

INSTITUTO FEDERAL
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

RELATÓRIO TÉCNICO

SOCKETS TCP E UDP

Arthur Cadore Matuella Barcella

TAREFA:

Comparando sockets UDP e TCP

Objetivos

- Entender o conceito de sockets relacionados aos protocolos UDP e TCP.
 - Processos que rodam em máquinas diferentes se comunicam entre si enviando mensagens para sockets. Um processo é semelhante a um prédio e o socket do processo é semelhante a uma porta em seu interior. A aplicação reside dentro do prédio e o protocolo da camada de transporte reside no mundo externo. Um programador de aplicação controla o interior do prédio mas tem pouco (ou nenhum) controle sobre o exterior.
- Simultaneamente explora-se os conceitos relativos aos protocolos UDP e TCP, observando-se a quantidade de mensagens necessárias para a troca de uma simples frase textual.
 - Observa-se a "agilidade" do UDP e a robustez do TCP.
- Por fim, propõe-se um comparativo entre os dois protocolos da camada de transporte: UDP e TCP.

Leia os slides de 1 à 12 e o 58: Capítulo 3 -- Camada de Transporte

Descrição da aplicação a ser desenvolvida em UDP e TCP

- Usaremos a aplicação cliente-servidor simples a seguir para demonstrar a programação de socket:
 - Um cliente lê uma linha de caracteres (dados) do teclado e a envia para o servidor.
 - O servidor recebe os dados e converte os caracteres para maiúsculas.
 - O servidor envia os dados modificados ao cliente.
 - O cliente recebe os dados modificados e apresenta a linha em sua tela.

Programação de sockets com TCP

Capturas da configuração inicial utilizando um notebook com ubuntu 22.04 (lembrando que: o código python precisou ser alterado para a versão do python 3).



INSTITUTO FEDERAL
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

```
cadore@cadore-ac:~/IFSC/redes$ cat TCPServer.py
from socket import *
serverPort = 33333
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind(('',serverPort))
#Escuta as requisicoes do TCP do cliente. Numero maximo de conexoes em fila = 1
serverSocket.listen(1)
print ('O servidor esta pronto')
while 1:
    #Quando o cliente bate a essa porta, o programa chama o metodo accept() para serverSocket,
    #que cria um novo socket no servidor, chamado connectionSocket, dedicado a esse cliente
    #especifico. Cliente e servidor, entao, completam a apresentacao, criando uma conexao TCP
    #entre o clientSocket do cliente e o connectionSocket do servidor.
    connectionSocket, addr = serverSocket.accept()
    message = connectionSocket.recv(1024)
    print (message)
    messageMaiuscula = message.upper()
    connectionSocket.send(messageMaiuscula)
    connectionSocket.close()
```

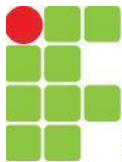
```
cadore@cadore-ac:~$ nmap -p 3333 127.0.0.1
Starting Nmap 7.80 ( https://nmap.org ) at 2022-10-10 10:17 -03
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000068s latency).

PORT      STATE SERVICE
3333/tcp  closed dec-notes

Nmap done: 1 IP address (1 host up) scanned in 0.03 seconds
cadore@cadore-ac:~$ nmap -p 33333 127.0.0.1
Starting Nmap 7.80 ( https://nmap.org ) at 2022-10-10 10:17 -03
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000069s latency).

PORT      STATE SERVICE
33333/tcp open  dgi-serv

Nmap done: 1 IP address (1 host up) scanned in 0.05 seconds
```



INSTITUTO FEDERAL
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

1) As três primeiras mensagens trocadas apresentam a camada de aplicação, sim ou não? Explique. O que elas significam?

Não apresentam, apenas mensagens até a camada de transporte. Servem para a troca de mensagens inicial entre as partes, para negociar parâmetros de sessão e iniciar a sessão.

