

Nome: Nathan Medeiros Cristiano.

Turma: RED129005

LABORATÓRIO 8

USANDO MINHA MÁQUINA / IFSC

TCP X UDP

Transferência utilizando o protocolo TCP.

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

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

```
-rw-r--r-- 1 root root 5327160 Oct 13 13:31 arquivoTCP
```

- Quanto tempo levou para transmiti-lo?

```
real    0m46.728s
user    0m0.004s
sys     0m0.031s
```

1.1- Analisando a captura de pacotes do WireShark responda:

- Quais as portas origem e destino escolhidas pelo cliente e servidor?

```
Src Port: 34294, Dst Port: 5555
```

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

CLIENTE:

1º Pacote

```
Sequence Number (raw): 4217614131
```

Último Pacote

```
Sequence Number (raw): 4222941293
```

SERVIDOR:

1º Pacote

```
Sequence Number (raw): 1125776685
```

Último Pacote

```
Sequence Number (raw): 1125776686
```

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

CLIENTE:

PRIMEIRO ACK

```
Sequence Number: 1    (relative sequence number)
Sequence Number (raw): 4217614132
```

ÚLTIMO ACK

```
Sequence Number: 5327162    (relative sequence number)
Sequence Number (raw): 4222941293
```

SERVIDOR:

PRIMEIRO ACK

```
Sequence Number: 1    (relative sequence number)
Sequence Number (raw): 1125776686
```

ÚLTIMO ACK

```
Sequence Number: 1    (relative sequence number)
Sequence Number (raw): 1125776686
```

- Calcule e mostre o procedimento de cálculo do tamanho do arquivo pela análise dos pacotes?

Apenas basta ver o primeiro "sequence number" do cliente somar com o último. Será contado o cabeçalho do TCP também!

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

Tem valor diferente (menor) pois, o último pacote foi o que sobrou da quebra do arquivo em segmentos

6181 80.997008	10.0.0.20	10.0.0.21	TCP	1514 34294 → 5555 [PSH, ACK]
----------------	-----------	-----------	-----	------------------------------

6184 81.045976	10.0.0.20	10.0.0.21	TCP	1482 34294 → 5555 [PSH, ACK]
----------------	-----------	-----------	-----	------------------------------

- 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)?

Sim estão de acordo, apresenta Syn, Syn Ack, e Ack, e os cabeçalhos confirmam o protocolo TCP corretamente.

The image shows a Wireshark packet capture of a TCP SYN packet. The packet list on the left shows 24 packets. The selected packet (No. 5) is a TCP segment from 10.0.0.20 to 10.0.0.21, port 34294 to 5555, with sequence number 0. The packet details pane on the right shows the following fields:

- Frame 5: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface -, id 0
- Ethernet II, Src: 42:00:aa:00:00:00 (42:00:aa:00:00:00), Dst: 42:00:aa:00:00:01 (42:00:aa:00:00:01)
- Internet Protocol Version 4, Src: 10.0.0.20, Dst: 10.0.0.21
- Transmission Control Protocol, Src Port: 34294, Dst Port: 5555, Seq: 0, Len: 0
 - Source Port: 34294
 - Destination Port: 5555
 - [Stream index: 0]
 - [Conversation completeness: Complete, WITH_DATA (31)]
 - [TCP Segment Len: 0]
 - Sequence Number: 0 (relative sequence number)
 - Sequence Number (raw): 4217614131
 - [Next Sequence Number: 1 (relative sequence number)]
 - Acknowledgment Number: 0
 - Acknowledgment number (raw): 0
 - 1010 = Header Length: 40 bytes (10)
 - Flags: 0x002 (SYN)
 - Window: 64240
 - [Calculated window size: 64240]
 - Checksum: 0x1457 [unverified]
 - [Checksum Status: Unverified]
 - Urgent Pointer: 0
 - Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window s
 - [Timestamps]

The packet bytes pane at the bottom shows the raw data for the Sequence Number (tcp.seq), 4 byte(s).

The image shows a Wireshark packet capture of a TCP SYN-ACK packet. The packet list on the left shows 24 packets. The selected packet (No. 6) is a TCP segment from 10.0.0.21 to 10.0.0.20, port 5555 to 34294, with sequence number 0 and acknowledgment number 1. The packet details pane on the right shows the following fields:

- Frame 6: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface -, id 0
- Ethernet II, Src: 42:00:aa:00:00:01 (42:00:aa:00:00:01), Dst: 42:00:aa:00:00:00 (42:00:aa:00:00:00)
- Internet Protocol Version 4, Src: 10.0.0.21, Dst: 10.0.0.20
- Transmission Control Protocol, Src Port: 5555, Dst Port: 34294, Seq: 0, Ack: 1, Len: 0
 - Source Port: 5555
 - Destination Port: 34294
 - [Stream index: 0]
 - [Conversation completeness: Complete, WITH_DATA (31)]
 - [TCP Segment Len: 0]
 - Sequence Number: 0 (relative sequence number)
 - Sequence Number (raw): 1125776685
 - [Next Sequence Number: 1 (relative sequence number)]
 - Acknowledgment Number: 1 (relative ack number)
 - Acknowledgment number (raw): 4217614132
 - 1010 = Header Length: 40 bytes (10)
 - Flags: 0x012 (SYN, ACK)
 - Window: 65160
 - [Calculated window size: 65160]
 - Checksum: 0x1457 [unverified]
 - [Checksum Status: Unverified]
 - Urgent Pointer: 0
 - Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window s
 - [Timestamps]
 - [SEQ/ACK analysis]

