

Redes de Computadores 2

Parte 02 – camada de enlace – protocolos para acesso múltiplo

Prof. Vinicius da Cunha (Slides concedidos gentilmente pelo Prof. Kleber Vieira Cardos)



INSTITUTO DE
INFORMÁTICA
UFG

Tópicos

- Protocolos para acesso múltiplo
 - Divisão (estática) do canal
 - Particionamento aleatório (dinâmico)
 - Revezamento

Enlaces e protocolos de acceso múltiplo

Dois tipos de enlaces:

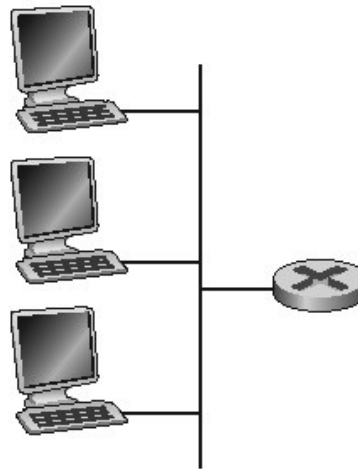
- Ponto-a-ponto:

- Acesso discado
- ADSL
- Conexão entre *switch* Ethernet e *host* (Ethernet comutado)

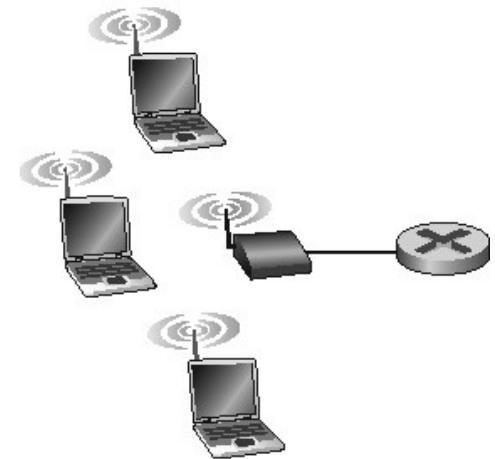
- *Difusão ou broadcast* (meio compartilhado)

- Ethernet tradicional
- *Upstream* HFC
- 802.11 (ou WiFi)
- Bluetooth
- Satélite

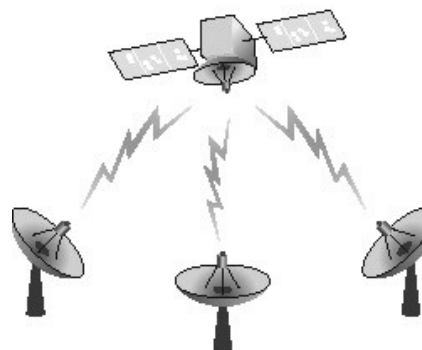
Compartilhado com fio
(por exemplo, Ethernet)



Compartilhado sem fio
(por exemplo, Wi-Fi)



Satélite



Coquetel



Protocolos de acesso múltiplo

- Principais características:
 - canal de comunicação único de difusão
 - interferência: quando dois ou mais nós transmitem simultaneamente
 - **colisão** se um nó receber dois ou mais sinais ao mesmo tempo

Protocolo de acesso múltiplo

- Algoritmo distribuído que determina como os nós compartilham o canal, isto é, determina quando um nó pode transmitir
- Comunicação sobre o compartilhamento do canal deve usar o próprio canal!
 - não há canal fora da faixa (ou canal de controle) para coordenar a transmissão
- Também conhecido protocolo de controle de acesso ao meio (*Media Access Control* ou MAC)
- É um dos problemas mais explorados na pesquisa em redes

Protocolo ideal de acesso múltiplo

Para um canal de difusão com taxa de R bps:

1. Quando apenas um nó tem dados para enviar, esse nó obtém uma vazão de R bps
2. Quando M nós têm dados para enviar, cada um desses nós poderá transmitir em média a uma taxa de R/M bps
3. Completamente descentralizado
 - nenhum nó especial (mestre) para coordenar as transmissões
 - nenhuma sincronização de relógios ou *slots*
4. Simples, para que sua implementação seja barata

