Teleprocessamento e Redes
Universidade Católica do Salvador

Aula 10 - Alocação Dinâmica do Canal *Ethernet*, CSMA/CD e IEEE802.3

Objetivo :

Estudaremos nesta aula os métodos de alocação dinâmica de canal utilizados nos sistemas por difusão. Concluiremos a aula analisando detalhadamente o padrão utilizado nas redes *ethernet*.

Roteiro da Aula :

O compartilhamento do meio físico por mais de uma estação implica em duas soluções possíveis : a alocação fixa, onde cada estação tem determinada fração de tempo para usar o canal e a alocação dinâmica, onde são criadas regras de convivência que permitem o compartilhamento do canal conforme a necessidade de transmissão de cada uma delas.

Nesta aula vamos iniciar o estudo destas técnicas que estabelecem a cada instante qual é a estação que tem o direito de utilizar o canal compartilhado por todas. Para começar, vamos conhecer primeiro algumas premissas que devem ser consideradas na alocação dinâmica de canal :

1) **Modelo da estação** : temos "n" estações independentes (definição de rede), onde a probabilidade de um quadro ser gerado é $\lambda.\Delta t$, onde λ é uma constante (taxa de chegada de novos quadros). Gerado um quadro, a estação não faz mais nada até que o quadro tenha sido transmitido com sucesso.

É importante lembrar que esta premissa considera um modelo bastante simplificado, que simplifica os cálculos. Em sistemas mais modernos, é comum as estações continuarem operando mesmo após o início da transmissão de um quadro.

2) **Premissa de canal único** : todas as estações transmitem e recebem através deste canal. Em termos de *hardware*, as estações são equivalentes, embora um *software* de protocolo possa determinar prioridades.

Novamente é importante observar que, no mundo real, as estações têm diferentes perfis e capacidades de gerar quadros. De qualquer sorte, a premissa do canal único não é afetada, já que não existe qualquer outra maneira da estação se comunicar que não seja através do canal compartilhado por todas.

3) **Premissa da colisão** : se duas ou mais estações transmitem ao mesmo tempo no canal, o sinal é deturbado, o que evidencia a ocorrência de uma colisão. Todas as estações conseguem detectar colisões. Um quadro que sofre colisões terá que ser re-transmitido, mesmo que esta colisão ocorra em apenas um bit dos quadros transmitidos. Não existem outros erros que não os gerados por colisões.

4) **Tempo contínuo X Tempo dividido** : os métodos de alocação dinâmica de canal podem operar de duas formas bem diferentes. Na alocação do tipo **tempo contínuo**, a transmissão de um quadro pode se iniciar a qualquer momento. Não existe nenhum relógio ou dispositivo mestre dividindo o tempo em intervalos discretos ou permitindo o início de uma transmissão. Na alocação do tipo **tempo dividido**, o tempo é dividido em *slots* (aberturas). A transmissão de um quadro sempre começa no início do *slot*. Este pode conter 0 quadros (*slot* livre), 1 quadro (transmissão bem sucedida) ou mais de um quadro (colisão).

5) **Detecção de portadora** : os métodos também se diferenciam pela capacidade de detectar a presença de colisões. Esta detecção é feita através da presença de um sinal elétrico particular no cabo. Alguns métodos permitem e outros não a detecção de colisões. Caso não detecte colisões, o método só poderá avaliar a existência de colisões analisando os erros

obtidos em um eventual sinal recebido. Com a detecção, a transmissão pode ser interrompida imediatamente após a detecção de uma colisão.

O modelo ALOHA : este modelo foi criado na universidade do Havaí na década de 70, e basicamente consiste em deixar qualquer estação transmitir sempre que tiver dados. Desta forma, obviamente ocorrerão colisões com destruição de quadros.

Os adaptadores usados no modelo ALOHA (e também em muitos outros) possuem uma propriedade de *feed-back* que garante que os mesmos são capazes de detectar a presença de erros de comunicação. Esta capacidade normalmente apresenta respostas bastante rápidas em uma rede local. No caso de ambientes mais lentos, o *feed-back* acontece após um tempo mais prolongado (270 ms no caso de uma transmissão por satélite). Como o modelo ALOHA foi originalmente implantado usando-se transmissões por rádio na Universidade do Havaí, esta consideração é importante.

