

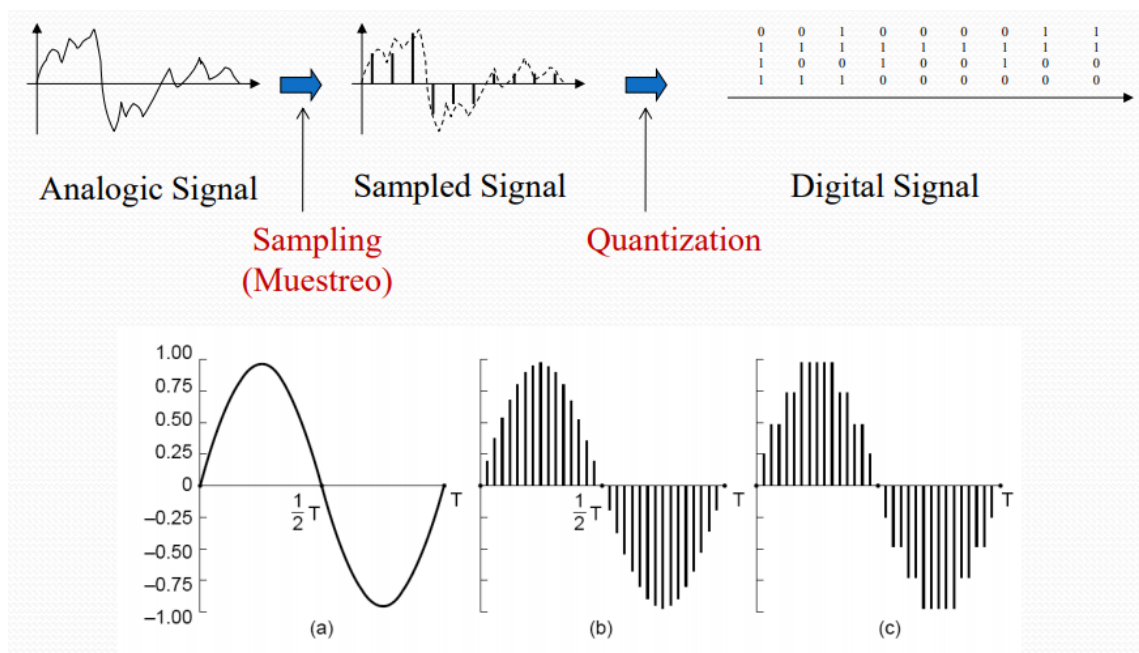
TEMA 6 – Multimedia y arquitecturas QoS

Objetivos

1. Codificación multimedia (técnicas)
2. Identificar las arquitecturas y protocolos involucrados en las aplicaciones multimedia
3. Entender las arquitecturas QoS de Internet

Importante

- **Conversión analógica → Digital (0 y 1s)**



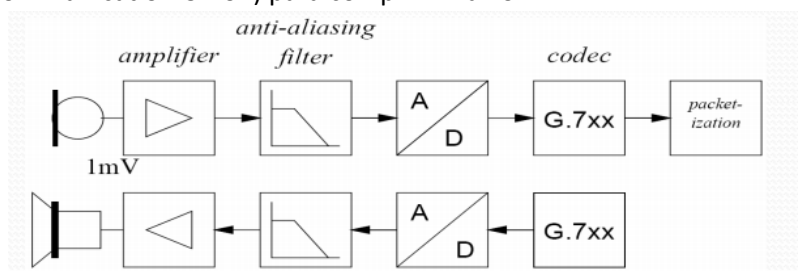
Teorema del muestreo: frecuencia muestreo $\geq 2 \cdot$ ancho de banda. (no sale en el examen)

Básicamente, es para llegar a que, por ejemplo, para convertir una señal de voz, digamos $F_m = 8 \text{ KHz}$ y $b_m = 8 \rightarrow$ Necesitaré $8 \cdot 8 = 64 \text{ bits/seg}$.

