

Pon un poco de FOSS en tu VoIP (por favor)

José Antonio Montes

ESNOG-32
UPC Barcelona 24/10/2024

Introducción y contexto

- Hoy vamos a hablar de VoIP y de FOSS.
- Cuando hablemos de VoIP nos referiremos a las llamadas de voz transportadas a través de las redes IP.
- Cuando hablemos de FOSS nos referiremos a los proyectos de Software Libre aplicables a la VoIP.
- Estos proyectos de Software Libre no son nada nuevo sobre el horizonte.
- La mayoría tienen bastante solera y son utilizados ampliamente en las redes de los operadores fijos y móviles de todo el mundo.

¿Por qué utilizar FOSS en tu VoIP?

Utilizar proyectos de Software Libre en entornos de producción tiene sus pros y sus contras.

- Ventajas:
 - El código fuente es accesible y modificable.
 - Hay soporte de la comunidad y también comercial.
 - Si el proyecto tiene actividad y madurez estará al día en las últimas tendencias.
 - El proyecto suele estar bien documentado.
 - Se puede hacer un seguimiento de los cambios.
 - Tu también puedes formar parte de la comunidad.
- Inconvenientes:
 - El soporte de la comunidad es voluntario y depende de la disponibilidad de los mantenedores.
 - El proyecto puede quedar abandonado o sufrir forks.
 - Requiere tiempo y esfuerzo para aprender a usarlo.

Vale, pero... ¿Por qué en tu VoIP?

- A veces es la única solución disponible.
- Otras veces es la mejor solución del momento.
- Los proveedores no llegan a tiempo con las últimas tendencias o soluciones a problemas reportados.
- No existe un proveedor que pueda darte todo.
- Te permite:
 - Tener un control sobre tu red muy difícil de conseguir de otra forma.
 - Diseñar servicios a la carta.
 - Solucionar problemas espontáneos.
 - Escalar bajo demanda.
- Proporciona estabilidad (a tu red) y paz mental.
- A veces existe una solución comercial, pero no está a tu alcance.

Hablemos (un poco) de SIP

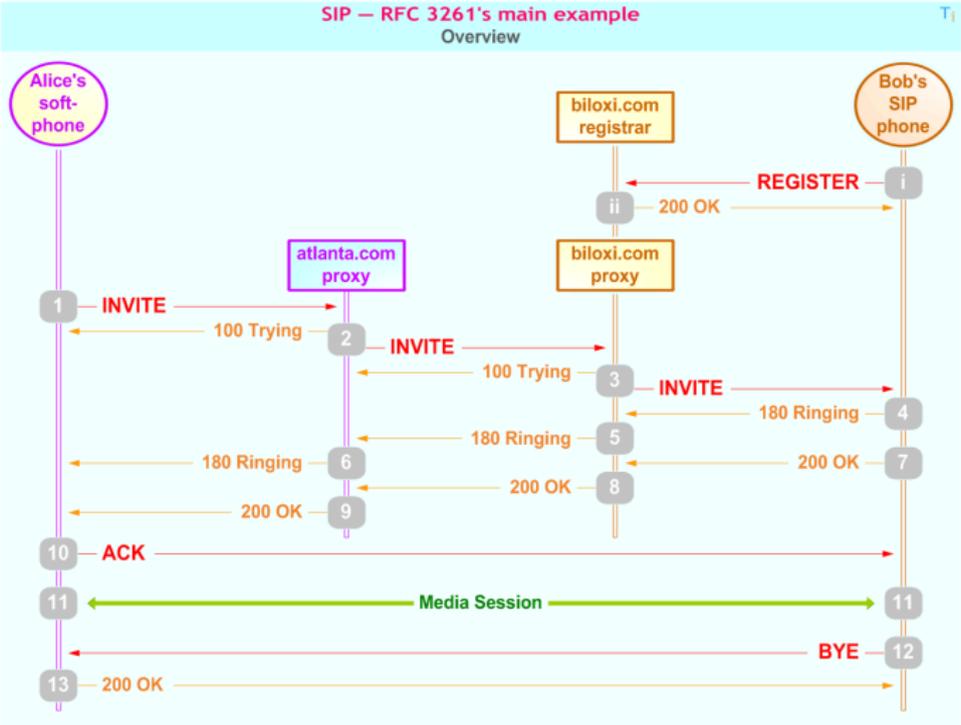
¿Qué es SIP?

