

# Paquetización de Voz y Video en Redes IP

---

# Introducción

---

PAQUETIZACIÓN DE VOZ Y VIDEO EN REDES IP

# Paquetización de los flujos multimedia

---

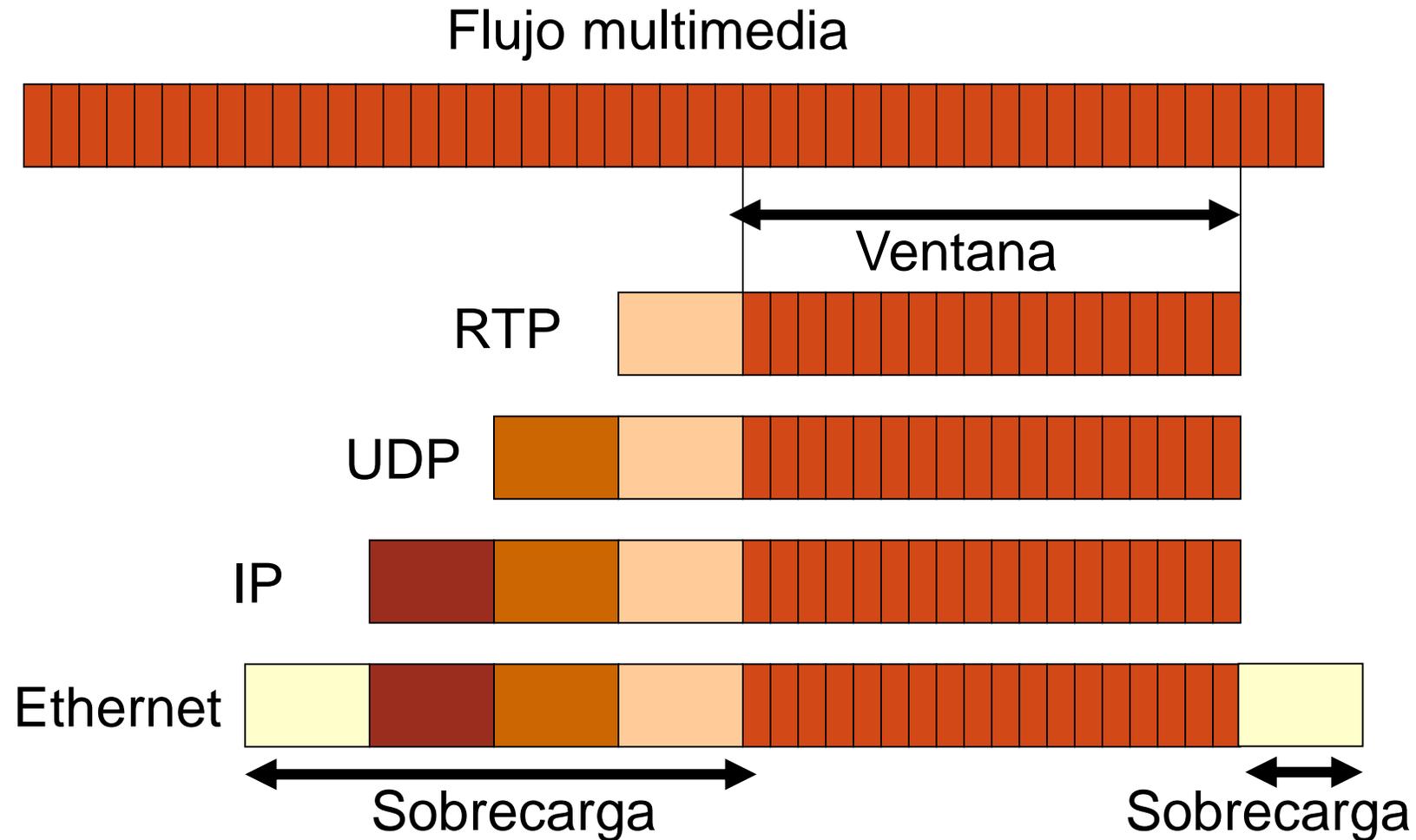
Para poder transmitir la información codificada de voz o video sobre redes de datos, es necesario armar “paquetes”.

Es necesario “juntar” un conjunto apropiado de información para armar un paquete.

Cada paquete tiene una cantidad mínima de información de control

- Cabecal del paquete
- Origen, destino
- Etc.

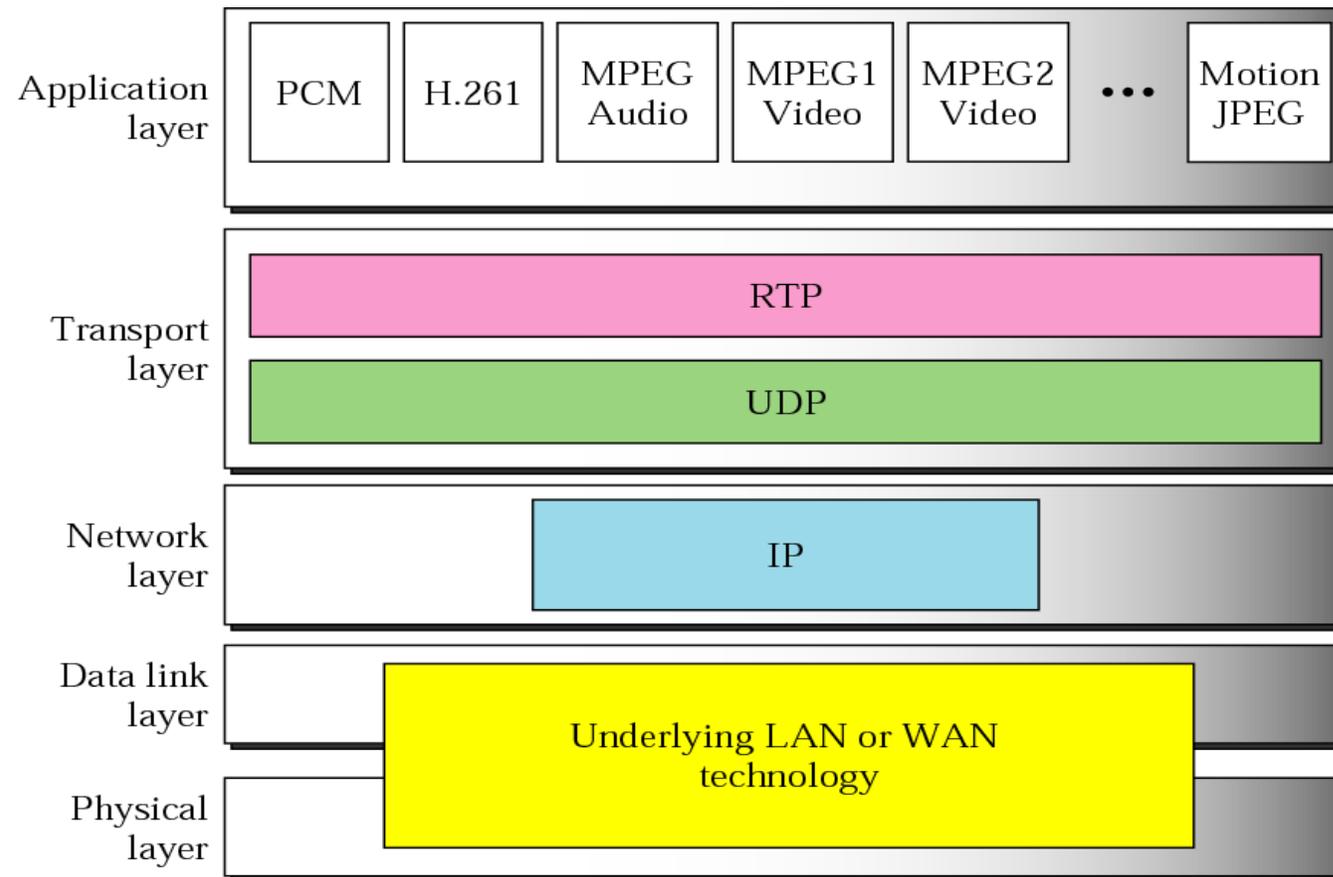
# Transmisión de multimedia sobre redes de datos



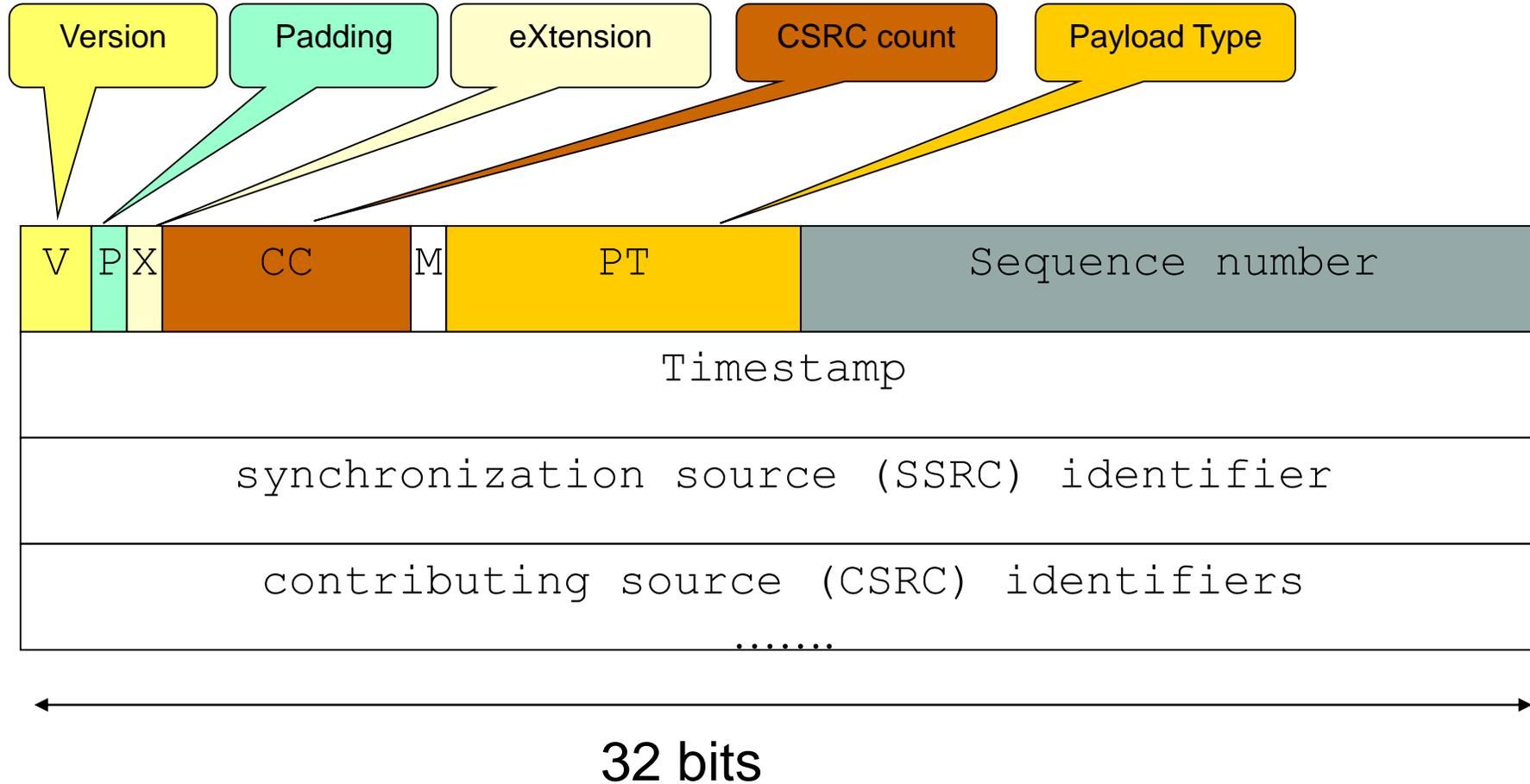
# RTP – Real Time Protocol

Es un protocolo para transmisión de datos de tiempo real (audio y video) sobre IP

