



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 2007.

Gabarito

1. Por que o UDP existe? Não teria sido suficiente deixar que os processos de usuário enviassem pacotes diretamente para a camada de rede? (0,5 pontos)

Resposta:

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 redes 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 (*checksum*).

2. Descreva por que um desenvolvedor de uma aplicação distribuída escolhe executar sua aplicação sobre UDP em vez de executá-la sobre TCP. (1,0 ponto)

Resposta:

A escolha está baseada nas seguintes características do UDP:

- **Não há estabelecimento de conexão:** No TCP existe o *three way handshake* para que a transferência de dados tenha seu início e esta etapa produz atraso.
- **Não há estado de conexão:** O TCP mantém o estado de conexão. Esse estado inclui *buffers* de envio e recepção, parâmetros de controle de congestionamento e parâmetros numéricos de sequência e de reconhecimento, desta forma esta manutenção do estado tem custos de processamento e espaço alocado.
- **Pequeno overhead no cabeçalho do pacote:** O segmento TCP tem 20 bytes, enquanto o do UDP tem somente 8 bytes.
- **Taxa de envio não regulada:** No TCP existem os mecanismos de controle de congestionamento e de fluxo que regulam a taxa de envio de segmentos, sem a interferência da aplicação transmissora. Este controle na taxa de envio de segmentos pode ser crítica para aplicações que requisitam uma taxa de transmissão mínima e que são tolerantes a perda de pacotes até um certo nível.

3. É possível que uma aplicação que executa sobre UDP possa se beneficiar da transferência de dados confiável? Se sua resposta é afirmativa, explique como isso é possível. (1,0 ponto)

Resposta:

Sim, desde que a transferência de dados confiável seja implementada na própria aplicação, caso contrário não seria possível, pois o UDP não implementa a transferência de dados confiável.

4. 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 tamanho da **RcvWindow** depende da quantidade de dados recebidos e ainda não entregues à aplicação, portanto a **RcvWindow** pode crescer ou diminuir 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. Isto é garantido pelo mecanismo de controle de fluxo em relação à janela de recepção, que sempre menor ou igual ao buffer de recepção em B. Portanto o hospedeiro A nunca terá um número de bytes enviados e não reconhecidos que seja superior ao buffer de recepção do hospedeiro B.

- 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 seqüência para um segmento transmitido nessa conexão é **m**, então o número de seqüência para o segmento subsequente é **m+1**.

Resposta:

Falso. O próximo número de segmento será **m** acrescido da quantidade de bytes de dados da aplicação enviados 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 seqüência 38. Nesse mesmo segmento, o número contido no campo de confirmação é obrigatoriamente 42.

Resposta:

Falso. O valor inicial para os números de seqüência e de confirmação, nos dois extremos da conexão são definidos durante o estabelecimento da conexão TCP (durante o *three way handshake*), e cada uma das partes é livre para escolher o número de seqüência inicial que desejar.

- 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:

Falso. O limiar é ajustado à metade do tamanho da janela de congestionamento quando da ocorrência da expiração do temporizador.

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

(1,0 ponto cada)

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

Resposta:

No problema temos o tamanho de janela igual a 3. A faixa de números de sequência possível é de $k - 3$ até $k + 2$, respeitados os limites nos quais a faixa de números de sequência começa a ser reutilizada.

Podemos justificar esta faixa de números de sequência através de seus casos extremos, apresentados a seguir:

- Janela do transmissor ($k-3$, $k-2$, $k-1$): Neste caso o transmissor já enviou os três pacotes e o receptor já transmitiu seus respectivos ACKs e estes estão à caminho do transmissor. Portanto o receptor aguarda o pacote com o número de sequência k .
- Janela do transmissor (k , $k+1$, $k+2$): Neste caso o transmissor já recebeu os ACKs até o número de sequência $k-1$, recebeu da aplicação os próximos três pacotes, já os enviou ao receptor e o receptor ainda não recebeu o pacote com número de sequência 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:

$k-3$, $k-2$, $k-1$

Se o receptor está aguardando pelo pacote com o número de sequência k , obrigatoriamente o ACK com o número de sequência $k-1$ já foi enviado. Dado que a janela do transmissor comporta até três pacotes podemos concluir que no máximo três pacotes de ACK estão se propagando na direção do transmissor. Portanto, os números de sequência destes três pacotes seriam $k-3$, $k-2$, $k-1$.

6. Considere o protocolo Repetição Seletiva para transferência de dados confiável. Considere ainda que a faixa de números de sequência é: $0, 1, 2, \dots, k-1$. Qual o maior tamanho de janela possível para que o protocolo não falhe? (1,0 ponto)

Resposta:

Tamanho da Janela = $k / 2$

7. Considere a transferência do hospedeiro A para o hospedeiro B, de um arquivo (muito grande) com L bytes. Assuma que o Tamanho Máximo do Segmento (*Maximum Segment Size* - MSS) é de 1460 bytes. Qual o máximo valor de L de modo a que os faixa de números de sequência não seja esgotada? Lembre-se que o campo para número de sequência no TCP tem quatro bytes. (1,0 ponto)

Resposta:

Os números de sequência no TCP não são incrementados de um a cada segmento e sim do número de bytes dos dados enviados. Assim o tamanho do MSS é irrelevante – o tamanho máximo do arquivo que pode ser enviado de A para B é simplesmente o número de bytes que podem ser representados por 2^{32} . Isto é aproximadamente 4 Gbytes.

8. Discorra sobre o mecanismo “Aumento Aditivo, Diminuição Multiplicativa” usado no controle de congestionamento realizado pelo TCP. (1,0 ponto)

Resposta:

O mecanismo “Aumento Aditivo, Diminuição Multiplicativa” mantém a conexão TCP em dois estados:

- **Início lento (*slow start* – SS):** A conexão TCP está neste estado quando a janela de congestionamento for inferior ao limiar (*threshold*). A cada confirmação positiva (ACK) recebida em sequência o tamanho da janela de

congestionamento é acrescido do tamanho de um MSS, o que resulta na duplicação da janela de congestionamento a cada RTT (*Round Trip Time*).

- **Prevenção de congestionamento (*congestion avoidance* - CA):** A conexão TCP está neste estado quando a janela de congestionamento (CongWin) for igual ou superior ao limiar (*threshold*). A cada confirmação positiva (ACK) recebida em sequência o tamanho da janela de congestionamento é acrescido através da fórmula:

$$\text{CongWin} = \text{CongWin} + \text{MSS} \times \text{MSS} / \text{CongWin}$$

A aplicação desta fórmula resulta no acréscimo do tamanho de um MSS ao tamanho da janela de congestionamento a cada RTT (*Round Trip Time*).

Outros eventos além da recepção de um ACK esperado e em sequência:

- Quando um ACK para um segmento já confirmado é recebido, é incrementado o “contador de ACKs em duplicata” e quando o terceiro ACK para o mesmo segmento chega, a janela de congestionamento é reduzida à metade e o limiar (*threshold*) também recebe este mesmo valor. A conexão entra no estado de prevenção do congestionamento.
- Quando o temporizador expira (*timeout*) o *threshold* recebe o valor da metade do tamanho da janela de congestionamento e a janela de congestionamento é reduzida para seu tamanho mínimo, ou seja, de um MSS. Portanto, a conexão TCP entra na fase de início lento (“Slow Start”).