

# Introdução às Redes de Comunicação

Ano Letivo de 2016/2017

## Trabalho 1

# Protocolos da Camada de Transporte



Realizado por:  
Renato Santos n.º 2015237457  
João Clara n.º 2015245403

Departamento de Engenharia Informática  
Faculdade de Ciências e Tecnologias da Universidade de Coimbra

Coimbra, 2016

## 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.

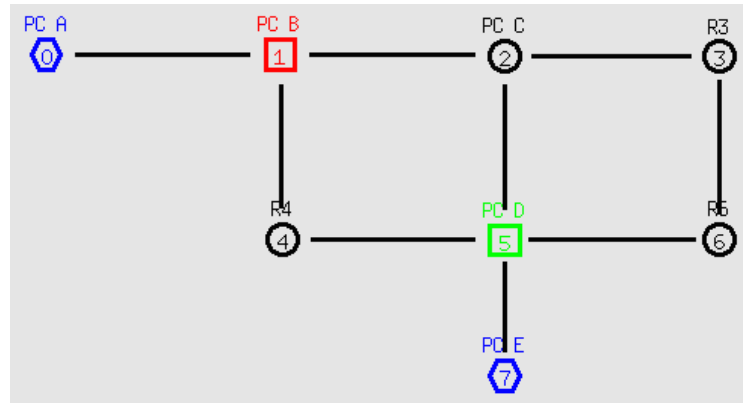


Figura 1 - Rede

Usando a rede especificada, constituída por PCs e *routers*, em que os PCs também fazem o *routing* de pacotes de dados, o “PC A” vai enviar ao “PC E” um bloco de dados de 2MB, que começa a ser transmitido no instante 0.5 segundos. Ao mesmo tempo, desde o “PC B” para o “PC D”, e do “PC D” para o “PC C”, existem *streams* de dados que estão a ser enviadas por UDP (que começa em ambos os casos no instante 0.5 segundos). Dependendo do cenário considerado as *streams* de dados estão activas ou desligadas.

### Características da rede

- Detalhes das ligações:
  - A ligação entre o “R4” e o “PC B” só permite transmissão no sentido “R4”->“PC B”, tem uma velocidade de 10 Mb/s e um tempo de propagação de 5ms
  - Todas as outras ligações são *full-duplex* a 10 Mb/s com tempos de propagação de 10 ms
  - Todas as filas são do tipo *DropTail* com o tamanho por *default*. (Ver Nota 1).
  - Será usado um protocolo de *routing* dinâmico (rtproto LS).

### Cenários

- **Cenário 1:**
  - Apenas tráfego entre o “PC A” e o “PC E”.
- **Cenário 2:**
  - Ao cenário 1 são acrescentadas 2 *streams* de dados UDP:
    - “PC B” -> “PC D”: 6 Mb/s.
    - “PC D” -> “PC C”: 5 Mb/s

## Exercício 1:

### 1.1.

Para minimizarmos o número de ficheiros diferentes (para um só), introduzimos por argumentos de linha os valores necessários para a projeção do cenário desejado. Assim, na linha de comando temos:

No caso de ser UDP:

```
$ns projeto1.tcl <cenário> <protocolo> <quebra>
```

No caso de ser TCP:

```
$ns projeto1.tcl <cenário> <protocolo> <quebra> <janela>
```

Com as seguintes possibilidades:

Cenário: 1 ou 2.

Protocolo: udp ou tcp

Quebra: 1 ou 0, neste caso, sim ou não, respetivamente.

Janela: qualquer valor positivo.

### 1.5.

O valor mínimo possível para a fila no “PC A” é 2098. Ao enviarmos um pacote de 2MB, ou seja, 2097152 bytes, este iria dividir-se durante a simulação deste projeto, pois por default um pacote tem apenas 1000 bytes.

Assim, o número de pacotes enviados será cerca de  $2097152 / 1000 = 2097.152$ . Por isso, o valor mínimo possível será 2098 (valor arredondado para cima).

Definindo assim no código:

```
$ns queue-limit $pca $pcb 2098
```

## Exercício 2:

Valores retirados no ficheiro que se encontra na seguinte diretoria: “./ns-2.35/tcl/lib/ns-default.tcl”.

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

## Exercício 3:

Os valores encontrados nos vários pontos foram recolhidos recorrendo ao ficheiro ‘trace\_analyzer.awk’ que é fornecido com o enunciado.

### NOTA (\*):

\*Obtivemos diferentes valores para os tempos, com exatamente o mesmo código.

Supondo o Cenário 1:

**3.1.** Determinámos o menor tempo total de transmissão do bloco de dados entre o “PC A” e o “PC E” usando TCP e UDP. No caso do TCP, usámos o menor valor possível da janela de transmissão para obter esse tempo.

