Sistemi Ipermediali
Multimedia in Rete I

Claudio Palazzi
Università degli Studi di Padova

Lucidi a cura di: Marco Roccetti, Paola Salomoni, Stefano Ferretti, Claudio Palazzi

Multimedia su Internet

- I criteri di classificazione per le applicazioni multimediali (operanti su Internet) sono fondamentalmente i seguenti:
  - ✓ Hard Real time / Soft Real Time/ Non real time
  - ✓ Sincrono (interattivo) / Asincrono
  - ✓ Unicast / Multicast / Broadcast
  - ✓ Unidirezionale /Bidirezionale

Classi di applicazioni multimediali su Internet

1. Distribuzione Audio/video non real-time: comunicazione non real time che ha come scopo il semplice delivery asincrono di file multimediali

2. Audio/video on demand: comunicazione real time non interattiva che ha come scopo il delivery di stream multimediali

3. Internet TV/radio (Live Streaming)
   - comunicazione asincrona, unidirezionale, 1 a molti

4. Internet Telephony
   - comunicazione sincrona, bidirezionale, 1 a 1

5. Video Conferenza
   - comunicazione sincrona, bidirezionale, molti a molti

1. Distribuzione Audio/Video non real-time

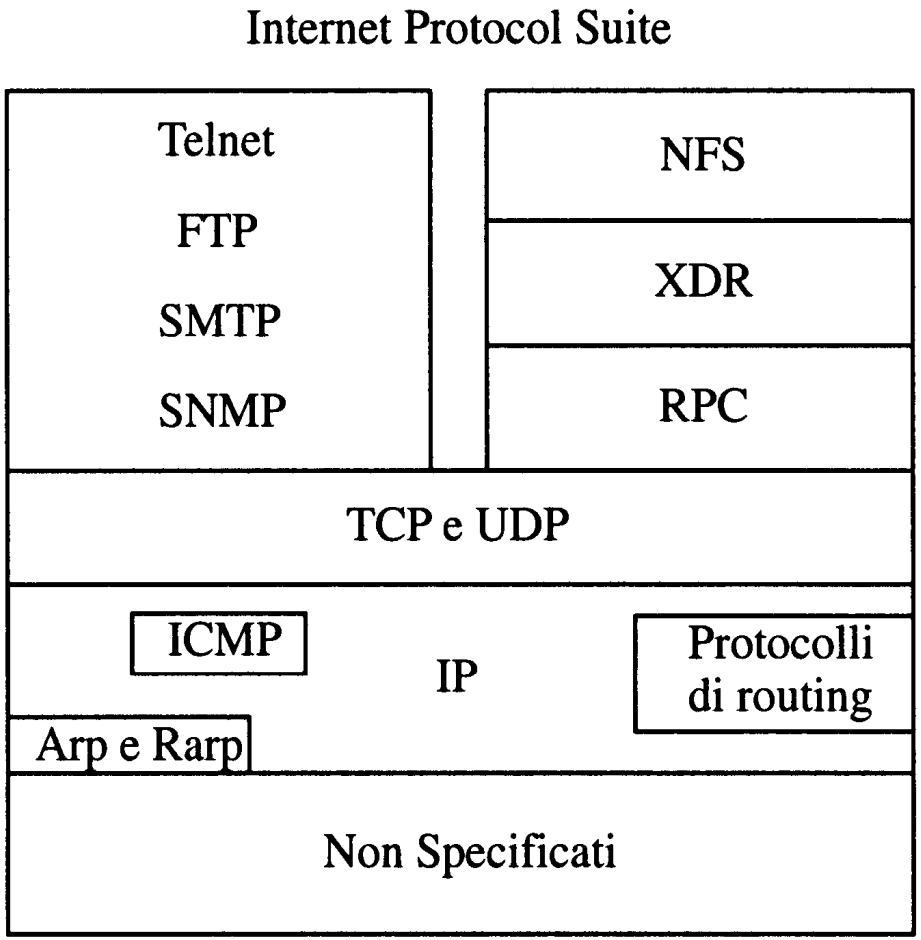
- Basato su protocollo IP che definisce il livello di rete di Internet
  - Non orientato alla connessione: ogni pacchetto contiene l'indirizzo di partenza e di destinazione e può seguire un percorso diverso (*routing*)
  - Non affidabile (*best effort*): ogni singolo pacchetto può non arrivare, o arrivare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

Livello di trasporto

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

TCP/IP

il livello di trasporto del TCP/IP



Numerosità delle destinazioni

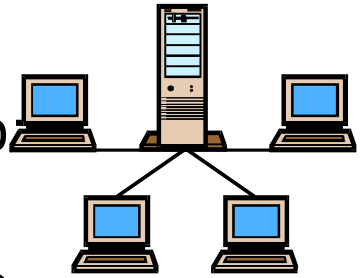
- Unicast: trasmissione in cui è individuato un *unico* host destinazione
- Multicast: trasmissione in cui è individuato un *gruppo* di host destinazione
- Broadcast: trasmissione in cui gli host destinazione sono *tutti* i nodi raggiungibili di una rete

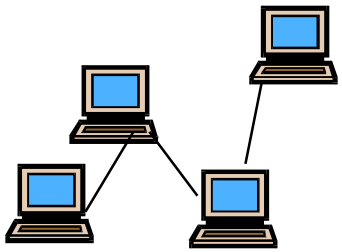
Multicast

- Multicast è un metodo per la distribuzione a un vasto numero di host di singoli pacchetti di dati.
  - I dati sono trasmessi *una volta soltanto* dall'host sorgente verso gruppi di destinazioni multiple, che possono essere connesse a differenti reti
  - Il gruppo *multicast* è tipicamente dinamico ovvero accetta entrate ed uscite in qualunque momento
  - La dinamicità permette di continuare ad operare anche con interruzioni di rete

Client/Server e P2P

- Le applicazioni di rete possono essere basate su due modelli architetturali diversi:
  - Nel modello client/server un programma (client) genera una richiesta ed un altro programma (server) gli risponde (esempio: Web)
  - Nel modello Peer to Peer (P2P) non esiste sistema fornitore di servizi "centrale" ma ogni partner nella comunicazione ha ruoli e funzioni equivalenti (esempio: Napster)





Distribuzione di risorse MM

- In tempi recenti si sono diffusi in modo capillare sistemi che consentono di distribuire/reperire risorse multimediali su Internet
- Il più famoso è Napster, un sistema per il reperimento di file musicali in formato MP3
- In questo tipo di applicazione, l'aspetto interessante della comunicazione non riguarda tanto il trasferimento finale della risorsa, quanto il suo reperimento

