
Protocolli a supporto delle applicazioni multimediali distribuite in Internet

Corso di **Applicazioni Telematiche**

A.A. 2006-07 – Lezione n.3

Prof. Roberto Canonico

Università degli Studi di Napoli Federico II

Facoltà di Ingegneria

Argomenti

- Trasferimento di informazioni multimediali in Internet
 - I protocolli RTP ed RTCP
 - Il protocollo RTSP
 - Formati di descrizione (SDP,...)
 - Esempi di applicazioni
-

Trasferimento di informazioni multimediali su rete

- **Problema:** trasferire informazioni multimediali (audio, video, ...) da una sorgente ad uno o più ricevitori attraverso una rete
- Per ridurre la quantità di informazioni trasferita sulla rete, il **trasmettitore** effettua una compressione mediante un'opportuna tecnica (MPEG 1-2-4, MJPEG, MP3, ...)
- Sulla **rete** l'informazione è trasferita a **pacchetti**
- Il **ricevitore** recupera l'informazione originaria dalla sequenza di pacchetti ricevuti, mediante un'operazione inversa a quella di compressione e una successiva trasformazione in forma sonora o in forma di video (sequenza di fotogrammi)

Trasferimento di informazioni multimediali su rete (2)

- Nel caso di **informazioni live**, l'informazione è prodotta dalla sorgente mediante un apposito sistema di acquisizione (microfono + scheda audio, telecamera + video capture board), opportunamente compressa (in tempo reale) e trasmessa sulla rete ai ricevitori
- Nel caso di **informazioni pre-registrate**, l'informazione è già registrata in formato compresso (MPEG, MJPEG, MP3, ...) in un file memorizzato su memoria di massa (hard-disk, CDROM, DVD, ...)

Informazioni multimediali pre-registrate

- Sono possibili due strategie per il trasferimento dell'informazione:
 - **Trasferimento dell'intero file da sorgente a ricevitore e successiva riproduzione: *file transfer***
 - La riproduzione può iniziare solo al termine del trasferimento del file (ritardo proporzionale alla dimensione del file)
 - E' necessaria una adeguata capacità di memorizzazione (su memoria di massa) da parte del ricevitore
 - Questa soluzione è idonea solo per documenti di piccole dimensioni (audio-clip e/o video-clip)
 - **Riproduzione progressiva del contenuto multimediale durante il trasferimento dell'informazione: *streaming***
 - Il ricevitore memorizza l'informazione ricevuta in un buffer (*playout buffer*) che viene continuamente alimentato dai dati ricevuti dalla rete e svuotato progressivamente
 - La riproduzione può iniziare non appena il buffer si è "sufficientemente" riempito
 - Il ricevitore non deve memorizzare l'intero file
 - La qualità della riproduzione può degradare se la rete non mantiene la continuità temporale del flusso di informazioni trasmesso dalla sorgente (*sensibilità al jitter*)

Informazioni multimediali *live*

- Nel caso di informazioni *live*, la sorgente produce un **flusso** continuo di informazioni
- Questo flusso di informazioni è spezzato in **pacchetti** che sono trasmessi individualmente sulla rete: trasmissione in **streaming**

Sensibilità dello streaming alla QoS

- Il ricevitore riceve i pacchetti, recupera l'informazione originaria e la riconverte in forma audio/video
 - Il **ricevitore** riesce a recuperare la **continuità del flusso di informazioni** prodotto dalla sorgente se tutti i pacchetti arrivano a destinazione, con la stessa tempificazione relativa
 - La **rete** può alterare la continuità temporale del flusso di informazioni in due modi:
 - Facendo occasionalmente perdere dei pacchetti
 - Consegnando i pacchetti al ricevitore con una tempificazione relativa diversa da quella con cui sono stati trasmessi (*jitter*)
 - Perché la rete possa effettivamente supportare la trasmissione di flussi multimediali occorre che alcuni parametri di Qualità del Servizio (QoS) siano soddisfatti
 - Percentuale di perdita di pacchetti, latenza, jitter, ...
-