TCP			UDP	
Tempo min	Janela min	Nº pacotes perdidos	Tempo min	Nº pacotes perdidos
002.264192 002.254160*	101	0	001.720122 001.709322*	0

**3.2.** Quebra de ligação entre o “PC C” e o “PC D” no instante 0.75 segundos até ao instante 0.90 segundos. Determinámos o menor tempo total de transmissão do bloco de dados entre o “PC A” e o “PC E” usando TCP e UDP. No caso do TCP, usamos o menor valor possível da janela de transmissão para obter esse tempo.

TCP			UDP	
Tempo min	Janela min	Nº pacotes perdidos	Tempo min	Nº pacotes perdidos
002.302520 002.292488*	101	0	001.741914 001.731114*	14

#### Exercício 4:

Supondo o Cenário 2:

**4.1.** Determinamos o tempo total de transmissão do bloco de dados entre os “PC A” e o “PC E” usando TCP e UDP. Usamos o TCP com uma janela de transmissão igual a 20.

TCP		UDP	
Tempo	Nº pacotes perdidos	Tempo	Nº pacotes perdidos
009.081232 009.071200*	0	001.736888 001.726088*	780

**4.2.** Determinamos o tempo total de transmissão do bloco de dados entre os “PC A” e o “PC E” usando TCP e UDP. Usamos o TCP com uma janela de transmissão igual a 20. Quebre a ligação entre o “PC C” e o “PC D” no instante 0.75 segundos até ao instante 0.90 segundos.

TCP		UDP	
Tempo	Nº pacotes perdidos	Tempo	Nº pacotes perdidos
009.043775 009.488336*	9	001.733642 001.722842*	837

**4.3.** Determinação do menor tempo total de transmissão do bloco de dados entre os “PC A” e o “PC E” usando TCP e UDP. No caso do TCP, usamos o menor valor possível da janela de transmissão para obter esse tempo. No caso do UDP alteramos a velocidade da ligação (em múltiplos de 1 Mb) entre o “PC A” e o “PC B” para o valor que permita perder o menor número de pacotes. A velocidade só foi alterada no caso do UDP, quando usar o TCP usamos os valores por omissão.

TCP			UDP		
Tempo min	Janela min	Nº pacotes enviados/recebidos	Tempo min	Nº pacotes perdidos	Velocidade “PC A”-“PC B”
006.677 006.666968 *	99	2201 / 2157	004.236642 004.224642 *	0	4Mb

## Exercício 5:

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

A comparação entre os dois protocolos, UDP e TCP, deve ser feita tendo em consideração vários aspetos tais como: o tamanho do bloco a ser transmitido, a estabilidade da rede e número de nós.

Em todos os cenários deste projeto, podemos observar que o protocolo UDP demora menos tempo no envio da informação, isto deve-se à reduzida quantidade de funcionalidades que o UDP possui. Este protocolo simplesmente envia a informação, sendo assim preferível nos casos em que sabemos que não vão haver interferências pelo meio (ex.3.1).

No entanto, quando encontramos um canal em que mais pacotes estão a ser transportados, se a capacidade do canal não for suficiente, vamos perder bastantes pacotes com o UDP. Para isso serve o TCP que implementa um mecanismo de retransmissão em caso de packet-loss, assim quando um pacote é enviado para ter a certeza que não se perde, o TCP espera por um acknowledge de volta, o que vai demorar tempo extra relativamente ao UDP, mas como podemos verificar nos ex3.2, ex4.1 e ex 4.2 a quantidade de pacotes perdidos é bastante inferior.

Concluindo, o protocolo UDP garantirá uma transmissão mais rápida, porém menos viável quanto às perdas de informação, devido a falta de certos mecanismos de proteção. Pelo contrário, o protocolo TCP cobrirá melhor a proteção, garantindo assim uma maior eficácia de transmissão, porém torna-se mais lento em certos casos.

## Exercício 6:

**Analise os problemas causados na ligação entre o “PC A” e o “PC E” pela interferência das 2 streams UDP adicionadas no Cenário 2. Analise a interferência individual de cada uma das streams durante a simulação tendo também em conta a altura em que existe quebra de ligação. Como poderiam esses problemas ser resolvidos?**

Quando não existe quebra de ligação a stream pcb->pcd (vermelha) vai fazer, no caso do UDP, com que os pacotes sejam perdidos no nó pcb pois a fila de espera vai exceder a capacidade do canal (com o TCP se a window for demasiado grande vai fazer perder pacotes também), já a ligação pcd->pcc não vai ter influência devido ao facto de ser na direção contrária.

Quando existe uma quebra nos 0.75s a ligação pcd->pcc(verde) vai ter que tomar outro caminho, que inclui a ligação pcb->pcc, deste modo essa ligação para além dos pacotes enviados pelo pca e pela stream vermelha, vai ter também os pacotes da stream verde na mesma direção, o que vai causar a uma perda ainda maior dos pacotes iniciais (a stream vermelha influencia da mesma maneira com ou sem quebra).

Estes problemas poderiam ser resolvidos de 2 maneiras, ou aumentar a capacidade das ligações congestionadas, ou aumentar as filas de espera nos nós dessas ligações.