Está estandarizado en el RFC 3550



# RTP - Cabezal



# RTP - Cabezal

---

## Payload Type

<b>Payload Type</b>	<b>Formato</b>	<b>Medio</b>	<b>Clock Rate</b>
0	PCM mu-law	Audio	8 kHz
3	GSM	Audio	8 kHz
4	G.723	Audio	8 kHz
8	PCM A-law	Audio	8 kHz
9	G.722	Audio	8 kHz
13	Confort Noise	Audio	
14	MPEG Audio	Audio	90 kHz
15	G.728	Audio	8 kHz
18	G.729	Audio	8 kHz
26	Motion JPEG	Video	90 kHz
31	H.261	Video	90 kHz
32	MPEG-1 o 2 Elementary Stream	Video	90 kHz
33	MPEG-1 o 2 Transport Stream	Video	90 kHz
34	H.263	Video	90 kHz
96 – 127	Dinámico		

# RTP - Cabezal

---

## Payload type

- Identifica el tipo de información que viaja en el paquete
- Indica el tipo de codificación de audio o video, o el contenido de información “especial”
  - CN (Comfort Noise)
  - Tipos dinámicos
    - RFC 2833 (Tonos DTMF, tonos de Fax, etc.)
    - ...

## Sequence number ( 16 bits)

- Número secuencial, generado en el origen. Es usado por el receptor para detectar paquetes perdidos

## Time Stamp (32 bits)

- Marca horaria, del momento de la generación del primer byte de la muestra enviada en el paquete

## Synchronization Source Identifier (32 bits)

- Identifica el origen

# Ejemplo RTP: Paquete de audio

The image shows a Wireshark capture of an RTP audio packet. The packet list pane shows packet 352, which is an RTP packet of 214 bytes. The packet details pane shows the following information:

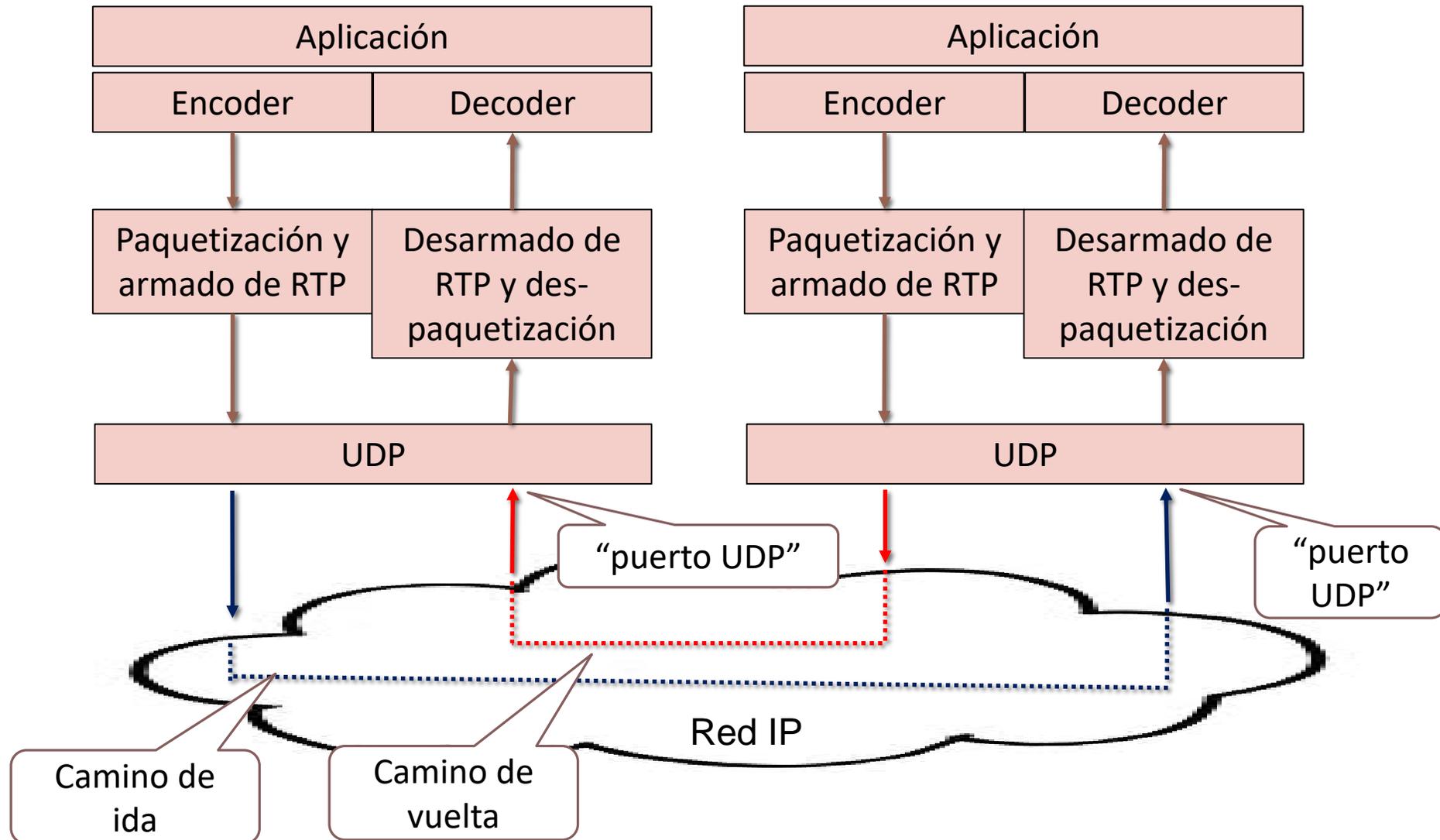
- Frame 352: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits)
- Ethernet II, Src: AlliedTe\_25:3a:cb (00:00:cd:25:3a:cb), Dst: Ibm\_46:89:52 (00:2...
- Internet Protocol, Src: 172.20.20.73 (172.20.20.73), Dst: 192.168.0.85 (192.168...
- User Datagram Protocol, Src Port: 5574 (5574), Dst Port: 12702 (12702)
- Real-Time Transport Protocol
  - 10.. .... = Version: RFC 1889 Version (2)
  - ..0. .... = Padding: False
  - ...0 .... = Extension: False
  - .... 0000 = Contributing source identifiers count: 0
  - 0... .... = Marker: False
  - Payload type: ITU-T G.711 PCMU (0)
  - Sequence number: 40647
  - Timestamp: 2136687751
  - Synchronization source identifier: 0xa499445a (2761507930)

The packet bytes pane shows the raw data of the packet, including the RTP header and the audio payload. The audio payload is shown as a series of hexadecimal bytes and their corresponding ASCII characters.

No.	Time	Source	Destination	Protocol	Info
352	2.174172	172.20.20.73	192.168.0.85	RTP	PT=ITU-T G.711 PCMU, SSRC=0xA499445A,

```
0010 00 c8 92 1b 40 00 7f 11 e6 f6 ac 14 14 49 c0 a8 .....@... ..I..
0020 00 55 15 c6 31 9e 00 b4 5e bd 80 00 9e c7 7f 5b .U..1... A.....[
0030 44 87 a4 99 44 5a ec fa 7e 7c 6f 70 78 69 63 69 D...DZ.. ~|opxici
0040 68 63 72 f8 fe ee e7 ee f4 f2 7a 6d f8 ef f6 e7 hcr.... ..zm...
0050 e6 f2 fa fe 6d 70 fa fc fe ef 7e 6b 6c 69 5f 67 ...mp.. ..~kli_g
0060 7a 6d 6d f0 f4 78 ec e3 eb e7 e1 ee 7e fa 76 70 zmm..x.. ....~.vp
0070 fc f2 f4 ef ef 7a 6c 69 61 60 69 68 6e f2 ed eb .....zli a'ihn...
0080 e9 ee ee ec f4 7e fc 7e 6e 70 78 6f 70 fc 78 7c .....~.~ npxop.x|
0090 ef f2 f4 e9 e5 ed f0 f6 7a 6d 68 64 62 61 69 6c ..... zmhdbai|
00a0 78 f8 f0 ec eb e8 e7 ed ee eb e9 ef f4 7a 70 6f x..... ..zpo
00b0 6c 6a 6e 76 70 6e 70 70 6f fe fa 7e ef ea f4 f4 ljnvpnpp o.....
00c0 f6 7c fe 7e fe fe 7e 7e 7a 7e fa f0 ed f0 f0 f6 .|..... z~.....
00d0 76 68 6b 6d 69 6d vhkmm
```

