


Redes de Computadores

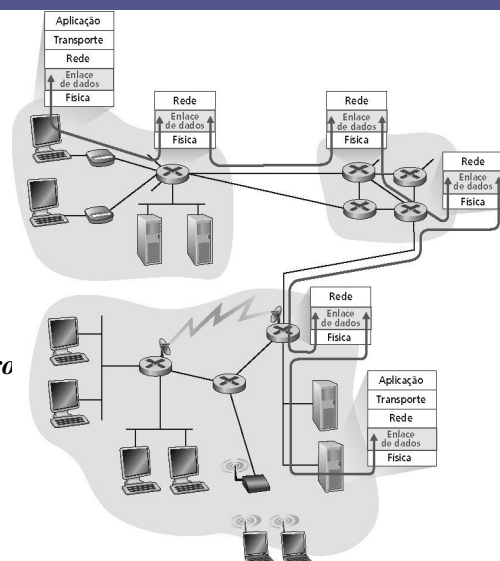
Parte III: Camada de Enlace

Professor: Reinaldo Gomes
reinaldo@computacao.ufcg.edu.br

 **Camada de enlace: introdução**

- Hospedeiros e roteadores são **nós**
- Canais de comunicação que conectam nós adjacentes ao longo do caminho de comunicação são **enlaces**
 - Enlaces guiados (cabado)
 - Enlaces não guiados (e.g., rádio)
- Entidade de camada 2: **quadro**

Camada de enlace tem a responsabilidade de transferir um datagrama de um nó para o nó adjacente sobre um enlace.






A camada de enlace: contexto

- Datagrama transferido por protocolos de enlace diferentes sobre enlaces diferentes:
 - ex.: *Ethernet* no primeiro enlace, *frame relay* nos enlaces intermediários, *802.11* no último enlace.
- Cada protocolo de enlace provê serviços diferentes
 - ex.: pode ou não prover transferência confiável sobre o enlace

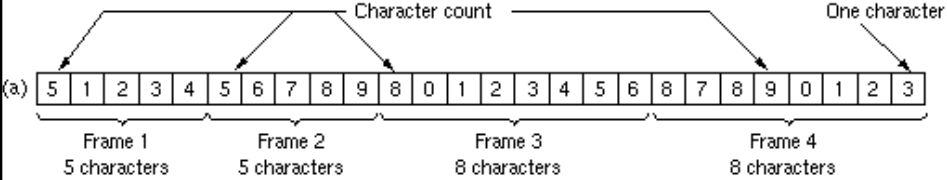


Serviços da Camada de Enlace

- Enquadramento e acesso ao enlace:
 - ▣ Encapsula datagramas em quadros acrescentando cabeçalhos e trailer
 - ▣ Implementa acesso ao canal se o meio é compartilhado - endereços físicos usados nos cabeçalhos dos quadros para Identificar a fonte e o destino dos quadros
 - ▣ Diferente do endereço IP !

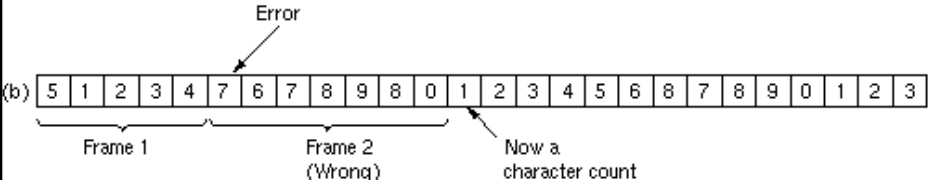

Delimitação dos Quadros

□ Contador de caracteres




(a) 5 1 2 3 4 5 6 7 8 9 8 0 1 2 3 4 5 6 8 7 8 9 0 1 2 3

Frame 1: 5 characters, Frame 2: 5 characters, Frame 3: 8 characters, Frame 4: 8 characters




(b) 5 1 2 3 4 7 6 7 8 9 8 0 1 2 3 4 5 6 8 7 8 9 0 1 2 3

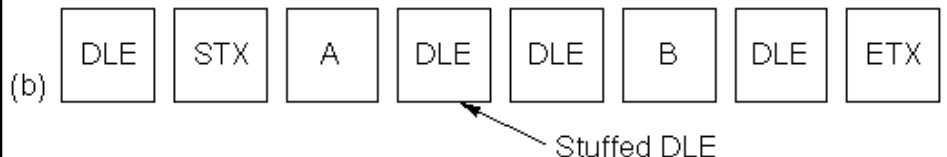
Frame 1, Frame 2 (Wrong), Now a character count


Delimitação dos Quadros

□ Caracteres de inicialização e finalização, com caracter de preenchimento




(a) DLE STX A DLE B DLE ETX




(b) DLE STX A DLE DLE B DLE ETX

Stuffed DLE



(c) DLE STX A DLE B DLE ETX

 **Delimitação dos Quadros**


- *Flags* de inicialização e finalização, com caracter de enchimento

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

Stuffed bits

(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

 **Serviços da Camada de Enlace**

- **Transferência dados da camada de rede** da máquina de origem para a camada de rede do destino
 - Não orientados a conexões e sem reconhecimento
 - Quadros independentes
 - Indicado para taxa de erros muito baixa ou tráfego em tempo real
 - Não orientados a conexões e com reconhecimento
 - Quadros confirmados individualmente
 - Útil em canais não confiáveis (ex.: sistemas sem fio)
 - Orientados a conexões e com reconhecimento
 - **Estabelecimento de conexão antes da transferência de dados**
 - Quadros confirmados individualmente
 - Garante que cada quadro será recebido uma única vez e na ordem correta



Serviços da Camada de Enlace

- Controle de fluxo:
 - Limitação da transmissão entre transmissor e receptor
- Detecção de erros:
 - Erros causados pela atenuação do sinal e por ruídos
 - O receptor detecta a presença de erros:
 - Avisa o transmissor para reenviar o quadro perdido
- Correção de erros:
 - O receptor identifica e corrige o(s) bit(s) errado(s) sem recorrer à retransmissão



Controle de Fluxo

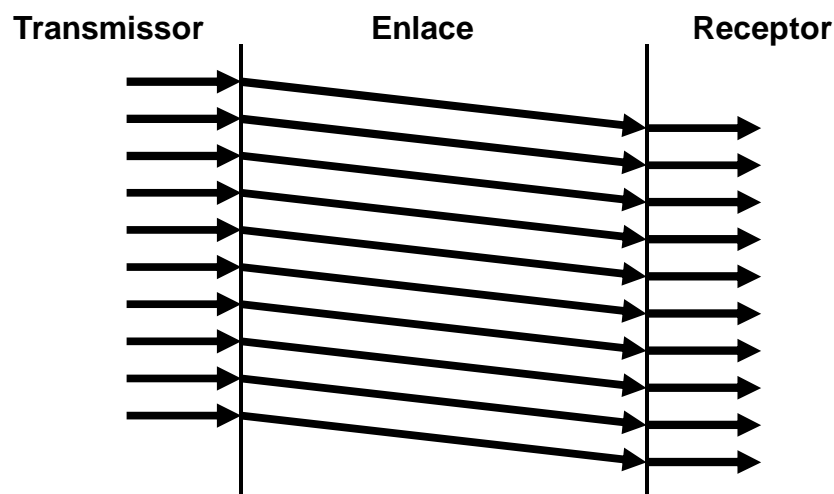



Protocolo Simplex sem restrições

- Transmissão de dados em um único sentido
- Camada de redes sempre pronta
- Buffer infinito
- Não há perda de quadros

