

strManager: replicador de paquets d'alta capacitat

Guillem Cabrera, Javier López, Xavier Calvo, Jesus Alcober
Departament d'Enginyeria Telemàtica, mediaENTEL
Universitat Politècnica de Catalunya i Fundació i2CAT
{guillem.cabrera, javi.lopez}@i2cat.net,
{xavier.calvo-brugal, jesus.alcober}@upc.edu

Maig de 2009

Resum

Actualment Internet suposa un nou medi per a la distribució de continguts multimèdia. Tot i així, les xarxes actuals tenen certes mancances per a ser utilitzades amb aquest propòsit.

Aquest article presenta el disseny i la implementació d'una nova aplicació de replicació de paquets per a salvar les limitacions imposades per la capa de xarxa. També es presenten els primers tests de rendiment realitzats amb aquesta eina i dues aplicacions satisfactòries on se'n fa ús.

1 Introducció

En els últims anys, gràcies al ràpid desenvolupament i implantació de les xarxes de dades d'alta velocitat, el paradigma de la distribució de continguts multimèdia ha canviat. Cada cop més els operadors tendeixen a transportar les dades sobre xarxes IP i inclús trobem exemples clars de distribució a usuaris finals a través d'Internet (3alacarta, rtve.es) o de xarxes d'operadors (Imagenio, Jazztelia).

Si a això sumem els avenços en tecnologies de codificació de vídeo i àudio i la seva forta orientació a Internet, és fàcil descobrir que la tendència del sector és clara a favor del nou paradigma de distribució.

A més a més, l'aparició de noves aplicacions de col·laboració a través de la xarxa també ha comportat un canvi en les funcionalitats requerides a la xarxa, cosa que sovint no és possible degut a les restriccions imposades per part dels operadors.

En aquest article es plantegen les problemàtiques derivades del nou paradigma de la distribució de continguts multimèdia. A continuació s'expliquen les motivacions que van comportar el desenvolupament d'una nova eina de replicació de paquets UDP i es presenta el seu disseny. Finalment es mostren els primers resultats de rendiment obtinguts en les proves del nou software i es plantegen les possibles línies de futur del projecte.

2 Problemàtica

La distribució de continguts multimèdia en temps real sobre xarxes IP s'acostuma a fer sobre fluxos de RTP¹ transportats en datagrames UDP². L'ús d'aquesta pila de protocols comporta normalment una comunicació entre dues parts de manera unidireccional: la font envia el contingut directament a l'usuari.

En entorns massius de distribució (anomenats sovint de broadcast) de continguts multimèdia sobre xarxes de dades, apareix la necessitat d'entregar un mateix contingut (representat com a flux de dades) a tots els destinataris de la informació.

Una primera solució a aquest problema consistiria en l'ús dels grups multicast a la capa de xarxa, però aquesta funcionalitat no està disponible en la majoria xarxes comercials públiques (com és el cas d'Internet). Això suposa un greu impediment per als distribuïdors de continguts, ja que requereixen d'acords amb operadors de xarxa per a disposar-ne, cosa que augmenta els seus costos i dificulta la vessant tècnica.

Tot i així, l'ús de grups multicast també presenta una sèrie de problemàtiques a resoldre. La més important consisteix en la seva rigidesa de funcionament: tot usuari subscrit a un grup rebrà tota la informació destinada a aquell grup, sense possibilitat d'altres comportaments si només s'utilitza un sol grup.

La solució doncs passa per a dissenyar estratègies a nivell d'aplicació que permetin l'entrega de paquets de manera lògica i flexible, permetent diferents comportaments i funcionant sobre xarxes públiques sense capacitats multicast.

3 Antecedents

En un primer moment, una cerca sobre les diferents possibilitats de codi obert d'aquest tipus d'aplicacions ens va portar a l'estudi d'una eina anomenada *rum* [3], desenvolupada per la Loughborough University. Aquest software es definia com a *un replicador de paquets unicast simple*, amb l'objectiu de ser utilitzat en entorns de videoconferències multipunt.

A partir d'aquesta eina, el CESNET [4] va desenvolupar un entorn complet de videoconferències multipunt d'alta qualitat [5]. Aquest entorn consistia en una xarxa mallada de replicadors de paquets, d'una lògica de control d'aquests replicadors a partir d'un nou protocol (RAP³) i clients per a reproduir els fluxos multimèdia rebuts i gestionar les sessions.

