

Camada de Enlace

Objetivos:

- ❖ Entender os serviços oferecidos:
 - *Detecção e correção de erros (erros nos bits: ruídos, atenuação do sinal)*
 - *Entrega confiável (enlace a enlace)*
 - *Compartilhamento de canal: **broadcast, múltiplo acesso***
 - *Endereçamento / delimitação de quadros*
- ❖ Implementação de várias tecnologias na camada de enlace
- ❖ Redes locais Ethernet, VLANs

Camada de enlace

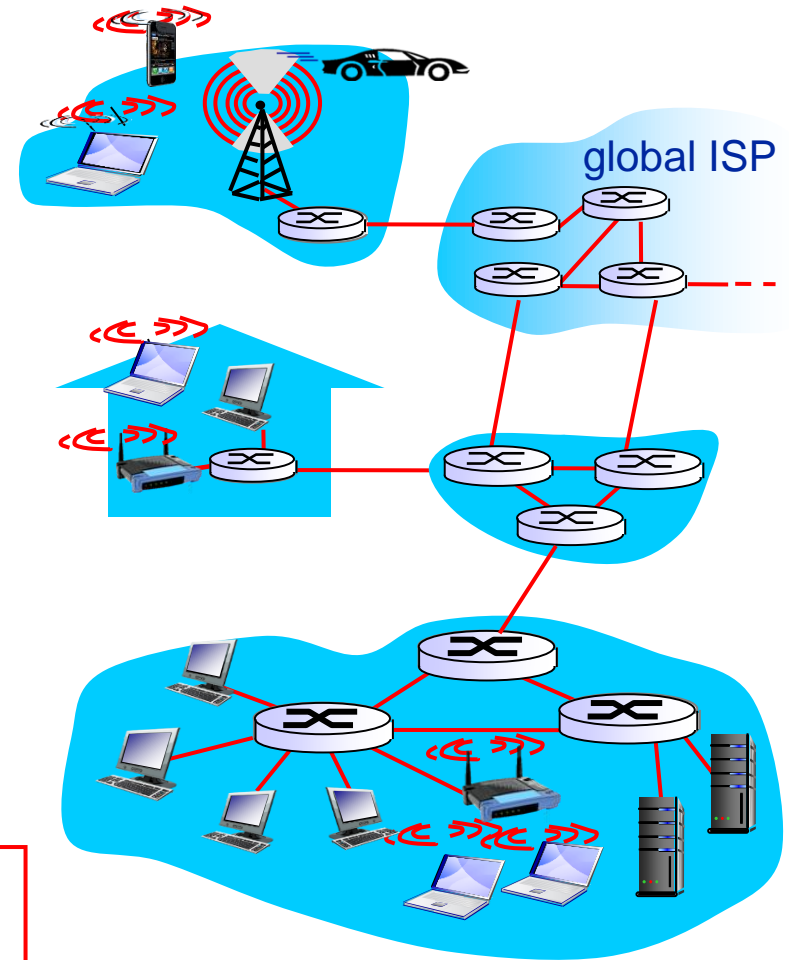
- **introdução, serviços**
- detecção e correção de erros
- protocolos de múltiplo acesso
- LANs
 - *endereçamento, ARP*
 - *Ethernet*
 - *comutadores*
 - *VLANS*
- redes de datacenter
- um dia na vida de uma requisição web

Introdução

Terminologia:

- ❖ **nós:** hosts e roteadores
- ❖ **enlaces:** canais de comunicação que conectam nós adjacentes ao longo do caminho
 - Enlaces com fio
 - Enlaces sem fio
 - LANs
- ❖ **quadro:** unidade de dados da camada de enlace que encapsula um datagrama

Camada de enlace: tem a responsabilidade de transferir um datagrama de um nó ao nó fisicamente *adjacente* sobre um enlace



Contexto

- ❖ Datagramas podem ser transferidos por diferentes protocolos sobre enlaces diferentes:
 - *Ex: Ethernet no 1º enlace, fibra ótica nos enlaces intermediários, IEEE 802.11 (Wi-Fi) no último enlace*
- ❖ Cada protocolo de enlace provê serviços diferentes
 - *Ex: pode ou não prover transferência confiável de dados. Ex: fibra (não provê)*

Analogia com transporte humano:

- ❖ Viagem de Resende p/ Lyon
 - *ônibus: Resende p/ GIG*
 - *avião: GIG p/ Paris*
 - *tgV: Paris p/ Lyon*
- ❖ Turista = **datagrama**
- ❖ Cada trecho = **enlace de comunicação**
- ❖ Tipo de transporte = **protocolo de enlace**
- ❖ Agente de viagem = **algoritmo de roteamento**

Serviços da Camada de Enlace

- ***Enquadramento:***
 - Enquadramento (delimitação) dos dados transmitidos
 - ✓ Camada Física: **Bits**
 - ✓ Camada de Enlace: **Quadros**
 - ✓ Camada de Rede: **Datagramas**
 - ✓ Camada de Transporte: **Segmentos**
 - ✓ Camada de Aplicação: **Mensagens**
 - encapsula datagrama num quadro incluindo cabeçalho
 - implementa acesso ao canal se meio for compartilhado
 - ***endereços MAC*** são usados em cabeçalhos de quadros para identificar origem e destino de quadros ***em enlaces multiponto diferente do endereço IP!!!***

Serviços da Camada de Enlace

- ***Entrega confiável entre nós adjacentes:***
 - Pouco usada em fibra óptica, cabo coaxial e alguns tipos de pares trançados devido a taxas de erro de bit muito baixas.
 - Usada em enlaces de rádio, onde a meta é reduzir erros evitando assim a retransmissão fim a fim.
 - Semelhante à camada de transporte (*fim-a-fim* versus *nó-a-nó*, em um único enlace).
- ***Controle de Fluxo:***
 - compatibilizar taxas de produção e consumo de quadros entre transmissores e receptores (nós adjacentes, semelhante à camada de transporte)

