

# Desenvolupament a baix nivell per aplicacions en Temps Real

MAS BONED, F. B., OLIVER TORRES, T.

**Resum—** El present article donarà a conèixer les diferents aplicacions a baix nivell en temps real que podem trobar a internet, centrant el seu contingut a les llibreries C++ i les aplicacions RTP tools (desenvolupades pel doctor Henning Schulzrinne i el seu departament). El programari que ens ofereix Sun Microsystems per dur a terme les mateixes accions (en aquest cas emprant el llenguatge de programació Java), i les canonades de GStreamer per aplicacions de capes superiors es comentaran molt breument. El protocol més implicat dins la transferència en temps real seran breument explicat.

## I. INTRODUCCIÓ

La transmissió d'informació multimèdia (àudio i vídeo) té un gran impacte en els sistemes de comunicació. Aquesta transmissió requereix no tan sols el suport de les xarxes, sinó també dels protocols de comunicació de nivells superiors. Aquest protocols són els adoptats per la IETF: RTP, SDP i SIP.

Podem parlar de transmissió d'informació en temps real quan es pot assegurar que la informació arriba al seu destí amb uns paràmetres determinats (retràs, rendiment, fiabilitat, etc.).

En general les aplicacions multimèdia requereixen una qualitat de servei (QoS) per part dels serveis de la xarxa. De les noves xarxes s'exigeix poder especificar aquesta qualitat de servei i assegurar el seu compliment.

Dissenyats els protocols i les xarxes capaces de mantenir un transferència en temps real, només necessitem les llibreries que ens permetin la creació d'una aplicació (programari) d'aquestes característiques. Així neixen les llibreries C, C++ i Java, entre d'altres.

## II. RTP

Les aplicacions multimèdia necessiten alguna classe de protocol de transport, amb característiques diferents de TCP i amb major funcionalitat que UDP. El protocol creat per satisfer aquestes necessitats és l'anomenat 'Real-time Transport Protocol' (RTP). No obstant RTP es considera un protocol de capa d'aplicació.

Les seves funcions són les següents:

- Comunicar la elecció de l'esquema de codificació de les dades.
- Determinar la relació temporal entre les dades rebudes.
- Sincronitzar els distints medis.
- Indicar la pèrdua de paquets.
- Indicar límits de *frames* en les dades.
- Identificació amigable d'usuaris.

Un requeriment addicional a les funcions senyalades és l'ús eficient de l'ample de banda.

En realitat l'estàndard RTP defineix dos protocols, RTP i RTCP (Real-time Transport Control Protocol). El primer es emprat per l'intercanvi de dades multimèdia, mentre que el segon és emprat per l'enviament periòdic d'informació de control associada a un determinat flux de dades.

### I. Format del paquet RTP

El format que segueixen les trames RTP és el representat a la figura 1:

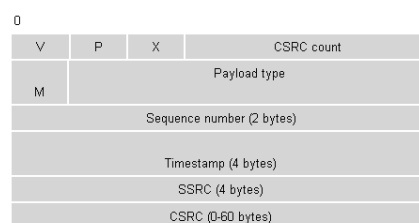


Figura 1. Capçalera del paquet RTP.

On 'V' són dos bits que ens informen de la versió del protocol. El bit 'P' ens indica si ha hagut farciment de bits per complir algun requeriment mínim. El bit 'X' senyala la presència d'extensions de l'encapçalament. El bit de marques 'M' s'empra per indicar *frames*.

### J. RTCP

RTP lliura un *stream* de control que està associat a un *stream* de dades per una aplicació multimèdia.

Proveeix tres funcions:

- Feedback en el desenvolupament de l'aplicació.
- Sincronització i correlació de diferents *streams* que provenen de la mateixa font.
- Un mode de transportar la identitat de l'usuari emissor pel desplegament en la interfície de l'usuari receptor.