No caso de detecção de falhas, o transmissor aguarda um tempo aleatório e re-transmite. Isto garante que não ocorram múltiplas colisões sincronizadas, o que eliminaria o carácter aleatório das colisões, diminuindo muito a performance do ambiente. Diz-se que a estação encontra-se em estado de **contenção** durante estes períodos, pois, apesar de possuir dados para transmitir, permanece aguardando alguns instantes antes de efetivar o processo. Sistemas sujeitos a este tipo de espera são também conhecidos como sistemas de **contenção**.

Outra característica importante do ALOHA é que todos os quadros possuem tamanho fixo. Isto aumenta o desempenho do ambiente. Para se chegar a esta conclusão, devemos lembrar que cada estação deve transmitir, em média, um número entre 0 e 1 quadros por unidade de tempo, onde a unidade de tempo é o tempo necessário para transmissão de um único quadro (que tem tamanho fixo). Se isto não fosse verdade, praticamente todo quadro sofreria colisão, já que temos que considerar que ocorrerá perda completa de um quadro mesmo que a colisão ocorra em apenas um bit do mesmo. Isto aconteceria se uma estação iniciasse a transmissão no instante da propagação do último bit da transmissão de uma outra estação. Em outras palavras, temos que considerar que não existem perdas parciais no modelo ALOHA.

Calculando-se matematicamente a eficiência obtida do modelo ALOHA, encontramos um número em torno de 18%. Esta ineficiência, embora aparentemente elevada, é aceitável considerando-se a simplicidade do método de acesso ao meio físico. Para reduzi-la, são necessárias alterações no método.

A primeira alteração, conhecida como ALOHA com aberturas, divide o tempo em intervalos discretos, chamados de *slots*. Desta forma abandonamos o tempo contínuo, reduzindo com isto a probabilidade de colisões em aproximadamente 50%, o que implica numa eficiência de canal próxima de 37%.

Em ambos os métodos, no entanto, existe uma dependência probabilística em relação ao tráfego, que pode aumentar dramaticamente o número de colisões em ambientes de alta utilização.

O método CSMA 1-persistente : neste caso, quando uma estação tem algo a transmitir, ela escuta o canal. Se este não estiver livre, ela aguarda até que desocupe. Imediatamente após a detecção do canal livre, ela transmite. No caso de colisão, a estação aguarda um tempo aleatório e tenta de novo. O método chama-se 1-persistente porque existe probabilidade 1 de transmitir caso o canal esteja livre e existam dados a serem transmitidos.

O desempenho do protocolo está associado ao retardo na propagação da informação pelo meio físico, já que o mesmo aumenta a probabilidade de colisões. Quanto maior for o retardo, pior o protocolo. Se o retardo for zero, teremos a menor probabilidade, mas esta não é zero !

O método CSMA não persistente : neste caso, a estação não permanece aguardando a liberação do meio para transmissão imediata. Na verdade o processo ocorre como no caso da detecção de colisão - a estação aguarda um tempo aleatório e então repete o processo. Isto implica em melhor utilização do canal e a possibilidade de retardos maiores.

O método CSMA p-persistente : utiliza canais com aberturas. Se o canal estiver desocupado, a estação inicia com probabilidade p. Com probabilidade $q = 1 - p$, ela espera até a próxima abertura.

O método CSMA/CD : normalmente utilizado em redes locais padrão IEEE 802.3, este método será melhor detalhado a seguir, com todo o processo de transmissão e recepção. A novidade é a capacidade de detectar a presença de colisões durante a transmissão, permitindo a interrupção do processo antes do final do quadro. Isto aumenta bastante a eficiência do uso do canal. Nos métodos vistos até agora, o quadro era completamente transmitido, mesmo que a colisão ocorresse no início do quadro. Isto implicava na perda de todo um quadro.

