Redes Multimedia – Prácticas 2019
Práctica 3: VoIP

Turno y pareja: 4312_02

Integrantes:

Jorge Gutiérrez Díaz

Javier Martín González

Contenido

Contenido

1 Introducción

A lo largo de la práctica, hemos configurado un servicio VoIP basado en el protocolo SIP mediante la utilización de la herramienta Yate, entre lo que se incluye la puesta en marcha del servidor, la configuración de dos usuarios que serán los clientes (cada uno con su usuario y contraseña) y la posibilidad de crear marcaciones rápidas basado en extensiones.

Una vez verificado el correcto funcionamiento y haber aprendido los aspectos de la configuración y cómo establecer las comunicaciones, hemos comprobado la interoperabilidad entre clientes y servidores que actúen bajo el mismo protocolo, para lo que se utiliza Yate, Ekiga en la parte cliente, y Yate y Linphone en la parte servidor.

2 Realización de la práctica

1. Cada pareja procederá a configurar el servidor Yate en su ordenador de trabajo. Para ejecutar este software deberá seguir la documentación disponible en el *site* de Yate (yate.ro). En los laboratorios ya existe una instancia del software Yate corriendo en el puerto 5060. Como no podemos parar dicha instancia, a la hora de lanzar nuestra propia instancia **utilizaremos a todos los efectos el puerto 5061** como puerto de SIP.

Indique aquí los pasos dados para configurar el servidor Yate. Los siguientes cambios se realizan en la carpeta conf.d donde se localizan todos los ficheros de configutación de *Yate*. Se modifica el puerto por defecto de [general] a **5060** dentro del fichero **ysipchan.conf.sample.**

2. A continuación se procederá a configurar dos usuarios (uno para cada miembro de la pareja) en el repositorio de Yate. Note que Yate permite diferentes modos de configuración de usuarios (base de datos, LDAP, fichero local o cualquier otro método que podamos implementar). En este caso esta configuración se realizará usando un fichero de texto.

   Asociado a esta configuración, la pareja procederá a configurar el *softphone* Yate contra el servidor local usando la información de autentificación que acaba de configurar en Yate. Puede comprobar que la configuración es correcta llamando desde el *softphone* a uno de los teléfonos de prueba configurados en Yate, p.ej. el 99991001[1]. Al llamar a ese número de teléfono, se debe establecer la llamada automáticamente y se deberá oír un tono de

1 Para que esto funcione es necesario cambiar el nombre del fichero conf.d/regexroute.conf.sample a conf.d/regexroute.conf antes de arrancar el servidor Yate.

marcado. El fichero de configuración utilizado debe ser adjuntado a la hora de entregar la práctica.[2]

Indique aquí los pasos dados para configurar cada uno de los usuarios. En el fichero conf.d/regfile.conf, se añaden los nuevos usuarios en el formato: [user] password=contraseña Se cambia la extensión del fichero conf.d/**regexroute.conf.sample** a conf.d/**regexroute.conf** para llamar a los teléfonos de prueba. Se ejecuta el script **autogen.sh** y el ejecutable **configure**, y por último ejecutamos **make** para obtener los siguiente desplegables: **yate** y **run**. Se ejecuta **run.** Dentro del panel de Yate, se accede a Settings < Accounts, y se añade uno de los usaurios que de han introducido en **regfile.conf** con **server** y **domain** en la dirección IP actual.

3. Una vez configurado inicialmente el servidor y los *softphones*, vamos a proceder a hacer un análisis básico del funcionamiento real del protocolo SIP usando Wireshark. Para eso dispondremos de dos ordenadores, en uno de ellos se ejecutará el servidor Yate y un *softphone* Yate (A) configurado con uno de los usuarios creados y en otro el otro *softphone* Yate (B) configurado con el otro usuario. A continuación se realizarán los siguientes experimentos, **para cada uno de ellos deberá realizar una captura de Wireshark**:
   3.1. Registrar los *softphones* con el servidor (para ello, ciérrelos previamente).
   3.2. Llamar desde el *softphone* B al A. El usuario A deberá aceptar la llamada y verificar que se ha establecido la llamada de voz. Pasados unos segundos el usuario A podrá colgar la llamada.
   3.3. Igual que en el caso anterior el usuario B llamará al A, pero en este caso se rechazará la llamada.
   3.4. Apagamos el *softphone* A y volvemos a intentar la llamada.

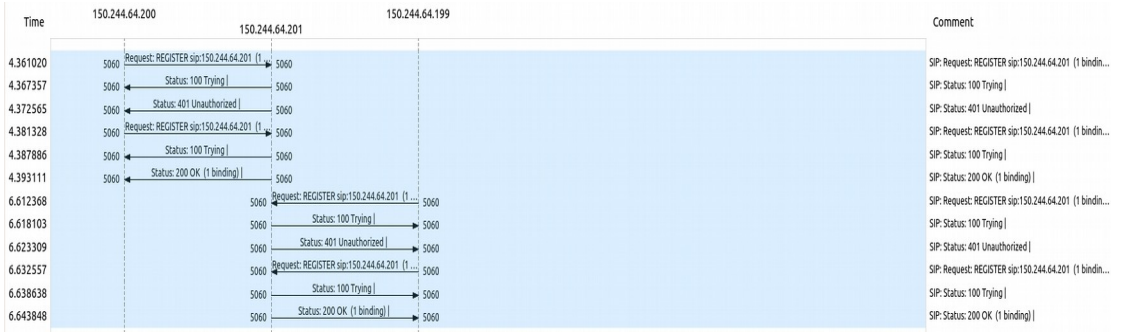
   Para cada uno de estos casos se analizará la captura de Wireshark y se realizarán las siguientes tareas:

   3.5. Dibujar un diagrama de secuencia en el que figuren el UA del usuario A, el UA del usuario B y el *proxy SIP*, y todo el intercambio de mensajes SIP que ha habido entre cada componente. Hay que tener en cuenta que en la captura Wireshark no aparecerán los mensajes intercambiados entre el UA del usuario A y el proxy si están en el mismo equipo Windows, por lo que se recomienda utilizar 3 ordenadores para ello.