Esta técnica de codificación es llamada **Pulse Code Modulation**.

- **Audio** (extra, no sale nada)

Se usa G.7xx. Son un grupo de estándares pertenecientes a ITU-T (International Telecommunication Union) para comprimir la voz.

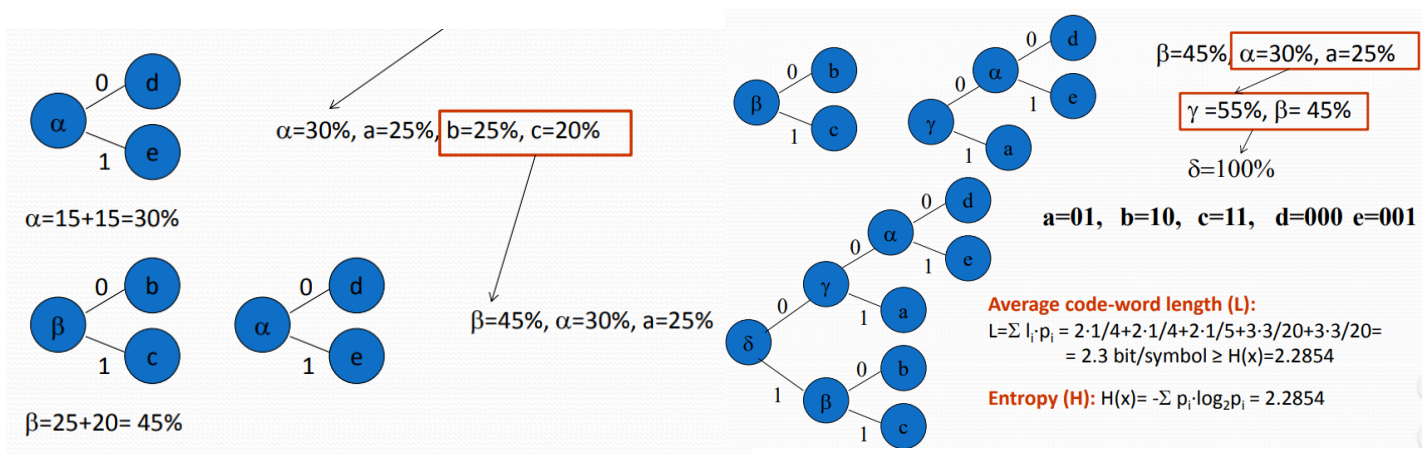


- **Compresión o source coding** → Técnica usada para reducir el número de bits necesarios para la transmisión o para el almacenamiento de información. Puede ser de dos tipos:
 - **Lossless Compression** (sin pérdidas). ZIP, GZIP, GIF...
 - **Lossy Compression** (con pérdidas). Alguna información es perdida, pero las relaciones de compresión son más altas. Usado en audio e imágenes. JPEG, MP3...

Huffman coding (lossless Compression)

Técnica optima de compresión. Saber cómo funciona, ejemplo:

A = 25%, b = 25%, c = 20%, d = 15%, e = 15%

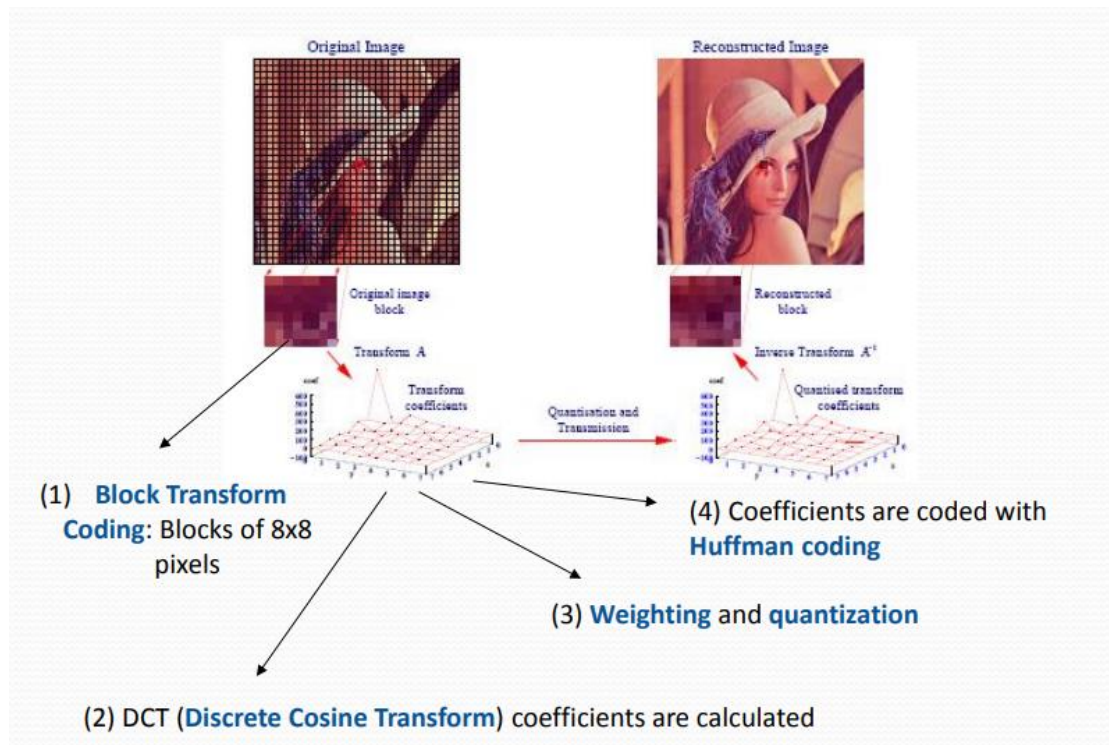


La entropía no la pregunta. Suele preguntar la L (bit/símbolo) y dado un fichero de X símbolos, calcula el tamaño del fichero. Una simple multiplicación.

Multimedia

- De tres tipos:
 - Un fichero multimedia es cargado antes de ser visto. (darle al play) → **stored media**
 - Si un fichero es suficientemente grande para ser cargado antes de verlo, se puede descargar unos cuantos segundos antes de darle al play → **Streaming**
 - La información tiene que llegar en pocos milisegundos desde que se juega en tiempo real. (Usado en VoIP o internet radio, en partidos como ejemplo) → **Continuos playout**
- Cada pixel de una imagen, se representa con una serie de bits (donde entra la iluminación y color)
- **Formatos** → Numero de pixeles por frame (imagen). Como ejemplo, CIF frame tiene 353x288 pixeles, sin compresión significa 24-bit/pixel RGB coding usando HSL o HSV (hecho en IDI)

- **Temporal Redundancy** → Los mismos puntos en imágenes consecutivas tienen luminancia y color similares.
- **Spatial Redundancy** → Los puntos cercanos en la misma imagen tienen luminancia y color similares.
- **Principios de multimedia:** (funcionamiento)



+ Transform → La fuente es dividida en bloques de mida N. Cada bloque se mapea en una secuencia de transformación utilizando un mapeo reversible.

+ Quantization → La secuencia de transformación anterior es cuantificada basadose en una estrategia.

+ Entropy coding (paso 4) → Los datos cuantificados son codificados usando Huffman u otros.

Se ve un ejemplo de esto con una imagen JPEG (pag 20)

Más tecnicas usadas...

- **Predictive coding** → Predecir el valor de los pixeles cuando estos son desconocidos
- **Motion Estimation (ME)** → Examina el movimiento de objetos en una secuencia de imágenes para tratar de obtener vectores que representen el movimiento estimado.
- **Motion Compensation (MC)** → Utiliza el conocimiento del movimiento del objeto así obtenido para lograr la compresión de datos
- **Tipos de frames:**
 - **I-frame (Intra Coded Frame)** → Aplica DCT a un bloque de píxeles NxN
 - **P-frame (Preductive)** → Aplica MC y seguidamente DCT sobre el error frame.

