



Catra Streaming Platform

Red Hat Magazine 2004 N° 2

Una piattaforma di streaming per il mondo mobile (UMTS, GPRS) e non solo

Oggi, trovare un sito web che non includa un qualche tipo di streaming di media è veramente difficile. Basti pensare alle radio su Internet, ai trailer di film oppure all'insegnamento a distanza. È inoltre importante evidenziare che Internet non è stato progettato per trasmettere media né per supportare un numero di utenti così elevato come quelli che oggi si connettono alla Rete.

Il 1993 è stato l'anno del rilascio di Mosaic, il primo browser per il Web che ha notevolmente contribuito alla diffusione della Rete. Internet è diventato così un modo efficiente di condividere qualunque tipo di file, inclusi file audio e video. Queste nuove tipologie di file, sfortunatamente, erano molto più grandi dei normali file di testo. Con la rapida crescita della popolarità del Web, inoltre, si è verificato un grande incremento del traffico sulla rete che obbligava gli utenti ad aspettare a lungo per scaricare e mandare file. Il "World Wide Web" è diventato il "World Wide Wait" a causa del traffico crescente sulla rete. L'attesa era, quindi, particolarmente dolorosa per coloro che volevano ascoltare file audio e video a causa delle dimensioni di queste tipologie di file. Il problema principale era che l'utente doveva aspettare il completo trasferimento del file sul proprio hard disk prima di poterlo ascoltare.

La tecnologia dello streaming offre un nuovo approccio ai media su Internet. Invece di aspettare che l'intero file sia trasferito sul computer dell'utente prima di iniziare il playback, lo streaming consente la riproduzione del file non appena inizia il suo trasferimento. I byte viaggiano attraverso la rete, viene eseguito il playback e poi vengono cestinati. Inoltre la tecnologia dello streaming permette di controllare i media durante il playback con comandi quali "stop", "play", "pause", "rewind" e "forward". Lo streaming può essere utilizzato sia per eventi live (come per esempio un concerto in tempo reale), on-demand e broadcast, sia per file già archiviati (streaming off-line). Lo streaming è in ogni modo condizionato dalla banda disponibile: poiché avviene in real-time, il numero

di byte che è possibile inviare dipende dalla capacità del canale che riceve.

Componenti di un sistema di streaming

Lo streaming di media è possibile grazie a differenti componenti software che comunicano tra di loro.

Un sistema di streaming di base è composto da tre componenti:

- Player: software usato per vedere e ascoltare una sessione di streaming.
- Server: software che realizza lo streaming mandando gli stream in pacchetti.
- Encoder: software che converte il flusso audio e video grezzo (non trattato) in un formato che può essere trasmesso in streaming.

Questi componenti comunicano tra di loro usando particolari protocolli e formati di file. Alcuni file contengono stream codificati secondo un particolare codec che rappresenta un algoritmo progettato per ridurre la dimensione di un file.

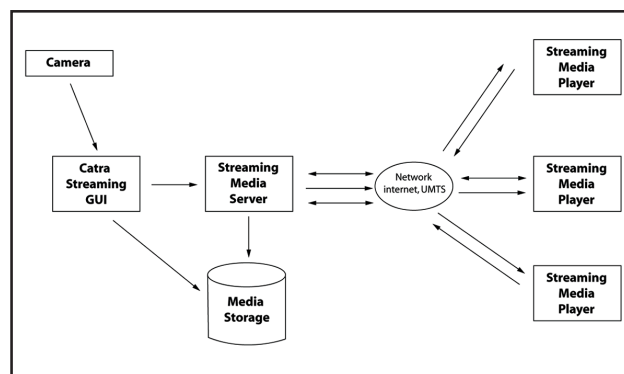


Figura 1: Illustrazione di un diagramma dei componenti, sopra citati e a come essi interagiscono in un sistema di streaming.

Prima che qualcuno possa guardare o ascoltare lo streaming di media, il flusso audio e video grezzo deve

essere convertito in un formato che può essere trasmesso in streaming. Per questa operazione viene utilizzato lo **Streaming Media Encoder**. L'encoder riduce la quantità di informazioni contenute nel file grezzo cercando di mantenere il più possibile la qualità del flusso originale.