Trasferimento della risorsa

- In effetti, la trasmissione della risorsa avviene come trasferimento di semplici file:
  - per cominciare a *sentire* il file occorre che esso sia stato del tutto scaricato
  - Il file viene trasmesso solitamente con FTP o HTTP (e quindi sulla base di TCP, con controllo d'ordine e consistenza)
  - I file sono di dimensioni notevoli, quindi, tipicamente i tempi di attesa prima dell'ascolto sono lunghi

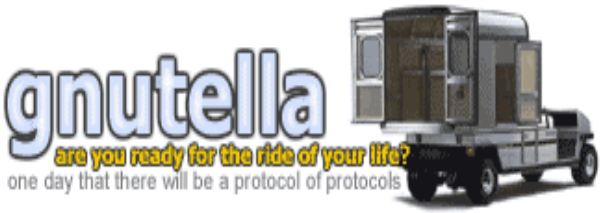
Distribuzione delle risorse

- Questo genere di applicazioni è tipicamente basato su architetture P2P: le risorse (i file MP3) sono distribuite sui diversi *peer* che partecipano al sistema
- Questo tipo di architettura ha diversi vantaggi:
  - Memorizzazione (e quindi richiesta di risorse) distribuita
  - Ridondanza: più peer possono offrire la stessa risorsa
  - Efficacia della reperibilità: e.g., tra i peer che offrono la risorsa, scelgo uno di quelli con tempi di risposta minimi

Distribuzione di risorse MM

- Napster

- Gnutella



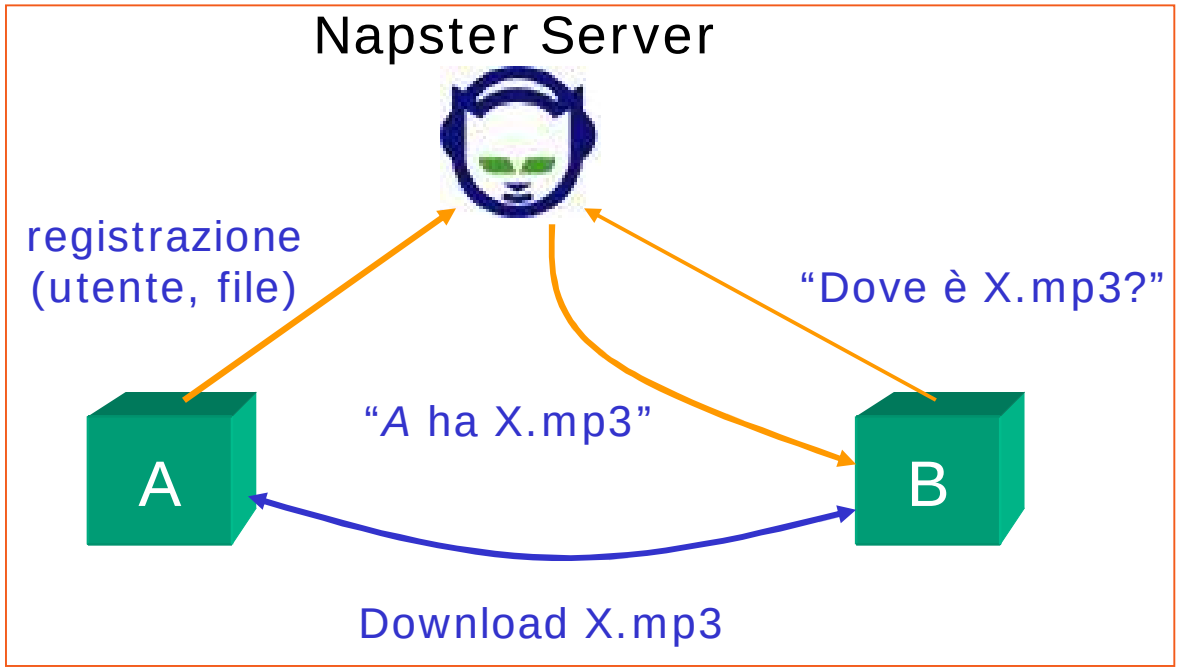
- KaZaA / Skype

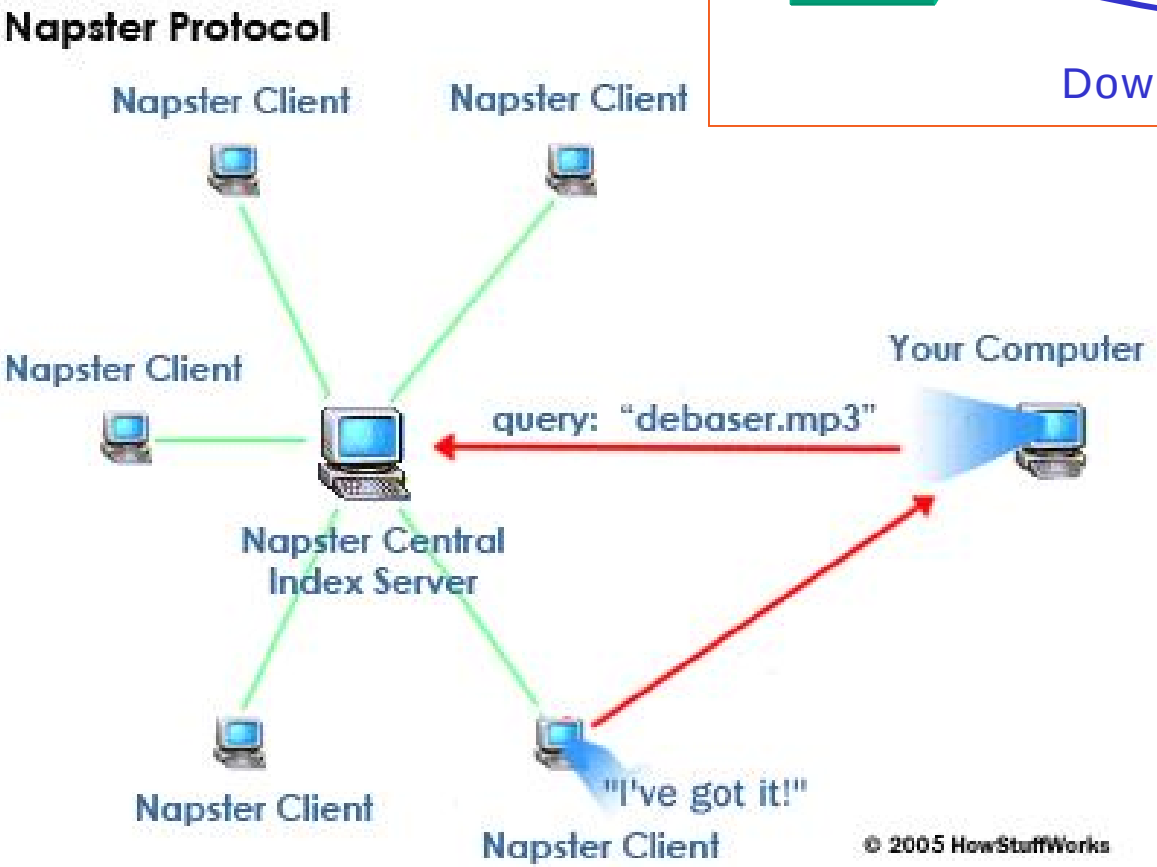
- Freenet

Napster

- Architettura mista P2P e client/server:
  - C/S: Un DB centrale che gestisce l'indice dei pezzi (MP3 e WMA) disponibili
  - C/S: I client si connettono al server e comunicano l'elenco dei file che mettono a disposizione
  - C/S: altri client connessi al server possono individuare le risorse che sono state condivise e successivamente ….
  - P2P: aprire, con altri client che mettono a disposizione tali risorse, una connessione tra pari per permetterne lo scaricamento

