

Relatório 03

Cristiano Medeiros Dalbem Mauricio Rodrigues Cruz

28 de setembro de 2011

1 Experimentos

1.1 UDP

1. Verificar as estruturas analisadas no diretório esqueleto-udp. Testar programa como está. Transformar o transmissor para enviar um número bits/s na rede de acordo com as seguintes linhas de comando:

-r n: onde n = soma da idade da dupla, que no caso é 44 e pode ser observada na Figura 1

-r 500: 500 kbit/s observada na Figura 2

-r 1000: 1Mbit/s observada na Figura 2

Poucas alterações foram necessárias no código fornecido pelo professor. O tamanho de pacote de dados enviado foi aumentado para garantir que a transmissão de taxas mais altas de dados sejam atingidas e algumas linhas de código puderam ser retiradas sem perda de funcionalidade. Além disso, o parâmetro $-r$ foi adicionado para que se possa regular a taxa de transmissão desejada. Isso é feito sabendo que a frequência na transmissão de pacotes desejada é obtida esperando o seu inverso, que significa o período de espera entre cada envio de pacote. Para executar o experimento, basta inicializar o Server com a seguinte linha de comando:

Receiver.exe -p < porta >

E o sender com o comando:

Sender -h < numeroipdoserver > -p < portadoserver > -r < taxadesejadaemKbits/s >

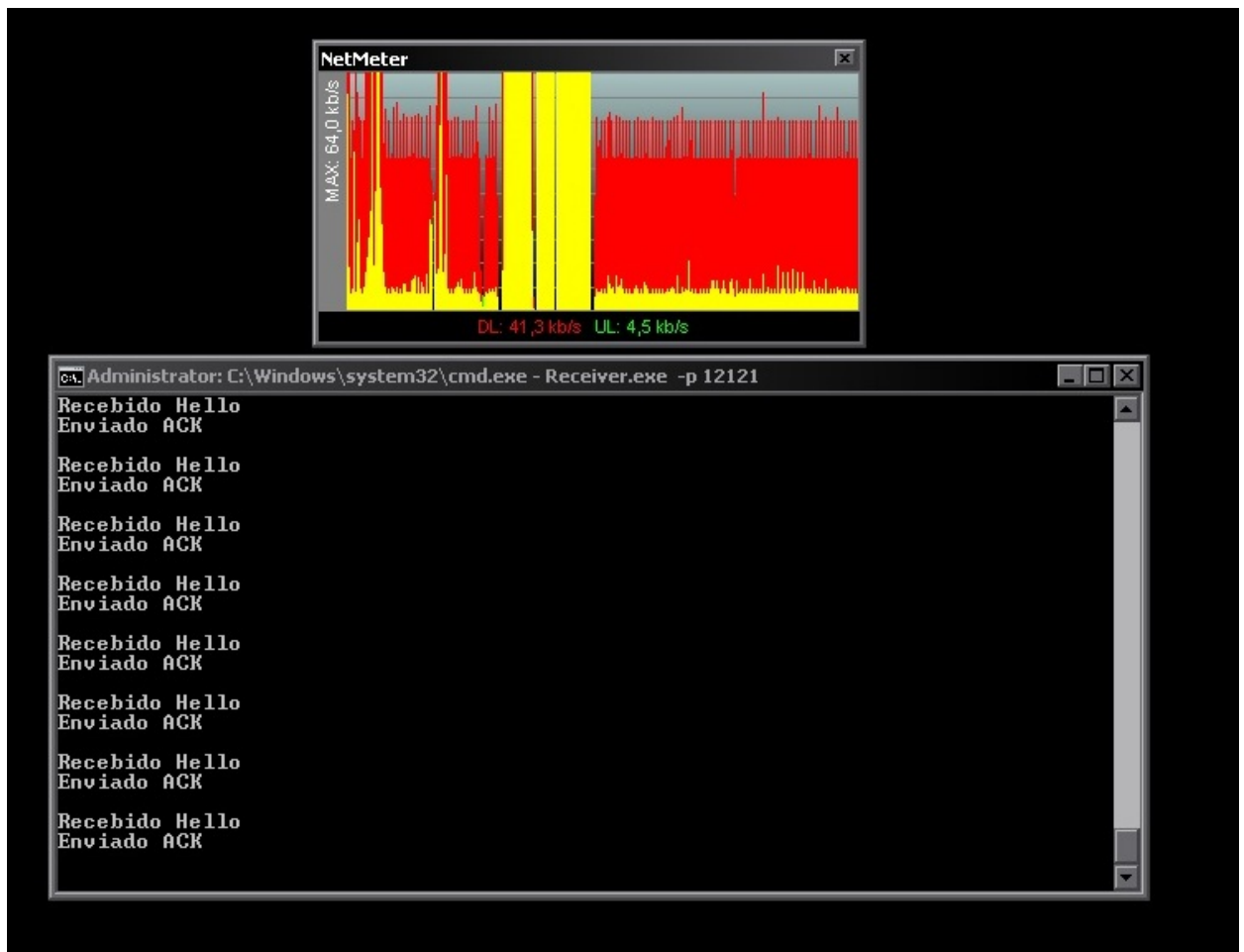


Figura 1: Pacotes sendo enviados a 44kb/s pelo protocolo UDP

Código de tratamento da entrada do valor de rate:

```
1 case 'r': //rate
2 {
3     i++;
4     int rate = atoi(argv[i]);
5     delayTime = (int)(sizeof(buffer)*8./rate*1000.);
6     delayTime = (int)(delayTime/1000.);
7 }
8 break;
```

Código do Main Loop de geração de banda:

```

1 while(1)
2 {
3     strcpy(buffer,"Hello");
4     sendto(s, buffer, sizeof(buffer), 0, (struct sockaddr *)&peer,
5         peerlen);
6     printf("Enviado Hello\n");
7     rc = recvfrom(s,buffer,sizeof(buffer),0,(struct sockaddr *)&peer, &
8         peerlen);
9     printf("Recebido %s\n\n",&buffer);
10    Sleep(delayTime);
11 }