- **Group of Pictures (GOP)**

Secuencia de frames que especifica el orden en que se organizan los frames Intra y los Inter frames.

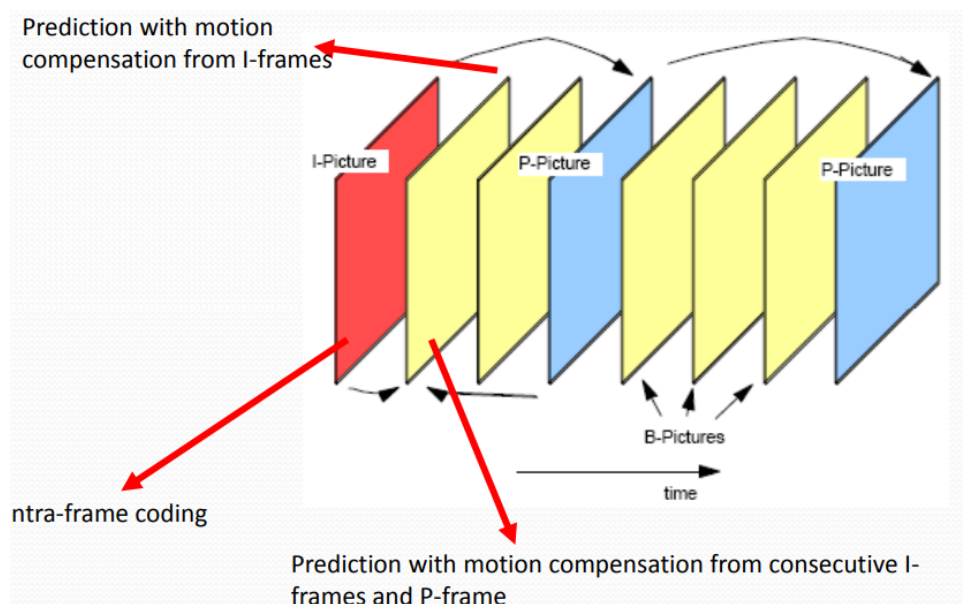
Cada stream de vídeo codificado consiste en sucesivos GOP, y es desde las imágenes que contienen estos GOP desde donde se generan las imágenes visibles.

Puede contener los diferentes tipos de imágenes/frames: (resultado de la compresión de la señal de video de los estándares ITU-T o MPEG)

- I-frames (Intra) → DCT compression
- P-frames (Predictive) → Motion-compensation de los frames anteriores I o P
- B-frames (Bidirectionally Predictive) → Motion-compensation de los frames anteriores y siguientes I o P

GOP (M,N), donde M = distancia entre P frames consecutivos y N = num total de la secuencia.

