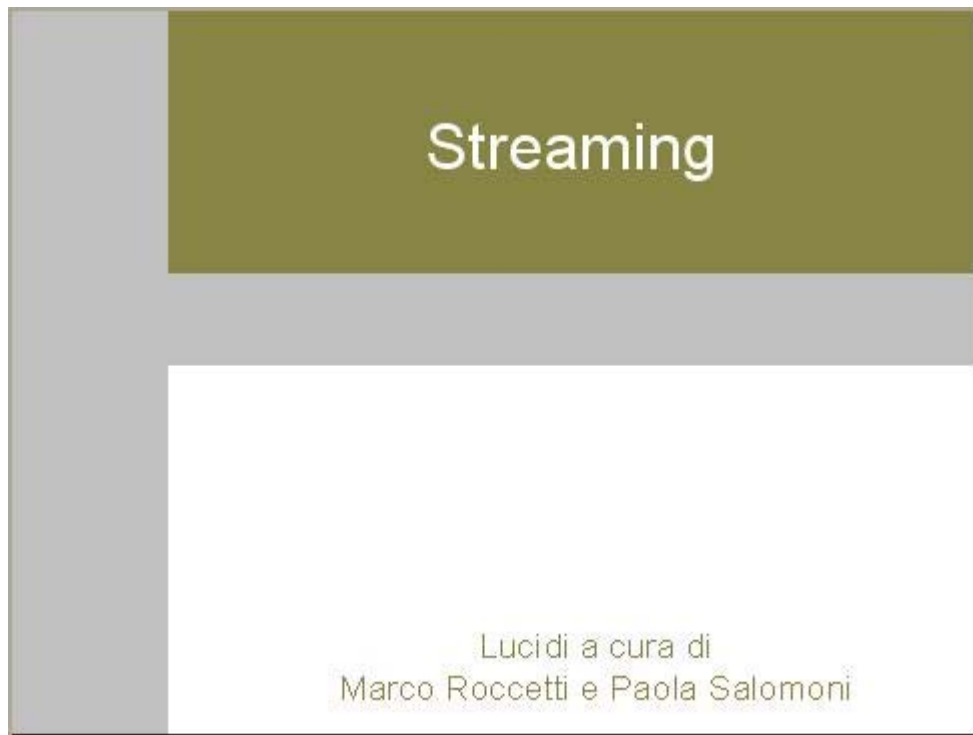


Streaming
Streaming



Buongiorno, io mi chiamo Paola Salomoni e insegno Sistemi Operativi e Sistemi Multimediali all'Università di Bologna. Lo scopo di questa lezione è quello di introdurre una particolare metodologia di trasferimento di file multimediali su Internet che è nota come Streaming. Il materiale didattico che utilizzeremo è stato preparato in collaborazione con un mio collega, il prof. Marco Rocchetti, che insegna Reti e Sistemi e Applicazioni Multimediali nella mia stessa Università.

Multimedia su Internet (1)

A presentation slide with a grey background. In the top left corner is the number "2". A dark green header contains the title "Multimedia su Internet (1)" in white. Below the header, two bullet points are listed in a teal color:

- Il delivery di dati multimediali digitali via supporto ottico non è (quasi mai) problematico in termini di dimensioni (in byte).
- La prospettiva cambia operando su rete: i media digitali, specialmente quelli continui, richiedono una **larghezza di banda** (in bit al secondo) difficilmente disponibile.

La distribuzione di dati multimediali digitali attraverso supporti ottici non è al giorno d'oggi quasi mai problematica in termini dimensionali. Sono infatti disponibili sul mercato supporti ottici di grandi dimensioni, come i DVD, che sono stati progettati con lo scopo di contenere, e quindi consentire la commercializzazione, di intere produzioni cinematografiche in formato digitale. All'interno di un DVD, possono essere contenute intere ore di video digitale, più tracce audio corrispondenti alla traccia video principale (che sono sostanzialmente le tracce audio delle diverse lingue disponibili) e, contemporaneamente, anche di elementi aggiuntivi, come per esempio i sottotitoli. I DVD attualmente sono in grado di contenere fino a 17 Gbyte e quindi diciamo che il trasporto di dati multimediali attraverso supporto ottico non è un problema in termini dimensionali.

Multimedia su Internet (2)



2

Multimedia su Internet (1)

- Il delivery di dati multimediali digitali via supporto ottico non è (quasi mai) problematico in termini di dimensioni (in byte).
- La prospettiva cambia operando su rete: i media digitali, specialmente quelli continui, richiedono una **larghezza di banda** (in bit al secondo) difficilmente disponibile.

La prospettiva cambia radicalmente quando si opera attraverso Internet. I media digitali e in modo particolare quelli continui, richiedono infatti una larghezza di banda che è difficilmente disponibile per tutti, al giorno d'oggi. Va ricordato, ovviamente, che i media continui (l'audio e il video, in modo particolare) producono in fase di digitalizzazione una certa quantità di byte al secondo e questa quantità di byte spiega sostanzialmente la richiesta in termini di larghezza di banda.

Multimedia su Internet (3)

3

Multimedia su Internet (2)

- Altre problematiche derivano dalla tipologia del servizio offerto da Internet (che è di tipo **best effort**).
- Per comprendere meglio queste problematiche e le metodologie di soluzione, richiamiamo molto brevemente alcuni concetti presentati nel **Modulo 5**.

Le difficoltà nel distribuire contenuti multimediali attraverso Internet non sono correlate soltanto alla disponibilità di larghezza di banda ma derivano soprattutto dalla tipologia di servizio che offre questa rete. Internet infatti offre un servizio cosiddetto di tipo best effort che si presta molto male al trasporto di dati di tipo continuo, come sono invece i dati multimediali come l'audio e il video. Per questo tipo di media, il segnale cambia nel tempo, abbiamo detto, e sostanzialmente viene prodotta una certa quantità di byte al secondo. Questa quantità di byte viene prodotta in maniera continua, come il nome stesso dice, e quindi si presuppone che il sistema di trasporto, cioè la rete, sia in grado di offrire uguale continuità nel distribuire questi dati e trasportarli dalla sorgente alla destinazione.

Multimedia su Internet (4)

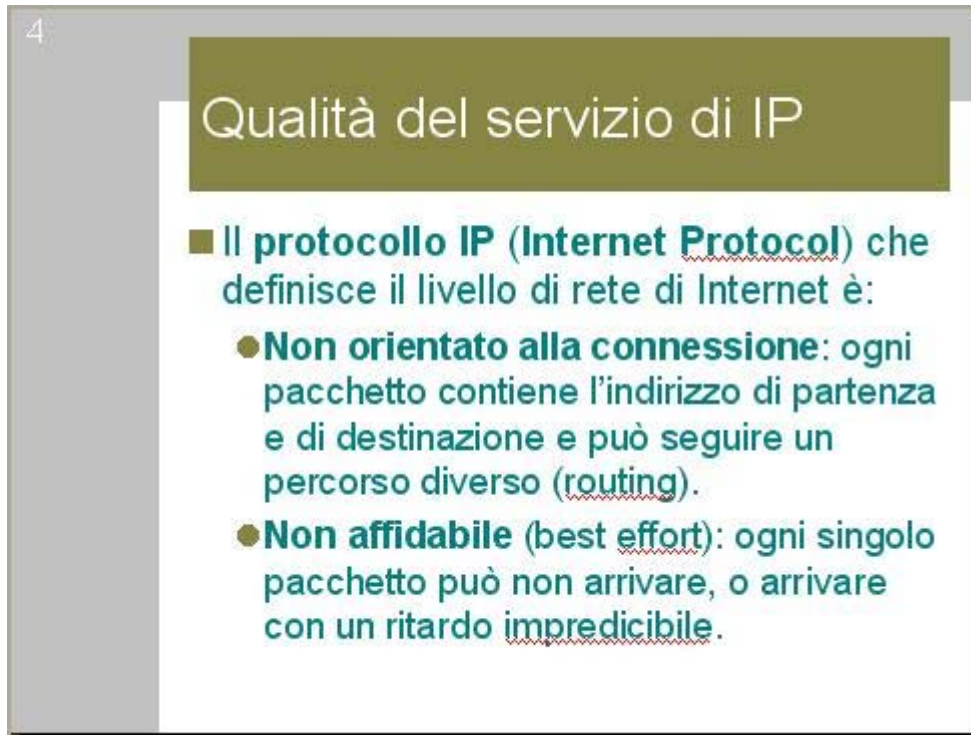
3

Multimedia su Internet (2)

- Altre problematiche derivano dalla tipologia del servizio offerto da Internet (che è di tipo **best effort**).
- Per comprendere meglio queste problematiche e le metodologie di soluzione, richiamiamo molto brevemente alcuni concetti presentati nel **Modulo 5**.