Tot i el bon funcionament d'aquesta eina, el seu ús estava molt orientat a entorns de videoconferències. Això feia que sempre s'imités el comportament dels grups multicast, reenviant tots els paquets a tots els usuaris actius en el replicador. A més

¹Real-time Transport Protocol [1]

²User Datagram Protocol [2]

³Reflector Administration Protocol [6]

a més, pel seu funcionament intern es forçava a que tot usuari que volgués rebre els fluxos multimèdia havia de ser també font d'informació.

4 Solució proposada: strManager

En l'àmbit de la Fundació i2CAT va aparèixer la necessitat de disposar d'una eina més flexible, que permetés la definició de diferents comportaments per a adaptar-se a diverses situacions. Una de les línies de treball està centrada en videoconferències multipunt en alta definició, on es requereix d'un element encarregat de fer arribar el vídeo i l'àudio generat per cada participant a la resta.

La Fundació també participa en la realització dels events de l'Anella Cultural [7], on els diferents actes són retransmesos des de qualsevol dels seus punts a tota la resta i on es creen retransmissions a partir de diferents fluxos multimèdia. Aquest cas representa un clar exemple de gestió i distribució de continguts multimèdia en temps real.

Degut a les limitacions que presentava el funcionament de l'eina *rum* i davant la falta d'alternatives obertes, es va plantejar la necessitat de començar el desenvolupament d'una eina més flexible des de zero. Amb aquesta intenció va néixer l'*strManager*, un replicador de paquets flexible i amb l'objectiu de ser capaç de replicar fluxos de gran ample de banda, com podrien ser vídeos en alta definició o d'alta qualitat, amb diferents comportaments.

A partir de l'anàlisi de les necessitats, s'ha dissenyat una eina que funciona a mode de router IP però a nivell de capa de transport, definint rutes d'entrada de paquets i les seves destinacions finals. Aquesta eina està ideada per capes, de manera que es defineix un pla de *media* molt senzill i ràpid que tracta els fluxos de paquets; i un pla de control que gestiona i configura els fluxos d'entrada i de sortida.

4.1 Disseny

La primera proposta de disseny de la nova aplicació intenta ser el més modular possible (veure Figura 1), de manera que pugui funcionar com a servei per a altres mòduls de software que oferissin serveis d'alt nivell com gestió d'usuaris, diferents protocols de senyalització (SIP⁴, H.323, MEGACO⁵) o inclús monitorització SNMP⁶.

El fet d'utilitzar mòduls d'expansió per a oferir serveis d'alt nivell, en lloc d'integrar-los dins del replicador de paquets en si, fa que aquest pugui ser molt simple, flexible i més ràpid. A més a més, permet integrar-lo en nous desenvolupaments a través de la seva interfície pública.

⁴Session Initiation Protocol [8]

⁵Media Gateway Controller [9]

⁶Simple Network Management Protocol [10]

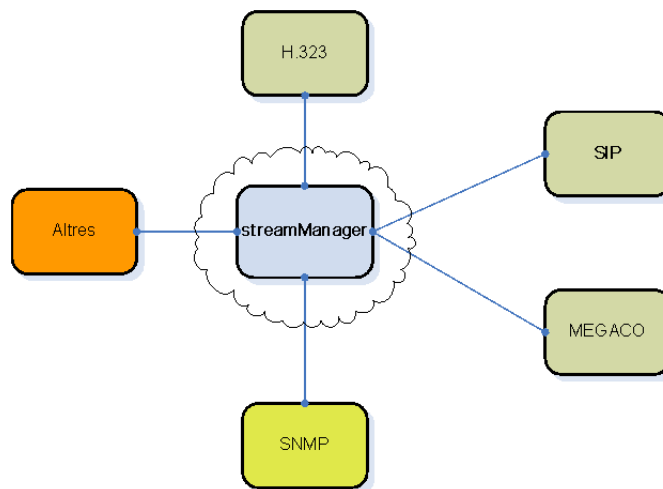


Figura 1: Disseny modular

Tal i com s'ha comentat anteriorment, per una banda, en el pla de *media* (*socketWorker*), la única i exclusiva funcionalitat és la lectura de paquets UDP i la seva retransmissió cap a cada un dels destinataris definits per a aquell flux d'entrada. Cal remarcar que tot el disseny d'aquesta eina està centrat en l'optimització d'aquest procés, de manera que sigui extremadament ràpid i permeti el tractament de fluxos d'un gran ample de banda.