Ej. GOP(3,12) → (IBBPBBPBBPBB)12 (IBBPBBPBBPBB)12 (IBBPBBPBBPBB)12

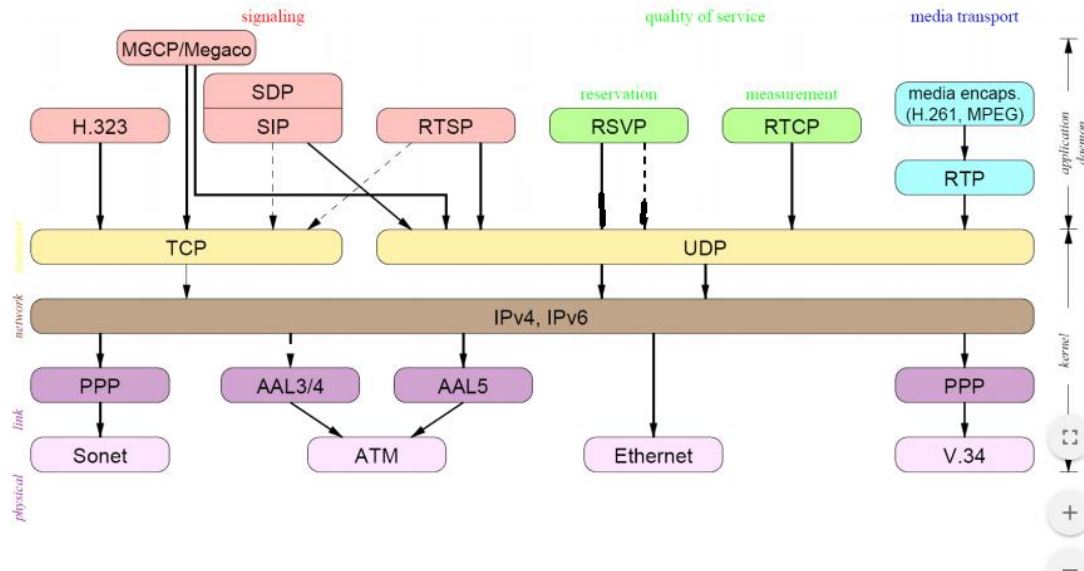


+ Un GOP siempre empieza con un frame tipo I y a continuación le siguen varios frames P, en cada caso, con varios frames de distancia. Esos huecos de imágenes se ocupan con frames B.

- **Protocolos importantes**

¿Cómo sabe el proceso el tiempo en el que tiene que volver a reproducir la información MM, paquetes perdidos, cambio en el orden de los paquetes, tipo de codificación...? (**necesidad de un protocolo para conseguir esta información**)

-
-
-



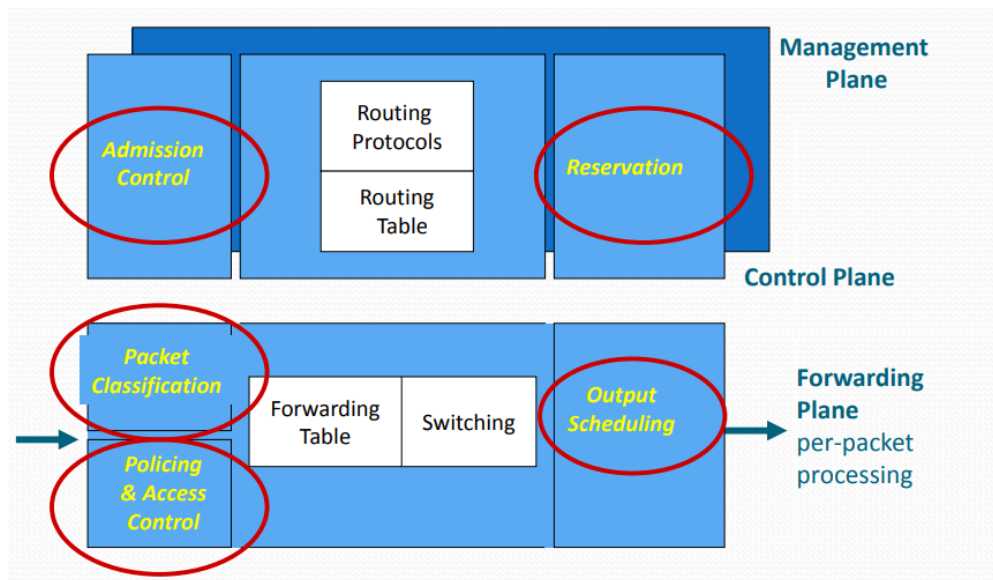
- **RTP (Real-time Transport Protocol)** → Se encarga de la transferencia de datos en real-time. Incluye información que facilita la tarea de recuperar la señal correcta en el receptor; son **timestamps** (para la sincronización), **números de secuencia** (para la pérdida de paquetes) y el **payload format** (que indica el formato de codificación de los datos)
- **RTCP (RTP Control Protocol)** → Usado juntamente con RTP que sirve para especificar la calidad del servicio (Qos). Permite el **envío de estadísticas** del transmisor al receptor **que son útiles en el momento de la codificación / decodificación** de la transmisión / recepción de información en tiempo real

