CAMADA DE ENLACE

Resumo

Leonardo Brito (LMPB)

SUMÁRIO

OBJETIVO E FUNCIONALIDADES

OBJETIVO

Transferir entre dois nós interligados por um único enlace.

SERVIÇOS

FRAMING

A maioria dos protocolos de enlace encapsula os dados recebidos em quadros (frames).

LINK ACCESS

Estabelecimento de regras para que se acesse o enlace.

CONFIABILIDADE

Fornecer confiabilidade de transferência de quadros no enlace.

DETECÇÃO E CORREÇÃO DE ERROS

Várias técnicas: checksum, bits de paridade, CRC etc.

HALF/FULL DUPLEX

Half-duplex: nós só podem ou transmitir ou receber dados de uma vez.

Full-duplex: nós podem transmitir e receber dados simultaneamente.

DETECÇÃO E CORREÇÃO DE ERROS

A detecção de erros também está presente em outras camadas. No entanto, na camada enlace ela pode ser mais sofisticada, pois geralmente é **implementada em hardware**, enquanto em camadas superiores em geral são implementadas via software.

BITS DE PARIDADE

É o tipo mais simples de detecção de erro.

Com 1 bit de paridade, pode-se detectar em média **50% dos erros**:

Supomos um esquema de paridade par (pacote correto → número par de 1s). Se houver 1 bit trocado no quadro, o erro será detectado. Se houver 2 bits trocados, o erro não será detectado, pois ainda haverá um número par de 1s. Ou seja, se um número **ímpar** de bits forem trocados o erro será detectado; já se um número **par** de bits for detectado, o erro não será detectado. Supondo que a probabilidade da quantidade de bits trocados ser ímpar é igual à probabilidade de ser par, temos uma chance de 50% de detecção.

Há também o esquema **bidimensional** de bits de paridade: os dados são vistos como uma matriz, de forma que uma coluna e linha erradas indicam o bit exato que foi trocado, o que permite a correção do erro (**FEC – forward error correction**).

Ambos os esquemas são casos específicos do conceito geral de bits de paridade n-dimensionais.

CHECKSUMMING

Interpreta-se os dados como cadeias de inteiros de *k* bits; a soma desses inteiros forma o checksum. No **internet checksum,** usado por TCP e UDP, o complemento a 1 (inversão dos bits) dessa sorma forma o checksum.

CYCLIC REDUNDANCY CHECK – CRC

A idéia do CRC é concatenar aos dados uma cadeia de bits de forma que a cadeia resultante seja divisível por uma outra cadeia específica, chamada **cadeia geradora**. CRC usa aritmética modular em seus cálculos.

Tem-se *d* bits de dados, D. Tem-se uma cadeia geradora G de *r+1* bits.

O emissor escolherá uma cadeia de *r* bits de forma que, concatenada aos bits menos significativos de D, a cadeia resultante de *d+r* bits é divisível por G.

Basta ao receptor dividir os *d+r* bits recebidos por G e verificar se o resto é zero. Caso não seja, ele terá certeza que houve erro.

O CRC detecta garantidamente erros de até *r* bits sucessivos, e detecta erros de mais de *r* bits sucessivos com probabilidade 1-0,5^r.

MULTIPLE ACCESS PROTOCOLS

Há basicamente dois tipos de enlaces: os **ponto-a-ponto**, onde o enlace é exclusivo aos dois hosts que interliga, e os **broadcast**, que são compartilhados por vários usuários.

Surge uma dificuldade em enlaces broadcast: se dois ou mais nós transferirem dados ao mesmo tempo, haverá uma **colisão** e poderá ocorrer perda dos dados.

Idealmente, os protocolos de acesso múltiplo devem ser:

1. Simples
2. Descentralizados
3. Quando um só nó estiver ativo, ele deve obter toda a banda passante
4. Quando n nós estiverem ativos, cada um deve ter, em média, throughput de R/n, sendo R a banda passante máxima do enlace

Há basicamente três abordagens para se construir um protocolo de acesso múltiplo: partição de canal, acesso aleatório e taking turns.

