



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Redes de Computadores I

AD2 - 1º semestre de 2009.

Gabarito

-
1. Suponha que a camada de rede oferece o seguinte serviço para a camada de transporte: a camada de rede no *host* de origem aceita da camada de transporte, um segmento de no máximo 1.100 *bytes* e o endereço do *host* de destino. A camada de rede garante a entrega do segmento na camada transporte do *host* de destino. Suponha que diversos processos de aplicações de rede podem estar em execução no *host* de destino.

a. Descreva um protocolo simples para a camada de transporte, que receberá o dado da aplicação no *host* destino. Assuma que o sistema operacional no *host* destino atribui o número de porta de três *bytes* para cada processo de aplicação em execução. (0,5 pontos)

Resposta:

Chamemos este protocolo de Protocolo de Transporte Simples (PTS). No lado remetente, o PTS aceita do processo remetente um conjunto de dados que não excede a 1097 *bytes*, um endereço do *host* destino e um número de porta destino. O PTS acrescenta um cabeçalho de três *bytes* a cada conjunto de dados e põe o número da porta do processo de destino neste cabeçalho. O PTS repassa o endereço do *host* destino e o segmento resultante então à camada de rede. A camada de rede entrega o segmento ao PTS do *host* destino. O PTS examina o número da porta no segmento, extrai os dados do segmento e repassa os dados ao processo identificado pelo número da porta.

b. Modifique o protocolo de modo que ele forneça o “endereço de retorno” ao processo de destino. (0,5 pontos)

Resposta:

O segmento, neste caso, tem dois campos no cabeçalho: um campo de porta origem e outro para a porta destino. No lado de remetente, o PTS aceita do processo remetente um conjunto de dados que não excede a 1094 *bytes*, um endereço de *host* destino, um número da porta origem e um número da porta destino. O PTS cria um segmento que contém os dados de aplicação, número da porta origem e número da porta destino. O segmento e o endereço do *host* destino são repassados à camada de rede. Depois de receber o segmento, o PTS do *host* destino entrega ao processo de aplicação os dados e o número porta do processo origem. A camada de rede fornece o endereço do *host* de origem.

c. Nos seus protocolos a camada de transporte “precisa executar alguma tarefa” no núcleo da rede de computadores (isto é, nos roteadores)? (0,5 pontos)

Resposta:

Não, a camada de transporte não tem que fazer qualquer coisa no núcleo da rede. A camada de transporte “reside” nos *hosts*, ou seja, fora do núcleo da rede.

2. 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. 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 duplicata do segmento de controle SYN duplicado mencionado anteriormente.

3. Um processo no hospedeiro 1 foi atribuído à porta *p*, e um processo no hospedeiro 2 foi atribuído à porta *q*. É possível haver duas ou mais conexões TCP entre essas duas portas ao mesmo tempo? Explique. **(0,5 pontos)**

Resposta:

Não é possível. De modo diferente ao que ocorre com o *socket* do UDP, o *socket* TCP é identificado por uma 4-tupla <endereço IP de origem, porta de origem, endereço IP de destino, porta de destino>. No processo de demultiplexação, quando o segmento chega ao hospedeiro de destino, o TCP usa todos esses quatro valores para direcionar o segmento para o *socket* apropriado.

4. Suponha um processo que executa no hospedeiro C tem um *socket* com número de porta 6789. Suponha que dois hospedeiros A e B, enviem segmentos UDP para a porta de destino 6789 do hospedeiro C. Responda: ambos os segmentos serão direcionados para o mesmo *socket* no hospedeiro C? Se sua resposta é sim, como o processo que executa no hospedeiro C, sabe que esses dois segmentos têm origem em dois hospedeiros diferentes? **(0,5 pontos)**

Resposta:

Sim, ambos os segmentos serão direcionados para o mesmo *socket* no hospedeiro C. Para cada segmento recebido, na interface do *socket*, o sistema operacional disponibilizará para o processo o endereço IP de forma a determinar as origens dos segmentos individuais.

