TECNICO DELLE RETI

MODULO 9

I° parte

Scuola Radio Elettra®

Unità di Apprendimento PG440_M09
N. 8.1
Titolo: APPLICAZIONI MULTIMEDIALI NEL NETWORKING
Obiettivo: Analisi delle principali applicazioni multimediali

Giunti a questo punto, abbiamo acquisito basi consistenti sui principi e la pratica del funzionamento delle reti di calcolatori. Tali basi saranno utili in questa unità per un argomento che interseca molti strati della pila protocollare: il funzionamento delle reti *(networking)* con le applicazioni multimediali.

Negli ultimi anni abbiamo assistito ad una crescita intensa nello sviluppo e dell'operatività di applicazioni funzionanti in rete che trasmettono e ricevono contenuti audio e video su Internet. Le nuove applicazioni multimediali funzionanti in rete *(multimedia nerworking applications),* cui ci si riferisce anche come alle applicazioni a media continui, (*continuous media applications)* - video streaming, telefonia IP, radio Internet, teleconferenze, giochi interattivi, mondi virtuali, apprendimento a distanza, e molto altro - sembra siano annunciate quotidianamente. I requisiti di servizio di queste applicazioni differiscono in modo significativo da quelli delle applicazioni elastiche (e-mail, Web, login remoto, condivisione di file). In particolare, le applicazioni multimediali sono altamente sensibili al ritardo da terminale a terminale *(end-to-end delay)* e alle variazioni del ritardo, ma possono tollerare l'occasionale perdita di dati. Esamineremo dapprima come le applicazioni multimediali possono essere progettate per sfruttare al meglio il servizio best-effort offerto da Internet, che non dà garanzie sul ritardo da estremo a estremo. Successivamente esamineremo una serie di attività che sono attualmente in corso per estendere l'architettura di Internet in modo da supportare i requisiti di servizio delle applicazioni multimediali.

Esempi di applicazioni multimediali

Internet trasporta una grande varietà di applicazioni multimediali; in seguito considereremo tre ampie classi di queste applicazioni: streaming di audio/video memorizzati, streaming di audio/video in tempo reale e audio/video interattivo in tempo reale.

In questo unità non ci occuperemo di applicazioni "scarica-e-poi-esegui", come il download di un intero file MP3 mediante un'applicazione di condivisione di file da pari a pari prima di

riprodurre il pezzo MP3. Le applicazioni “scarica-e-poi-esegui” sono applicazioni elastiche di trasferimento di file senza requisiti di ritardo speciali.

Streaming, audio e video in memoria

In questa classe di applicazioni, i client richiedono on-demand file audio o video compressi che sono memorizzati su un server. Un client può richiedere un file audio o video in qualsiasi momento e nella maggior parte dei casi dopo un ritardo di pochi secondi può cominciare a riprodurre il file mentre lo stesso continua ad essere scaricato dal server (tale caratteristica è chiamata streaming). Il ritardo tra il momento in cui l’utente fa la richiesta e il momento in cui l’utente comincia effettivamente a ricevere il file può variare tra uno e dieci secondi per una reattività accettabile. Spesso tali applicazioni sono interattive: l’utente può ad esempio bloccare o riprendere la riproduzione, saltare avanti o indietro nel file. I requisiti per il ritardo di pacchetto e il jitter non sono così stringenti come nel caso delle applicazioni real-time.

- *Media in memoria.* Il contenuto multimediale è stato preregistrato e archiviato sul server. Come risultato un utente può usare pausa, riavvolgimento, avanzamento rapido o indicizzare il contenuto multimediale. Per una sensibilità accettabile, il tempo dal momento in cui il client fa una richiesta di questo tipo al momento in cui l’azione si manifesta (al client) dovrebbe essere circa di 1- 10 s.

- *Streaming.* In molte applicazioni audio e video memorizzate, un client comincia a riprodurre l’audio/video pochi secondi dopo che è cominciata la ricezione del file dal server. Questo significa che il client può riprodurre file audio/video in un certo punto del file mentre sta ancora ricevendo l’ultima parte dal server. Questa tecnica, conosciuta come streaming, evita di dover scaricare l’intero file (e di incorrere in lunghi ritardi) prima di cominciarne la riproduzione. Ci sono molti prodotti per lo streaming multimediale, compreso il RealPlayer della RealNetworks, il QuickTime della Apple e il Windows Media della Microsoft

- *Riproduzione continua.* Dal momento in cui inizia la riproduzione del file multimediale, essa procede in accordo ai tempi di registrazione originali. Questo impone limiti critici di ritardo per l’arrivo dei dati. I dati devono essere ricevuti dal server in tempo per la loro riproduzione al client. Sebbene le applicazioni multimediali memorizzate presentino requisiti di riproduzione continua, i loro vincoli sul ritardo end-to-end sono

tipicamente meno rigidi di quelli per applicazioni interattive dal vivo, come la telefonia Internet e le videoconferenze

Streaming audio e video dal vivo

Questa classe di applicazioni è simile alla tradizionale diffusione radio e televisiva, a eccezione che la trasmissione avviene su Internet. Queste applicazioni permettono a un utente di ricevere dal vivo trasmissioni radio o televisive diffuse da qualsiasi parte del mondo. Poiché audio/video in tempo reale non è registrato in memoria, un client non può usare l'avanzamento rapido all'interno del media. Però, con l'archiviazione in una memoria locale dei dati ricevuti, per alcune applicazioni sono possibili altre operazioni interattive, come la pausa e il riavvolgimento nel corso delle trasmissioni multimediali dal vivo. Le applicazioni tipo le trasmissioni dal vivo hanno spesso molti client che ricevono lo stesso programma audio/video. La distribuzione a molti receiver di audio/video dal vivo può essere eseguita efficientemente usando le tecniche multicast IP Come con lo streaming dei multimedia in memoria, è richiesta la continuità della riproduzione, sebbene i limiti sulla tempistica siano meno restrittivi rispetto alle applicazioni interattive dal vivo. Possono essere tollerati ritardi fino a decine di secondi da quando l'utente richiede la spedizione/riproduzione di una trasmissione dal vivo a quando la riproduzione inizia.

Audio e video interattivi in tempo reale

Questa classe di applicazioni permette alle persone di usare audio/video per comunicare tra loro in tempo reale. All'audio interattivo in tempo reale ci si riferisce spesso come alla telefonia Internet, poiché, dalla prospettiva dell'utente, è simile al tradizionale servizio di telefonia a commutazione di circuito. La telefonia Internet potenzialmente può fornire i servizi telefonici PBX (centralini privati), locali e a lunga distanza a costi molto bassi. Può anche facilitare la realizzazione di nuovi servizi che non sono facilmente supportati dalle reti a commutazione di circuito tradizionali, tra cui l'integrazione Web-telefono, la comunicazione di gruppo in tempo reale, i servizi di consultazione di elenco, il filtraggio del chiamante, e altro. Attualmente sono disponibili centinaia di prodotti per la telefonia su Internet. Per esempio, gli utenti di Microsoft Instant Messenger possono effettuare chiamate PC-telefono e PC-PC. Con il video interattivo in tempo reale, chiamato anche videoconferenza, gli individui comunicano visivamente oltre che verbalmente. Oggi sono molti i prodotti disponibili per il video interattivo in tempo reale in Internet, compreso il Microsoft's NetMeeting. Notate che in un'applicazione interattiva audio/video un utente

può parlare o muoversi in qualunque momento. Per una conversazione con l'interazione di più partecipanti, il ritardo da quando un utente parla o si muove a quello in cui l'azione si rende manifesta all'host ricevente dovrebbe essere inferiore a poche centinaia di millisecondi. Per la voce, ritardi inferiori a 150 millisecondi non sono percepiti dall'ascoltatore, ritardi fra *150* e 400 millisecondi possono essere accettabili, ritardi che superano i 400 millisecondì possono risultare frustranti, se non completamente in-comprensibili, per la conversazione vocale.

Ostacoli per la multimedialità nell'internet odierna

Ricordiamo che il protocollo dello strato di rete per l'Internet attuale dà un servizio best-effort a tutti i datagramma che trasporta. In altre parole, Internet fa del suo meglio per muovere ciascun datagramma dal sender al receiver il più velocemente possibile, ma il servizio best-effort non fa alcuna promessa circa il ritardo end-to-end per qualsiasi singolo pacchetto. Né fa alcuna promessa circa le variazioni del ritardo di un pacchetto all'interno di un flusso di pacchetti. Poiché TCP e UDP funzionano su TP. nessuno di questi protocolli può fornire alcuna garanzia sul ritardo alle applicazioni che a essi si rivolgono. A causa della mancanza di qualsiasi impegno per inviare i pacchetti tempestivamente, il problema di sviluppare applicazioni multimediali per il funzionamento in rete costituisce una sfida estrema nel caso di Internet. A tutt' oggi la multimedialità Internet ha raggiunto un significativo ma limitato successo. Per esempio, lo streaming di audio/video in memoria con ritardi di interazione dell'utente da cinque a dieci secondi costituiscono ora la norma in Internet. Ma durante i periodi di picco del traffico le prestazioni possono non soddisfare, particolarmente quando i link coinvolti sono congestionati (come i congestionati link transoceanici).

La telefonia Internet e il video interattivo in tempo reale hanno avuto a tutt'oggi meno successo dello streaming di audio/video in memoria. Infatti, il jitter nel caso delle applicazioni audio/vocali interattive in tempo reale ha limiti molto stretti. Il jitter dei pacchetti *(racket jitter)* è la variabilità dei ritardi dei pacchetti all'interno dello stesso flusso di pacchetti. Voce e video in tempo reale possono funzionare bene in regioni dove la larghezza di banda è abbondante, e quindi ritardo e jitter sono minimi. Ma la qualità può deteriorarsi fino a livelli inaccettabili appena il flusso di pacchetti vocali o video in tempo reale si imbatte in un link moderatamente congestionato.

Il progetto delle applicazioni multimediali risulterà di certo più semplice se ci sarà qualcosa come un servizio Internet di prima e uno di seconda classe, per mezzo del quale i pacchetti di prima classe siano in numero limitato e ricevano la priorità di servizio nelle code nei router. Questo tipo di servizio di prima classe dovrebbe essere soddisfacente per le applicazioni sensibili al ritardo. Ma fino a oggi Internet ha assunto in gran parte un approccio ugualitario per i pacchetti accodati ai router. Tutti i pacchetti ricevono ugual servizio; nessun pacchetto, compresi quelli audio e video sensibili al ritardo, riceve una priorità particolare nelle code nei router.

Per il momento, dobbiamo convivere con il servizio best-effort. Ma, data questa limitazione, possiamo applicare molte decisioni di progetto e usare alcuni trucchi per migliorare la qualità delle applicazioni multimediali percepita dall'utente. Per esempio, possiamo spedire audio e video su UDP, e in questo modo evitare il basso throughput del TCP quando entra nella sua fase di partenza lenta. Possiamo ritardare la riproduzione al receiver di 100 ms o più, in modo da diminuire gli effetti del jitter indotto dalla rete. Al sender, possiamo aggiungere una marcatura temporale *(timestamp)* ai pacchetti in modo che il receiver sappia quando devono essere riprodotti. Per audio/video in memoria possiamo pre-scaricare i dati durante la riproduzione quando al client sono disponibili memoria e larghezza dì banda extra. Possiamo anche inviare informazioni ridondanti per mitigare gli effetti della perdita di pacchetti indotta dalla rete. Analizzeremo molte di queste tecniche più avanti.

Compressione audio video

La compressione dei dati è il processo di conversione dei dati puri in un flusso d'uscita di dimensione inferiore. La compressione si basa su una tecnica di riduzione di ridondanza delle informazioni, con perdita o senza perdita. Un processo di compressione senza perdita potrebbe essere indispensabile nella trasmissione di dati di importanza vitale o di codici segreti di banca. La compressione con perdita, dall'altra parte, aumenta il fattore di compressione, a scapito però della precisione. Nonostante ciò si rivela veramente efficace se applicata a immagini e suoni; l'occhio e l'orecchio umano sono infatti in grado di tollerare una certa imperfezione nella riproduzione. Comunque, anche se con la perdita di alcune informazioni originarie, la compressione dei dati audio/video ha portato ad un miglioramento notevole nell'utilizzo di Internet. La compressione dei dati consiste nel prendere un flusso di simboli e trasformarli in una sequenza di codici.

Il principio fondamentale della compressione si basa sull'assegnare codici brevi ai termini che ricorrono più frequentemente, lasciando i codici lunghi a quei termini che invece capitano di rado. Se la compressione deve essere efficace, la sequenza di codici risulterà molto più breve di quella dei simboli originali. La decisione di scegliere proprio un certo codice in corrispondenza ad un particolare simbolo si basa su di un modello matematico. Tale modello è costituito da un insieme di dati e di regole che stabiliscono il criterio secondo il quale ai simboli elaborati in ingresso è associato un determinato codice. Un programma usa questo modello per definire con precisione la probabilità per ogni simbolo e sulla base di queste probabilità il codificatore genera un codice appropriato. La somma di modellizzazione e codifica produce la compressione dei dati. Generalmente, però, si è soliti indicare l'intero processo col termine di codifica *(coding).*

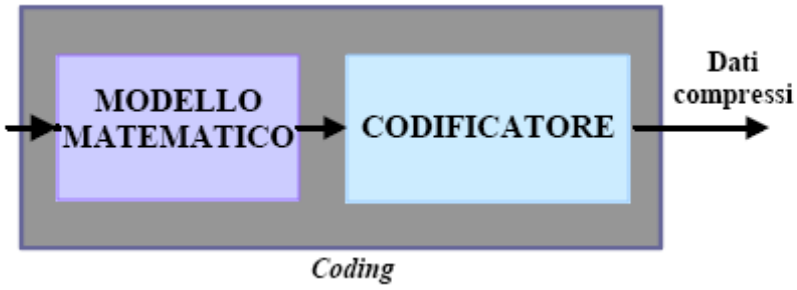


Figura 8.1 – Schema a blocchi di un modello di compressione

Alcuni metodi di compressione agiscono su un flusso di dati; elaborano cioè la sequenza di ingresso con continuità finché non incontrano la fine del file. Altri metodi raggruppano invece i dati in blocchi; ciascun blocco è letto e codificato separatamente dagli altri. Vi sono metodi di compressione, chiamati *fisici,* che leggono i dati, ma ignorano il loro significato fisico. L'efficacia di un particolare schema di compressione si misura attraverso grandezze caratteristiche:

- **Frequenza di compressione = $\frac{\text{dimensione del flusso di uscita}}{\text{dimensione del flusso di ingresso}}$**
- **Fattore di compressione = 1 / Frequenza di compressione**
- **Efficacia della compressione = 100 x (1 - Frequenza di compressione)**
- **Bpp = numero bit necessari per comprimere un pixel**

Applicazioni	Fattore approssimativo di compressione	Requisiti di banda (Mbps) (dopo la compressione)
Videoconferenza	300:1	0.384
Cataloghi video	100:1	1.5
TV digitale (MPEG-2) • **su richiesta** • **dal vivo** • **sport**	100:1 50:1 25:1	1.5 3 6
Varietà (HDTV 1080 X 1920 X 24)	100:1	da 19.3 a 38.6

Figura 8.2 – Requisiti di banda in relazione a determinati fattori di compressione

Voce e musica producono un segnale audio analogico continuamente variabile che viene normalmente convertito in un segnale digitale:

- Il segnale analogico è prima campionato ad una certa frequenza fissa, ad esempio a 8,000 campioni al secondo. Il valore di ogni campione è un arbitrario numero reale.
- Ogni campione è poi arrotondato ad uno di un numero finito di valori. Si tratta dell'operazione di *quantizzazione*; per valori di quantizzazione si scelgono solitamente potenze di due.
- Ogni numero di quantizzazione è rappresentato da un numero fisso di bit. Ogni campione viene convertito nella sua rappresentazione in bit e le rappresentazioni di ciascun campione sono poi concatenate insieme a formare la rappresentazione digitale del segnale.

Ad esempio, se un segnale analogico è campionato a 8,000 campioni al secondo, ogni campione è quantizzato e rappresentato per mezzo di 8 bit, quindi il segnale digitale risultante avrà una frequenza di 64,000 bit al secondo. Dopo la trasmissione questo segnale viene riconvertito, ossia decodificato, in un segnale analogico pronto per essere

riascoltato. Il segnale analogico decodificato risulta in ogni caso leggermente differente da quello originale. L'approssimazione risulta tanto migliore quanto maggiori sono la frequenza di campionamento e il numero di valori di quantizzazione. E' chiaro dunque che esiste un compromesso tra la qualità del segnale decodificato e i requisiti di memoria e di banda del segnale digitale.
La tecnica base di codifica è chiamata *Pulse Code Modulation (PCM).* Il PCM è diffuso per la codifica del parlato, con una frequenza di campionamento di 8000 campioni al secondo e 8 bit per campione, generando una frequenza di 64 Kbps. I compact disk audio (CD) usano anche il PCM, solo con una frequenza di campionamento di 44,100 campioni al secondo con 16 bit per campione; la velocità di trasmissione risultante è dunque di 705.6 Kbps per il mono e 1.411 Mbps per lo stereo. La velocità di 1.411 Mbps della musica stereo è superiore a moltissime frequenze di accesso e perfino i 64 Kbps del parlato eccedono la frequenza di accesso di un modem di utente. Per questa ragione la codifica PCM è usata raramente in Internet per la musica e il parlato. Si usano invece delle tecniche di compressione in grado di ridurre la frequenza di bit del flusso di dati. Le più popolari tecniche di compressione comprendono GSM (13 Kbps), G.729 (8.5 Kbps), G.723 (6.4 e 5.3 Kbps), G.711, DVI e ancora un gran numero di tecniche proprietarie. La musica stereo dei CD usa lo standard di compressione MPEG livello 3, meglio conosciuto come MP3. MP3 comprime il suono a 28 o 112 Kbps e produce solo una leggerissima degradazione della qualità musicale.

Un video è una sequenza di immagini, dove ciascuna immagine è aggiornata ad una frequenza costante, ad esempio di 24 o 30 immagini al secondo. Un'immagine digitale non compressa consiste in una sequenza di pixel, con ciascun pixel codificato da un certo numero di bit necessario per rappresentare la luminanza e la crominanza. Ci sono due tipi di ridondanza nel video ed entrambe vengono sfruttate nelle tecniche di compressione. La ridondanza spaziale è la ridondanza insita in una data immagine; se è costituita per lo più da spazi bianchi, l'immagine può essere ad esempio efficacemente compressa. Due immagini identiche o comunque molto simili in successione generano invece una ridondanza temporale. Non è ragionevole ricodificare entrambe le immagini, se sono identiche; é molto più efficiente indicare semplicemente durante la codifica che si tratta della stessa immagine. Gli standard di compressione MPEG sono tra le più popolari tecniche di compressione video; MPEG1 per la qualità video CD-ROM (1.5 Mbps), MPEG2 per il video DVD ad alta qualità (3-6 Mbps) e MPEG4 per la compressione video

orientata all’oggetto. Molto diffusi in Internet sono anche gli standard JPEG, H.261 e numerosi altri standard proprietari.

