

T1 Laboratório de Redes de Computadores

Erick Machado, Eduardo Enes

1. Execute o Wireshark para monitorar o tráfego UDP gerado pelo programa.

Identifique os pacotes UDP que estão sendo enviados para cada um dos servidores.

Quais portas de origem e destino estão sendo utilizadas pelos pacotes?

R: Exemplo capturado do registro de um novo usuário no servidor. Porta origem (usuario): 34725, porta destino (servidor): 8686.

35.927574108	192.168.123.221	192.168.123.13	UDP	52	34725 → 8686	Len=10
Frame 5: 52 bytes on wire (416 bits), 52 bytes captured (416 bits) on interface wlan0, id 0						
Ethernet II, Src: ChongqingFug_d0:86:29 (c8:94:02:d0:86:29), Dst: Intel_16:6b:47 (dc:21:48:16:6b:47)						
Internet Protocol Version 4, Src: 192.168.123.221, Dst: 192.168.123.13						
User Datagram Protocol, Src Port: 34725, Dst Port: 8686						
Source Port: 34725						
Destination Port: 8686						
Length: 18						
Checksum: 0x7327 [unverified]						
[Checksum Status: Unverified]						
[Stream index: 0]						
[Stream Packet Number: 1]						
[Timestamps]						
UDP payload (10 bytes)						
Data (10 bytes)						

0000	dc 21 48 16 6b 47 c8 94	02 d0 86 29 08 00 45 00	ih kg	E
0010	00 26 c2 73 40 00 40 11	00 18 c0 a8 7b dd c0 a8	& s0 0	{ . . .
0020	7b 0d 87 a5 21 ee 00 12	73 27 2f 52 45 47 20 65	{	s'/REG e
0030	72 69 63 6b		rick	

2. Há diferença, em termos de volume de tráfego na rede, entre a aplicação com socket TCP e a aplicação com socket UDP?

R: Sim, a aplicação com socket TCP precisa realizar mais trocas de mensagens para confirmar o envio das mensagens (TCP handshake), e, caso algum pacote não tenha sido entregue ou sua entrega não tenha sido confirmada, são realizadas retransmissões de pacotes. Ao contrário do UDP, que apenas realiza apenas uma vez o envio dos pacotes.

3. Há diferença, em termos de desempenho da aplicação, entre a aplicação com socket TCP e a aplicação com socket UDP?

R: Sim, pelo fato da aplicação com socket TCP possuir mais etapas de confirmação no envio de uma mensagem que o UDP, então ela será mais lenta.

4. Compare a transmissão de um arquivo de 1200 bytes usando a socket TCP e socket UDP

Para ambas aplicações TCP e UDP, foi criado um protocolo para envio de arquivos de texto em que a primeira mensagem enviada indica metadados do arquivo enviado e a última mensagem é um aviso ao destinatário que ele recebeu um arquivo, os demais pacotes são referentes aos bytes do arquivo.

Trasmissão UDP:

7	24.598042950	192.168.123.13	192.168.123.221	UDP	72	8686 → 59077	Len=30
8	24.632621281	192.168.123.13	192.168.123.221	UDP	442	8686 → 59077	Len=400
9	24.633072868	192.168.123.13	192.168.123.221	UDP	442	8686 → 59077	Len=400
10	24.633352858	192.168.123.13	192.168.123.221	UDP	442	8686 → 59077	Len=400
11	24.633412012	192.168.123.13	192.168.123.221	UDP	84	8686 → 59077	Len=42

Transmissão TCP:

