

Transmissió

Alexandre Ribelles García

PID_00198472



Els textos i imatges publicats en aquesta obra estan subjectes –llevat que s'indiqui el contrari– a una llicència de Reconeixement-NoComercial-SenseObraDerivada (BY-NC-ND) v.3.0 Espanya de Creative Commons. Podeu copiar-los, distribuir-los i transmetre'ls públicament sempre que en citeu l'autor i la font (FUOC. Fundació per a la Universitat Oberta de Catalunya), no en feu un ús comercial i no en feu obra derivada. La llicència completa es pot consultar a <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.ca>

Índex

Introducció	5
1. Plantejament bàsic	7
2. Codificació i transmissió	8
3. La problemàtica de transmissió àudio/vídeo sobre xarxa IP.	11
3.1. La taxa de transmissió de dades (amplada de banda necessària i disponible)	11
3.2. La cadència del flux enfront de les variacions del retard	14
3.3. La pèrdua de dades	15
4. Transmissió de vídeo en una xarxa IP	17
4.1. Les xarxes IP i els seus protocols	17
4.1.1. Protocols de xarxa (IP)	18
4.1.2. Protocols de transport (UDP, TCP i SCTP)	19
4.2. Tècniques de multidesinació sobre xarxes IP	21
4.3. Els protocols de transmissió en temps real	24
4.3.1. Protocols de temps real (RTP, RTCP i RTSP)	24
4.3.2. Altres protocols de transmissió en temps real	26
4.4. El protocol MPEG-2 TS	28
4.4.1. Paquets i taules del TS	28
5. Les tecnologies de publicació	30
5.1. Tecnologies de publicació	30
5.1.1. Vídeo incrustat	30
5.1.2. Baixada progressiva	31
5.1.3. Transmissió en temps real de vídeo	33
5.1.4. P2P	34
5.2. Els contenidors de vídeo per transmissió en temps real	35

Introducció

Una vegada codificada digitalment la informació i comprimida de manera tan òptima com es pugui segons el seu contingut (en general, una compressió sense pèrdues per a dades pures, i amb pèrdues o sense per a àudio i vídeo), és imprescindible definir mecanismes per a fer-ne la transmissió.

La relació entre codificació i transmissió és intensa, ja que s'han definit mecanismes de codificació específics per a combatre les limitacions en la transmissió, com per exemple els formats multimèdia de distribució que ja coneixeu. És una de les maneres en què la “marea” de la convergència digital ha pogut englobar també el contingut audiovisual en el procés que duu a terme d'unificació de la informació digital sobre els mateixos mitjans digitals de distribució. En la primera part d'aquest mòdul ens centrarem en els problemes presents en la distribució audiovisual sobre xarxes de paquets com les xarxes IP.

El vídeo, sens dubte a causa de les necessitats específiques que té de volum de dades i cadència, ha estat de les últimes informacions a integrar-se, però una vegada fet el pas, és present a tot arreu en la nostra vida diària, com un servei més. El procés d'integració té, doncs, dos elements paral·lels: els estàndards de format digital de vídeo i la implementació de sistemes digitals avançats de transmissió.

La transferència de vídeo i àudio digital en temps real (*streaming*) són bàsicament serveis de valor afegit que tenen una qualitat adaptada a la capacitat i al tipus de receptor client (telèfon mòbil, tauleta, punts d'informació, etc.) gràcies a estàndards més o menys oberts que possibiliten oferir aquest servei al vol. Veurem les maneres de distribució, els mecanismes de xarxa que ho fan possible i els protocols xarxa que es fan servir habitualment. Acabarem el mòdul veient un exemple complet d'implementació de transmissió en temps real.

1. Plantejament bàsic

En una xarxa IP com Internet, pensada per a transmetre dades, els senyals de vídeo no són fàcils de transportar, bàsicament per dues raons. La primera és que no és una xarxa pensada per a vídeo, sinó que el trànsit per la xarxa del vídeo i àudio comprimits es fa amb paquets de dades genèriques (paquets IP), com tots els altres tipus de dades (textos, imatges, etc.) que flueixen per la Xarxa. Aquests paquets de dades circulen per les mateixes xarxes i són encaaminats pels mateixos centres de distribució (*routers*) com qualsevol altra dada, de vegades desordenant-se o perdent-se. La segona raó és un flux de dades molt alt, gairebé constant, de manera que necessita una gran quantitat de paquets per a assegurar la reproducció amb la cadència necessària; per exemple, un vídeo de YouTube d'un minut pot exigir un lliurament d'uns mil paquets sense perdre'n l'ordre i amb una cadència fixa.

Sens dubte, entre el vídeo per Internet (*streaming*) i el vídeo per *broadcast* (difusió àmplia) –per exemple, la TDT– hi ha diferències crítiques en la distribució que obliguen a reptes tècnics en part superats. Més enllà de la tecnologia, però, hi ha una revolució més profunda i de conseqüències més vastes: l'usuari tria el que vol veure des del punt en què ho vol veure, en contraposició amb el fet de veure el que t'ofereixen tal com indica la programació estàtica del canal.

Aquesta nova capacitat de decisió de l'usuari final obliga a replantejar l'atracció del producte i crear nous mercats en què es pugui aprofitar la interacció. Aquesta revolució ja està en marxa, i arguments com la falta de qualitat d'imatge o de tractament professional assegurat dels continguts de la Xarxa perden força any rere any perquè s'incorporen professionals preparats per a treballar en aquest mitjà com el seu entorn natural.

2. Codificació i transmissió

Els dos entorns habituals d'aplicació dels còdecs són, d'una banda, la transmissió d'àudio i vídeo en temps real¹ i, de l'altra, l'emmagatzematge. No obstant això, en cadascun d'aquests dos casos s'ha de seleccionar una codificació adequada a les condicions en què es duu a terme.

⁽¹⁾En anglès, *streaming*.

1) Emmagatzematge

En els sistemes de codificació orientats a emmagatzematge, el codificador té accés a tot el material, de manera que pot fer previsions més fiables i aconseguir taxes altes de compressió alhora que manté la qualitat audiovisual. El temps que trigui a generar el fitxer final no és un factor crític, per la qual cosa els requisits de capacitat de càlcul i necessitat de memòria no són exageradament elevats. Això possibilita que hi pugui haver codificadors de programari dignes per a complir aquesta missió i fins i tot funcionals en equips informàtics domèstics.

2) Transmissió en temps real

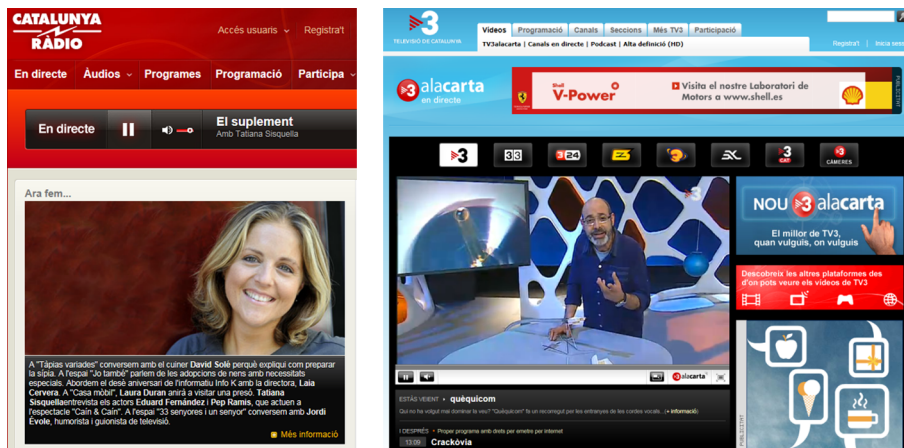
En els sistemes de transmissió en temps real com Internet Ràdio o IPTV (televisió basada en el protocol IP), el material de què es disposa per a fer la codificació va arribant a la mateixa velocitat que es genera i s'ha de codificar a aquesta mateixa velocitat, de manera que fa falta un equip d'alta disponibilitat, fiabilitat, gran capacitat de memòria i excel·lent capacitat de càlcul. La majoria són equips dedicats de maquinari d'alt valor situats a l'empresa que proveeix el servei: la majoria de webs de companyies de *broadcast* (difusió àmplia) actuals de ràdio i televisió en són un exemple.

Il·lustració 1. Equip codificador HD per a TDT o IPTV



En altres serveis de transmissió en temps real menys ambiciosos (com per exemple la videoconferència), en els quals la qualitat visual no és rellevant i la cadència de les imatges no s'assegura, els requisits del codificador es relaxen i n'hi pot haver fins i tot en programari (FaceTime, Messenger, etc.).

Il·lustració 2. Altres serveis de transmissió en temps real menys ambiciosos



La majoria de les empreses de ràdio i televisió emeten també part de la programació per Internet en paral·lel a l'emissió pel mitjà habitual.

3) El cas intermedi: vídeo a la carta

Accedir a un vídeo de YouTube (vídeo a la carta²) sembla que se situa entre tots dos casos, ja que és visualitzar en temps real un material emmagatzemat abans en un servidor remot. El que és crucial en aquest cas és poder assegurar la cadència de recepció, i això s'ha previst abans en la generació del material, que es crea en un format audiovisual fàcil de transportar i que es pot començar a visualitzar des de l'arribada de les primeres dades al client.

⁽²⁾En anglès, *video-on-demand*.

Aquesta previsió del format en l'emmagatzematge fa que generalment aquest cas s'inclogui entre els sistemes de transmissió en temps real, i així ho farem a partir d'aquest moment, encara que amb puntualitzacions. Per això, serveis de vídeo com Netflix, Hulu, Google TV o la majoria de serveis a la carta són sistemes de transmissió en temps real.

4) El receptor

Respecte al receptor (tant pel que fa a la transmissió en temps real com a la lectura de fitxers emmagatzemats), disposa d'un descodificador que no necessita gaires prestacions, ja que els sistemes de codificació són els que fan la feina d'anàlisi i generen la informació mínima necessària per a reconstruir el material audiovisual. Així, hi ha programari domèstic capaç tant de reproduir un Blu-ray en alta definició com de visualitzar una conferència emesa per Internet a l'altra banda del món.

Il·lustració 3. Reproductors de programari de propòsit general



VLC, QuickTime Player, Windows Media Player, etc., són exemples de reproductors de programari de propòsit general executables en entorns domèstics de poques prestacions.

3. La problemàtica de transmissió àudio/vídeo sobre xarxa IP

La transmissió d'informació audiovisual en temps real per una xarxa IP, en què la infraestructura i els protocols de xarxa estan optimitzats per a transportar petits paquets de dades genèriques, obliga a enfrontar-se a una sèrie de problemes, que detallem a continuació:

- 1) La taxa de transmissió de dades (amplada de banda necessària i disponible)
- 2) La cadència del flux enfront de les variacions del retard
- 3) La pèrdua de dades

3.1. La taxa de transmissió de dades (amplada de banda necessària i disponible)

En primer lloc, cal tenir present l'ordre del volum de dades que es requereixen en un flux audiovisual per a detectar un dels problemes a què s'enfronta la transmissió.

Comencem veient el cas d'un mer flux de dades d'àudio, i a continuació el cas d'un flux de vídeo.