CHANNEL PARTITIONING

O enlace é dividido logicamente entre os nós.

TEMPO: TDMA (TIME-DIVISION MULTIPLE ACCESS)

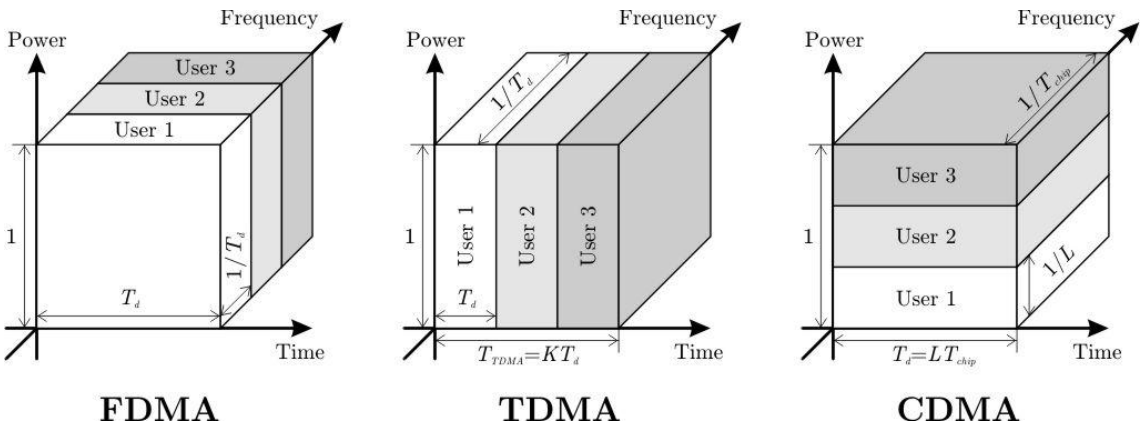
O canal é dividido em “frames” (não os quadros da camada enlace), cada qual com um número de “slots”. Cada nó tem o tempo de um slot para transmitir, evitando assim as colisões.

FREQÜÊNCIA: FDMA (FREQUENCY-DIVISION MULTIPLE ACCESS)

O canal é dividido em faixas de frequência; cada nó pode transmitir numa faixa, evitando assim as colisões.

CÓDIGO: CDMA (CODE-DIVISION MULTIPLE ACCESS)

Cada bit de dado é relacionado com vários bits de código, de forma que vários nós podem transmitir simultaneamente e ainda assim receberem seus dados corretos. Em vez de dividir tempo ou “espaço” (frequência), CDMA divide o *espaço de códigos* entre os nós.



RANDOM ACCESS

Ao contrário do Channel Partitioning, quando um nó tem permissão para transmitir, ele o faz sempre na capacidade máxima do enlace. Quando uma colisão ocorre, os nós envolvidos esperam algum tempo e tentam transmitir novamente, até que os quadros sejam recebidos corretamente.

ALOHA PURO

Totalmente descentralizado. Foi um dos primeiros protocolos de acesso aleatório, surgido nos anos 1970.

Quando um quadro é gerado pelo host, ele imediatamente começa a transmiti-lo pelo canal. Se houver uma colisão, o host retransmite imediatamente com uma probabilidade *p*, ou espera por algum tempo com uma probabilidade *1-p* e tenta transmitir novamente com a mesma probabilidade *p*, até que se consiga transmitir o pacote.

O ALOHA puro tem metade da eficiência do slotted ALOHA: *1/2e*.

SLOTTED ALOHA

Evolução do ALOHA, tem o dobro da eficácia: *1/e* (~37%). A melhor eficiência se dá devido ao uso de slots:

- Cada slot tem L/R segundos, ou seja, é suficiente para transmitir exatamente 1 quadro
- Quadros só podem ser transmitidos no começo de cada slot
- Nós estão sincronizados quanto ao começo de cada slot

Funcionamento de um nó querendo transmitir um quadro:

1. Nó espera começo do próximo slot e inicia a transmissão do quadro
2. Se houve colisão, nó espera até o próximo slot e retransmite com probabilidade *p*, ou espera (não faz nada) com probabilidade *1-p*

Ou seja, basicamente a mesma idéia do ALOHA mas com a rigidez de slots bem-definidos.