Test

1	Le applicazioni multimediali non sono sensibili al ritardo da terminale a terminale e alle variazioni del ritardo.	SI
		NO
2	La compressione dei dati è il processo di conversione dei dati puri in un flusso d'uscita di dimensione inferiore.	SI
		NO
3	L'efficacia di un particolare schema di compressione si misura attraverso grandezze caratteristiche. frequenza di compressione e fattore di compressione.	SI
		NO

TECNICO DELLE RETI

MODULO 9

2° parte

Scuola Radio Elettra®

Unità di Apprendimento
N. 8.2
Titolo: Lo streaming di audio video in memoria
Obiettivo: Analisi del funzionamento dello streaming

Lo streaming indica la possibilità di presentare via Internet un filmato, o un pezzo audio, direttamente, ossia senza dover prima aspettare che l'intero file sia stato scaricato. E' necessaria una codifica digitale per convertire il segnale analogico in un formato digitale compresso per la trasmissione e la riproduzione. Fare lo streaming di un file significa dunque mandare in uscita un flusso continuo di informazioni audio/video; i dati possono essere recuperati da un archivio nel caso di applicazioni su richiesta (ad esempio da parte di un telespettatore), oppure possono essere trasmessi in tempo reale. E' importante osservare che lo streaming è molto più che un semplice segnale digitale trasmesso via Internet. Offre al pubblico la possibilità di intervenire interattivamente e incredibili forme di comunicazione bidirezionale. Il processo di streaming audio/video interattivo è conosciuto come *Webcasting.* Una larga diffusione del *Webcasting* sarà comunque ancora impensabile, almeno finché il pubblico degli utenti non potrà disporre di un accesso alla rete ad una velocità minima di 1.5 Mbps . Il problema principale risiede nei dati video, che richiedono un'occupazione di banda enorme. E' per questo che tale applicazione è ancora molto lontana dal poter offrire un servizio audio/video paragonabile a quello offerto dalla tradizionale diffusione radio televisiva (*broadcast).* Mentre una parabola satellitare a diffusione diretta riceve i dati a 2 Mbps, un modem è attualmente limitato a 0.05 Mbps. Le stime più ottimistiche dicono che nel 2006 circa 40 milioni di case avranno un accesso per il modem alla velocità di 1.5 Mbps. L'utilizzo di algoritmi per la compressione dei dati riduce ad un valore molto più accettabile la frequenza necessaria per fare lo streaming di file video; la qualità audio/video, ad una data larghezza di banda, dipende sensibilmente dallo specifico formato di compressione. La tecnologia solo recentemente ha raggiunto un punto in cui il video può essere digitalizzato e compresso ad un livello tale da permetterne una riproduzione accettabile durante la trasmissione su reti digitali.

Accedere ad audio e video da un server web

L'audio/video in memoria possono risiedere sia su un server web che consegna gli audio/video al client su HTTP, sia su uno streaming server di audio/video che consegna gli audio/video al client su protocolli non-HTTP (protocolli che possono essere standard

proprietari o aperti). Esamineremo la consegna di audio/video da un server Web; successivamente vedremo la consegna da uno streaming server.

Consideriamo prima il caso di uno streaming audio. Quando un file audio risiede su un server Web, il file audio è un oggetto generico nel file system del server, proprio come i file HTML e JPEG. Quando un utente vuole ascoltare il file audio, l'host dell'utente stabilisce una connessione TCP con il server Web e invia una richiesta HTTP per l'oggetto). Dopo aver ricevuto una richiesta, il server Web inserisce il file audio in un messaggio di risposta HTTP e restituisce questo messaggio nella connessione TCP.

Il caso del video è leggermente più delicato, perché le parti audio e video del "video" devono essere inserite in due file diversi, vale a dire, esse devono costituire due oggetti diversi nel file system del server Web. In questo caso, al server sono inviate due richieste HTTP separate (su due connessioni TCP separate per HTTP/1.0), e i file audio e video arrivano al client in parallelo. La gestione della sincronizzazione dei due flussi è lasciata al client. È anche possibile che audio e video siano lasciati assieme nello stesso file, così che al client debba essere inviato un solo oggetto. Per mantenere semplice la nostra esposizione, nel caso del "video" assumeremo che audio e video siano contenuti in un file. Un'architettura primitiva per lo streaming audio/video è mostrata nella Figura 8.1. In questa architettura:

- Il processo del browser stabilisce una connessione TCP con il server Web e richiede il file audio/video con un messaggio di richiesta HTTP.
- Il server Web invia al browser il file audio/video in un messaggio di risposta HTTP.
- La linea di intestazione tipo di contenuto nel messaggio di risposta HTTP indica una codifica audio/video specifica. Il browser del client esamina il tipo di contenuto del messaggio di risposta, avvia il corrispondente media player, e passa il file al media player.
- Il media player riproduce quindi il file audio/video.

Sebbene questo approccio sia molto semplice, ha un grande inconveniente: il media player (cioè, l'applicazione helper) deve interagire con il server attraverso l'intermediazione di un browser Web. Questo può portare a molti problemi. In particolare, quando il browser è intermediario, l'intero oggetto deve essere scaricato prima che il

browser lo passi all'applicazione helper. Il risultante ritardo prima che la riproduzione possa cominciare può essere inaccettabile per clip audio/video di moderata lunghezza. Per questa ragione, nelle implementazioni dello streaming audio/video tipicamente il server invia il file audio/video direttamente al processo del media player. In altre parole, è stabilita una connessione socket diretta fra il processo del server e quello del media player.

- Audio: il file inviato come oggetto http
- Video: audio ed immagini interleaved in un singolo file, oppure due files separati inviati al client che sincronizza il display, inviati come oggetti http
- Il Browser richiede glio ggetti che vengono completamente scaricati e poi passati ad un helper per il display
- No pipelining
- Ritardo non accettabile per file di moderata lunghezza

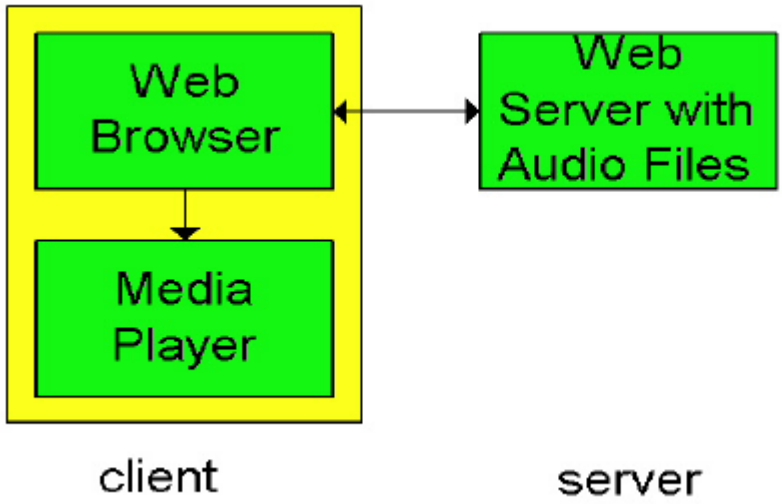


Figura 6.1 – Implementazione primitiva per lo streaming audio

Come mostrato nella Figura 8.2, questo si ottiene di solito usando un meta file, un file che fornisce informazioni (per esempio, URL, tipo di codifica) sul file audio/video che deve essere spedito. Una connessione TCP diretta fra il server e il media player è ottenuta nel modo seguente:

1. L'utente clicca su un hyperlink per un file audio/video.

2. L'hyperlink non punta direttamente sul file audio/video, punta invece su un meta file. Il meta file contiene l'URL del vero file audio/video. Il messaggio di risposta HTTP che incapsula il meta file comprende una linea di intestazione tipo del contenuto che indica la specifica applicazione audio/video.

3. Il browser del client esamina la linea di intestazione tipo del contenuto del messaggio di risposta, lancia il media player associato e passa l'intero corpo del messaggio di risposta (cioè, il meta file) al media player.

4. Il media player imposta una connessione TCP direttamente con il server HTTP. Il media player invia un messaggio di richiesta HTTP per il file audio/video nella connessione TCP.

5. Il file audio/video è inviato all'interno di un messaggio di risposta HTTP al media player. Il media player estrae lo stream del file audio/video.

L'importanza del passo intermedio per l'acquisizione del meta file è chiara. Quando il browser vede il tipo di contenuto del file, esso può lanciare l'appropriato media player, e in questo modo far sì che il media player contatti direttamente il server. Abbiamo appena imparato che un meta file può permettere a un media player di dialogare direttamente con un server Web in cui dimora un audio/video. Però, molte società che vendono prodotti per lo strearning audio/video non raccomandano l' architettura appena descritta. Questo perché nell' architettura il media player comunica con il server su HTTP e quindi anche su TCP. L'HTTP è spesso considerato insufficientemente ricco per un interazione soddisfacente dell'utente con il server, in particolare, l'http non permette a un utente di inviare con facilità al server i comandi pausa/riprendi, avanzamento rapido, salti temporali.

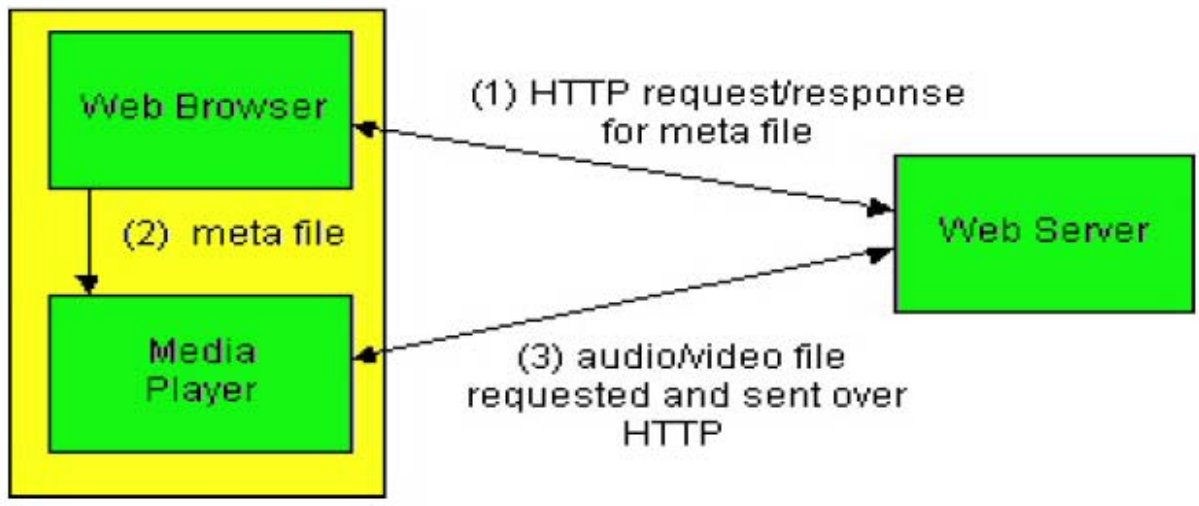


Figura 8.2 – Il servere Web invia audio/video direttamente al media player

In sintesi:

- Alternativa: stabilisci un collegamento socket diretto tra server ed media player
- Web browser richiede e riceve un **Meta File** (un file che descrive l'oggetto da scaricare) invece del file stesso
- Il browser lancia l'appropriato helper e gli passa il Meta File;
- Il media player stabilisce una connessione HTTP con il Web Server ed invia un messaggio di richiesta
- Il file audio/video è inviato dal server al media player

Osserviamo che la richiesta del metafile:

- Non permette di interagire in modo strutturato con il server, ex: pause, rewind
- E'vincolato ad usare TCP

Invio di multimedia da uno streaming server a un'applicazione helper

Per poter evitare l'HTTP e/o il TCP, audio/video possono essere immagazzinati su e inviati da uno streaming server al media player. Questo streaming server potrebbe essere uno streaming server proprietario, come quelli commercializzati da RealNetworks e Microsoft, o potrebbe essere uno streaming server di pubblico dominio. Con uno streaming server, audio/video possono essere spediti su UDP (piuttosto che su TCP) usando protocolli dello strato delle applicazioni che possono essere più adeguati dell'HTTP allo streaming audio/video.

Quest'architettura richiede due server, come mostrato nella Figura 8.3. Un server, il server HTTP, serve pagine Web (compresi i meta file). Il secondo server, lo streaming server, serve i file audio/video. I due server possono funzionare sullo stesso terminale o su due terminali distinti. I passi per quest'architettura sono simili a quelli descritti nell'architettura precedente. Comunque, ora il media player richiede i file da uno streaming server invece che da un server Web, e ora il media player e lo streaming server possono interagire usando loro propri protocolli. Questi protocolli possono permettere una ricca interazione dell'utente con lo stream audio/video. Nella architettura di Figura 8.3, ci sono diverse opzioni per la consegna degli audio/video dallo streaming server al media player. Un elenco parziale delle opzioni è riportato di seguito**.**

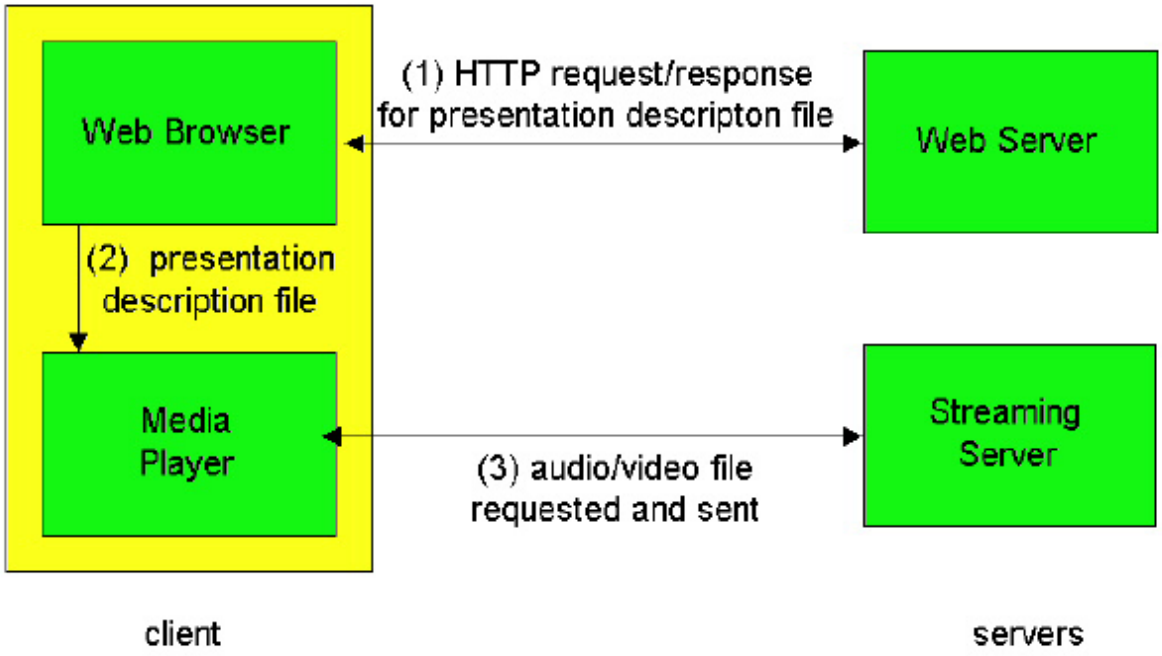


Figura 8.3 – streaming da uno streaming server ad un media player

- Usa UDP, ed il Server invia ad una velocità (Compressionee Trasmissione) appropriata per il client; per ridurre lo jitter, il Player bufferizza inizialmente per 2-5 secondi, quindi inizia il display
- Sender usaTCP alla massima velocità possibile; ritrasmette quando un errore viene incontrato; il Player utilizza un buffer di dimensioni molto maggiori per ammortizzare la velocità di trasmissione fluttuante di TCP

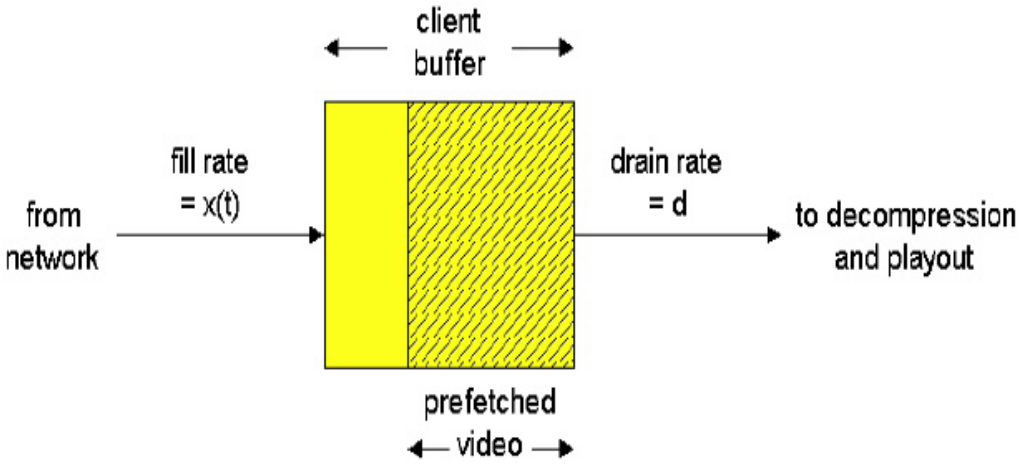


Figura 8.4 – il buffer del client viene riempito al tasso x(t) e svuotato alla velocità d

Protocollo di streaming in tempo reale

Molti utenti dei multimedia in Internet (soprattutto quelli che sono cresciuti con un telecomando TV in mano) vorrebbero poter *controllare* la riproduzione dei media continui *(continuous-media)* con pause, riposizionamenti in avanti o all'indietro, avanzamento rapido, riavvolgimento, e così via. Queste funzionalità sono simili a quelle permesse all'utente da un DVD player durante la visione di video DVD o da un riproduttore per CD durante l'ascolto di musica. Per permettere all'utente di controllare la riproduzione, il media player e il server necessitano di un protocollo per scambiarsi le informazioni di controllo della riproduzione. Il protocollo per lo streaming in tempo reale (RTSP, *Real-Time Streaming Protocol),* definito nella RFC 2326, è questo protocollo.
Ma prima di entrare nei dettagli dell'RTSP, indichiamo che cosa *non* può fare.

• L'RTSP non definisce gli schemi di compressione per audio e video.

• L'RTSP non definisce come audio e video sono incapsulati in pacchetti per la trasmissione sulla rete; l'incapsulamento per i media dello stream deve essere fornito dall'RTP o da un protocollo proprietario. Per esempio, server e player audio/video della RealNetwork usano l'RTSP per scambiarsi le informazioni di controllo, ma il flusso dei media è incapsulato in pacchetti RTP o in qualche formato proprietario dei dati.

• L'RTSP non limita il modo in cui i dati dello stream dati del media sono trasportati; possono essere trasportati su UDP o TCP.

• L'RTSP non pone limiti a come il media player memorizza gli audio/video. Gli audio/video possono essere riprodotti nello stesso momento in cui arrivano al client, possono essere riprodotti dopo un ritardo di pochi secondi o possono essere scaricati interamente prima della riproduzione.