Un cop esmentat el protocol RTP descriurem les eines de desenvolupament a baix nivell en Temps Real per a diferents llenguatges de programació.

## III. RTP TOOLS

Les RTP tools són eines que ens proporciona la Universitat de Columbia, creades pel doctor Henning Schulzrinne i els seus col·laboradors [1]. Aquestes aplicacions estan codificades en C, per tant poden ser visualitzades per qualsevol compilador C.

### K. Descripció

La distribució de les RTP tools es compon d'una sèrie de petites aplicacions que poden ser emprades per el processament de dades RTP.

- rtpplay: grava la sessió RTP enregistrada per rtpdump.

- `rtpsend`: genera paquet RTP amb la descripció textual.
- `rtpdump`: analitza sintàcticament i imprimeix els paquets RTP.
- `rtptrans`: traductor RTP entre xarxes *multicast* i *unicast*

#### L. Instal·lació

Les RTP tools han de poder ser compilades sobre qualsevol plataforma Posix compatible amb els sockets de suport.

#### M. Suggeriment d'ús general

Les adreces de xarxa poden ser adreces *multicast* o *unicast*. Aquestes poden ser especificades amb nombres decimals o com el nom del host. El nombre del port s'ha d'especificar amb un nombre decimal dins el rang 1..65535.

A no ser que s'indiqui el contrari, les entrades arribaran per *stdin* i les sortides per *stdout* (fitxers d'entrada i de sortida respectivament).

#### N. `rtpplay`

`rtpplay` llegeix la sessió RTP de dades, emmagatzemada per `rtpdump` del fitxer 'f' o *stdin*.

#### 5) Format de trama

`rtpplay [-T][-v][-f][-p][-s][-b][-e][t/tl]`

On 'T' i 'v' són banderes.

'f' és el fitxer d'entrada.

'p' és el fitxer del perfil de la freqüència del rellotge.

's' és el port font.

'b' i 'e' són el temps d'inici i final respectivament, mesurats en segons.

#### 6) Funcionament

Si el fitxer no està especificat, s'enviaran les dades a l'adreça de xarxa de destí i port amb un temps de vida de 'ttl'.

Si la bandera 'T' està activada, el temps entre paquets correspon al temps d'arribada en lloc del *timestamps* de RTP. Per una altra banda podem emprar el *timestamps* per suavitzar el *jitter* i restaurar la seqüència del paquets. Aquest fet pot provocar un desordre en el paquets de RTP i RTCP.

El port font és el port de sortida dels paquets, es pot establir amb la bandera 's'. S'escull un port a l'atzar si no activem la bandera.

El fitxer del perfil conté línies amb dos camps cada una: la primera és la carrega útil de tipus numèric, a segona és la freqüència del rellotge. En cas de que la bandera 'T' estigui desactivada els perfils seran ignorats.

La bandera 'v' ens indica que s'han generat paquets a *stdout*.

#### O. `rtpsend`

L'aplicació `rtpsend` envia un flux de paquets RTP amb paràmetres configurables. Amb això s'intenta posar a prova les característiques del protocol RTP. Les capçaleres de RTP i RTCP són llegides d'un fitxer, generat a mà, per un programa o per `rtpdump` (format *ascii*).

#### 1) Format de trama

`rtpsend [-a][-l][-s][-f]destí/port[t/tl]`

On 'a', 'l' i 's' són banderes.

'f' és el fitxer descriptiu.

'ttl' és el temps de vida.

#### 2) Funcionament

Els paquets se envien amb un temps de vida de valor 'ttl'.

Si les dades es llegeixen d'un arxiu en lloc de *stdin*, la bandera 'l' reenvia la mateixa seqüència de paquets una i una altra vegada.

El port d'origen per els paquets de sortida es pot configurar amb la bandera 's'. Un port a l'atzar s'escull si no s'especifica aquest indicador.

Si la bandera 'a' s'especifica, `rtpsend` inclou un router IP opció d'alertar els paquets RTCP. Això s'empra per els recursos YESSIR de reserva de protocol.

