Capitolo 7

1)Nei video ci sono due tipi di ridondanza: descriveteli e spiegate come possono essere sfruttati per ottenere una compressione efficiente.

Nella compressione video vengono sfruttati due tipi di ridondanza. La ridondanza spaziale è quella che avviene all'interno di un'immagine/frame di un video, ed è dovuta al fatto che quest'ultima è rappresentata mediante pixel (ognuno dei quali codificato con un numero di bit per rappresentare luminanza e crominanza). Questo tipo di ridondanza consiste nel fatto che magari alcune immagini, che si susseguono in un video, presentano una colorazione omogenea -> ergo può essere compressa in modo efficace senza sacrificarne in modo significativo la qualità. Altro tipo di ridondanza è quella temporale che è data dalla ripetizione della stessa immagine in due tempi successivi. Questo tipo di ridondanza mi permette di utilizzare una stessa codifica più volte durante la compressione di un video.

2) Supponete che un segnale audio analogico sia campionato 16.000 volte al secondo e che ogni campione sia quantizzato in 1024 livelli. Qual è il bit rate risultante del segnale audio digitale PCM?

FC=16000 volte al secondo
LIV=1024=2^10 → 10 bit per rappresentare un campione
RATE=FC*LIV=160000bps

3) Le applicazioni multimediali possono essere classificate in tre categorie: elencatele e descrivetele.

Le applicazioni multimediali sono classificate in tre categorie:

- Streaming audio e video di contenuti registrati. In questa classe di applicazioni i contenuti sono memorizzati su server a disposizione (su richiesta) degli utenti. Le caratteristiche che contraddistinguono questa classe sono: *Streaming,* (il client tipicamente inizia la riproduzione audio/video pochi secondi dopo aver iniziato a ricevere il file dal server in modo da evitare lo scaricamento dell'intero file e quindi possibilità di incombere in lunghi ritardi) *Interattività,* (il contenuto multimediale è registrato e archiviato sul server. Quindi,

gli utenti possono usare le funzioni di pausa, riavvolgimento e avanzamento rapido.) *Riproduzione continua.* (quando la riproduzione inizia, dovrebbe procedere secondi i tempi di registrazione originali. Ciò impone che i dati debbano essere ricevuti dal client in tempo utile per la loro riproduzione.

- Conversazione voce/video su IP. Questa classe di applicazioni viene comunemente definita telefonia Internet o Voice-over-IP (VoIP) in quanto per l'utente è simile al tradizionale servizio telefonico a commutazione di circuito. L'aspetto della temporizzazione è importante, in quanto queste applicazioni sono altamente sensibili ai ritardi. D'altro canto queste applicazioni sono tolleranti alle perdite.

- Streaming Audio e video in tempo real-time. Questa terza classe di applicazioni è simile alle tradizionali trasmissioni radiotelevisive, a parte il fatto che avviene su Internet. Come per lo streaming di contenuti registrati, anche in questo caso la rete deve fornire a ogni flusso un throughput medio maggiore del suo bit rate per garantire la continuità della riproduzione. Siccome l'evento è in diretta, anche il ritardo potrebbe essere un problema. Tuttavia possono essere tollerati ritardi fino a una decina di secondi da quando l'utente richiede l'invio a quando inizia la riproduzione.

4) I sistemi di video streaming possono essere classificati in tre categorie: elencatele e descrivetele brevemente.

I sistemi di video streaming sono classificabili in tre categorie:

- Streaming UDP. Nello streaming UDP il server trasmette il video allo stesso bit rate a cui il client lo consuma, inviando i blocchi video in pacchetti UDP a un tasso costante.

Lo streaming UDP usa generalmente un buffer lato client molto piccolo, in grado di contenere meno di un secondo di video. I

l server, prima di passare i blocchi video a UDP, li incapsula in pacchetti di trasporto appositamente progettati per audio e video, usando il protocollo di trasporto in tempo reale (RTP). Un'altra proprietà che contraddistingue lo streaming UDP è il fatto che client e server mantengono oltre al flusso video anche, in parallelo, una connessione di controllo separata sulla quale il client invia i comandi

riguardanti i cambiamenti di stato della sessione, quali la pausa, la ripresa della riproduzione etc. Gli svantaggi di queso tipo di streaming sono stati inseriti nella domanda successiva.

- Streaming HTTP. Nello streaming HTTP il video viene semplicemente memorizzato in un server HTTP come un file ordinario con un URL specifico.

Quando un utente vuole vedere un video, il client stabilisce una connessione TCP con il server e invia una richiesta GET HTTP per il suo URL. Il server invia il file video, all'interno di un messaggio di risposta HTTP, più velocemente possibile, vale a dire tanto più velocemente quanto il controllo di flusso e di congestione TCP lo permettono.

Sul lato client i byte vengono memorizzati in un buffer dell'applicazione client. Quando il numero di byte nel buffer supera una soglia fissata, l'applicazione client inizia la riproduzione.

Inoltre l'uso di HTTP e TCP permette ai video di attraversare più facilmente firewall e i NAT che sono spesso configurati per bloccare la maggior parte del traffico UDP, ma lasciano passare la maggior parte del traffico HTTP. Inoltre, lo streaming HTTP elimina la necessità di avere un server di controllo, come un server RTSP, riducendo i costi di un'installazione su larga scala su Internet.

Usando buffer e prefetching si può migliorare la riproduzione continua anche in presenza dei meccanismi di trasferimento dati affidabili e controllo di congestione propri di TCP.(L’uso del buffer consiste nell’utilizzo di un buffer lato client che ha la funzionalità di mantenere i dati ottenuti dal server prima della riproduzione vera e propria. Mentre il prefetching consiste nel fatto che il client potrebbe tentare di scaricare il contenuto ad un tasso più alto di quello di consumo.)

- Streaming HTTP adattativi. Questo servizio di streaming nasce dal grande svantaggio dello Streaming HTTP di fornire contenuti ad una qualità fissa, indipendentemente dalla disponibilità di banda del cliente. Lo streaming adattativo (DASH) fornisce dunque un servizio analogo al comune streaming HTTP, ma con l’unica differenza che sul server sono presente più versioni di uno stesso contenuto con

differenti qualità ed indirizzate ad URL diversi. DASH permette dunque ai clienti di adattare la scelta della versione del contenuto in base alla banda disponibile durante la sessione, monitorando in modo dinamico la banda disponibile ed il livello del buffer del client.

