



## Ficha 4 – TCP vs UDP

Ano Letivo de 2020/2021

**Data de entrega:** O trabalho deverá ser entregue até dia **08/maio/2021**, no Moodle.

### Defesa:

- A defesa é obrigatória e será realizada na semana de 9/maio/2021, por Zoom;
- É necessária inscrição num dos *slots* de defesa disponibilizados no Nónio;
- Na defesa serão colocadas perguntas sobre o TCP e UDP;

### Descrição do trabalho

Este trabalho pretende analisar e comparar a transmissão de dados usando os protocolos UDP e TCP. Para esta análise foi construída a rede da Fig. 1, a qual vai ser simulada recorrendo ao NS2.

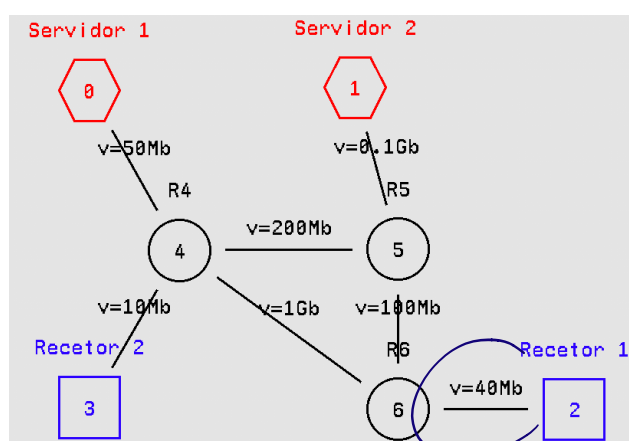
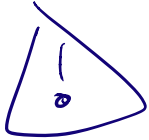


Fig. 1 - Rede

Usando a rede especificada, o “Servidor 1” vai enviar ao “Recetor 1” um bloco de dados de 2MB, que começa a ser transmitido no instante 0.5 segundos. Ao mesmo tempo, entre o “Servidor 2” e os Recetores 1 e 2, existe tráfego que corresponde a uma *stream* de dados que está a ser enviada por UDP (que também começa no instante 0.5 segundos). Dependendo do cenário considerado, a *stream* de dados está ativa ou desligada.

### Características da rede

- Detalhes das ligações:
  - Servidor 1 – Router 4: ligação a 50Mb/s



- Servidor 2 – Router 5: ligação a 0.1Gb/s
- Router 4 - Router 5: Ligação a 200Mb/s
- Router 4 – Router 6: Ligação a 1Gb/s
- Router 5 – Router 6: ligação a 100Mb/s
- Router 6 – Recetor 1: ligação a 40Mb/s
- Router 4 – Recetor 2: Ligação a 10Mb/s
- Os tempos de propagação são todos de 10 ms, com exceção da ligação entre o “Router 6” e o “Recetor 1” que será de 3ms.
- Será usado um protocolo de *routing* dinâmico (rtproto DV).

## Cenários

- **Cenário 1:**
  - Sem tráfego originado no “Servidor 2”.
- **Cenário 2:**
  - 2 *streams* de dados ativas:
    - Servidor 2 – Recetor 1: 3 Mb/s.
    - Servidor 2 – Recetor 2: 3 Mb/s

## Simulação

Será fornecido junto com a ficha um ficheiro de simulação do NS-2 escrito em TCL. Este ficheiro tem a seguinte sintaxe:

```
$ ns Ficha04_simul.tcl {proto} {tcp_w} {stream_r} {break} {s1_r4} {sim_t} {nam}
```

- `proto`: protocolo, que pode ser *tcp* ou *udp*
- `tcp_w`: janela do TCP; no caso de o protocolo ser UDP utilizar 0;
- `stream_r`: velocidade das *streams* de dados;
- `break`: quebra de ligação; pode ser *yes* ou *no*;
- `s1_r4`: velocidade entre o “Servidor 1” e o “Router 4” (0 = velocidade por omissão); pode usar kb, Mb, Gb, ...
- `sim_t`: tempo de simulação
- `nam`: para mostrar o Network Animator (NAM) colocar *on*