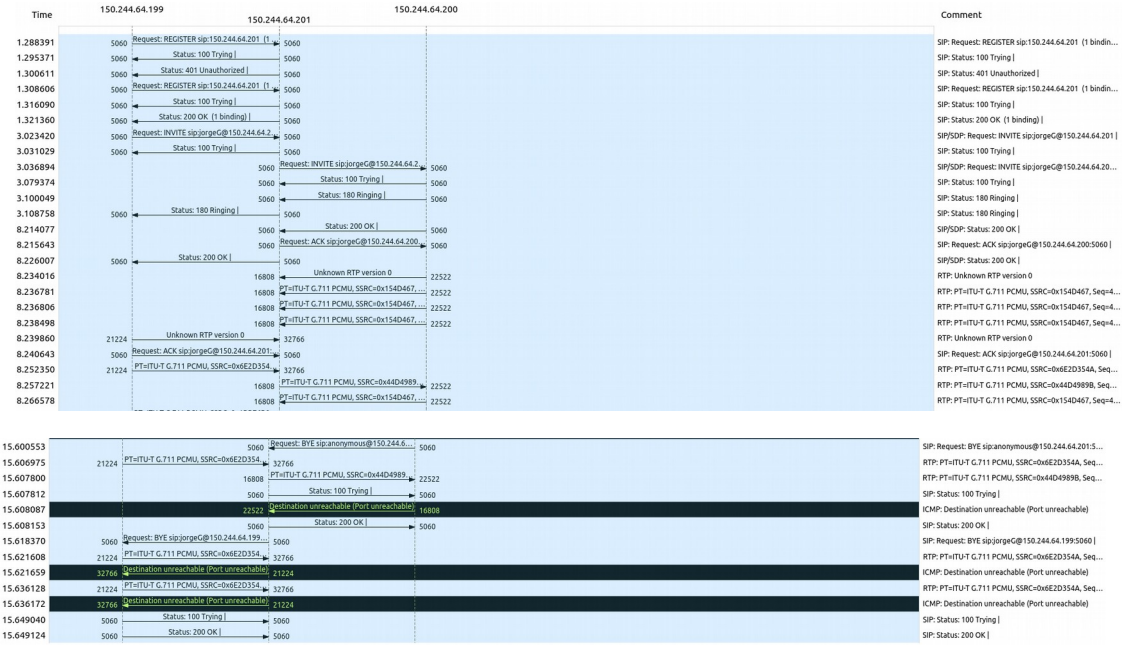
---

2 La ruta de los ficheros de configuración de Yate se encuentra en la carpeta yate/conf.d. Para que los ficheros sean válidos debe guardarse una copia del fichero sin la extensión .sample.

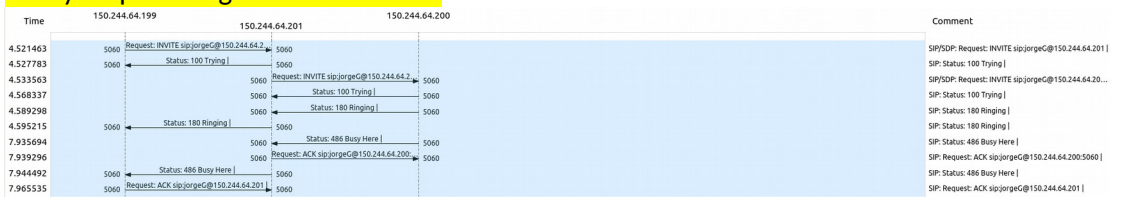
Incluya aquí el diagrama del caso 3.1



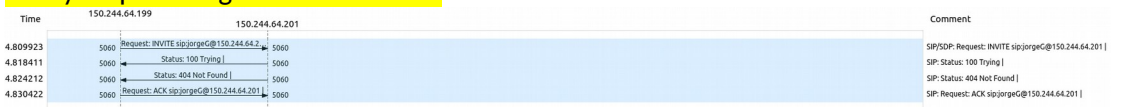
Incluya aquí el diagrama del caso 3.2



Incluya aquí el diagrama del caso 3.3



Incluya aquí el diagrama del caso 3.4



3.6. Recuperar el contenido del mensaje SDP (embebido en SIP) que se intercambia cada UA y de ahí inferir los tipos de datos/codecs multimedia que cada UA ofrece, y determinar el tipo de códec y tasa binaria finalmente usados para los datos de audio intercambiados.

Indique el contenido del mensaje SDP y los tipos de Codecs ofrecidos por los agentes de usuario en aquellos casos en que se presente esta información, indicando en qué casos se presenta.

Filtramos los paquetes por “sip or rtp” y visualizamos aquellos con mensajes SDP:

```
Message Body
  Session Description Protocol
    Session Description Protocol Version (v): 0
    Owner/Creator, Session Id (o): yate 1554137257 1554137257 IN IP4 150.244.64.199
    Session Name (s): SIP Call
    Connection Information (c): IN IP4 150.244.64.199
    Time Description, active time (t): 0 0
    Media Description, name and address (m): audio 21224 RTP/AVP 0 8 11 98 97 105 106 101
      Media Type: audio
      Media Port: 21224
      Media Protocol: RTP/AVP
      Media Format: ITU-T G.711 PCMU
      Media Format: ITU-T G.711 PCMA
      Media Format: 16-bit uncompressed audio, monaural
      Media Format: DynamicRTP-Type-98
      Media Format: DynamicRTP-Type-97
      Media Format: DynamicRTP-Type-105
      Media Format: DynamicRTP-Type-106
      Media Format: DynamicRTP-Type-101
    Media Attribute (a): rtpmap:0 PCMU/8000
    Media Attribute (a): rtpmap:8 PCMA/8000
    Media Attribute (a): rtpmap:11 L16/8000
    Media Attribute (a): rtpmap:98 iLBC/8000
    Media Attribute (a): fmtp:98 mode=20
    Media Attribute (a): rtpmap:97 iLBC/8000
    Media Attribute (a): fmtp:97 mode=30
    Media Attribute (a): rtpmap:105 iSAC/16000
    Media Attribute (a): rtpmap:106 iSAC/32000
    Media Attribute (a): rtpmap:101 telephone-event/8000
    Media Attribute (a): ptime:30
```

En ambos agentes usuarios se oferecen los códec G.711 PCMU y G.711 PCMA. Adicionalmente, los códec que se encuentran debajo son subtipos de códec para carga no estática de RTP.

