

Laboratório 01 – Programação Cliente-Servidor: UDP

24 de Novembro de 2020

UDP Pinger

Nesta tarefa, você irá estudar um simples servidor de Ping da Internet escrito em linguagem Java e implementar um cliente correspondente. A funcionalidade provida por esses programas é similar à dos programas de Ping padrão disponíveis nos sistemas operacionais modernos, exceto que será utilizado o UDP em vez do ICMP (Internet Control Message Protocol) para se comunicar (Java não provê meios diretos para interagir com o ICMP.)

O protocolo Ping permite a uma máquina cliente enviar um pacote de dados para uma máquina remota, a qual retornará o dado para o cliente sem modificações (uma ação conhecida como eco). O protocolo Ping permite aos hospedeiros determinarem o tempo de resposta de outras máquinas. O código do servidor é fornecido no TIDIA. Seu trabalho será escrever o cliente Ping.

Parte 1: Código do Servidor

O código disponível no TIDIA implementa por completo o servidor de Ping. Você precisará compilar e executar este código. Estude-o cuidadosamente, pois ele irá ajudá-lo a escrever seu cliente de Ping. O servidor fica num loop infinito de escuta pela chegada de pacotes UDP. Quando um pacote chega, o servidor simplesmente envia o dado encapsulado de volta para o cliente.

Perda de pacotes

O UDP provê aplicações com serviço de transporte não confiável, pois as mensagens podem se perder pela rede devido a um *overflow* na fila do roteador ou por outras razões. Aplicações que usam o UDP para comunicação precisam implementar alguma confiabilidade separadamente no nível de aplicação (cada aplicação pode implementar uma política diferente, de acordo com necessidades específicas).

Devido ao fato de a perda de pacotes ser rara, ou até mesmo inexistente, em uma rede típica, o servidor neste laboratório injeta perda artificial para simular os efeitos da perda de pacotes na rede. O servidor possui um parâmetro `LOSS_RATE`, que determina qual a porcentagem de pacotes deve ser perdida.

O servidor também possui outro parâmetro, `AVERAGE_DELAY`, que é usado para simular o atraso de transmissão ao enviar um pacote pela Internet. Você deve ajustar o `AVERAGE_DELAY` com um valor positivo quando o cliente e o servidor forem estar na mesma máquina, ou quando as máquinas estiverem muito perto fisicamente na rede. Você pode ajustar o `AVERAGE_DELAY` em 0 (zero) para encontrar o tempo de transmissão verdadeiro dos seus pacotes.

Compilando e executando o servidor

Para compilar o servidor, faça o seguinte:

```
javac PingServer.java
```

Para executar o servidor, faça o seguinte:

java PingServer port

onde *port* é o número da porta que o servidor escuta. Lembre que você deve usar um número de porta maior ou igual a 1024, pois apenas os processos executando no modo root (administrador) possuem privilégio de usar portas menores que 1024.

Nota: Se você obtiver um erro de classe não encontrada quando executar o comando acima, você precisará indicar ao compilador Java o diretório atual para resolver as referências de classe. Nesse caso, o comando de compilação é:

java -classpath . PingServer port

Parte 2: Programando o Cliente

Você deve escrever o cliente de modo que ele envie 10 requisições de Ping para o servidor, separadas por aproximadamente 1 segundo. Verifique o funcionamento do comando ping executando:

ping -c 10 127.0.0.1

Você deve escrever o cliente de modo que ele inicie com o seguinte comando:

java PingClient host port

onde *host* é o nome do computador em que o servidor está sendo executado e *port* é o número da porta que ele está escutando. Note que você pode executar o cliente e o servidor em diferentes máquinas ou na mesma.

O cliente deve enviar 10 Pings para o servidor. Como o UDP é um protocolo não confiável, alguns dos pacotes enviados pelo cliente ou pelo servidor podem ser perdidos. Por essa razão, o cliente não pode esperar indefinidamente pela resposta a uma mensagem de Ping. Você deve fazer com que o cliente espere por um tempo TIMEOUT por uma resposta; se nenhuma resposta for recebida, presume-se que o pacote foi perdido durante a transmissão. Você precisará pesquisar a API para o DatagramSocket de modo a descobrir como se ajusta o valor de tempo de expiração em um socket de datagrama.

Ao desenvolver seu código, você deve executar o servidor de Ping em sua máquina e testar seu cliente enviando pacotes para *localhost* (ou, 127.0.0.1). Após o completo debug do seu código, você deve ver como sua aplicação se comunica através da rede com um servidor de Ping sendo executado por um outro membro da classe.

Formato das mensagens

As mensagens de Ping neste laboratório são formatadas de modo simples. Cada mensagem contém uma sequência de caracteres terminados por um carácter de retorno (\r) e um carácter de mudança de linha (\n). A mensagem contém a seguinte string:

PING sequence_number time CRLF

onde *sequence_number* começa em 0 (zero) e progride até 9, para cada mensagem sucessiva de Ping enviada pelo cliente; *time* é o tempo do momento em que o cliente enviou a mensagem e *CRLF* representa o retorno e linha de caracteres que finalizam a linha.