

1. Porque não se deve usar um valor fixo para os temporizadores de uma conexão TCP? Comente, portanto, sobre como é escolhido o valor do temporizador de uma conexão TCP? (1,0 ponto)

Resposta: Não se deve usar um valor fixo pois para o temporizador, pois se um valor muito longo é usado, espera-se muito tempo para retransmitir um pacote perdido e, assim, o atraso de transmissão aumenta (0,3 pontos). Por outro lado, se o temporizador é curto, pacotes que podem estar apenas atrasados são considerados perdidos e retransmissões desnecessárias acontecem (0,2 pontos). O valor do temporizador é escolhido baseado na estimativa da média e do desvio do tempo de ida-e-volta (RTT – *round trip time*) dos segmentos e de seus respectivos reconhecimentos positivos (ACKs), desconsiderando as retransmissões, caso ocorram (0,5 pontos).

2. Suponha que um processo executa em um sistema final C e abre um socket UDP com número de porta 1234. Suponha que dois sistemas finais, A e B, enviem segmentos UDP para a porta de destino 1234 do sistema final C. Responda: Ambos os segmentos serão direcionados para o mesmo socket e mesmo processo no sistema final C? Se sua resposta é sim, como o processo que executa no sistema final C sabe que esses dois segmentos têm origem em dois hospedeiros diferentes? (0,5 pontos)

Resposta: Sim, no caso do protocolo UDP, os dois segmentos serão direcionados para o mesmo socket e mesmo processo no sistema final C (0,2 pontos). O processo diferencia a origem dos segmentos pelo endereço IP de origem (0,3 pontos).

3. O UDP também é chamado de protocolo de transporte mínimo. Por quê? Ele é realmente necessário na Internet? Porque algumas aplicações preferem usar UDP e não usar TCP? Justifique suas respostas e cite dois protocolos da camada de aplicação que o utilizam. (1,0 ponto)

Resposta: O UDP é chamado de protocolo de transporte mínimo porque só oferece os serviços mínimos da camada de transporte que são: multiplexação, demultiplexação e verificação de integridade (0,4 pontos). O UDP é necessário por possuir menor latência se comparado ao TCP, não manter estados no transmissor e receptor e por permitir que um transmissor possa enviar dados tão rápido quanto desejado e possível. Isso porque o UDP não exige estabelecimento de conexão, não

emprega mecanismos de controle de fluxo e congestionamento e seu cabeçalho tem tamanho reduzido (0,4 pontos). São exemplos de protocolos da camada de aplicação que o utilizam o DNS (*Domain Name System*), o NFS (*Network File System*), o SNMP (*Simple Management Network Protocol*), entre outros (0,1 pontos por cada protocolo citado corretamente – até 0,2 pontos).

4. Descreva o mecanismo de retransmissão rápida empregado pelo TCP. Por que esse mecanismo é necessário? (1,5 pontos)

Resposta: Durante uma transmissão na conexão TCP, alguns segmentos podem ser recebidos no destino fora de ordem. Esta desordem pode ter sido ocasionada pelo roteamento no percurso entre o hospedeiro de origem e o hospedeiro de destino, e não pela perda de um segmento enviado. Os ACKs relativos aos segmentos recebidos fora de ordem, são enviados, informando ao transmissor a sequência esperada pelo receptor (ou em outras palavras, o número de sequência do segmento faltante). Para evitar o acionamento do mecanismo para de “partida lenta”, o transmissor aguarda para ter a certeza de que o segmento foi realmente perdido (já que o segmento pode não ter sido perdido mas estar atrasado, devido ao roteamento). A certeza da necessidade de transmissão, no caso do TCP, é obtida de duas formas: ou pela recepção do terceiro ACK em duplicata ou pelo timeout no transmissor. No caso da recepção de três ACKs em duplicata o transmissor constata que segmentos estão chegando ao receptor (já que ACKs em duplicata estão sendo recebidos), e por essa razão além de retransmitir o segmento perdido (“retransmissão rápida”, isto é, o transmissor não espera a temporização do segmento), o transmissor, para uma melhor utilização da banda passante, aciona o mecanismo para “evitar congestionamento” e não o de “partida lenta”.

5. Descreva e diferencie os mecanismos de controle de fluxo e de congestionamento do TCP. (2,0 pontos)

Resposta: O objetivo do controle de fluxo é não sobrecarregar o receptor com mais dados do que ele pode receber. Por outro lado, o controle de congestionamento tem como objetivo inferir um congestionamento e não sobrecarregar ainda mais a rede nessa situação. Essa é a principal diferença entre os dois mecanismos (0,4 pontos). O controle de fluxo funciona da seguinte forma. O receptor anuncia o espaço livre em seu buffer para o transmissor através do campo janela de recepção (RcvWindow) presente no cabeçalho de cada segmento enviado. Ao receber essa informação, o transmissor limita sua janela de transmissão, ou seja, a quantidade de dados ainda não reconhecidos, ao tamanho informado no campo. Dessa forma, o receptor não é afogado pelo transmissor (0,8 pontos). O controle de congestionamento também limita a quantidade de dados ainda não reconhecidos, porém, de acordo com o nível de congestionamento da rede. Para tanto, perdas são inferidas através da recepção de três ACKs duplicados e pelo estouro do temporizador de espera de um ACK. Nos dois casos, a medida tomada após o

evento de perda é reduzir a janela de congestionamento pela metade ou para 1 segmento e redefinir o valor do `ssthresh`. Também são definidas diferentes formas de crescimento da janela em função do valor atual da janela de congestionamento: partida lenta (crescimento exponencial) e prevenção de congestionamento (crescimento linear) (0,8 pontos).