Degradazione introdotta dalla rete

- Gli effetti sono diversi a seconda della natura del media (audio/video), a seconda della tecnica di compressione utilizzata ed a seconda del grado di alterazione introdotto
 - nel caso di flusso audio, vengono percepite dei "disturbi" (*hiccup*s)
 - nel caso di flusso video, si hanno dei disturbi (*glitches*) che possono essere più o meno localizzati nel tempo e nello spazio
 - Sia audio che video possono in genere tollerare una parziale degradazione, ma quando si oltrepassano dei valori di soglia l'informazione diventa inintelligibile
-

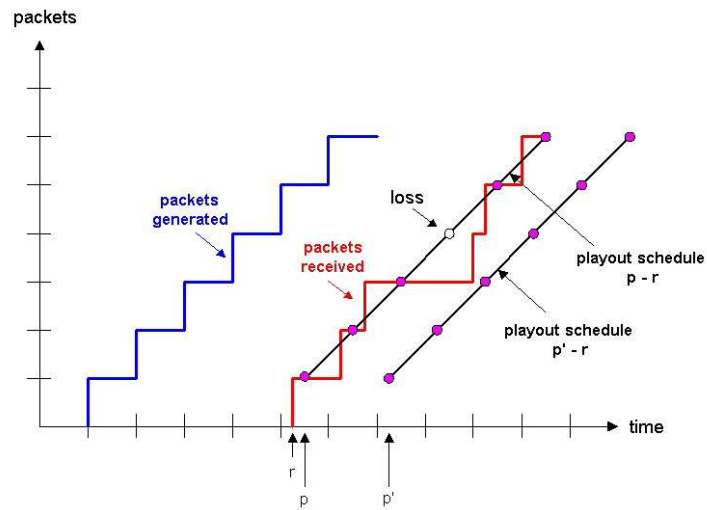
Esempio di distorsione video prodotta da errori di trasmissione



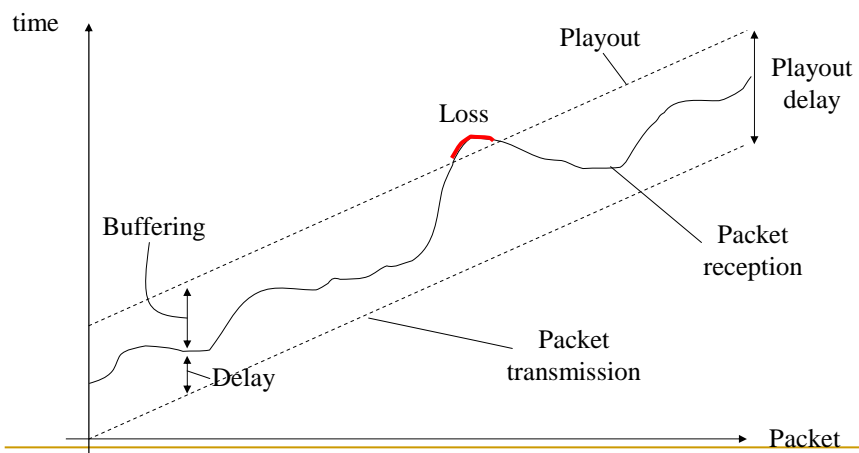
Contromisure

- Rispetto alla perdita occasionale di pacchetti, ci si difende mediante l'adozione di tecniche di compressione **robuste**, per le quali l'informazione audio/video ricostruita non è sensibilmente degradata quando occasionalmente si perde un pacchetto
 - In alcuni casi si adottano tecniche di *Forward Error Correction* (FEC)
 - L'adozione di tecniche basate sulla ritrasmissione (alla TCP) non sono considerate idonee per lo streaming
- Per limitare gli effetti del *jitter* si adotta una strategia di bufferizzazione: un buffer in ricezione fa da volano e compensa (introducendo un ritardo extra) la variabilità del ritardo di attraversamento della rete
- **NOTA:** non sarebbe necessario introdurre delle contromisure se la rete fosse in grado di offrire servizi a **qualità garantita**
 - **Internet offre un servizio best-effort !**

Bufferizzazione con ritardo di riproduzione costante



Bufferizzazione con ritardo di riproduzione costante (2)