- SIP es **The Session Initiation Protocol**.
- Es un protocolo de señalización diseñado para iniciar, mantener y terminar sesiones de media.
- Diseñado en el IETF (Network WG) y adoptado por el 3GPP para la red IMS.
- La primera versión fue publicada en 1999, la segunda (actual) en 2002.
- La RFC principal es la 3261 ¹ con sus extensiones ².
- Usa UDP, TCP, TLS, SCTP o WSS como transporte.
- Está basado en texto, toma ideas del HTTP y SMTP.
- Modelo petición/respuesta alternado los roles.
- Utiliza el SDP para describir las sesiones.

¹ <https://datatracker.ietf.org/doc/html/rfc3261>

² <https://datatracker.ietf.org/doc/search/?name=SIP&rfcs=on&sort=date>

¿Qué es SIP?



<https://www.tech-invite.com/y-sip/tinv-rfc-SIP-mainex.html>

¿Qué es SIP?

Petición

```
INVITE sip:bob@biloxi.com SIP/2.0  
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776asdhd5  
Max-Forwards: 70  
To: Bob <sip:bob@biloxi.com>  
From: Alice <sip:alice@atlanta.com>;tag=1928301774  
Call-ID: a84b4c76e66710@pc33.atlanta.com  
CSeq: 314159 INVITE  
Contact: <sip:alice@pc33.atlanta.com>  
Content-Type: application/sdp  
Content-Length: 264
```

```
v=0  
o=- 0 0 IN IP4 192.0.2.1  
s=-  
c=IN IP4 192.0.2.1  
t=0 0  
m=audio 20138 RTP/AVP 8 0 18 96  
a=rtpmap:8 PCMA/8000  
a=rtpmap:0 PCMU/8000  
a=rtpmap:18 G729/8000  
a=rtpmap:96 telephone-event/8000  
a=sendrecv  
a=maxptime:20  
a=ptime:20
```

Respuesta

```
SIP/2.0 200 OK  
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776asdhd5  
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf  
From: Alice <sip:alice@atlanta.com>;tag=1928301774  
Call-ID: a84b4c76e66710@pc33.atlanta.com  
CSeq: 314159 INVITE  
Contact: <sip:bob@192.0.2.4>  
Content-Type: application/sdp  
Content-Length: 302
```

```
v=0  
o=- 0 2 IN IP4 192.0.2.4  
s=-  
c=IN IP4 192.0.2.4  
t=0 0  
m=audio 20140 RTP/AVP 0 8 18 96  
a=rtpmap:0 PCMU/8000  
a=rtpmap:8 PCMA/8000  
a=rtpmap:18 G729/8000  
a=fmtp:18 annexb=no  
a=rtpmap:96 telephone-event/8000  
a=fmtp:96 0-16  
a=sendrecv  
a=ptime:20  
a=maxptime:150
```

← primera línea →

cabeceras

← cuerpo (SDP) →

← esta línea falta en la petición (espóiler)

Hablemos de Kamailio

¿Qué es Kamailio?

Un poco de historia



- Es un **SIP** (signaling) **router**.
- Es un proyecto con bastante madurez (>20 años).
- Iniciado en la Fraunhofer FOKUS Research Institute como SER (SIP Express Router) en 2001.
- En 2002 el código fue liberado bajo licencia GPLv2.
- En 2005 se produjo el fork a OpenSER.
- En 2008 OpenSER cambió el nombre a Kamailio.
- Poco después SER y Kamailio se reunificaron ¹.
- Liderado por Daniel-Constantin Mierla (@miconda).

¹<https://web.archive.org/web/20180429092231/http://sip-router.org/history/>

¿Qué es Kamailio?