```

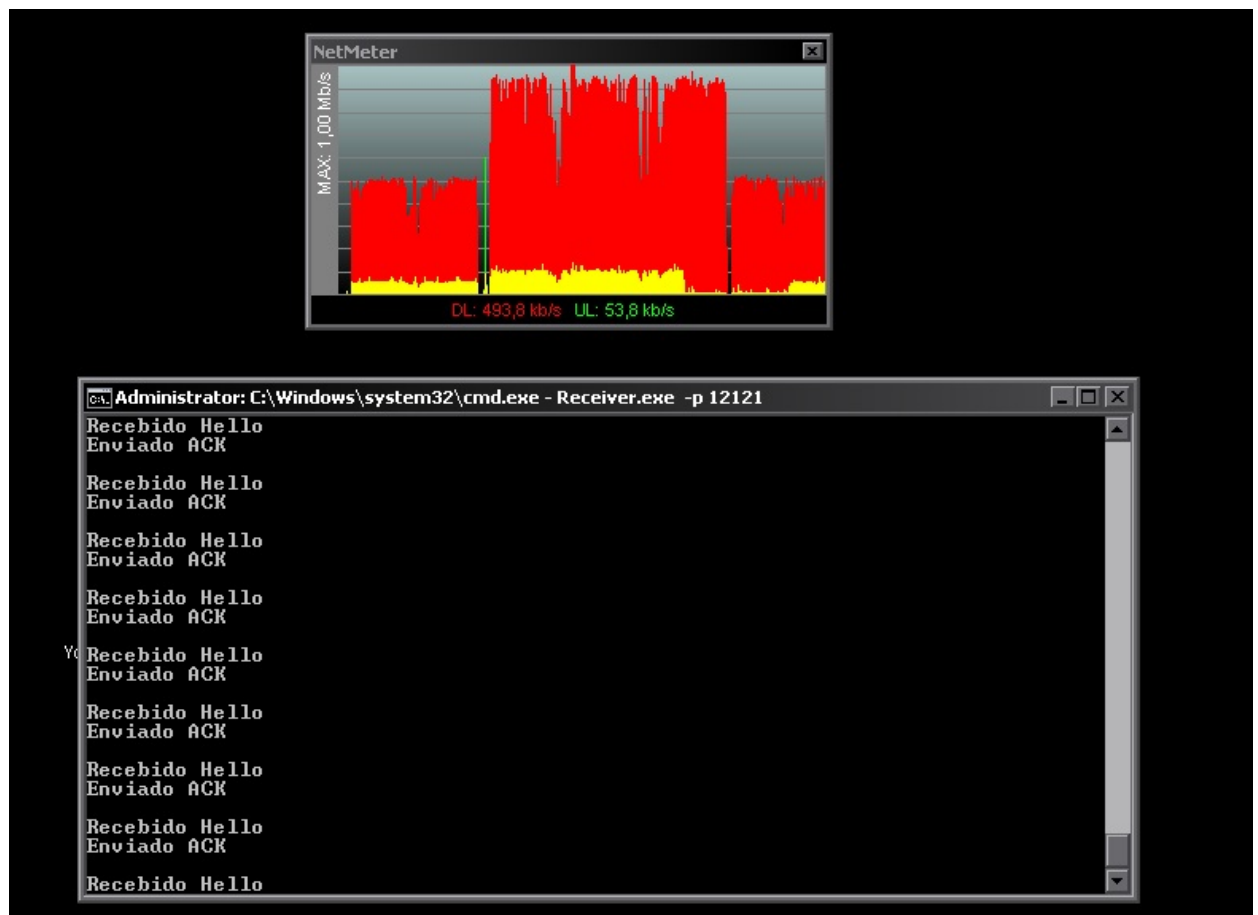


Figura 2: Pacotes sendo enviados a 500kbts/s e, logo após, enviados à 1000kbts/s

1.2 TCP

2. Verificar as estruturas analisadas no diretório esqueleto-tcp. Testar programa como está. Analisar equidade do tráfego na rede, alterando o programa para ficar enviando dados na máxima velocidade possível, e testando com várias transmissões simultâneas (mínimo de 2 conexões dois servidores (duas instâncias do servidor em portas diferentes) e dois clientes). Gerar log com a média de tráfego por segundo. Pode-se realizar o relatório com duas máquinas, mas se o grupo quiser, poderá utilizar 3 máquinas ou mais (para ver na ferramenta de redes, além do log, a variação de carga na rede)