Source	Destination	Protocol	Length	Info
127.0.0.1	127.0.0.1	TCP	92	37890 → 1500 [PSH, ACK] Seq=1 Ack=1 Win=512 Len=26 TSval=58283
127.0.0.1	127.0.0.1	TCP	66	1500 → 37890 [ACK] Seq=1 Ack=27 Win=512 Len=0 TSval=58283
127.0.0.1	127.0.0.1	TCP	1266	37890 → 1500 [PSH, ACK] Seq=27 Ack=1 Win=512 Len=1200 TSval=58283
127.0.0.1	127.0.0.1	TCP	66	1500 → 37890 [ACK] Seq=1 Ack=1227 Win=512 Len=0 TSval=58283
127.0.0.1	127.0.0.1	TCP	87	1500 → 37888 [PSH, ACK] Seq=1 Ack=1 Win=512 Len=21 TSval=58283
127.0.0.1	127.0.0.1	TCP	66	37888 → 1500 [ACK] Seq=1 Ack=22 Win=509 Len=0 TSval=58283
127.0.0.1	127.0.0.1	TCP	466	1500 → 37888 [PSH, ACK] Seq=22 Ack=1 Win=512 Len=400 TSval=58283
127.0.0.1	127.0.0.1	TCP	66	37888 → 1500 [ACK] Seq=1 Ack=422 Win=507 Len=0 TSval=58283
127.0.0.1	127.0.0.1	TCP	466	1500 → 37888 [PSH, ACK] Seq=422 Ack=1 Win=512 Len=400 TSval=58283
127.0.0.1	127.0.0.1	TCP	66	37888 → 1500 [ACK] Seq=1 Ack=822 Win=507 Len=0 TSval=58283
127.0.0.1	127.0.0.1	TCP	466	1500 → 37888 [PSH, ACK] Seq=822 Ack=1 Win=512 Len=400 TSval=58283
127.0.0.1	127.0.0.1	TCP	66	37888 → 1500 [ACK] Seq=1 Ack=1222 Win=507 Len=0 TSval=58283
127.0.0.1	127.0.0.1	TCP	108	1500 → 37888 [PSH, ACK] Seq=1222 Ack=1 Win=512 Len=42 TSval=58283
127.0.0.1	127.0.0.1	TCP	66	37888 → 1500 [ACK] Seq=1 Ack=1264 Win=509 Len=0 TSval=58283

Percebemos que o protocolo TCP gera mais tráfego de rede em comparação com o UDP pois é necessário confirmar o envio de cada pacote.

5. Compare a transmissão de um arquivo de 2000 bytes usando a socket TCP e socket UDP

Transmissão UDP:

2288	138.479098885	192.168.123.13	192.168.123.221	UDP	72	8686 → 59077	Len=30
2289	138.479781154	192.168.123.13	192.168.123.221	UDP	442	8686 → 59077	Len=400
2290	138.480518247	192.168.123.13	192.168.123.221	UDP	442	8686 → 59077	Len=400
2291	138.481106650	192.168.123.13	192.168.123.221	UDP	442	8686 → 59077	Len=400
2292	138.481749529	192.168.123.13	192.168.123.221	UDP	442	8686 → 59077	Len=400
2293	138.482172550	192.168.123.13	192.168.123.221	UDP	442	8686 → 59077	Len=400
2294	138.482246651	192.168.123.13	192.168.123.221	UDP	84	8686 → 59077	Len=42

Transmissão TCP:

Source	Destination	Protocol	Length	Info
127.0.0.1	127.0.0.1	TCP	92	37890 → 1500 [PSH, ACK] Seq=1 Ack=1 Win=512 Len=26 TSval=5830996
127.0.0.1	127.0.0.1	TCP	66	1500 → 37890 [ACK] Seq=1 Ack=27 Win=512 Len=0 TSval=583099659
127.0.0.1	127.0.0.1	TCP	2066	37890 → 1500 [PSH, ACK] Seq=27 Ack=1 Win=512 Len=2000 TSval=583100911
127.0.0.1	127.0.0.1	TCP	66	1500 → 37890 [ACK] Seq=1 Ack=2027 Win=512 Len=0 TSval=583100660
127.0.0.1	127.0.0.1	TCP	87	1500 → 37888 [PSH, ACK] Seq=1 Ack=1 Win=512 Len=21 TSval=583100660
127.0.0.1	127.0.0.1	TCP	66	37888 → 1500 [ACK] Seq=1 Ack=22 Win=509 Len=0 TSval=583100660
127.0.0.1	127.0.0.1	TCP	466	1500 → 37888 [PSH, ACK] Seq=22 Ack=1 Win=512 Len=400 TSval=583100911
127.0.0.1	127.0.0.1	TCP	66	37888 → 1500 [ACK] Seq=1 Ack=422 Win=507 Len=0 TSval=583100911
127.0.0.1	127.0.0.1	TCP	466	1500 → 37888 [PSH, ACK] Seq=422 Ack=1 Win=512 Len=400 TSval=583100911
127.0.0.1	127.0.0.1	TCP	66	37888 → 1500 [ACK] Seq=1 Ack=822 Win=507 Len=0 TSval=583101011
127.0.0.1	127.0.0.1	TCP	466	1500 → 37888 [PSH, ACK] Seq=822 Ack=1 Win=512 Len=400 TSval=583101011
127.0.0.1	127.0.0.1	TCP	66	37888 → 1500 [ACK] Seq=1 Ack=1222 Win=507 Len=0 TSval=583101112
127.0.0.1	127.0.0.1	TCP	466	1500 → 37888 [PSH, ACK] Seq=1222 Ack=1 Win=512 Len=400 TSval=583101112
127.0.0.1	127.0.0.1	TCP	66	37888 → 1500 [ACK] Seq=1 Ack=1622 Win=507 Len=0 TSval=583101212
127.0.0.1	127.0.0.1	TCP	466	1500 → 37888 [PSH, ACK] Seq=1622 Ack=1 Win=512 Len=400 TSval=583101312
127.0.0.1	127.0.0.1	TCP	66	37888 → 1500 [ACK] Seq=1 Ack=2022 Win=507 Len=0 TSval=583101312
127.0.0.1	127.0.0.1	TCP	108	1500 → 37888 [PSH, ACK] Seq=2022 Ack=1 Win=512 Len=42 TSval=583101413
127.0.0.1	127.0.0.1	TCP	66	37888 → 1500 [ACK] Seq=1 Ack=2064 Win=509 Len=0 TSval=583101413

Assim como identificado no envio do arquivo de 1200 bytes, o protocolo TCP gera mais tráfego de rede em comparação com o UDP pois ele precisa controlar o recebimento dos pacotes.

6. Configure a interface de rede da máquina para incluir perda de pacotes.

a. Qual a diferença, em termos de tráfego na rede, entre o socket TCP e UDP?

Houve alguma retransmissão usando TCP?

TCP Com packet loss de 70%:

Realiza retransmissão de pacotes ao receber ACKs duplicados, identificando que algum pacote se perdeu no caminho ao cliente. A captura abaixo demonstra a troca de mensagens entre clientes.