D'altra banda, el pla de control (*streamManager*) consisteix en una sèrie d'estructures que permeten gestionar el pla de *media* de forma eficient. Bàsicament es controla l'obertura de sockets i la definició de llistes d'entrada i sortida de paquets (*streamList*), oferint a una interfície de configuració pràctica. Bàsicament s'ofereixen mètodes d'inserció i eliminació de fluxos d'entrada i fluxos de sortida (*udpStream*).

El desenvolupament d'aquesta eina es va fer de forma íntegra en C++, utilitzant les llibreries STL⁷ i Boost [11] per a la gestió de memòria, d'objectes i de *threads*. Tot i així i per tal d'optimitzar-ne el rendiment, es va decidir utilitzar la gestió nativa de socket per a sistemes GNU/Linux. Això fa que l'aplicació sigui només compatible amb sistemes d'aquest tipus, tot i que l'adaptació a altres plataformes no suposaria un gran esforç.

4.1.1 streamManager

Aquest representa l'objecte que ofereix la interfície externa de configuració del replicador de paquets. Permet la gestió de fluxos UDP entrants i sortints. Gràcies al seu ús, és possible aïllar a les lògiques que utilitzen aquesta eina de la gestió de *threads* i *sockets*.

⁷Standard Template Library

Bàsicament, aquesta capa ofereix 4 mètodes públics que permeten la configuració ràpida i senzilla del replicador de paquets:

addSource(identificador, IP_origen, port_destí) crea un *socket* per a rebre un flux multimèdia provinent de la IP origen i rebut per el port destí.

removeSource(identificador) elimina el flux entrant corresponent l'identificador i tots els fluxos de destí creats per a ell.

addDestination(identificador_origen, identificador, IP_destí, port_destí) afegeix un destí (adreça IP i port) per al flux entrant corresponent a l'identificador d'origen.

removeDestination(identificador_origen, identificador) elimina el destí corresponent al identificador del flux corresponent a l'identificador origen.

4.1.2 socketWorker

Aquest és l'objecte que representa els *threads* responsables de la lectura i replicació de paquets amb les ordres rebudes per l'*streamManager* (veure Figura 2).

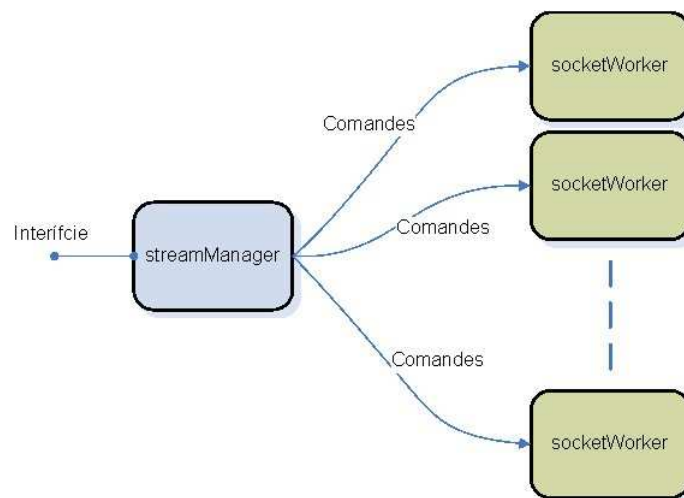


Figura 2: streamManager i socketWorkers

Cada objecte pot gestionar múltiples fluxos provinents d'adreces IP diferents i rebuts per diferents ports. La informació referent als fluxos d'entrada i sortida és configurada en temps d'execució, cosa que representa la gran diferència en front a les eines similars esmentades anteriorment i dóna la flexibilitat requerida. És possible també oferir el servei en forma de múltiples *threads*, de manera que múltiples processadors i/o nuclis puguin ser utilitzats al màxim.

Aquest objecte rep les crides dels mètodes de la interfície pública per a la gestió de fluxos entrants i sortints i n'emmagatzema la informació en objectes *udpStream* i *streamList*. Gràcies a aquesta informació és possible crear els *sockets* necessaris per a rebre els fluxos entrants i replicar-los als destins configurats.