L'arxiu conté la descripció dels paquets que s'enviaran. Dins l'arxiu, cada entrada comença amb un valor de temps, en segon, en relació amb el temps de començament de la trama. El valor temporal ha d'aparèixer a l'inici d'una línia, sense espais en blanc. Dins d'un RTP, RTCP o descripció de paquets, els paràmetres poden aparèixer en qualsevol ordre, sense espais en blanc al voltant del signe igual.

Les línies se continuen amb els espais en blanc inicials en la línia següent.

Les línies comencen amb el signe #. Els *strings* s'inclouen entre cometes.

#### P. `rtpdump`

Analitza els paquets RTP per poder ser enviats posteriorment.

#### 1) Format

`rtpdump [-F][-t][-x][-f][-o]`

On 'F' és el format.

't' és la duració.

'x' són les dades (*payload*).

'f' és l'arxiu d'especificació d'adreça de xarxa.

'o' és el fitxer de sortida.

#### 2) Funcionament

L'aplicació `rtpdump` escolta l'adreça i el port del parell de paquets RTP i RTCP i descarrega la versió al fitxer de sortida si s'especifica o al *stdout*.

Si el fitxer 'f' s'especifica, el fitxer s'empra en lloc de l'adreça de xarxa.

Format	Text/binari	Format	Text/binari
Dump	binari	Ascii	text
Header		Hex	
Payload		rtp	
		short	

TAULA 1. FORMAT DE LES TRAMES.

#### Q. `rtptrans`

Els paquets RTP/RTCP arriben de una de les direccions a totes les direccions.

Les direccions poden ser *multicast* o *unicast*. El valor 'ttl' per les direccions *unicast* s'ignoren (en realitat, no comprova si són paquets RTP o no).

#### 1) Format de trama

`rtptrans [host]/port[t/tl][host]/port[t/tl][...]`

On 'ttl' és el temps de vida.

## 2) Funcionament

Amés el traductor pot traduir paquets VAT en paquets RTP. El control de VAT es tradueix en paquets RTCP SDES amb n CNAME i un nom d'entrada. No obstant, això només està destinat a ser emprat en les següents configuracions: paquets VAT que arriben a una connexió *multicast* es tradueixen en RTP i es transmeten a través d'una connexió *unicast*.

Per lo tant, en la actualitat es compte principalment amb el suport de les següents tipologies: *multicast* VAT → Traductor → *unicast* RTP, i en el camí de tornada *multicast* VAT ← traductor ← *unicast* VAT.

Això significa que l'agent d'àudio i vídeo en l'enllaç *unicast* ha de poder emprar tant VAT com RTP.

## IV. GNU ccRTP

ccRTP és una llibreria de C++ basades en GNU Common C++ [2], que proporciona un alt rendiment, flexibilitat i extensibilitat amb els estàndards de a pila de protocols RTP i amb total suport de RTCP. El disseny i l'aplicació de ccRTP és adequat per servidors d'alta capacitat i passarel·les, així com per aplicacions d'usuaris.

La llibreria està creada com un protocol de *framework* i no com un protocol de transport típic tal com TCP o UDP. Així, RTP es casi mai implementable fora de la capa d'aplicació. Les aplicacions RTP moltes vegades han de personalitzar la capacitat d'adaptació del disseny i processament dels paquets RTP, les limitacions del temps, període de sessions, així com altres mecanismes de RTP i RTCP.

El suport a les dades d'àudio i vídeo també es consideren a les llibreries ccRTP, que pot fer marcs parcials (ensamblatge i desensamblatge). Suporta el models de transport *unicast*, *multicast* i *multi-unicast*, així com múltiples fonts actives sincronitzades, múltiples sessions RTP i múltiples aplicacions RTP. Això permet el seu ús per a la construcció de totes les formes d'internet basades en les normes d'àudio i vídeo per sistemes de conferència.

ccRTP proporciona seguretat i alt rendiment. Empra la cua de les llistes de paquets per la repetició i transmissió de paquets de dades. GNU ccRTP dona suport a RTCP i molt d'altres estàndards que es necessiten per avançades aplicacions d'*streaming*.

