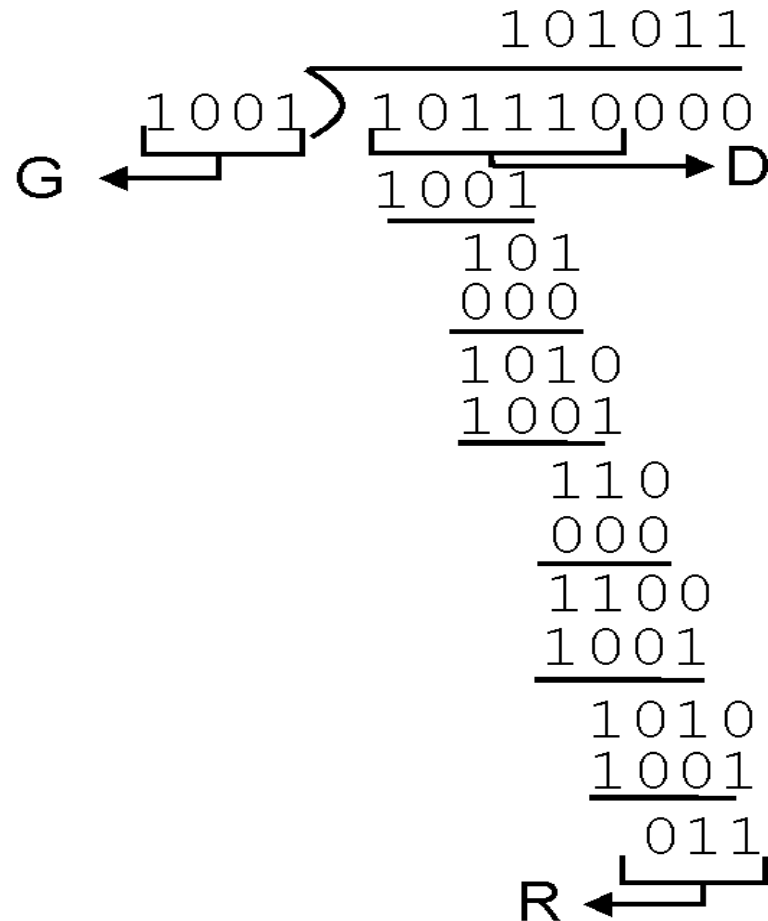
*Que é equivalente a:*

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

*Que é equivalente a:*

se se dividir  $D \cdot 2^r$   
por  $G$ , obtem-se o  
resto  $R$

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



# Sumário

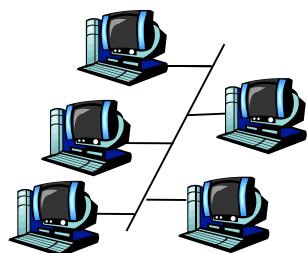
- ❑ 5.1 Introdução e serviços
- ❑ 5.2 Detecção e correcção de erros
- ❑ 5.3 Protocolos de acesso múltiplo
- ❑ 5.4 Endereçamento de Nível 2 e ARP
- ❑ 5.5 Ethernet
- ❑ 5.6 Concentradores, pontes e comutadores
- ❑ 5.7 Ligações sem fios e LAN's
- ❑ 5.8 PPP
- ❑ 5.9 Link virtualização: ATM, MPLS
- ❑ 5.10 Frame Relay



# Ligações e protocolos de acesso múltiplo

Dois tipos de "ligações":

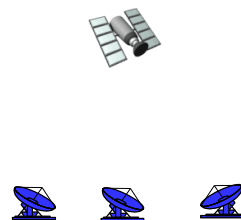
- ❑ **Ponto a ponto** (ligação física dedicada para a ligação)
  - PPP para acesso pelas linhas telefónicas
  - Ligação ponto a ponto entre comutador Ethernet e uma máquina
- ❑ **Difusão broadcast** (ligação física partilhada por várias ligações)
  - Ethernet tradicional
  - 802.11 wireless LAN



Cabo partilhado (e.g., Ethernet coax)



partilha RF  
(e.g., 802.11 WiFi)



partilha RF  
(satelite)



Pessoas em reunião  
(acústica)

# Protocolos de Acesso Múltiplo

- ❑ Canal único e partilhado para difusão
- ❑ Duas ou mais transmissões simultâneas em cada nó:
  - **interferência**
  - **colisão** se o nó recebe dois ou mais sinais em simultâneo

## Protocolo de acesso múltiplo

- ❑ Algoritmo distribuído que determina como é que os nós partilham o canal, i.e., quando um nó pode transmitir
- ❑ Comunicação sobre a partilha do canal tem de usar o próprio canal!
- ❑ O que procurar num protocolo de acesso múltiplo?
  - Sincrono ou assíncrono
  - Necessidade de informação sobre os outros nós
  - Robustez e desempenho

# Protocolo de acesso múltiplo ideal

## Canal de difusão com ritmo $R$ bps

1. Quando apenas um nó quer transmitir, pode enviar dados ao ritmo  $R$ .
2. Quando  $M$  nós querem transmitir, cada nó pode enviar a um ritmo médio de  $R/M$
3. Totalmente descentralizado:
  - não há um nó especial para coordenar as transmissões
  - Não há sincronização de relógios nem slots
4. simples

# Protocolos MAC: taxonomia

Três classes de protocolos:

## ❑ Partição de canal - Channel Partitioning

- Divide o canal em partes mais pequenas
  - No tempo (time slots) fatias de tempo por canal - **TDMA**
  - Na frequência - bandas de frequência por canal - **FDMA**
  - Em códigos - códigos diferentes por canal - **CDMA**
- Atribui cada parte exclusivamente a um nó

## ❑ Acesso aleatório - Random Access

- Canal não dividido, permite colisões
- Permite recuperar das colisões

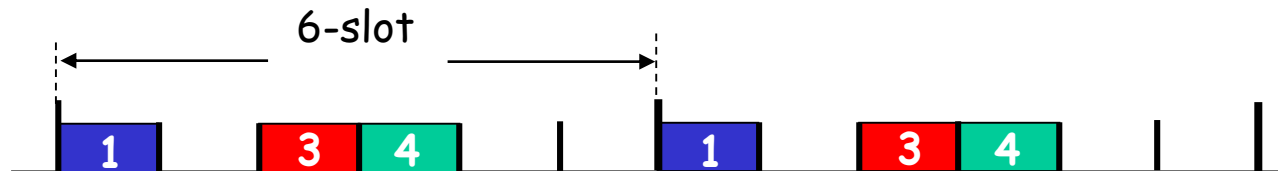
## ❑ Por turnos - "Taking turns"

- Acesso muito coordenado ao meio partilhado, evitando colisões, por divisão no tempo.

# Protocolos MAC de partição de canal: TDMA

## TDMA: time division multiple access

- ❑ Acesso ao canal por "ciclos"
- ❑ Cada nó obtém uma fatia de tempo (slot) de dimensão fixa em cada ciclo
- ❑ Slots não utilizados vão vazios
- ❑ Exemplo: LAN com 6 nós
  - nós 1,3,4 têm pacotes para transmitir
  - slots 2,5,6 vazios



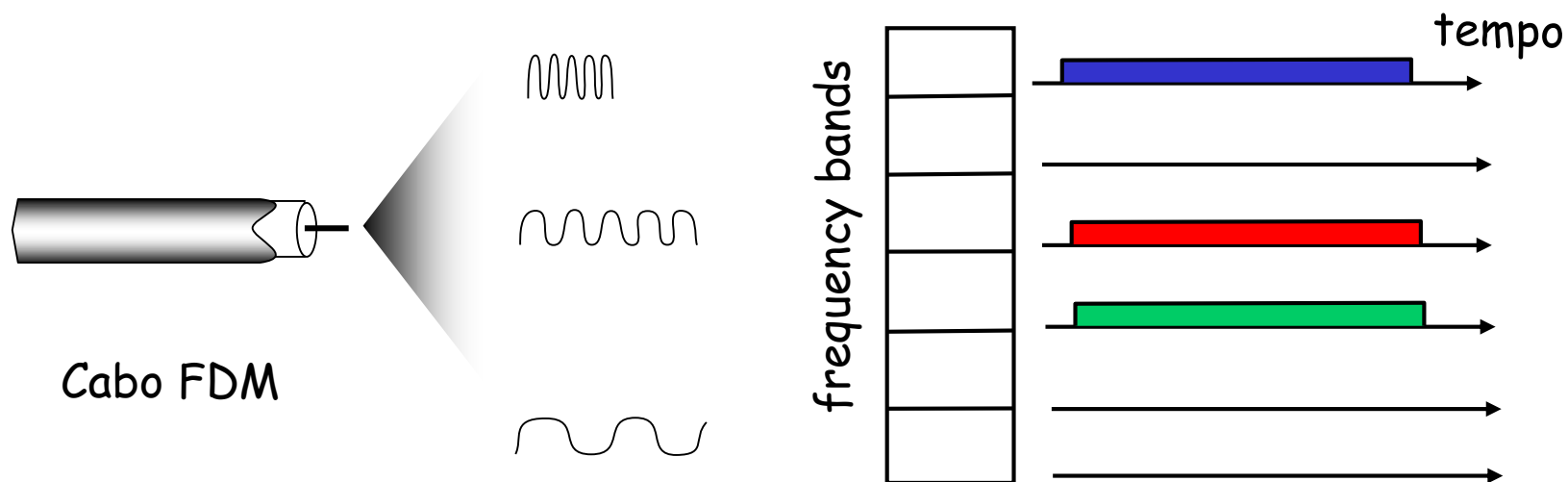
**TDMA - Canal dividido em  $N$  slots, um por estação, é ineficiente:**

- Longos ciclos de espera,
- Slots vazios não aproveitados

# Protocolos MAC de partição de canal: FDMA

## FDMA: frequency division multiple access

- ❑ O espectro do canal é dividido em bandas de frequência
- ❑ A cada nó está atribuída uma banda de frequência
- ❑ Banda de frequência vazia se posto não transmite
- ❑ exemplo: LAN com 6 estações, 1,3,4 têm pacotes para transmitir, bandas de frequência 2,5,6 estão vazias



# Protocolos MAC de partição de canal: CDMA

## CDMA: Code division multiple access

- ❑ Código único atribuído a cada estação
- ❑ Usado essencialmente em canais de difusão sem fios (redes celulares, satélite, etc)
- ❑ Todas as estações partilham a mesma frequência, mas possuem um código próprio para codificar os seus dados
- ❑ Sinal codificado = dados originais  $\times$  sequência de código
- ❑ Sinal decodificado = produto interno entre o sinal codificado e a sequência do seu código
- ❑ Permite que múltiplas estações coexistam, transmitam em simultâneo com um mínimo de interferência.

# Protocolos de acesso aleatório

- ❑ Quando o nó tem pacotes para transmitir
  - Transmite ao ritmo máximo do canal R.
  - Não há coordenação *a priori* entre os nós
- ❑ Dois ou mais nós a transmitir → "colisão",
- ❑ **Protocolos MAC de acesso** especificam:
  - Como detectar colisões
  - Como recuperar de colisões (e.g., através de atrasos na retransmissão)
- ❑ Exemplos de protocolos MAC de acesso aleatório:
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA



# CSMA (Carrier Sense Multiple Access)

**CSMA**: Escuta antes de transmitir:

- ❑ Se o canal escutado estiver livre: transmite a trama
- ❑ Se o canal escutado estiver ocupado: atrasa a transmissão
  - CSMA persistente: tenta imediatamente quando o canal fica livre (pode causar instabilidade)
  - CSMA não persistente: tenta após um intervalo aleatório
- ❑ Analogia humana: Não interromper os outros!