Trasferimento di informazioni multimediali su Internet

- Il trasferimento di informazioni multimediali su Internet mediante la tecnica del **file transfer** è tipicamente realizzato adottando il protocollo applicativo HTTP, il quale si appoggia sul protocollo di trasporto TCP
- Per la trasmissione in **streaming** sono adottate due tecniche:
 - mediante un protocollo ad-hoc (RTP) su UDP
 - mediante HTTP su TCP

RTP

- RTP sta per “real-time transport protocol”
- definito dal Working Group “Audio/Video Transport” dell’ IETF inizialmente in RFC 1889 (gennaio 1996)
 - <http://www.ietf.org/rfc/rfc1889.txt>
- e successivamente in RFC 3550 (luglio 2003)
 - <http://www.ietf.org/rfc/rfc3550.txt>
- RTP offre un servizio di livello trasporto specificamente progettato per i requisiti di flussi multimediali
- I pacchetti RTP sono incapsulati in datagrammi UDP
 - Un protocollo di livello trasporto su un altro di livello trasporto

RTP (2)

- RTP è un protocollo concepito secondo il modello *Application Level Framing* proposto in
 - Clark D., and D. Tennenhouse, “Architectural Considerations for a New Generation of Protocols”, IEEE Computer Communications Review, Vol. 20(4), September 1990.
- è concepito per essere implementato direttamente nelle applicazioni, e non come uno strato aggiuntivo dello stack protocollare
- offre le funzionalità minimali richieste dalla trasmissione di flussi continui tipici delle applicazioni multimediali
- è neutrale rispetto alla codifica utilizzata (MPEG, ecc...)

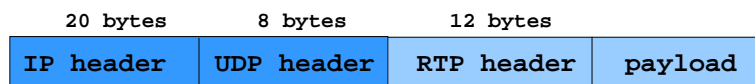
RTP (3)

- RTP fornisce informazioni di tempificazione (timestamp) per consentire
 - **sincronizzazione intra-media**: ricostruzione della corretta tempificazione della sequenza di pacchetti ricevuti
 - **sincronizzazione inter-media**: finalizzata a mantenere “al passo” flussi multimediali trasmessi separatamente (es. audio e video: sincronizzazione “lip-sync”)

RTP (4)

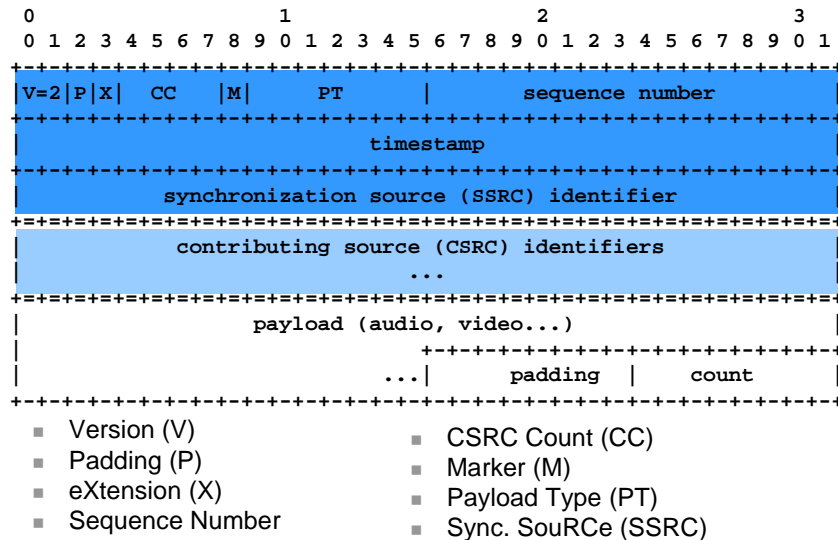
- supporta sia la trasmissione unicast che la trasmissione multicast
- i suoi meccanismi sono scalabili rispetto al numero di appartenenti al gruppo multicast
- separa la trasmissione dei dati dalla trasmissione delle informazioni di controllo
- RTP è definito congiuntamente ad un protocollo di controllo (RTCP) utilizzato per scambiare informazioni di servizio e di controllo sulla qualità della trasmissione
- fornisce informazioni necessarie a combinare flussi di informazioni differenti mediante appositi mixer software