Este tipo de información se transmite en los paquetes INVITE y en aquellos paquetes que aceptan la llamada con el código OK.

Indique el tipo de códec utilizado y la tasa binaria en aquellos casos en que se presente esta información, indicando en qué casos se presenta.

Los mensajes SIP no utilizan ningún tipo de códec.

A continuación se muestra el bit rate de los mensajes SIP de los clientes al servidor:

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
150.244.64.199	5060	150.244.64.201	5060	6	2.813	2	1.110	4	1.703	6.612368	0.0315	282 k	432 k
150.244.64.200	5060	150.244.64.201	5060	6	2.791	2	1.101	4	1.690	4.361020	0.0321	274 k	421 k

3.7. Determinar direcciones IP y puertos de origen y de destino de los flujos RTP intercambiados, si los hubiera.

Indique las direcciones IP y puertos de origen y destino de los flujos RTP en aquellos casos en que se presente esta información, indicando en qué casos se presenta.

Agente Usuario A a Servidor:

IP origen – 150.244.64.199

IP destino – 150.244.64.201

Puerto origen - 21224

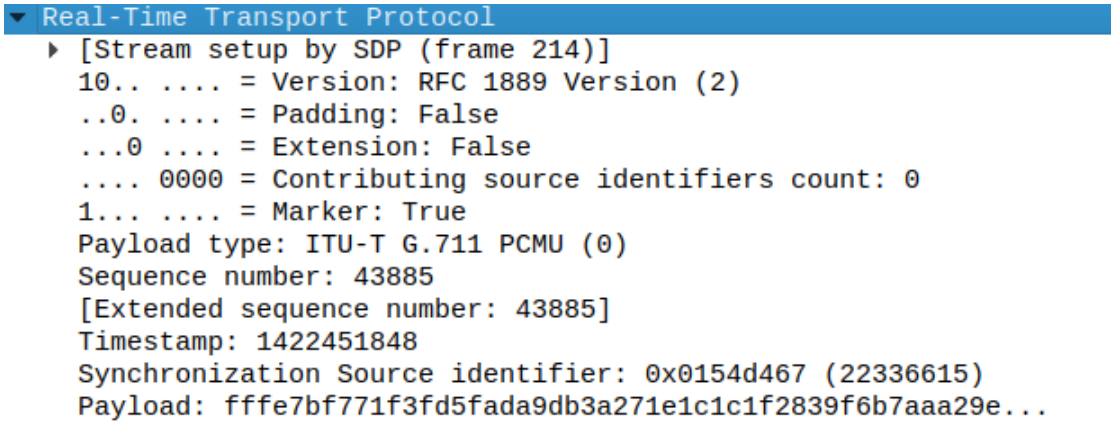
Puerto destino – 32766

Agente Usuario B a Servidor:
IP origen – 150.244.64.200
IP destino – 150.244.64.201
Puerto origen - 22522
Puerto destino – 16808

Servidor a Agente Usuario A:
IP origen – 150.244.64.201
IP destino – 150.244.64.199
Puerto origen - 32766
Puerto destino – 21224

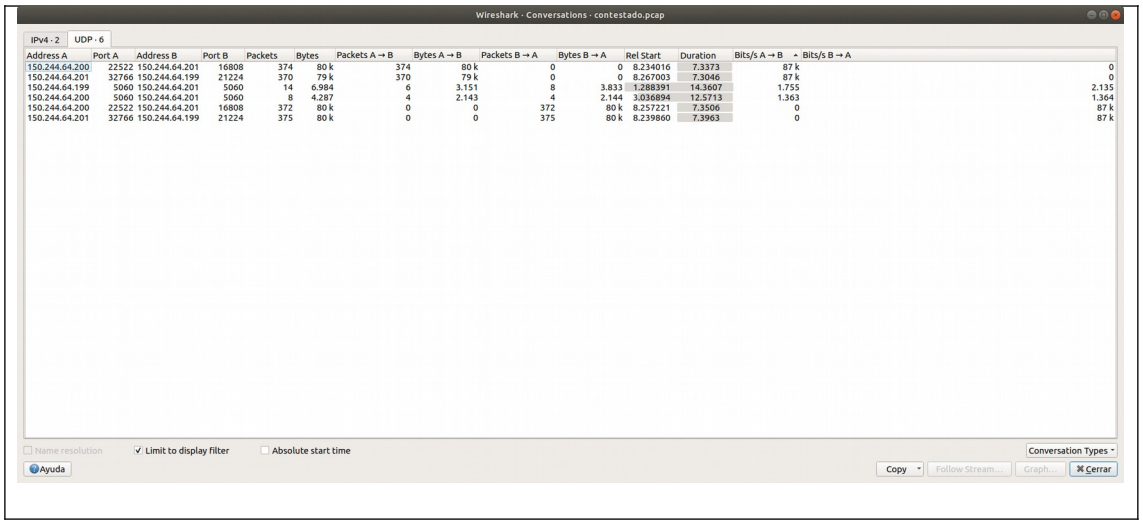
Servidor a Agente Usuario B:
IP origen – 150.244.64.201
IP destino – 150.244.64.199
Puerto origen - 32766
Puerto destino – 21224

Indique el tipo de códec utilizado y la tasa binaria en aquellos casos en que se presente esta información, indicando en qué casos se presenta.