Napster

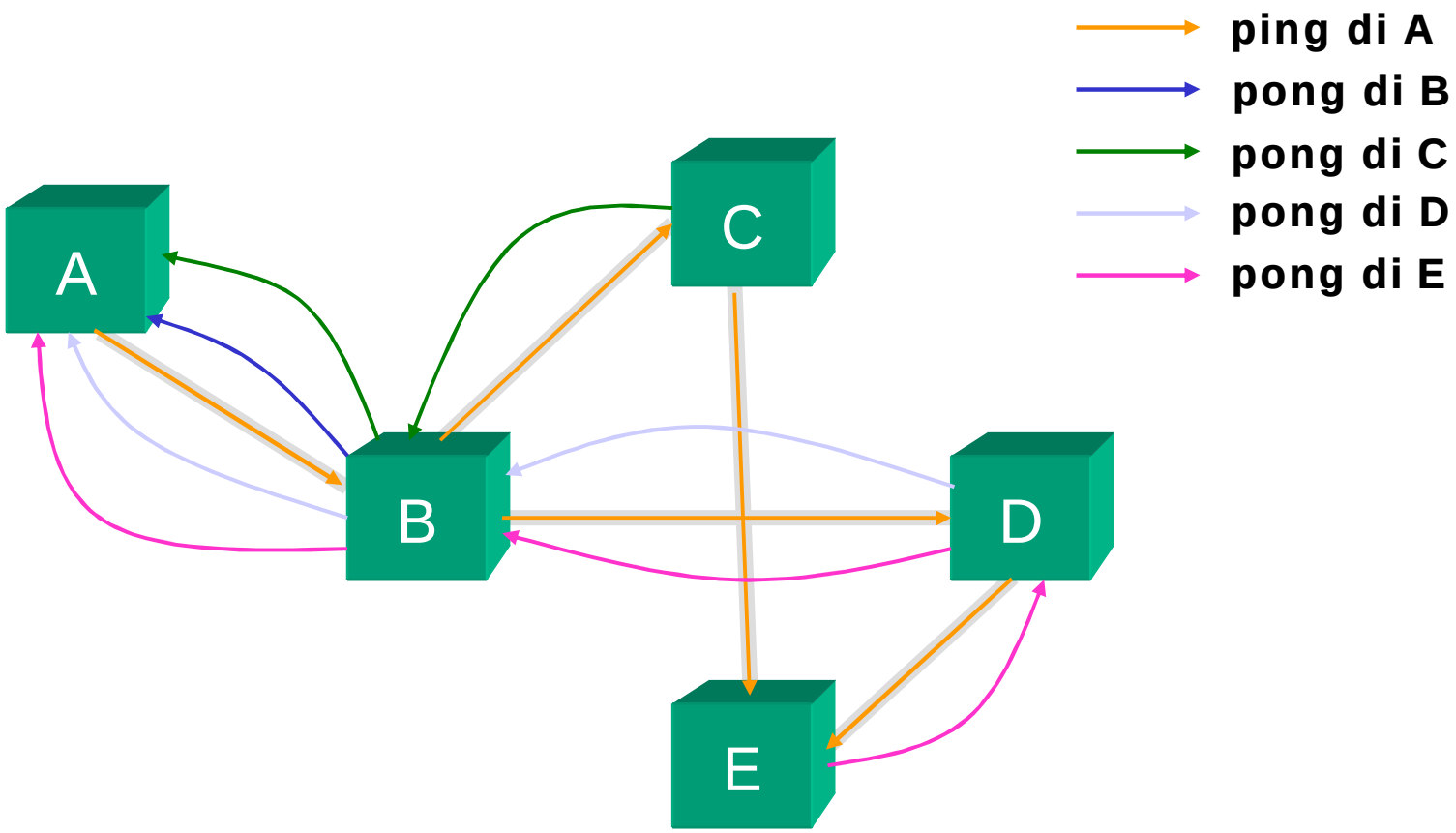

Napster Server
registrazione
(utente, file)
"Dove è X.mp3?"
"A ha X.mp3"
A
B
Download X.mp3


Napster Protocol
Napster Client
Napster Client
Napster Client
Your Computer
query: "debaser.mp3"
Napster Central
Index Server
Napster Client
"I've got it!"
Napster Client
© 2005 HowStuffWorks