# Comunicación RTP de extremo a extremo



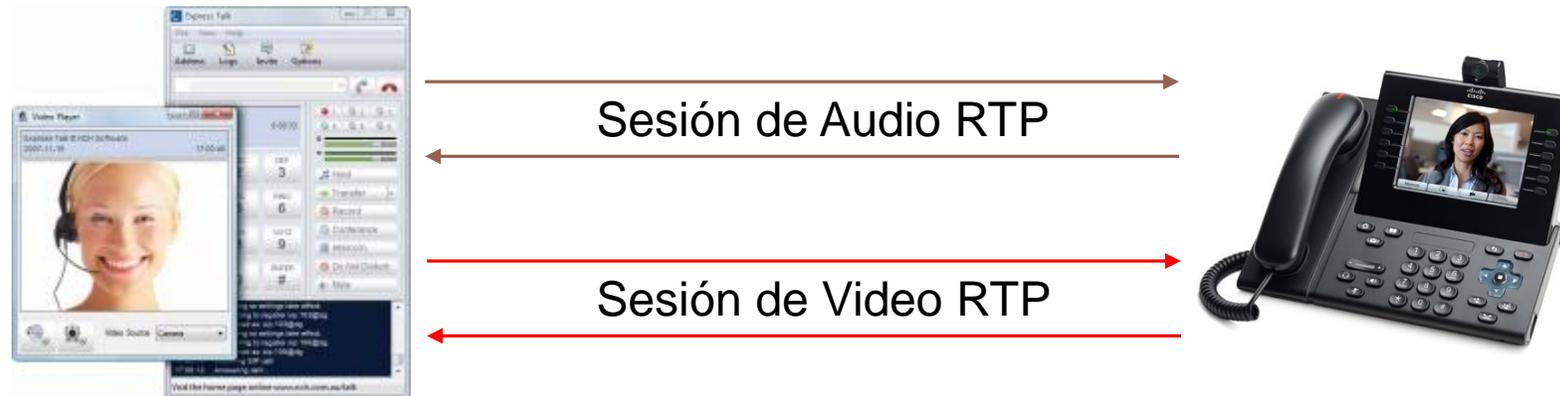
# Sesiones RTP

---

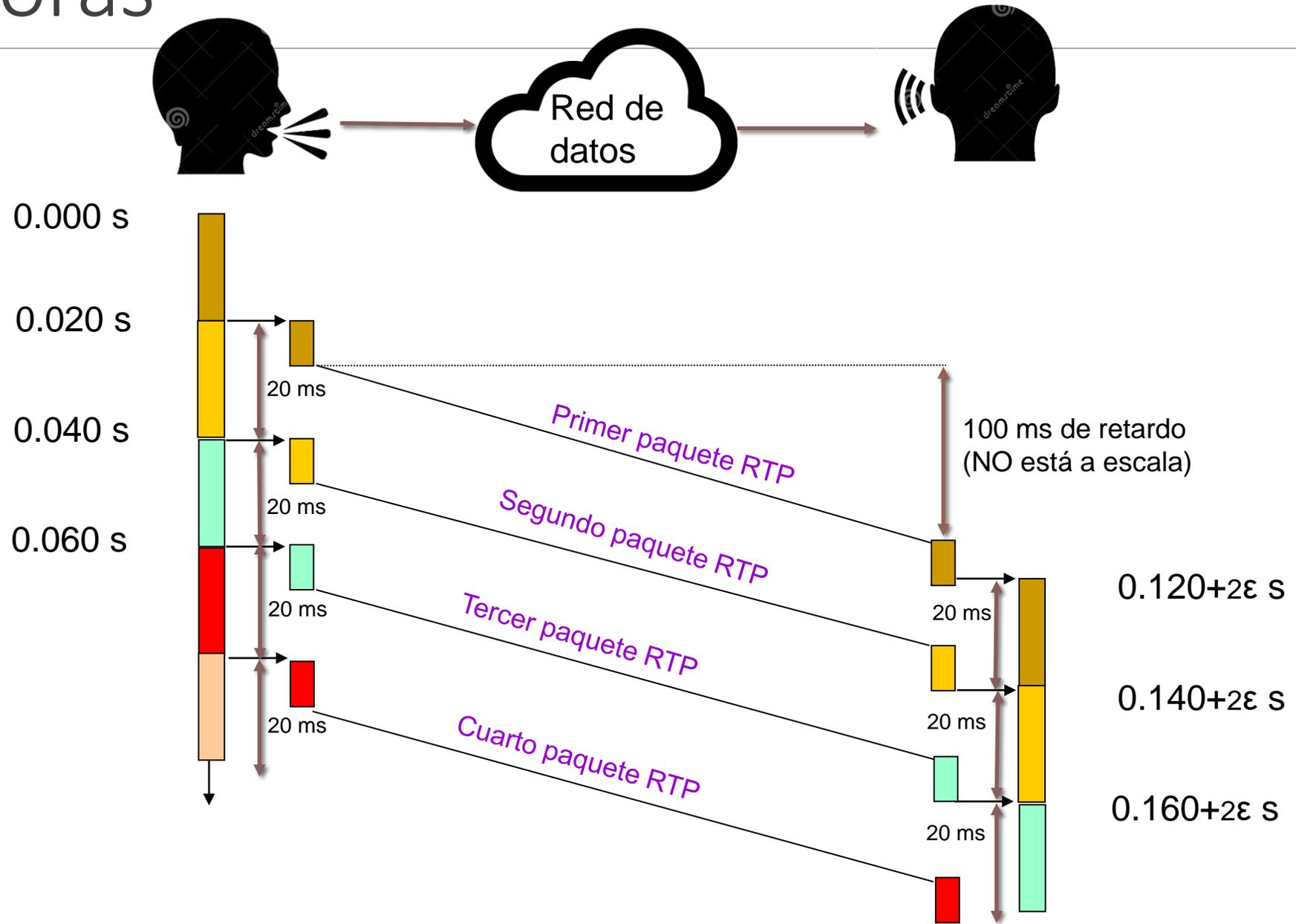
Los terminales establecen *sesiones* para cada tipo de medio

- Audio
- Video

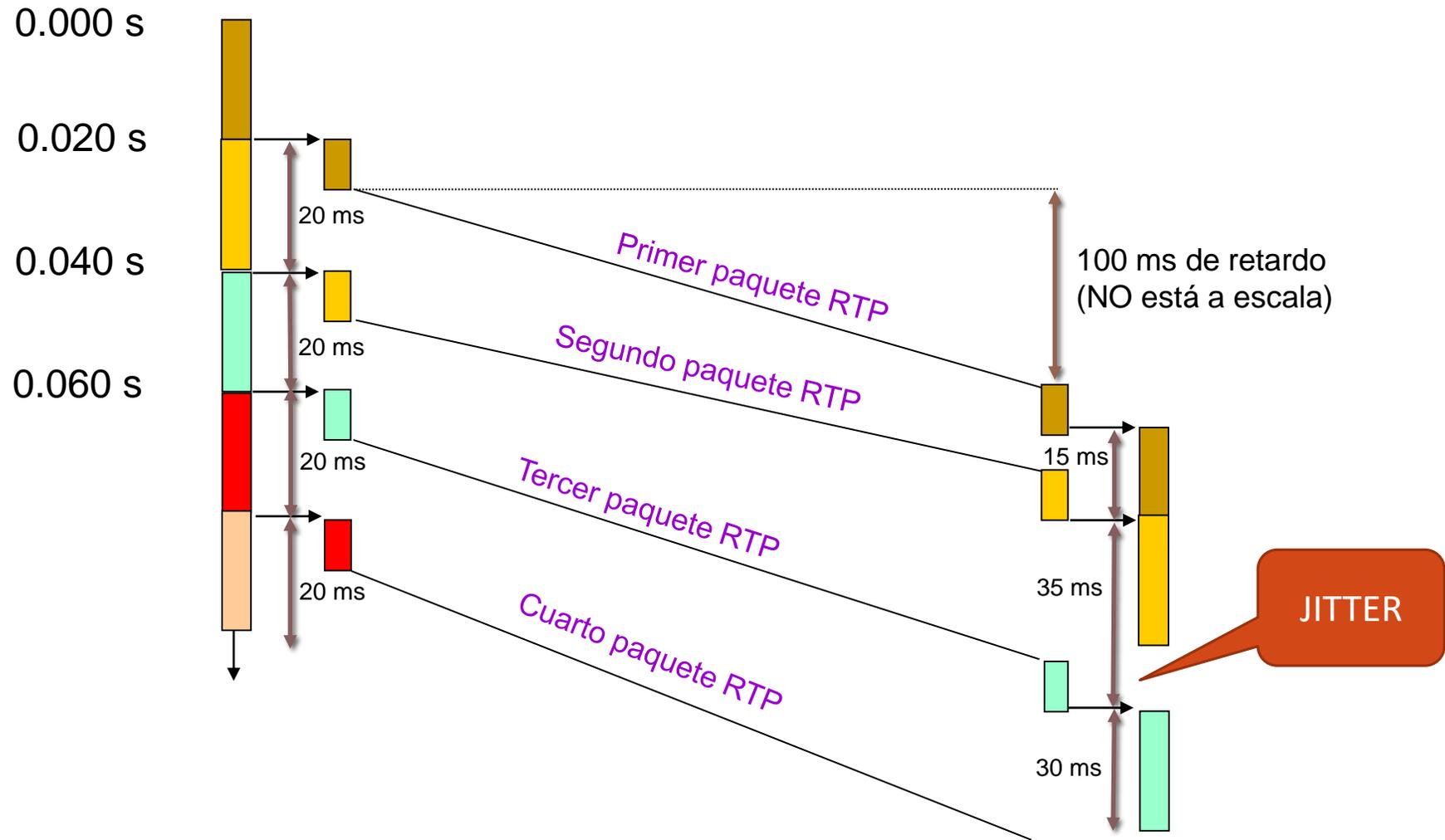
Típicamente cada sesión es “independiente”



