



Introdução às Redes de Comunicação

Trabalho 1

Realizado por:

Rui Mendes 2013136967

Ana Fidalgo 2013134819

Introdução

Um protocolo define o formato e a ordem das mensagens trocadas entre duas ou mais entidades e as ações tomadas durante a transmissão ou recepção de mensagens, sendo que os três protocolos na área de redes mais conhecidos são o UDP, o TCP e o IP. O TCP-IP é um conjunto de protocolos bem definidos.

Os dois primeiros protocolos – UDP e TCP – pertencem à camada de transporte do modelo de comunicação TCP/IP. Como tal, são responsáveis pelo envio e recepção de pacotes de informação entre máquinas distintas, referenciadas por IPs, assumindo funções como fragmentação e empacotamento e ainda pela implementação de mecanismos de controlo de fluxo e deteção de erros.

O TCP-IP é uma abreviatura de Transmission Control Protocol – Internet Protocol. Possui apenas 5 camadas, o que dá uma diminuição de overhead em relação ao modelo OSI. Cada camada comunica com a camada correspondente no receptor, tal como na OSI.

O protocolo TCP oferece garantias de integridade, fidelidade e certeza de que os pacotes chegam ao destino. Cada pacote que chega ao seu destino envia ao seu emissor um outro pacote, denominado ‘pacote acknowledge’ (ACK), que funciona como verificação da chegada do pacote ao destino. Tem ainda a vantagem de não sobrecarregar a rede com excesso de pacotes (redefinindo a ‘janela de congestão’), sendo, por isso, um protocolo com boas capacidades de controlo de fluxo. A sua grande desvantagem é o tempo que demora a enviar a informação (muito superior ao obtido quando se utiliza UDP), relacionado com o facto de ser um protocolo ‘pesado’, adicionado mais informação do que o UDP aos dados e colocando mais restrições na tolerância a falhas.

No protocolo UDP não existem verificações nem confirmações de que os dados foram recebidos corretamente (que não se perderam os pacotes nem que estão pela ordem certa). Para além disso, os pacotes de informação são enviados apenas uma vez;

se houver algum problema na sua transmissão ou ressecção, perdem-se permanentemente. É, por isso, um protocolo vantajoso no que toca à velocidade de envio dos dados, uma vez que não possui overhead com confirmações nem com criações/destruições de conexões (ao contrário do TCP, o UDP é ‘connectionless’). Não é, no entanto, aconselhado quando há necessidade de garantias de confiabilidade e ordenação ou a necessidade de mecanismos de controlo de fluxo.

Com este trabalho pretendemos analisar e comparar a transmissão de dados usando os protocolos UDP e TCP utilizando o NS2.

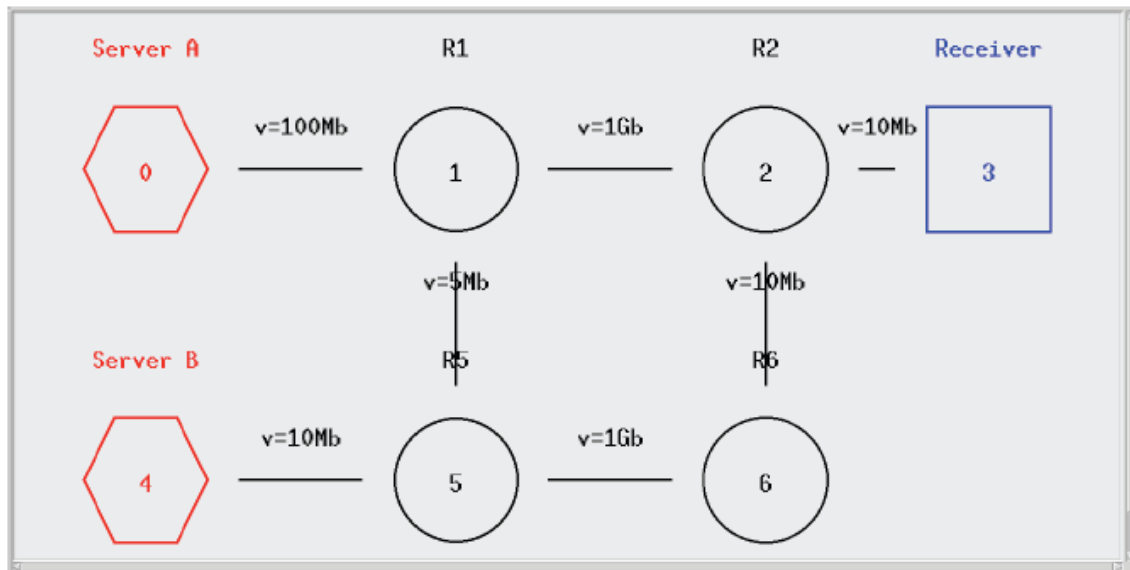
Temos um “Servidor A” que vai enviar a um “Receptor” um bloco de dados de 3MB, que começa a ser transmitido no instante 0.5 segundos. Ao mesmo tempo, poderá existir tráfego entre um “Servidor B” e o “Receptor”, que corresponde a uma stream de música que o “Receptor” está a receber (que também começa no instante 0.5 segundos).

