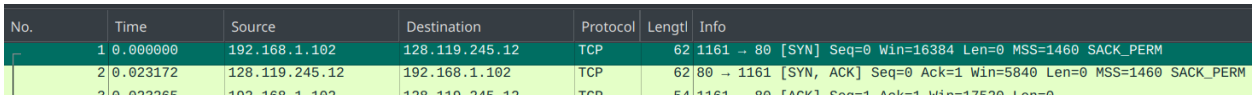
Redes de Computadoras II
TCP

NOMBRE: Daniel Alberto Vinzia

Una primera vista de la traza

1. ¿Cuál es la dirección IP y número de puerto usado por la computadora que realiza la transferencia del archivo a gaia.cs.umass.edu?

```
No.     Time           Source                Destination           Protocol Length Info
      1 0.000000       192.168.1.102         128.119.245.12        TCP      62     1161 → 80 [SYN] Seq=0 Win=16384
```



No.	Time	Source	Destination	Protocol	Lengtl	Info
1	0.000000	192.168.1.102	128.119.245.12	TCP	62	1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM
2	0.023172	128.119.245.12	192.168.1.102	TCP	62	80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM

Como se puede ver tanto en las dos imágenes (una de la impresión completa y luego recortada del primer paquete tcp y la otra una captura de pantalla del programa wireshark), se puede apreciar entre otros datos, los que estamos buscando:

Direccion IP de origen: *192.168.1.102*

Número de puerto: *1161*

2. ¿Cuál es la dirección IP de gaia.cs.umass.edu? ¿En que número de puerto envía y recibe los segmentos TCP para esta conexión?

```
No.     Time           Source                Destination           Protocol Length Info
      2 0.023172       128.119.245.12        192.168.1.102         TCP      62     80 → 1161 [SYN, ACK] Seq=0 Ack=1
```

realizamos el mismo proceso en imprimir ahora un paquete del servidor, y obtenemos los datos

Direccion IP del servidor: *128.119.245.12*

Puerto que envía y recibe datos: *80*

TCP básico

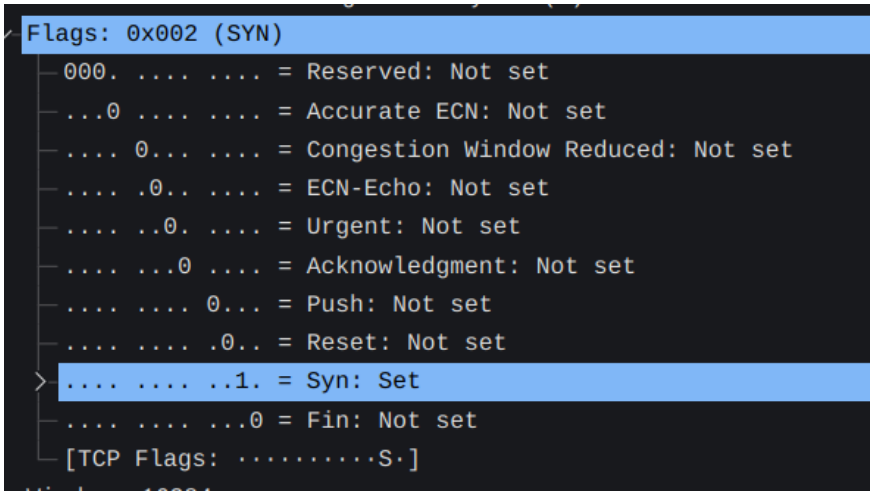
3. ¿Cuál es el número de secuencia del segmento TCP SYN que es usado para iniciar la conexión TCP entre la computadora cliente y gaia.cs.umass.edu? ¿Qué es lo que identifica en el segmento que es un "segmento SYN"?

```
Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 1161
    Destination Port: 80
    [Stream index: 0]
    [Stream Packet Number: 1]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 232129012
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    0111 .... = Header Length: 28 bytes (7)
    Flags: 0x002 (SYN)
    Window: 16384
    [Calculated window size: 16384]
    Checksum: 0xf6e9 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
    Options: (8 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted
    [Timestamps]
```

como vemos en la imagen en la sección de Transmission Control Protocol (TCP), el número de secuencia real es el que dice entre paréntesis *raw* pero aparece debajo de uno que nos proporciona Wireshark para hacer más fácil el seguimiento.

