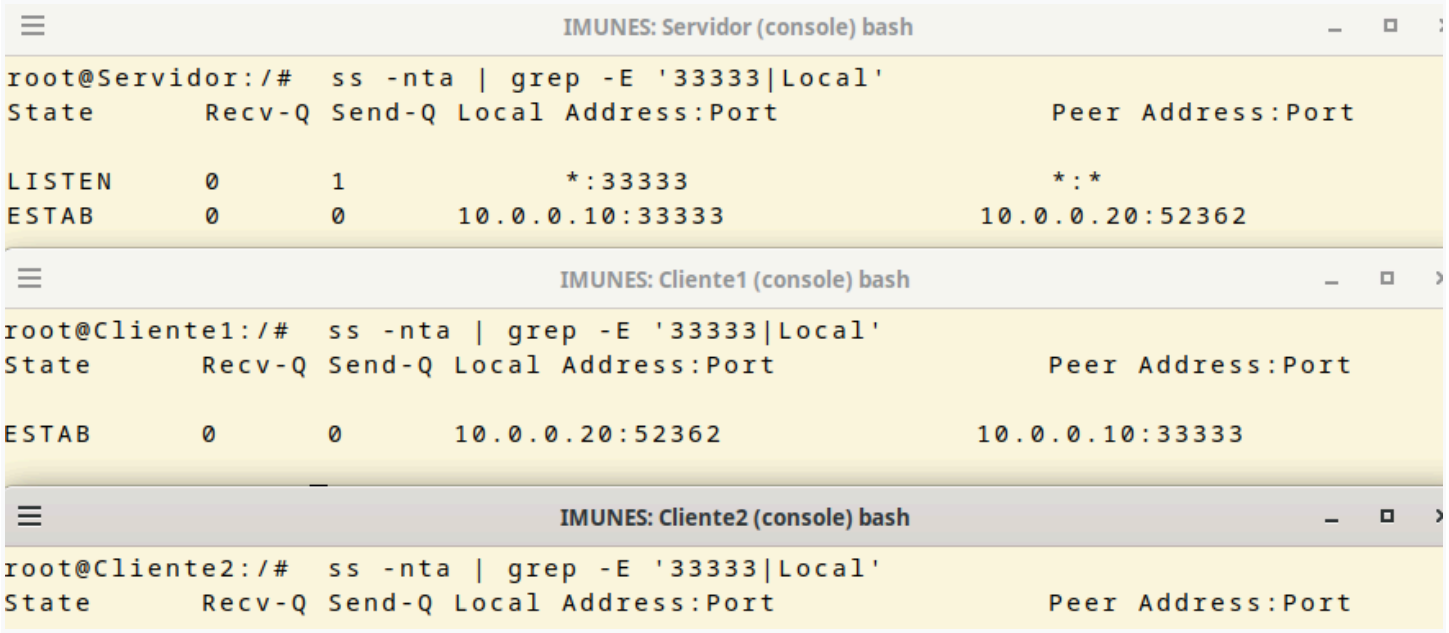
Comparando *sockets* UDP e TCP

Nome: Luiza Kuze Gomes
Disciplina: RCO786202

Programação de sockets com TCP

12. Em novos terminais do Cliente 1 e 2 e Servidor, verifique os *sockets* TCP ativos: `ss -nta | grep -E '33333|Local'`



1. **Você encontrou sockets abertos em todos os hosts (máquinas)?**
   Somente no cliente 1 e servidor.

2. **Quais parâmetros apresentados para cada um deles?**

   Para o servidor:

   i. Socket em estado LISTEN

   - State: Listen (modo de escuta, aguardando novas conexões)

- Recv-Q: 0 (não há dados pendentes)
- Send-Q: 1 (há 1 dado na fila de envio)
- Local Address: *:33333 (endereço local, socket ouvindo na porta 33333)
- Peer Address: *:* (endereço do par, socket)

ii. Socket em estado ESTAB

- State: Listen
- Recv-Q: 0
- Send-Q: 0
- Local Address: 10.0.0.10:33333
- Peer Address: 10.0.0.20:52362

Para o Cliente1:

i. Socket em estado

- State: Listen
- Recv-Q: 0
- Send-Q: 0
- Local Address: 10.0.0.1:52362
- Peer Address: 10.0.0.20:33333

3. Qual a relação entre os *sockets* clientes e servidor, número IP, portas etc?

O Local Address e o Peer Address estão invertidos.