Primeira mensagem SYN:

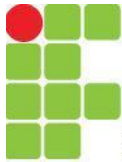
No.	Time	Source	Destination	Length	Protocol	Info
81	65.47876...	127.0.0.1	127.0.0.1	74	TCP	41290 → 33333 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2607286637 TSecr=0 WS=128
82	65.47878...	127.0.0.1	127.0.0.1	74	TCP	33333 → 41290 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=2607286638 TSecr=2607286637 WS=128
83	65.47878...	127.0.0.1	127.0.0.1	66	TCP	41290 → 33333 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2607286638 TSecr=2607286638

Frame 81: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface lo, id 0
Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 41290, Dst Port: 33333, Seq: 0, Len: 0
Source Port: 41290
Destination Port: 33333
[Stream index: 0]
[Conversation completeness: Complete, NO_DATA (39)]
[TCP Segment Len: 0]
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 235426504
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 0
Acknowledgment number (raw): 0
1010 ... = Header Length: 40 bytes (10)
Flags: 0x002 (SYN)
Window: 65495
[Calculated window size: 65495]
Checksum: 0xfe30 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
[Timestamps]

Segunda mensagem SYN, ACK:

No.	Time	Source	Destination	Length	Protocol	Info
81	65.47876...	127.0.0.1	127.0.0.1	74	TCP	41290 → 33333 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2607286637 TSecr=0 WS=128
82	65.47878...	127.0.0.1	127.0.0.1	74	TCP	33333 → 41290 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=2607286638 TSecr=2607286637 WS=128
83	65.47878...	127.0.0.1	127.0.0.1	66	TCP	41290 → 33333 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2607286638 TSecr=2607286638

Frame 82: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface lo, id 0
Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 33333, Dst Port: 41290, Seq: 0, Ack: 1, Len: 0
Source Port: 33333
Destination Port: 41290
[Stream index: 0]
[Conversation completeness: Complete, NO_DATA (39)]
[TCP Segment Len: 0]
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 2997291581
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 235426505
1010 ... = Header Length: 40 bytes (10)
Flags: 0x012 (SYN, ACK)
Window: 65483
[Calculated window size: 65483]
Checksum: 0xfe30 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
[Timestamps]
[SEQ/ACK analysis]



INSTITUTO FEDERAL
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

Terceira mensagem ACK:

No.	Time	Source	Destination	Length	Protocol	Info
81	65.47876...	127.0.0.1	127.0.0.1	74	TCP	41290 → 33333 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2607286637 TSecr=0 WS=128
82	65.47878...	127.0.0.1	127.0.0.1	74	TCP	33333 → 41290 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=2607286638 TSecr=2607286637 WS=128
83	65.47878...	127.0.0.1	127.0.0.1	66	TCP	41290 → 33333 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2607286638 TSecr=2607286638

4

- Frame 83: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface lo, id 0
- Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
- Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
- Transmission Control Protocol, Src Port: 41290, Dst Port: 33333, Seq: 1, Ack: 1, Len: 0
 - Source Port: 41290
 - Destination Port: 33333
 - [Stream index: 8]
 - [Conversation completeness: Complete, NO_DATA (39)]
 - [TCP Segment Len: 0]
 - Sequence Number: 1 (relative sequence number)
 - Sequence Number (raw): 235426505
 - [Next Sequence Number: 1 (relative sequence number)]
 - Acknowledgment Number: 1 (relative ack number)
 - Acknowledgment number (raw): 2997291582
 - 1000 = Header Length: 32 bytes (8)
 - Flags: 0x010 (ACK)
 - Window: 512
 - [Calculated window size: 65536]
 - [Window size scaling factor: 128]
 - Checksum: 0xfe28 [unverified]
 - [Checksum Status: Unverified]
 - Urgent Pointer: 0
 - Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
 - [Timestamps]
 - [SEQ/ACK analysis]

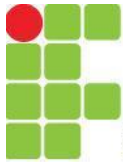
2) Em qual mensagem (número) é que a frase escrita é enviada ao servidor?