Os tempos de propagação são todos de 10 ms, com excepção da ligação entre o “Router 2” e o “Receptor” que será de 3ms.

Para efeitos de simulação, todo o tráfego do “Servidor A” para o “Receptor” foi criado usando o gerador de tráfego CBR existente no NS2 o qual gerou um pacote de dados com 3MB.

Dependendo do cenário considerado o “Receptor” tem a stream activa ou não. Neste trabalho existiam dois cenários, um 1º sem tráfego entre o “Servidor B” e o “Receptor” e um 2º onde o tráfego entre o “Servidor B” e o “Receptor” ocupa 50% da largura de banda existente entre o “Servidor B” e o “Router 5”.

Ou seja:



Onde as ligações apresentam as seguintes características:

- Servidor A – Router 1: ligação a 100Mb/s
- Router 1 - Router 2: Ligação a 1Gb/s
- Router 2 - Receptor: Ligação a 10Mb/s
- Servidor B - Router 5: ligação a 10Mb/s
- Router 5 - Router 6: Ligação a 1Gb/s
- Router 1 - Router 5: Ligação a 5Mb/s
- Router 2 - Router 6: Ligação a 10Mb/s

Exercício 1

Ao chamarmos o ficheiro no terminal, os argumentos devem ser passados pela seguinte ordem:

- 1- O tipo de ligação a definir para o percurso servidor B ao receptor, ou seja qual o protocolo a usar. Inserimos 0 para UDP e 1 para TCP;
- 2- O cenário que estamos a testar: inserir 1 para o primeiro cenário (Sem tráfego entre o “Servidor B” e o “Receptor”) e 2 para o segundo cenário (o tráfego entre o “Servidor B” e o “Receptor” ocupa 50% da largura de banda existente entre o “Servidor B” e o “Router 5”);
- 3- O tamanho da janela para ligações TCP. O seu melhor valor foi deduzido nos exercícios seguintes;
- 4- Utilização de quebra na ligação: inserir 1 para utilizar.

Com o enunciado foi disponibilizado o ficheiro ‘trace_analyzer.awk’ que nos permitiu obter estatísticas sobre o tráfego enviado.

Comandos a utilizar para as estatísticas:

Para o TCP: `awk -f trace_analyzer.awk type=tcp src=0 dest=6 flow=2 out.tr`

Para o UDP: `awk -f trace_analyzer.awk type=cbr src=0 dest=6 flow=2 out.tr`

Utilizamos sempre um flow de 2 pois, tal como está referido nos slides, é o valor que estava definido para o IPv6.

Exercício 2

Neste exercício era-nos pedido que preenchesse-mos a uma tabela com os seguintes valores:

Tamanho por omissão das filas dos nós	50 Pacotes
Tamanho por omissão dos pacotes TCP	1000 bytes
Tamanho por omissão dos pacotes UDP	1000 bytes
Tamanho por omissão da janela do TCP	20
Valor por omissão da fila no Servidor A	3146 bytes

Todos estes valores foram retirados do ficheiro “./ns-2.35/tcl/lib/ns-default.tcl”. As instruções seguintes (presentes no ficheiro ns-default.tcl”) são aquelas que nos permitiram completar a tabela:

- Queue set limit_ 50
- Agent/TCP set packetSize_ 1000
- Agent/UDP set packetSize_ 1000
- Agent/TCP set window_ 20
- 3MB(3145729 bytes)/1000 bytes arredondado dá 3146 bytes.

Exercício 3

3.1- Para este exercício apenas necessário iniciar a simulação para cada um dos casos e analisar as estatísticas obtidas com a função `trace_analyzer.awk`, que podem ser observadas na tabela seguinte:

TCP			UDP	
Temp min	Janela min	Nº Pacotes Perdidos	Temp min	Nº Pacotes Perdidos
2.869846 s	105	0	0.3143 s	2782

Para chegarmos à conclusão de que a janela mínima (uma janela que leve a uma perda de pacotes mínima e um tempo de transmissão mínimo) era 105 tivemos de a procurar por pesquisa binária, ou seja, testamos um valor, seguidamente testamos o valor metade do inicial; se esse for ‘pior’ que o inicial, continuamos a procura no intervalo superior, caso contrário, continuamos no intervalo inferior, e assim sucessivamente até não podermos dividir mais. Este método aplica-se devido à relação aproximadamente linear entre o tamanho da janela e a velocidade de transferência, bem como a relação, também aproximadamente linear, entre o tamanho da janela e o número de pacotes perdidos. Deste modo, uma vez num cenário em que não há perda de pacotes, podemos ter garantias de que a melhor janela será a anterior à que cause a perda dos primeiros pacotes.