4. Identifique e "printe" o socket receptivo no Servidor.

É a primeira figura do relatório.

15. Inspecione o conteúdo da resposta do servidor, segunda mensagem. O servidor retornou explicitamente o conteúdo do arquivo? Como você pode dizer isso?

tcp.port==33333

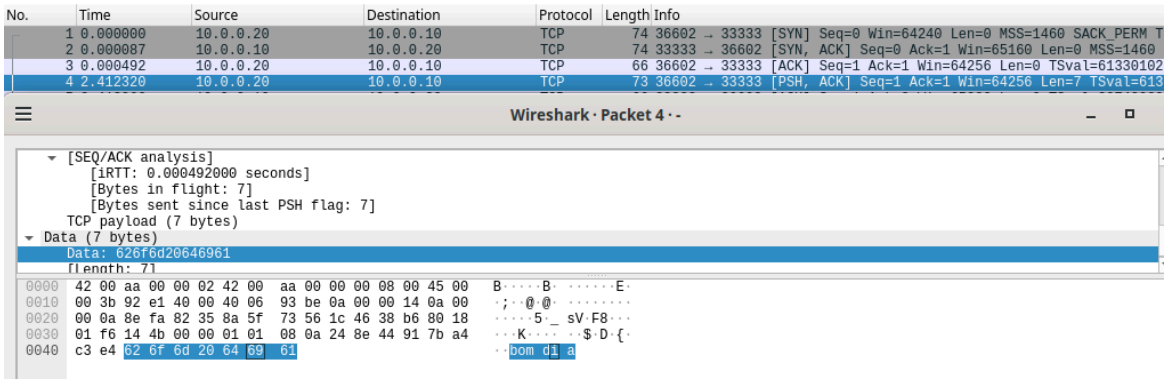
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.20	10.0.0.10	TCP	74	36602 → 33333 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=613301028 TSecr=0 WS=128
2	0.000087	10.0.0.10	10.0.0.20	TCP	74	33333 → 36602 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2074395620 TSecr=61
3	0.000492	10.0.0.20	10.0.0.10	TCP	66	36602 → 33333 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=613301029 TSecr=2074395620
4	2.412320	10.0.0.20	10.0.0.10	TCP	73	36602 → 33333 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=7 TSval=613303441 TSecr=2074395620
5	2.412366	10.0.0.10	10.0.0.20	TCP	66	33333 → 36602 [ACK] Seq=1 Ack=8 Win=65280 Len=0 TSval=2074398032 TSecr=613303441
6	2.412485	10.0.0.10	10.0.0.20	TCP	73	33333 → 36602 [PSH, ACK] Seq=1 Ack=8 Win=65280 Len=7 TSval=2074398032 TSecr=613303441
7	2.412505	10.0.0.20	10.0.0.10	TCP	66	36602 → 33333 [ACK] Seq=8 Ack=8 Win=64256 Len=0 TSval=613303441 TSecr=2074398032
8	2.412590	10.0.0.20	10.0.0.10	TCP	66	36602 → 33333 [FIN, ACK] Seq=8 Ack=8 Win=64256 Len=0 TSval=613303441 TSecr=2074398032
9	2.413248	10.0.0.10	10.0.0.20	TCP	66	33333 → 36602 [FIN, ACK] Seq=8 Ack=9 Win=65280 Len=0 TSval=2074398033 TSecr=613303441
10	2.413336	10.0.0.20	10.0.0.10	TCP	66	36602 → 33333 [ACK] Seq=9 Ack=9 Win=64256 Len=0 TSval=613303442 TSecr=2074398033
13	15.700627	10.0.0.21	10.0.0.10	TCP	74	34804 → 33333 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2435427587 TSecr=0 WS=128
14	15.700708	10.0.0.10	10.0.0.21	TCP	74	33333 → 34804 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=1306674288 TSecr=24
15	15.701046	10.0.0.21	10.0.0.10	TCP	66	34804 → 33333 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2435427589 TSecr=1306674288
16	18.292210	10.0.0.21	10.0.0.10	TCP	75	34804 → 33333 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=9 TSval=2435430180 TSecr=1306674288
17	18.292259	10.0.0.10	10.0.0.21	TCP	66	33333 → 34804 [ACK] Seq=1 Ack=10 Win=65152 Len=0 TSval=1306676880 TSecr=2435430180
18	18.292462	10.0.0.10	10.0.0.21	TCP	75	33333 → 34804 [PSH, ACK] Seq=1 Ack=10 Win=65152 Len=9 TSval=1306676880 TSecr=2435430180
19	18.292557	10.0.0.21	10.0.0.10	TCP	66	34804 → 33333 [ACK] Seq=10 Ack=10 Win=64256 Len=0 TSval=2435430180 TSecr=1306676880
20	18.292629	10.0.0.10	10.0.0.21	TCP	66	33333 → 34804 [FIN, ACK] Seq=10 Ack=10 Win=65152 Len=0 TSval=1306676880 TSecr=2435430180
21	18.292791	10.0.0.21	10.0.0.10	TCP	66	34804 → 33333 [FIN, ACK] Seq=10 Ack=11 Win=64256 Len=0 TSval=2435430180 TSecr=1306676880
22	18.292848	10.0.0.10	10.0.0.21	TCP	66	33333 → 34804 [ACK] Seq=11 Ack=11 Win=65152 Len=0 TSval=1306676880 TSecr=2435430180