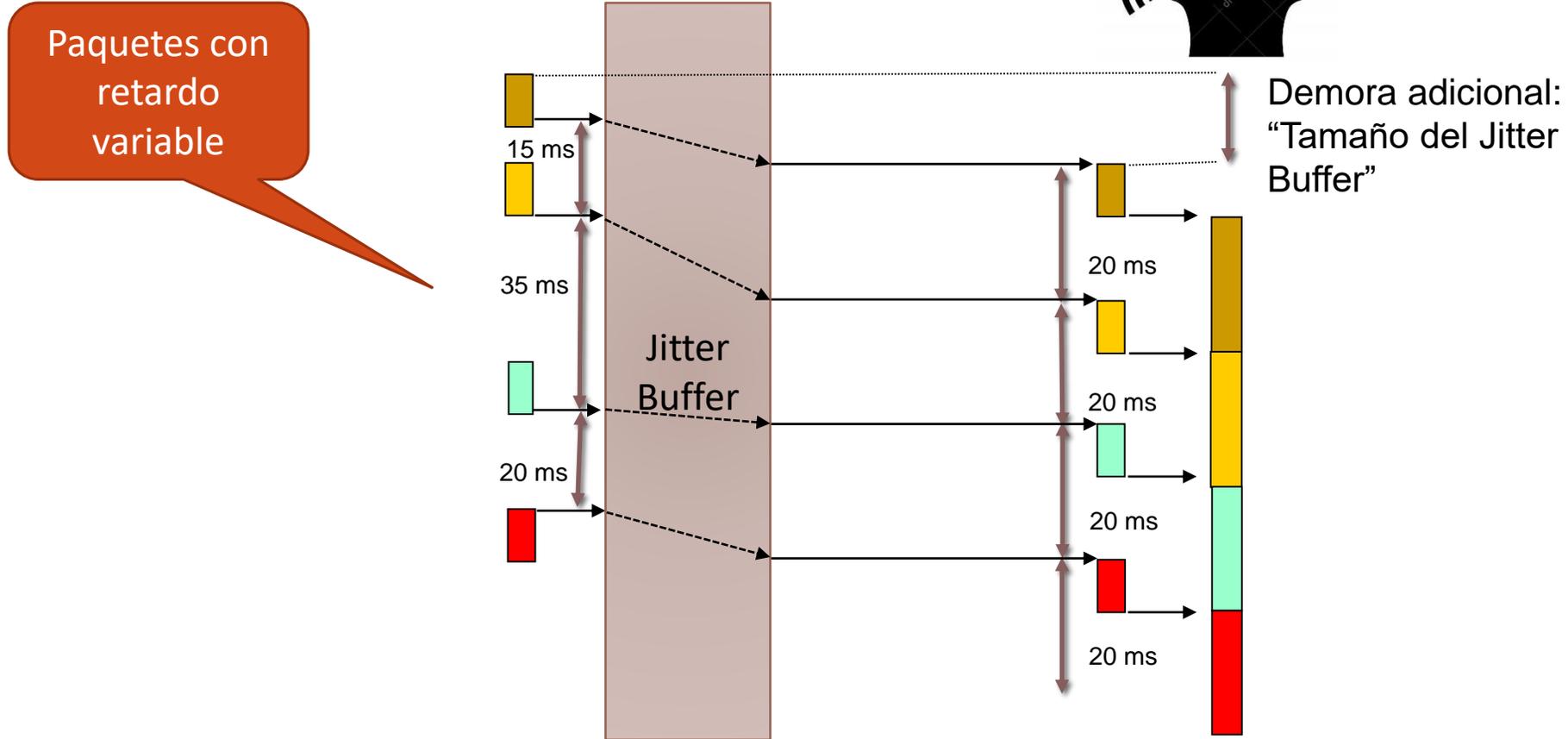
# Demoras



# Demoras variables - Jitter



# Jitter Buffer



El tamaño del Jitter-Buffer depende del tiempo del Jitter.

En VoIP el tamaño típico es de 20 ms a 60 ms

# RTCP –RTP Control Protocol

---

El RFC 3550 establece, además del protocolo RTP, un protocolo de control, RTCP

- Encargado de enviar periódicamente paquetes de control entre los participantes de una sesión
- Proveer realimentación acerca de la calidad de los datos distribuidos (por ejemplo, de la calidad percibida de VoIP).

# RTCP – tipos de datos

---

SR (Sender Report): Envía estadísticas de los participantes “origen” (sender)

RR (Receiver Report): Envía estadísticas de los participantes “destino” (receivers)

SDES (Source Description): Envía ítems de descripción del origen

BYE: Indica el fin de la participación en el intercambio de mensajes RTCP

APP: Funciones específicas para las aplicaciones participantes

# RTCP – Ejemplo de SR y SDES

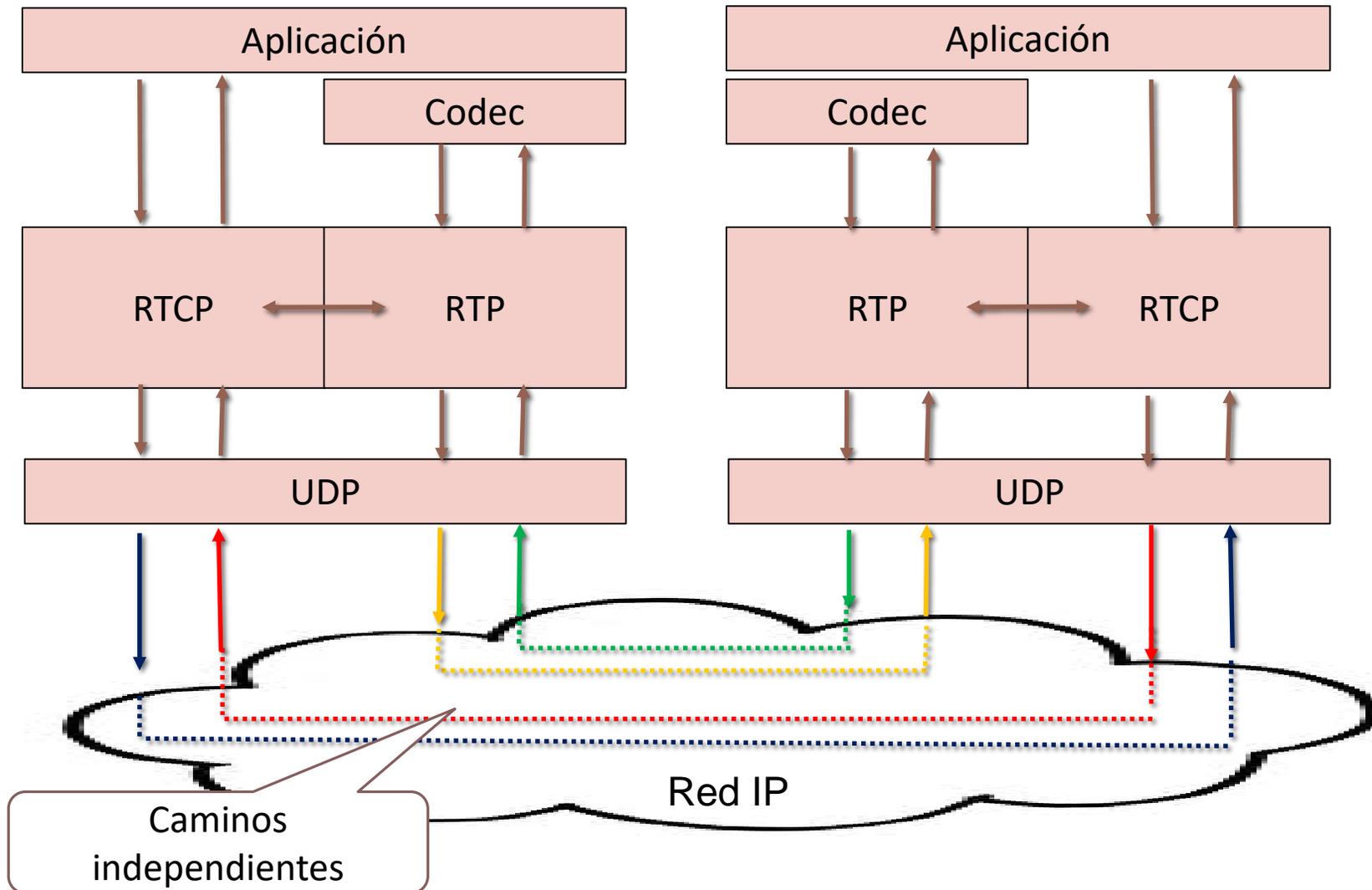
The image shows a Wireshark capture of an RTCP frame. The filter is set to 'rtcp'. The selected packet is number 1223, received at 43.32172 seconds, from source 172.31.0.1 to destination 172.31.0.11. The protocol is RTCP, and the info is 'Sender Report source description'. The packet details pane shows the following structure:

- User Datagram Protocol, Src Port: 19259 (19259), Dst Port: 51777 (51777)
- Real-time Transport Control Protocol (Sender Report)
- Real-time Transport Control Protocol (Source description)
  - [Stream setup by SDP (frame 7)]
    - 10.. .... = Version: RFC 1889 version (2)
    - ..0. .... = Padding: False
    - ...0 0001 = Source count: 1
    - Packet type: source description (202)
    - Length: 19 (80 bytes)
  - Chunk 1, SSRC/CSRC 0x1c360001
    - Identifier: 0x1c360001 (473300993)
    - SDES items
      - Type: CNAME (user and domain) (1)
        - Length: 16
        - Text: 0.0.0@172.31.0.1
      - Type: NAME (common name) (2)
        - Length: 23
        - Text: Cisco IOS, VoIP Gateway
      - Type: TOOL (name/version of source app) (6)
        - Length: 23
        - Text: Cisco IOS, VoIP Gateway
      - Type: END (0)

[RTCP frame length check: OK - 132 bytes]

The packet bytes pane shows the raw data in hexadecimal and ASCII. The ASCII part shows the SDES items: 'c.# .....Eh', 'b.....', 'K;.A. ....6', '.....2v3 =m.....', and 'A'.

# RTCP y RTP de extremo a extremo



# Audio sobre Redes de Datos

---

PAQUETIZACIÓN DE VOZ Y VIDEO EN REDES IP

# RTP – Pacote de audio

The image shows a Wireshark capture of an RTP audio packet. The packet list pane shows packet 352 at time 2.174172, source 172.20.20.73, and destination 192.168.0.85. The packet details pane shows the following structure:

- Frame 352: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits)
- Ethernet II, Src: AlliedTe\_25:3a:cb (00:00:cd:25:3a:cb), Dst: Ibm\_46:89:52 (00:2...
- Internet Protocol, Src: 172.20.20.73 (172.20.20.73), Dst: 192.168.0.85 (192.168...
- User Datagram Protocol, Src Port: 5574 (5574), Dst Port: 12702 (12702)
- Real-Time Transport Protocol
  - 10.. .... = Version: RFC 1889 version (2)
  - ..0. .... = Padding: False
  - ...0 .... = Extension: False
  - .... 0000 = Contributing source identifiers count: 0
  - 0... .... = Marker: False
  - Payload type: ITU-T G.711 PCMU (0)
  - Sequence number: 40647
  - Timestamp: 2136687751
  - Synchronization source identifier: 0xa499445a (2761507930)

The packet bytes pane shows the raw data in hexadecimal and ASCII. The ASCII column shows the beginning of the audio payload, which is a PCM stream.

Offset	Hex	ASCII
0010	00 c8 92 1b 40 00 7f 11 e6 f6 ac 14 14 49 c0 a8	....@... ..I..
0020	00 55 15 c6 31 9e 00 b4 5e bd 80 00 9e c7 7f 5b	.U..1... A.....
0030	44 87 a4 99 44 5a ec fa 7e 7c 6f 70 78 69 63 69	D...DZ.. ~ opxici
0040	68 63 72 f8 fe ee e7 ee f4 f2 7a 6d f8 ef f6 e7	hcr..... ..zm...
0050	e6 f2 fa fe 6d 70 fa fc fe ef 7e 6b 6c 69 5f 67	...mp.. ..~kli_g
0060	7a 6d 6d f0 f4 78 ec e3 eb e7 e1 ee 7e fa 76 70	zmm..x. ....~.vp
0070	fc f2 f4 ef ef 7a 6c 69 61 60 69 68 6e f2 ed eb	.....zli a`ihn...
0080	e9 ee ee ec f4 7e fc 7e 6e 70 78 6f 70 fc 78 7c	.....~.~ npxop.x
0090	ef f2 f4 e9 e5 ed f0 f6 7a 6d 68 64 62 61 69 6c	..... zmhdbail
00a0	78 f8 f0 ec eb e8 e7 ed ee eb e9 ef f4 7a 70 6f	x..... ..zpo
00b0	6c 6a 6e 76 70 6e 70 70 6f fe fa 7e ef ea f4 f4	ljnvpnpp o..~....
00c0	f6 7c fe 7e fe fe 7e 7e 7a 7e fa f0 ed f0 f0 f6	.]..~.~ Z~.....
00d0	76 68 6b 6d 69 6d	vhkmim

# No alcanza con enviar audio

---

La telefonía analógica y luego la digital incluyeron métodos de señalización “en banda” (dentro de la banda de audio).