El senyal d'àudio digital segueix el format AES/EBU, amb cada mostra de 16 bits i a velocitats de 32.000, 44.100 i 48.000 mostres segons l'amplada de banda analògica que es vulgui representar (fins a 16 kHz, qualitat FM; fins a 22 kHz, qualitat CD; i fins a 24 kHz, qualitat d'estudi professional, respectivament). Així, un senyal estèreo digital AES comporta un flux de dades no comprimides de:

$$\text{Flux d'àudio} = 44.100 \text{ mostres} \times 16 \frac{\text{bits}}{\text{mostra}} \times 2 \text{ canals} = 1,41 \text{ Mbps}$$

Si a més comptem que l'estàndard AES reserva 32 bits per transportar cada mostra i altres dades que hi estan associades (com el codi de temps), resulta:

$$\text{Flux de transport d'àudio} = \text{Flux d'àudio} \times 2 = 2,82 \text{ Mbps}$$

Això comportaria gairebé una connexió de 3 Mbps només per a reproduir l'àudio estèreo mitjançant una xarxa. El cas d'àudio multicanal o envolupant multiplica per 3 o 6 aquesta xifra. Evidentment es comprimeix sense perdre gaire qualitat per a poder distribuir àudio per xarxa.

Exercici 1

Instal·leu el programa de reproducció VLC. Amb aquest programa, obriu un fitxer d'àudio i mentre està en reproducció accediu a *Eines - Informació multimèdia*. La pestanya *Detalls del còdec* indica el format en què està comprimit l'àudio, i la pestanya *Estadística* presenta una sèrie de dades en les quals destaquen sobretot *Taxa de bits d'entrada*, és a dir, els kilobits per segon que el programa llegeix del fitxer d'àudio, i *Taxa de bits de contingut*, que indica quants dels kilobits per segon llegits són realment per a àudio (els altres són de control, qualitat, sincronia, etc.).

↳ Entrada/lectura	
Tamaño de datos del medio	1504 KiB
Tasa de bits de entrada	128 kb/s
Tamaño de datos demuxados	1458 KiB
Tasa de bits del contenido	125 kb/s
Descartados (corruptos)	0

Comproveu la taxa de bits de contingut amb el flux d'àudio (no el flux de transport d'àudio) que hauria de tenir en format AES per a donar-nos una idea de la compressió que s'assoleix amb el format d'àudio del fitxer i de la pèrdua de la qualitat de so que es pugui detectar.

Verifiqueu-ho amb diferents formats (MP3, WMA, WAV, etc.).

Exercici 2

Feu el mateix amb material audiovisual; per exemple, una pel·lícula emmagatzemada en un DVD o Blu-ray domèstic. Per a fer-ho, des del menú *Mitjà - Obriu disc*, seleccioneu la unitat de disc i reproduïu una pel·lícula i accediu a qualsevol moment en el menú *Eines - Informació multimèdia - Detalls del còdec*, on l'apartat *Vídeo* indica el format de codificació (MPEG-2 en DVD, H.264/AVC en Blu-ray), l'amplada i l'alçada de la imatge en píxels i els fotogrames per segon. És fàcil calcular, doncs, que el flux de dades de lectura en bits per segon hauria de ser, sense compressió, de:

$$\text{Flux de dades} = \text{amplada}(\text{píxels}) \times \text{alçada}(\text{píxels}) \times 8 \frac{\text{bits}}{\text{color}} \times 3 \text{ colors} \times \frac{\text{fotogrames}}{\text{segon}}$$

Una xifra que s'acosta als 240 Mbps per a material de qualitat estàndard i a més d'1 Gbps en qualitat d'alta definició. Tots dos valors són difícils de transmetre en les xarxes actuals per a un sol flux audiovisual.

Ara, en la pestanya *Estadística*, es pot veure la *Taxa de bits del contingut*, els bits per segon necessaris per a reproduir correctament el fitxer comprimit:

Entrada/lectura	
Tamaño de datos del medio	9079 KiB
Tasa de bits de entrada	973 kb/s
Tamaño de datos demuxados	6801 KiB
Tasa de bits del contenido	1030 kb/s
Descartados (corruptos)	0

La compressió que ha assolit, en general molt alta, és calculable, com:

$$\text{Factor de compressió} = \frac{\text{Flux de dades sense comprimir (bps)}}{\text{Flux de dades comprimit (bps)}}$$

Exercici 3

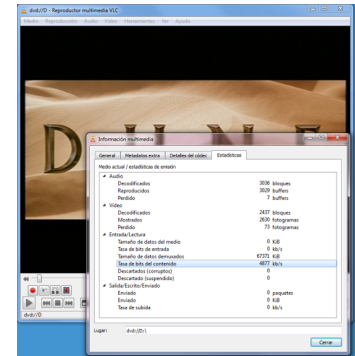
Feu la mateixa operació però en aquest cas amb un fitxer de vídeo que s'hagi baixat d'Internet (WMV, MP4, AVI, etc.). Per a visualitzar-lo amb VLC, seleccioneu el menú *Mitjà - Obriu arxiu* i recolliu les dades que presenta. Calculeu el factor de compressió i compareu-lo amb el del cas anterior. Per quina raó hi ha aquesta diferència?

En segon lloc, no solament és ressenyable el volum de transmissió de dades, sinó que hi ha variacions en la taxa de bits del contingut.

Això passa per la taxa de bits variable de la codificació del vídeo (VBR) típica de DVD i Blu-ray, que és molt interessant perquè manté la qualitat de la codificació i és ideal quan l'objectiu és emmagatzemar les dades ja que redueix la necessitat d'espai, però que genera inquietud en la transmissió: s'ha de dimensionar el canal per a la taxa màxima, o per a una taxa mitjana i acceptar llavors pèrdues de dades?

Hi ha diverses solucions, de les quals donarem una explicació breu ja que aquesta qüestió se surt dels objectius de l'assignatura:

- Posar una memòria intermèdia³ entre la sortida del codificador i l'entrada de la xarxa, de manera que redueixi les variacions del flux VBR i l'acosti a un pseudoflux estable. Tanmateix augmenta el retard global del senyal.



Il·lustració 4. Taxa de bits donada pel programa VLC

Reflexió

Si en comptes de DVD es fa servir Blu-ray, compareu-lo amb un fitxer de vídeo que sigui en alta definició, o la comparativa no tindria sentit.

⁽³⁾En anglès, *buffer*.

- Sumar diferents fluxos VBR abans d'injectar-los a la xarxa, ja que la suma de fluxos variables sempre és un flux més estable; aquesta solució només té sentit si tots els fluxos van del mateix punt al mateix punt de la xarxa.

Com a tercer i últim punt, destaquem que l'amplada de banda disponible en una xarxa IP també sol ser variable.

En aquestes xarxes no es pot fer una reserva d'amplada de banda, sinó que és dinàmica. Si el flux de dades supera l'amplada de banda, es congestionarà la xarxa, perdrà dades i disminuirà la qualitat de recepció. Si el flux és inferior a l'amplada de banda, malbaratarem amplada de banda que es podria dedicar a altres transmissions.

Hi ha diverses solucions per a aquest problema:

- Que el codificador emissor generi un flux de dades⁴ de sortida segons l'amplada de banda lliure a la xarxa a cada moment.
- Que el descodificador receptor triï un nivell de qualitat del flux de dades que rep. Això només és possible si el flux de dades està codificat per capes o és escalable (per exemple, MPEG-4).

⁽⁴⁾En anglès, *stream*.

3.2. La cadència del flux enfront de les variacions del retard

Tota xarxa genera un retard entre l'emissor i el receptor. Tanmateix, en les xarxes IP aquest retard és variable ja que, com veurem a continuació, el camí de les dades a través d'aquesta xarxa pot anar canviant en plena transmissió.

Tant àudio com vídeo són informacions que requereixen una cadència constant d'enviament i recepció de dades per a poder reconstruir l'efecte de moviment en vídeo i la intel·ligibilitat en àudio: quan comença la reproducció s'ha de mantenir el flux presentant una imatge i una porció d'àudio cada cert temps. I si el cas és una comunicació audiovisual en tots dos sentits, o interactiva, la situació és més complicada.

Així, un retard de 150 ms en l'àudio no és detectat per l'oïda humana, i per sobre de 400 ms el fa intel·ligible. En vídeo, el retard comporta una congelació de la imatge o de part d'aquesta imatge. I retards diferents entre àudio i vídeo són notables si el vídeo antecedeix l'àudio en 100 ms, o el segueix més enllà de 20 ms.

Totes aquestes variacions del retard⁵ obliguen a afegir intel·ligència i memòria en els equips, sobretot en l'equip receptor, cosa que n'augmenta el cost: s'emmagatzemen les dades en el receptor en una cua de memòria abans de reproduir-les a l'usuari, de manera que no es perceben variacions en l'arribada d'aquestes dades, encara que es genera un retard total proporcional al volum de la cua. Malgrat tot, les dades que arribin massa tard (és a dir, més enllà del moment en què s'haurien d'haver mostrat) es descarten i representen errors.

⁽⁵⁾En anglès, *jitter*.

3.3. La pèrdua de dades

Com que tota xarxa és susceptible de pèrdua de dades (de paquets per caiguda d'un node de la xarxa o d'un enllaç entre nodes, o malmetement del contingut del paquet per ràfegues de soroll), hi ha mecanismes per a protegir-les d'aquestes situacions fins on és possible.

Bàsicament hi ha quatre tècniques de control d'errors, agrupades en dues famílies:

1) Control d'errors per codificació de canal. Per exemple:

- **Retransmissió de paquets (ARQ⁶):** el receptor detecta els paquets que falten segons un criteri (ordre, importància) i els reclama per retransmetre'ls.
- **Tècniques de FEC⁷:** el receptor o qualsevol dels nodes que travessa pel seu camí és capaç de detectar la majoria de bits malmesos que té el paquet i reconstruir-lo correctament.

⁽⁶⁾La sigla *RQ* correspon a l'expressió anglesa *automatic repeat request*.

⁽⁷⁾La sigla *FEC* correspon a *forward error correction*.

En el cas del mecanisme ARQ, gràcies a un comptador que marca de manera única i seqüencial cada paquet en emissió, el receptor detecta els que falten i els reclama. Els paquets reclamats es reenvien sempre que se segueixi un criteri: el criteri més simple és que s'estimi que pugui arribar al receptor abans de l'instant en què s'han de presentar al destinatari; una altra possibilitat és reclamar els paquets perduts no per l'ordre sinó per la importància que tenen en la descodificació del material audiovisual.

I és que hi ha bits més importants que d'altres; per exemple, en la família MPEG, els que pertanyen a les imatges I sempre són més importants que els que pertanyen a les imatges P, i els de les imatges P ho són més que els de les imatges B, que fins i tot es poden descartar per complet. Un altre exemple és en una codificació escalable: el flux principal ha de tenir més importància que el primer flux de millora, i aquest primer flux de millora n'ha de tenir més que el segon i els subsegüents fluxos de millora del contingut.

Els controls d'errors per codificació de canal es poden combinar, de manera que s'intenta recuperar el màxim nombre de paquets malmesos mitjançant FEC, i els paquets que no s'aconsegueixen recuperar o que ni tan sols arriben a

la destinació es retransmeten mitjançant ARQ. Una taula resum com la següent representa la idea de la seva aplicació combinada en la transmissió d'un flux MPEG:

	Imatge I	Imatge P	Imatge B
FEC	Màxima	Mitjana	Mínima
ARQ	Màxima	Mitjana	Descartable

I en la transmissió d'un flux escalable:

	Flux base	Flux millora 1	Flux millora 2
FEC	Màxim	Mitjà	Mínim
ARQ	Màxim	Mitjà	Descartable

2) **Control d'errors per codificació de font.** Per exemple:

- **Ocultació de l'error⁸:** s'estima la informació perduda per a poder-la així ocultar o dissimular en la presentació. La mateixa correlació o redundància que té el vídeo i l'àudio (i que tan bé ha servit per a definir estratègies de codificació) ara ofereix una manera senzilla de valorar aquesta informació perduda.
- **Codificació de vídeo resistent als errors⁹:** s'apliquen tècniques de codificació que augmenten la resistència davant errors específics; per exemple, la codificació de Huffman i la codificació *run-length*.