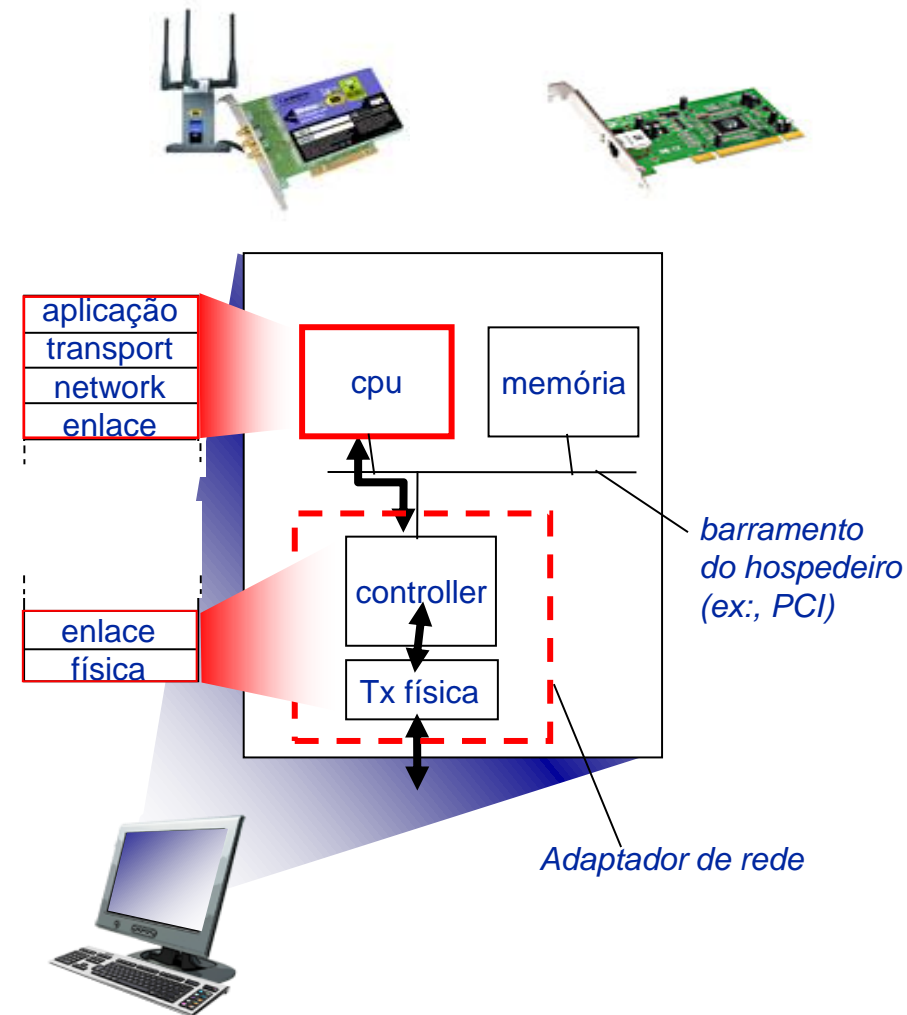
Serviços da Camada de Enlace

- *Detecção de Erros:*
 - erros são causados por atenuação do sinal e por ruído
 - receptor detecta presença de erros
 - receptor sinaliza ao remetente para retransmissão ou simplesmente descarta o quadro com erro
- *Correção de Erros:*
 - mecanismo que permite que o receptor localize e corrija o erro (de bit) sem precisar da retransmissão
- *Transmissão half-duplex e full-duplex*

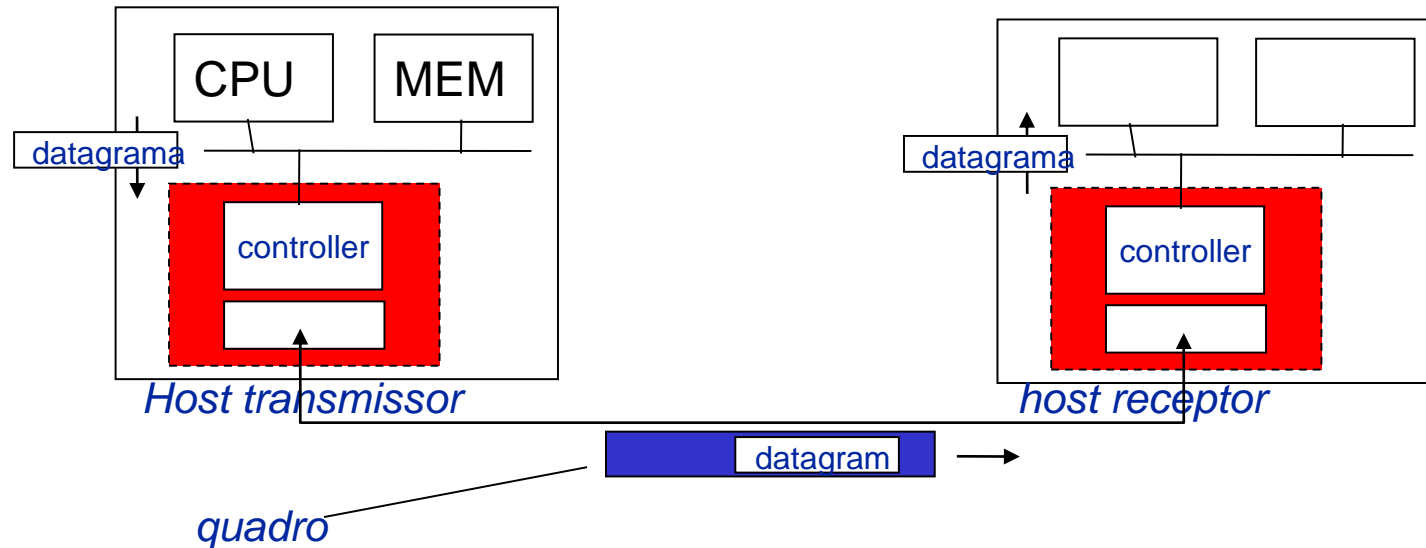
Obs: Com half duplex, nós em ambos os lados do enlace podem transmitir, mas não ao mesmo tempo

Onde é implementada a camada de enlace

- ❖ Em todos os hospedeiros
- ❖ Implementada na placa de rede (*network interface card* NIC) ou em um chip
 - Placa Ethernet, placa 802.11, Ethernet chipset
 - Placa implementa tanto a camada de enlace qto a física
- ❖ Conectada ao barramento do Sistema do computador
- ❖ Combinação de hardware e software



Comunicação entre as placas de rede



❖ Lado transmissor:

- *encapsula o datagrama no quadro*
- *adiciona bits de verificação de erros, rdt, controle de fluxo, etc.*

❖ Lado receptor

- *procura por erros, rdt, controle de fluxo, etc*
- *extrai o datagrama e o passa para a camada superior*

Comunicação em um enlace

Exemplo de procedimentos envolvidos na comunicação:

1. Quadro é recebido em um roteador em um enlace
2. *Hardware* verifica *checksum* (detecção de erros) e repassa o quadro para o software da camada de enlace
3. Software verifica se o quadro recebido é realmente o quadro esperado
Ex.: verifica se o quadro está na ordem correta
4. Em caso afirmativo, camada de enlace extrai o datagrama de dentro do quadro e o entrega ao *software* da camada de rede para roteamento
5. *Software* de roteamento escolhe o enlace de saída apropriado e repassa o datagrama para o *software* de camada de enlace responsável por aquele enlace

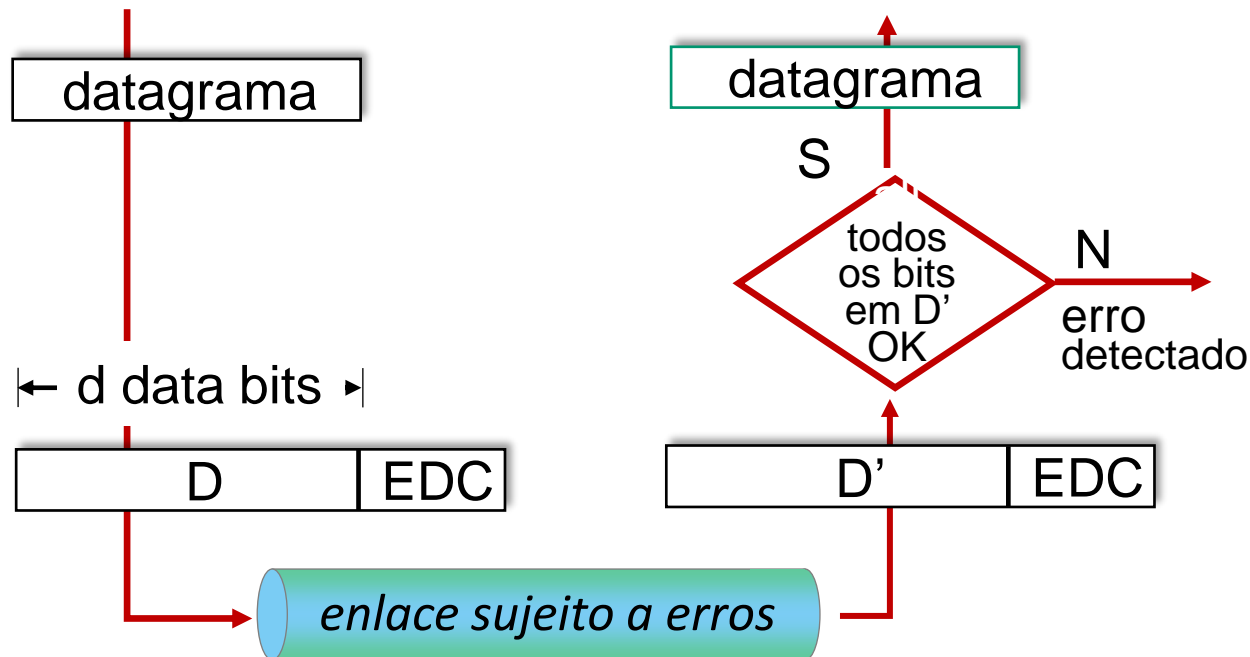
Detecção de erros

EDC = bits de Detecção e Correção de Erros (redundância)