Na verdade, o padrão IEEE 802.3 teve sua origem no ALOHA. Com o acréscimo da detecção de portadora e de colisões, a XEROX criou uma pequena rede de 2,94 Mbps, batizada de rede *Ethernet*. O sistema funcionou tão bem que a XEROX, a INTEL e a DEC redigiram um padrão para uma rede *Ethernet* a 10 Mbps. Este padrão foi a base do padrão IEEE 802.3. Na verdade, o padrão IEEE 802.3 é diferente do *Ethernet* da XEROX, INTEL e DEC, pois trata de uma família de padrões CSMA/CD 1-persistentes rodando a taxas entre 1 e 10 Mbps sobre diversos meios. De todos eles, o padrão que se tornou o líder de mercado a nível mundial foi o padrão em 10 Mbps sobre cabos de 50 ohms em banda base, ou seja, sem modulação. Este padrão utiliza a codificação Manchester pura, que será vista em aulas futuras, com sinais de +0,85 V e -0,85V, resultando em uma componente de tensão nula.

Todos estes métodos resultam em eficiências de canal variáveis com o tráfego. Para melhor entender o comportamento de cada método, mostramos abaixo um gráfico ilustrativo das tendências de eficiência do canal.

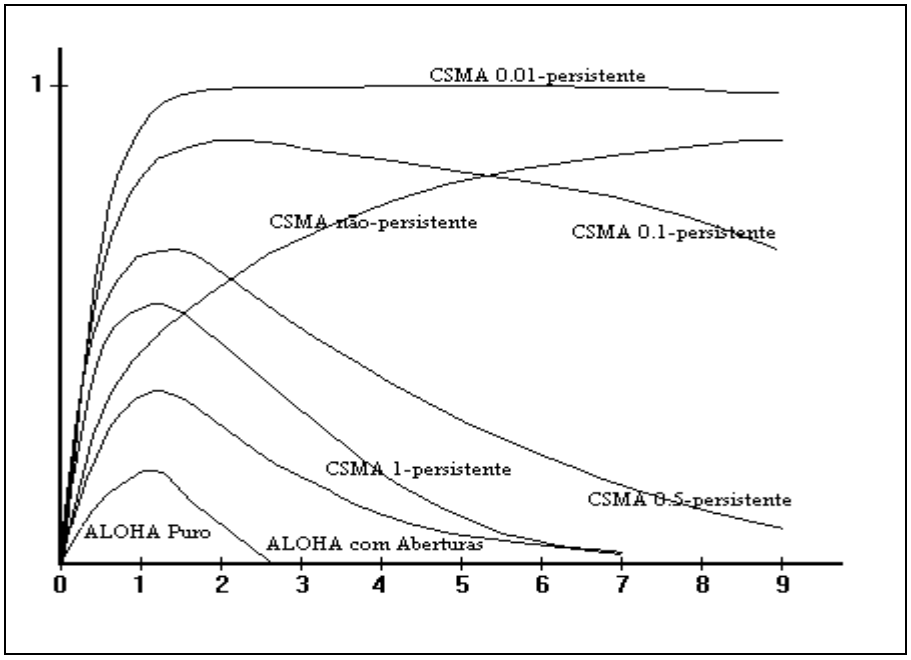

Figura 10.1 - Utilização do Canal X Carga em diversos protocolos com alocação dinâmica.

CSMA/CD - Operação Detalhada :

TRANSMISSÃO

Em função da grande quantidade de pontos no mundo inteiro baseados na tecnologia *Ethernet*, vamos fazer um detalhamento mais cuidadoso do método de acesso CSMA/CD e suas características de operação. Tal como já havíamos visto acima, são estas as características básicas do protocolo para transmissão :

a) A estação deve escutar o barramento antes de iniciar uma transmissão.

b) Caso exista tráfego, a estação deve aguardar até o final da transmissão. Caso contrário, ela pode iniciar a sua tranmissão imediatamente.

c) Se duas estações ou mais tem mensagens para enviar, pode ocorrer o fenômeno da colisão. A colisão interrompe a transmisão do quadro, liberando o meio físico para novas tentativas. O método baseia-se na determinação de um tempo máximo de verificação de colisões. Este tempo normalmente é igual ao tempo total de propagação do sinal pelo meio físico. Isto determina a impossibilidade de ocorrência posterior de colisões, já que todas as estações já têm conhecimento da existência de uma transmissão.