⁽⁸⁾En anglès, *error concealment*.

⁽⁹⁾En anglès, *error resilient video coding*.

Els errors que afecten aquest tipus de control habitualment són dos:

a) La **pèrdua de la sincronització del flux de bits**, de manera que el descodificador no sap quins bits corresponen a quins paràmetres. És molt greu si s'usa un codi de longitud variable (com Huffman o *run-length*) ja que l'error en un sol bit fa que el descodificador dedueixi una longitud incorrecta de la paraula, de manera que interpreta també incorrectament les següents fins que s'arribi a alguna resincronització.

b) Els **estats incorrectes i la propagació d'errors**, que sobretot afecten els descodificadors que treballen sobre codificació predictiva (per exemple, DPCM sobre àudio).

4. Transmissió de vídeo en una xarxa IP

Les tècniques per a transmetre vídeo en les xarxes IP es coneixen normalment com a **mecanismes de *streaming***, que es pot definir com la transmissió en temps real (en viu) d'àudio i vídeo sobre una xarxa.

Abans de l'aparició de les tècniques de transmissió en temps real, les aplicacions multimèdia empraven Internet sobretot per a transferir arxius. Així, quan s'havien baixat del tot els continguts, es podien reproduir. Avui dia, usant transmissió en temps real, es pot visualitzar un contingut multimèdia a mesura que es transfereix aquest contingut.

Com hem vist en els estàndards digitals anteriors, els fluxos de dades d'àudio i vídeo s'empaqueten de manera que cadascun forma un flux elemental (ES¹⁰) que a més guarda informació de sincronia entre uns i altres. A continuació es combinen (almenys un de vídeo i un altre d'àudio) en un flux elemental empaquetat (PES¹¹), cosa que possibilita que es combinin més.

⁽¹⁰⁾La sigla *ES* correspon a *elementary stream*.

⁽¹¹⁾La sigla *PES* correspon a *packetized elementary stream*.

A partir d'aquest moment podem fer dues coses: emmagatzemar aquest contingut en un contenidor per reproduir-lo, o transmetre'l (en el nostre cas, per una xarxa IP). En altres apartats hem vist per sobre com es condicionen els fluxos de dades per a emmagatzemar-los; ara veurem com els hem d'enviar per la xarxa.

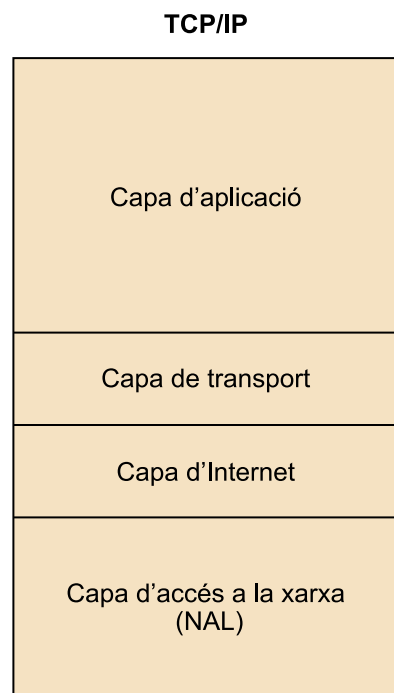
Hi ha moltes tècniques de transmissió en temps real segons la realitat particular del sistema de vídeo digital. En aquest apartat veurem les més importants en el context de les xarxes basades en IP, com 3G, IPTV o Internet, i a continuació farem una revisió dels protocols de les xarxes IP i dels protocols de transmissió en temps real.

4.1. Les xarxes IP i els seus protocols

Una xarxa es pot analitzar a diferents nivells. Per exemple, es pot estudiar des d'un punt de vista elèctric, mesurant les tensions dels senyals que hi circulen i els tipus de connectors que s'hi necessiten. Aquest punt de vista és l'anomenat **nivell físic** o **nivell d'accés a la xarxa** (NAL¹²), més propi d'investigadors i instal·ladors.

⁽¹²⁾La sigla *NAL* correspon a l'expressió anglesa *nivell d'accés a la xarxa*.

Il·lustració 5. Arquitectura de xarxa TCP/IP



Un altre punt de vista és el **nivell d'enllaç**, que estudia la identificació de les targetes de xarxa i el protocol d'intercanvi de bits entre aquestes targetes. Aquest nivell també és més propi de desenvolupadors i programadors que no pas nostre, ja que som en un nivell d'ús més abstracte.

Els nivells que ens interessin són els dos següents: el **nivell de xarxa**, en el qual parlem de dades que circulen per aquesta xarxa des d'un equip en direcció cap a un altre equip o uns altres equips, i el **nivell de transport**, que consisteix en els mecanismes per a assegurar que les dades arriben correctament, i si no ho fan, es demana reenviar-les o no. En aquesta quarta i última capa ja hi tindriem les aplicacions informàtiques que usen la Xarxa. En aquest apartat només es fa esment d'aquestes dues últimes capes (xarxa i transport) de les xarxes IP, amb la finalitat d'entendre els mecanismes de transmissió en temps real que presentem tot seguit.

4.1.1. Protocols de xarxa (IP)

Les xarxes IP¹³ són un territori inestable i sempre canviant de ruta des d'un punt fins a un altre. Perquè un paquet de dades arribi a la seva destinació, ha de tenir indicada aquesta destinació en cadascun dels paquets que formen les dades que s'envien.

⁽¹³⁾ La sigla *IP* correspon a *Internet protocol*.

El protocol IP és un protocol robust que no espera que hi hagi un camí assegurat i estable entre origen i destinació, sinó que les dades es divideixen en paquets que s'envien de manera consecutiva i cadascun "navega" en una xarxa plena de paquets fins a arribar a la destinació que té marcada.

Dit tècnicament, el protocol IP és no orientat a connexió per a la comunicació mitjançant una xarxa de paquets commutats. És el protocol d'Internet i dels sistemes d'IPTV.

Aquests paquets són enviats amb informació extra per a poder dur a terme accions relatives a garantir la seva pròpia integritat, però no proporciona cap mena de seguretat que les dades arribaran realment a la destinació (d'això se n'encarrega el protocol superior, el protocol de transport).

4.1.2. Protocols de transport (UDP, TCP i SCTP)

Per sobre d'aquest "carter" que és el protocol IP, sempre hi ha un protocol de transport que té la responsabilitat d'assegurar la transferència sense errors de dades entre l'emissor i el receptor. Això no obstant, ha de ser àgil o la seva pròpia acció pot alentir la celeritat amb què es reben aquestes dades. Els bits d'aquest protocol van dins del paquet de xarxa.

Depenent del tipus de servei que vulguem, podem triar un dels tres protocols de transport següents:

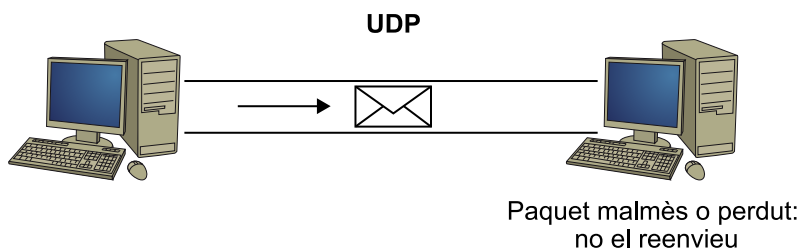
a) Protocol UDP

UDP¹⁴ és un protocol lleuger que no dona garantia sobre la pèrdua de paquets ni la recepció per duplicat de paquets. Només disposa de mecanismes de comprovació de dades; si cal gestionar els errors en la transmissió, s'haurà de fer a la banda del destinatari (en l'aplicació que els rebi).

⁽¹⁴⁾La sigla UDP correspon a *user datagram protocol*.

És el protocol de transport natural per a transmissió en temps real de vídeo i àudio.

Il·lustració 6. Protocol UDP



En UDP, una pèrdua, un error o una duplicació de paquet no genera cap petició de reenviament per part del destinatari.

La raó per la qual tota videoconferència que es preui, tota sessió de Spotify o tot web pirata de sèries de televisió no estrenades a Europa faci servir aquest protocol és que consumeix pocs bits del total de bits de dades i va de pressa a inicialitzar-se (ja que no fa gaire gestió). Certament no és tan segur com

d'altres, però en transmissió en temps real es considera inútil una retransmissió de la informació perduda, ja que no arribaria a temps (l'*streaming* presenta restriccions de temps real).

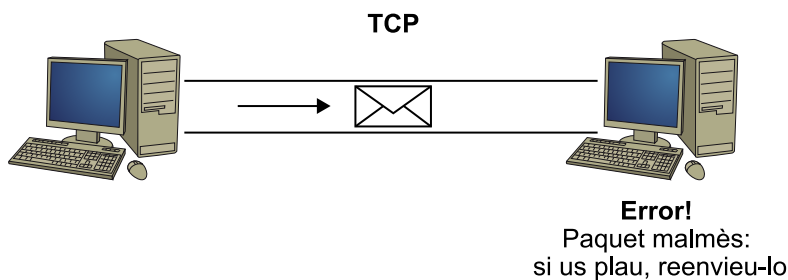
b) Protocol TCP

El protocol TCP¹⁵ ens assegura que els paquets arribaran i ho faran en ordre.

⁽¹⁵⁾La sigla *TCP* correspon a *transmission control protocol*.

Això s'aconsegueix mitjançant un mecanisme en què, quan el destinatari no rep la confirmació del paquet durant cert temps, aquest paquet es reenvia. A causa d'aquest comportament es poden detectar paquets perduts i es pot demanar que es retransmetin. En el cas de transmissió de flux de dades, quan es perden paquets la retransmissió augmenta el retard i el consum de l'amplada de banda, cosa que pot provocar que es buidi la memòria intermèdia del reproductor (i per tant que s'interrompi la reproducció del flux de dades).

Il·lustració 7. Protocol TCP



En TCP s'assegura la recepció correcta de tots els paquets.