Taxonomia dos protocolos MAC

Três categorias gerais:

□ Divisão (estática) de canal

- divide o canal em pequenos “pedaços” (*slots*/fatias de tempo, frequência, código)
- aloca pedaço a um dado nó para uso exclusivo deste

□ Acesso aleatório (dinâmico)

- canal não é dividido, podem ocorrer colisões
- realiza a “recuperação” das colisões

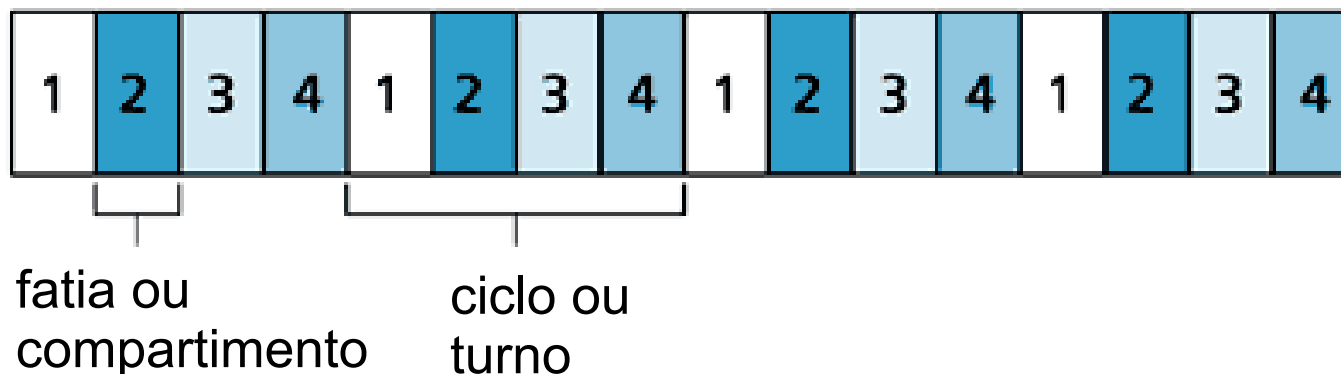
□ Revezamento

- nós se alternam em revezamento, mas nós que possuem mais dados a transmitir podem demorar mais quando chegar a sua vez

