Capítulo 3

A CAMADA DE ENLACE

A camada de enlace é responsável pela detecção e recuperação de erros ocorridos na camada física, oferecendo às camadas superiores um transporte de dados mais confiável. O *enlace* é uma conexão virtual por onde fluem os quadros. Esta conexão implementa protocolos simples de detecção e recuperação de erros, por exemplo detecção através de *checksum* e recuperação por retransmissão.

Em redes de longa distância e metropolitanas o enlace se dá entre roteadores. A rota física por onde os quadros fluem é exclusiva dos roteadores por ela conectados (o meio físico é decomposto em segmentos que conectam dois, e somente dois, roteadores). Ao decidir transmitir um quadro, um roteador simplesmente seleciona o segmento conectando ao destinatário e invoca os serviços da camada física referente ao segmento.

Em redes locais, o meio físico é compartilhado por todos os hosts. A transmissão de um quadro em redes locais requer antes um procedimento de acesso ao meio. Este procedimento é denominado MAC (Medium Access Control) e varia em complexidade em função da topologia e demais características da rede.

Dada a importância do controle de acesso ao meio em redes locais, é comum reservar uma subcamada no modelo de rede exclusiva para tal. A *subcamada de acesso ao meio* é parte da camada de enlace e situa-se na interface desta com a camada física. O restante da camada de enlace denomina-se Controle de Enlace Lógico (LLC: Logical Link Control) e podemos considerá-la como uma segunda subcamada, acima da subcamada de acesso ao meio.

3.1 A Subcamada de Acesso ao Meio (MAC)

A subcamada de acesso ao meio implementa uma disciplina (seguida à risca por todos os hosts) de acesso ao meio físico. As mais difundidas técnicas de controle de acesso ao meio são as baseadas em *acesso aleatório (ou de contenção)* e *passagem de permissão*.

3.1.1 Técnicas de Acesso Aleatório

As técnicas de acesso aleatório são utilizadas em redes com topologia de barramento. Dois métodos de acesso aleatório serão descritos: métodos ALOHA (pioneiros) e sua evolução para

os métodos de acesso múltiplo com detecção de portadora (CSMA: Carrier Sense Multiple Access).

Método ALOHA Puro

Este médodo foi concebido na Universidade do Hawaii para uma rede conectando unidades em quatro ilhas via rádio. A técnica necessita de reconhecimento por parte do receptor e segue o algoritmo abaixo:

1. transmita o quadro;
2. aguarde o reconhecimento da recepção por T unidades de tempo; se recebido, fim;
3. gere um número aleatório (r) entre 0 e R;
4. vá para 1 após r unidades de tempo.

A técnica ALOHA pura é basteante simples. Sempre que necessitar transmitir um quadro, o host simplesmente o faz. Caso ocorra colisão (interferência entre duas transmissões), o quadro será propagado com erro, causando o seu descarte pelo destinatário. O emissor detecta colisão pelo não recebimento do reconhecimento. Neste caso, a próxima retransmissão se dará após um intervalo de tempo aleatório. É importante tentar nova transmissão após um intervalo de tempo aleatório, pois, caso contrário, uma nova colisão certamente ocorrerá se ambos os hosts colidentes tentarem a retransmissão ao mesmo tempo.

A técnica ALOHA pura apresenta baixa eficiência na utilização do canal pois uma transmissão em curso está sempre sujeita a interferência de outra que se inicia. A variante a seguir impede interferências em uma transmissão em curso.

Método ALOHA Particionado

Esta variante do ALOHA puro permite que transmissões se iniciem em intervalos de tempo bem definidos (partições). Se o período das partições for superior ao tempo de transmissão de um quadro, uma transmissão que se iniciou sem colisão será concluída sem colisão. Como desvantagem, o host deve esperar o início do próxima partição para transmitir, mesmo que o meio esteja livre. O algoritmo é dado abaixo:

1. aguarde o *beep* de início de partição (fornecido por uma estação mestre);
2. transmita o quadro;
3. aguarde o reconhecimento da recepção por T unidades de tempo; se recebido, fim.
4. gere um número aleatório (r) entre 0 e R;
5. vá para 1 após r unidades de tempo.

Método CSMA Não Persistente

Os métodos da família CSMA têm em comum a capacidade de *escutar* o meio físico para a detecção de uma transmissão em curso. Um host somente inicia a transmissão se detectar o meio em repouso (sem transições, no caso de transmissão digital). Colisões ainda podem ocorrer, se dois hosts detectarem o meio em repouso e iniciarem a transmissão ao mesmo tempo. Neste caso, a confirmação por parte do receptor, como nos métodos ALOHA, também é imprescindível.

O método CSMA Não Persistente opera segundo o algoritmo:

1. escute o meio;
2. se o meio estiver em repouso:
   (a) transmita o quadro;
   (b) aguarde o reconhecimento da recepção por T unidades de tempo; se recebido, fim;
   (c) vá para 1.
3. caso contrário (transmissão em curso):
   (a) gere um número aleatório (r) entre 0 e R;
   (b) vá para 1 após r unidades de tempo.

Quando detectada uma transmissão em curso, o método aguarda um intervalo aleatório antes de reiniciar a escuta do meio a fim de aguardar a sua liberação. Se a transmissão terminar logo após o início do intervalo aleatório, uma sub-utilização do meio é acarretada.

Método CSMA 1-Persistente

Este método é idêntico ao anterior, apenas fazendo o intervalo aleatório igual zero (escuta permanente do meio até cessar a transmissão em curso). Este método evita as esperas com o meio em repouso do anterior (aumentando portanto a utilização do canal) sob pena de um aumento da possibilidade de colisões quando dois hosts estão sensoriando o meio ocupado por um terceiro.

Método CSMA p-Persistente

É um meio termo entre o método Não Persistente e o 1-Persistente. O método detecta o meio permanentemente até a transmissão em curso se encerrar. Neste ponto, o método pode transmitir ou suspender a transmissão por um intervalo de tempo aleatório. Os passos do algoritmo CSMA p-Persistente são os seguintes:

1. escute o meio até ser detectada a condição de repouso;
2. gere um número aleatório (s) entre 0 e 1;
3. Se $s \geq p$:

   (a) transmita o quadro;
   (b) aguarde o reconhecimento da recepção por T unidades de tempo; se recebido, fim;
   (c) vá para 1.