Ejemplos:

- DTMF (RFC 2833)
- FAX (T.38)

Los nuevos codecs incluyen mecanismos de transmisión discontinua (DTX), y pueden indicarlo “en banda”

# RTP – Ejemplo RFC 2833

The image shows a Wireshark capture of an RTP event packet. The packet list shows frame 501, 58 bytes on wire, from 172.31.0.11 to 172.31.0.1. The protocol is RTP, and the payload type is RTP Event, DTMF Two 2. The packet details pane shows the following structure:

- Frame 501: 58 bytes on wire (464 bits), 58 bytes captured (464 bits)
- Ethernet II, Src: HewlettP\_63:e1:20 (00:1f:29:63:e1:20), Dst: Cisco\_81:ed:06 (00:0c:29:81:ed:06)
- Internet Protocol, Src: 172.31.0.11 (172.31.0.11), Dst: 172.31.0.1 (172.31.0.1)
- User Datagram Protocol, Src Port: 51776 (51776), Dst Port: 19258 (19258)
- Real-Time Transport Protocol
  - [Stream setup by SDP (frame 33)]
    - 10.. .... = Version: RFC 1889 version (2)
    - ..0. .... = Padding: False
    - ...0 .... = Extension: False
    - .... 0000 = Contributing source identifiers count: 0
    - 0... .... = Marker: False
    - Payload type: DynamicRTP-Type-101 (101)
    - Sequence number: 17478
    - [Extended sequence number: 83014]
    - Timestamp: 3792001414
    - Synchronization source identifier: 0x76333d6d (1983069549)
  - RFC 2833 RTP Event
    - Event ID: DTMF Two 2 (2)
    - 0... .... = End of Event: False
    - .0.. .... = Reserved: False
    - ..00 1010 = Volume: 10
    - Event Duration: 320

The packet bytes pane shows the following hex data:

```
0000 00 23 04 81 ed 06 00 1f 29 63 e1 20 08 00 45 00  .#. .... )C. ..E.  
0010 00 2c 6e 2b 40 00 80 11 00 00 ac 1f 00 0b ac 1f  .,n+@... ..  
0020 00 01 ca 40 4b 3a 00 18 58 74 80 65 44 46 e2 05  .!.@K:.. Xt.eDF..  
0030 59 86 76 33 3d 6d 02 0a 01 40                    Y.v3=m... .@
```

# RTP – Ejemplo Comfort Noise

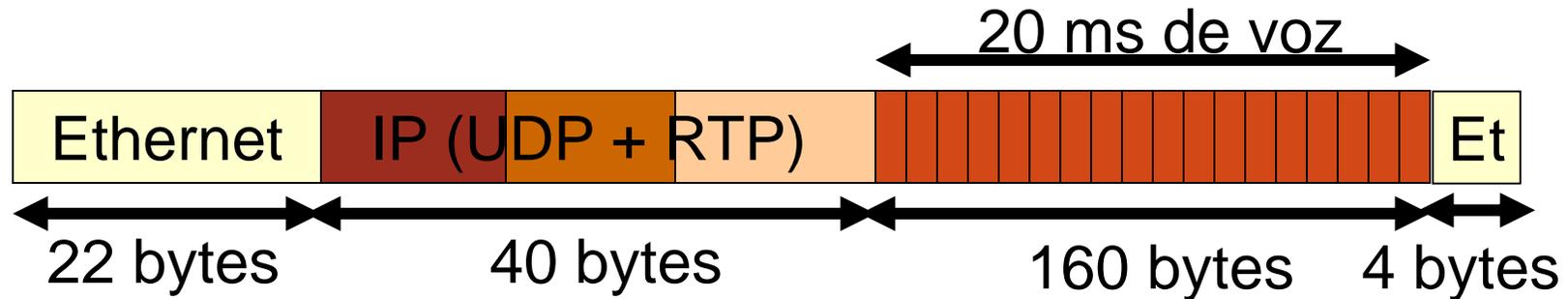
The image shows a Wireshark capture of an RTP packet. The packet list pane shows packet 1130 at time 40.68172.31.0.1, destined to 172.31.0.11, with protocol RTP and info PT=Comfort noise, SSRC=0x1c360001. The packet details pane shows the following structure:

- Frame 1130: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
- Ethernet II, Src: Cisco\_81:ed:06 (00:23:04:81:ed:06), Dst: HewlettP\_63:e1:20 (00:0c:29:13:9e:00)
- Internet Protocol, Src: 172.31.0.1 (172.31.0.1), Dst: 172.31.0.11 (172.31.0.11)
- User Datagram Protocol, Src Port: 19258 (19258), Dst Port: 51776 (51776)
- Real-Time Transport Protocol
  - [Stream setup by SDP (frame 7)]
  - 10.. .... = Version: RFC 1889 version (2)
  - ..0. .... = Padding: False
  - ...0 .... = Extension: False
  - .... 0000 = Contributing source identifiers count: 0
  - 0... .... = Marker: False
  - Payload type: Comfort noise (13)
  - Sequence number: 5637
  - [Extended sequence number: 71173]
  - Timestamp: 3813669695
  - Synchronization source identifier: 0x1c360001 (473300993)
  - Payload: 5a

The packet bytes pane shows the following hex and ASCII data:

```
0000 00 1f 29 63 e1 20 00 23 04 81 ed 06 08 00 45 b8 ..)c. .# .....E.
0010 00 29 13 9e 00 00 fe 11 50 23 ac 1f 00 01 ac 1f .). .... P# .....
0020 00 0b 4b 3a ca 40 00 15 00 00 80 0d 16 05 e3 4f ..K:!.@.. .. ....O
0030 fb 3f 1c 36 00 01 5a 00 00 00 00 00 ..?.6..Z. ....
```

# Ancho de banda para G.711



Ventana = 20 ms

Bytes de voz/trama =  $64 \text{ kb/s} * 20 \text{ ms} / 8 = 160 \text{ bytes}$

Bytes de paquete IP =  $160 + 40 = 200 \text{ bytes}$

Bytes de Trama Ethernet =  $200 + 26 = 226 \text{ bytes}$

Ancho de banda LAN =  $226 * 8 / 20 \text{ ms} = 90.4 \text{ kb/s}$

Este ancho de banda es para la voz en UN sentido. Se debe duplicar para tener en cuenta ambos sentidos

# Ancho de banda

---

Bytes de voz/trama = Velocidad de muestreo \* duración de trama / 8

Bytes de paquete IP = Bytes de voz/trama + 40

Bytes de Trama Ethernet = Bytes de paquete IP + 26

Ancho de banda LAN = Bytes de Trama Ethernet \* 8 / duración de trama

# Ancho de banda de LAN en un sentido

<b>Tipo de Codec</b>	<b>Duración de Trama (ms)</b>	<b>Bytes de voz/Trama</b>	<b>Bytes de paquete IP</b>	<b>Bytes de trama Ethernet</b>	<b>Ancho de Banda en LAN (kbps)</b>
<b>G.711</b>	10	80	120	146	116,8
<b>(64 kbps)</b>	20	160	200	226	90,4
	30	240	280	306	81,6
<b>G.729</b>	10	10	50	76	60,8
<b>(8 kbps)</b>	20	20	60	86	34,4
	30	30	70	96	25,6
<b>G.723.1</b>					
<b>(6.3 kbps)</b>	30	24	64	90	23,9
<b>G.723.1</b>					
<b>5.3 kbps</b>	30	20	60	86	22,9

# Sitios “on line” para calcular ancho de banda

---

<http://www.erlang.com/calculator/lipb/>

# Video sobre Redes de Datos

---

PAQUETIZACIÓN DE VOZ Y VIDEO EN REDES IP

# Transmisión de video sobre redes de datos

---

Las secuencias de video (Elementary Streams) son paquetizadas en unidades llamadas PES (Packetized Elementary Streams), consistentes en un cabezal y hasta 8 kbytes de datos de secuencia.

Estos PES a su vez, son paquetizados en pequeños paquetes, de 184 bytes, los que, junto a un cabezal de 4 bytes (totalizando 188 bytes) conforman el “MPEG Transport Stream” (MTS) y pueden ser transmitidos por diversos medios.

# Transmisión de video sobre redes de datos

---

## RFC 2250:

- Establece los procedimientos para transportar video MPEG-1 y MPEG-2 sobre RTP. Varios paquetes MTS de 188 bytes pueden ser transportados en un único paquete RTP, para mejorar la eficiencia

## RFC 3016 y RFC 3640

- Establecen los procedimientos para transportar flujos de audio y video MPEG-4

## RFC 3984

- Establece los procedimientos para transportar flujos de video codificados en H.264

## RFC 7798

- Describe la forma de transportar H.265 (HEVC) sobre RTP

# MPEG-2 sobre RTP

The image shows a Wireshark capture of an RTP packet. The packet list pane shows several packets, with packet 2 selected. The packet details pane shows the following structure:

- Frame 2: 1370 bytes on wire (10960 bits), 1370 bytes captured (10960 bits)
- Ethernet II, Src: AsustekC\_10:a0:ad (00:23:54:10:a0:ad), Dst: IntelCor\_a0:77:fa (00:1c:c0:a0:77:fa)
- Internet Protocol, Src: 192.168.170.38 (192.168.170.38), Dst: 192.168.170.32 (192.168.170.32)
- User Datagram Protocol, Src Port: terabase (4000), Dst Port: search-agent (1234)
- Real-Time Transport Protocol
  - 10.. .... = Version: RFC 1889 version (2)
  - ..0. .... = Padding: False
  - ...0 .... = Extension: False
  - .... 0000 = Contributing source identifiers count: 0
  - 1... .... = Marker: True
  - Payload type: MPEG-II transport streams (33)
  - Sequence number: 18957
  - Timestamp: 484245819
  - Synchronization source identifier: 0x79dd8f64 (2044563300)
  - ISO/IEC 13818-1 PID=0x100 CC=4
  - ISO/IEC 13818-1 PID=0x100 CC=5
  - ISO/IEC 13818-1 PID=0x100 CC=6
  - ISO/IEC 13818-1 PID=0x100 CC=7
  - ISO/IEC 13818-1 PID=0x100 CC=8
  - ISO/IEC 13818-1 PID=0x100 CC=9
  - ISO/IEC 13818-1 PID=0x100 CC=10

Annotations in the image:

- A red box highlights the "Payload type: MPEG-II transport streams (33)" field, with a red arrow pointing to it and the text "Payload Type: MTS (MPEG-2 Transport Stream)".
- A red box highlights the list of ISO/IEC 13818-1 PIDs, with a red bracket and the text "7 paquetes MTS (MPEG-2 Transport Stream) dentro de un mismo paquete RTP".