Per comprendere meglio queste problematiche, correlate alla tipologia di servizio offerto dalla rete Internet, e anche per capire quali sono le principali soluzioni, richiameremo molto brevemente alcuni concetti che sono stati presentati nel Modulo 5, sull'introduzione alle reti.

Qualità del servizio di IP (1)



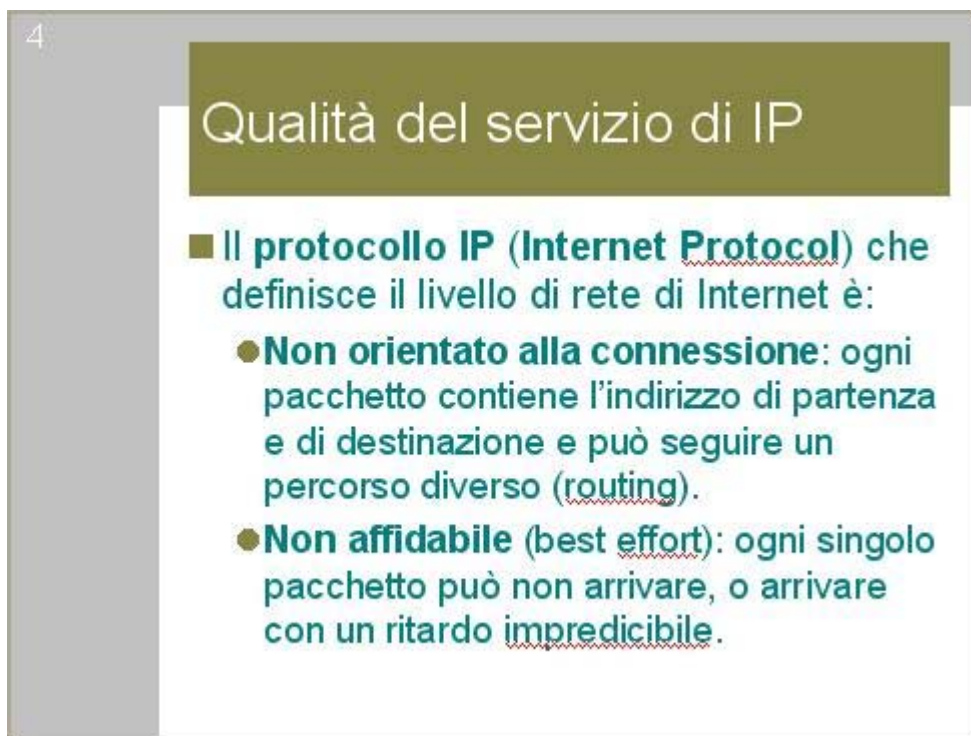
4

Qualità del servizio di IP

- Il **protocollo IP (Internet Protocol)** che definisce il livello di rete di Internet è:
 - **Non orientato alla connessione:** ogni pacchetto contiene l'indirizzo di partenza e di destinazione e può seguire un percorso diverso (routing).
 - **Non affidabile (best effort):** ogni singolo pacchetto può non arrivare, o arrivare con un ritardo imprevedibile.

Il protocollo di rete di Internet è IP, che sta per Internet Protocol ed è un protocollo che ha due caratteristiche fondamentali: non è orientato alla connessione e non è affidabile.

Qualità del servizio di IP (2)



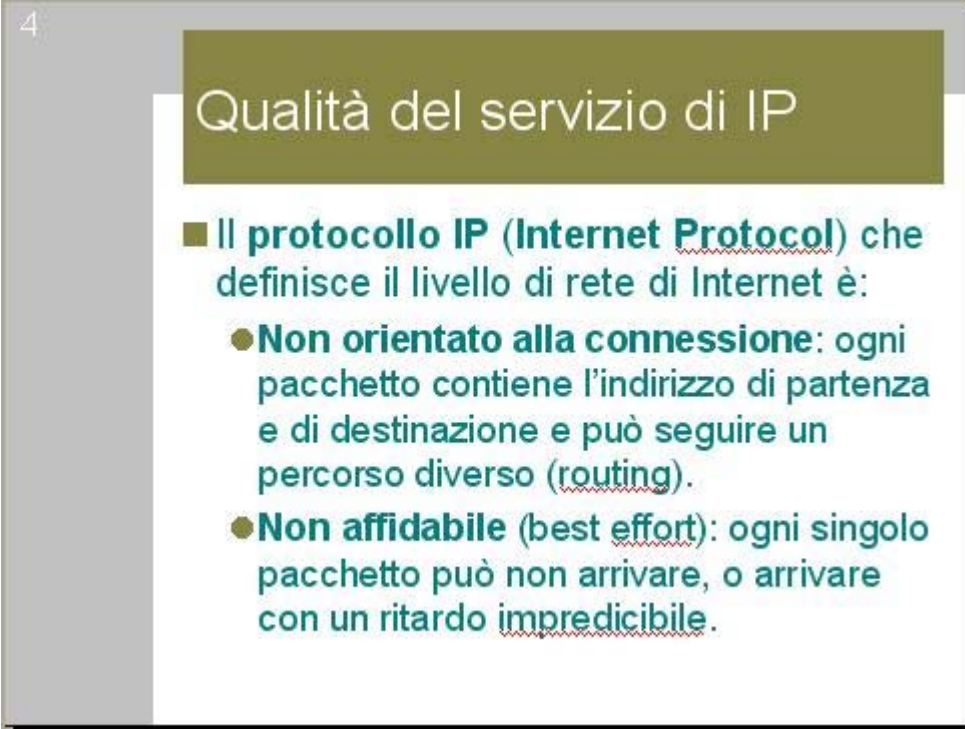
4

Qualità del servizio di IP

- Il **protocollo IP (Internet Protocol)** che definisce il livello di rete di Internet è:
 - **Non orientato alla connessione:** ogni pacchetto contiene l'indirizzo di partenza e di destinazione e può seguire un percorso diverso (routing).
 - **Non affidabile (best effort):** ogni singolo pacchetto può non arrivare, o arrivare con un ritardo imprevedibile.

Con "Non orientato alla connessione" si intende che ogni pacchetto che viene instradato su Internet contiene l'indirizzo di partenza e l'indirizzo di destinazione, ma può seguire un percorso qualunque attraverso i nodi intermedi. Pacchetti successivi possono seguire percorsi diversi, che sono determinati dai protocolli di routing che sono stati introdotti dai moduli precedenti.

Qualità del servizio di IP (3)



4

Qualità del servizio di IP

- Il protocollo IP (Internet Protocol) che definisce il livello di rete di Internet è:
 - **Non orientato alla connessione**: ogni pacchetto contiene l'indirizzo di partenza e di destinazione e può seguire un percorso diverso (routing).
 - **Non affidabile (best effort)**: ogni singolo pacchetto può non arrivare, o arrivare con un ritardo impredicibile.

Con "Non affidabile", invece, si intende che ogni singolo pacchetto potrebbe anche non arrivare alla destinazione, oppure che, nel caso che arrivi, il ritardo che ha accumulato durante il trasporto sulla rete è comunque impredicibile sia al lato destinazione che al lato sorgente.

Qualità del servizio di IP (4)

4

Qualità del servizio di IP

- Il **protocollo IP (Internet Protocol)** che definisce il livello di rete di Internet è:
 - **Non orientato alla connessione**: ogni pacchetto contiene l'indirizzo di partenza e di destinazione e può seguire un percorso diverso (**routing**).
 - **Non affidabile (best effort)**: ogni singolo pacchetto può non arrivare, o arrivare con un ritardo **impredicibile**.

Questa proprietà è alla base della definizione della qualità del servizio di Internet, che è appunto detta di tipo "best effort".

Livello di trasporto (1)

5

Livello di trasporto

- Nella suite di protocolli IP sono disponibili due diversi protocolli di **trasporto**:
 - **TCP (Transmission Control Protocol)**: è un protocollo con connessione di tipo affidabile (ossia tutti i pacchetti arrivano, e rispettando l'ordine d'invio).
 - **UDP (User Datagram Protocol)**: è un protocollo senza connessione e di tipo non affidabile, i pacchetti possono arrivare in ordine diverso (rispetto all'invio) o non arrivare affatto.

Salendo di un livello nella suite di protocolli di Internet, cioè nella suite di protocolli che poggia sopra IP, abbiamo a disposizione due diversi protocolli di livello trasporto: TCP e UDP.

Livello di trasporto (2)

5

Livello di trasporto

- Nella suite di protocolli IP sono disponibili due diversi protocolli di **trasporto**:
 - **TCP (Transmission Control Protocol)**: è un protocollo con connessione di tipo affidabile (ossia tutti i pacchetti arrivano, e rispettando l'ordine d'invio).
 - **UDP (User Datagram Protocol)**: è un protocollo senza connessione e di tipo non affidabile, i pacchetti possono arrivare in ordine diverso (rispetto all'invio) o non arrivare affatto.