Representando o método de acesso por um algoritmo, teríamos :

```
if MESSAGE_READY=true then
    while BUS_BUSY=true
            defer
    end while
    while MESSAGE_BUFFER_FULL=true
            output MESSAGE_BUFFER
            if COLLISION=true then
                    wait RANDOM_TIME
            end if
    end while
end if
```

Imaginando a necessidade de transmissão, poderíamos detalhar o processo da seguinte forma :

a) Escutando antes de transmitir :

As estações ficam continuamente monitorando o meio físico, verificando a existência de um sinal (*Carrier On*). Este sinal é normalmente reconhecido pela presença de uma voltagem indicando a utilização do cabo. A ausência deste sinal indica que o cabo está livre e autoriza o início da transmissão, com monitoração do meio físico para detectar possíveis colisões.

b) Aguardando o final de outra transmissão para início :

Para evitar colisões, as estações devem esperar caso detectem a utilização do cabo. Quando cessa a utilização do meio físico, a estação aguarda apenas o tempo de estabilização do meio (9,6 μS para ambientes de 10MBps). Após este período, a estação inicia a sua transmissão, monitorando o meio físico para detectar possíveis colisões.

c) Detectando a ocorrência de uma colisão :

Se duas ou mais estações tem mensagens para enviar e estão separadas por distâncias significativas de cabo, uma estação pode começar a transmitir sem saber da existência do sinal de outra que tenha iniciado o processo anteriormente. Isto ocorre porque, apesar de rápida (no mínimo igual a 58% da velocidade da luz), a velocidade de propagação do sinal no meio físico provoca um atraso entre a transmissão de um sinal em um dos extremos do barramento e a recepção do mesmo no outro extremo. Desta forma, os sinais são colocados simultaneamente no meio físico e não poderão ser identificados pelas estações. O método de acesso exige então que a estação responsável pela transmissão monitore o barramento após o início da transmissão, para detectar possíveis "colisões".

Esta monitoração acontece durante os primeiros instantes da transmissão, enquanto o sinal não tiver tido tempo de atingir todas as estações da rede e retornar (*round trip time*). Em sistemas de 10MBps, este tempo equivale ao tempo para transmissão de 64 bytes. Eventuais colisões ocorridas após este período são conhecidas como *late collisions* e indicam possíveis falhas, geralmente na estrutura de cabos.

A detecção de uma colisão é realizada com base na medição do sinal no barramento. Caso este sinal seja igual ou maior que o sinal produzido por duas estações transmitindo simultaneamente, temos uma colisão.

Ao detectar uma colisão, uma das primeiras providências que é tomada pela estação transmissora é comunicar às outras estações esta condição especial para que elas evitem transmitir. Isto é feito através da transmissão de uma seqüência especial de bits (32 bytes para ambientes de 10 MBps), conhecida como *jam*. Somente depois do envio do *jam* a transmissão é encerrada. O comprimento do *jam* é exatamente a metade do *round trip time*, já que agora só é necessária a propagação em um sentido (no caso da avaliação de possíveis colisões, é necessário aguardar o retorno do sinal).

Após isto cada uma das estações que deseja transmitir deve esperar um tempo aleatório para tentar de novo. A escolha de um número aleatório vai garantir que será bastante improvável a ocorrência de uma nova colisão. Para tanto, é necessário que o algoritmo para esta escolha seja bem elaborado - ele será o um dos fatores determinantes do comportamento do método de acesso. O método utilizado para o cálculo é chamado de *Binary Exponential Backoff* e surgiu baseado em algumas considerações :

1. Estudos demonstram que o tráfego de mensagens em uma rede local *Ethernet* corretamente dimensionada tende a ser muito variável. Chegamos a ter longos períodos durantes os quais o tráfego é bastante reduzido. Durante estes períodos, escolher um período curto para tentar a re-transmissão resulta em um atraso menor no tratamento da colisão, e portanto, maior eficiência.

2. Da mesma forma, é interessante aumentar as possibilidades de escolha do tempo de espera para situações de alto tráfego, onde a probabilidade de ocorrência de uma novas colisões é maior.

3. A determinação do tempo de espera deve ser aleatória, para que estatisticamente tenhamos o melhor resultado sobre qualquer condição de tráfego.