- No es una PBX (ni un B2BUA ¹) y no procesa media.
- Es un **SIP router** programable *explicitamente*.
- Trabaja a nivel de paquetes (mensajes) SIP.
- Mas de 250 módulos, muchas funcionalidades ².
- Es **muy** estable, eficiente y escalable.
- Soporta miles de transacciones SIP por segundo.
- Altamente portable y con muy pocas dependencias.
- Pieza clave en las redes VoIP de los operadores grandes y pequeños donde se utiliza.

¹ https://en.wikipedia.org/wiki/Back-to-back_user_agent

² <https://www.kamailio.org/docs/modules/stable/>

¿Qué es Kamailio?



- No es una PBX (ni un B2BUA ¹) y no procesa media.
- Es un **SIP router** programable *explicitamente*.
- Trabaja a nivel de paquetes (mensajes) SIP.
- Mas de 250 módulos, muchas funcionalidades ².
- Es **muy** estable, eficiente y escalable.
- Soporta miles de transacciones SIP por segundo.
- Altamente portable y con muy pocas dependencias.
- Pieza clave en las redes VoIP de los operadores grandes y pequeños donde se utiliza.
- Requiere una curva especial de aprendizaje.
- Una receta del gremio: deberías de aprender *"tres partes de SIP por cada parte de Kamailio"*

¹ https://en.wikipedia.org/wiki/Back-to-back_user_agent

² <https://www.kamailio.org/docs/modules/stable/>

¿Cómo se utiliza Kamailio?

En una red VoIP

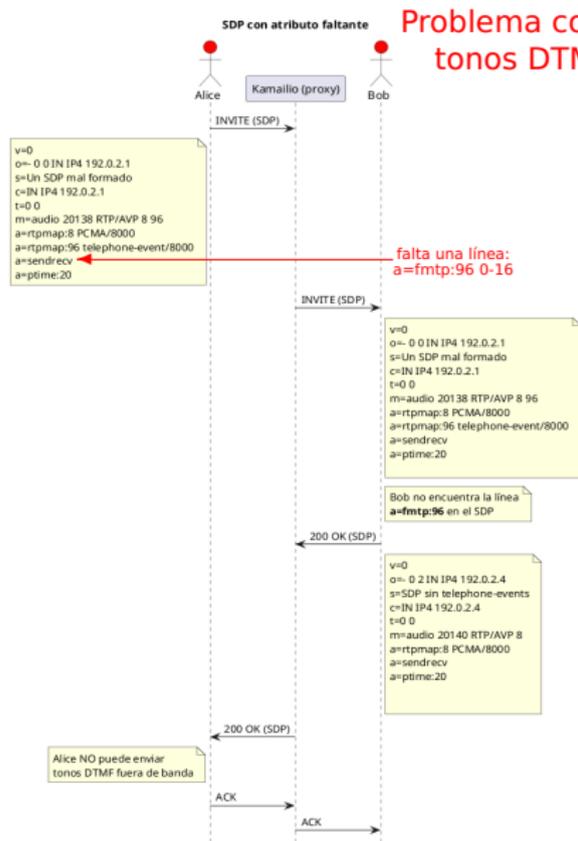


- Kamailio se utiliza principalmente como:
 - Proxy
 - Registrar
 - Load Balancer
 - Redirect Server
 - Application Server
 - Base para diseñar servicios VoIP a la carta.

