



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 2010.

Gabarito

-
1. É possível que uma aplicação desfrute da transferência confiável de dados mesmo quando executa sobre UDP? Caso a resposta seja afirmativa, como isso acontece? **(1,0 ponto)**

Resposta:

Sim, desde que seja implementado na própria aplicação o código do protocolo que realiza a transferência de dados confiável. Sem isso é impossível garantir a transferência de dados confiável, já que o UDP não oferece esse serviço.

2. Suponha que um servidor *Web* é executado no Computador C na porta 80. Esse servidor utiliza conexões persistentes e no momento, está recebendo solicitações de dois computadores diferentes: o Computador A e o Computador B. Todas as solicitações estão sendo enviadas através do mesmo *socket* no Computador C? Se as solicitações estão passando por diferentes *sockets*, podem ambos os *sockets* ter o número de porta 80? Discuta e explique. **(1,0 ponto)**

Resposta:

Para a primeira pergunta a resposta é não. Para cada conexão persistente, o servidor de *Web* cria um *socket* de conexão separado. Cada *socket* de conexão é identificado pela 4-tupla <endereço IP de origem, número de porta de origem, endereço IP de destino, número de porta de destino >. Quando hospedeiro C recebe um datagrama IP, ele examina esses quatro campos nos cabeçalhos das camadas de rede e de transporte (datagrama/segmento) para determinar para qual *socket* deve passar a carga útil (*payload*) do segmento TCP.

Respondendo a segunda pergunta, como vimos na anterior, cada *socket* de conexão é identificado por quatro informações: <endereço IP origem, número da porta de origem, endereço IP destino e número da porta destino>. Quando o hospedeiro C recebe um datagrama IP, este examina estes quatro campos no datagrama/segmento para determinar para qual *socket* o conteúdo de informação do segmento TCP deve ser repassado. Assim, as requisições de A e B vão para *sockets* diferentes. O identificador para ambos *sockets* tem 80 como porta de destino, porém, os identificadores para estes *sockets* têm valores diferentes para o endereço IP de origem. Diferentemente do que ocorre no UDP, quando a

camada de transporte repassa o conteúdo de informações de um segmento de TCP para o processo da aplicação, não especifica o IP de origem, pois isto é implicitamente especificado pelo identificador do *socket*.

3. Suponha que o *round trip time* (RTT, ou tempo de ida e volta) 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 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.

4. Considere um protocolo para transferência confiável de dados que utiliza somente confirmações negativas (NAKs). Suponha que o transmissor envie dados pouco frequentemente. Nessa situação, um protocolo que utiliza somente NAKs é preferível a um protocolo que utiliza ACKs? Por quê? Agora suponha que o transmissor tenha uma grande quantidade de dados para enviar e que a conexão fim a fim sofra poucas perdas. Nesse segundo caso, um protocolo que utiliza somente NAKs é preferível a um protocolo que utiliza ACKs? Por quê? **(1,0 ponto)**

Resposta:

A resposta a primeira pergunta é não. Em um protocolo que somente utilize NAKs, a perda de um pacote x só é percebida pelo receptor quando o pacote $x+1$ é recebido. Portanto, o receptor recebe os pacotes $x-1$ e $x+1$, somente quando $x+1$ é recebido que o receptor percebe x foi perdido. Se houver um intervalo longo entre a transmissão de x e a transmissão de $x+1$, então teremos um longo tempo até que x possa ser recuperado, com um protocolo que utilize apenas NAKs.

A resposta a segunda pergunta é sim. Se os dados estão sendo enviados frequentemente, então a recuperação de um protocolo que utilize apenas NAKs acontece de forma mais rápida. Além disso, se os erros não forem frequentes, então os NAKs só são enviados ocasionalmente, apenas quando necessário, e ACKs nunca serão enviados. Neste caso teremos uma redução significativa dos pacotes de controle.