En xarxes d'IPTV (Imagenio, per exemple), en les quals hi ha una xarxa que hi està dedicada en exclusiva i que assegura una gran qualitat de servei i no hi ha congestió, el mecanisme de transmissió en temps real continua essent el tradicional basat en UDP. En canvi, en xarxes d'ús general en què no hi ha mecanismes per assegurar la qualitat de servei, com Internet, es pot optar per fer la transmissió en temps real sobre TCP; això no obstant, aquest protocol té dos problemes: en primer lloc, perquè el trànsit de paquets retransmesos afecta la cadència i qualitat del flux de dades, i en segon lloc, perquè pel mateix funcionament intrínsec del protocol TCP es generen oscil·lacions de la taxa mitjana de transmissió en detriment periòdic de la transmissió en temps real.

c) Protocol SCTP

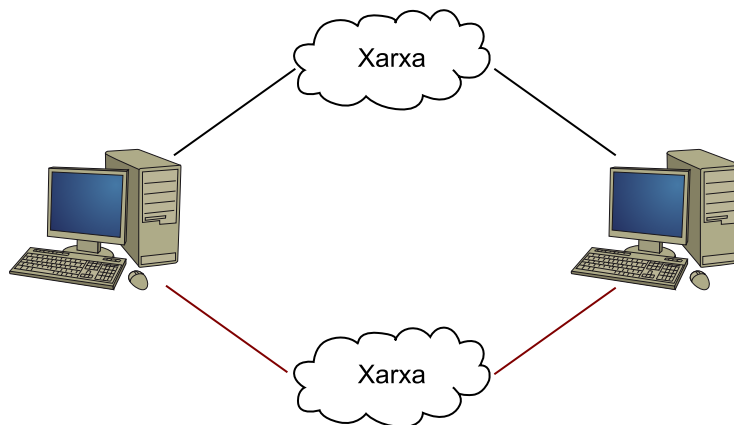
El protocol de transport SCTP¹⁶ és una solució de compromís entre els tradicionals UDP i TCP. Malgrat que està del tot estandarditzat, ha tingut una difusió lenta. Té una potencial utilitat en la transmissió en temps real de vídeo en xarxes sense fil i altres contextos amb grans pèrdues puntuals de dades.

⁽¹⁶⁾ La sigla *SCTP* correspon a *stream control transmission protocol*.

Aquest protocol es basa en datagrames, però a diferència d'UDP, té registre de números de seqüència amb els quals fa controls de paquets fora d'ordre i perduts, que seran retransmesos.

A diferència de TCP, aquest protocol permet diferents adreces IP dins d'una mateixa connexió (origen SCTP - destinació SCTP). O sigui que cadascun dels extrems de connexió pot tenir diferents IP (i mantenir el port de connexió) i els paquets s'envien indistintament per qualsevol IP i a qualsevol IP corresponent a cada extrem SCTP.

Il·lustració 8. Protocol SCTP



SCTP possibilita diferents adreces IP per un mateix port.

El protocol SCTP també suporta el trànsit sobre la mateixa connexió de diferents tipus de paquets i manté un nombre de seqüència per a cadascun d'aquests paquets.

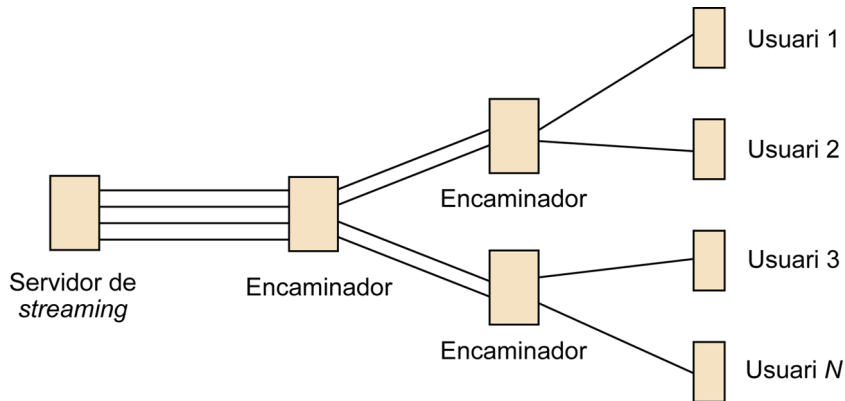
4.2. Tècniques de multidestinació sobre xarxes IP

En les xarxes IP hi ha diferents tècniques per a difondre la informació a diversos destinataris simultàniament:

- **Unidestinació**¹⁷: l'enviament d'un punt a un altre en una xarxa s'anomena *unidestinació*. El mecanisme que solen fer servir totes les aplicacions té un nombre N màxim d'usuaris limitat.

⁽¹⁷⁾En anglès, *unicast*.

Il·lustració 9. Unidestinació

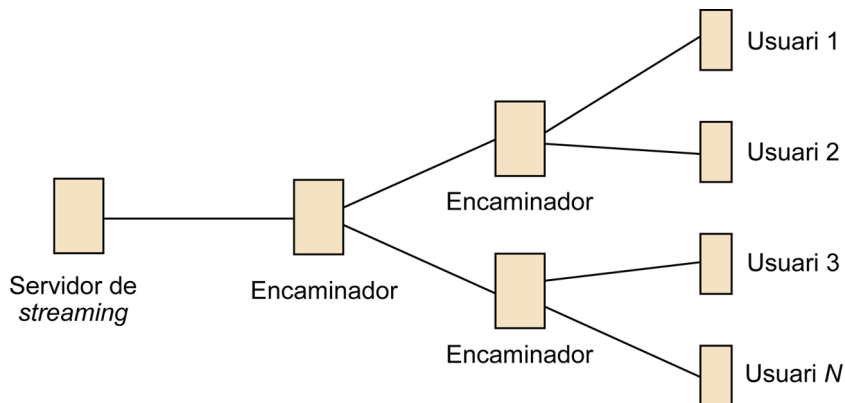


N usuaris visualitzen simultàniament un programa de televisió per Internet; en unidestinació el servidor ha d'enviar el flux de dades per a cada usuari.

- **Multidestinació**¹⁸: l'enviament de la informació a moltes destinacions simultàniament amb l'ajuda de centres de difusió preparats que, si estan enviant el flux de dades a un usuari, quan reben una altra petició dupliquen les dades del flux de dades. El servidor queda molt més descarregat, però obliga a tenir encaminadors capacitats per a aquest protocol.

⁽¹⁸⁾En anglès, *multicast*.

Il·lustració 10. Multidestinació



En multidestinació, el servidor envia d'entrada un flux de dades al primer usuari. Els altres en reben el flux mitjançant reenviaments dels centres de difusió, que s'adonen que tenen més d'un usuari amb la mateixa petició.

- **Broadcast**: l'enviament a tots els nodes en una xarxa s'anomena *broadcast*. Lògicament només es pot aplicar en contextos reduïts, per exemple, per al descobriment de recursos en una xarxa local (LAN¹⁹).

⁽¹⁹⁾La sigla anglesa *LAN* correspon a *xarxa d'àrea local*.

La tècnica de multidestinació és fàcil de descriure i complicada d'implementar.

La **multidestinació** ha d'usar l'estratègia més eficient per a enviar els missatges sobre cada usuari de la xarxa (amb la finalitat de disminuir el consum d'amplada de banda), emprant cada enllaç com a màxim una vegada perquè es difongui cada paquet i creant còpies en els centres de difusió quan els enllaços en les destinacions es divideixen. Això s'acostuma a aconseguir mitjançant l'ús d'**arbres de cobriment**, en sorgir un protocol per a mantenir aquest arbre a través dels centres de difusió.

Des del punt de vista de l'emissor i dels receptors, el procés és senzill: l'emissor envia la informació una sola vegada a una adreça IP especial (de manera idèntica a una unidestinació); els receptors se subscriuen al grup de multidestinació associat a aquesta IP especial i reben una còpia de tot el que l'emissor ha enviat. Per tant, abans d'enviar la informació en multidestinació, s'han d'establir una sèrie de paràmetres.

Per a poder-la rebre, cal establir el que s'anomena *grup de multidestinació*. En IPv4, les IP especials pertanyen al rang 224.0.0.0 a 239.255.255.255 i el protocol usat per a manejar i associar-se als grups de multidestinació s'anomena *protocol d'Internet per a la gestió de grups* (*Internet group management protocol* o IGMP).

Depenent de la informació, es tria el mecanisme ideal per a difondre-la.

En IPTV, la transmissió de canals en viu s'acostuma a fer mitjançant multidestinació (*multicast*), ja que s'espera que els vegin una audiència nombrosa – l'enviament individual a cada espectador (*unicast*) redundaria en un dimensionament inadmissible de la xarxa.

En canvi, el **vídeo a la carta** (VoD²⁰), per naturalesa, és punt a punt i per tant se sol fer servir unidestinació. Però això no és cap regla, ni tan sols en el context d'IPTV. Per exemple:

- Hi ha tècniques per a reduir el temps de zàping entre els canals en viu (enviats per multidestinació) que impliquen l'enviament unidestinació d'una ràfega inicial de gran amplada de banda per a omplir ràpidament la memòria intermèdia del receptor i reproduir de seguida després del canvi de canal.
- D'altra banda, en els sistemes de VoD, quan un contingut a la carta és molt sol·licitat, es pot reproduir en forma d'nVoD²¹, que implica l'enviament per multidestinació i que el client hagi de fer una espera inicial d'uns quants minuts abans no arriba la multidestinació següent. Una tècnica una mica més sofisticada implica l'enviament d'una bona porció del VoD per unidestinació i l'emmagatzematge en la memòria intermèdia de la trans-

⁽²⁰⁾ La sigla VoD correspon a *video-on-demand*.

⁽²¹⁾ La sigla nVoD es refereix a l'expressió *near VoD*.

missió multidestinació antiga més pròxima en el temps. Això evita l'espera inicial del sistema nVoD però exigeix als receptors disposar d'un espai de memòria d'emmagatzematge molt més gran. Aquest tipus de mecanisme mixt encara s'ha d'investigar.

Cal concloure que en els sistemes de vídeo digital especialitzats com IPTV, l'ús eficient de les tècniques de difusió redunda en una més bona qualitat d'experiència i aprofitament dels recursos de la xarxa.

La situació en el context d'**Internet** és molt diferent. En els començaments d'Internet, la multidestinació no estava estandarditzada i era difícil que els sistemes operatius i el maquinari dedicat de l'època hi funcionessin. Avui, per raons d'indole comercial es manté aquesta restricció sobre Internet; per tant, no és compatible amb la multidestinació (i per raons lògiques d'escala, tampoc amb *broadcast*). Una esperança per a la compatibilitat de multidestinació a Internet és amb el protocol IPv6, la nova versió del protocol IP, amb el qual funciona nativament.

Si Internet funcionés amb multidestinació, els sistemes dedicats d'IPTV presentarien una forta competència, que seria una oferta de contingut global per a un públic global (per aquest motiu els proveïdors de connectivitat no veuen atractiu oferir aquests beneficis als proveïdors de serveis d'Internet). Els actuals sistemes de vídeo a Internet fan, segons les possibilitats que tenen, una "multidestinació en capa d'aplicació" mitjançant el desplegament global dels centres de dades²² i la rèplica intel·ligent de la informació.

Malgrat que són tècniques més eficients de multidestinació, en el tram més compromès subsisteix el problema, és a dir, en la xarxa d'accés, on aquestes xarxes solament poden fer distribució unidestinació (YouTube, Google Vídeos, Netflix, etc.).

4.3. Els protocols de transmissió en temps real

Hi ha una tercera categoria de protocols dissenyats específicament per a transmissió en temps real, coneguts com a *protocols de temps real*, que intenten assegurar l'enviament de fluxos de dades amb una cadència assegurada, tal com necessita el material audiovisual.

4.3.1. Protocols de temps real (RTP, RTCP i RTSP)

Quan hem parlat al principi del mòdul dels problemes inherents a les variacions de l'amplada de banda de les xarxes IP, una de les dues solucions era ajustar la taxa de codificació amb la capacitat real a cada moment de la xarxa perquè no se saturés i perdés dades. Aquesta solució es basa en aquests protocols, mitjançant els quals el receptor avisa el codificador d'origen sobre l'amplada de banda efectiva en cada instant.



Il·lustració 11. Unidestinació i multidestinació a Internet
Tant les tècniques de multidestinació com les d'unidestinació avui dia estan restringides a Internet.