Sendo assim, o *Binary Exponential Backoff* funciona dentro dos seguintes princípios :

1. Para garantir uma menor probabilidade de ocorrência de colisões, as estações envolvidas no processo determinam um valor aleatório inteiro, que será multiplicado por um período fixo de tempo conhecido como *time slot*.

2. O *time slot* é calculado com base no tempo necessário para a propagação de um sinal de uma estação no extremo do barramento até o outro extremo, seu retorno e detecção (este tempo foi visto anteriormente e é conhecido como *round trip time*). Isto significa que a escolha de números aleatórios diferentes por duas estações, mesmo que consecutivos, vai evitar a ocorrência de uma nova colisão. Em sistemas de 10 MBps, o *time-slot* equivale ao tempo de transmissão de 64 bytes.

3. De forma a aumentar o tempo médio de forma proporcional ao tráfego, a escolha do número aleatório ocorre dentro do intervalo [0,1] para a primeira colisão, dobrando-se o intervalo a cada nova colisão. Ex : [0,1,2,3] para a segunda colisão; [0,1,2,3,4,5,6,7] para a terceira etc. Este processo de duplicação ocorre até a décima tentativa. Após isto, mantém-se o intervalo até a décima-sexta tentativa.

4. Se forem executadas 16 tentativas consecutivas de transmissão com colisões, o processo de retransmissão é abandonado, sendo gerada uma mensagem de erro (normalmente chamada de *Excess Collision*).

O método acima tem a grande vantagem de apresentar um tempo de recuperação de colisões baixo para situações de tráfego reduzido e maior à medida que o tráfego aumenta. Ele foi utilizado nas primeiras redes *Ethernet*, estando agora padronizado pelo *IEEE 802.3 CSMA/CD local network standard*, ou simplesmente *IEEE 802.3*.

Após o sucesso em uma transmissão, uma estação pode realizar um teste de sua capacidade de detecção de colisões, para checar o funcionamento perfeito da mesma. Para tanto, basta habilitar a função SQE (*Signal Quality Error*). No caso da existência de repetidores, é importante desabilitar o SQE. Isto evita que eles considerem a presença de colisões na rede, o que provocaria a transmissão do *jam*, prejudicando o funcionamento do ambiente.

Para finalizar, é interessante abordar uma das maiores preocupações com o padrão *Ethernet* - a característica estatística do tempo de acesso. Ela permite a distribuição do acesso ao meio físico conforme a geração de tráfego. Isto torna o ambiente *Ethernet* extremamente eficaz em ambientes de carga heterogênea, razão do sucesso deste método no mercado.

Apresentamos a seguir um fluxograma do processo de transmissão em uma rede *Ethernet* para análise. A seqüência dos procedimentos apresentada deve ser devidamente acompanhada por uma seqüência de recepção nas estações destino, que será analisada em seguida.