D = Dados protegidos por verificação de erros;
podem incluir alguns campos do cabeçalho

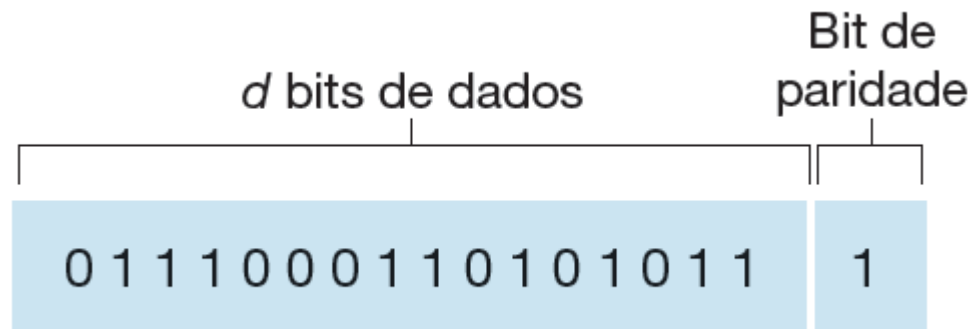
Detecção de erros não é 100% perfeita:

- ✓ protocolo pode não identificar alguns erros, mas é raro
- ✓ qto maior o campo de EDC maior a probabilidade de detecção e correção



Técnicas de detecção e correção de erros

- O desafio do receptor é determinar se D' é ou não igual ao D original, uma vez que recebeu apenas D' e EDC' .



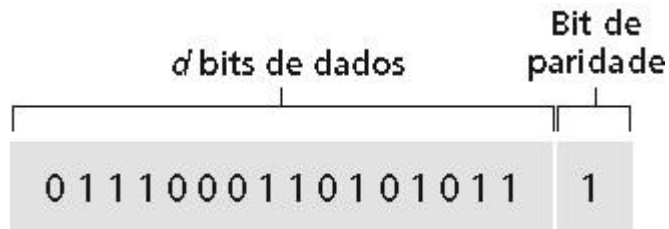
Paridade par:

define o bit de paridade para que haja um número par de 1s

Verificações de Paridade

Paridade de 1 Bit:

Detecta erros em um único bit

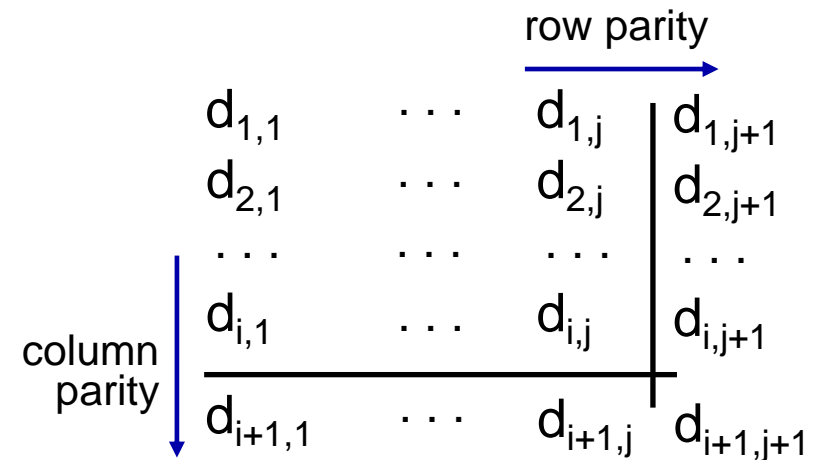


sem erros:

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

Paridade Bidimensional:

Detecta e corrige erro em um único bit



detectados
e corrigíveis
erros de bit
único

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

parity
error

Checksum da Internet (revisão)

Objetivo: detectar "erros" (ex., bits trocados) no segmento transmitido

Transmissor:

- Considera dados como compostos de inteiros de 16 bits; soma todos os campos de 16 bits.
- O complemento de 1 desta soma é calculado
- Coloca o *checksum* no campo do segmento da camada de transporte

Receptor:

- Calcula o checksum do segmento recebido e verifica se o resultado é igual ao do campo de checksum. Não, erro detectado. Caso contrário, segmento está OK.

Enlaces e protocolos de múltiplo acesso

Redes de Difusão (*Broadcast*) vs Redes Ponto-a-Ponto

➤ Redes Ponto-a-Ponto:

- ✓ Meio de transmissão dedicado
- ✓ Sem disputa pelo acesso ao meio
- ✓ Sem necessidade de endereçamento
- ✓ Apenas um destino possível para cada transmissão

➤ Redes de Difusão

- ✓ Meio de transmissão compartilhado
- ✓ Necessidade de disciplinar o acesso ao meio
- ✓ Cada estação possui um endereço único
- ✓ A mesma transmissão pode ser recebida por várias estações

Enlaces e protocolos de múltiplo acesso

Dois tipos de enlaces:

- **ponto-a-ponto:**
 - Enlace ponto-a-ponto entre um switch Ethernet e um host
 - PPP para acesso discado
- **difusão (*broadcast* – meio compartilhado com fio ou sem fio)**
 - Ethernet mais antiga (hubs)
 - Enlace HFC (Hybrid Fiber Coax) em rede de acesso baseada em cabo
 - 802.11 (Wi-fi) LAN, 4G/5G. satélite



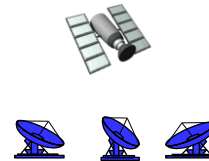
Compartilhado com fio
(Ethernet antiga)



Compartilhado sem fio: 4G/5G



Compartilhado sem fio: WiFi



satélite



Analogia: festa

Protocolos de múltiplo acesso

Considerações sobre o problema de alocação do canal

➤ Colisões

- ✓ canal único de *broadcast* compartilhado
- ✓ dois (ou mais) quadros transmitidos simultaneamente se sobrepõem, gerando lixo
 - quadros são perdidos
 - mesmo que apenas um bit seja afetado!
- ✓ assume-se que colisões são a única fonte de erros
- ✓ todas as estações podem detectar a ocorrência de colisões
 - inclusive as estações transmissoras (*broadcast*)
- ✓ alguns protocolos evitam a ocorrência de colisões

Protocolos de múltiplo acesso

Considerações sobre o problema de alocação do canal (2)

➤ **Uso do tempo:**

✓ *Tempo contínuo*

- transmissão de quadros pode ter início em qualquer instante

✓ *Tempo demarcado (slotted time)*

- um relógio mestre divide o tempo em intervalos discretos (*slots*)
- transmissão de quadros só pode começar no início de um *slot*

