

UFPE – Centro de Informática – Infraestrutura de Comunicação
3ª Prova – 2020-3

Nome do(a) aluno(a): Gabriel Nogueira Leite
Número de Matrícula/CPF: 398.068.608-61

1ª Questão – 1,5 Pontos

Analise as seguintes afirmativas:

- I. No controle de congestionamento do TCP Reno, quando 3 ACKs duplicados são recebidos pelo emissor, a janela de controle de congestionamento é setada para 1 MSS.
- II. A utilização (U) de um canal de comunicação ao ser utilizado exclusivamente para uma transferência de dados com o *Selective Repeat* é proporcional ao tamanho w da janela. Com esse protocolo e supondo o uso dos *números de sequência de 1 a 5*, o tamanho da janela para se buscar a maior utilização possível e, ao mesmo tempo, o correto funcionamento do protocolo é igual a 2.
- III. Suponha uma rede reservada exclusivamente para teste dos protocolos GoBack-N e Selective Repeat, estando ambos configurados de forma correta e garantindo-se as mesmas condições para ambos. Testa-se inicialmente o desempenho do GoBack-N transferindo 1000 segmentos de 500 bytes com janela comportando 10 segmentos. Em seguida, testa-se o desempenho do Selective Repeat transferindo os mesmos 1000 segmentos de 500 bytes com janela comportando 10 segmentos. Supondo o cenário onde não ocorra desordenação de pacotes (dados e ACKs), não ocorra corrupção de pacotes (dados e ACKs), não ocorra nenhum *timeout* e nem perdas de pacotes (dados e ACKs), o desempenho de transferência de dados do Selective Repeat é menor do que o desempenho do GoBack-N.

Para cada afirmativa, diga se ela é verdadeira ou falsa e explique o porquê.

- I. **Falso.** Se tratando de TCP reno, a janela é ajustada para $cwnd = 6 \cdot MSS$, após isso seu crescimento é linear.
- II. **Verdadeiro.** O número de sequência deve ser menor ou igual a metade do número de números de sequência. Além disso deve ser uma potência de 2.
- III. **Falso.** O correto é afirmar que o desempenho de transferência de dados do Selective Repeat é o mesmo desempenho do GoBack-N ($N / (1+2^a)$).

2ª Questão – 1,5 Pontos

Analise as seguintes afirmativas:

- I. No TCP, o RTT estimado é dado por $EstimatedRTT = (1 - \alpha) * EstimatedRTT + \alpha * SampleRTT$. Sabe-se que o TCP adota α igual a 0,125. Isso permite sempre atribuir um peso maior para amostras passadas visto que $(1 - \alpha)$ é maior que α .
- II. O *Go-Back-N* utiliza o conceito de ACK cumulativo. Nesse protocolo, ACK (N) confirma todos os bytes recebidos até o byte N-1.

- III. Um socket UDP é identificado pela tupla {Endereço IP de Origem, Número da Porta de Origem}.

Para cada afirmativa, diga se ela é verdadeira ou falsa e explique o porquê.

I.**Falso.** Nesse caso temos que a fórmula atribui um peso maior para as amostras mais recentes. Isso ocorre pois amostras mais recentes representam de uma melhor forma o estado atual de congestionamento da rede.

II.**Falso.** Temos que a confirmação é feita em todos os pacotes com número de sequência até N.

III.**Falso.** Um socket UDP é identificado por um número de porta de destino e um endereço IP de destino.

3ª Questão – 1,5 Pontos

Analise as seguintes afirmativas:

- I. É possível a troca bidirecional de dados entre dois processos em *hosts* diferentes quando um lado utiliza o TCP Reno e o outro o TCP Tahoe.
- II. Todo protocolo de transporte requer controle de fluxo para evitar a sobrecarga de dados no receptor.
- III. O tamanho da janela de envio de segmentos com dados no TCP é limitado pelo valor máximo entre o tamanho da janela de controle de congestionamento do emissor e o espaço disponível conhecido da janela de recepção do receptor.

Para cada afirmativa, diga se ela é verdadeira ou falsa e explique o porquê.

I.**Falso.** O TCP Reno e TCP Tahoe possuem diferentes controles de congestionamento, por conta disso eles não poderão trocar informações de forma bidirecional.