TCP è un protocollo con connessione, di tipo affidabile, e dunque è un protocollo che verifica l'arrivo di tutti i pacchetti che vengono trasmessi e ne ricostruisce anche l'ordine. D'altro canto, invece, UDP è un protocollo senza connessione e di tipo non affidabile. Quindi i pacchetti spediti con UDP possono arrivare in ordine diverso rispetto all'ordine di invio oppure potrebbero non arrivare affatto. Rispetto alle caratteristiche fino ad ora definite, quindi connessione e affidabilità, sostanzialmente UDP non aggiunge nulla rispetto al protocollo di rete su cui si appoggia, cioè rispetto a IP.

TCP vs UDP (1)

6

TCP vs UDP

- In termini molto semplici, le **diffarenze** tra TCP e UDP possono essere ridotte a:
 - TCP garantisce la consegna (ordinata) dei pacchetti, UDP no.
 - TCP richiede più banda e più tempo di comunicazione (e di sistema) rispetto a UDP poiché, per garantire la consegna dei pacchetti, deve effettuare controlli e ritrasmissioni.
- UDP può essere utilizzato quando si può tollerare che non tutti i pacchetti arrivino a destinazione o quando si demanda all'applicazione il problema di gestire perdite o ritardi dei pacchetti.

In termini molto semplici, le principali differenze che abbiamo individuato tra TCP e UDP possono

essere riassunte in questo modo.

TCP vs UDP (2)

6

TCP vs UDP

- In termini molto semplici, le **differenze** tra TCP e UDP possono essere ridotte a:
 - TCP garantisce la consegna (ordinata) dei pacchetti, UDP no.
 - TCP richiede più banda e più tempo di comunicazione (e di sistema) rispetto a UDP poiché, per garantire la consegna dei pacchetti, deve effettuare controlli e ritrasmissioni.
- UDP può essere utilizzato quando si può tollerare che non tutti i pacchetti arrivino a destinazione o quando si demanda all'applicazione il problema di gestire perdite o ritardi dei pacchetti.

TCP garantisce la consegna ordinata dei pacchetti, mentre UDP no. Per svolgere questa attività che è di verifica e di controllo, TCP richiede l'uso di una maggiore larghezza di banda e anche più tempo di comunicazione, e anche di sistema rispetto a quanto non richiede UDP. Queste risorse vengono impiegate per garantire la consegna dei pacchetti ed effettuare controlli e ritrasmissioni in caso che questa consegna non avvenga correttamente.

TCP vs UDP (3)

6

TCP vs UDP

- In termini molto semplici, le **differenze** tra TCP e UDP possono essere ridotte a:
 - TCP garantisce la consegna (ordinata) dei pacchetti, UDP no.
 - TCP richiede più banda e più tempo di comunicazione (e di sistema) rispetto a UDP poiché, per garantire la consegna dei pacchetti, deve effettuare controlli e ritrasmissioni.
- UDP può essere utilizzato quando si può tollerare che non tutti i pacchetti arrivino a destinazione o quando si demanda all'applicazione il problema di gestire perdite o ritardi dei pacchetti.

Quando quindi può essere utilizzato UDP? UDP può essere utilizzato quando si può tollerare che non tutti i pacchetti arrivino a destinazione o quando si demanda al livello di applicazione, cioè al livello che sta sopra il livello di trasporto, il problema di gestire le perdite o i ritardi dei pacchetti.

Applicazioni multimediali (1)

7

Applicazioni Multimediali (1)

- Possiamo distinguere due tipi di applicazioni:
 - Quelle che richiedono o comunque privilegiano il download completo delle risorse multimediali (ES: applicazioni Peer to Peer per il delivery di multimedia).
 - Quelle che non necessitano o non sono sostengono il download completo delle risorse ma operano sui flussi generati dal media continuo nel tempo.

La scelta del protocollo di trasporto da utilizzare in ambito multimediale dipende fortemente dal tipo di applicazione multimediale che si vuole realizzare.

Applicazioni multimediali (2)


7

Applicazioni Multimediali (1)

- Possiamo distinguere due tipi di applicazioni:
 - Quelle che richiedono o comunque privilegiano il download completo delle risorse multimediali (ES: applicazioni Peer to Peer per il delivery di multimedia).
 - Quelle che non necessitano o non sono sostengono il download completo delle risorse ma operano sui flussi generati dal media continuo nel tempo.

Possiamo classificare le applicazioni che hanno come obiettivo quello di trasmettere informazioni di tipo multimediale su Internet, in due macrocategorie che vengono individuate discriminando in base tipo di scaricamento delle risorse che queste applicazioni effettuano.

Applicazioni multimediali (3)



7

Applicazioni Multimediali (1)

- Possiamo distinguere due tipi di applicazioni:
 - Quelle che richiedono o comunque privilegiano il download completo delle risorse multimediali (ES: applicazioni Peer to Peer per il delivery di multimedia).
 - Quelle che non necessitano o non sono sostengono il download completo delle risorse ma operano sui flussi generati dal media continuo nel tempo.

Esiste un primo gruppo di applicazioni che richiedono o che comunque privilegiano il download completo delle risorse multimediali. Sono applicazioni di questo tipo tutti i sistemi Peer to Peer che vengono utilizzati per lo scambio di file musicali o anche di film attraverso la rete e un esempio molto famoso di questo genere di applicazione è Napster. È evidente che in casi come questo lo scopo del software è quello di portare una copia del file multimediale all'utente che lo ha richiesto e quindi lo scaricamento del file deve essere totale affinché l'applicazione possa dire di aver completato con successo il suo compito.

Applicazioni multimediali (4)

7

Applicazioni Multimediali (1)

- Possiamo distinguere due tipi di applicazioni:
 - Quelle che richiedono o comunque privilegiano il download completo delle risorse multimediali (ES: applicazioni Peer to Peer per il delivery di multimedia).
 - Quelle che non necessitano o non sono sostengono il download completo delle risorse ma operano sui flussi generati dal media continuo nel tempo.

Per contro esistono applicazioni che non necessitano del download completo delle risorse o che addirittura in alcuni casi non possono sostenerlo. In questo secondo caso l'applicazione opera in alcuni casi direttamente sui flussi generati dal media continuo nel tempo e da questo momento della lezione ci interesseremo in realtà soltanto di questo secondo genere di applicazione multimediale.

Applicazioni multimediali (5)

8

Applicazioni Multimediali (2)

- Non sempre il download completo del file è una strategia utilizzabile.
- In molti casi l'attesa per un download completo è insostenibile :
 - **Tempo reale/Liveness**: per esempio, Internet-Radio e Internet-TV vogliono raggiungere *prima possibile* i propri ascoltatori e/o telespettatori.
 - **Dimensione del file/ Attesa**: per esempio, il download di un intero film (tipicamente, di notevoli dimensioni in byte) può richiedere ore.

Non sempre il download completo del file è una strategia utilizzabile ed esistono addirittura casi in cui è una metodica di tipo insostenibile. Non è possibile, per esempio, il download completo in caso di applicazioni che operino in tempo reale, come può essere per esempio la radio via Internet, o analogamente la TV via Internet. In questo caso la trasmissione sostanzialmente è in corso, non è

completata registrazione e dunque la produzione di un file multimediale, ma si può discutere esclusivamente di flussi, per cui è possibile la trasmissione del flusso ma non è possibile lo scaricamento del file.

Applicazioni multimediali (6)



8

Applicazioni Multimediali (2)

- Non sempre il download completo del file è una strategia utilizzabile.
- In molti casi l'attesa per un download completo è insostenibile :
 - **Tempo reale/Liveness:** per esempio, Internet-Radio e Internet-TV vogliono raggiungere *prima possibile* i propri ascoltatori e/o telespettatori.
 - **Dimensione del file/ Attesa:** per esempio, il download di un intero film (tipicamente, di notevoli dimensioni in byte) può richiedere ore.

Esistono poi casi in cui il download sarebbe possibile ma sarebbe così lungo da scoraggiare l'utente. Per esempio nel caso del video on demand, il download di un intero film può richiedere anche ore e l'utente non è tipicamente disponibile ad attendere così tanto per iniziare a fruire del servizio che ha richiesto.

Streaming

9

Streaming

- In questi i casi, si usa una tecnica nota come **streaming** che opera considerando il media come stream (**flusso**) e non come file.
- L'host ricevente comincia il playout del media continuo (audio o video) prima che sia conclusa la comunicazione con l'host trasmittente.