5) Elencate tre svantaggi dello streaming UDP.

Gli svantaggi dello streaming UDP sono tre e sono:

- Il fatto che lo streaming UDP a tasso costante può non fornire riproduzione continua, data la quantità di banda disponibile tra server e client non solo variabile, ma anche impredicibile.

- Il secondo svantaggio dello streaming UDP è il fatto che richiede un server di controllo, come un server RTSP, per elaborare le richieste interattive da client a server e tracciare lo stato del client, aumenta la complessità e i costi.

- Il terzo svantaggio deriva dal fatto che molti utenti non possono ricevere video UDP, in quanto i loro firewall sono configurati per bloccare tale traffico.

6) Nello streaming HTTP il buffer di ricezione TCP e quello dell'applicazione client sono la stessa cosa? In caso negativo, come interagiscono?

Nello streaming HTTP il buffer di ricezione TCP e quello dell’applicazione client sono differenti! Il buffer di ricezione TCP ha lo scopo di ricevere il contenuto compresso ed inoltrato dal server sulla connessione socket. Il buffer dell’applicazione client invece viene riempito dall’applicazione client (media player), la quale tramite la sua socket legge i byte del buffer di ricezione TCP e li pone nel buffer dell’applicazione client. Quest’ultima parallelamente e periodicamente preleva i frame video dal buffer, li decomprime e li visualizza. Si noti che se il buffer dell’applicazione client si riempisse, automaticamente si saturerebbero sia il buffer TCP di ricezione che di invio e dunque il server interromperebbe la trasmissione del contenuto.

7) Considerate il semplice modello di streaming HTTP. Supponete che il server invii i bit a un tasso costante pari a 2 Mbps e la

riproduzione inizi quando sono stati ricevuti 8 milioni di bit. Qual è il ritardo iniziale di buffer *Tb*?

RATE=2 Mbps= 2^21 bps oppure 2*10^6 bps;

START=8.000.000 bit;

Tb = START/RATE=4sec.

8) Le CDN adottano in generale due diversi approcci per la dislocazione dei server. Elencateli e descriveteli brevemente.

- Il primo approccio per la dislocazione dei server CDN è l'Enter deep: l'idea è quella di entrare profondamente nelle reti di accesso degli ISP installando gruppi di server detti cluster, con l'obiettivo di stare vicini agli utenti finali in modo da migliorare il ritardo percepito dall'utente e il throughput, diminuendo il numero di collegamenti e router tra l'utente finale ed il cluster CDN da cui riceve i contenuti

- Il secondo approccio per la dislocazione dei server CDN e il Bring Home: l'idea è quella di portare in casa l'ISP costruendo grandi cluster in pochi punti chiave ed interconnetterli usando una rete privata ad alta velocità. Invece di entrare negli ISP di accesso, queste CDN pongono ogni cluster in un luogo vicino ai PoP (punto di accesso alla rete) di molti ISP di livello1.

9) Nel Paragrafo 7.4.2 abbiamo descritto alcune strategie di selezione dei cluster. Di queste quale trova un buon cluster rispetto all'LDNS del client? Quale rispetto al client stesso?

La strategia di selezione del cluster che trova un buon cluster rispetto all'LDNS del client è la strategia che assegna ad un client il cluster *geograficamente più vicino*. Con questa strategia la CDN, quando riceve una richiesta DNS da un particolare LDNS, sceglie il cluster più vicino ad esso in linea d'aria. L'approccio *anycast IP* invece ha il vantaggio di trovare il cluster più vicino al client piuttosto che quello più vicino all'LDNS del client. Il meccanismo alla base di quest'approccio è che il router BGP, seguendo le procedure standard, prende il percorso "migliore" verso l'indirizzo IP secondo i meccanismi di selezione dei percorsi locali.

10) Oltre alle considerazioni legate alla rete, come ritardi, perdite e

prestazioni di banda, ci sono molti altri fattori importanti che intervengono nella scelta della strategia di selezione dei cluster. Quali sono?

- Carico di lavoro dei cluster, in quanto i client non dovrebbero essere diretti verso cluster sovraccarichi.

- Costo di consegna degli ISP, i cluster dovrebbero essere selezionati tenendo conto di rapporti contrattuali e costi.

11) Qual è la differenza fra ritardo end-to-end e jitter di un pacchetto? Quali sono le cause del jitter?

Con il termine ritardo *end-to-end* si intende la somma dei ritardi di trasmissione, di elaborazione e di accodamento nei router, più quelli di propagazione e di elaborazione sui terminali (per un'applicazione VoIP questo ritardo non dovrebbe superare i 400ms, altrimenti potrei avere limitazioni serie della conversazione). Con il termine ritardo di *jitter* invece si fa riferimento alla variabilità statistica nel ritardo di ricezione dei pacchetti trasmessi, causata dalle code interne dei router congestionati.

12) Perché un pacchetto ricevuto dopo il tempo previsto per la riproduzione è considerato perso?

Perché molto spesso si fa riferimento ad applicazioni real time, ergo è importante l'ordine di ricezione, di riproduzione, ma soprattutto l'istante di riproduzione dell'informazione.

13) Riassumete i due schemi FEC descritti nel Paragrafo 7.3. 3. Si noti che entrambi aumentano il tasso trasmissivo del flusso aggiungendo ridondanza. Anche l'interleaving incrementa il tasso trasmissivo (larghezza di banda) ?

- Primo meccanismo FEC: l'idea di base di questo schema è quella di aggiungere informazioni ridondanti al flusso originale di pacchetti. Il primo meccanismo invia, dopo ogni N blocchi, un blocco ridondante ottenuto da un'operazione di OR esclusivo degli N blocchi originali. In questo modo, se qualche pacchetto del gruppo degli $N+1$ pacchetti va perso, il ricevente lo può ricostruire integralmente (questo solo se il pacchetto perso è singolo, poiché se già i pacchetti

persi fossero due il ricevente non potrebbe ricostruire l'informazione).

- Secondo meccanismo FEC: il secondo meccanismo consiste nell'inviare uno stream audio a bassa risoluzione come informazione ridondante (il trasmittente crea *l'n-esimo* pacchetto aggiungendo il blocco *n-1-esimo* dello stream ridondante all'n-esimo blocco dello stream nominale).