Incapsulamento di pacchetti RTP



- Un pacchetto RTP è trasmesso in un datagramma UDP
- L'header UDP contiene i numeri di porto sorgente e destinazione
- RTP usa numeri di porto destinazione pari per la trasmissione dei flussi dati
- Se $2n$ è il numero di porto destinazione usato per uno flusso, il numero successivo $2n+1$ è usato da RTCP per trasmettere le informazioni di controllo relative a quel flusso

Header RTP



Campi dell'header RTP

- **Payload Type:** 7 bit, specifica la codifica utilizzata per i dati (PCM, MPEG2 video, ecc.)
- **Sequence Number:** 16 bit, serve ad identificare perdite di pacchetti
- **Timestamp:** 32 bit, specifica il tempo di campionamento del primo byte del payload; serve a rimuovere il jitter introdotto dalla rete mediante bufferizzazione
- **Synchronization Source identifier (SSRC):** 32 bit, identifica la sorgente del flusso, ed è scelto casualmente dalla sorgente stessa; è introdotto per non dover fare affidamento sull'indirizzo IP per identificare la sorgente;
 - problema: sono possibili conflitti ...

Campi dell'header RTP (2)

- **Contributing Source identifier list (CSRC):** sequenza di n campi da 32 bit ($0 \leq n \leq 15$), ciascuno dei quali identifica la sorgente originaria in un flusso prodotto dalla “fusione” di flussi diversi mediante un mixer software
 - esempio: audio-conferenza a più partecipanti
 - SSRC identifica il mixer
 - CSRC indica lo speaker corrente

Sessione RTP

- Una associazione tra un gruppo di entità che comunicano mediante RTP
- Alcune applicazioni danno vita a sessioni RTP differenti per media differenti (es. audio e video), a meno che la tecnica di codifica adottata non effettui un multiplexing di flussi differenti in un singolo flusso di dati
- Sessioni RTP differenti (es. audio e video) vengono distinte da un ricevitore mediante il port number di livello trasporto (UDP)

RTP timestamp e numero di sequenza

- Il valore di timestamp inserito in ogni pacchetto riferisce la tempificazione dei dati inseriti nel payload rispetto ad un clock specifico per il media trasportato
- Possono essere generati pacchetti RTP consecutivi con lo stesso timestamp
- Il numero di sequenza identifica un pacchetto rispetto agli altri principalmente per consentire di identificare pacchetti persi
- non possono essere generati due pacchetti con lo stesso numero di sequenza

RTCP

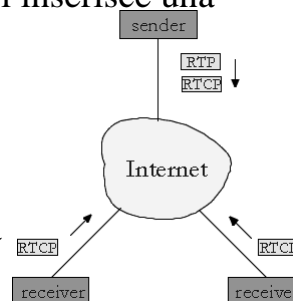
- Protocollo utilizzato congiuntamente ad RTP per la trasmissione di informazioni di controllo
- I pacchetti RTCP vengono inviati con una certa periodicità e trasportano informazioni di varia natura:
 - feedback sulla qualità della ricezione dei dati (perc. pacchetti persi, ...)
 - identificazione dei partecipanti ad una sessione RTP mediante un identificativo detto CNAME
- Nel caso di trasmissione RTP tra partecipanti ad un gruppo multicast, RTCP consente ad ogni partecipante di conoscere il numero di partecipanti

Messaggi RTCP

- Il protocollo RTCP definisce cinque tipi diversi di messaggi:
 - **Source Report (SR)**
 - **Receiver Report (RR)**
 - **Source Description (SD)**
 - **BYE**
 - **APP**
- I messaggi di tipo *report* contengono statistiche sul numero di pacchetti inviati, numero di pacchetti ricevuti, percentuale di pacchetti persi, jitter dei tempi di interarrivo, ecc. e servono a monitorare la qualità della trasmissione
- I messaggi di tipo *description*, invece, descrivono la sorgente del flusso (contengono tra l'altro il CNAME)
- BYE serve a notificare l'uscita da una sessione
- APP è un tipo di messaggio le cui funzioni sono definibili dall'applicazione

