

# *Il Protocollo RTP*

audio e video per internet



**Relazione tratta da RFC 1889 e RFC 3550**

[info@andreatv.it](mailto:info@andreatv.it)  
[www.andreatv.it](http://www.andreatv.it)



## Sommario

1. Introduzione:.....	3
2. Formato del pacchetto RTP: .....	4
2.1. Definizione campi del pacchetto RTP:.....	4
3. Come funziona il pacchetto RTP: .....	7
4. RTCP:.....	8
4.1. Compiti del pacchetto RTCP: .....	8
4.2. I pacchetti RTCP:.....	9
4.3. Intervallo di trasmissione RTCP:.....	9
4.4. Sender e Receiver Report: .....	11
4.5. SDES:.....	11
4.6. BYE: .....	12
4.7. APP:.....	12
5. Differenze fra RFC 1889 e RFC 3550: .....	12
6. Note: .....	13
7. Tipologie payload: .....	14

## 1. Introduzione:

Per non limitare questo protocollo, i suoi ideatori, H. Schulzrinne, S.L. Casner, R. Frederick e V. Jacobson, hanno pensato di crearlo per:

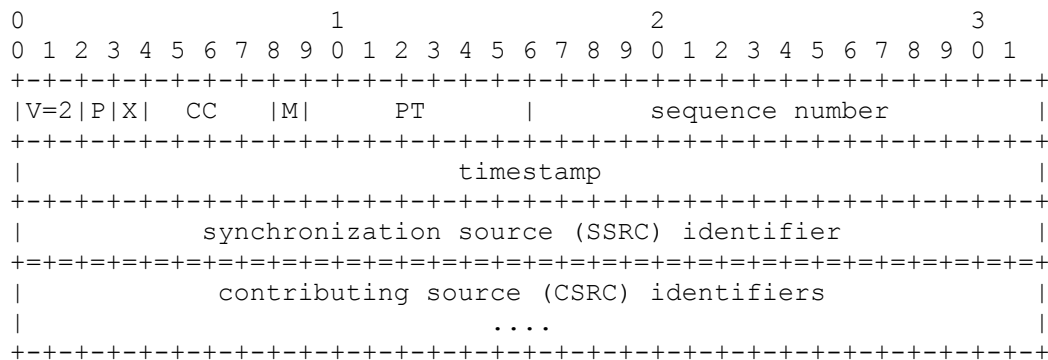
- Memorizzare un flusso continuo di dati
- Simulazioni interattive distribuite
- Controllo del contenuto
- Soddisfare il bisogno di molteplici partecipanti ad una conferenza senza implementarlo in una specifica applicazione.

Queste caratteristiche, come specificato nel RFC 1889 e poi anche in RFC 3550, sono state espressamente rese il più malleabile possibile per ogni possibile utilizzo, quindi la documentazione fornita è incompleta, specifica le funzioni, il minimo necessario per lasciare libere idee per future applicazioni. Per una completa specifica riguardante l'intero protocollo RTP, sono richiesti molti RFC, per esempio per le specifiche riguardanti le applicazioni audio e video, si può leggere l'RFC 3551, oppure per i diversi tipi di payload, si consultano gli RFC solitamente intitolati " RTP payload Format for XYZ audio/video encoding".

Il protocollo RTP, non ha nessun meccanismo per assicurare la consegna e tanto meno garantisce Quality of Service (QoS) delegando però i livelli più bassi del modello a pila ISO/OSI per svolgere questo compito. Non garantisce la consegna e tanto meno l'ordine di arrivo di tali pacchetti. La sequenza di numeri inclusi nel pacchetto RTP permette a chi riceve di ricostruire l'esatta sequenza e può essere usata per determinare l'esatta locazione di un pacchetto all'interno di una codifica video senza il bisogno di decodificare i pacchetti nella giusta sequenza.

Anche questa relazione come il Request for Comments presi in esame è separata in 2 parti strettamente collegate che si completano, la parte riguardante il protocollo RTP e quella riguardante il protocollo di controllo RTCP, necessario a monitorare la qualità del servizio e la trasmissione d'informazioni sui partecipanti durante la sessione.

## 2. Formato del pacchetto RTP:



I primi dodici byte del pacchetto RTP, sono sempre presenti, mentre la lista degli identificatori CSRC lo è solamente quando è inserito un mixer, definito dal RFC come un sistema intermedio, che ha il compito di ricevere lo stream di dati da una o più sorgenti, per esempio sorgente audio e video e unirle in un unico stream, verificando la sincronizzazione fra i flussi. Tutti i pacchetti elaborati dal mixer devono essere marcati con uno specifico identificatore SSRC, che è inserito nel campo CSRC per preservare l'identità della sorgente.

