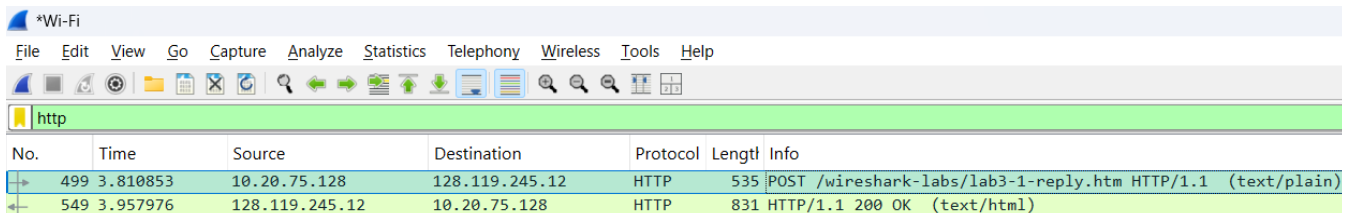


Lab Redes TCP 4/4/2025

1.



No.	Time	Source	Destination	Protocol	Length	Info
499	3.810853	10.20.75.128	128.119.245.12	HTTP	535	POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1 (text/plain)
549	3.957976	128.119.245.12	10.20.75.128	HTTP	831	HTTP/1.1 200 OK (text/html)

Source Port: 50467
Destination Port: 80
[Stream index: 7]

Source IP e Source Port = cliente

Destination IP e Port = servidor

2.

Porta do servidor: 80

3

- IP: 10.20.75.128
- Porta: 50467

4.

- IP do cliente: 10.20.75.128
- IP do servidor (gaia.cs.umass.edu): 128.119.245.12
- Porta origem: 50467
- Porta destino: 80
- Flags: [SYN]
- Sequence Number: 0 (está visível no campo Seq=0)

Resposta:

- **Número de sequência (Sequence Number): 0**
- **Identificado como SYN porque: o campo Info mostra [SYN], e nas flags do pacote está ativo apenas o bit SYN (tcp.flags.syn == 1 e tcp.flags.ack == 0).**

5

- **Source (servidor): 128.119.245.12**
- **Destination (cliente): 10.20.75.128**
- **Portas: 80 → 50467**
- **Sequence Number: 0**
- **Acknowledgment Number: 1**
- **Flags: [SYN, ACK]**
(confirmado no campo Info: Seq=0 Ack=1)

Resposta:

- **Sequence Number: 0**
- **Acknowledgment Number: 1**
Porque o cliente enviou um SYN com número de sequência 0, e o servidor responde com ACK = 0 + 1 = 1.
- **É identificado como SYN-ACK porque tem as flags SYN = 1 e ACK = 1 ativas, como mostrado na coluna "Info" ([SYN, ACK]).**

6

Número de sequência do segmento com o HTTP POST: 152578

7.

- **Os 6 primeiros segmentos TCP têm números de sequência começando em 152578, com incrementos de 1460.**
- **Todos têm comprimento de 1460 bytes.**
- **O RTT médio está entre 0.1089 e 0.1141 s.**

- **O EstimatedRTT estabiliza perto de 0.110 segundos.**

#	Seq Number	Tempo de envio (s)	Próx. ACK esperado	Tempo do ACK (estimado)	RTT (s)
1	152578	3.416831	154038	3.525749	0.1089
2	154038	3.417566	155498	3.530351	0.1128
3	155498	3.417566	156958	3.531622	0.1141
4	156958	3.417566	158418	3.531622	0.1141
5	158418	3.417566	159878	3.531622	0.1141
6	159878	3.417566	161338	3.531622	0.1141

```
EstimatedRTT = (1 - α) * EstimatedRTT_anterior + α * SampleRTT
α = 0.125
```

8

O primeiro segmento TCP (com o HTTP POST) tem 737 bytes de payload.

Os cinco seguintes têm o valor padrão de 1460 bytes, que é o Maximum Segment Size (MSS) para redes Ethernet típicas.