Entrambi gli approcci FEC incrementano l'informazione da trasmettere, poiché aggiungono informazione ridondante a quella nominale. In alternativa alla correzione anticipata degli errori (FEC) si può pensare di utilizzare la tecnica di *interleaving* poiché non richiede l'aumento di larghezza di banda, anche se però aumenta la latenza.

14) Come può un ricevente identificare differenti flussi RTP in diverse sessioni? Come possono essere identificati quelli della stessa sessione?

Un ricevente può identificare differenti flussi RTP in diverse sessioni, come anche in una stessa sessione, grazie al fatto che i flussi di pacchetti sono indipendenti tra loro e caratterizzati da un identificativo per la sorgente (SSRC) presente nell'intestazione del pacchetto RTP.

15) Qual è il ruolo di un server di registrazione SIP e che cosa lo differenzia da quello di un agente domestico in IP Mobile?

Ad ogni utente che fa uso del protocollo SIP per comunicare, è associato un server di registrazione SIP (*SIP registrar*) al quale l'applicazione SIP su un dispositivo, quando viene lanciata, invia un messaggio di registrazione contenente l'attuale indirizzo IP presso cui l'utente può essere contattato.

Il messaggio di registrazione sarà un INVITE,che poi verrà utilizzato dal proxy server del mittente che lo inoltrerà al proxy server del destinatario che a sua volta lo inivierà al dispositivo SIP che il destinatario sta usando in quel momento, solo a quel punto il destinatario potrà inviare una risposta SIP al mittente. DA FINIRE

16) Che cos'è la priorità di accodamento senza prelazione descritta nel paragrafo 7.5? Può avere senso l'impiego di una priorità di accodamento con prelazione per una rete di calcolatori?

La priorità di accordamento senza prelazione è una modalità di accodamento che sfrutta comunque un accodamento con priorità (cioè facendo riferimento ad opportune classi di priorità) in cui la trasmissione dei pacchetti non può essere interrotta una volta iniziata.

Invece, utilizzare una priorità di accodamento con prelazione può essere utile dal momento in cui non si vuole dare la possibile ad una classe con priorità di poter essere unicamente schedulata per prima delle altre ma appunto, switchare tra le varie classi, permettendo cosi a pacchetti di diverse classi di essere eseguiti.

17) Fate un esempio di modalità di scheduling che non sia conservativa.

Se per "conservativa" intende "sempre stesso odine/modalità" allora direi Round Robin non lo è poiché è circolare come cosa, però magari con più persone o risorse non è sempre lo stesso l'ordine. Infatti RR è fa una sorta di context switch tra le varie classi cosi da permette ad ogni pacchetto di essere di essere eseguito e non di dover aspettare che tutti quelli della classe con priorità più alta vengano schedulati.

18) Fornire un esempio delle code FIFO, a priorità, Round Robin e WFQ nella vita reale.

FIFO= ticket alle poste italiane, dove vi è una file determinata da chi arriva prima prima viene servito

ROUND ROBIN= assegnazione ferie estive.

WFQ= priorità ad anziani e donne in gravidanza sui posti a sedere sui mezzi pubblici.

19) Si assuma che in coda ad un router siano presenti i pacchetti P1,P2,P3,P4,P5. I pacchetti P1 e P4 sono di una classe C1 di priorità inferiore rispetto alla classe C2 di P2,P3 e P5.

Qual è la sequenza di output dal router per gli scheduling:

FIFO, PRIORITA' e ROUND ROBIN?

Cosa succede nel caso in cui lo scheduling sia di tipo wieghted fair queuing se il peso della classe C1 è 1 e il peso di C2 è 2?

Nel caso di FIFO avrò una politica di scheduling di tipo First In first out, quindi verrano schedulati nell'ordine in cui vengono ricevuti, quindi: P1,P2,P3,P4,P5

Per quanto riguarda le CODE CON PRIORITÀ verranno eseguiti prima quelli con priorità elevata poi quelli con priorità più bassa: P2,P3,P5,P1,P4

Nel caso di RR i pacchetti delle diverse classi aventi diversa priorità verranno alternati fino ad esaurimento quindi: P2,P1,P3,P4,P5

Per ultimo se lo cheduling è di tipo WFQ(Weighted Faie Queuing) avrò che verrà schedulato 1 pacchetto dalla classe C1 e 2 dalla Classe C2 per poi ricominciare dalla classe C1: P2,P3,P1,P5,P4

20) Si consideri un flusso che deve attraversare 4 routers tra sorgente e destinazione, su ognuno dei quali detiene risorse di buffer per al più 20 pacchetti ed un rate garantito di 200 pacchetti al secondo. Calcolare il massimo ritardo end-to-end di un pacchetto del flusso dovuto ai tempi di attesa ai buffer dei 4 routers.

Nella situazione peggiore, in ogni buffer il flusso sta occupando il massimo delle risorse cioè esattamente 20 pacchetti. Ciò vuol dire che un pacchetto che è appena arrivato dovrà aspettare la trasmissione dei precendenti.

Tempo attesa trasmissione degli altri è pari a (Npacchetti/BitRate)=(20/200)=1/10=0.1sec

Per i complessivi 4 router= 0.1 *4 =0.4sec

21)Si consideri uno schema di trasmissione con ritardo adattativo che utilizzi il protocollo RTP per il trasporto del segnale vocale campionato a 64 Kb/sec in cui viene emesso un pacchetto ogni 20 ms. Come si può determinare il primo pacchetto di un periodo di parlato?

Sapendo che una differenza nei timestamp *(marcatura temporale)* di almeno 20 msec tra due pacchetti indica un nuovo periodo di attività di parlato. Quindi andando a controllare il timestamp, nell'header del pacchetto RTP, possiamo stabilire quando c'è un nuovo periodo di parlato.

Il primo pacchetto di un periodo di parlato, quindi, lo abbiamo quando dati due pacchetti consecutivi A e B, se timestamp(A) - timestamp(B) = almeno 20msec allora abbiamo B come primo pacchetto di un nuovo periodo di parlato.

Non può andar bene considerare il tempo di arrivo poichè potrebbe subire ritardi e quindi falsare i risultati, ma appunto per determinare primo pacchetto di un periodo di parlato bisogna considerare il timestamp, cioè il tempo in cui quel pacchetto è stato generato.