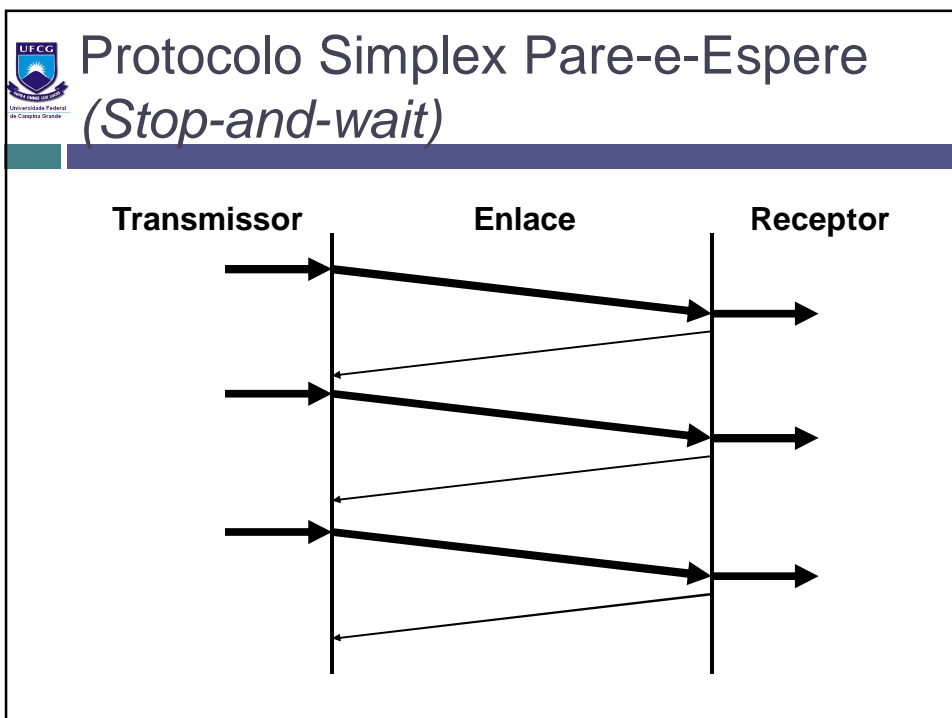


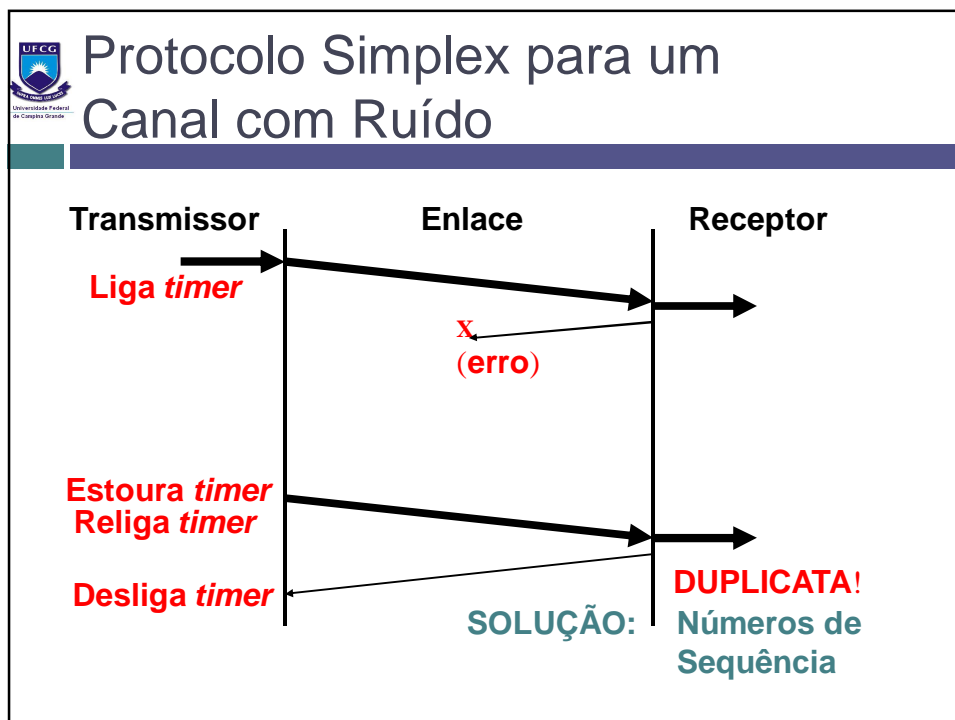
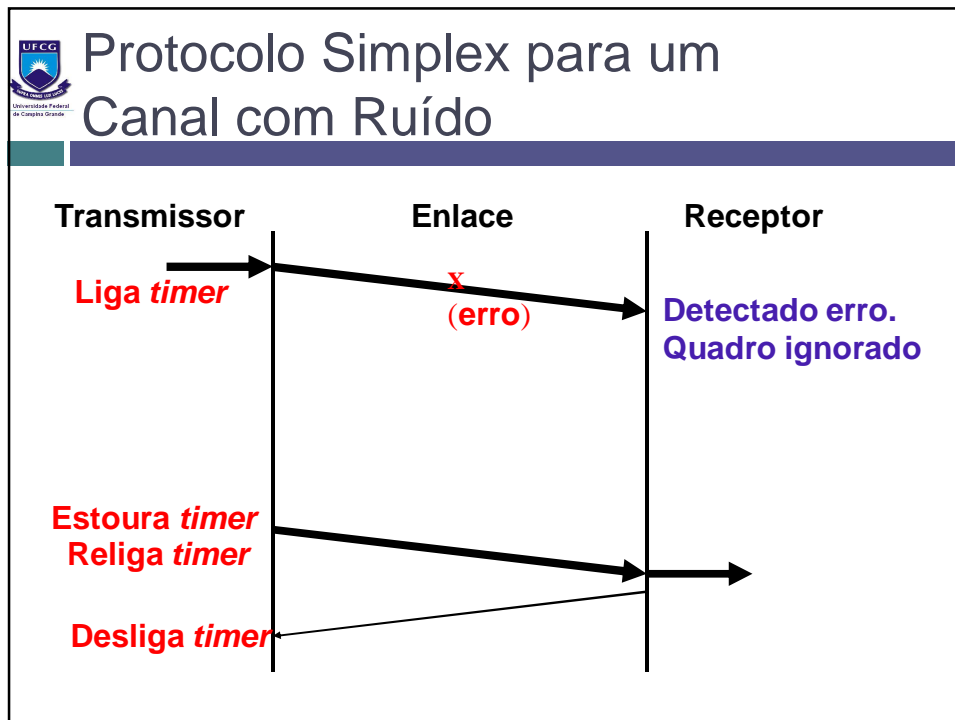
Protocolo Simplex sem restrições



 **Protocolo Simplex Pare-e-Espere
(Stop-and-wait)**

- Os buffers não são infinitos (controle de fluxo)
- O tempo de processamento não é ignorado
- O transmissor não envia outra mensagem até que a anterior tenha sido aceita como correta pelo receptor
- Embora o tráfego de dados seja simplex, há fluxo de quadros em ambos os sentidos







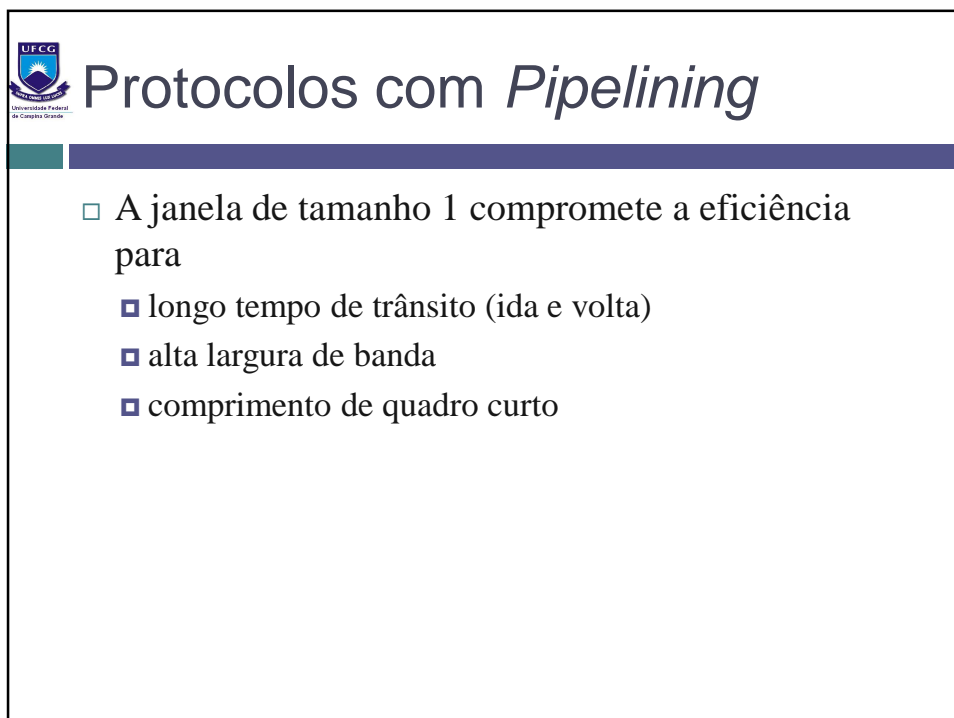
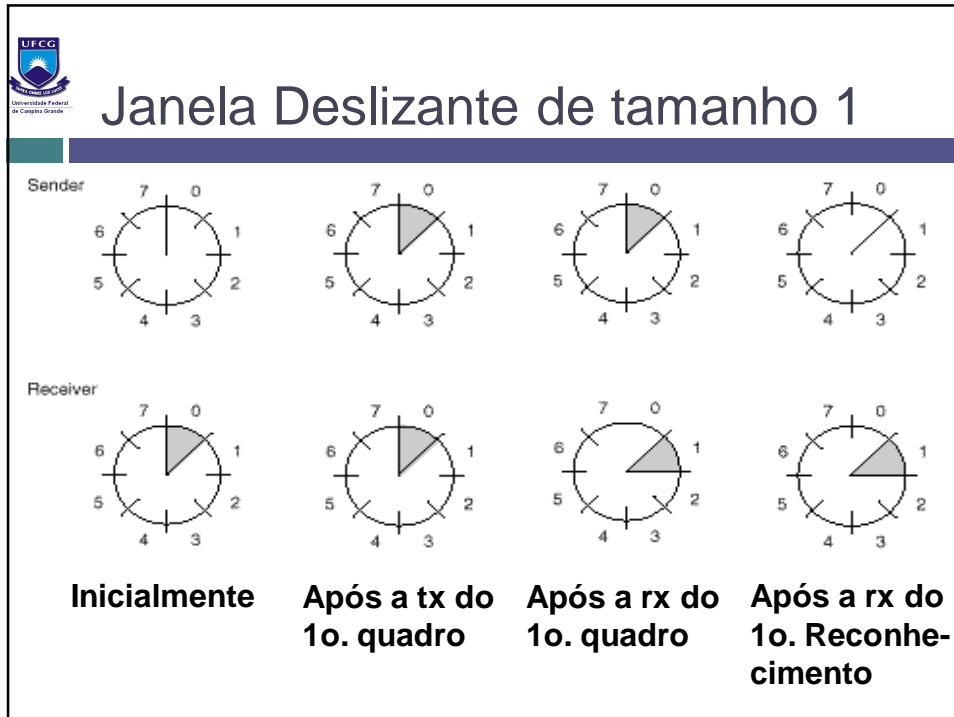
Protocolo Simplex para um Canal com Ruído

- Os quadros são numerados sequencialmente
- O transmissor transmite um quadro
- O receptor envia um quadro de reconhecimento se o quadro for recebido corretamente, caso contrário, há um descarte e é aguardada uma retransmissão
- Quadros não reconhecidos são retransmitidos (*timer*)



Protocolos de Janela Deslizante

- Transmissão de dados bidirecional
- Técnica de carona (*piggybacking*)
- Janelas para transmissão e recepção
 - Janela de transmissão
 - números de sequência habilitados para transmissão
 - Janela de recepção
 - números de sequência habilitados para recepção
- Os quadros são mantidos na memória para possível retransmissão





Protocolos com *Pipelining*

- Solução:
 - ▣ Deixar o transmissor transmitir até w quadros (sem receber o reconhecimento do primeiro) antes de ser bloqueado.
 - ▣ w deve ser escolhido de modo que o transmissor possa transmitir quadros por um tempo igual ao de trânsito, antes de *encher* a janela



Protocolos com *Pipelining*

- O que fazer se um **quadro no meio da janela for danificado ou perdido?**
- Abordagens:
 - ▣ ***Go-back n***
 - ▣ **Retransmissão Seletiva** (*Selective Reject*)