## 3) Situació actual

GNU ccRTP 1.5.0 introdueix suport per el perfil Secure RTP.

Les emissions 1.4.x s'han centrat en la introducció del protocol IPv6

## 4) Manual de referència

Aquest petit manual defineix les classes C++ de les llibreries de ccRTP per desenvolupar una comunicació en temps real.

### a) Iniciar sessió:

Iniciar sessió RTP amb ccRTP implica la construcció d'un objecte de la classe i crida alguns mètodes d'inicialització opcionals de *RTPSession*. El propòsit de la pila de protocols de ccRTP, com *RTPSession*, ha de ser signar abans de començar l'execució. Això es fa cridant el mètode *startRunning()*. També hi ha un *enableStack()*, aquest tan sols activa la pila, però no s'inicia l'execució.

El període de sessions i altres paràmetres es pot establir com:

- Afegir destinataris.
- Establir format de càrrega útil.
- Conjunt local d'articles SDES.
- Permetre el farciment de paquets de dades.
- Establiment de les direccions.
- Establir l'ample de banda cridant el mètode *setSessionBandwidth()*. De no ser així, tindrem per defecte un ample de 64 Kbytes, i un 5% d'aquest per RTCP.

### b) Enviament de dades:

El paquets de dades s'envien a través del mètode *putData()*.

Per defecte, el bit marcador de paquets enviats no està definit. El seu valor es pot configurar a través del mètode *setMark()*, que té un *boolean* com argument.

ccRTP també suporta fragmentació de blocs de dades en varis paquets RTP. Quan els blocs de dades major que el màxim tamany del segment es presenta a través de *putData()*, dos o més paquets s'introdueixen en la cua de paquets sortints.

### c) Recepció de dades:

Rebudes les dades s'extreu el paquet de la cua d'entrada a través del mètode *getData()*, el qual ens dona el *timestamp* i una font opcional de sincronització.

El nou paquet de cues s'encarrega de funcions tal com reordenació de paquets o filtrat de paquets duplicats.

Es pot saber si hi ha paquets en la cua de recepció mitjançant *isWaiting()*. Ambdós, *getData()* i *isWaiting()*, tenen un paràmetre opcional el qual selecciona una determinada font de sincronització de la unitat de dades. Si no s'especifica les unitat de dades es retornen, independentment del seu origen.

El períodes de les sessions de RTP poden tancar suprimint o destruint els objectes *RTPSession*. La pila envia un paquet BYE a cada destí quan el destructor de sessions es crida. També es possible explícitament enviar un paquet BYE a través del mètode *dispatchBYE()*.

### d) Tipus de càrrega útil i formats:

En el context de RTP, una càrrega útil a RTP és un identificador numèric de set bits. Per tipus de càrrega útil, GNU ccRTP defineix el tipus enter *PayloadType* i el tipus enumerat *StaticPayloadType*, com l'enumeració de la càrrega útil de tipus de RTP assignats per àudio i els formats de vídeo estàndard.

### e) Manipulació de successos:

Hi ha una sèrie de casos que exigeixen una especial resposta de la sol·licitud. ccRTP defineix *plug-ins* per manejar aquests successos.

- Paquets RTP d'arribada:
  - *onRTPPacketRecv*
- Paquets RTCP d'arribada:
  - *onGotSR*
  - *onGotRR onGotRR*
  - *onGotSDESCChunk*
  - *onGotAPP onGotAPP*
  - *onGotRRSRExtension*
- Sincronització d'estats font:
  - *onNewSyncSource*
- Col·lisions SSRC:
  - *onSSRCCollision*
- Paquets de caducitat de RTP
  - *onExpireSend*
  - *onExpireRecv*
  - *end2EndDelayed end2EndDelayed*

## V. LLIBRERIES JAVA

JMF és una API que ens serveix per incloure multimèdia a les nostres aplicacions de Java [3], funciona bàsicament, rebent el contingut multimèdia d'alguna font, per després processar-lo i entregant-lo en alguna sortida multimèdia.