Il mixer è molto utile perché limita la banda “aiutando” le connessioni lente, senza diminuire la qualità del servizio di tutti i nodi client.

Un ulteriore problema potrebbe riguardare l'impossibilità di ricevere pacchetti multicast, per esempio per colpa di un firewall. Per risolvere anche questo tipo di problema gli ideatori hanno pensato di utilizzare un translator, anzi due, prima e dopo il firewall. Quello esterno incanala tutti i pacchetti multicast in una connessione sicura verso il translator sul lato opposto del firewall, che li converte nella forma originale e li inoltra alla rete interna.

### 2.1. Definizione campi del pacchetto RTP:

- ✘ **Versione (V):** 2 bit, identifica la versione RTP, attualmente la 2.
- ✘ **Padding (P):** 1 bit, se impostato a 1, il pacchetto contiene uno o più padding bytes che non fanno parte del payload. L'ultimo ottetto di questo campo contiene il numero di byte da ignorare. Il padding è utile per utilizzare un sistema di crittografia a blocchi di lunghezza fissa, o per trasportare parecchi pacchetti RTP

verso gli strati più bassi del protocollo, in questo caso però deve subire la frammentazione, a scapito del riconoscimento del tipo di payload.

- ✘ **Estensione (X):** 1 bit, se settato, l'header fisso deve essere obbligatoriamente seguito esattamente da un header ampliato con un formato definito, che viene appeso alla lista CSRC se presente.

Il campo estensione, permette implementazioni singole per sperimentare nuovi formati di payload indipendenti, per funzioni che richiedono informazioni aggiuntive da trasportate all'interno del pacchetto RTP. È un campo intenzionalmente limitato, visto che esistono metodi migliori di questo meccanismo.

- ✘ **CSRC Count (CC):** 4 bit, contiene il numero di identificatori CSRC.
- ✘ **Marker (M):** 1 bit, l'interpretazione di questo campo è definita da un profilo, che permette di evidenziare eventi significativi come i limiti dei frame in un flusso di pacchetti.

Con il campo payload, trasporta specifiche informazioni per le applicazioni, vengono quindi aggiunti ulteriori 32 bit per contenerle, ridefinendo il profilo.

È raccomandato impostare a uno il bit più significativo, per trovare un modello statistico dei pacchetti persi che hanno il marker attivato.

- ✘ **Payload Type (PT):** 7 bit, identifica il formato del pacchetto RTP e determina quale applicazione usare e come deve interpretarlo. Il formato è specificato attraverso una tabella dei formati di payload specificati nel [RFC 3551](#) riguardanti audio e video [paragrafo 7], oppure attraverso un formato dinamico di profili che non appartengono a RTP ma che però sono necessari a rendere il servizio utilizzabile, come il protocollo SIP ([RFC 3261](#)).

Questo campo può essere codificato, fornendo un'alternativa alla codifica a livello IP oppure a livello RTP, magari definendo nuovi profili per codifica e decodifica. Questa decodifica deve specificare il padding e altri aspetti legati alla codifica. In questo modo si codifica solamente i dati lasciando l'header inalterato, utile alle applicazioni e per mantenere un livello di confidenzialità verso i livelli sottostanti del modello ISO/OSI.

Le costanti usate in questo campo, sono definite all'interno [dell'RFC 3550](#). Esistono però un paio di valori riservati, associati a payload e marker, che debbono essere evitati, i numeri decimali 200 e 201, per distinguere il pacchetto

RTP dal pacchetto di controllo *Sender Report* (SR) e *Receiver Report* (RR) durante la validazione, solitamente sostituiti da 72 e 73.

- ✘ **Sequence Number:** 16 bit, valore incrementato ad ogni pacchetto RTP inviato, può essere usato dal destinatario per un riscontro sui pacchetti persi. Il valore iniziale del campo deve essere casuale, anche se i successivi, possono non esserlo. In questo modo la decodifica in un attacco di tipo know-plaintext è più complicata, anche se non è stato accordato nessun cifrario da utilizzare.

Questo campo è usato per determinare la giusta sequenza di pacchetti da parte del destinatario, così facendo si può utilizzare immediatamente il pacchetto ricevuto, se corretto, senza aspettare l'arrivo dell'intera sequenza, stimando allo stesso tempo il numero di pacchetti perduti.