The packet bytes pane at the bottom shows the raw data for the Sequence Number (tcp.seq), 4 byte(s).

tcp.stream eq 0						
No.	Time	Source	Destination	Protocol	Length	Info
5	34.609673	10.0.0.20	10.0.0.21	TCP	74	34294 → 5555 [SYN] Seq=
6	34.610942	10.0.0.21	10.0.0.20	TCP	74	5555 → 34294 [SYN, ACK]
7	34.610964	10.0.0.20	10.0.0.21	TCP	66	34294 → 5555 [ACK] Seq=
8	34.611057	10.0.0.20	10.0.0.21	TCP	1514	34294 → 5555 [ACK] Seq=
9	34.611060	10.0.0.20	10.0.0.21	TCP	1514	34294 → 5555 [ACK] Seq=
10	34.611062	10.0.0.20	10.0.0.21	TCP	1514	34294 → 5555 [ACK] Seq=
11	34.611064	10.0.0.20	10.0.0.21	TCP	1514	34294 → 5555 [ACK] Seq=
12	34.611065	10.0.0.20	10.0.0.21	TCP	1514	34294 → 5555 [PSH, ACK]
13	34.611094	10.0.0.20	10.0.0.21	TCP	1514	34294 → 5555 [ACK] Seq=
14	34.611096	10.0.0.20	10.0.0.21	TCP	1514	34294 → 5555 [ACK] Seq=
15	34.611098	10.0.0.20	10.0.0.21	TCP	1514	34294 → 5555 [ACK] Seq=
16	34.611100	10.0.0.20	10.0.0.21	TCP	1514	34294 → 5555 [ACK] Seq=
17	34.611103	10.0.0.20	10.0.0.21	TCP	1514	34294 → 5555 [PSH, ACK]
18	34.620077	10.0.0.20	10.0.0.21	TCP	1514	34294 → 5555 [ACK] Seq=
19	34.624331	10.0.0.21	10.0.0.20	TCP	66	5555 → 34294 [ACK] Seq=
20	34.624362	10.0.0.20	10.0.0.21	TCP	1514	34294 → 5555 [ACK] Seq=
21	34.636941	10.0.0.21	10.0.0.20	TCP	66	5555 → 34294 [ACK] Seq=
22	34.636988	10.0.0.20	10.0.0.21	TCP	1514	34294 → 5555 [ACK] Seq=
23	34.636993	10.0.0.20	10.0.0.21	TCP	1514	34294 → 5555 [PSH, ACK]
24	34.648746	10.0.0.21	10.0.0.20	TCP	66	5555 → 34294 [ACK] Seq=

▶ Frame 7: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface -, id 0
 ▶ Ethernet II, Src: 42:00:aa:00:00:00 (42:00:aa:00:00:00), Dst: 42:00:aa:00:00:01 (42:00:aa:00:00:01)
 ▶ Internet Protocol Version 4, Src: 10.0.0.20, Dst: 10.0.0.21
 ▶ Transmission Control Protocol, Src Port: 34294, Dst Port: 5555, Seq: 1, Ack: 1, Len: 0
 Source Port: 34294
 Destination Port: 5555
 [Stream index: 0]
 [Conversation completeness: Complete, WITH_DATA (31)]
 [TCP Segment Len: 0]
 Sequence Number: 1 (relative sequence number)
 Sequence Number (raw): 4217614132
 [Next Sequence Number: 1 (relative sequence number)]
 Acknowledgment Number: 1 (relative ack number)
 Acknowledgment number (raw): 1125776686
 1000 = Header Length: 32 bytes (8)
 Flags: 0x010 (ACK)
 Window: 502
 [Calculated window size: 64256]
 [Window size scaling factor: 128]
 Checksum: 0x144f [unverified]
 [Checksum Status: Unverified]
 Urgent Pointer: 0
 Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
 [Timestamps]
 [SEQ/ACK analysis]

- 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)?