⁽²²⁾En anglès, *data centers*.

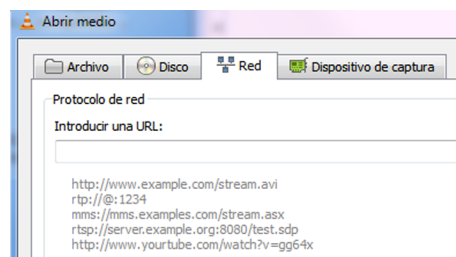
a) Protocol RTP

RTP²³ és un protocol de transport desenvolupat per a transmissió en temps real, normalment muntat sobre UDP. El transport d'RTP sobre TCP també està definit; malgrat això, no es fa servir tant perquè presenta una latència i sobrecàrrega més gran que UDP (la qual cosa és un desavantatge per a aplicacions de temps real). Aquest protocol inclou dades extres no presents en TCP, com marcatge de temps (guarda en cada paquet l'hora en què es transmet a partir d'un rellotge de referència, cosa que facilita la sincronització) i nombre de seqüència (per si arriba desordenat o si es perd), i això contribueix a facilitar el transport en forma contínua. També hi ha dades de control que permeten al servidor fer la transmissió en temps real a una taxa de flux definida i correcta.

⁽²³⁾La sigla RTP correspon a *real-time transport protocol*.

Lamentablement, RTP no té informació de sincronització ni mecanismes nadius per a recuperar fallades. Hi ha una sèrie de còdecs que estan estandarditzats per a enviar-los sobre RTP, uns quants dins de la línia MPEG-2.

Il·lustració 12. El programa VideoLAN



En indicar en l'aplicació l'adreça de la font de transmissió en temps real, cal indicar el protocol de temps real en què es transmet.

b) Protocol RTCP

RTCP²⁴ és un protocol usat en conjunció amb RTP per a rebre informes estadístics. Permet, per exemple, detectar fallades en l'arbre de distribució de clients de multidestinació, del nombre de paquets perduts i estadístiques de fluctuacions (*jitter*).

⁽²⁴⁾La sigla RTCP correspon a *real-time transport control protocol*.

Recepció d'informes estadístics

Hi ha informes enviats per l'emissor i informes enviats pel receptor. Per exemple, els de l'emissor solen ser el volum de quadres enviats o quantitat de bytes, i els informes del receptor presenten els quadres perduts o la taxa de quadres lliurats.

Una característica particular que té aquest protocol és que, per a amortitzar la sobrecàrrega²⁵ que comporta consumir bits per al protocol en comptes de fer-ho per a dades, es poden reunir diversos missatges RTCP i enviar-los en un missatge RTCP compost. Els paquets han d'estar compostos almenys per un missatge de receptor o emissor i el nom del participant, i han de ser enviats periòdicament sense arribar a consumir el 5% de l'amplada de banda de la sessió. Els paquets RTCP es transporten sobre UDP.

⁽²⁵⁾En anglès, *overhead*.

RTCP no permet ni encriptació, ni autenticació ni autorització. Una actualització amb aquestes funcionalitats és l'estàndard SRTCP⁽²⁶⁾.

⁽²⁶⁾La sigla *SRTCP* correspon a *secure real-time transport protocol*.

c) Protocol RTSP

RTSP⁽²⁷⁾ fa control sobre dades multimèdia de temps real i proporciona la possibilitat d'interactivitat amb el reproductor, de manera semblant a un vídeo reproductor domèstic. L'RTSP permet reproduir, aturar, avançar, i més accions. També pot reaccionar a congestions a la xarxa i reduir l'amplada de banda.

⁽²⁷⁾La sigla *RTSP* correspon a *real-time streaming protocol*.

Les ordres estandarditzades en RTSP són:

- *Describe*: amb aquesta ordre es rep una descripció del recurs, una llista dels fluxos de dades necessaris.
- *Setup*: s'indiquen els paràmetres de configuració del flux de dades.
- *Play*: executa la reproducció del flux de dades especificat.
- *Pause*: atura la reproducció del flux de dades indicat, reiniciable amb l'ordre *Play*.
- *Teardown*: atura i allibera els recursos usats, i finalitza el flux de dades.

L'RTSP es va inspirar en el protocol web (HTTP 1.1) però amb la millora que es pot mantenir l'estat de la connexió (HTTP no manté l'estat) i que tots dos (client i servidor) poden executar ordres. RTSP és compatible amb RTP com a protocol de transport.

Una de les utilitats que té és que proporciona una manera inicial d'escollir el canal de distribució òptim cap al client. Per exemple, alguns clients poden tenir filtrats en el tallafoc els paquets UDP, de manera que el servidor de transmissió en temps real hauria de proveir la possibilitat d'escollir entre diferents protocols de transport com UDP, TCP o UDP de multidestinació.

4.3.2. Altres protocols de transmissió en temps real

a) Protocol RDT

RDT⁽²⁸⁾ és un protocol propietari per a transmissió d'àudio i vídeo. El va desenvolupar RealNetworks el 1995. Igual que RTP, treballava en conjunt amb RTSP per al control de la transmissió en temps real, i com que sembla que no té gaires més avantatges que el seu equivalent estàndard RTP, se n'ha posat en dubte la continuïtat.

⁽²⁸⁾La sigla *RDT* correspon a *real data transport*.

b) Protocol HTTP

Malgrat que s'ha dissenyat amb finalitats completament diferents (és el protocol pel qual podem navegar per web), HTTP²⁹ és usat per a la transmissió en temps real d'àudio i vídeo per diverses raons. La principal és que la majoria de les connexions a Internet estan protegides per un tallafoc o un servidor intermediari³⁰, per la qual cosa la manera més senzilla de fer transmissió en temps real d'un contingut en aquest context és usant HTTP.

⁽²⁹⁾La sigla *HTTP* correspon a *hypertext transfer protocol*.

⁽³⁰⁾En anglès, *proxy*.

Lògicament té més sobrecàrrega³¹ que la família de protocols de temps real (RTP, RTCP i RTSP), de manera que no està ben considerat en molts contextos, com per exemple en el grup MPEG. No obstant això, com ja hem comentat, és una opció vàlida per a Internet gràcies a l'existència de tallafocs o servidors intermediaris i al fet que representa una manera simple d'afrontar els errors per pèrdues a la xarxa (mitjançant retransmissió).

⁽³¹⁾En anglès, *overhead*.

La transmissió en temps real sobre HTTP no està gaire estandarditzada malgrat que és, per exemple, la més usada per les ràdios a Internet.

Ràdios a Internet

Podeu veure un directori d'emissores de ràdio per Internet al lloc web Showtcast Radio Internet.

Avui dia, sobre HTTP es poden transmetre diversos tipus de formats de vídeo. A més d'ES crus (com els acceptats per RTP), sobre HTTP es pot enviar MPEG-TS, MPEG-PS, OGG, ASF i altres formats.

Recordeu que RTP es pot muntar sobre TCP (a més de la manera tradicional sobre UDP), de manera que pot recuperar fallades en errors de la xarxa.

c) Protocol MMS (*Microsoft media services*)

MMS³² és un protocol propietari per a la transmissió d'àudio i vídeo. Desenvolupat per Microsoft, es va abandonar oficialment el 2003 i s'hi va deixar de proporcionar manteniment el 2008.

⁽³²⁾La sigla *MMS* correspon a *Microsoft media services*.

Operava sobre TCP, UDP o HTTP, i client i servidor negociaven quin havien de triar d'aquests tres, o unívocament el servidor segons l'estat de la xarxa. Si el client no podia negociar una bona connexió usant MSS sobre UDP (abreujat *MMSU*), ho intentava amb MMS sobre TCP (abreujat *MMST*). Si això fallava, feia servir una versió modificada d'HTTP per a establir la connexió (abreujat *MMSH*). En cada pas anava creixent la sobrecàrrega i n'augmentava la ineficiència.

4.4. El protocol MPEG-2 TS

MPEG TS³³ és un protocol que ofereix un mecanisme per a multiplexar (és a dir, combinar en un sol flux de dades) els fluxos de dades d'àudio, vídeo i dades, per a transmetre'ls així per una xarxa. Es fa servir molt en els sistemes de vídeo digital de televisió per cable i TDT.

⁽³³⁾TS és la sigla de *transport stream*.

Com hem dit en l'apartat sobre MPEG-2, un ES³⁴ és bàsicament la sortida del codificador, i n'hi ha de dos tipus: els fluxos de dades de vídeo (VES) i els d'àudio (AES). Per a manejar els diferents ES, es divideixen en paquets de diferent mida, segons les característiques de l'aplicació i del descodificador. Del procés de partició en paquets de l'ES se'n diu *empaquetatge* i un ES empaquetat és un PES³⁵.

⁽³⁴⁾ES és la sigla de *elementary stream*.

⁽³⁵⁾PES és la sigla de *packetized elementary stream*.

Quan es fa servir MPEG-TS, es poden transmetre diversos PES de manera conjunta en un mateix flux de dades TS. D'aquest procés se'n diu **multiplexació**.

En la majoria de les aplicacions es necessita àudio i vídeo; per tant, es requereix la multiplexació d'almenys un PES d'àudio i un PES de vídeo. Els PES transmesos dins d'un TS s'agrupen en programes. Normalment un flux de dades TS transmet diversos canals de televisió simultàniament; cadascun és un programa format almenys per un PES de vídeo i un PES d'àudio. A més dels PES, un flux de dades TS transmet informació de control en format de taules; per exemple, en aquestes taules es pot enviar informació de la programació present en el flux de dades.

L'MPEG-TS no proporciona només una manera adequada de fer la multiplexació dels diferents ES, sinó que també ataca el problema de recrear el rellotge de la font en cadascun dels receptors, per aconseguir així una descodificació i un sincronisme correctes de l'àudio i del vídeo.

4.4.1. Paquets i taules del TS

Els paquets TS tenen una mida fixa de 188 bytes. Un flux TS està compost per un o més d'un programa. Per a identificar quin PES pertany a cada programa, el TS envia periòdicament taules amb aquesta informació.

Dins de la capçalera dels paquets TS hi ha un camp anomenat *PID*³⁶, que identifica el contingut específic d'aquest paquet TS. El contingut d'un paquet TS pot ser:

⁽³⁶⁾La sigla PID correspon a *packet identifier*.

- part d'un PES d'àudio o vídeo, o

- informació de control, conegut com a *program specific information* (PSI).

