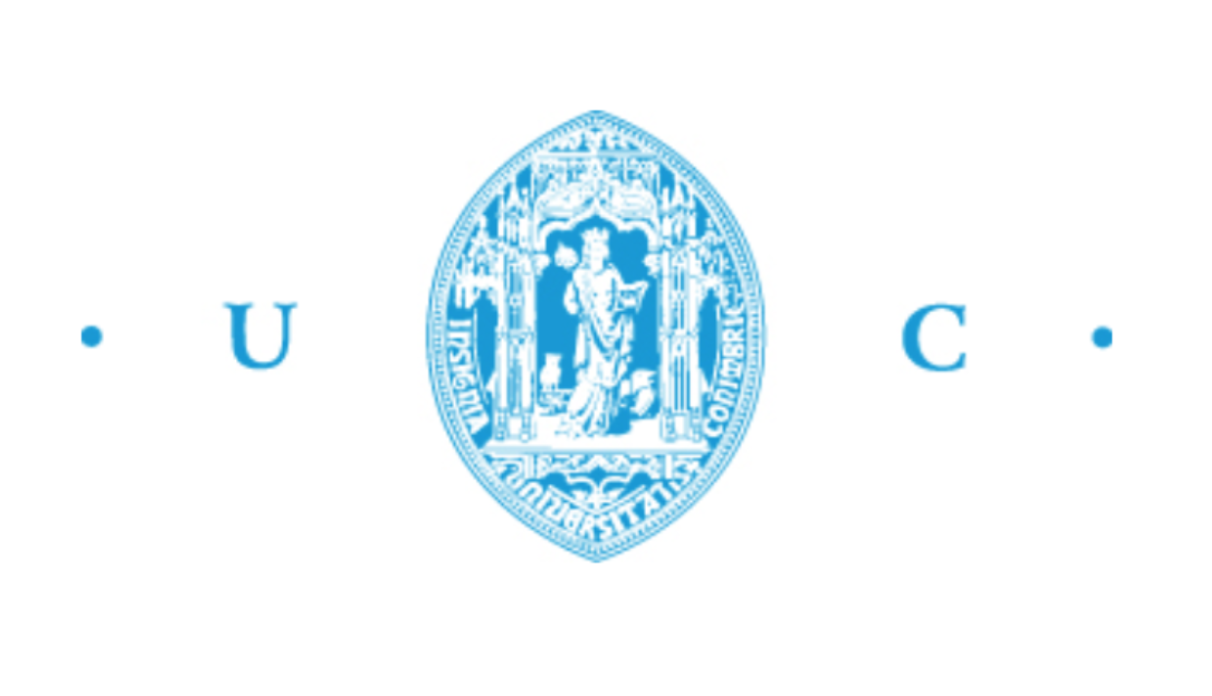
UNIVERSIDADE DE COIMBRA

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

INTRODUÇÃO ÀS REDES DE COMUNICAÇÃO

RELATÓRIO TRABALHO 1



Novembro, 2016

Autores:

Teresa Salazar, 2015234237

Gonçalo Amaral

**INTRODUÇÃO**

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

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.

Dependendo do cenário, poderá haver um envio adicional de informação por UDP.

O Protocolo UDP (User Datagram Protocol) é um protocolo de envio de informação que, entre os dois protocolos, é o mais rápido e mais simples pois não fornece garantia na entrega dos pacotes.

O Protocolo TCP (Transmission Control Protocol) é mais seguro porque, ao contrário

do Protocolo UDP, existe um checksum que vai actualizando o Receptor da informação que já recebeu.

**NOTAS**

**Argv0: Cenário – Permite escolher o cenário.**

**Argv1: Protocolo – Permite alternar o protocolo usado - TCP ou UDP.**

**Argv2: Janela – Permite definir a janela de transmissão para envios para o protocolo do tipo TCP.**

**Argv3: Quebra – Permite escolher se queremos activar a quebra de liagação entre “PC C” e o “PC D” – 0 ou 1.**

**Argv4: velocidade – Permite escolher a velocidade de ligação entre o “PC A” e o “PC B”. Para a todos os exercícios, exceto o 4.3, deverá ter o valor de 10Mb.**

**Para correr o projecto é necessário correr o comando:**

**ns project.tcl <cenario> <protocol> <window> <break> <velocidade>**

**Para chamar o trace\_analyzer fomos modificando os valores de type - cbr ou tcp caso a**

**ligação seja udp ou tcp. O float, source e destination são sempre 1, 0 e 7.**

**EXERCÍCIO 2**

|  |  |
| --- | --- |
| 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 3.1**

**ns project.tcl 1 udp 0 0 10Mb**

**awk -f trace\_analyzer.awk type=cbr src=0 dest=7 flow=1 out.tr**

**Statistics for cbr from node 0 to 7 in flow 1**

**Total sent: 2098**

**Total received: 2098**

**Lost packets: 0**

**Average delay: 000,859867**

**Total transmission time: 002,000000**

**No caso do TCP, a janela influencia o resultado. Para janelas abaixo de 34, o número total de pacotes enviados não era enviado. No entanto, à medida que a janela aumenta, o Average delay e o total transmission time variam.**

