# TECNICHE PER QUALITY-OF-SERVICE E MULTICAST

Elena Pagani, Gian Paolo Rossi {pagae , rossi}@dsi.unimi.it

Dip. Informatica - Università degli Studi di Milano

Corso Sistemi Multimediali, A.A. 1999-2000

#### Multimedia e QoS

- dati devono essere ricevuti a destinazione in forma comprensibile
- es. trasmissioni audio/video per videoconferenza, teledidattica, TV via cavo, telefonia su Internet, trasmissione documenti web...
- due meccanismi utilizzabili
- 1. **bufferizzazione** e riproduzione: non sempre va bene
- applicazioni interattive, invio di dati sensibili (medici, finanziari...)
- 2. QoS in tempo reale: oggetto di questa lezione
- protocolli di rete tali che il flusso di dati ricevuto a destinazione può essere immediatamente riprodotto con le caratteristiche desiderate

# Servizio di trasmissione dati di IP

- servizio best-effort non va bene
- non garantiti limiti superiori al ritardo di trasmissione
- non garantita consegna (ordinata) a destinazione
- protocollo TCP inadatto
- ullet deve essere disponibile un servizio di trasporto  $di~qualit\grave{a}$
- Problema: IP su tutti i router presenti in Internet

 $\leftarrow$ 

- vanno effettuate delle modifiche ai protocolli dello stack TCP/IP
- vanno aggiunti protocolli
- modifiche/aggiunte devono avere il minor impatto possibile sulla struttura esistente di Internet

# Definizione di Quality-of-Service

- definizione dipendente da caratteristiche applicazioni estensione: end-to-end, hop-by-hop, su dominio granularità: per pkt, per flusso, per aggregato parametri: pkt rate, banda, ritardo, jitter, loss...
- specifica qualititiva vs. quantitativa

#### • esempi:

- audio richiede basso jitter, bassa loss
- dati sensibili richiedono basso ritardo
- video richiede grande banda

## TASSONOMIA DEI REQUISITI

#### • proprietà **temporali**

iness  za o richiesto, tolleranza a ottimale, tollerabile azioni host		frame size
iness  za o richiesto, tolleranza a ottimale , tollerabile	video: 15 frame/sec	prestazioni host
iness za o richiesto, tolleranza	videoconferenza	banda ottimale , tollerabile
iness	e-mail: 5 min. , 24 h	tempo richiesto, tolleranza
Iness	conferenza: 20 msec.	latenza
	audio: 1 msec. cbr	jitter
	audio compresso	burstiness
	lip-sync: 80 msec.	sincronicità
rd/soft real time   medico: hard	medico: hard	hard/soft real time

## TASSONOMIA DEI REQUISITI

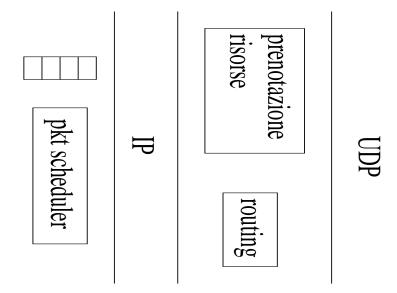
#### • affidabilità

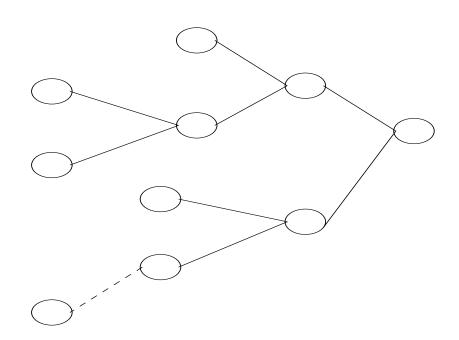
file transfer: 10 sec.	recovery da fault
	rilevamento fault
perdita semantica $  x \text{ frame OK}, y \text{ persi}  $	perdita semantica
video: 20%	perdita tollerata

#### • sessione

#	flus	flus
sender/receiver	flussi attivi	flussi potenziali
video multicast: 10 <sup>5</sup>	football match: 6	football match: 15

• sicurezza: autenticazione, accounting





## Meccanismi per la QoS

prenotazione risorse rete

banda: parte del canale è riservata per pkt di un'applicazione

delay: dipende da banda disponibile e ritardo che i pkt subiscono nei router

loss: dipende da affidabilità canali attraversati e disponibilità spazio buffer nei router

- scelta cammini
- normali protocolli routing considerano metrica = # hop
- servono protocolli che individuino i cammini in accordo a metriche di QoS
- problema: metriche statiche vs. dinamiche
- basso impatto su best effort