# Colisões no CSMA

As colisões podem ocorrer porque:

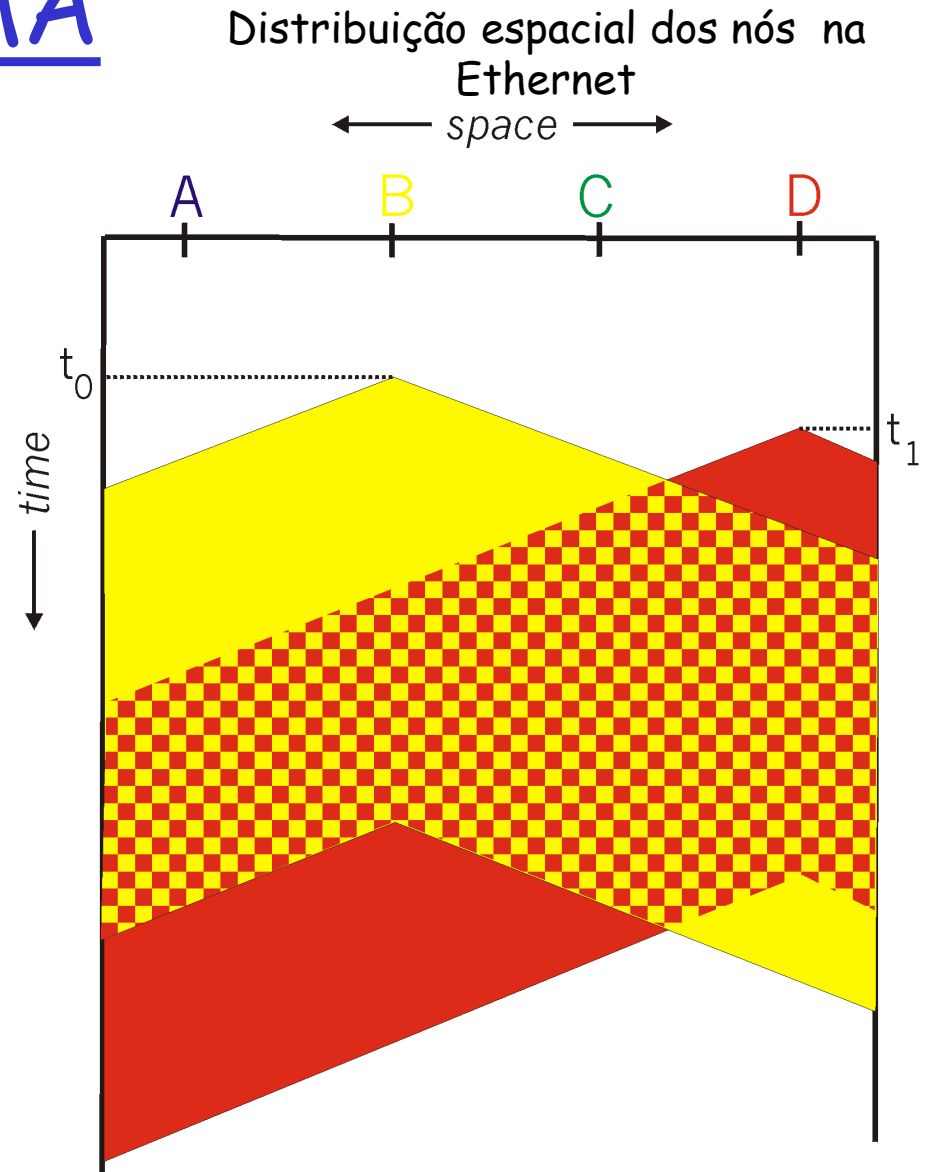
Atrasos de propagação significam que os nós podem não se "ouvir" uns aos outros enquanto transmitem

**Colisão:**

Todo o tempo de transmissão é desperdiçado

**nota:**

Importância da distância e do tempo de propagação na determinação da probabilidade de colisão



# CSMA/CD (Collision Detection)

## CSMA/CD:

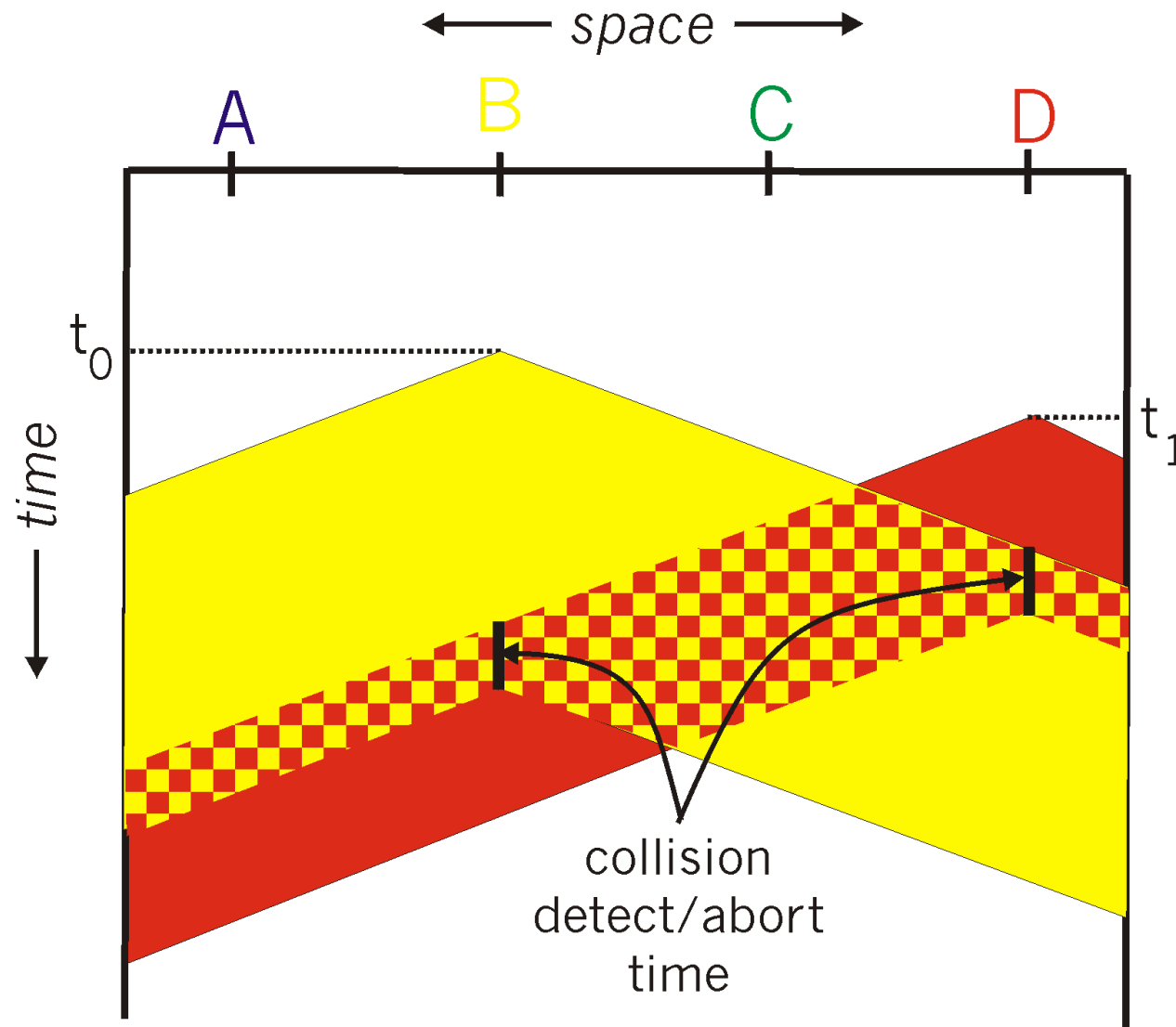
- Colisões detectadas num tempo curto
- Transmissões em tempo de colisão são abortadas, reduzindo o desperdício de ocupação do canal

### □ Detecção de colisão:

- Fácil nas LANs com fios:
  - Medição da potência do sinal
  - Comparação do sinal recebido com o transmitido
- Difícil em redes sem fios:
  - Potência do sinal pode variar durante a transmissão
  - Recepção desligada durante a transmissão

### □ Analogia humana: o conversador educado

# CSMA/CD detecção de colisão



# Protocolos MAC por turnos

## Protocolos MAC de partição de canal:

- Partilham o canal de forma eficiente para carga elevada
- Ineficiente para cargas baixas: atrasos no acesso ao canal,  $1/N$  da largura de banda atribuída a cada nó, mesmo quando há apenas 1 nó activo!

## Protocolos MAC de acesso aleatório

- Eficiente para cargas baixas: um único nó pode utilizar a totalidade do canal
- Carga elevada: sobrecarga de colisões

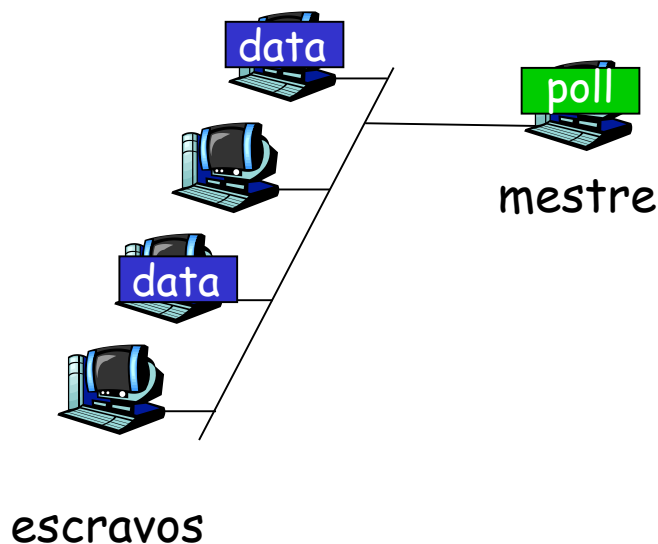
## Protocolos "por turnos"

Procura o melhor dos dois mundos!