4. Caso contrário (s < p):
   (a) gere um número aleatório (r) entre 0 e R;
   (b) aguarde r unidades de tempo;
   (c) escute o meio; se em repouso vá para 2;
   (d) caso contrário (transmissão em curso):
      i. gere um número aleatório (u) entre 0 e U;
      ii. vá para 1 após u unidades de tempo.

O método CSMA p-Persistente apresenta uma boa taxa de utilização do meio com baixa probabilidade da de ocorrência de colisões caso p seja ajustado às características do tráfego.

Método CSMA-CD

O método CSMA-CD (Collision Detection) adiciona aos métodos CSMA a detecção de colisões *sem a necessidade de aguardar reconhecimento por parte do receptor*. Detectada uma colisão, o host interrompe imediatamente a transmissão, entrando em seguida em um processo de retransmissão. O processo de detecção de colisão é simples: durante uma transmissão o host escuta o meio, comparando o sinal no meio com aquele sendo transmitido. Ocorrida uma diferença, o host conclui que uma segunda transmissão está se sobrepondo à sua.

Detectada uma colisão, o host reforça a colisão com a injeção de sinais espúrios no meio *(jamming)* a fim de que os demais hosts transmitindo detectem imediatamente a colisão e suspendam igualmente a transmissão. O algoritmo é composto dos seguintes passos:

1. escute o meio até ser detectada a condição de repouso;
2. inicie a transmissão do quadro, escutando o meio para se certificar que apenas esta transmissão está em curso; encerrada a transmissão do quadro sem colisão, fim;
3. reforce a colisão (jamming);
4. caso o número de colisões (c) na transmissão deste quadro exceder um limite, sinalize um erro à camada superior e termine;
5. gere um número aleatório (r) entre 0 e R(c);
6. vá para 1 após r unidades de tempo.

Como o método CSMA-CD detecta colisões independente do reconhecimento por parte do receptor, esta técnica pode suportar serviços de datagrama sem confirmação.

A rede Ethernet foi pioneira na introdução do método CSMA-CD, sendo comum empregar-se o termo *Ethernet* para este método de acesso ao meio.

CSMA-CD no Padrão IEEE 802.3

O padrão IEEE 802.3 estabelece o formato de quadros apresentado na figura 3.1.

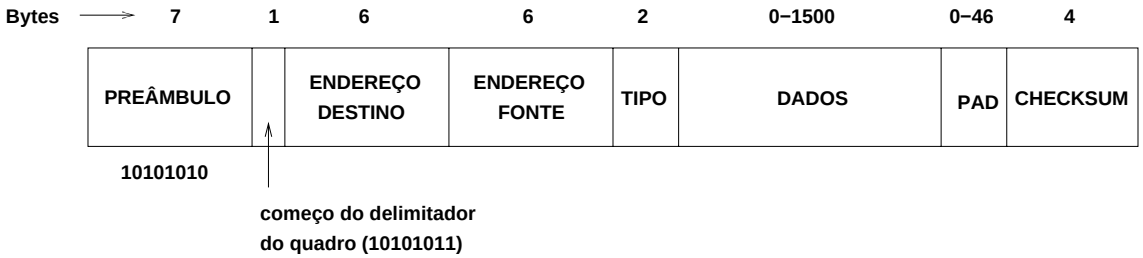

Figura 3.1: Formato dos quadros no IEEE 802.3

O quadro inicia com um preâmbulo de 7 bytes composto dos bits 10101010. A seguir vem um byte de início de quadro composto dos bits 10101011. Duas seqüências de 6 bytes estabelecem os endereços do destinatário e do emissor, respectivamente. Seguem 2 bytes contendo o tipo da informação contida no quadro e os bytes correspondentes (1500 máximo). Caso o número de bytes da informação contida no quadro seja insuficiente para atingir o tamanho mínimo de quadro (64 bytes a partir do byte de início), um *pad* de 0 a 46 bytes completa as informações do quadro. Finalmente, 4 bytes são reservados para *checksum*.

A imposição de um tamanho mínimo de quadro se dá por duas razões:

1. quadros muitos curtos emitidos nos extremos do cabo podem entrar em colisão sem que os respectivos emissores a detectem (isto é, quando um quadro atinge o extremo oposto a tansmissão do quadro neste extremo já foi concluída);

2. reforçar o *checksum*, diminuindo a probabilidade de diferentes arranjos de bits gerarem o mesmo *checksum*.

3.1.2 Métodos Baseados em Passagem de Permissão

Os métodos baseados em passagem de permissão foram desenvolvidas para redes com topologia em anel. A idéia básica é ter-se uma ficha *(token)* circulando pelo anel, de host para host. O host que detiver o *token* está autorizado a transmitir. Transmitido um quadro, este circula pelo anel até atingir o host destino. Recebido sem erros no destino, o host ativa no próprio quadro um bit de reconhecimento e transmite ao seu sucessor até atingir o host que o emitiu (note que um quadro sempre dá uma volta completa pelo anel). O emissor pode então se certificar que o quadro foi corretamente recebido ou ignorado (devido a erros ocorridos na camada física ou inexistência do destinatário), drenando-o do anel.

Nenhum host pode manter a posse do *token* por um intervalo de tempo superior a um limite pré-estabelecido. Efetuadas as transmissões ou expirado o tempo máximo de posse do *token*, o host a passa para seu sucessor. Redes com topologia em anel que empregam passagem de permissão como método de acesso ao meio são denominadas redes *token ring*.

Métodos baseados em passagem de permissão apresentam duas características básicas: inexistência de colisões e tempo máximo de espera para acessar o meio (este tempo, em teoria, é infinito para os métodos de acesso aleatório).

O anel pode conter uma *estação mestre* que tem como função verificar se o *token* não se perdeu, reiniciar o anel em caso de falhas, remover quadros corrompidos, etc. Em caso de falha desta estação, outra é eleita como mestre (automaticamente ou com a intervenção do operador).

Apesar da simplicidade do conceito, métodos de acesso ao meio baseados na passagem de permissão são bem mais complexos que os métodos de acesso aleatório. A seguintes situações devem ser tratadas:

- iniciação do anel quando o primeiro host é ligado;
- inserção de novos hosts no anel;
- reconfiguração do anel ante a falha ou o desligamento programado de hosts.

Apesar de associados à topologia em anel, métodos baseados em passagem de permissão também se aplicam a topologia de barramento. Neste caso, forma-se um anel lógico ordenando os hosts de acordo com algum critério, por exemplo, seus endereços. O *token* circula neste anel lógico, dando permissão ao host que o detém de difundir quadros no meio físico. Redes com topologia de barramento que empregam passagem de permissão como método de acesso ao meio são denominadas redes *token bus* (figura 3.2).