Protocolos MAC de divisão de canal: TDM

TDM: Acesso Múltiplo por Divisão de Tempo

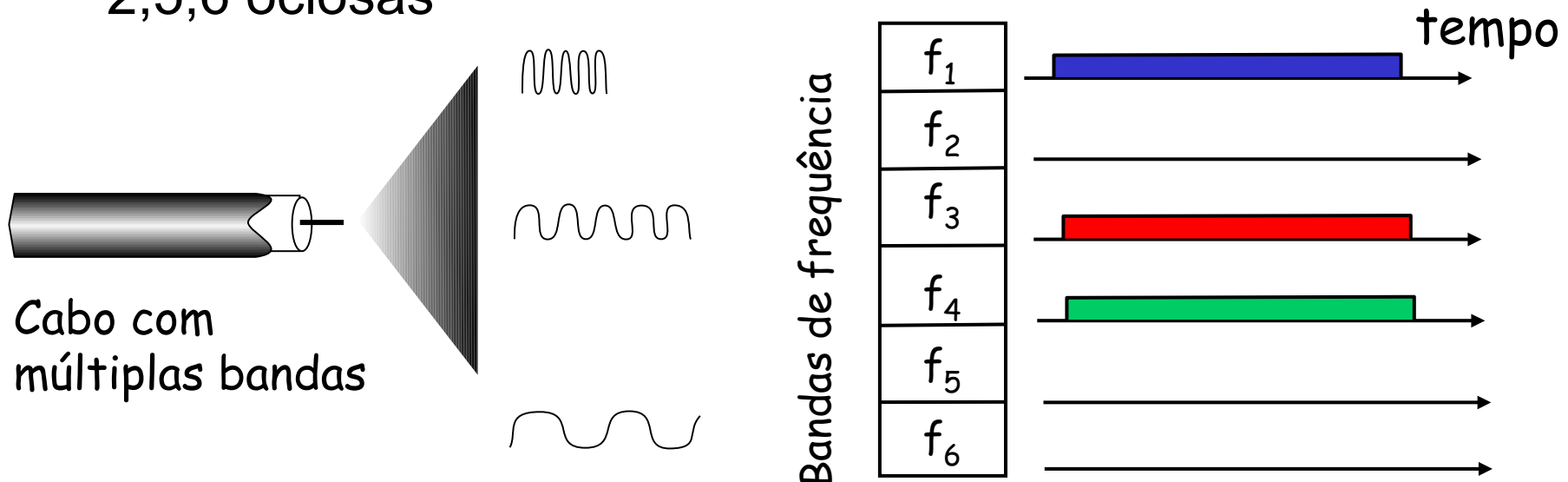
- acesso ao canal em “turnos” ou ciclos
- cada estação recebe um comprimento fixo de compartimento (comprimento = tempo de Tx do pacote) em cada ciclo
- compartimentos não usados permanecem ociosos
- Exemplo: LAN com 4 estações; compartimentos 2, 3 e 4 com pacotes; compartimento 1 ocioso



Protocolos MAC de divisão de canal: FDM

FDM: Acesso Múltiplo por Divisão de Frequência

- espectro do canal dividido em bandas de frequência
- a cada estação é atribuída uma banda fixa de frequência
- tempo de transmissão não usado nas bandas permanecem ociosos
- Exemplo: LAN com 6 estações; 1,3,4 com pacotes; bandas 2,5,6 ociosas



Protocolos MAC de divisão de canal: CDM

CDM: Acesso Múltiplo por Divisão de Código

- atribui um código diferente para cada estação
- com escolha adequada de código, permite transmissão simultânea das estações
- apresenta propriedades interessantes, como boa imunidade a interferências
- Sua principal aplicação é em rede sem fio, portanto será abordado posteriormente em mais detalhes

Protocolos de acesso aleatório

- Quando nó tem um pacote para transmitir
 - transmite na taxa máxima R
 - nenhuma coordenação *a priori* entre os nós
- Se dois ou mais nós transmitem → “colisão”
- O **protocolo MAC de acesso aleatório** especifica:
 - como detectar colisões
 - como se recuperar delas (por exemplo, através de retransmissões aleatoriamente adiadas)
- Exemplos de protocolos MAC de acesso aleatório:
 - ALOHA, *Slotted* ALOHA
 - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