4.1.3 udpStream

Aquests objectes representen un flux de dades entrant o sortint i són creats en el moment en que s'afegeix un origen o un destí. La informació que contenen és un identificador únic, una adreça IP i un port UDP.

En el cas de fluxos entrants, l'adreça IP representa l'adreça d'origen dels paquets i el port el port local per el que es reben els paquets. En el cas de fluxos sortints, tant l'adreça IP com el port representen les dades de destí.

4.1.4 streamList

Aquests objectes són agrupacions en forma de llista d'objectes *udpStream*. Cada llista representa un flux d'entrada i tots els fluxos de sortida corresponents a aquell flux d'entrada (veure Figura 3). El fet d'agrupar els destins en una mateixa llista fa que la replicació sigui molt ràpida, ja que cal només recórrer aquesta llista per a aconseguir la informació dels destins.

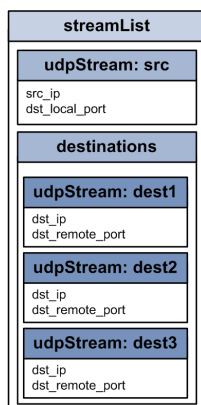


Figura 3: Objecte streamList agrupant udpStreams

Aquestes llistes són creades dins dels objectes socketWorker i per a cada paquet entrant, se'n busca la llista del qual n'és origen. Posteriorment es recorre tota la llista de destinacions i s'hi reenvia el contingut del paquet original.

4.2 Resultats preliminars

Per tal de determinar el rendiment que podia oferir *strManager*, vam disposar l'escenari mostrat a la Figura 4. Tots els ordinadors utilitzats eren ordinadors de treball amb capacitats computacionals no excessivament altes, excepte l'ordinador sobre el que funcionava el replicador de paquets, que era un doble AMD Dual Opteron amb 4 Gbytes de memòria RAM i sistema operatiu GNU/Linux Ubuntu Server 8.04.

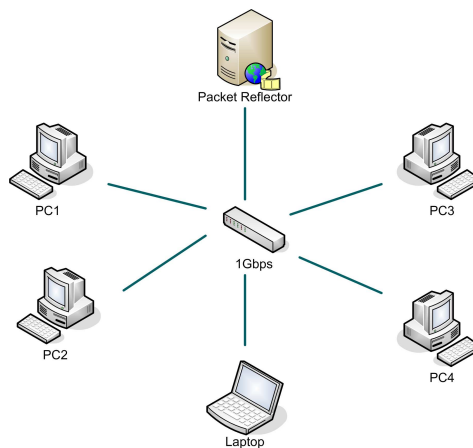


Figura 4: Escenari de proves

A partir d'aquest escenari, vam configurar el replicador de paquets per crear un esquema de fluxos com el mostrat a la Figura 5 i vam utilitzar *VideoLAN* [12] als PCs per a generar fluxos RTP a partir de vídeos en alta definició. Aquests vídeos eren MPEG-2 amb resolucions de 1080 línies progressives (1920x1080i) amb un *bitrate* de 25 Mbps.

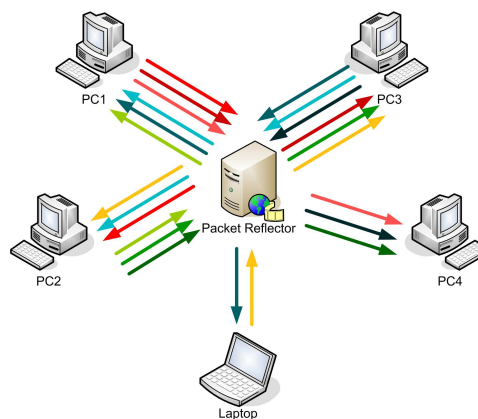


Figura 5: Fluxos a l'escenari de proves

Comptant el número de fluxos generats (23) i l'ample de banda que representa cada un d'ells (25 Mbps), obteníem un ample de banda total gestionat de 575 Mbps sense cap pèrdua en la recepció dels fluxos multimèdia (determinat a partir de les estadístiques ofertes per *VideoLAN*). En aquesta situació, la càrrega de l'ordinador que oferia el servei de replicació de paquets era sempre inferior al 50% d'un dels nuclis de les CPUs.