O slotted ALOHA representa uma evolução em relação ao ALOHA puro, porém deixa de ser descentralizado: todos os nós tem de estar sincronizados com um servidor central que defina os slots; caso esse servidor falhe, todo o sistema falha.

CSMA (CARRIER SENSE MULTIPLE ACCESS)

A eficácia de protocolos estilo ALOHA cai por eles não *ouvirem* (sensing) o canal antes de começar a transmitir, o que poderia impedir colisões de ocorrerem. Protocolos CSMA escutem o canal antes de iniciar uma transmissão. Se detectarem que o canal está ocupado, esperam por algum tempo e escutam o canal novamente, até que ele esteja livre para transmissão.

CSMA/CD (CSMA COM COLLISION DETECTION)

É o protocolo usado por redes Ethernet.

Além de escutar o canal *antes* de começar a transmitir, CSMA/CD *continua a escutar* o canal *durante* a transmissão, de forma que ao detectar uma colisão, interrompe imediatamente a transmissão. Portanto, CSMA/CD está fortemente escorado na capacidade dos nós perceberem se o canal está ou não sendo usado, o que é fácil em redes Ethernet.

O CSMA/CD usado no Ethernet usa o mecanismo de **exponential backoff** para decidir quanto tempo se espera após uma colisão: após sofrer a n-ésima colisão consecutiva, o adaptador escolhe aleatoriamente K = {0, 1, 2, ... , 2^(m-1)}, sendo m = min(n,10), e espera por 512*K ciclos.

Funcionamento:

1. Adaptador gera um quadro a ser transmitido. Faz carrier-sensing no enlace: se o enlace estiver ocupado, ele espera até que detecte que ele foi desocupado. Caso contrário, começa a transmitir o quadro.
2. Se, durante a transmissão, o adaptador detectou que outro nó está transmitindo no enlace, ele transmite um **jam signal** de 48 bits e inicia o backoff exponencial:
    a. Considerando que essa foi a n-ésima colisão para esse quadro, escolhe K = {0, 1, 2, ... , 2^(m-1)}, sendo m = min(n,10)
    b. Espera por K*512 ciclos
    c. Retorna para passo 1

CSMA/CA (CSMA COM COLLISION AVOIDANCE)

Como dissemos antes, o CSMA/CD depende fortemente da capacidade dos adaptadores de saberem se outros nós estão ou não transmitindo no enlace. Apesar de fácil em Ethernet, nem todos os tipos de enlace fornecem essa funcionalidade. Por exemplo, em redes sem fio 802.11, há algumas peculiaridades (**hidden terminal problem, signal fading**) que impediriam o funcionamento correto de um protocolo CSMA/CD.

O CSMA/CA não tenta detectar colisões. Em vez disso, ele tenta *evita-las*, fazendo uso de requisições e acks explícitos. Quadros CSMA/CA implementam:

- Campo de duração, que indica aos nós que o receberem que o canal estará ocupado por aquele tempo especificado.
- **Request to Send (RTS)**, que é enviado pelo emissor ao receptor e indica o comprimento dos pacotes de dados e do ACK
- **Clear to Send (CTS),** que é enviado em resposta ao RTS e dá permissão ao emissor começar a transmitir. O CTS é transmitido pelo (futuro) receptor do pacote, o que resolve o problema do hidden terminal
- ACKs confirmando o recebimento correto do pacote.

TAKING-TURNS

Nos protocolos taking-turns, cada nó espera pela sua "vez" de transmitir.

Isso pode ser feito basicamente de duas maneiras: **polling** ou **token-passing**.

No **polling**, um nó mestre coordena a rede inteira, perguntando a cada nó se ele deseja transmitir. É introduzido o polling delay: independente da quantidade de nós que realmente precisam transmitir quadros, o mestre é obrigado a perguntar a todos os nós se eles querem transmitir. Se o nó mestre falhar, o sistema inteiro falha.