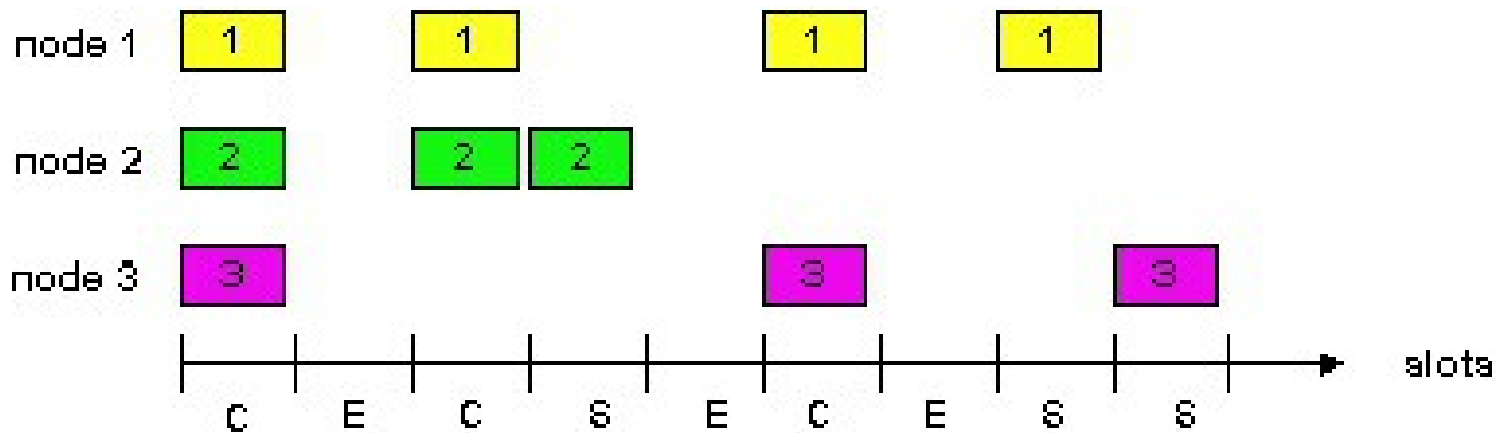
Hipóteses

- todos os quadros têm o mesmo tamanho (L bits)
- tempo é dividido em slots de tamanho igual, tempo para transmitir 1 quadro (L/R seg)
- nós começam a transmitir quadros apenas no início dos intervalos (*slots*)
- nós são sincronizados
- se 2 ou mais nós transmitirem num *slot*, todos os nós detectam a colisão

Operação

- quando o nó obtém um novo quadro, ele espera até o início do próximo *slot* e transmite o quadro inteiro
- Se não houver colisão, nó poderá enviar um novo quadro no próximo *slot*
- caso haja uma colisão (detectada antes do final do intervalo), nó retransmite o quadro em cada intervalo subsequente com probabilidade p até obter sucesso

Slotted ALOHA



Vantagens

- único nó ativo pode transmitir continuamente na taxa máxima do canal
- Altamente descentralizado: apenas *slots* nos nós precisam estar sincronizados
- simples

Desvantagens

- colisões: *slots* desperdiçados
- *slots* ociosos (desperdício)
- nós podem ser capazes de detectar colisões num tempo inferior ao da transmissão do pacote
- sincronização dos relógios

Eficiência do *Slotted* ALOHA

Eficiência é a fração de longo prazo de *slots* bem sucedidos quando há muitos nós cada um com muitos quadros para transmitir

