

UDP x TCP

Nome: Luiza Kuze Gomes

Disciplina: RCO786202

Transferência utilizando o protocolo TCP

10. No terminal do Receptor, após o término do processo, verifique o tamanho do arquivo recebido com o comando "ls -l".

1. O tamanho é igual ao do arquivo seq_num.txt?

Sim, 5327160 bytes.

```
root@Receptor:/# ls -l
total 6604
-rw-r--r-- 1 root root 5327160 Nov 26 13:56 arquivoTCP
```

2. Quanto tempo levou para transmiti-lo?

17.173 segundos.

```
IMUNES: Transmissor (console) bash
root@Transmissor:/# time nc -w1 -vvn 10.0.0.21 5555 < seq_num.txt
(UNKNOWN) [10.0.0.21] 5555 (?) open
net timeout
sent 5327160, rcvd 0

real    0m17.173s
user    0m0.000s
sys     0m0.018s
root@Transmissor:/#

IMUNES: Receptor (console) bash
root@Receptor:/# nc -vvn1 -p 5555 > arquivoTCP
listening on [any] 5555 ...
connect to [10.0.0.21] from (UNKNOWN) [10.0.0.20] 51920
sent 0, rcvd 5327160
root@Receptor:/#
```

11. Analisando a captura de pacotes do WireShark responda:

ip.addr==10.0.0.21						
No.	Time	Source	Destination	Protocol	Length	Info
9	257.236729	10.0.0.20	10.0.0.21	TCP	74	51920 → 5555 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3521755089 TSecr=0 WS=128
10	257.237411	10.0.0.21	10.0.0.20	TCP	74	5555 → 51920 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2098509345 TSecr=3521755089 WS=128
11	257.237994	10.0.0.20	10.0.0.21	TCP	66	51920 → 5555 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3521755091 TSecr=2098509345
12	257.298593	10.0.0.20	10.0.0.21	TCP	7306	51920 → 5555 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=7240 TSval=3521755091 TSecr=2098509345
13	257.299344	10.0.0.21	10.0.0.20	TCP	66	5555 → 51920 [ACK] Seq=1 Ack=7241 Win=61312 Len=0 TSval=2098509407 TSecr=3521755091
14	257.306770	10.0.0.20	10.0.0.21	TCP	1018	51920 → 5555 [PSH, ACK] Seq=7241 Ack=1 Win=64256 Len=952 TSval=3521755091 TSecr=2098509345
15	257.307377	10.0.0.21	10.0.0.20	TCP	66	5555 → 51920 [ACK] Seq=1 Ack=8193 Win=64128 Len=0 TSval=2098509415 TSecr=3521755091
16	257.355258	10.0.0.20	10.0.0.21	TCP	5858	51920 → 5555 [PSH, ACK] Seq=8193 Ack=1 Win=64256 Len=5792 TSval=3521755091 TSecr=2098509345
17	257.355915	10.0.0.21	10.0.0.20	TCP	66	5555 → 51920 [ACK] Seq=1 Ack=13985 Win=62080 Len=0 TSval=2098509464 TSecr=3521755091
18	257.367368	10.0.0.20	10.0.0.21	TCP	1514	51920 → 5555 [ACK] Seq=13985 Ack=1 Win=64256 Len=1448 TSval=3521755105 TSecr=2098509345
19	257.368924	10.0.0.21	10.0.0.20	TCP	66	5555 → 51920 [ACK] Seq=1 Ack=15433 Win=64128 Len=0 TSval=2098509476 TSecr=3521755105
20	257.427894	10.0.0.20	10.0.0.21	TCP	7306	51920 → 5555 [PSH, ACK] Seq=15433 Ack=1 Win=64256 Len=7240 TSval=3521755153 TSecr=2098509407

1. Quais as portas origem e destino escolhidas pelo cliente e servidor?

Cliente (Transmissor): Porta de origem é 51920 e porta de destino é 5555.

Servidor (Receptor): Porta de origem é 5555 e porta de destino é 51920.