- ✘ **Timestamp:** 32 bit, è il tempo di campionamento del primo ottetto del pacchetto RTP, proveniente da un segnale di clock incrementato linearmente, permettendo così la sincronizzazione ed il calcolo del jitter ovvero del ritardo d'invio di un pacchetto da parte dei router. Il valore della frequenza del clock dipende dal formato dei dati e dal payload che è specificato statisticamente nel profilo o nel formato del payload statico o dinamico a seconda del tipo di profilo. Se il pacchetto RTP è generato periodicamente il campionamento, è determinato dal segnale di clock e non dall'orologio di sistema. Come per il numero di sequenza, anche questo valore è casuale, può capitare che pacchetti consecutivi abbiano lo stesso timestamp, come per esempio accade quando si trasmette lo stesso frame video, allo stesso modo. Il valore fra media differenti campionati nello stesso istante, può essere differente perché solitamente hanno un differente numero iniziale!
- ✘ **SSRC:** 32 bit. Responsabile della sincronizzazione. Nel documento è raccomandata una scelta casuale di tale valore, con l'intento, però, di avere le due sorgenti (audio e video) sincronizzate con diverso numero di sessione, ma con lo stesso identificatore SSRC. Possono però verificarsi delle collisioni proprio perché più sorgenti hanno lo stesso identificatore, per risolvere questo problema, si utilizza l'indirizzo di rete locale per identificarsi, ricordando che transator e mixer operano con reti che a loro volta usano differenti spazi d'indirizzamento, aumentato ancora di più la possibilità di collisione. Si è pensato quindi di abbassare tale probabilità, inviando ad una nuova sorgente un pacchetto per ogni destinatario, per tenere traccia dei partecipanti prima di iniziare la trasmissione

e controllando allo stesso tempo che l'identificatore ricevuto non sia in contrasto con qualche altro, altrimenti se ne sceglie uno nuovo. Può accadere però che la trasmissione sia già iniziata è un nuovo destinatario con lo stesso identificatore SSRC voglia partecipare, allora la sorgente deve inviare un pacchetto *RTCP BYE* per il vecchio identificatore e sceglierne un altro.

Per risolvere gran parte delle collisioni, basta inviare nello stesso pacchetto RTP gli identificatori SSRC di ogni media utilizzato, in questo modo per esempio si evitano problemi derivanti da un cambio di codifica, o di un riavvio dell'applicazione utilizzata.

- ✘ **CSRC list:** da 0 a 15 campi, ognuno composto da 32 bit. È la lista degli identificatori SSRC, risultato del flusso prodotto dal mixer. Solamente 15 sorgenti possono essere identificate, gli altri identificatori vengono esclusi. Il numero o identificatore è preso dal campo CSRC Count.

### 3. Come funziona il pacchetto RTP:

Questo protocollo trasporta *end-to-end* i dati, in tempo reale. Utile quindi per applicazioni riguardanti la videoconferenza, oppure solamente per la trasmissione in real-time di audio o video. Per soddisfare tali richieste, utilizza il protocollo UDP, che più si adatta a questo scopo, anche se non prevede nessun meccanismo di controllo, perché non mantiene lo stato della connessione, è quindi eventualmente accompagnato da un protocollo di controllo RTCP.

Il flusso di dati, viene da prima incapsulato in un pacchetto RTP, poi UDP ed accoppiato ad un paio di porte, corrispondenti al pacchetto RTP e RTCP.

Come già detto, ogni media ha un proprio identificatore unico, quindi una propria sessione, che permette la ricezione di tutti i media oppure solamente di alcuni, secondo la scelta dell'utente.

Il protocollo RTP è stato pensato per conferenze, è stato quindi realizzato per utilizzare il protocollo multicast, dove tutti i partecipanti alla sessione condividono un paio d'indirizzi di destinazione comuni, fin dall'inizio, però le applicazioni lo hanno usato anche unicast, dove i partecipanti possono ricevere da tutti gli utenti della sessione, il numero delle porte da utilizzare, che possono essere uguali o differenti per ognuno.

## 4. RTCP:

Lo scopo principale di questo protocollo è la ricerca di feedback nella qualità del servizio RTP che sono poi utili per il controllo della codifica del pacchetto, ma anche per verificare eventuali errori nell'invio dei dati. Basato sulla periodica trasmissione dei pacchetti di controllo verso tutti i partecipanti della sessione, riutilizzando lo stesso meccanismo per l'invio dei pacchetti RTP, questo protocollo utilizza una porta differente rispetto al RTP per dividere i pacchetti di controllo da quelli dati e poiché si basa sempre sul multicast, è utile anche per ricevere feedback della diagnosi della rete, grazie ai pacchetti RTCP sender e RTPC receiver.