A continuació, varem afegir un altre flux UDP generat amb *iPerf*⁸ que anava del PC1 al PC4 a través del replicador de paquets. Aquest flux representava una taxa constant de 150 Mbps entrant i sortint al replicador, que afegida als 575 Mbps anteriors representava un ample de banda total gestionat d'uns 875 Mbps. Tot i així, la càrrega de l'ordinador seguia sent inferior al 50% d'un dels nuclis de les CPUs.

Adicionalment es van realitzar mesures sobre els retards afegits en els fluxos que passen pel replicador [14]. Per a fer-ho, es van afegir marques de temps en el codi del processat de cada paquet, calculant el temps entre la recepció i l'enviament de cada paquet. Els resultats obtinguts van ser que el replicador de paquets afegia un retard d'entre 50 μ s i 70 μ s, depenent de la longitud de la llista de destinacions per a cada flux d'entrada.

Un cop arribats a aquest punt, l'escenari de proves disposat quedava limitat per la velocitat de 1 Gbps de la xarxa. Tot i així, vam poder disposar durant un breu període de temps d'equipament amb xarxa de 10 Gbps i vam comprovar que el límit del replicador de paquets era superior a 1 Gbps de tràfic gestionat.

Mentre vam disposar d'aquest equipament (Doble Xeon 3.0 GHz, 2 GByte de memòria RAM, targeta de xarxa 10 Gbps Ethernet i sistema operatiu GNU/Linux Debian 5.0), vam realitzar tests amb fluxos RTP de 900Mbps, generats amb *UltraGrid*⁹. Vam disposar un escenari on un flux entrant era replicat a dues destinacions, representant un ample de banda total gestionat de 2,7 Gbps. En aquesta situació, la càrrega de l'ordinador que oferia el servei de replicació de paquets era un altre cop inferior al 50% d'un dels processadors, la qual cosa ens porta a pensar que aquest no era encara el límit de la nova eina.

5 Casos d'ús

Després de les primeres proves amb la nova eina i de comprovar el seu correcte funcionament, es varen crear diferents lògiques de control del replicador de paquets per adaptar-lo a les aplicacions per a les que va ser dissenyat.

⁸*Iperf* [13] és un software que permet l'enviament de fluxos de dades amb les característiques d'ample de banda específicats. Normalment s'utilitza per a fer tests del funcionament de xarxes.

⁹*UltraGrid* [15] és un software d'enviament i recepció de vídeo sense compressió, que genera fluxos d'entre 800 Mbps i 1,5 Gbps per a imatges en alta definició.

5.1 Videoconferències multipunt

Tal i com s'ha comentat anteriorment, una de les línies de treball de la Fundació i2CAT està centrada en videoconferències multipunt, intentant oferir servei per a multitud de fluxos multimèdia heterogenis (vídeo en alta definició o en 3D, múltiples canals d'àudio d'alta qualitat, gràfiques d'aparells mèdics i un llarg etcètera).

Un dels majors problemes en les videoconferències multipunt és la necessitat de que els fluxos multimèdia (generalment vídeo i àudio) generats per cada participant arribin a tota la resta de punts de la conferència. Si a això sumem l'absència de multicasting en xarxes públiques, cal trobar solucions alternatives.

La primera aproximació és la de crear múltiples connexions entre parelles de clients (tal i com mostra la Figura 6), fent que cada un d'ells sigui l'encarregat d'enviar els seus fluxos a tota la resta. Aquesta solució obliga a tots els participants de la reunió a tenir un gran ample de banda en el canal de pujada, cosa que normalment no està disponible (sobretot pensant en connexions asimètriques com ADSL).

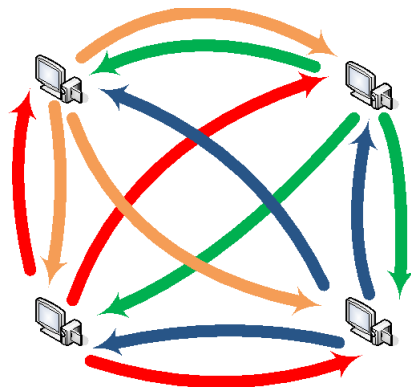


Figura 6: Multiconferència amb parelles punt a punt

La segona aproximació consisteix en afegir un element central, anomenat MCU¹⁰, que rebi els fluxos de tots els participants i s'encarregui de fer arribar la informació a tota la resta. Dins d'aquesta solució existeixen dues propostes clarament diferenciades: l'ús de transcodificadors multimèdia o l'ús de replicadors de paquets.