Il flusso prodotto dall'encoder viene ricevuto dallo **Streaming Media Server** (direttamente dall'encoder in caso di eventi live, tramite il Media Storage in caso di streaming off-line) che concettualmente si comporta come un web server. Entrambi sono progettati per servire un grande numero di richieste, ma mentre il web server instaura una comunicazione con il client solo per il tempo necessario per la consegna di una pagina web, lo Streaming Media Server mantiene la comunicazione attiva per tutta la durata della sessione di streaming.

Lo Streaming Media Player è un software che sa come comunicare con lo Streaming Media Server e come eseguire il playback degli stream ricevuti. Lo Streaming Media Player generalmente offre un controllo interattivo sugli stream che consente di eseguire comandi come "stop", "pause", "rewind" e "forward".

☞ Catra Streaming Platform (CSP)

Catra Streaming Platform comprende una piattaforma di streaming aperta ed estendibile, un'interfaccia grafica che centralizza la gestione di tutti i server distribuiti nel territorio e una serie di tool per la gestione di file MP4.

◆ Architettura

Il Catra Streaming Server è basato completamente su architetture e standard aperti. Aderisce a numerosi protocolli standard 3GPP e Internet come RTSP, RTP/RTCP, SDP. Questi standard sono descritti nel paragrafo 'La tecnologia dello streaming'.

Quando il Catra Streaming Server viene eseguito, legge un file di configurazione per ottenere numerosi parametri, tra cui il path dove si trovano i file media, il path per i file di log e altri. Così, comincia ad ascoltare sulla scheda di rete e su una specifica porta per eventuali richieste di streaming. Lo Streaming Media Player inizia una sessione di streaming fornendo al server un URL RTSP al contenuto di cui si vuole eseguire lo streaming. Il player stabilisce una connessione su TCP/IP, richiede e controlla lo stream attraverso metodi RTSP, quali ad esempio SETUP, PLAY, PAUSE. Durante lo streaming, server e player comunicano periodicamente tramite il protocollo RTCP, scambiandosi informazioni statistiche riguardo la sessione di streaming. Il Catra Streaming Server manterrà tutte le informazioni che riguardano la sessione di streaming nei suoi file di log ai quali assicura dimensione, rotazione e compressione appropriate. Per motivi di sicurezza, il Catra Streaming Server gestisce anche una serie di timeout per evitare che le sue risorse rimangano definitivamente allocate senza mai essere liberate. Il Catra Streaming Server usa i seguenti protocolli:

- RTSP per il controllo della sessione;

- SDP per fornire al player le informazioni sul media richiesto;
- RTP per pacchettizzare lo stream da mandare al player;
- RTCP per comunicare e monitorare informazioni statistiche che riguardano gli stream.

La Catra Streaming GUI è una applet java che in fase di esecuzione riceve un file di configurazione con tutte le informazioni riguardo i server e la loro distribuzione sul territorio. Comunica, tramite CORBA, con ogni server per reperire informazioni su ognuno di essi.

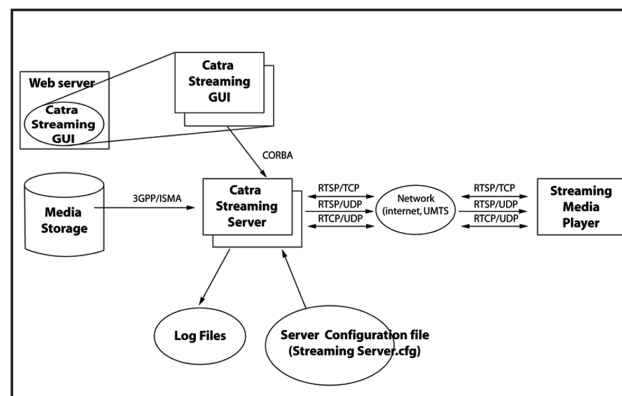


Figura 2: Descrizione dell'architettura della Catra Streaming Platform.

◆ Catra Streaming Server

Catra Streaming Server segue gli standard **3GPP** e **ISMA** ed è in grado di comunicare con qualunque player che aderisca a questi standard come il player 3GPP di Philips (<http://www.digitalnetworks.philips.com/InformationCenter/Global/FHomepage-NoXCache.asp?lNodeId=756&channel=756&channelId=N756A2167>), il player ISMA MPEG4IP (<http://mpeg4ip.sourceforge.net/>) e ogni player 3GPP installato sui terminali UMTS.

