VOLTAR

Princípios de detecção e correção de erros, princípios de controle de link e princípios de acesso múltiplo

Verificar na camada de enlace do modelo OSI os principais mecanismos de detecção e correção de erros, controle do link e acesso ao meio de transmissão.

NESTE TÓPICO

Marcar tópico

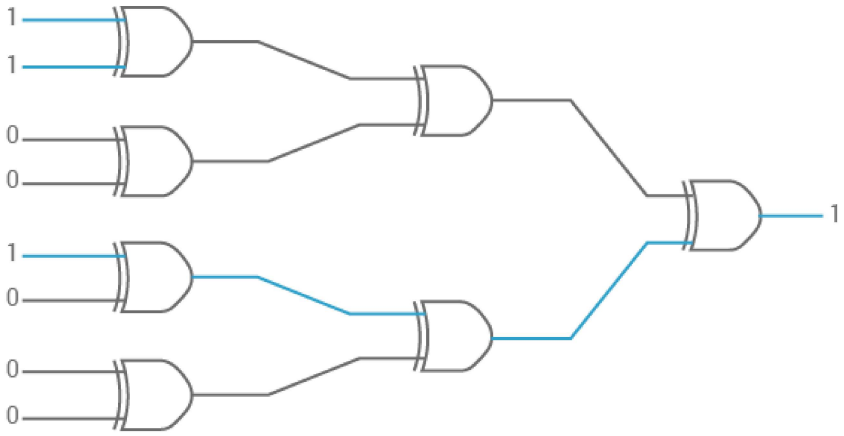
Em uma transmissão em rede o que se espera é a integridade dos dados da origem até o seu destino final. Entretanto vários fatores contribuem para o surgimento de erros durante a transmissão, portanto surge a necessidade de mecanismos que permitam detectar e corrigir estes erros.

TANENBAUM, Andrew. S. (2011) menciona que a forma mais comum de garantir uma entrega confiável é dar ao transmissor algum tipo de feedback sobre o que está acontecendo no outro extremo da linha. Normalmente, o protocolo solicita que o receptor retorne quadros de controle especiais com confirmações positivas ou negativas sobre os quadros recebidos. Se receber uma confirmação positiva sobre um quadro, o transmissor saberá que o quadro chegou em segurança ao destino. Por outro lado, uma confirmação negativa significa que algo saiu errado e que o quadro deve ser retransmitido.

Em canais altamente confiáveis, como os de fibra, é mais econômico utilizar um código de detecção de erros e simplesmente retransmitir o bloco com erro. Porém, em canais como enlaces sem fio que geram muitos erros, é melhor adicionar a cada bloco redundância suficiente para que o receptor seja capaz de descobrir qual era o bloco original, em vez de confiar em uma retransmissão.

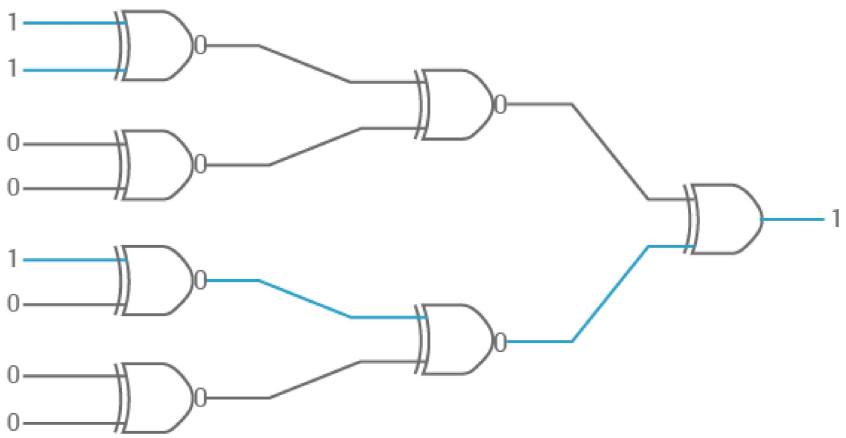
Detecção de Erro - Paridade Par e Paridade Ímpar