La primera proposta comporta una gran complexitat computacional, ja que la MCU ha de rebre els fluxos multimèdia de tots els participants i fer una composició en temps real a partir de tots ells per a crear un sol flux (veure Figura 7). El resultat obtingut és un sol flux de vídeo i àudio, que requereix un ample de banda contingut, però que limita l'apreciació de detalls degut a la reducció de les imatges per a ser mostrades en una sola pantalla. A més a més, el fet de fer un tractament intensiu sobre la *media*, afegeix un retard que pot ser crític si es pretén oferir interac-

¹⁰Multi-point Control Unit

ció entre usuaris i requereix de computadors molt potents, normalment amb costos d'adquisició molt elevats.

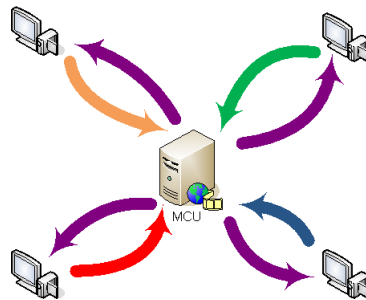


Figura 7: Multiconferència amb composició de fluxos a la MCU

La segona proposta afronta el problema de manera totalment diferent: la MCU no ha de fer cap tractament sobre els fluxos multimèdia, simplement els ha de reenviar a la resta de participants (veure Figura 8). L'ample de banda necessari en aquest cas és més elevat, però permet a l'usuari mostrar els diferents fluxos de vídeo en diferents pantalles i apreciar-ne els detalls. Al no haver-hi cap mena de processat sobre les dades, el retard afegit és mínim i permet oferir una gran interacció entre els diferents punts de la conferència. Finalment, el servei pot ser ofert des d'un servidor de baixes prestacions (i, en conseqüència, baix cost), ja que l'absència de processat fa que no sigui necessari.

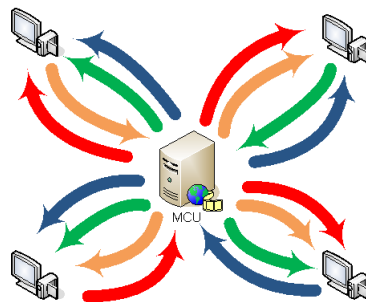


Figura 8: Multiconferència amb replicació de paquets a la MCU

A partir d'aquesta última opció, es va desenvolupar una capa de control sobre el replicador de paquets basada en SIP [14]. La idea era crear un servei complet de videoconferències establertes a través de SIP i SDP i que fos capaç d'operar amb fluxos de *media* heterogenis i de gran ample de banda.

L'elecció de SIP no és casual, ja que la seva implantació en el món de VoIP¹¹ fa que ja existeixi la infraestructura necessària per al seu ús. A més a més, en combinació

¹¹Voice over IP, Veu sobre IP

amb l'SDP, esdevé un mecanisme molt flexible per l'establiment i control de sessions multimèdia, com per exemple una videoconferència.

La capa SIP funcionava a mode d'un Server User Agent, acceptant trucades i negociant amb SDP els paràmetres multimèdia de la sessió (bàsicament adreces IP, ports, tipus de *media* i CODECs). A partir d'aquests paràmetres i utilitzant l'interfície oferta per *strManager*, es configuraven els fluxos d'entrada i sortida necessaris per a crear les conferències (tal i com es mostra a la Figura 9). Es va dotar al servei de la capacitat d'operar amb diverses sales simultànies, fent possible l'existència de diferents reunions al mateix temps utilitzant el mateix servidor.

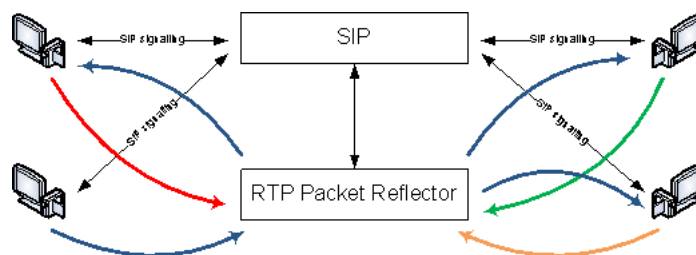


Figura 9: Capa SIP amb el replicador de paquets

A més a més, i a mode de recerca, es va implementar la possibilitat de substituir els fluxos RTP per fluxos SRTP¹² [17], oferint un servei de videoconferències segures. En aquest cas, en la negociació SDP es realitzava un intercanvi de claus amb el protocol MIKEY¹³ i s'autentiquen i xifren els fluxos RTP.