Número de secuencia: *0*

Número de secuencia (raw): *232129012*



Luego en la parte de **Flags** (ampliado en la imagen de arriba), el número hexadecimal *0x002* responde al **SYN**, o en binario *0000 0000 0010*. Esto lo identifica como un segmento SYN.

4. ¿Cuál es el número de secuencia del segmento SYNACK enviado por gaia.cs.umass.edu a la computadora cliente en respuesta al SYN? ¿Cuál es el valor del campo Acknowledgement en el segmento SYNACK? ¿Cómo determinó gaia.cs.umass.edu ese valor? ¿Qué es lo que identifica en el segmento que es un "segmento SYNACK"?

```
Transmission Control Protocol, Src Port: 80, Dst Port: 1161, Seq: 0, Ack: 1, Len: 0
    Source Port: 80
    Destination Port: 1161
    [Stream index: 0]
    [Stream Packet Number: 2]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 883061785
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 232129013
    0111 .... = Header Length: 28 bytes (7)
    Flags: 0x012 (SYN, ACK)
    Window: 5840
    [Calculated window size: 5840]
    Checksum: 0x774d [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
    Options: (8 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted
    [Timestamps]
    [SEQ/ACK analysis]
```

Como se puede ver en la imagen superior:

Numero de segmento: *0*

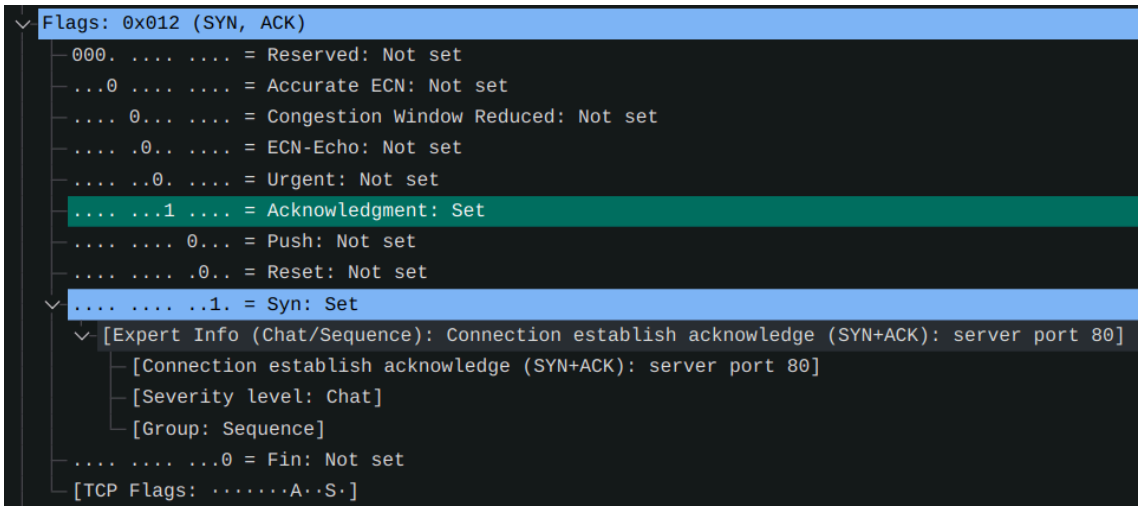
Numero de segmento (raw): *883061785*

Valor del campo Acknowledgement: *0*

Valor del campo Acknowledgement (raw): *232129013*

gaia determina el valor de su segmento de forma aleatoria como lo hizo el cliente.

gaia determina el valor del **ACK** sumandole uno (+1) al valor de segmento que envío el cliente



Lo que identifica el segmento como un **SYN ACK** son los *Flags,* como se ve en la imagen superior. El valor es **0x012** o **0000 0001 0010.**

5. ¿Cuál es el número de secuencia del segmento TCP conteniendo el comando HTTP POST?