Nestas redes o hosts emissor não dispõe de confirmação por parte do receptor como nas redes *token ring* visto que o quadro é simplesmente difundido no barramento.

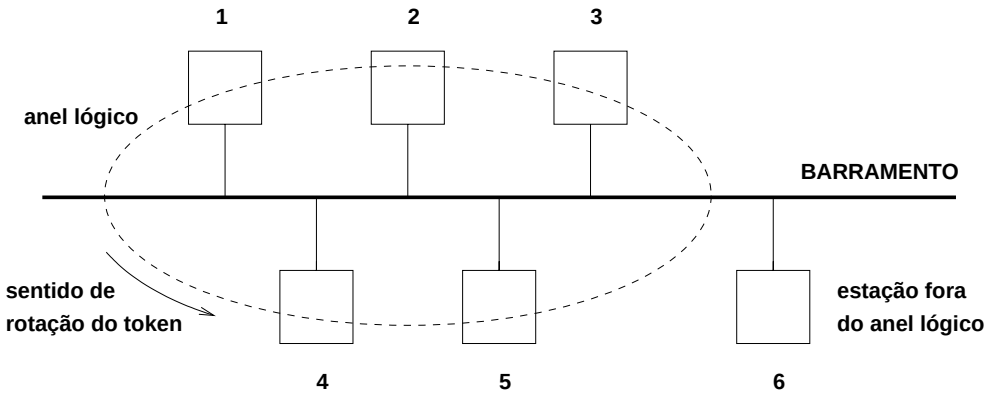

Figura 3.2: Passagem de permissão em redes com topologia de barramento.

Redes *token bus* apresentam um atrativo adicional em relação a redes *token ring*: um host pode receber mensagens sem estar participando do anel lógico (host 6 na figura 3.2), posto que todas as mensagem são transmitidas em difusão pelo meio físico. Tais hosts são incapazes de transmitir, pois jamais estarão de posse do *token* (no máximo podem responder a uma mensagem a eles direcionada). Esta característica das redes *token bus* viabiliza a inclusão de processadores extremamente simples na rede (microcontroladores, por exemplo), sem dotá-los da capacidade plena de acesso ao meio (e portanto sem empregar todas as funcionalidades das sete camadas do modelo OSI).

3.1.3 Passagem de Permissão no Padrão IEEE 802.5

O padrão IEEE 802.5 *(token ring)* é bem mais complexo que o CSMA-CD do 802.3. Duas características não presentes no 802.3 são definidas no 802.5: prioridade de acesso ao meio e reserva do meio.

O *token* é composto de 3 bytes: DI (delimitador de início), CA (controle de acesso) e TIPO. O primeito byte, DI, identifica o início do *token* e é formado por transições inválidas do código Manchester Diferencial (transições que jamais ocorrem em cadeias de 0s e 1s tais como duas transições positivas ou negativas seguidas). O segundo byte, CA, é utilizado para controle de acesso ao meio, sendo composto de agrupamentos de bits em 4 categorias:

1. status (1 bit): se o *token* está livre ou não;

2. monitor (1 bit): se o *token* passou pela estação mestre ou não. A estação mestre ativa este campo sempre que um *token* passar por ela;

3. prioridade (3 bits): estipula a prioridade mínima dos quadros que podem ser transmitidos com a captura do *token*. Se o valor deste campo for N, um host detentor do *token* pode transmitir quadros de prioridade maior ou igual a N;

4. reserva (3 bits): determina a prioridade do próximo *token* livre. Ao gerar um novo *token*, caso o valor da reserva seja maior que o valor da prioridade do token capturado, o host atribui o valor da reserva como prioridade do *token*. Se um host deseja reservar o *token*, este atribui ao campo de reserva a prioridade dos quadros que tem para transmitir, caso esta prioridade seja maior que a prioridade de reserva corrente. Para evitar que a prioridade do *token* cresça indefinidamente, todo o host que aumentar a prioridade da reserva e capturar o *token* se compromete a liberá-lo com prioridade menor.

O byte de TIPO estipula o tipo da informação que o quadro carrega: dados oriundos das camadas superiores ou controle (este último utilizado para a manutenção do anel).

É freqüente na literatura a afirmação que redes *token ring* são determinísticas, isto é, apresentam um tempo de acesso ao meio limitado (aproximadamente o tempo máximo de retenção do *token* multiplicado pelo número de hosts no anel). Tal propriedade ocorre somente se o esquema de prioridade e reserva não seja utilizado.

A figura 3.3 mostra o formato de um quadro no padrão IEEE 802.5.

Figura 3.3: Formato dos quadros no IEEE 802.5

Capturado o *token*, o quadro é injetado no anel logo a seguir. Os campos de endereço e *checksum* são idênticos ao IEEE 802.3. A quantidade de dados é ilimitada (a rigor, limitada pelo tempo máximo que um host pode reter o *token*). O campo DF (delimitador de final)

também é composto por transições Manchester Diferencial inválidas. Um campo após o DF, ST (status) contém dois bits: A e C. O bit A é ativado pelo host de destino e informa o host de origem que o destinatário tomou conhecimento do quadro a ele endereçado. O bit C é ativado se o destinatário aceitou o quadro (pode tê-lo rejeitado por falta de área para armazenamento, por exemplo).

Duas observações importantes:

1. um quadro no IEEE 802.5 não define campo de tamanho do quadro; o campo de dados começa 14 bytes após o campo DI do *token* e termina 4 bytes antes do campo DF do quadro;

2. o campo DF deve obrigatoriamente preceder o campo ST; o destinatário está em condições de aceitar um quadro (bit C do campo ST) somente após computar o *checksum* e, como não existe quadro de tamanho de dados, o campo de *checksum* só é definido após o recebimento do campo DF.

Manutenção do Anel

Uma das estações do anel é rotulada como estação mestre (EM). Via de regra, é a primeira estação a completar o procedimento de *boot*. Caso esta estação falhe, uma nova EM é eleita. Periodicamente, a EM circula um *token* com o campo TIPO contendo a informação ACTIVE_MONITOR_PRESENT. Se este *token* ficar sem circular por determinado período, inicia-se um procedimento de escolha de uma nova EM.