JMF es construeix entorn a una arquitectura de components: mitjans manipuladors, font de dades, codecs, renderers i mux/demux.

Exemples de mètodes: start(), stop(), realize(), prefetch(), deallocate(), setRate(), setMediaTime(), getVisualComponent(), getControlPanelComponent(), etc.

JMF és una molt flexible arquitectura multimèdia que ens mostra moltes promeses. En el futur, cal esperar que Sun treballarà per fer-ho més estable, així com documentar el marc i proporcionant un major nombre de codis exemple.

## VI. CANONADES GSTREAMER

El suport de RTP en Gstreamer [4] actualment és parcila, encara així, ja és possible executar algunes canonades que ens transmeten vídeo i àudio emprant RTP damunt UDP.

Per poder emprar aquestes funcions cal tenir instal·lats els següents *plugins*: gstreamer0.10-plugins-base, gstreamer0.10-farsight, gstreamer0.10-plugins-good, gstreamer0.10-plugins-ugly, gstreamer0.10-alsa, gstreamer0.10-ffmpeg, gstreamer0.10-x.

Les funcions que en permeten fer la transferència són:

- rtpbin: s'adapta a qualsevol necessitat.
- rtpjitterbuffer: control del jitter i ordena els paquets.
- Rtpdemux: detecta qualsevol nou tipus de càrrega útil que es rebí en el flux RTP.

## VII. CONCLUSIONS

La creació d'aquest article ens ha fet veure la difícil tasca que tenen els investigadors. La redacció de l'article implica un gran aprofundiment dins el tema o temes esmentats a aquest. Si bé amb menys mesura que els investigadors de la

Universitat de les Illes Balears, nosaltres em procurat capficar-nos i extreure la major informació possible sobre la transferència en temps real.

Referent a l'article, cal esmentar que la transferència en temps real per dos usuaris qualsevol de la xarxa, emprant programari propi té una certa dificultat. Els coneixement en llenguatge C++, C i Java han de ser elevats per tal d'aconseguir un transferència d'informació amb unes condicions acceptables.

Les canonades GStreamer es poden utilitzar una vegada aconseguida la transferència i fer la reproducció de la informació enviada.

En definitiva, podem concloure que la creació de l'article ha fomentat el nostre esperit crític i que la codificació d'un programa, mitjançant programari lliure, necessita evolucionar per tal d'oferir els mateixos i millors serveis, de forma més senzilla que la esmentada a aquest article.

## REFERÈNCIES

- [19] RTP tools. Disponible on-line: <http://www.cs.columbia.edu/~hgs/>
- [20] GNU ccRTP. Disponible on-line: <http://www.gnu.org/software/ccrtp/>
- [21] Java media framework. Disponible on-line: <http://java.sun.com/javase/technologies/desktop/media/jmf/>
- [4] Canonades GStreamer. Disponible on-line: <http://crvsl.inf-cr.uclm.es/node/655>



Nom: Francisco de Borja Mas Boned.  
Titulació: Enginyeria Tècnica en Telecomunicacions, especialitat en Telemàtica.  
Direcció: [fborja.mas@gmail.com](mailto:fborja.mas@gmail.com)



Nom: Tomeu Oliver Torres.  
Titulació: Enginyeria Tècnica Industrial, especialitat en electrònica industrial.  
Enginyeria Tècnica en Telecomunicacions, especialitat en Telemàtica.  
Direcció: [tomeu.oliver@gmail.com](mailto:tomeu.oliver@gmail.com)

Assignatura impartida per: Magdalena Payeras Capellà