1. **Para cada cliente, as três primeiras mensagens trocadas apresentam a camada de aplicação, sim ou não? Explique. O que elas significam (*3-way handshake*)?**

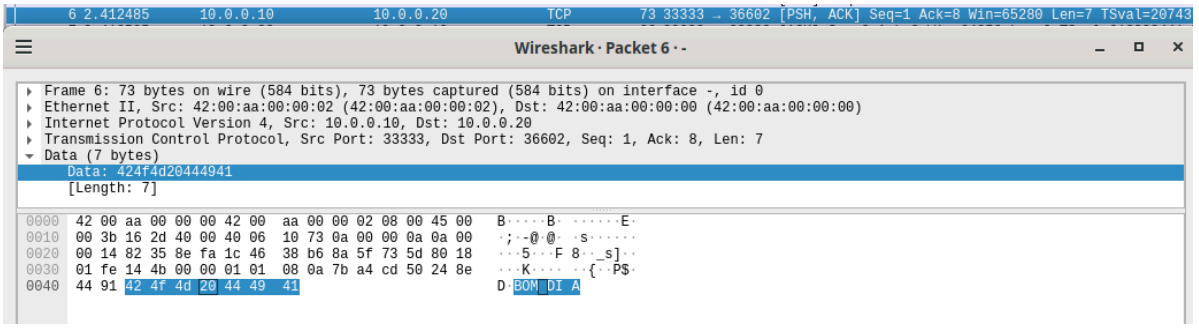
Não, a camada de aplicação começa a ser utilizada após a conexão TCP estar estabelecida. A conexão TCP é estabelecida nas três primeiras mensagens trocadas que fazem parte do 3-way handshake.

2. **Encontre a frase/palavra escrita enviada ao servidor (minúscula) e a resposta em maiúscula?**

   Frase enviada ao servidor (minúscula):