6. Um sistema final A envia para um sistema final B três segmentos, um logo após o outro, em uma mesma conexão TCP. Os dois primeiros têm tamanho igual a 70 bytes e o terceiro igual a 50 bytes. Assuma que o segundo é perdido e os demais são recebidos por B. Defina possíveis números de sequência dos segmentos enviados de A para B e dos reconhecimentos enviados de B para A (0,5 pontos).

Resposta: Supondo que o número de sequência do primeiro segmento enviado por A é n , os números de sequência dos segmentos seguintes serão $n+70$ e $n+140$. Por exemplo, se $n = 0$, logo o número de sequência dos segmentos enviados por A será 0, 70 e 140. Ao receber o primeiro segmento, B enviará um ACK com número de reconhecimento $n+70$, que é o próximo byte esperado por ele. O segundo segmento é perdido e não será recebido. Ao receber o terceiro segmento, B enviará novamente um ACK com número de reconhecimento $n+70$ que continua sendo o próximo byte esperado por B. Se $n=0$, por exemplo, os dois ACKs enviados por B têm número de reconhecimento igual a 70.

7. Sobre os processos de abertura e encerramento de conexão do TCP, diga se cada uma das afirmativas a seguir é VERDADEIRA ou FALSA. Justifique sua resposta. (0,5 pontos por item)

- a. O mecanismo de abertura de conexão do TCP é composto por 3 etapas, por isso a denominação de *three-way handshake*.

Resposta: A afirmativa é VERDADEIRA (0,2 pontos). O cliente TCP inicia o processo de abertura de conexão enviando um segmento SYN para o servidor TCP. O servidor responde com um segmento SYN+ACK e, por fim, o cliente TCP confirma a abertura de conexão enviando um ACK para o servidor, após receber o SYN+ACK (0,3 pontos).

- b. Na primeira etapa do processo de abertura de conexão, o cliente TCP envia para o servidor TCP um segmento de controle SYN que parâmetros de configuração e contém dados.

Resposta: A afirmativa é FALSA (0,2 pontos). Na primeira etapa, o cliente TCP envia para o servidor TCP um segmento de controle SYN, especificando, por exemplo, o número de sequência inicial no sentido cliente-servidor. Porém, não são enviados dados neste segmento, uma vez que o servidor TCP ainda não confirmou a abertura de conexão (0,3 pontos).

- c. Um cliente TCP envia um segmento SYN+ACK para confirmar a abertura da conexão.

Resposta: A afirmativa é FALSA (0,2 pontos). O servidor TCP é quem envia para o cliente um segmento de controle SYN+ACK sinalizando que aceita abrir uma conexão e que alocou espaço em seu *buffer* (0,3 pontos).

- d. Uma conexão é sempre encerrada pelo servidor TCP que envia apenas um segmento FIN para o cliente TCP e libera imediatamente os recursos alocados para essa conexão.

Resposta: A afirmativa é FALSA (0,2 pontos). Tipicamente, um cliente TCP inicia o processo de encerramento de conexão enviando para o servidor TCP um segmento FIN. O servidor, então, responde enviando um segmento ACK para confirmar o encerramento da conexão. Logo em seguida, o servidor também envia um segmento FIN e entra no estado de espera temporizada até receber um ACK do cliente. Só então os recursos são liberados (0,3 pontos).

8. Sobre números de sequência e de reconhecimento do TCP, diga se cada uma das afirmativas a seguir é VERDADEIRA ou FALSA. Justifique sua resposta. (1,5 pontos)

- a. Um sistema final A envia para um sistema final B um arquivo de 10 GB em uma conexão TCP. Nessa situação, se o número de sequência para um segmento dessa conexão for n , então o número de sequência para o segmento subsequente será necessariamente $n+1$.

Resposta: FALSO. O protocolo TCP é orientado a byte.

- b. Suponha que um sistema final A envie a um sistema final B, por uma conexão TCP, um segmento contendo 16 bytes de dados e com número de sequência 60, nesse mesmo segmento, o número contido no campo de confirmação é obrigatoriamente 76?

Resposta: FALSO. O campo de confirmação refere-se ao próximo número de sequência esperado.

- c. Um sistema final A envia para um sistema final B dois segmentos, um logo após o outro, em uma mesma conexão TCP. O primeiro tem número de sequência 100 e o segundo 120. Se o primeiro for perdido e o segundo recebido, o número de reconhecimento enviado no ACK de B para A é 100.

Resposta: VERDADEIRO. O campo de confirmação refere-se ao próximo número de sequência esperado.