Trabajo Práctico N°2

En grupos (dentro de lo posible) de dos o tres integrantes, desarrollen las siguientes consignas:

a) Elijan un **protocolo de la capa de aplicación** (por ejemplo: http, ftp, smtp, etc).
b) Realicen un trabajo de **investigación** (4 puntos) e **implementación** (3 puntos) del protocolo elegido. Tengan en cuenta que lo anterior debe estar subido a un repositorio en github.
c) Preparen una **presentación** (3 puntos), a modo de clase, de lo trabajado en el punto anterior para todo el curso.

Es condición para aprobar el trabajo práctico haber hecho todos los puntos a) b) y c). No olviden de especificar toda la bibliografía consultada.

Fechas de entrega:

- a) y b) hasta el 29/5 inclusive.
- Presentaciones de c): 5/6, 9/6 y 12/6.

Investigación:

Reseña, informe técnico, documentación.
Lectura de textos.
lista de preguntas básicas generales (qué es, cómo funciona, qué problema soluciona, historia, cómo se aplica)

Real-Time Transport Protocol
Investigación

Real-Time Transport Protocol (RTP) es un protocolo de la capa de aplicación normalmente utilizado para la transmisión de paquetes multimedia en tiempo real. Se utiliza, precisamente, para videoconferencias, transmisiones de video/audio individualmente, telefonía, servicios de televisión y servicios de streaming en tiempo real, entre otros.

Funcionamiento:
RTP suele trabajar sobre User Datagram Protocol (UDP) , junto a Real-Time Control Protocol (RTCP) y Session Initiation Protocol (SIP). Al tratarse de una conexión en tiempo real, los paquetes son enviados desde el host hasta el destino sin establecer previamente la conexión y sin validar la transmisión correcta de los datagramas. Por este motivo, los mismos pueden perderse o llegar en desorden.

Para dar un poco de orden a este tipo de conexión, los paquetes enviados mediante RTP se emiten con una cabecera que contiene información adicional, como puede ser un número de secuencia, que es un valor entero que se incrementa por cada paquete enviado, y una marca de tiempo, que indica el momento exacto en el que se envía el paquete. De esta manera, se tiene una constancia sobre los paquetes enviados y su respectivo orden. También en la cabecera se incluye información sobre el tipo de payload (carga útil), que permite al receptor identificar el formato de codificación. El payload es el conjunto de información transmitido por RTP en un paquete. Este puede ser de audio o video, y para cada uno hay diferentes tipos de codificación.

Para garantizar cierta calidad de servicio, RTP funciona en conjunto con RTCP (Real-Time Control Protocol) de manera paralela. RTCP se encarga de supervisar las estadísticas de transmisión, perfeccionar la sincronización de múltiples flujos, garantizando así QoS.

RTP es comúnmente utilizado con SIP (Session Initiation Protocol), otro protocolo que permite y gestiona las conexiones establecidas entre los usuarios.

Breve explicación de conexión peer to peer utilizando SIP, UDP, RTP y RTCP:

En una conexión P2P utilizando SIP, UDP, RTP y RTCP, los participantes establecen una sesión de comunicación utilizando SIP. Uno de ellos debe enviar una solicitud denominada INVITE, que luego será aceptada por el destinatario para consolidar la conexión. Una vez establecida la sesión, los datos de audio y video se transmiten entre los participantes utilizando RTP sobre UDP. RTCP se utiliza para el monitoreo y control de la calidad de la transmisión durante la sesión.

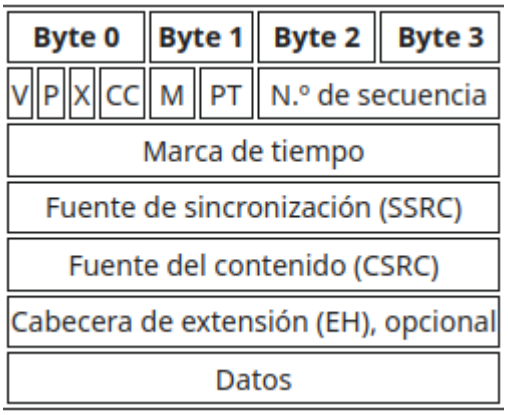
Entonces, ¿cómo es que interviene exactamente RTP en esta comunicación?