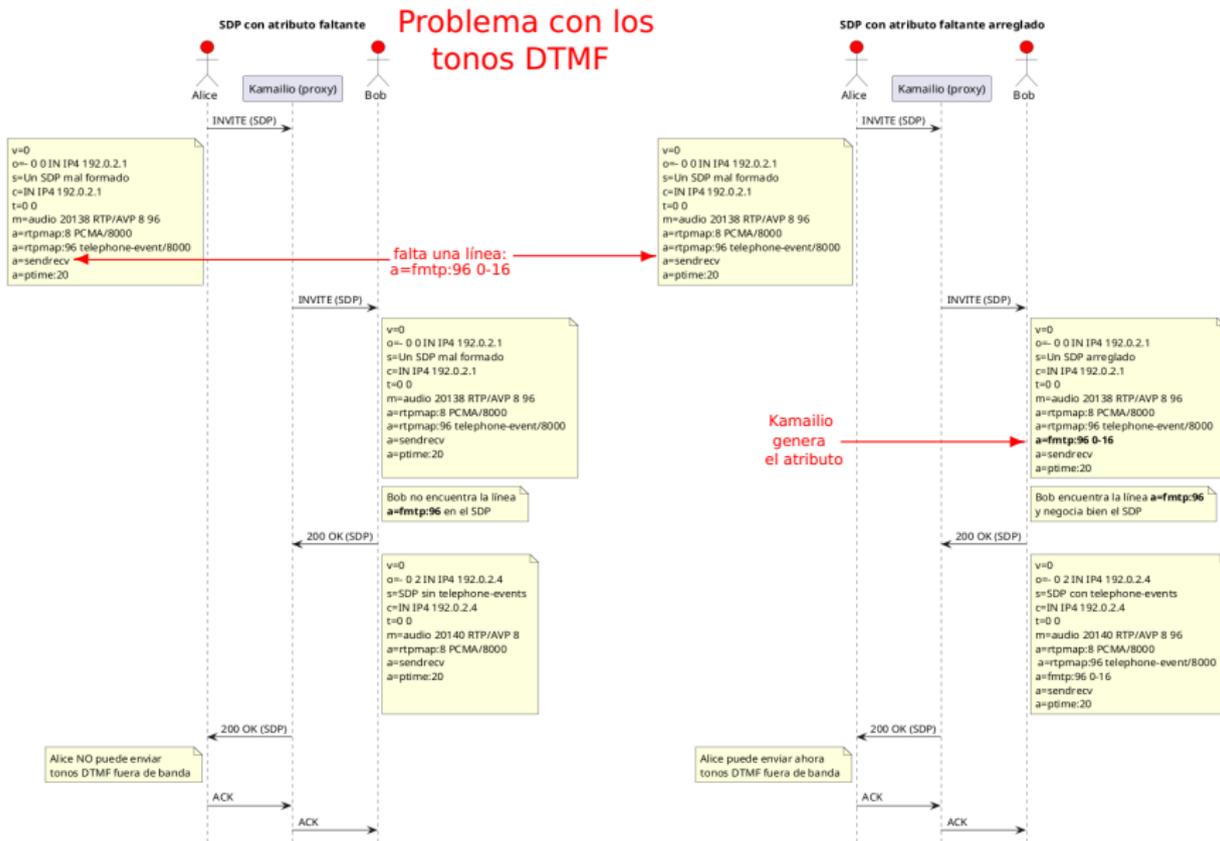
- Kamailio no se utiliza como:
 - Softphone
 - B2BUA
 - PBX
 - Media Server

Hora de Aventuras

Aventuras bizarras con tu proxy (I)

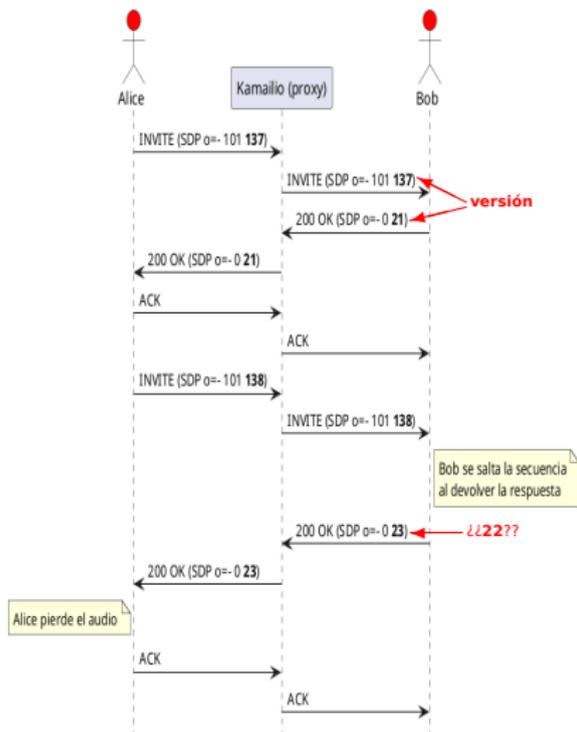


Aventuras bizarras con tu proxy (I)



