

# Lab 2 : Protocolos de Transporte – UDP/TCP

Rodrigo Seiji Piubeli Hirao (186837)

16 de dezembro de 2021

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Metodologia</b>	<b>2</b>
2.1	Requisição TCP/UDP	2
2.2	Requisição TCP com tráfego ao fundo	2
<b>3</b>	<b>Resultados</b>	<b>3</b>
3.1	Requisição TCP/UDP	3
3.1.1	Primeira requisição UDP (1.2kB)	3
3.1.2	Segunda requisição UDP (1.2kB)	3
3.1.3	Requisição TCP (1.2kB)	4
3.1.4	Requisição UDP (4kB)	4
3.1.5	Requisição TCP (4kB)	5
3.1.6	TCP em tempo real	5
3.2	Requisição TCP com tráfego ao fundo	6
<b>4</b>	<b>Discussão</b>	<b>7</b>
4.1	Requisição TCP/UDP	7
4.2	Requisição TCP com tráfego ao fundo	8
<b>5</b>	<b>Conclusão</b>	<b>8</b>
<b>6</b>	<b>Feedback</b>	<b>8</b>

# 1 Introdução

Nesse laboratório será estudado o comportamento dos protocolos TCP e UDP, e vendo como seus comportamentos se diferem com o tamanho do pacote enviado e com tráfegos de fundo presentes.

## 2 Metodologia

### 2.1 Requisição TCP/UDP

Foi usado a topologia da figura 1 para analisar os protocolos UDP e TCP

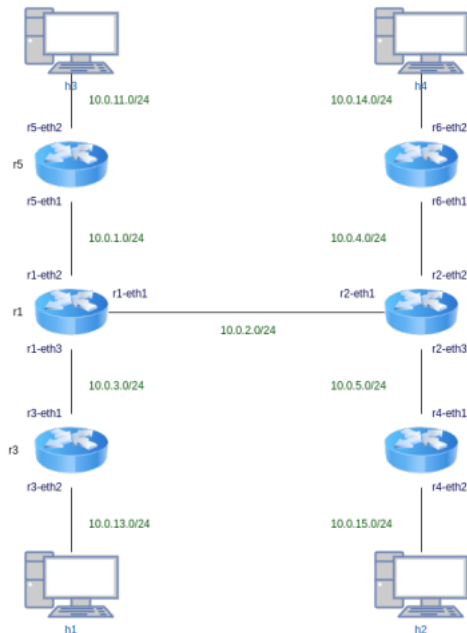


Figura 1: Topologia Experimental UDP/TCP

Então foram transmitidos pacotes de 1200 bytes de h1 até h4 por UDP e TCP, para fim de estudos do comportamento pelo Wireshark.

Usando os comandos do netcat para UDP:

```
h1> nc -l 4444 -u > data.txt
```

```
h4> nc -w 3 10.0.13.3 4444 -u < data.txt
```

E para o TCP:

```
h1> nc -l 4444 > data.txt
```

```
h4> nc -w 3 10.0.13.3 4444 < data.txt
```

O processo foi repetido então para 4000 bytes.

Então foi analisado o TCP em tempo real, abrindo uma conexão, enviando "hello"(6 bytes) e fechando a conexão.

### 2.2 Requisição TCP com tráfego ao fundo

Por fim foi criado um tráfego de fundo entre h2 e h3 usando os comandos do D-ITG:

```
h2> ./ITGRecv
```

```
h3> ./ITGSend -a 10.0.15.3 -C 10000000000000 -c 40000 -t 60000 -T UDP
```

E enviados pacotes de 100MB de h4 para h1 por aproximadamente 30 segundos, enquanto analisado o comportamento de h1 pelo Wireshark:

```
h4> for i in {1..10} ; do cat data3.txt ; done | nc 10.0.13.3 4444
```