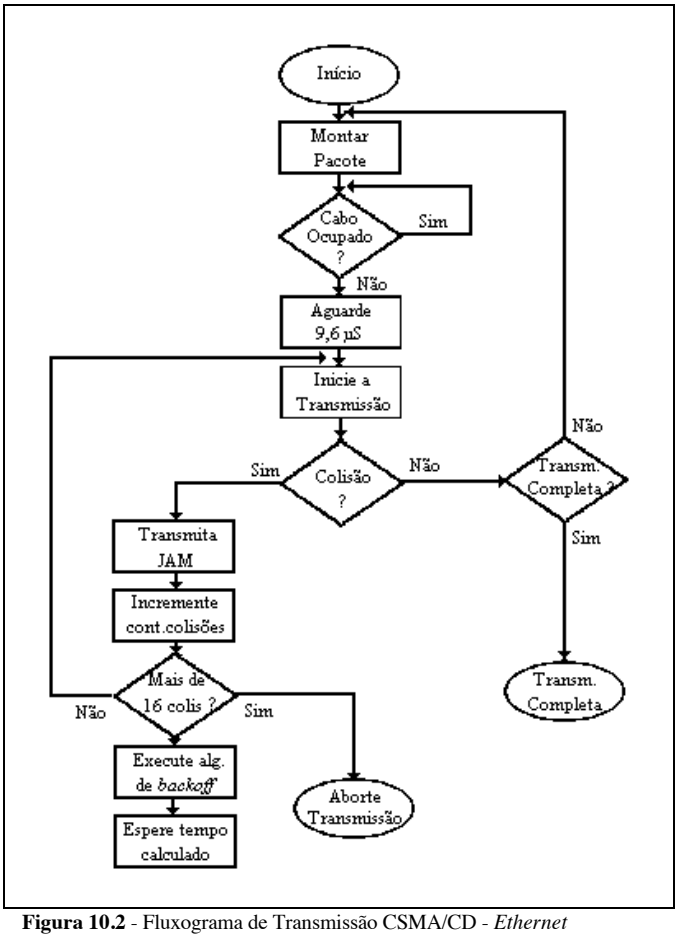

Figura 10.2 - Fluxograma de Transmissão CSMA/CD - *Ethernet*

RECEPÇÃO

Da mesma forma que analisamos o processo de transmissão de dados com método de acesso CSMA/CD em uma rede *Ethernet*, vamos agora atentar para a seqüência de operação para recepção :

a) Analise de todos os pacotes recebidos, dispensando os fragmentos. Este passo deve ser executado por todas as estações conectadas ao segmento, mesmo que estas não tenham nenhuma participação na comunicação em análise. Afinal, até este momento, não se sabe qual é a estação destino.

b) Checar o endereço de destino. Novamente é importante lembrar que este passo deve ser executado por todas as estações do segmento.

c) Após conferir o endereço de destino, checar a integridade do pacote de informações. Agora podemos limitar a execução deste passo à(s) estação(ões) de destino.

d) Processar o pacote. Este passo também só deve ser executado pela(s) estação(ões) destino.

Representando a recepção por um algoritmo, teríamos :

```
if MESSAGE_READY=true then
    if PACKET_LENGTH>64 then
            if DESTINATION_ADRESS=valid then
                    if PACKET_LENGHT>1518 then
                            abort
                    end if
                    if mod(BIT_COUNT,8)#0 then
                            abort
                    end if
                    if CRC_OK=false then
                            abort
                    end if
                    call RUN_RECEPTION
            end if
    end if
end if
```

Podemos detalhar o processo da seguinte forma :

a) Analisar os pacotes disponíveis :

Em redes *Ethernet*, todas as estações interligadas vêem cada pacote no momento em que ele passa no cabo, independente do fato deste ter ou não aquela estação como destino. Esta característica é conseqüência da topologia em barramento, onde o meio físico é compartilhado por todas as estações. Uma das principais preocupações, portanto, é analisar o endereço de destino do pacote.

Antes disto, porém, conhecendo a possibilidade de ocorrência de colisões, a estação receptora toma o cuidado de analisar apenas aqueles pacotes com tamanho superior a 64 bytes (para ambientes de 10MBps). Isto garante que o pacote foi transmitido sem a ocorrência de colisões. Como já vimos, no caso da ocorrência de colisões, a transmissão é interrompida antes do envio do 65° byte. Desta forma, pacotes com comprimento inferior a 64 bytes certamente foram gerados durante a ocorrência de colisões, e por isto mesmo devem ser desprezados.

b) Checagem do endereço de destino :

É importante lembrar que os pacotes podem conter não só endereços específicos, mas também endereços *broadcast* e *multicast*. Um endereço de *broadcast* indica que a mensagem contida no pacote deve ser analisada por **todas** as estações, enquanto que um endereço de *multicast* indica que a mensagem deverá ser analisada por mais de uma estação. A forma de indicar o endereço de destino é uma característica que se encontra descrita mais detalhadamente no final desta aula, quando conheceremos o formato de uma *frame Ethernet*.

c) Checar a integridade do pacote :

Após determinar que o pacote não foi gerado durante uma colisão e também que o mesmo tem um endereço de destino válido, o método determina a análise do pacote para checagem da integridade das informações. Desta forma podemos evitar pacotes defeituosos (as causas podem estar na estação transmissora ou no meio físico de comunicação). A checagem é feita através de 3 testes :

1) É verificado o comprimento total do pacote, que não pode ultrapassar 1518 bytes (tamanho máximo permitido pelo padrão *Ethernet*).