Arquitecturas QoS

- **Qos (Quality of Service)** → Es la habilidad de proveer diferentes prioridades a diferentes aplicaciones, usuarios o flujo de datos. Se mide de diferentes maneras (visto en el tema 5)
- Tenemos **dos tipos de arquitecturas Qos**:
 - **IntServ (Servicios integrados)** → Es una arquitectura que especifica los elementos para garantizar la QoS en las redes. Usa una arquitectura llamada **arquitectura por flujo**, que consiste en que **cada flujo es tratado específicamente** en cada ruta para garantizar una QoS. Usa RSVP como protocolo.
 - **DiffServ (Servicios diferenciados)** → Arquitectura que especifica un mecanismo simple y escalable para **clasificar, gestionar el tráfico de red y proporcionar QoS**. Usa una arquitectura llamada **arquitectura por clase**, que consiste en agrupar los flujos en clases de manera que los de una misma clase, reciban el mismo tratamiento en la red.

- IntServ

1. Necesita una **señalización para establecer una nueva conexión**, los routers deben crear y mantener un estado por cada flujo
2. Se usa el **protocolo RSVP para reservar recursos en los routers**, mantienen una **reservación por flujo**.
3. Los routers **deben implementar un control de admisión** para verificar que el recurso necesitado está disponible
4. Cuando el router **recibe el paquete, lo clasifica** dependiendo de donde pertenezca el flujo y lo procesa como le toca.

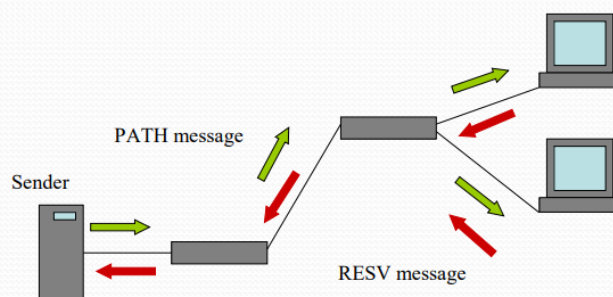


+ (Resource Reservation Protocol) → Es un protocolo de señalización que permite al sender o receiver reservar recursos a lo largo de un camino de comunicación. Funcionamiento básico:

After receiving a PATH message, receivers can request a reservation By sending back RESV messages. These messages:

- **PATH messages** specify the resource requirements
- **RESV messages** set-up the state of routers along the path

After receiving the RESV, sender can start to transmit packets

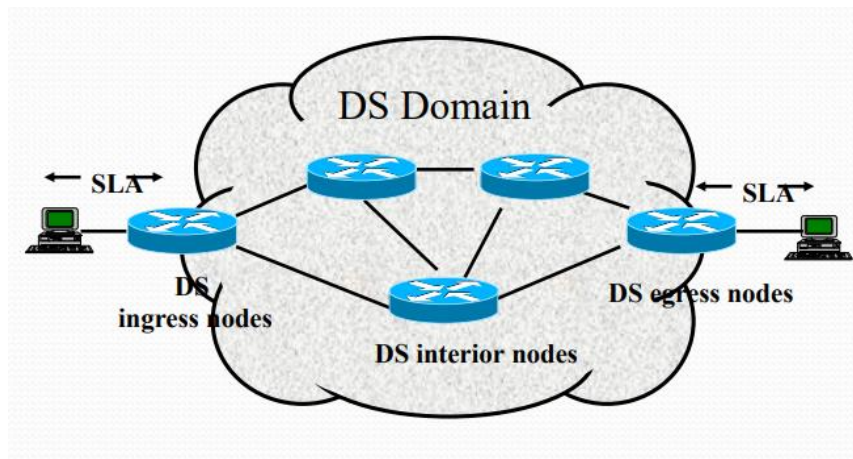


Esta arquitectura presenta problemas de escalabilidad en cuanto al plano de control y el de forwarding. Ya que en una “core network”, los routers deben procesar como 100.000 flujos por link, lo que hace a la red inestable.