# Protocolos MAC por turnos

## Polling:

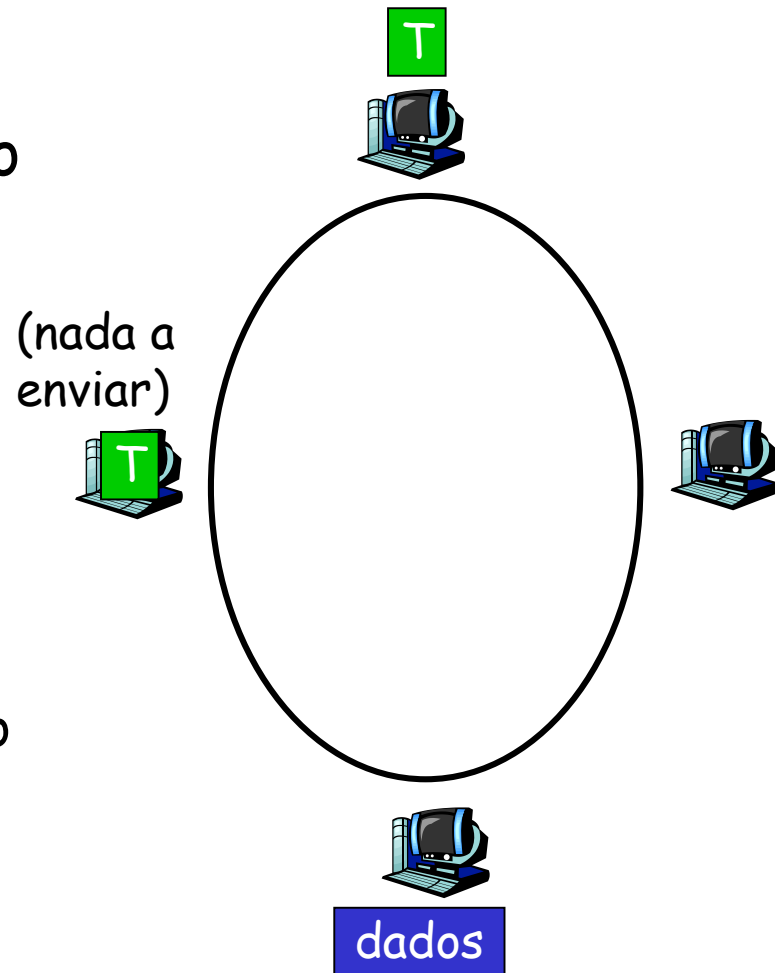
- ❑ Nó mestre "convida" nós "escravos" a transmitir
- ❑ Tipicamente utilizado em escravos "estupidos"
- ❑ Mensagens:
  - Clear to Send (CTS)
  - Request to Send (RTS)
- ❑ problemas:
  - Overhead do polling
  - latência
  - Único ponto de falha (mestre)



# Protocolos MAC por turnos

## Passagem de testemunho (token passing):

- ❑ Testemunho de controlo transferido sequencialmente entre nós.
- ❑ Mensagem de testemunho
- ❑ Problemas:
  - Overhead do testemunho
  - latência
  - Único ponto de falha (token)



# Sumário dos protocolos MAC

- ❑ *O que fazer com um meio partilhado?*
- ❑ *Partição do canal*, por tempo, frequência ou código  
TDMA, FDMA, CDMA
- ❑ *Partição aleatória* (dinâmica),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - Escuta da portadora: fácil em algumas tecnologias (cabos), difícil noutras (wireless)
  - CSMA/CD usado na Ethernet
  - CSMA/CA usado na 802.11 (wireless)
- ❑ *Por turnos*
  - polling a partir de um posto central,
  - passagem de testemunho
  - Bluetooth, FDDI, IBM Token Ring



# Sumário

- ❑ 5.1 Introdução e serviços
- ❑ 5.2 Detecção e correcção de erros
- ❑ 5.3 Protocolos de acesso múltiplo
- ❑ 5.4 Endereçamento de Nível 2 e ARP
- ❑ 5.5 Ethernet
- ❑ 5.6 Concentradores, pontes e comutadores
- ❑ 5.7 Ligações sem fios e LAN's
- ❑ 5.8 PPP
- ❑ 5.9 Link virtualização: ATM, MPLS
- ❑ 5.10 Frame Relay

# Endereços MAC e ARP

## ❑ Endereços IP de 32 ou 128 bit :

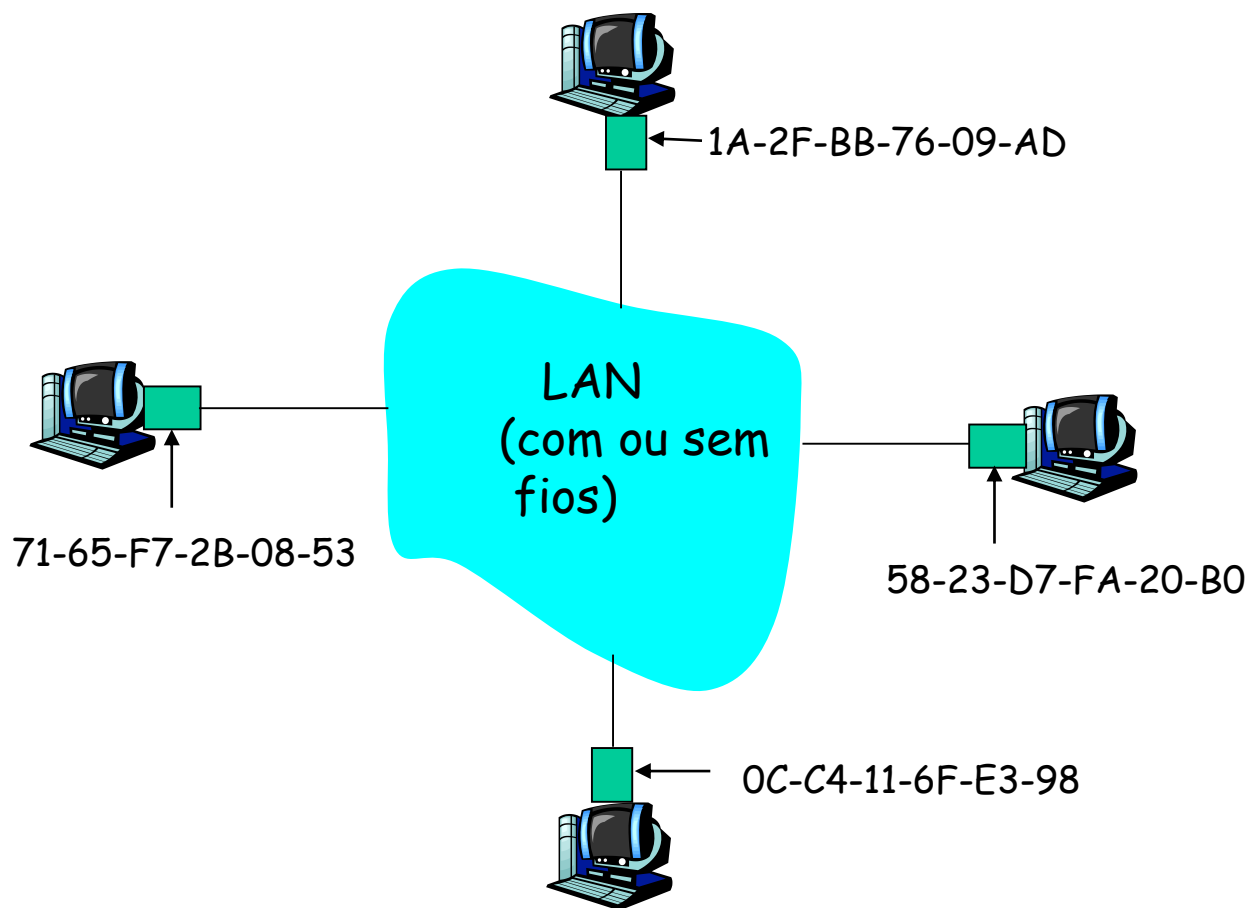
- Endereços de nível de rede
- Utilizados para enviar os datagramas até à rede de destino

## ❑ Endereços MAC (ou LAN ou físicos ou Ethernet):

- função: enviar o datagrama de uma interface para outra interface fisicamente ligada à mesma rede
- Endereço MAC de 48 bits Ex: 1A-2F-BB-76-09-AD
  - gravado na ROM da placa de rede

# Endereços de LAN e ARP

Cada placa de rede numa LAN tem um endereço MAC diferente



Endereço de broadcast=  
FF-FF-FF-FF-FF-FF

■ = placa de rede

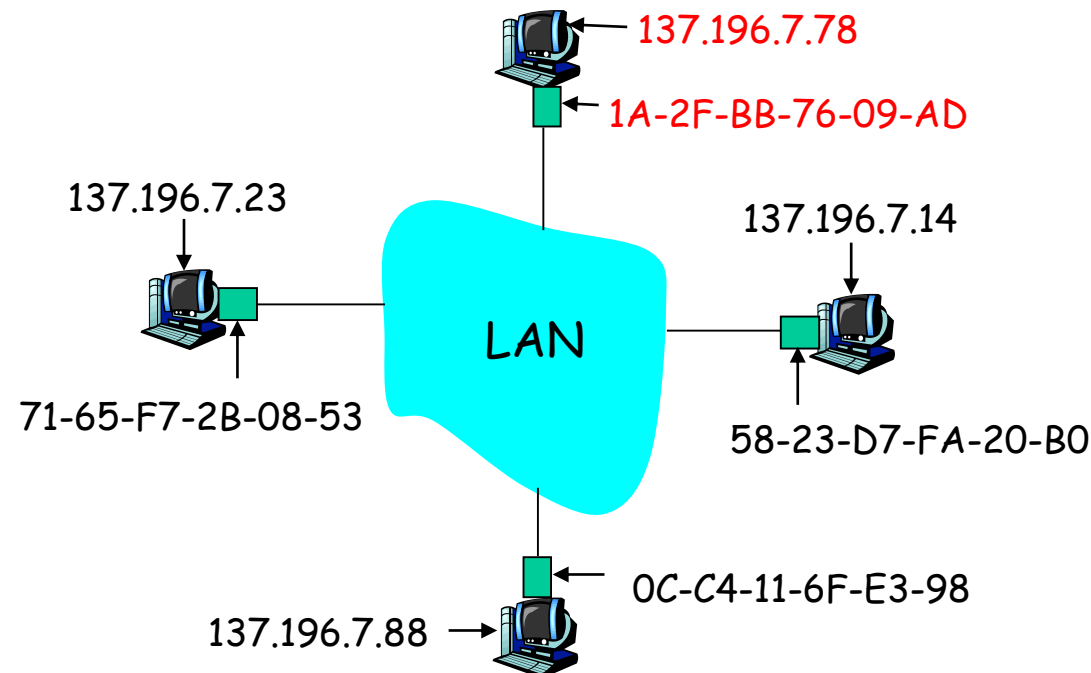
# Endereços de LAN

- ❑ Atribuição de endereços MAC administrada pelo IEEE
- ❑ Fabricantes compram partes do espaço de endereços MAC para garantir a unicidade
- ❑ analogia:
  - (a) endereços MAC: como o n° do BI
  - (b) endereços IP: como endereço postal
- ❑ Endereçamento **MAC é plano** → portabilidade
  - Pode transportar uma carta de rede de uma LAN para outra, sem efectuar nenhuma alteração
- ❑ Endereçamento **IP tem hierarquia** e não é transportável
  - endereço depende da rede IP a que se está ligado

# ARP: Address Resolution Protocol

Questão: como saber o endereço MAC de B sabendo o endereço IP de B?