Assim que uma estação termina o procedimento de *boot*, ela aguarda a passagem do *token* ou um quadro de ACTIVE_MONITOR_PRESENT. Expirado este tempo de espera, a estação gera um quadro de controle com a informação CLAIM_TOKEN. Se este quadro circular sem alteração, a estação que o emitiu se torna a estação mestre. Se já existir uma EM, a mesma altera o quadro de CLAIM_TOKEN, informando ao emissor do quadro a sua existência.

São atribuições da estação mestre:

- drenar quadros corrompidos do anel;

- drenar quadros órfãos do anel (quadros não drenados pelo emissor por falhas de hardware ou software);

- verificar se o *token* não se perdeu (o host detentor do *token* falha em injetar um novo *token* no anel). Neste caso, a estação mestre gera um novo *token* no anel.

O dreno de quadros pela estação mestre é feito caso o bit de monitor do quadro CA estiver ativado quando o quadro passar pela EM (indicando que se trata de uma segunda passagem).

Finalmente, quando um host suspeitar da ruptura do anel (tempo longo sem a passagem de *tokens*), este injeta um *token* de controle com a informação BEACON no campo TIPO. Se o quadro voltar ao emissor, este supõe que o problema foi sanado. Caso contrário, a estação entra em um estado de *standby* aguardando o reestabelecimento do anel.

3.2 A Subcamada de Enlace Lógico (LLC)

A subcamada LLC provê serviços à camada de rede através de SAPs (ver Cap. 1) denominadas LSAPs. Tipicamente, 3 serviços podem estar disponíveis nas LSAPs (detalhados a seguir):

1. serviço sem conexão e sem reconhecimento;
2. serviço sem conexão com reconhecimento;
3. serviço orientado a conexão.

Para prover tais serviços, a subcamada LLC deve dispor das seguintes funcionalidades:

- partição dos pacotes oriundos da camada de rede em quadros compatíveis com a subcamada de acesso ao meio;
- detecção de quadros corrompidos por erros de transmissão;
- recuperação de quadros corrompidos;
- estabelecimento e gerenciamento de conexões;
- controle do fluxo de quadros.

3.2.1 Montagem de Quadros

Os quadros são definidos de forma diferente para redes locais ou públicas. Em redes locais, os quadros têm o formato imposto pela subcamada de acesso ao meio e os dados referentes ao protocolo de enlace são incorporados em uma PDU (ver Cap. 1) que trafega no campo de dados do quadro.

Em redes públicas, não existe subcamadada de acesso ao meio, e os quadros são definidos justamente para implementar os protocolos de enlace. Neste caso, os dados do protocolo possuem campos exclusivos no quadro.

Na montagem de quadros, a delimitação dos mesmos merece alguns comentários. Caso os quadros sejam compostos exclusivamente de caracteres (o que é raro na atualidade) pode-se definir caracteres especiais para a delimitação de quadros. Por exemplo, o código ASCII reserva alguns códigos especiais para delimitação de blocos de dados como o *Data Link Escape* (DLE), o *Start of Text* (STX) e o *End of Text* (ETX). Assim, um quadro pode ser delimitado por:

```
DLE STX ... <texto em ASCII> ... DLE ETX
```

Os caracteres DLE, STX e ETX não ocorrem no interior do texto, pois são caracteres de controle.

Atualmente, os dados que compõem o quadro são seqüências arbitrárias de bits (números, segmentos de memória, etc) que não podem ser representados como seqüência de caracteres (texto). Nestes casos, duas técnicas são bastante empregadas:

Enchimento de Bits *(bit stuffing)*

Escolhe-se um delimitador de ínicio de quadro (por exemplo 01111110) e previne-se de sua ocorrência nos dados com uma regra simples como: *A cada ocorrência de cinco 1s adiciona-se um 0.* O receptor do quadro executa procedimento inverso na recepção. A figura 3.4 ilustra este procedimento.

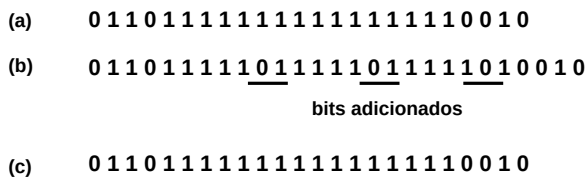

Figura 3.4: Enchimento de bits: (a) dados originais, (b) dados no quadro, (c) dados decodificados pelo receptor.

Violação de Códigos da Camada Física

Delimita-se o quadro com códigos inválidos empregados pela camada física [1]. Dado que tais códigos jamais ocorrem na codificação de 0s e 1s, sua detecção indica o início ou o final de quadros. No código Manchester Diferencial (empregado pelo 802.5) utiliza-se transições positivas (H) seguidas de negativas (L): HHLLHHLL. Na modulação FSK-coerente (empregada no 802.4), pode-se misturar freqüências altas e baixas em um mesmo período do bit, gerando assim um código inválido (figura 3.5).

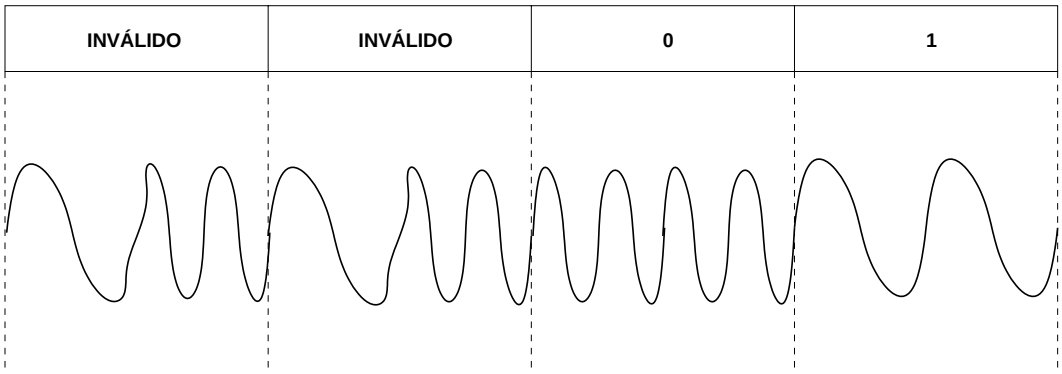

Figura 3.5: Exemplo de modulação FSK-coerente com códigos inválidos.

3.2.2 Detecção de Erros

A técnica mais usual para detecção de erros[2] durante a transmissão de quadros é a *Redundância Cíclica* (CRC). Esta técnica não utiliza bits de redundância o suficiente para

[1] A rigor, quem gera os quadros inválidos é a camada física a partir de indicador suprido pela LLC.