Protocolos de múltiplo acesso

Considerações sobre o problema de alocação do canal (3)

➤ **Detecção do estado do canal:**

✓ *Protocolos com monitoração do canal (carrier sensing)*

- estações verificam se o canal está sendo usado **antes** de tentarem transmitir \Rightarrow reduz a probabilidade de colisões
- método geralmente usado em redes locais

✓ *Protocolos sem monitoração do canal*

- quando a monitoração prévia do estado do canal não é prática
 - ✓ **ex.: em redes de satélite (longos atrasos)**
- estações transmitem sem verificar se o canal está livre
- detectam a colisão e retransmitem

Protocolos de múltiplo acesso

- ❖ Canal único para compartilhamento *broadcast*
- ❖ Duas ou mais transmissões simultâneas pelos nós: interferência
 - *Colisão: se nó recebe dois ou mais sinais ao mesmo tempo*

Protocolo de múltiplo acesso

- ❖ Algoritmo distribuído determina como os nós compartilham o canal, isto é, determina quando o nó pode transmitir
- ❖ Comunicação sobre o compartilhamento do canal deve usar o próprio canal!
 - *Não existe um canal específico (“fora da banda”) para coordenar a transmissão*

Um protocolo de múltiplo acesso ideal

Dado um canal *broadcast* com taxa de R bps, deseja-se:

1. *quando um nó quer transmitir, ele pode enviar à taxa R .*
2. *quando M nós querem transmitir, cada um pode enviar a uma taxa média de R/M*
3. *totalmente descentralizado:*
 - *Não existe um nó especial para coordenar as transmissões*
4. *simples*

Protocolos MAC

Três grandes classes:

❖ *Particionamento do canal*

- *divide canal em “pedaços” menores (slots de tempo, frequência, código)*
- *aloca um “pedaço” para um nó, para uso exclusive*
- *não existe colisões*

❖ *Acesso aleatório*

- *Canal permite colisões*
- *“Recupera-se” de colisões*

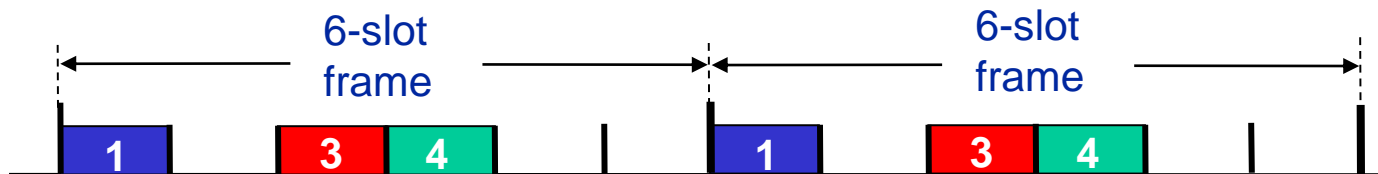
❖ *Revezamento*

- *Nós se revezam nas transmissões, mas aqueles com mais dados a enviar podem tomar um tempo maior do canal*

Protocolos MAC de particionamento do canal: TDMA

TDMA: múltiplo acesso por divisão de tempo

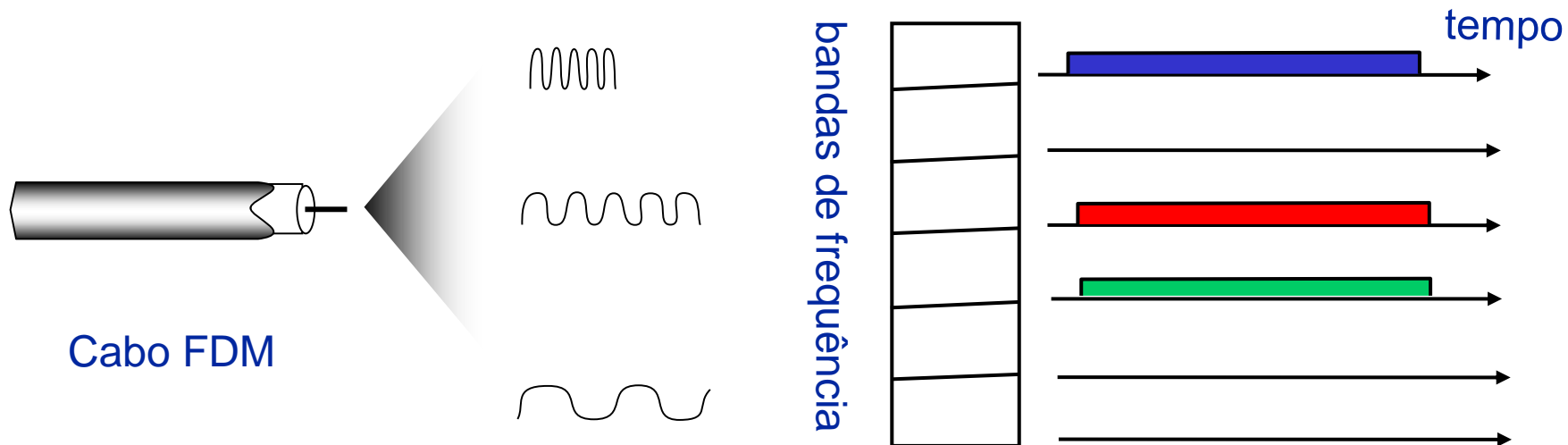
- ❖ Cada um acessa o canal na sua “vez”
- ❖ Cada estação ganha um slot de comprimento fixo (comprimento = tempo de transmissão do quadro) em cada vez
- ❖ Slots não usados ficam ociosos
- ❖ exemplo: 6 estações numa rede local (LAN): 1,3,4 têm quadros para transmitir, slots 2,5,6 ociosos



Protocolos MAC de particionamento do canal: FDMA

FDMA: múltiplo acesso por divisão de frequência

- ❖ Espectro do canal é dividido em bandas de frequência
- ❖ Para cada estação é atribuída uma banda de frequência fixa
- ❖ Bandas de frequência não usadas ficam ociosas
- ❖ exemplo: 6-estações LAN, 1,3,4 têm pkts, bandas de frequência 2,5,6 ociosas



Protocolos de acesso aleatório

- ❖ Quando nó tem pacotes a enviar
 - Transmite à taxa de R bps.
 - Não existe uma coordenação *a priori* entre os nós
- ❖ Dois ou mais nós transmitindo → “colisão”
- ❖ Protocolo MAC de acesso aleatório especifica:
 - Como detectar colisões
 - Como se recuperar de colisões (ex., via atraso nas retransmissões)
- ❖ Exemplos de protocolos MAC de acesso aleatório:
 - ALOHA
 - Slotted ALOHA
 - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

- Tempo é dividido em intervalos discretos do mesmo tamanho
 - ✓ Equivalentes ao tempo de transmissão de um quadro
 - **considerando quadros de mesmo tamanho**
 - ✓ Uma estação especial transmite um sinal periódico de temporização para marcar o início dos *slots*
 - ✓ Se 2 ou mais nós transmitem no mesmo *slot*, todos os nós detectam a colisão
- Estação transmissora precisa esperar o início de um novo *slot* para que possa transmitir