- Cada nó IP (posto, router) na LAN tem uma tabela **ARP**
- tabela ARP :  
mapeamento de endereços IP/MAC para os nós  
< IP address; MAC address; TTL >
  - TTL (Time To Live):  
tempo após o qual o mapeamento de endereços é "esquecido" (20 min típico)



# ARP: Address Resolution Protocol

*Resultado do Comando arp -a*

```
PS C:\Windows\system32> arp -a

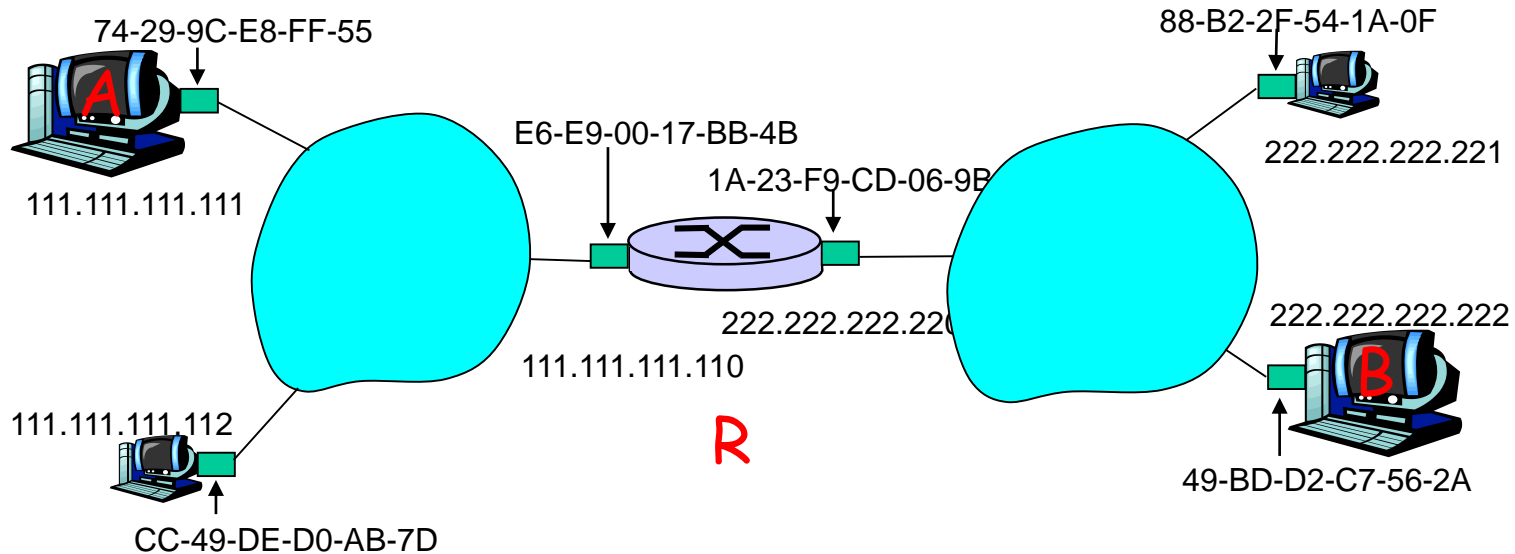
Interface: 192.168.1.67 --- 0xc
Internet Address      Physical Address      Type
192.168.1.65          dc-66-72-9d-c6-4b     dynamic
192.168.1.254         00-06-91-cf-c7-0f     dynamic
224.0.0.2             01-00-5e-00-00-02     static
224.0.0.22            01-00-5e-00-00-16     static
224.0.0.251           01-00-5e-00-00-fb     static
224.0.0.252           01-00-5e-00-00-fc     static
239.255.255.250       01-00-5e-7f-ff-fa     static
255.255.255.255       ff-ff-ff-ff-ff-ff     static
PS C:\Windows\system32>
```

# Protocolo ARP: mesma LAN (rede)

- ❑ **Posto A** quer enviar datagrama para **posto B**, e o MAC não está na tabela ARP de A
- ❑ **Posto A** envia pedido **broadcasts** ARP, contendo o endereço IP de B
  - dest MAC address = FF-FF-FF-FF-FF-FF
  - Todas as máquinas na rede LAN recebem o pedido ARP
- ❑ **Posto B** recebe o pedido ARP, responde apenas a A com o seu endereço MAC (unicast)
- ❑ **Posto A** guarda o par IP/MAC na sua memória (tabela cache ARP) até esta informação ficar velha (timeout)
  - soft state: a informação que expira é descartada, a menos que seja refrescada
- ❑ ARP é "plug-and-play":
  - Os nós criam as suas tabelas ARP *sem intervenção do administrador de rede*

# Encaminhamento para outra LAN

Passo a passo: **envia datagrama de A para B via R**  
assume-se que A sabe endereço IP de B

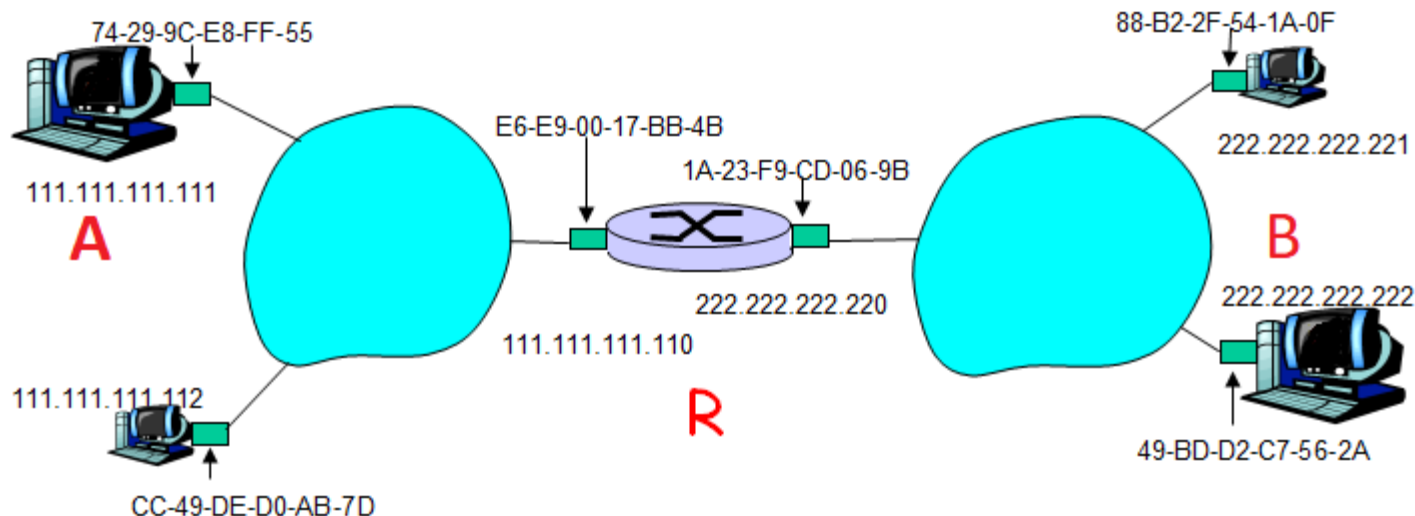


- **Duas tabelas ARP no router R**, uma para cada rede IP (LAN)



- ❑ **A** cria datagrama IP com origem **A** e destino **B**
- ❑ **A** usa ARP para obter o endereço MAC de **R** (111.111.111.110)
- ❑ **A** cria uma trama Ethernet com o MAC de **R** como destino, a trama tem o datagrama IP de **A-para-B**
- ❑ Placa de rede de **A** envia trama
- ❑ Placa de rede de **R** recebe trama
- ❑ **R** remove o datagrama IP da trama Ethernet, e verifica que é destinada para **B**
- ❑ **R** usa ARP na LAN2 para obter o MAC de **B**
- ❑ **R** cria trama que contem o datagrama a enviar de **A** para **B**

Isto é **muito** importante  
- tenha a certeza que  
compreendeu!



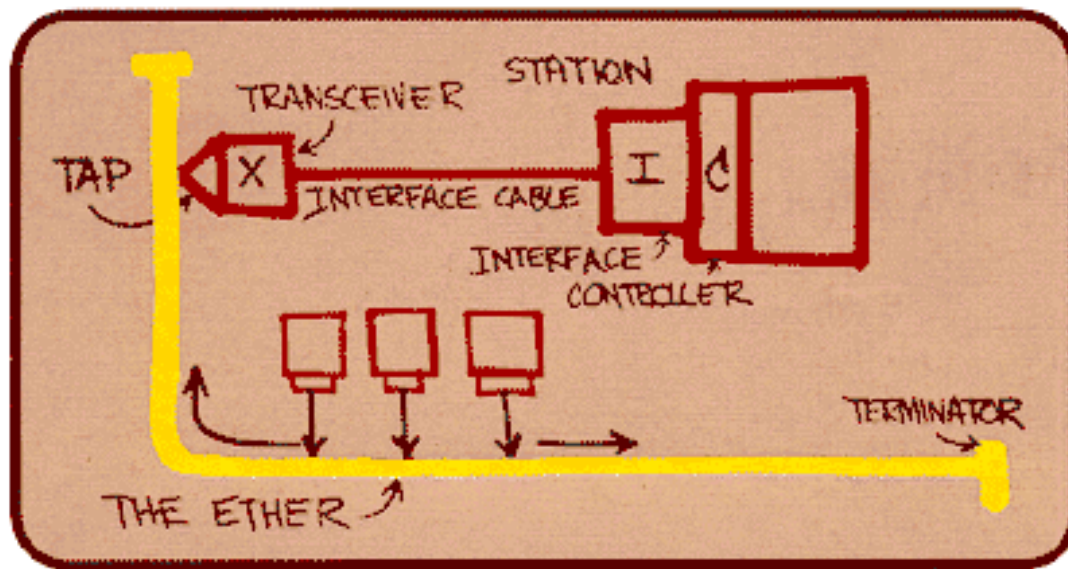
# Sumário

- ❑ 5.1 Introdução e serviços
- ❑ 5.2 Detecção e correcção de erros
- ❑ 5.3 Protocolos de acesso múltiplo
- ❑ 5.4 Endereçamento de Nível 2 e ARP
- ❑ 5.5 Ethernet
- ❑ 5.6 Concentradores, pontes e comutadores
- ❑ 5.7 Ligações sem fios e LAN's
- ❑ 5.8 PPP
- ❑ 5.9 Link virtualização: ATM, MPLS
- ❑ 5.10 Frame Relay

# Ethernet

Tecnologia "dominante" de LAN:

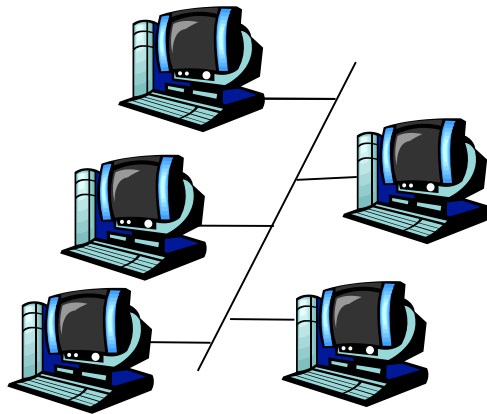
- ❑ Barata: carta de rede a 20€ para 100Mbps
- ❑ Primeira tecnologia de LAN amplamente utilizada
- ❑ Mais simples e barata que LAN's com testemunho e ATM
- ❑ Acompanha o aumento das velocidades: 10 Mbps - 10 Gbps