2) É verificada a possibilidade de falhas no comprimento do pacote em bits, que deve ser múltiplo de 8, já que todo pacote *Ethernet* é formado por octetos (*bytes*).

3) Testa-se o CRC (*Cyclical Redundancy Check*) do pacote. Eventuais trocas de valores de bits poderão ser detectadas neste último teste.

d) Processamento do pacote :

Garantindo todos estes testes, o pacote é encaminhado para as camadas superiores da máquina receptora, para que seja decodificado e utilizado. Eventuais erros ainda existentes (bastante improváveis), deverão ser detectados e corrigidos por camadas superiores do receptor, já que provavelmente foram gerados em camadas superiores do transmissor.

FORMATO DA *FRAME* IEEE 802.3

A estrutura de quadro (*frame*) definida pelo IEEE 802.3 em 1985 é mostrada na figura abaixo.

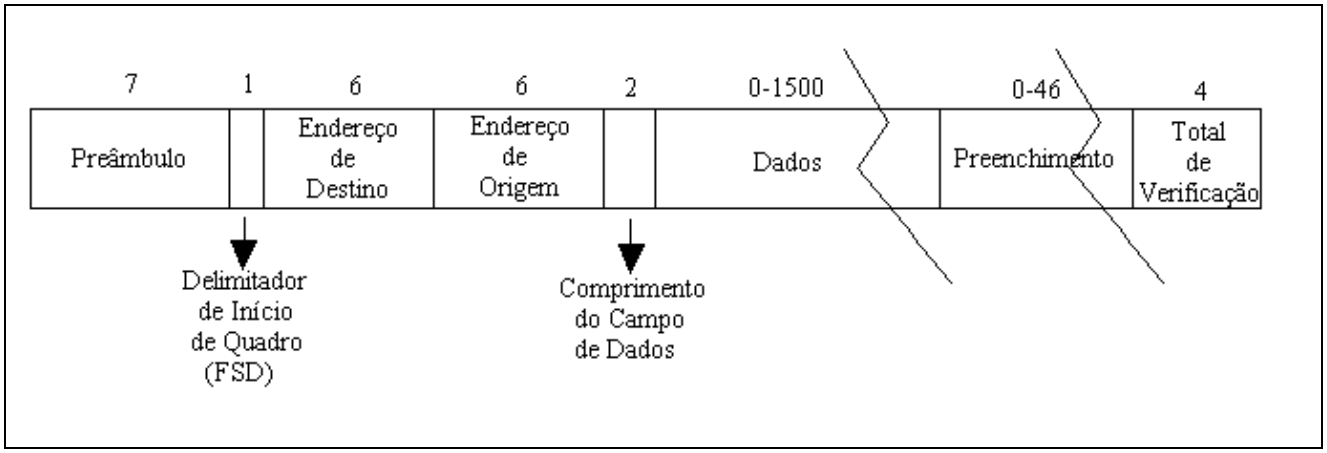

Figura 10.3 - Formato da *frame* (quadro) no IEEE 802.3

Como podemos ver na figura, o quadro começa com um campo de **preâmbulo** de 7 bytes. Cada byte deste campo possui o valor 10101010. Com a codificação *Manchester* deste valor por 7 bytes, temos uma onda quadrada de 10 Mhz durante 5,6 μS. Esta onde permite a sincronização do receptor com o transmissor. Isto é essencial se consideramos que temos diversas máquinas independentes conectadas ao barramento, o que pode implicar em diferenças de *clock* entre as mesmas.

Em seguida, temos o quadro **delimitador de início**, também conhecido como *frame start delimiter*. Seu conteúdo é o mesmo dos bytes do preâmbulo, com uma pequena diferença : os últimos dois bits iguais a 1 (10101011). Após este campo, começa o quadro efetivamente dito. Desta forma, devemos considerar que a partir do próximo quadro, não poderemos ter mais do que 1518 bytes.

Os próximos dois quadros indicam respectivamente o **endereço de destino e o de origem**. A presença do quadro de destino em primeiro lugar garante uma maior agilidade na avaliação do endereço de destino da mensagem, que é uma das primeiras atividades na interpretação de *frames* recebidas pelas estações. Isto se torna ainda mais importante quando consideramos que esta atividade será realizada simultaneamento por todos os pontos interligados. Os endereços de destino e origem têm comprimento de 6 bytes.