Si nos vamos al final del paquete **TCP número 4,** en la parte de ***data*** encontraremos lo siguiente. Donde claramente podemos ver la palabra **POST** y un poco mas adelante se vera el protocolo ***HTTP 1.1***

6. Considera el segmento TCP conteniendo el HTTP POST como el primer segmento en la conexión TCP. ¿Cuáles son los números de secuencia de los primeros seis segmentos en la conexión TCP (incluyendo al segmento que contiene el HTTP POST)? ¿En que momento de tiempo se envió cada segmento? ¿Cuándo se recibió cada ACK? Dada la diferencia entre cada segmento TCP enviado y cuando su asentimiento fué recibido ¿cuál es el valor del RTT para cada uno de los

seis segmentos? ¿Cuál es la estimación del valor del RTT despues de la recepción de cada ACK? Asuma que dicho valor llamado EstimatedRTT es igual al la medición del RTT para el primer segmento y luego es calculado usando la ecuación:

$$EstimatedRTT = 0.875 * EstimatedRTT + 0.125 * SampleRTT$$

	Time	Source	Destination	Protocol	Lengtl	Info
4	0.026477	192.168.1.102	128.119.245.12	TCP	619	1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565
5	0.041737	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460
6	0.053937	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
7	0.054026	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460
8	0.054690	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460
9	0.077294	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
10	0.077405	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460
11	0.078157	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460
12	0.124085	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=3486 Win=11680 Len=0
13	0.124185	192.168.1.102	128.119.245.12	TCP	1201	1161 → 80 [PSH, ACK] Seq=7866 Ack=1 Win=17520 Len=1147
14	0.169118	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=4946 Win=14600 Len=0
15	0.217299	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=6406 Win=17520 Len=0
16	0.267802	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=7866 Win=20440 Len=0

Numero	Secuencia	Secuencia Raw	Tiempo	Numero de respuesta	Tiempo	Tiempo Respuesta
4	1	232129013	0.026477			
5	566	232129578	0.041737	6	0.053937	0.0122
7	2026	232131038	0.054026	9	0.077294	0.02033675
8	3486	232132498	0.05469	12	0.124085	0.03330528125
10	4946	232133958	0.077405	14	0.169118	0.05028187109
11	6406	232135418	0.078157	15	0.217299	0.07115901221

7. ¿Cuál es la longitud de cada uno de los primeros seis segmentos TCP?

- Como se ve en la imagen en la pregunta 6 los tamaños de los primeros 6 paquetes son:

Numero	Tamaño
4	565
5	1460
7	1460
8	1460
10	1460
11	1460

8. ¿Cuál es el mínimo monto disponible de espacio en el buffer advertido en la recepción para toda la traza? ¿La falta de espacio en el buffer del receptor estrangula alguna vez al emisor?

- el mínimo advertido por el receptor en la ventana se ve en el paquete Número 2 de toda la comunicación en **5840**, dicho número va a ir incrementando en cada envío hasta **62780.** Pero como se ve en los envíos, la ventana crece antes de poder generar algún estrangulamiento al emisor.

9. ¿Hay algún segmento transmitido en el archivo de la traza? ¿Qué es lo que constató (en la traza) para poder responder a esta cuestión?

- Si vemos paquete por paquete en el envío, se verá que el envío se trata de un archivo de texto y el receptor solo asienta que va recibiendo cada paquete. el archivo de texto es el libro “Alicia en el país de las maravillas”

10. ¿Cuanta cantidad de datos suele reconocer el receptor en un ACK? ¿Puedes identificar casos donde el receptor esté asintiendo todos los demás segmentos recibidos?