1997	314.248739923	34.107.243.93	10.32.143.23	TCP	66	443 → 44914 [ACK] Seq=3793 Ack=1074 Win=65536 Len=0 TSval=2596280998 TSecr=1390815888
1998	314.248977189	34.107.243.93	10.32.143.23	TCP	66	443 → 44914 [FIN, ACK] Seq=3793 Ack=1074 Win=65536 Len=0 TSval=2596280998 TSecr=1390815888
2000	314.471642809	34.107.243.93	10.32.143.23	TCP	66	[TCP Retransmission] 443 → 44914 [FIN, ACK] Seq=3793 Ack=1074 Win=65536 Len=0 TSval=2596281221 TSecr=...
2001	314.695626152	34.107.243.93	10.32.143.23	TCP	66	[TCP Retransmission] 443 → 44914 [FIN, ACK] Seq=3793 Ack=1074 Win=65536 Len=0 TSval=2596281445 TSecr=...
2003	315.135692577	34.107.243.93	10.32.143.23	TCP	66	[TCP Retransmission] 443 → 44914 [FIN, ACK] Seq=3793 Ack=1074 Win=65536 Len=0 TSval=2596281885 TSecr=...
2004	315.252582675	10.32.143.23	34.160.144.191	TCP	66	[TCP Previous segment not captured] 57874 → 443 [FIN, ACK] Seq=1204 Ack=9458 Win=64128 Len=0 TSval=76...
2005	315.268784589	34.160.144.191	10.32.143.23	TCP	78	[TCP Dup ACK 1879#1] 443 → 57874 [ACK] Seq=9458 Ack=1127 Win=69888 Len=0 TSval=3074156937 TSecr=76770...
2007	315.492589891	10.32.143.23	34.160.144.191	TCP	143	[TCP Retransmission] 57874 → 443 [PSH, ACK] Seq=1127 Ack=9458 Win=64128 Len=77 TSval=767755105 TSecr=...
2008	315.508831981	34.160.144.191	10.32.143.23	TCP	66	443 → 57874 [ACK] Seq=9458 Ack=1205 Win=69888 Len=0 TSval=3074157177 TSecr=767755105
2009	315.599064077	34.160.144.191	10.32.143.23	TCP	66	443 → 57874 [FIN, ACK] Seq=9458 Ack=1205 Win=69888 Len=0 TSval=3074157177 TSecr=767755105
2010	315.727495343	34.160.144.191	10.32.143.23	TCP	66	[TCP Retransmission] 443 → 57874 [FIN, ACK] Seq=9458 Ack=1205 Win=69888 Len=0 TSval=3074157396 TSecr=...
2011	315.727539027	10.32.143.23	34.160.144.191	TCP	66	57874 → 443 [ACK] Seq=1205 Ack=9459 Win=64128 Len=0 TSval=167755339 TSecr=3074157177
2012	316.007656254	34.107.243.93	10.32.143.23	TCP	66	[TCP Retransmission] 443 → 44914 [FIN, ACK] Seq=3793 Ack=1074 Win=65536 Len=0 TSval=2596282757 TSecr=...
2013	316.206858195	10.32.143.23	34.149.100.209	TLSv1.2	112	Application Data
2014	316.227045827	34.149.100.209	10.32.143.23	TCP	66	443 → 53654 [ACK] Seq=285996 Ack=2167 Win=72192 Len=0 TSval=2149502177 TSecr=2951442102
2015	316.268580576	10.32.143.23	34.149.100.209	TCP	66	[TCP Previous segment not captured] 53654 → 443 [FIN, ACK] Seq=2198 Ack=285996 Win=487040 Len=0 TSval=...
2016	316.283780534	34.149.100.209	10.32.143.23	TCP	78	[TCP Dup ACK 2014#1] 443 → 53654 [ACK] Seq=285996 Ack=2167 Win=72192 Len=0 TSval=2149502234 TSecr=295...
2024	317.806023191	34.107.243.93	10.32.143.23	TCP	66	[TCP Retransmission] 443 → 44914 [FIN, ACK] Seq=3793 Ack=1074 Win=65536 Len=0 TSval=2596284549 TSecr=...
2025	317.800062944	10.32.143.23	34.107.243.93	TCP	66	44914 → 443 [ACK] Seq=1074 Ack=3794 Win=64128 Len=0 TSval=1390819455 TSecr=2596280998
2027	318.243506574	10.32.143.23	201.54.156.25	TCP	66	[TCP Keep-Alive] 48896 → 443 [ACK] Seq=517 Ack=1449 Win=64128 Len=0 TSval=3390827800 TSecr=344054065
2028	318.849236682	201.54.156.25	10.32.143.23	TCP	66	[TCP Keep-Alive ACK] 443 → 48896 [ACK] Seq=1449 Ack=518 Win=30208 Len=0 TSval=344064068 TSecr=33908337...
2054	330.604585565	10.32.143.23	34.149.100.209	TCP	97	[TCP Retransmission] 53654 → 443 [PSH, ACK] Seq=2167 Ack=285996 Win=487040 Len=31 TSval=2951456500 TS...

