



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Redes de Computadores I

Gabarito da AP2 - 1º semestre de 2010.

1. **(1,5 pontos)** Explique para que serve o processo de demultiplexação realizado na camada de transporte.

Resposta:

Um processo, parte de uma aplicação de rede, pode ter um ou mais *sockets*, portas pelas quais dados passam da rede para o processo e do processo para a rede. A tarefa de entregar dados contidos em um segmento da camada de transporte para a porta correta é denominada demultiplexação. Para tanto, na extremidade receptora, a camada de transporte examina os campos do cabeçalho do segmento para identificar a porta receptora e direciona o segmento para o *socket* correto.

2. Suponha que um processo que executa no **Computador C** tem um *socket* UDP com número de porta 6789. Suponha ainda que processos que executam no **Computador A** e no **Computador B** enviam segmentos UDP para o **Computador C** tendo 6789 como porta de destino. Responda:

1. **(0,5 pontos)** Esses dois segmentos serão enviados para o mesmo *socket* no **Computador C**?

Resposta:

Sim, ambos os segmentos serão enviados para o mesmo *socket*.

2. **(1,5 pontos)** Se sim, como o processo do **Computador C** sabe que esses segmentos foram enviados por dois computadores distintos.

Resposta:

Para cada segmento recebido em um *socket*, o sistema operacional informa ao processo endereços IP de modo que seja possível determinar a origem de cada segmento.

3. 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**. Responda:

1. **(0,5 pontos)** Todas as solicitações estão sendo enviadas através do mesmo *socket* no **Computador C**?

Resposta:

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.

2. **(1,5 pontos)** Se as solicitações estão passando por diferentes *sockets*, podem ambos os *sockets* ter o número de porta 80? Discuta e explique.

Resposta:

Como vimos na resposta 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*.

4. **(1,5 pontos)** Para que serve o campo “Janela de Recepção” (ou “RcvWindow”) no cabeçalho do segmento TCP?

Resposta:

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.

5. O **hospedeiro A** está enviando um arquivo enorme para o **hospedeiro B** através de uma conexão TCP. Nessa conexão não há perda de pacotes e os temporizadores nunca se esgotam. Seja **R** bps a taxa de transmissão do enlace que liga o **hospedeiro A** à Internet. Suponha que o processo que executa no **hospedeiro A** é capaz de enviar dados para o seu *socket* TCP à uma taxa de **S** bps, onde **S** = 10 x **R**. Suponha ainda que o *buffer* de recepção do TCP (no **hospedeiro B**) seja grande o suficiente para conter o arquivo inteiro, e que o *buffer* de envio TCP (no **hospedeiro A**) pode conter apenas um por cento do arquivo. O que impediria o processo que executa no **hospedeiro A** de passar dados continuamente para o seu *socket* TCP a taxa de **S** bps:

1. **(1,0 ponto)** O controle de fluxo TCP? Explique sua resposta.

Resposta:

Neste problema não existe o perigo do transmissor sobrecarregar (inundar) o receptor, pois o *buffer* do receptor é capaz de armazenar o arquivo inteiro, logo o controle de fluxo não impede o **hospedeiro A** de passar dados continuamente para o seu *socket* TCP a taxa de **S** bps.

2. **(1,0 ponto)** O controle de congestionamento TCP? Explique sua resposta.

Resposta:

Considerando que não existem perdas e que os ACKs retornam antes dos temporizadores expirarem, o mecanismo de controle do congestionamento não influi na taxa do transmissor.

3. **(1,0 ponto)** Alguma outra coisa? Explique sua resposta.

Resposta:

Contudo, o processo do **hospedeiro A** não poderá passar dados continuamente para o *socket*, pois o *buffer* de transmissão é pequeno e logo fica cheio. Portanto, tão logo que o *buffer* de transmissão está cheio, o processo repassa os dados à uma taxa média R que é muito menor que S .