- la transmisión, solo varía en los primeros dos paquetes número 4 y 5, que el asentimiento es conjunto, luego cada paquete tiene su ACK particular.

11. ¿Cuál es el throughput (bytes transferred per unit time) para la conexión TCP? Explica cómo calculaste este valor.

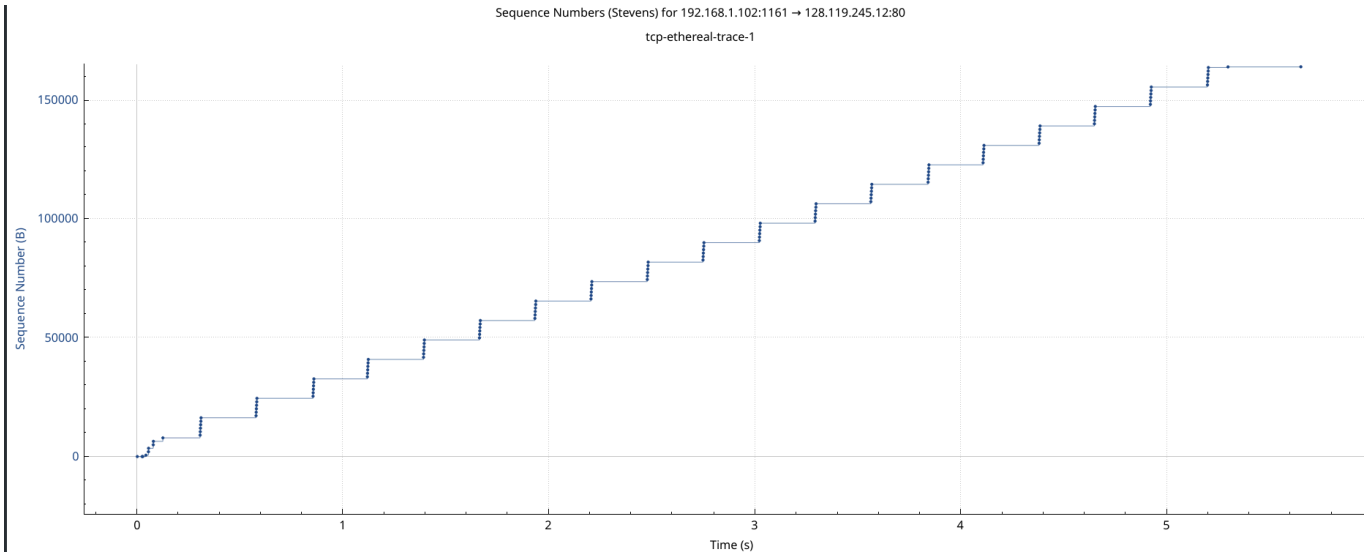
- Como dijimos antes el tamaño empieza en 565 y al siguiente envío es de 1460 Bytes, este número se puede encontrar de manera sencilla con el número de secuencia que nos da Wireshark, que no es el mismo que el número de secuencia RAW. el número de secuencia que se nos brinda, comienza en 1, para poder hacer más legible todo.