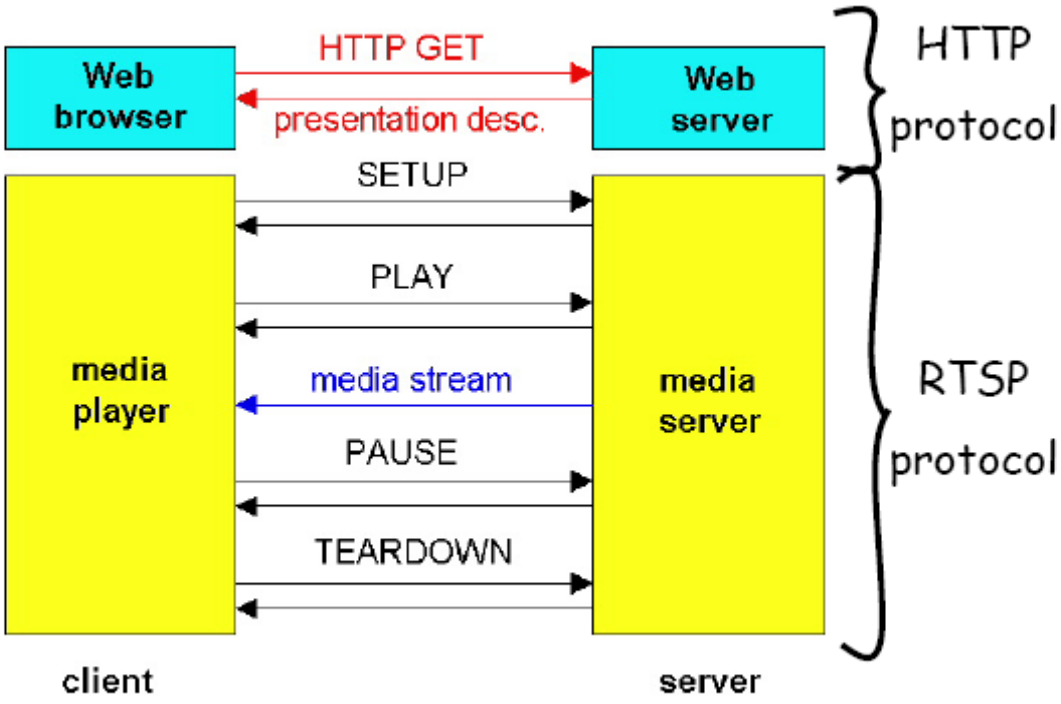
L'RTSP è un protocollo che permette a un media player di controllare la trasmissione di un flusso *(stream)* di media. Come detto sopra, le azioni di controllo comprendono pausa/ripristino, riposizionamento della riproduzione, avanzamento rapido e riavvolgimento. L'RTSP è un cosiddetto protocollo fuori banda *(out-of-band protocol).* In particolare, i messaggi RTSP sono inviati fuori banda, mentre il flusso dei media, la cui struttura del pacchetto non è definita da RTSP, è considerato "in banda." I messaggi RTSP usano un numero di porta diverso, 544, dal flusso dei media.

La specifica dell'RTSP [RFC 2326] consente ai messaggi RTSP di essere inviati sia su TCP sia su UDP. Ricordiamo che anche il protocollo di trasferimento dei file (FTP, *File Transfer Protocol)* usa il concetto di fuori banda. In particolare, l'FTP usa due coppie di socket client/server, ciascuna coppia con il suo proprio numero di porta: una coppia di socket client/server supporta una connessione TCP che trasporta le informazioni di controllo; l'altra coppia di socket client/server supporta una connessione TCP che effettivamente trasposta il file. Il canale RTSP è per molti versi simile al canale di controllo dell'FTP:

Esempio di Metafile

```
<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>
```

Comandi RTSP



Esempio di comunicazione 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

Test

1	Lo streaming indica la possibilità di presentare via Internet un filmato, o un pezzo audio, direttamente, ossia senza dover prima aspettare che l'intero file sia stato scaricato	SI NO
2	Per evitare che il ritardo prima che la riproduzione possa cominciare possa essere inaccettabile per clip audio/video di moderata lunghezza nelle implementazioni dello streaming audio/video tipicamente il server invia il file audio/video direttamente al processo del media player	SI NO
3	Con uno streaming server, audio/video possono essere spediti su su TCP usando protocolli dello strato delle applicazioni che possono essere più adeguati dell'HTTP allo streaming audio/video.	SI NO

TECNICO DELLE RETI

MODULO 9

3° parte

Scuola Radio Elettra®

Unità di Apprendimento
N. 8.3
Titolo: UN ESEMPIO DI TELEFONIA INTERNET
Obiettivo: Comprendere il funzionamento della telefonia Internet

Il protocollo dello strato di rete di Internet, IP, fornisce un servizio best-effort. Vale a dire che Internet fa del suo meglio per muovere ciascun datagramma dalla sorgente alla destinazione il più velocemente possibile. Comunque, il servizio best-effort non fa alcuna promessa relativa all'ammontare del ritardo end-to-end per un pacchetto individuale, o all'ammontare del jittere alla perdita dei pachetti all'interno del flusso degli stessi. La mancanza di garanzie sul ritardo e sul jitter di pacchetto crea notevoli problemi per la progettazione di applicazioni multimediali in tempo reale come la telefonia via Internet e la videoconferenza in tempo reale, che sono fortemente sensibili al ritardo di pacchetto, al jitter, e alla perdita. Fortunatamente i progettisti di queste applicazioni possono introdurre molti utili meccanismi che mantengono una buona qualità audio e video nei casi in cui ritardi, jitter e perdite non siano eccessivi.

Lo speaker nella nostra applicazione di telefonia Internet genera un segnale audio che consiste di un'alternanza di periodi di attività e periodi di silenzio. Per poter risparmiare la larghezza di banda, la nostra applicazione di telefonia Internet genera pacchetti solo nella fase di attività vocale. Durante la fase del parlato il sender genera byte a un tasso di 8000 byte al secondo, e ogni 20 millisecondi riunisce i byte in blocchi. Quindi, il numero di byte in un blocco è (20 ms) (8 kbyte/s) = 160 byte. A ciascun blocco è collegata una speciale intestazione, il cui contenuto sarà descritto qui sotto. Il blocco e la sua intestazione sono incapsulati in un segmento UDP, attraverso l'interfaccia del socket. Quindi, durante la fase di emissione delle parole, ogni 20 ms viene inviato un segmento UDP. Se ciascun pacchetto arriva al receiver con un piccolo e costante ritardo end-to-end, allora durante la fase di emissione delle parole i pacchetti arrivano al receiver periodicamente ogni 20 ms. In queste condizioni ideali, il receiver può semplicemente riprodurre ciascun blocco nel momento del suo arrivo. Ma, sfortunatamente, alcuni pacchetti possono essere persi e la maggior parte di essi non avrà lo stesso ritardo end-to-end, anche in un Internet leggermente congestionata. Per questo motivo, il receiver deve fare molta attenzione nel (1) determinare il momento in cui riprodurre un blocco, e (2) decidere che cosa fare con un blocco mancante.

Limitazioni di un servizio best - effort

Abbiamo detto che il servizio best – effort può portare a perdita di pacchetti, eccessivo ritardo end-to-end e jitter dei ritardi. Esaminiamo questi argomenti.

Perdita dei pacchetti

Consideriamo uno dei segmenti UDP generati dalla nostra applicazione di telefonia Internet. Il segmento UDP è incapsulato in un datagramma IP. Mentre il datagramma vaga per la rete, esso passa attraverso buffer (cioè, code) nei router per poter accedere ai link di uscita. È possibile che uno o più buffer sul percorso dal sender al receiver siano pieni e non possano ammettere il datagramma IP. In questo caso, esso viene scartato, e non arriverà mai all' applicazione del receiver.

Le perdite possono essere eliminate inviando i pacchetti su TCP invece che su UDP. Ricordiamo che il TCP ritrasmette i pacchetti che non arrivano a destinazione. Comunque, i meccanismi di ritrsmissione sono spesso considerati inaccettabili per le applicazioni interattive audio in tempo reale come la telefonia Internet, perché aumentano il ritardo end-to-end. inoltre, a causa del meccanismo di controllo della congestione del TCP, dopo la perdita del pacchetto la velocità di trasmissione al sender si può ridurre a un valore che è più basso di quello di svuotamento al receiver. Questo può avere un impatto severo sull'intelligibilità della voce al receiver. Per questi motivi, quasi tutte le applicazioni esistenti di telefonia Internet funzionano su UDP e non si curano della ritrasmissione dei pacchetti persi.

Ma la perdita dei pacchetti non è necessariamente così grave come si può pensare. Infatti, tassi di perdita dei pacchetti fra l'1 e il 20% possono essere tollerati, in funzione di come la voce è codificata e trasmessa, e di come la perdita è mascherata al receiver

Ritardo end - to - end

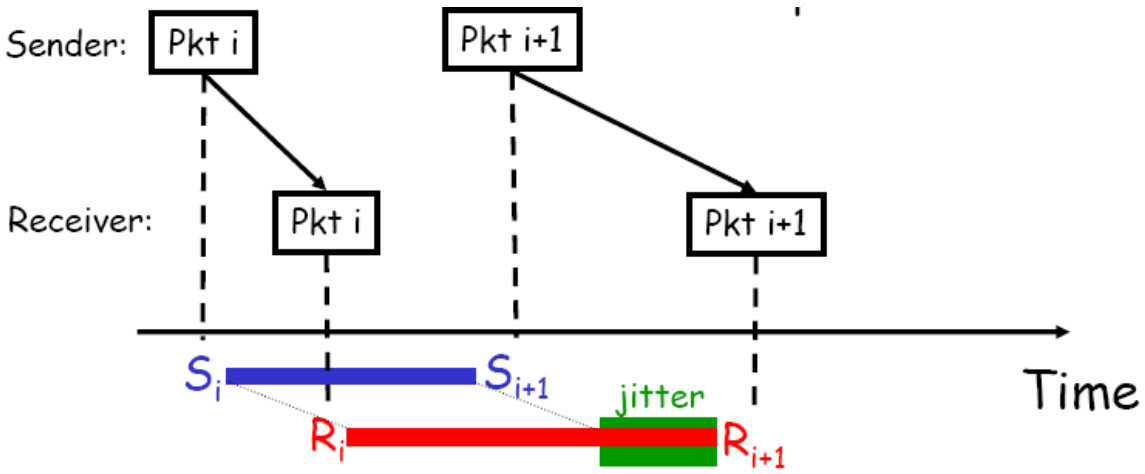
Il ritardo end-to-end è la somma dei ritardi di accodamento, dei ritardi di elaborazione, di trasmissione e di coda ai router, dei ritardi di propagazione, e dei ritardi di elaborazione nei terminali lungo un percorso fra sorgente e destinazione. Per le applicazioni audio altamente interattive, come la telefonia Internet, ritardi end-to-end infenon a 150 ms non sono percepiti all'ascoltatore umano; ritardi fra 150 e 400 ms possono essere accettabili ma non ideali; e ritardi che superano i 400 ms possono ostacolare seriamente l'interattività nella conversazione. Il receiver in un'applicazione di telefonia Internet tipicamente trascurerà qualsiasi pacchetto con ritardi superiori a una certa soglia, per esempio, oltre i 400 ms. Quindi, i pacchetti che hanno ritardi che superano la soglia sono effettivamente persi.

Jitter dei ritardi

Un componente cruciale del ritardo end-to-end è il ritardo di coda casuale ai.router. A causa di questi ritardi variabili nella rete, il tempo da quando un pacchetto è generato alla sorgente a quando è ricevuto al receiver può fluttuare da pacchetto a pacchetto. Questo fenomeno è detto jitter.

Come esempio, consfderiamo due pacchetti consecutivi all'interno di un periodo di attività nella nostra applicazione di telefonia Internet. Il sender invia il secondo pacchetto 20 ms dopo aver inviato il primo. Ma al receiver, la distanza tra i due pac

Lo jitter di una coppia di pacchetti è la differenza tra l'intervallo di tempo che intercorre tra la trasmissione e la ricezione dei due pacchetti



Intervallo rcv desiderato: $S_{i+1} – S_i$, mentre l'intrervallo rcv è: $R_{i+1} - R_i$

Bitter tra pacchetti i e i+1: $(R_{i+1} - R_i) – (S_{i+1} – S_i)$

Audio: rimozione dello jitter al receiver

Per un' applicazione vocale come la telefonia Internet o per l' audio a richiesta, il receiver deve tentare di fornire la riproduzione sincrona dei blocchi vocali in presenza di jitter di rete variabile. Ciò si ottiene di solito combinando tre meccanismi:

• *Facendo precedere ciascun pezzo da un* numero di sequenza. Il sender incrementa numero di sequenza di un'unità per ciascun pacchetto che genera.

• *Facendo precedere ciascun pezzo da una* marcatura temporale. Il sender contrassegna ciascun blocco con il tempo al quale il blocco è generato.

• *Ritardando la riproduzione del blocco al receiver.* Il ritardo di riproduzione al receiver per i blocchi audio deve essere abbastanza lungo da consentire la ricezione della maggior parte dei pacchetti prima del tempo fissato per la loro riproduzione. Questo ritardo di riproduzione può essere fisso per la durata della conferenza, o può essere variato durante la conferenza stessa. I pacchetti che non arrivano in tempo sono considerati persi e dimenticati; come notato sopra, il receiver può usare una sorta di interpolazione del parlato per tentare di nascondere le perdite. Vedremo ora come questi tre meccanismi, quando usati in combinazione, possono alleviare o anche eliminare del tutto gli effetti del jitter. Esamineremo due principali strategie di riproduzione: con ritardo di riproduzione fissato e con ritardo di riproduzione adattativo.

Ritardo di riproduzione fissato

Con la strategia del ritardo fissato, il receiver tenta di riprodurre ciascun blocco esattamente q millisecondi dopo la sua generazione. Così se un blocco è contrassegnato da un tempo di generazione t, il receiver lo riproduce al fempo t + q, assumendo che il blocco sia arrivato entro quel tempo. I pacchetti che arrivano dopo il tempo programmato per la loro riproduzione sono scartati e considerati persi. Qual è la miglior scelta per q? La telefonia Internet può supportare ritardi fino a 400 ms, sebbene un'esperienza interattiva più soddisfacente si raggiunge con valori di q più piccoli. D'altra parte, se q è scelto molto più piccolo di 400 ms, allora molti pacchetti mancheranno il tempo programmato per la loro riproduzione a causa del jitter del ritardo indotto dalla rete. Approssimativamente parlando, se sono tipiche ampie variazioni nel ritardo end-to-end, è preferibile usare q grande; d'altra parte, se il ritardo è piccolo yariazioni nel ritardo sono anch'esse piccole, è preferibile avere q piccolo, magari inferiore a 150 ms. Il legame fra il ritardo di riproduzione e la perdita dei pacchetti è illustrato nella Figura 8.6. La figura mostra i tempi a cui i pacchetti sono generati e riprodotti per un singolo flusso di parole. Sono considerati due ritardi iniziali di riproduzione distinti. Come mostra la linea a gradini più a sinistra, il sender genera pacchetti a intervalli regolari: diciamo, ogni 20 ms. Il primo pacchetto in questo flusso di parole è ricevuto al tempo *r.* Come illustrato nella figura, gli arrivi dei pacchetti successivi non sono ugualmente spaziati a causa del jitter di rete. Per i tempi programmati per la prima riproduzione, il ritardo di riproduzione iniziale fissato è impostato a p - *r.* Con questa programmazione, il quarto pacchetto non arriva entro il tempo programmato per la sua riproduzione, e il receiver lo considera perso.

Per il secondo schema dei tempi di riproduzione, il ritardo di riproduzione iniziale fissato è impostato a p' - *r.* Per questa programmazione, tutti i pacchetti arrivano prima dei tempi programmati per la loro riproduzione, e non ci sono perdite.

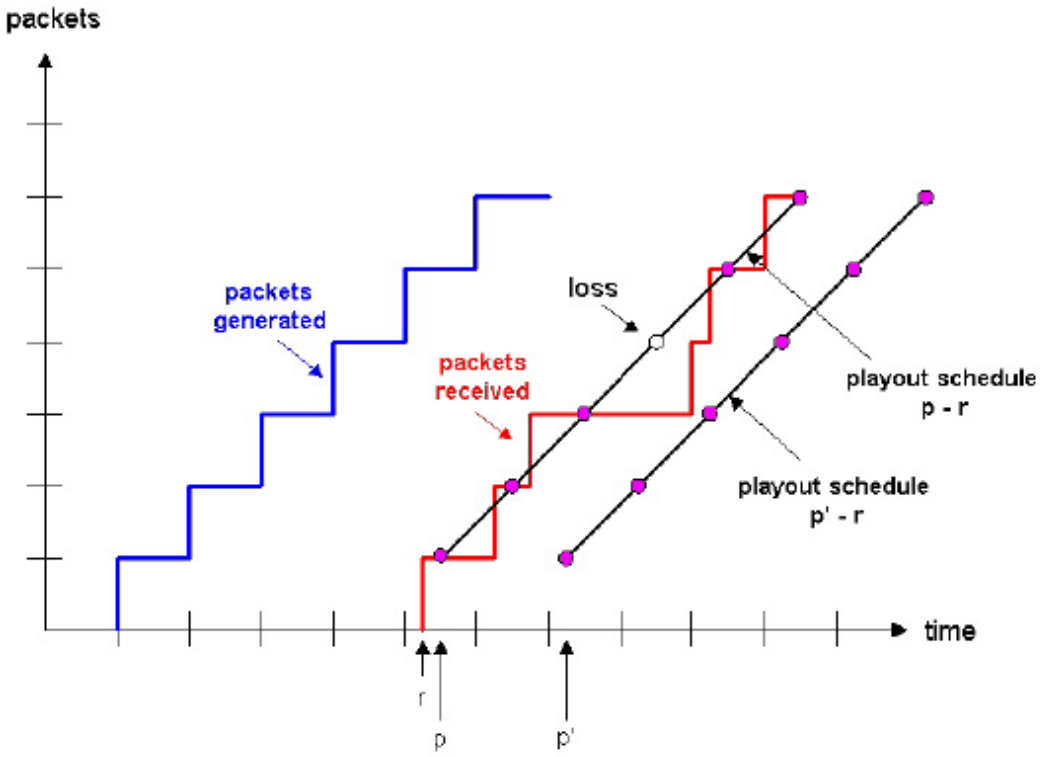


Figura 8.6 – Perdita di pacchetti per diversi ritardi di riproduzione fissati

L'esempio sopra dimostra un importante compromesso fra ritardo e perdita dei pacchetti che si stabilisce quando si progetta una strategia con ritardi di riproduzione fissati. Scegliendo un ritardo di riproduzione ampio, molti pacchetti arriveranno in tempo utile e ci saranno quindi perdite trascurabili; comunque, per un servizio interattivo, come la telefonia Internet, lunghi ritardi possono diventare irritanti se non intollerabili. Idealmente, vorremmo un ritardo di riproduzione minimo con il vincolo che la perdita sia al di sotto di pochi punti percentuali.

Il modo naturale di trattare questo comportamento è stimare il ritardo della rete e le sue variazioni, e regolare il ritardo di riproduzione di conseguenza all'linizio di ciascun periodo di attività vocale. Questa regolazione adattata del ritardo di riproduzione all'inizio dei periodi di attività vocale farà sì che i periodi silenti del sender saranno compressi o prolungati; comunque, la compressione e l'allungamento dei silenzi di una piccola quantità non è rilevabile nel parlato.