In questi i casi, si usa una tecnica che nota come streaming e che opera considerando il media come stream, ovvero come flusso, e non come file. Lo streaming è applicabile a media continui e tipicamente è utilizzato per la trasmissione di audio e video. La continuità del media fa sì che, a fronte dello scorrere del tempo, si produca attraverso le fasi di digitalizzazione e compressione un flusso di dati che dovrà essere trasportato attraverso una sequenza di pacchetti che vengono inviati sulla rete. Nello streaming l'host ricevente comincia il playout del media continuo, cioè dell'audio oppure del video, prima che l'intera sequenza di pacchetti sia stata ricevuta e dunque prima che sia conclusa la comunicazione con l'host trasmittente.

Streaming: trasporto su UDP (1)

10

Streaming: trasporto su UDP

- Tipicamente i sistemi di streaming operano su **UDP** (cioè il protocollo di trasporto che **NON** controlla l'integrità della trasmissione, ne' l'ordine di arrivo dei pacchetti):
 - **PRO**: viene velocizzata la trasmissione liberandola dai controlli,
 - **CON**: possono esserci pacchetti persi e pacchetti "disordinati", causati dal servizio *best effort* offerto da IP.

Tipicamente i sistemi di streaming operano su UDP, ovvero a livello di trasporto la sequenza di pacchetti viene trasmessa senza controllare né l'integrità della trasmissione, né l'ordine di arrivo. Questa scelta consente di ottenere una trasmissione complessivamente più leggera, perché è liberata dai controlli. Contemporaneamente però, possono esserci pacchetti persi e pacchetti "disordinati" (cioè che arrivano in un ordine diverso da quello in cui sono stati trasmessi) e questi due fenomeni sono causati dal servizio *best effort* che viene fornito dalla rete Internet.

Streaming: trasporto su UDP (2)



10

Streaming: trasporto su UDP

- Tipicamente i sistemi di streaming operano su **UDP** (cioè il protocollo di trasporto che **NON** controlla l'integrità della trasmissione, né l'ordine di arrivo dei pacchetti):
 - **PRO**: viene velocizzata la trasmissione liberandola dai controlli,
 - **CON**: possono esserci pacchetti persi e pacchetti "disordinati", causati dal servizio *best effort* offerto da IP.

Da notare che l'ordine deve comunque essere ricostruito a livello applicazione mentre per i pacchetti persi vengono adottate tipicamente politiche tolleranti all'errore. Ovviamente l'errore è tollerabile entro certi limiti perché il tipo di dati trasferito è percepito dall'utente in modo accettabile anche in presenza di piccole perdite.

Jitter

11

Jitter

- Se il controllo di ordine ed integrità non è effettuato a livello di trasporto, deve essere effettuato a livello applicazione.
- Il disordine che si produce presso la stazione ricevente è causato dal fatto che ciascun pacchetto impiega un tempo diverso a raggiungere la destinazione.
- Questa variabilità del ritardo prodotto dalla trasmissione è detta **delay jitter** (variazione del ritardo).

Se il controllo di ordine e di integrità non viene effettuato a livello di trasporto, perché stiamo utilizzando UDP, allora deve essere effettuato a livello applicazione. Il disordine che si produce presso la stazione ricevente è sostanzialmente causato dal fatto che ciascun pacchetto impiega un tempo diverso a raggiungere la destinazione. Questo effetto è proprio del best effort di IP, e non viene corretto dal livello di trasporto perché abbiamo scelto di utilizzare UDP, quindi si riflette sul livello di applicazione. La variabilità del ritardo prodotto dalla trasmissione è detta delay jitter.

Streaming (1)

12

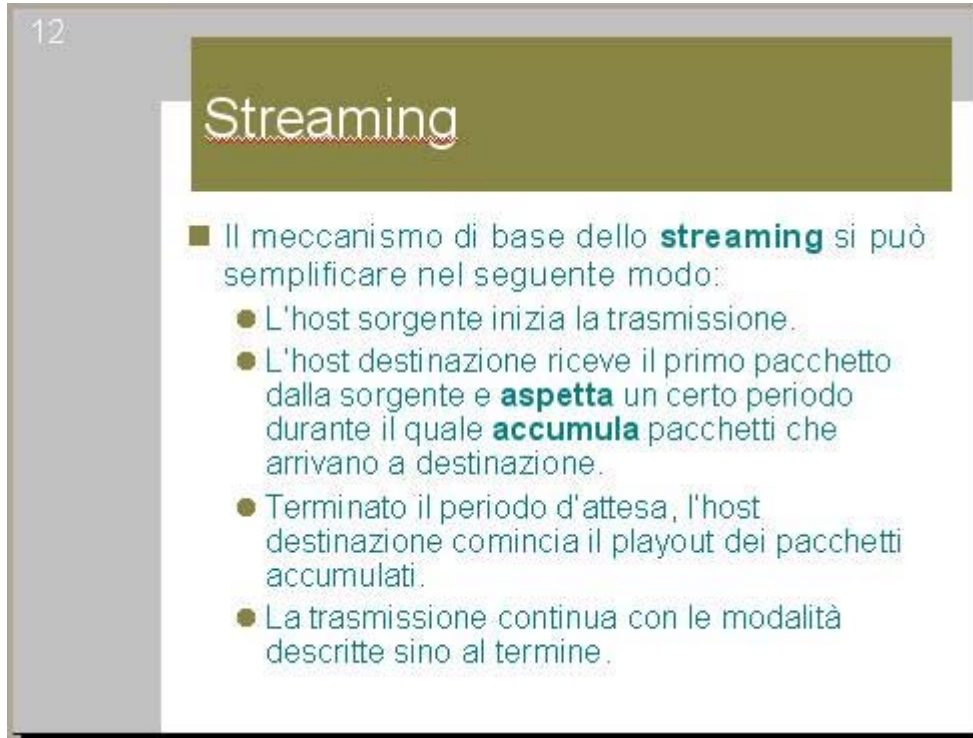
Streaming

- Il meccanismo di base dello **streaming** si può semplificare nel seguente modo:
 - L'host sorgente inizia la trasmissione.
 - L'host destinazione riceve il primo pacchetto dalla sorgente e **aspetta** un certo periodo durante il quale **accumula** pacchetti che arrivano a destinazione.
 - Terminato il periodo d'attesa, l'host destinazione comincia il playout dei pacchetti accumulati.
 - La trasmissione continua con le modalità descritte sino al termine.

Il meccanismo alla base dello streaming si può semplificare nella seguente sequenza: La prima cosa che avviene è che l'host sorgente inizia la trasmissione del flusso, sottoforma di pacchetti che

contengono le informazioni multimediali. L'host destinazione riceve il primo pacchetto dalla sorgente e prima di mandarlo in playout (ovvero prima di farlo ascoltare all'utente o farlo vedere all'utente), aspetta un certo periodo durante il quale accumula pacchetti che arrivano a destinazione.

Streaming (2)



12

Streaming

- Il meccanismo di base dello **streaming** si può semplificare nel seguente modo:
 - L'host sorgente inizia la trasmissione.
 - L'host destinazione riceve il primo pacchetto dalla sorgente e **aspetta** un certo periodo durante il quale **accumula** pacchetti che arrivano a destinazione.
 - Terminato il periodo d'attesa, l'host destinazione comincia il playout dei pacchetti accumulati.
 - La trasmissione continua con le modalità descritte sino al termine.

Terminato il periodo d'attesa, l'host destinazione comincia il playout dei pacchetti che ha accumulato e poi la trasmissione continua così, con queste modalità, fino al termine. L'attesa introdotta al ricevente è strategica, come vedremo nel seguito della lezione, perché attenua il jitter. Infatti l'immediata partenza del playout all'arrivo del primo pacchetto renderebbe la fruizione estremamente sensibile alle variazioni del ritardo che i pacchetti successivi al primo possono avere. La presenza dei pacchetti accumulati, attenua sostanzialmente l'effetto di questa variabilità.

Buffer

13

Buffer

- L'host destinazione deve quindi predisporre uno spazio di memoria in cui accumulare i pacchetti in attesa del playout (**buffer**).
- Il buffer è una struttura che viene riempita da un lato e svuotata dall'altro.
- In realtà la struttura non si riempirà in modo regolare:
 - Perché i pacchetti possono arrivare disordinatamente.
 - Perché alcuni pacchetti non arrivano affatto.

L'host destinazione deve quindi predisporre uno spazio di memoria in cui accumulare i pacchetti in attesa del playout, ovvero deve predisporre un buffer. Tipicamente un buffer è una struttura che viene riempita da un lato e svuotata dall'altro ma in realtà nel nostro caso la struttura non si riempirà in modo regolare, sia perché i pacchetti possono arrivare in un ordine qualunque, sia perché alcuni pacchetti non arriveranno affatto. La dimensione del buffer è strettamente correlata al periodo d'attesa, più pacchetti contiene il buffer e più lungo sarà il tempo necessario a riempirlo quindi più lunga sarà l'attesa prima del playout del primo pacchetto.

