



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Redes de Computadores I

AD2 - 2º semestre de 2009.

Gabarito

-
1. Por que? Não teria sido suficiente deixar que os processos de usuário enviassem pacotes diretamente para a camada de rede? **(1,0 ponto)**

Resposta:

Não seria suficiente. O UDP foi criado, na camada de transportes, para oferecer às aplicações um serviço não orientado à conexão, disponibilizando para elas o serviço do “melhor esforço” da camada de rede. A utilização direta da camada de rede subverteria a pilha de protocolos da Internet, pois a camada de rede é responsável pela transferência dos dados entre dois hospedeiros (origem e destino da transferência). Já camada de transporte é responsável pela transferência de dados entre aplicações distribuídas, realizando para tal as tarefas de multiplexação, demultiplexação (e, no caso do UDP, uma verificação de erros). De forma simplificada, a tarefa de entregar no hospedeiro de destino, os dados contidos em um segmento da camada de transporte à porta correta é denominada de **demultiplexação**. O trabalho de reunir no hospedeiro de origem, porções de dados provenientes de diferentes portas, encapsular cada porção de dados com informações de cabeçalho (que mais tarde serão usadas na demultiplexação) para criar segmentos, e passar esses segmentos para a camada de rede é denominado **multiplexação**.

2. Suponha que o receptor UDP, determina o *checksum* para segmento UDP recebido e verifica que ele é igual ao valor encontrado no campo *checksum* do cabeçalho UDP. Pode o receptor ter certeza absoluta de que não ocorreram erros em *bits* do segmento? Explique sua resposta. **(1,0 ponto)**

Resposta:

Não, o receptor não pode ter certeza absoluta de que não ocorreram erros em *bits* no segmento. Isto se deve a forma pela qual o *checksum* para o pacote é calculado. Se dois *bits* que ocupam a mesma posição em duas palavras de 16-*bits* do segmento são no momento da transmissão 0 e 1, e se ambos são modificados, respectivamente para 1 e 0 durante a transmissão, a soma desses dois *bits* permanece a mesma. Consequentemente, o complemento a 1 que o receptor calcula também será o mesmo. Isto significa que a detecção de erros falhou. O exemplo abaixo, com palavras de com dois *bits*, os dados no receptor chegaram errados (houve troca dos dois últimos *bits* de cada uma das palavras), mas a soma obtida é a mesma:

Transmissão: 01 e 10; soma: 11

Recepção (trocando os dois últimos *bits* de cada palavra): 00 e 11, soma 11

3. Imagine que no estabelecimento de uma conexão TCP, em vez do *handshake* de três vias, tenha sido empregado o *handshake* de duas vias. Em outras

palavras, a terceira mensagem não é usada. É possível que ocorra algum problema transferência confiável de dados? Forneça um exemplo do problema ou mostre que não existe qualquer problema em se usar o *handshake* de duas vias. (1,5 pontos)

Resposta:

Se o estabelecimento da conexão se der em duas vias é possível sim que que a transferência confiável de dados não ocorra. O ponto crucial do problema é a existência de duplicatas atrasadas na rede. Imagine a situação que cada pacote sofre temporização e é retransmitido duas ou três vezes. Alguns pacotes, tanto de controle como de dados, podem ficar retidos no núcleo da rede e emergir muito tempo depois. Lembremos que quando do estabelecimento da conexão cada lado da conexão começa com número de sequência diferente e ambos devem concordar em relação ao número de sequência escolhido por cada lado. Agora vamos ver como o *handshake* de três vias resolve o problema de pacotes de controle duplicados. Imagine que emerge no hospedeiro 2 (o servidor) uma duplicata do segmento SYN de uma antiga conexão. Essa duplicata chega ao hospedeiro 2 sem o conhecimento do hospedeiro 1 (o cliente). O servidor responde ao cliente com um segmento SYNACK, para verificar se o cliente deseja realmente estabelecer uma nova conexão. O cliente percebe que o número de sequência que está sendo confirmado corresponde a um segmento SYN antigo que ficou retido na rede e rejeita o estabelecimento da conexão. Sem a terceira via do estabelecimento da conexão, isto é o ACK do SYNACK, o servidor poderia receber segmentos de dados duplicados que ficaram retidos na rede, e emergiram após a duplicata do segmento de controle SYN recebida no hospedeiro 2, como mencionado anteriormente.