O bit de paridade é escolhido de forma que o número de bits 1 da palavra de código seja par (ou ímpar). Por exemplo, quando 1011010 é enviado com paridade par, é acrescentado um bit ao final para formar 10110100. Com paridade ímpar, 1011010 passa a ser 10110101. Um código com um único bit de paridade tem uma distância igual a 2, pois qualquer erro de um único bit produz uma palavra de código com a paridade errada. Isso pode ser usado para detectar erros isolados. Como um exemplo simples de código de detecção de erros por paridade par, considere um circuito com OU exclusivo **(XOR)** para 11001000:(KUROSE, J. F 2013).



Portanto a mensagem com o bit de paridade par seria: 110010001

Para a paridade ímpar o circuito poderia ser desenvolvido com portas não OU exclusivo (X-NOR) conforme exemplificado abaixo, portanto, para a mesma mensagem (110010001) o bit de paridade ímpar seria 0.



Detecção de Erro - *CRC – Cyclic Redundancy Code* **(Códigos Cíclicos de Detecção de Erros)**

São capazes de detectar uma grande faixa de erros de transmissão, isolados ou em rajadas, possuem algoritmo de cálculo mais complexos do que a paridade e podem ser calculados por hardware ou software.

No CRC cada bit da mensagem "m" codificada em binário, é considerado como um coeficiente de um polinômio M(X) base 2. A mensagem é deslocada para a esquerda de "r" posições, onde "r" é o número de bits do CRC (ordem do polinômio verificador = número de bits da representação do polinômio verificador – 1). A mensagem deslocada é dividida por um polinômio característico "G(X)". O resto da divisão é somado à mensagem deslocada para formar a mensagem composta "T(X)".que é transmitida. O receptor divide "T(X)" por "G(X)". Se o resultado for "0", existe grande probabilidade da mensagem estar correta, caso contrário, existe um erro.

Por exemplo: deseja-se enviar os bits de dados: 111100101 e o polinômio verificador é 101101, portanto "r" será: quantidade de bits do polinômio verificador (seis bits) menos um, ou seja 6 - 1 = 5. A mensagem com deslocamento à esquerda de "r" (cinco bits zeros) fica: 11110010100000. A seguir dividimos este valor pelo polinômio verificador, TANENBAUM, Andrew. S. (2011).

Vídeo

0:00 / 2:47

CRC na origem

Com a determinação do CRC este é adicionado ao final da mensagem, que quando recebida no destino é verificada com a divisão do polinômio verificador conforme ilustrado a seguir.

Vídeo

0:00 / 2:25

CRC no destino

Código de Hamming

O código de Hamming é um código de detecção e correção de erro, isto é, permite não apenas detectar erro de um bit, mas também a localização do bit incorreto.

Os bits da palavra de código são numerados consecutivamente, começando com o bit 1 da extremidade esquerda, com o bit 2 imediatamente à sua direita e assim por diante.

Os bits que são potências de 2 (1, 2, 4, 8, 16 etc.) são bits de verificação. Os outros (3, 5, 6, 7 etc.) são preenchidos com os m bits de dados. Cada bit de verificação força a paridade de algum conjunto de bits, incluindo seu próprio conjunto, a ser par (ou ímpar). Um bit pode ser incluído em vários cálculos de paridade. Se quiser ver para quais bits de verificação o bit de dados da posição k contribui, reescreva k como a soma de potências de 2.

Por exemplo, 11 = 1 + 2 + 8 e 29 = 1 + 4 + 8 + 16.

Um bit é verificado apenas pelos bits de verificação que ocorrem em sua expansão (por exemplo, o bit 11 é verificado pelos bits 1, 2 e 8). Quando uma palavra de código é recebida, o receptor inicializa um contador como zero.

Em seguida, ele examina cada bit de verificação k (k = 1, 2, 4, 8,...) para confirmar se a paridade está correta. Caso não esteja, k é incluído no contador.