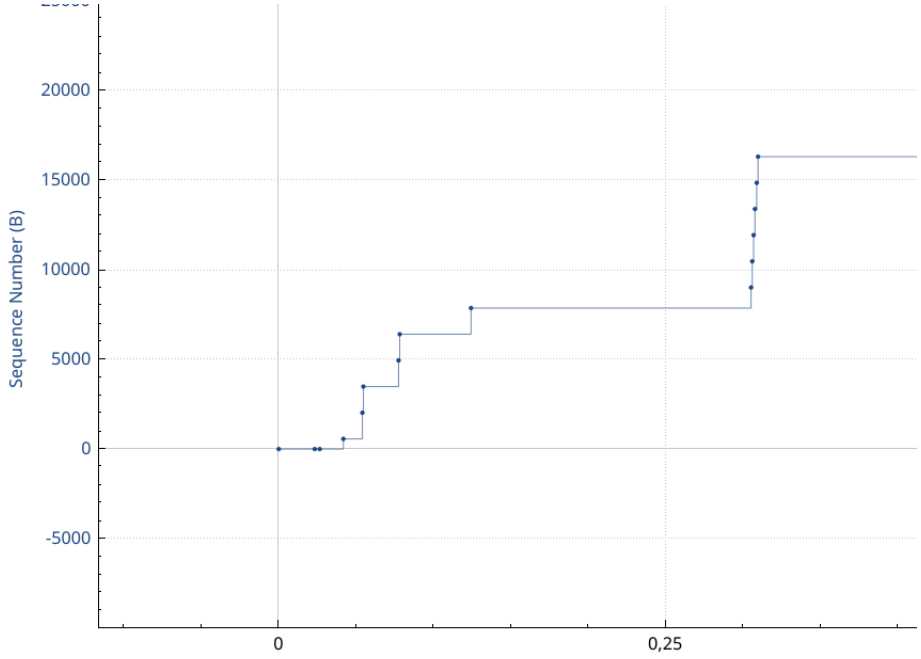
  A este número de secuencia que acompaña a cada envío, se le puede restar el número anterior de secuencia y nos dará el valor de la cantidad de datos transferidos.

  También lo podemos ver individualmente si entramos en cada paquete, dirigirnos a la sección de TCP y buscar el valor que nos dice en [TCP Segment Len: xxxx] donde xxxx es el valor de longitud del segmento.

Congestión en TCP

12. Usando la herramienta de graficación Time-Sequence-Graph(Stevens) para ver los números de secuencia vs el tiempo en que los segmentos fueron enviados. ¿Puedes identificar donde comienza y termina la fase slowstart y donde se hace cargo el congestion avoidance? Comenta en que medida los datos observados difieren del comportamiento idealizado de TCP según se ve en Tanenbaum.





Como podemos ver en la segunda imagen más ampliado el slowstart se da en los primeros envíos y esto cambia, y luego el emisor empieza a mandar 6 paquetes a partir del paquete 18

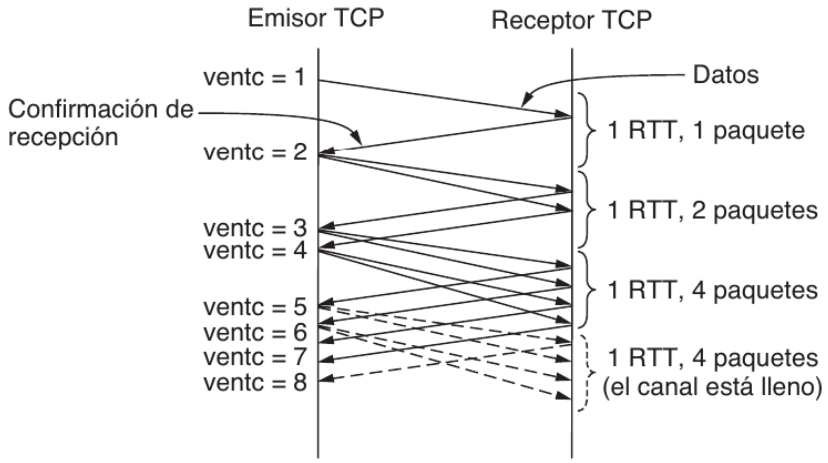


Figura 6-44. Inicio lento desde una ventana de congestión inicial de un segmento.

como podemos ver en este ejemplo que nos muestra el Tanenbaum se comporta de la misma forma al muestreo, aumenta la cantidad de envíos ya que siempre se le confirma que los paquetes han llegado