Segmento	Número de Sequência	Comprimento TCP
1 (POST)	152578	737 bytes
2	153315	1460 bytes
3	154775	1460 bytes
4	156235	1460 bytes
5	157695	1460 bytes
6	159155	1460 bytes

273	3.309049	10.20.75.128	128.119.245.12	TCP	66 50467 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
308	3.416798	10.20.75.128	128.119.245.12	TCP	54 50466 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0
309	3.416831	10.20.75.128	128.119.245.12	TCP	54 50467 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0
310	3.417300	10.20.75.128	128.119.245.12	TCP	791 50467 → 80 [PSH, ACK] Seq=1 Ack=1 Win=65280 Len=737 [TCP PDU reassembled in 499]
311	3.417566	10.20.75.128	128.119.245.12	TCP	1514 50467 → 80 [ACK] Seq=738 Ack=1 Win=65280 Len=1460 [TCP PDU reassembled in 499]
312	3.417566	10.20.75.128	128.119.245.12	TCP	1514 50467 → 80 [ACK] Seq=2198 Ack=1 Win=65280 Len=1460 [TCP PDU reassembled in 499]
313	3.417566	10.20.75.128	128.119.245.12	TCP	1514 50467 → 80 [ACK] Seq=3658 Ack=1 Win=65280 Len=1460 [TCP PDU reassembled in 499]
314	3.417566	10.20.75.128	128.119.245.12	TCP	1514 50467 → 80 [ACK] Seq=5118 Ack=1 Win=65280 Len=1460 [TCP PDU reassembled in 499]
315	3.417566	10.20.75.128	128.119.245.12	TCP	1514 50467 → 80 [ACK] Seq=6578 Ack=1 Win=65280 Len=1460 [TCP PDU reassembled in 499]
316	3.417566	10.20.75.128	128.119.245.12	TCP	1514 50467 → 80 [ACK] Seq=8038 Ack=1 Win=65280 Len=1460 [TCP PDU reassembled in 499]
317	3.417566	10.20.75.128	128.119.245.12	TCP	1514 50467 → 80 [ACK] Seq=9498 Ack=1 Win=65280 Len=1460 [TCP PDU reassembled in 499]
318	3.417566	10.20.75.128	128.119.245.12	TCP	1514 50467 → 80 [ACK] Seq=10958 Ack=1 Win=65280 Len=1460 [TCP PDU reassembled in 499]
319	3.417566	10.20.75.128	128.119.245.12	TCP	1514 50467 → 80 [ACK] Seq=12418 Ack=1 Win=65280 Len=1460 [TCP PDU reassembled in 499]
333	3.525749	10.20.75.128	128.119.245.12	TCP	1514 50467 → 80 [ACK] Seq=13878 Ack=1 Win=65280 Len=1460 [TCP PDU reassembled in 499]
335	3.530351	10.20.75.128	128.119.245.12	TCP	1514 50467 → 80 [ACK] Seq=15338 Ack=1 Win=65280 Len=1460 [TCP PDU reassembled in 499]
336	3.530351	10.20.75.128	128.119.245.12	TCP	1514 50467 → 80 [PSH, ACK] Seq=16798 Ack=1 Win=65280 Len=1460 [TCP PDU reassembled in 499]
340	3.531622	10.20.75.128	128.119.245.12	TCP	1514 50467 → 80 [ACK] Seq=18258 Ack=1 Win=65280 Len=1460 [TCP PDU reassembled in 499]
341	3.531622	10.20.75.128	128.119.245.12	TCP	1514 50467 → 80 [ACK] Seq=19718 Ack=1 Win=65280 Len=1460 [TCP PDU reassembled in 499]
342	3.531622	10.20.75.128	128.119.245.12	TCP	1514 50467 → 80 [ACK] Seq=21178 Ack=1 Win=65280 Len=1460 [TCP PDU reassembled in 499]
343	3.531622	10.20.75.128	128.119.245.12	TCP	1514 50467 → 80 [ACK] Seq=22638 Ack=1 Win=65280 Len=1460 [TCP PDU reassembled in 499]
344	3.531622	10.20.75.128	128.119.245.12	TCP	1514 50467 → 80 [ACK] Seq=24098 Ack=1 Win=65280 Len=1460 [TCP PDU reassembled in 499]
345	3.531622	10.20.75.128	128.119.245.12	TCP	1514 50467 → 80 [ACK] Seq=25558 Ack=1 Win=65280 Len=1460 [TCP PDU reassembled in 499]

9

O espaço mínimo de buffer anunciado foi: 65280 bytes.

