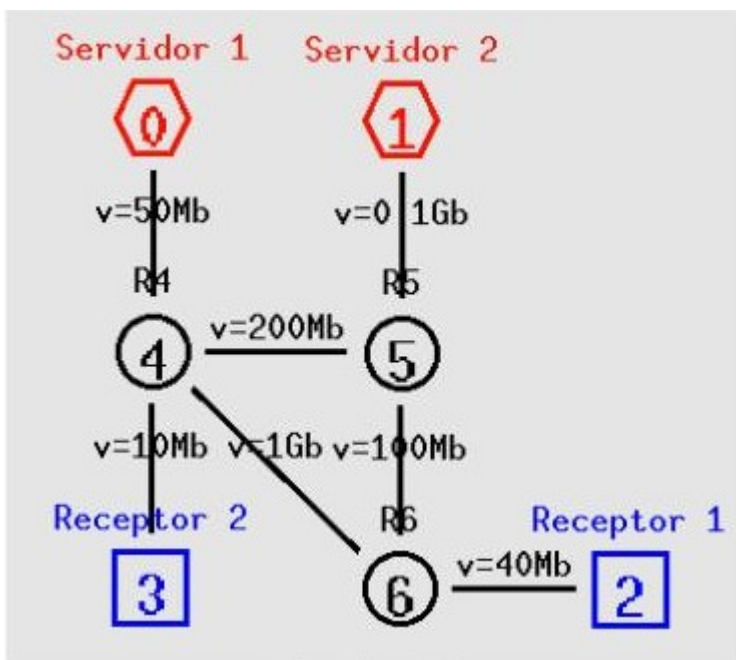


Trabalho 1 – Protocolos da Camada de Transporte

Relatório IRC

Este trabalho pretendia analisar e comparar a transmissão de dados usando os protocolos UDP e TCP. Para tal foram criados dois cenários: Cenário 1 que pretendia transmitir um bloco de 2MB usando o protocolo UDP e TCP, com tráfego originado no “Servidor 1” e como destino os recetores “1” e “2”, e o Cenário 2 onde são ativadas duas *streams* de dados, com origem no “Servidor 2” e destino, uma no “Recetor 1” e outra no “Recetor 2” enviadas por UDP, de acordo com a seguinte imagem:



1.

No exercício 1 é-nos pedido, entre outras coisas, para minimizar o número de ficheiros usados, para resolver este problema, criámos condições para que, os argumentos dados por um utilizador na linha de comando, sejam os valores necessários para criar os vários cenários. Os valores necessários são o tipo de cenário (1 ou 2) e o tipo de protocolo (“udp” ou “tcp”). Em relação há ligação entre o nó 4 e o nó 6, no caso de não ser introduzido um terceiro argumento na linha de comando, a ligação permanecerá ativa (assim como se o argumento for “continua”), no caso do argumento introduzido ser “quebra”, então a ligação será temporariamente interrompida e o tráfego de dados realizar-se-á do nó 4 para o nó 5 e deste para o nó 6.

2.

Para preencher a tabela fornecida, foi necessário utilizar a diretoria indicada no enunciado `“./ns-2.35/tcl/lib/ns-default.tcl”`. O ficheiro `“ns-default.tcl”` contém as informações sobre os valores por default que o ns2 utiliza.

Tamanho por omissão das filas nos nós	50
Tamanho por omissão dos pacotes TCP	1000
Tamanho por omissão dos pacotes UDP	1000
Tamanho por omissão da janela do TCP	20

3.

Este exercício, sobre o cenário 1, foi realizado através de tentativas, para obter o cálculo da janela mínima. Para obter o tempo mínimo e o número de pacotes perdidos durante toda a transmissão, tanto no UDP como para o TCP, foi utilizado o ficheiro `“trace_analyzer.awk”` assim como os comandos (respetivamente):

- `awk -f trace_analyzer.awk type=cbr src=0 dest=2 flow=4 out.tr;`
- `awk -f trace_analyzer.awk type=tcp src=0 dest=2 flow=3 out.tr;`

3.1

TCP			UDP	
Tempo min	Janela min	Nº pacotes perdidos	Tempo min	Nº pacotes perdidos
0.777538s	224	0	0.368368s	372

3.2

TCP			UDP	
Tempo min	Janela min	Nº pacotes perdidos	Tempo min	Nº pacotes perdidos
0.817780s	224	0	0.368401s	424

4

No exercício 4 era-nos pedido o mesmo que no exercício 3, mas com um cenário diferente (Cenário 2) e com uma janela de transmissão de 20 pacotes (para as alíneas 4.1 e 4.2). Por esta razão, o nosso método de resolução foi o mesmo que no exercício 3 (o uso do programa “trace_analyzer” para obter os valores do tempo e de pacotes perdidos).