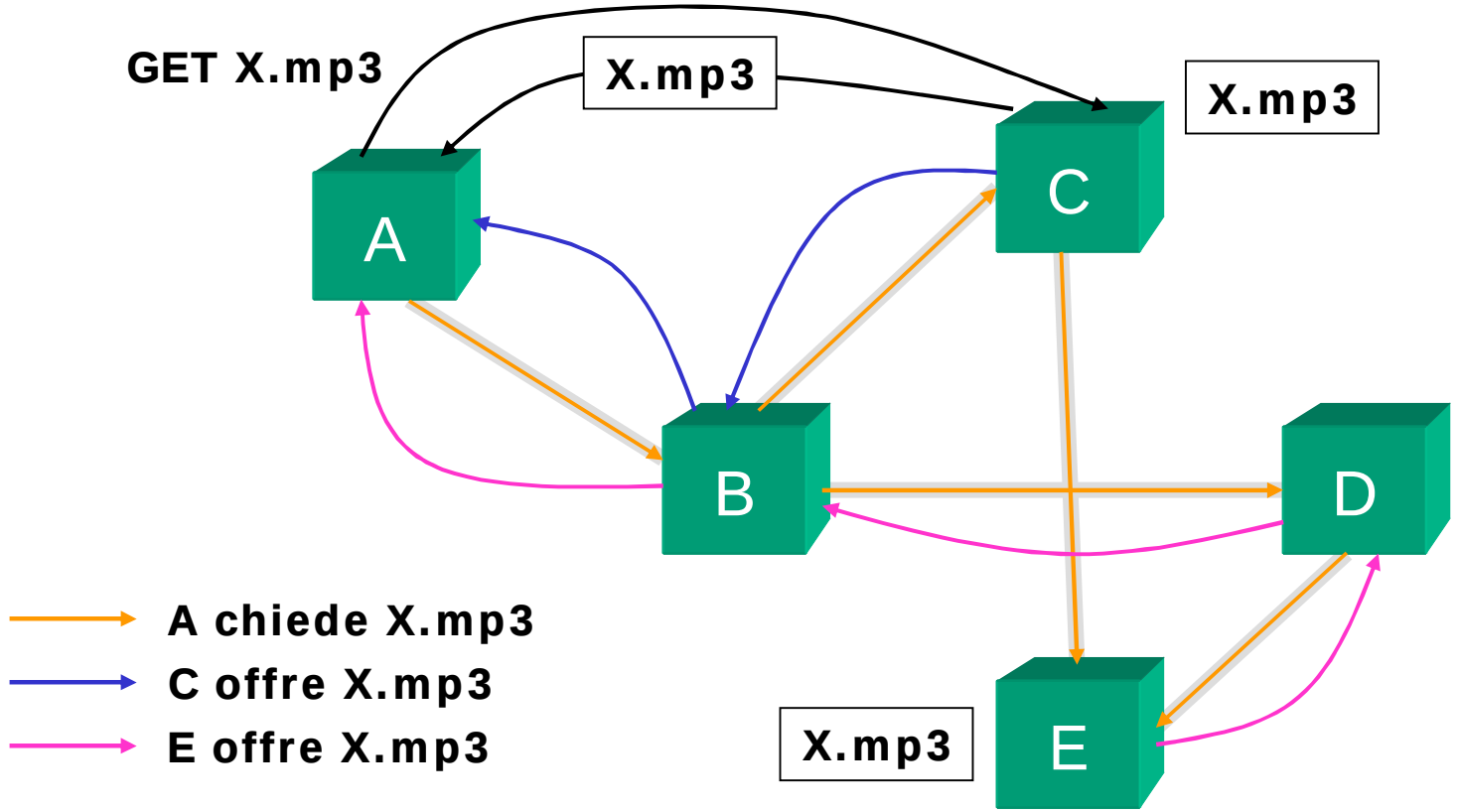
Gnutella

- Caratteristiche salienti dell'architettura P2P:
  - non c'è server centrale
  - si effettua un *broadcast limitato* per individuare i pari con cui dialogare (ping/pong)
  - ogni peer invia tutti i pacchetti che riceve ad ognuno dei suoi pari (tipicamente 4) – constrained broadcast
  - Il pacchetto ha un tempo di vita limitato (tipicamente 7 passi) e un numero univoco di identificazione per evitare cicli
- Dalla versione 0.6 introduzione dei superpeers
  - Leafnodes si connettono a superpeers
  - Ogni superpeer ha dai 10 ai 100 leafnodes
  - Superpeer autoeletti fra nodi con migliori connessioni
- Concetto superpeers ripreso da Kazaa
  - Che però usa protocollo FastTrack e traffico criptato

Ping/Pong

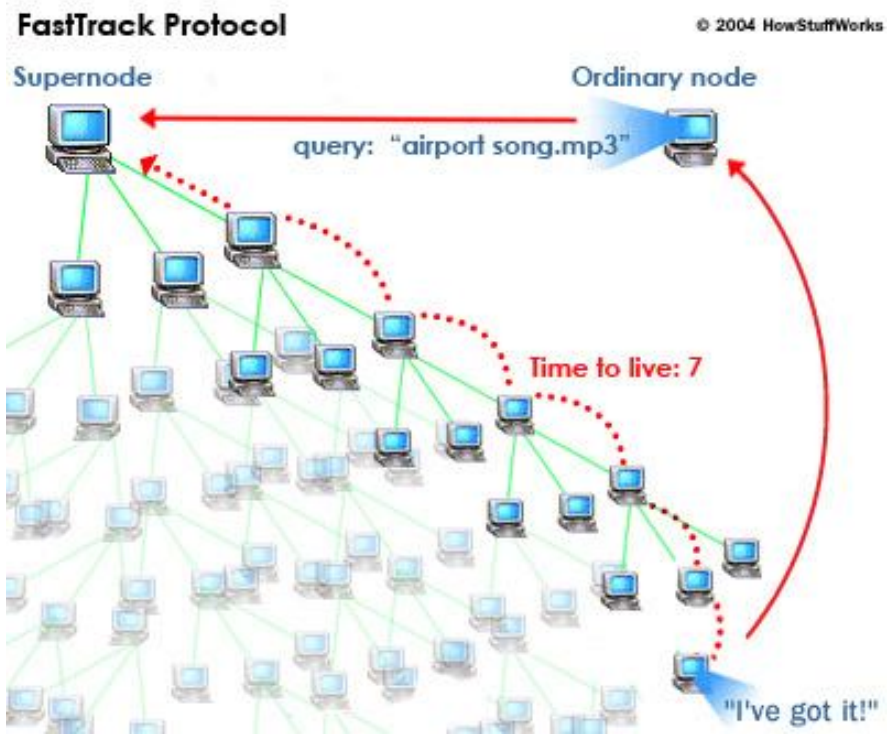


Download


GET X.mp3
X.mp3
X.mp3
A
C
B
D
E
X.mp3
A chiede X.mp3
C offre X.mp3
E offre X.mp3

KaZaA

- Rete P2P di seconda generazione (supernodes)
- Problemi legali (copyright) cominciano ad arrivare



- Usato a supporto di SKYPE
- Funziona meglio se ci sono tanti utenti
  - Skype inizialmente gratuito
  - Poi a pagamento solo per chiamate da/a telefoni
  - Ora si sta diffondendo su smartphone
  - eBay comprò per 2.6G$
  - Microsoft comprato per 8.5G$
    - ✓ 700M utenti
    - ✓ 12$ a utente

Freenet

- Architettura P2P:
  - non c'è server centrale
  - Non c'è *constrained broadcast* come in Gnutella
  - Il sistema di pubblicazione, replicazione e reperimento è *adattivo*
    - ✓ informazioni vengano crittografate e replicate su molti diversi nodi in continuo cambiamento in tutto il mondo
  - I riferimenti ai file sono creati e gestiti in modo indipendente dalla loro localizzazione fisica
  - I dati sono replicati dinamicamente

Freenet

- Il sistema protegge l'anonimato di autori e lettori:
  - È impossibile determinare l'origine e la destinazioni dei dati che passano attraverso un pari.
  - Per il pari è difficile determinare cosa sta memorizzando e mettendo a disposizione
    - ✓ Riparo da reati di detenzione
- La struttura del grafo di pari evolve dinamicamente nel tempo:
  - Si formano nuovi link tra i pari
  - I file migrano da un pari all'altro
  - Il routing è adattivo

