

# Práctica 5 - Sesión SIP

Protocolos para la Transmisión de Audio y Vídeo en Internet

Versión 10.0 - 23.11.2020

Nota: Esta práctica se puede entregar para su evaluación como parte de la nota de prácticas. Para las instrucciones de entrega, mira al final del documento.

## 1 Introducción

El protocolo de iniciación de sesión (SIP) es un protocolo que se limita solamente al establecimiento y control de una sesión (RFC 3550). Los detalles del intercambio de datos, como por ejemplo la codificación o decodificación del audio/vídeo, no son controlados por SIP sino que se llevan a cabo por otros protocolos (por ejemplo, RTP). Los principales objetivos de SIP son:

- SIP permite el establecimiento de una localización de usuario (o sea, traducir de un nombre de usuario a su dirección de red actual)
- SIP provee funcionalidad para la negociación de las características de una sesión, de manera que los participantes en una sesión pueden consensuar características soportadas por todos ellos.
- SIP es un mecanismo para la gestión de llamadas, por ejemplo para añadir, eliminar o transferir participantes
- SIP permite la modificación de características de la sesión durante su transcurso.

## 2 Objetivos de la práctica

1. Conocer el protocolo SIP y otros protocolos utilizados en una sesión con clientes SIP (como Ekiga o linphone).
2. Profundizar en el uso de Wireshark: análisis, captura y filtrado.

### 3 Conocimientos previos necesarios

- Nociones de SIP y RTP (las de clase de teoría)
- Funcionamiento de Wireshark.

Tiempo estimado (para un alumno medio): 8 horas

### 4 Ejercicios

#### Creación de repositorio para la práctica

1. Con el navegador, dirígete al repositorio `ptavi-p5` en la cuenta de la asignatura en GitLab<sup>1</sup> y realiza un `fork`, de manera que consigas tener una copia del repositorio en tu cuenta de GitLab. Clona el repositorio que acabas de crear a local para poder editar los archivos. Trabaja a partir de ahora en ese repositorio, sincronizando los cambios que vayas realizando.

Como tarde al final de la práctica, deberás realizar un `push` para subir tus cambios a tu repositorio en GitLab. En esta práctica, al contrario que con las demás, se recomienda hacer frecuentes `commits`, pero el `push` al final.

#### Análisis de una sesión SIP

Se ha capturado una sesión SIP con el cliente SIP Ekiga (archivo `sip.cap.gz`), que se puede abrir con Wireshark<sup>2</sup>. Se pide rellenar las cuestiones que se plantean en este guión en el fichero `p5.txt` que encontrarás también en el repositorio.

2. Observa que las tramas capturadas corresponden a una sesión SIP con Ekiga, un cliente de VoIP para GNOME. Responde a las siguientes cuestiones:
  - ¿Cuántos paquetes componen la captura?
  - ¿Cuánto tiempo dura la captura?
  - ¿Qué IP tiene la máquina donde se ha efectuado la captura? ¿Se trata de una IP pública o de una IP privada? ¿Por qué lo sabes?

---

<sup>1</sup><http://gitlab.etsit.urjc.es/ptavi/ptavi-p5>

<sup>2</sup>Desde la shell: `$ wireshark sip.cap.gz` - recuerda que puedes utilizar el tabulador en la shell para autocompletar.

3. Antes de analizar las tramas, mira las estadísticas generales que aparecen en el menú de **Statistics**. En el apartado de jerarquía de protocolos (**Protocol Hierarchy**) se puede ver el porcentaje del tráfico correspondiente al protocolo TCP y UDP.
  - ¿Cuál de los dos es mayor? ¿Tiene esto sentido si estamos hablando de una aplicación que transmite en tiempo real?
  - ¿Qué otros protocolos podemos ver en la jerarquía de protocolos? ¿Cuales crees que son señal y cuales ruido?
4. Observa por encima el flujo de tramas en el menú de **Statistics** en **IO Graphs**. La captura que estamos viendo incluye desde la inicialización (registro) de la aplicación hasta su finalización, con una llamada entremedias.
  - Filtra por **sip** para conocer cuándo se envían paquetes SIP. ¿En qué segundos tienen lugar esos envíos?
  - Y los paquetes con RTP, ¿cuándo se envían?