Descriviamo ora un algoritmo generico che il receiver può usare per la regolazione adattata dei suoi ritardi di riproduzione. Poniamo:

t_i= marcatura temporale dell'i-esimo pacchetto = tempo a cui il pacchetto è generato dal sender
r_i = tempo a cui il pacchetto è ricevuto dal receiver

p_i = tempo a cui il pacchetto è riprodotto al receiver

Il ritardo di rete end-to-end dell'i-esimo pacchetto è $r_i - t_i$. A causa del jitter della rete, questo ritardo varierà da pacchetto a pacchetto. Indichiamo con d_i una stima del valore medio del ritardo di rete alla ricezione dell'i-esimo pacchetto. Questa stima è ricavata dalle marcature temporali come segue:

$$d_i = (1-u)\, d_{i-1} + u\,(r_i - ti)$$

dove *u è* una costante fissata (per esempio, *u* = 0, 01). Quindi *d è* una media livellata dei ritardi di rete osservati $r_1 — t_1, \ldots, r_i — t_i$.

La stima assegna più peso ai ritardi di rete più recentemente osservati rispetto a quelli osservati nel lontano passato. Questa forma di stima non dovrebbe essere del tutto sconosciuta.. Indichiamo con *v* una stima della deviazione media del ritardo dal ritardo medio stimato. Anche questa stima è ricavata dalle marcature temporali:

$$vi = (1-u)\, v_{i-1} + u\,|r_i - ti - d_i|$$

Le stime di d_i e v_i sono calcolate per ciascun pacchetto ricevuto, sebbene possano essere usate solo per determinare il punto di riproduzione per il primo pacchetto in qualsiasi periodo di attività.

Una volta calcolate queste stime, il receiver impiega i seguenti algoritmi per la riproduzione dei pacchetti. Se il pacchetto *i* è il primo pacchetto di un periodo di attività, il suo tempo di riproduzione, p, è calcolato come:

$$p_i = t_i + d_i + K\, v_i$$

dove K è una costante positiva (per esempio, $K = 4$). Lo scopo del termine $K\ v_i$ è di impostare un tempo di riproduzione abbastanza lontano nel futuro così che solo una piccola frazione dei pacchetti nel periodo di attività in arrivo saranno persi a causa del ritardo nell'arrivo. Il punto di riproduzione per ogni pacchetto successivo in un periodo di attività è calcolato come uno spostamento dal punto nel tempo in cui il primo pacchetto del periodo di attività è riprodotto. In particolare poniamo che

$$q_j = p_i - t_i$$

sia il tempo che trascorre da quando il primo pacchetto nel periodo di attività è generato al momento della sua riproduzione. Se anche il pacchetto *j* appartiene a questo periodo di attività, esso viene riprodotto al tempo

$$p_j = t_j + q_i$$

L'algoritmo appena descritto funziona perfettamente assumendo che il receiver possa dire se un pacchetto è il primo pacchetto del periodo di attività. Se non c'è perdita di pacchetti, allora il receiver può determinare se il pacchetto *i* è il primo pacchetto del periodo di attività confrontando la marcatura temporale dell' i-esimo pacchetto con la marcatura temporale dell'(i - 1)-esimo pacchetto. Infatti, se t_i — t_{i-1}> 20 ms, allora il receiver sa che l' i-esimo pacchetto è l'inizio di un nuovo periodo di attività. Ma supponete ora che intervenga un'occasionale perdita di un pacchetto. In questo caso, due pacchetti ricevuti consecutivamente alla destinazione possono avere marcature temporali che differiscono di oltre 20 ms sebbene i pacchetti appartengano allo stesso periodo di attività. Qui i numeri di sequenza possono essere particolarmente utili. Il receiver può usare i numeri di sequenza per determinare se una differenza superiore ai 20 ms nelle marcature temporali è dovuta all'inizio di un nuovo periodo di attività o alla perdita di pacchetti.

Recupero dalla perdita dei pacchetti

Descriveremo brevemente diversi schemi che tentano di conservare una qualità audio accettabile in presenza di perdita dei pacchetti. Questi schemi sono detti schemi di recupero dalle perdite *(loss recovery schemes)*. Qui definiamo la perdita dei pacchetti in senso ampio: un pacchetto è perso se non arriva mai al receiver o se arriva dopo il tempo programmato per la sua riproduzione. Il nostro esempio di telefonia Internet servirà ancora da contesto per descrivere gli schemi di recupero dalle perdite. Come abbiamo visto, la

ritrasmissione dei pacchetti persi non è appropriata per un' applicazione interattiva in tempo reale come la telefonia Internet. Infatti, la ritrasmissione di un pacchetto che ha mancato il suo tempo di riproduzione non ha assolutamente scopo. E la ritrasmissione di un pacchetto che ha sovraccaricato la coda a un router non può avvenire abbastanza in fretta. Date queste considerazioni, le applicazioni di telefonia di Internet spesso usano alcuni schemi di anticipazione delle perdite. Due tipi di questi schemi sono la correzione dell'errore in avanti (FEC, *Forward Error Correction)* e l'interallacciamento(*interieaving*).

Correzione dell'errore in avanti (FEC)

L'idea base del FEC è di aggiungere un'informazione ridondante al flusso originale dei pacchetti. In cambio di un aumento marginale della velocità di trasmissione dell'audio nello sfream, l'informazione ridondante può essere usata per ricostruire una "approssimazione" dell'esatta versione di alcuni dei pacchetti persi. Tratteremo due meccanismi FEC. Il primo meccanismo invia un blocco di codifica ridondante dopo ogni n blocchi. Il blocco ridondante è ottenuto da un'operazione di OR esclusivo degli n blocchi originali. In questo modo se qualcuno dei pacchetti del gruppo degli n ±1 pacchetti è perso, il receiver può ricostruire completamente il pacchetto perso. Ma se in un gruppo sono persi due o più pacchetti, il receiver non può ricostruirli. Mantenendo piccolo n + 1, la dimensione del gruppo, gran parte dei pacchetti persi possono essere recuperati quando le perdite non sono eccessive. Comunque, più piccole sono le dimensioni del gruppo, più grande è l'incremento relativo della velocità di trasmissione dello stream audio. In particolare, la velocità di trasmissione aumenta di un fattore 1/n; per esempio, se n = 3, allora la velocità di trasmissione aumenta del 33%. Inoltre, questo semplice schema aumenta il ritardo di riproduzione, poiché il receiver deve aspettare di aver ricevuto l'intero gruppo di pacchetti prima di poterne iniziare la riproduzione.

Il secondo meccanismo FEC consiste nell'inviare uno stream audio a bassa risoluzione come informazione ridondante.

Interallacciamento

Come alternativa alla trasmissione ridondante, un'applicazione di telefonia Internet può inviare audio interallacciato. Come mostra la Figura 8.7, il sender risequenzia le unità di dati audio prima della trasmissione, in modo che le unità originariamente adiacenti siano separate da una certa distanza nel flusso trasmesso. L'interallacciamento può ridurre gli

effetti della perdita di pacchetti. Se, per esempio, le unità sono lunghe *5* ms e i blocchi sono di 20 ms (cioè, 4 unità per blocco), allora il primo blocco può contenere le unità 1, *5,* 9, 13; il secondo le unità 2, 6, 10, 14; e così via. La Figura 6.8 mostra che la perdita di un singolo pacchetto da un flusso interallacciato risulta in molte piccole lacune nel flusso ricostruito, all'opposto della singola grande lacuna che interviene in uno stream non interallacciato.

L'interleaving può migliorare significativamente la qualità percepita di uno stream audio Esso ha anche bassa ridondanza. Lo svantaggio ovvio è che incrementa la latenza. Questo limita il suo uso per le applicazioni interattive come la telefonia Internet, sebbene possa dare buone prestazioni con lo streaming di audio memorizzato. Il più grande vantaggio dell'interallacciamento è che non aumenta la larghezza di banda di uno stream.

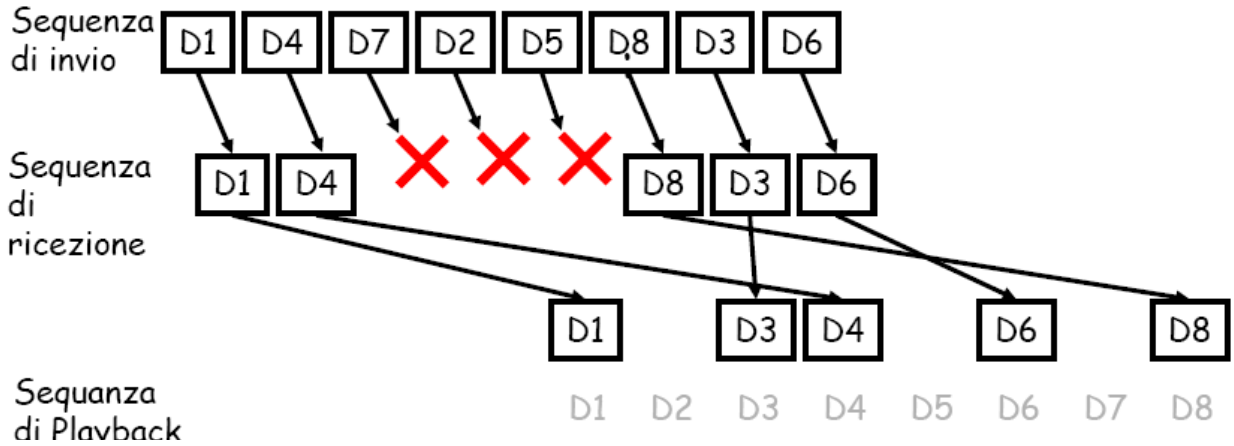


Figura8 . 7 – L'invio di un audio interallacciato

Lo steaming di audio e video memorizzato

Tipicamente le applicazioni per lo streaming di audio e video memorizzato usano anch'esse numeri di sequenza, marcature temporali e ritardi di riproduzione per alleviare o anche eliminare gli effetti del jitter di rete. Comunque c'è un'importante differenza fra l'audio/video interattivo e lo streaming di *audio/video* memorizzato. Nello specifico, lo streaming di audio/video memorizzato può tollerare ritardi significativamente grandi. Infatti, quando un utente richiede un clip audio/video, egli può trovare accettabile aspettare cinque o più secondi per l'inizio della riproduzione. Questa grande tolleranza sui ritardi consente ai programmatori una grande flessibilità nella progettazione delle applicazioni per i media immagazzinati in memoria.

Test

1	Il protocollo dello strato di rete di Internet, IP, fornisce un servizio best-effort.	SI
		NO
2	Le limitazioni del servizio best – effort sono: la perdita dei pacchetti e lo jitter dei ritardi	SI
		NO
3	Lo jitter di una coppia di pacchetti è la differenza tra l'intervallo di tempo che intercorre tra la trasmissione e la ricezione dei due pacchetti	SI
		NO

TECNICO DELLE RETI

MODULO 9

4° parte

Scuola Radio Elettra®

Unità di Apprendimento
N. 8.4
Titolo: PROTOCOLLI PER APPLICAZIONI INTERATTIVE IN TEMPO REALE
Obiettivo: Analisi dei protocolli per la applicazioni multimediali

Il lato sender di un'applicazione multimediale aggiunge campi di intestazione blocchi audio/video prima di passarli allo strato di trasporto. Questi campi di intestazione comprendono numeri di sequenza e marcature temporali. Poiché molte applicazioni multimediali funzionanti in rete possono usare numeri di sequenza e marcature temporali, conviene avere una struttura standardizzata di pacchetto che comprende i campi per audio/video, numeri di sequenza e marcature temporali così come altri campi potenzialmente utili. L'RTP, definito nella RFC 1889, è questo standard. L'RTP può essere usato per il trasporto di formati comuni come PCM, GSM e MP3 per il suono e MPEG e H.263 per il video: può anche essere utile per trasportare formati musica e video proprietari. Oggi, RTP viene implementato in centinaia prodotti e prototipi di ricerca. Esso è anche complementare ad altri importanti protocolli interattivi in tempo reale, come SIP e H.323.

Le basi di RTP

L'RTP funziona tipicamente sopra UDP. Il lato trasmittente incapsula un blocco di media all'interno di un pacchetto RTP, quindi incapsula il pacchetto in un segmento UDP e poi affida il segmento a IP. Il lato ricevente estrae il pacchetto RTP dal segmento UDP, quindi estrae il blocco di media dal pacchetto RTP e passa il blocco al media player per la decodifica e la riproduzione.

Come esempio, consideriamo l'uso dell'RTP per il trasporto della voce. Supponiamo che la sorgente vocale sia codificata PCM (cioè, campionata, quantizzata e digitalizzata) a 64 kbit/s e che l'applicazione raccolga i dati codificati in blocchi di 20 ms, vale a dire, 160 byte in un blocco. L'applicazione fa precedere ciascun blocco dei dati audio da un'intestazione RTP che comprende il tipo di codifica audio, un numero di sequenza e una marcatura temporale. L'intestazione RTP è normalmente di 12 byte. Il pezzo audio insieme all'intestazione RTP forma il pacchetto RTP. Il pacchetto RTP è quindi inviato nel socket di interfaccia UDP. Al lato receiver, l'applicazione riceve i pacchetti RTP dal suo socket di interfaccia.

L’applicazione estrae il blocco audio dal pacchetto RTP e usa i campi dell’intestazione del pacchetto per decodificare e riprodurre opportunamente il blocco audio. Se un’applicazione incorpora l’RTP (invece di uno schema proprietario per fornire tipo di carico utile, numero di sequenza o marcatura temporale) allora potrà interagire con altre applicazioni multimediali funzionanti in rete. Per esempio, se due diverse società sviluppano un software per la telefonia in Internet e nei loro prodotti incorporano l’RTP, c’è qualche speranza che un utente che usa uno dei due prodotti sarà in grado di comunicare con un utente che usa l’altro. L’RTP è spesso usato insieme agli standard per la telefonia in Internet. Deve essere enfatizzato che l’RTP di per se stesso non fornisce alcun meccanismo per assicurare la spedizione tempestiva dei dati o per fornire altre garanzie di qualità del servizio; allo stesso modo esso non garantisce la consegna dei pacchetti e non previene la consegna fuori ordine degli stessi. Infatti, l’incapsulamento di RTP è visto solo al terminale. I router non distinguono fra i datagramma IP che trasportano pacchetti RTP e quelli che non lo fanno. L’RTP permette che a ciascuna sorgente (per esempio, una videocamera o un microfono) sia assegnato il suo proprio indipendente stream (flusso) di pacchetti RTP. Per esempio, per una videoconferenza fra due partecipanti, possono essere aperti quattro stream RTP: due per trasmettere l’audio (uno in ciascuna direzione) due per il video (ancora uno in ciascuna direzione). Comunque, molte popolari tecniche di codifica (compresi MPEG 1 e MPEG2) riuniscono audio e video in un singolo stream durante il processo di codifica. Quando audio e video sono assembiati dal codificatore, è generato un solo stream RTP in ciascuna direzione.

I pacchetti RTP non sono limitati alle applicazioni unicast. Possono anche essere spediti su alberi multicast da uno - a -molti e da molti - a - molti. Per una sessione multicast da molti - a - molti, tutti i sender e le sorgenti della sessione usano tipicamente lo stesso gruppo multicast per inviare i loro stream RTP. Gli stream multicast RTP che hanno la stessa appartenenza, come gli stream audio e video emessi da più sender in un’applicazione di videoconferenza, appartengono a una sessione RTP.

- Real-Time Multimedia protocol di livello generale
  - Scalabile a sessioni con molti mittenti e molti riceventi
  - Dati di sessione inviati via RTP (Real-time Transfer Protocol)
- RTP deve essere integrato all'interno dell'applicazione
  - Applicazioni invia pacchetti RTP all'interno di un socket di interfaccia UDP
  - Programmatore deve prevedere l'estrazione dei dati applicazione dai pacchetti RTP
- Applicazioni che utilizzano codifiche diverse possono comunque interagire se inseriscono la specifica della codificane ll'header del pacchetto RTP
- Pacchetti RTP possono anche essere inviati su trasmissioni Multicast. Tutti i partecipanti usano lo stesso gruppo IP di multicast.

Campi di intestazione dei pacchetti RTP

Come illustrato nella Figura 8.9, i quattro campi principali dell'intestazione del pacchetto RTP sono il tipo di carico utile *(payload rype),*il numero di sequenza *(sequence number),* la marcatura temporale *(timestarnp)* e l'identificatore della sorgente *(source identfier).*

Figura 8.9 – Campi di intestazione RTP

Il campo tipo di carico utile nel pacchetto RTP ha lunghezza di sette bit. Per uno stream audio, il campo tipo di carico utile è usato per indicare il tipo di codifica dell' audio (per esempio, PCM, modulazione delta adattativa, codifica lineare predittiva) che è stata usata. Se un sender decide di variare la codifica nel mezzo di una sessione, può informare il receiver della variazione attraverso questo campo tipo di carico utile Il sender può voler cambiare la codifica per aumentare la qualità audio o per diminuire la velocità di bit dello stream RTP.

Per uno stream video, il tipo di carico utile è usato per indicare il tipo di codifica video (per esempio, JPEG dinamici, MPEG 1, MPEG 2, H.261). Ancora, il sender può variare la codifica video in corso di trasmissione all'interno di una sessione. Campi importanti sono:

• *Campo numero di sequenza.* Il campo numero di sequenza è lungo 16 bit. Il numero di sequenza incrementa di uno per ciascun pacchetto RTP inviato, e può essere usato dal receiver per rilevare perdite e per ricostruire la sequenza dei pacchetti. Per esempio, se il lato receiver dell' applicazione riceve un flusso di pacchetti RTP con un buco fra i numeri di sequenza 86 e 89, allora il receiver sa che i pacchetti 87 e 88 sono mancanti. Il receiver può allora tentare di coprire i dati persi.

• *Campo marcatura temporale.* Il campo marcatura temporale *(timestamp field) è* lungo 32 bit. Esso riflette l'istante del campionamento del primo byte nel pacchetto dati RTP. Come abbiamo visto nel paragrafo precedente, il receiver può usare la marcatura temporale per poter rimuovere il jitter dei pacchetti introdotto nella rete e per fornire la riproduzione sincronizzata al receiver. La marcatura temporale è derivata da un orologio di campionamento del sender.

• *Identficatore della sorgente di sincronismo (SSRC,).* Il campo SSRC è lungo 32 bit e identifica la sorgente dello stream RTP. Tipicamente, ciascuno stream in una sessione RTP ha un diverso SSRC. L'SSRC non è l'indirizzo IP del sender, ma piuttosto un numero che la sorgente assegna arbitrariamente quando inizia un nuovo stream. La probabilità che a due stream venga assegnato lo stesso numero è molto bassa.

Protocollo di controllo RTP

La RFC 1889 specifica anche l'RTCP *(RTP Control Protocol),* un protocollo che un'applicazione multimediale può usare insieme all'RTP. Come mostrato nello scenario multicast di Figura 8.12, i pacchetti RTCP sono trasmessi da ciascun partecipante a una sessione RTP a tutti gli altri partecipanti alla sessione usando l'IP multicast. Tipicamente, per una sessione RTP c'è un singolo indirizzo multicast e tutti i pacchetti RTP e RTCP appartenenti alla sessione lo usano. I pacchetti RTP e RTCP sono distinguibili l'un l'altro dai loro diversi numeri di porta. (Il numero di porta di RTCP è posto uguale al numero di porta RTP più uno.)