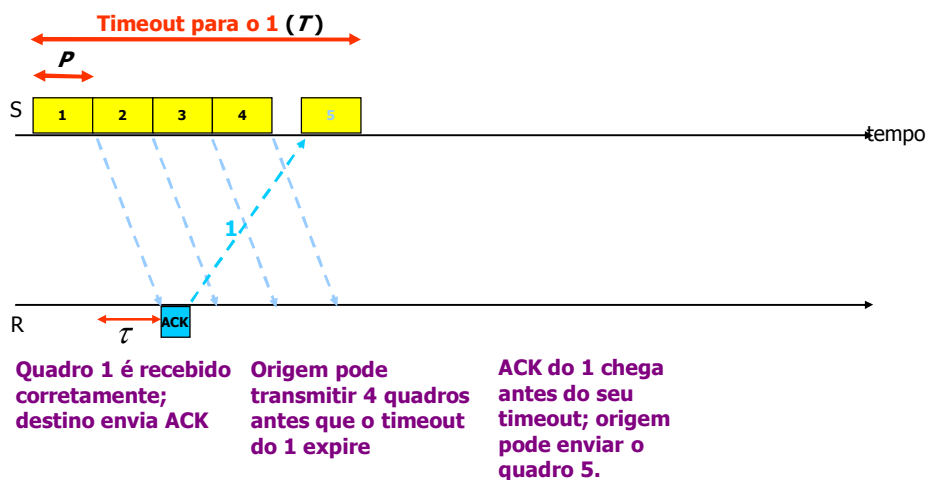


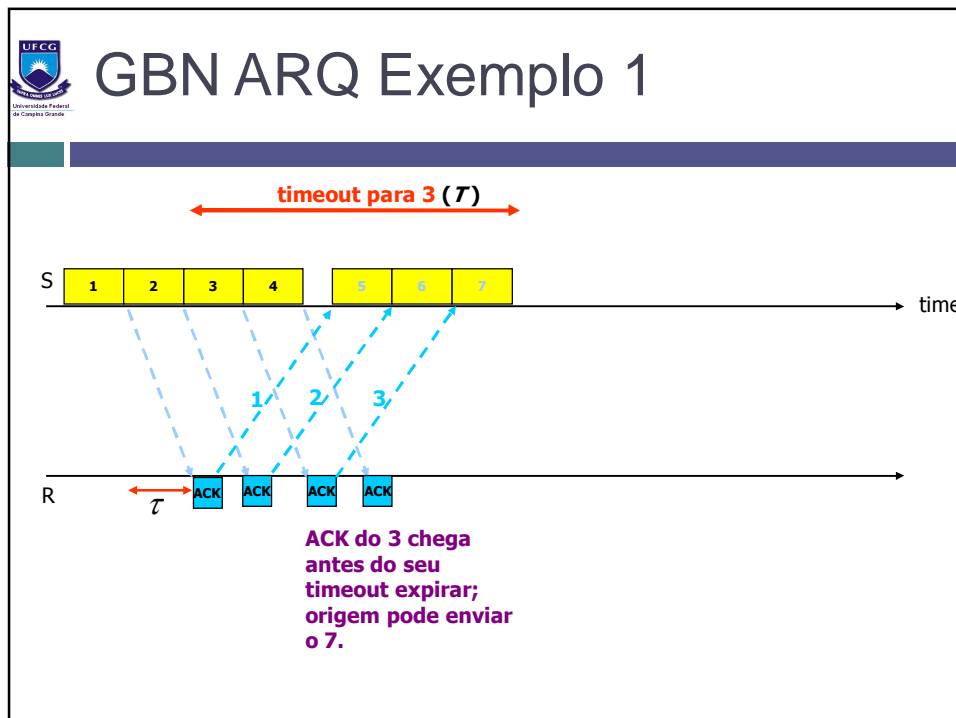
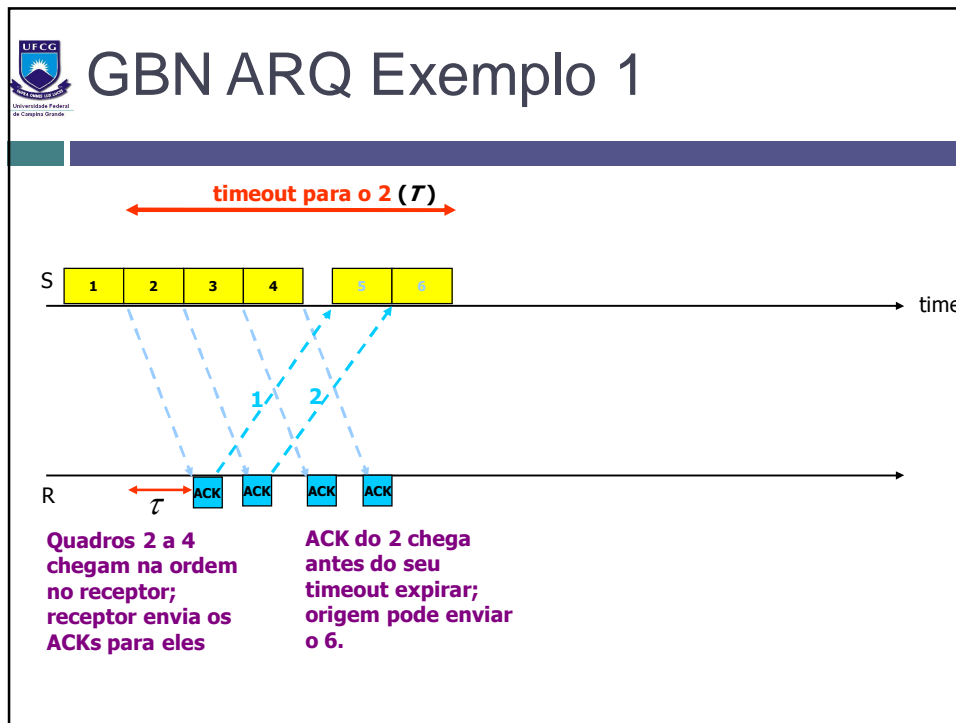
Protocolos com *Pipelining*

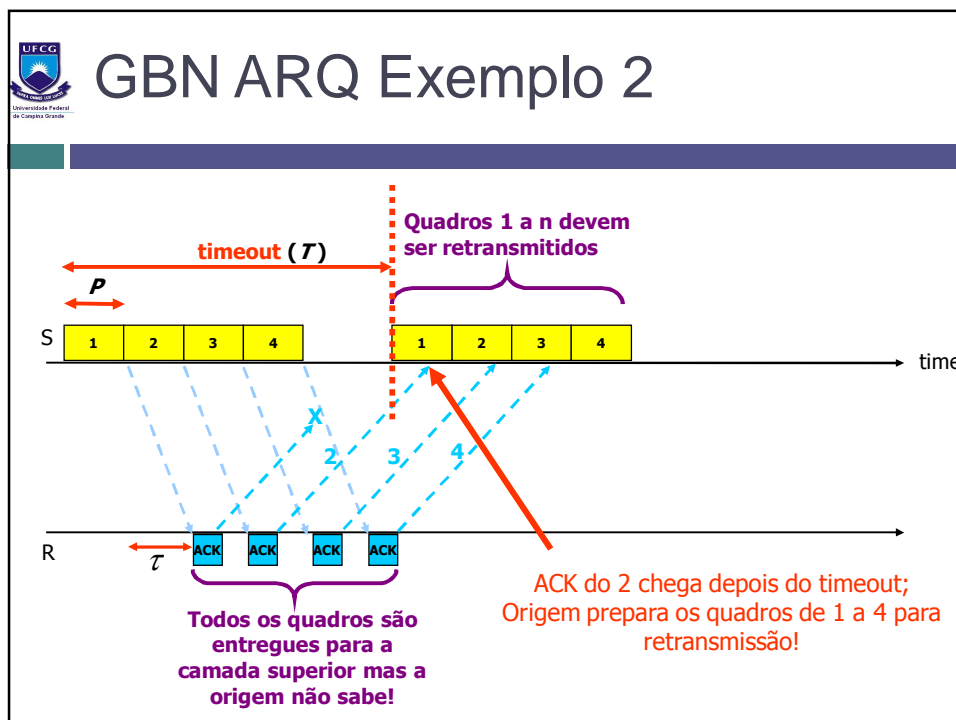
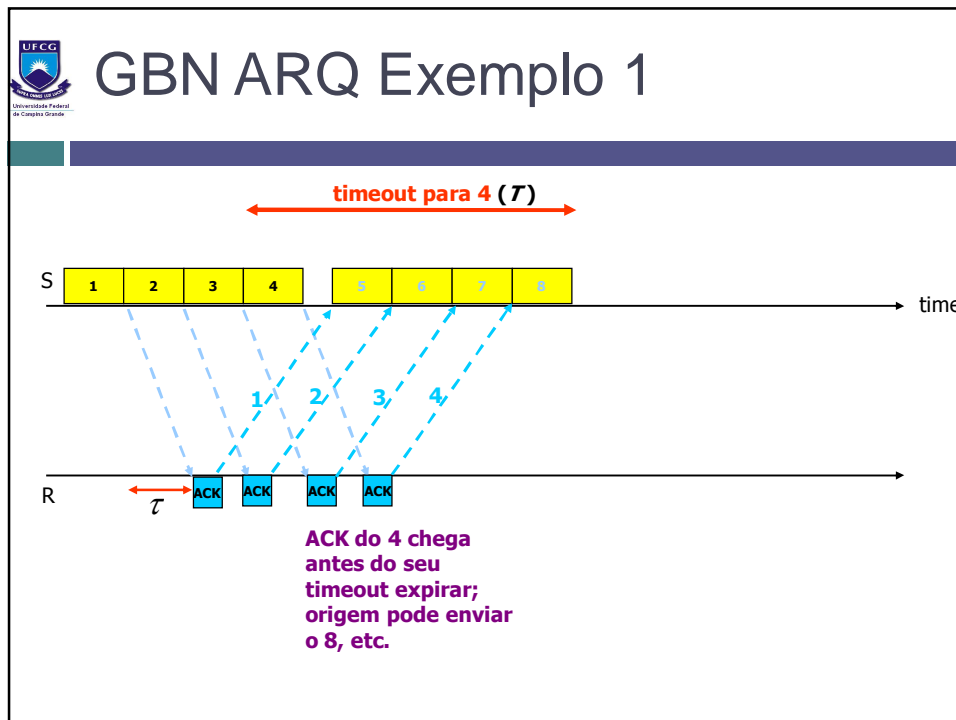
- Com GBN, o destino descarta qualquer quadro fora de ordem; portanto, não necessita de um *buffer*.
- Destino confirma (i.e., ACK) um quadro recebido corretamente com o número de sequência do último quadro recebido em ordem.
- A fonte inicializa um tempo de espera para cada quadro transmitido. Caso não receba confirmação dentro deste tempo, a fonte retransmite o quadro expirado e todos os quadros enviados após aquele quadro.
- A fonte pode ter até W quadros esperando por confirmação.

