

Peer-to-Peer video streaming

Le dimensioni di un sistema di p2p video streaming

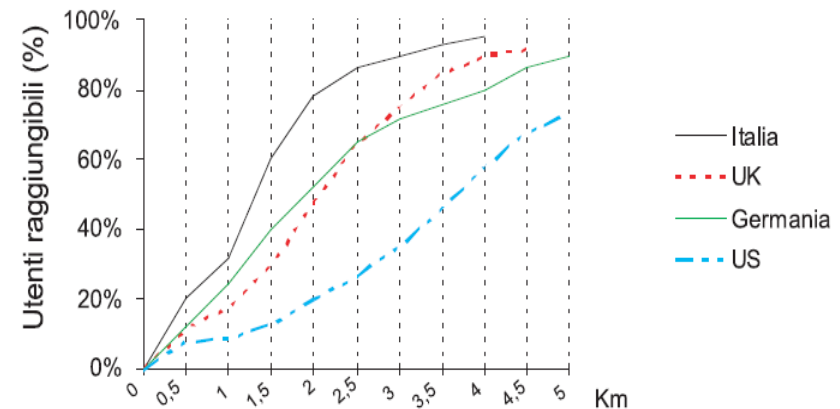
- 1. struttura della distribuzione
- 2. costruzione della topologia
- 3. topologia
- 4. sorgenti
- 5. coordinamento e startup
- 6. join
- 7. leave
- 8. performance
- 9. crittografia

Fattori abilitanti dei sistemi p2p video streaming

- Banda larga residenziale: ADSL, la tecnologia più diffusa, permette la distribuzione di contenuti video in modalità streaming
- Potenza dei personal computer di fascia medio-bassa, che permette di realizzare le funzionalità di nodo della rete peer-to-peer

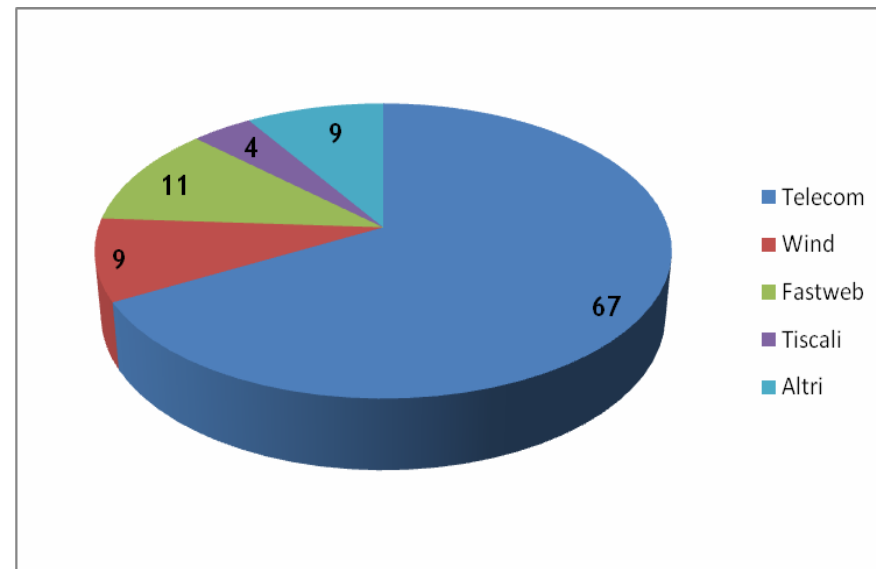
Le tecnologie ADSL

- Le tecnologie xDSL utilizzano il tradizionale doppino telefonico in rame
- La capacità trasmissiva dipende sia dal numero di utenti collegati, sia da fattori fisici come la sezione dei cavi di rame, la lunghezza e le interferenze che questi sistemi presentano
- I doppini telefonici riescono a dare massima efficienza ed elevata velocità se la distanza da percorrere, tra l'utente e le centrali Telecom, risulta breve (inferiore a 3 Km)
- In Italia si è deciso di puntare su questo tipo di tecnologia perché, al contrario degli altri paesi, nella maggior parte dei casi questa distanza non supera i 2 Km