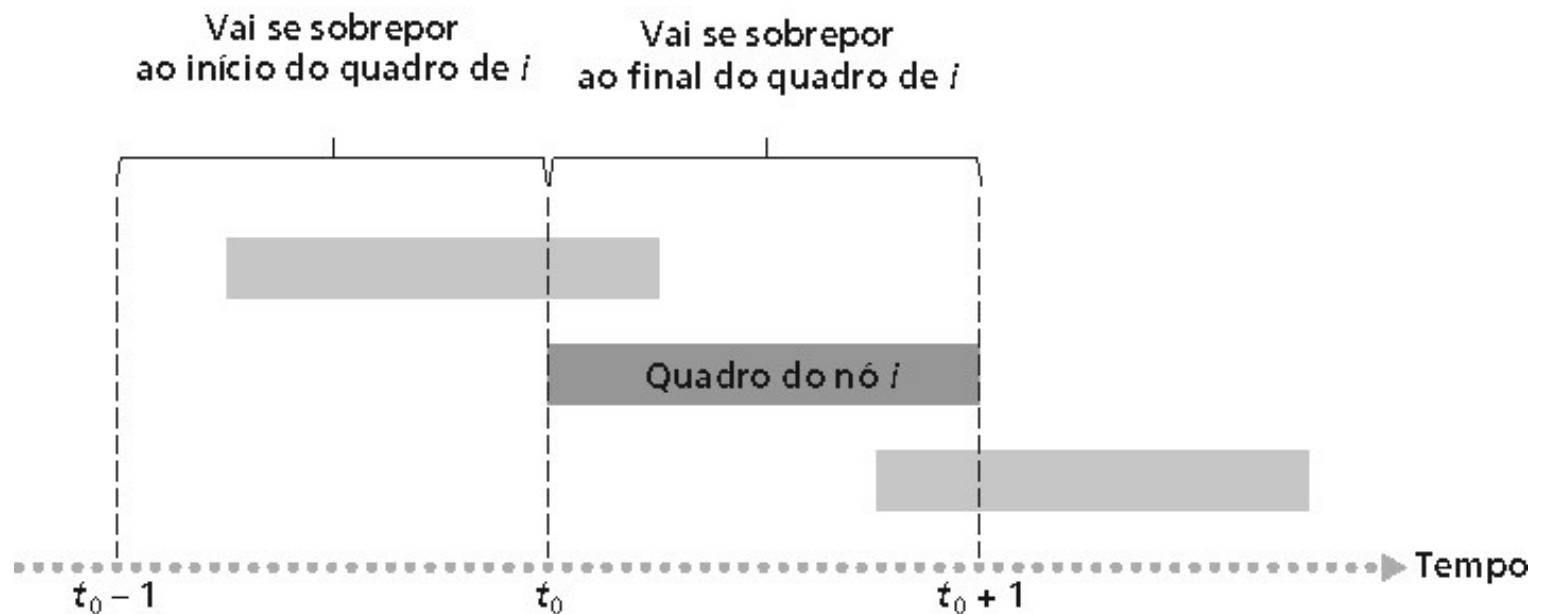
- Assuma N nós com muitos quadros para enviar, cada um transmite num *slot* com probabilidade p
- probabilidade que nó 1 tenha sucesso em um *slot* = $p(1-p)^{N-1}$
- probabilidade que qualquer nó tenha sucesso = $Np(1-p)^{N-1}$

- Para eficiência máxima com N nós, encontre p^* que maximiza $Np(1-p)^{N-1}$
- Assumindo muitos nós, calcule $Np^*(1-p^*)^{N-1}$ com N tendendo a infinito, o que resulta em $1/e = 0,37$

Melhor caso: canal usado para transmissões úteis em 37% do tempo!

ALOHA Puro (sem *slots*)

- ALOHA puro (sem *slots*): mais simples, sem sincronização
- Ao chegar um quadro no nó
 - transmite imediatamente
- Probabilidade de colisão aumenta:
 - quadro enviado em t_0 colide com outros quadros enviados em $[t_0-1, t_0+1]$



Eficiência do ALOHA puro

$$\begin{aligned} P(\text{sucesso de um dado nó}) &= P(\text{nó transmite}) \times \\ &\quad P(\text{nenhum outro nó transmite em } [t_0-1, t_0]) \times \\ &\quad P(\text{nenhum outro nó transmite em } [t_0, t_0+1]) \\ &= p \times (1-p)^{N-1} \times (1-p)^{N-1} \\ &= p \times (1-p)^{2(N-1)} \end{aligned}$$

... escolhendo o valor ótimo de p e deixando $n \rightarrow \text{infinito}$...

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

Ainda pior que o Slotted ALOHA!

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 (por um tempo aleatório)

Colisões no CSMA

colisões ainda *podem* acontecer?