Se o contador indicar zero após todos os bits de verificação terem sido examinados (ou seja, se todos estiverem corretos), a palavra de código será aceita como válida. Se o contador não for igual a zero, ele conterá o número do bit incorreto. Por exemplo, se os bits de verificação 1, 2 e 8 estiverem incorretos, o bit invertido será igual a 11, pois ele é o único verificado pelos bits 1, 2 e 8.

Vejamos abaixo um exemplo de geração do código Hamming para a mensagem: 0101

		m1		m2	m3	m4
x1	x2	0	x3	1	0	1
1	2	3	4	5	6	7
2^0	2^1		2^2			

Primeiro geramos uma numeração de 1 a 7;

Depois verificamos neste intervalo quais valores de posição são potência de dois, no exemplo a posição 1 é representativa de 20 , a posição 2 é relativa a 21 e a posição quatro é igual 22.

Estas posições que são potência de dois serão as posições do bit de verificação e são denominadas como x1, x2 e x3.

As demais posições devem ser preenchidas com os bits da mensagem e denominadas de m1, m2, m3 e m4.

Vídeo

Vamos calcular os bits de verificação para cada *bit* da mensagem:

m1 = 3 = 2 + 1 = x1 + x2

m2 = 5 = 4 + 1 = x3 + x1

m3 = 6 = 4 + 2 = x3 + x2

m4 = 7 = 4 + 2 +1 = x3 + x2 + x1

Determinando os valores de x1, x2 e x3 (utilizando XOR)

x1 ocorre em m1, m2 e m4, portanto:

$x1 = m1 \oplus m2 \oplus m4$

os valores de m1 = 0, m2 = 1 e m4 =1

$x1 = 0 \oplus 1 \oplus 1 = 0$

x2 ocorre em m1, m3 e m4 portanto:

$x2 = m1 \oplus m3 \oplus m4$

os valores de m1 = 0, m3 = 0 e m4 = 1

$x2 = 0 \oplus 0 \oplus 1 = 1$

x3 ocorre em m2, m3 e m4 portanto:

$x3 = m2 \oplus m3 \oplus m4$

os valores de m2 = 1, m3 = 0 e m4 = 1

$x3 = 1 \oplus 0 \oplus 1 = 0$

O código de Hamming será:

0:00 / 5:55

Código de Hamming

Vídeo

$x1 = m1 \oplus m2 \oplus m4$

$x2 = m1 \oplus m3 \oplus m4$

$x3 = m2 \oplus m3 \oplus m4$

No receptor é realizada verificação por paridade par:

P1 = x1, m1, m2 e m4, ou seja, 0, 0, 1, 1 paridade par = 0

P2 = x2, m1, m3 e m4, ou seja, 1, 0, 0, 1 paridade par = 0

P3 = x3, m2, m3 e m4, ou seja, 0, 1, 0, 1 paridade par = 0

Quando todos os *bits* de paridade estão em zero não há erros.

Entretanto se a mensagem recebida for 0111

P1 = x1, m1, m2 e m4, ou seja, 0, 0, 1, 1 paridade par = 0

P2 = x2, m1, m3 e m4, ou seja, 1, 0, 1, 1 paridade par = 1

P3 = x3, m2, m3 e m4, ou seja, 0, 1, 1, 1 paridade par = 1

Pela paridade há erro, como determinar o bit com erro?

Primeiro, inverte-se os *bits* de paridade: 110

Segundo converter o valor para decimal:

110 = 6

Portanto o *bit* incorreto está na posição seis que é o *bit* m3:

0:00 / 2:50

Correção de erro - HAMMING

Controle de Link

Também na camada de enlace além da detecção e correção de erro a subcamada Logical Link Control (LLC) é responsável por estabelecer uma interface padronizada para muitos protocolos de comunicação diferentes, controlando o fluxo de dados e garantindo que os dados cheguem ao seu destino.

O controle de ligação lógica insere os dados no que é conhecido como frames (quadros). Os quadros são encaminhados para a subcamada MAC, que atribui os endereços de hardware específicos para os quadros.