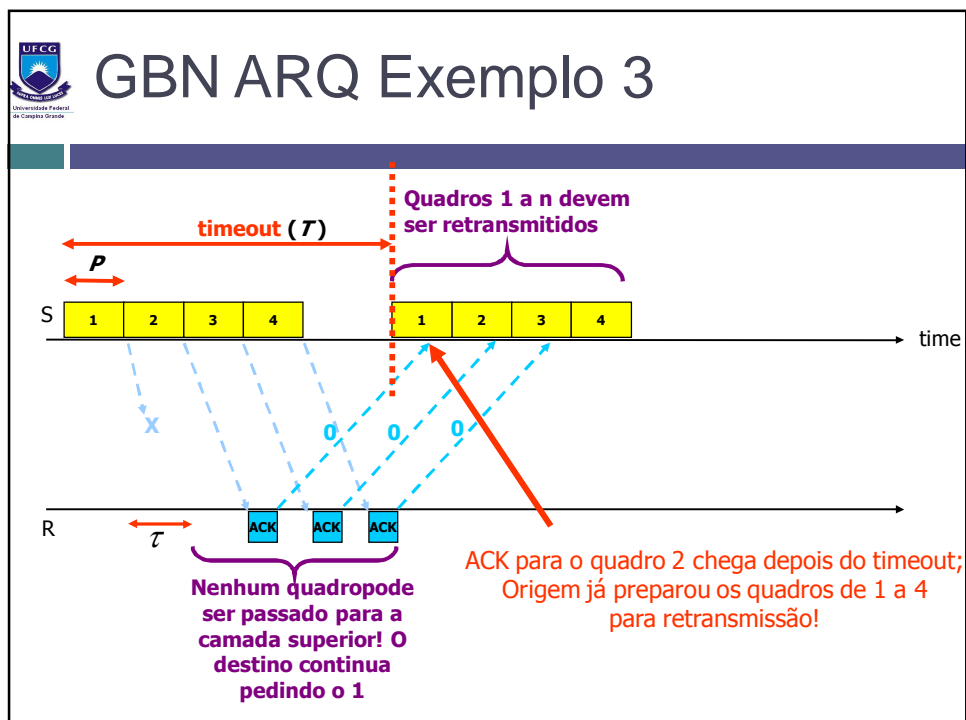
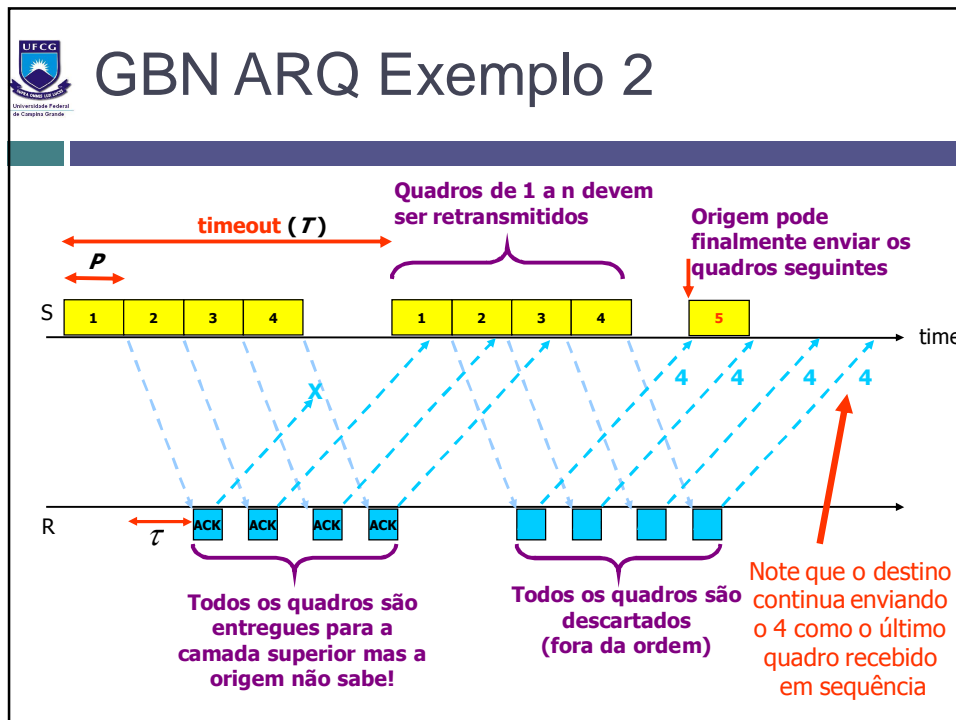


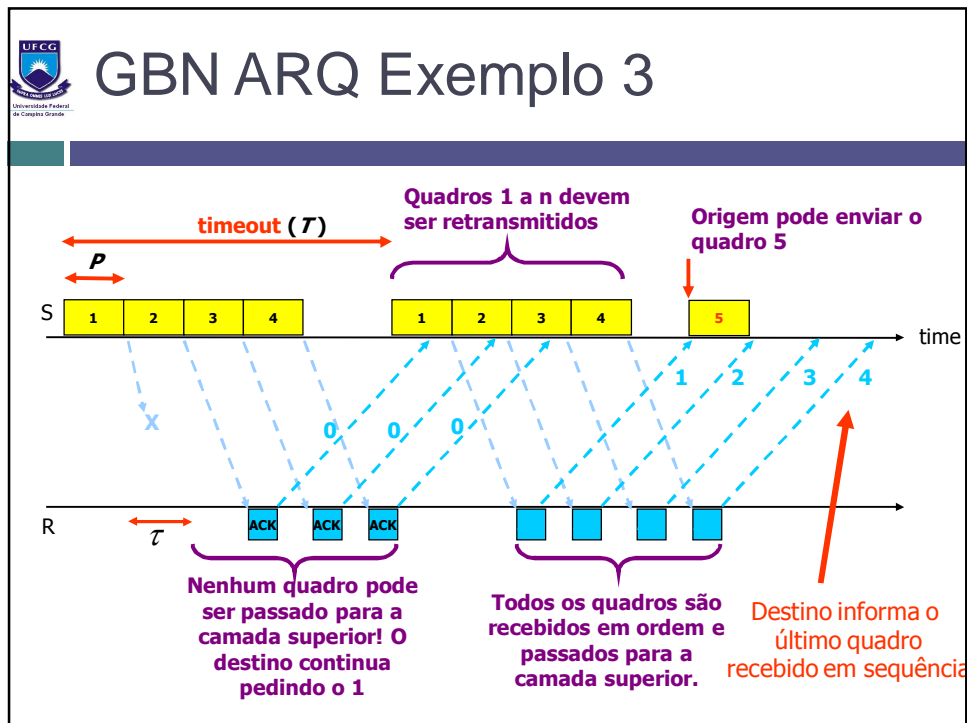
GBN ARQ Exemplo 1







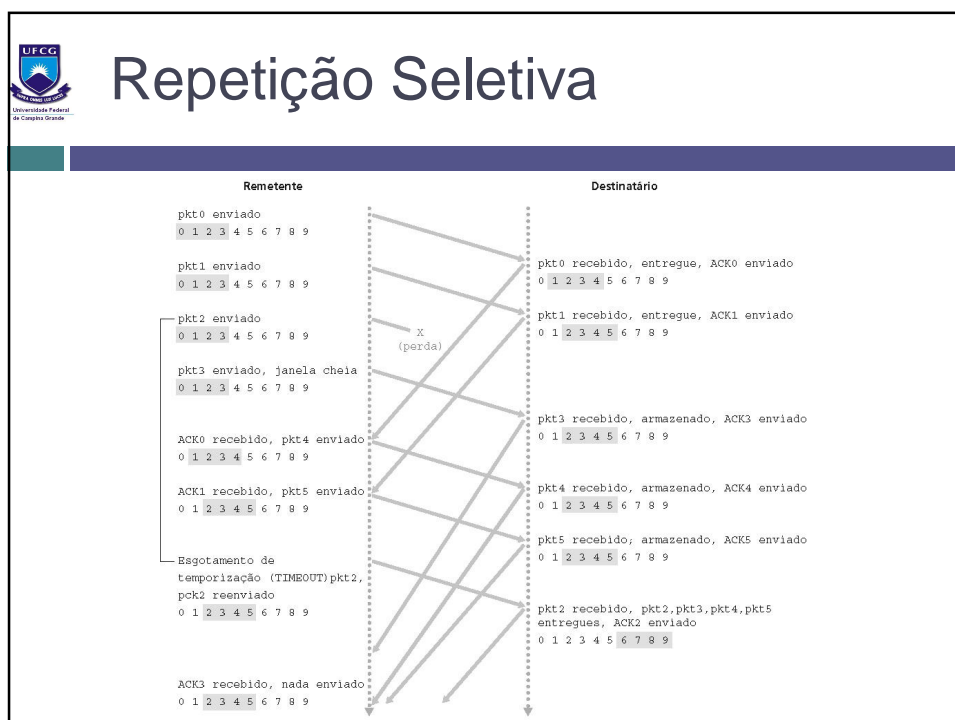
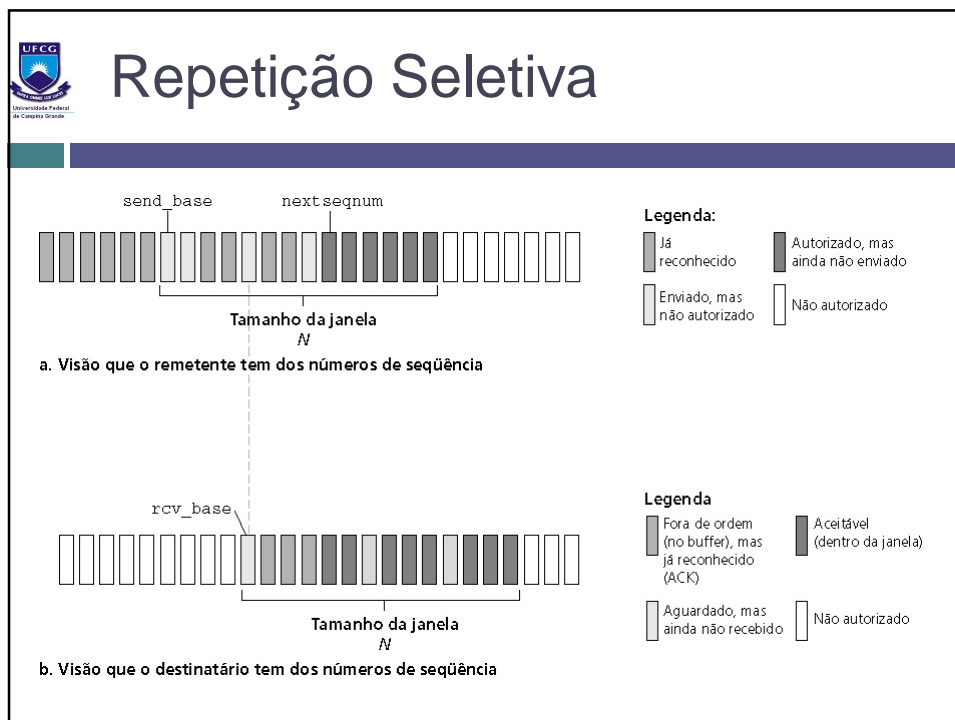





Protocolos com *Pipelining*

- **Retransmissão seletiva**
 - O nó armazena os quadros corretos que chegarem após o com erro. O tx retransmite apenas o quadro com erro.
 - Ao receber o quadro que faltava, o nó entrega os diversos quadros já recebidos rapidamente e envia um reconhecimento do quadro de ordem mais alta
 - Necessita de maior quantidade de memória no nó


Janela de recepção → maior que 1





Universidade Federal
de Campina Grande


Detecção e Correção de Erros



Universidade Federal
de Campina Grande

Detecção e Correção de Erros

- Detecção X Correção
 - ▣ Meio confiável → **detecção**
(+ retransmissão)
 - ▣ Meio com muito ruído → **correção**

 **Detecção e Correção de Erros**

□ Verificação de Paridade

Paridade com bit único:
Detecta erro de um único bit

Paridade bidimensional:
Detecta e corrige erro de um único bit

Diagrama de Paridade:

Paridade de linha →

Paridade de coluna ↓

Exemplo de Paridade com bit único:

d bits de dados | Bit de paridade

0 1 1 1 0 0 0 1 1 0 1 0 1 0 1 1 | 1

Verificação de Paridade:


Nenhum erro

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

Erro de bit único corrigível:

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

Erro de paridade

 **Detecção e Correção de Erros**

□ Quadro = dados + bits de verificação
(checksum)

$n = m + r$ Palavra do Código

□ Código de Hamming

□ Código de Redundância Cíclica (CRC)



Detecção e Correção de Erros

□ Código de Hamming

▣ Distância de Hamming:

$$\begin{array}{r} 10001001 \\ \text{XOR } 10110001 \\ \hline 00111000 \end{array}$$