Novamente foram necessárias poucas mudanças no código já fornecido e as que foram feitas, são similares as já apresentadas na primeira parte desse relatório. Entretanto, o propósito desse experimento era um pouco diferente e a taxa de transmissão deveria ser máxima ao longo da execução, portanto, o parâmetro -r de seleção desse quesito não era necessário, portanto não foi implementado.

Comando para iniciar um servidor de escuta TCP

server < port >

Comando para iniciar um cliente simulando envio de dados para um servidor TCP.

client - c < port_client > -s < port_server > -i < ip_server >

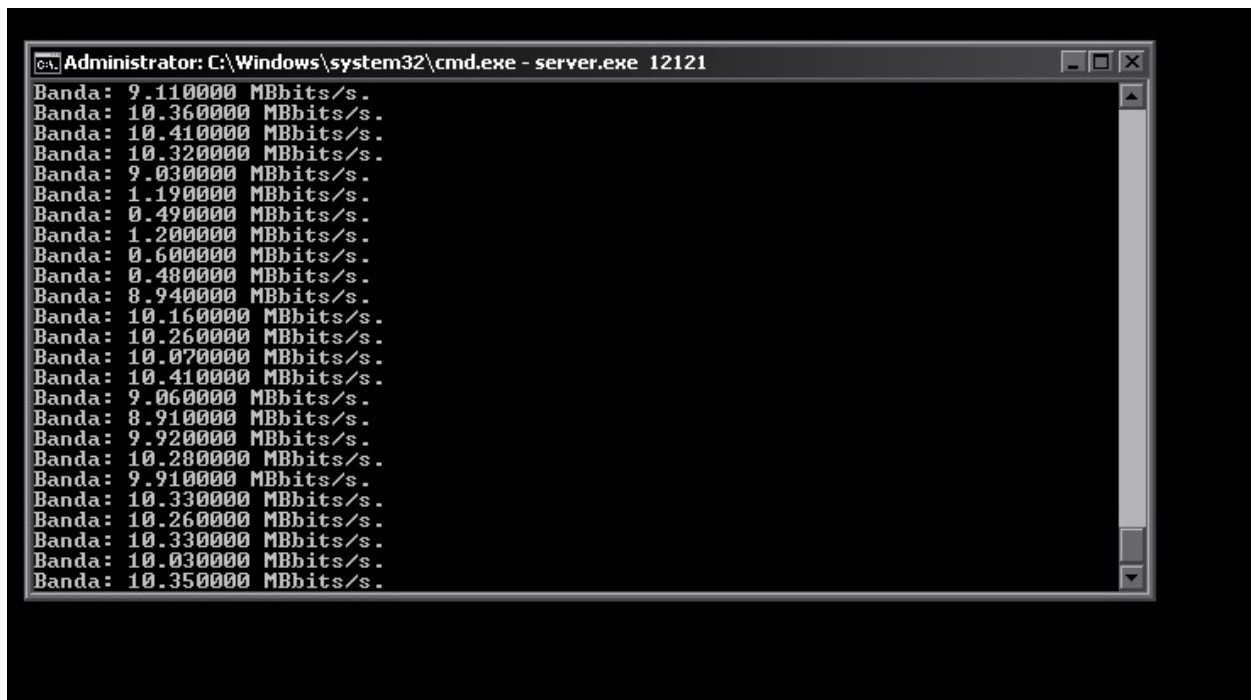


Figura 3: Um servidor TCP recebendo dados de um cliente.

Na figura 3, pode se observar um servidor recebendo na capacidade máxima da conexão disponível de um cliente remoto.