## 3 Resultados

### 3.1 Requisição TCP/UDP

#### 3.1.1 Primeira requisição UDP (1.2kB)

```
Frame 5: 1242 bytes on wire (9936 bits), 1242 bytes captured (9936 bits) on interface h1-eth1, id 0
Ethernet II, Src: ae:7f:51:3f:96:7e (ae:7f:51:3f:96:7e), Dst: ca:19:a0:99:c4:50 (ca:19:a0:99:c4:50)
  Destination: ca:19:a0:99:c4:50 (ca:19:a0:99:c4:50)
  Source: ae:7f:51:3f:96:7e (ae:7f:51:3f:96:7e)
  Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.0.14.3, Dst: 10.0.13.3
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 1228
  Identification: 0x5790 (22416)
  Flags: 0x4000, Don't fragment
  Fragment offset: 0
  Time to live: 60
  Protocol: UDP (17)
  Header checksum: 0xb38b [validation disabled]
  [Header checksum status: Unverified]
  Source: 10.0.14.3
  Destination: 10.0.13.3
User Datagram Protocol, Src Port: 58439, Dst Port: 4444
  Source Port: 58439
  Destination Port: 4444
  Length: 1208
  Checksum: 0x33cf [unverified]
  [Checksum Status: Unverified]
  [Stream index: 0]
  [Timestamps]
Data (1200 bytes)
  Data: 0000000000000000000000000000000000000000000000000000000000000000...
  [Length: 1200]
```

#### 3.1.2 Segunda requisição UDP (1.2kB)

Segunda requisição feita com diferenças destacadas em negrito.

```
Frame 14: 1242 bytes on wire (9936 bits), 1242 bytes captured (9936 bits) on interface h1-eth1
Ethernet II, Src: ae:7f:51:3f:96:7e (ae:7f:51:3f:96:7e), Dst: ca:19:a0:99:c4:50 (ca:19:a0:99:c4:50)
  Destination: ca:19:a0:99:c4:50 (ca:19:a0:99:c4:50)
  Source: ae:7f:51:3f:96:7e (ae:7f:51:3f:96:7e)
  Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.0.14.3, Dst: 10.0.13.3
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 1228
  Identification: 0x6397 (25495)
```

```

Flags: 0x4000, Don't fragment
Fragment offset: 0
Time to live: 60
Protocol: UDP (17)
Header checksum: 0xa784 [validation disabled]
[Header checksum status: Unverified]
Source: 10.0.14.3
Destination: 10.0.13.3
User Datagram Protocol, Src Port: 33578, Dst Port: 4444
Source Port: 33578
Destination Port: 4444
Length: 1208
Checksum: 0x33cf [unverified]
[Checksum Status: Unverified]
[Stream index: 1]
[Timestamps]
Data (1200 bytes)
Data: 0000000000000000000000000000000000000000000000000000000000000000...
[Length: 1200]

```

### 3.1.3 Requisição TCP (1.2kB)

Fonte	Destino	Tamanho	Tipo	Seq	Ack	Len
10.0.14.3	10.0.13.3	74	[SYN]	0	0	0
10.0.13.3	10.0.14.3	74	[SYN, ACK]	0	1	0
10.0.14.3	10.0.13.3	66	[ACK]	1	1	0
10.0.14.3	10.0.13.3	1266	[PSH, ACK]	1	1	1200
10.0.13.3	10.0.14.3	66	[ACK]	1	1201	0
10.0.14.3	10.0.13.3	66	[FIN, ACK]	1201	1	0
10.0.13.3	10.0.14.3	66	[FIN, ACK]	1	1202	0
10.0.14.3	10.0.13.3	66	[ACK]	1202	2	0

Figura 2: Requisição TCP (1.2kB)