Apresentamos então alguns dos valores que obtemos com este método:

TCP (sem filas ou quebras) Cenário 1			
Tempo de Propagação	Nº de Pacotes Perdidos	Janela	Tempo de Transmissão
10 ms	0	20	7,549 s
10 ms	0	50	3.1967 s
10 ms	0	100	2.869851 s
10 ms	0	105	2.869846 s
10 ms	1	106	3.105644 s

Considerando que o número de pacotes na fila é de 3146 (pois temos um “Servidor A” que vai enviar a um “Receptor” um bloco de dados de 3MB, ou seja, 3145728 bytes. O tamanho por omissão dos pacotes é de 1000 bytes logo, $3145729/1000=3145.729$, que arredondado fica então 3146) existe uma perda de quase 85% dos pacotes no caso do UDP, mesmo que o tempo de transmissão seja muito mais reduzido que o do TCP, como este por comparação tem uma taxa de perda de pacotes de 0%, é muito mais vantajoso usar o TCP neste caso.

3.2- Neste exercício é-nos pedido que quebrems a ligação entre o “Router 1” e o “Router 2” no instante 0.6 segundos e que a mantenhamos durante 0.1 segundos, ou seja, até aos 0.7 segundos. Com a função `trace_analyzer.awk` obtivemos os seguintes resultados:

TCP			UDP	
Temp min	Janela min	Nº Pacotes Perdidos	Temp min	Nº Pacotes Perdidos
2.9344934 s	105	0	0.3383 s	2798

A janela ideal foi encontrada mais uma vez por tentativa erro:

TCP (sem filas ou quebras) Cenário 1			
Tempo de Propagação	Nº de Pacotes Perdidos	Janela	Tempo de Transmissão
10 ms	0	20	2,4039 s
10 ms	0	100	2.934939 s
10 ms	0	104	2.934939 s
10 ms	0	105	2,934934 s
10 ms	1	106	3.170738 s

Exercício 4

4.1- Neste exercício é-nos pedido que com o Cenário 2 determinássemos o tempo total de transmissão do bloco de dados entre os “Servidor A” e o “Receptor” usando tanto o TCP como o UDP (para o exercício em questão, a janela do TCP tem o valor 20).

Obtivemos os seguintes resultados:

TCP		UDP	
Tempo min	Nº de Pacotes Perdidos	Tempo min	Nº de Pacotes Perdidos
7.5740 s	0	0.3101 s	2799

4.2- Determinámos o menor tempo total de transmissão do bloco de dados entre os “Servidor A” e o “Receptor” usando TCP e UDP. No caso do TCP voltámos a calcular o menor valor possível da janela de transmissão, alterando ainda a fila do “Router 2” para o valor que o faça perder o menor número de pacotes no caso do UDP. A fila só foi alterada no caso de UDP, visto que no TCP foram usados os valores por omissão.

TCP			UDP		
Temp min	Janela min	Nº de Pacotes Perdidos	Temp min	Nº de Pacotes Perdidos	Tamanho min fila
5.426079 s	36	0	2.660462 s	0	3517

A nossa janela ideal é 36, mais uma vez foi calculada por pesquisa binária:

TCP			
Tempo de Propagação	Nº de Pacotes Perdidos	Janela	Tempo de Transmissão
10 ms	0	20	7,574 s
10 ms	0	30	5,438181 s
10 ms	0	36	5,426079 s
10 ms	2	37	5,616231 s
10 ms	18	50	6,710322 s

O tamanho da nossa fila mínima é 3517 pois é a partir deste valor onde não há perda de pacotes, ou seja, uma fila menor que 3517 tem sempre perda de pacotes e superior a este valor não tem perda.