Já na figura 4, dois servidores dividem de forma igualitária a mesma conexão, ocupando-a completamente. Um detalhe que se pode observar é a disparidade entre a conexão total máxima atingida com somente um servidor, observado na figura 3, da atingida com duas instancias do mesmo servidor. Isso se dá pelo fato de que o protocolo TCP/IP modifica o tamanho dos pacotes de dados enviados, *camuflando* o tamanho real do pacote, e assim confundindo o servidor sobre a taxa individual de transmissão.

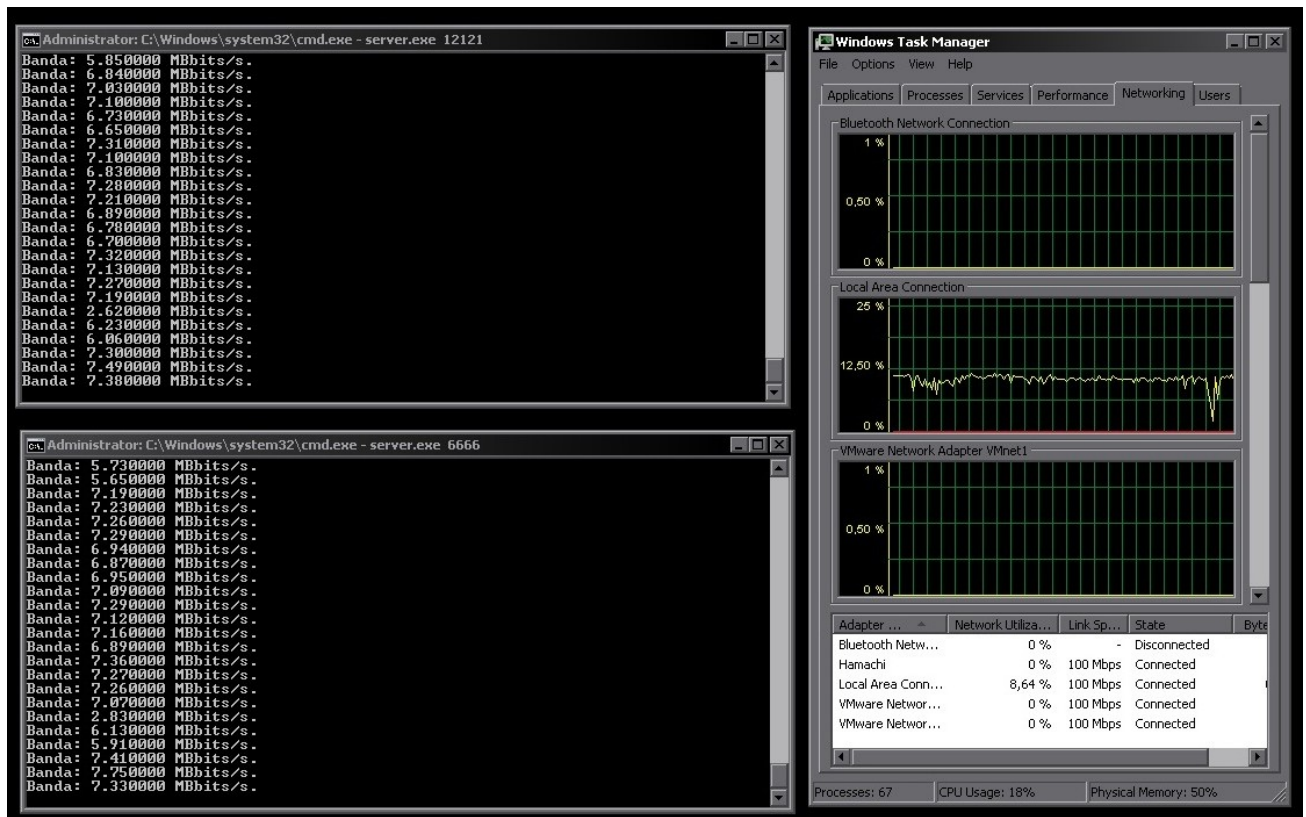


Figura 4: Dois servidores TCP dividindo banda igualmente.

2 Exercícios

3. O que é equidade de tráfego e em qual protocolo ela foi observada nativamente no laboratório?

Equidade de tráfego se trata da distribuição igualitária de banda por todas as transmissões ocorrendo simultaneamente. Ou seja, a banda total é dividida pelo número de transmissões concorrentes. Essa característica é observada no protocolo TCP, utilizando a multiplexação por divisão temporal.

4. Qual o nível do modelo OSI para fazer controle de fluxo quando o programador utiliza TCP e quando utiliza UDP?

Ambos os protocolos tratam do controle de fluxo na camada 4 (de Transporte) do modelo OSI.

O protocolo TCP utiliza o controle de fluxo por janela deslizando, adaptando a velocidade de transmissão de acordo com as capacidades de cliente e servidor. Já o protocolo UDP não utiliza nenhum controle de fluxo, mantendo seu fluxo constante ao longo da transmissão.