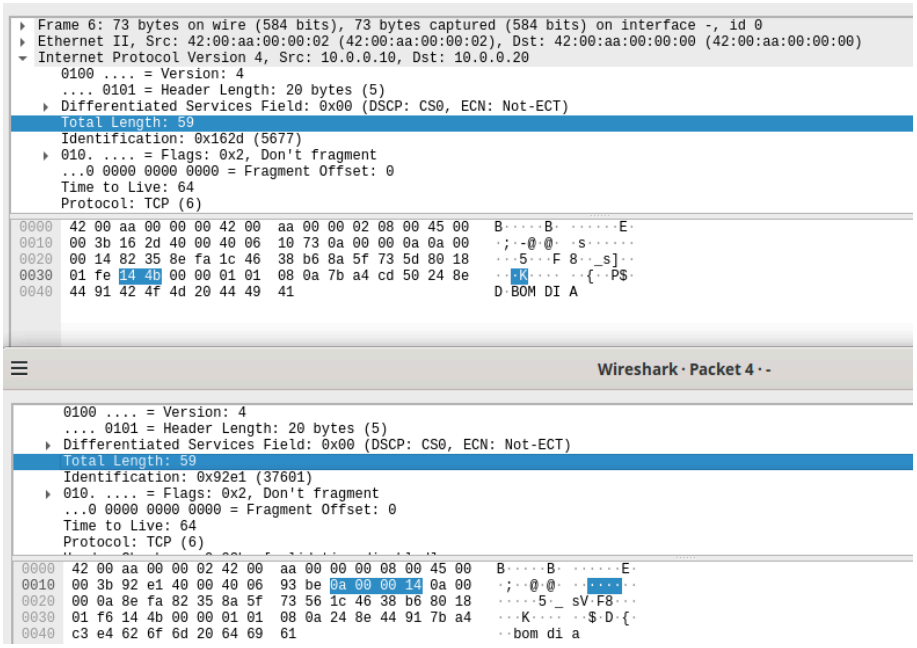
   A resposta (maiúscula):



3. **Qual o tamanho dos pacotes contendo as mensagens: i) *Data* (camada 5) e ii) *Total Length* (camada 3). Qual a relação entre estes valores?**

   i. Data (camada 5): Ambos os pacotes 7 bytes.

   ii. Total Length (camada 3): Ambos 59 bytes



O tamanho dos pacotes na camada de aplicação (camada 5) refere-se ao tamanho da mensagem que o cliente envia ao servidor. O tamanho total (Total Length na camada 3, IP) é a soma do tamanho do conteúdo da camada de aplicação e dos cabeçalhos das camadas de transporte (TCP) e de rede (IP).

4. **As últimas 3 mensagens, de cada cliente, contém o fechamento de conexão, explique-as.**

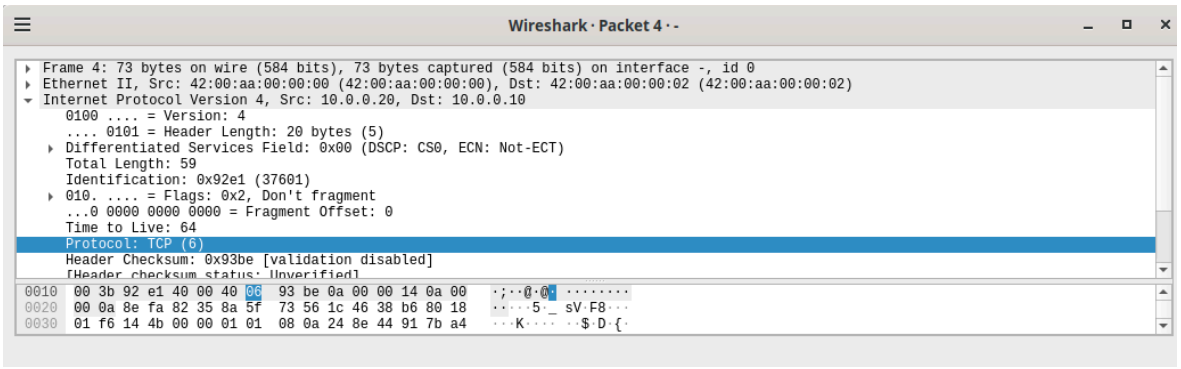
   O cliente envia um pacote FIN para o servidor, indicando que deseja encerrar a conexão, após isso o servidor responde com um ACK para reconhecer o pedido de encerramento do cliente. Ao final, o servidor, ao terminar suas atividades, envia seu próprio pacote FIN para indicar que também deseja encerrar a conexão.

5. **Qual é o protocolo da camada de transporte nessa troca de mensagens?**

   O protocolo TCP.

6. **Qual o número identificador de protocolo TCP no pacote IP? Dica: na janela central abra o campo *Internet Protocol* e procure a string *Protocol***

   Identificador é 6.