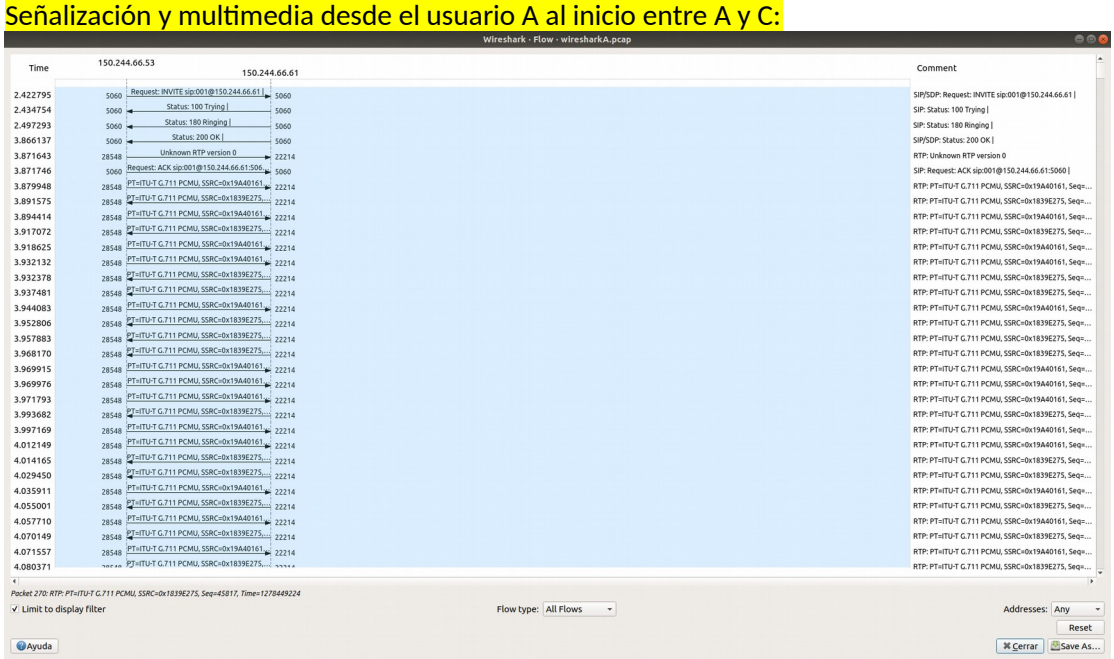
El tipo de códec utilizado se pude observar en los paquetes RTP que vienen a continuación de los mensajes de negociación. Para este caso, el códec utilizado es G.711 PCMU.

Para obtener el bit rate de la conversación accedemos a Statistics < Conversation y observamos los paquetes UDP:

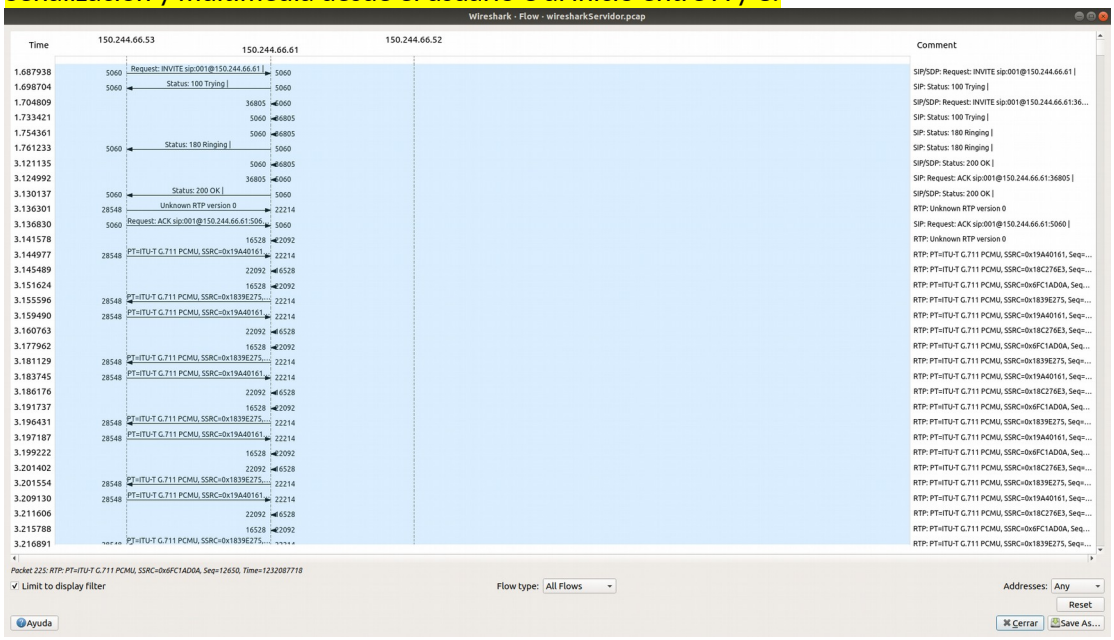


4. Posteriormente, se estudiará la arquitectura de multi-llamada de Yate. Para ello, se debe añadir un nuevo usuario a la configuración de usuarios del servidor. Una vez añadido, abra dos usuarios (A y B) en un ordenador, y el usuario (C) restante y el servidor en otro ordenador. Realice sendas capturas de Wireshark en los dos ordenadores utilizados mientras realiza el siguiente experimento:
   4.1. Comience una llamada entre el usuario A y el usuario C. Acepte la llamada y, acto seguido, añada desde el usuario A al usuario B en la comunicación.
   4.2. Obtenga el diagrama de flujo de las dos capturas realizadas y compruebe la transmisión de mensajes entre los participantes (tanto de señalización como multimedia). Compare lo observado con el mecanismo de establecimiento de multi-llamadas mediante adición de usuarios especificado en la RFC 5359, y explique a qué se deben las diferencias.