# MPEG-2 sobre RTP

Test6\_720p\_rtp\_MPEG2.pcap - Wireshark

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
2	0.000322	192.168.170.3	192.168.170.32	RTP	PT=MPEG-II transport streams, SSRC=0x79DD8F64, S...

Frame 2: 1370 bytes on wire (10960 bits), 1370 bytes captured (10960 bits)

- Ethernet II, Src: AsustekC\_10:a0:ad (00:23:54:10:a0:ad), Dst: IntelCor\_a0:77:fa (00:1c:c0:a0:77:fa)
- Internet Protocol, Src: 192.168.170.38 (192.168.170.38), Dst: 192.168.170.32 (192.168.170.32)
- User Datagram Protocol, Src Port: terabase (4000), Dst Port: search-agent (1234)
- Real-Time Transport Protocol
  - 10.. .... = Version: RFC 1889 Version (2)
  - ..0. .... = Padding: False
  - ...0 .... = Extension: False
  - .... 0000 = Contributing source identifiers count: 0
  - 1... .... = Marker: True
  - Payload type: MPEG-II transport streams (33)
  - Sequence number: 18957
  - Timestamp: 484245819
  - synchronization source identifier: 0x79dd8f64 (2044563300)
- ISO/IEC 13818-1 PID=0x100 CC=4
  - Header: 0x47010014
    - 0100 0111 .... = Sync Byte: Correct (0x00000047)
    - ..... 0... .. = Transport Error Indicator: 0
    - ..... .0.. .. = Payload Unit Start Indicator: 0
    - ..... .0. .... = Transport Priority: 0
    - ..... ..0 0001 0000 0000 .... = PID: Unknown (0x00000100)
    - ..... ..00.. .... = Transport Scrambling Control: Not scrambled
    - ..... ..01 .... = Adaption Field Control: Payload only (0x00000000)
    - ..... ..0100 = Continuity Counter: 4
  - [MPEG2 PCR Analysis]
  - Payload: 2e34302061713d313a312e3030008000000016588840033ff...
- ISO/IEC 13818-1 PID=0x100 CC=5
- ISO/IEC 13818-1 PID=0x100 CC=6
- ISO/IEC 13818-1 PID=0x100 CC=7
- ISO/IEC 13818-1 PID=0x100 CC=8

0030 01 3b 79 dd 8f 64 47 01 00 14 2e 34 30 20 61 71 ..y..dG...40 aq

0040 3d 31 3a 31 2e 30 30 00 80 00 00 01 65 88 84 00 =1:1.00...e..

0050 33 ff f7 02 1a e8 13 5c 59 2f e9 52 cd 85 e2 7b 3.....\Y/R...{

0060 2e 99 80 2c af e2 6f 10 4a 6b 6d 27 82 40 63 ae .....o jkm'@c.

0070 bc 0c ad b8 f3 2a 1a 4a 6d 71 69 f9 1e 76 d8 92 .....\*]mqj.v.v.

0080 27 2e 86 7c b3 af 65 24 63 08 70 f6 76 0f 0c 50 7.....43p8&~&

ISO/IEC 13818-1 (mp2t), 188 bytes

Packets: 27436 Displayed: 27436 Marked: 0 Load time: 0:00... Profile: Default

Cabezal de MTS  
(4 bytes)

Payload de MTS  
(184 bytes)

# H.264 sobre RTP

Test1\_vga\_rtp\_h264.pcap - Wireshark

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
249	1.296720	192.168.170.3	192.168.170.32	RTP	PT=DynamicRTP-Type-96, SSRC=0x5F1213D0, Seq: 27668
250	1.312343	192.168.170.3	192.168.170.32	RTP	PT=DynamicRTP-Type-96, SSRC=0x5F1213D0, Seq: 27669
251	1.312353	192.168.170.3	192.168.170.32	RTP	PT=DynamicRTP-Type-96, SSRC=0x5F1213D0, Seq: 27670
252	1.327966	192.168.170.3	192.168.170.32	RTP	PT=DynamicRTP-Type-96, SSRC=0x5F1213D0, Seq: 27671
253	1.328213	192.168.170.3	192.168.170.32	RTP	PT=DynamicRTP-Type-96, SSRC=0x5F1213D0, Seq: 27672
254	1.328270	192.168.170.3	192.168.170.32	RTP	PT=DynamicRTP-Type-96, SSRC=0x5F1213D0, Seq: 27673
255	1.343582	192.168.170.3	192.168.170.32	RTP	PT=DynamicRTP-Type-96, SSRC=0x5F1213D0, Seq: 27674
256	1.343806	192.168.170.3	192.168.170.32	RTP	PT=DynamicRTP-Type-96, SSRC=0x5F1213D0, Seq: 27675
257	1.343863	192.168.170.3	192.168.170.32	RTP	PT=DynamicRTP-Type-96, SSRC=0x5F1213D0, Seq: 27676

Frame 249: 1484 bytes on wire (11872 bits), 1484 bytes captured (11872 bits)

Ethernet II, Src: AsustekC\_10:a0:ad (00:23:54:10:a0:ad), Dst: IntelCor\_a0:77:fa

Internet Protocol, Src: 192.168.170.38 (192.168.170.38), Dst: 192.168.170.32 (192.168.170.32)

User Datagram Protocol, Src Port: terabase (4000), Dst Port: search-agent (1234)

Real-Time Transport Protocol

- 10... .. = Version: RFC 1889 Version (2)
- ..0. .... = Padding: False
- ...0 .... = Extension: False
- .... 0000 = Contributing source identifiers count: 0
- 0... .... = Marker: False
- Payload type: DynamicRTP-Type-96 (96)
- Sequence number: 27668
- Timestamp: 2618277648
- Synchronization Source identifier: 0x5f1213d0 (1595020240)
- Payload: 1c013d7f7b4f4233db6202da1fe1de60e534e01429c00e07...

0020 aa 20 0f a0 04 d2 05 aa 99 12 80 60 6c 14 9c 0f  
0030 bf 10 5f 12 13 d0 1c 01 3d 7f 7b 4f 42 33 db 62  
0040 02 da 1f e1 de 60 e5 34 e0 14 29 c0 0e 07 e1 3e  
0050 09 6b 4d c4 b8 64 ce 7b de 6c 6e 8b 9c 00 10 01  
0060 80 e4 a1 40 62 09 0a a5 08 61 9f ab 1e 79 1b a6  
0070 38 ea 6b 5c 70 27 2c ff 8d 17 28 b8 b6 4d d7 06

Real-Time Transport Protocol (rtp), 1442 bytes      Packets: 3830 Displayed: 3830 Marked: 0 Load time: ...      e Default

Payload del tipo  
"dinámico"

Payload de H.264  
(1430 bytes)