No **token-passing**, um quadro especial (token) é repassado entre todos os nós da rede (e.g. num **token ring network**). Quem detiver o token pode transmitir um certo número de quadros e depois tem de repassá-lo. É introduzido outro delay: em redes com grande perímetro, haverá delay significativo para que cada nó possa começar a transmitir, pois o token tem de ser repassado a cada nó da rede, um a um, até chegar no nó ativo que precise transmitir. Se algum nó falhar ou retiver o token, todo o sistema poderá falhar.

LANS

ENDEREÇAMENTO

LANs usam canais broadcast. No entanto, geralmente cada nó quer se comunicar com um nó específico, e não com toda a LAN. Para que isso seja possível, é preciso um **endereço de LAN**.

O endereço de LAN tem 48 bits e é expresso em seis pares de dígitos hexadecimais. Os endereços são únicos e fixos a cada interface, e não há hierarquia. Fabricantes compram lotes de endereços diretamente do IEEE, evitando assim duplicação de endereços.

ARP (ADDRESS RESOLUTION PROTOCOL)

Traduz **endereços IP** em **endereços LAN**, sendo análogo ao DNS. Assim, quando um nó A gera um datagrama destinado ao nó B na mesma LAN, o adaptador do nó A irá buscar na tabela ARP o endereço LAN referente ao endereço IP de destino do datagrama, e colocará o datagrama dentro do payload do quadro endereçado a este endereço LAN.

Caso o destino do datagrama esteja fora da LAN, o ARP será usado para traduzir não o endereço IP de destino final, mas o endereço IP do *next router*, e assim sucessivamente até que o quadro chegue ao destino final.

COMO A TABELA É FORMADA

Ao receber um datagrama , o adaptador percorre sua tabela ARP à procura do endereço LAN referente ao endereço IP de destino desse datagrama.

Se ele não o encontrar, ele gera um **ARP query packet**, que é um quadro com destino a todos os nós da LAN (endereço broadcast FF-FF-FF-FF-FF-FF) e que contém o endereço IP procurado.

Eventualmente, o adaptador com o endereço IP procurado vai receber o ARP query packet e vai enviá-lo à camada de rede de seu nó-pai. Lá, o nó vai checar se o endereço IP do adaptador é o mesmo endereço IP de destino do ARP query. Se os dois são iguais, então este adaptador é o que está sendo procurado, e o adaptador envia um **ARP response packet** de volta ao emissor do query packet, que então atualiza sua tabela ARP com a nova entrada e enfim consegue enviar o quadro ao host correto.

ETHERNET

As tecnologias Ethernet são não-orientadas a conexão (quadros são simplesmente enviados, sem setup prévio) e fornecem um serviço sem confiabilidade (há detecção de erro, mas não há envio de ACKs confirmando o recebimento correto).

A confiabilidade não é necessária no Ethernet, pois pode ser implementada num nível mais alto (transporte – TCP).

Ethernet tem padrões com velocidades entre 10 Mbps e 10 Gbps, e topologias em barramento ou estrela.

O protocolo de acesso múltiplo do Ethernet é o **CSMA/CD.** Como o tempo de propagação no enlace geralmente é muito pequeno, a eficiência é bastante alta:

$$eficiência = \frac{1}{1 + 5t_{propagação}/t_{transmissão}}$$

QUADRO ETHERNET

A estrutura do quadro Ethernet é igual em todas as versões.

preâmbulo	end. Destino	end. Origem	tipo	dados	CRC
8 bytes	6 bytes	6 bytes	2 bytes	46 a 1500 bytes	4 bytes

- **Preâmbulo**: sete bytes consecutivos com valor 10101010 seguidos por um byte com valor 10101011. Serve para “acordar” o receptor e sincronizar seu clock com o do emissor (lembrando que há vários padrões diferentes com diferentes frequências);
- **Endereços** destino e origem: endereços LAN de 6 bytes;
- **Tipo**: a qual protocolo de rede deve se entregar os dados (e.g. IP, IPX, ARP, etc). Faz a multiplexação, análoga às portas da camada transporte e ao campo “protocol” do IP;

- **Dados**: os dados em si, e.g. um datagrama IP. O MTU do Ethernet é de 1500 bytes.
- **CRC**: 4 bytes de CRC = 4*2^8 bits = 1024; logo o CRC é capaz de detectar erros de até 1024 bits consecutivos.