### 3.1.4 Requisição UDP (4kB)

Requisição de 4kB feita com diferenças por conta do tamanho do pacote descartadas em negrito.

```

Frame 3: 1082 bytes on wire (8656 bits), 1082 bytes captured (8656 bits) on interface h1-e
Ethernet II, Src: ae:7f:51:3f:96:7e (ae:7f:51:3f:96:7e), Dst: ca:19:a0:99:c4:50 (ca:19:a0:
Destination: ca:19:a0:99:c4:50 (ca:19:a0:99:c4:50)
Source: ae:7f:51:3f:96:7e (ae:7f:51:3f:96:7e)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.0.14.3, Dst: 10.0.13.3
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 1068
Identification: 0x5a38 (23096)
Flags: 0x0172
Fragment offset: 2960
Time to live: 60
Protocol: UDP (17)
Header checksum: 0xf011 [validation disabled]
[Header checksum status: Unverified]
Source: 10.0.14.3
Destination: 10.0.13.3
[3 IPv4 Fragments (4008 bytes): #1(1480), #2(1480), #3(1048)]

```

```

User Datagram Protocol, Src Port: 48831, Dst Port: 4444
Source Port: 48831
Destination Port: 4444
Length: 4008
Checksum: 0xe17c [unverified]
[Checksum Status: Unverified]
[Stream index: 0]
[Timestamps]
Data (4000 bytes)
Data: 0000000000000000000000000000000000000000000000000000000000000000...
[Length: 4000]

```

### 3.1.5 Requisição TCP (4kB)

Fonte	Destino	Tamanho	Tipo	Seq	Ack	Len	MSS	Window Size
10.0.14.3	10.0.13.3	74	[SYN]	0	0	0	1460	42340
10.0.13.3	10.0.14.3	74	[SYN, ACK]	0	1	0	1460	43440
10.0.14.3	10.0.13.3	66	[ACK]	1	1	0	-	83
10.0.14.3	10.0.13.3	1514	[ACK]	1	1	1448	-	83
10.0.13.3	10.0.14.3	66	[ACK]	1	1449	0	-	83
10.0.14.3	10.0.13.3	1514	[ACK]	1449	1	1448	-	83
10.0.13.3	10.0.14.3	66	[ACK]	1	2897	0	-	82
10.0.14.3	10.0.13.3	1170	[PSH, ACK]	2897	1	1104	-	83
10.0.13.3	10.0.14.3	66	[ACK]	1	4001	0	-	83
10.0.14.3	10.0.13.3	66	[FIN, ACK]	4001	1	0	-	83
10.0.13.3	10.0.14.3	66	[FIN, ACK]	1	4002	0	-	83
10.0.14.3	10.0.13.3	66	[ACK]	4002	2	0	-	83

Figura 3: Requisição TCP (4kB)

### 3.1.6 TCP em tempo real

Fonte	Destino	Tamanho	Tipo	Seq	Ack	Len	MSS	Window Size
10.0.14.3	10.0.13.3	74	[SYN]	0	0	0	1460	42340
10.0.13.3	10.0.14.3	74	[SYN, ACK]	0	1	0	1460	43440
10.0.14.3	10.0.13.3	66	[ACK]	1	1	0	-	83
10.0.14.3	10.0.13.3	72	[PSH, ACK]	1	1	6	-	83
10.0.13.3	10.0.14.3	66	[ACK]	1	7	0	-	85
10.0.14.3	10.0.13.3	66	[FIN, ACK]	7	1	0	-	83
10.0.13.3	10.0.14.3	66	[FIN, ACK]	1	8	0	-	85
10.0.14.3	10.0.13.3	66	[ACK]	8	2	0	-	83