Programação de *sockets* com UDP

7. Verifique os sockets em um novo terminal das máquinas Cliente 1 e 2 e Servidor, com o comando: `ss -ua`

IMUNES: Servidor (console) bash

```
root@Servidor:/# ss -ua
State       Recv-Q Send-Q Local Address:Port                 Peer Address:Port

UNCONN      0      0           *:sunrpc                      *:*
UNCONN      0      0           *:642                         *:*
UNCONN      0      0           *:22222                       *:*
UNCONN      0      0          :::sunrpc                     :::*
UNCONN      0      0          :::642                        :::*
```

IMUNES: Cliente2 (console) bash

```
root@Cliente2:/# ss -ua
State       Recv-Q Send-Q Local Address:Port                 Peer Address:Port

ESTAB       0      0      10.0.0.21:59574                  10.0.0.10:22222
```

IMUNES: Cliente1 (console) bash

```
root@Cliente1:/# ss -ua
State       Recv-Q Send-Q Local Address:Port                 Peer Address:Port

ESTAB       0      0      10.0.0.20:36249                  10.0.0.10:22222
```

1. **Identifique e anote os sockets abertos.**

   Do servidor são 5 com state “UNCONN” e 1 com state “ESTAB” para o Cliente1 e Cliente2, conforme imagem acima.

2. **Identifique e anote o socket receptivo do servidor.**

   O socket receptivo no servidor está na porta 22222, com o endereço local * .

14. PERGUNTAS baseadas na captura:

udp.port==22222

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	42:00:aa:00:00:01	Broadcast	ARP	42	Who has 10.0.0.10? Tell 10.0.0.21
2	0.000023	42:00:aa:00:00:02	42:00:aa:00:00:01	ARP	42	10.0.0.10 is at 42:00:aa:00:00:02
3	0.000290	10.0.0.21	10.0.0.10	RTPpro…	50	Request: dia[Malformed Packet]
4	0.000585	10.0.0.10	10.0.0.21	RTPpro…	50	Request: DIA[Malformed Packet]
5	5.078341	42:00:aa:00:00:02	42:00:aa:00:00:01	ARP	42	Who has 10.0.0.21? Tell 10.0.0.10
6	5.078796	42:00:aa:00:00:01	42:00:aa:00:00:02	ARP	42	10.0.0.21 is at 42:00:aa:00:00:01
7	5.310755	10.0.0.20	10.0.0.10	RTPpro…	52	
8	5.311043	10.0.0.10	10.0.0.20	RTPpro…	52	
9	10.454315	42:00:aa:00:00:02	42:00:aa:00:00:00	ARP	42	Who has 10.0.0.20? Tell 10.0.0.10
10	10.454744	42:00:aa:00:00:00	42:00:aa:00:00:02	ARP	42	Who has 10.0.0.10? Tell 10.0.0.20
11	10.454764	42:00:aa:00:00:02	42:00:aa:00:00:00	ARP	42	10.0.0.10 is at 42:00:aa:00:00:02
12	10.454775	42:00:aa:00:00:00	42:00:aa:00:00:02	ARP	42	10.0.0.20 is at 42:00:aa:00:00:00

1. **Em algum momento foi identificado algum procedimento para estabelecimento de conexão?**