A janela nunca foi reduzida para zero, nem para um valor muito pequeno.

Conclusão: O emissor nunca foi travado (throttled) por falta de espaço no buffer do recetor.

No.	Time	Source	Destination	Protocol	Length	Info
99	1.167283	142.250.184.164	10.20.75.128	TCP	6	Packet length (bytes) [] Seq=1 Ack=2 Win=1027 Len=0 SLE=1 SRE=2
206	2.754894	10.20.75.128	142.250.184.10	TCP	55	50300 → 443 [ACK] Seq=1 Ack=1 Win=252 Len=1
210	2.774767	142.250.184.10	10.20.75.128	TCP	66	443 → 50300 [ACK] Seq=1 Ack=2 Win=1035 Len=0 SLE=1 SRE=2
227	2.972496	10.20.75.128	35.190.80.1	TCP	55	50312 → 443 [ACK] Seq=1 Ack=1 Win=251 Len=1
228	2.992049	35.190.80.1	10.20.75.128	TCP	66	443 → 50312 [ACK] Seq=1 Ack=2 Win=1044 Len=0 SLE=1 SRE=2
252	3.207275	10.20.75.128	142.250.200.78	TCP	55	50302 → 443 [ACK] Seq=1 Ack=1 Win=255 Len=1
253	3.225656	142.250.200.78	10.20.75.128	TCP	66	443 → 50302 [ACK] Seq=1 Ack=2 Win=1031 Len=0 SLE=1 SRE=2
263	3.304395	10.20.75.128	142.250.201.74	TLSv1.2	119	Application Data
264	3.304440	10.20.75.128	142.250.201.74	TLSv1.2	93	Application Data
265	3.304462	10.20.75.128	142.250.201.74	TLSv1.2	195	Application Data
272	3.308734	10.20.75.128	128.119.245.12	TCP	66	50466 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
273	3.309049	10.20.75.128	128.119.245.12	TCP	66	50467 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
274	3.320735	142.250.201.74	10.20.75.128	TCP	60	443 → 50303 [ACK] Seq=1 Ack=66 Win=1050 Len=0
275	3.320735	142.250.201.74	10.20.75.128	TCP	60	443 → 50303 [ACK] Seq=1 Ack=105 Win=1050 Len=0
276	3.320735	142.250.201.74	10.20.75.128	TCP	60	443 → 50303 [ACK] Seq=1 Ack=246 Win=1050 Len=0
277	3.320735	142.250.201.74	10.20.75.128	TLSv1.2	93	Application Data
278	3.365660	10.20.75.128	142.250.201.74	TCP	54	50303 → 443 [ACK] Seq=246 Ack=40 Win=253 Len=0
290	3.383364	142.250.201.74	10.20.75.128	TLSv1.2	127	Application Data
291	3.383364	142.250.201.74	10.20.75.128	TLSv1.2	560	Application Data
292	3.383364	142.250.201.74	10.20.75.128	TLSv1.2	85	Application Data
293	3.383364	142.250.201.74	10.20.75.128	TLSv1.2	93	Application Data
294	3.383487	10.20.75.128	142.250.201.74	TCP	54	50303 → 443 [ACK] Seq=246 Ack=689 Win=250 Len=0
295	3.384682	10.20.75.128	142.250.201.74	TLSv1.2	89	Application Data

com essa imagem confirmamos mesmo que nenhum dos pacotes mostra Window size value = 0

10.

