

Questão 01)

O protocolo de acesso ao meio utilizado nas redes Ethernet é o CSMA/CD. Esse protocolo, é um protocolo de acesso aleatório, ou seja, um nó transmissor sempre transmite à taxa total do canal e quando ocorrem colisões o nó transmissor, após esperar um tempo aleatório, volta a retransmitir o pacote até que ele seja transmitido pelo canal sem haver colisões.

Nesse protocolo, para reduzir o número de colisões os seguintes mecanismos são utilizados. Primeiro, um nó ouve o canal antes de transmitir para ter certeza de que não existe nenhum nó, naquele momento, transmitindo (detecção de portadora). Caso tenha algum nó transmitindo no momento em que o nó em questão está escutando, ele espera até q o nó que está transmitindo termine para que ele possa transmitir. Ou seja, o nó ouve o canal antes de transmitir e só começa a transmitir depois de um período sem detectar transmissões.

Depois disso, quando um nó está transmitindo e ouvindo o canal enquanto transmite percebe que existe outro nó transmitindo (detecção de colisão), ele para a transmissão e espera por um tempo aleatório antes de voltar a transmitir.

Porém, existe a grande possibilidade dos mesmos nós colidirem novamente. Isso ocorre, pois um nó pode escutar o canal e não detectar nenhuma transmissão naquele momento e começar a transmitir, e, uma transmissão que já havia sido feita e ainda não tinha chegado nesse nó pode colidir com a transmissão que foi feita agora. Quanto maior for o atraso de propagação do canal de fusão maiores as chances de haverem colisões, já que o canal fica ocupado por mais tempo.

Para solucionar essa questão, no protocolo CSMA/CD é utilizado o algoritmo de recuo exponencial binário. Esse algoritmo faz com que o tempo de espera de um nó seja probabilisticamente maior de acordo com o número de colisões que ele já tenha experimentado. Probabilisticamente pois os tempos de espera dos nós são aleatórios, porém, nesse algoritmo, ele aumenta a quantidade de tempos de bits a serem sorteados para a espera daquele nó. Então, quanto mais colisões o nó tenha experimentado, maiores as chances de esperar por tempos maiores.

Questão 02)

- a) No algoritmo de estado de enlace, um roteador envia informações de roteamento para todos os outros roteadores por difusão. Já no algoritmo de vetor de distância, um roteador envia informações de roteamento apenas para os roteadores vizinhos (roteadores diretamente conectados).
- b) No algoritmo de estado de enlace, as informações de roteamento enviadas pelos roteadores são os custos dos enlaces ligados diretamente a ele. Já no algoritmo de vetor de distância, as informações de roteamento enviadas pelos roteadores são as estimativas de menor custo entre o próprio roteador que mandou a mensagem e todos os outros roteadores da rede (ou todos que ele sabe que existe).
- c) O problema de contagem até o infinito é um problema do algoritmo de vetor de distância. Para explicar o problema de contagem até o infinito, consideremos o seguinte exemplo: considere três roteadores X, Y e Z, onde X se conecta diretamente a Y e Z, Y se conecta diretamente a X e Z, e Z se conecta diretamente a X e Y. O custo do enlace de X até Y é inicialmente 1, o custo do enlace de Y até Z é inicialmente 2 e o custo do enlace de Z até X de inicialmente 25. Então, inicialmente, a distância mínima de Y para X é 1, de Y para Z 2, de Z para Y 2 e de Z para X 3. Agora, considere que o custo do enlace de X até Y mude de 1 para 20. Y vai calcular sua nova distância mínima até X, considerando que Z vai até X com um custo de 3 (por não saber que essa distância foi calculada passando pelo Y), logo, o resultado

achado vai ser 5 (a distância de Y até Z mais a distância de Z até X). Após isso, um loop de roteamento é acionado, já que um pacote direcionado a X que chegou em Y ou Z vai ficar indo de um para o outro até que as tabelas de repasse sejam mudadas, ou, pior, pra sempre (daí o nome de problema de contagem até infinito). Continuando nossa ordem cronológica, após o Y calcular sua nova distância mínima até X, ele vai passar essa informação para o Z. Com isso, Z recebe a informação de que Y vai até X em 5 e, sabendo que Z vai até Y em 2, ele considera que sua distância mínima até X agora é 7 (2 de Z até Y mais 5 de Y até X) e assim sucessivamente até que Z finalmente calcule que a distância mínima dele até X seja 19 e Y perceba que a distância mínima dele até X é de 20 (caminho sem precisar passar por Z, já que o custo passando por Z é igual a 21). Para isso, serão necessárias 8 iterações.