# Ancho de Banda de Video

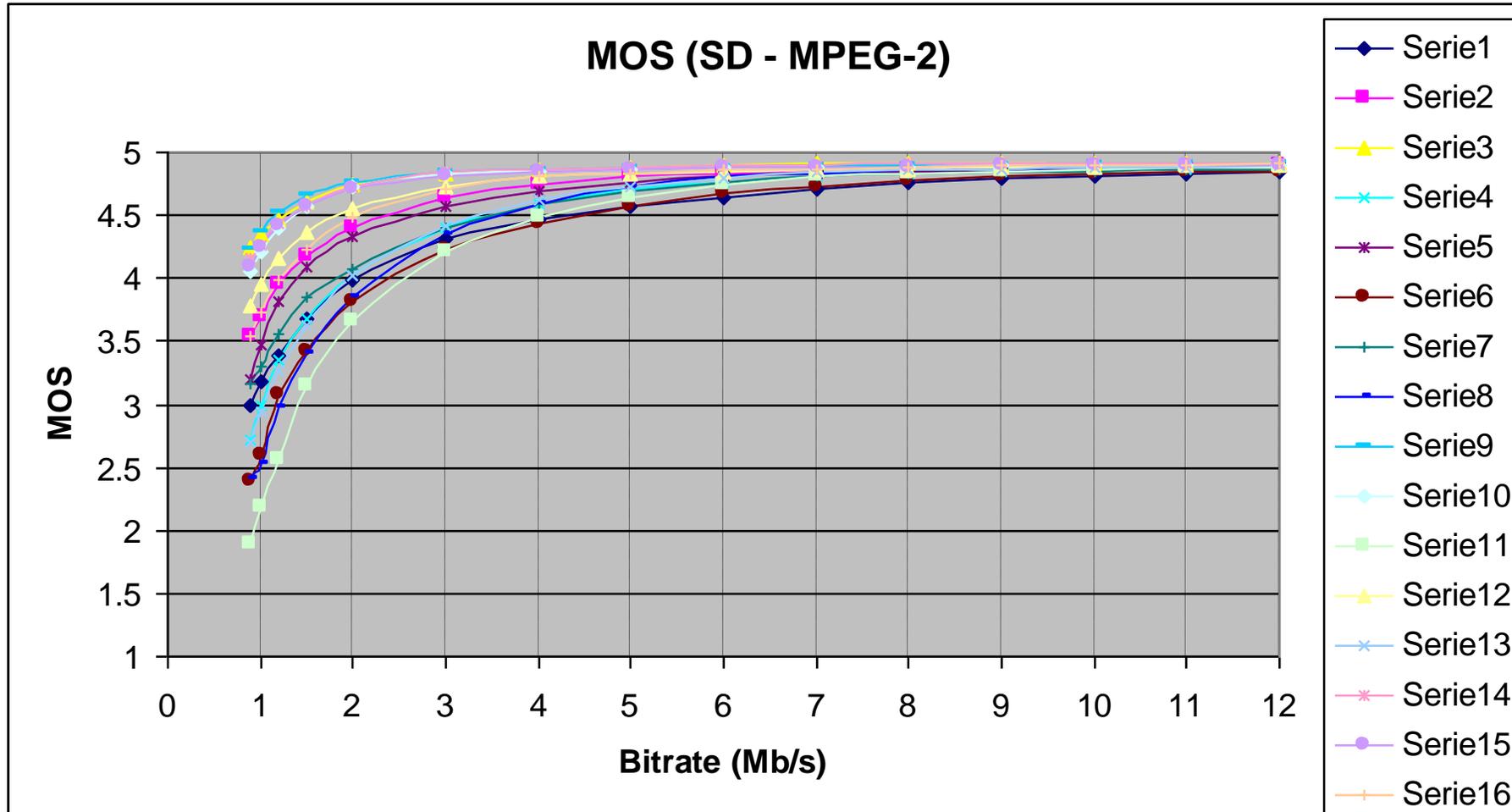
---

El ancho de banda requerido depende de

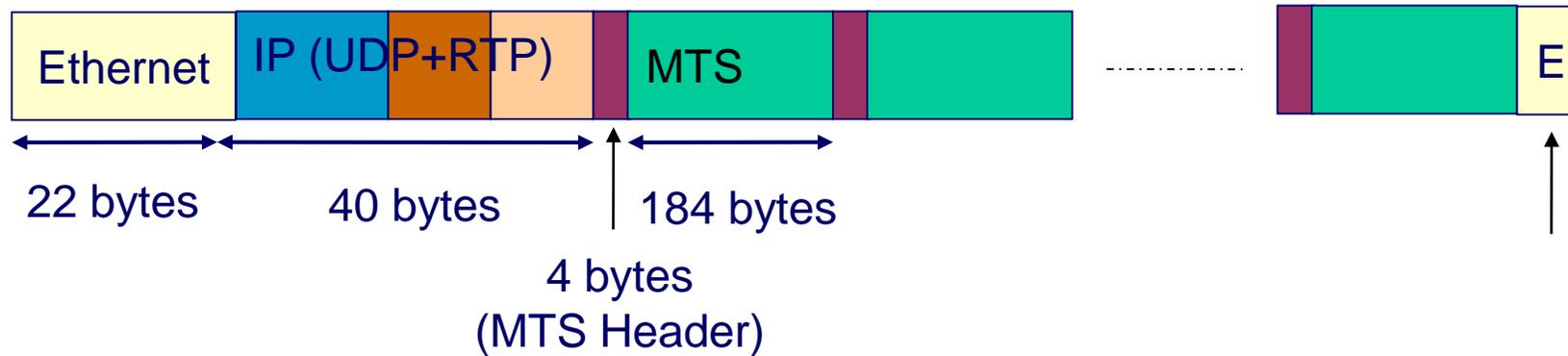
- Tipo de codificación utilizada (MPEG-1, 2, 4, H264, etc.)
- Resolución (tamaño de los cuadros SD, CIF, QCIF, etc.)
- Tipo de cuantización seleccionado
- Movimiento
- Textura

La codificación de video es estadística, y depende de la imagen transmitida

# Calidad vs Ancho de Banda



# Ancho de banda en LAN para MPEG-2 con MTS



$7 \times 184 = 1288$  bytes de contenido MPEG-2

$40 + 4 \times 7 = 68$  bytes de cabezales a nivel de capa 3 (IP)

26 bytes de cabezales adicionales a nivel de capa 2

# Ancho de banda en LAN para MPEG-2 con MTS

---

El ancho de banda de MPEG-2 transportado en RTP

- 5.3% (68/1288) mayor que el ancho de banda propio del video en capa 3 (IP)
- 7.3 % (94/1288) mayor que el ancho de banda propio del video en capa 2 (Ethernet)

# Ancho de banda en LAN para H.264

---

H.264 encapsulado directamente sobre RTP (sin utilizar TS)

- Se pueden enviar hasta 1430 bytes de “payload” en un paquete IP/UDP/RTP
- El ancho de banda en capa 3 es 2.8% (40/1430) mayor que el del propio video codificado
- En capa 2 es 4.6% (66/1430) mayor que el del propio video codificado.

# Seguridad en RTP

---

PAQUETIZACIÓN DE VOZ Y VIDEO EN REDES IP

# RTP es inseguro

El protocolo RTP es “abierto”.

- Los paquetes capturados pueden ser fácilmente decodificados
- Herramientas habituales (como wireshark) permiten escuchar el audio codificado en flujos RTP

Prueba 1.pcap [Wireshark 1.8.7 (SVN Rev 49382 from /trunk-1.8)]

File Edit View Go Capture Analyze Statistics Telephony Tools Inte

Filter: rtp

No.	Time	Source	Destination	Protocol
18	74.614669	172.31.0.11	172.31.0.1	RTP
28	79.147107	172.31.0.1	172.31.0.11	RTP
29	79.166980	172.31.0.1	172.31.0.11	RTP
30	79.186980	172.31.0.1	172.31.0.11	RTP
31	79.216977	172.31.0.1	172.31.0.11	RTP
32	79.226978	172.31.0.1	172.31.0.11	RTP
33	79.246976	172.31.0.1	172.31.0.11	RTP
34	79.266977	172.31.0.1	172.31.0.11	RTP
35	79.286975	172.31.0.1	172.31.0.11	RTP
36	79.306973	172.31.0.1	172.31.0.11	RTP
38	79.327100	172.31.0.1	172.31.0.11	RTP
39	79.347098	172.31.0.1	172.31.0.11	RTP
40	79.367096	172.31.0.1	172.31.0.11	RTP
41	79.386972	172.31.0.1	172.31.0.11	RTP
42	79.406970	172.31.0.1	172.31.0.11	RTP
43	79.426969	172.31.0.1	172.31.0.11	RTP
44	79.446968	172.31.0.1	172.31.0.11	RTP
45	79.466967	172.31.0.1	172.31.0.11	RTP
46	79.487091	172.31.0.1	172.31.0.11	RTP
47	79.507091	172.31.0.1	172.31.0.11	RTP

Wireshark: RTP Stream Analysis

Forward Direction Reversed Direction

Analysing stream from 172.31.0.1 port 19348 to 172.31.0.11 port 52740 SSRC = 0x13150001

Packet	Sequence	Delta(ms)	Filtered Jitter(ms)	Skew(ms)	IP BW(kbps)	Marker	Status
28	8165	0.00	0.00	0.00	1.60	SET	[ Ok ]
29	8166	19.87	0.01	0.13	3.20		[ Ok ]
30	8167	20.00	0.01	0.13	4.80		[ Ok ]
31	8168	30.00	0.63	-9.87	6.40		[ Ok ]
32	8169	10.00	1.22	0.13	8.00		[ Ok ]
33	8170	20.00	1.14	0.13	9.60		[ Ok ]
34	8171	20.00	1.07	0.13	11.20		[ Ok ]
35	8172	20.00	1.00	0.13	12.80		[ Ok ]

Max delta = 30.00 ms at packet no. 31  
Max jitter = 1.64 ms. Mean jitter = 0.04 ms.  
Max skew = 10.55 ms.  
Total RTP packets = 912 (expected 912) Lost RTP packets = 0 (0.00%) Sequence errors = 0  
Duration 26.62 s (-3344 ms clock drift, corresponding to 6995 Hz)

Save payload... Save as CSV... Refresh Jump to Gran Player Ne nor

# SRTP

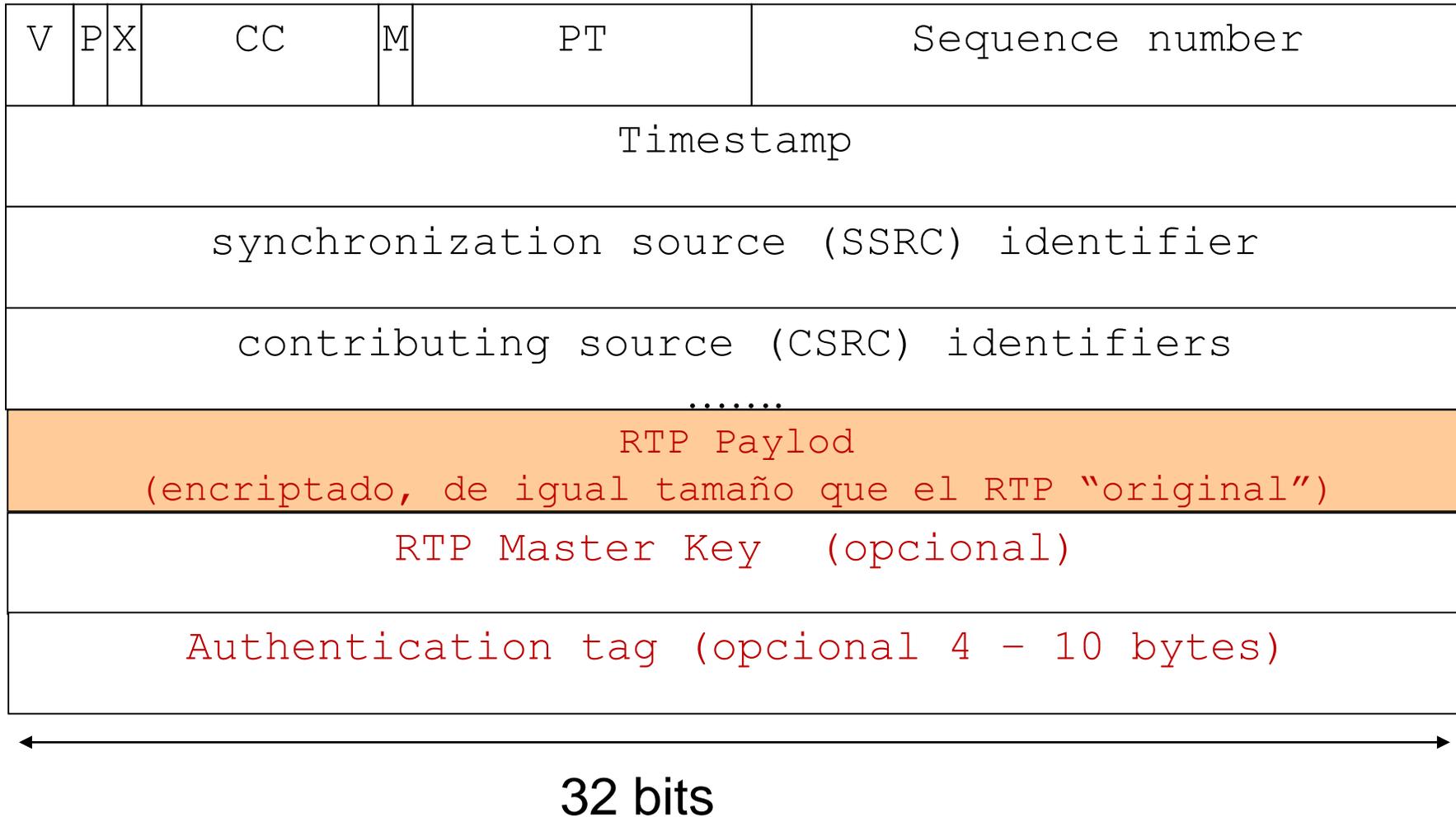
---

Es posible cifrar el medio, a través del protocolo **SRTP** (Secure RTP) (y **SRTCP**), estandarizado en el RFC 3711

Se utilizan técnicas de cifrado AES (Advanced Encryption Standard) para el “payload”

También es posible autenticar el contenido completo del paquete

# SRTCP



# Encriptación

---

Se utilizan técnicas de cifrado AES (Advanced Encryption Standard)