Figura 4: TCP analisado em tempo real

### 3.2 Requisição TCP com tráfego ao fundo

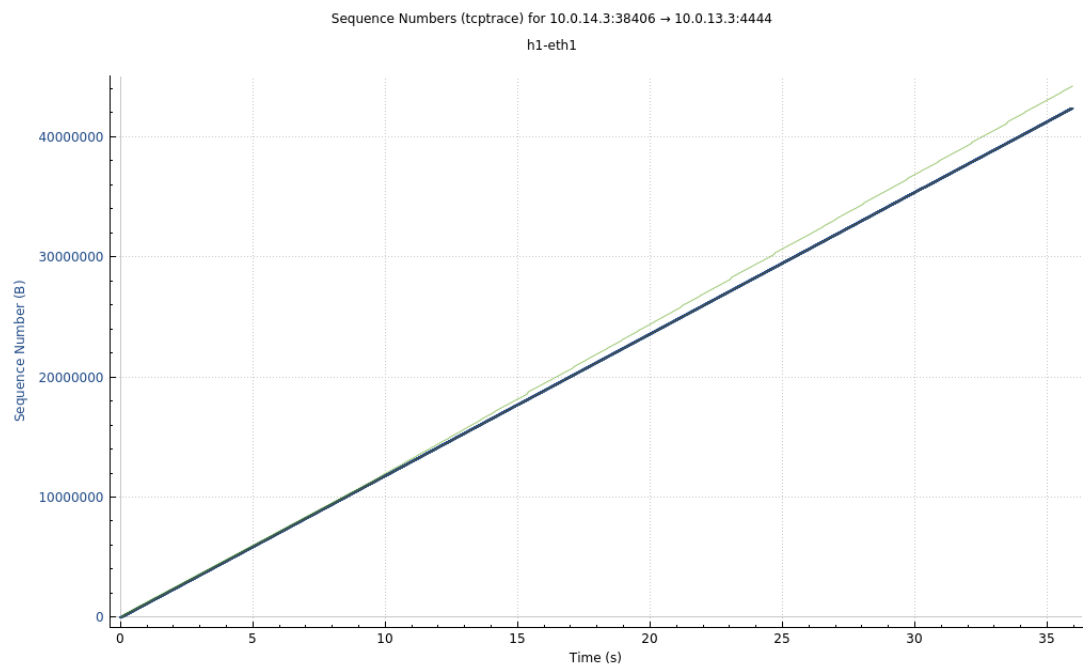


Figura 5: Time-Sequence Graph da requisição TCP com tráfego ao fundo

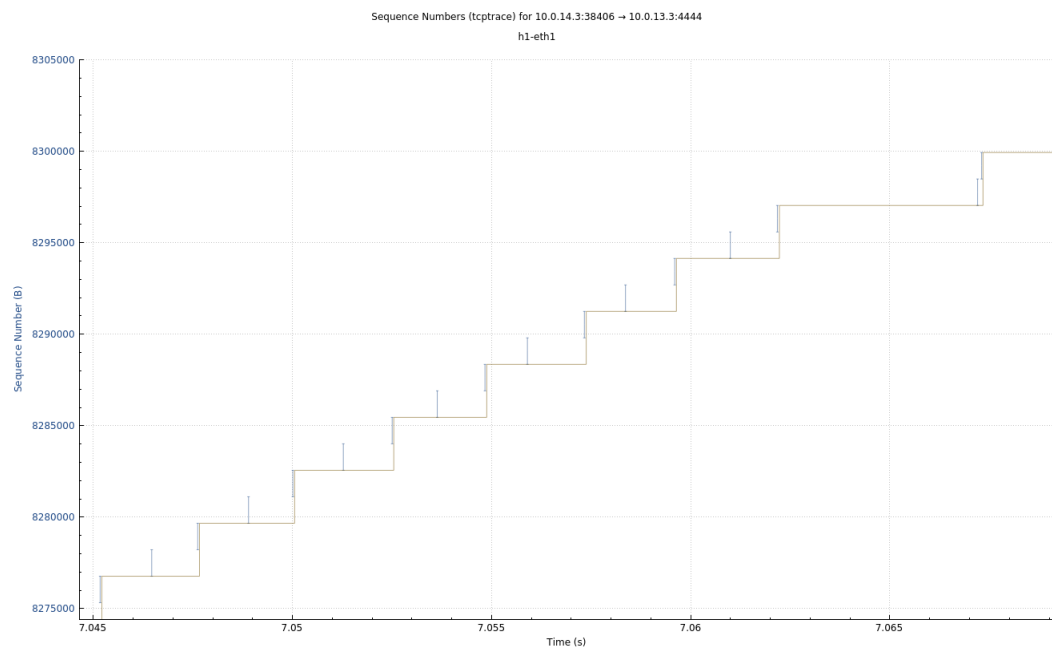


Figura 6: Time-Sequence Graph da requisição TCP com tráfego ao fundo focado em um pacote mais devagar