12

Throughput = Total de dados transferidos / Tempo total da transferência

Total de dados transferidos = $29938 - 152578 = 1460 \times 10 = 14.600 \text{ bytes}$

(Mas atenção, o campo `Ack = 29938` já é em bytes absolutos, ou seja, estamos a medir os bytes enviados desde o início do POST até esse ponto. Como a contagem começa com `152578`, a diferença real é:

$29938 - 0 = 29938 \text{ bytes}$)

✓ Total transferido: 29.938 bytes

✚ 2. Tempo total da transferência

- Início: Pacote com o POST — tempo `≈ 3.416831 s`
- Fim: Último ACK — tempo `≈ 3.671951 s`

📋 Copiar ✎ Editar

Duração = $3.671951 - 3.416831 = 0.255120 \text{ segundos}$

✓ Resposta final:

- Throughput $\approx 117.35 \text{ KB/s}$ (ou $117,350 \text{ bytes/s}$)
- Cálculo:

python

📋 Copiar ✎ Editar

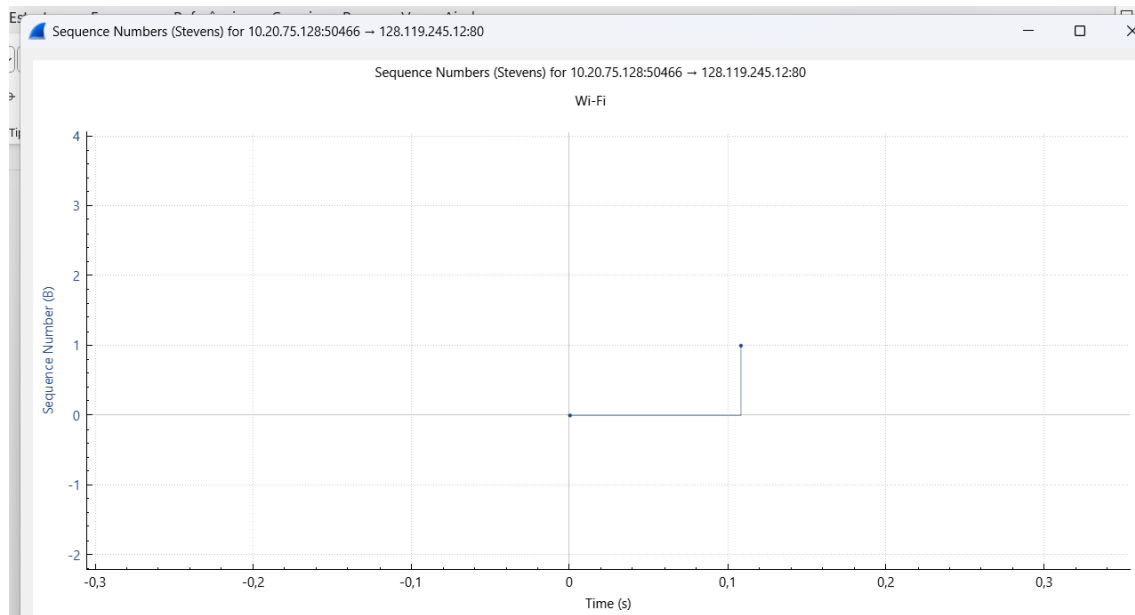
```
Dados transferidos: 29.938 bytes
Tempo da transferência: 0.255 segundos
Throughput = 29938 / 0.255 ≈ 117,350 bytes/s
```

13.

Na minha própria captura, o gráfico “Time-Sequence Graph (Stevens)” não apresentou dados suficientes para observar claramente as fases de slow start e congestion avoidance.

Isso pode ter acontecido porque:

- A captura foi demasiado curta.
- Ou nem todos os pacotes foram capturados no início ou fim da transmissão.



14.

Mesmo sem um gráfico visível, a análise dos segmentos enviados e reconhecidos indica que o TCP iniciou com **slow start**, enviando pacotes em ritmo acelerado, e depois entrou na **fase de congestion avoidance**, com ritmo de envio estabilizado e constante — em conformidade com o comportamento esperado do TCP.