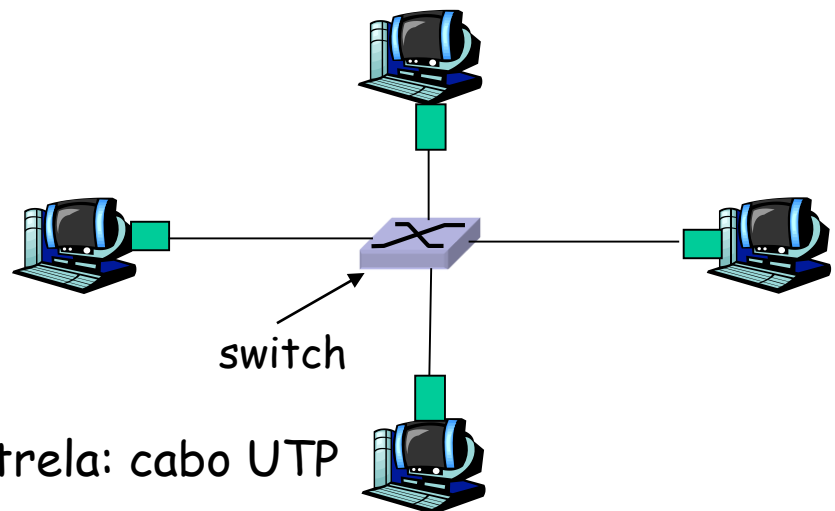
Esboço da Ethernet  
realizado por Metcalfe

# Topologia em estrela

- ❑ Topologia em BUS utilizada até fins da década de 90
  - Todos os nós no mesmo domínio de colisão (as mensagens de um nó pode colidir com as dos outros nós)
- ❑ hoje: prevalência da topologia em estrela
  - *switch* activos no centro
  - Cada nó usa um troço Nível 2 próprio (nós não colidem com outros)



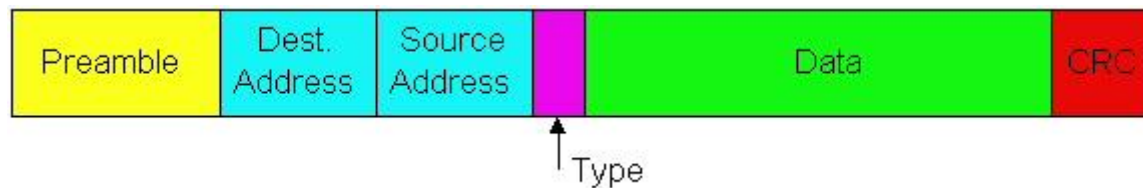
bus: cabo coaxial



estrela: cabo UTP

# Estrutura de uma trama Ethernet

A placa de rede do emissor encapsula datagrama IP (ou pacote de outro protocolo de nível de rede) numa **trama Ethernet**

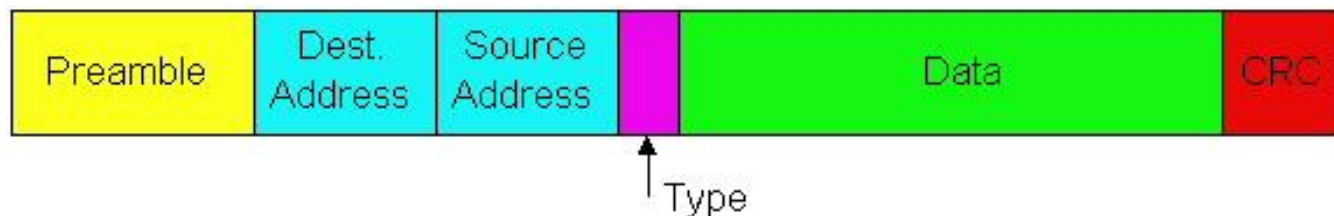


## **Preâmbulo (Preamble):**

- ❑ 7 bytes com padrão 10101010 seguidos de um byte com padrão 10101011
- ❑ usado para sincronizar os ritmos de relógio do emissor e do receptor

# Estrutura de uma trama Ethernet

- ❑ **Endereços:** 6 bytes cada
  - Se a placa de rede recebe uma trama com o seu endereço de destino, ou com endereço de broadcast (eg pacote ARP), passa os dados da trama para o protocolo de camada de rede
  - De outra forma, a placa de rede descarta a trama
- ❑ **Type:** (2 bytes) indica o protocolo de nível superior
  - Normalmente IP
  - Suporta outros protocolos (Novell IPX, AppleTalk, ...)
- ❑ **CRC:** (4 bytes) verificado no receptor. Se se detectarem erros, a trama é descartada



# Ethernet: sem ligação, não fiável

- ❑ **Sem ligação:** Não há "handshaking" entre placas de rede do emissor e do receptor - não garante a entrega!!!
  
- ❑ **Não fiável:** a placa de rede do receptor não confirma as recepções positivas (ACK) ou negativas (NACK)
  - Fluxo de datagramas passado para a camada superior pode ter falhas (datagramas em falta)
  - As falhas podem ser preenchidas se a aplicação usar TCP
  - De outra forma, a aplicação verá as falhas

# Ethernet usa CSMA/CD

- ❑ Sem slots
- ❑ Uma placa de rede não transmite se detectar que outra placa está a transmitir. Isto chama-se **detecção da portadora (carrier sense - CS)**
- ❑ Uma placa de rede não espera autorização para transmitir. Isto chama-se **acesso múltiplo (multiple access - MA)**
- ❑ Uma placa de rede em transmissão aborta se detectar outra placa de rede em actividade. Isto chama-se de **detecção de colisão (collision detection - CD)**
- ❑ Antes de tentar a retransmissão, a placa espera um tempo aleatório. Isto chama-se de **acesso aleatório (random access - RA)**



# Algoritmo CSMA/CD Ethernet

1. A placa de rede recebe o datagrama e cria a trama
2. Se a placa de rede detecta o canal desocupado, começa a transmitir a trama. Se detecta o canal ocupado, espera que este fique livre e então transmite
3. Se a placa de rede transmite a trama inteira sem detectar outra transmissão, a placa de rede acabou de servir a trama!
4. Se a placa de rede detecta outra transmissão enquanto transmite, aborta e envia um sinal de reforço de colisão (jam signal)
5. Depois de abortar, a placa de rede entra em **retenção exponencial (exponential backoff)**: depois da m-ésima colisão, a placa de rede escolhe um K aleatório de  $\{0,1,2,\dots,2^m-1\}$ . A placa espera  $K \times 512$  tempos de bit e volta ao passo 2

# Algoritmo CSMA/CD Ethernet

**Sinal de reforço de colisão (Jam Signal):** serve para assegurar que todas as outras transmissões toma conhecimento da colisão;  
48 bits

**Tempo de Bit:** .1 microsec para Ethernet a 10 Mbps;  
para  $K=1023$ , a espera é de cerca de 50 mseg

## **Retenção exponencial (Exponential Backoff)**

- ❑ **objectivo:** adaptar as tentativas de retransmissão à situação de carga na rede
  - Carga elevada: espera aleatória será elevada.
- ❑ Primeira colisão: escolhe  $K$  entre  $\{0,1\}$ ; atraso é  $K \cdot 512$  tempos de bit
- ❑ Depois da 2ª colisão: escolhe  $K$  de  $\{0,1,2,3\}$ ...
- ❑ Depois de 10 ou mais colisões, escolhe  $K$  de  $\{0,1,2,3,4,...,1023\}$

# Algoritmo CSMA/CD Ethernet

A: sense channel

if idle

then {

transmite and monitor channel;

if detect another transmission

then {

abort and send jam signal

Update # collisions

delay as required by exponential backoff algorithm

goto A

}

else done with this frame; set collision to zero

}

else (wait until ongoing transmission is over and goto A

# Eficiência CSMA/CD

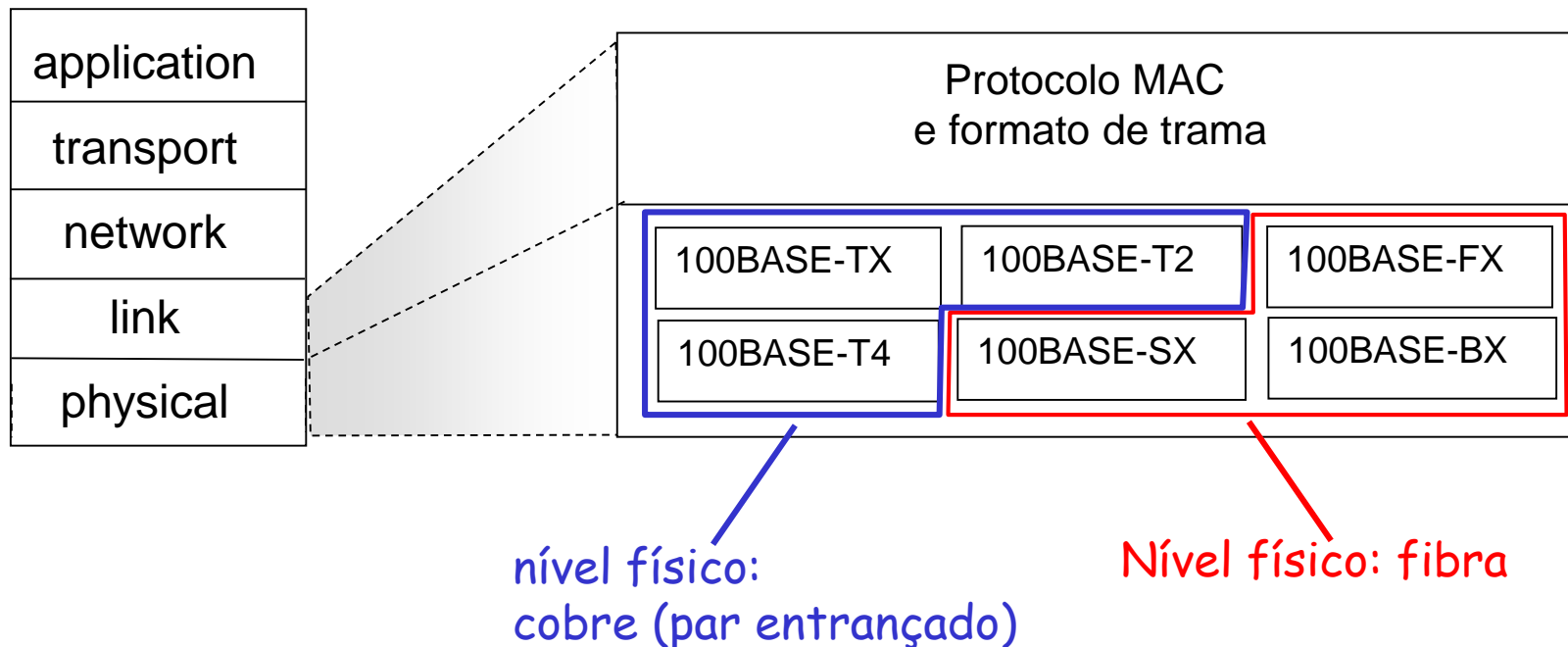
- $T_{\text{prop}}$  = tempo de propagação máximo entre 2 nós na LAN
- $t_{\text{trans}}$  = tempo de transmissão de trama de dimensão máxima

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

- Eficiência tende para 1
  - Quando  $t_{\text{prop}}$  tende para 0
  - Quando  $t_{\text{trans}}$  tende para infinito
- Muito melhor performance que ALOHA: mas ainda mais simples, barato e descentralizado!