Banda usata da RTCP

- Nel caso di trasmissione multicast, ciascun ricevitore invia periodicamente (allo stesso gruppo multicast) i report RTCP
- Cosa succede se il numero di membri del gruppo diventa molto elevato ?
- Per contenere il traffico di controllo, si inserisce una minima forma di coordinamento:
- L'intervallo temporale tra due report è proporzionale al numero di partecipanti alla sessione
- in modo che la banda consumata da RTCP non superi il 5% della banda usata dalla sessione



RTSP: Real Time Streaming Protocol

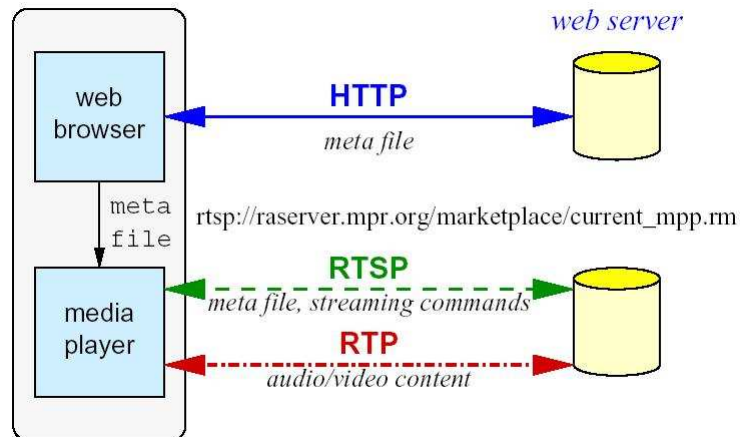
- Protocollo nato per il controllo dello streaming di flussi audio/video trasmessi da una media server verso un client
- Definito in RFC 2326 (<http://www.ietf.org/rfc/rfc2326.txt>)
- Consente al client di "comandare" lo stream trasmesso dal server mediante tipiche azioni di controllo: rewind, fast forward, pause, ecc.
- Protocollo Out-of-band i cui messaggi possono essere inviati sia tramite TCP che UDP (porta 554)
- La trasmissione dei flussi multimediali avviene poi tipicamente mediante RTP
- Un meta-file informa il client dei media coinvolti e identifica (tramite una URL) il server che li fornisce
 - Formati di meta-file: SDP (IETF RFC 2327), SMIL (W3C)

Esempio di meta-file (SMIL)

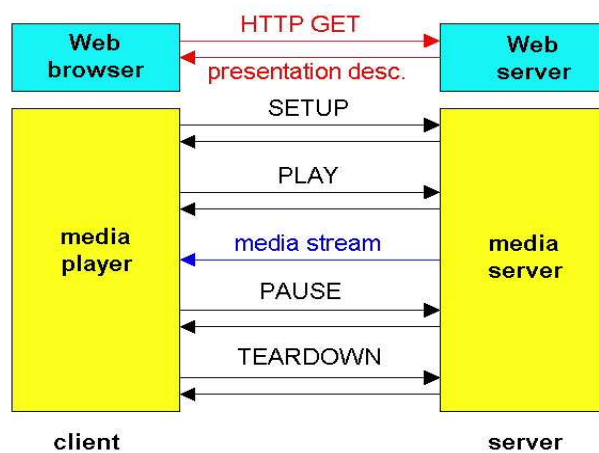
```
<title>Twister</title>
<session>
  <group language=en lipsync>
    <switch>
      <track type=audio
        e="PCMU/8000/1"
        src = "rtsp://audio.example.com/twister/audio.en/lofi">
      <track type=audio
        e="DVI4/16000/2" pt="90 DVI4/8000/1"
        src="rtsp://audio.example.com/twister/audio.en/hifi">
    </switch>
    <track type="video/jpeg"
      src="rtsp://video.example.com/twister/video">
  </group>
</session>
```

Scenario tipico

- HTTP per reperire il meta-file + RTSP&RTP per lo streaming



Scenario tipico: messaggi



RTSP: caratteristiche del protocollo

- RTSP è stato progettato sulla falsariga di HTTP
- Protocollo testuale: i messaggi sono codificati in ASCII organizzati per linee
- Adotta “\r\n” (CR+LF) come sequenza di caratteri indicatori di fine linea
- Schema di interazione richiesta-risposta (come HTTP)
- A differenza di HTTP, il server deve mantenere uno stato
 - non *stateless*, ma *stateful*
- Una sessione di controllo RTSP può essere realizzata sia su TCP che su UDP
 - L'implementazione più frequente è su TCP