Vamos a poner como ejemplo una conferencia de audio y video, en la cual pueden participar múltiples usuarios. Cada canal de información se transmite mediante una sesión de RTP independiente, cada una con su puerto UDP respectivamente. Esta se conforma de todas las fuentes de un mismo tipo que formen parte de la conferencia. Cada paquete de información contiene alrededor de 20 ms de

contenido. Dependiendo de cuantos milisegundos se opte por transmitir, mas rapida o lenta tendrá que ser la conexión. Cuantos menos milisegundos, más ancho de banda se requerirá, y viceversa.

Uno de los beneficios por los que el audio y el video se transmiten por sesiones diferentes es que permite al destinatario/s de la conferencia recibir solo un medio, si así lo desea (ej. desactivar las cámaras del resto de usuarios en una sesión de meet, o tu propia cámara también).

Como ya hemos mencionado, cada paquete transmitido por RTP contiene una cabecera o "header".



Cabecera de paquete RTP

En síntesis, esta cabecera contiene datos fundamentales como la versión de RTC (V), la marca de tiempo, el número de secuencia, los datos en sí, un bit de relleno para identificar el tipo de cifrado (P), si es que tiene, y el tipo de carga útil (PT). Otros de los datos más importantes son el SSRC y el CSRC. El SSRC es un identificador único que permite al receptor agrupar los paquetes correctamente para su reproducción. Cada sesión RTP contiene un SSRC único que persiste durante toda la duración de la misma. El CSRC es una lista que contiene todos los contribuyentes de paquetes RTP al SSRC específico. En adición, hay un contador que marca la cantidad total de fuentes que conforman al CSRC (CC). El bit de extensión (R), de estar activado, indica que hay otro paquete opcional que funciona como extensión (EH).

RTP Mixer: Sistema intermediario que recibe paquetes RTP de una o más fuentes, que puede o no cambiar el formato de lo recibido. Dado que estas fuentes generalmente no están sincronizadas y se superponen, el mixer hace ajustes y genera una mezcla sincronizada. Un RTP Mixer puede funcionar como una fuente de sincronización.

Seguridad:

Se espera que los protocolos de las demás capas se encarguen de garantizar la seguridad para las aplicaciones RTP, sin embargo, al comienzo, estas aplicaciones necesitaban cierta confidencialidad que las demas capas todavia no podian brindar, por lo que se definió un estándar de confidencialidad para codificar las aplicaciones mediante la encriptación de los paquetes, la confidencialidad consiste en que solo los receptores esperados reciban y puedan decodificar los paquetes que se les envía.
Al momento de publicar el RFC 3550, se estaba desarrollando un perfil de extensión a RTP llamado Secure RTP (SRTP), que fue finalmente estandarizado en Marzo de 2004 por Cisco Systems (véase

RFC 3711), para proveer confidencialidad en el payload del RTP. Aún así, para poder seguir utilizando los algoritmos ya establecidos de interpretación de RTP, lo único que se encriptaba era el payload. El resto del header permanecía igual.

Historia:
RTP fue desarrollado por la Internet Engineering Task Force (IETF) y publicado en Enero de 1996, debido a la creciente demanda de transmisiones en tiempo real a través de internet. Hasta ese momento no había un protocolo capaz de transmitir audio y video en tiempo real entre 2 o más clientes, por lo que se utilizaban conexiones mediante UDP, TCP o mediante Real-Time Streaming Protocol, que solo servía para recibir paquetes (de un canal de televisión, por ejemplo). Originalmente, se estandarizó junto a la publicación del RFC 1889. Luego, en julio de 2003, la universidad de Columbia publicó un nuevo RFC para este protocolo, el RFC 3550. Esta nueva edición dejó obsoleto al RFC mencionado anteriormente, ya que a pesar de mantener casi todo el contenido, incorporó nuevas reglas y algoritmos que siguen siendo respetados hasta hoy.

Bibliografía:

- RTP Wikipedia
- RFC 3550
- Chat GPT
- ¿Qué es RTP?