



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Redes de Computadores I**

**Gabarito AP2 - 2º semestre de 2013.**

**Aluno:** \_\_\_\_\_

**Assinatura:** \_\_\_\_\_

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
3. Você pode usar lápis para responder as questões.
4. Ao final da prova devolva as folhas de questões e as de respostas.
5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.

1. O UDP (*User Datagram Protocol*) também é chamado de protocolo de transporte mínimo. Por quê? Ele é realmente necessário na Internet? 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).

2. Sobre os protocolos de transferência confiável responda:
  - a. Qual a principal deficiência de protocolos que operam no modo para-e-espera (*stop and wait*)? (0,3 pontos)

Resposta: A principal deficiência é a baixa utilização do meio de transmissão, ou seja, tais protocolos limitam o uso dos recursos físicos. Isso porque um segmento só pode ser enviado pelo transmissor após a recepção do reconhecimento positivo

do segmento anterior enviado por esse transmissor. Enquanto isso não ocorre, o transmissor fica ocioso (0,3 pontos).

b. Qual a solução adotada por alguns protocolos para contornar esse problema? (0,2 pontos)

Resposta: A solução é permitir que um transmissor envie um conjunto de segmentos consecutivamente sem que antes seja necessário receber a confirmação de recepção dos segmentos anteriores dentro deste conjunto. Essa técnica é chamada de paralelismo (*pipelining*) (0,2 pontos).

c. Quais são as duas principais modificações que os protocolos que empregam a solução do Item b devem adotar? (0,5 pontos)

Resposta: As duas principais modificações são: (i) aumento da faixa dos números de sequência (0,3 pontos) e (ii) emprego de *buffers* no transmissor e/ou no receptor (0,2 pontos).

3. Descreva sucintamente o mecanismo de abertura de conexão usado pelo TCP. (1,0 ponto)

Resposta: O mecanismo de abertura de conexão do TCP é composto por 3 etapas, por isso a denominação de *three-way handshake* (0,1 ponto). Na primeira etapa, o cliente TCP envia para o servidor TCP um segmento de controle SYN, especificando o número de sequência inicial no sentido cliente-servidor, entre outros parâmetros de configuração. Não são enviados dados neste segmento (0,3 pontos). Na segunda etapa, após receber o SYN, o servidor envia para o cliente um segmento de controle SYN+ACK sinalizando que aceita abrir uma conexão e que alocou espaço em seu *buffer*. Nesse segmento, é especificado, por exemplo, o número de sequência inicial no sentido servidor-cliente (0,3 pontos). Na terceira etapa, após receber o SYN+ACK, o cliente envia um segmento de controle ACK, confirmando, assim, a abertura da conexão. Este último segmento pode conter dados (0,3 pontos).

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

Resposta: O mecanismo de retransmissão rápida funciona da seguinte maneira. Um transmissor ao receber três ACKs duplicados, ou seja, segmentos com o mesmo número de reconhecimento, infere que houve uma perda do segmento correspondente e o reenvia. Dessa forma, não é preciso esperar até o estouro do temporizador para inferir uma perda de segmento e reenviá-lo (0,8 pontos). Nesse caso, a janela de congestionamento é reduzida de forma menos agressiva, ou seja, para metade do seu valor antes do evento de perda e o *ssthresh* assume este mesmo valor. Assim, o mecanismo de controle de congestionamento entra na fase de prevenção de congestionamento e não na fase de partida lenta (0,2 pontos). Esse comportamento menos agressivo é adotado porque, nesse caso, se assume

que o nível de congestionamento da rede é mais baixo uma vez que o receptor está enviando ACKs, mesmo que duplicados, para o transmissor, ou seja, o receptor está recebendo alguns segmentos. A retransmissão rápida é necessária para que o transmissor detecte mais rápido um evento de perda, uma vez que o valor do temporizador para recepção de um ACK é baseado o tempo de ida-e-volta (*round-trip time* - RTT) dos segmentos (0,5 pontos). O RTT varia ao longo do tempo em função do nível de ocupação dos *buffers* dos roteadores no caminho entre o transmissor e o receptor, da rota entre eles, entre outros fatores. É possível esperar demasiadamente para reenviar um segmento perdido, pois o estado atual da rede não é refletido imediatamente no cálculo do temporizador.

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. Sobre números de sequência e de reconhecimento do TCP, diga se cada uma das afirmativas a seguir é VERDADEIRA ou FALSA. Justifique sua resposta. (0,5 pontos por item)
- a. Um sistema final A envia para um sistema final B um arquivo de 10 GB em uma conexão TCP. Nessa situação, o número de bytes não reconhecidos que A envia para B não pode exceder o tamanho do *buffer* de recepção de B.

Resposta: VERDADEIRA (0,2 pontos). Esse é o papel do controle de fluxo: evitar que o transmissor envie uma quantidade de bytes maior do que o espaço disponível no *buffer* de recepção do receptor (0,3 pontos).

- b. 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: FALSA (0,2 pontos). O TCP é orientado a bytes, ou seja, o número de sequência representa o primeiro byte de dados de um segmento (0,3 pontos).

- 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 80 e o segundo 100. Se o primeiro for perdido e o segundo recebido, o número de reconhecimento enviado no ACK de B para A é 100.

Resposta: FALSA (0,2 pontos). O ACK terá o número de reconhecimento 80, que é o número de sequência do próximo byte esperado por ele, uma vez que o segmento com esse byte foi perdido (0,3 pontos).

7. O protocolo TCP experimenta o comportamento mostrado no gráfico abaixo. Com base nesse gráfico, responda às seguintes perguntas e justifique resumidamente suas respostas. (0,4 pontos cada item)

- a. Qual(is) o(s) intervalo(s) de tempo em que a partida lenta do TCP está em execução?

Resposta: A partida lenta é executada nos intervalos [1,6] e [11,15] (0,2 pontos), pois nesses intervalos o crescimento da janela de congestionamento é exponencial (0,2 pontos).

- b. Qual(is) o(s) intervalo(s) de tempo em que a prevenção de congestionamento do TCP está em execução?

Resposta: A prevenção de congestionamento é executada nos intervalos [6,10], [15,18] e [19,23] (0,2 pontos), pois nesses intervalos o crescimento da janela de congestionamento é linear (0,2 pontos).

- c. Após a 10ª rodada de transmissão, a perda de segmento será detectada por três ACKs duplicados ou por um estouro do temporizador?

Resposta: Neste caso, a perda será detectada por estouro do temporizador (0,2 pontos), pois a janela de congestionamento foi reduzida para apenas 1 segmento (0,2 pontos).

- d. Após a 18ª rodada de transmissão, a perda de segmento será detectada por três ACKs duplicados ou por um estouro do temporizador?

Resposta: Neste caso, a perda será detectada por três ACKs duplicados (0,2 pontos), pois a janela de congestionamento foi reduzida para metade do seu

valor antes do evento de perda, ou seja, de 20 segmentos na 18ª rodada para 10 segmentos na 19ª rodada (0,2 pontos).

- e. Qual o valor do limiar de partida lenta (*ssthresh*) na 13ª rodada e na 21ª rodada?

Resposta: Nos dois casos, o valor do limiar *ssthresh* é definido como metade do valor da janela de congestionamento antes do último evento de perda. Portanto, na 13ª rodada, o valor do *ssthresh* é 18 (0,2 pontos) e na 21ª é 10 (0,2 pontos).