[Al terminar el ejercicio es recomendable hacer **commit** de los ficheros modificados]
5. Analiza las dos primeras tramas de la captura.
  - ¿Qué servicio es el utilizado en estas tramas?
  - ¿Cuál es la dirección IP del servidor de nombres del ordenador que ha lanzado Ekiga?
  - ¿Qué dirección IP (de **ekiga.net**) devuelve el servicio de nombres?
6. A continuación, hay más de una docena de tramas TCP/HTTP.
  - ¿Podrías decir la URL que se está pidiendo?
  - ¿Qué **user agent** (UA) la está pidiendo?
  - ¿Qué devuelve el servidor?
  - Si lanzamos el navegador web, por ejemplo, **Mozilla Firefox**, y vamos a la misma URL, ¿qué recibimos? ¿Qué es, entonces, lo que está respondiendo el servidor?
7. Hasta la trama 45 se puede observar una secuencia de tramas del protocolo STUN.
  - ¿Por qué se hace uso de este protocolo?
  - ¿Podrías decir si estamos tras un NAT o no?

8. La trama 46 es la primera trama SIP. En un entorno como el de Internet, lo habitual es desconocer la dirección IP de la otra parte al realizar una llamada. Por eso, todo usuario registra su localización en un servidor **Registrar**. El **Registrar** guarda información sobre los usuarios en un servidor de localización que puede ser utilizado para localizar usuarios.

- ¿Qué dirección IP tiene el servidor **Registrar**?
- ¿A qué puerto (del servidor **Registrar**) se envían los paquetes SIP?
- ¿Qué método SIP utiliza el UA para registrarse?
- Además de **REGISTER**, ¿podrías decir qué instrucciones SIP entiende el UA?

[Al terminar el ejercicio es recomendable hacer **commit** de los ficheros modificados]

9. Fijémonos en las tramas siguientes a la número 46:

- ¿Se registra con éxito en el primer intento?
- ¿Cómo sabemos si el registro se ha realizado correctamente o no?
- ¿Podrías identificar las diferencias entre el primer intento y el segundo de registro? (fíjate en el tamaño de los paquetes y mira a qué se debe el cambio)
- ¿Cuánto es el valor del tiempo de expiración de la sesión? Indica las unidades.

10. Una vez registrados, podemos efectuar una llamada. Vamos a probar con el servicio de eco de Ekiga que nos permite comprobar si nos hemos conectado correctamente. El servicio de eco tiene la dirección **sip:500@ekiga.net**. Veamos el **INVITE** de cerca.

- ¿Puede verse el nombre del que efectúa la llamada, así como su dirección SIP?
- ¿Qué es lo que contiene el cuerpo de la trama? ¿En qué formato/protocolo está?
- ¿Tiene éxito el primer intento? ¿Cómo lo sabes?
- ¿En qué se diferencia el segundo **INVITE** más abajo del primero? ¿A qué crees que se debe esto?

11. Una vez conectado, estudia el intercambio de tramas.

- ¿Qué protocolo(s) se utiliza(n)? ¿Para qué sirven estos protocolos?
- ¿Cuál es el tamaño de paquete de los mismos?

- ¿Se utilizan bits de **padding**?
- ¿Cuál es la periodicidad de los paquetes (en origen; nota que la captura es en destino)?
- ¿Cuántos bits/segundo se envían?

[Al terminar el ejercicio es recomendable hacer **commit** de los ficheros modificados]

12. Vamos a ver más a fondo el intercambio RTP. En **Telephony** hay una opción RTP. Empecemos mirando los flujos RTP.

- ¿Cuántos flujos hay? ¿por qué?
- ¿Cuántos paquetes se pierden?
- ¿Cuál es el valor máximo del **delta**? ¿Y qué es lo que significa el valor de **delta**?
- ¿Cuáles son los valores de **jitter** (medio y máximo)? ¿Qué quiere decir eso? ¿Crees que estamos ante una conversación de calidad?

13. Elige un paquete RTP de audio. Analiza el flujo de audio en **Telephony** -> RTP -> **Stream Analysis**.

- ¿Cuánto valen el **delta** y el **jitter** para el primer paquete que ha llegado?
- ¿Podemos saber si éste es el primer paquete que nos han enviado?
- Los valores de **jitter** son menores de 10ms hasta un paquete dado. ¿Cuál?
- ¿A qué se debe el cambio tan brusco del **jitter**?
- ¿Es comparable el cambio en el valor de **jitter** con el del **delta**? ¿Cual es más grande?

14. En **Telephony** selecciona el menú **VoIP calls**. Verás que se lista la llamada de voz IP capturada en una ventana emergente. Selecciona esa llamada y pulsa el botón **Play Streams**.

- ¿Cuánto dura la conversación?
- ¿Cuáles son sus SSRC? ¿Por qué hay varios SSRCs? ¿Hay CSRCs?