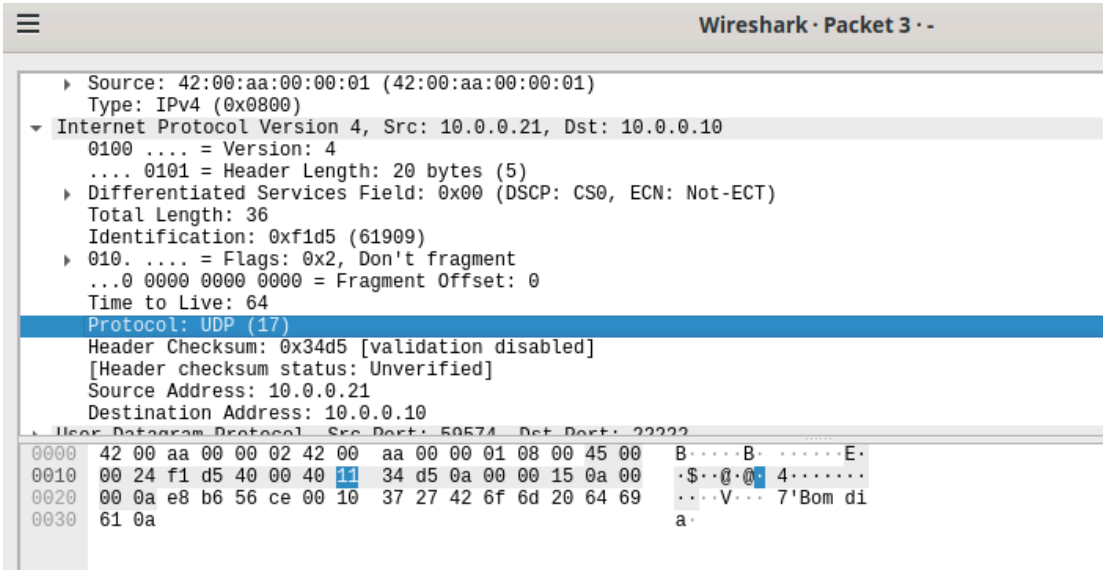
   Não, o protocolo UDP é um protocolo sem conexão (connectionless). Diferentemente do TCP.

2. **Em algum campo do UDP existe numeração de mensagens?**

   Não. O cabeçalho UDP não inclui numeração de sequência.

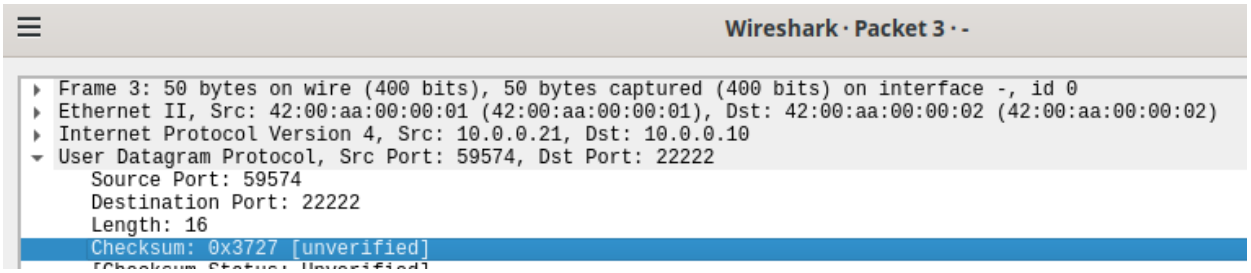
3. **Qual o número identificador de protocolo UDP no pacote IP? Dica: na janela central abra o campo *Internet Protocol* e procure a string *Protocol*.**

   Identificador é 17.



4. **Qual é o *checksum* no pacote (datagrama) UDP? Qual é o formato apresentado? Quantos bits ele possui?**

   O checksum do UDP está na imagem abaixo, ele possui 16 bits. No Wireshark, tem formato hexadecimal.



5. **É possível capturar toda a troca de mensagens e inclusive capturar o texto passado do cliente para o servidor?**

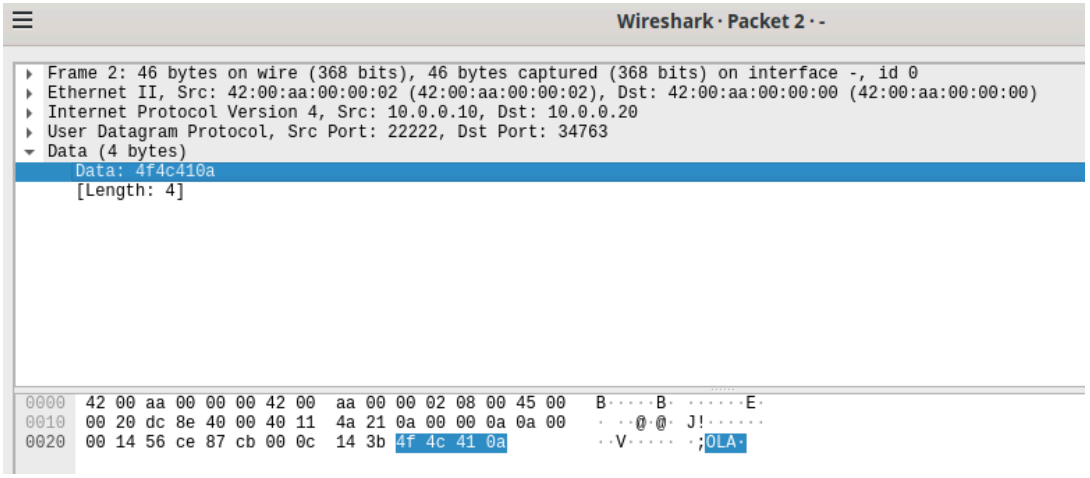
   Sim, pois o UDP não é criptografado.