22) Si consideri un percorso fatto di 3 router che accettano pacchetti di dimensione 1Kb appartenenti a due classi di priorità P1 e P2. Tutti i router implementano la politica di scheduling WFQ e hanno la medesima banda in uscita di 1Mbps.
La dimensione delle code assegnate a P1 e P2 è di 10 pacchetti.
1. Assegnare il peso W1 alla classe P1 affinché il ritardo massimo di un suo pacchetto nell'attraversare i 3 router sia di 150ms.
2. Qual è in questo caso il ritardo massimo di un pacchetto che appartiene a P2?

1) ritMAXP1=150mSec
Mi posso ricavare w1/w1+w2
→ 3 (numero router) * 10(n pacchetti)*(dim pacchetti)= R(bitrate) (*w1/w1+w2)

→ (3*10*10^3)/[(1*10^6)(w1/(w1+w2))]=ritMAXp1
w1/w1+w2=1/5

w1=1
w2=4
questo perché il totale è su 5

2) Una volta che abbiamo i pesi si ricava immediatamente il ritardo massimo per i pacchetti P2,
applicando la solita formula:
[3*bufferP2]/[Rate(w2/(w1+w2))]=ritMAXp2=
→ [3*10*10^3]/[1*10^6*(4/5)]=3,75mSec

23)Si consideri un leaky bucket di dimensione del secchio = 10 pacchetti e token rate = 20 pacchetti al secondo. Nell'ipotesi che il traffico in entrata sia caratterizzato da 1 pacchetto ogni 2 msec, quanti pacchetti escono al più dal leaky bucket?

b (dimensione del bucket);
r (token rate);
p (traffico in entrata)

Il massimo numero di pacchetti che escono dal leaky bucket nell'unità di tempo è dato da: *N=b+(r*t)*

dove t è il tempo (che poniamo pari ad 1);
ovvero N = 10+(20*1)=30 pk
Notiamo dunque che nel leaky bucket il numero dei pacchetti uscenti è indipendente dal tasso dei pacchetti in entrata.

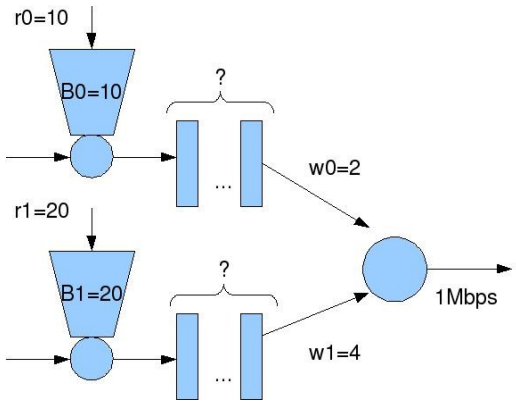
24) Si consideri un percorso fatto di 2 router. Il primo implementa la politica weighted fair queuing (WFQ) con due classi di priorità P1 di peso 2 e P2 di peso 4, entrambe le classi di priorità hanno un buffer di 100 pacchetti.
Il secondo implementa la semplice politica FIFO, ha un packet rate in uscita di 1000 pacchetti per secondo ed un buffer di 200 pacchetti. Qual è il packet rate che deve avere il primo router affinché il massimo ritardo di un pacchetto con priorità P1 che attraversa entrambi i router sia 500ms?

packetRate rate del primo router (WFQ)
w1 peso della classe di priorità P1
w2 peso della classe di priorità P2

bufferP1 buffer della coda di priorità P1
bufferR2 buffer del secondo router (FIFO)

[(bufferP1)/((packetRate)(w1/w1+w2))]+[(bufferR2/
rateR2)]=500mSec
→ (100/(pR*2/6))+(200/1000)=500msec
→ pR=1001 pacc/sec
ovviamente può essere anche maggiore quindi in definita avremo
packetRateR1 >= 1001pps

25) Si consideri un percorso fatto di 2 router con rate di 600Kbps ciascuno che accetta pacchetti di dimensione 1 Kb in ingresso a due distinte code. Il traffico in ingresso alle code è regolato da un sistema di leaky bucket con r0=10 token per secondo e B0=10 pacchetti e r1=30 token per secondo e B1=30 pacchetti.
Infine la politica di scheduling è WFQ con peso w0=2 per la coda 0 e w1=4 per la coda 1.
1)Qual è la dimensione minima delle code del router affinché nessun pacchetto vada perduto ?
2)Qual è il massimo ritardo end-to-end?



1)Sapendo che (r0 * t+ b0)/t <R * (w0/w0+w1)
avrò che

r0t+B0]/t < 600kbps(w0/(w1+w0)) con t=1sec
[r1t+B1]/t < 600kbps(w1/(w1+w0)) con t=1sec

deduco che la massima coda necessaria sia pari ai massimi burst, quindi a rispettivamente 20 e 60 pacchetti.
queste sarebbero le dimensioni minime delle code ovvero, se possono uscire al più b+rt pacchetti/sec, devo contenere almeno b+rt pacchetti/sec

Detto questo i ritardi massimi sono naturalmente
(dim_coda/flusso)*2(numero router):

ritardo nella coda 0) (20/200)*2=0.2 sec
ritardo nella coda 1) (60/400)*2=0.3 sec
E quindi il ritardo massimo generale per un singolo pacchetto sarebbe 0.3sec

26) Un servizio real-time è caratterizzato da un R_spec con banda pari a 100Kbps e tempo di slack pari a 0.3s. Si considerino pacchetti di dimensione 1Kb e tre possibili percorsi alternativi costituiti da due router identici, che assegnano al servizio rispettivamente:
1.Banda in uscita 70Kbps, buffer 5 pacchetti
2.Banda in uscita 200Kps, buffer 10 pacchetti
3.Banda in uscita 200Kps, buffer 100 pacchetti
Quale dei percorsi garantisce il rispetto del R_spec? Giustificare la risposta.

1) Non può essere perché se la R_spec comunica una velocità di invio di pacchetti a 100Kbps, il router deve essere in grado di smaltire questi pacchetti. Se il rate di uscita del router è inferiore a quello specificato nella R_spec, il buffer si riempirebbe subito, e c'è possibilità di perdita di pacchetti.