A partir da quarta mensagem, até o momento da finalização da conexão, todas as demais mensagens são encaminhadas entre servidor e cliente para troca de dados úteis.

No.	Time	Source	Destination	Length	Protocol	Info
81	65.47876...	127.0.0.1	127.0.0.1	74	TCP	41290 → 33333 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2607286637 TSecr=0 WS=128
82	65.47878...	127.0.0.1	127.0.0.1	74	TCP	33333 → 41290 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=2607286638 TSecr=2607286637 WS=128
83	65.47878...	127.0.0.1	127.0.0.1	66	TCP	41290 → 33333 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2607286638 TSecr=2607286638
84	65.47880...	127.0.0.1	127.0.0.1	66	TCP	41290 → 33333 [RST, ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2607286638 TSecr=2607286638

4

- Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
- Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
- Transmission Control Protocol, Src Port: 41290, Dst Port: 33333, Seq: 1, Ack: 1, Len: 0
 - Source Port: 41290
 - Destination Port: 33333
 - [Stream index: 8]
 - [Conversation completeness: Complete, NO_DATA (39)]
 - [TCP Segment Len: 0]
 - Sequence Number: 1 (relative sequence number)
 - Sequence Number (raw): 235426505
 - [Next Sequence Number: 1 (relative sequence number)]
 - Acknowledgment Number: 1 (relative ack number)
 - Acknowledgment number (raw): 2997291582
 - 1000 = Header Length: 32 bytes (8)
 - Flags: 0x014 (RST, ACK)
 - 000. = Reserved: Not set
 - ...0 = Nonce: Not set
 -0... = Congestion Window Reduced (CWR): Not set
 -0... = ECN-Echo: Not set
 -0... = Urgent: Not set
 -1... = Acknowledgment: Set
 -0... = Push: Not set
 -1... = Reset: Set
 -0... = Syn: Not set
 -0... = Fin: Not set
 - [TCP Flags:A.R..]
 - Window: 512
 - [Calculated window size: 65536]
 - [Window size scaling factor: 128]
 - Checksum: 0xfe28 [unverified]
 - [Checksum Status: Unverified]
 - Urgent Pointer: 0
 - Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
 - [Timestamps]



INSTITUTO FEDERAL
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

3))A mensagem seguinte (quinta) apresenta a camada de aplicação? Clique na camada TCP no Wireshark e observe o campo Flags: 0x010 (ACK). O que você acha que isso significa?

R: Significa que junto da mensagem que voltou ao cliente, foi encaminhado também um ACK como “carona” no pacote.

4) Qual o conteúdo da mensagem seguinte (sexta)? E da sétima? Explique.

R: Durante a sexta e sétima mensagem, o cliente solicitou uma página ao servidor através de HTTP e o servidor encaminha um ACK através do TCP para informar ao cliente que a solicitação foi recebida.

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

R: TCP

Quais são os números de porta e os IPs utilizados?

R: Origem 41290

Destino 33333

6) Quais foram os números de sequência utilizados em todas as mensagens?

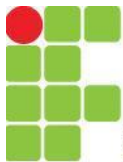
Iniciou-se em 0 (convertido pelo wireshark) e finalizou em 0, pois o outro dispositivo não transmitiu dados, apenas recebeu do dispositivo inicial.

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

Campo “Protocol” sendo que seu valor é: 6.

Quantos sockets foram abertos no servidor com um cliente "conectado"? E com dois clientes?

Com um cliente foi aberto apenas um socket, com dois clientes, foram abertos dois sockets.