Slotted ALOHA

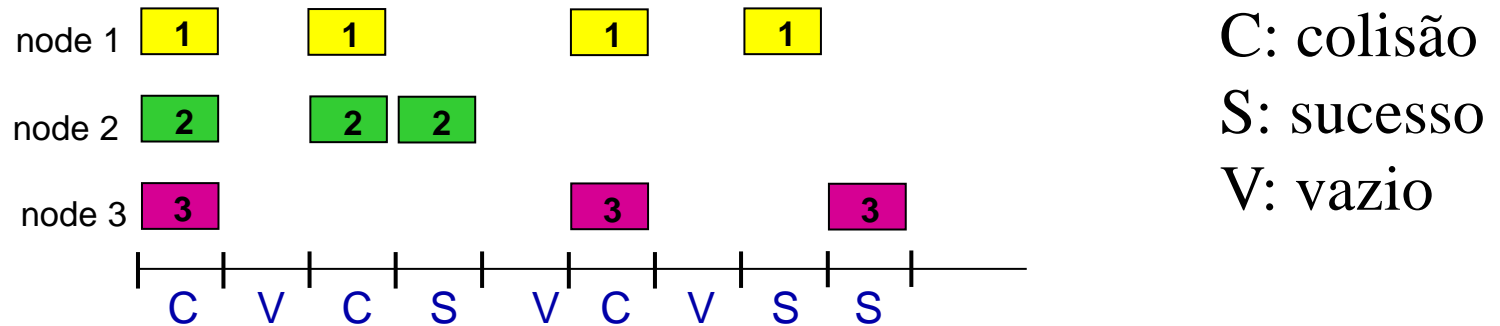
Operação:

quando um nó obtém um novo quadro, o transmite no próximo *slot*

- *Se não ocorrer colisão, nó pode enviar outro quadro no próximo slot*
- *Se ocorrer colisão, nó retransmite o quadro no slot subsequente com probabilidade p até que obtenha sucesso*

Acesso aleatório: tenta evitar novas colisões

Slotted ALOHA



Prós:

- um único nó ativo pode transmitir continuamente na taxa total do canal
- altamente descentralizado: somente os slots e os nós devem ser sincronizados
- simples

Contras:

- colisões, desperdiçando slots
- pode-se ter *slots* vazios
- exigência de sincronização

Eficiência do *Slotted Aloha*

Eficiência:

é a fração de *slots* onde a transmissão foi bem sucedida quando há muitos nós, cada um com muitos quadros para transmitir

- Assuma N nós com muitos quadros para enviar, cada um transmite num *slot* com probabilidade p
- probabilidade que um nó tenha sucesso em um *slot* = $p(1-p)^{N-1}$
- probabilidade que qualquer nó tenha sucesso = $Np(1-p)^{N-1}$

- Para eficiência máxima com N nós, encontre p^* que maximiza $Np(1-p)^{N-1}$
- Para muitos nós, faça limite para $Np^*(1-p^*)^{N-1}$ quando N tende a infinito, dá

eficiência máxima = $1/e = 0,37$

Melhor caso:

canal usado para transmissões úteis em apenas 37% do tempo!



ALOHA Puro

- Estações transmitem quadros sempre que houver dados a serem enviados sem monitorar o estado do meio
- Não requer sincronização
- Colisões são frequentes
 - ✓ quadros são perdidos mesmo que apenas o primeiro ou o último bit tenha colidido, aumentando a probabilidade de colisão
- Transmissor detecta a colisão e retransmite o quadro após um intervalo aleatório de tempo, para tentar evitar novas colisões