2. Qual é o número de sequência, para ambas as máquinas, do primeiro e do último pacote?

Primeiro pacote

Para o cliente: Seq=0

Para o servidor: Seq=0

Último pacote

Para o cliente: Seq=5327162

Para o servidor: Seq=1

3. Qual é o número de sequência, para ambas as máquinas, do primeiro e do último ACK?

Primeiro ACK

Para o cliente: Ack=1

Para o servidor: Ack=1

Último ACK

Para o cliente: Ack=2

Para o servidor: Ack=5327162

4. Calcule e mostre o procedimento de cálculo do tamanho do arquivo pela análise dos pacotes? Qual é a maneira mais fácil? Apresente os cálculos ou descreva a maneira de obtenção do valor. Dica: observe o primeiro e o último número de sequência e faça uma correlação com o tamanho do arquivo.

Tamanho do arquivo=(Seq final)+(Len do último segmento)

Tamanho do arquivo=(5327162)+(1912)=5327160bytes

3580	301.782379	10.0.0.20	10.0.0.21	TCP	2962	51920 → 5555 [PSH, ACK] Seq=5322353 Ack=1 Win=64256 Len=2896 TSval=35
3581	301.798767	10.0.0.20	10.0.0.21	TCP	1978	51920 → 5555 [FIN, PSH, ACK] Seq=5325249 Ack=1 Win=64256 Len=1912 TSv

5. Qual é o tamanho do último segmento de dados recebido? Perceba que ele é diferente dos demais, que vem "cheios", que tem tamanho grandes.

1912 bytes.

3580	301.782370	10.0.0.20	10.0.0.21	TCP	2962	51920 → 5555	[PSH, ACK] Seq=5322353 Ack=1 Win=64256 Len=2896 TSval=3521786156 TSecr=2098540410
3581	301.798767	10.0.0.20	10.0.0.21	TCP	1978	51920 → 5555	[FIN, PSH, ACK] Seq=5325249 Ack=1 Win=64256 Len=1912 TSval=3521786156 TSecr=2098540410
3584	301.799584	10.0.0.21	10.0.0.20	TCP	66	5555 → 51920	[ACK] Seq=1 Ack=4993657 Win=3144448 Len=0 TSval=2098551117 TSecr=3521784303
3587	301.800130	10.0.0.21	10.0.0.20	TCP	66	5555 → 51920	[ACK] Seq=1 Ack=4996553 Win=3144448 Len=0 TSval=2098551141 TSecr=3521784303

6. Apresente os segmentos do 3-way handshake e analise os campos do cabeçalho, que os identificam. Estão de acordo com a norma apresentada na literatura (em sala de aula)?

Está conforme o que foi visto em aula.

O sistema final do cliente envia o segmento SYN do TCP ao servidor. Servidor recebe FIN, responde com ACK Fecha conexão, envia FIN (FINACK). O cliente recebe FINACK, responde com ACK.

ip.addr==10.0.0.21						
No.	Time	Source	Destination	Protocol	Length Info	
9	257.236720	10.0.0.20	10.0.0.21	TCP	74	51920 → 5555 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3521755089 TSecr=0 WS=128
10	257.237411	10.0.0.21	10.0.0.20	TCP	74	5555 → 51920 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2098509345 TSecr=3521755089 WS=128
11	257.237994	10.0.0.20	10.0.0.21	TCP	66	51920 → 5555 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3521755091 TSecr=2098509345

7. Apresente os segmentos do fechamento de conexão e analise os campos do cabeçalho, que os identificam. Estão de acordo com a norma apresentada na literatura (em sala de aula)?

Está conforme o que foi visto em aula.

O sistema final do cliente envia o segmento de controle TCP FIN ao servidor. O servidor recebe FIN, responde com ACK. Fecha a conexão, envia FIN. O cliente recebe FIN ACK e responde com ACK.