- ▣ As propriedades de detecção e correção de erros de um código dependem de sua distância de Hamming

Detecção: d erros \rightarrow distância $d + 1$

Correção: d erros \rightarrow distância $2d + 1$



Detecção e Correção de Erros

□ Código de Hamming

- ▣ $n = m + r$

- ▣ r ocupa os bits que são potência de 2

- ▣ m ocupa os outros bits

- ▣ r força paridade par




Detecção e Correção de Erros

- Código de Hamming
 - ▣ Numere os bits
 - ▣ Preencha os bits da mensagem
 - ▣ Os bits de verificação resultam de um XOR das posições que influênciam



Detecção e Correção de Erros


- Código de Hamming
 - ▣ Exemplo:
- $m = 0 \quad 1 \quad 1 \quad 0$
 $\quad \quad m_1 \quad m_2 \quad m_3 \quad m_4$
 $T(x) = \underline{\quad} \underline{0} \underline{\quad} \underline{1} \underline{1} \underline{0} \underline{\quad} \underline{\quad}$
 $\quad \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8$
 $\quad \quad 2^0 \quad 2^1 \quad \quad 2^2 \quad \quad \quad 2^3$
 $\quad \quad x^1 \quad x^2 \quad m^1 \quad x^3 \quad m^2 \quad m^3 \quad m^4 \quad x^4$


Detecção e Correção de Erros

☐ Código de Hamming

☐ Exemplo: $m = 0 \ 1 \ 1 \ 0$

$x_1 = m_1 \oplus m_2 \oplus m_4 = 0 \oplus 1 \oplus 0 = 1$
 $x_2 = m_1 \oplus m_3 \oplus m_4 = 0 \oplus 1 \oplus 0 = 1$
 $x_3 = m_2 \oplus m_3 \oplus m_4 = 1 \oplus 1 \oplus 0 = 0$


Detecção e Correção de Erros

☐ Código de Hamming

☐ Exemplo:

$m = 0 \ 1 \ 1 \ 0$


$m_1 \ m_2 \ m_3 \ m_4$

$T(x) = \underline{1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ -}$

$1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8$

$2^0 \ 2^1 \ \quad 2^2 \quad \quad 2^3$

$x^1 \ x^2 \ m^1 \ x^3 \ m^2 \ m^3 \ m^4 \ x^4$


Detecção e Correção de Erros

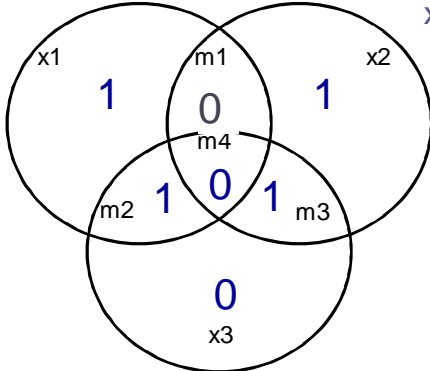
☐ Código de Hamming


☐ Exemplo:

$T(x) = 1100110$ $T(x) = 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0$

\rightarrow

$x1 \ x2 \ m1 \ x3 \ m2 \ m3 \ m4$




Detecção e Correção de Erros

☐ Código de Hamming

☐ Exemplo: $T(x) = 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0$

$x1 \ x2 \ m1 \ x3 \ m2 \ m3 \ m4$

$2^0 \ 2^1 \ 2^2$

$m1(3) = 1 + 2 = 2^0 + 2^1 = x1, x2$

$m2(5) = 1 + 4 = 2^0 + 2^2 = x1, x3$


$m3(6) = 2 + 4 = 2^1 + 2^2 = x2, x3$

$m4(7) = 1 + 2 + 4 = 2^0 + 2^1 + 2^2 = x1, x2, x3$

$x1 = m1 \oplus m2 \oplus m4$


$x2 = m1 \oplus m3 \oplus m4$

$x3 = m2 \oplus m3 \oplus m4$



Detecção e Correção de Erros

- Código de Hamming
 - ▣ Exemplo: $T(x) = \overset{1}{1} \overset{2}{1} \overset{3}{0} \overset{4}{0} \overset{5}{1} \overset{6}{1} \overset{7}{0}$
 $\quad \quad \quad x_1 \ x_2 \ m_1 \ x_3 \ m_2 \ m_3 \ m_4$
 $\quad \quad \quad 2^0 \ 2^1 \quad \quad 2^2$

	X1	X2	X3
Sem ruído	1	1	0
Ruído em m1	0	0	0
Ruído em m2	0	1	1
Ruído em m3	1	0	1
Ruído em m4	0	0	1



Detecção e Correção de Erros

- Código Polinomial ou Código de Redundância Cíclica (CRC)
 - ▣ Polinômio: **110001** $\rightarrow x^5 + x^4 + x^0$
 (6 bits \rightarrow grau 5)
- Procedimento no transmissor
 - ▣ Seja $G(x)$ o gerador polinomial de grau r . Acrescente r bits 0 ao quadro
 - ▣ Divida o novo quadro pelo gerador $G(x)$
 - ▣ Subtraia do quadro original o resto da divisão (*sem ÷ vai umö*)


Detecção e Correção de Erros

- ☐ Código de Redundância Cíclica (CRC)
 - ☐ Exemplo: $M(x) = 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 0$
 $G(x) = 1\ 0\ 0\ 1\ 1$

$1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 0 \mid 1\ 0\ 0\ 1\ 1$


Detecção e Correção de Erros

- ☐ Código de Redundância Cíclica (CRC)
 - $1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 0 \mid 1\ 0\ 0\ 1\ 1$
 $1\ 0\ 0\ 1\ 1$

$1\ 0\ 0\ 1\ 1$

$1\ 0\ 0\ 1\ 1$

$1\ 0\ 0\ 1\ 1$

$0\ 0\ 0\ 0\ 1$


$0\ 0\ 0\ 0\ 0$

$0\ 0\ 0\ 1\ 0$


\dots

$1\ 1\ 1\ 0 \leftarrow \text{Resto}$

$T(x) = 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ 0$

 **UFCG**
Universidade Federal
de Campina Grande

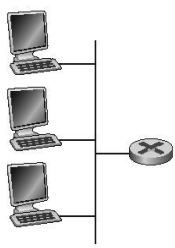
Protocolos de acesso a meio compartilhado

 **UFCG**
Universidade Federal
de Campina Grande


Enlaces de acesso múltiplo e protocolos

- Ponto-a-ponto: fio único (e.g., PPP, SLIP)
- Broadcast: fio ou meio compartilhado:
 - Ethernet tradicional
 - 802.11 LAN sem fio

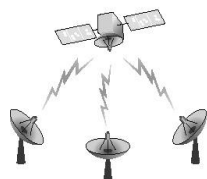
Compartilhado com fio
(por exemplo, Ethernet)



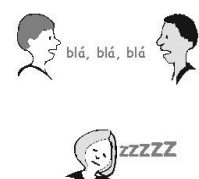
Compartilhado sem fio
(por exemplo, Wi-Fi)



Satélite



Coquetel





Protocolos de acesso múltiplo

- Canal de comunicação **único e compartilhado**
- Duas ou mais transmissões simultâneas pelos nós: **interferência!**
 - **Colisão:** um nó recebe dois ou mais sinais simultaneamente!
- **Protocolo de múltiplo acesso:**
 - Algoritmo distribuído que determina **como as entidades compartilham o canal**; isto é, determinam **quando cada estação pode transmitir**
 - Comunicação sobre o compartilhamento do canal deve utilizar o próprio canal!
 - Ou seja, nenhum canal fora-de-banda (i.e., out-of-band) para coordenação!



Protocolo ideal de múltiplo acesso

Exemplo: canal de broadcast de taxa R bps

1. Quando um nó quer transmitir, ele pode enviar a uma taxa R .
2. Quando M nós querem transmitir, cada um envia a uma taxa média R/M
3. Totalmente **descentralizada**:
 - Nenhum nó especial para coordenar transmissões
 - Nenhuma sincronização de relógios e compartimentos
4. Simples



Protocolos MAC: uma taxonomia

Três grandes classes:


- **Particionamento de canal**
 - Divide o canal em pedaços (i.e., *slots*) menores : compartimentos de tempo, frequência, etc.
 - Aloca um pedaço para uso exclusivo de cada nó
- **Acesso aleatório**
 - Canal não dividido, permite colisões
 - Necessita mecanismos para recuperação das colisões
- **Passagem de permissão**
 - Nós transmitem nos seus turnos (mas com mais volume para enviar podem usar turnos mais longos)



Protocolos MAC com particionamento de canal: TDMA

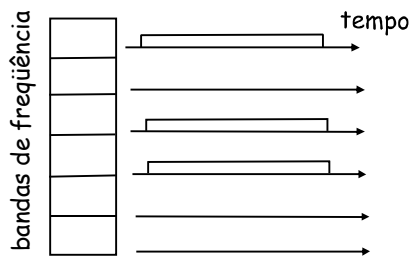
TDMA: acesso múltiplo por divisão temporal


- Acesso ao canal é feito por "turnos"
- Cada estação controla um compartimento (slot) de tamanho fixo (tamanho = tempo de transmissão de quadro) em cada turno
- Compartimentos não usados são desperdiçados
- Exemplo: rede local com 6 estações: 1, 3, 4 têm quadros, compartimentos 2, 5, 6 ficam vazios

 **Protocolos MAC com particionamento de canal: FDMA**


FDMA: acesso múltiplo por divisão de frequência

- O espectro do canal é dividido em bandas de frequência
- Cada estação recebe uma banda de frequência
- Tempo de transmissão não usado nas bandas de frequência é desperdiçado
- Exemplo: rede local com 6 estações: 1, 3, 4 têm quadros, as bandas de frequência 2, 5, 6 ficam vazias



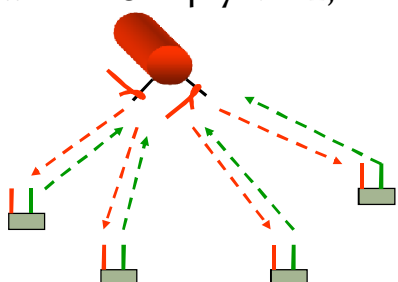
 **Protocolos de acesso aleatório**


- Quando o nó tem um quadro a enviar:
 - Transmite com toda a taxa do canal R.
 - Não há uma regra de coordenação **a priori** entre os nós
- Dois ou mais nós transmitindo -> colisão
- **Protocolo MAC de acesso aleatório** especifica:
 - Como detectar colisões
 - Como as estações se recuperam das colisões (ex., via retransmissões atrasadas)
- Exemplos de protocolos MAC de acesso aleatório:
 - **Slotted ALOHA**
 - **ALOHA**
 - **CSMA e CSMA/CD**