4.3- Neste exercício foi-nos pedido que quebrássemos a ligação entre o “Router 1” e “Router 2” no instante 0.6 segundos e a mantivéssemos quebrada durante 0.1 segundos e determinássemos o menor tempo total de transmissão do bloco de dados entre o “Servidor A” e o “Receptor” usando TCP e UDP. No caso do TCP usámos o menor valor possível da janela de transmissão. Obtivemos os seguintes valores:

TCP			UDP	
Temp min	Janela min	Nº de Pacotes Perdidos	Temp min	Nº de Pacotes Perdidos
5,492271 s	36	0	0.342899 s	2825

Obtivemos a janela mínima 36 por pesquisa binária:

TCP			
Tempo de Propagação	Nº de Pacotes Perdidos	Janela	Tempo de Transmissão
10 ms	0	20	7,639 s
10 ms	0	30	5,503 s
10 ms	0	36	5,492271 s
10 ms	18	50	7,5069 s

Exercício 5

Neste trabalho foram analisadas as diferenças entre estes dois tipos de protocolos e o impacto que vários cenários e problemas poderiam ter em cada um deles. O TCP é um protocolo que inclui mecanismos para iniciar, manter e finalizar a comunicação (ao contrário do UDP, não é ‘connectionless’) bem como negociar tamanho de pacotes. Estes factores contribuem para evitar o congestionamento da rede (controlo de fluxo) e permitir a retransmissão dos pacotes corrompidos ou perdidos. Deste modo, como até aqui temos visto, o TCP tem a grande vantagem de conseguir detectar e, de algum modo, reparar os erros, bem como controlar o fluxo, lidando com os congestionamentos e com as variações bruscas de conectividade de alguns nós na rede.

Em relação ao UDP, um protocolo ‘connectionless’ (i.e: em que não há uma ligação formal entre dois nós, apenas envio e recepção de pacotes), essas verificações e confirmações não existem, os dados serão transmitidos apenas uma vez (cabe à camada de aplicação, se quiser, implementar mecanismos adicionais no topo do UDP), o que pode dar problemas ao nível de recepção de pacotes, pois nem receptor, nem transmissor conseguem detectar eventuais perdas ou corrupções de pacotes (ainda que em várias implementações o campo de ‘checksum’ do protocolo UDP seja utilizado para fins de detecção de erros). Por outro lado, com a ausência de estruturas mais complexas (o overhead físico do UDP é muito baixo) é garantida uma maior eficiência, como foi possível constatar em exercícios anteriores. De destacar que esta maior eficiência não deve ser encarada como a possibilidade de enviar pacotes de maneira sucessiva o mais rapidamente possível. De facto, tal pode ser prejudicial, sobrecarregando a rede e levando a perdas de pacotes nas filas de espera do router – falta ao UDP o mecanismo de controlo de fluxo que o TCP possui para lidar com estas situações.

Mais concretamente, analisando os resultados obtidos na realização do trabalho, ao longo do exercício 3 e 4 concluímos que só é rentável o uso do UDP na presença de uma stream, ou seja, quando é importante que algo seja enviado em tempo real e a perda de pacotes não seja uma prioridade.

É o que se observa no exercício 4 pois estamos na presença de uma stream de música entre o “Servidor B” e o “Receptor”. Enquanto que no exercício 3 é mais vantajoso usar o TCP pois o tempo de transmissão não é uma prioridade e sim o envio de pacotes sem perdas.

No exercício 3.1 observamos que o envio por UDP apresenta uma perda de pacotes de quase 85% e o TCP de 0%, tendo sido os pacotes enviados sem erros. Como neste caso não se trata de uma stream, é mais vantajoso usar o TCP mesmo que tenha um tempo de transmissão muito superior ao do UDP. No caso do 3.2 como temos agora uma quebra na ligação o tempo de transmissão aumentou tanto no TCP como no UDP e este continua a apresentar uma percentagem muito elevada de perda de pacotes. É tal como referido anteriormente mais rentável usar o TCP outra vez.

No exercício 4.1 já estamos na presença do cenário 2, que tem o tráfego entre o “Servidor B” e o “Receptor” que envia uma stream de uma música. Perdem-se 2799 pacotes, mas como não é prioridade enviar os pacotes com uma percentagem de sucesso de 100% é mais vantajoso usar o UDP neste caso por causa do seu tempo de transmissão muito reduzido. No exercício 4.2 a fila do router 2 foi alterada no caso do UDP para que o menor número de pacotes fosse , chegamos a um tamanho de fila mínimo de 3517, que permitiu que com que nenhum pacote fosse perdido, no TCP também nenhum pacote é perdido mas comparativamente o tempo de transmissão do UDP é inferior, logo este é mais uma vez mais vantajoso. O exercício 4.3 apresenta uma quebra entre o router 1 e o router 2 no instante 0.6 segundos que dura até 0.7 e segundo os valores obtidos embora se percam 2825 pacotes, como a prioridade numa stream é o menor tempo de transmissão possível mais uma vez se verifica que o UDP é o mais rentável em cenários com streams.