3700	301.858815	10.0.0.21	10.0.0.20	TCP	66	5555 → 51920	[ACK] Seq=1 Ack=5325249 Win=3144448 Len=0 TSval=2098553891 TSecr=3521786156
3701	301.859367	10.0.0.21	10.0.0.20	TCP	66	5555 → 51920	[ACK] Seq=1 Ack=5327162 Win=3144448 Len=0 TSval=2098553907 TSecr=3521786156
3702	301.859815	10.0.0.21	10.0.0.20	TCP	66	5555 → 51920	[FIN, ACK] Seq=1 Ack=5327162 Win=3144448 Len=0 TSval=2098553907 TSecr=3521786156
3703	301.860429	10.0.0.20	10.0.0.21	TCP	66	51920 → 5555	[ACK] Seq=5327162 Ack=2 Win=64256 Len=0 TSval=3521799713 TSecr=2098553907
3580	301.782370	10.0.0.20	10.0.0.21	TCP	2962	51920 → 5555	[PSH, ACK] Seq=5322353 Ack=1 Win=64256 Len=2896 TSval=3521786156 TSecr=2098540410
3581	301.798767	10.0.0.20	10.0.0.21	TCP	1978	51920 → 5555	[FIN, PSH, ACK] Seq=5325249 Ack=1 Win=64256 Len=1912 TSval=3521786156 TSecr=2098540410
3584	301.799584	10.0.0.21	10.0.0.20	TCP	66	5555 → 51920	[ACK] Seq=1 Ack=4993657 Win=3144448 Len=0 TSval=2098551117 TSecr=3521784303
3587	301.800130	10.0.0.21	10.0.0.20	TCP	66	5555 → 51920	[ACK] Seq=1 Ack=4996553 Win=3144448 Len=0 TSval=2098551141 TSecr=3521784303

Transferência utilizando o protocolo UDP

1. No terminal do Receptor, verifique o tamanho do arquivo recebido com o comando "ls -l".

```
IMUNES: Transmissor (console) bash
root@Transmissor:/# time nc -w1 -vvnu 10.0.0.21 6666 < seq_num.txt
(UNKNOWN) [10.0.0.21] 6666 (?) open
net timeout
sent 5327160, rcvd 0

real    0m2.028s
user    0m0.000s
sys     0m0.028s

IMUNES: Receptor (console) bash
root@Receptor:/# nc -vvnl -p 6666 > arquivoUDP
listening on [any] 6666 ...
connect to [10.0.0.21] from (UNKNOWN) [10.0.0.20] 57241
```

1. O tamanho é igual ao do arquivo seq_num.txt?

```
IMUNES: Receptor (console) bash
root@Receptor:/# ls -l
total 6604
-rw-r--r-- 1 root root 5327160 Nov 26 13:56 arquivoTCP
-rw-r--r-- 1 root root 1359872 Nov 26 15:30 arquivoUDP
```

Não, é bem menor.

2. Quanto tempo levou para transmiti-lo?

2.028s

2. Analisando a captura de pacotes do WireShark responda:

1. Qual é o identificador (número de sequência) do primeiro e do último pacote? Existe?

Não existe.

2. É possível calcular o tamanho do arquivo pela análise dos pacotes? É mais fácil ou difícil que no caso da transferência via TCP?

Mais difícil, é necessário realizar o somatório do campo Len de cada um dos pacotes.

3. Compare as transferências feitas com os protocolos TCP e UDP em relação, principalmente, ao tempo gasto para transmitir o arquivo e a integridade de dados.

1. O que eles têm em comum?

Ambos dividem os dados em pacotes menores e assim conseguem transmitir dados entre dispositivos.

2. Que diferenças lhe pareceram mais pronunciadas?

Tempo gasto: UDP mais rápido, já que não há confirmação do recebimento.

Controle de conexão: TCP é orientado à conexão, enquanto o UDP não é.

3. Como isso deve afetar as aplicações que usam esses protocolos?

As aplicações que utilizam o TCP tem confiabilidade garantida na transmissão, porém podem ter um tempo de transmissão maior. Já as aplicações que utilizam o UDP tem transferência rápida, porém com potencial perda de dados.