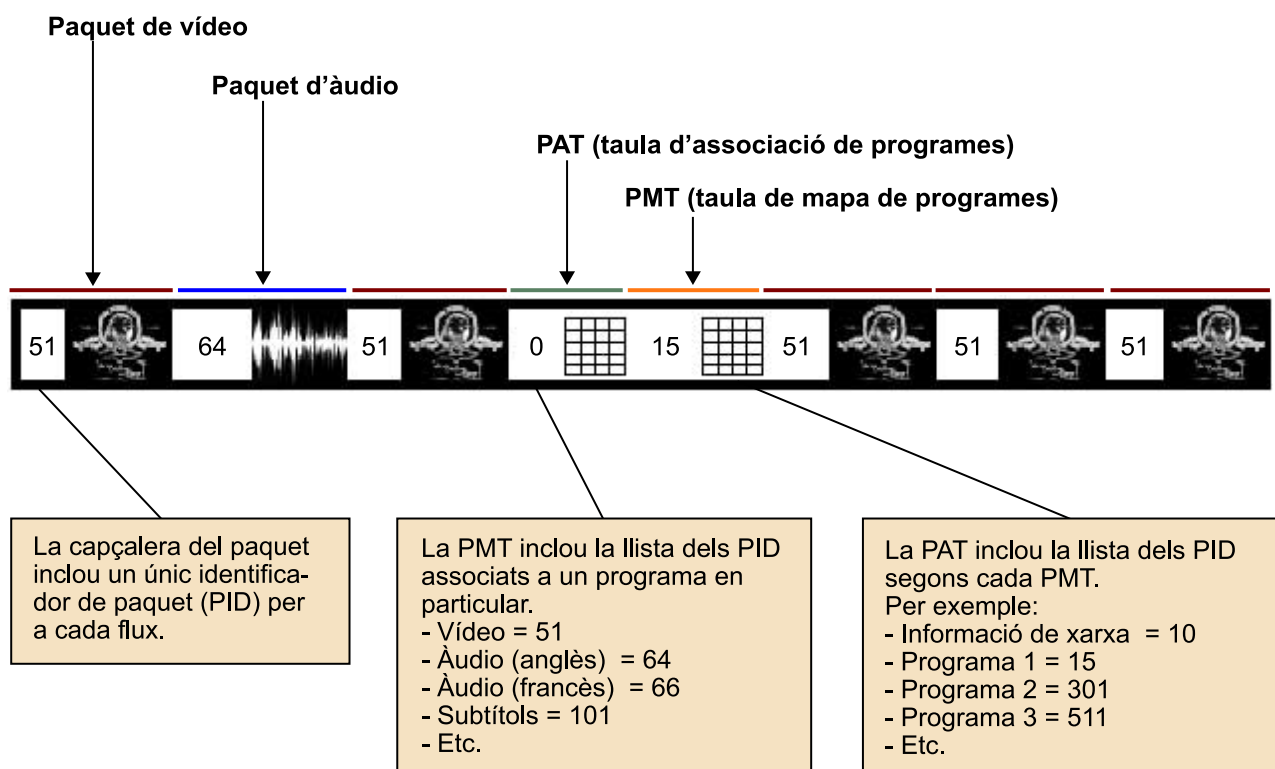
El PSI està compost per quatre taules diferents:

- la taula d'associació de programa (*program association table* o PAT),
- la taula de mapa de programa (*program map table* o PMT),
- la taula d'accés condicional (*conditional access table* o CAT) i
- la taula d'informació de xarxa (*network information table* o NIT).

Dins d'aquestes taules hi ha tota la informació necessària per a extreure de manera correcta cada flux de vídeo, àudio i dades i seleccionar així els diferents programes en el receptor. Les taules PAT i PMT sempre són presents dins del flux de dades, mentre que les altres dues depenen del flux de dades que es transmeti.

En la taula PAT (identificades amb PID = 0x0000), s'associa el número de programa amb el PID en què vindrà la PMT per a aquest programa. Per exemple, en la il·lustració 13 es mostra que en els paquets amb PID número 15 es transmetran els paquets corresponents a la PMT del programa 1. Al seu torn, dins de la taula PMT, entre altres coses, hi ha quin PID –i per tant, quin ES– està associat a cada programa. En aquest exemple es defineix que el PID número 64 és un ES d'àudio corresponent al programa número 1.

Il·lustració 13. Exemple de taules PAT i PMT



5. Les tecnologies de publicació

Arribat el moment en què s'ha d'escollir la manera de publicar el material audiovisual per xarxa, cal plantejar-se quina tecnologia s'ha de fer servir i el tipus de format del material. Al llarg dels anys han millorat en paral·lel tant les tecnologies de publicació a la xarxa com els formats contenidors de vídeo que agiliten aquesta publicació. Totes dues línies de desenvolupament estan molt relacionades.

5.1. Tecnologies de publicació

De tecnologies de publicació audiovisual a Internet en podem definir quatre, que presentem en ordre cronològic (coincident amb una complexitat més gran):

- 1) Vídeo incrustat
- 2) Baixada progressiva
- 3) Transmissió en temps real de vídeo
- 4) P2P

5.1.1. Vídeo incrustat

Aquest model de plataforma de publicació bàsica va ser la primera aproximació a la integració d'àudio i vídeo a la web. Requeria un servidor web, de manera que feia ús del protocol HTTP clàssic, senzill i amb mecanisme de confirmació de les dades enviades pel client, pensat per a transmetre fitxers petits a un conjunt de clients alhora. El fitxer de vídeo “s'incrustava” o integrava en una pàgina web en HTML com qualsevol altre fitxer mitjançant les instruccions *HREF*, *EMBED*, *BGSOUND*, etc.

El client sol·licitava el fitxer de vídeo, que baixava temporalment a la memòria cau³⁷ del programa navegador que feia servir, exactament igual que la de qualsevol altre tipus de fitxer que es pugui publicar en una pàgina web (imatges, .pdf, .doc, etc.).

⁽³⁷⁾En anglès, *cache memory*.

La majoria de formats d'arxiu només permeten obrir el contingut quan s'ha baixat del tot: qualsevol document d'Office, els antics Adobe PDF (versió 5 o una d'anterior), etc. No obstant això, n'hi ha d'altres que sí que possibiliten

iniciar la visualització després de començar la baixada i accedir a un mínim del contingut. Per exemple, les imatges JPEG enregistrades en format progressiu i els documents PDF moderns.

En aquells temps, els formats contenidors de vídeo no estaven preparats per a ser reproduïts progressivament, de manera que es generava una espera directament proporcional a la durada del material. Una conseqüència més greu que aquest retard és que l'equip client havia de tenir prou recursos per a carregar el fitxer en memòria i obrir-lo: sens dubte no era vàlid per a dispositius mòbils. Malgrat totes aquestes limitacions, aquest sistema de vídeo incrustat avui dia es fa servir per a publicar petits vídeos de durada breu.

Exemple: Macromedia Flash SWF (1998)

El format SWF és el més antic i el primer que va possibilitar incrustar àudio i vídeo en una animació Flash. En publicar l'SWF en una pàgina d'un servidor web tradicional, el connector (*plug-in*) o complement Flash del navegador usat per a visualitzar-la havia de baixar el fitxer per complet per reproduir-lo, de manera que aquest sistema només era interessant per a vídeos d'una durada de no gaires segons.

Altres limitacions clares eren que, per a canviar una sola seqüència d'aquest vídeo, s'havia de refer tot el fitxer SWF i tornar-lo a publicar. Encara més: la càrrega en memòria de vídeos de llarga durada generava dessincronitzacions amb l'àudio al cap de la reproducció de dos o tres minuts de la pel·lícula.

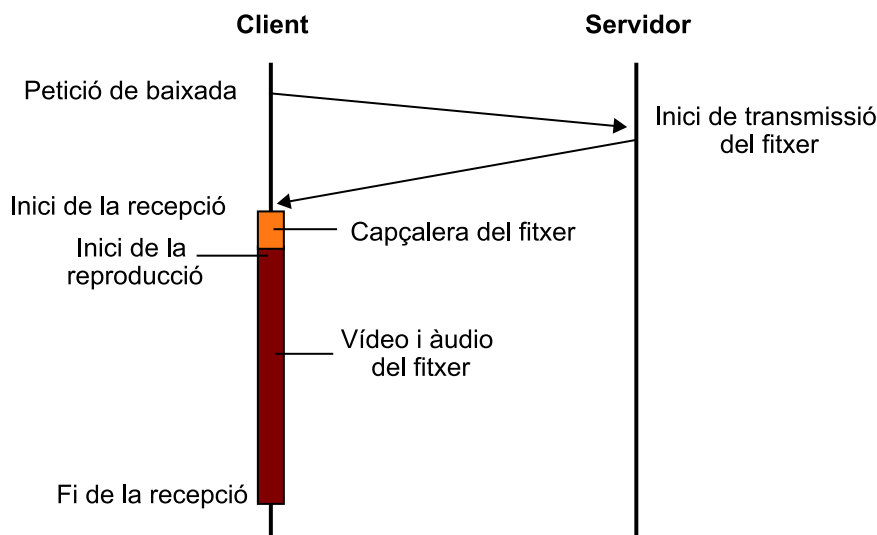


5.1.2. Baixada progressiva

Aquestes limitacions i la demanda creixent per contingut multimèdia a la web van fer que es desenvolupés un mecanisme de distribució de **baixada progressiva** gràcies al qual es podia reproduir perquè rebia una part "suficient" de dades.

La baixada progressiva, a més de reduir l'espera del client, eludeix els problemes de sincronia àudio-vídeo i, sobretot, simplifica els requisits tècnics del reproductor en termes de memòria, cosa que obre la porta als dispositius mòbils.

Il·lustració 14. Esquema de la baixada progressiva d'un fitxer de vídeo



Tots els formats no són compatibles amb aquest sistema de baixada. El fitxer de vídeo ha de tenir un format contenidor que sigui capaç de funcionar-hi. L'estructura d'aquest fitxer ha de tenir, almenys, una capçalera que prepari el programa reproductor i hi indiqui les dades bàsiques del tipus de codificació i la durada per a preveure les necessitats de memòria que tindrà, i a més que les dades de vídeo i àudio estiguin combinades com cal, ja que si l'àudio és després de totes les dades de vídeo (o viceversa) no es podrà reproduir res en bones condicions fins que no es rebi tot el fitxer. Tal com es pot veure en la taula resum de tipus de contenidors, en el cas d'arxius de vídeo, alguns tipus de formats contenidors estan preparats per a transmissió en temps real, és a dir, per a visualitzar-los sense haver d'esperar que baixin del tot.

No obstant això, la tècnica de baixada progressiva es continua basant en l'ús d'un servidor web genèric, dissenyat sobretot per a emmagatzemar webs. Quan un client fa una petició, tota l'amplada de banda disponible es fa servir per a baixar les dades al client de la manera més ràpida que es pugui.

En condicions normals, un servidor web envia nombrosos fitxers petits a un nombre limitat d'usuaris alhora, cosa que divideix l'amplada de banda entre els enviaments simultanis, però com que és de mida petita, és suficient. Tanmateix, en el moment en què ha d'enviar fitxers d'àudio i vídeo a un nombre creixent de clients alhora, l'amplada de banda sempre és un recurs escàs.

No hi ha una distribució intel·ligent de l'amplada de banda entre els clients: els que baixen un vídeo codificat a 256 Kbps el rebran a la mateixa velocitat que els que en baixen un a 2 Mbps, ja que el servidor web no analitza abans les característiques del fitxer.

El tradicional protocol web HTTP opera a més fent servir protocol de transport TCP, que assegura la recepció de les dades (després dels enviaments ha de rebre les confirmacions de recepció del client; si no les rep, reenvia les dades). Aquesta seguretat en la transmissió no és eficient per a àudio i vídeo, no tant pel volum de dades sinó perquè la pèrdua d'alguna d'aquestes dades no afecta sensiblement els ulls i les orelles del client, i perquè si reenvia dades que s'han perdut abans no es pot assegurar que arribin a temps de ser reproduïdes.

Exemple: baixada progressiva i el format contenidor FV de Macromedia Flash (2003)

Macromedia incorpora la baixada progressiva introduint el nou format contenidor FLV (Flash Video) preparat per a aquest. A part de les millores incorporades a la codificació d'àudio i vídeo, el fitxer SWF es podia associar a un fitxer extern FLV de vídeo, i així independitzava el projecte del contingut audiovisual.