UDP Com packet loss de 50%:

Para demonstrar a perda de pacotes, foi enviado o arquivo de 1200 bytes para um cliente conectado.

Em uma transmissão de 5 pacotes (conforme a captura da questão 4), o cliente apenas recebeu 2 pacotes, gerando um arquivo de texto incompleto.

No.	Time	Source	Destination	Protocol	Length	Info
165	18.916545	192.168.0.53	192.168.0.32	UDP	442	1500 → 56060 Len=400
166	18.916698	192.168.0.53	192.168.0.32	UDP	442	1500 → 56060 Len=400

```
Usuário conectado: eduardo (/192.168.0.32:56060)
ece nossa experiência e nos lembra que n?o estamos sozinhos.

A busca pelo conhecimento também é crucial. Ler livros, ouvir podcasts e participar de cursos expande nossa c
ompreensão do mundo. A curiosidade nos impulsiona a explorar novas ideias e a desafiar nossas crenças.

O autocuidado é essencial para nossa saúde física e mental. Encontrar tempo para relaxar e praticar esportes
pode fa
zer maravilhas por nosso bem-estar. Priorizar isso nos torna mais capazes de enfrentar os desafios da vida.

Por fim, a gratid?o transforma nossa perspectiva. Reconhecer as coisas boas, mesmo as pequenas, nos ajuda a e
ncontrar alegria. Agradecer pelas vitórias diárias pode mudar nossa atitude e nos permitir ver a beleza ao no
sso redor.

A vida é uma tapeçaria de experiências que nos moldam.a
```

Como podemos ver, o arquivo foi gerado pela metade, sem o pacote que continha os bytes iniciais do texto.

7) Configurar a interface de rede da máquina para incluir latência variável

a. Qual a diferença, em termos de tráfego na rede, entre o socket TCP e UDP?

Houve alguma retransmissão usando TCP?

TCP utilizando delay de 10.000 ms

Realiza retransmissão de pacotes quando o timer de recebimento de ACK estora, buscando garantir a entrega dos pacotes ao destinatário.