Il server di streaming è scritto utilizzando il linguaggio Ansi C++ e rappresenta anche un server CORBA per rispondere alle richieste avanzate dall'interfaccia grafica. L'implementazione CORBA utilizzata è Mico (MICO Is CORBA: implementazione gratuita e completamente conforme dello standard CORBA).

A oggi il server è in grado di eseguire lo streaming di file MP4 e successivamente supporterà anche lo streaming di eventi live.

I codec audio attualmente supportati sono AAC, GSMAMR, WBAMR, mentre quelli video sono MPEG4 e H263.

Di seguito sono elencati gli **standard** a cui il Catra Streaming Server aderisce:

- 3GPP TS 26.234 - PSS Protocols and codecs (Release 5)
- rfc2326: Real-Time Streaming Protocol (RTSP)
- rfc2327: Session Description Protocol (SDP)

-rfc1889: A Transport Protocol for Real-Time Applications (RTP)
 -rfc2429: RTP Payload Format for the 1998 Version of ITU-T Rec. H.263 Video (H.263+)
 -rfc3016: RTP Payload Format for MPEG-4 Audio/Visual Streams
 -rfc3267: Real-Time Transport Protocol (RTP) Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs

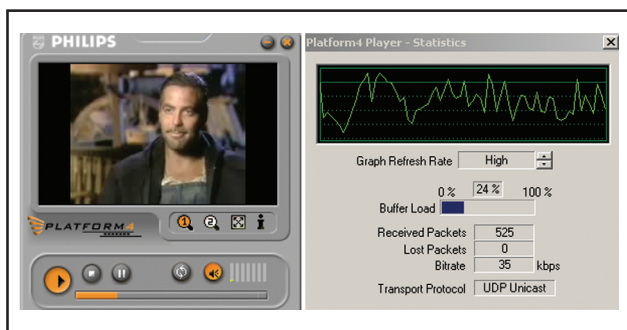


Figura 3: Player Philips durante una sessione di streaming.

◆ Catra Streaming GUI

L'interfaccia grafica centralizza la gestione e l'amministrazione di tutti i server di streaming distribuiti nel territorio. È scritta in Java e comunica coi server di streaming tramite CORBA e l'utilizzo di JacORB (l'implementazione Java gratuita dello standard CORBA di OMG).

Le principali funzionalità di questo componente sono:

- Configurazione del sistema
- Gestione dei contenuti
- Monitoring dei server
- Generazione di report statistici
- Gestione dei log

Per rappresentare tutti i siti in cui la piattaforma di streaming è distribuita, si utilizza una struttura ad albero. Ogni sito avrà un insieme di server di streaming, ciascuno dei quali avrà la sua specifica configurazione. Nella Figura 4 troviamo il server di streaming chiamato 'Piazza Fiera' con la sua configurazione all'interno del sito 'Trento'. 'Piazza Fiera' è ovviamente il nome logico che è associato al server di streaming.

Sulla sinistra troviamo gli stessi siti e server di streaming che abbiamo visto nella configurazione (vedi Figura 4). Tramite questa finestra è possibile monitorare le richieste di uno specifico server, di tutti i server di un sito oppure di tutti i server di tutti i siti. Selezionando il server 'Piazza Fiera' è possibile anche monitorare lo stato del server tramite informazioni per esempio sulla memoria o la percentuale di CPU usata, il numero di utenti connessi, la banda erogata e altro.

◆ Configurazione della piattaforma

La Catra Streaming Platform è configurata dai seguenti file di configurazione: GUIConfiguration.xml, GUIContent.xml, GUIMonitor.xml e StreamingServer.cfg.

I primi tre file sono usati dalla Catra Streaming GUI e risiedono su un web server. Non necessitano di alcun cambiamento perché sono completamente gestiti dalla stessa Catra Streaming GUI.

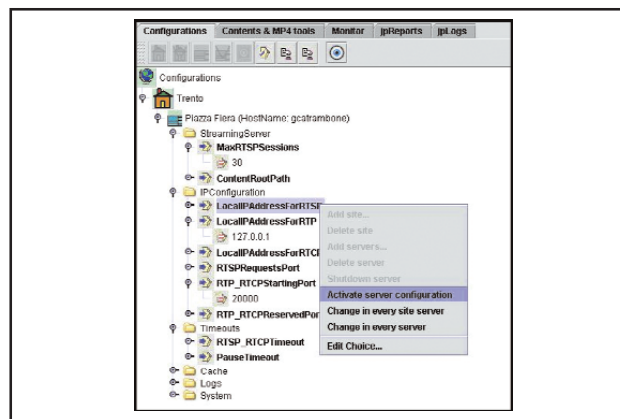


Figura 4: Mostra la finestra relativa alla configurazione del sistema.