2. Audio/Video on Demand

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

Streaming

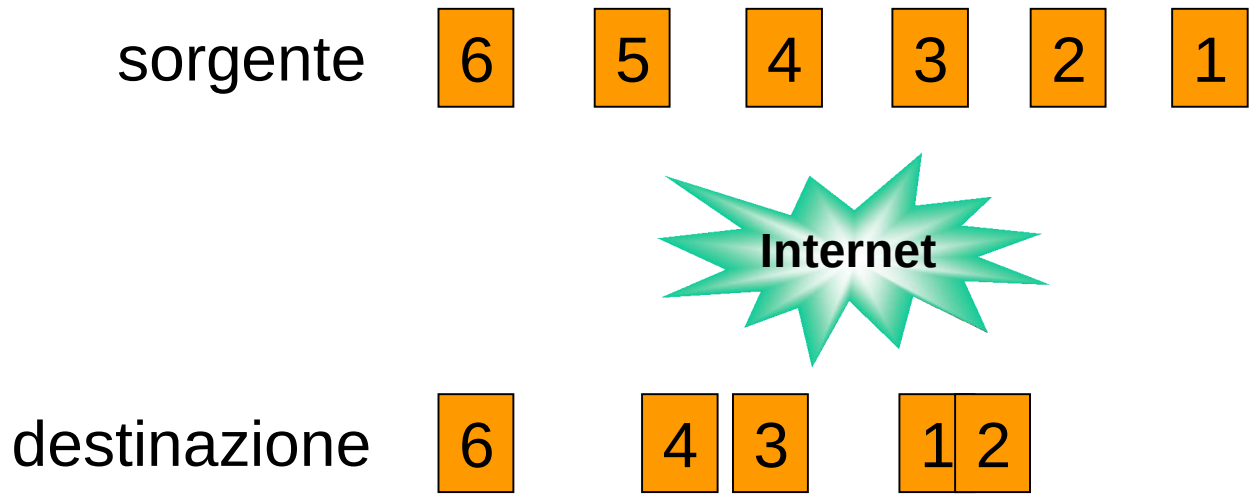
- In questi i casi, si usa una tecnica nota come streaming che opera considerando il media come stream (flusso) e non come file
- L'host ricevente comincia il playout del media (audio o video) prima che sia conclusa la comunicazione con l'host trasmittente
- Il tempo di attesa si può ridurre:
  - Quanto?
  - Come?

Streaming

- Tipicamente i sistemi di streaming operano su UDP (cioè il protocollo di trasporto che NON controlla l'integrità della trasmissione, nè 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

Ordine

- Se il controllo di ordine ed integrità non è effettuato a livello di trasporto, deve essere effettuato a livello applicazione



Jitter

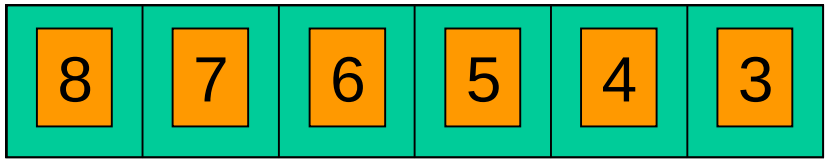
- Il disordine che si produce presso la stazione ricevente è causato dal fatto che ciascun pacchetto impiega un tempo diverso a raggiungere la destinazione
- Questa variabilità del ritardo prodotto dalla trasmissione è detta delay jitter (variazione del ritardo)
  - Formula jitter secondo RFC 1889: $D(j, i) = |(Rj - Sj) - (Ri - Si)|$

Streaming

- Il meccanismo di base dello streaming si può riassumere 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

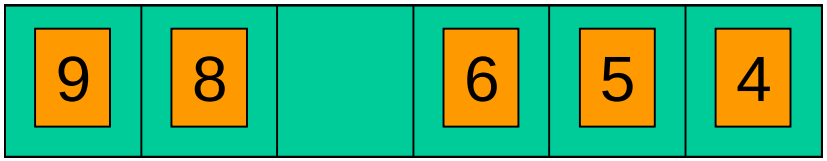
Buffer

- 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 di tipo FIFO (First In, First Out) che viene riempita da un lato e svuotata dall'altro



Buffer

- In realtà la struttura non si riempirà in modo così regolare
  - Perché i pacchetti possono arrivare disordinatamente
  - Perché alcuni pacchetti non arrivano affatto



Playout di 3

Packet Loss

- Nell'esempio precedente un pacchetto (2) non è arrivato in tempo per essere ascoltato
- Questo tipo di errore, indicato con packet loss, genera un interruzione del segnale audio
- 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