Uma vez que a camada de enlace de dados concluiu seu trabalho, ela passa os dados para a camada mais baixa do modelo OSI, a camada física, que, em seguida, transforma os dados em um fluxo de sinais elétricos para o meio de transmissão.

Desta forma, a LLC fornece a capacidade para qualquer das camadas superiores transmitirem dados sem ter que saber qualquer coisa sobre o tipo de rede que serão utilizadas para o encaminhamento de dados. A camada de enlace de dados pode ser projetada de modo a oferecer diversos serviços, que podem variar de sistema para sistema.

Três possibilidades são oferecidas com frequência:

1. Serviço sem conexão e sem confirmação.

2. Serviço sem conexão com confirmação.

3. Serviço orientado a conexões com confirmação.

O serviço sem conexão e sem confirmação consiste em fazer a máquina de origem enviar quadros independentes à máquina de destino, sem que a máquina de destino confirme o recebimento desses quadros. Nenhuma conexão lógica é estabelecida antes ou liberada depois do processo. Se um quadro for perdido devido a ruídos na linha, não haverá nenhuma tentativa de detectar a perda ou de recuperá-lo na camada de enlace de dados. Essa classe de serviço é apropriada quando a taxa de erros é muito baixa, e a recuperação fica a cargo de camadas mais altas. Ela também é apropriada para o tráfego em tempo real, no qual, a exemplo da fala humana, os dados atrasados causam mais problemas que dados recebidos com falhas. A maior parte das LANs utiliza serviços sem conexão e sem confirmação na camada de enlace de dados.

O próximo é o serviço sem conexão com confirmação. Quando esse serviço é oferecido, ainda não há conexões lógicas sendo usadas, mas cada quadro enviado é individualmente confirmado. Dessa forma, o transmissor sabe se um quadro chegou corretamente ou não. Caso não tenha chegado dentro de um intervalo de tempo específico, o quadro poderá ser enviado outra vez. Esse serviço é útil em canais não confiáveis, como os sistemas sem fio. A camada de rede sempre pode enviar um pacote e esperar que ele seja confirmado. Se a confirmação não chegar durante o intervalo do timer, o transmissor poderá enviar a mensagem inteira mais uma vez.

O terceiro serviço oferecido é o orientado a conexões. Com ele, as máquinas de origem e destino estabelecem uma conexão antes de os dados serem transferidos. Cada quadro enviado pela conexão é numerado, e a camada de enlace de dados garante que cada quadro será de fato recebido. Além disso, essa camada garante que todos os quadros serão recebidos uma única vez e

na ordem correta. Na primeira fase, a conexão é estabelecida, fazendo-se ambos os lados inicializarem as variáveis e os contadores necessários para controlar os quadros que são recebidos e os que não são. Na segunda fase, um ou mais quadros são realmente transmitidos. Na terceira e última fase, a conexão é desfeita, liberando-se as variáveis, os buffers e os outros recursos usados para mantê-la.

Controle de fluxo

Na camada de enlace de dados um transmissor pode querer enviar quadros mais rapidamente do que o receptor é capaz de aceitar.

Duas abordagens são utilizadas para solucionar este tipo de situação.

Na primeira, chamada controle de fluxo baseado em feedback, o receptor envia de volta ao transmissor informações que permitem ao transmissor enviar mais dados, ou que pelo menos mostram ao transmissor qual a situação real do receptor.

Na segunda, chamada controle de fluxo baseado na velocidade, o protocolo tem um mecanismo interno que limita a velocidade com que os transmissores podem enviar os dados, sem usar o feedback do receptor. Existem diversos esquemas de controle de fluxo. No entanto, a maioria deles utiliza o mesmo princípio básico. O protocolo contém regras bem definidas sobre quando um transmissor pode enviar o quadro seguinte. Com frequência, essas regras impedem que os quadros sejam enviados até que o receptor tenha concedido permissão para transmissão, implícita ou explicitamente. Por exemplo, quando uma conexão é estabelecida, o receptor pode informar a quantidade de quadros que podem ser enviados e aguardar a confirmação para novo envio.