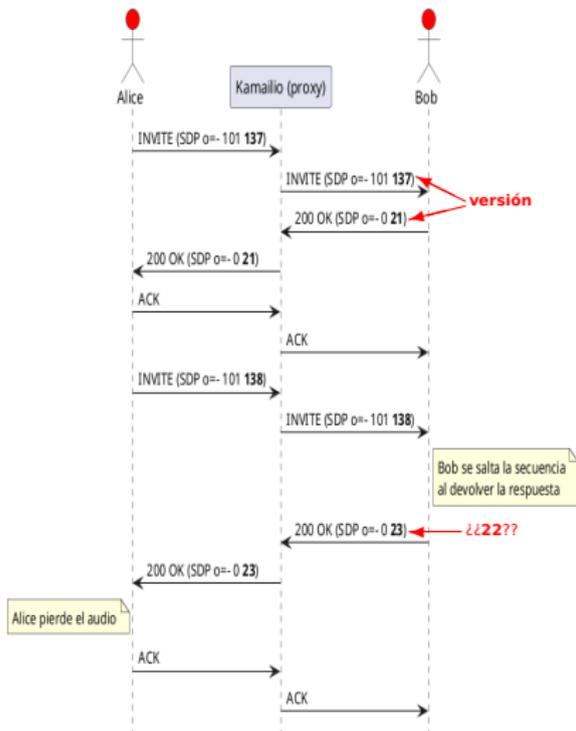
Aventuras bizarras con tu proxy (II)

SDP con número de versión incorrecto

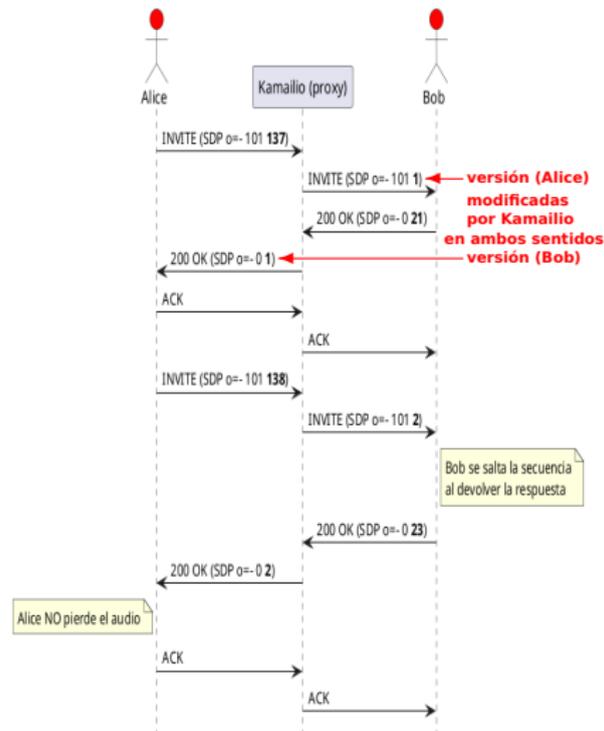


Aventuras bizarras con tu proxy (II)

