

Redes de Computadores – 2019

Segundo Trabalho

Transmissão Confiável

1 Descrição do Trabalho

Neste trabalho, você implementará uma versão simplificada do protocolo TCP para transmissão confiável de dados. A sua implementação consistirá em uma aplicação cliente e outra servidora. A aplicação cliente transmitirá dados de um arquivo encapsulados em segmentos TCP, mas transmitidos em datagramas UDP. Essa aplicação deve implementar o algoritmo de janela deslizante do TCP para transmissão confiável, levando em consideração o valor anunciado na janela do receptor. Os pacotes transmitidos e não confirmados devem ser mantidos em um buffer na memória, ou seja, eles não podem ser lidos diretamente do disco para retransmissão. A aplicação servidora irá receber os datagramas UDP, desencapsular os segmentos TCP, enviar segmentos ACK de confirmação e armazenar os dados recebidos em disco somente quando eles estiverem em ordem, ou seja, os segmentos recebidos fora de ordem devem ser colocados em um buffer de recepção.

As aplicações cliente e servidora devem ser executadas com as linhas de comando abaixo:

```
$ tcp_client fn sip sport wnd rto mss dupack lp
```

```
fn:      nome do arquivo a ser enviado
sip:      endereço IP do servidor
sport:    porta UDP do servidor
wnd:      tamanho da janela do transmissor e receptor em bytes
rto:      valor inicial de timeout para retransmissão de um segmento em
           milisegundos
mss:      TCP Maximum Segment Size
dupack:   deve ser um para usar retransmissão via ACKs duplicados e zero caso
           contrário
lp:       probabilidade de um datagrama UDP ser descartado
```

```
$ tcp_server fn sport wnd lp
```

```
fn:      nome do arquivo a ser recebido e gravado em disco
sport:    porta UDP que o servidor deve escutar
wnd:      tamanho da janela do transmissor e receptor em bytes
lp:       probabilidade de um datagrama UDP ser descartado
```

1.1 Estabelecimento de Sessão

Você deve implementar a fase de estabelecimento de sessão do TCP (*three-way handshake*) antes de começar a transmitir os dados. Nesta fase, os valores da janela do receptor e de números de sequência iniciais devem ser negociados. Lembre-se que o número de sequência inicial de uma sessão TCP é um valor aleatório de 32 bits.

1.2 Confirmações Cumulativas

Os segmentos de confirmação (ACKs) devem ser cumulativos, como no TCP.

1.3 Retransmissões

Você deve implementar duas formas de retransmissão de segmentos perdidos: ACKs duplicados e timeout. Ao receber três ACKs com o mesmo valor, você assume que o segmento com número de sequência no campo ACK do segmento TCP foi perdido e precisa ser retransmitido. Para retransmissões com timeout, você deve calcular o valor do timeout em função dos tempos de envio dos segmentos e os tempos de recebimento de suas confirmações. Utilize o algoritmo padrão do TCP para atualização do valor do timeout. Para os segmentos SYN e SYN+ACK, os valores de timeout a serem utilizados são os fornecidos na linha de comando. A retransmissão por ACKs duplicados pode ser habilitada ou desabilitada pelo parâmetro `dupack` da linha de comando. A retransmissão por timeout é sempre ativa, mas um segmento pode ser retransmitido antes por ACKs duplicados.

1.4 Simulando Perdas de Pacote

Você deve implementar uma função de envio de datagramas UDP que recebe como parâmetro um valor de probabilidade de descarte de pacotes (esse valor é passado na linha de comando). Para cada datagrama a ser transmitido, você deve gerar um número aleatório entre 0 e 1. Se o valor aleatório gerado for maior do que a probabilidade de descarte, o datagrama deve ser transmitido. Caso contrário, o datagrama deve ser descartado.

2 Entrega do Trabalho

O trabalho pode ser feito em grupo de no máximo dois alunos e deve ser entregue até o dia **19 de junho de 2019**. A entrega do trabalho consistirá em uma demonstração das funcionalidades do proxy no laboratório. O grupo deve preparar uma breve apresentação e um roteiro para a demonstração. O grupo deve indicar explicitamente as funcionalidades que foram implementadas e as que não foram. Na demonstração, o grupo deve incluir casos que demonstrem claramente as funcionalidades implementadas, como, por exemplo, o tratamento de várias conexões simultâneas. Além disso, o grupo deve entregar o código fonte e um breve relatório

descrevendo o trabalho. Neste relatório, o grupo deve incluir uma breve introdução, decisões de implementação, funcionalidades não implementadas, problemas enfrentados na implementação, etc. O relatório deve ser entregue em um arquivo PDF.

3 Avaliação Experimental do Protocolo

No relatório do trabalho, você deve incluir dois gráficos com os tempos de transmissão de um arquivo em função da probabilidade de perda de pacotes e diferentes valores de MSS. O eixo X deve conter os valores de perda de pacote 0%, 1%, 2%, 4%, 8%, 16% e 32%. Um gráfico deve utilizar o valor de MSS igual a 512 e outro o valor 1024. O arquivo a ser transmitido deve possuir pelo menos 1GB de tamanho e a conexão entre o cliente e o servidor deve ser feita em uma rede local, ou seja, o cliente e o servidor não podem estar rodando na mesma máquina. Para cada valor de probabilidade, você deve fazer a mesma transmissão pelo menos dez vezes e calcular o intervalo de confiança. Em seu relatório, você deve descrever o cenário usado no experimento, ou seja, as características dos computadores usados no experimento e a velocidade da rede que conecta os computadores.

4 Avaliação

Além da correção do programa, o professor e/ou assistente de ensino poderão fazer perguntas durante a apresentação do trabalho. Durante a apresentação, o grupo deverá explicar o funcionamento do programa e responder a perguntas relativas ao projeto.

5 Referência

Transmission Control Protocol <https://tools.ietf.org/html/rfc793>