4. Suponha que o *round trip time* (RTT) entre o transmissor e o receptor TCP é constante e conhecido pelo transmissor. Assumindo que perdas de segmentos podem ocorrer, explique: Será necessário manter a temporização no protocolo rdt 3.0 apresentado no nosso livro texto? (1,0 ponto)

Resposta:

Sim, a temporização ainda será necessária no protocolo rdt 3.0. Se o RTT é conhecido a única vantagem é que o lado transmissor da comunicação saberá, com certeza, que o pacote ou o ACK (ou NACK) do pacote foi perdido, quando comparamos com o cenário real, no qual o pacote pode estar a caminho do lado transmissor, após a expiração do temporizador. No entanto, para detectar a perda, para cada pacote, um temporizador com duração constante será necessário no lado transmissor da comunicação.

5. Responda verdadeiro ou falso, explicando sua escolha: (0,5 pontos cada)

a) O tamanho de RcvWindow (definida na Seção 3.5 do seu livro texto) do TCP nunca muda enquanto dura a conexão.

Resposta: Falso. O campo "RcvWindow" indica a quantidade de *bytes* disponível no "*buffer* de recepção" no lado receptor de uma conexão TCP. O conteúdo desse campo é usado para evitar que o transmissor envie mais dados do que o receptor é capaz de receber, evitando assim o descarte de dados, por falta de espaço de armazenamento no lado receptor de uma conexão TCP. Portanto, ele varia durante a conexão.

b) Suponha que o hospedeiro A esteja enviando para o hospedeiro B um arquivo grande por meio de uma conexão TCP. O número de *bytes* não

reconhecidos que o hospedeiro A envia não pode exceder o tamanho do *buffer* de recepção do hospedeiro B.

Resposta: Verdadeiro. O controle de fluxo do TCP através do campo “RcvWindow” indica para o lado transmissor a sua capacidade de recepção de dados através do campo “RcvWindow” (vide o item a).

c) Suponha que o hospedeiro A esteja enviando para o hospedeiro B um arquivo grande por meio de uma conexão TCP. Se o número de sequência para um segmento transmitido nessa conexão é m , então o número de sequência para o segmento subsequente é $m+1$.

Resposta: Falso. No caso do TCP, o número de sequência do segmento subsequente seria: $m + k$, onde k é a quantidade de *bytes* de dados contidos no segmento anterior.

d) Imagine que o hospedeiro A envie ao hospedeiro B, por uma conexão TCP, um segmento contendo 4 *bytes* de dados e com número de sequência 38. Nesse mesmo segmento, o número contido no campo de confirmação é obrigatoriamente 42.

Resposta: Falso. Os valores iniciais para os números de sequência para cada um dos lados da conexão TCP são definidos durante o estabelecimento da conexão TCP (*three way handshake*), e cada uma das partes faz a escolha independentemente. Portanto o campo de confirmação, ao longo a transmissão na conexão TCP, reflete essa escolha inicial.

e) Considere o controle de congestionamento no TCP. Quando um temporizador expira no transmissor, o limiar (*threshold*) é ajustado para a metade do seu valor anterior.