Questão 03)

A janela de recepção (window) é um serviço de controle de fluxo do TCP que é usada para dar ao remetente uma ideia do espaço disponível no buffer do destinatário. A window do próximo segmento vai ser 3400 (4000 do buffer de recepção de B menos 600 de bytes de dados enviados de A para B + 0 dos bytes lidos anteriormente, 0 por que nenhum byte foi lido por B ainda).

O acknowledgement number (que é o início da próxima sequência de dados que ele espera receber, ou seja, o sequence number vindo de A após ser lidos todos os bytes de dados enviados por A) vai ser 1601 já que sequence number (que representa o início da sequência de dados enviados por A) vindo de A era 1001, ou seja, após ler um byte de dados, o sequence number passaria a ser 1002 e como foram enviados 600 bytes de dados, após ler todos os bytes, o sequence number de A seria igual a 1601. Seguindo a mesma lógica, o sequence number enviado vai ser o acknowledgement number recebido, ou seja 2020.

Como, em uma conexão TCP, o acknowledgement number de um é igual ao sequence number do outro e vice e versa (isso por que sempre são enviados um para o outro o início da sequência de dados enviados e o início da próxima sequência de dados que ele espera receber) então o acknowledgement number de A vai virar o sequence number de B, ou seja, 2020.

Questão 04)

Primeiro, para facilitar a explicação do problema, vou chamar de A o nó 111.111.111.111, que vai tentar contactar o servidor Web. Por sua vez, esse servidor Web está hospedado no nó 222.222.222.221, que eu vou chamar de B. Vamos chamar o nó onde o servidor DNS local é executado (111.111.111.112) de C. Um ponto importante a se destacar é que A só tem ciência do URL do servidor Web e que C contém o mapeamento entre URL e IP.

O primeiro passo após digitar a URL do servidor Web no navegador é utilizar o protocolo DHCP (camada de aplicação) para descobrir o endereço do servidor DNS local. Então A manda uma solicitação DHCP encapsulada em UDP, encapsulada no IP, encapsulado em Ethernet. O quadro Ethernet é difundido na LAN1 e é recebido pelo roteador que executa o servidor DHCP. Ethernet é demultiplexado para IP, UDP e DHCP.

Agora, o servidor DHCP vai preparar a resposta para A contendo o nome e endereço IP do servidor DNS e o endereço do roteador. Depois disso, ocorre um encapsulamento no servidor DHCP, o quadro é repassado através da LAN1 e há uma demultiplexação no cliente. Agora, o cliente DHCP recebe a resposta do server DHCP.

Depois disso, uma consulta DNS é criada, encapsulada em UDP, encapsulada no IP e encapsulada no Ethernet. Para enviar o quadro ao roteador, é preciso saber o endereço MAC da interface do

roteador, para isso, uma consulta ARP é difundida e recebida pelo roteador, que manda uma resposta pra A com o endereço MAC dele.

Sabendo o endereço MAC do roteador, A agora envia o quadro contendo a consulta DNS para o roteador. O datagrama IP contém a consulta DNS que vai ser encaminhada através do LAN1 de A até o roteador, esse datagrama IP é repassado para C (servidor DNS) e o C responde a A com o endereço IP da URL do servidor que A está tentando contactar.

Agora, sabendo o endereço IP de B, A vai enviar um pedido de conexão HTTP. E, para isso, A abre um socket TCP para o C. Após a conexão TCP ser estabelecida, a solicitação HTTP é enviada para o socket TCP, o datagrama IP que contém a solicitação HTTP é enviado para C. Agora, C responde a solicitação HTTP contendo a página web e o datagrama IP com a resposta HTTP é encaminhada de volta pra A.