II.**Verdadeiro.** O controle de fluxo permite controlar a quantidade de dados no receptor. Portanto, se o objetivo é evitar a sobrecarga, deve haver um controle de fluxo no protocolo de transporte.

III.**Falso.** O tamanho da janela de envio de segmentos com dados no TCP é limitado pelo valor mínimo.

4ª Questão – 1,5 Pontos

Analise as afirmativas sobre o TCP em cada item a seguir:

- I. Considere 1 fluxo UDP que consome 10% da banda passante R de um enlace de gargalo. Supondo que k sessões TCP passem a competir nesse mesmo enlace no qual o fluxo UDP está sempre presente, cada sessão TCP deve ter uma taxa média de R/k .
- II. O overhead do cabeçalho do segmento UDP é de ao menos 8 bytes enquanto o cabeçalho do segmento do TCP é de ao menos 20 bytes.

- III. Um socket TCP é identificado pela tupla quádrupla {Endereço IP de Origem, Número da Porta de Origem, Endereço IP de Destino, Número de Porta de Destino}. Todas essas informações estão contidas no cabeçalho de cada segmento TCP para que o host receptor entregue o segmento ao socket correto.

Para cada item analisado, explique se ele é verdadeiro ou falso.

I.**Falso.** O valor deveria ser de $(0,9 \cdot R)/k$.

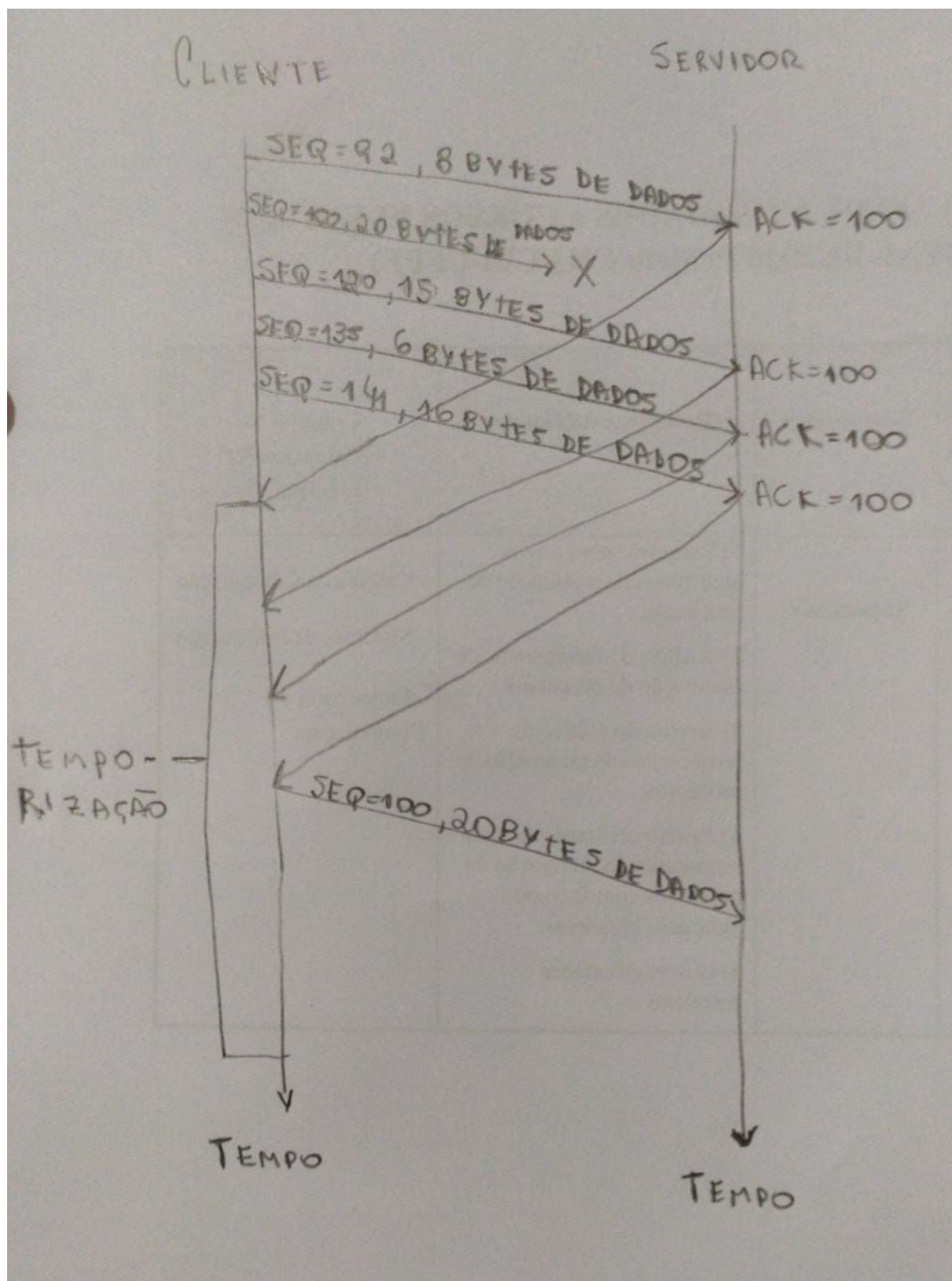
II.**Falso.** Segundo Kurose o UDP tem tipicamente 8 bytes, mas não ao menos 8. Bem como o TCP tem tipicamente 20 bytes, mas não ao menos 20 (Kurose, página 171).