ALOHA

Estação

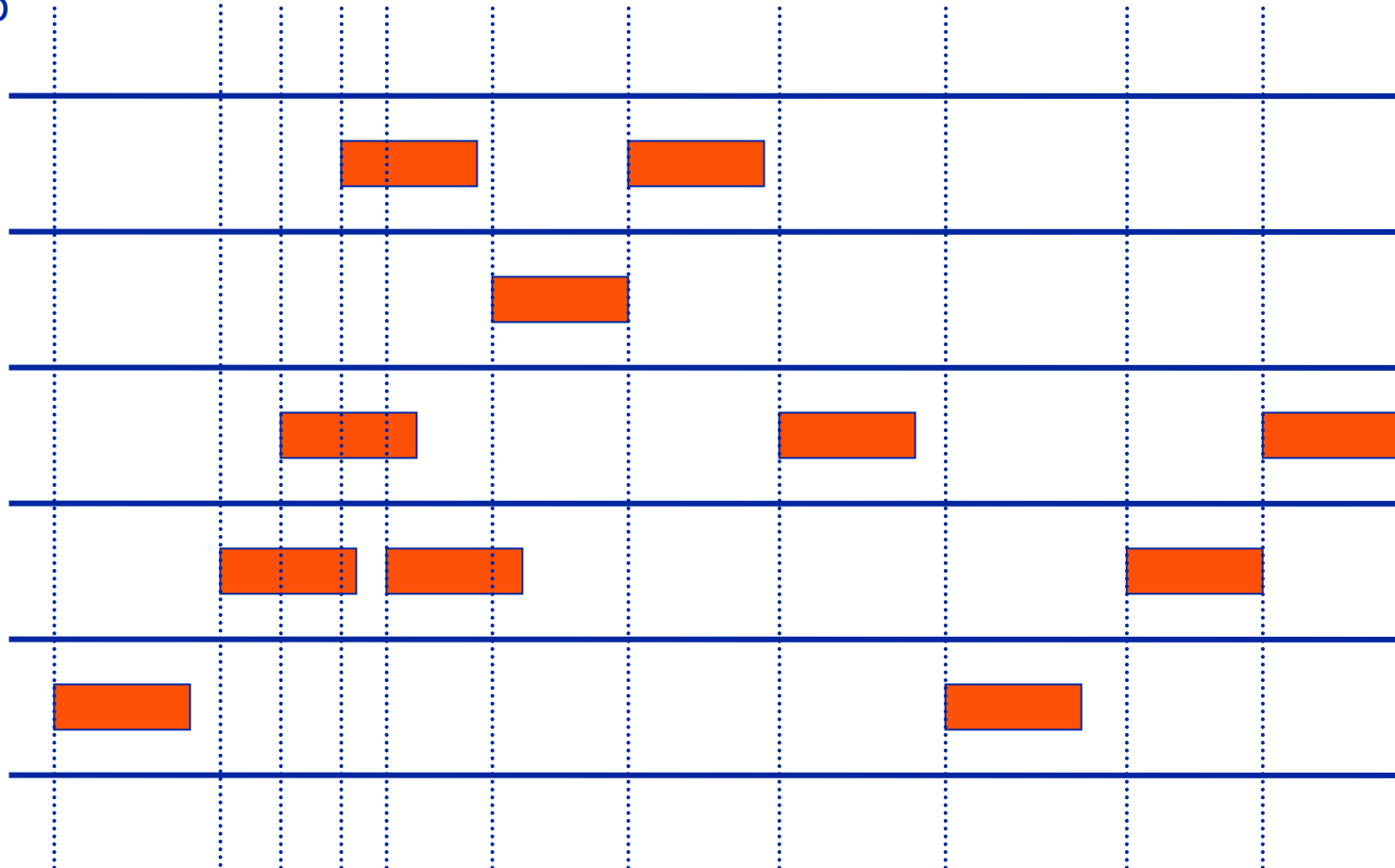
A

B

C

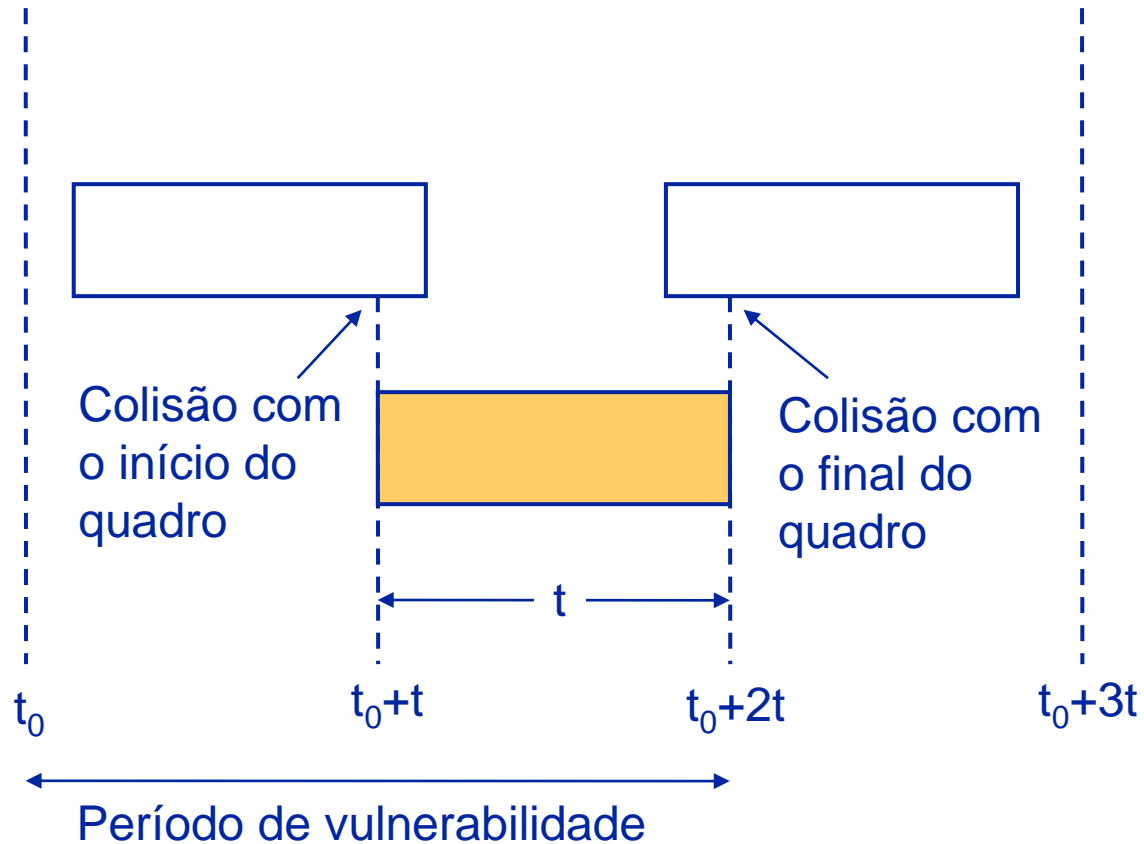
D

E



ALOHA

Período de disputa do meio:



ALOHA

Eficiência

- Decresce com o aumento do número de estações tentando transmitir ao mesmo tempo
 - ✓ Gera mais colisões...
 - ✓ ... que por sua vez geram mais retransmissões
 - ✓ ... que geram mais colisões
 - ✓ Efeito em cascata
- *Eficiência máxima do protocolo: 18%*

Eficiência do Aloha

$P(\text{sucesso para um dado nó}) = P(\text{nó transmite c/ sucesso}) \cdot$

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

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

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

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

... escolhendo o valor ótimo de p e deixando $n \rightarrow \infty$

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

CSMA (Acesso múltiplo com detecção de portadora)

CSMA (*Carrier Sense Multiple Access*)

Escuta antes de transmitir (detecção de portadora)

Se o canal estiver livre: transmite todo o quadro

Se o canal estiver ocupado, adia a transmissão

Analogia humana: não interrompa os outros!

Colisões no CSMA

colisões ainda podem acontecer:

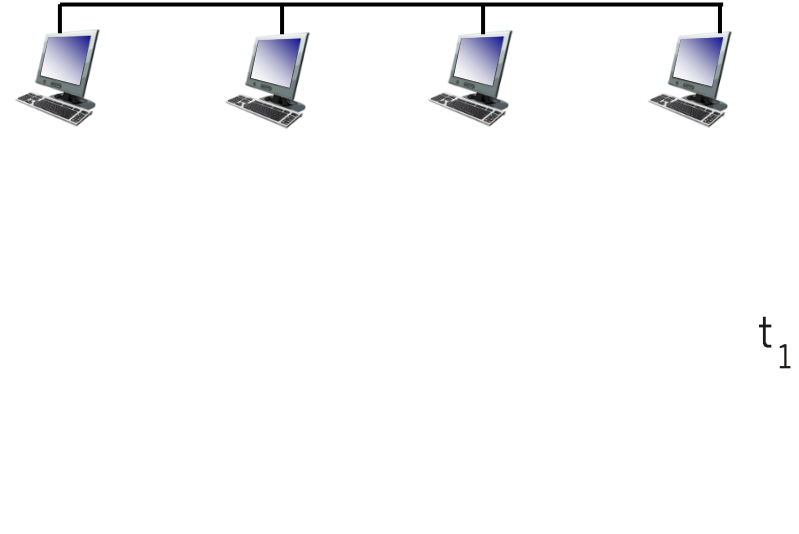
**atraso de propagação significa que
dois nós podem não ouvir
a transmissão do outro**

colisão:

**todo o tempo de transmissão
é desperdiçado**

nota:

**papel da distância e atraso de
propagação na determinação da
probabilidade de colisão**



CSMA/CD (Detecção de Colisões)

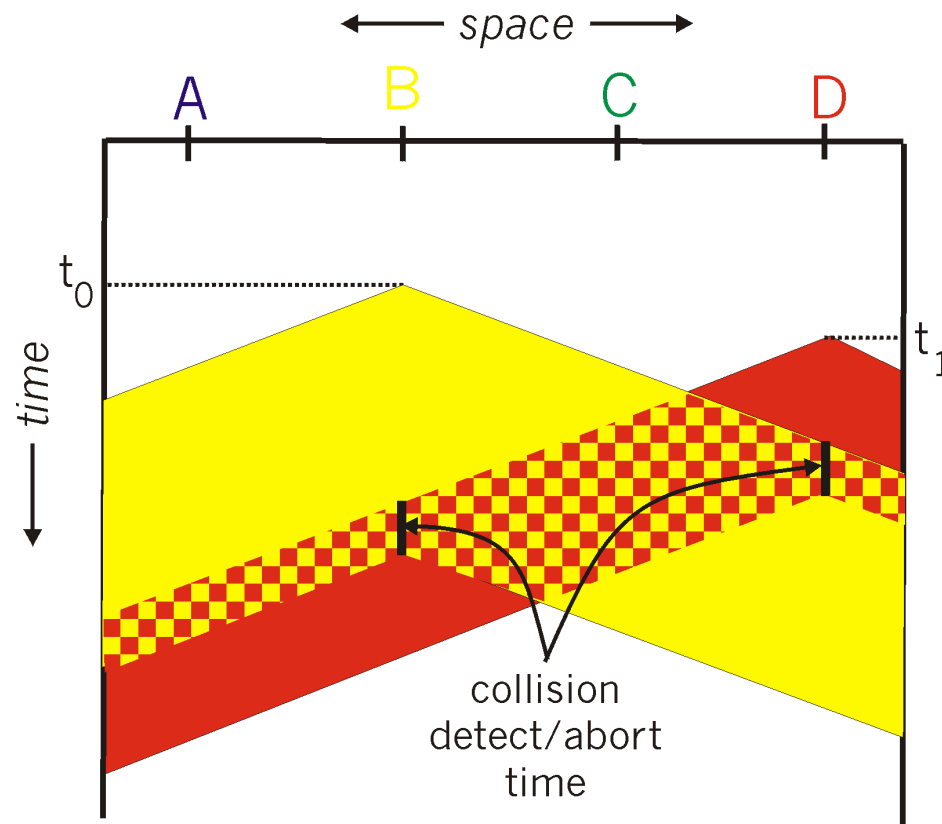
CSMA/CD: detecção da portadora adia a transmissão, como no CSMA