[2] Erro neste contexto é a inversão do valor de um ou mais bits. Inversões de N bits em seqüência é denominada *erro de rajada de comprimento N.*

correção do erro. Via de regra, detectado que um quadro chegou com erro, o receptor solicita ao emissor sua retransmissão. Em redes de computadores, a retransmissão é a técnica mais usual de correção (recuperação) de erros.

A técnica CRC é similar aos *dígitos de controle* presentes no CPF, contas bancárias, etc. O dígito de controle é computado por uma seqüência de operações bem definida na origem e transmitido ao destino que o recomputa. Em havendo diferença entre os valores transmitido e computado, conclui-se sobre a invalidade do número.

Na técnica CRC pode-se imaginar que os bits do quadro formam um número inteiro de grande maginitude. Se o quadro é composto por k bits (numerados de 0 a k-1), este número é dado pelo polinômio:

$$P(x) = b(k-1).x^{k-1} + b(k-2).x^{k-2} + ...b(1).x + b(0)$$

onde b(j) é o valor do bit na posição j (0 ou 1) e x é a base da representação (2). Por exemplo, a seqüência de 10 bits 1101011011 é representada pelo polinômio $x^9+x^8+x^6+x^4+x^3+x+1$.

A técnica CRC reserva um número arbitrário de bits (n) para detecção de erros *(checksum)*. Assim, são transmitidos n+k bits, sendo k de informação seguidos de n bits de *checksum*. Os n bits de *checksum* são computados de forma que os n+k bits do quadro sejam representados por um polinômio $F(x).P(x)$, sendo $F(x)$ de ordem n.

Seja um polinômio de referência, $G(x)$ de ordem n (denominado *polinômio gerador*). Podemos escrever a seguinte igualdade:

$$F(x).P(x) = Q(x).G(x) + R(x)$$

onde Q(x) é de ordem k-1 e R(x) de ordem n-1.

Na técnica CRC escolhe-se $F(x) = x^n$ e computa-se $R(x)$ dividindo-se os inteiros (não os polinômios !) $F(x).P(x)$ por $Q(x)$. Neste caso, $R(x)$ formará os bits de *checksum*.

As operações são feitas em aritmética módulo 2 (sem o "vai 1"):

Adição	Subtração	Multiplic.	Divisão
0 + 0 = 0	0 - 0 = 0	0 x 0 = 0	0 ÷ 1 = 0
0 + 1 = 1	0 - 1 = 1	0 x 1 = 0	1 ÷ 1 = 1
1 + 0 = 1	1 - 0 = 1	1 x 0 = 0	
1 + 1 = 0	1 - 1 = 0	1 x 1 = 1	

Em aritmética módulo 2, pode-se escrever

$$F(x).P(x) = Q(x).G(x) + R(x)$$

como

$$F(x).P(x) + R(x) = Q(x).G(x)$$

Como $F(x) = x^n$ o lado esquerdo da igualdade acima representa os bits originais acrescidos dos bits referentes a R(x) a direita.

No exemplo da seqüência 1101011011, considerando-se $n = 4$ (4 bits de *checksum*) e tomando-se o polinômio gerador: $G(x) = x^4 + x + 1$ temos

$$Q(x) = x^9 + x^8 + x^3 + x$$

$$R(x) = x^3 + x^2 + x$$

A figura 3.6 ilustra as operações, que nas implementações práticas são efetuadas por hardware.

Asssim os bits 1101011011 são transmitidos com quatro bits adicionais: 1101011011-1110

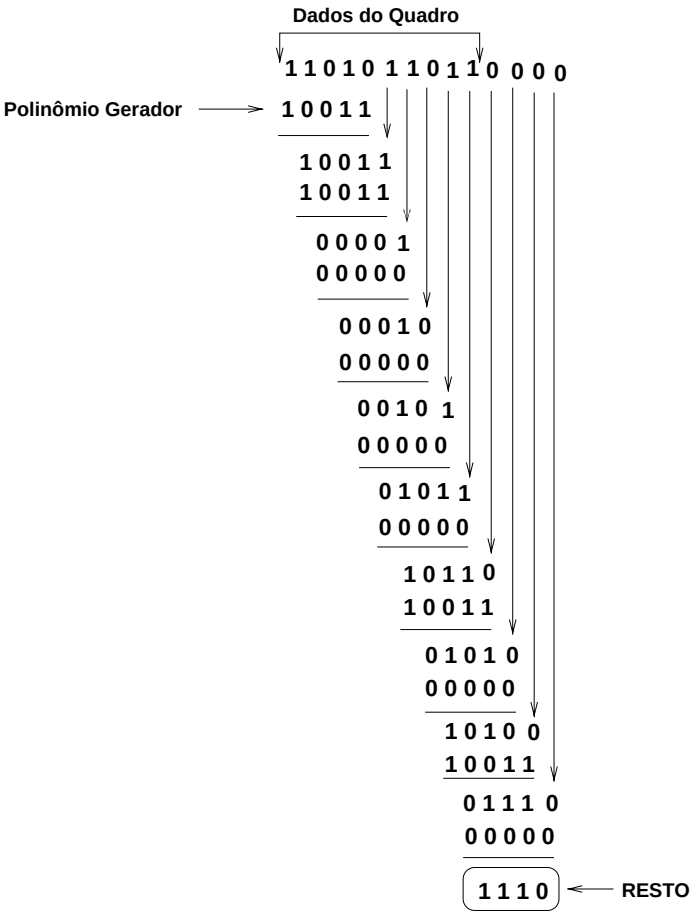

Figura 3.6: Exemplo do cômputo do *checksum*.

Tanto a ITU-T como o IEEE padronizaram polinômios geradores para o cômputo de *checksum*.

$CCR - 16 : x^{16} + x^{15} + x^2 + 1$

$CCR - CCITT : x^{16} + x^{12} + x^5 + 1$

$IEEE - 802.2 : x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Os polinomios CCR-16 e CCR-CCITT, para 16 bits de *checksum* são capazes de detectar:

- inversões de 1 ou 2 bits;

- inversões de um número ímpar de bits;
- rajadas de comprimento menor ou igual a 16;
- 99,997% das rajadas de comprimento 17;
- 99,998% das rajadas de comprimento 18.

3.2.3 Técnicas de Recuperação de Erros por Retransmissão

As técnicas mais usuais de recuperação de erros por retransmissão são baseadas em reconhecimento.