No caso do endereço de destino, costuma-se padronizar que o bit de mais alta ordem seja sempre zero. Caso este primeiro bit seja um, teremos uma indicação de endereço de *multicast*. No caso de todos os bits do endereço de destino serem iguais a 1, temos um endereço de *broadcast*, o que garante que todas as estações interpretarão aquela informação. O bit número 46 (posterior ao bit de mais alta ordem) é utilizado para indicar endereços locais, que não tem sentido fora dos limites da rede local. Os outros 46 bits são usados para indicação de endereços globais. A imensa quantidade de possibilidades de endereço (2^{46}), garante que não existirão dois endereços globais iguais em todo o mundo. A distribuição de endeços globais entre os fabricantes é uma atribuição do IEEE. Nas próximas aulas, veremos como a camada superior de rede é capaz de ajudar na localização de determinado endereço global.

O campo de **comprimento** indica o tamanho do campo de dados. Estes dois bytes podem indicar quantidades entre 0 e 1500. Porém, devemos avaliar um possível problema : se usarmos um comprimento de dados igual a zero, teremos um comprimento total de quadro inferior a 64 bytes. Este quadro será confundido com quadros gerados por colisões. Desta forma, torna-se essencial a presença de um campo de **preenchimento**, que deve existir sempre que o comprimento total do campo de dados for inferior a 46 bytes. Desta forma, ele pode complementar o quadro garantindo um comprimento mínimo de 64 bytes para os quadros válidos.

Outro motivo para a existência do campo de comprimento reside na utilização da transmissão em banda base. Como não existe modulação, os únicos sinais possíveis são os valores binários de 0 e 1. Sendo assim, não há como detectar o final do quadro através de um sinal especial. O uso do campo de comprimento informa para a estação receptora qual é o comprimento esperado para o quadro, garantindo que a mesma possa interpretá-lo sem falhas.

Par finalizar, temos o campo com o **total de verificação**. Através de uma técnica especial de checagem de erros conhecida como CRC (*Ciclic Redundancy Check*), pode-se garantir a integridade dos dados transmitidos. Isto ocorre porque a falha em um ou mais bits do quadro provavelmente provocará a mudança dos 32 bits do CRC, garantindo que um cálculo realizado no receptor seja suficiente para detectar possíveis quadros defeituosos.

Exercícios :

1) Associe as afirmações abaixo :

( ) Apesar de reduzir a eficiência, reduz a probabilidade de ocorrência de múltiplas colisões consecutivas.
( ) Qualquer estação pode transmitir a qualquer momento, desde que existam dados a serem transmitidos.
( ) Determinou, após o seu aprimoramento, o surgimento das redes *Ethernet*.
( ) A detecção de uma colisão no início da transmissão do quadro permite a interrupção da transmissão.
( ) Garante o sincronismo entre transmissor e receptor antes da recepção do quadro.
( ) Permite adequar o tempo de espera ao tráfego, garantindo tempos menores quando houver menor tráfego.
( ) Ajusta-se automaticamente, garantindo um comprimento mínimo de 64 bytes para os quadros.

(a) ALOHA
(b) *jam*
(c) IEEE 802.3
(d) Contenção
(e) *Binary Exponential Backoff*
(f) Preâmbulo
(g) N.R.A.

2) Indique abaixo a ordem correta dos eventos durante a recepção de um quadro por uma estação IEEE 802.3 (podem existir eventos intermediários faltando) :

( ) Verificação do comprimento do pacote em bits (este deve ser múltiplo de oito).
( ) Verificação do endereço de destino.
( ) Teste de integridade do campo de CRC.
( ) Processamento do quadro.
( ) Verificação do comprimento do pacote em bytes (este deve ser igual ou superior a 64 bytes).

3) Ao realizar um projeto de comunicação de dados, se decidiu implementar uma rede IEEE 802.3 para interligação de um grupo de servidores de arquivo com distribuição homogênea de performance e altos índices de tráfego. Esta escolha foi correta ? Porque ?

_____

_____

_____

_____

_____

_____

_____