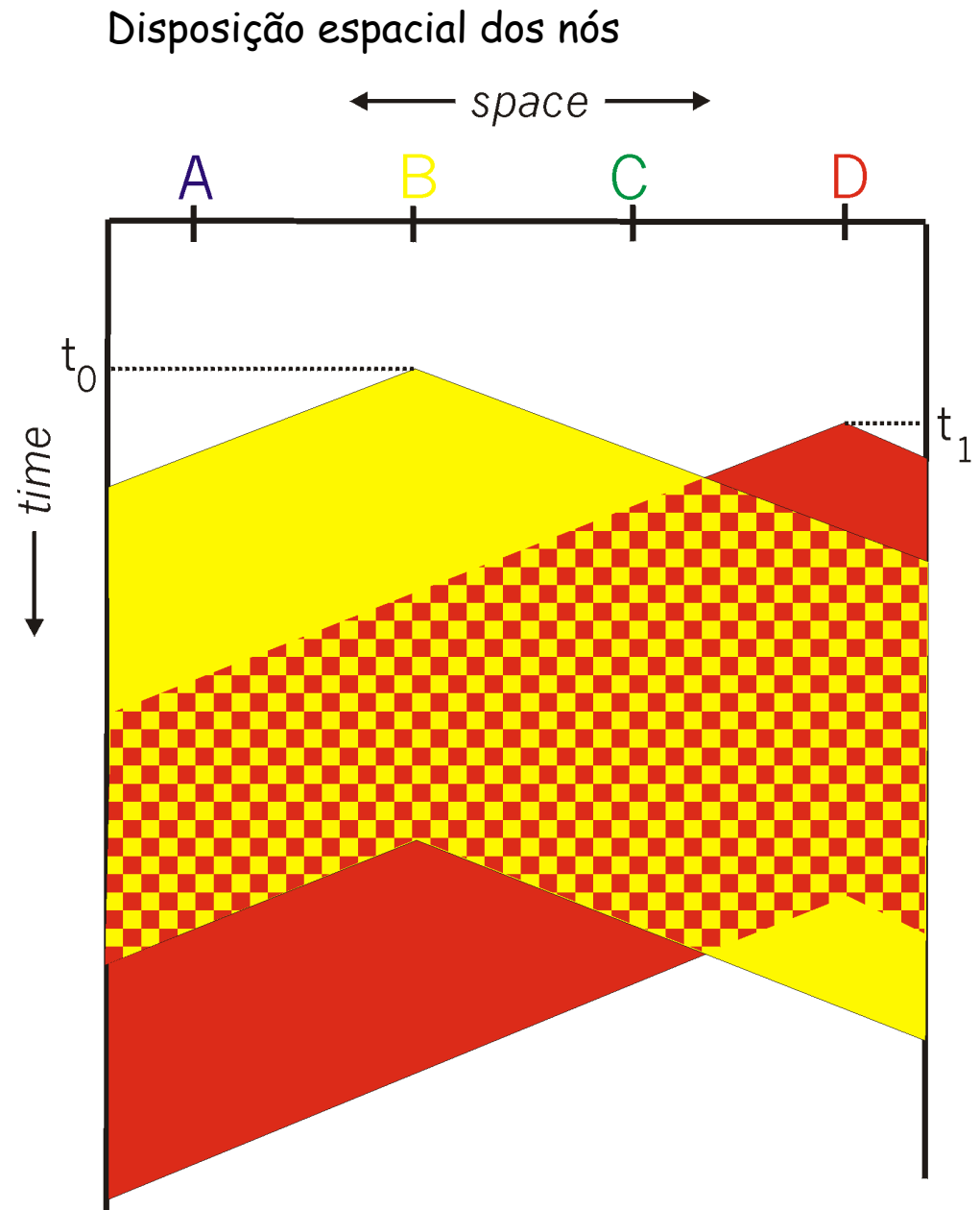
Sim, pois devido ao atraso de propagação, dois nós podem não ouvir a transmissão do outro

colisão:

todo o tempo de transmissão é desperdiçado

nota:

distância e atraso de propagação influenciam a probabilidade de colisão

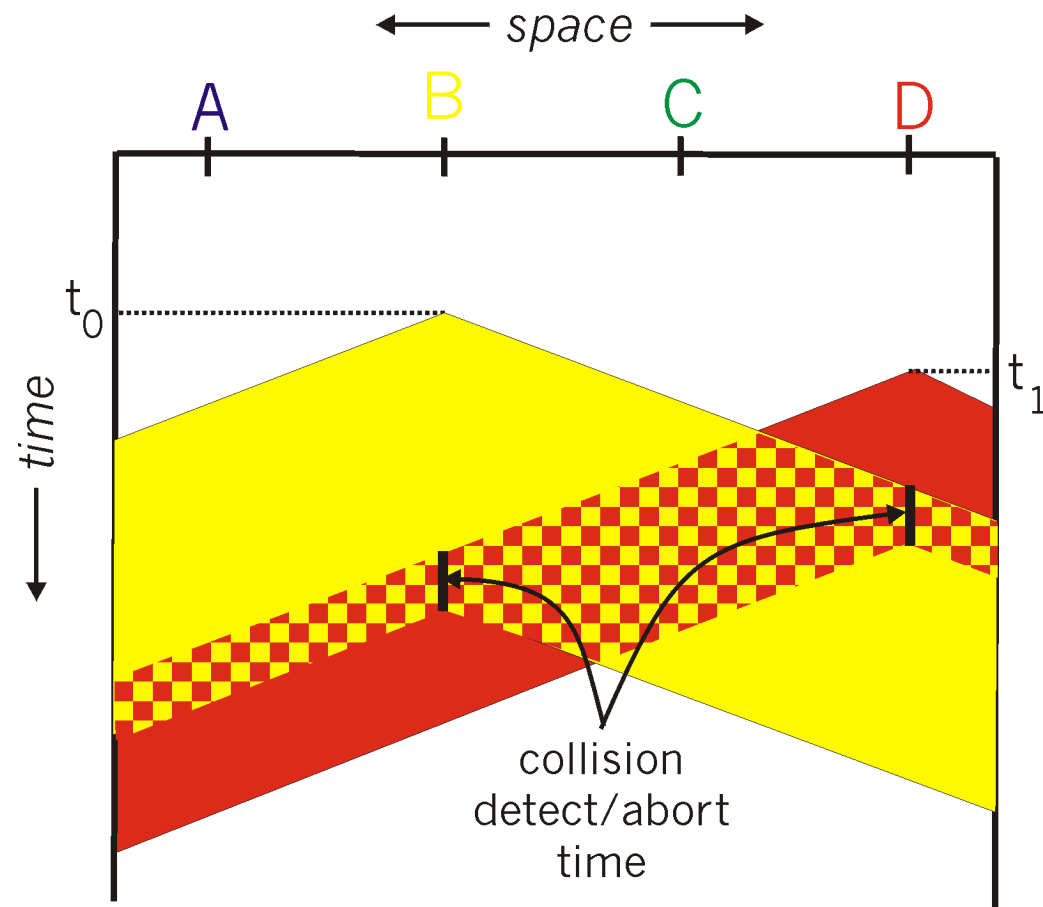


CSMA/CD (detecção de colisões)

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: não há garantia de detecção da colisão na interface que está transmitindo; comumente, as interfaces são *half-duplex*

Detecção de colisões em CSMA/CD



CSMA/CD – operação em uma NIC

1. A interface obtém um pacote da camada de rede, prepara o quadro correspondente e o coloca no *buffer* de transmissão
2. Se a interface observa o meio livre (não há energia no canal), tenta transmitir. Se o meio está ocupado, aguarda ficar livre e então tenta transmitir
3. Enquanto transmite, a interface verifica se há sinal (energia) de outras interfaces no meio
4. Se ao transmitir todo o quadro, não observa sinal de outras interfaces (ou seja, colisão), então quadro foi transmitido com sucesso e é removido do *buffer*. Se ao transmitir, observa colisão, então aborta transmissão
5. Após abortar a transmissão, a interface aguarda um tempo aleatório e então retorna ao passo 2