Buffer: esempio (1)

14

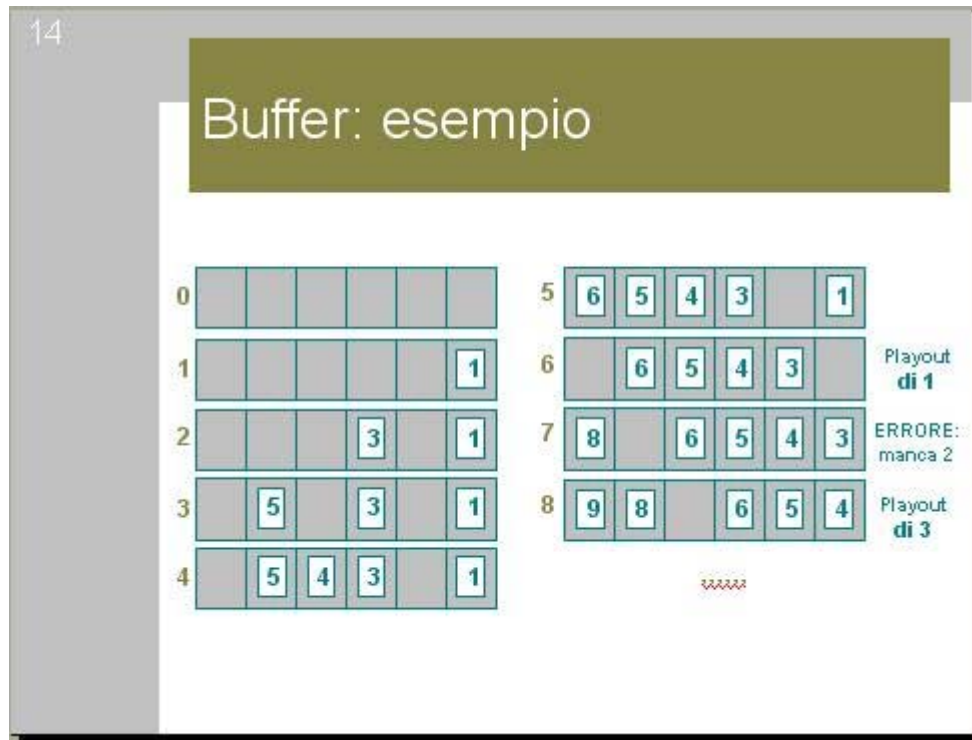
Buffer: esempio

0									
1									1
2					3				1
3		5			3				1
4		5	4	3					1
5	6	5	4	3					1
6		6	5	4	3				Playout di 1
7	8		6	5	4	3			ERRORE: manca 2
8	9	8		6	5	4			Playout di 3
9									

In questa slide vediamo un esempio di avvio dello streaming, con il riempimento del buffer e il

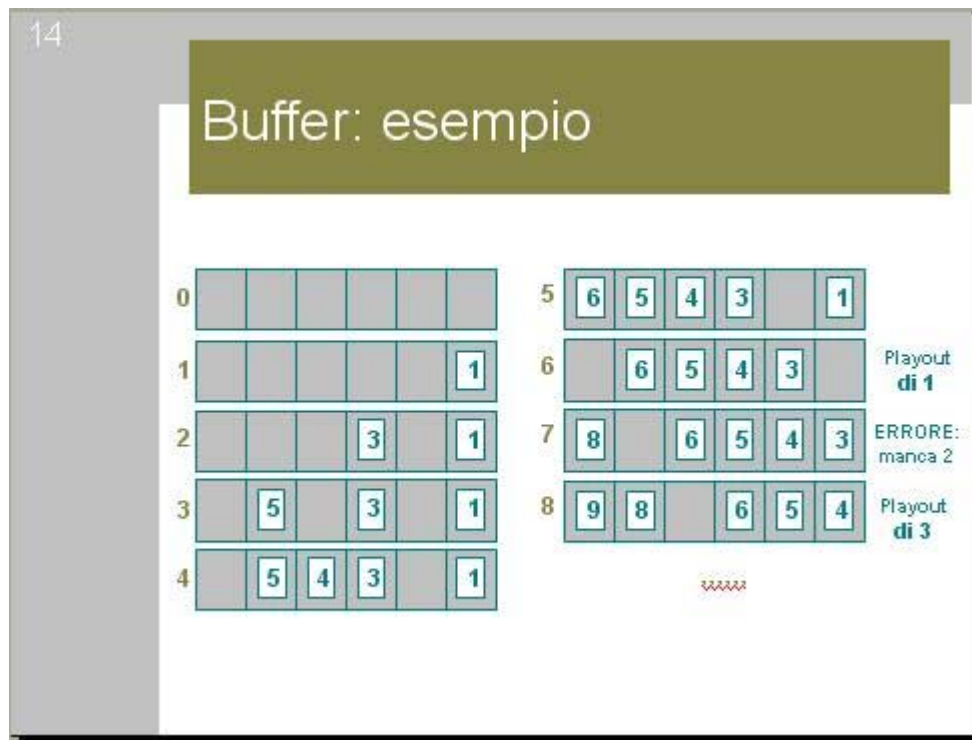
parallelo meccanismo di svuotamento. Nel nostro caso il buffer è composto da 6 aree di memoria, ovvero può contenere 6 pacchetti. L'attesa al ricevente sarà quindi pari alla lunghezza del flusso che viene contenuto nei sei pacchetti. Scandiamo il tempo su sei intervalli, intervalli di tipo discreto, ricordando che ognuno corrisponde in termini di tempo in secondi alla dimensione del flusso che viene contenuto in un pacchetto.

Buffer: esempio (2)



All'inizio, cioè al tempo zero della simulazione, il buffer è vuoto. Lo streaming inizia al tempo 1, con l'arrivo del primo pacchetto che viene inserito nella cella che per prima verrà svuotata (e convenzionalmente quella più a destra nella figura). Il tempo 2 visualizza l'arrivo del pacchetto 3. Si noti che il pacchetto 2 non è ancora arrivato ma nel mentre è arrivato il 3, questo fenomeno è introdotto dal delay jitter. Il tempo 3 mostra l'arrivo del pacchetto 5 e il tempo 4 l'arrivo del pacchetto 4. Anche l'inversione nell'arrivo dei pacchetti 4 e 5 è un fenomeno tipico del delay jitter.

Buffer: esempio (3)



Al tempo 5 arriva il pacchetto 6. Essendo passati 6 intervalli di tempo, al tempo 6 inizia il playout, col pacchetto 1. Il buffer viene quindi svuotato a partire da destra e i pacchetti che sono in giacenza scorrono verso destra, creando così lo spazio per ospitare i pacchetti successivi in arrivo. Al tempo 7 arriva il pacchetto 8 e contemporaneamente dovrebbe andare in playout il pacchetto 2, che però non è arrivato. Si verifica dunque un errore nel flusso fruito dall'utente. L'ultimo tempo che viene riportato nell'esempio, l'8, vede contemporaneamente l'arrivo del pacchetto 9 e il playout del pacchetto 3. Lo streaming continua così fino a conclusione del flusso. Si noti che possono esistere istanti di tempo in cui non arrivano pacchetti, come accade, nell'esempio che abbiamo riportato per il tempo 6.

Packet Loss (1)

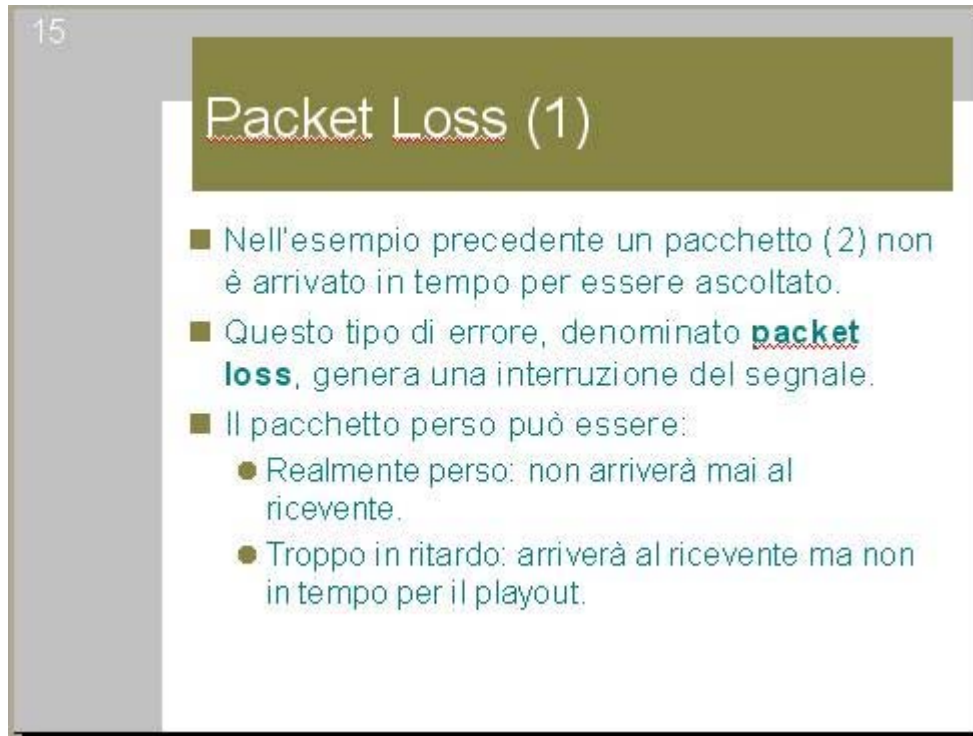
15

Packet Loss (1)

- Nell'esempio precedente un pacchetto (2) non è arrivato in tempo per essere ascoltato.
- Questo tipo di errore, denominato **packet loss**, genera una interruzione del segnale.
- Il pacchetto perso può essere:
 - Realmente perso: non arriverà mai al ricevente.
 - Troppo in ritardo: arriverà al ricevente ma non in tempo per il playout.