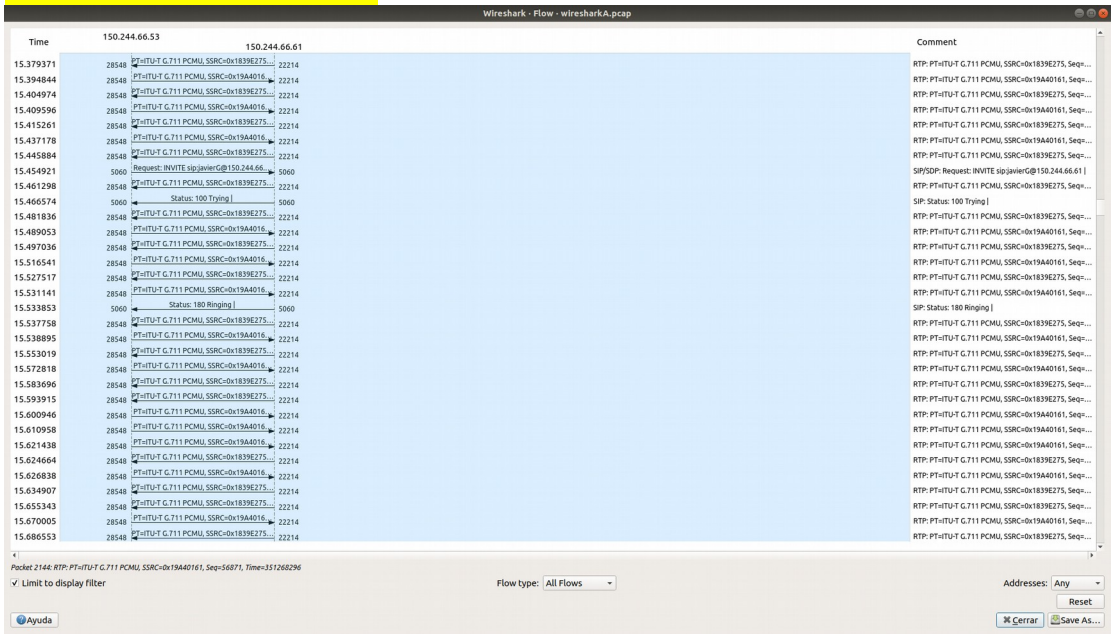
Incluya aquí los diagramas del caso 4.2
Señalización y multimedia desde el usuario A al inicio entre A y C:



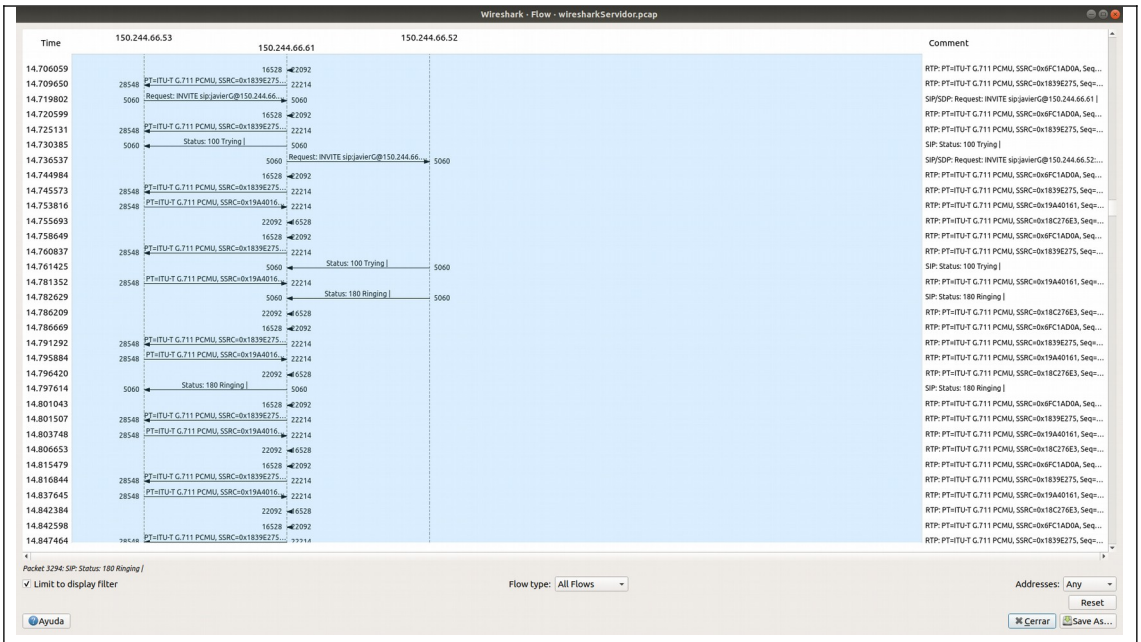
Señalización y multimedia desde el usuario C al inicio entre A y C:



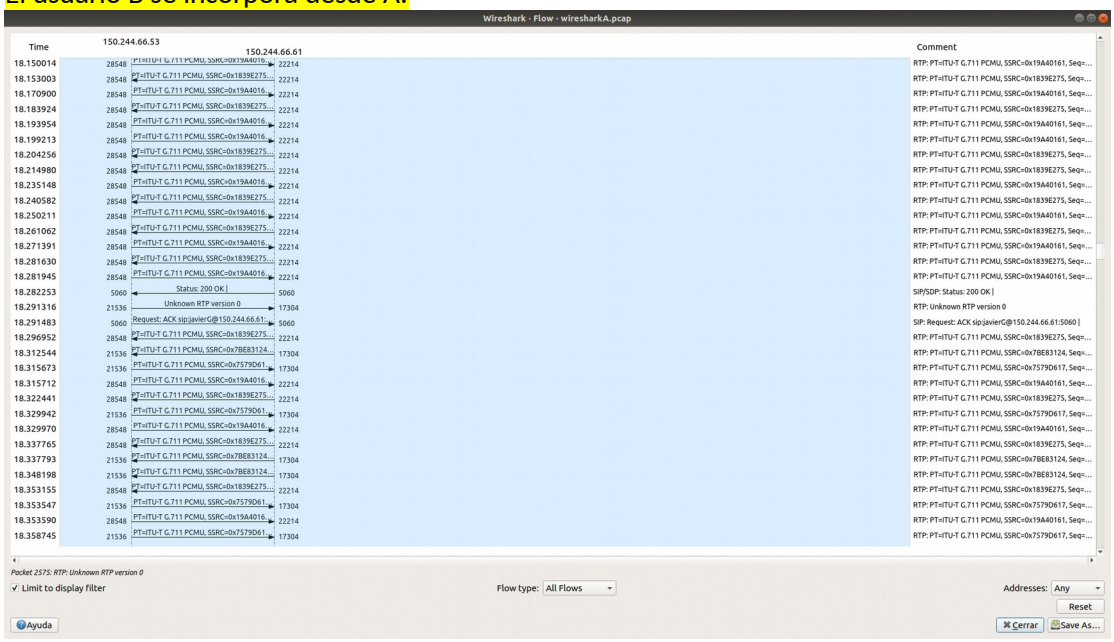
Llamando al usuario B desde A:



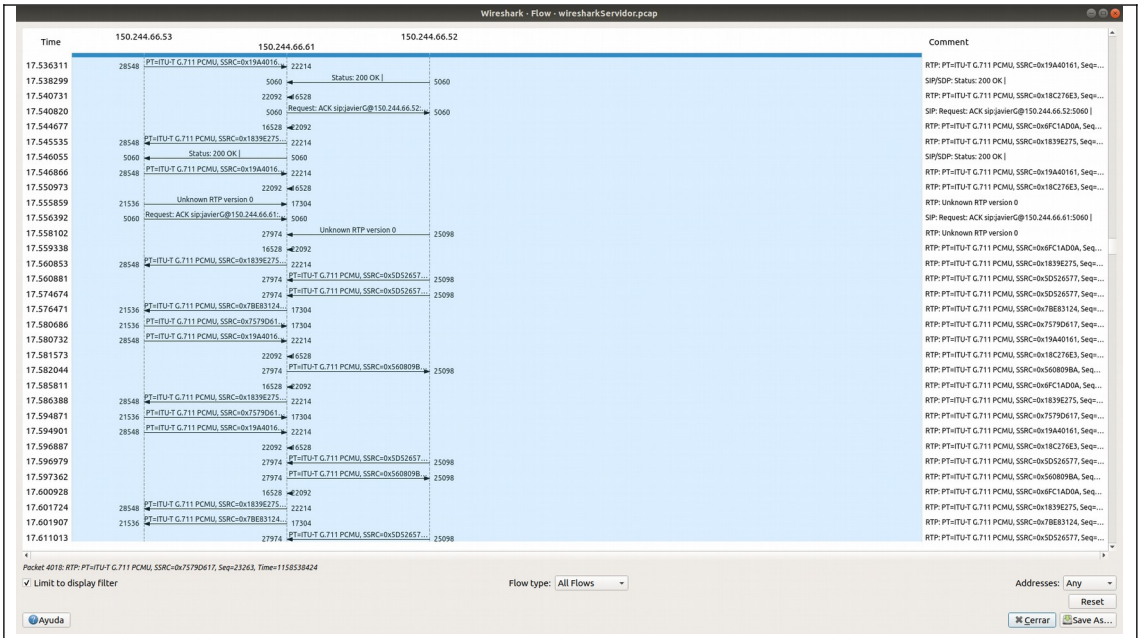
Llamando al usuario B desde C:



El usuario B se incorpora desde A:



El usuario B se incorpora desde C:



Indique aquí a qué se deben las diferencias entre el comportamiento observado y lo especificado en la RFC 5359.

A diferencia del RFC 5359 el usuario A no envía un paquete RE-INVITE al usuario C para cambiar el Contact URI y así actuar como foco de la llamada. En su lugar, el usuario A mete a la llamada al usuario C sin cambiar ningún parámetro de la llamada.

5. A continuación analizaremos algunas opciones de rutado de llamadas. El rutado de llamadas es una funcionalidad que realiza internamente el servidor SIP y que permite, por ejemplo, asociar una extensión corta a un usuario para que al llamarle sea suficiente con escribir la extensión corta en vez de toda la URI SIP. Para ello se realizarán las siguientes tareas:
   5.1. Se asociará a cada uno de los dos usuarios creados anteriormente una extensión corta (01 al primero y 02 al segundo). Para ello se podrá usar o bien la configuración de rutado usando expresiones regulares (regexroute.conf) o usando javascript (javascript.conf). Adjunte la expresión regular o el fichero javascript usado para rutar la llamada.

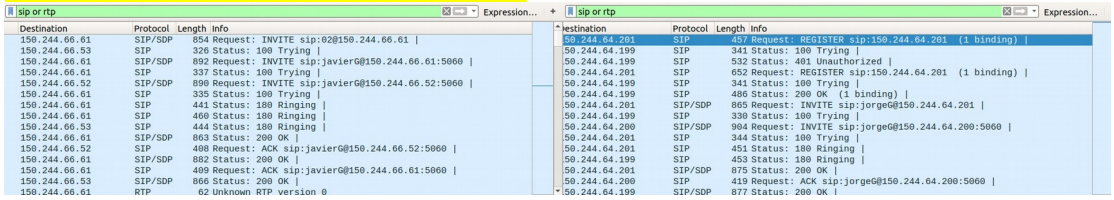
Indique aquí los pasos dados para configurar las extensiones cortas.

En el fichero regexroute.conf se han añadido las siguientes líneas:

^01$=sip/sip:jorgeG@150.244.66.61:5060
^02$=sip/sip:javierG@150.244.66.61:5060

   5.2. Realice una llamada de prueba entre los dos usuarios mientras captura el tráfico con Wireshark. Para que la captura se obtenga correctamente, realice la llamada desde el cliente que **no** está en el mismo ordenador que el servidor de Yate. Explique qué cambia en los paquetes SIP intercambiados con respecto al apartado 3.

Explique aquí en qué cambian los paquetes.



Como se aprecia en la imagen, al utilizar la extensión corta la cabecera del paquete INVITE tiene el formato **INVITE sip:[extension corta]@IP**. En el caso de que no utilizemos la extensión corta, la cabecera del paquete INVITE tiene el formato: **INVITE sip:[nombre del usuario]@IP**.

A continuación vamos a probar la interoperabilidad de elementos SIP. Para eso se utilizará otro *softphone*. Se puede descargar Ekiga de www.ekiga.org

6. La pareja creará dos cuentas en . Anote la información de las cuentas (usuario, contraseña, dirección del *proxy SIP* y puerto). Verifique que la instalación de Ekiga es correcta realizando una llamada de prueba. En caso de que el servicio de Ekiga sea deficiente, puede probar con algún otro proveedor de VoIP. Revise para ello el siguiente enlace: http://www.voip-info.org/wiki/view/Free+VoIP+Networks

Indique aquí los pasos dados para configurar los usuarios de Ekiga u otro proveedor de VoIP.