## Notas gerais

- Para efeitos de simulação, o “Servidor 1” enviará para o “Recetor 1” um pacote de dados com 2MB (1kB = 1024 bytes).
- O tráfego iniciado no “Servidor 2” será sempre UDP.
- Indique sempre as unidades utilizadas.
- Os tempos de processamento existentes durante a transmissão dos dados são desprezados.
- Justifique as respostas usando os conhecimentos que tem sobre os protocolos TCP e UDP.
- Adeque os tempos de simulação a cada um dos cenários simulados.
- Para analisar mais facilmente os cenários pode recorrer ao ficheiro ‘`trace_analyzer.awk`’ que é fornecido com o enunciado.
- Tempos de simulação altos geram ficheiros de trace grandes. Apague alguns destes ficheiros, não deixando que a VM fornecida fique com o disco completamente cheio!! Mantenha sempre um espaço livre de segurança!
- Dados do NS2 a ter em conta:

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

## Trabalho

1 - Supondo o “Cenário 1”:

\*tcp\* tcp 1 0 NO 0 100.0 ON  
// udp 0 0 NO 0 2.0 ON

1.1 - Determine o menor tempo total de transmissão do bloco de dados entre o “Servidor 1” e o “Recetor 1” usando TCP e UDP. No caso do TCP, use o menor valor possível da janela de transmissão para obter esse tempo. Preencha os resultados na tabela seguinte:

Temp Simulação

$$0.5 + 0.003 + (0.010 \times 2) + 0.4 = 0.913 \text{ seg.}$$

ou (proporção) N° Pacotes (ideia de velocidade)  
 $0,046 \times 2038 = 96,5 \text{ segundos}$   
(sevel 2 s)  
(felte + 0.5 s)

TCP			UDP	
Tempo min	Janela min	Nº pacotes perdidos	Tempo min	Nº pacotes perdidos
97,364894	1	0	0,368368	372

1.2 - Corra a simulação colocando no parâmetro *break* o valor *yes*. Isso irá quebrar a ligação entre o “Router 4” e o “Router 6” no instante 0.6 segundos, e durante 0.1 segundos. Determine o menor tempo total de transmissão do bloco de dados entre o “Servidor 1” e o “Recetor 1” usando TCP e UDP. No caso do TCP, use o menor valor possível da janela de transmissão para obter esse tempo. Preencha os resultados na tabela seguinte:

\*tcp\* (yes)

TCP			UDP	
Tempo min	Janela min	Nº pacotes perdidos	Tempo min	Nº pacotes perdidos
97,405086	1	0	0,368401	424

2 - Supondo o “Cenário 2”:

\*tcp\* tcp 20 3MB NO 0 6.0 ON  
// udp 0 3MB NO 0 2.0 ON

2.1 - Determine o tempo total de transmissão do bloco de dados entre os “Servidor 1” e o “Recetor 1” usando TCP e UDP. Use o TCP com uma janela de transmissão igual a 20. Preencha os resultados na tabela seguinte:

(yes)

TCP		UDP	
Tempo	Nº pacotes perdidos	Tempo	Nº pacotes perdidos
5,035942	0	0,366636	455

2.2 - Determine o tempo total de transmissão do bloco de dados entre os “Servidor 1” e o “Recetor 1” usando TCP e UDP. Use o TCP com uma janela de transmissão igual a 20. Corra a simulação colocando no parâmetro *break* o valor *yes*. Isso irá quebrar a ligação entre o “Router 4” e o “Router 6” no instante 0.6 segundos, e durante 0.1 segundos. Preencha os resultados na tabela seguinte:

TCP		UDP	
Tempo	Nº pacotes perdidos	Tempo	Nº pacotes perdidos
5,076178	0	0,366787	507

2.3 - Determine o menor tempo total de transmissão do bloco de dados entre os “Servidor 1” e o “Recetor 1” usando TCP e UDP. No caso do TCP, use o menor valor possível da janela de transmissão para obter esse tempo. No caso do UDP altere a velocidade da ligação (em múltiplos de 1 Mb) entre o “Servidor 1” e o “Router 4” para o valor que permita perder o menor número de pacotes. A velocidade só será alterada no caso do UDP, quando usar o TCP use os valores por omissão.