# Standards Ethernet 802.3 : níveis de ligação de dados e físico

- ❑ **Muitos** diferentes standards Ethernet
  - Protocolo MAC e formato de trama comum
  - Velocidades diferentes: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10Gbps
  - Meios físicos diferentes: fibra, cabo de pares

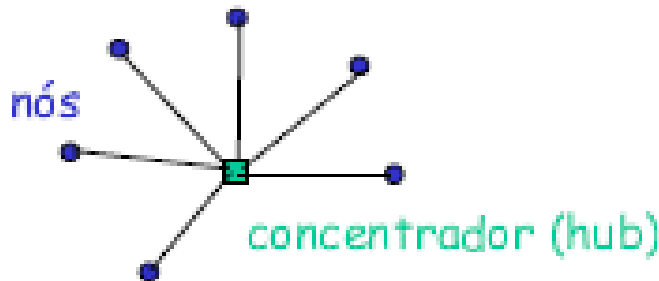


# Tecnologia Ethernet:

<b>100base...</b>	<p><b>TX:</b> Usa dois pares de cobre de um sistema de cablagem tipo 5 ou superior, o comprimento máximo de um segmento é 100 metros.</p> <p><b>FX:</b> Usa duas fibras multimodo, o comprimento máximo de um segmento é 2 Km.</p>
<b>1000base...</b>	<p><b>T:</b> Usa quatro pares de cobre de um sistema de cablagem tipo 5E ou superior, o comprimento máximo de um segmento é 100 metros. Não suporta <i>full-duplex</i>.</p> <p><b>SX:</b> Usa duas fibras multimodo, o comprimento máximo de um segmento é 220 metros ou 550 metros, respetivamente para fibras de 62,5 ou 50 microns.</p> <p><b>LX:</b> Usa duas fibras monomodo, o comprimento máximo de um segmento é de 5 Km, mas pode ser superior, de acordo com as especificações do fabricante.</p>
<b>10Gbase...</b>	<p><b>SR/LRM/LR/ER/LX4:</b> são várias normas correspondentes a diversos tipos de fibra ótica que resultam em várias distâncias máximas suportadas que podem ir desde as dezenas de metros até à centena de Km.</p> <p><b>CX4/Kx/T:</b> utilizam vários tipos de cablagem de cobre especial, com características elétricas especiais, o 10GbaseT usa 4 pares CAT6A</p>
<b>40Gbase...</b>	<p><b>CR4/SR4/LR4:</b> a primeira usa 4 cabos de cobre coaxiais de características especiais, a segunda utiliza quatro fibras multimodo e a terceira utiliza quatro fibras monomodo.</p>
<b>100Gbase...</b>	<p><b>CR10/SR10/LR4/ER4:</b> As primeiras duas utilizam, respetivamente, 10 cabos de cobre coaxial especial e 10 fibras multimodo. As duas últimas utilizam quatro fibras monomodo e diferem no comprimento máximo que podem atingir</p>

# 10BaseT e 100BaseT

- ❑ 10/100: Ritmos de 10/100 Mbps
  - 100 Mbps é chamada de "fast Ethernet"
- ❑ Base: transmissão em banda de base (sem modulação)
- ❑ T: Twisted pair (pares de fios entrançados)
- ❑ Nós são ligados através de um concentrador (HUB) utilizando pares entrançados numa topologia em estrela
- ❑ 100 metros de distância máxima entre nós e concentrador



# 10BaseT e 100BaseT

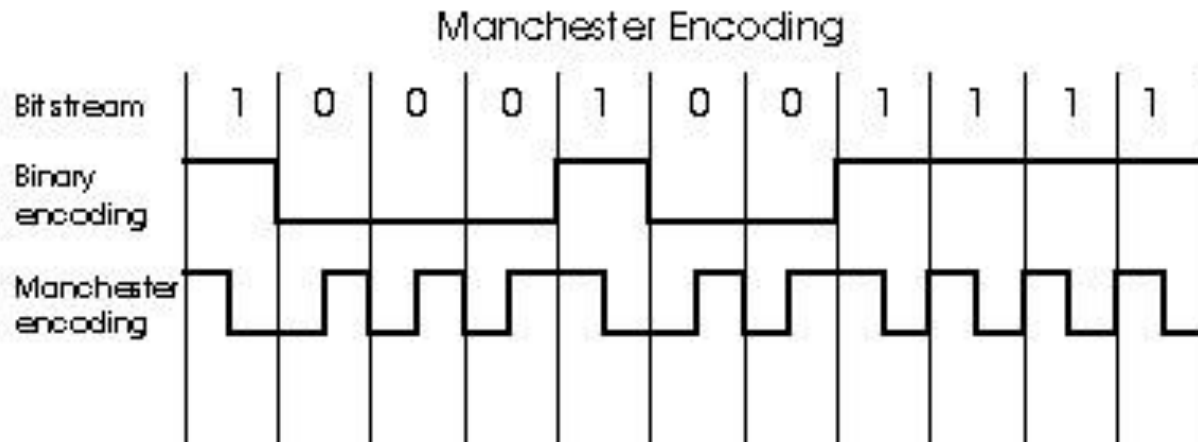
- Os concentradores (hub) são essencialmente repetidores de nível físico:
  - Os bits que chegam numa linha de entrada saem em todas as outras linhas
  - Não há memória (buffer) para as tramas
  - Não há CSMA/CD no concentrador: as placas de rede detectam colisões - domínio de colisão
  - Pode fornecer funcionalidades de gestão de rede



# Gigabit Ethernet

- ❑ Usa o formato de trama standard da Ethernet
- ❑ Permite linhas ponto a ponto e canais de difusão partilhados
- ❑ No modo partilha, usa CSMA/CD: para ser eficiente, a distância entre nós deve ser curta
- ❑ Usa concentradores (hub), chamados de "buffered Distributors"
- ❑ Full duplex a 1 Gbps para linhas ponto a ponto
- ❑ 10 Gbps já existente

# Codificação Manchester



- ❑ Usado em 10BaseT e 10Base2
- ❑ Cada bit tem uma transição de valor positivo para negativo
- ❑ Permite que os relógio emissor e receptor se sincronizem
  - Não é necessário relógio global e centralizado para todos os nós!
- ❑ Isto faz parte do Nível Físico!

# Sumário

- ❑ 5.1 Introdução e serviços
- ❑ 5.2 Detecção e correcção de erros
- ❑ 5.3 Protocolos de acesso múltiplo
- ❑ 5.4 Endereçamento de Nível 2 e ARP
- ❑ 5.5 Ethernet
- ❑ 5.6 Concentradores, pontes e comutadores
- ❑ 5.7 Ligações sem fios e LAN's
- ❑ 5.8 PPP
- ❑ 5.9 Link virtualização: ATM, MPLS
- ❑ 5.10 Frame Relay

# Interligação de segmentos de LANs

## Q: porque não uma LAN grande?

- ❑ Quantidade de tráfego suportado é limitada
  - Numa só LAN todas as estações têm de partilhar a mesma largura de banda (**difusão**)
- ❑ Domínio de colisão grande - ineficiente
- ❑ Podem ocorrer colisões com muitas estações
- ❑ Comprimento limitado
  - 802.3 especifica comprimento máximo para o cabo
- ❑ N° limitado de estações

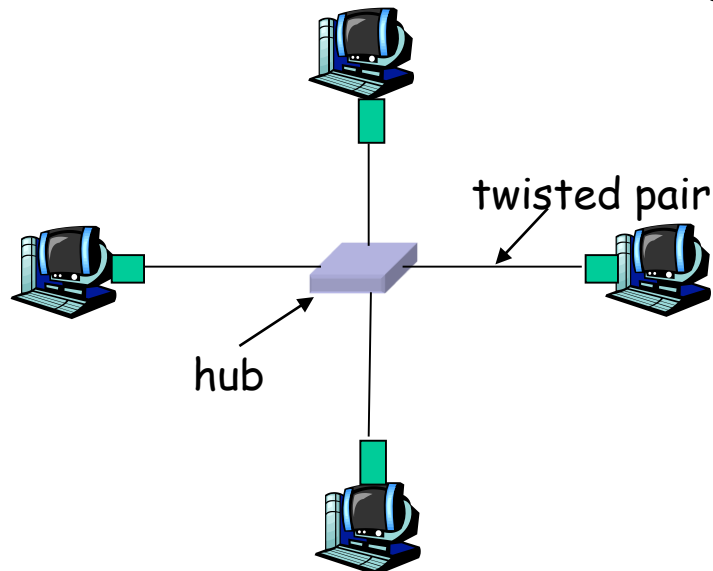
# Interligação de segmentos de LANs

- ❑ Concentradores (HUBs)
- ❑ Pontes (Bridges)
- ❑ Comutadores (Switches)
  - Comutadores são pontes com múltiplas portas

# Concentradores - Hubs

... repetidores "estúpidos" de nível físico:

- Os bits que chegam numa linha de entrada saem em todas as outras linhas
- Não há memória (buffer) para as tramas
- Não há CSMA/CD no concentrador: as placas de rede detectam colisões - domínio de colisão
- Pode fornecer funcionalidades de gestão de rede



# Concentradores - Hubs

- ❑ Cada LAN que se liga ao concentrador é um **segmento** de LAN
- ❑ Os concentradores **não isolam** domínios de colisão
  - Um nó pode colidir com qualquer nó que resida em qualquer segmento de LAN
- ❑ Vantagens dos concentradores:
  - Dispositivo simples e de baixo custo
  - Hierarquia multi nível permite uma degradação suave: se um concentrador deixar de funcionar, apenas uma porção da LAN deixa de funcionar
  - Aumenta a distância entre pares de nós (100 metros por concentrador)

# Limitação dos concentradores

- ❑ Um único domínio de colisão não permite aumentar o débito máximo
  - Débito da hierarquia multi nível e de um só nível é o mesmo
- ❑ Restrição das LANs individuais impõe limitações no nº máximo de nós no mesmo domínio de colisões e na área geográfica que é possível cobrir
- ❑ Não pode interligar diferentes tipos de Ethernet (e.g. 10BaseT e 100BaseT)