## COMUNICAZIONI MULTICAST

- molte applicazioni hanno pattern di comunicazione uno-a-molti o molti-a-molti
- QoS sul gruppo introduce ulteriori problemi in termini di:
- requisiti da soddisfare: applicazioni interattive o che distribuiscono dati sensibili possono richiedere basso fair delay
- prenotazione risorse efficiente verso più destinazioni, adattabile in funzione dei cambiamenti di membership
- adattamento a receiver eterogenei (sindrome  $crying\ baby$ )
- controllo della congestione al sender (implosione feedback)
- 2 benchmark: 1 sender, 10<sup>5</sup> receiver (efficienza multicast); oppure 10<sup>2</sup> sender, 10<sup>2</sup> receiver (congestione)

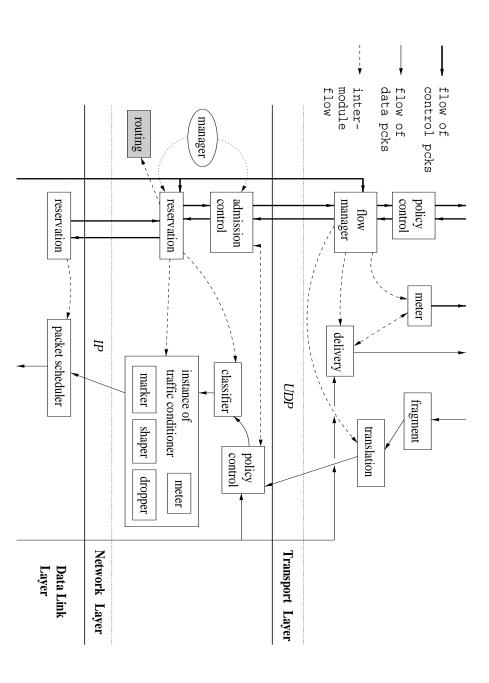
## ESEMPI REQUISITI APPLICAZIONI

		BANDA	TOLLERANZA	TOLLERANZA
		RICHIESTA	DELAY	LOSS
1M	distrib. audio/video	alta	media	media
1M	push news, meteo	bassa	alta	alta
1M	push quotazioni borsa	media	bassa	bassa
1M	caching web site	media	alta	bassa
MM	multimedia conferencing	alta	bassa	media
MM	concurrent processing	alta	bassa	bassa
MM	MM cooperative work	media	$\operatorname{media}$	bassa

# Tassonomia delle applicazioni e classi di traffico

- appl. elastiche: servizio best effort
- ullet appl. **real-time**: dati utili se arrivano entro  $\Delta t$
- appl. playback: latenza (talk vs. TV), fedeltà del segnale
- l. appl. **intolleranti**: fissato offset delay; servizio garantito che rispetta perfettamente un upper bound
- 2. appl. tolleranti: offset delay dinamico; servizio predittivo
- traffico **CBR**: MPEG1, voce senza soppressione silenzio
- traffico **VBR**: MPEG2, voce con soppressione silenzio

## ARCHITETTURA DI RIFERIMENTO



# CLASSIFICAZIONE DELLE FUNZIONALITÀ

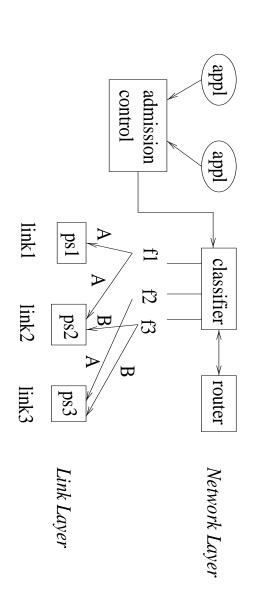
- 1. set-up servizio
- prenotazione risorse, configurazione
- 2. gestione traffico QoS in rete
- politica di forwarding
- gestione traffico fuori profilo
- interazioni tra servizi
- 3. metering
- ullet adattamento della sorgente (cambiamento codifica)
- adattamento dinamico risorse / configurazioni
- 4. problemi sicurezza
- accounting, autorizzazioni (denial-of-service, theft-of-service)

## APPROCCI IN LETTERATURA

- Servizi Integrati (Int-Serv, IS)
- lavorano per flusso di pck
- applicazione deve specificare QoS desiderata per ogni flusso
- poco scalabile ma servizio accurato
- Servizi **Differenziati** (*Diff-Serv*, *DS*)
- lavorano per aggregati di flussi
- applicazione sceglie tra servizi disponibili
- scalabile ma servizio poco accurato
- proposti servizi standard e architetture di riferimento

#### Servizi integrati

• ogni router deve ricordare informazioni per ogni flusso da cui è attraversato



- multicast: composizione richieste di flussi diversi
- $\bullet$  QoS specificata come TSpec: token bucket + tasso di picco + taglia frame

## IS A CARICO CONTROLLATO

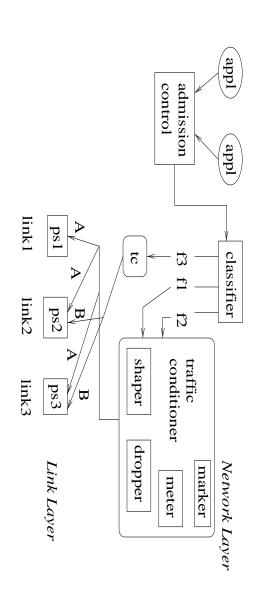
- servizio come a rete scarica: appl. real-time tolleranti
- alta percentuale pkt consegnati
- alta percentuale pkt con delay prossimo al minimo
- ridotti congestione e ritardi in coda
- stima traffico generato
- flessibilità per gestione burst
- traffico in eccesso: best effort
- NO impatto su best effort e flussi regolari
- esempio: distribuzione audio/video