6. **Qual foi a sequência numérica do campo *Data* em seu teste? Qual o significado?**

Verifiquei que o pacote número 3 estava com “Malformed Packet”, então refiz o procedimento desta sessão para obter essa nova captura do Wireshark:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.20	10.0.0.10	UDP	46	34763 → 22222 Len=4
2	0.000302	10.0.0.10	10.0.0.20	UDP	46	22222 → 34763 Len=4
3	5.127569	42:00:aa:00:00:02	42:00:aa:00:00:00	ARP	42	Who has 10.0.0.20? Tell 10.0.0.10
4	5.127897	42:00:aa:00:00:00	42:00:aa:00:00:02	ARP	42	10.0.0.20 is at 42:00:aa:00:00:00
5	5.127909	42:00:aa:00:00:00	42:00:aa:00:00:02	ARP	42	Who has 10.0.0.10? Tell 10.0.0.20
6	5.127917	42:00:aa:00:00:02	42:00:aa:00:00:00	ARP	42	10.0.0.10 is at 42:00:aa:00:00:02
7	13.296058	10.0.0.21	10.0.0.10	UDP	45	60148 → 22222 Len=3
8	13.296325	10.0.0.10	10.0.0.21	UDP	45	22222 → 60148 Len=3
9	18.439560	42:00:aa:00:00:02	42:00:aa:00:00:01	ARP	42	Who has 10.0.0.21? Tell 10.0.0.10
10	18.440004	42:00:aa:00:00:01	42:00:aa:00:00:02	ARP	42	Who has 10.0.0.10? Tell 10.0.0.21
11	18.440025	42:00:aa:00:00:01	42:00:aa:00:00:02	ARP	42	10.0.0.21 is at 42:00:aa:00:00:01
12	18.440029	42:00:aa:00:00:02	42:00:aa:00:00:01	ARP	42	10.0.0.10 is at 42:00:aa:00:00:02

Aqui consegui verificar o campo Data corretamente:



Este é um número em hexadecimal. “4f 4c 41” representa os caracteres ASCII "OLA" e “0a” é um caractere de nova linha.

7. **Qual é o protocolo da camada de transporte nessa troca de mensagens?**
   O protocolo UDP.

15. Comparativo entre TCP e UDP:

1. **Quantas mensagens foram trocadas entre o servidor e o cliente em cada um dos protocolos para atingir o mesmo objetivo?**

   UDP utilizou 2 mensagens de dados e o TCP utilizou 9 mensagens.

2. **O que justifica a diferença na quantidade de mensagens trocadas?**

   UDP é um protocolo sem conexão e o TCP é um protocolo orientado à conexão. Ao ser orientado à conexão, o TCP utilizará o 3-way handshake e também enviará confirmações (ACKs) para cada pacote recebido. Já o UDP, não requer confirmações de recebimento.

3. **Discuta as vantagens e desvantagens de cada protocolo.**

i. UDP

- Vantagens: Menor atraso que pode se dar por conta de não haver necessidade de estabelecer conexão. Além disso, como não há estados de conexão, há um custo menor.
- Desvantagens: Não é confiável, não há garantia de entrega dos pacotes. Não tem controle de congestionamento, o que pode sobrecarregar a rede com muitos pacotes.

ii. TCP

- Vantagens: Confiabilidade, pois garante a entrega de dados. Controle de congestionamento, o qual ajusta a taxa de envio para evitar sobrecarga.
- Desvantagem: Maior latência, pois o handshake e confirmações aumentam o tempo de resposta.