### 4.1. Compiti del pacchetto RTCP:

Invia verso il livello di trasporto l'identificativo per la sorgente RTP chiamato CNAME e collega l'identificatore SSRC all'identificatore della sorgente mittente o destinatario che sia. Quindi come il campo SSRC, il CNAME è unico all'interno della sessione ed ha un formato simile a "user@host" oppure solamente "host" dove host indica il dominio completo della sorgente dei dati. Altri tipi di rappresentazione sono possibili utilizzando la codifica ASCII e indirizzi unici come l'indirizzo MAC.

Se l'identificatore SSRC cambia per i problemi descritti sopra, CNAME, ha il compito di tenere traccia, per ogni partecipante.

Esiste un'altra funzione opzionale per questo pacchetto, ovvero un'illustrazione minima delle informazioni di controllo della sessione, come per esempio tradurre l'identificatore con il nome nell'interfaccia utente.

## 4.2. I pacchetti RTCP:

Ogni pacchetto RTCP ha la parte iniziale molto simile al pacchetto RTP, il seguito, però è strutturato in modo variabile per quanto riguarda la lunghezza del pacchetto, in accordo con il tipo di pacchetto, che non può superare i 32 bit.

Non esiste un numero minimo di pacchetti RTCP, poiché ognuno di essi è indipendente dall'ordine di arrivo o di partenza.

I vari pacchetti RTCP per controllo e verifica informazioni sono:

<b>Sender Report (SR)</b>	Fornisce statistiche di trasmissione e ricezione dei partecipanti.
<b>Receiver Report (RR)</b>	Responsabile della ricezione delle statistiche dei partecipanti che non hanno un compito attivo per quanto riguarda l'invio di pacchetti.
<b>SDES:</b>	Contiene il CNAME che deve essere incluso in ogni pacchetto RTCP.
<b>BYE:</b>	Indica la fine della trasmissione a un partecipante.
<b>APP:</b>	Funzioni specifiche riguardanti l'applicazione.

## 4.3. Intervallo di trasmissione RTCP:

I pacchetti RTCP sono stati disegnati per permettere la scalabilità automatica durante la sessione.

Ogni sessione è soggetta ad una larghezza di banda definita, divisa fra i partecipanti.

Tale banda può essere riservata e limitata secondo la connessione utilizzata, se non ci sono prenotazioni o altri impedimenti, la banda utilizzabile dipende dalla struttura.

Viene stabilita così una banda massima per ogni sessione. Tale valore può essere definito

a priori, conoscendo le caratteristiche della connessione, e dipende in larga misura dal tipo di codifica del media utilizzato da parte del mittente.

Per ogni sessione è definita una larghezza di banda utilizzabile, tale valore è calcolato da un'applicazione di controllo della sessione che, quando è richiamata, può impostare un valore di default, spesso però la larghezza di banda della sessione è la somma delle larghezze di banda nominali dei mittenti attivi, a patto che tutti i partecipanti utilizzino la stessa larghezza di banda, così facendo si ottiene il pacchetto RTCP nello stesso intervallo.

In generale, la banda è riservata ai pacchetti RTCP è circa il 5%, è necessario però se i mittenti sono maggiori rispetto ai destinatari, riservare un quarto della banda RTCP a chi trasmette dati, in questo modo in una sessione con tanti riceventi e pochi mittenti, i partecipanti ricevono in tempi minori il CNAME. Quando la proporzione dei mittenti è più grande di un quarto, i mittenti possono utilizzare l'intera larghezza di banda, finché i valori non diventano critici. Se il rapporto mittenti partecipanti è più grande di  $S/(S+R)$ , allora il mittente fa la proporzione della somma di questi partecipanti.

L'esatto uso di questi due parametri, S e R, permette la ricezione dei report, per fissare la larghezza di banda di chi non trasmette a zero, mentre per chi trasmette ad un valore diverso da zero, per facilitare l'invio dei Sender Report e sincronizzare i media.

Esiste un ritardo voluto, che è la metà dell'intervallo più piccolo ricevuto, fra l'avvio di un'applicazione e l'invio della prima serie di pacchetti RTCP, per permettere a tutti i partecipanti di ricevere i pacchetti RTCP, arrivando al tempo d'intervallo ottimale più velocemente fra i partecipanti "vicini" e "lontani". Il valore raccomandato per l'intervallo è di 5 secondi.