Sim, está de acordo, pode se verificar isso pelo "Sequence Number" e pelo "Acknowledgment Number"

6189	81.332005	10.0.0.20	10.0.0.21	TCP	66	34294 → 5555 [FIN, ACK]
6190	81.333293	10.0.0.21	10.0.0.20	TCP	66	5555 → 34294 [FIN, ACK]
6191	81.333326	10.0.0.20	10.0.0.21	TCP	66	34294 → 5555 [ACK] Seq=

Transmission Control Protocol, Src Port: 34294, Dst Port: 5555, Seq: 5327161, Ack: 1, Len: 0
 Source Port: 34294
 Destination Port: 5555
 [Stream index: 0]
 [Conversation completeness: Complete, WITH_DATA (31)]
 [TCP Segment Len: 0]
 Sequence Number: 5327161 (relative sequence number)
 Sequence Number (raw): 4222941292
 [Next Sequence Number: 5327162 (relative sequence number)]
 Acknowledgment Number: 1 (relative ack number)
 Acknowledgment number (raw): 1125776686
 1000 = Header Length: 32 bytes (8)
 Flags: 0x011 (FIN, ACK)
 Window: 502
 [Calculated window size: 64256]
 [Window size scaling factor: 128]
 Checksum: 0x144f [unverified]
 [Checksum Status: Unverified]
 Urgent Pointer: 0
 Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
 [Timestamps]

```

6189 81.332005 10.0.0.20 10.0.0.21 TCP 66 34294 → 5555 [FIN, ACK]
6190 81.333293 10.0.0.21 10.0.0.20 TCP 66 5555 → 34294 [FIN, ACK]
6191 81.333326 10.0.0.20 10.0.0.21 TCP 66 34294 → 5555 [ACK] Seq=
Transmission Control Protocol, Src Port: 5555, Dst Port: 34294, Seq: 1, Ack: 5327162, Len: 0
Source Port: 5555
Destination Port: 34294
[Stream index: 0]
[Conversation completeness: Complete, WITH_DATA (31)]
[TCP Segment Len: 0]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 1125776686
[Next Sequence Number: 2 (relative sequence number)]
Acknowledgment Number: 5327162 (relative ack number)
Acknowledgment number (raw): 4222941293
1000 .... = Header Length: 32 bytes (8)
Flags: 0x011 (FIN, ACK)
Window: 4255
[Calculated window size: 544640]
[Window size scaling factor: 128]
Checksum: 0x144f [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[Timestamps]
6189 81.332005 10.0.0.20 10.0.0.21 TCP 66 34294 → 5555 [FIN, ACK]
6190 81.333293 10.0.0.21 10.0.0.20 TCP 66 5555 → 34294 [FIN, ACK]
6191 81.333326 10.0.0.20 10.0.0.21 TCP 66 34294 → 5555 [ACK] Seq=
Source Port: 34294
Destination Port: 5555
[Stream index: 0]
[Conversation completeness: Complete, WITH_DATA (31)]
[TCP Segment Len: 0]
Sequence Number: 5327162 (relative sequence number)
Sequence Number (raw): 4222941293
[Next Sequence Number: 5327162 (relative sequence number)]
Acknowledgment Number: 2 (relative ack number)
Acknowledgment number (raw): 1125776687
1000 .... = Header Length: 32 bytes (8)
Flags: 0x010 (ACK)
Window: 502
[Calculated window size: 64256]
[Window size scaling factor: 128]
Checksum: 0x144f [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[Timestamps]
[SEQ/ACK analysis]
This is a reply to the segment in frame 6190

```

Transferência utilizando o protocolo UDP.

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

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

```

-rw-r--r-- 1 root root 5327160 Oct 13 13:31 arquivoTCP
-rw-r--r-- 1 root root 753667 Oct 13 19:27 arquivoUDP

```

- Quanto tempo levou para transmiti-lo?

```

real    0m2.052s
user    0m0.000s
sys     0m0.035s
root@Transmissor:/#

```

1.1- Analisando a captura de pacotes do WireShark responda:

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

Não existe pois não é um cabeçalho do UDP.

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

O processo é mais chato de se fazer, precisa - se do Length de cada pacote enviado pelo cliente ou servidor, e somar todos eles. É menos confiável.

1.2- 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.

- **O que eles têm em comum?**

Ambos usam portas de Origem e Destino para diferenciação de processos, fazendo multiplexação, permitindo que várias aplicações ou mesmo, funcionem em uma única máquina com um único endereço de IP.

Ambos operam na Camada 4, camada de Transporte. E ambos também, se "envelopam" para entregarem e serem usados pela Camada 3, seja o TCP criando Segmentos ou o UDP criando Datagramas.

- **Que diferenças lhe pareceram mais pronunciadas?**

As principais diferenças que identifiquei. Foram as de confiabilidade, processamento, velocidade, entrega, segmentação e datagramas, e UDP não precisa de "conexão", já o TCP opera com 3-Way Handshake.

- **Como isso deve afetar as aplicações que usam esses protocolos?**

Cada aplicação vai requisitar de um caso de uso diferente, sendo isso é possível ser viabilizado qual protocolo usar para determinada aplicação. Algumas aplicações vão demandar velocidade, sem tanta integridade de dados, outras porventura, vão demandar de extrema confiabilidade dos dados enviados e requisitados.

Podemos ver essa diferenciação comparando as seguintes aplicações, Uma aplicação Bancária e um Streaming de Vídeo.