2)
Ritardo =(2(router)*10(pacchetti)*10^3(dim pacchetti))/bitRate

Ritardo rit2=2*10*10^3/(200kbps)=0,1sec
Avendo uno slack
→ 0.1 < 0.3
→ viene garatito il rispetto del R_spec

3)
Ritardo rit3=2*100*10^3/(200kbps)=1 sec
1sec >slack
→ non viene garantito R_spec

27) Si consideri il seguente traffico nel tempo (t):
0<=t<2 secondi 100 pacchetti al secondo, t=2 secondi 1000 pacchetti istantanei t>2 secondi 35 pacchetti al secondo.
Se questo traffico arriva in ingresso ad un leaky bucket di parametri b=100 pacchetti e r = 30 gettoni per secondo, come viene modificato in uscita? Disegnare l'andamento nel tempo.

N(numero pacchetti in uscita) = b + r*t

Supponendo di avere il bucket “pieno” in partenza (A t=0 ho il burst di 100.), al primo secondo arrivano 100 pacchetti e viene “svuotato”, uscendo 100 pacchetti.
Dopodiché la velocità di “riempimento”è di 30 gettoni per secondo. Quindi per quanti pacchetti arrivino, solo 30 ne potranno passare.
Al secondo 1, ho quindi 30 pk nel bucket, col traffico in entrata di 100: quindi possono uscire 100+30 pacchetti=130 pk

Quindi al t=2 ho 1000 pacchetti instantanei, ma a questo punto, però il bucket non si riempie, quindi da 2 in poi ho solo 30 pk/s

ricapitolando
0<=t<2 ho 130 pk
t=2 ho 30 pk/s
t>2 ho 30 pk/s

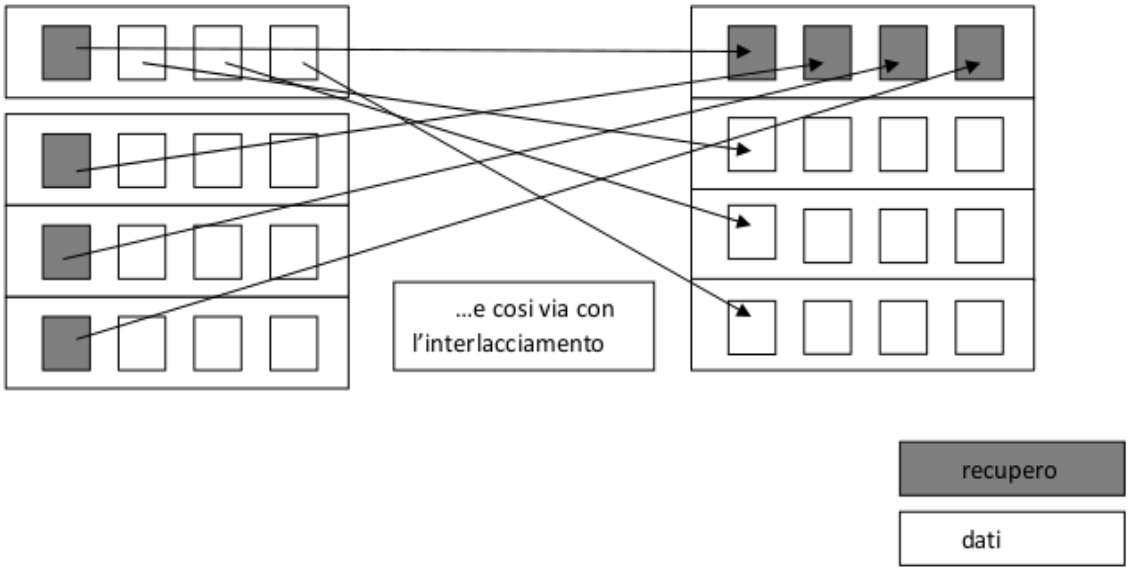
28) Per quale motivo l'header dei pacchetti RTP contiene un numero di sequenza ed una marcatura temporale?
Discutere in che modo si potrebbe realizzare un protocollo affidabile usando solo datagrammi UDP

Utilizzando numero di sequenza e timestamp (marcatura temporale). Numero di sequenza perché UDP non assicura che i pacchetti arrivino con lo stesso ordine con cui l'abbiamo inviati. Il timestamp è dovuto al fatto che, ad esempio nella telefonia, scartiamo i pacchetti che arrivano con più di 400 msec di ritardo, poiché ritardi eccessivi sono inaccettabili. Inoltre il timestamp, in unione ai numeri di sequenza servono a distinguere i periodi di parlato da quelli di silenzio nel caso di ritardo di playout adattivo.

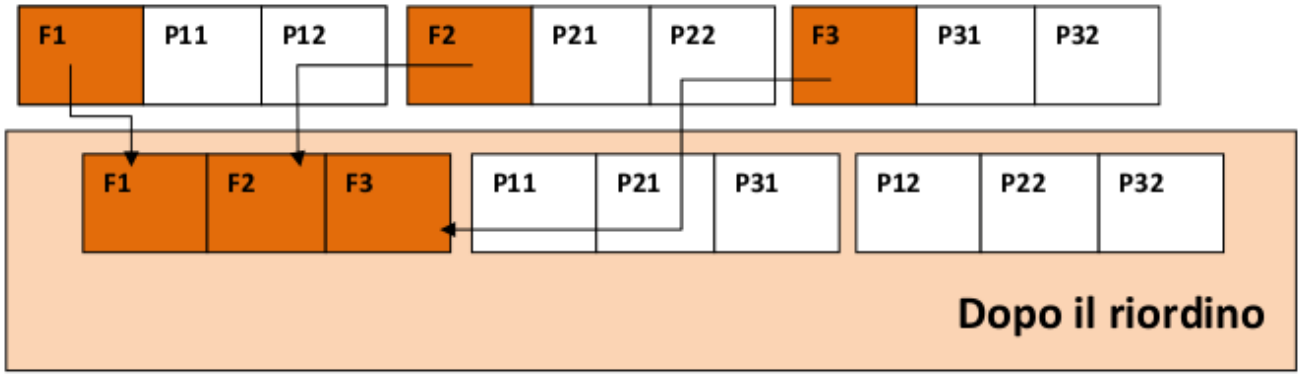
Si potrebbe realizzare un protocollo affidabile nel momento in cui tu vai ad utilizzare meccanismi FEC e interleaving per recupero di blocchi di pacchetti persi. In questo modo cerchi di assicurare il trasferimento dati affidabile.