5. Considere a transferência de um grande arquivo de L bytes do Hospedeiro A para o Hospedeiro B. Assuma um tamanho máximo do segmento (MSS- *Maximum Segment Size*) de 536 bytes.
(a) Qual o máximo valor de L , tal que, não sejam esgotados os números de sequência do TCP? Lembre-se que o campo para o número de sequência TCP tem quatro bytes. **(1,0 ponto)**

Resposta:

Existem $2^{32} = 4.294.967.296$ possíveis números de sequência. O número de sequência não é incrementado de 1 a cada segmento, na verdade é incrementado pela quantidade de *bytes* dos dados enviados no segmento. Desta forma, o tamanho do segmento é irrelevante e o tamanho máximo do arquivo que pode ser enviado de A para B sem que sejam esgotados os números de sequência, considerando que o TCP tem 4 *bytes* para acomodar este número é de 2^{32} , que representa aproximadamente 4,19 Gbytes.

(b) Para o L que você obtiver em (a) descubra quanto tempo demora transmitir o arquivo. Para isso, considere que um total de 66 *bytes* dos cabeçalhos de transporte, rede e de enlace, são adicionados a cada segmento antes que o pacote resultante seja enviado por um enlace de 155Mbps. Ignore o controle de fluxo e o controle de congestionamento, de tal modo que o Hospedeiro A envie os segmentos, um atrás do outro continuamente. **(1,0 ponto)**

Resposta:

O número de segmentos é o teto de $2^{32} / 536 = 8.012.999$. Adicionando 66 *bytes* de cabeçalho a cada segmento perfaz um total de 528.857.934 *bytes* de cabeçalhos. O número total de *bytes* transmitidos é de $2^{32} + 528.857.934 = 4,824 \times 10^9$ *bytes*. Portanto, o tempo de transmissão do arquivo será de 249 segundos.

6. Compare os três protocolos: Retorne a N (*Go-Back-N*), Repetição Seletiva e o TCP (sem ACKs atrasados). Admita que os valores de temporização para os três protocolos sejam suficientemente longos, tal que cinco segmentos de dados consecutivos e seus correspondentes ACKs, possam ser recebidos (se não forem perdidos no canal) respectivamente por um hospedeiro receptor, o Hospedeiro B, e pelo hospedeiro emissor, o Hospedeiro A. Suponha que o Hospedeiro A envie cinco segmentos para o Hospedeiro B, e que o segundo segmento enviado, seja perdido. Considere que ao final da transmissão, todos os cinco segmentos de dados foram recebidos corretamente no Hospedeiro B.

(a) No total, quantos segmentos o Hospedeiro A enviou e no total, quantos ACKs o Hospedeiro B enviou? Quais foram seus números de sequência? Responda essas perguntas para os três protocolos.

(2,0 pontos)

Resposta:

Go-Back-N:

A envia 9 segmentos no total. Inicialmente são enviados os segmentos 1, 2, 3, 4 e 5 e após são reenviados os segmentos 2, 3, 4 e 5.

B envia 8 ACKs. Destes, 4 ACKs com o número de sequência 1 e 4 ACKs com os números de sequência 2, 3, 4 e 5.

Repetição Seletiva:

A envia 6 segmentos no total. Inicialmente são enviados os segmentos 1, 2, 3, 4 e 5 e após é reenviado o segmento 2.

B envia 5 ACKs. Destes, 4 ACKs com os números de sequência 1, 3, 4 e 5. O ACK restante tem o número de sequência 2.

TCP:

A envia 6 segmentos no total. Inicialmente são enviados os segmentos 1, 2, 3, 4 e 5 e após é reenviado o segmento 2.

B envia 5 ACKs. Destes, 4 ACKs tem o número de sequência 2. O outro ACK restante tem o número de sequência 6. Note que o TCP sempre envia um ACK com o número de sequência do segmento esperado.

(b) Se os valores da temporização de todos os três protocolos são muito maiores que 5 RTT, qual dos três protocolos entrega corretamente, no menor tempo, todos os cinco segmentos? **(2,0 pontos)**

Resposta:

O TCP. Isto porque o TCP usa o mecanismo de retransmissão rápida, sem esperar que o temporizador expire.