El bon funcionament d'aquesta aplicació ha estat comprovat en el marc del projecte HDVIPER¹⁴, on s'han definit les bases per a la creació d'una plataforma oberta i escalable de videoconferències en alta definició.

5.2 Anella Cultural

L'Anella Cultural [7] és un projecte de promoció i distribució de cultura a tot Catalunya. Gràcies a la interconnexió de diferents centres culturals (bàsicament museus i teatres) es promou la producció, l'experimentació, la creativitat, el intercanvi de coneixement i la circulació d'idees entre diferents territoris.

Tots els events de l'Anella són retransmesos en directe als altres punts, a través de xarxes heterogènies (Figura 10). Normalment s'utilitzen fluxos de vídeo DV¹⁵ (resolució PAL a 30Mbps) amb l'àudio integrat. Amb aquests fluxos és possible projectar els events en pantalles de grans dimensions i oferir una bona qualitat subjectiva per als assistents als actes.

¹²Secure Real-time Transport Protocol[16]

¹³Multimedia Internet KEYing [18]

¹⁴High Definition Videoconferencing over IP Environment [19]

¹⁵Digital Video

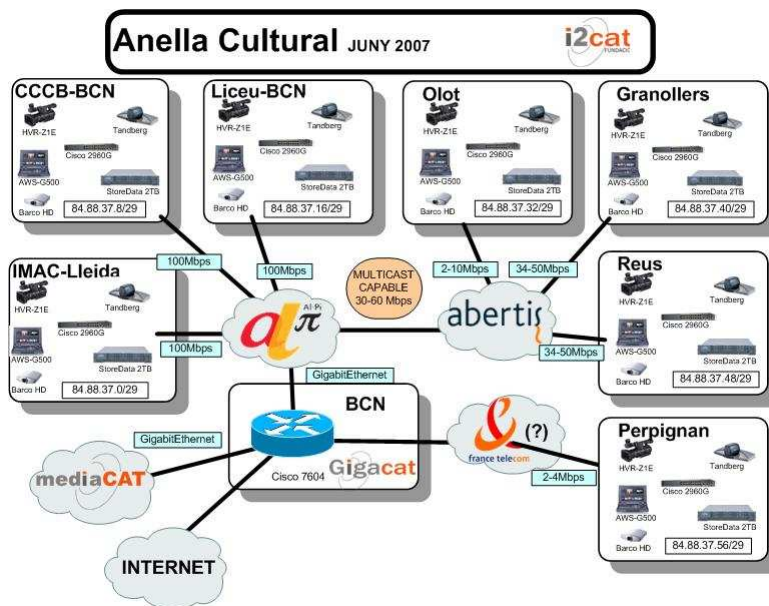


Figura 10: Esquema de transmissió d'un acte de l'Anella Cultural

El repte tecnològic en aquests events era de nou la distribució de continguts a múltiples punts sobre xarxes sense capacitats multicast. L'ús d'un replicador de paquets representava doncs una solució eficient i permetia adaptar-se a diferents situacions. A més a més, gràcies a la facilitat de reconfiguració de *strManager*, és possible fer una realització a nivell de fluxos de *media*, canviant les destinacions de cada flux d'informació segons el contingut que es vol mostrar.

6 Conclusions

Gràcies al desenvolupament d'aquesta eina, s'han pogut solucionar problemàtiques molt diverses plantejades en diferents projectes de la Fundació i2CAT.

El seu rendiment ha demostrat que la nova eina de replicació de paquets és prou potent per a tractar amb fluxos de gran ample de banda i prou flexible com per adaptar-se a diferents situacions. A més a més, aquesta flexibilitat, traduïda en la possibilitat de crear diferents lògiques de control de la replicació de paquets, ha obert les portes al desenvolupament de nous serveis.

Finalment, apuntar que no s'ha trobat encara el límit en el rendiment, tot i que les línies futures de treball pretenen fer-ho, i que hi ha plans per implementar noves funcionalitats, com la lectura o escriptura de fluxos multimèdia a disc o el control de taxa a partir de la lectura de capçaleres RTP o informes RTCP.