Dentro de Ekiga, en la opción para configurar las cuentas tenemos dos opciones, una que ya viene preconfigurada en la que se deben introducir los datos de la cuenta Ekiga, y otra opción llamada “Cuenta SIP” donde se pueden configurar los parámetros de una cuenta SIP genérica indicando el servidor. En nuestro caso como no ha sido posible registrarnos en Ekiga, lo hemos hecho con una cuenta Linphone mediante la opción “Cuenta SIP”.

7. A continuación se procederá a integrar el servidor Yate local con el servidor remoto de Ekiga. Para eso vamos a operar de la siguiente manera: vamos a configurar Yate para que pueda actuar como un cliente SIP frente al servidor remoto de Ekiga. Esto además permitirá a Yate rutar llamadas por la línea que establece entre Yate y el servidor remoto de Ekiga:
   7.1. Configure la conectividad entre el servidor Yate y el servidor remoto de Ekiga. Para eso use la información de autentificación, IP y puerto del primero de los usuarios creados en Ekiga. Use el fichero de configuración accfile.conf de Yate. Para verificar que la autenticación funciona correctamente se debe observar con Wireshark los mensajes que se intercambian el servidor Yate y el servidor de Ekiga. Comente dichos mensajes.

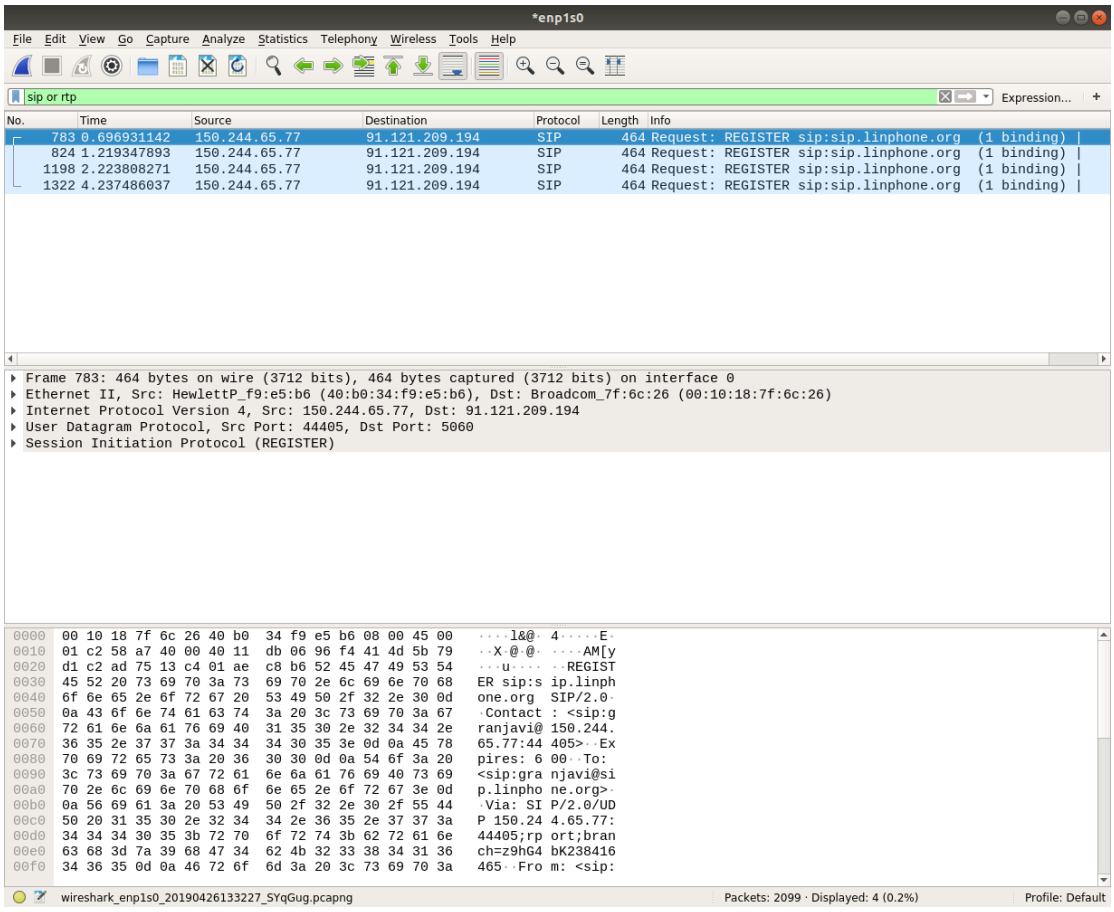
Indique aquí los pasos dados para proporcionar la conectividad con el servidor remoto Ekiga desde el servidor Yate.

Se ha añadido el siguiente código al fichero accfile.conf:

[sip_linphone_javi]
enabled=yes
protocol=sip
username=granjavi
description=Javi SIP account
;interval=600
authname= granjavi@sip.linphone.org
password=1234
domain= sip.linphone.org
registrar= sip.linphone.org
;outbound=10.0.0.1:5061
;localaddress=192.168.0.1:5062

Comente los mensajes que intercambian ambos servidores durante la autenticación.

Hemos realizado una prueba de conexión en la red de la UAM y obtuvimos el siguiente resultado:



Como se puede observar, el cliente Yate envía mensajes REGISTER al servidor de linphone pero no recibe respuesta. Esto es debido a que el firewall de la UAM está tirando los paquetes entrantes de respuesta o los paquetes salientes de REGISTER.

Al igual que desde la UAM desde nuestras casas se obtuvo el mismo resultado.

De forma teórica, tomando como referencia el escenario de estar en los laboratorios de la UAM y que nos estamos registrando con una cuenta de Ekiga, para registrar el cliente Ekiga manda una petición REGISTER sin autentificación, que será respondida con un mensaje de error 401 informándole de los parámetros de autentificación.

Cuando Ekiga ha recibido esta información vuelve a mandar el mensaje REGISTER con la autentificación y recibe la confirmación del servidor Yate. Una vez obtenido la autentificación, el cliente Ekiga envía un mensaje SUBSCRIBE donde en el campo Event le indica a qué tipo de eventos se quiere subscribir. Inmediatamente el servidor Yate responde con un mensaje de confirmación y con un mensaje NOTIFY que incluye la información que el cliente ha subscrito.