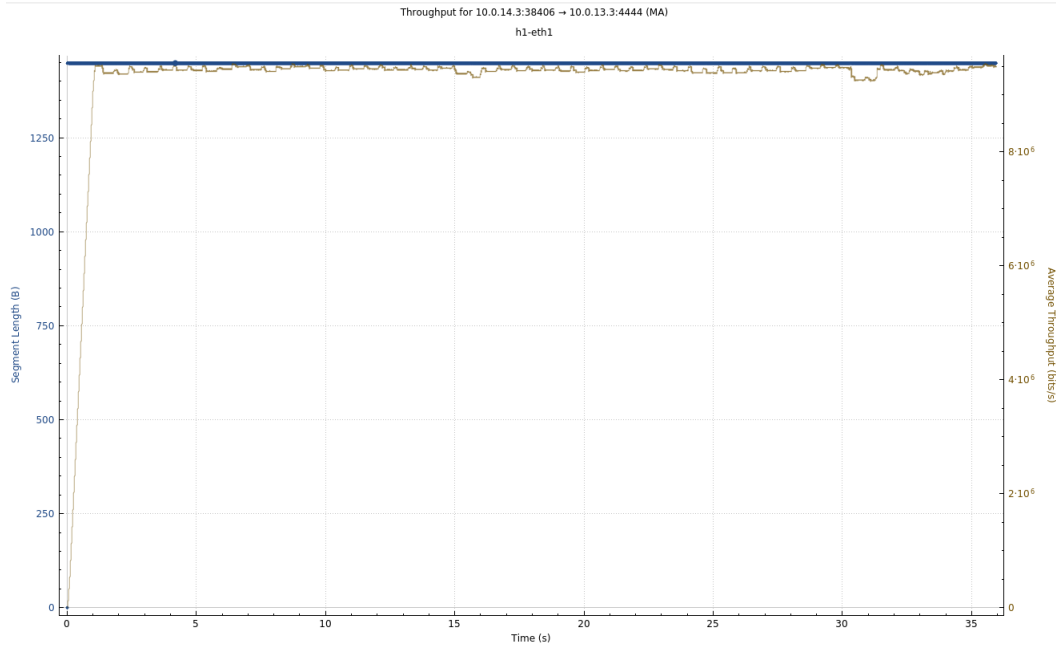


Figura 7: Throughput da requisição TCP com tráfego ao fundo

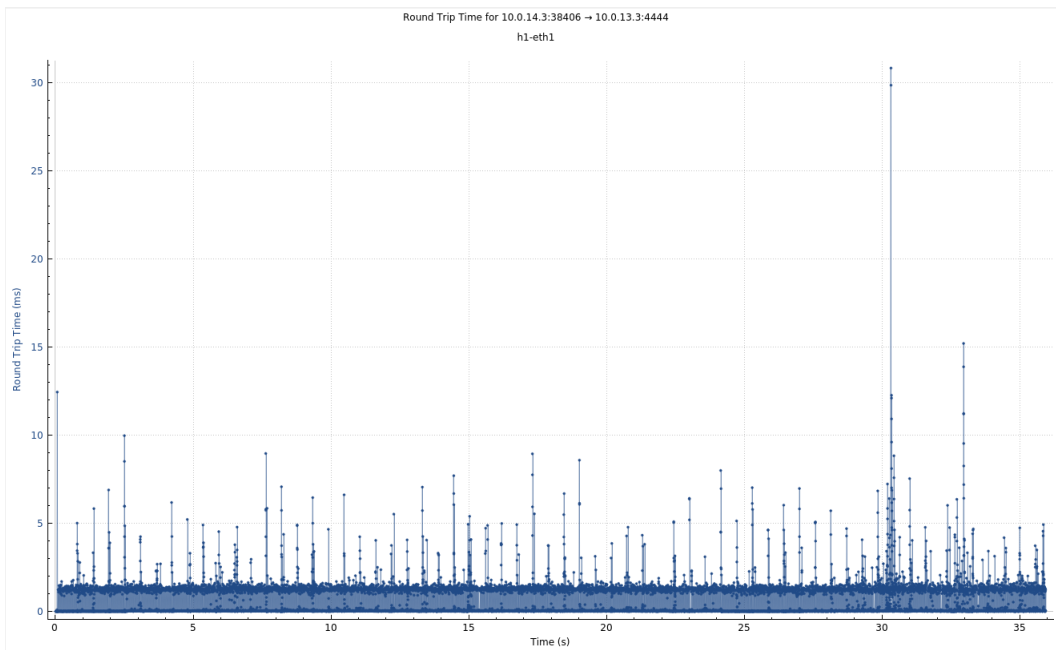


Figura 8: Round-Trip Time da requisição TCP com tráfego ao fundo

## 4 Discussão

### 4.1 Requisição TCP/UDP

Pelos logs do UDP é possível ver seu comportamento connectionless, que apenas manda um pacote sem esperar resposta, e que ao mandar o mesmo pacote de novo possui comportamento parecido, apenas com diferença de porta fonte e informações a nível de IP. Que é o mesmo comportamento com um pacote maior (4kB), que tem diferença apenas na fragmentação do IP.

Já o TCP, por ser orientado à conexão, possui um 3-way handshake anterior ao envio dos dados para abrir a conexão, seguindo o procedimento:

- Enviar requisição de sincronização (SYN) com sequência (SEQ) inicial (nos casos estudados 0)

- Receber resposta de sincronização (SYN) com o SEQ inicial e um ACK (nos casos estudados 1) esperando esse número como número de sequência
- Enviar número de sequência recebido pelo ACK e um novo ACK (nos casos estudados 1) que espera como próximo número de sequência

Deve ser notado que nas etapas de sincronização é enviado o MSS para determinar o tamanho limite do pacote e o Window Size para determinar o tamanho da janela.

Depois de aberta a conexão é enviado os dados com o último ACK recebido como SEQ o tamanho (LEN), que não deve exceder MSS e um PSH caso seja o último pacote com dados. No caso do teste com 4kB ele enviou um pacote com 1448B de dados, recebeu o ACK novo ( $ACK_{novo} = ACK_{anterior} + bytesrecebidos$ ) e repete o procedimento com o resto dos dados.