7 Llicència

Aquest article es distribueix sota una llicència Creative Commons Reconeixement-Sense obres derivades 2.5 Espanya. Veieu <http://creativecommons.org/licenses/by-nd/2.5/es/deed.ca> per més informació.

Els desenvolupaments de la nova eina està previst que siguin lliberats sota llicència Lesser General Public License (LGPL) [20] per tal de facilitar-ne el seu ús en múltiples entorns. Actualment el projecte es troba en la fase final de desenvolupament i en els propers mesos serà publicat a la comunitat.

Finançament

Aquest treball ha estat finançat parcialment per el Ministerio de Indústria, Turismo y Comercio en el marc del programa “Avanza I+D” (projecte TSI-020400-2008-147), el Ministerio de Ciencia e Innovación sota el projecte TSI2007-66637-C02-01 i per el Centro para el Desarrollo Tecnológico Industrial (CDTI) sota el projecte CENIT-VISION 2007-1007.

Referències

- [1] H. Schulzrinne, *et al.* *RTP: A Transport Protocol for Real-Time Applications*. Internet Engineering Task Force, RFC 3550. Juliol 2003.
- [2] J. Postel. *User Datagram Protocol*. Internet Engineering Task Force, RFC 768. Agost 1980.
- [3] J. Highfield. *UDP Packet Reflector Hacks, rum*. Disponible on-line a [<http://spirit.lboro.ac.uk/mug/mug.html>]. Última visita al 28-4-2009.
- [4] *CESNET - Czech NREN Operator*. Disponible on-line a [<http://www.ces.net>]. Última visita al 28-4-2009.
- [5] CESNET. *Synchronizing RTP Packet Reflector*. Disponible on-line a [<http://www.cesnet.cz/doc/techzpravy/2003/rtpreflector>]. Última visita al 28-4-2009.
- [6] J. Denemark, P. Holub, E. Hladká. *RAP: Reflector Administration Protocol*. CESNET Technical Report 9/2003, November 2003.
- [7] *Anella Cultural*. Disponible on-line a [<http://www.anellacultural.cat>]. Última visita al 8-5-2009.
- [8] J. Rosenberg *et al.* *SIP: Session Initiation Protocol*. Internet Engineering Task Force, RFC 3261, Juny 2002.

- [9] C. Groves *et al.* *Gateway Control Protocol Version 1*. Internet Engineering Task Force, RFC 3525, Juny 2003.
- [10] J. Case *et al.* *A Simple Network Management Protocol*. Internet Engineering Task Force, RFC 1157, Maig 1990.
- [11] *Boost C++ Libraries*. Disponible on-line a [<http://www.boost.org>]. Última visita al 28-5-2009.
- [12] *VideoLAN - VLC media player*. Disponible on-line a [<http://www.videolan.org>]. Última visita al 13-5-2009.
- [13] *Iperf*. Disponible on-line a [<http://iperf.sourceforge.net>]. Última visita al 28-5-2009.
- [14] G. Cabrera, E. Eliasson. *Secure High Definition Videoconferencing*. Telecommunications System LABORatory, Kungliga Tekniska högskolan, Juny 2008.
- [15] *UltraGrid* iGridWiki. Disponible on-line a [<https://www.sitola.cz/igrid/index.php/UltraGrid>]. Última visita al 28-5-2009.
- [16] M. Baugher *et al.* *The Secure Real-time Transport Protocol*. Internet Engineering Task Force, RFC 3711, Març 2004.
- [17] G. Cabrera, J. López *et al.* *MCU basada en SIP para videoconferencias seguras*. Jornadas Técnicas de RedIRIS 2008, Novembre 2008.
- [18] J. Arkko *et al.* *MIKEY: Multimedia Internet KEYing*. Internet Engineering Task Force, RFC 3830, Agost 2004.
- [19] *High Definition Videoconferencing over IP Environment (HDVIPER)*. Disponible on-line a [<http://www.hdviper.org>]. Última visita al 12-5-2009.
- [20] *GNU LESSER GENERAL PUBLIC LICENSE*, Versió 3, Juny 2007. Disponible on-line a [<http://www.gnu.org/copyleft/lesser.html>]. Última visita al 4-6-2009.