Per a preveure pauses involuntàries de la reproducció de l'àudio i vídeo per falta de dades en connexions lentes, Flash va redefinir més endavant la suficiència de dades com la característica imprescindible per a assegurar la reproducció completa del fitxer FLV i va fer una previsió conservadora segons la velocitat de baixada estimada, el pes total del fitxer i la durada del vídeo.

Per aquestes raons es va definir un tercer mecanisme de distribució, la **transmissió en temps real** (*streaming*), mitjançant l'addició d'un servidor addicional al servidor web (o servei addicional en el mateix servidor web) que té unes característiques que superen les limitacions indicades.

5.1.3. Transmissió en temps real de vídeo

El servidor de transmissió en temps real (*streaming*) analitza la capçalera del contenidor del material abans de fer-ne la transmissió, de manera que sap la velocitat de codificació³⁸ necessària per a fer-ne la reproducció correcta, i reserva així una amplada de banda idèntica per a la transmissió en temps real. Aquesta gestió intel·ligent personalitzada optimitza els recursos més bé que cap dels altres mecanismes de transmissió vistos.

⁽³⁸⁾En anglès, *bit rate*.

Encara més, a diferència de la baixada progressiva, es pot començar la reproducció de la transmissió en temps real des de qualsevol punt del fitxer. L'estructura del contenidor dels fitxers preparats per a transmissió en temps real es divideix en petites unitats, cadascuna amb informació relativa a la porció que té de material audiovisual i amb una marca de temps que n'indica la posició en el conjunt del fitxer.

Com que el servidor de transmissió en temps real analitza las característiques del material que ha de transmetre, es condicionen els formats contenidors que es poden fer servir en cada tipus de servidor de transmissió en temps real: un servidor Windows Media Server no pot transmetre fitxers en Real Video o Real Audio, però sí que ho pot fer Windows Media Video o Windows Media Audio.

Sempre hi ha un retard entre l'arribada de les dades al client i l'inici de la reproducció en omplir una memòria intermèdia que prevé de les variacions en el temps d'arribada dels paquets per canvis de congestió de la xarxa.

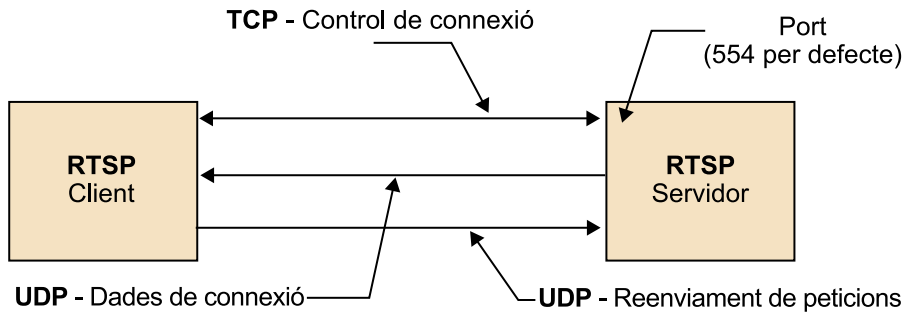
La transmissió en temps real prevé a més de la baixada dels fitxers d'àudio i vídeo per part del client, i redueix així les possibilitats de fer-ne còpies ("pirateria"), perquè envia els paquets de dades directament a l'aplicació client i els descarta una vegada reproduïts.

Així, fa ús del protocol UDP, que no retransmet les dades perdudes amb la finalitat d'aconseguir un més bon rendiment en transmissions de dades en temps real, tolerants a la pèrdua de dades. Per la seva naturalesa, UDP té més prioritat que TCP en els centres de distribució de trànsit, fins i tot en xarxes congestionades, de manera que n'assegura el servei. Alguns fabricants fan servir fins i tot UDP Resend, un esquema de retransmissió de dades perdudes que només en reenvia al client si hi ha temps per a reproduir-les (per exemple, Microsoft en els antics Windows Media Server).

Entre el client i el servidor de transmissió en temps real es crea una connexió paral·lela a la UDP però en protocol TCP per a transmetre les ordres de control del reproductor client (*PLAY*, *PAUSE*, etc.) al servidor o viceversa per al control remot de càmeres IP, per a enviar un codi de temps quadre a quadre, etc. Aquest sistema d'ús d'UDP per a dades i TCP per a control s'anomena *RTSP*³⁹.

⁽³⁹⁾La sigla *RTSP* correspon a *real-time streaming protocol*.

Il·lustració 15. Connexió RTSP



En les situacions en què el protocol RTSP no està permès, alguns fabricants implementen solucions compatibles amb HTTP i el seu port 80, com per exemple en les xarxes privades. Aquestes solucions es basen en HTTP Streaming, una versió modificada d'HTTP. I en xarxes de molt alta velocitat i d'alta qualitat (de baixes pèrdues de dades) com les xarxes òptiques, s'està tornant cap enrere per a fer servir TCP com a protocol de tot el sistema per a assegurar la màxima qualitat de reproducció del material transmès.

Exemple: Adobe Flash Media Server (2005)

Adobe fa un altre pas i planteja una solució més robusta i consistent per a distribuir àudio i vídeo que no pas la baixada progressiva en un servidor web tradicional, i ofereix un servidor dedicat (Adobe Flash Media Server) que gestiona de manera intel·ligent la capacitat de connexió de cada client i optimitza l'ús dels seus recursos: transmissió en temps real (*streaming*). Per mitjà d'aquest servidor, el vídeo comença abans que amb la baixada progressiva, consumeix menys recursos del servidor i del client, i facilita el seguiment d'incidències i la facturació del servei.

No aposta per RTSP, sinó per un protocol de transmissió en temps real propietari RTMP o, des de la versió 10.1 de Flash Player, per HTTP Streaming. En aquest últim cas, lligat a l'ús del contenidor FLV amb còdec MPEG-4.



5.1.4. P2P

Aquesta nova línia en ple desenvolupament trenca el model client-servidor usat fins ara i aposta per un model descentralitzat basat en les **xarxes d'igual a igual** (*peer-to-peer* o **P2P**), en les quals cada client pot ser consumidor i, alhora, publica i distribueix. Es recolza en algun dels protocols de transmissió en temps real que hem explicat.

Un exemple il·lustratiu: Adobe Flash Player 10 (2010)

Des d'aquesta versió es capacita els clients Flash Player a compartir àudio, vídeo i dades mitjançant una connexió directa a altres clients sense passar pel servidor. Fa servir un protocol propietari d'Adobe, RTMFP, basat en UDF.



5.2. Els contenidors de vídeo per transmissió en temps real

No tots els formats contenidors són compatibles amb transmissió en temps real, tal com hem vist en altres mòduls. Per exemple, l'antic format AVI 1.0 (que més endavant es va modificar per a acceptar transmissió en temps real) s'estructura en blocs de dades numerades anomenades *chunks*, té una única capçalera al començament de l'arxiu on indica les característiques del contingut: durada total del vídeo, velocitat màxima de transferència per a la baixada progressiva, nombre de fluxos d'àudio, vídeo i dades (subtítols) que conté, espai recomanat que cal reservar en la memòria intermèdia del receptor per a emmagatzemar *chunks*, amplada i alçada del vídeo en píxels, freqüència de quadre, profunditat de color, còdec de vídeo i d'àudio usats, etc.

En principi sembla que té tota la informació necessària per a reproduir-lo, però només des del començament. No es pot reproduir a partir de cert punt si no s'ha baixat abans el fitxer fins a aquest punt com a mínim. És vàlid, doncs, per a baixada progressiva però no per a transmissió en temps real. Un contenidor de vídeo compatible amb transmissió en temps real ha de possibilitar la reproducció des de qualsevol punt del fitxer, en particular:

- Ha de tenir una capçalera que indiqui les característiques principals del contingut d'àudio i vídeo per a notificar al servidor de transmissió en temps real quins recursos de xarxa ha de reservar per a la transmissió en condicions òptimes.
- S'ha d'estructurar en blocs independents d'àudio i vídeo anomenats comunament *àtoms*, cadascun amb una capçalera que el caracteritzi i que tingui marques de sincronia per a identificar-lo dins del conjunt del fitxer i possibilitar l'accés directe a aquest punt.
- Cada àtom ha de tenir prou informació audiovisual per a poder iniciar la reproducció en aquell punt, és a dir, ha d'abastar un segment sincronitzat de cada flux (àudio, vídeo i dades). En general, un àtom són uns quants segons d'àudio o vídeo.

Il·lustració 16. Estructura de qualsevol fitxer de vídeo preparat per a transmissió en temps real

ftyp	moov	A	V	A	V	A	V	moov	A	V	A	V
------	------	---	---	---	---	---	---	------	---	---	---	---

Una capçalera amb informació general i una sèrie d'àtoms amb contingut fragmentat d'àudio i vídeo. Si el fitxer és molt gran, és habitual afegir algun àtom intermediari.

L'emplaçament dels fluxos en cada àtom ha de ser multiplexat, és a dir, a dins s'hi han de combinar alternativament i de manera intel·ligent les dades de tots els fluxos. Si s'emmagatzemés primer el flux de vídeo, després el d'àudio i finalment el de dades, fins a la lectura completa del bloc de vídeo no es podria començar a reproduir el vídeo juntament amb l'àudio.

Un exemple: Flash Video i el format base ISO

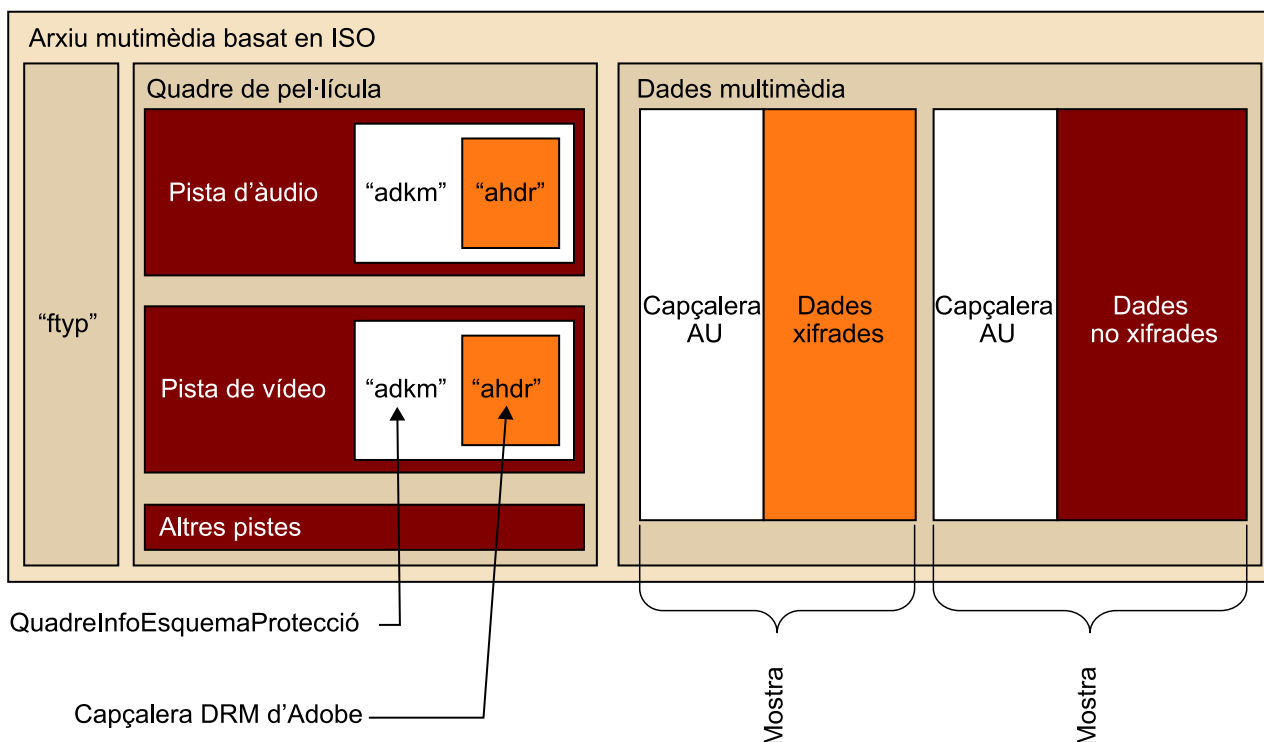
El contenidor Flash Video és l'estàndard de transmissió en temps real més estès avui dia, i seleccionat com el de referència per a empreses que ofereixen aquest servei com YouTube, BBC Online, Hulu o Yahoo Video. Aquest contenidor s'emmagatzema com a fitxer de vídeo en els formats FLV i el millorat F4V (2010), i es pot integrar dins d'un fitxer SWF.