Nell'esempio riportato il pacchetto numero 2 non è arrivato in tempo per essere ascoltato. Questo tipo di errore è denominato **packet loss**, e genera una interruzione nella continuità del segnale che ha effetti differenti nel caso audio e nel caso del video. Il pacchetto perso può essere realmente perso oppure può essere arrivato troppo in ritardo per essere mandato in playout. Se il pacchetto è realmente perso, non arriverà mai al ricevente, ovvero il ricevente riceverà la richiesta di concludere lo streaming senza che il pacchetto perduto sia mai arrivato. Se invece il pacchetto è troppo in ritardo, allora arriverà al ricevente ma non in tempo per il playout, ovvero arriverà in un momento in cui è già avvenuto il playout di pacchetti successivi a quello smarrito. Non è possibile recuperare l'errore perché in realtà, nel tempo, si è già verificato.

Packet Loss (2)



15

Packet Loss (1)

- Nell'esempio precedente un pacchetto (2) non è arrivato in tempo per essere ascoltato.
- Questo tipo di errore, denominato **packet loss**, genera una interruzione del segnale.
- Il pacchetto perso può essere:
 - Realmente perso: non arriverà mai al ricevente.
 - Troppo in ritardo: arriverà al ricevente ma non in tempo per il playout.

Dal punto di vista sostanziale, per l'applicazione e quindi in definitiva anche per l'utente, si tratta in entrambi i casi di pacchetti persi che producono una interruzione nel flusso e che quindi possono anche avere anche avere un riscontro percettivo.

Packet Loss (3)

16

Packet Loss (2)

- Quando un pacchetto è perso, il ricevente può adottare diverse strategie per rimediare durante il playout.
- Le più semplici sono:
 - **Silenzio**: non viene riprodotto nulla.
 - **Ripetizione**: viene riprodotto l'ultimo pacchetto arrivato.
 - **Interpolazione**, se sono già disponibili un pacchetto precedente ed uno successivo viene effettuata una interpolazione tra i due.

Quando un pacchetto è perso, l'applicazione all'host ricevente può adottare diverse strategie per rimediare durante il playout. Vediamo insieme alcune tra le più semplici. La prima è quella di riprodurre silenzio ovvero non riprodurre nulla; nel caso dell'audio si tratterà di vero e proprio silenzio, nel caso del video si spezzerà la sequenza dei frame. È una strategia banale ma poco efficace perché l'utente è molto sensibile a questo tipo di errore e lo percepisce facilmente. Un altro metodo consiste nella ripetizione dei pacchetti disponibili. In questo caso viene riprodotto l'ultimo pacchetto arrivato e dunque un certo segnale viene ripetuto più volte auspicando che la ripetizione non sia percepita dall'utente. Può essere una scelta efficace per intervalli di tempo molto brevi. Infine un altro metodo consiste nell'effettuare delle interpolazioni. Se sono già disponibili un pacchetto precedente ed uno successivo, viene effettuata una interpolazione tra i due, ottenendo un segnale che mescoli in modo opportuno il segnale che lo precede con quello che lo segue e quindi renda difficilmente percepibile all'utente la perdita che è avvenuta.

Buffer e QoS

17

Buffer e QoS

- La struttura di memorizzazione introdotta induce sulla trasmissione alcuni **vantaggi**:
 - Effettua automaticamente il riordino.
 - Tenta di compensare e gestire il delay jitter.
- Contestualmente appare evidente un **effetto negativo**: i pacchetti messi in attesa ritardano forzatamente il momento del loro playout.

La presenza del buffer induce sulla trasmissione alcuni vantaggi. Il primo è che il buffer effettua automaticamente il riordino, poiché i pacchetti sono collocati nel buffer nella loro corretta posizione. L'ordine quindi è ripristinato, come avevamo premesso, a livello di applicazione. Il buffer in parte compensa anche e gestisce il delay jitter. La struttura di memorizzazione attenua l'effetto della variazione del ritardo perché, entro certi limiti, pacchetti più veloci o i pacchetti più lenti del primo vengono comunque bufferizzati e mandati in playout correttamente. Contestualmente appare evidente anche un effetto negativo: i pacchetti messi in attesa ritardano forzatamente il momento del loro playout. Tornando all'esempio iniziale, il pacchetto 1 che era disponibile al tempo 1, viene in realtà ascoltato dall'utente al tempo 6.

Parametri

18

Parametri

- Riassumendo, i parametri per valutare la **qualità** di trasmissioni audio/video su IP sono i seguenti:
 - Il numero di **pacchetti persi** (packet loss)
 - Il **ritardo** tra il momento in cui la trasmissione parte e quello in cui viene visto/ascoltato il media (**playout delay**)
 - la variazione che questo ritardo può assumere a causa dell'imprevedibilità del comportamento della rete (**delay jitter**).

Riassumendo, i parametri per valutare la qualità di trasmissioni audio e video su rete Internet, sono i sostanzialmente tre. Il primo è il numero di pacchetti persi, detto viene detto anche packet loss. Il secondo è il ritardo tra il momento in cui la trasmissione parte, inizia, e quello in cui viene visto oppure ascoltato il media, ovvero il momento in cui viene fatto effettivamente il playout. Infine l'ultimo parametro, è la variazione che questo ritardo può assumere a causa dell'impredicibilità del comportamento della rete, che viene anche detto delay jitter. La misura di questi indici è significativa perché hanno un impatto sulla percezione che l'utente ha della qualità della fruizione e in specifico sulla qualità dello streaming. Va infatti ricordato che, se l'utente stesse operando con file scaricati, il playout sarebbe esente da tutti e tre questi tipi di fenomeni.

Percezione (1)

19

Percezione

- La **percezione** che l'utente ha della **qualità** dello **streaming** è legata a:
 - **Pacchetti persi**: può essere percepibile, come discontinuità il mancato arrivo del pacchetto.
 - **Ritardo**: ritardo iniziale di **buffering** o attesa per **rebuffering**.
 - **Jitter delay**: diminuzione della larghezza di banda che provoca l'esaurirsi del buffer. L'**host** ricevente riparte con il riempimento del buffer e l'utente ha una discontinuità pari al tempo di **rebuffering**.

La percezione che l'utente ha della qualità dello streaming è legata alle tre misure che abbiamo visto in modo diverso. La perdita di pacchetti viene percepita come una discontinuità del flusso. Se si tratta di singoli pacchetti il fenomeno è di solito poco rilevante e viene in parte corretto a livello di applicazione con vari meccanismi ad hoc che non abbiamo visto perché sono abbastanza complessi. Più difficile è invece recuperare da perdite multiple e consecutive che danno all'utente la sensazione di una interruzione molto lunga.

Percezione (2)

19

Percezione

- La **percezione** che l'utente ha della **qualità** dello streaming è legata a:
 - **Pacchetti persi**: può essere percepibile, come discontinuità il mancato arrivo del pacchetto.
 - **Ritardo**: ritardo iniziale di buffering o attesa per rebuffering.
 - **Jitter delay**: diminuzione della larghezza di banda che provoca l'esaurirsi del buffer. L'host ricevente riparte con il riempimento del buffer e l'utente ha una discontinuità pari al tempo di rebuffering.