15. Identifica la trama donde se finaliza la conversación.

- ¿Qué método SIP se utiliza?
- ¿En qué trama(s)?
- ¿Por qué crees que se envía varias veces?

16. Finalmente, se cierra la aplicación de VoIP.

- ¿Por qué aparece una instrucción SIP del tipo **REGISTER**?
- ¿En qué trama sucede esto?
- ¿En qué se diferencia con la instrucción que se utilizó con anterioridad (al principio de la sesión)?

[Al terminar el ejercicio es recomendable hacer **commit** de los ficheros modificados]

## Captura de una sesión SIP

- Dirígete a la web de Linphone<sup>3</sup> con el navegador y créate una cuenta SIP. Recibirás un correo electrónico de confirmación en la dirección que has indicado al registrarte (mira en tu carpeta de *spam* si no es así).
- Lanza **linphone**, y configúralo con los datos de la cuenta que te acabas de crear. Para ello, puedes ir al menú “Ayuda” y seleccionar “Asistente de Configuración de Cuenta”. Al terminar, cierra completamente **linphone**.
- Captura una sesión SIP de una conversación con el número SIP **sip:music@sip.iptel.org**. Recuerda que has de comenzar a capturar tramas antes de arrancar **linphone** para ver todo el proceso<sup>4</sup>.
- Observa las diferencias en el inicio de la conversación entre el entorno del laboratorio y el del ejercicio anterior<sup>5</sup>:
  - ¿Se utilizan DNS y STUN? ¿Por qué?
  - ¿Son diferentes el registro y la descripción de la sesión?
- Identifica las diferencias existentes entre esta conversación y la conversación anterior:
  - ¿Cuántos flujos tenemos?
  - ¿Cuál es su periodicidad?
  - ¿Cuánto es el valor máximo del delta y los valores medios y máximo del jitter?
  - ¿Podrías reproducir la conversación desde **Wireshark**? ¿Cómo? Comprueba que poniendo un valor demasiado pequeño para el **buffer** de **jitter**, la conversación puede no tener la calidad necesaria.

---

<sup>3</sup><https://www.linphone.org/freesip/home>

<sup>4</sup>Puedes *matar* **linphone** desde la línea de **shell** buscando el identificador del proceso con `$ ps aux` y terminándolo con `$ kill -9 pid`, donde *pid* es el identificador del proceso.

<sup>5</sup>Si estás utilizando tu portátil en vez de una máquina del laboratorio, ten en cuenta que la captura puede ser diferente, porque con el portátil estarás conectado a la red *wifi*, mientras que los ordenadores de los laboratorios lo están a una LAN.

- ¿Sabrías decir qué tipo de servicio ofrece `sip:music@iptel.org`?

[Al terminar el ejercicio es recomendable hacer `commit` de los ficheros modificados]

22. Filtra por los paquetes SIP de la captura y guarda **únicamente** los paquetes SIP como `p5.pcapng`. Abre el fichero guardado para cerciorarte de que lo has hecho bien. Deberás añadirlo al repositorio.

[Al terminar el ejercicio es recomendable hacer `commit` de los ficheros modificados]

[Al terminar la práctica, realiza un `push` para sincronizar tu repositorio GitLab]

## 5 Fecha y modo de entrega

La entrega de práctica se deberá hacer antes del miércoles 2 de diciembre de 2020 a las 23:59. Para entonces, se debe:

- Tener un repositorio git en GitLab con:
  1. Un archivo de texto con los comentarios a los ejercicios en `p5.txt`.
  2. Un archivo con una captura filtrada en `p5.pcapng`.
  3. Un test de Moodle rellenado sobre la práctica 5. Este test tendrá preguntas relacionadas con los ejercicios que se han realizado y tiene un tiempo de realización de 20 minutos.
  4. 4 ficheros adicionales (además de `.git`): `README.md`, `LICENSE`, `check-p5.py` y `.gitignore`.

Se ha de tener en cuenta las siguientes consideraciones:

- Se valorará que al menos haya diez `commits` realizados en al menos dos días diferentes
- Se valorará que la captura esté bien filtrada y sólo contenga los paquetes que se indican en el guión.

Se puede comprobar la correcta entrega de la práctica utilizando el programa `check-p5.py`. Este programa se ejecuta desde la línea de comandos de la siguiente manera:

```
$ python check-p5.py login
```

donde `login` es tu nombre de usuario en los laboratorios docentes. El programa comprueba que se han entregado los ficheros que se solicitan (y sólo esos).