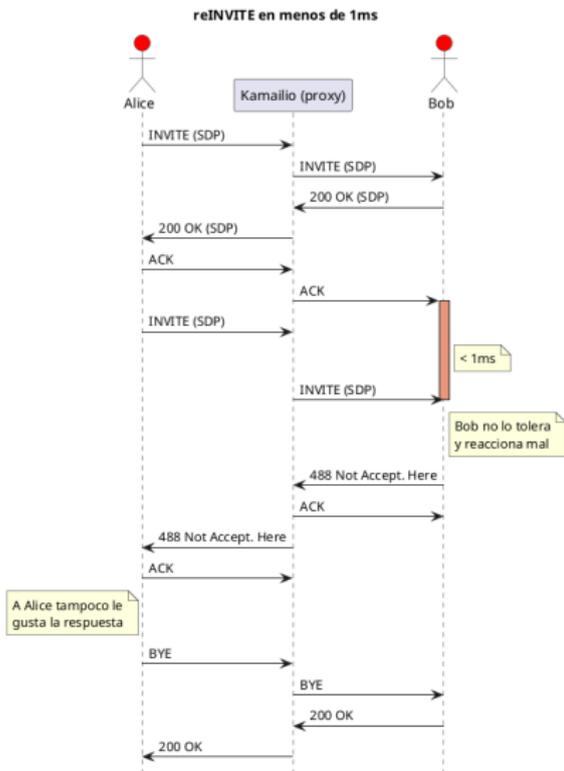
SDP con número de versión incorrecto



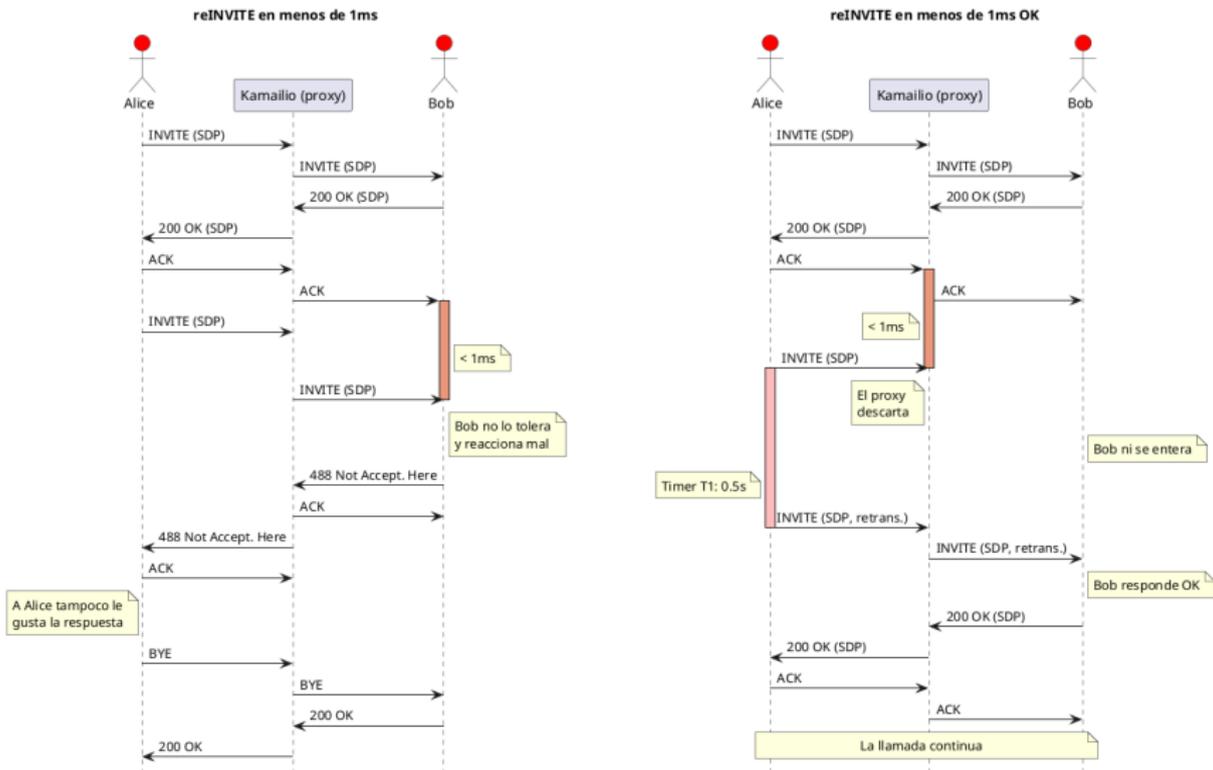
SDP con número de versión OK



Aventuras bizarras con tu proxy (III)