5. 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 pontos)

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.

6. Os hospedeiros A e B estão conectados diretamente por um enlace de 200 Mbps. Existe uma conexão TCP entre esses dois hospedeiros, e o hospedeiro A está enviando um grande arquivo para o hospedeiro B através dessa conexão. O hospedeiro A pode enviar dados da aplicação a taxa de 100Mbps porém no hospedeiro B o *buffer* de recepção TCP (*receive buffer*) é lido pela aplicação a taxa de 50Mbps. Descreva o efeito do controle de fluxo do TCP nesse cenário. (1,0 pontos)

Resposta:

O hospedeiro A envia dados para o *buffer* do receptor mais rapidamente do que o receptor do hospedeiro B pode retirar dados do *buffer*. O *buffer* de recepção em B rapidamente fica cheio devido a maior velocidade na entrada de dados que na retirada dos mesmos. Quando o *buffer* estiver cheio, o hospedeiro B sinaliza para A que pare de enviar dados atribuindo à janela de recepção o valor zero (RecWindow=0). O hospedeiro A para de enviar dados até que receba um valor para a janela de recepção superior a zero (RecWindow>0). Desta forma, o hospedeiro A pára e inicia a transmissão de dados como uma função dos valores recebidos para a janela de recepção do hospedeiro B. Na média, ao longo do tempo, a taxa de transmissão na qual o hospedeiro A envia dados para o hospedeiro B, durante a conexão não será superior a 50Mbps, evitando que o lado transmissor suplante a capacidade de recepção do outro lado da conexão.

7. 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 pontos)

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.

8. Os hospedeiros A e B comunicam-se através de uma conexão TCP, e B já recebeu de A todos os *bytes* até o *byte* 248. Suponha que nessa situação, A envia dois segmentos para B. O primeiro e o segundo segmento contêm respectivamente 40 e 60 *bytes* de dados. No primeiro segmento o número de sequência é 249, o número da porta de origem é 503, e o número da porta de destino é 80. O hospedeiro B envia uma confirmação sempre que ele recebe um segmento de A.

a. No segundo segmento enviado de A para B, quais são os valores do número de sequência, do número da porta de origem e do o número da porta de destino? Justifique sua resposta. (1,0 ponto)

Resposta:

No segundo segmento do hospedeiro A para o B, o número de sequência é 289, a porta de origem é 503 e o número da porta de destino é 80. Isto se justifica porque o primeiro segmento tem 40 *bytes* e a sequência é acrescida deste valor, ou seja, passa de 249 para 289. As portas de origem e destino dependem da direção de envio dos pacotes, de A para B a origem é 503 e o destino é 80, na direção contrária (B para A) a porta de origem é 80 e destino 503, este mesmo raciocínio justifica a definição das portas origem e destino neste item bem como nos itens b e c.

b. Se o primeiro segmento chega antes do segundo segmento no destino, na confirmação enviada para o segmento, quais são os valores contidos nos campos de confirmação, de número da porta de origem e de o número da porta de destino? Justifique sua resposta. (1,0 ponto)

Resposta:

Se o primeiro segmento chega antes do segundo, na confirmação da recepção do primeiro o valor é 289, a porta de origem é 80 e o número da porta de destino é 503. A justificativa é que o TCP sempre confirma o último pacote recebido corretamente e na ordem esperada.

c. Se o segundo segmento chega antes do primeiro segmento no destino, na confirmação enviada para o segmento, quais são os valores contidos nos campos de confirmação, de número da porta de origem e de o número da porta de destino? Justifique sua resposta. (1,0 ponto)

Resposta:

Se o segundo segmento chega antes do primeiro, na confirmação da recepção o valor é 249, a porta de origem é 80 e o número da porta de destino é 503. A justificativa é que o TCP sempre confirma o último pacote recebido corretamente e na ordem esperada.