 **Protocolo ALOHA**

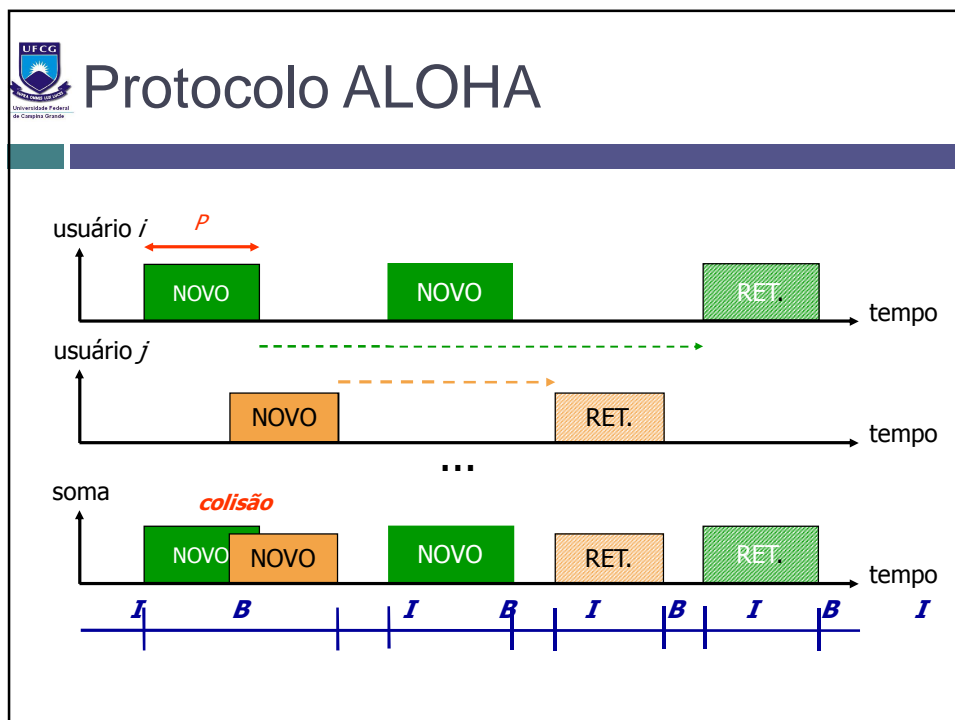
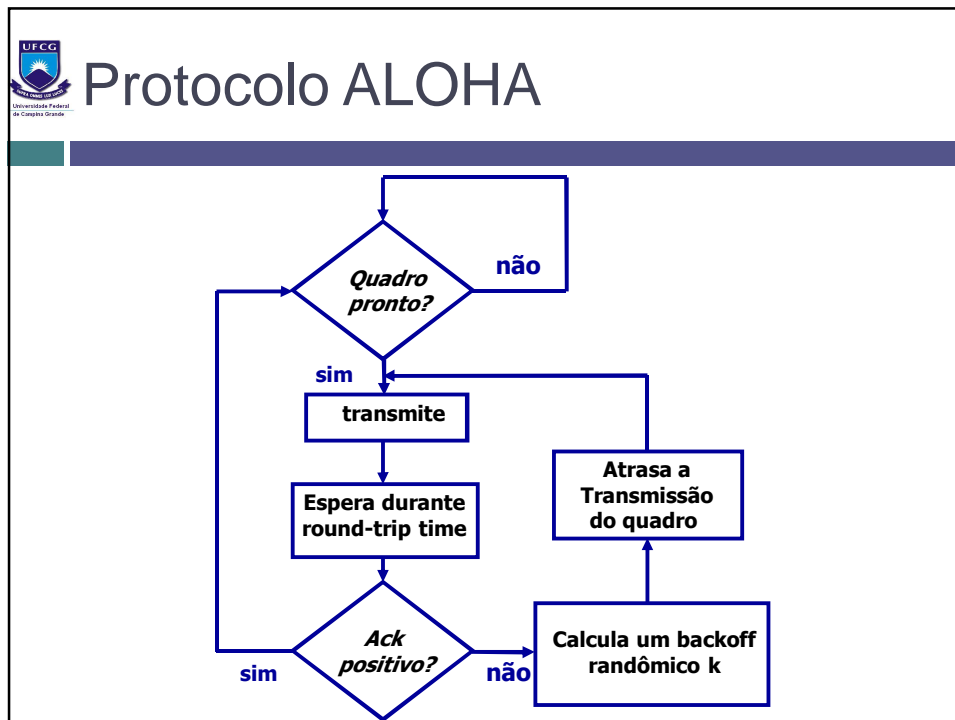
- Primeiro protocolo para canais de múltiplo acesso
- Originalmente planejado para sistemas com base stations centrais ou comunicação via satélite
- Usava duas frequências
 - Uplink em 413 MHz e Downlink em 407 MHz, trabalhando a 9600 bps

O nó central retransmite todos os quadros recebidos



 **Protocolo ALOHA**

- A rede é composta por um grande número de **estações que transmitem dados em rajadas**
- Cada estação transmite um quadro assim que ele é recebido do usuário – **não existe qualquer coordenação com as outras estações para que apenas uma envie dados**
- O nó central retransmite todos os quadros (tenham sido recebidos corretamente ou não) através de seu canal de down link
- As estações decidem se devem retransmitir baseado nas informações que elas recebem do nó central



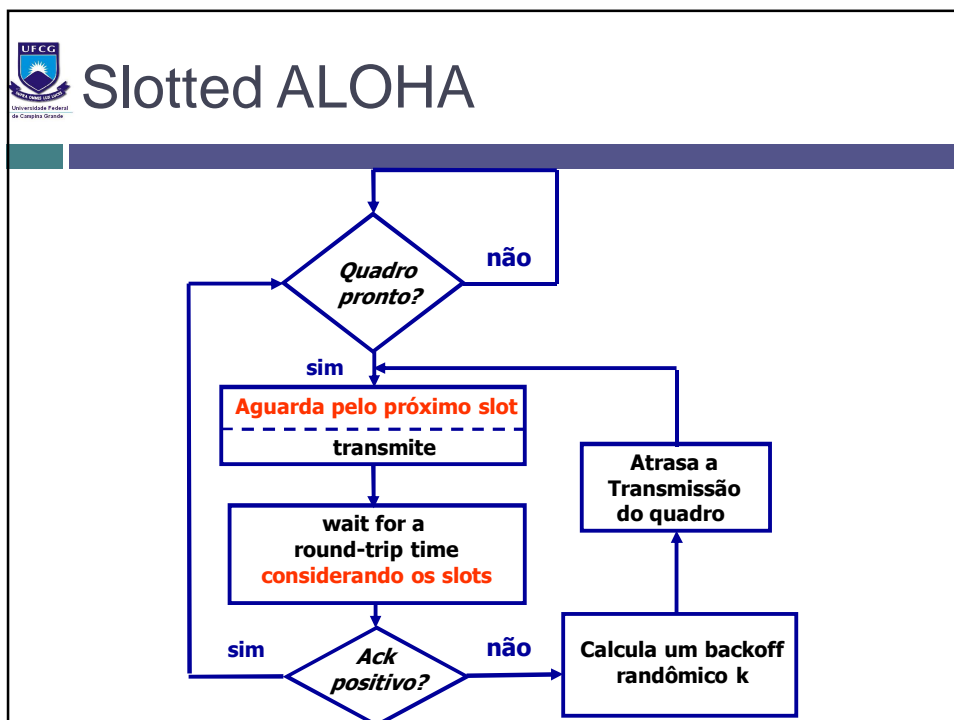


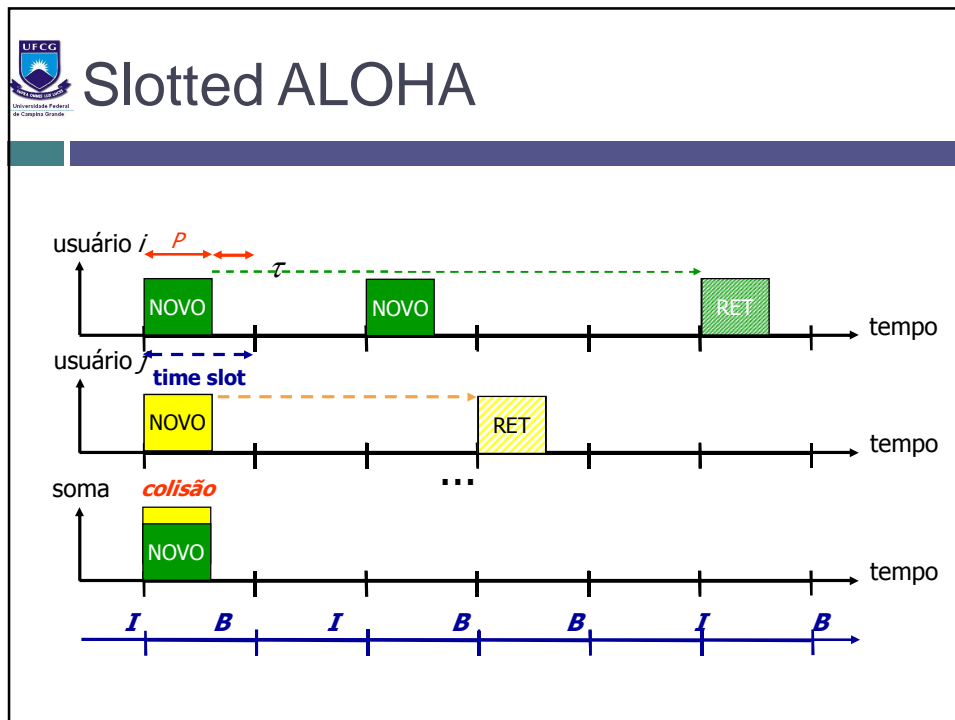
Slotted ALOHA


- O throughput do ALOHA pode ser melhorado se reduzirmos o período em que o quadro se encontra “vulnerável” a interferência de outros quadros
- O Slotted ALOHA trabalha em um canal que é particionado em slots de tempo
- As estações só podem transmitir quadros no início dos slots de tempo
- É necessário que seja realizada a sincronização das estações. Essa sincronização é realizada através da camada física e controlada pela estação central



Slotted ALOHA






 **CSMA: Carrier Sense Multiple Access**

CSMA: escuta antes de transmitir:

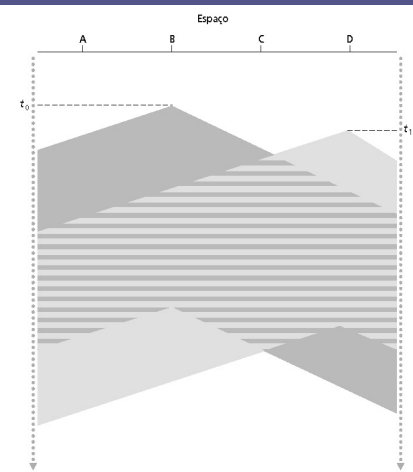
- Se o canal parece vazio: transmite o quadro
- Se o canal está ocupado, adia a transmissão
- Persistente:
 - Caso o meio estiver ocupado, o nó persiste escutando e quando o meio ficar livre, temos duas possibilidades:
 - 1-Persistente: transmite o quadro (i.e., p , probabilidade, igual a 1).
 - P-Persistente: com probabilidade P transmite e com probabilidade $1-P$ recua.
- Não Persistente:
 - Calcula tempo de recuo (i.e., *backoff*) e permanece inativo pelo tempo estipulado (tempo de recuo é decrementado somente enquanto o meio estiver LIVRE. Caso o meio ficar ocupado, congela contador até o meio ficar livre novamente. Necessariamente o meio estará livre quando o contador zerar!!!).