Il file di configurazione StreamingServer.cfg permette la configurazione del server di streaming e normalmente può essere modificato attraverso l'uso della 'Catra Streaming GUI'. All'interno del file, solo un parametro non può essere modificato tramite l'interfaccia grafica: parliamo del nome logico che si vuole dare al server di streaming. Il file di configurazione deve essere modificato con un normale editor di testo nel caso in cui non si vuole installare la 'Catra Streaming GUI'.

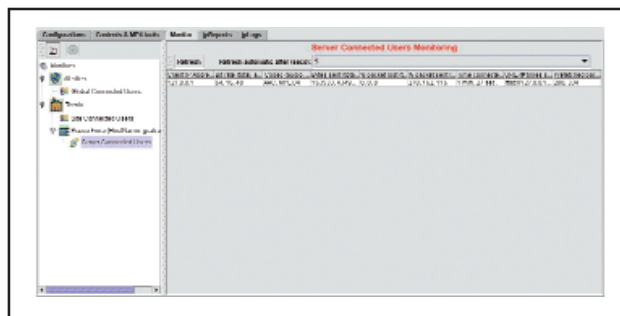


Figura 5: Descrizione della finestra GUI adibita al monitoring dei sistemi.

◆ Installazione ed esecuzione

Scaricate i file

CatraStreamingServer_v_1_45.tar.gz,
 CatraStreamingGUI_v_1_1.zip e
 mp4Tools_v_1_1.tar.gz dal sito
<http://members.xoom.it/catrambone/Streaming/CatraStreamingPlatform.htm>, sezione Download. Ogni componente della

piattaforma (Server, GUI e MP4 tools) è indipendente e, quindi, può essere installato e mandato in esecuzione indipendentemente dagli altri componenti.

Create la directory che ospiterà la Catra Streaming Platform:

```
-mkdir CatraStreamingPlatform
-cd CatraStreamingPlatform
```

D'ora in poi con STREAMINGPLATFORM_PATH s'indicherà il path della directory CatraStreamingPlatform appena creata.

Inizializzate la variabile d'ambiente

```
STREAMINGPLATFORM_PATH
```

Per una shell csh:

```
setenv STREAMINGPLATFORM_PATH
<...>/CatraStreamingPlatform/CatraStreamingServer_v_1_45/lib
```

Per una shell sh:

```
export STREAMINGPLATFORM_PATH
<...>/CatraStreamingPlatform/CatraStreamingServer_v_1_45/lib
```

◆ Installazione del Catra Streaming Server

```
-tar xvfz CatraStreamingServer_v_1_45.tar.gz
-aggiungete il path
$STREAMINGPLATFORM_PATH/CatraStreamingServer_v_1_45/lib alla variabile d'ambiente
LD_LIBRARY_PATH.
```

Per una shell csh:

```
setenv LD_LIBRARY_PATH
$LD_LIBRARY_PATH:$STREAMINGPLATFORM_PATH/CatraStreamingServer_v_1_45/lib
```

Per una shell sh:

```
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$STREAMINGPLATFORM_PATH/CatraStreamingServer_v_1_45/lib
```

◆ Installazione della Catra Streaming GUI

La GUI è stata testata con la JRE 1.4 perciò non è noto il suo comportamento con altre versioni di JRE.

Per installare la Catra Streaming GUI è necessario quindi: installare la JRE 1.4 (effettuare il download dal sito <http://java.sun.com/j2se/1.4.2/download.html>), estrarre i file da CatraStreamingGUI_v_1_1.zip nella directory CatraStreamingPlatform creata precedentemente, copiare il file \$STREAMINGPLATFORM_PATH/CatraStreamingGUI_v_1_1/security/jacorb.properties nella home directory dell'utente (~<username>) che eseguirà l'applet.

copiare il file \$STREAMINGPLATFORM_PATH/CatraStreamingGUI_v_1_1/security/orb.properties nella directory lib della JRE (<JRE1.4_HOME>\lib).
copiare il file \$STREAMINGPLATFORM_PATH/CatraStreamingGUI_v_1_1/security/CatraStreamingGUI.policy nella directory <JRE1.4_HOME>\lib\security.
-Modificare il file <JRE1.4_HOME>\lib\security\java.security. In particolare è necessario modificare il valore del campo 'policy.url.1' con il valore 'file:\${java.home}/lib/security/CatraStreamingGUI.policy'.