INSTITUTO FEDERAL
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

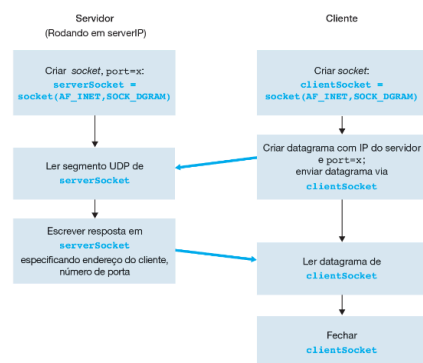
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

PARTE 2 (SOCKETS UDP):

Programação de sockets com UDP

A aplicação cliente-servidor usando UDP tem a estrutura apresentada na Figura abaixo. Utilizamos a linguagem Python por expor com clareza os principais conceitos de sockets. Quem desejar pode implementar em outras linguagens, por exemplo um modelo para programação de sockets utilizando a API Posix encontra-se [aqui](#).

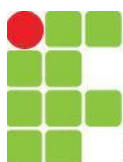


Como fica evidente na Figura acima, há dois processos cliente e servidor que podem ou não rodar em máquinas distintas e se comunicam justamente enviando mensagens via sockets, que abstrai qualquer necessidade de conhecimento das camadas subjacentes.

Verificando as conexões ativas no netstat, veja que a porta 22222 está aberta para o protocolo UDP.

```
cadore@cadore-ac:~$ netstat -eau
Conexões Internet Ativas (servidores e estabelecidas)
Proto Recv-Q Send-Q Endereço Local          Endereço Remoto          Estado      Usuário Inode
udp      0      0 localhost:domain        0.0.0.0:*                systemd-resolve 23551
udp      0      0 cadore-ac:bootpc       191.36.8.66:bootps       ESTABELECID root      3329634
udp      0      0 0.0.0.0:631            0.0.0.0:*                root         1632225
udp      0      0 224.0.0.251:mdns       0.0.0.0:*                cadore       3331328
udp      0      0 224.0.0.251:mdns       0.0.0.0:*                cadore       2352414
udp      0      0 224.0.0.251:mdns       0.0.0.0:*                cadore       95216
udp      0      0 224.0.0.251:mdns       0.0.0.0:*                cadore       95214
udp      0      0 0.0.0.0:mdns           0.0.0.0:*                avahi        26863
udp      0      0 cadore-ac:38318        52.97.26.34:https       ESTABELECID cadore    3330453
udp      0      0 0.0.0.0:22222          0.0.0.0:*                cadore       3444663
udp      0      0 0.0.0.0:35145          0.0.0.0:*                avahi        26865
udp6     0      0 cadore-ac:58104        2800:3f0:4001:828:https ESTABELECID cadore    3442536
udp6     0      0 [::]:mdns              [::]:*                   avahi        26864
udp6     0      0 [::]:42805             [::]:*                   avahi        26866
udp6     0      0 cadore-ac:55418        2800:3f0:4001:829:https ESTABELECID cadore    3446903
udp6     0      0 cadore-ac:47241        2800:3f0:4001:82e:https ESTABELECID cadore    3403292
udp6     0      0 cadore-ac:47281        2800:3f0:4001:82e:https ESTABELECID cadore    3409182
udp6     0      0 cadore-ac:60098        2800:3f0:4001:832:https ESTABELECID cadore    3444694
udp6     0      0 cadore-ac:48772        2800:3f0:4001:80f:https ESTABELECID cadore    3447886
udp6     0      0 cadore-ac:40662        2800:3f0:4001:81f:https ESTABELECID cadore    3442652
```

Feita uma captura no wireshark e testado o programa para socket UDP.



INSTITUTO FEDERAL
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

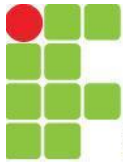
CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

Note que não há pacotes de estabelecimento de conexão antes da passagem de dados, os três primeiros pacotes já são responsáveis por encaminhar dados UDP.