\*tcp\*    tcp 1 3MB NO 0 1000 On  
 //        udp 0 3MB NO — 2.0 On  
 ↳ 1MB, 2MB, ...

Preencha os resultados na tabela seguinte:

TCP			UDP		
Tempo min	Janela min	Nº pacotes enviados/recebidos	Tempo min	Nº pacotes perdidos	Velocidade S1-R4
97,370241 seg.	1	2099 / 2099	1,421238 seg.	0	12 Mb

3 – Analise os resultados das perguntas anteriores de modo a comparar a performance entre uma ligação TCP e UDP.

4 – Analise os problemas que a existência de cada uma das 2 streams UDP cria na ligação entre o “Servidor 1” e o “Recetor 1”.

3.

No TCP o tempo de transmissão depende do tamanho da janela (1 versus 20) e que, independentemente da ligação, o TCP garante a entrega dos pacotes (não há pacotes perdidos). No caso do UDP, e independentemente do cenário, o tempo de transmissão não varia, ou seja, há um envio de pacotes constante mesmo que as condições variem, e por isso ocorre perdas de pacotes. As quebras de ligações entre routers é transparente, desde que haja uma rota alternativa com as mesmas condições de tráfego.

4.

Com a existência das streams, dependendo das velocidades de transmissão das ligações, pode sobrecarregar as ligações. Com isto, no caso do UDP (entre o “Servidor 1” e o “Recetor 1”) o número de pacotes perdidos aumentam (apesar de um tempo de transmissão semelhante). O TCP, nestas condições, permanece praticamente igual.

## Trace\_analyzer

Junto com o enunciado será disponibilizado o ficheiro ‘trace\_analyzer.awk’ que lhe permitirá de uma forma rápida obter estatísticas sobre o tráfego enviado. Este ficheiro irá ter como *input* um ficheiro de *trace* do NS. Nesta Ficha, o código de NS fornecido irá gerar 2 ficheiros: *out.tr* e *out.nam*. O primeiro será usado como *input* do trace\_analyzer, o segundo para uso com o Network Animator (NAM).

Para executar o ficheiro awk deverá executar o seguinte comando na janela do terminal:

```
awk -f trace_analyzer.awk type=<tipo_pacote> src=<origem> dest=<destino>
    flow=<fluxo de pacotes> <ficheiro de trace>
```

- type: tipo de pacote (para TCP usar *tcp*, para UDP usar *cbr*)
- flow: colocar o número 1 neste parâmetro
- <ficheiro de trace>: usar o *out.tr*

Exemplos:

```
awk -f trace_analyzer.awk type=cbr src=0 dest=2 flow=1 out.tr
awk -f trace_analyzer.awk type=tcp src=0 dest=2 flow=1 out.tr
```

Como resultado obterá a seguinte informação:

- *Total sent* – número de pacotes enviados da origem em determinado fluxo
- *Total received* – número de pacotes recebidos no destino em determinado fluxo
- *Lost packets* – número de pacotes perdidos
- *Average delay* – média de atraso dos pacotes entre emissor e receptor
- *Total transmission time* - diferença entre o tempo de chegada do último pacote e o tempo do envio do primeiro

### Nota:

Em alguns casos o awk pode ter um comportamento não esperado devido a configurações de local – pode por exemplo estar a usar a vírgula como separador decimal e não o ponto. Caso isto aconteça, use “LC\_ALL=C” antes do comando awk - esta opção irá fazer o set da variável LC\_ALL antes de invocar o awk.

Exemplo:

```
LC_ALL=C awk -f trace_analyzer.awk type=cbr src=0 dest=2 flow=1 out.tr
```

## Ficheiro de trace do NS2

Na figura seguinte mostra-se a estrutura de um ficheiro de trace do NS2.

