

2. Un usuario envía un correo usando un programa similar al implementado en clase de la siguiente forma:

```
alumno@pc512~$ ./myEmailClient
```

```
From: rysd-profesor@rysd.es
```

```
To: rysd-alumno@rysd.es
```

```
Mensaje:
```

```
Para hacer bien este examen hay que saber del protocolo SMTP.
```

```
¡Correo enviado con éxito!
```

```
alumno@pc512:~$
```

a) Indicar qué mensajes SMTP envía el cliente, en qué orden y con qué contenido:

C0: HELO myEmailClient

C1: MAIL FROM: rysd-profesor@rysd.es

C2: RCPT TO: rysd-alumno

C3: DATA

C4: {G xqqs

C5: }

C6: QUIT

C7:

C8:

C9:

Cliente

$s=1$  seq 200  $len=20$   $data=0$

Secu

$s=1$  seq 200  $len=20$

$A=1$  ack 300  $T=20$

$A=1$  seq 300  $T=20$

$A=1$  ack 400

$A=1$  seq 400  $T=30$

Send (

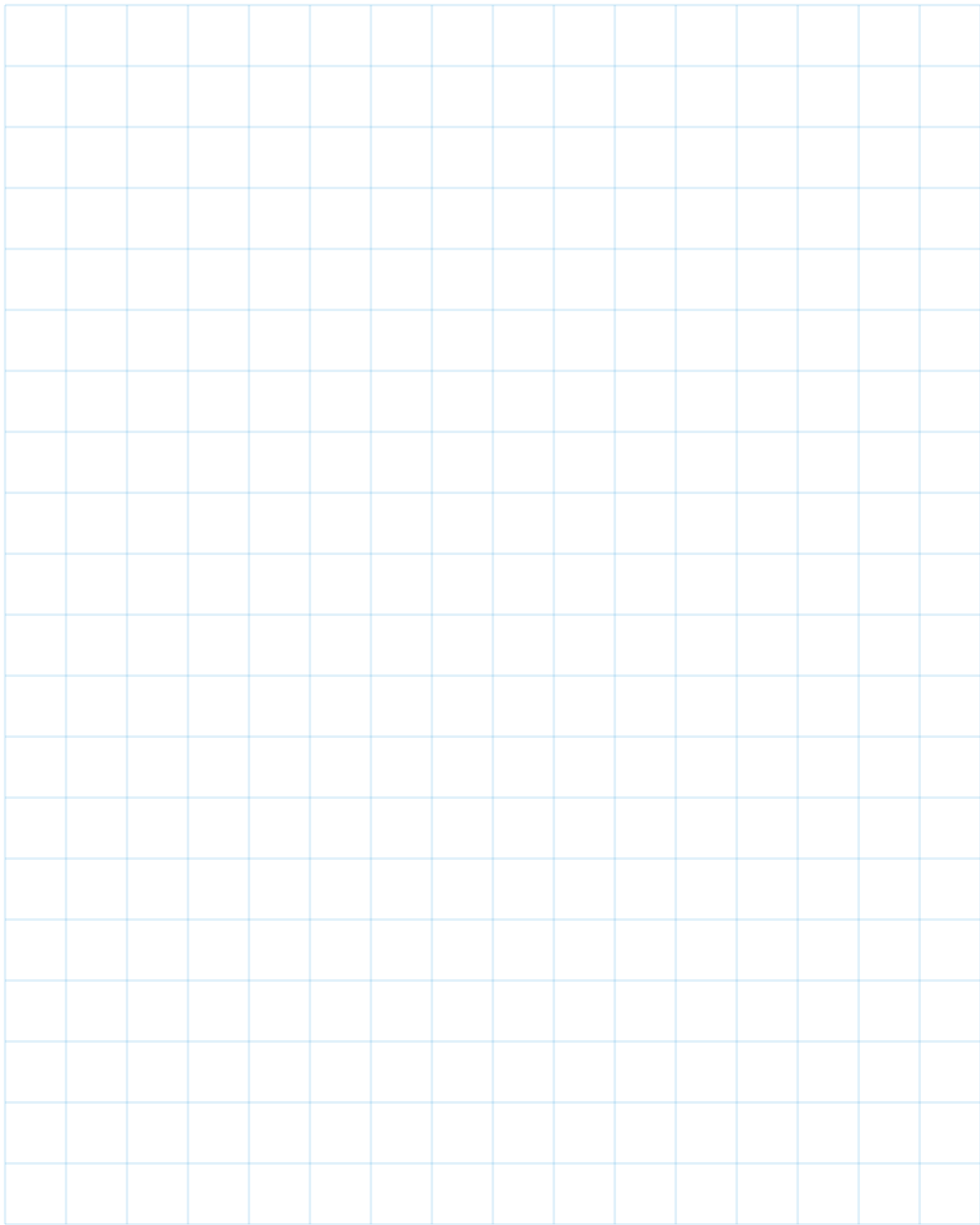
$A=1$  ack 300

$A=1$  seq 300

$A=1$  ack 400

Send('HELO')

$4 \times 2 = 6$



3. Un usuario anónimo (IP: 192.168.177.49) quiere conectarse al servidor de FTP público de la UMA (IP: 150.214.40.67). La comunicación que se establece es la siguiente:

*El servidor empieza a escuchar en el puerto estándar de FTP.*

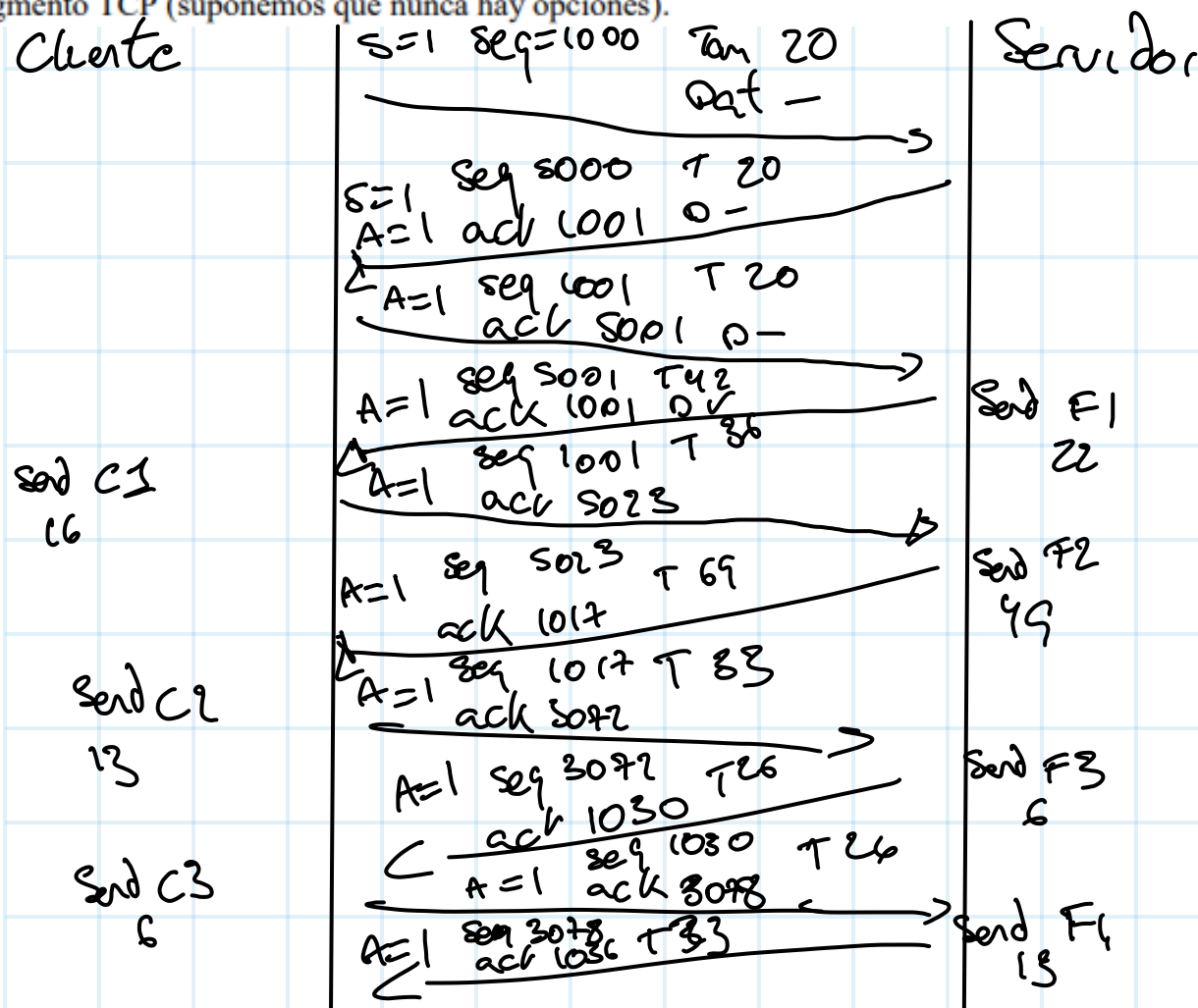
*El cliente inicia la conexión al servidor.*