 **Colisões no CSMA**

Colisões podem ocorrer:
o atraso de propagação implica que dois nós quaisquer podem não ouvir as transmissões do outro

Colisão:
todo o tempo de transmissão do quadro é desperdiçado

Note:
papel da distância e do atraso de propagação na determinação da probabilidade de colisão.

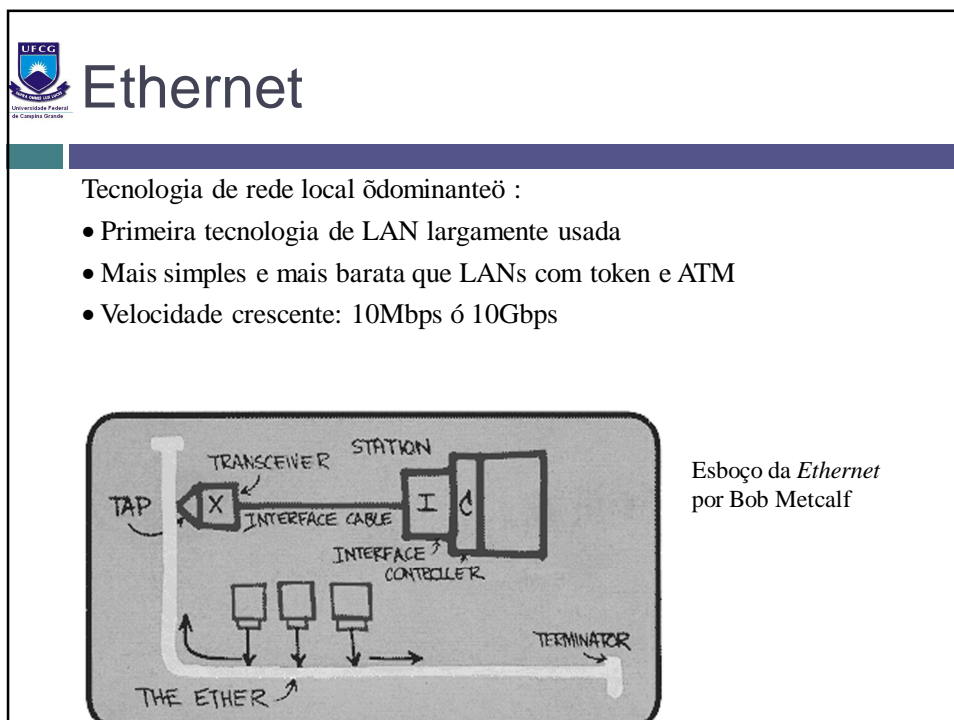
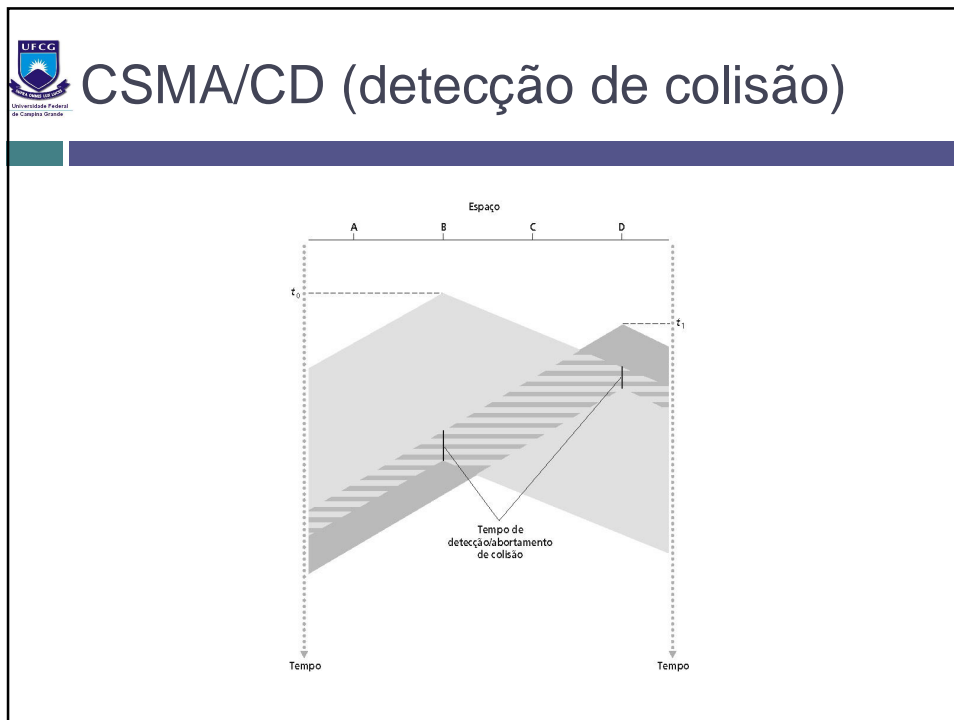



arranjo espacial dos nós na rede

 **CSMA/CD (detecção de colisão)**


CSMA/CD: detecção de portadora como no CSMA

- Colisões **detectadas** num tempo mais curto (proporcional ao tempo de propagação).
- Transmissões com colisões são interrompidas, reduzindo o desperdício do canal
- Detecção de colisão:
 - Fácil em LANs cabeadas: medição da intensidade do sinal, comparação dos sinais transmitidos e recebidos
 - Difícil em LANs sem fio: receptor desligado enquanto transmitindo (e além do mais, colisão acontece no receptor, não no transmissor!!!!).
 - Analogia humana: o ôbom de papoê educado




Ethernet

- **Sem conexão:** não ocorre conexão entre o adaptador transmissor e o receptor.
- **Não confiável:** adaptador receptor não envia ACKs ou nacks para o adaptador transmissor
 - O fluxo de datagramas que passa para a camada de rede pode deixar lacunas
 - Lacunas serão preenchidas se a aplicação estiver usando TCP.
 - Caso contrário, a aplicação verá as lacunas
- Ethernet se baseia no mecanismo do CSMA/CD
 - Sem slots de tempo
 - Adaptador não transmite se ele detectar algum outro adaptador transmitindo, isto é, **carrier sense**
 - O adaptador transmissor aborta quando detecta outro adaptador transmitindo, isto é, **collision detection**
 - Antes de tentar uma retransmissão, o adaptador espera um período aleatório, isto é, **random access**



Quadro Ethernet

Adaptador do transmissor encapsula o datagrama IP (ou outro pacote de protocolo da camada de rede) num **quadro Ethernet**

Preâmbulo	Endereço de destino	Endereço da origem	Tipo	Dados	CRC
-----------	---------------------	--------------------	------	-------	-----


Preâmbulo:

- 7 bytes com padrão 10101010 seguido por um byte com padrão 10101011
- **usado para sincronizar as taxas de relógio do transmissor e do receptor**


Quadro Ethernet

- **Endereços:** 6 Bytes
 - Se o adaptador recebe um quadro com endereço de destino coincidente, ou com endereço de broadcast (ex., pacote ARP), ele passa o dado no quadro para o protocolo da camada de rede
- **Tipo:** indica o protocolo da camada superior; geralmente é o protocolo IP, mas outros podem ser suportados, tais como Novell IPX e AppleTalk)
- **CRC:** verificado no receptor; se um erro é detectado, o quadro é simplesmente descartado

Preâmbulo	Endereço de destino	Endereço da origem	Tipo	Dados	CRC
-----------	---------------------	--------------------	------	-------	-----


CSMA/CD do Ethernet

1. Adaptador recebe um datagrama da camada de rede e cria um quadro.
2. Se o adaptador detecta um canal livre, ele começa a transmitir o quadro. Se ele detecta o canal ocupado, espera até ele ficar livre e então transmite.
3. Se o adaptador transmite o quadro todo sem detectar outra transmissão, sua missão com esse quadro está cumprida!
4. Se o adaptador detecta outra transmissão enquanto transmite, ele aborta e envia um *jam signal*
5. Após abortar, o adaptador entra em **exponential backoff**: após a m-ésima colisão, o adaptador escolhe um K aleatório de $\{0, 1, 2, \dots, 2^m - 1\}$. O adaptador espera K 512 tempos de bit e retorna ao passo 2.



CSMA/CD do Ethernet

Jam signal: garante que todos os outros transmissores estão cientes da colisão; 48 bits;

Bit time: .1 microseg para Ethernet de 10 Mbps;
para K=1023, o tempo de espera é cerca de 50 mseg

Exponential backoff:

- **Objetivo:** adaptar tentativas de retransmissão para carga atual da rede
 - Carga pesada: espera aleatória será mais longa
- Primeira colisão: escolha K entre {0,1}; espera é
K x 512 tempos de transmissão de bit
- Após a segunda colisão: escolha K entre {0, 1, 2, 3}
- Após 10 ou mais colisões, escolha K entre {0, 1, 2, 3, 4, ..., 1023}

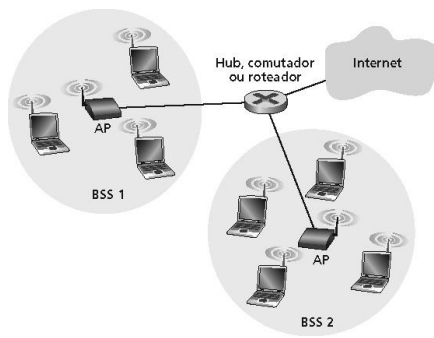


IEEE 802.11 LAN sem fio

- **802.11b**
 - 2,4 GHz faixa de rádio sem licença
 - Até 11 Mbps
 - *Direct sequence spread spectrum* (DSSS) na camada física
 - Todos os hospedeiros usam a mesma sequência de código
 - Largamente empregado, usando estações-base (pontos de acesso)
- **802.11a**
 - Faixa 5 GHz
 - Até 54 Mbps
- **802.11g**
 - Faixa 2,4 GHz
 - Até 54 Mbps
- Todos usam CSMA/CA para acesso múltiplo
- Todos têm estações-base e versão para redes ad hoc



802.11 arquitetura de LAN

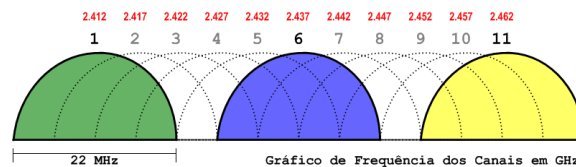


- Hospedeiro sem fio se comunica com a estação-base
- Estação-base = ponto de acesso (AP)
- Basic Service Set (BSS) (ou “célula”) no modo infra-estrutura contém:
 - Hospedeiros sem fio
 - Ponto de acesso (AP): estação-base
 - Modo ad hoc: somente hospedeiros




802.11: Canais, associação

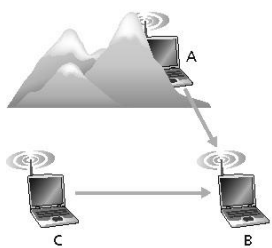
- 802.11b: o espectro de 2,4 GHz-2,485 GHz é dividido em 11 canais de diferentes frequências
- O administrador escolhe a frequência para o AP
- Possível interferência: canal pode ser o mesmo que aquele escolhido por um AP vizinho!