Reconhecimento Positivo

Neste método, ao transmitir um quadro, o emissor aguarda outro de reconhecimento por parte do receptor, caso o primeiro tenha sido recebido livre de erros. Recebido um quadro de reconhecimento, o emissor envia o próximo quadro e assim sucessivamente. Caso o receptor tenha detectado um erro (por exemplo, diferença entre o *checksum* computado e enviado), este simplesmente suprime o envio do reconhecimento. Expirado o tempo de espera pelo reconhecimento, o emissor retransmite o quadro.

Reconhecimento Negativo

Neste método o receptor sempre transmite um quadro de reconhecimento imediatamente após a recepção de um quadro: reconhecimento positivo, caso nenhum erro tenha sido detectado, ou negativo, caso contrário. A recepção de um reconhecimento negativo faz o emissor retransmitir prontamente o quadro, sem a necessidade de espera como no método anterior.

Reconhecimento Contínuo

Os métodos de reconhecimento positivo e negativo apresentam dois inconvenientes:

1. duplicação de quadros: 0 receptor recebe quadros duplicados em duas situações: quando o reconhecimento chegar após ter-se expirado seu tempo de espera; ou quando o reconhecimento é descartado na sua recepção devido a erros;
2. baixa eficiência: para cada quadro transmitido, circula outro de controle (reconhecimento) no sentido inverso.

O método de reconhecimento contínuo permite o emissor transmitir até N (largura da janela) quadros sem a necessidade de espera por reconhecimento. Cada quadro carrega dois contadores, N(S) e N(R), cujos valores dependem do tipo do quadro (de informação ou reconhecimento).

A figura 3.7 compara o reconhecimento contínuo com o reconhecimento negativo.

Nos quadros de informação (que fluem no sentido emissor-receptor), N(S) indica o número do quadro sendo transmitido e N(R) o número do próximo quadro esperado pelo emissor.

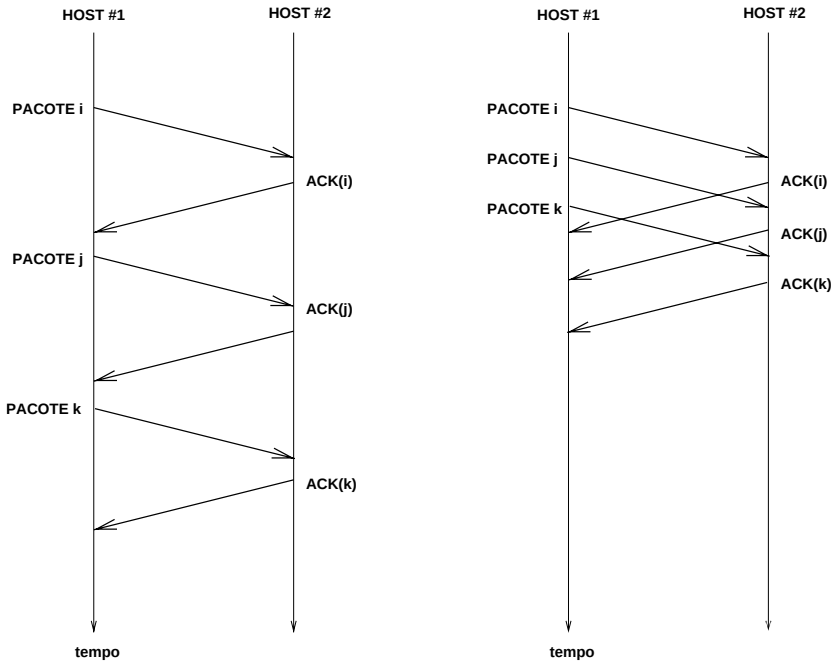

Figura 3.7: Reconhecimento negativo (esquerda) comparado com reconhecimento contínuo com janela de tamanho 3.

Nos quadros de reconhecimento, N(R) indica o próximo quadro sendo esperado (implicitamente os quadros numerados até N(R)-1 foram recebidos livre de erros). N(S) e N(R) ocupam poucos bits, tipicamente 3, sendo neste caso um contador em módulo 8. Isto justifica a introdução da janela em curso, permitindo maior segurança na identificação dos quadros.

O reconhecimento contínuo permite duas formas de operação quanto ao reconhecimento:

1. um quadro de reconhecimento para toda a janela;

2. quadros individuais de reconhecimeto.

No primeiro caso o emissor verifica se N(R) indica que a janela em curso se completou com sucesso. Neste caso, uma nova janela é iniciada com a transmissão dos próximos N quadros. Caso contrário, o emissor retransmite os quadros a partir de N(R) até o último quadro da janela, aguardando novo reconhecimento. Esta técnica reduz considravelmente os quadros de reconhecimento, aumentando a eficiência do enlace.

No segundo caso (figura 3.7) o transmissor *desliza* (avança) a janela a cada quadro de reconhecimento recebido. Como para cada quadro de dado existe um de reconhecimento, a janela desliza continuamente, o que não ocorre no primeito caso. Esta técnica minimiza as retransmissões, sendo atrativa para enlaces sujeitos a altas taxas de erro.

Quadros duplicados podem ainda ocorrer (por exemplo, quando um quadro de reconhecimento é corrompido), mas sua detecção é simples, posto que os quadros são identificados um a um.

3.2.4 Formas de Estabelecimento do Enlace

O enlace (conexão virtual) pode se dar entre dois hosts ou emanando de um host para vários outros. No primeiro caso o enlace é dito *ponto-a-ponto*, enquanto no segundo tem-se um enlace *multiponto*. Um enlace ponto-a-ponto ocorre, por exemplo, quando dois hosts se conectam para efetuar a transferência de um arquivo. Exemplo de enlace multiponto é um computador central controlando vários terminais dispersos geograficamente.

Podemos identificar três tipos de estações (hosts) quanto suas responsabilidades pelo enlace:

1. estações primárias: controlam totalmente o enlace;

2. estações secundárias: recebem comandos da primária, podendo transmitir pelo enlace somente quando autorizadas por esta;

3. estações combinadas: atuam de forma dual, ora como primária ora como secundária, dependendo do contexto.

Enlaces ponto-a-ponto são constituídos tipicamente por duas estações combinadas, enquanto enlaces multiponto são formados por uma primária e várias secundárias.