È importante calcolare il tempo d'intervallo, perché i pacchetti RTCP, al contrario di quelli RTP non sono in nessun modo limitati, quindi se i pacchetti di tipo RR fossero inviati da tutti i partecipanti ad una frequenza costante, la banda utilizzata aumenterebbe linearmente con il numero di partecipanti.

#### 4.4. Sender e Receiver Report:

La differenza fra i due pacchetti oltre al tipo di codice, è che il SR include 20 byte dove sono presenti le informazioni riguardanti i mittenti attivi. Entrambi i pacchetti possono contenere da zero a 31 blocchi di resoconto, uno per ogni sorgente da cui ha ricevuto pacchetti RTP provenienti dall'ultimo report. Ogni blocco contiene statistiche sui dati ricevuti da una particolare sorgente, se sono necessari più blocchi, quelli mancanti sono aggiunti ad un pacchetto di tipo RR. Se ci sono molte sorgenti da accontentare, tutti i pacchetti RR, sono inclusi in un pacchetto RTCP senza superare l'MTU (Maximum Transmission Unit) della rete.

#### 4.5. SDES:

Pacchetto composto da un header e da zero o più chunks, ognuno composto da oggetti descritti dall'identificatrice sorgente all'interno del chunk, lungo al massimo 32 bit, contiene un identificatore SRRS/CSRS seguito da una lista di zero o più oggetti complementari, letti in successione, fino a quando un oggetto ha valore zero, ovvero fine lista.

Solamente l'oggetto CNAME è obbligatorio, tutti gli altri oggetti possono essere usati da particolari profili.

#### 4.6. BYE:

Indica che una o più sorgenti non sono più attive.

Se questo pacchetto è ricevuto da un mixer, allora può inoltrare tale pacchetto senza modificare l'identificatore SSRC/CSRC.

#### 4.7. APP:

Pacchetto sperimentale, usato nelle nuove applicazioni e per nuove funzioni. Se l'applicazione non riconosce il nome del pacchetto, semplicemente, lo ignora.

### 5. Differenze fra RFC 1889 e RFC 3550:

Le modifiche fra i due RFC sono essenzialmente dei miglioramenti che prendono significato quando il numero di partecipanti alla sessione è dell'ordine delle centinaia e quando molti utenti entrano ed escono contemporaneamente. Tali modifiche sono state progettate per coesistere con la versione precedente del RFC, diminuendo la banda RTCP durante l'accesso, in modo proporzionale al rapporto dei partecipanti che implementano i nuovi algoritmi.

I miglioramenti riguardano principalmente l'algoritmo d'intervallo, che minimizza la trasmissione in eccesso quando c'è un ingresso organizzato di partecipanti, ed un diverso modo di ridurre l'errore di timeout causato durante l'eliminazione veloce di partecipanti.

## 6. Note:

Seguendo la terminologia interpretata e descritta nel [BCP 14](#), che utilizzano entrambi gli RFC, ho deciso volutamente di non inserire le modalità specifiche con cui operano i vari pacchetti RTCP, perché a quel punto non avrei ottenuto una relazione ma una documentazione RFC tradotta. Con lo stesso metro di giudizio, ho eliminato parti troppo tecniche ed esempi specifici.

## 7. Tipologie payload:

Tabella dei payload types del protocollo RTP, aggiornata al 22/07/2008.

PT	encoding name	audio/video (A/V)	clock rate (Hz)	channels (audio)
0	PCMU	A	8000	1
1	Reserved			
2	Reserved			
3	GSM	A	8000	1
4	G723	A	8000	1
5	DVI4	A	8000	1
6	DVI4	A	16000	1
7	LPC	A	8000	1
8	PCMA	A	8000	1
9	G722	A	8000	1
10	L16	A	44100	2
11	L16	A	44100	1
12	QCELP	A	8000	1
13	CN	A	8000	1
14	MPA	A	90000	
15	G728	A	8000	1
16	DVI4	A	11025	1
17	DVI4	A	22050	1
18	G729	A	8000	1
19	reserved	A		
20	unassigned	A		
21	unassigned	A		
22	unassigned	A		
23	unassigned	A		
24	unassigned	V		
25	CelB	V	90000	
26	JPEG	V	90000	
27	unassigned	V		
28	nv	V	90000	
29	unassigned	V		
30	unassigned	V		
31	H261	V	90000	
32	MPV	V	90000	
33	MP2T	AV	90000	
34	H263	V	90000	
35--71	unassigned	?		
72--76	reserved for	RTCP conflict avoidance		
77--95	unassigned	?		
96--127	dynamic	?		