CSMA/CD – *backoff* exponencial binário

- Qual o melhor valor para a espera aleatória?
 - Um valor pequeno é adequado quando há poucas estações competindo
 - Um valor grande é adequado quando há muitas estações competindo
 - Solução: usar um valor que começa pequeno e aumenta caso as colisões se repitam
- *Backoff* exponencial binário
 - Após n colisões, a espera aleatória é um valor K sorteado no intervalo entre $\{0, 1, 2, \dots, 2^n - 1\}$
 - Exemplo: Ethernet, a espera aleatória é igual $K \times$ (tempo para transmitir 512 bits) e o n é no máximo 10
 - Importante: para cada novo quadro, *backoff* é reiniciado

CSMA/CD – eficiência

- Quando apenas uma estação transmite, toda capacidade da rede é utilizada
- Quando há múltiplas estações tentando transmitir, colisões reduzem a eficiência, ou seja, as transmissões com sucesso
- Qual a eficiência com muitas estações tentando transmitir e com grande número de pacotes para transmitir?

- Uma aproximação é dada pela seguinte expressão:

$$Eficiência = \frac{1}{1 + 5d_{prop} / d_{trans}}$$

- Onde:
 - d_{prop} – tempo máximo para a energia do sinal se propagar entre quaisquer duas NICs
 - Quanto mais próximo de zero, mais rápida a colisão é detectada
 - d_{trans} – tempo para transmitir um quadro de tamanho máximo
 - Quanto maior, mais tempo o canal realiza trabalho útil

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 colisões

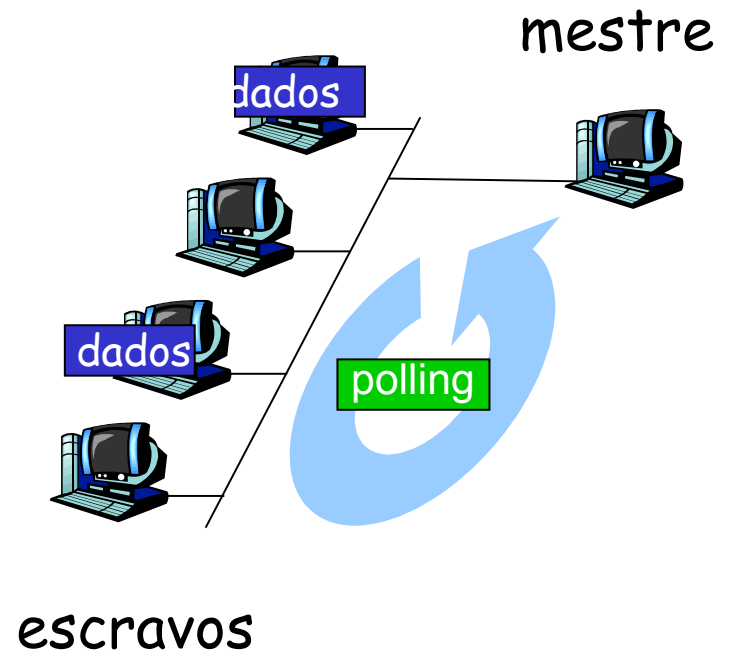
Protocolos de revezamento:

Procura oferecer o melhor das duas abordagens!

Protocolos MAC de “revezamento”

Seleção (*Polling*):

- Nó mestre “convida” nós escravos a transmitir em revezamento
- Cada nó escravo pode transmitir até um número máximo de quadros
- Preocupações:
 - Sobrecarga com as consultas (*polling*), pois mesmos nós inativos são consultados
 - Ponto único de falha (mestre)
- Exemplo: Bluetooth



Protocolos MAC de “revezamento”

Passagem de permissão (*token*):

- Controla **permissão** passada de um nó para o próximo de forma sequencial
 - Nó que tem dados a transmitir, transmite os dados e depois o *token*
 - Nó que não tem dados a transmitir, retransmite imediatamente o *token*
- Preocupações:
 - Sobrecarga com a passagem de permissão
 - Perda do *token*, por falha do nó, ou retenção do *token* por falha ou intencionalmente