La percezione del ritardo è dovuta a due fattori fondamentali: il ritardo accumulato che è accumulato dal primo pacchetto nella rete, e su questo non c'è modo di influire, e il ritardo di buffering, ovvero il tempo che viene introdotto per il riempimento del buffer. Questa seconda quantità, è introdotta artificialmente dal sistema, quindi potrebbe essere ridotta, ma non può essere troppo limitata perché si rischia che un buffer troppo piccolo sia inefficace nell'attenuare il delay jitter. La percezione del delay jitter, infine, è attenuata, nei limiti del possibile, dal buffer. Se il jitter però supera l'elasticità che viene introdotta dal buffer, per esempio a causa della netta diminuzione della larghezza di banda, allora il buffer si esaurisce. Il sistema ricevente riparte con il riempimento del buffer e l'utente si trova di fronte ad una forte discontinuità, che pari al tempo necessario per il rebuffering, cioè al tempo che serve al nuovo riempimento della memoria tampone.

Protocolli (1)

20

Protocolli

- Le applicazioni di streaming in ambito Web, sono tipicamente basate sui seguenti protocolli:
 - Protocolli per le trasmissioni multimediali real time che usano anche TCP, ma preferibilmente UDP: Real Time Protocol (RTP)/ Real Time Control Protocol (RTCP)
 - Un protocollo per il controllo dell'attività di streaming **Real Time Streaming Protocol (RTSP)** che fornisce:
 - ◆ Compressione e decompressione
 - ◆ Rimozione del Jitter delay
 - ◆ Correzione degli errori

Le applicazioni che utilizzano lo streaming e altre tecniche di buffering sono numerose e differenti in termini di caratteristiche basilari. Sono dunque numerosi anche i protocolli che sono usati in questi ambiti applicativi e questi protocolli dipendono fortemente dal tipo di applicazione che utilizza lo streaming per il trasporto dei dati multimediali. Non questa è lezione la sede giusta per descrivere una realtà applicativa molto complessa in cui architetture, applicazioni, stack di protocolli sono usati con finalità molto varie. Ci limiteremo invece ad una trattazione, piuttosto semplice, dei protocolli che sono usati in ambito Web per lo streaming audio e video.

Protocolli (2)

20


Protocolli

- Le applicazioni di streaming in ambito Web, sono tipicamente basate sui seguenti protocolli:
 - Protocolli per le trasmissioni multimediali real time che usano anche TCP, ma preferibilmente UDP: Real Time Protocol (RTP)/ Real Time Control Protocol (RTCP)
 - Un protocollo per il controllo dell'attività di streaming **Real Time Streaming Protocol (RTSP)** che fornisce:
 - ◆ Compressione e decompressione
 - ◆ Rimozione del Jitter delay
 - ◆ Correzione degli errori

I protocolli utilizzati in questo contesto in modo significativo, sono sostanzialmente due: Il primo è

un protocollo che viene usato per la trasmissione del flusso multimediale e che tipicamente poggia su UDP. Questo protocollo in realtà è definito attraverso una coppia di protocolli che si chiamano rispettivamente RTP (Real Time Protocol) e RTCP (Real Time Control Protocol) che servono alla trasmissione del flusso dei dati multimediali e al suo controllo. Entrambi i protocolli sono definiti ovviamente dall'IETF e fanno parte della suite di protocolli TCP/IP. Un secondo protocollo è quello dedicato alla vera e propria attività di streaming, si chiama RTSP (Real Time Streaming Protocol) e fa sempre parte della suite TCP/IP. RTSP fornisce i meccanismi per la compressione e la decompressione, per la rimozione del delay jitter e per la correzione degli errori.

RTSP



21

RTSP

- Alcune caratteristiche di RTSP sono:
 - Supporto per il controllo dell'erogazione in streaming del flusso da parte dell'applicazione (tra client e server).
 - Supporto per il controllo del flusso da parte dell'utente: *rewind*, *fast forward*, *pause*, *resume*, etc.
 - Protocollo di controllo "fuori banda": vengono in realtà usate due connessioni una per lo stream multimediale e una per il controllo.

Elenchiamo brevemente le principali funzionalità offerte da RTSP. In primo luogo RTSP offre il supporto per il controllo dell'erogazione in streaming del flusso, sia da parte dell'applicazione (cioè tra client e server), sia da parte dell'utente. Quest'ultimo in particolare avviene attraverso comandi che rispecchiano le tradizionali meccaniche con cui si interagisce con i sistemi di riproduzione di audio e video, come per esempio la pausa, il resume, il rewind e così via. Inoltre RTSP opera anche utilizzando un protocollo di controllo "fuori banda", e vengono quindi in realtà usate due connessioni una per lo stream multimediale e una per il controllo del flusso dello stream multimediale.

Meta File (1)

22

Meta File

- La richiesta avviene tramite il Web, ovvero come richiesta HTTP da un browser ad un HTTP server, che avvia la comunicazione di streaming tra client e server:
 - Il Web browser richiede una risorsa che in realtà è un **meta file**, ovvero un file che descrive il media di cui verrà fatto lo streaming.
 - Il Server HTTP risponde inviandolo.
 - Il Browser lancia l'appropriato Player e gli passa il metafile.
 - Il Player seguendo le indicazioni del Metafile avvia lo streaming.

La richiesta di attivazione di una connessione di tipo RTSP avviene tipicamente attraverso il Web. L'utente naviga ed entra in una pagina che contiene dei riferimenti ad oggetti multimediali che verranno poi forniti dal sistema in modalità di streaming. Questi riferimenti vengono attivati dall'utente con un click e in quel momento parte una richiesta dal browser, ovvero dal client HTTP con cui sta interagendo l'utente, verso il server HTTP. Questa richiesta è in realtà riferita a un metafile, ovvero un file che descrive il media di cui verrà fatto lo streaming. Il Server HTTP risponde inviando il metafile al browser.

Meta File (2)

22

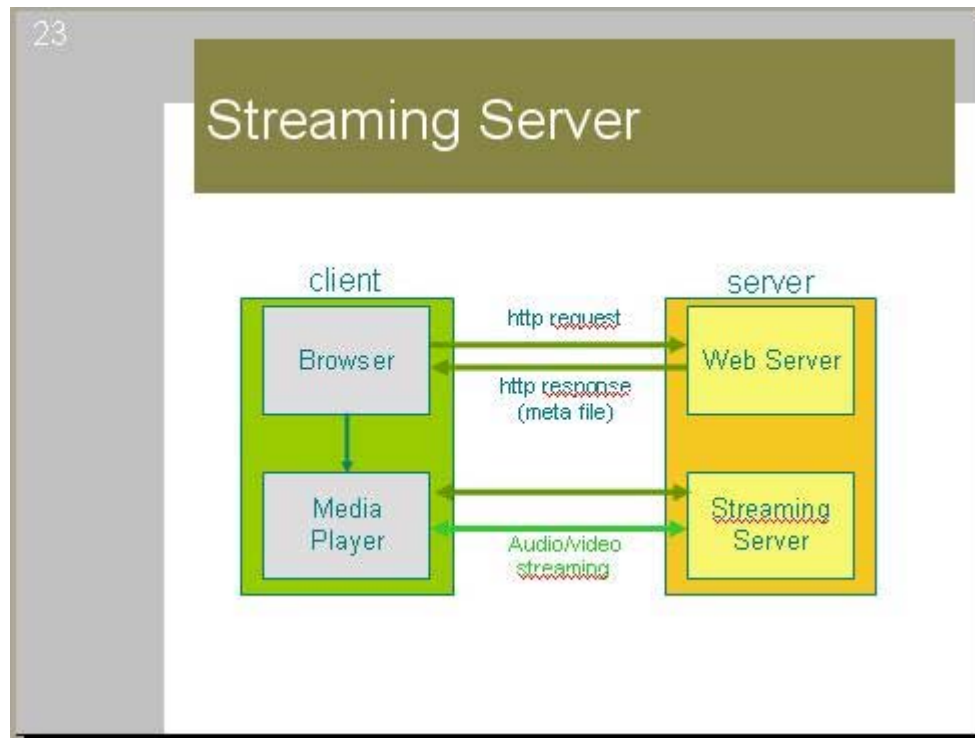
Meta File

- La richiesta avviene tramite il Web, ovvero come richiesta HTTP da un browser ad un HTTP server, che avvia la comunicazione di streaming tra client e server:
 - Il Web browser richiede una risorsa che in realtà è un **meta file**, ovvero un file che descrive il media di cui verrà fatto lo streaming.
 - Il Server HTTP risponde inviandolo.
 - Il Browser lancia l'appropriato Player e gli passa il metafile.
 - Il Player seguendo le indicazioni del Metafile avvia lo streaming.