- As colisões são *detectadas* em pouco tempo
- Transmissões que sofreram colisões são abortadas, reduzindo o desperdício do canal

Detecção de colisões:

- Fácil em LANs cabeadas: mede a potência do sinal, compara o sinal recebido com o transmitido
- Difícil em LANs sem fio (potência menor e problema do terminal escondido)

Detecção de colisões em CSMA/CD



Algoritmo CSMA/CD do Ethernet

1. Adaptador recebe datagrama da camada de rede e cria um quadro
2. Se o adaptador percebe que o canal está ocioso, começa a transmitir o quadro. Se percebe que o canal está ocupado, espera que o canal fique livre e transmite
3. Se o adaptador transmitir todo o quadro sem detectar outra transmissão, o adaptador concluiu a operação com o quadro!
4. Se o adaptador detectar outra transmissão enquanto estiver transmitindo, aborta e envia sinal de **reforço de colisão**
5. Após abortar a transmissão, o adaptador entra em **recuo (backoff) exponencial binário**:
 - após a m -ésima colisão, o adaptador escolhe um K aleatoriamente entre $\{0, 1, 2, \dots, 2^m - 1\}$. O adaptador espera $K \cdot 512$ tempos de bit e retorna ao Passo 2
 - quanto mais colisões houver, mais longos serão os intervalos entre as retransmissões.

Eficiência do CSMA/CD

t_{prop} = atraso máximo de propagação entre 2 nós na LAN

t_{trans} = tempo para transmitir quadro de tamanho máximo

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

Eficiência vai para 1 à medida que:

- ✓ t_{prop} vai para 0
- ✓ t_{trans} vai para infinito

Muito melhor que a do ALOHA, além de ser descentralizado, simples, e barato!

Protocolos MAC de “revezamento”

Protocolos MAC de divisão de canal:

- ❖ Compartilha o canal eficientemente e de forma justa em altas cargas
- ❖ Ineficiente em baixas cargas: atraso no canal de acesso, alocação de $1/N$ da largura de banda mesmo com apenas 1 nó ativo!

Protocolos MAC de acesso aleatório:

- ❖ Eficiente em baixas cargas: um único nó pode utilizar completamente o canal
- ❖ Altas cargas: sobrecarga, com excesso de colisões

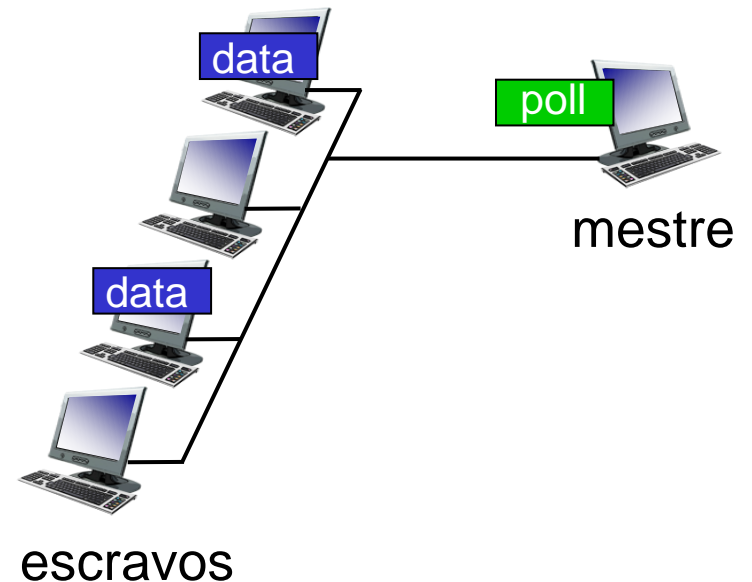
Protocolos de revezamento:

Procura oferecer o melhor dos dois mundos!

Protocolos MAC de “revezamento”

Seleção (*Polling*):

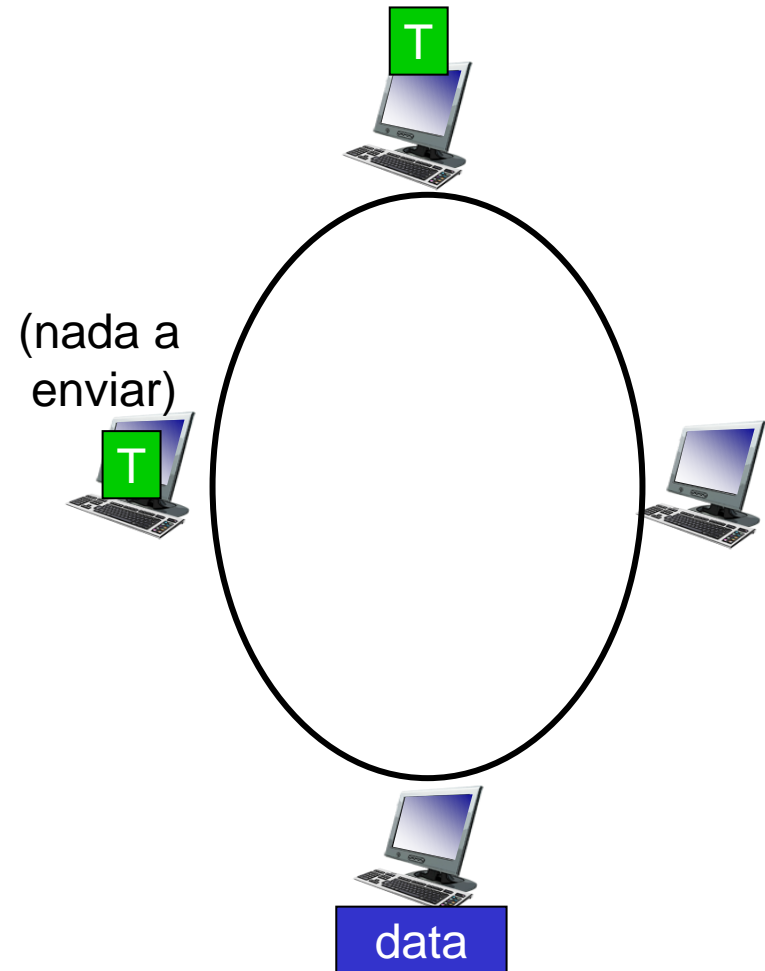
- ✓ Nó mestre “convida” nós escravos a transmitir em revezamento
- ✓ Usado tipicamente com dispositivos escravos “burros”.
- ✓ Preocupações:
 - Sobrecarga com as consultas (*polling*)
 - Latência
 - Ponto único de falha (mestre)