Por fim a conexão é fechada enviando um FIN de cada lado (ainda seguindo o protocolo de ACK/SEQ e a conexão é fechada.

## 4.2 Requisição TCP com tráfego ao fundo

Pela Figura 5 é possível ver o comportamento das requisições TCP de h4 até h1 com um tráfego de fundo de h2 até h3. Com inclinação de aproximadamente 1.2MB/s eu ao dar zoom (Figura 6) é possível enxergar pontos em que os pacotes são enviados mais devagar, Possivelmente por culpa do tráfego de fundo, mas mesmo assim no total não houve uma diferença significativa, como pode ser visto na Figura 5, pela diferença entre a linha azul e a verde.

O mesmo comportamento pode ser visto nas Figuras 7 e 8, que apenas alguns pacotes possuem uma velocidade mais devagar, mantendo bem próximo do ideal.

## 5 Conclusão

Foi possível ver nitidamente a diferença entre o comportamento sem conexão do UDP e o orientado à conexão do TCP, onde o primeiro deixa todo o gerenciamento para a camada anterior enquanto o TCP toma conta de sua própria fragmentação. Além do comportamento estável do TCP mesmo com outros tráfegos ocorrendo no fundo, embora ligeiramente reduzido.

## 6 Feedback

O laboratório estava muito bem escrito, meu maior problema foi com a opção de TSO que estava habilitada antes da aula parcial, mas isso foi rapidamente resolvido.