Um enlace pode ser estabelecido para operar em um dos três modos abaixo:

1. modo de resposta normal: a estação primária envia um quadro de consulta para as suas secundárias inquirindo sobre a existência de quadros para transmitir; em caso positivo, a secundária transmite imediatamente após ter recebido o quadro de consulta;

2. modo de resposta assíncrono: as estações secundárias transmitem independentemente da consulta por parte da primária; este modo é geralmente empregado quando as estações secundárias se conectam à primária por canais exclusivos (por exemplo, linhas seriais); este modo não é empregado em redes de computadores;

3. modo de resposta assíncrono balanceado: utilizado em enlaces ponto-a-ponto envolvendo apenas estações combinadas; as estações gerenciam o enlace de acordo com um protocolo definido pelas camadas de enlace dos hosts comunicantes.

Finalmente, um enlace é governado por um protocolo composto de quatro fases:

1. estabelecimento do enlace, onde um host toma a iniciativa do estabelecimento de uma conexão virtual com um ou mais hosts;

2. transferência de dados, onde os hosts que compõem a conexão trocam quadros de informação através desta;

3. encerramento do enlace, onde um dos hosts toma a iniciativa de propor o encerramento da conexão;

4. reiniciação do enlace, onde um host toma a iniciativa de reinicializar o protocolo de transferência de quadros pelo enlace pela ocorrência de um erro irrecuperável.

3.2.5 Controle do Fluxo de Quadros

O controle de fluxo é necessário quando um receptor de quadros vê-se na impossibilidade momentânea de continuar a recepção. Várias são as razões para isso ocorrer: exaustão de buffers, ocorrência de erros internos de hardware ou software, atendimento de atividades de comunicação mais prioritárias, etc.

Em protocolos baseados no reconhecimento positivo ou negativo, o controle do fluxo é desnecessário, pois o emissor só envia o próximo quadro após o recebimento do reconhecimento do receptor. Caso o receptor deseje a suspensão temporária do envio de quadros, este simplesmente deixa de enviar os quadros de reconhecimento.

Em protocolos que empregam o reconhecimento contínuo, dois quadros de controle são empregados para o controle de fluxo:

- quadros RR (Receiver Ready): informa o emissor que o receptor está pronto para iniciar ou continuar a recepção de quadros;
- quadros RNR (Receiver Not Ready): informa o emissor que o receptor está impossibilitado temporariamente de receber quadros.

Os quadros RR e RNR visam aumentar a eficiência do canal compartilhado evitando a circulação de quadros de dados que com certeza serão descartados pelo receptor.

3.3 Protocolos de Enlace

3.3.1 O Protocolo ISO HDLC para Redes Públicas

O protocolo HDLC (High-level Data Link Control) é padronizado pela ISO, servindo de base para o X.25/camada 2 que emprega um subconjunto denominado LAPB (Link Access Procedure, Balanced)[3].

No HDLC, os quadros têm o formato apresentado na figura 3.8. O quadro possui um delimitador de início e final composto dos bits 01111110. Um procedimento de codificação de bits é empregado para evitar a ocorrência desta sequência no quadro. O campo de endereço é utilizado em conexões multiponto para identificar as estações secundárias da conexão. Em conexões ponto-a-ponto este campo é utilizado para distinguir quadros de comandos dos de resposta.

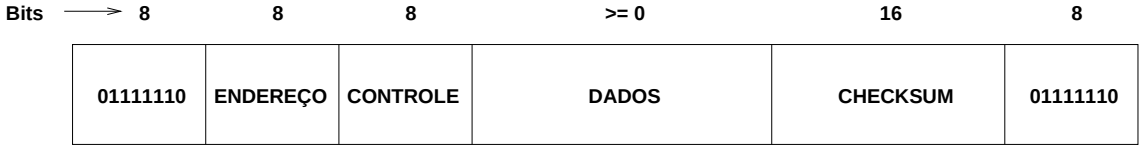

Figura 3.8: Formato do quadro no protocolo HDLC.

[3] O LAPB restringe o HDLC por permitir apenas enlace no modo de resposta assíncrono balanceado - ver subseção 3.2.4.

O campo de dados possui tamanho arbitrário, mas normalmente limitado por restrições impostas pela camada física.

O campo de *checksum* é computado pelo polinômio gerador $x^{16} + x^{12} + x^5 + 1$.

O campo de controle define três tipos de quadros: de informação (I), de supervisão (S) e não numerados (N), sendo os dois últimos quadros de controle (figura 3.9).

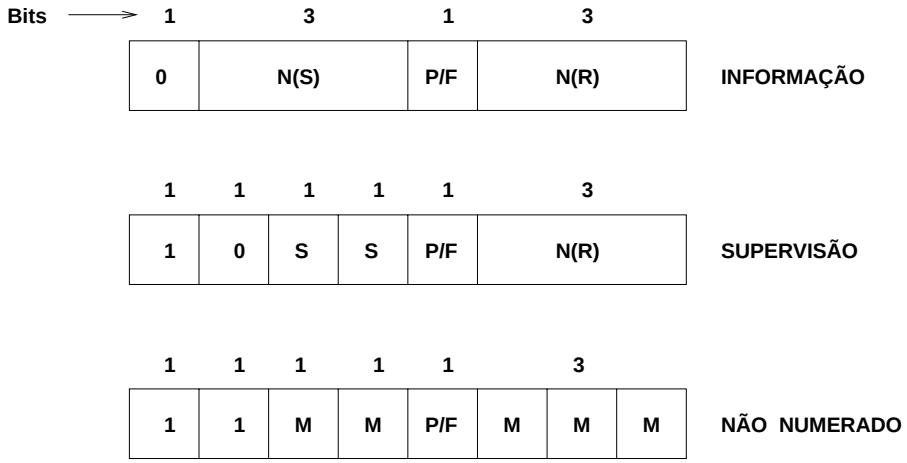

Figura 3.9: Formatos do campo de controle no protocolo HDLC.

Os quadros de informação carregam os contadores N(S) e N(R) para reconhecimento contínuo do fluxo de quadros.

Os quadros de supervisão são empregados no controle de fluxo e como reconhecimento da recepção de quadros. Três tipos instruções podem estar contidas no campo SS:

1. RR (Receiver Ready): informa que o receptor está pronto para continuar a recepção de quadros a partir de N(R), inclusive.

2. RNR (Receiver Not Ready): informa que o receptor reconhece o recebimento de todos os quadros até N(R)-1, e solicita a suspensão temporária do envio de novos quadros.