Il browser delega il controllo al player, che è il software deputato al playout del media ma anche a

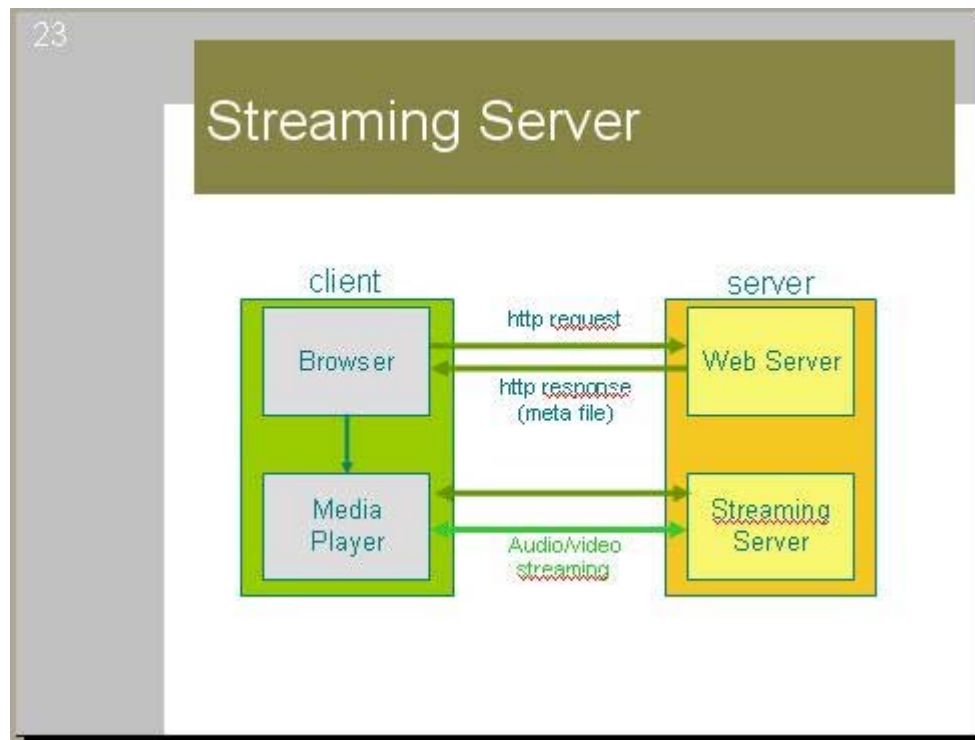
fornire il supporto al lato client allo streaming di tipo RTSP. Il browser passa al player il metafile, in modo che seguendo le indicazioni che sono specificate attraverso il metafile, possa essere avviato lo streaming, cosa che avviene contattando l'opportuno server RTSP e richiedendo la risorsa da mandare in playout.

Streaming Server (1)



Questa slide mostra l'interazione tra l'applicazione Web e l'applicazione di streaming. La piattaforma client (che è visualizzata sulla sinistra della slide) ospita sia il client HTTP, cioè il browser, che il client RTSP che si occuperà del playout del media continuo (audio o video che sia). La piattaforma server, parallelamente (è quella disegnata sulla destra della slide) ospita sia il server HTTP, che è indicato come server Web, sia il server RTSP, che è indicato come server di streaming. Dal lato server l'architettura è semplificata perché in realtà il server Web e il server di streaming potrebbero risiedere su due macchine differenti e con due diversi indirizzi Internet. L'URL della risorsa multimediale, distribuita dal server Web, consente all'applicativo client di individuare il server di streaming, indipendentemente da dove esso sia situato sulla rete.

Streaming Server (2)



La sessione inizia dal lato client con il browser che richiede una risorsa multimediale. Il server risponde a questa richiesta con un metafile che viene inviato indietro al browser. Il browser trova nel metafile le indicazioni su quale specifico player multimediale deve essere invocato per gestire lo streaming. Il browser lancia quindi il player e gli passa il metafile sulla base del quale il player contatta il rispettivo server e inizia contemporaneamente lo streaming e la corrispondente sessione di controllo.

Applicazioni (1)

24

Applicazioni

- I più diffusi sistemi di streaming, utilizzabili in ambito Web, sono:
 - Apple Quicktime (Darwin Streaming Server)
 - ◆ <http://www.apple.com/it/quicktime/>
 - Microsoft Windows Media Technologies
 - ◆ <http://www.microsoft.com/windows/windowsmedia/>
 - RealNetworks
 - ◆ <http://www.realnetworks.com>

A tutt'oggi i più diffusi sistemi di streaming sono caratterizzati dai seguenti marchi: il primo è Apple Quicktime, che è un marchio Apple, appunto, utilizzato per una suite di prodotti che include un

formato multimediale, diversi player multimediali con il supporto per lo streaming e alcune piattaforme server, di cui una che è di tipo open source. Maggiori informazioni sulla suite sono reperibili attraverso la URL che è riportata nella slide. La seconda piattaforma, il secondo insieme di tecnologie è Microsoft Windows Media Technologies. Esiste ovviamente una famiglia di tecnologie prodotte da Microsoft per la fruizione di multimedia, sia in streaming che non. Sono fornite con il sistema operativo, rispettivamente client per il player e server per i servizi di streaming. Maggiori informazioni sui prodotti Microsoft Media sono reperibili attraverso l'URL riportato nella slide.

Applicazioni (2)

24

Applicazioni

- I più diffusi sistemi di streaming, utilizzabili in ambito Web, sono:
 - Apple Quicktime (Darwin Streaming Server)
 - ◆ <http://www.apple.com/it/quicktime/>
 - Microsoft Windows Media Technologies
 - ◆ <http://www.microsoft.com/windows/windowsmedia/>
 - RealNetworks
 - ◆ <http://www.realnetworks.com>

Ultimo, ma solo in ordine alfabetico, è RealNetworks, un produttore che offre numerose soluzioni per produrre e distribuire multimedia. In particolare esistono diverse versioni del player e del server, tra cui anche una ad-hoc per le piattaforme mobili. Maggiori informazioni sulle soluzioni Real sono reperibili di nuovo attraverso l'URL riportato nella slide. Attraverso gli indirizzi riportati nella slide possono anche essere scaricati i player, che tipicamente sono sempre gratuiti e in alcuni casi sono gratuite o sono disponibili comunque anche alcune versioni dei server. Tutti questi sistemi sono attualmente molto diffusi per cui è frequente che sulle postazioni ad uso personale, ovviamente compatibilmente con la piattaforma e con il sistema operativo, siano installati tutti e tre i player.

Applicazioni (3)

25

Applicazioni

- I sistemi disponibili hanno caratteristiche simili:
 - Streaming audio e video,
 - Adattività alla larghezza di banda disponibile,
 - Scalabilità rispetto al numero di client collegati,
 - Alcune migliaia di stream contemporanei
- Spesso, ogni sistema ha il suo formato per lo stream (per cui non esiste interoperabilità).

I sistemi disponibili hanno caratteristiche simili. Offrono sia streaming audio che streaming video e sono in grado di adattare la qualità del flusso alla larghezza di banda a disposizione. Questo significa che il server conserva più stream corrispondenti alla stessa sorgente multimediale e poi si accorda col player su quale versione mandare in streaming, in considerazione della larghezza di banda che è a disposizione di quello specifico client. Tipicamente i server sono scalabili rispetto al numero di client collegati e sopportano, ovviamente questo compatibilmente con le caratteristiche della rete e dell'hardware che li ospita, anche diverse migliaia di connessioni contemporanee. Tutti i server citati utilizzano protocolli standard per le parti fondamentali dell'applicazione ma tipicamente, ogni sistema ha il suo formato per lo stream, per cui non esiste una vera interoperabilità e se un certo contenuto è offerto da uno specifico server, l'utente deve munirsi dell'apposito player per poterne fruire.

Applicazioni (4)

25

Applicazioni

- I sistemi disponibili hanno caratteristiche simili:
 - Streaming audio e video,
 - Adattività alla larghezza di banda disponibile,
 - Scalabilità rispetto al numero di client collegati,
 - Alcune migliaia di stream contemporanei
- Spesso, ogni sistema ha il suo formato per lo stream (per cui non esiste interoperabilità).

Infine i server supportano sia comunicazioni unicast, tradizionali di tipo 1 a 1, che comunicazioni multicast di tipo 1 a molti.

Riferimenti

26

Riferimenti

- N. P. Chapman, J. Chapman, *Digital Multimedia*, John Wiley & Sons.
- R. Steinmetz, K. Nahrstedt, *Multimedia Fundamentals, Volume I: Media Coding and Content Processing*, Prentice Hall PTR.
- Fred Halsall, *Multimedia Communications*, Addison-Wesley, 2001
- Nalin K. Sharda: *Multimedia Information Networking*, Prentice Hall, 1999.

Lo scopo della lezione è stato quello di illustrare i principali meccanismi alla base dello streaming di flussi multimediali su Internet, e siamo arrivati quindi fino a introdurre brevemente i protocolli e le principali applicazioni che si utilizzano per la fruizione di stream multimediali integrati all'interno di pagine Web. L'argomento "streaming" è stato trattato in questa sede in modo molto rapido, ed è possibile approfondirlo attraverso i testi elencati in questa slide.