Resposta: Verdadeiro. Esta é a estratégia adotada pelo mecanismo de controle do congestionamento do TCP. Além da redução do limiar à metade da janela de congestionamento anterior ao evento, a sua janela de congestionamento é reduzida para o tamanho de 1 MSS e a transmissão entra na fase de *Slow Start*. O caso de um temporizador expirar indica ao TCP que a rede se encontra congestionada, por isto uma medida mais cautelosa é adotada. Diferente do caso da recepção de três ACKs em duplicata, pois, neste caso, os pacotes de ACK ainda estão sendo recebidos no lado transmissor.

6. Sabemos que o TCP espera até receber três duplicatas de uma confirmação (ACK) para executar a fase de “retransmissão rápida”. Na sua opinião qual razão dos projetistas do TCP não terem decidido iniciar a “retransmissão rápida” após a recepção da segunda duplicata. **(1,0 ponto)**

Resposta: Vamos supor que os pacotes n , $n+1$ e $n+2$ foram enviados e o pacote n foi recebido e confirmado. Se os pacotes $n+1$ e $n+2$ foram reordenados no caminho (ou seja, foram recebidos na ordem $n+2$ e $n+1$), então o receptor do pacote $n+2$ gera uma confirmação duplicada para o pacote n . No caso de uma política que aguarda apenas a segunda confirmação em duplicata para retransmitir dados, o transmissor dispararia uma retransmissão. Na situação onde o transmissor espera pela terceira confirmação em duplicata, o receptor tem que receber dois pacotes corretamente após o pacote n , antes de receber o pacote $n+1$. Os projetistas provavelmente usaram a estratégia de esperar pela terceira confirmação em duplicata por acharem que esta teria o melhor

compromisso entre disparar a retransmissão rápida quando necessária e evitar a retransmissão prematura de pacotes no caso da reordenação de pacotes ao longo do caminho entre os dois hospedeiros.

7. Considere o protocolo Retorne a N (*Go-Back-N*) com tamanho de janela do transmissor igual a 3 e uma faixa de números de sequência de 1024. Suponha que no tempo t o pacote que o receptor está esperando, tenha o número de sequência k . Suponha também que o meio não reordene as mensagens. Responda: **(1,5 pontos cada)**

a) Quais são os possíveis de números de sequência dentro da janela do transmissor no tempo t ? Justifique sua resposta.

Resposta:

Temos nesse caso o tamanho da janela $N=3$. Suponha que o receptor tenha recebido o pacote $k-1$ e tenha confirmado todos os outros pacotes anteriores. Se todas essas confirmações foram recebidas no transmissor, então a janela do transmissor contém pacotes com números de sequência no intervalo $[k, k+N-1]$. Suponha agora que nenhuma dessas confirmações tenha sido recebida no transmissor. Nesse segundo, caso a janela do transmissor contém o pacote $k-1$, $k-2$ e $k-3$. Desse modo a janela do transmissor contém pacotes com números de sequência no intervalo $[k-N, k-1]$. Considerando essas duas situações, a janela do transmissor tem tamanho 3 e o pacote mais antigo contido na janela tem números de sequência contidos no intervalo $[k-N, k]$.

b) Quais os possíveis valores do campo ACK na mensagem que está correntemente se propagando de volta para o transmissor no tempo t ? Justifique sua resposta.

Resposta:

Se o receptor está aguardando pelo pacote com o número de sequência k , obrigatoriamente ele recebeu (e confirmou) o pacote com o número de sequência $k-1$ e os $N-1$ pacotes recebidos antes desse. Desse modo, se nenhum desses N ACKs tiverem sido recebidos no transmissor, então as mensagens de ACK com valores no intervalo $[k-N, k-1]$ ainda estão se propagando na rede em direção ao transmissor. Como o transmissor enviou os pacotes no intervalo $[k-N, k-1]$, necessariamente ele já recebeu o ACK para o pacote $k-N-1$. Como o receptor já enviou o ACK para o pacote $k-N-1$, ele não enviará confirmações que sejam inferiores a $k-N-1$. Assim a faixa de números das confirmações em trânsito vai de $k-N-1$ até $k-1$.