4.1

TCP		UDP	
Tempo	Nº pacotes perdidos	Tempo	Nº pacotes perdidos
5.035942s	0	0.366636s	455

4.2

TCP		UDP	
Tempo	Nº pacotes perdidos	Tempo	Nº pacotes perdidos
5.076178s	0	0.366787s	507

4.3

TCP			UDP		
Tempo min	Janela min	Nº pacotes e enviados/recebidos	Tempo min	Nº pacotes perdidos	Velocidade S1-R4
0.913621s	164	2099/2099	0.47666s	0	37 mb/s

5.

Através da análise dos resultados das perguntas anteriores conseguimos tirar conclusões sobre a performance do TCP e do UDP.

Tendo como base os resultados obtidos podemos concluir que:

- Nas mesmas condições e em relação à velocidade, o UDP é melhor pois possui menor tempo total de transmissão que o TCP.
- Em termos de eficiência de transmissão de pacotes, o TCP é melhor pois em todos os casos acima analisados usando o TCP, não foi perdido nenhum pacote enquanto que, no caso do UDP, houve perdas de pacotes.
- Estes factos são concordantes com a teoria estudada já que, como o UDP apenas transmite os dados uma vez e não possui mecanismos de verificação ou confirmação de chegada de pacotes, é esperado que transmita os pacotes mais rapidamente que o TCP, não havendo certezas da receção dos pacotes, tornando este protocolo menos confiável. O TCP possui mecanismos de confirmação, como o ACK (*Acknowledgement*), que são enviados pelo recetor aquando da chegada dos pacotes, de modo a confirmar a chegada destes. Se os pacotes não tiverem chegado, são retransmitidos. Este mecanismo torna o processo mais confiável o que torna o tempo de transmissão maior.
- No TCP também é usada uma *window* para controlar o fluxo de dados, a *window* é a quantidade máxima de dados que podem ser transmitidos sem haver a receção do ACK.

6.

No caso da stream "Servidor 2"- "Recetor 1", provoca congestionamento na ligação "R6" - "Recetor 1", o que vai fazer com que haja perdas de pacotes no caso da ligação UDP("Servidor 1"- "Recetor 1"), no caso TCP, não vai haver perdas de pacotes, mas vai aumentar o tempo de transmissão devido à congestão da ligação.

No caso da stream "Servidor 2"- "Recetor 2", não haverá alterações substantivas na ligação "Servidor 1"- "Recetor 1" tanto no UDP, como no TCP, já que a ligação "R4"- "R5", é uma ligação full-duplex, existindo dois sentidos de transmissão havendo pouca interferência entre os sentidos "R4"- "R5" e "R5"- "R4".

Uma das soluções para evitar ou reduzir o congestionamento seria aumentar o tamanho das filas, para diminuir o congestionamento.

Outra solução seria diminuir a velocidade de transmissão, o que permitiria mais capacidade de vazão, embora aumentasse o tempo de transmissão.

Conclusão

Com base nos resultados das tabelas dos exercícios três e quatro, podemos concluir que, apesar do tempo mínimo de transmissão de dados ser substancialmente menor para o protocolo UDP e, conseqüentemente, maior para o TCP, existe uma perda significativa de pacotes nas transmissões através do protocolo UDP, enquanto que o TCP garante a entrega de todos os pacotes enviados. Assim, quando for requerida a segurança no envio dos dados, o TCP será mais indicado, enquanto que o UDP será mais indicado, por exemplo, para o envio de ficheiros de áudio/vídeo via web, em que é preferível velocidade, no caso de um pacote não ter sido transmitido também seria melhor ignorá-lo que voltar a retransmitir o pacote em casos de ficheiro vídeo/áudio.

Trabalho realizado por:
Daniel Azevedo nº2014200607
Nuno Afonso Santos nº2014226541