◆ Esecuzione

I processi da eseguire sono:

```
-nsd (Naming Service di CORBA)
-streamingServer (Catra Streaming Server)
-streamingGUI (Catra Streaming GUI)
```

Avviate il servizio CORBA 'Naming Service'. È necessario lanciare nsd.csh. Questo file batch avvia il naming service di CORBA la cui sintassi è:

```
nsd -ORBIIOPAddr inet:<NSD IP address>:<NSD port>
```

<NSD IP address> è l'indirizzo IP della macchina dove gira il processo nsd. Può anche essere 127.0.0.1 se gli altri processi gireranno sulla stessa macchina.
<NSD port> è una qualunque porta socket disponibile (ad esempio 12456).

Lanciate il Catra Streaming Server. È necessario lanciare CatraStreamingServer.csh. È possibile anche modificare il file di configurazione del Catra Streaming Server:

```
$STREAMINGPLATFORM_PATH
/CatraStreamingServer/cfg/streamingServer.cfg.
```

I parametri più importanti sono:

```
-Name (Default: CatraServer): rappresenta la chiave per identificare il server. È usato dalla Catra Streaming GUI per comunicare con il server.
-ContentRootPath (Default: .\Contents): è la document root del server dove devono essere inseriti i file MP4/3GPP.
-LocalIPAddressForRTSP (Default: 127.0.0.1): è l'indirizzo IP usato dal server per la comunicazione RTSP.
-LocalIPAddressForRTP (Default: 127.0.0.1): è l'indirizzo IP usato dal server per la comunicazione RTP.
-LocalIPAddressForRTCP (Default: 127.0.0.1): è l'indirizzo IP usato dal server per la comunicazione RTCP.
BaseTraceFileName (Default: .\trace\StreamingServerSystemLogs): rappresenta la base del nome del file di trace usata dal server per costruire i nomi dei file di trace.
```

Ora verrà avviata una sessione di streaming di un file 3GPP:

-Scaricate un file 3GPP dalla sezione Download del sito <http://members.xoom.it/catrambone/Streaming/CatraStreamingPlatform.htm> e copiatelo nella directory \$STREAMINGPLATFORM_PATH/CatraStreamingServer/Contents.
 -Eseguite il download e l'installazione del player 3GPP della Philips dal relativo sito (<http://www.software.philips.com/InformationCenter/Globa1/FArticleDetail.asp?lArticleId=2606>) su un sistema Windows. In alternativa si può usare il player ISMA MPEG4IP (<http://mpeg4ip.sourceforge.net/>).
 -Inserite nel player l'URL:
 "rtsp://127.0.0.1:7777/<file name>".

Avviate la Catra Streaming GUI:
 A questo punto, potete vedere il video.
 -fate doppio clic su streamingGUI.html;
 -eseguite il log alla GUI come 'guest'.

La tecnologia dello streaming

Il Catra Streaming Server fornisce media digitali così come un web server fornisce pagine web. Divide il media digitale in pacchetti e li manda al player. Il player ricostruisce i pacchetti e li renderizza.

♦ Cos'è lo streaming basato su file?

Il Catra Streaming Server consente lo streaming di file MP4 che sono disponibili al server tramite un filesystem locale o remoto. Il player manda una richiesta al Catra Streaming Server per l'esecuzione dello streaming di un file tramite il relativo URL e l'appropriata richiesta RTSP. Poiché un file MP4 può essere formato anche da più di due stream, il player deve richiedere ogni stream separatamente mandando un comando di SETUP per ogni stream. Quando il player esegue una richiesta di streaming di un file, il server identifica il file e lo analizza in modo