3. REJ (Reject): informa que o receptor detectou um erro na recepção do quadro N(R) e solicita o envio a partir deste.

Os quadros não numerados são utilizados para estabelecimento e término de conexões. Seis tipos de intruções são definidas:

1. SNRM (Set Normal Response Mode): estabelece uma conexão entre estação primária e secundária.

2. SABM (Set Asynchronous Balanced Mode): estabelece uma conexão entre estações combinadas.

3. DISC (Disconnect): informa o outro extremo da conexão que este host deseja terminar o enlace.

4. DM (Disconnect Mode): Informa que uma conexão solicitada não pode ser estabelecida (por exemplo, por falta de espaço para armazenar os parâmetros de controle da conexão).

5. UA (Unnumbered Acknowledgement): reconhece positivamente comandos de estabelecimento, término e reinicialização de conexão.

6. FRMR (Frame Reject): um quadro foi recebido com erro através da conexão e seu conteúdo não pôde ser identificado.

O bit P/F (Pool/Final) é ativado por uma estação primária quando em consulta a uma secundária. Caso tenha quadros para transmitir, a estação secundária o faz com o bit P/F ativado, até o último quadro onde o bit P/F é desativado, indicando o final da transmissão.

3.3.2 O Protocolo IEEE 802.2 para Redes Locais

O protocolo de enlace definido pelo IEEE no escopo do padrão 802 é denominado LLC (Logical Link Control) e foi inspirado no HDLC descrito acima. No padrão 802, os quadros têm o formato como o da figura 3.1. O LLC especifica o formato da PDU que é parte dos dados do quadro. Esta PDU, denominada LPDU, tem o formato dado pela figura 3.10.

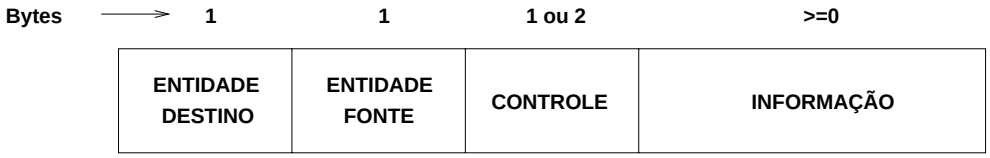

Figura 3.10: Formatos de uma LPDU.

O campo ENTIDADE FONTE especifica qual entidade gerou a PDU, enquanto o campo ENTIDADE DESTINO estipula a entidade para a qual a PDU se destina. O protocolo prevê serviços sem conexão (com e sem reconhecimento) e os serviços orientados a conexão oferecidos pelo protocolo LLC. A camada de rede invoca estes serviços através de primitivas definidas pela subcamada de enlace lógico.

As famílias de primitivas são:

- L-CONNECT: estabelecimento de conexão (com confirmação);
- L-DISCONNECT: término de conexão (sem confirmação);
- L-DATA: envio de quadros de dados através da conexão (sem confirmação);
- L-RESET: reinicialização de conexão (com confirmação);
- L-ERROR-REPORT: informe de erros pelo provedor (indicação apenas);
- L-EXPEDITED-DATA: envio de dados urgentes através da conexão (sem confirmação);
- L-UNIT-DATA: envio de dados sem conexão (sem corfirmação).

L-CONNECT.request(end_local, end_remoto, classe_de_serviço) L-CONNECT.indication(end_local, end_remoto, classe_de_serviço) L-CONNECT.response(end_local, end_remoto, classe_de_serviço) L-CONNECT.confirm(end_local, end_remoto, classe_de_serviço)
L-DISCONNECT.request(end_local, end_remoto) L-DISCONNECT.indication(end_local, end_remoto)
L-DATA.request(end_local, end_remoto, l_sdu) L-DATA.indication(end_local, end_remoto, l_sdu) L-EXPEDITED-DATA.request(end_local, end_remoto, l_sdu) L-EXPEDITED-DATA.indication(end_local, end_remoto, l_sdu)
L-RESET.request(end_local, end_remoto) L-RESET.indication(end_local, end_remoto) L-RESET.response(end_local, end_remoto) L-RESET.confirm(end_local, end_remoto)
L-ERROR-REPORT.indication(razão)
L-UNIT-DATA.request(end_local, end_remoto, l_sdu, classe_de_serviço) L-UNIT-DATA.indication(end_local, end_remoto, l_sdu, classe_de_serviço)

Tabela 3.1: Primitivas OSI de enlace. L-UNIT-DATA é utilizada para os serviços sem conexão; as demais para serviços com conexão.

A tabela 3.1 ilustra o formato destas primitivas.

A interface LLC/MAC é do tipo sem conexão, já que a comunicação entre ambas é local (interior à camada de enlace ou interface de rede). Uma única primitiva, MA-DATA, é empregada para transferir informação entre as subcamadas LLC e MAC.

O campo de controle da LPDU é inspirado no HDLC. Os três tipos de quadros do HDLC (informação, supervisão e não numerados) são definidos no protocolo LLC.

Quadros de informação são subdivididos em dois grupos: tipo I (Information) que transportam dados através de conexões, e tipo UI (Unnumbered Information) que transportam datagramas.

Quadros de supervisão são do tipo RR, RNR, REJ (idênticos ao X.25).

Quadros não numerados são do tipo SABM[4], DISC, DM, UA e FRMR (também idênticos ao X.25). Dois tipos de quadros não numerados, ausentes no X.25, são listados a seguir:

1. TEST: utilizado para testar uma conexão, fazendo com que o outro lado envie quadro similar.

2. XID (Exchange Information): utilizado para um host comunicar os tipos de serviços providos e tamanhos de janelas.

[4]O LLC suporta apenas o modo de resposta assíncrono balanceado, isto é, todas as estações são combinadas.

3.4 Problemas

1. Descreva o funcionamento dos principais métodos de acesso ao meio para redes locais.
2. Descreva os campos que compõem o quadro dos padrões IEEE 802.3 e 802.5.
3. Descreva os campos que compõem o *token* do padrão IEEE 802.5.
4. Descreva a manutenção do anel no padrão IEEE 802.5.
5. Descreva as técnicas mais usuais para delimitação de quadros e em que situações são empregadas.
6. Por que protocolos de enlace baseados em caracter estão superados ?
7. Compute o *checksum* utilizando a técnica CRC para a seqüência de 12 bits 111001100010 e polinômio gerador $x^4 + x^2 + x + 1$.
8. Faça um diagrama temporal mostrando a transmissão de quadros com reconhecimento contínuo e largura de janela 3, utilizando quadros de controle tipo RR e RNR.
9. Descreva os quadros não numerados empregados pelo padrão X.25/camada 2.
10. Faça um diagrama temporal mostrando o estabelecimento de enlace, envio de quadros e término do enlace utilizando os quadros do protocolo HDLC.