29) Si consideri un codice FEC che agisce su blocchi fatti di 3 pacchetti +1 di recupero. Progettare e disegnare uno schema di interleaving affinché l'uso congiunto di codice FEC ed interleaving sia in grado di tollerare la perdita di un intero blocco di interleaving

Basta mettere i pacchetti di recupero tutti nello stesso blocco, così posso perdere un blocco intero a burst.



30) Si consideri un codice a correzione per un segnale vocale con codifica PCM che introduca un pacchetto di recupero (per esempio reed-solomon) ogni due pacchetti dati. Si consideri inoltre una codifica interleaving che opera su insiemi di tre blocchi formati ognuno da tre pacchetti, due dati ed il corrispondente pacchetto di recupero.
Si determini la massima lunghezza di un singolo burst di perdita di pacchetti a cui la codifica è tollerante, il minimo ritardo di riproduzione introdotto dalla codifica e lo spreco di banda introdotto.



Blocco1=<F1, P11, P12>
Blocco2=<F2, P21, P22>
Blocco3=<F3, P31, P32>
Dove F1, F2, F3 sono i pacchetti di recupero FEC e P11, P21, P31, P12, P22, P32 sono i dati.

Questo schema può tollerare solo un errore nel primo blocco (che correggi con F1), un errore nel secondo blocco (che correggi con F2) ed un errore nel terzo blocco (che correggi con F3).

Applichiamo il seguente riordino (come in figura):

Blocco1=<F1, F2, F3>
Blocco2=<P11, P21, P31>
Blocco3=<P12, P22, P32>

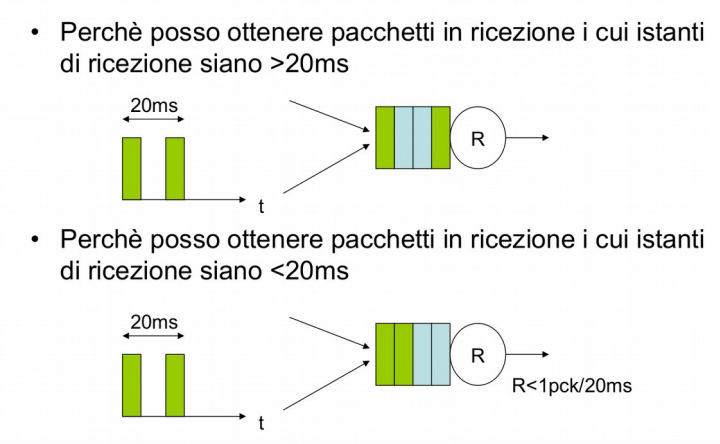
Immaginiamo ora di perdere TUTTO il blocco 2.

Adesso lo possiamo recuperare.
Con F1 recupero P11, con F2 recupero P21, con F3 recupero P31
Siamo tolleranti alla perdita di 1 burst su 3, lo spreco di banda è 1/3
Il minimo ritardo riproduzione è determinato dal ritardo di trasmissione + ritardo per fare l'interleaving + ritardo del riordino + eventuale correzione.

31) Supponiamo che un trasmittente invii pacchetti ogni 20ms.
1)Perchè posso ottenere pacchetti in ricezione i cui istanti di ricezione siano >20ms ?

2) Perchè posso ottenere pacchetti in ricezione i cui istanti di ricezione siano <20ms ?
3) Come faccio ad ottenere una riproduzione sincrona anche in presenza di jitter?
4) Qual'è il legame tra ritardo di riproduzione e la perdita di pacchetti?

1) e 2)



1) Se i pacchetti stanno in posizioni diverse nel router, verrano quindi smaltti in un tempo breve m avranno tra di loro in ricezione con un tempo > 20ms

2) Se i pacchetti di uno stesso flusso sono accodati uno dopo l'altro nel routerm passono essere smaltti con un tempo maggiore e arriveranno in ricezione con un tempo tra di loro < 20 ms

3) Ottengo riproduzione sincrona utilizzando: il numeri di sequenza che viene incrementato ad ogni pacchetti, ho bisogno anche della marcatura temporale,cioè vado ad indicare il tempo al quale quel pacchetto è stato generato cosi che se ho dei ritardi o delle perdita posso risalire all'istante in cui il pacchetto è stato generato, e per finire introduco anche un ritardo di riproduzione che può essere fisso oppure adattivo in base alle esigenze. Solo in questo modo posso rimuovere il jitter e avere una riproduzione sincrona.

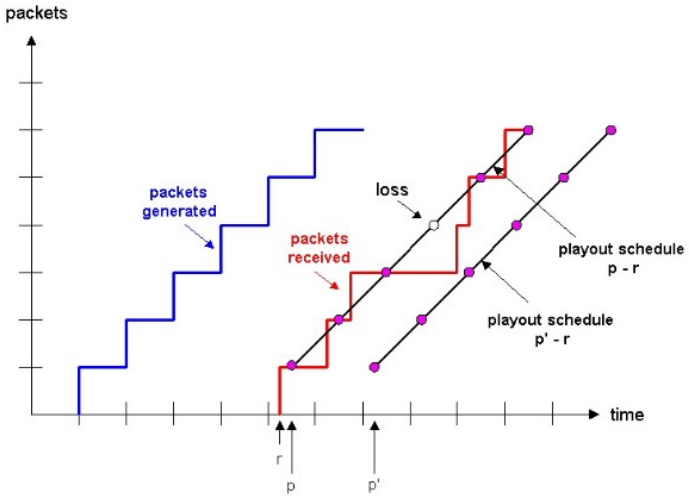
Avrò che se
t2-t1= 20 ms (s2-s1) potrò avere un periodo di no silenzio ma possibili perdite

dove s2,s1 sono numeri di sequenza
oppure
se t2-t1 > 20ms (s2-s1) avrò un perido di silezio ma sempre possibili perdite

4) Supponiamo che il trasmittente generi pacchetti ad intervalli regolari, ogni 20 ms. Il primo pacchetto è ricevuto al tempo r, mentre quelli successivi sono equidistanti a causa del jitter.
Supponiamo che in un primo caso il pacchetto venga riprodotto ad un instante p, molto vicino ad r, quindi ritardo di riproduzione sarà p-r. Con questo valore il 4 pacchetto (vedi figura) non arriverà entrò il tempo programmato per la riproduzione e il ricevente lo considererà perso.