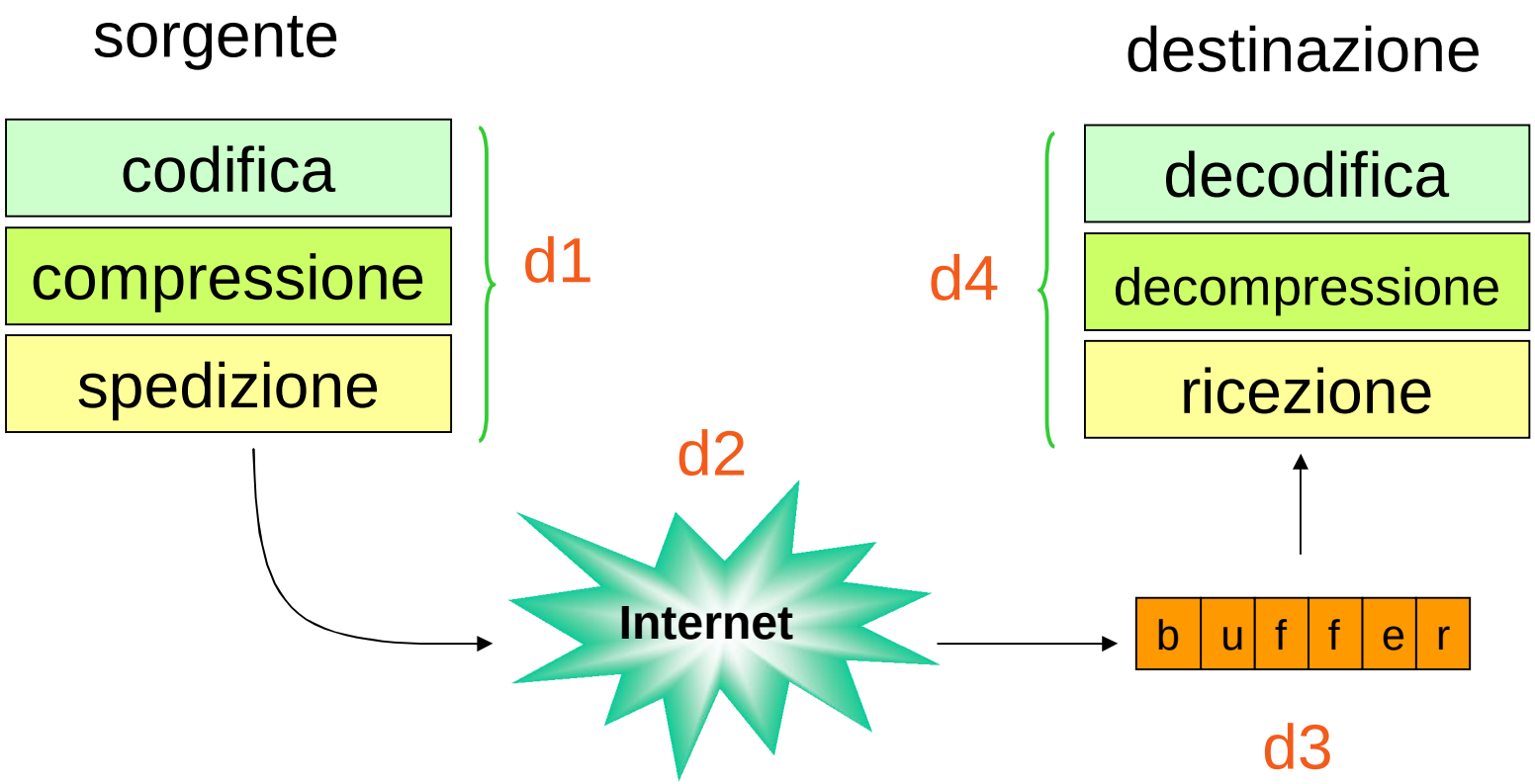
Packet Loss

- 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

Buffer

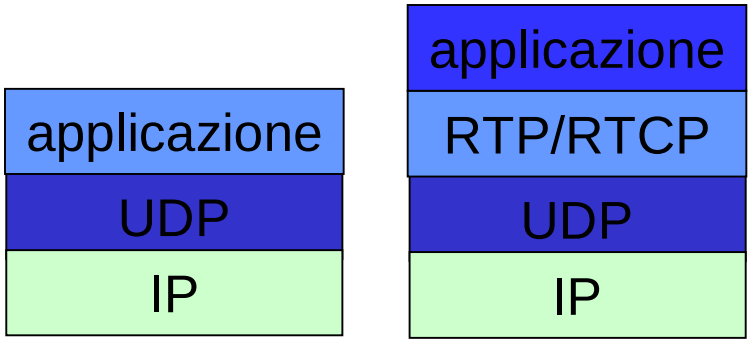
- La struttura di memorizzazione introdotta induce sulla trasmissione alcuni vantaggi:
  - Effettua automaticamente il riordino
  - Tenta di compensare e gestire il delay jitter
- Contestualmente appare evidente un effetto negativo: i pacchetti messi in attesa ritardano forzatamente il momento del loro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

Ritardo


sorgente
destinazione
codifica
compressione
spedizione
d1
d4
decodifica
decompressione
ricezione
d2
Internet
b u f f e r
d3

Ritardo

- d1: ritardo predicibile prodotto dall'host sorgente (processing delay):
  - la scheda audio campiona
  - il flusso viene compresso e
  - impacchettato (nell'audio circa 1 pk ogni 20/30 ms)
- d4 è simmetrico



Ritardo

- d2 : ritardo, impredicibile, introdotto dalla rete Internet (network delay) a causa di:
  - commutazione di pacchetto
  - best effort (sull'Internet attuale non è garantito nessun altro tipo di QoS)
- d3 : ritardo predicibile (o meglio regolabile dall'applicazione) a livello del buffer

Playout Delay

- Il *playout delay* (o ritardo di presentazione) è dato:
  - dal ritardo predicibile ma fisso accumulato nella digitalizzazione e nell'impacchettamento (e simmetricamente dallo spacchettamento e dalla riproduzione) d1+d4
  - dal ritardo predicibile e modulabile accumulato nel buffer d3 (la lunghezza del buffer può essere specificata dal software)
  - dal ritardo impredicibile dato dalla rete d2

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'impredicibilità del comportamento della rete (delay jitter) complica le cose ….

Influenza del Buffer

- La tecnica di buffering influenza il comportamento dell'applicazione nel modo seguente:
  - Più grande è il buffer:
    - ✓ più aumenta il playout delay
    - ✓ più diminuiscono i pacchetti persi (2 faceva in tempo ad arrivare)
  - Più piccolo è il buffer:
    - ✓ Più diminuisce il playout delay
    - ✓ Più aumentano i pacchetti persi
- NB: pacchetti persi o arrivati troppo tardi, sono la stessa cosa!

3. Live Streaming

- Alcune applicazioni aggiungono alle caratteristiche viste in precedenza la necessità di campionare, produrre e distribuire il media in maniera live
- Il media non è quindi già disponibile, ma viene prodotto o (emesso on line) dal *mondo reale*, per consentire l'accesso ad un evento in corso:
  - Internet Radio e Internet TV
  - VideoConferenza

Protocolli per live streaming

- Le applicazioni di streaming sono tipicamente basate sui seguenti protocolli:
  - Protocolli per le trasmissioni multimediali real time che sfruttano TCP, ma preferibilmente UDP: Real Time Protocol (RTP)/ Real Time Control Protocol (RTCP) - RFC 1889
  - Un protocollo per il controllo dell'attività di streaming Real Time Streaming Protocol (RTSP - RFC 2326) che fornisce:
    - ✓ Compressione e decompressione
    - ✓ Rimozione del Jitter
    - ✓ Correzione degli errori