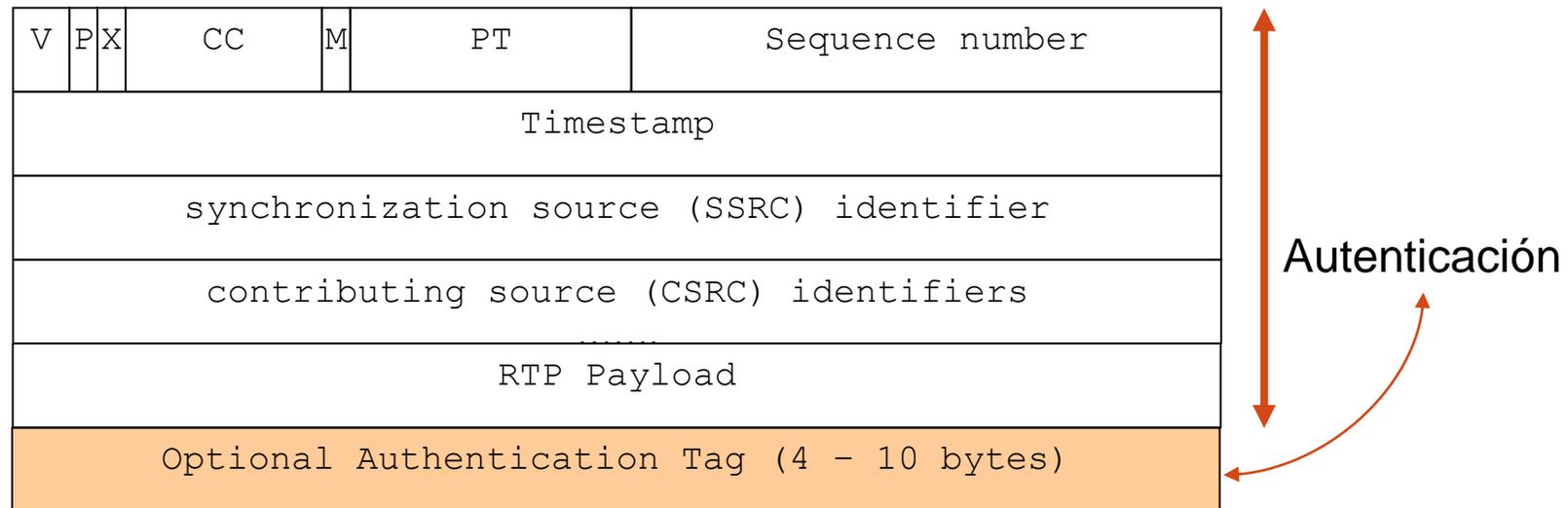
En cada paquete de RTP y RTCP se encripta el contenido con una “clave de sesión” simétrica

Esta clave simétrica debe ser conocida por ambos extremos

- RTP no establece el mecanismo de distribución de ésta clave
- Típicamente se realiza junto con la señalización en el proceso de establecimiento de llamada (x ej. en SIP)

# Autenticación

Se realiza utilizando los algoritmos HMAC (Hash-based Message Authentication Code) -SHA-1 (Secure Hash Algorithm 1)

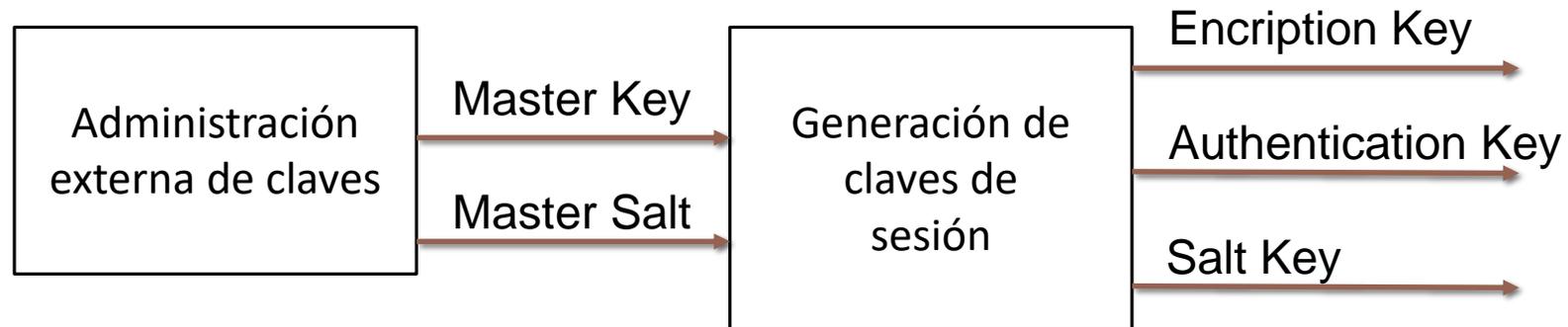


# Claves de encriptación y autenticación

---

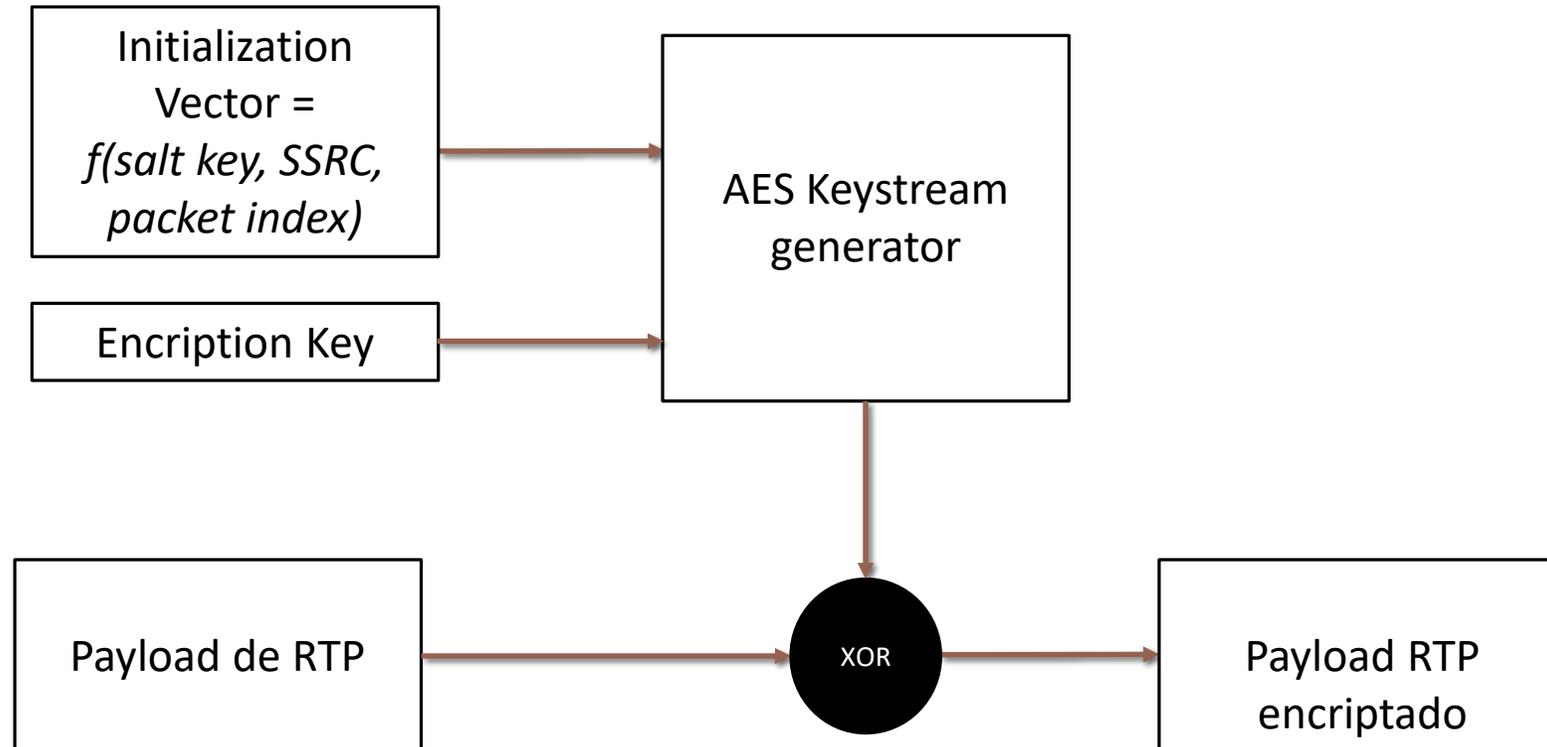
Las claves de sesión son derivadas de una “clave maestra”

- Esta clave maestra debe ser compartida entre los usuarios, y puede ser obtenida de una entidad externa de administración de claves de cifrado



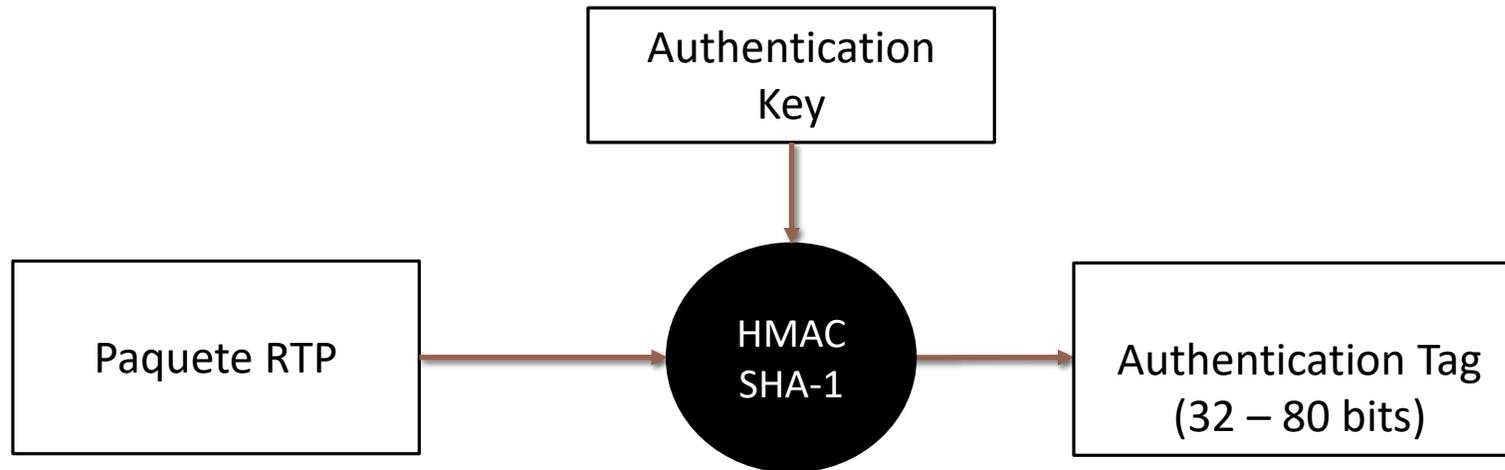
# Encriptación del payload

---



# Autenticación del paquete

---



# Muchas Gracias!

---

PAQUETIZACIÓN DE VOZ Y VIDEO EN REDES IP