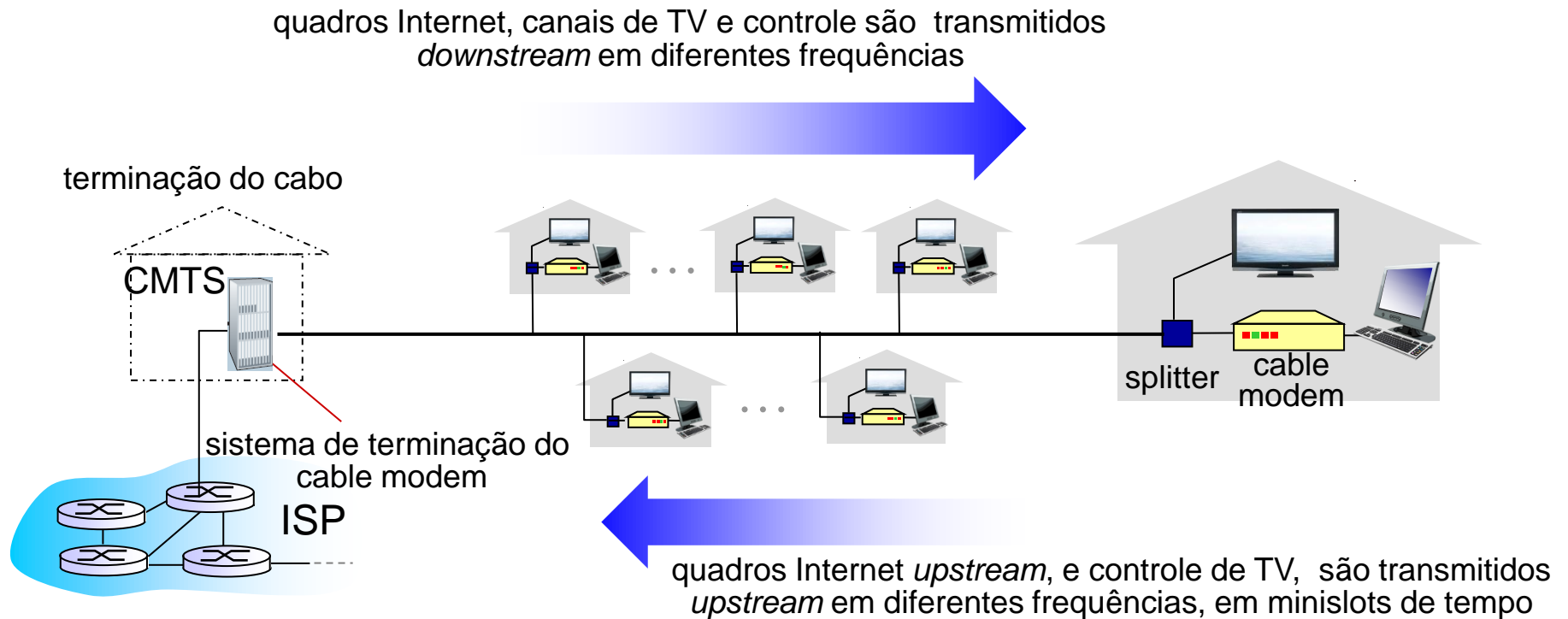
Protocolos MAC de “revezamento”

Passagem de permissão (*token*):

- ✓ controla ***permissão*** passada de um nó para o próximo de forma sequencial.
- ✓ mensagem de passagem da ***permissão***
- ✓ preocupações:
 - *overhead* com a passagem de permissão
 - latência
 - ponto único de falha (permissão)



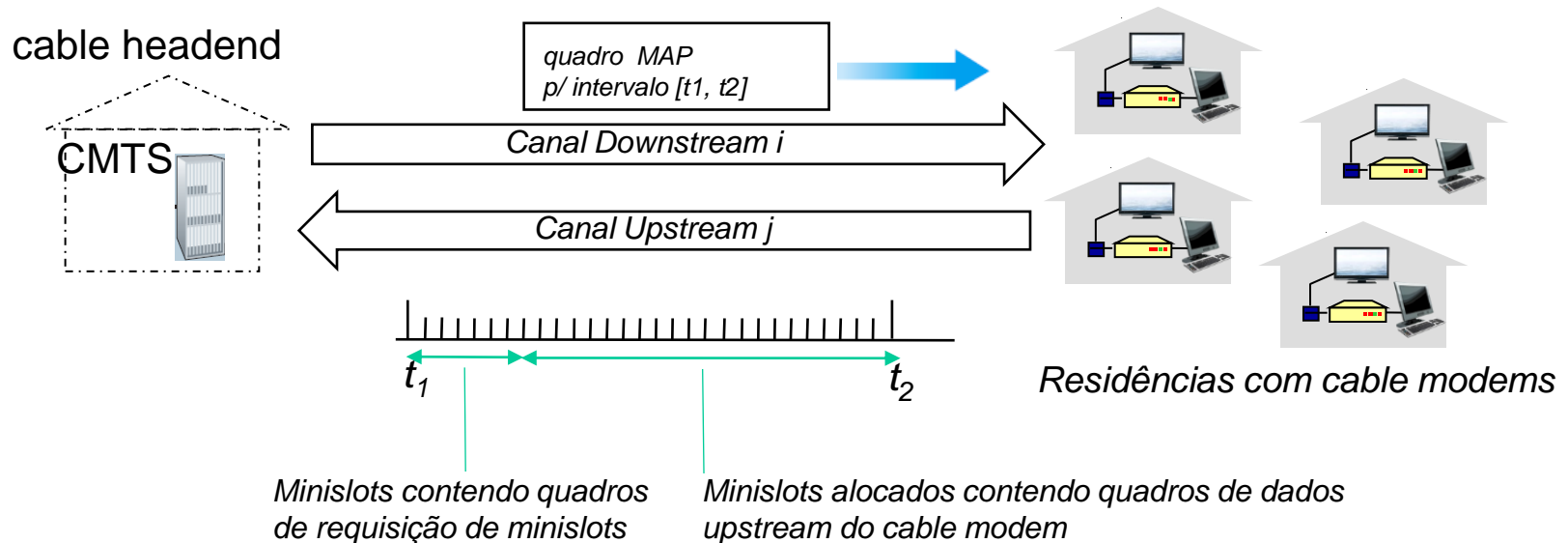
Rede de acesso a cabo: FDM, TDM e acesso aleatório



- **Múltiplos** canais de difusão *downstream* (de até 1.6 Gbps/canal)
 - Um único CMTS transmite para os diversos canais **(sem colisão!)**
- **Múltiplos** canais *upstream* (de até 1 Gbps/canal)

Acesso múltiplo: todos os usuários (cable modems) disputam alguns **slots** de tempo do canal *upstream* para seus quadros de requisição (O CMTS regula quem pode transmitir em cada slot e por qto tempo).

Rede de acesso a cabo



DOCSIS: especificação da interface de serviço de dados sobre cabo

FDM sobre as frequências dos canais *up* e *downstream*

TDM *upstream*: alguns slots são alocados, outros têm disputa

- quadro de mapeamento *downstream*: aloca slots *upstream*
- requisição de slots *upstream* (e dados) são transmitidos através de acesso aleatório (com *backoff exponencial*) em slots selecionados

Resumo dos protocolos MAC

- ***divisão do canal*** por tempo, frequência ou código
 - **divisão de tempo, divisão de frequência**
- ***acesso aleatório*** (dinâmico):
 - ALOHA, S-ALOHA, CSMA, CSMA/CD
 - CSMA/CD usado na Ethernet
 - CSMA/CA usado no 802.11 (Wi-fi)

Obs:

Protocolos com detecção de portadora: fácil em algumas tecnologias (cabeadas), difícil em outras (sem fio)

- ***Revezamento***
 - Seleção (*polling*) a partir de um ponto central, passagem de permissões
- Ex: *Bluetooth, FDDI, Token Ring***