¿Por qué? Por cada nuevo paquete escrito o leído del router, se tiene que acceder a la variable estado en tiempo real! (**plano forwarding**) y por cada flujo, hay que procesar varios paquetes RSVP, los cuales modifican tablas de estado y más cosas. (**plano de control**)

- **DiffServ**

Las clases de tráfico están clasificadas y marcadas en el header IP. **Cada clase recibe un tratamiento específico** en el dominio definido en el SLA. (Service Level Agreement)



Tenemos dos tipos de routers:

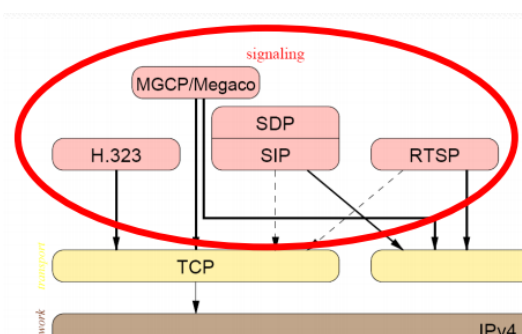
- **Core Routers** → Dan el tratamiento diferente a cada paquete dependiendo del byte ToS (Type of Service) en el header IP.
- **Access Routers** → Tiene la responsabilidad de clasificar y marcar los paquetes en una de las clases.

+ Las funciones más costosas se implementan en el borde de la red, mientras que, en la red central, los routers solo tienen que dar diferentes tratamientos a los paquetes en función de un número reducido de clases.

El campo DS (DiffServ) sustituye el type of service del header de IPv4 (no sale en el examen)

Protocolos multimedia (continuación)

Faltan los **protocolos de señalización**, ver dibujo.



- **Antes de enviar un paquete MM** (es decir, encapsular los datos mediante RTP-UDP-IP), **el cliente debe comunicarse con el servidor y aprender**; ¿dónde están los datos MM? ¿Qué tipo de codificación se usa? ¿Qué protocolos se van a usar de transporte y sesión?
- **Protocolos que se usan:**
 - **RTSP** (Real Time Streaming Protocol) → Para streaming de video
 - **SIP** (Session Initiated Protocol) → Para telefonía a través de Internet (VoIP), pero también permite video streaming
- **RTSP** (Real Time Streaming Protocol)

Es un **protocolo de control para el envío de contenido multimedia** a través de redes IP. Está **basado en TCP** para una entrega fiable y tiene operaciones/sintaxis similares a las de HTTP.

Es usado por el cliente de la aplicación para comunicarse con la información del servidor.

El **contenido multimedia real generalmente NO se entrega a través de la (s) conexión (es) RTSP**, aunque puede intercalarse si es necesario. **RTSP es análogo al control remoto** de los protocolos de transmisión.

Ejemplo de funcionamiento en pag 55 (creo que no sale en el examen, además se parece a http)

+ **VoIP** → Es una tecnología que permite hacer llamadas telefónicas usando una conexión a internet en vez de una línea telefónica regular. Algunos solo permiten hacer llamadas a los que usan el mismo servicio, pero otros permiten llamar usando un número de teléfono.

- **SIP** (Session Initiated Protocol)

Es un **protocolo de señalización a nivel de aplicación** para **poner en marcha, modificar y terminar sesiones en tiempo real** entre participantes a través de una red IP. Puede aceptar cualquier tipo de sesión sigle-media o multi-media.

SDP → Define el contenido de la **sesión** utilizando un conjunto de tipos similar a MIME (Extensiones Multipropósito de Mail de Internet)