10BASE2

Uso de **cabo coaxial** e topologia de **barramento**: todos os nós estão conectados diretamente a um cabo coaxial, cujas pontas são isoladas com **terminators**.

Limitações: 10Mbps, 30 nós por segmento, cada segmento com 185 metros, máximo de 5 segmentos.

10BASET / 100BASET

Uso de **par trançado** (fios de cobre) ou **fibra óptica**. Topologia **estrela**: hub (ou switch) central ao qual se conectam os nós da LAN.

Não usam Manchester encoding.

GIGABIT / 10GBPS

802.3z, fibra óptica ou par trançado.

HUBS, BRIDGES E SWITCHES

HUBS

É um mero repetidor de bits: ao receber um pacote numa interface, replica todos seus bits e envia as cópias às outras interfaces, **sem usar protocolos da camada enlace como CSMA/CD**. Portanto, hubs são dispositivos **da camada física**. Todos as interfaces conectadas tem de usar a mesma tecnologia Ethernet.

Todos os nós conectados a um hub fazem parte de um mesmo **domínio de colisão**. Quanto maior o domínio de colisão, maior a ineficiência da LAN, pois pacotes são replicados sem necessidade, gerando tráfego desnecessário.

BRIDGE

Ao contrário do hub, o bridge é um dispositivo verdadeiramente da **camada enlace**. Ele envia quadros **seletivamente**, descartando-os quando necessário. Dessa forma, cada interface de um bridge forma um domínio de colisão isolado, evitando os desperdícios do hub.

Bridges encaminham quadros de maneira seletiva com um auxílio de uma **tabela de bridge**, que relaciona interfaces do bridge a endereços LAN. Portanto, ao receber um quadro, o bridge verifica qual a interface de saída apropriada para enviar o quadro. Se a interface encontrada for diferente da interface pela qual o quadro chegou ao bridge, o quadro é enviado. Caso as interfaces de entrada e saída sejam iguais, isso quer dizer que o quadro já foi devidamente propagado no domínio de colisão adequado, e o bridge não precisa mais fazer nada; resta-lhe descartar o quadro.

Ao contrário dos hubs, bridges usam o protocolo CSMA/CD. Bridges são usados também para conectar segmentos Ethernet que usam diferentes tecnologias (e.g. 10BT e Gigabit).

Bridges também previnem erros que os hubs não enxergam, como **jabbering** de algum nó da LAN, desativando o segmento necessário.

TABELA BRIDGE

A tabela bridge é do tipo “plug-and-play” e “self-learning”.

Consideremos uma tabela inicialmente vazia. Ao receber um quadro, o bridge não saberá exatamente para qual interface deverá encaminhá-lo. Portanto, o bridge encaminha o quadro para *todas as demais interfaces*. Eventualmente a resposta passará pelo bridge, e ele registrará o número da interface ao lado do endereço LAN de origem do pacote.

Assim, cada nó que envie um quadro que passe pelo bridge vai ficar registrado na tabela bridge.

SWITCH

São bridges que operam em **full-duplex** e tem um número maior de interfaces, cada qual podendo ser de uma tecnologia Ethernet distinta.

Switches obtém transmissão full-duplex pois tem conexões ponto-a-ponto entre o switch e cada nó (topologia estrela). Assim, não há perigo de colisão em cada conexão, pois a conexão é dedicada; desabilita-se então o protocolo de acesso múltiplo e obtém-se conexão full-duplex, com uso simultâneo tanto do par de fios de cobre upstream quanto o par de downstream.