Aventuras bizarras con tu proxy (III)



Algunos consejos básicos

para que escale tu Kamailio



- Elige bien tus estructuras de datos (htable ymtree son tus mejores aliados).
- Mantén tu lógica sencilla (aplica el KISS).
- Divide los Kamailios por funciones específicas.
- Si tienes que usar una base de datos:
 - Monta la base de datos en local.
 - Usa la replicación para los datos de solo lectura.
 - Usa tablas en memoria para los datos temporales.
 - Diseña tus tablas para obtener una fila por consulta.
- Evita crear transacciones donde sea posible.
- (Ab)usa de la memoria y evita el disco duro.
- Los cuellos de botella son la I/O y las consultas síncronas externas.

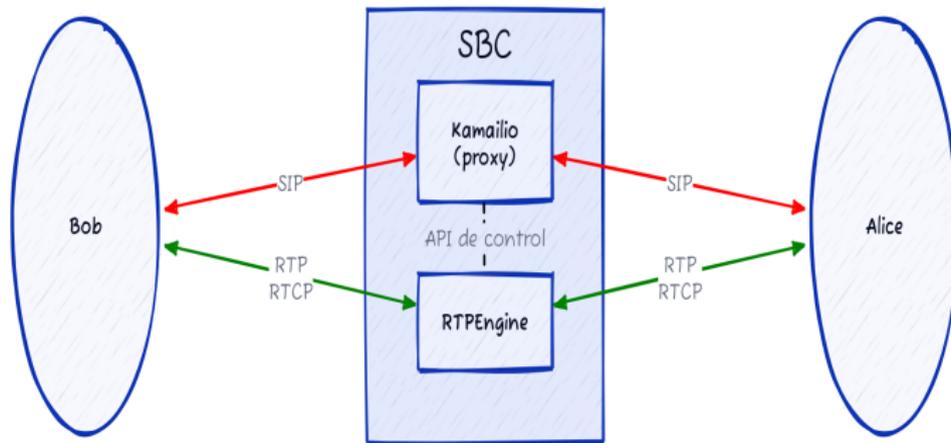
Hablemos de RTPEngine

¿Qué es RTPEngine?

- Es un media server robusto y muy completo.
- Hace de proxy para los paquetes RTP y RTCP.
- Es el complemento ideal para Kamailio.
- Gestiona el media de las llamadas mediante parámetros a través de un interfaz de control.
- Procesa y re-escibe el SDP en las negociaciones.
- Desarrollado por Sipwise bajo licencia GPLv3.
- Soporta transcoding y repaquetización del media.
- Realiza la conmutación del RTP en el kernel (Linux).
- Capaz de gestionar miles de llamadas por servidor.
- La lista de funcionalidades no para de crecer ¹.

¹<https://github.com/sipwise/rtpengine>

¿Como se utiliza RTPEngine?



Ejemplo de uso de RTPEngine con Kamailio

Hablemos de SEMS

¿Qué es SEMS?

Un poco de historia



- Es el SIP Express Media Server.
- Iniciado en la Fraunhofer FOKUS en 2002.
- El código está liberado bajo licencia GPLv2+.
- El complemento a SER para gestionar el media.
- Desarrollado inicialmente por Raphael Coëffic.
- Mantenido por iptel.org, IPTEGO y Frafos.
- Se distribuye también bajo licencia comercial.
- Stefan Sayer (Frafos) impulsa el proyecto.
- Frafos desarrolla su solución comercial ABC SBC.
- Desde la v1.6.0 SEMS es solo mantenido por la comunidad ¹.

¹<https://github.com/sems-server/sems>

¿Qué hace SEMS?