Después, cuando el cliente ya ha recibido la información solicitada responde al servidor Yate con un mensaje de confirmación. Finalmente, como Ekiga no puede desactivar la mensajería instantánea, hace uso del mensaje PUBLISH, pero como Yate no implementa un servidor de mensajería instantánea, la respuesta es un mensaje de error 501.

7.2. Configure una nueva extensión saliente (03) que se rute como llamada saliente a través de la línea SIP configurada anteriormente al segundo usuario creado en Ekiga.

Indique aquí los pasos dados para proporcionar la conectividad con el servidor remoto Ekiga desde el servidor Yate desde la extensión 03.

[sip_linphone_jorge]
enabled=yes
protocol=sip
username=georgegd
description=Jorge SIP account
;interval=600
authname= georgegd@sip.linphone.org
password=1234
domain= sip.linphone.org
registrar= sip.linphone.org
;outbound=10.0.0.1:5061
;localaddress=192.168.0.1:5062

En el fichero regexroute.conf se añade la siguiente extensión:
^03$=line/georgegd;line=linphone

7.3. Arranque un Ekiga en un ordenador diferente del de Yate y configúrelo para usar el segundo usuario creado en el servidor de Ekiga.

Indique aquí los pasos dados para configurar el cliente de Ekiga.

Tal y como se ha comentado en el apartado 7.1, en la opción de Ekiga para añadir las cuentas SIP, hemos introducido los datos de usuario, contraseña y servidor proporcionados

por Linphone y que son los mismos que los configurados en Yate en username, password y domain, respectivamente.

7.4. Realice los siguientes experimentos y comente qué ocurre:

7.4.1. Realice una llamada desde el *softphone* Yate a la extensión creada anteriormente (03) y verifique que la llamada llega al *softphone* de Ekiga.

Incluya aquí el diagrama de secuencia para la llamada, incluyendo todos los sistemas posibles. No ha sido posible realizar la prueba de la forma indicada por los bloqueos de puertos de la escuela y de nuestras casas.
Indique el contenido del mensaje SDP y los tipos de Codecs ofrecidos por los agentes de usuario en aquellos casos en que se presente esta información, indicando en qué casos se presenta. No hemos podido probarlo
Indique el tipo de códec utilizado y la tasa binaria en aquellos casos en que se presente esta información, indicando en qué casos se presenta. No hemos podido probarlo
Indique las direcciones IP y puertos de origen y destino de los flujos RTP en aquellos casos en que se presente esta información, indicando en qué casos se presenta. No hemos podido probarlo
Indique el tipo de códec utilizado y la tasa binaria en aquellos casos en que se presente esta información, indicando en qué casos se presenta. No hemos podido probarlo

7.4.2. Realice la llamada inversa: llame desde el *softphone* Ekiga al primer usuario creado en el servidor de Ekiga y verifique que llega al *softphone* Yate.

Incluya aquí el diagrama de secuencia para la llamada, incluyendo todos los sistemas posibles. No ha sido posible realizar la prueba de la forma indicada por los bloqueos de puertos de la escuela y de nuestras casas.
Indique el contenido del mensaje SDP y los tipos de Codecs ofrecidos por los agentes de usuario en aquellos casos en que se presente esta información, indicando en qué casos se presenta. No hemos podido probarlo
Indique el tipo de códec utilizado y la tasa binaria en aquellos casos en que se presente esta información, indicando en qué casos se presenta. No hemos podido probarlo
Indique las direcciones IP y puertos de origen y destino de los flujos RTP en aquellos casos

en que se presente esta información, indicando en qué casos se presenta. No hemos podido probarlo
Indique el tipo de códec utilizado y la tasa binaria en aquellos casos en que se presente esta información, indicando en qué casos se presenta. No hemos podido probarlo
¿Observa alguna diferencia significativa frente al caso anterior? No hemos podido probarlo

7.5. Analice la calidad de la comunicación, para eso procederá de la siguiente manera:

7.5.1. En Ekiga configure como única codificación de audio soportada la G.711.

7.5.2. Realice una llamada como en el punto anterior entre el *softphone* Yate y la extensión 03 realizando una captura con Wireshark.

7.5.3. Con la captura del tráfico SIP y RTP intercambiado, use las herramientas de Wireshark para reproducir la llamada.

7.5.4. Por último, usando también las herramientas de Wireshark consiga una estimación del jitter y la tasa de pérdida de paquetes de la llamada realizada. ¿En qué se diferencia la estimación del jitter de la que se realizaba en la práctica anterior?

Incluya aquí el diagrama de secuencia para la llamada, incluyendo todos los sistemas posibles. No hemos podido probarlo
Incluya los pasos realizados para poder reproducir la llamada. Seleccionamos la opción Telephony < VoIP Calls. En la ventana que aparecerá muestra las conversaciones que han tenido lugar. Pinchamos sobre la conversación que nos interesa y seleccionamos **Play Streams.** En la siguiente ventana que aparece muestra los paquetes con los audios que se pueden reproducir de la conversación.
Incluya los pasos realizados para poder estimar la calidad de la llamada, y los resultados obtenidos. No hemos podido probarlo
Incluya en qué se diferencia la estimación del jitter que hace Wireshark de la que realizada en la práctica 2. No hemos podido probarlo

3 Conclusiones

Tras haber realizado la práctica, hemos aprendido los distintos tipos de mensajes que se transmiten a la hora de establecer una llamada telefónica entre dos usuarios a través de Internet utilizando el protocolo SIP. Hay que destacar los comandos INVITE, OK, ACK y BYE. Algo que hemos aprendido en esta práctica y que en teoría no se había visto es el hecho de la autentificación y cómo hacen dos peticiones como medida de seguridad, recibiendo un error en la primera y un OK en la segunda en caso de que las credenciales sean correctas.

Tras ello, hemos comprobado la interoperabilidad del protocolo SIP entre distintos clientes y servidores, lo que ha logrado que para las llamadas IP sea algo universal, sencillo y con poco consumo de recursos de red.