III.**Falso.** Ao invés da tupla quádrupla, temos o número de sequência e de reconhecimento.

5ª Questão – 3,0 Pontos

O TCP utiliza *timeout* e contagem de ACKs duplicados como indicativos de perda de segmento. Explique porque é razoável utilizar ACKs duplicados para tal fim. Apresente um desenho de auxílio à sua explicação, mostrando a transferência de segmentos TCP entre um cliente e um servidor.

Resposta: Para responder essa questão iremos primeiro definir o que é um ACK duplicado, ACK duplicado é um ACK que reconhece novamente um segmento para o qual o remetente já recebeu um reconhecimento anterior. Cada pacote possui um número de sequência, quando um TCP destinatário recebe um segmento com número de sequência maior do que o próximo na sequência, ele detecta uma lacuna no fluxo de dados, identificando que um pacote está faltando. Usualmente um remetente envia um alto número de segmentos, um seguido do outro, caso um segmento seja perdido, provavelmente irão existir muitos ACKs duplicados, também um seguido do outro. Se o TCP remetente receber três ACKs duplicados para os mesmos dados, ele identifica com isso que o segmento seguinte ao segmento reconhecido três vezes foi perdido. No caso de receber três ACKs duplicados, o TCP remetente realiza uma retransmissão rápida, retransmitindo o segmento que falta antes da expiração do temporizador do segmento. Segue abaixo o desenho de auxílio.



6ª Questão – 1,0 Ponto

Explique porque setar o temporizador de expiração do TCP para $\text{TimeoutInterval} = 2 * \text{EstimatedRTT}$ se mostra inadequado.

Resposta: Temos que o $EstimatedRTT = (1 - \alpha) * EstimatedRTT + \alpha * SampleRTT$, já o $SampleRTT$ é o tempo transcorrido entre o momento em que o segmento é enviado e o momento em que é recebido um reconhecimento para ele. Tendo isso em vista precisamos determinar uma relação para determinar a temporização de retransmissão do TCP, esse intervalo deve ser maior ou igual ao $EstimatedRTT$, pois, caso contrário, seriam enviadas retransmissões desnecessárias. Queremos que nossa temporização de retransmissão não seja muito maior do que o $EstimatedRTT$, caso contrário quando um segmento fosse perdido, o TCP não o retransmitiria rápido, se usarmos a relação citada no enunciado de $TimeoutInterval = 2 * EstimatedRTT$, teremos esse problema lentidão na retransmissão, por ser sempre um valor fixo, o que causaria grandes atrasos. Logo o valor ideal seria o $EstimatedRTT$ mais determinada margem, que deverá ser maior quando houver muita variação nos valores de $SampleRTT$ e menor quando houver pouca variação. Dessa forma usaremos outro parâmetro, o $DevRTT$. Dessa forma a relação ideal seria a seguinte, $TimeoutInterval = EstimatedRTT + 4 * DevRTT$.

Boa Sorte!