I pacchetti RTCP non incapsulano pezzi di audio o video: piuttosto, essi sono inviati periodicamente e contengono rapporti del sender/receiver che comunicano dati statistici utili all' applicazione. Questi dati statistici comprendono numero di pacchetti spediti, numero di pacchetti persi e jitter tra gli arrivi. La specifica RTP [RFC 1889] non prescrive che cosa l'applicazione dovrebbe fare con queste informazioni di feedback; la decisione è lasciata al programmatore. I sender possono usare le informazioni di feedback, per esempio, per modificare le loro velocità di trasmissione. Queste informazioni possono essere anche usate per scopi diagnostici; per esempio, i receiver possono determinare se un problema è locale, regionale o globale.

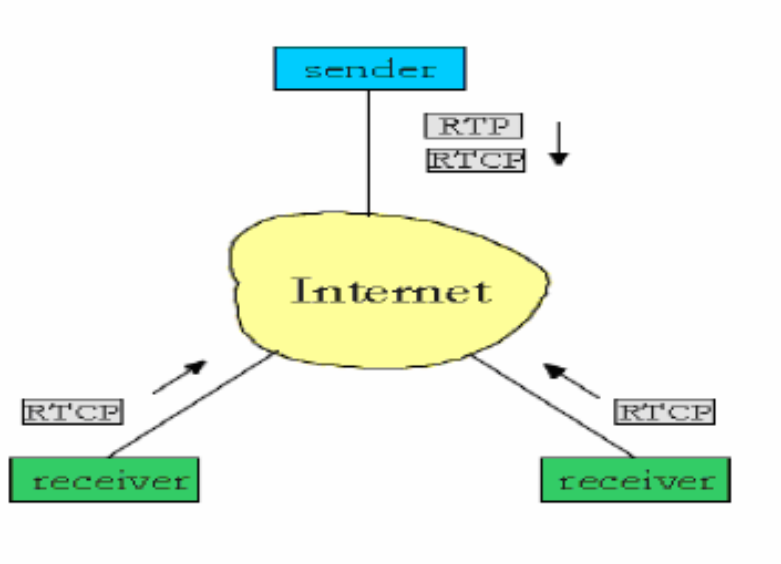


Figura 8.12 – Sender e riceiver inviano entrambi messaggi RTP

Tipi di pacchetti RTCP

Per ciascuno stream RTP che arriva a un receiver come parte di una sessione, il receiver genera un rapporto di ricezione. Il receiver raggruppa i suoi rapporti di ricezione in un singolo pacchetto RTCP. Il pacchetto è poi inviato nell'albero multicast che collega insieme tutti i partecipanti alla sessione. Il rapporto di ricezione comprende molti campi, i più importanti dei quali sono elencati di seguito.

- SR: sender report –statistiche di trasmissionee ricezione. Usato per sincronizzare diversi stream media o per associare lo stream alla sua sorgente
- RR: receiver report –stato di ricezione. Usato per modificare velocità di trasmissione o codifica

- SDES: descrizione della sorgente
- BYE: end-of-participation
- APP: funzione specifiche dell'applicazione

Pacchetti RTCP di diverso tipo possono essere trasmessi all'interno dello stesso pacchetto UDP

Funzionalità di RTCP

- Info per determinare collisione nell'identificatore dello stream
- Informazioni sull'identità dei partecipanti
- Informazioni per stabilire il numero di sessioni partecipanti
- Qualità della ricezione dei partecipanti

Controllo della congestione in RTCP

Per il controllo della congestione in RTCP vi è una semplice regola: la banda totale usata per pacchetti RTCP deveessere il 5% della banda usata per la sessione RTP

- 75% della banda RTCP per i riceventi
- 25% per il mittente

Riassumendo il periodo per la trasmissione dei pacchetti RTCP per un sender è:

$$T = \frac{\text{numero dei sender}}{0{,}25 \times 0{,}05 \times \text{larghezza di banda sessione}} \text{ (dim. media pacchetto RTCP)}$$

E il periodo usato per trasmettere i pacchetti RTCP per un riceiver è:

$$T = \frac{\text{numero dei receiver}}{0{,}75 \times 0{,}05 \times \text{larghezza di banda sessione}} \text{ (dim. media pacchetto RTCP)}$$

- Obiettivo: impedire l'esplosione di banda nel caso in cui tutti i partecipanti si uniscono nello stesso momento

- I riceventi che si uniscono inizialmente rilevano un piccolo numero di partecipanti nella sessione
- Soluzione per il momento iniziale di partecipazione:

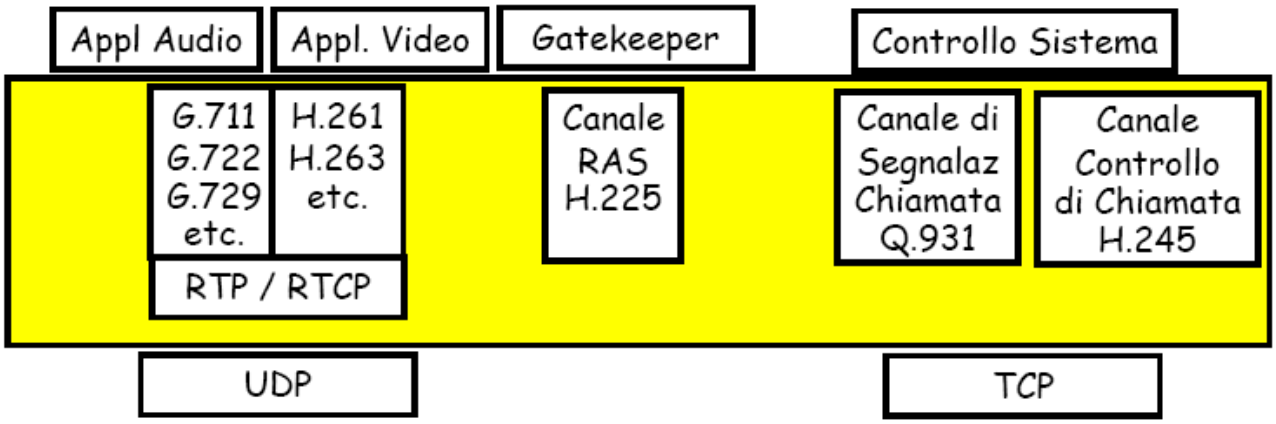
    1. CalcolaT, ed attendi un intervallo di tempo casuale
    2. Alla fine dell'intervallo, stima nuovamente il numero di partecipanti
    3. Se il numero di partecipanti è aumentato, calcola nuovamente T'
    4. Se T'< T, invia immediatamente
    5. Se T'>= T, attendi un ulteriore T', vai al passo2
- Successivamente, usa il normale periodo di attesa

H.323

Si tratta di un diffuso standard per conferenze audio e video in tempo reale tra terminali Internet. Lo standard si occupa anche di come i terminali attaccati ad Iternet comunicano con telefoni attaccati a normali reti telefoniche a commutazione di circuito.

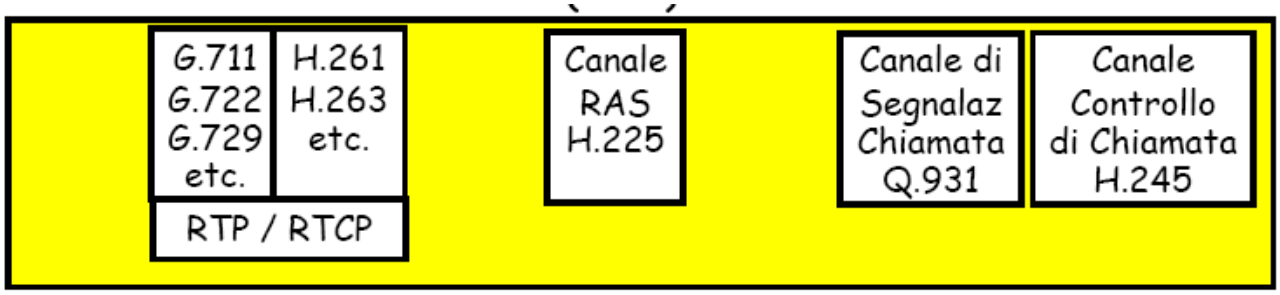
Componenti di Rete:

- **terminali**: host terminali H.323-compliant
- **gateways**: interfacce tra terminali H.323-compliant e tecnologie precedenti (ex: retetelefonica)
- **gatekeepers**: forniscono servizi ai terminali (ex: traduzione di indirizzi, tariffazione, autorizzazione, etc...)



- H.225: notifica gatekeepers dell'inizio della sessione
- Q.931: protocollo di segnalazione per stabilire e terminare le chiamate

- H.245: protocollo fuoribanda per negoziare i codici di compressione audio/video da utilizzare durante la sessione(TCP)



Servizi forniti ai terminali H.323:

- Traduzione da alias dei terminali ad indirizzi IP
- Gestione larghezza di banda per preservare la qualità
- Terminali H.323 registrano presso Gatekeeperdi zona con IP ed alias
- Terminali chiedono a Gatekeeperil permesso di realizzare una chiamata

Test

1	L’RTP funziona tipicamente sopra UDP. Il lato trasmittente incapsula un blocco di media all’interno di un pacchetto RTP, quindi incapsula il pacchetto in un segmento UDP e poi affida il segmento a IP	SI
		NO
2	I pacchetti RTP sono limitati solo alle applicazioni unicast	SI
		NO
3	I pacchetti RTCP sono trasmessi da ciascun partecipante a una sessione RTP a tutti gli altri partecipanti alla sessione usando l’IP multicast.	SI
		NO

TECNICO DELLE RETI

MODULO 9

5° parte

Scuola Radio Elettra®

Unità di Apprendimento
N. 8.5
Titolo: OLTRE IL BEST - EFFORT
Obiettivo: identificazione di componenti per proteggere la un'applicazione dalle congestioni

L'Internet attuale fornisce un servizio besteffort a tutte le sue applicazioni, vale a dire, non fa alcuna promessa sulla qualità del servizio (QoS) che un applicazione riceve. Un applicazione riceverà qualsiasi livello di prestazioni (per esempio, ritardo end-to-end dei pacchetti e perdite) che la rete è in grado di fornire in quel momento. Ricordiamo inoltre che l'Internet pubblica odierna non permette alle applicazioni multimediali sensibili al ritardo di richiedere alcun trattamento speciale. Ai router tutti i pacchetti sono trattati nello stesso modo, compresi quelli audio e video sensibili ai ritardi. Poiché tutti i pacchetti sono trattati equamente, per rovinare la qualità di una chiamata telefonica IP in corso in Internet basta una quantità sufficiente di traffico (cioè, congestione della rete) per aumentare considerevolmente ritardi e perdita di pacchetti già subiti da una chiamata telefonica IP.

In questo unità identificheremo nuovi componenti strutturali che possono essere aggiunti all' architettura di Internet per proteggere un' applicazione da queste congestioni e far sì che l'alta qualità delle applicazioni multimediali funzionanti in rete diventi realtà.

I scenario: un'applicazione audio a 1 Mbit/s e un trasferimento FTP

Il primo scenario è illustrato nella Figura 8.13. Qui, un'applicazione audio a 1 Mbit/s (per esempio, una chiamata audio con qualità CD) condivide il link a *1,5* Mbit/s fra R1 e R2 con un'applicazione FTP che sta trasferendo un file da H2 a H4. Nell'Internet best-effort, i pacchetti audio e FTP sono mescolati nella coda in uscita da R1 e (tipicamente) trasmessi nell'ordine primo-arrivato-primo-servito (FIFO, *First-In-FirstOut).* In questo scenario, una raffica di pacchetti dalla sorgente FTP può potenzialmente riempire la coda, provocando eccessivo ritardo o perdita dei pacchetti audio TP a causa della saturazione del buffer in R1.

Come potremmo risolvere questo potenziale problema? Dato che l' applicazione FTP non ha vincoli di tempo, la nostra intuizione potrebbe essere di dare una stretta priorità ai pacchetti audio in R1.

Con una condizione di stretta priorità, un pacchetto audio nel buffer di uscita di R1 dovrebbe sempre essere trasmesso prima di qualsiasi pacchetto FTP che si trovi nello stesso buffer R1. Il link da R1 a R2 apparirebbe come un link a *1,5* Mbit/s dedicato al traffico audio, con il traffico FTP che usa il link da R1 a R2 solo quando non c'è traffico audio accodato al buffer.

Perché R1 possa distinguere fra traffico audio e pacchetti FTP nella sua coda, ciascun pacchetto deve essere comrassegnato come appartenente a una delle due "classi"di traffico. Per quanto ovvio possa sembrare, questo è allora il nostro primo principio alla base della fornitura di garanzie di qualità del servizio:

I principio: La marcatura dei pacchetti permette a un router di distinguere fra pacchetti appartenenti a due diverse classi di traffico.

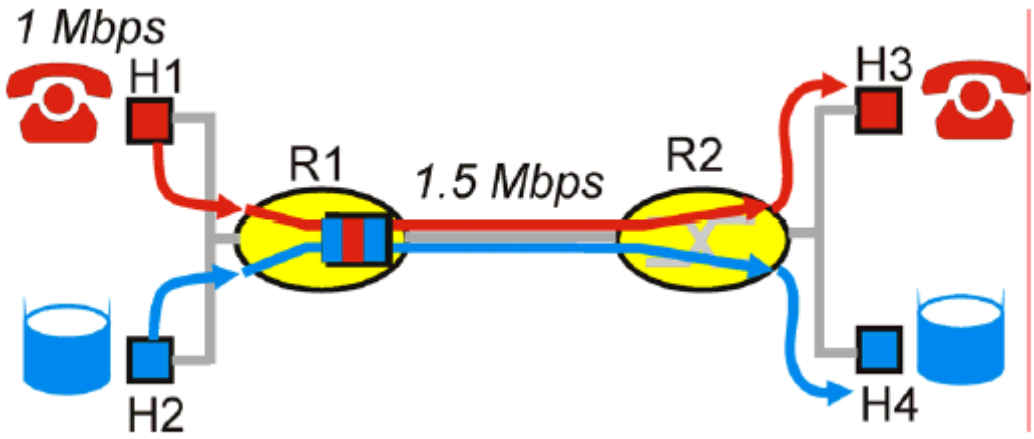


Figura 8.13 – Competizione fra applicazioni audio e FTP

Il scenario: un'applicazione audio a 1 Mbit/s e un trasferimento FTP ad alta priorità

Il nostro secondo scenario è solo leggermente diverso dal primo. Supponiamo che l'utente FTP abbia acquistato il "servizio platino" (cioè, ad alto prezzo) di accesso a Internet dal suo ISP, mentre l'utente audio abbia acquistato un servizio Internet economico, di basso budget, che costa solo una minima frazione del "servizio platino". In questo caso i pacchetti audio dell'utente con contratto economico potrebbero ottenere la priorità sui pacchetti FTP? Assolutamente no. Qui, sembrerebbe più ragionevole distinguere i pacchetti sulla base dell'indirizzo IP del sender.

Più in generale, vedremo che per un router è necessario classificare i pacchetti in base ad alcuni criteri. Quanto sopra allora richiede una piccola modifica al I principio:

II principio: La classificazione dei pacchetti permette a un router di distinguere fra pacchetti apppartenenti a diverse classi di traffico.

La marcatura esplicita è un mezzo con il quale distinguere i pacchetti. Comunque, il contrassegno portato da un pacchetto non può, di per sé, implicare che un pacchetto riceva una data qualità di servizio. La marcatura è un *meccanismo* per distinguere i pacchetti. Il modo in cui un router distingue tra i pacchetti per trattarli in modo diverso è una decisione politica.

III scenario: un'applicazione audio che si comporta in modo scorretto e un trasferimento FTP

Supponiamo ora che in qualche modo il router sappia che deve dare la priorità ai pacchetti dell' applicazione audio a 1 Mbit/s. Poiché la velocità del link in uscita è 1,5 Mbit/s, anche se i pacchetti FTP ricevono una bassa priorità, essi riceveranno, in media, 0,5 Mbit/s del servizio di trasmissione. Ma cosa succede se l'applicazione audio comincia a inviare pacchetti al tasso di 1,5 Mbit/s o più alto (volutamente o a causa di un errore nell' applicazione)? In questo caso, i pacchetti FTP non riceveranno alcun servizio sul link da R1 a R2. Problemi simili accadono se una molteplicità di applicazioni (per esempio, chiamate audio multiple), tutte con la stessa priorità, deve spartirsi la larghezza di banda di un link; un flusso non conforme potrebbe degradare e rovinare le prestazioni degli altri flussi. Idealmente, sarebbe desiderabile qualche grado di isolamento tra i flussi, per proteggere un flusso da un altro che si comporta in modo scorretto. Questo, allora, è un secondo principio alla base della fornitura di garanzie di QoS:

II principio: È desiderabile fornire un grado di isolamento fra i flussi di traffico, in modo che un flusso non subisca gli effetti avversi di un altro flusso che si comporta in modo scorretto.

Se l'applicazione esaminata si comporta in modo scorretto, il meccanismo di sorveglianza prenderà alcune decisioni (per esempio, ritarderà o scarterà i pacchetti che stanno violando i criteri) per far sì che il traffico che entra nella rete sia conforme ai criteri stabiliti.

Il leacky bucket (contenitore bucato) è forse il meccanismo di sorveglianza *(policing)* più ampiamente usato.

Un approccio alternativo per fornire l'isolamento tra i flussi di traffico è che il meccanismo di scheduling dei pacchetti a livello di link assegni esplicitamente a ciascun flusso delle applicazioni una quantità fissa di larghezza di banda. Per esempio, in R1 al flusso audio potrebbe essere assegnato 1 Mbit/s e al flusso FTP potrebbero essere assegnati 0,5 Mbit/s. In questo caso, i flussi audio e FTP vedono un link logico con capacità di 1,0 e 0*,5* Mbit/s, rispettivamente. Con un preciso controllo della larghezza di banda allocata a livello del link, un flusso può impiegare solo la larghezza di banda che gli è stata assegnata; in particolare, non può utilizzare larghezza di banda che al momento non è usata da altre applicazioni. Per esempio, se il flusso audio diventa silente (per esempio, se lo speaker prende una pausa e non genera pacchetti audio), sul link da R1 a R2 il flusso FTP non potrà ancora superare la velocità di trasmissione di 0*,*5 Mbit/s, anche se la larghezza di banda di 1 Mbi/ls riservata al flusso audio al momento non è utilizzata. È quindi desiderabile usare la larghezza di banda nel modo più efficientemente possibile:

III principio: Mentre si fornisce l'isolamento tra i flussi, è desiderabile usare le risorse il più efficientemente possibile.

IV scenario: due applicazioni audio a 1 Mbit/s su un link a sovraccarico 1,5 Mbit/s

In questo ultimo scenario due connessioni audio a 1 Mbit/s trasmettono i loro pacchetti su un link a 1,5 Mbit/s, come mostrato nella figura 8.14. la velocità di dati combinati dei due flussi eccede la capacità del link.