L'F4V, igual que altres contenidors com MP4 o 3GPP, s'inspira en una idea més universal, l'especificació MPEG-4 Part 12, anomenada comunament *ISO base*, un model global consensuat i ben especificat que té universalment el suport de l'organització ISO, que el va dissenyar a partir del contenidor QuickTime per facilitar l'edició, la presentació, la gestió i l'intercanvi del material per qualsevol mitjà, independentment del protocol usat.

Centrant-nos en el cas F4V, l'estructura és bàsicament ISO, afegint-hi les seves pròpies interpretacions:

Il·lustració 17. Estructura de Flash F4V



Es pot veure que és modular, descomponible en objectes molt senzills, cadascun amb una missió específica. Cada objecte s'anomena *caixa*⁴⁰, no hi ha res que no formi part d'alguna, i cada caixa pot contenir subcaixes. Sens dubte, la idea general d'àtom que hem comentat ara fa unes pàgines coincideix del tot amb l'ús de la caixa com a element bàsic d'aquest contenidor.

⁽⁴⁰⁾En anglès, *box*.

a) Caixa *ftyp*

Primer de tot, hi ha una caixa *tipus de fitxer (ftyp)* de valor 3 o 4 caràcters que informa així el programa reproductor del tipus de material que conté. Se'n pot trobar una llista a MP4REG > Registrered types. S'hi pot veure ben clar que no apareix indicada cap *ftyp* que indiqui Flash Video. Adobe es va acollir a l'ISO base però de manera unilateral, sense registrar-se. Per a tenir una llista de tots els *ftyp* (registrats o no), s'ha d'accedir a la pàgina web Complete List of all known MP4 / QuickTime 'ftyp' designations, en la qual apareix *f4v* com el de Flash Video.

b) Caixa MovieBox *moov*

La caixa o àtom *moov* és la capçalera única del fitxer *f4v*; pot contenir altres caixes que defineixen en conjunt l'estructura de les dades –un *TrackBox* o *trak* per al vídeo, un altre *TrackBox* o *trak* per a l'àudio, *MediaBox* o *mdia*, etc. (vegeu el gràfic adjunt)–, i és present en tots els contenidors que s'acullen a ISO base. Inclou un índex de *frames*, *track hint*, útil tant en transmissió en temps real com en baixada progressiva.

c) Caixa *MediaData mdat*

A continuació, i de manera multiplexada o paral·lela, les pistes de vídeo i d'àudio dins de la gran caixa *mdat*. Cadascuna de les pistes se subdivideix en petits subblocs, cadascun marcat amb un codi de temps, i poden estar protegits o no amb contrasenya o un altre mecanisme de protecció; tota una manera de fer una picada d'ullet a l'ús en aplicacions de valor afegit.

Així, no és possible accedir a l'instant a qualsevol punt del vídeo i l'àudio: com que es vol reproduir des d'un punt específic, que se situa sempre dins d'un subbloc, és imprescindible llegir la capçalera d'aquest subbloc per a confirmar els drets d'accés al material i desplaçar-se dins del subbloc fins a accedir al punt exacte volgut. Això sempre comporta cert temps d'anàlisi fins a poder reproduir el material audiovisual. Sens dubte, com més petits són els subblocs, més curt és el temps d'accés, però més gran és la sobrecàrrega en bits que implica el nombre creixent de capçaleres, de manera que cal un compromís entre l'eficiència en l'accés i la mida del fitxer.

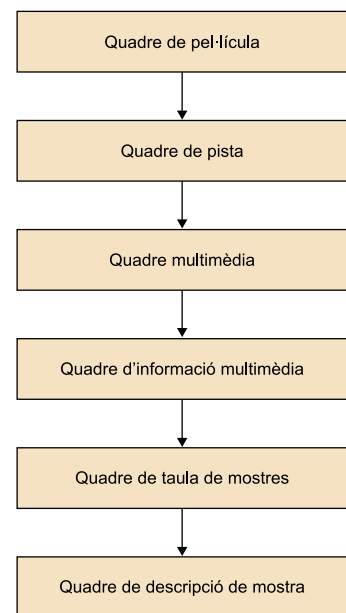
Exercici: validació de fitxers *flv*

Abans de penjar un fitxer FLV o F4V a la xarxa, és recomanable aplicar-hi l'eina *flvcheck* (versió Windows o Linux), perquè valida l'estructura de caixes del fitxer. Està limitat a fitxers de menys de 4 Gb i pensat per a FLV, però funciona amb la majoria d'F4V.

S'executa en la línia d'ordre i la sintaxi és la següent:

```
flvcheck -f "fitxer_a_analitzar.f4v" -v
```

El resultat és la validació del fitxer i a més un abocament dels valors de cada caixa *ftyp* i *moov*. Per exemple:



Baixada de l'eina *flvcheck*

L'eina *flvcheck* es pot baixar des de la pàgina web Productivity tools and sample downloads d'Adobe.

```
C:\Users\Alex>flvcheck -f "Flash10 P2P.f4v" -v
```

```
FLVCheck version 1.0 - Utility to validate flv and mp4 media files.  
Copyright (C) 2008 Adobe Systems Incorporated. All rights reserved.  
www.adobe.com
```

```
File: Flash10 P2P.f4v  
ftyp/ 28  
moov/ 648485  
moov/mvhd/ 108  
moov/trak/ 373763  
moov/trak/tkhd/ 92  
moov/trak/mdia/ 373663  
moov/trak/mdia/mdhd/ 32  
moov/trak/mdia/hdlr/ 44  
moov/trak/mdia/minf/ 373579  
moov/trak/mdia/minf/vmhd/ 20  
moov/trak/mdia/minf/dinf/ 36  
moov/trak/mdia/minf/stbl/ 373515  
moov/trak/mdia/minf/stbl/stsd/ 171  
moov/trak/mdia/minf/stbl/stsd/avc1/ 155  
moov/trak/mdia/minf/stbl/stsd/avc1/avcC/ 69  
moov/trak/mdia/minf/stbl/stts/ 117560  
moov/trak/mdia/minf/stbl/stsc/ 12372  
moov/trak/mdia/minf/stbl/stsz/ 58796  
moov/trak/mdia/minf/stbl/co64/ 8256  
moov/trak/mdia/minf/stbl/ctts/ 117560  
moov/trak/mdia/minf/stbl/stss/ 58792  
moov/trak/ 274183  
moov/trak/tkhd/ 92  
moov/trak/mdia/ 274083  
moov/trak/mdia/mdhd/ 32  
moov/trak/mdia/hdlr/ 68  
moov/trak/mdia/minf/ 273975  
moov/trak/mdia/minf/smhd/ 16  
moov/trak/mdia/minf/dinf/ 36  
moov/trak/mdia/minf/stbl/ 273915  
moov/trak/mdia/minf/stbl/stsd/ 103  
moov/trak/mdia/minf/stbl/stsd/mp4a/ 87  
moov/trak/mdia/minf/stbl/stsd/mp4a/esds/ 51  
moov/trak/mdia/minf/stbl/stts/ 32  
moov/trak/mdia/minf/stbl/stsc/ 12372  
moov/trak/mdia/minf/stbl/stsz/ 253144  
moov/trak/mdia/minf/stbl/co64/ 8256  
moov/trak/ 423  
moov/trak/tkhd/ 92  
moov/trak/mdia/ 323  
moov/trak/mdia/mdhd/ 32  
moov/trak/mdia/hdlr/ 55  
moov/trak/mdia/minf/ 228  
moov/trak/mdia/minf/nmhd/ 12  
moov/trak/mdia/minf/dinf/ 36  
moov/trak/mdia/minf/stbl/ 172  
moov/trak/mdia/minf/stbl/stsd/ 32  
moov/trak/mdia/minf/stbl/stsd/amf0/ 16  
moov/trak/mdia/minf/stbl/stts/ 32  
moov/trak/mdia/minf/stbl/stsc/ 36  
moov/trak/mdia/minf/stbl/stsz/ 32  
moov/trak/mdia/minf/stbl/co64/ 32  
uuid/ 56013  
mdat/ 48922658  
11-07-17 20:19:04 Flash10 P2P.f4v passed
```

Si l'anàlisi retorna un error, aquesta eina només el pot corregir si és un fitxer FLV i l'error és en la metadada (si és en l'estructura no hi pot fer gaire res). En aquest cas s'aplica:

```
flvcheck -m "fitxer_malmès.flv" -v
```

La llista de missatges d'error i d'avisos generats per flvcheck és a la pàgina web [Checking video files](#).

Exercici: validació de fitxer *f4v*

Hi ha una eina específica d'aplicació a fitxers F4V que detecta errors i corregeix problemes d'estructura: Adobe F4V Post Processor.

Entre les opcions d'aquesta eina ens centrem en la no documentada `-c`, que corregeix estructura:

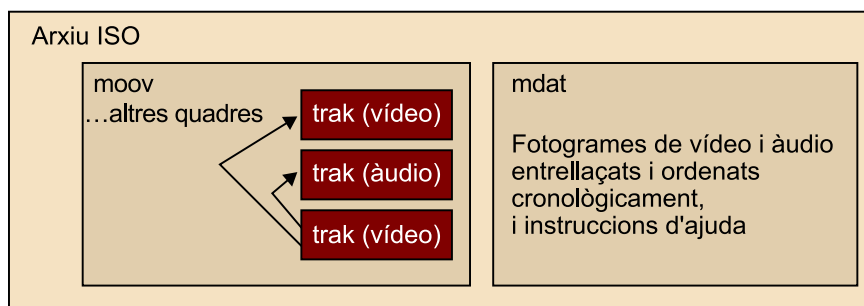
```
f4vpp -c "fitxer.f4v"
```

En la documentació associada de l'eina s'indiquen els diferents missatges d'error que pot generar en la detecció de problemes.

Un altre exemple: 3GPP2 (2007)

Com a informació, el format 3GPP2 (els fitxers del qual tenen l'extensió `.3g2`) també es basa en el contenidor base ISO, i l'estructura d'aquest format es pot representar de la manera següent:

Il·lustració 18. Estructura de 3GPP2



És similar a la de Flash Video perquè tenen un origen comú.

Baixada de l'eina *f4vpp*

L'eina *f4vpp* es pot baixar des de la pàgina web Productivity tools and sample downloads d'Adobe.