Mensaje	ID	Origen	Tamaño
200 FTP Server Ready	F1	Servidor	22 bytes
USER anonymous	C1	Cliente	16 bytes
331 Anonymous login ok, send your email address	F2	Servidor	49 bytes
PASS a@a.es	C2	Cliente	13 bytes
230-	F3	Servidor	6 bytes
QUIT	C3	Cliente	6 bytes
221 Goodbye	F4	Servidor	13 bytes

*El servidor cierra la conexión con el cliente.*

*El cliente también cierra la conexión con el servidor.*

En la tabla anterior, el texto en *courier* son los mensajes enviados, el resto es información adicional: identificador de mensaje (no forma parte del protocolo, solo se utiliza para identificar los mensajes en el ejercicio), origen del mensaje y el tamaño del mismo. Suponiendo la anterior conversación realice el diagrama de secuencia TCP. Suponga que el número de secuencia utilizado por el cliente es 1000 y el del servidor 5000. En cada envío debe aparecer: número de secuencia, número de confirmación (si el bit ACK está activo), flags activos (A para ACK, S para SYN y F para FIN), mensaje enviado (si se envían datos, utilice los IDs de los mensajes en vez de su contenido) y el tamaño del segmento TCP (suponemos que nunca hay opciones).



$F=1$   $A=1$  seq 1036  
 $acn$  3091 T20  
 $F=1$  seq 8091  
 $A=1$   $acn$  1036 T20  
 $A=1$  seq 1036  
 $acn$  3091 T20

5. Un usuario en un navegador introduce la URL `http://www.et.com/micasa.html` y se abre una página cuyo contenido es poco más que una imagen de un teléfono. Examinando las trazas capturadas se observa que se establece una conexión HTTP persistente con el servidor `www.et.com` y hace dos peticiones HTTP de la siguiente forma:

Realice el diagrama de secuencia TCP relacionado con dicha conversación. Suponga que el **número de secuencia** utilizado por el **cliente** es **2000** y el del **servidor** **4000**, que el **MSS** es **1460** y que **cada segmento es confirmado** (usando **piggybacking** cuando sea posible). En cada envío debe aparecer: **número de secuencia** (seq), **número de confirmación** (ack) (si el bit ACK está activo), **flags** activos (A para ACK, S para SYN y F para FIN), **mensaje enviado** (si se envían datos, se pueden utilizar los IDs -P1, R1, P2, R2- en vez de los datos completos) y el **tamaño completo del segmento TCP**.