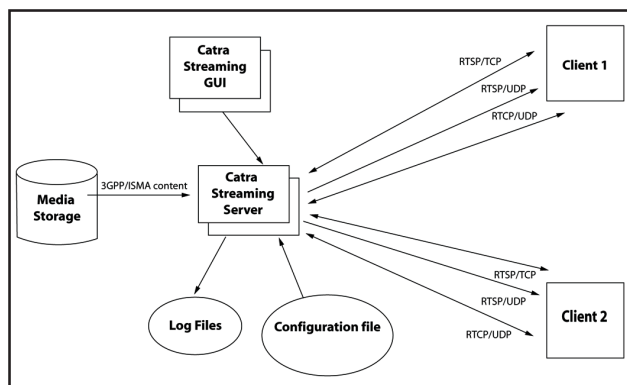


Figura 6: Descrive lo streaming basato su file.

da ottenere le informazioni di ogni stream contenuto nel file. Per ognuno di questi stream, il server legge i dati dal file e manda un flusso di pacchetti RTP al player. Inoltre per ogni stream, il server comunica regolarmente con il player inviando e ricevendo pacchetti RTCP che descrivono la qualità della comunicazione. Lo stream termina in corrispondenza della fine dell'esecuzione del media, a

meno che dal player non arrivi un comando di PAUSE oppure TEARDOWN. La comunicazione RTCP termina quando il player invia il comando TEARDOWN.

♦ Real-Time Transport Protocol (RTP)

Il Catra Streaming Server utilizza il protocollo RTP per mandare gli stream al player.

Usando le informazioni contenute all'interno del file MP4, il server divide gli stream in pacchetti RTP e li trasmette al player su un socket UDP.

Ogni pacchetto RTP trasmesso dal server comprende:

- l'header RTP
- l'header payload RTP
- dati payload RTP

L'header RTP tra l'altro comprende:

- informazioni sulla versione RTP
- numero di sequenza
- timestamp
- Synchronization Source Identifier (SSRC)

Il server usa la versione 2 del protocollo RTP. Il numero di sequenza per il primo pacchetto di una sessione RTP è generato in modo random dal server. I successivi pacchetti avranno un numero di sequenza che viene incrementato secondo una funzione monotona. I numeri di sequenza sono fondamentali per il player per capire l'ordine esatto in cui deve gestire i pacchetti RTP, e anche se qualche pacchetto si è perso. Ogni pacchetto RTP contiene inoltre il timestamp che indica il tempo esatto in cui i dati (dati payload RTP) devono essere renderizzati dal player. Come per il numero di sequenza, anche il timestamp del primo pacchetto RTP è generato in modo random dal server.

Il campo 'SSRC' è generato in modo random dal server ed è incluso in ogni pacchetto RTP. Esso serve al player per identificare univocamente da chi ha ricevuto il pacchetto RTP.

Il server, che inizia una sessione RTP, assicura che vengono allocate due porte socket per ogni stream (una per RTP e l'altra per RTCP). La porta per RTP è sempre un numero pari mentre la porta per RTCP è il successivo numero dispari. Per esempio, 13240 e 13241.

Il server costruisce e manda i pacchetti RTP tenendo conto dei tempi della presentazione. Per migliorare la qualità dello stream, il server costruisce in anticipo un numero configurabile di pacchetti RTP.

♦ Real-Time Transport Control Protocol (RTCP)

Il protocollo RTCP (Real-Time Control Protocol) viene utilizzato per la comunicazione di informazioni statistiche dello stream tra player e server.

In una sessione di streaming

- Il server manda RTCP sender report per fornire al player informazioni riguardanti lo stream (per esempio informazioni sulla sincronizzazione tra intra e inter stream).
- Il player manda RTCP receiver report per fornire

informazioni riguardanti la qualità di ciò che ha ricevuto. Queste informazioni possono essere utilizzate dal server per migliorare la qualità dello stream.

◆ Real-Time Streaming Protocol (RTSP)

Il server implementa il Real-Time Streaming Protocol (RTSP) su un socket TCP. Il server utilizza i metodi del protocollo RTSP per controllare lo streaming della presentazione.

La porta socket utilizzata dal server per il protocollo RTSP è di default la 554. Questa porta può essere riconfigurata.

Quando il player inizia una nuova connessione RTSP, il server verifica che il numero massimo di connessioni RTSP non sia stato già raggiunto. Una volta che il server si è accertato di poter accettare la connessione, si mette in ascolto di richieste RTSP.

Il server interpreta i metodi, gli attributi e l'header del protocollo RTSP per controllare la presentazione multimediale. I metodi del protocollo RTSP che il player manda al server sono:

-OPTIONS - Il server risponde con la lista dei metodi RTSP supportati dal server.