Invece in un secondo caso il pacchetto venga riprodotto ad un istante p', non troppo vicino ad r, ora il ritardo di riproduzione sarà p'-r. I pacchetti arriveranno entro i tempi programmati per la loro riproduzione e non ci saranno perdite.

Nel primo caso abbiamo una perdita causata dal fatto di aver scelto un "q", un ritardo di riproduzione, troppo vicino alla ricezione del pacchetto e quindi diversi pacchetti arrivano tardi , non verranno trasmessi. Invece nel secondo caso avendo scelto un "q" ragionevole si avrà che i pacchetti arriveranno tutti in tempo per essere poi riprodotti normalmente.



32)
1)Per quale motivo il ritardo di riproduzione adattivo è utile ?
2)Come stimo il ritardo di riproduzione d_i?
3)Come stimo l'istante di riproduzione p_i?
4)Supponendo che un pacchetto sia inviato ogni 20ms come determino il primo pacchetto di un periodo di riproduzione? In caso di perdite?

1)Il ritardo di riproduzione adattivo è utile in quanto permette che le pause dei trasmittenti siano compresse o prolungate, secondo la necessità.

2)Il ritardo di riporduzione $d_i=(1-u)d_{i-1}+u(r_i-t_i)$, dove u è una costante fissa, (r_i-t_i) è il ritardo end-to-end dell' i-esimo pacchetto. Questo vuol dire che d_i è una media livellata dei ritardi di rete osservati $r_1-t_1,..,r_i-t_i$

3)L'istante di riproduzione $p_i=t_i+d_i+Kv_i$, dove K è una costante positiva e lo scopo del termine Kv_i è quello di impostare l'istante di inizio della riproduzione con sufficiente ritardo in modo da consentire che solo una piccola frazione dei pacchetti durante il periodo di attività vada persa a causa del ritardo.

4)Determino il primo pacchetto dal timestamp, cioè dalla marcatura temporale, infatti in questo modo è possibile arrivare a vedere il tempo in cui è stato generato. Nel caso delle perdite vedo il numero di sequenza.

33) Si consideri un meccanismo di registrazione weighted fair queueing presso uno switch su cui incidono 3 flussi di differenti classi di traffico. I flussi delle tre classi sono descritti da una specifica leaky bucket del tipo: b1=20, r1=100; b2=60, r2=200, b3=20, r3=400. (Se la dimensione non è specificata i pacchetti sono di 1KB)

Le classi sono ordinate secondo priorità crescente. Lo switch ha massimo rate disponibile di 800 pacchetti al secondo.

Assegnare i pesi alle tre classi in modo tale che siano soddisfatti i requisiti sulla specifica del traffico per ogni flusso ed il ritardo allo switch non superi i 200 msec.
b1=20 r1=100

b2=60 r2=200
b3=20 r3=400

BitRate=R=800 bps
dmax=ritardo per ogni flusso<=200 msec

Sapendo che dmax= $b_i/(R*w_i/(\sum w_i))<=200$
b1/(R*w1/w1+w2+w3)
→ w1/(w1+w2+w3)=1/8*10^-3

→ w2/(w1+w2+w3)=3/8*10^-3

→ w3/(w1+w2+w3)=1/8*10^-3

→ w1=w3

w1/w2=1/3

quindi w1=1, w2=3,w3=1
wtot=5

DA CONTROLLARE

34) Si consideri un'applicazione telefonica su Internet con codifica PCM ed un ritardo di trasmissione medio di 100 msec.
1. Si progetti una codifica che permetta di recuperare un pacchetto perso ogni 4 pacchetti trasmessi.
2. Determinare il minimo ritardo di playout necessario per la trasmissione.
3. Determinare la perdita di banda introdotta dalla codifica.
4. Si derivi dal precedente uno schema basato sull'interleaving con tolleranza a burst di 2 pacchetti.
5. Come aumenta il ritardo di playout?

1) Supponiamo che di avere 5 blocchi

Blocco1=<F1, P11, P12,P13,P14>
Blocco2=<F2, P21, P22,P23,P24>
Blocco3=<F3, P31, P32,P33,P34>

Blocco4=<F4,P41,P42,P43,P44>
Blocco5=<F5,P51,P52,P53,P54>

Questo vuol dire posso tollerare solo un errore nel primo blocco un errore nel secondo blocco,un errore nel terzo blocco e un errore nel quarto blocco.

Dopo il riordino

Blocco1=<F1, F2, F3, F4, F5>
Blocco2=<P11, P21, P31,P41,P51>
Blocco3=<P12, P22, P32,P42,P52>
Blocco4=<P13, P23, P33,P43,P53>
Blocco5=<P14, P24, P34,P44,P54>

Immaginiamo ora di perdere TUTTO il blocco 2.

Adesso lo possiamo recuperare.

Con F1 recupero P11, con F2 recupero P21, con F3 recupero P31,con F4 recupero P41,con F5 recupero P51

Siamo tolleranti alla perdita di 1 burst su 5 (5 pacchetti su 25) lo spreco di banda è 1/5

Il minimo ritardo riproduzione è determinato dal ritardo di trasmissione + ritardo per fare l'interleaving + ritardo del riordino + eventuale correzione

DA CONTROLLARE E FINIRE

35) Che differenza c’è tra il buffer TCP e il buffer dell’applicazione?

Il buffer TCP riceve byte del video richiesto dal buffer TCP invio del server, dopo di che il client legge i dati dal buffer TCP e li mette nel buffer dell’applicazione client, il quale dopo un tot di byte presi trasmetterà il video.
Se il buffer dell’applicazione è pieno smetto di leggere pacchetti da buffer TCP finchè non si crea spazio.