Acesso Múltiplo

CSMA *(Carrier Sense Multiple Access)*

CS *(Carrier Sense)*: Capacidade de identificar se está ocorrendo transmissão;

MA *(Multiple Access)*: Capacidade de múltiplos nós concorrerem pela utilização da mídia;

Protocolo de controle de acesso ao meio que busca ao máximo evitar a colisão de quadros (pacotes da camada de enlace) em redes com múltiplo acesso ao meio. O mecanismo usado para coordenar a transmissão numa rede Ethernet, não evita as colisões por completo. Elas ocorrem quando o sensoriamento do canal é simultâneo, fazendo com que dois ou mais hosts suponham não haver transmissão e as iniciam concomitantemente. Quando um nó pretende emitir dados, verifica se o meio de transmissão está livre, se for esse o caso procede à emissão.

Se o meio de transmissão está ocupado, existem vários algoritmos possíveis:

? CSMA NÃO PERSISTENTE - se o meio de transmissão está ocupado, ele espera um período de tempo aleatório e voltar a tentar, a desvantagem é que o meio de transmissão pode estar desocupado enquanto existem dados para transmitir.

? CSMA PERSISTENTE - continua a escutar o meio até que esteja livre e emiti os dados. Se existe mais do que um nó nestas condições ocorre uma colisão, nesse caso espera um período de tempo aleatório e volta a tentar.

? CSMA P PERSISTENTE - tenta diminuir as colisões evitando que o meio de transmissão seja subutilizado: espera até que o meio esteja livre, então transmite com uma probabilidade p, em alternativa espera um período de tempo equivalente ao atraso máximo de propagação no meio de transmissão e volta ao início.

Funcionamento

O CSMA identifica quando a mídia está disponível para a transmissão. Neste momento a transmissão é iniciada. O mecanismo CD (Collision Detection ou em português detecção de colisão) ao mesmo tempo obriga que os nós escutem a rede enquanto emitem dados, razão pela qual o CSMA/CD é também conhecido por “Listen While Talk“ (traduzido como "escute enquanto conversa") (LWT).

Se o mesmo detecta uma colisão, toda transmissão é interrompida e é emitido um sinal de 48 bits para anunciar que ocorreu uma colisão.

Para evitar colisões sucessivas o nó espera um período aleatório e volta a tentar transmitir.

Muitas variações são usadas para aumentar a eficiência do método, como CSMA/CD e CSMA/CA.

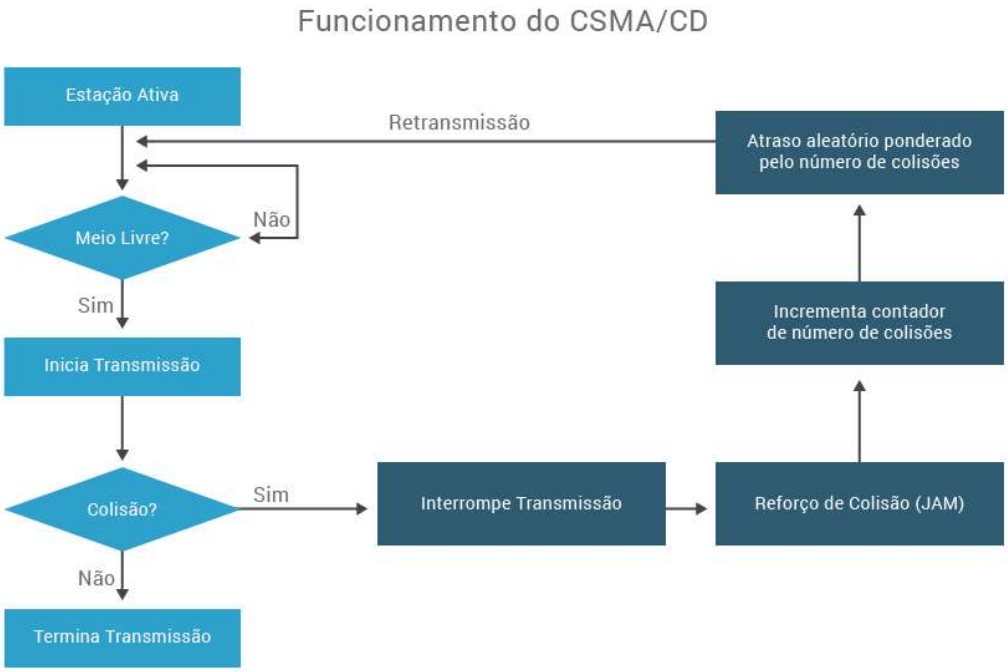
CSMA/CD ("Carrier-Sense Multiple Acces with Collision Detection", com detecção de colisão) é usado em redes Ethernet, presente em quase todas as redes locais atuais.