**Após várias tentativas, a janela que minimiza o tempo total de transmissão do bloco de dados entre o “PC A” e o “PC E” sem perda de pacotes é 88.**

**ns project.tcl 1 tcp 88 0 10Mb**

**awk -f trace\_analyzer.awk type=tcp src=0 dest=7 flow=1 out.tr**

Statistics for tcp from node 0 to 7 in flow 1

Total sent: 2099

Total received: 2099

Lost packets: 0

Average delay: 000,021915

Total transmission time: 002,000000

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TCP | | | UDP | |
| Tempo min | Janela min | Nº pacotes perdidos | Tempo min | Nº pacotes perdidos |
| **002,000000** | 88 | 0 | **002,000000** | 0 |

**EXERCÍCIO 3.2**

Statistics for cbr from node 0 to 7 in flow 1

Total sent: 2098

Total received: 1304

Lost packets: 794

Average delay: 000,879601

Total transmission time: 002,000000

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TCP | | | UDP | |
| Tempo min | Janela min | Nº pacotes perdidos | Tempo min | Nº pacotes perdidos |
|  |  |  | 002,000000 | 794 |

**EXERCÍCIO 4.1**

**ns project.tcl 2 udp 0 0 10Mb**

**awk -f trace\_analyzer.awk type=cbr src=0 dest=7 flow=1 out.tr**

Statistics for cbr from node 0 to 7 in flow 1

Total sent: 2098

Total received: 1318

Lost packets: 780

Average delay: 000,872534

Total transmission time: 002,000000

ns project.tcl 2 tcp 20 0 10Mb

awk -f trace\_analyzer.awk type=tcp src=0 dest=7 flow=1 out.tr

Statistics for tcp from node 0 to 7 in flow 1

Total sent: 1251

Total received: 1245

Lost packets: 6

Average delay: 000,011245

Total transmission time: 005,000000

|  |  |  |  |
| --- | --- | --- | --- |
| TCP | | UDP | |
| Tempo | Nº pacotes perdidos | Tempo | Nº pacotes perdidos |
| 5,000000 | 6 | 2,000000 | 780 |

**EXERCÍCIO 4.2**

**ns project.tcl 2 udp 0 1 10Mb**

**awk -f trace\_analyzer.awk type=cbr src=0 dest=7 flow=1 out.tr**

**Statistics for cbr from node 0 to 7 in flow 1**

**Total sent: 2098**

**Total received: 1304**

**Lost packets: 794**

**Average delay: 000,879601**

**Total transmission time: 002,000000**

**ns project.tcl 2 tcp 20 1 10Mb**

**awk -f trace\_analyzer.awk type=tcp src=0 dest=7 flow=1 out.tr**

**Statistics for tcp from node 0 to 7 in flow 1**

**Total sent: 1101**

**Total received: 1087**

**Lost packets: 14**

**Average delay: 000,036799**

**Total transmission time: 005,000000**

|  |  |  |  |
| --- | --- | --- | --- |
| TCP | | UDP | |
| Tempo | Nº pacotes perdidos | Tempo | Nº pacotes perdidos |
| **005,000000** | 14 | **002,000000** | **794** |

**EXERCÍCIO 4.3**

**ns project.tcl 2 udp 0 0 39Mb**

**awk -f trace\_analyzer.awk type=cbr src=0 dest=7 flow=1 out.tr**

**Statistics for cbr from node 0 to 7 in flow 1**

**Total sent: 2098**

**Total received: 507**

**Lost packets: 1591**

**Average delay: 000,000000**

**Total transmission time: 000,000000**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TCP | | | UDP | | |
| Tempo min | Janela min | Nº pacotes enviados/recebidos | Tempo min | Nº pacotes  perdidos | Velocidade |
|  |  |  | 0,000 | 1591 | 39Mb |

**EXERCÍCIO 5**

**Primeiramente, confirma-se que o Protocolo UDP é menos eficiente pois existe perda de pacotes. Prova-se então a importância da presença do checksum no protocolo TCP para garantir o envio da informação.**

**Como consequência, o Protocolo UDP consegue reduzir o seu tempo de execução, enquanto que o TCP tem que confirmar qual a informação que já foi recebida e assim perde mais tempo.**

**EXERCÍCIO 6**

**O problema surge com o facto de o PC E ter de manusear mais informação. Isto faz com que o PC E tenha mais trabalho, logo mais tempo de execução e maior**

**probabilidade de perda de pacotes.**

**A janela mínima do checksum (no TCP) tem que ser maior para que haja o mínimo de reenvio de pacotes, e a velocidade de envio de pacotes (no UDP) tenha de ser menor, de modo a conseguir minimizar perda de pacotes.**

**Além disso, o facto de ambas as streams passarem em nós comuns faz com eles tenham de diferenciar a informação e de onde vem.**

**Esta espera e alternância de vários pacotes vai fazer com que a fila de receção de pacotes aumente significativamente.**

**As soluções para resolver estes problemas serão: diminuir a velocidade de envio dos pacotes para a fila não crescer substancialmente e não existir uma maior perda de pacotes; aumentar o tamanho da fila, permitindo assim uma maior acumulação de pacotes, contribuindo para a não perda dos mesmos.**