RTSP: Caratteristiche del Protocollo e Funzionalità

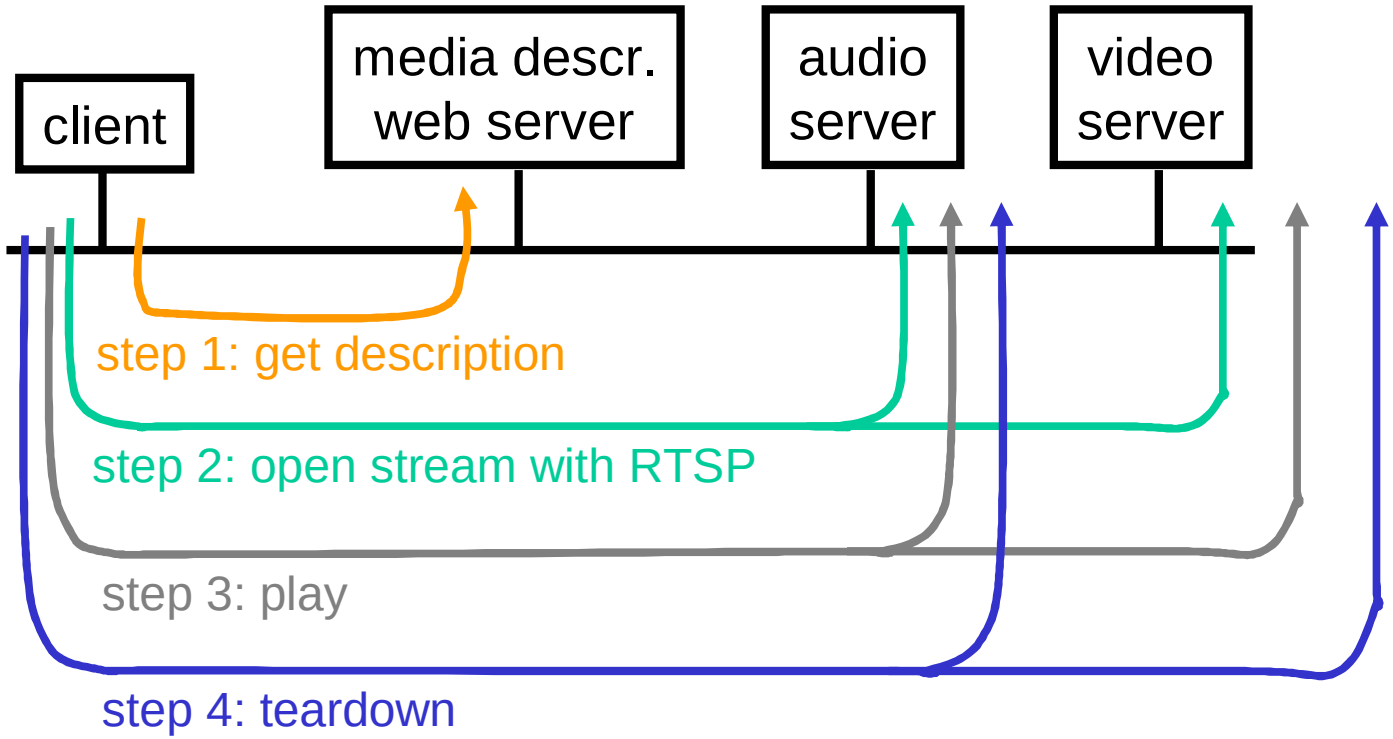
- Protocollo text-based
- Protocollo transport independent
- Simile all'HTTP con numerose differenze:
  - Richieste sia di tipo client → server che di tipo server → client
  - Il server mantiene lo stato della sessione
  - Opera sia unicast che multicast
- Supporto per il controllo dell'erogazione in 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.
- Gestione delle conferenze: invito, registrazione e rilascio della conferenza

RTSP: Principali METODI

- **SETUP:** il server alloca le risorse per lo stream e inizia una sessione RTSP
- **PLAY**: inizia la trasmissione dello stream
- **RECORD**: inizia la registrazione di uno stream
- **PAUSE**: sospende temporanemente la trasmissione dello stream
- **SET_PARAMETER**: cambia la codifica (device, compressione, ecc.)
- **TEARDOWN**: libera le risorse allocate per lo stream e conclude la sessione RTSP

Esempio

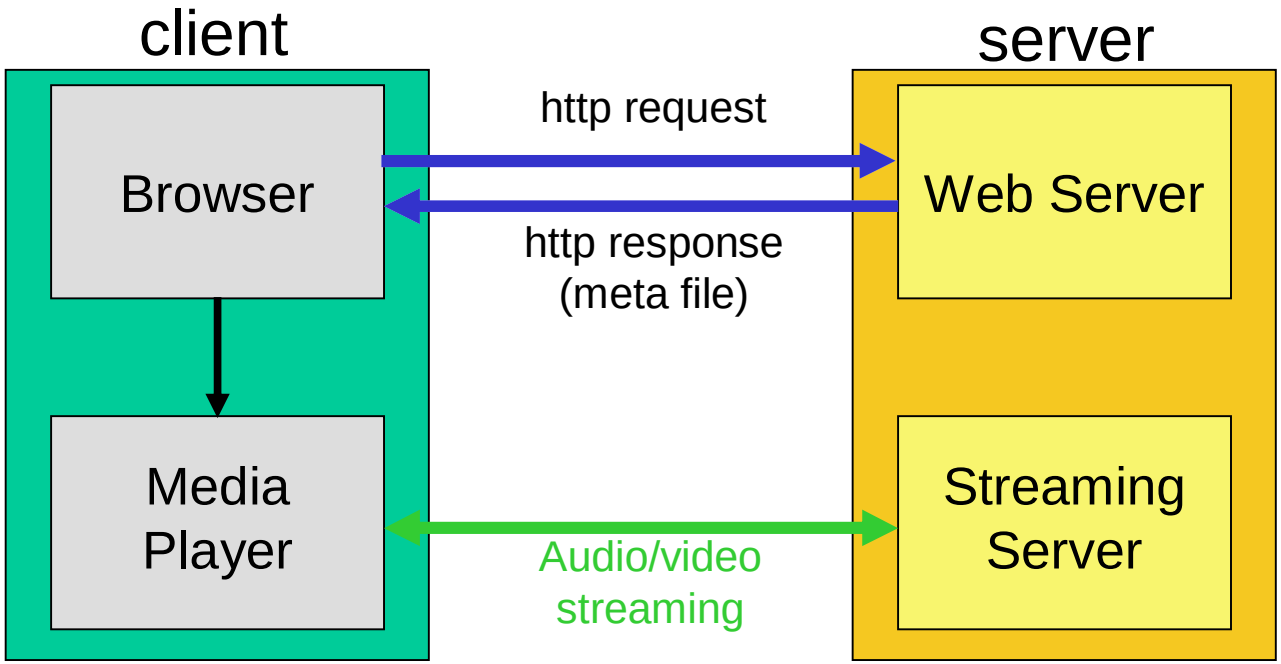
- Media on demand unicast



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

Streaming Server



Applicazioni

- Classici sistemi di streaming sono:
  - RealNetworks 
  - Microsoft Windows Media Technologies 
  - Apple Quicktime (Darwin Streaming Server) 

4./5. Real Time interattivo su IP

- Alcune delle applicazioni viste nella lezione precedente (e.g., semplice download di file multimediali) non richiedono il rispetto di requisiti stringenti sui ritardi (e dunque, tra l'altro, sulla lunghezza del buffer)
- In caso di audio/video conferenza su Internet (uno a uno, o uno a molti) è invece fondamentale che il ritardo complessivo resti entro una soglia, senza per questo aumentare le perdite di pacchetti