### IS CON QOS GARANTITA

- ullet bound su delay end-to-end matematicamente dimostrabile
- delay = partefissa + code
- NO fallimenti o cambiamenti routing
- NO minimizzazione jitter
- parametri: error rate, varianza sul tempo di transito
- taglia buffer dipendente da tali parametri: se posso riservare un buffer sufficiente allora il servizio è accettato

#### Servizi differenziati

- ullet set di servizi tra cui scegliere prestabilito dall'ISP ( $Per ext{-}Hop ext{-}Behaviour)$
- $\bullet$  tutti i microflussi che devono subire lo stesso trattamento vengono aggregati e gestiti insieme



- prenotazione risorse per aggregato
- accordo tra ISP (configurazione router) ai confini di dominio

#### PER HOP BEHAVIOUR

- indicato da TOS byte (DS byte) nel singolo pacchetto
- proposti standard per bassa perdita, basso jitter, bassa latenza
- assured forwarding (AF): servizio qualitativo
- 4 classi AF, 3 livelli drop per ogni classe
- se x < y: pkt di classe x hanno tempo di forwarding non inferiore a quello dei pkt di
- se p < q: pkt con livello drop p inoltrati con probabilità non inferiore a quella dei pkt con livello q
- **expedited forwarding (EF)**: virtual leased line (Premium Service)
- configurazione tempo min. uscita da un nodo
- condizionamento traffico: frequenza arrivo

#### Protocolli Standard

## **RSVP**: prenotazione risorse (livello net)

- trasporta messaggi di controllo per prenotazione
- receiver-oriented
- collabora con moduli di reservation e admission control

## RTP (RTCP): trasporto dati real-time

- invio/consegna dati real-time
- monitoraggio QoS fornita
- adattamento dinamico condizioni di rete

## RTSP: gestione flussi multimediali

- playback flussi in accordo a scelte utenti
- risincronizzazione flussi di diversi media

#### Real-Time Protocol

- ipotizza set-up eseguito da altri protocolli
- può sfruttare servizi di UDP
- funzione trasporto dati:
- consegna dati ordinata ed in accordo a deadline temporale
- ri-sincronizza flussi di uno stesso stream
- adatta codifica informazioni
- funzioni di monitoring:
- serve per adattamento dinamico flussi inviati
- serve per stima risorse necessarie per soddisfare richieste successive

#### RTP: CONSEGNA DATI

- pkt numerati in sequenza e marcati con timestamp
- diversi flussi di uno stesso stream gestiti separatamente
- time-stamp usati per risincronizzazione di diversi media
- moduli *mixer* e *translator*:
- cambiano la codifica dei dati per adattarsi alle risorse di rete disponibili
- possono fare trasformazione multicast  $\rightarrow$  unicast
- consentono attraversamento firewall
- multicast: adattamento codifica nei punti di branch

## RTCP: FUNZIONI DI MONITORING

- raccolta misure QoS ad ogni pacchetto ricevuto
- periodico invio di report dalle/alle sorgenti
- permette anche controllo di congestione, rilevamento errori
- $\bullet$ pkt inviati a tuttii partecipanti
- porzione banda riservata per pkt controllo

#### PROBLEMI

- ullet implementazione incompleta: non è ancora chiaro come realizzare mixer e translator
- specifica lacunosa: classificato come protocollo di livello 4, ha componenti di livello 3
- multicast: adattamento per diversi riceventi?

### QoS: PROBLEMI APERTI

- sia IS che DS richiedono modifiche alla struttura di ogni router
- possibilità di esistenza di domini non-compliant
- traduzione servizi tra domini adiacenti DS-DS o DS-IS
- DS: PHB deve garantire QoS end-to-end
- letteratura frammentaria e incompleta
- protocolli proposti come standard sono poco scalabili (RSVP), oppure si occupano di problematiche di alto livelli trascurando strati inferiori (RTP, RTSP)
- ullet definizione standard è incompleta (non esiste modulo translator di RTP)

## SPUNTI PROGETTO D'ESAME

- esistono diversi tool e implementazioni di protocolli
- tool per generazione traffico sorgente sintetico/reale
- strumenti monitoraggio e misurazione prestazioni
- implementazioni prototipali degli standard
- di molti sono disponibili i codici sorgenti free (MBone, standard...)

 $\leftarrow$ 

- 1. installazione tool e osservazione del funzionamento
- 2. costruzione di semplici architetture per QoS
- 3. misurazione comparata prestazioni di diversi tool che realizzano le medesime funzionalità