946	784.436683194	10.32.143.23	10.32.143.139	TCP	76 [TCP Retransmission] 1500 → 54426 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2663023059 TSecr=2663023059
946	784.436683194	10.32.143.139	10.32.143.23	TCP	62 42574 → 1500 [RST] Seq=33 Win=0 Len=0
947	784.436683432	10.32.143.139	10.32.143.23	TCP	68 [TCP Previous segment not captured] 54426 → 1500 [ACK] Seq=14 Ack=1 Win=64256 Len=0 TSval=3973665789 TSecr=2663023059
948	784.436759062	10.32.143.139	10.32.143.23	TCP	62 42574 → 1500 [RST] Seq=33 Win=0 Len=0
949	784.436759205	10.32.143.139	10.32.143.23	TCP	68 [TCP Dup ACK 947#1] 54426 → 1500 [ACK] Seq=14 Ack=1 Win=64256 Len=0 TSval=3973667830 TSecr=2663023059
955	786.459420512	10.32.143.23	10.32.143.139	TCP	76 [TCP Retransmission] 1500 → 54426 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2663023059 TSecr=2663023059
956	786.459667973	10.32.143.139	10.32.143.23	TCP	68 [TCP Dup ACK 947#2] 54426 → 1500 [ACK] Seq=14 Ack=1 Win=64256 Len=0 TSval=3973669853 TSecr=2663023059
964	786.683417865	10.32.143.23	10.32.143.139	TCP	76 [TCP Retransmission] 1500 → 54426 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2663023059 TSecr=2663023059
965	786.683455710	10.32.143.139	10.32.143.23	TCP	68 [TCP Dup ACK 947#3] 54426 → 1500 [ACK] Seq=14 Ack=1 Win=64256 Len=0 TSval=3973674077 TSecr=2663023059
969	791.707414737	10.32.143.23	54.161.152.147	TCP	68 [TCP Keep-Alive] 49972 → 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=2397315196 TSecr=2646480697
970	791.857521148	54.161.152.147	10.32.143.23	TCP	68 [TCP Keep-Alive ACK] 443 → 49972 [ACK] Seq=1 Ack=2 Win=484 Len=0 TSval=2646521195 TSecr=2396731782
971	791.963421730	10.32.143.23	18.194.180.142	TCP	68 [TCP Keep-Alive] 51606 → 80 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=3788167905 TSecr=2485069314
972	792.207651023	18.194.180.142	10.32.143.23	TCP	68 [TCP Keep-Alive ACK] 80 → 51606 [ACK] Seq=1 Ack=2 Win=479 Len=0 TSval=2485109778 TSecr=3787836322
977	798.889932456	10.32.143.23	34.117.188.166	TLSv1.3	107 Application Data
983	802.004446970	10.32.143.23	10.32.143.23	TCP	62 42574 → 1500 [RST] Seq=33 Win=0 Len=0
994	806.892068126	10.32.143.23	34.117.188.166	TLSv1.3	92 Application Data
995	806.892292594	10.32.143.23	34.117.188.166	TLSv1.3	92 Application Data
999	808.905624369	34.117.188.166	10.32.143.23	TCP	68 443 → 46120 [ACK] Seq=5907 Ack=1221 Win=67072 Len=0 TSval=1089845131 TSecr=286534777
1000	808.905624825	34.117.188.166	10.32.143.23	TLSv1.3	107 Application Data
1001	808.905624905	34.117.188.166	10.32.143.23	TCP	107 [TCP Retransmission] 443 → 46120 [PSH, ACK] Seq=5907 Ack=1221 Win=67072 Len=39 TSval=1089847411 TSecr=286534777
1002	808.905624975	34.117.188.166	10.32.143.23	TCP	107 [TCP Retransmission] 443 → 46120 [PSH, ACK] Seq=5907 Ack=1221 Win=67072 Len=39 TSval=1089850291 TSecr=286534777
1003	808.905625043	54.161.152.147	10.32.143.23	TCP	68 [TCP Keep-Alive] 443 → 49972 [ACK] Seq=0 Ack=2 Win=484 Len=0 TSval=2646533747 TSecr=2396731782
1004	808.905625113	34.117.188.166	10.32.143.23	TCP	68 443 → 46120 [ACK] Seq=5946 Ack=1285 Win=67072 Len=0 TSval=1089853134 TSecr=286542779
1007	809.952683758	34.117.188.166	10.32.143.23	TCP	107 [TCP Retransmission] 443 → 46120 [PSH, ACK] Seq=5907 Ack=1285 Win=67072 Len=39 TSval=1089856179 TSecr=286542779
1012	812.084528987	10.32.143.23	10.32.143.139	TCP	68 1500 → 54426 [ACK] Seq=1 Ack=14 Win=65152 Len=0 TSval=2663053754 TSecr=3973685478
1013	812.084899126	10.32.143.23	10.32.143.139	TCP	97 1500 → 54426 [PSH, ACK] Seq=1 Ack=14 Win=65152 Len=29 TSval=2663053754 TSecr=3973685478
1014	812.085092010	10.32.143.139	10.32.143.23	TCP	68 54426 → 1500 [ACK] Seq=14 Ack=39 Win=64256 Len=0 TSval=3973695478 TSecr=2663053754
1015	813.892437841	10.32.143.23	142.250.219.142	TLSv1.3	107 Application Data
1016	813.907735333	142.250.219.142	10.32.143.23	TCP	68 443 → 54520 [ACK] Seq=1535 Ack=2472 Win=75264 Len=0 TSval=1550088425 TSecr=301187391
1017	813.907735659	142.250.219.142	10.32.143.23	TLSv1.3	107 Application Data

A aplicação utilizando sockets UDP não realiza a retransmissão. O destinatário recebe os pacotes 10s depois do envio do remetente, pois em nossa implementação o programa não desconecta até receber os pacotes do remetente.