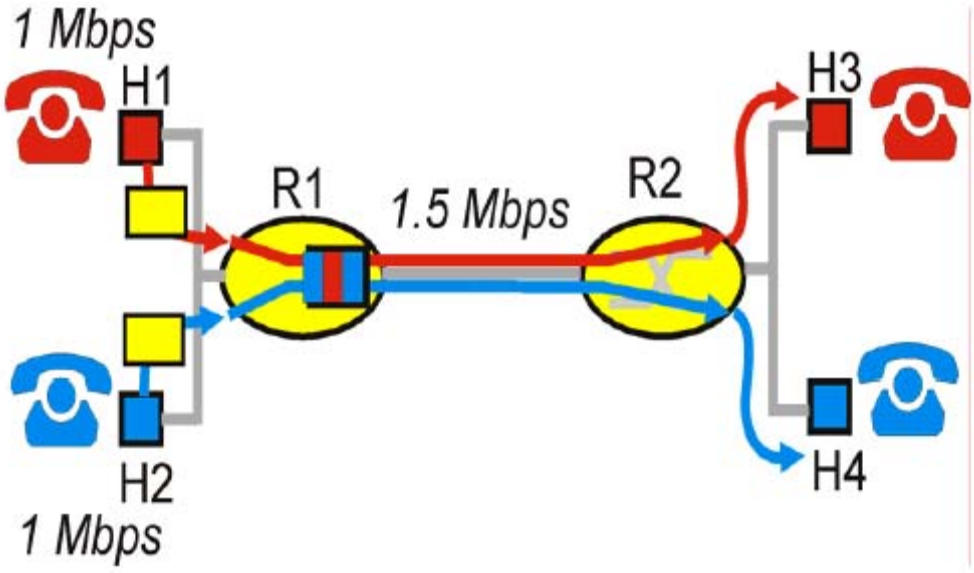


Figura 8.14 – due applicazioni in competizione sovraccaricano il link fra R1 e R2

IV Principio 4: Utilizzare politica di ammissione di chiamata (Call Admission): un'applicazione dichiara cio' che serve e puo' essere accettata o rifiutata (se non c'e'abbastanza banda)

Test

1	Il primo principio afferma che: Marcare i pacchetti (Marking) per permettere ai router di distinguere fra classi di servizio (e avere nuovi router che lo facciano)	SI
		NO
2	La marcatura esplicita è un mezzo con il quale distinguere i pacchetti	SI
		NO
3	L'Internet attuale fornisce un servizio besteffort a tutte le sue applicazioni, vale a dire, non fa alcuna promessa sulla qualità del servizio (QoS) che un applicazione riceve	SI
		NO

TECNICO DELLE RETI

MODULO 9

6° parte

Scuola Radio Elettra®

Unità di Apprendimento

N. 8.6

Titolo: MECCANISMI DI REGISTRAZIONE E REGOLAZIONE

Obiettivo: analisi dei meccanismi di scheduling e policing

Meccanismi di registrazione

Ricordiamo che i pacchetti appartenenti a diversi flussi della rete sono multiplati insieme e accodati per la trasmissione ai buffer di uscita associati con un link. Il modo in cui i pacchetti nelle code sono selezionati per la trasmissione sul link è detto modalità di scheduling del link. Abbiamo visto che la modalità di scheduling nel link ha un ruolo importante nel fornire le garanzie di QoS. Consideriamo ora in dettaglio alcuni delle più importanti modalità di scheduling del link, ossia dei criteri di scelta dei pacchetti da spedire.

First – In – First - Out (FIFO)

La Figura 8.21 mostra l'astrazione del modello di coda per la modalità di scheduling nel link *First-In-First-Out* (FIFO, primo entrato-primo uscito). I pacchetti che arrivano alla coda di uscita del link, se il link è occupato dalla trasmissione di un altro pacchetto, vengono accodati all'uscita. Se lo spazio nel buffer non è sufficiente per contenere il pacchetto in arrivo, la politica di scarto del pacchetto della coda determina se il pacchetto deve essere scartato (perso) o se altri pacchetti possono essere rimossi dalla coda per fare spazio al pacchetto in arrivo. Quando un pacchetto è completamente trasmesso su un link in uscita (cioè, ha ricevuto il servizio) viene rimosso dalla coda.

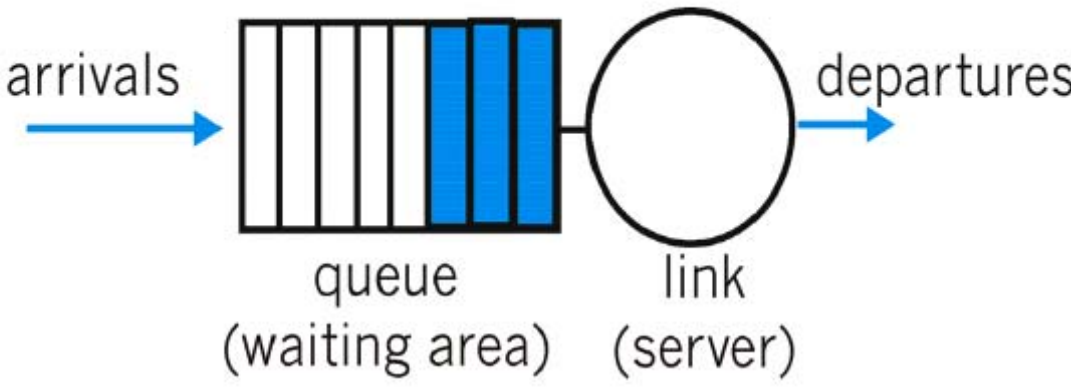


Figura 8 – 21 Astrazione di accoramento FIFO

La politica di scheduling FIFO (detta anche *First-Come-First-Served,* FCFS) seleziona i pacchetti per la trasmissione sul link nello stesso ordine in cui essi arrivano alla coda in uscita del link. Sperimentiamo spesso l'accodamento FIFO per esempio alle fermate dei bus o in altri centri di servizio, dove i clienti che arrivano si uniscono alla fine di una singola linea di attesa, rimanendo in ordine, per essere serviti solo quando raggiungono l'inizio.

La Figura 8.22 mostra un esempio di coda FIFO in funzione. I pacchetti in arrivo sono indicati da frecce numerate al di sopra della linea superiore dei tempi, con i numeri che indicano il loro ordine di arrivo. La partenza dei pacchetti è individuata sotto la linea inferiore. Poiché l'ordine di scheduling è FIFO, i pacchetti partono nello stesso ordine in cui arrivano.

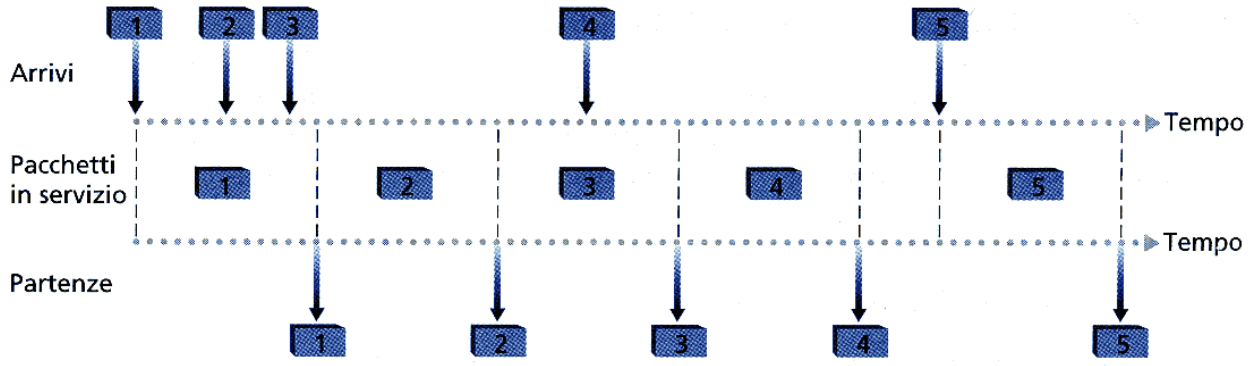


Figura 8.22 - La coda FIFO in funzione

FIFO: in ordine di arrivo alla coda; pacchetti trovano buffer pieno sono scartati

Accodamento prioritario

Nell'accodamento prioritario, i pacchetti che arrivano al link di uscita sono classificati all'interno di due o più classi di priorità nella coda di uscita, come mostrato nella Figura 8.23. Come abbiamo visto nella sezione precedente, la classe di priorità di un pacchetto può dipendere da una marcatura esplicita che è riportata nell'intestazione del pacchetto (per esempio, il valore dei bit tipo di servizio, ToS, in un pacchetto IPv4), dal suo indirizzo

IP di sorgente o di destinazione, dal numero di porta di destinazione o da altri criteri. Tipicamente ciascuna classe di priorità ha la sua coda. Nella scelta del pacchetto da trasmettere, la modalità di accodamento prioritario trasmetterà un pacchetto della più alta classe di priorità la cui coda non sia vuota (vale a dire, che contiene un pacchetto in attesa della trasmissione). La scelta fra i pacchetti *di una stessa classe di priorità* di solito è effettuata in modo FIFO.

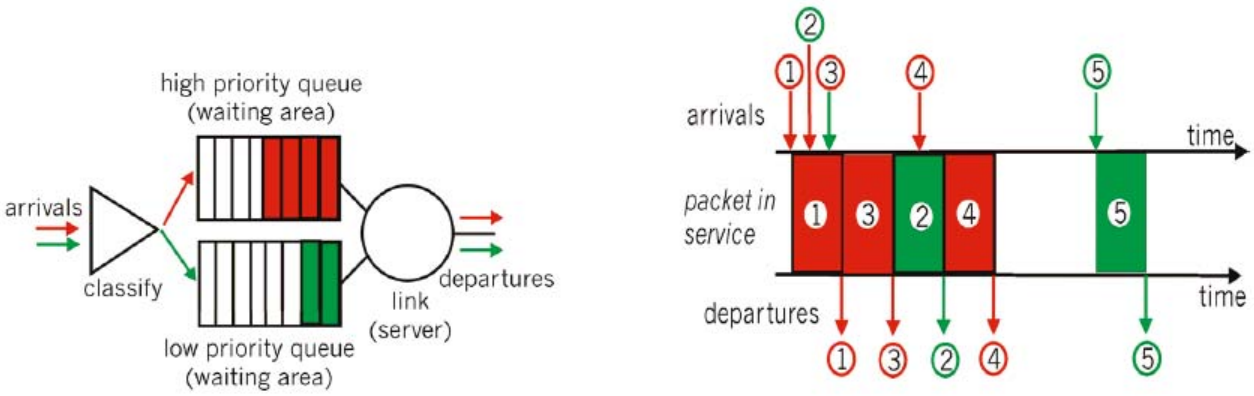


Figura 8.23 – Accoramento prioritario

Priority Queuing: pacchetti hanno diverse priorità
Cisono piu'code (una per ciascuna classe di priorità
Trasmetti prima pacchetti di priorità più alta

Round robin e accodamento equo pesato (WFQ)

Nella modalità di accodamento round robin *(round robin queuing discipline),* i pacchetti sono ancora smistati in classi, come con l'accodamento prioritario. Comunque, invece di avere una stretta priorità assegnata alle classi, la modalità round robin altema i servizi fra le classi. Nella forma più semplice di scheduling round robin, è trasmesso un pacchetto di classe 1, seguito da uno di classe 2, seguito da uno di classe 1, poi da uno di classe 2, e così via. Una modalità di accodamento round robin che non dissipa il lavoro *(work-conserving)* non permetterà mai al link di restare inattivo se ci sono pacchetti (di qualunque classe) accodati per la trasmissione.

Una modalità round robin non dissipativa che cerca un pacchetto di una data classe ma non trova nulla cercherà immediatamente nella classe successiva della sequenza round robin.

La Figura 8.24 illustra le operazioni di una coda round robin a due classi. In questo esempio, i pacchetti 1, 2 e 4 appartengono alla prima classe, e i pacchetti 3 e *5* appartengono alla seconda classe. Il pacchetto i viene trasmesso appena dopo il suo arrivo alla coda in uscita. I pacchetti 2 e 3 arrivano durante la trasmissione del pacchetto i e quindi si accodano per la trasmissione. Al termine della trasmissione del pacchetto 1, il meccanismo di scheduling del link cerca un pacchetto di classe 2 e quindi trasmette il pacchetto 3. Dopo la trasmissione del pacchetto 3, il meccanismo di. scheduling guarda per un pacchetto di classe i e trasmette allora il pacchetto 2. Dopo la trasmissione di questo pacchetto rimane accodato il solo pacchetto 4, che viene trasmesso immediatamente dopo il pacchetto 2.

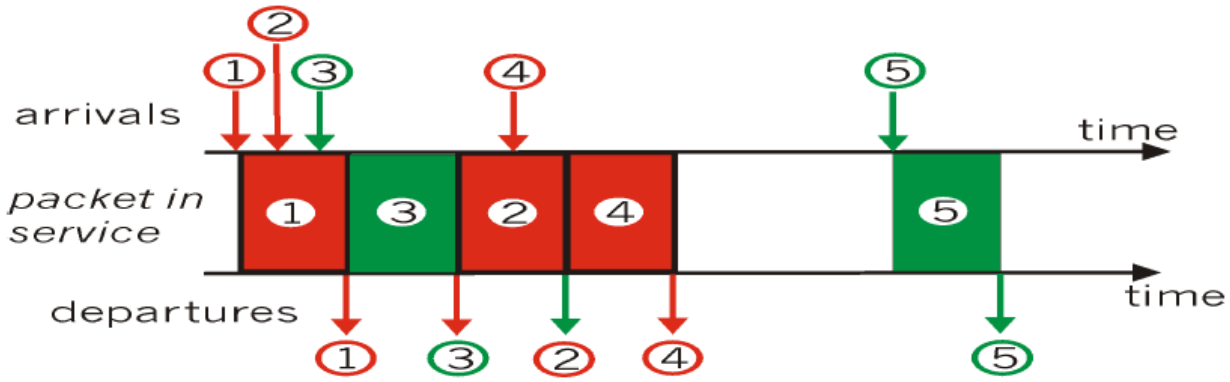


Figura 8.24 – Operazioni della coda round robin a due classi

Un'astrazione generalizzata dell'accodamento round robin che ha trovato impiego considerevole nelle architetture per la QoS è la cosiddetta modalità di accoramento equo pesato (WFO, Weigheted Fair Queuing). I pacchetti in arrivo sono ancora classificati e vengono accodati nella loro area di attesa della classe appropriata. Come nella modalità round robin, la modalità WFQ servirà ancora le classi in modo ciclico: prima la classe 1, poi la 2, poi la 3 e quindi (assumendo l'esistenza di tre classi) ripeterà lo schema di servizio. Anche la WFQ è una modalità di lavoro work – conservino, e quindi si sposterà immediatamente a servire la classe successiva quando troverà la coda di una clessae vuota.

La WFQ differisce dal round robin in quanto ogni classe può ricevere una quantità di servizio differenziata in qualsiasi intervallo di tempo.

Weighted Fair Queuing: generalizza round robin fornendo un servizio differenziato ad ogni classe per un dato periodo di tempo

Regolazione: il leaky bucket

Possiamo identificare tre importanti criteri di sorveglianza, ciascuno che differisce dall'altro in accordo alla scala dei tempi su cui il flusso di pacchetti è sorvegliato:

- *Velocità media.* La rete può desiderare di limitare la velocità media a lungo termine (pacchetti per intervallo di tempo) a cui un flusso di pacchetti può esservi spedito. Un argomento cruciale qui è l'intervallo di tempo su cui la velocità media sarà sorvegliata. Un flusso il cui la velocità media è limitata a 100 pacchetti per secondo è più vincolato rispetto a una sorgente che ha come limite 6000 pacchetti al minuto, anche se su un intervallo di tempo sufficientemente lungo entrambi hanno la stessa velocità media. Per esempio, l'ultima limitazione permetterebbe l'invio di un flusso di 1000 pacchetti nell'intervallo di un secondo (con il vincolo che la velocità sia inferiore ai 6000 pacchetti in ogni intervallo di i minuto contenente questi 1000 pacchetti), mentre il vincolo precedente impedirà questo tipo di comportamento.
- *Velocità di picco.* Mentre il vincolo sulla velocità media limita la quantità di traffico che può essere inviato nella rete in un periodo relativamente lungo di tempo, un vincolo sulla velocità di picco limita il massimo numero di pacchetti che possono essere inviati in un breve periodo di tempo. Usando l'esempio precedente, la rete può ammettere un flusso a un tasso medio di 6000 pacchetti al minuto, mentre limita la velocità di picco del flusso a 1500 pacchetti al secondo.
- *Dimensione della raffica (burst).* La rete può anche desiderare di limitare il numero massimo di pacchetti (la "raffica di pacchetti") che possono essere inviati nella rete in un periodo di tempo molto breve. Al limite, quando la lunghezza dell'intervallo tende a zero, la dimensione della raffica limita il numero di pacchetti che possono essere istantaneamente inviati nella rete. Anche se è fisicamente impossibile inviare istantaneamente in rete più pacchetti (dopo tutto, ciascun link ha un limite fisico alla velocità di trasmissione che non può essere superato), l'astrazione di una massima dimensione della raffica è utile.

Il meccanismo *leaky bucket* (contenitore o secchio bucato) è un'astrazione che può essere usata per caratterizzare questi limiti di sorveglianza. Come mostrato nella Figura 8.25, un leaky bucket consiste di un contenitore che può contenere fino a *b token* (gettoni). I token sono aggiunti al contenitore come segue. Nuovi token, che potenzialmente potrebbero essere aggiunti al contenitore, sono sempre generati a una velocità di *r* token al secondo. (Qui assumeremo per semplicità che l'unità di tempo sia il secondo.) Se il contenitore contiene meno di *b* token al momento della generazione di un token, il token appena generato è aggiunto al contenitore; altrimenti viene ignorato, e il contenitore rimane con *b* token (cioè pieno).

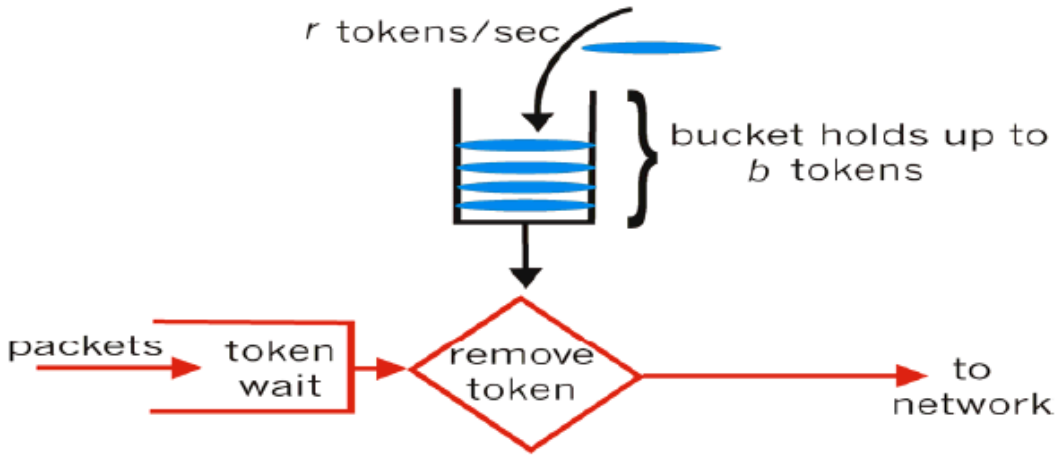


Figura 8.25 – il policer leacky bucket

Consideriamo ora come leaky bucket possa essere usato per sorvegliare un flusso di pacchetti. Supponete che prima che un pacchetto sia inviato nella rete, esso debba rimuovere un token dal contenitore. Se il contenitore dei token è vuoto, il pacchetto deve aspettare un token. (Un'alternativa per il pacchetto è quella di essere scartato, sebbene qui non la prendiamo in considerazione.) Consideriamo ora come questa politica influisca sul flusso di traffico. Poiché ci possono essere al massimo *b* token nel contenitore, la massima dimensione della raffica per un flusso sorvegliato con leaky bucket è di *b* pacchetti. Inoltre, poiché la velocità di generazione dei token è *r,* il numero massimo di pacchetti che possono entrare nella rete in qualsiasi intervallo di tempo di lunghezza t *è rt* + *b.* Quindi, la velocità di generazione dei token, *r,* serve a limitare la velocità media a lungo termine a cui i pacchetti possono entrare nella rete. È anche possibile usare il leaky

bucket (nello specifico, due leaky bucket in serie) per regolare la velocità di picco del flusso oltre alla velocità media a lungo termine.