Messaggi RTSP (“metodi”)

- OPTIONS get available methods
- SETUP establish transport
- ANNOUNCE change description of media object
- DESCRIBE get (low-level) description of media object
- PLAY start playback, reposition
- RECORD start recording
- REDIRECT redirect client to new server
- PAUSE halt delivery, but keep state
- SET PARAMETER device or encoding control
- TEARDOWN remove state

Esempio di sessione RTSP

```
C: SETUP rtsp://audio.example.com/twister/audio RTSP/1.0
  Transport: rtp/udp; compression; port=3056; mode=PLAY

S: RTSP/1.0 200 1 OK
  Session 4231

C: PLAY rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
  Session: 4231
  Range: npt=0-

C: PAUSE rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
  Session: 4231
  Range: npt=37

C: TEARDOWN rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
  Session: 4231

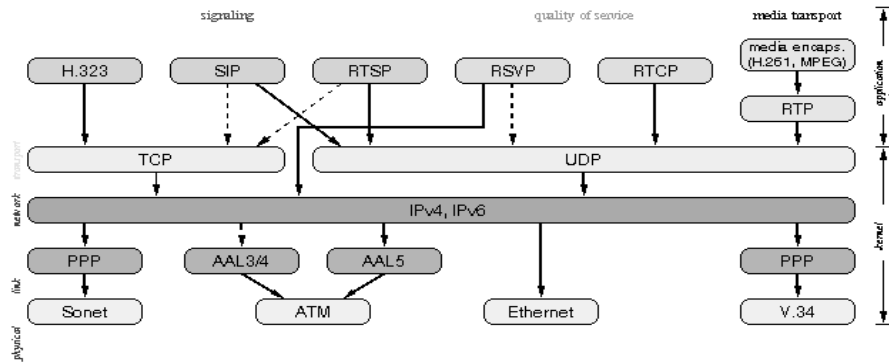
S: 200 3 OK
```

Esempio: i messaggi in dettaglio

```
C->S:
SETUP rtsp://audio.com/twister/audio.en RTSP/1.0
CSeq: 1
Transport: RTP/AVP/UDP;unicast
;client_port=3056-3057
S->C:
RTSP/1.0 200 OK
CSeq: 1
Session: 12345678
Transport: RTP/AVP/UDP;unicast
;client_port=3056-3057;
;server_port=5000-5001
```

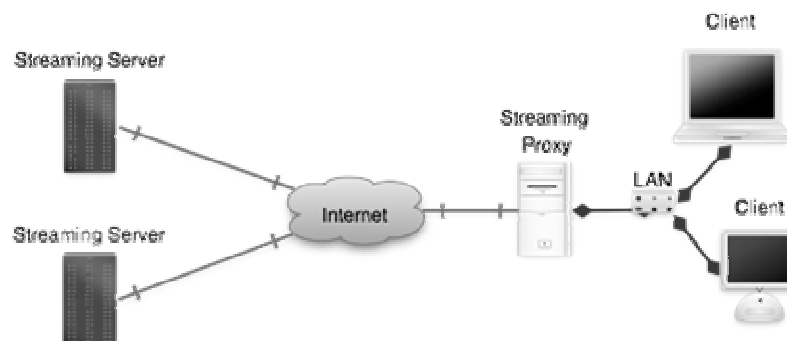
Il server restituisce un numero di sessione che dovrà essere usato dal client in tutti i messaggi successivi

Protocolli per il multimedia in Internet



- Non ci interesseremo dei protocolli di basso livello
- Altri protocolli di alto livello saranno esaminati in seguito

Streaming RTSP attraverso proxy



Riferimenti sul web

- Sito web IETF per gli RFC
 - <http://www.ietf.org>
- RTP & RTCP
 - <http://www.cs.columbia.edu/~hgs/rtp/>
- RTSP
 - <http://www.cs.columbia.edu/~hgs/rtsp/>
 - <http://www.rtsp.org>
 - Internet Streaming Alliance <http://www.isma.tv>

Domande?