No.	Time	Source	Destination	Length	Protocol	Info
12...	2177.289...	127.0.0.1	127.0.0.1	42	UDP	54633 → 22222 Len=0
12...	2177.289...	127.0.0.1	127.0.0.1	42	UDP	22222 → 54633 Len=0
12...	2177.289...	127.0.0.1	127.0.0.1	70	ICMP	Destination unreachable (Port unreachable)
Frame 1226: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface lo, id 0						
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)						
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1						
0100 = Version: 4						
.... 0101 = Header Length: 20 bytes (5)						
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)						
Total Length: 28						
Identification: 0x3212 (12818)						
Flags: 0x00						
...0 0000 0000 0000 = Fragment Offset: 0						
Time to Live: 39						
Protocol: UDP (17)						
Header Checksum: 0x63bd [validation disabled]						
[Header checksum status: Unverified]						
Source Address: 127.0.0.1						
Destination Address: 127.0.0.1						
User Datagram Protocol, Src Port: 54633, Dst Port: 22222						
Source Port: 54633						
Destination Port: 22222						
Length: 8						
Checksum: 0xd5a3 [unverified]						
[Checksum Status: Unverified]						
[Stream index: 304]						

O Segundo pacote responde ao solicitante com dados de retorno, mas da mesma maneira como visto na mensagem anterior, não houve estabelecimento de conexão.

No.	Time	Source	Destination	Length	Protocol	Info
12...	2177.289...	127.0.0.1	127.0.0.1	42	UDP	54633 → 22222 Len=0
12...	2177.289...	127.0.0.1	127.0.0.1	42	UDP	22222 → 54633 Len=0
12...	2177.289...	127.0.0.1	127.0.0.1	70	ICMP	Destination unreachable (Port unreachable)
Frame 1227: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface lo, id 0						
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)						
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1						
0100 = Version: 4						
.... 0101 = Header Length: 20 bytes (5)						
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)						
Total Length: 28						
Identification: 0x5665 (22117)						
Flags: 0x40, Don't fragment						
...0 0000 0000 0000 = Fragment Offset: 0						
Time to Live: 64						
Protocol: UDP (17)						
Header Checksum: 0xe669 [validation disabled]						
[Header checksum status: Unverified]						
Source Address: 127.0.0.1						
Destination Address: 127.0.0.1						
User Datagram Protocol, Src Port: 22222, Dst Port: 54633						
Source Port: 22222						
Destination Port: 54633						
Length: 8						
Checksum: 0xfe1b [unverified]						
[Checksum Status: Unverified]						
[Stream index: 304]						



INSTITUTO FEDERAL
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

PERGUNTAS baseadas na captura:

- 1) Em algum momento foi identificado algum procedimento para estabelecimento de conexão?

Para o protocolo UDP não.

- 2) Em algum campo do UDP existe numeração de mensagens?

Não, o protocolo apenas encaminha as mensagens, sem numeração.

- 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.

Numero 17.

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

É formado pelo campo “checksum”, ele possui 4 caracteres hexadecimais, portanto 16bits.

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

Sim, pois ele não foi cifrado antes do encaminhamento da mensagem.

Comparativo entre TCP e UDP:

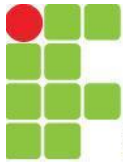
- 1) Se a mensagem digitada for teste, do cliente para o servidor deve aparecer o campo Data: 7465737465 e a resposta do servidor deve aparecer Data: 5445535445. O que significa isso? Dica, olhe na internet o código ASCII.

Significa que o conteúdo da mensagem do pacote é 5445535445.

- 2) Qual foi a sequência numérica do campo Data em seu teste? Qual o significado?

Significa que o pacote é a sequência X da conexão.

- 3) Qual é o protocolo da camada de transporte nessa troca de mensagens?



INSTITUTO FEDERAL
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

TCP devido a cifra utilizada, sequência de conexão, etc.

4) Quantos sockets foram abertos no servidor com um cliente "conectado"? E com dois clientes?

Apenas 1, no caso de ter 2 clientes, dois sockets seriam abertos.