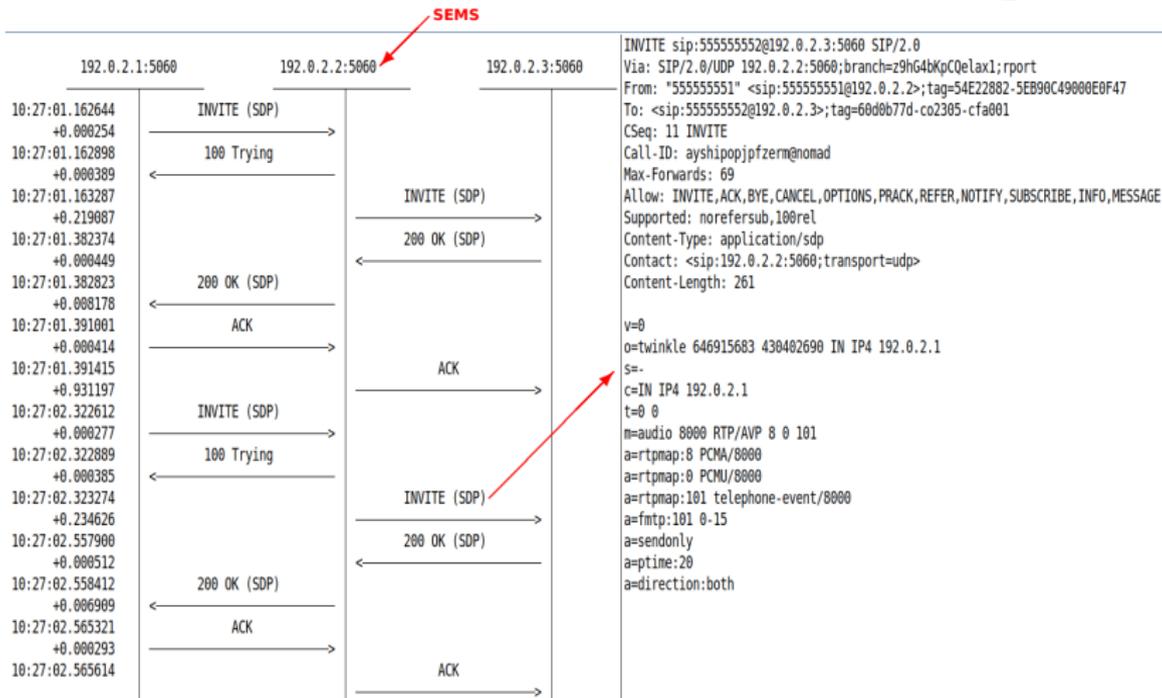


- SEMS está diseñado como servidor de aplicaciones.
- Tiene varias aplicaciones predefinidas en módulos.
- Es **muy** estable, escalable y eficiente en recursos.
- Las aplicaciones se pueden programar mediante C++, Python, o DSM (lenguaje nativo).
- La aplicación de **SBC**¹ es muy interesante²:
 - Hace transcoding del media y filtrado de codecs.
 - Gestiona el NAT, los SST, registros (caching).
 - Define el comportamiento por perfiles.
 - Tiene módulos para el control de las llamadas.
 - Filtra, modifica o añade cabeceras usando variables.
 - Usa los nombres de variables de SER (kamailio).
 - Los perfiles se pueden recargar en caliente.

¹ Session Border Controller: <https://datatracker.ietf.org/doc/html/rfc5853>

² <https://www.kamailio.org/events/2014-KamailioWorld/day0/w4-Stefan.Sayer-SEMS-Workshop.pdf>

SEMS es ligero



Ejemplo de SEMS como B2BUA (SBC)

SEMS está activo en varios proyectos Open Source ¹:

- El proyecto original mantenido por la comunidad:
<https://github.com/sems-server/sems>
- El fork mantenido por Sipwise para el B2BUA de su solución sip:provider CE:
<https://github.com/sipwise/sems>
- El fork mantenido por el proyecto Yeti para el SBC de su Yeti-Switch ²:
<https://github.com/yeti-switch/sems>

¹ El ABC SBC de Frafos basado en SEMS es un producto comercial.

² <https://yeti-switch.org>

¿Preguntas?

iGracias!