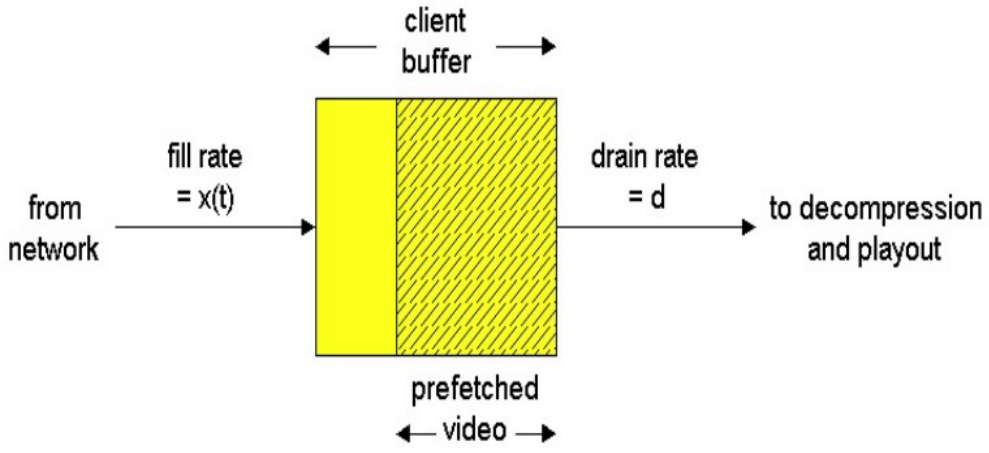
36) Consideriamo una sessione RTP con 4 utenti che inviano e ricevono pacchetti RTP allo stesso indirizzo multicast. Ogni utente spedisce video a 100Kbps

1) A quale ritmo RTCP limiterà il suo traffico?
2) Quanta banda RTCP sarà allocata al ricevente e quanta al trasmittente?

1)
T_{sender} = (# senders * avg RTCP pkt size)/ (0.25 * 0.05 * RTP bandwidth)
T_{rcvr} = (# receivers * avg RTCP pkt size)/(0.75 * .05 * RTP bandwidth)

2) la banda totale usata è 4(numeroutenti) * 100Kbps (BitRate) = 400Kbps
5% per RTCP = 20Kbps
questo perché ciascun utente è sia ricevente che trasmittente quindi 20/4=5Kbps (in generale 25% trasmittente, 75% ricevente)

37)
1)Cosa succede se uso TCP ?
2)Cosa succede se usiamo TCP, la larghezza di banda TCP è molto maggiore di d e il buffer può contenere solo 1/3 delle dimensioni del file ?



1) x(t) fluttua nel tempo a causa di controllo di congestione
Inoltre vi può essere una perdita pacchetti quindi che x(t) può essere inferiore a d per molto tempo.

2) Quando il buffer è pieno (o semi-pieno) blocco o rallento x(t) questo implica una fluttuazione di x(t) intorno a d quindi un aumento probabilità di starvation

DA CONTROLLARE

38) Si illustri lo streaming attraverso il protocollo DASH e lo schema dell'infrastruttura Cloud e CDN adottato da Netflix.

- Lo streaming DASH o detto anche Streaming HTTP adattivo consiste nel fatto i video vengono codificati in diverse versioni in base alla larghezza di banda disponibile per quel Client che richiede il video.
I video sono memorizzati in un server HTTP, il quale ha anche un file di descrizione contenente la versione del video e l'URL corrispondente.

Sapendo che stiamo usando un protocollo HTTP, che utilizza TCP, si dovrà andare a stabilire una connessione TCP, questo comporterà al fatto che il video verrà inviato utilizzando un buffer TCP di invio del server,che memorizzerà i dati del video man mano.

Il server invierà i byte che si stanno accumulando nel buffer invio TCP al buffer di ricezione TCP del client ,i quale oltre al buffer TCP di ricezione ha anche un application buffer che leggerà i byte del TCP buffer di ricezione tramite la socket. Dopo che nell'application Buffer si saranno raggiunti tot byte per la riporduzione del video, il video verrà trasmesso all'utente.

- Con cloud computing in particolare si indicano una serie di tecnologie che permettono di elaborare, archiviare e memorizzare dati grazie all'utilizzo di risorse hardware e software distribuite nella rete.

L'innovazione apportata dalle configurazioni cloud riguarda la distribuzione in rete dei servizi, la semplice scalabilità dell'infrastruttura, la maggiore affidabilità e continuità del servizio e l'erogazione in tempi molto rapidi di nuove risorse di calcolo e memorizzazione.

Quando diciamo servizi “cloud” stiamo parlando di server pilotati da un software che ne mette a disposizione le capacità di calcolo (CPU) e di memorizzazione (dischi); i servizi forniti vengono dislocati automaticamente tra tutti i server disponibili e in caso di necessità nuovi server possono essere facilmente aggiunti per aumentare la capacità complessiva del sistema.

Il cloud computing prevede tre modelli di servizio principali:
1) Software as a Service – SaaS

- Il cloud fornisce servizi (applicazioni) – che vengono utilizzati direttamente dall’utente finale

2) Platform as a Service – PaaS

- Il cloud fornisce servizi che sono piattaforme runtime, che supportano l’esecuzione di programmi – ad es., una piattaforma per l’esecuzione di applicazioni web

- l’utente è uno sviluppatore di applicazioni per quella piattaforma

3) Infrastructure as a Service – IaaS

- il cloud fornisce servizi infrastrutturali – come server (CPU e sistemi operativi), storage e connettività
- l’utente è uno sviluppatore o un amministratore di sistema per un’infrastruttura che va configurata

- La CDN adottata da Netflix usa la stategia del cluste “più vicino” al Client,detto anche AnycastIP, che consiste nel utilizzare un protocollo BGP per andare a stabilire il percorso dall’utente verso un qualsiasi cluster della CDN, le politiche riguardo la scelta del percorso più breve sono nella tabella di inoltro preconfigurata da BGP.

Infatti il Client dopo aver fatto una richiesta ad un video, il DNS della CDN gli restituisce l’indirizzo anycast. Il client invia un pacchetto a

quell’IP, il pacchetto verrà instradato al cluster identificato come il più vicino.

CASO NETFLIX:

Bob accede a netflix tramite il server di registrazioe di Netflix dopo di che e richiede un determinato video ad Amazon cloud egli gli restituira il manifest (file che contiene le diverse verisioni per quel video), nel frattempo vengono caricate le copie delle diverse versioni ai server delle diverse CDN. A quel punto Bob richiederà la versione a lui conveniente dato la sua larghezza di banda disponibile e riceverà quindi il video richiesto dal cluster a lui più vicino utilizzando la strategia dell’AnycastIP.