CSMA/CA ("Carrier-Sense Multiple Acces with Collision Avoidance", com prevenção de colisão) é popular em redes locais sem fio (WLANs). (Tanenbaum, A.S. 2011).

CSMA/CD - *Collision Detection*

Responsável por identificar colisões na rede, permite recuperar a transmissão de dados, quando surge uma colisão na emissão de pacotes de dados, o meio de transmissão fica inutilizado durante toda a transmissão dos mesmos.

Utiliza um algoritmo 1-persistente que é o mais eficiente sob o ponto de vista da utilização do meio de transmissão, em lugar de minimizar o número de colisões, tenta-se reduzir as suas consequências. O mecanismo CD obriga a que os nós escutem a rede enquanto emitem dados. Como o CD tem a capacidade de ?ouvir? enquanto ?fala?, o mesmo compara se a amplitude do sinal recebido é a mesma do sinal enviado. Desta forma, quando se ouve algo diferente do que foi dito, é identificada uma colisão.



CSMA/CA - *Collision Avoidance.*

Similar ao CSMA/CD, porém ele implementa a prevenção de colisão em vez de detecção de colisão.

Define quadros especiais denominada solicitação de envio e liberação para envio, que auxiliam a minimizar as colisões. Um nó emissor envia um quadro RTS (solicitação de envio) ao nó receptor.

Se o canal está livre, o nó receptor envia um quadro CTS (liberação para envio) ao nó emissor.

Consiste em transmitir um pequeno pacote de controle ao receptor. Sua confirmação assegura que o transmissor poderá transmitir para o outro computador. Mesmo que dois computadores comecem a transmitir, cada um, um pacote de controle simultaneamente, o CSMA/CA permitirá que o receptor detecte os sinais como uma interferência ou colisão, nesse caso, ambos os transmissores ficarão sem resposta e aguardarão um tempo aleatório antes de tentar iniciar novamente suas transmissões.

Existe um aspecto importante a considerar para que as colisões sejam detectadas com sucesso, o tamanho mínimo dos pacotes deve ser tal que o seu tempo de transmissão seja superior ao dobro do atraso de propagação. Se isto não acontecer uma estação pode completar a emissão do pacote sem que o sinal produzido pela colisão chegue a tempo.

Quiz

Exercício

Princípios de detecção e correção de erros, princípios de controle de link e princípios de acesso múltiplo

INICIAR

Quiz

Exercício Final

Princípios de detecção e correção de erros, princípios de controle de link e princípios de acesso múltiplo

INICIAR

Referências

TANENBAUM, Andrew. S.; WETHEREALL, D. Redes de Computadores, sa Ed. São Paulo: Pearson, 2011.

KUROSE, J. F.: ROSS, K. W. Redes de Computadores e a Internet: uma abordagem top-down; 6a Ed., São Paulo: Pearson, 2013.

Avalie este tópico

ANTERIOR
Qualidade de serviço. Protocolo IPv6

Índice

PRÓXIMO
Protocolo Ethernet

Ajuda?
(https://ava.uninove.br/seu/AVA/topico/topico.php?idCurso=)

Biblioteca
(https://www.uninove.br/conheca-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/)
Portal Uninove
(http://www.uninove.br)
Mapa do Site