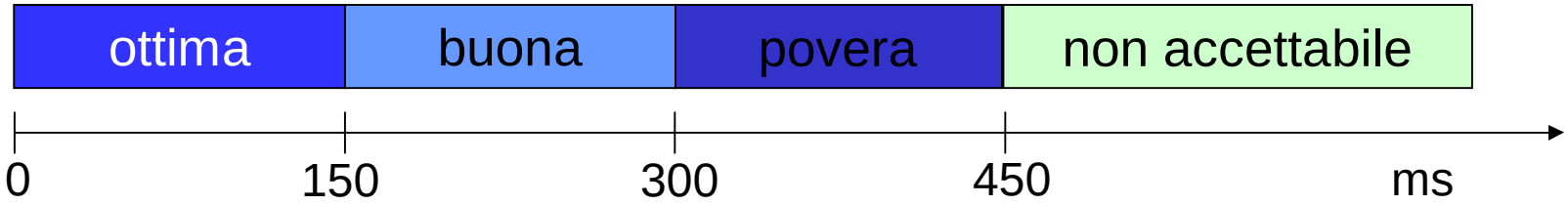
Caratteristiche dei media

- Il trade-off tra pacchetti persi e ritardo deve essere gestito con molta attenzione
- Ad es. per il *parlato*, la qualità minima accettabile in termini di:
  - Numero di pacchetti persi
  - Ritardo

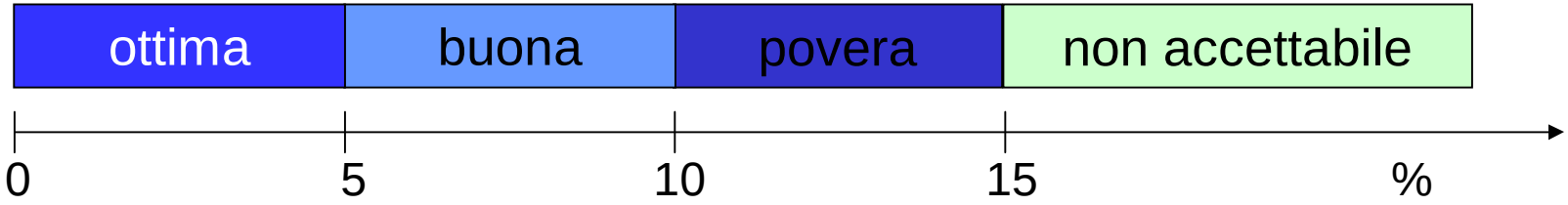
  può essere definita da una scala che prevede 3 livelli di qualità considerata accettabile (ottima, buona, povera)

Qualità del parlato

- Rispetto ai ritardi



- Rispetto alle perdite di pacchetto



Caratteristiche dei media

- Tipicamente, per questo tipo di applicazione vengono aperti canali sincroni interattivi di tipo full duplex
- Questo significa che devono essere previsti meccanismi di sincronizzazione tra la comunicazione (ovviamente unidirezionale) tra sorgente e destinazione e quella inversa (di nuovo unidirezionale), tra destinazione e sorgente

Migliorie per lo streaming

- Rispetto ai meccanismi di streaming visti fino ad ora, per le applicazioni real time di tipo interattivo (es. audio/video conferenza) sono messe in opera in numerose migliorie, tra cui:
  - Riduzione del processing delay alla sorgente ed alla destinazione
  - Gestione del silenzio
  - Compensazione dell'eco
  - Adattività del livello di compressione
  - Adattività del buffer

Riduzione del processing delay

- E' possibile diminuire il processing delay:
  - Via software: utilizzando sistemi operativi real time
  - Via hardware: utilizzando hardware dedicato per campionare e comprimere il flusso multimediale (audio)
- Sono diponibili sul mercato schede *ad hoc* di tipo DSP, *Digital Signal Processing*, che integrano gli algoritmi di campionamento e di compressione

Gestione del silenzio

- Circa il 60% di una comunicazione verbale full duplex è silenzio
- Il silenzio può essere rilevato ed eliminato (inutile trasmettere pacchetti che contengono solo rumore bianco)
- Il rumore di fondo va comunque ricostruito al ricevente perché l'assenza di playout può far pensare all'utente a una interruzione del servizio

Gestione dell'eco

- In una comunicazione full duplex sono presenti fenomeni di eco
- ESEMPIO: quando si utilizzano le casse, il playout del messaggio del mittente viene ricampionato al ricevente e spedito di nuovo al mittente, che sente la sua stessa voce con un certo ritardo
- L'eco deve essere eliminato (via HW oppure via SW)

Modulazione della compressione

- Il sistema (alla sorgente) stima la larghezza di banda disponibile e inizia la trasmissione codificando e comprimendo utilizzando un livello di compressione adeguato
- Nel caso la larghezza di banda diminuisca, o aumenti, il sistema attiva un codec differente e “avverte” il sistema client che “si allinea”
  - G.726, G.729, G.723.1

Adattabilità del buffer

- Tipicamente la lunghezza del buffer viene definita ad inizio comunicazione e rimane inalterata finché la comunicazione non è conclusa
- Si possono verificare due diversi tipi di fenomeni che procurano perdita di pacchetti:
  - Buffer underflow: i pacchetti arrivano troppo tardi per essere mandati in playout (la loro posizione nel buffer non è “più disponibile”)
  - Buffer overflow I pacchetti arrivano troppo presto per essere memorizzati nel buffer, ovvero la loro locazione non è “ancora disponibile”

Adattività del buffer

- Questi problemi possono determinarsi a causa del fatto che le prestazioni della rete (variabili nel tempo) possono cambiare (rispetto a quanto fissato all'inizio della comunicazione)
  - In meglio: ===➔ overflow
  - In peggio: ===➔ underflow
- Alcuni algoritmi reagiscono prevedendo che la dimensione del buffer si adatti dinamicamente alle prestazioni della rete

Adattività del buffer

- Se la rete cambia nel tempo la velocità con cui consegna i pacchetti all'applicazione, allora il ritardo di playout (e la dimensione de buffer) deve potersi:
  - Accorciare: le prestazioni della rete sono migliorate e si vuole ridurre il ritardo (mantenendo possibilmente inalterato il numero di pacchetti persi)
  - Allungare: le prestazioni della rete sono peggiorate e il numero dei pacchetti persi ha superato la soglia tollerabile. Si vuole diminuire il numero dei pacchetti persi a discapito del ritardo (che ovviamente aumenterà)

Allungare il buffer

- Per allungare il buffer occorre:
  - Individuare un silenzio nella conversazione
  - Inserire (al lato ricevente) nel playout un silenzio fittizio che consenta di aumentare il tempo di attesa per l'ascolto
  - Allungare e riallineare il buffer (al lato ricevente)
- Se la comunicazione è full duplex, deve essere riallineata anche l'applicazione dall'altro lato

Accorciare il buffer

- Per accorciare il buffer occorre:
  - Individuare un silenzio nella conversazione
  - Eliminarlo (al lato ricevente) nel playout
  - Accorciare e riallineare il buffer (al lato ricevente)
- Se il silenzio individuato non è sufficiente a coprire il taglio alla lunghezza del buffer, il procedimento può essere distribuito su più silenzi consecutivi
- Se la comunicazione è full duplex, deve essere riallineata anche l'applicazione dall'altro lato