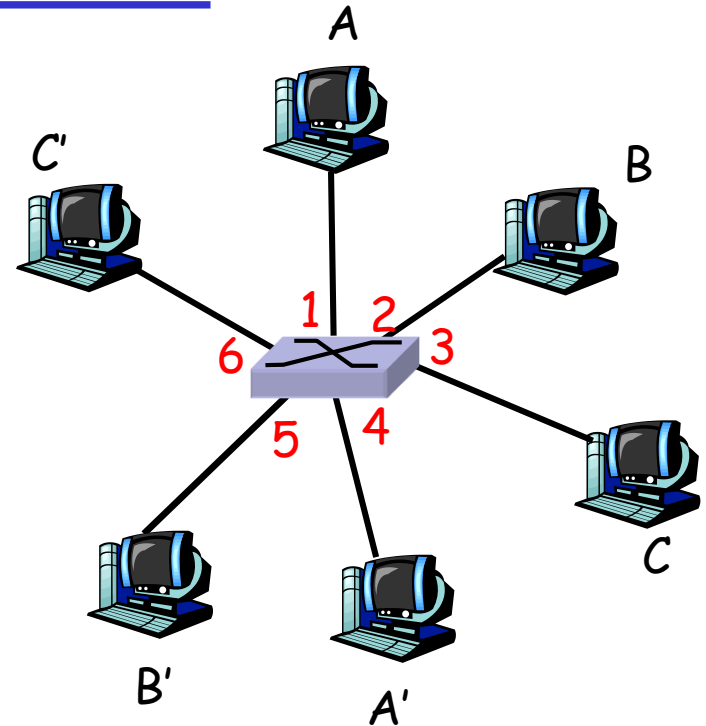


# Pontes (Bridges)

- ❑ Dispositivo do nível de ligação de dados: mais “espertos” que os hubs, tomam parte activa
  - Armazenam e enviam tramas Ethernet
  - examinam os cabeçalhos das tramas, encaminham, *selectivamente* as tramas, com base no endereço MAC de destino
  - Quando uma trama vai ser enviada num segmento, usa CSMA/CD para aceder ao segmento e transmitir
- ❑ *transparente*
  - Os sistemas terminais não se apercebem da presença das pontes
- ❑ *plug-and-play, auto-aprendizagem*
  - As pontes não necessitam configuração

# Switch: permitem múltiplas transmissões em simultâneo

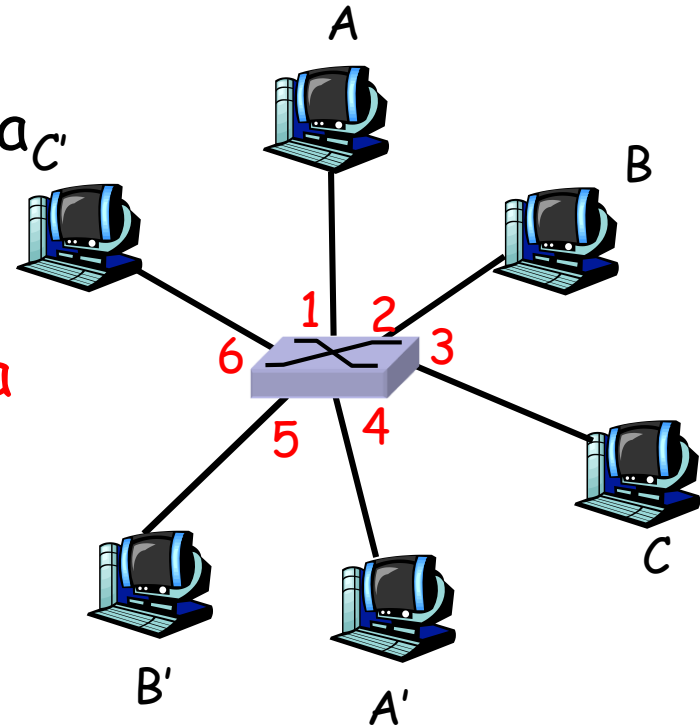
- ❑ Nós têm ligação directa e dedicada ao switch
- ❑ switches armazenam pacotes
- ❑ protocolo Ethernet usado em cada ligação de entrada, mas sem colisões; full duplex
  - Cada ligação é o seu próprio domínio de colisão
- ❑ **switching:** A-para-A' e B-para-B' em simultâneo, sem colisões
  - Não é possível com um hub



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

# Tabela de comutação

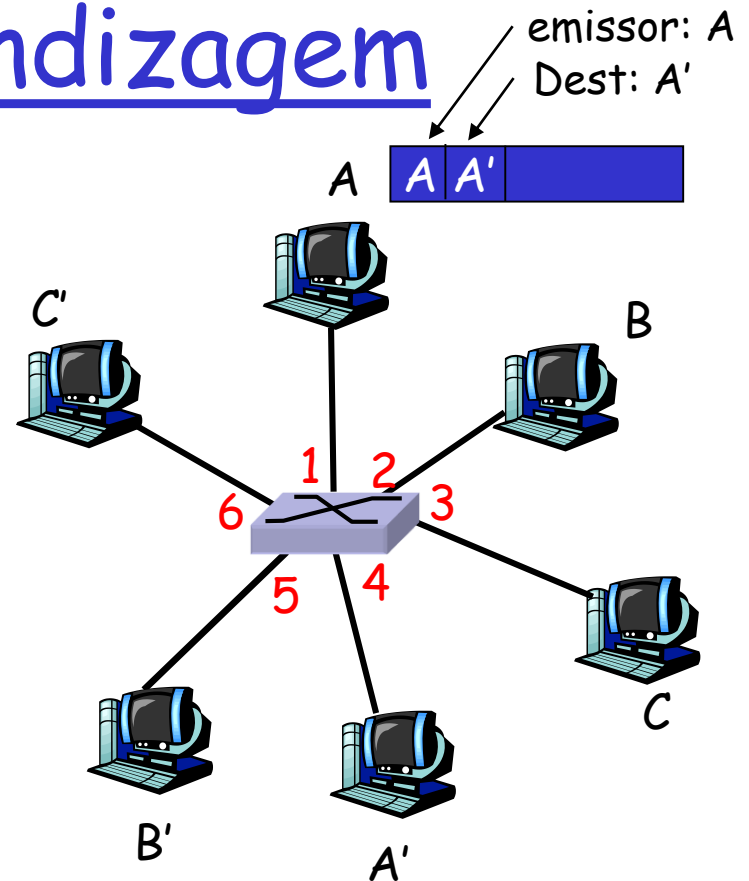
- ❑ Q: como é que um switch sabe que A' está acessível através da porta 4 e B' acessível pela porta 5?
- ❑ A: cada switch tem uma **tabela de comutação**, com os valores:
  - (MAC address do nó, interface para o nó, time stamp)
- ❑ Parece uma tabela de encaminhamento!
- ❑ Q: como é que as entradas são criadas na tabela?



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

# Switch: aprendizagem

- switch *aprende* quais os nós que podem ser acedidos através das interfaces
  - Quando recebe uma trama, o switch aprende a localização do emissor: segmento de entrada
  - Grava o par emissor/porta na tabela de comutação



MAC addr	interface	TTL
A	1	60

*Tabela de comutação  
(inicialmente vazia)*

# Switch: filtragem/encaminhamento de tramas

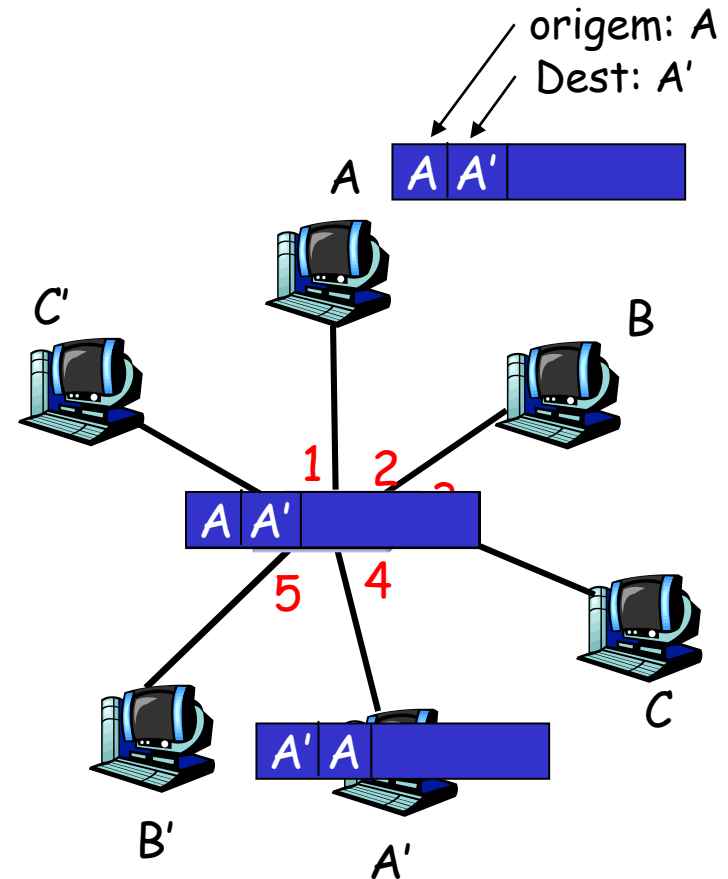
## Quando recebe uma trama:

1. Grava ligação associada com nó emissor
  2. Procura MAC destino na tabela de comutação
  3. **If** encontra MAC destino na tabela  
    **then** {  
        **if** MAC dest no segmento por onde a trama chegou  
        **then** descarta a trama  
        **else** envia a trama para a interface indicada  
    }  
    **else** inunda
- Encaminha para todas as interfaces, excepto para a qual em que chegou*

# Aprendizagem, encaminhamento: exemplo

- ❑ Destino de trama desconhecido: *inunda*
- ❑ destino A conhecido:

*Envio selectivo*

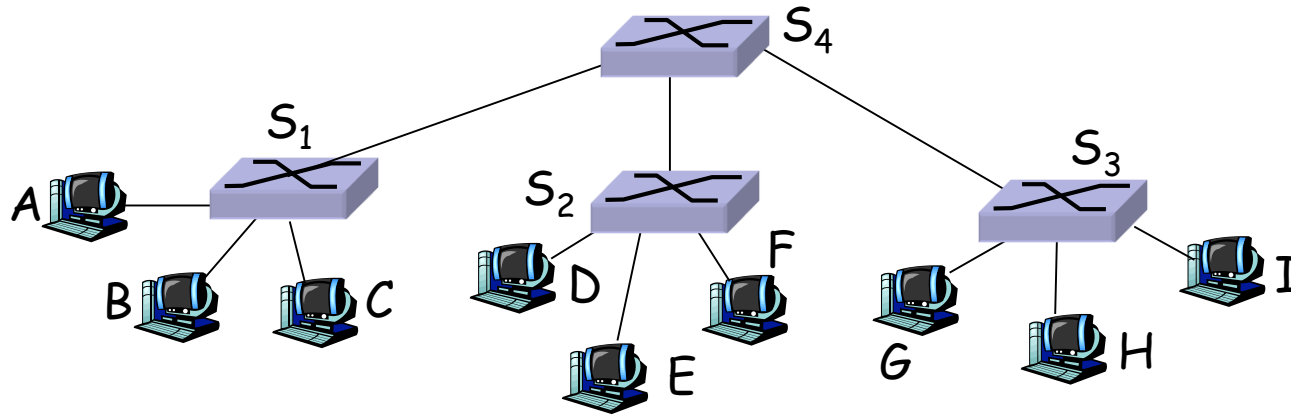


MAC addr	interface	TTL
A	1	60
A'	4	60

*Tabela encaminhamento  
(inicialmente vazia)*

# Interligando switches

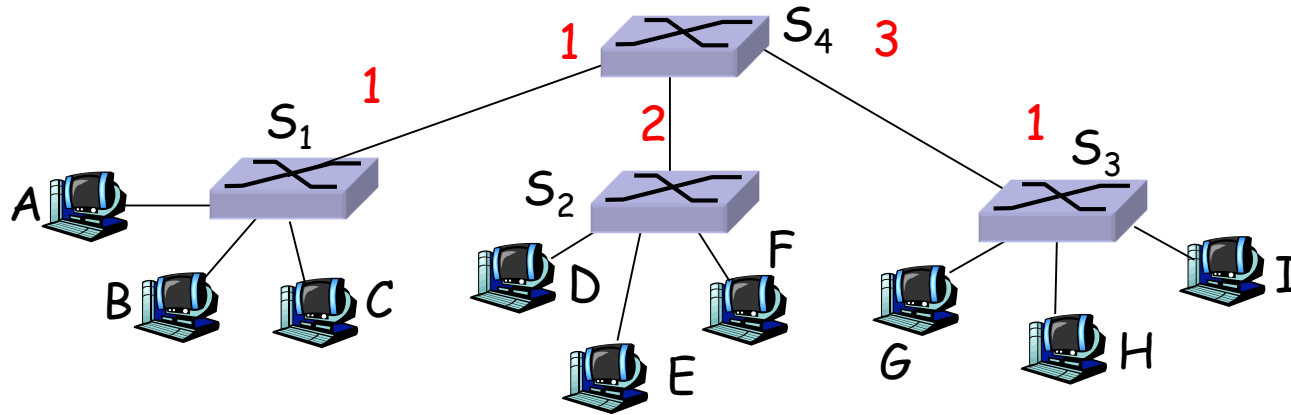
- switches podem ser interligados



- Q: enviando de A para G, como é que S1 sabe que o caminho passa por S4 e S3?
- A: auto aprendizagem! (funciona da mesma forma como para o caso de um único switch!)

# Exemplo de aprendizagem com multi-switch

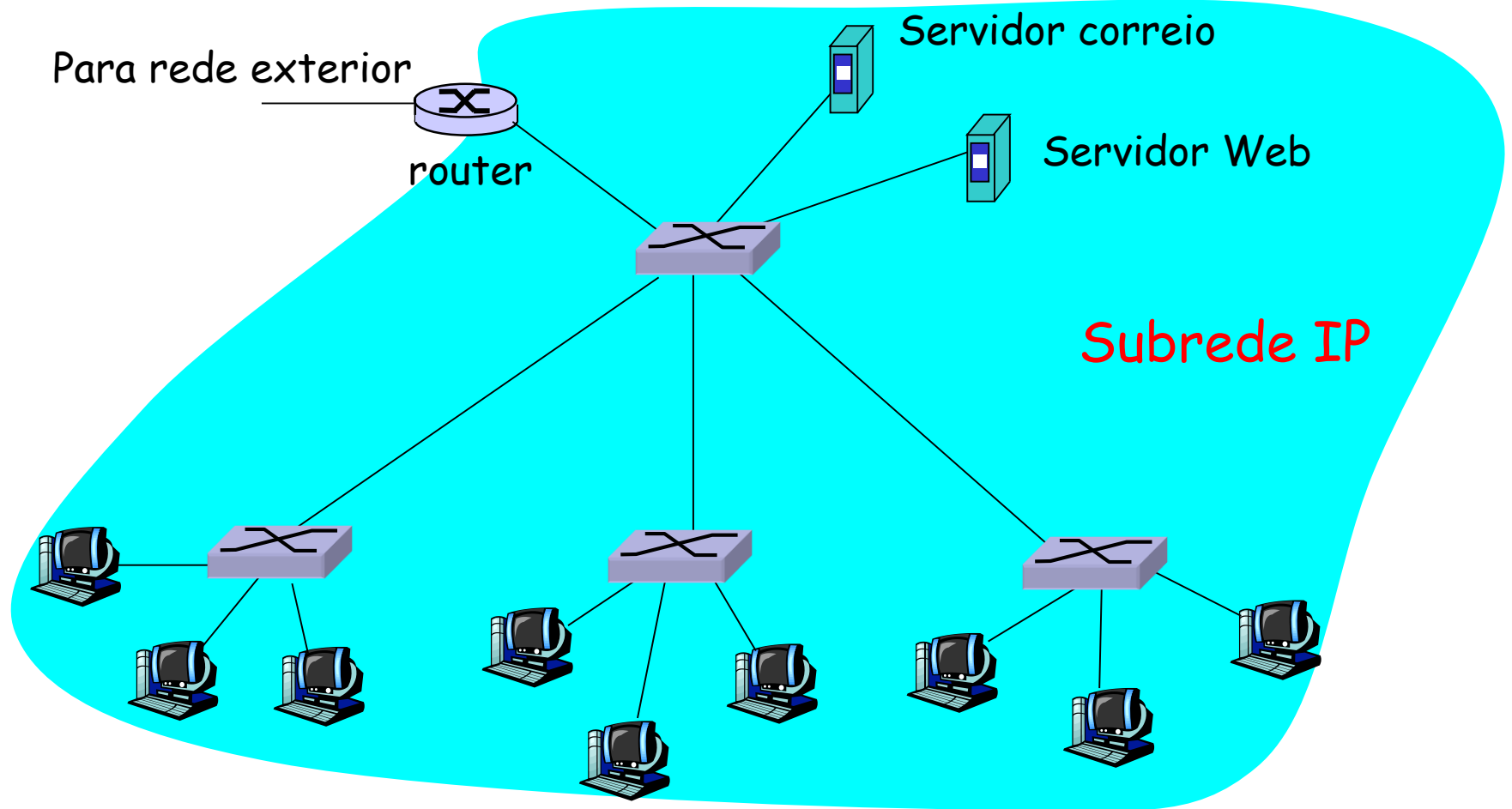
Supondo que C envia msg para I e I responde a C



□ Q: mostre as tabelas de comutação e encaminhamento em S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, S<sub>4</sub>

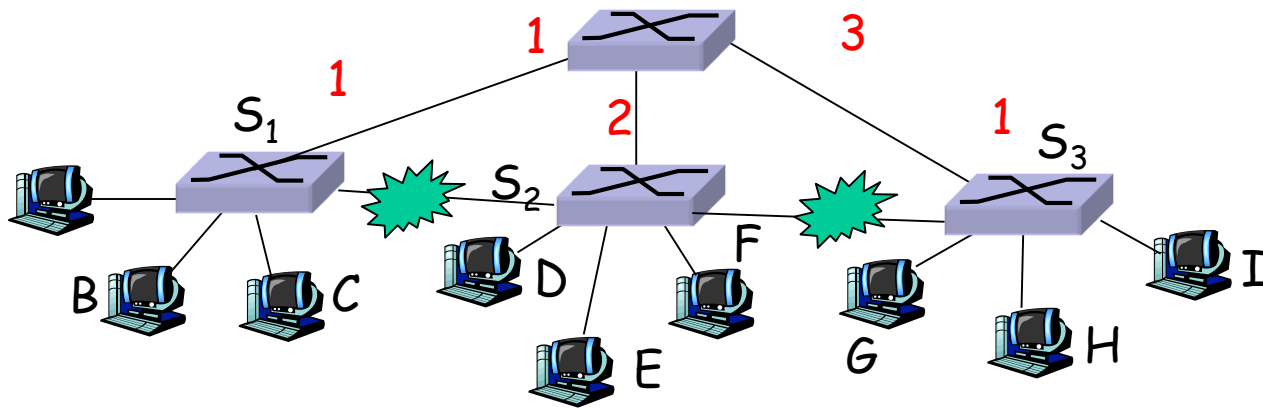


# Rede institucional



# Árvore geradora de uma ponte (spanning tree)

- ❑ Para melhor fiabilidade, é vantajoso ter caminhos redundantes (alternativos) da origem para o destino
- ❑ Com múltiplos caminhos alternativos, podem ocorrer círculos!!!
  - Os switches podem multiplicar tramas e encaminhá-las eternamente!!
- ❑ Solução: organizar os switches numa árvore, desactivando ligações que fazem círculos - Protocolo de Spanning Tree

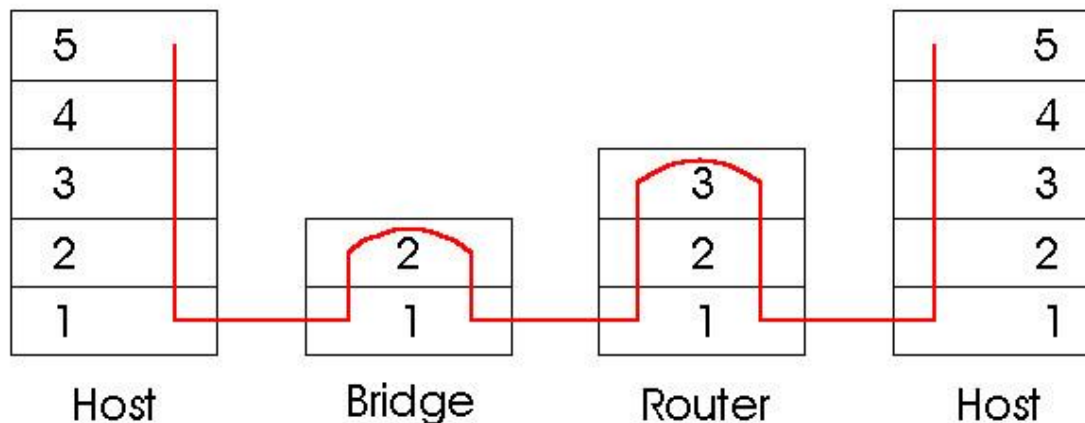


# Algumas funcionalidades dos Switches

- ❑ Os switches isolam domínios de colisão, resultando num débito total superior
- ❑ Não impõe limitações ao número de nós, nem à área geográfica coberta
- ❑ Pode ligar diferentes tipos de Ethernet, uma vez que é um dispositivo que armazena e envia, se necessário, já convertido ao novo tipo
- ❑ Transparente ("plug and play"): não é necessário configuração

# Switches vs. Routers

- ❑ Ambos são dispositivos que armazenam e enviam pacotes
  - routers: dispositivo de nível de rede (examina cabeçalho de nível de rede)
  - switches são do nível de ligação de dados
- ❑ routers mantêm tabelas de encaminhamento, realizam algoritmos de encaminhamento
- ❑ switches mantêm tabelas de comutação, realizam algoritmos de filtragem e de aprendizagem



# Routers vs. switches

## Switches

- (+) A operação de um switch é mais simples e requer menos processamento de pacotes
- (+) As tabelas de um switch são auto aprendidas
- (-) Todo o tráfego é restringido à árvore geradora (spanning tree) apesar de haver ligações alternativas (desactivadas)
- (-) Os switches não oferecem protecção contra "broadcast storm" (tempestades difusão de pacotes). Os switches encaminham todos os broadcasts

# Routers vs. switches

## Routers

- (+) Pode suportar qualquer tipo de topologia, os círculos - os círculos são limitados pelo contador de TTL
- (+) Fornece protecção de firewall contra broadcast storms
- (-) Requer configuração de endereços IP (não é "plug and play")
- (-) Requer mais processamento por pacote
- ❑ Os switches são adequados para redes pequenas (poucas centenas de nós) enquanto os routers são utilizados em redes grandes (milhares de nós)