-DESCRIBE - Il server identifica il file indicato dall'URL e risponde con informazioni sul media da descrivere. Queste informazioni vengono presentate dal server tramite il protocollo SDP.

-SETUP - Il player identifica le tracce del media di cui vuole ricevere gli stream (ogni stream tipicamente corrisponde a una differente tipologia di media: audio e video). Il server cerca la sessione RTP associata a questa richiesta e se non la trova (qualora il player non avesse precedentemente mandato la DESCRIBE), crea una sessione RTP nuova. Alla fine il server restituisce lo stato dell'operazione di SETUP con un 'session ID' se l'operazione ha avuto successo.

-PLAY - Questa richiesta inizia il trasferimento dei dati per ogni stream specificato dal player nei precedenti comandi di SETUP. I pacchetti RTP vengono costruiti in anticipo per ogni stream della sessione RTSP. Il server restituisce al player il numero di sequenza e il timestamp del primo pacchetto RTP di ogni stream.

-PAUSE - Questa richiesta indica al server di bloccare temporaneamente il trasferimento dei dati per lo streaming. Il server mantiene comunque la sessione senza liberare alcuna risorsa in modo che possa successivamente riprendere il trasferimento dei dati. Il server non manda più pacchetti RTP ma continua a mandare i pacchetti RTCP.

-TEARDOWN - Questa richiesta causa il termine della sessione RTSP. Il server ferma tutte le transazioni RTP e RTCP, chiude la comunicazione con il player e libera tutte le risorse allocate per la sessione di streaming.

L'interazione tra server e player per l'esecuzione di una sessione di streaming è tipicamente descritta dai seguenti passi:

-Viene stabilita una connessione Real-Time Streaming Protocol (RTSP) tra server e player.

-Il player esegue una richiesta RTSP DESCRIBE e il server risponde con il Session Description Protocol (SDP) per la

descrizione del media e delle stream disponibili nella presentazione.

Il player esegue una richiesta RTSP SETUP per ogni stream di cui vuole eseguire lo streaming. Il server inizia una sessione RTP per ogni stream e inizializza una coppia di porte socket per RTP e RTCP per ogni stream di cui si deve effettuare lo streaming.

Il player controlla la presentazione con l'uso dei comandi PLAY, PAUSE e TEARDOWN.

◆ Session Description Protocol (SDP)

Il server utilizza il Session Description Protocol (SDP) per fornire informazioni al player riguardo al media richiesto. Quando il player manda una richiesta di DESCRIBE, il server costruisce e trasmette le informazioni secondo l'SDP come risposta alla DESCRIBE. Il player utilizza queste informazioni per richiedere lo streaming per uno specifico stream.

Tipicamente, SDP fornisce le seguenti informazioni al player:

- nome della sessione;
- descrizione della presentazione;
- descrizione del media: bit rate, durata del media;
- informazioni sul network (come indirizzo IP, porte socket) per ricevere il media;
- informazioni sulla banda necessaria per la sessione di streaming;
- informazioni sugli stream disponibili per lo streaming (bit rate, codec).

◆ Cosa sono le "hint track"?

Le "hint track" sono un tipo specifico di traccia (stream) all'interno del file MP4. Una "hint track" contiene informazioni che indicano al server esattamente come eseguire lo streaming. Per esempio, una "hint track" indica al server esattamente dove prendere le informazioni nel file per costruire i pacchetti RTP per quello specifico stream. Le "hint track" rendono il server indipendente dal codec usato in fase di encoding in quanto non serve sapere le caratteristiche dello stream.

☞ Conclusioni

In questo articolo è stata descritta una piattaforma di streaming che implementa lo standard 3GPP per il mondo mobile (GPRS e UMTS) e lo standard ISMA per il mondo Internet. Si tratta di un sistema sicuramente giovane che però è strutturato e realizzato in maniera tale da poter diventare un'affidabile, efficiente e completa piattaforma di streaming.

Altre informazioni possono essere reperite al sito <http://members.xoom.it/catrambone/Streaming/CatraStreamingPlatform.htm> oppure prendendo contatto direttamente con il responsabile di progetto tramite l'indirizzo di posta elettronica giuliano.catrambone@tin.it.

Giuliano Catrambone

Software engineer