Leaky bucket e accodamento equo pesato possono essere combinati.

Test

1	La politica di scheduling FIFOseleziona i pacchetti per la trasmissione sul link in ordine inverso da come essi arrivano alla coda in uscita del link.	SI
		NO
2	Nella scelta del pacchetto da trasmettere, la modalità di accodamento prioritario trasmette un pacchetto della più alta classe di priorità la cui coda non sia vuota	SI
		NO
3	I criteri di sorveglianza, sono tre: - *Velocità media.* - *Velocità di picco.* - *Dimensione della raffica (burst).*	SI
		NO

TECNICO DELLE RETI

MODULO 9

7° parte

Scuola Radio Elettra®

Unità di Apprendimento
N. 8.7
Titolo: Servizi integrati
Obiettivo: analisi di architetture per fornire la qualità del servizio in Internet

Nelle unità precedenti abbiamo identificato sia i principi sia i meccanismi usati per fornire la qualità del servizio in Internet. In questa unità consideriamo come queste idee sono utilizzate in una particolare architettura per fornire la qualità di servizio in Internet: la cosiddetta architettura Intserv *(Integrated Services)* di Internet. L'intserv è un progetto sviluppato per fornire garanzie di qualità di servizio personalizzate a sessioni applicative individuali. Due caratteristiche basilari costituiscono il nucleo dell'architettura Intserv:

- *Risorse riservate.* A un router è richiesto di conoscere la quantità delle sue risorse (buffer, larghezza di banda del link) che sono già state riservate alle sessioni in corso.

- *Impostazione della chiamata.* Una sessione che richiede garanzie di QoS deve prima essere in grado di prenotare risorse sufficienti a ciascun router della rete sul suo percorso da sorgente a destinazione, per assicurare che i suo requisiti di QoS end-to-end siano soddisfatti. Questo processo di impostazione *(setup)* della chiamata (detto anche ammissione della chiamata) richiede la partecipazione di ciascun router sul percorso. Ogni router deve determinare le risorse locali richieste dalla sessione, considerare la quantità delle sue risorse che sono già impegnate da altre sessioni in corso, e determinare se ha risorse sufficienti per soddisfare i requisiti di QoS della sessione a questo router, senza violare le garanzie di QoS date a una sessione precedentemente ammessa.

La Figura 8.26 illustra il processo di impostazione della chiamata. Consideriamo i passi coinvolti nell'ammissione della chiamata più in dettaglio:

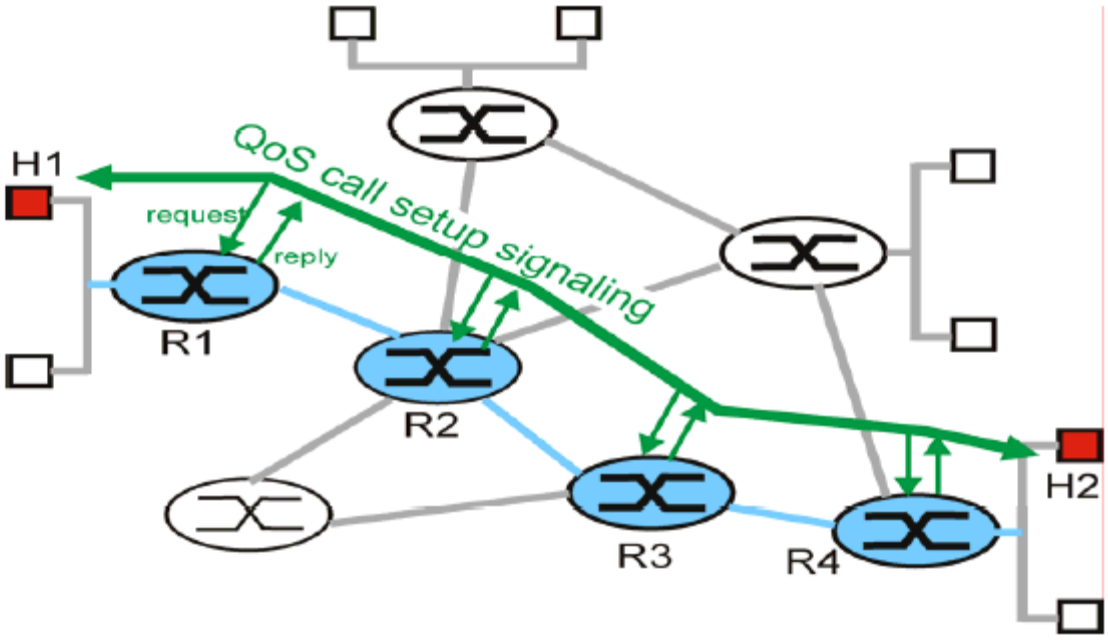


Figura 8.26 – Il processo di instaurazione della chiamata

1. *Caratterizzazione del traffico e specifica della QoS desiderata.* Perché un router sia in grado di determinare se le sue risorse sono o no sufficienti a soddisfare i requisiti di QoS di una sessione, la sessione deve prima dichiarare le sue necessità di QoS, come anche caratterizzare il traffico che invierà nella rete, e per il quale richiede le garanzie di QoS. Nell'architettura Intserv, il cosiddetto Rspec (R sta per *Reserved,* prenotato) definisce la specifica QoS che è stata richiesta da una connessione; il cosiddetto Tspec (T sta per traffico) caratterizza il traffico che il sender si appresta a inviare nella rete, o che il receiver riceverà dalla rete. La forma specifica di Rspec e Tspec varierà in funzione del servizio richiesto.

2. *Segnalazione per l'impostazione della chiamata.* Tspec e Rspec di una sessione devono essere trasportati ai router in cui saranno prenotate le risorse per la sessione. In Internet, il protocollo RSVP, trattato in dettaglio nel paragrafo seguente, è attualmente il protocollo scelto per la segnalazione.

3. *Ammissione della chiamata per elemento.* Quando un router riceve i Tspec e Rspec per una sessione che richiede garanzie di QoS, esso può determinare se può o non può ammettere la chiamata. Questa decisione di ammissione della chiamata dipenderà dalla specifica del traffico, dal tipo di servizio richiesto e dall'impegno delle risorse già

assegnate dal router alle sessioni in corso. L' architettura Intserv definisce due maggiori classi di servizi: servizio garantito e servizio di carico controllato.

Servizi integrati

- Un architettura per fornire QoSgarantita in reti IP per sessioni individuali
- si basa sulla assegnazione delle risorse; routers devono mantenere informazioni di stato, registrare le risorse assegnate e decide su questa base a riguardo di nuove richieste di connessione.

Qualità del servizio garantita

La specifica di servizio garantito, fornisce precisi limiti ai ritardi di coda che un pacchetto può sperimentare in un router. Mentre i dettagli alla base della garanzia del servizio sono piuttosto complicati, l'idea base è davvero molto semplice. A una prima approssimazione, la caratterizzazione del traffico di una sorgente è data da un leaky bucket con i parametri *(r,b)* e il servizio richiesto è caratterizzato dalla velocità di trasmissione, *R,* a cui i pacchetti saranno trasmessi. In sostanza, una sessione che richiede un servizio garantito sta chiedendo che ai bit nel suo pacchetto sia garantito l'inoltro alla velocità di *R* bits/s. Dato che il traffico è specificato usando una caratterizzazione leaky bucket, e che è stata richiesta una velocità garantita di *R,* è anche possibile limitare il massimo ritardo di coda al router. Ricordiamo che con la caratterizzazione leaky bucket del traffico, la quantità di traffico (in bit) generata in un intervallo di lunghezza t è limitata da *rt + b.* Ricordiamo anche che quando una sorgente leaky bucket è alimentata in una coda che garantisce che il traffico nella coda sarà servito almeno alla velocità di *R* bit al secondo, il massimo ritardo di coda sperimentato da qualsiasi pacchetto sarà limitato a *b/R,* finché *R* è più grande di *r.* Il reale limite del ritardo garantito nella definizione di servizio garantito è leggermente più complicato, a causa dell'effetto di pacchettizzazione (il semplice limite *b/R* assume che i dati abbiano forma simile a un flusso fluido piuttosto che a pacchetti discreti), del fatto che il processo di arrivo del traffico è soggetto alla limitazione della velocità di picco al link di ingresso (il semplice limite *b/R* assume che una raffica di *b* bit possa arrivare in tempo zero) e di altre possibili variazioni nel tempo di trasmissione di un pacchetto.

QoS garantita

- □ fornisce una garanzia rigida sul ritardo di accoramento ai routers;
- □ intesa per applicazioni con vincoli real-time stretti che sono altamente sensibili al valore atteso ed alla varianza del ritardo end-to-end

Servizio di carico controllato

Una sessione che riceve un servizio di carico controllato riceverà un una qualità di servizio che si avvicina molto alla QoS che lo stesso flusso riceverebbe da un elemento di rete scarico. In altre parole, la sessione potrebbe considerare che una percentuale molto alta dei suoi pacchetti passerà con successo attraverso i router senza essere scartata e con ritardi di coda prossimi allo zero. Cosa interessante, il servizio di carico controllato non dà garanzie quantitative sulle prestazioni: esso non specifica che cosa costituisce una percentuale molto alta di pacchetti né quale qualità dei servizio approssima strettamente quella di un elemento di rete scarico. Il target del servizio di carico controllato sono le applicazioni multimediali in tempo reale sviluppate per l'Internet attuale. Come abbiamo visto, queste applicazioni funzionano molto bene quando la rete è scarica, ma le loro prestazioni degradano rapidamente quando la rete diventa più carica.

Carico Controllato

- □ fornisce QoS simile a quella fornita da internet quando non congestionata
- □ intesa per applicazioni real-time che operano bene nelle reti IP odierne quando non congestionate

Test

1	L’intserv è un progetto sviluppato per fornire garanzie di qualità di servizio personalizzate a sessioni applicative individuali.	SI
		NO
2	Una sessione che riceve un servizio di carico controllato riceverà un una qualità di servizio che si discosta dalla QoS	SI
		NO
3	La specifica di servizio garantito, fornisce precisi limiti ai ritardi di coda che un pacchetto può sperimentare in un router	SI
		NO

TECNICO DELLE RETI

MODULO 9

8° parte

Scuola Radio Elettra®

Unità di Apprendimento
N. 8.8
Titolo: RSVP
Obiettivo: Analisi del protocollo RVSP

Il protocollo RSVP permette alle applicazioni di prenotare larghezza di banda per i loro flussi di dati. È usato da un host, a nome di un flusso di dati di un'applicazione, per richiedere alla rete una quantità specifica di larghezza di banda. L'RSVP è anche usato dai router per inoltrare le richieste di prenotazione della larghezza di banda. Per implementare l'RSVP, il software RSVP deve essere presente in receiver, sender e router. Le due principali caratteristiche dell'RSVP sono:

1. Fornisce la prenotazione della larghezza di banda negli alberi multicast (gli unicast sono gestiti come caso particolare dei multicast).

2. È orientato al receiver, cioè, il receiver di un flusso di dati inizia e mantiene la prenotazione delle risorse usate per quel flusso.

- □ Int-Serv specifica solo il framework per la riservazione delle risorse di rete
- □ Occorre un protocollo usato dai routers per trasportare le informazione sulla riservazione delle risorse
- □ Resource Reservation Protocol
- è il protocollo usato per trasportare e coordinare le informazioni di setup delle chiamate
- progettato per adattarsi anche a riservazione multicast
- ricevitore prende l'iniziativa(più facile per multicast)
- fornisce supporto per unire flussi destinati ad un receiver da sorgenti multiple in un singolo gruppo di multicast

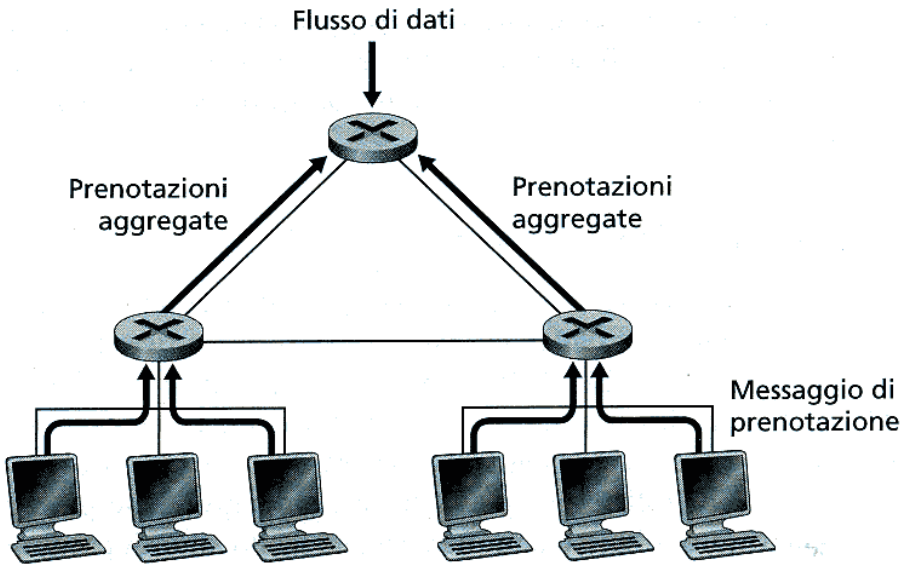


Figura : 8.27 RVSP orientato al multicast e al receiver

Queste due caratteristiche sono illustrate nella Figura 8.27. Lo schema mostra un albero multicast con il flusso di dati dal vertice dell'albero verso gli host nella parte inferiore. Sebbene i dati originino dal sender, i messaggi di prenotazione originano dai receiver. Quando un router inoltra un messaggio di prenotazione a monte verso il sender, il router può integrare il messaggio di prenotazione con altri messaggi che arrivano dai nodi a valle.

Prima di trattare l'RSVP a grandi linee, dobbiamo considerare la nozione di sessione. Una sessione può essere costituita da più flussi di dati multicast. Ciascun sender in una sessione è la sorgente di uno o più flussi di dati; per esempio, un sender può essere la sorgente di un flusso di dati video e di un flusso di dati audio. Ciascun flusso di dati in una sessione ha lo stesso indirizzo multicast. Per mantenere concreta la discussione, assumiamo che router e host identifichino la sessione a cui appartiene un pacchetto attraverso l'indirizzo multicast del pacchetto. Quest' assunzione è in qualche modo restrittiva; la reale specifica di RSVP consente metodi più generici per identificare una sessione. All'interno di una sessione, anche il flusso di dati cui appartiene un pacchetto richiede di essere identificato. Questo può essere fatto, per esempio, con il campo di identificazione del flusso in IPv6.

Che cosa non è RSVP

Osserviamo che lo standard RSVP [RFC 2205] non specifica come la rete fornisca la larghezza di banda prenotata ai flussi di dati. È solo un protocollo che permette alle applicazioni di prenotare la necessaria larghezza di banda del link. Una volta effettuata la prenotazione, è compito dei router in Internet fornire veramente la larghezza di banda ai flussi di dati. È anche importante capire che l'RSVP non è un protocollo di instradamento: esso non determina i link in cui deve essere fatta la prenotazione. Invece, per determinare i router per il flusso esso dipende da un sottostante protocollo di instradamento (unicast o multicast). Quando i percorsi sono stabiliti, l'RSVP può prenotare la larghezza di banda nei link lungo i percorsi. Quando le prenotazioni sono operative, gli scheduler dei pacchetti dei router devono davvero fornire la larghezza di banda ai flussi dei dati. Quindi, l'RSVP è solo una parte, benché una parte importante, del puzzle della garanzia di QoS.

All'RSVP qualche volta ci si riferisce come a un protocollo di segnalazione. Con questo si vuole indicare che l'RSVP è un protocollo che permette agli host di stabilire e mantenere le prenotazioni per i flussi di dati.

Receiver eterogenei

Alcuni receiver possono ricevere un flusso a 28,8 kbit/s, altri a 128 kbit/s, e altri ancora a 10 Mbit/s o più alti. Questa eterogeneità dei receiver pone un'interessante questione. Se un sender invia multicast un video a un gruppo di receiver eterogenei, esso dovrebbe codificare il video a 28,8 kbit/s per bassa qualità, a 128 kbit/s per qualità media o a 10 Mbit/s per l'alta qualità? Se il video è codificato a 10 Mbit/s, potrà essere visto dai soli utenti con l'accesso a 10 Mbit/s. D'altra parte, se il video è codificato a 28,8 kbitls, gli utenti con 10 Mbit/s riceveranno immagini di bassa qualità pur sapendo che possono vedere di meglio. Per risolvere questo dilemma si suggerisce spesso di codificare audio e video in strati. Per esempio, un video potrebbe essere codificato in due strati: uno strato base e uno strato aggiuntivo. Lo strato base potrebbe avere una velocità di 20 kbit/s mentre quello aggiuntivo una velocità di 100 kbit/s; in questo modo i receiver con accesso a 28,8 kbit/s potrebbero ricevere le immagini dello strato base a bassa qualità mentre quelli con accesso a 128 kbit/s potrebbero ricevere entrambi gli strati per costruire un'immagine di alta qualità.

Notate che il sender non ha la necessità di conoscere la velocità di ricezione di tutti i receiver. Deve solo conoscere la velocità massima di tutti i suoi receiver. Il sender codifica il video o l' audio in strati multipli e invia tutti gli strati fino alla velocità massima nell' albero multicast. I receiver estraggono gli strati che sono appropriati per le proprie la velocità di ricezione. Per evitare lo spreco della larghezza di banda nei link della rete, i receiver eterogenei devono comunicare alla rete le velocità che possono gestire.