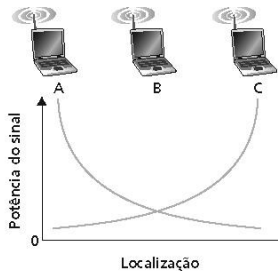
- **Hospedeiro:** deve **associar-se** com um AP
 - Percorre (scanning) canais, buscando quadros *beacon* que contêm o nome do AP (SSID) e o endereço MAC
 - Escolhe um AP para se associar
 - Pode realizar autenticação

 **IEEE 802.11: acesso múltiplo**


- Evita colisões: 2 ou mais nós transmitindo ao mesmo tempo
- 802.11: CSMA - escuta antes de transmitir
 - Não colide com transmissões em curso de outros nós
- 802.11: não faz detecção de colisão!
 - Difícil de receber (sentir as colisões) quando transmitindo devido ao sinal recebido ser muito fraco (desvanecimento)
 - Pode não perceber as colisões: colisão acontece no receptor!!!h
 - Meta: **evitar colisões**: CSMA/C(collision)A(voidance)



a.



b.

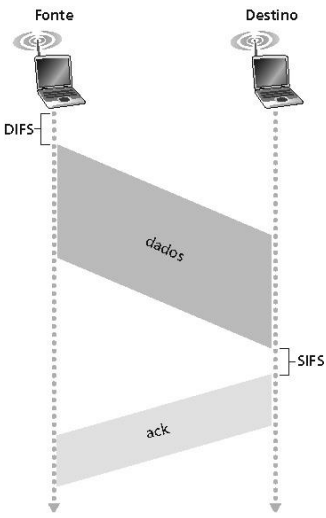
 **IEEE 802.11 Protocolo MAC: CSMA/CA**


Transmissor 802.11

1. Se o canal é percebido quieto (idle) por DIFS então
 - Transmite o quadro inteiro (sem CD).
2. Se o canal é percebido ocupado, então
 - Inicia um tempo de *backoff* aleatório
 - Temporizador decremente contador enquanto o canal estiver **ocioso**
 - Transmite quando temporizador expira
 - Se não recebe ACK, aumenta o intervalo de recuo (*backoff*) aleatório; repete passo 2.

Receptor 802.11

- Se o quadro é recebido corretamente retorna ACK depois de SIFS (ACK é necessário devido ao problema do terminal oculto)






Evitando colisões

Idéia: permite o transmissor “reservar” o canal em vez de acessar aleatoriamente ao enviar quadros de dados: evita colisões de quadros grandes

- Transmissor envia primeiro um pequeno quadro chamado **request to send** (RTS) à estação-base usando CSMA
 - RTSs podem ainda colidir uns com os outros, mas são pequenos
- BS envia em **broadcast clear to send** CTS em resposta ao RTS
 - CTS é ouvido por todos os nós
 - Transmissor envia o quadro de dados
 - Outras estações deferem suas transmissões
 - Após recepção do quadro, destinatário envia confirmação (i.e., ACK).



Quadro 802.11: endereçamento

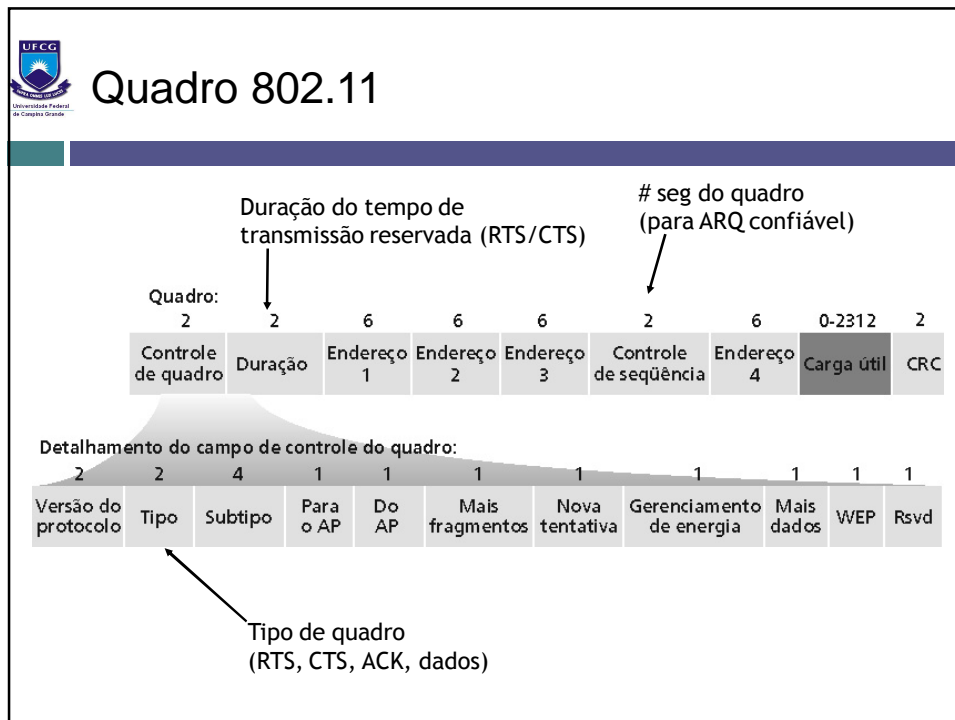
Quadro:	2	2	6	6	6	2	6	0-2312	2
	Controle de quadro	Duração	Endereço 1	Endereço 2	Endereço 3	Controle de sequência	Endereço 4	Carga útil	CRC


Endereço 1: endereço MAC do Hospedeiro ou Access Point (AP) que deve receber o quadro

Endereço 2: endereço MAC do hospedeiro sem fio ou AP transmitindo este quadro

Endereço 3: endereço MAC da interface do roteador à qual o AP é ligado

Endereço 4: usado apenas no modo ad hoc



 **Protocolos MAC com passagem de permissão**


Protocolos MAC com particionamento de canais:

- Compartilham o canal eficientemente quando a carga é alta e bem distribuída
- Ineficiente nas cargas baixas: atraso no acesso ao canal. A estação consegue uma banda de $1/N$ da capacidade do canal, mesmo que haja apenas 1 nó ativo!

Protocolos MAC de acesso aleatório

- Eficiente nas cargas baixas: um único nó pode usar todo o canal
- Cargas altas: excesso de colisões

Protocolos de passagem de permissão
Buscam o melhor dos dois mundos!


 **Protocolos MAC com passagem de permissão**

Polling:


- Nó mestre ãconvidaã os escravos a transmitirem um de cada vez
- Problemas:
 - Polling overhead
 - Latência
 - Ponto único de falha (mestre)

Token passing:

- Controla um **token** passado de um nó a outro seqüencialmente.
- Mensagem *token*
- Problemas:
 - *Token overhead*
 - Latência
 - Ponto único de falha (token)


 **Sumário dos protocolos MAC**

- Como se faz com um canal compartilhado?
 - Particionamento de canal, no tempo, por frequência ou por código
 - Particionamento aleatório (dinâmico),
 - ALOHA, Slotted-ALOHA, CSMA, CSMA/CD
 - Detecção de portadora: fácil em alguns meios físicos (e.g., cabos) e difícil em outros (e.g., *wireless*)
 - CSMA/CD usado na rede *Ethernet*
 - CSMA/CA (Collision Avoidance) usado em 802.11 mode *ad hoc*
- Passagem de permissão
 - *Polling* a partir de um nó central, passagem de token



Universidade Federal
de Campina Grande

Endereçamento da camada de enlace



Universidade Federal
de Campina Grande

Endereços de LAN e ARP

Endereços IP de 32-bit:

- Endereços da *camada de rede*
- Usados para levar o datagrama até a rede de destino (lembre-se da definição de rede IP)

Endereço de LAN (ou MAC ou físico):

- Usado para levar o datagrama de uma interface física a outra fisicamente conectada com a primeira (isto é, na mesma rede)
- Endereços MAC com 48 bits (na maioria das LANs) gravados na memória fixa (ROM) do adaptador de rede



Endereços de LAN e ARP

- A alocação de endereços MAC é administrada pelo IEEE
- O fabricante compra porções do espaço de endereço MAC (para assegurar a unicidade)
- Analogia:
 - (a) endereço MAC: semelhante ao número do RG
 - (b) endereço IP: semelhante a um endereço postal
- Endereçamento MAC é ôflatô => portabilidade
 - É possível mover uma placa de LAN de uma rede para outra sem reconfiguração de endereço MAC
- Endereçamento IP ôhierárquicoô => NÃO portátil
 - Depende da rede na qual se está ligado



ARP: Address Resolution Protocol (Protocolo de resolução de endereços)

**Questão: como determinar o endereço MAC de B
dado o endereço IP de B?**

- Cada nó IP (hospedeiro, roteador) numa LAN tem um módulo e uma tabela **ARP**
- Tabela ARP: mapeamento de endereços IP/MAC para alguns nós da LAN
< endereço IP; endereço MAC; TTL>
- < IP address; MAC address; TTL>
- TTL (Time To Live): tempo depois do qual o mapeamento de endereços será esquecido (tipicamente 20 min)



ARP na mesma rede local

- A quer enviar um datagrama para B, e o endereço MAC de B não está na tabela ARP de A
- A faz **broadcast** de pacote de consulta ARP, contendo o endereço IP de B
 - end. MAC de destino = FF-FF-FF-FF-FF-FF
 - todas as máquinas na LAN recebem a consulta ARP
- B recebe o pacote ARP, responde para A com seu endereço MAC (de B).
 - Quadro enviado para o end. MAC de A (unicast)
- A faz um cache (salva) o par de endereços IP para MAC em sua tabela ARP até que a informação se torne antiga (expirada) soft state: informação que expira (é descartada) sem atualização
- ARP é *plug-and-play*:
 - Nós criamos nossas tabelas ARP sem intervenção do administrador da rede



Roteamento e ARP

objetivo: **envia datagrama de A para B via R**
supõe que A conhece o endereço IP de B

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



- A cria o pacote IP com origem A, destino B
- A usa ARP para obter o endereço de camada física de R correspondente ao endereço IP 111.111.111.110
- A cria um quadro Ethernet com o endereço físico de R como destino, o quadro Ethernet contém o datagrama IP de A para B
- A camada de enlace de A envia o quadro Ethernet
- A camada de enlace de R recebe o quadro Ethernet
- R remove o datagrama IP do quadro Ethernet, verifica que ele se destina a B
- R usa ARP para obter o endereço físico de B
- R cria quadro contendo um datagrama de A para B e envia para B