Test

1	Il protocollo RSVP permette alle applicazioni di prenotare larghezza di banda per i loro flussi di dati	SI
		NO
2	La caratteristica fondamentale dell'RSVP è che esso fornisce la prenotazione della larghezza di banda negli alberi multicast.	SI
		NO
3	Resource Reservation Protocol - è il protocollo usato per trasportare e coordinare le informazioni di setup delle chiamate - progettato per adattarsi anche a riservazione multicast - ricevitore prende l'iniziativa(più facile per multicast) - fornisce supporto per unire flussi destinati ad un receiver da sorgenti multiple in un singolo gruppo di multicast	SI
		NO

TECNICO DELLE RETI

MODULO 9

9° parte

Scuola Radio Elettra®

Unità di Apprendimento
N. 8.9
Titolo: Servizi differenziati
Obiettivo: Analisi dei servizi differenziati

Abbiamo visto come l'RSVP può essere usato per prenotare risorse *per-flusso* ai router nella rete. La capacità di richiedere e prenotare risorse per- flusso, a sua volta, rende possibile per la struttura Intserv fornire garanzie di qualità di servizio a flussi individuali. Al procedere del lavoro su Intserv e RSVP, comunque, i ricercatori coinvolti in questi sforzi hanno cominciato a scoprire alcune delle difficoltà associate con il modello Intserv e con la prenotazione delle risorse per-flusso:

- *Problema di scala.* La prenotazione delle risorse per-flusso usando l'RSVP implica la necessità per un router di elaborare le prenotazioni di risorse e di mantenere uno stato per-flusso per *ogni* flusso che lo attraversa. L'elaborazione delle prenotazioni per-flusso nei router della backbone può quindi portare a notevoli sovraccarichi nelle grandi reti.

- *Modelli di servizio flessibili.* La struttura Intserv supporta un piccolo numero di classi di servizio prespecificate. Questo particolare set di classi di servizio non permette la distinzione di classi più qualitative o la definizione relativa dei servizi

Queste considerazioni hanno portato alla recente cosiddetta attività "Diffserv" *(Differentiated Services,* servizi differenziati) all'interno dell'Internet Engineering Task Force. Il gruppo di lavoro Diffserv sta sviluppando un'architettura per fornire una differenziazione fra servizi *scalabile* e *flessibile:* vale a dire, l'abilità di gestire differenti "classi" di traffico in diversi modi all'interno di Internet. La necessità di un servizio di *scala* nasce dal fatto che centinaia di migliaia di flussi di traffico simultanei da sorgente a destinazione si possono presentare a un router della backbone di Internet. Presto vedremo che questa necessità è soddisfatta collocando semplici funzionalità nella sezione interna della rete, con operazioni di controllo più complesse implementate alle sue estremità.

Diffserv: inteso per rispondere alle difficolta di Intserv e RSVP;

- **Scalabilità**: mantenere lo stato ai routers in reti ad alta velocità è difficile a causa del grande numero di flussi
- **Modello di Servizio Flessibile**: Intserv ha solo due classi, si desidera fornire piùclassi di servizio; ex: distinzioni 'relative'di servizio (Platinum, Gold, Silver, …)
- **Segnalazione piùsemplice**: (rispetto RSVP) molte applicazioni ed utenti possono desiderare di specificare una nozione piùqualitativa di QoS

Per impostare la struttura per la definizione dei componenti dell'architettura di un modello di servizi differenziati, cominciamo con la semplice rete illustrata nella Figura 8.28.

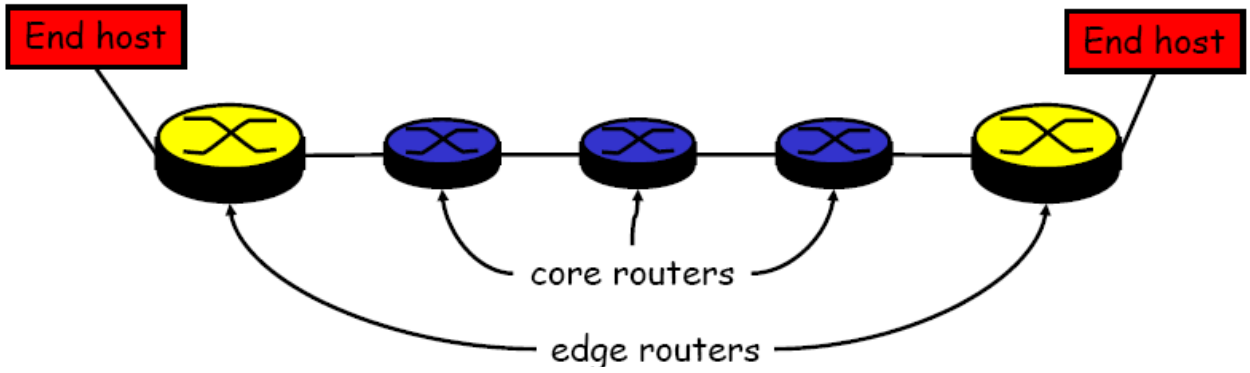


Figura 8.28 – Un semplice esempio di rete Diffserv

Nel seguito, descriveremo un possibile uso dei componenti Diffserv. Molte altre varianti sono possibili, il nostro obiettivo qui è di fornire un'introduzione agli aspetti chiave dei servizi differenziati, piuttosto che di descrivere il modello architetturale in dettagli esaustivi. L' architettura dei servizi differenziati è costituita da due insiemi di elementi funzionali:

- *Funzioni di estremità:* classificazione dei pacchetti e condizionamento del traffico. Alla "estremità" di ingresso della rete (cioè, a un host Diffserv-compatibile che genera traffico o al primo router Diffserv-compatibile attraverso il quale passa il traffico), i pacchetti in arrivo sono contrassegnati. Più specificamente, il campo servizio differenziato (DS, *Dfferentiated Service)* dell'intestazione del pacchetto è posto a un certo valore. ve identifica la classe di traffico a cui appartiene. Diverse classi di traffico riceveranno differenti servizi nella sezione interna della rete. Dopo essere stato

marcato, un pacchetto può essere inoltrato immediatamente nella rete, inviato nella rete dopo un certo ritardo, o essere scartato.

- *Funzioni della sezione interna: inoltro.* Quando un pacchetto marcato DS arriva a un router Diffserv-compatibile, il pacchetto è inoltrato al suo prossimo hop in accordo al cosiddetto comportamento per-hop associato alla classe del pacchetto. Un principio cruciale dell'architettura Diffserv è che il comportamento per-hop di un router sarà basato solo sul marchio del pacchetto, cioè, sulla classe di traffico a cui il pacchetto appartiene.

Un'analogia può tornare utile. In un evento sociale su scala molto ampia (per esempio, un grande congresso pubblico, una grande discoteca, un concerto, una partita di calcio), le persone che entrano per partecipare all'evento ricevono un "pass" di un tipo o di un altro. Ci sono i pass VIP per i Very Importarit People; ci sono pass over-21 per persone che, hanno 21 anni o più (per esempio, se vengono serviti alcolici); ci sono pass backstage (dietro la quinte) per i concerti; ci sono pass stampa per i reporter; ci sono pass comuni per le persone comuni. Questi pass sono tipicamente distribuiti all'ingresso dell'evento, cioè, alla sua "estremità". È qui, all'estremità, il luogo in cui si eseguono le intense operazioni computazionali, come il pagare l'ingresso, il controllo dell'appropriato tipo di invito e la verifica dell'identità dell'invitato. Inoltre, potrebbe esserci un limite al numero delle persone di un dato tipo cui è permesso di partecipare all'evento. Se esiste questo limite, le persone possono dover aspettare prima di entrare alla manifestazione. Una volta entrata, un pass permette a una persona di ricevere un servizio differenziato in diversi luoghi dell'evento: un VIP può avere drink gratuiti, un tavolo migliore, pranzo offerto, entrare in stanze esclusive e un servizio adulante. All' opposto, una persona comune è esclusa da certe aree, paga per i drink e riceve solo un servizio standard. In entrambi i casi, il servizio ricevuto all'interno dell'evento dipende solo dal tipo di pass posseduto. Inoltre, tutte le persone di una classe sono trattate in modo simile.

Classificazione e condizionamento del traffico

Nell'architettura dei servizi differenziati, la marcatura di un pacchetto *è* portata all'interno del campo DS nell'intestazione del pacchetto di IPv4 o IPv6. La definizione del campo DS si intende sostitutiva delle definizioni precedenti dei campi tipo di servizio di IPv4 e classe di traffico di IPv6 La struttura di questo campo a otto bit è illustrata nella Figura 8*.29*

Figura 8.29 – Struttura del campo DS nell'intestazione IPv4 e IPv6

Il sottocampo codice di servizio differenziato (DSCP, *Differentiated Service Code Point)* a sei bit determina il cosiddetto comportamento per-hop che il pacchetto riceverà all'interno della rete. Il sottocampo CU a due bit del campo DS attualmente non è usato. Per i nostri scopi attuali, dobbiamo solo notare che la marcatura di un pacchetto, il suo "code point" nella terminologia Diffserv. è inserito nel campo Diffserv a otto bit. Come notato sopra, un pacchetto è contrassegnato (marcato) assegnando il suo valore del campo Diffserv alle estremità della rete. Questo può avvenire sia in un host Diffserv-compatibile sia al primo punto in cui un pacchetto incontra un router Diffserv-compatibile. Assumiamo che la marcatura avvenga a un router di estremità che è direttamente collegato a un sender.

Il ruolo della funzione di misura (delle proprietà temporali), mostrato nella Figura 8.30 è di confrontare il flusso dei pacchetti in arrivo con il profilo di traffico negoziato e di determinare se un pacchetto rientra in questo profilo. La vera decisione se marcare immediatamente, istradare, ritardare o scartare un pacchetto non è specificata nell' architettura del Diffserv. L'architettura del Diffserv fornisce solo la struttura per l'esecuzione della marcatura dei pacchetti e per la sagomatura/scarto; essa non impone alcuna politica specifica secondo cui marcatura e condizionamento (sagomatura o scarto) in realtà devono essere fatti. La speranza, naturalmente, è che i componenti architetturali di Diffserv abbiano nell'insieme abbastanza flessibilità da permettere un set di servizi e in costante evoluzione agli utenti finali.

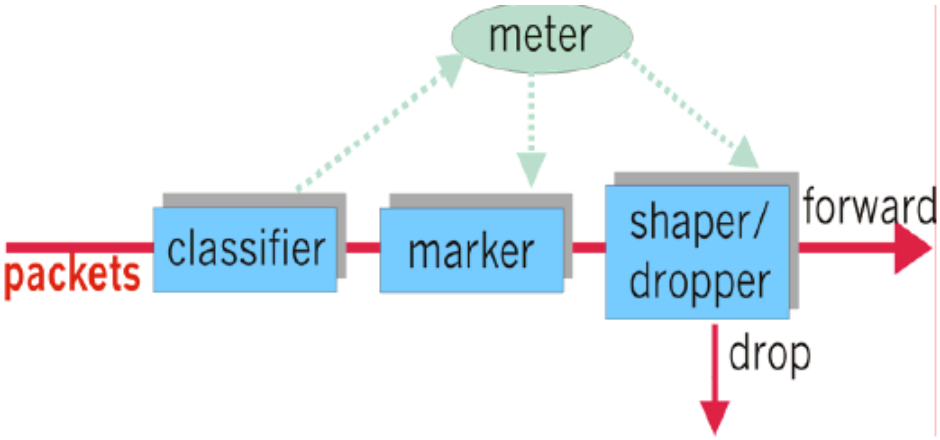


Figura 8.30 – Vista logica della classificazione dei pacchetti e condizionamento del traffico a un router d'estremità

Comportamenti per-hop

Il secondo componente chiave dell' architettura Diffserv coinvolge il comportamento per-hop implementato dai router Diffserv-compatibili. Il comportamento per-hop (PHB, *Per-Hop Behavior)* è definito come "una descrizione osservabile dall'esterno del comportamento di lancio di un nodo Diffserv applicato a un particolare insieme di aggregati di comportamento *(behaviour aggregate)* Diffserv" Esaminando questa definizione, possiamo vedere che incorpora alcune importanti considerazioni:

1. Un PHB può far sì che diverse classi di traffico ricevano diverse prestazioni (cioè, diversi comportamenti di instradamento osservabili dall'esterno).

2. Mentre un PHB definisce differenze in prestazioni (comportamenti) fra le classi, esso non comanda alcun meccanismo particolare per raggiungere questi comportamenti. Finché è soddisfatto il criterio delle prestazioni osservabili dall'esterno, possono esseri usati qualsiasi meccanismo di implementazione e qualsiasi politica di allocazione di buffer e larghezza di banda. Per esempio, un PHB non richiede che per raggiungere un particolare comportamento sia usata una particolare disciplina di accodamento dei pacchetti, per esempio, accodamento prioritario oppure accodamento equo pesato oppure coda con modalità primo arrivato-primo servito, li PI{B è il "fine" per il quale l'allocazione delle risorse e i meccanismi di implementazione sono il "mezzo."

3. Le differenze in prestazioni devono essere osservabili, e quindi misurabili.

Un esempio di un semplice PHI3 è quello che garantisce che una data classe di pacchetti marcati ricevano almeno x% della larghezza di banda del link in uscita in un certo intervallo di tempo. Un altro comportamento per-hop può specificare che una classe di traffico riceverà sempre un'inflessibile priorità rispetto a un'altra: vale a dire, se un pacchetto ad alta priorità e uno a bassa priorità sono contemporaneamente presenti nella coda di un router, il pacchetto ad alta priorità lascerà sempre il router per primo. Notate che mentre una modalità di accodamento prioritario potrebbe essere la scelta naturale per implementare questo secondo PHB, qualsiasi modalità di accodamento che implementi il comportamento osservabile richiesto è accettabile.

Il PHB di inoltro spedito specifica che il tasso di partenza di una classe di traffico da un router debba essere uguale o superiore a un tasso stabilito. Cioè, in un certo intervallo di tempo, alla classe di traffico deve essere garantito di ricevere abbastanza larghezza di banda perché il tasso di uscita del traffico uguagli o superi il tasso minimo configurato.

4. Il PHB di inoltro assicurato è più complesso. L'AF divide il traffico in quattro classi, e a ciascuna di queste quattro classi AF *è* data garanzia di una quantità minima di larghezza di banda e di buffering. All'interno di ciascuna classe i pacchetti sono ulteriormente ripartiti in una di tre categorie con "preferenza di scarto". Quando si verifica la congestione all'interno di una classe AF, un router può allora scartare i pacchetti in base al loro valore di preferenza di scarto.

   PHB AF può essere usato come blocco base per fornire differenti tipi di servizio ai terminali, per esempio, classi di servizio oro, argento e bronzo. Ma cosa serve per poterlo realizzare? Se il servizio oro è infatti destinato a essere "meglio" (e presumibilmente il più costoso!) del servizio argento, allora l'ISP deve assicurare che i pacchetti "oro" debbano essere soggetti a ritardi più bassi e a meno perdite di quelli "argento". Ricordiamo, comunque, che un minimo di larghezza di banda e di buffering deve essere allocato a ciascuna classe. Che cosa accadrebbe se al servizio oro fosse allocato *x%* di una larghezza di banda e al servizio argento ne fosse allocato x/2%, ma l'intensità del traffico dei pacchetti oro fosse 100 volte più grande di quella dei pacchetti argento? In questo caso, è probabile che i pacchetti argento riceveranno prestazioni *migliori* dei pacchetti oro! (Questo *è* un risultato che rende felici gli acquirenti del servizio argento, ma molto infelici quelli dei ben più costosi servizi oro!) Chiaramente, quando si crea un servizio a partire dal PHB, entra in gioco più del solo PHB stesso: in questo esempio, il dimensionamento delle

risorse (che determina quante risorse saranno allocate a ciascuna classe di servizio) deve essere fatto tenendo conto delle richieste delle varie classi di traffico.

Domande di riepilogo

1. Da cosa sono caratterizzate applicazioni di tipo streaming?
   Dal fatto di generare e dover ricevere un flusso continuo di dati.

2. Perchè le applicazioni di comunicazione audio interattiva sono particolarmente critiche?
   Perchè richiedono che l'utente percepisca ritardi (cosidetti end-to-end) molto limitati, normalmente inferiori a 150 - 200 ms per ogni verso della comunicazione (one way delay).

3. Quali sono le implicazioni di applicazioni streaming sulla rete?
   Avendo bisogno di ricevere un flusso di dati con lo stesso profilo del flusso originariamente trasmesso, il tempo di attraversamento della rete deve essere rigorosamente costante. Sebbene piccole variazioni possano essere compensate in fase di ricezione, grandi variazioni sono problematiche in quanto la loro compensazione richiede buffer di grosse dimensioni e l'introduzione di ritardi considerevoli.

4. A cosa è principalmente dovuta la variabilità dei tempi di attraversamento di una rete a pacchetto?
   Al carico di elaborazione dei router, ma soprattutto alla contesa dei pacchetti per i collegamenti di uscita. Quando più di un pacchetto deve essere inoltrato su un collegamento un pacchetto viene trasmesso e gli altri vengono nel frattempo accodati in un buffer in attesa di essere a loro volta trasmessi. Il tempo trascorso nei buffer varia significativamente a seconda del carico istantaneo che attraversa la rete e delle direttrici di traffico dei pacchetti.

5. A cosa server il protocollo RTP (real-time transport protocol)?
   Esso trasporta informazioni ausiliarie per il corretto fuzionamento delle applicazioni multimediali. Queste possono essere informazioni sui partecipanti ad una sessione, informazioni di temporizzazione per la ricostruzione del profilo di traffico generato dal trasmettitore o per la sincronizzazione reciproca dei vari media.

6. Cosa sono gli algoritmi di scheduling?
   Si tratta di algoritmi che permettono di scegliere all'interno di una coda di pacchetti il prossimo pacchetto da trasmettere in modo da ottimizzare qualche criterio. Il criterio di ottimizzazione può essere massimizzare l'utilizzo delle risorse di rete, minimizzare i ritardi, minimizzare le variazioni di ritardo, minimizare le perdite.

7. Cosa si intende per controllo dell'accesso (access control) e perchè è importante?
   La verifica e limitazione del traffico che accede alla rete. Questa verifica può essere fatta a vari livelli che vanno dal singolo pacchetto, al flusso di pacchetti, al numero di utenti che hanno accesso alla rete.

8. Cosa è il traffic engineering?
   La possibilità di distribuire il traffico sulla rete in modo da limitare o eliminare punti di congestione, senza dover necessariamente potenziare la rete.

9. Descrivere i due tipi di servizio previsti dalla soluzione Integrated Services (IntServ)
   Il servizio controlled load prevede un servizio equivalente a quello offerto da una rete scarica. In altre parole si tratta di un servizio senza garanzie sulla qualità, tuttavia il fatto che la rete si comporti come una rete scarica garantisce un buon livello di servizio, probabilmente sufficiente per la maggior parte delle applicazioni.
   Guaranteed service prevede che l'applicazione richieda alla rete un servizio con caratteristche (in termini di ritardo e perdite) ben precise e che la rete confermi o meno la possibilità di offrire tale servizio. Nel primo caso la rete garantisce la forntura di un servio con le caratteristiche richieste.

10. A cosa serve RSVP (resource reservation protocol)?
    Alle applicazioni per dichiararele caratteristiche del traffico che generano e del servizio di cui hanno bisogno. Alla rete per comunicare queste informazioni ai nodi coinvolti e prenotare risorse negli stessi in modo che siano in grado di erogare il servizio richiesto.

11. Qual è l'obiettivo dell'approccio Differentiated Services (DiffServ)?
    Separare i pacchetti in transito nella rete in classi differenti che vengono trattate dai nodi di rete in modo differente.

12. Quali sono i vantaggi dell'approccio DiffServ?

Elevata semplicità di implementazione e di utilizzo cui corrispondono bassi costi.

13. Cosa è il per-hop behavior (PHB)?

    La modalità con cui pacchetti appartenenti ad una certa classe di traffico vengono trattati da ogni nodo attraversato.