UtENZE ADSL per provider

- In figura è riportata la ripartizione (2007) degli utenti ADSL in Italia

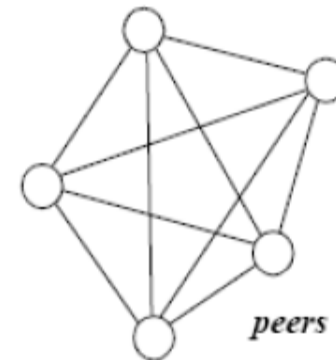
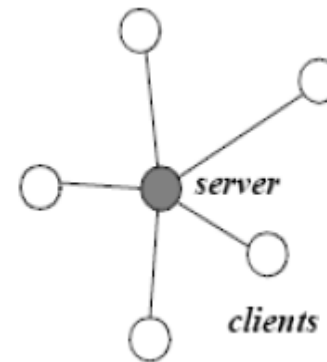


Alcune configurazioni ADSL disponibili sul mercato

Operatore	Download	Upload	
Fastweb (ADSL)	20	1	
Fastweb (Fibra)	10	10	
Telecom (Alice 4)	4	0,256	
Telecom (Alice Free)	0,640	0,256	
Telecom (Alice 20)	20	1	
Tiscali 4	4	0,256	
Tiscali 12	12	1	
Tiscali 24	24	1,5	
Wind 4	4	0,256	
Wind 12	12	1	

I sistemi peer-to-peer

- I sistemi peer-to-peer superano la classica struttura client-server delle applicazioni distribuite
- Nei sistemi peer-to-peer, ogni peer agisce sia da client che da server



Peer-to-peer: pro e contro

- Si basano sui protocolli http, ftp, etc.. quindi lo scambio di informazioni può essere gestito attraverso i router ed i firewall
- Permettono una migliore gestione della banda disponibile, cosa che non è possibile con il modello client/server che genera rallentamenti all'aumentare del numero di client
- Garantiscono una elevata scalabilità del sistema: un aumento del numero di peer non provoca problemi di gestione o di traffico nella rete
- Non si ha la necessità di avere amministratori di rete
- Sono molto tolleranti ai guasti proprio perché nessun nodo è indispensabile al funzionamento dell'architettura
- Non è sempre facile trovare i dati e le informazioni cercate, questo comporta la necessità di implementare nuovi sistemi di ricerca
- Problemi riguardano anche la difficoltà nel fermare la diffusione di materiale protetto da copyright
- Essendo una rete aperta è facilmente attaccabile da hackers
- Permette la diffusione di file malevoli come virus, spyware, materiale pedo-pornografico
- La banda disponibile in upload dei peer può essere un collo di bottiglia

P2P video streaming e P2P file sharing

	Streaming	Download
Server	È richiesto uno streaming server	È sufficiente un normale web server
Protocollo di rete	TCP/UDP/IP	TCP/IP
Protocollo applicazione	RTP/RTSP RTCP	HTTP
Perdita pacchetti	Possibile la perdita pacchetti	Non possono essere persi dei pacchetti
Time performance	Real time. La durata del media trasferito è uguale all'originale	I pacchetti possono essere ritrasmessi
Qualità del media	Alcuni pacchetti possono essere eliminati in funzione dei vincoli di tempo e banda	Alta qualità garantita
Connessione dell'utente	Può corrispondere con la banda dell'utente	Il file viene distribuito senza tener conto della banda dell'utente
Playback	La visualizzazione parte immediatamente	La visualizzazione parte dopo lo scaricamento del file
Impegno	Maggior sforzo per il provider (sono richiesti nuovi servers e connessioni veloci)	Maggior sforzo per l'utente finale (spazio su HD e velocità di connessione)
Firewall	Possono ostacolare la visualizzazione del media	Dipendono dalla configurazione ma in genere non costituiscono problemi
Memorizzazione	I file non vengono memorizzati nel PC dell'utente	I file vengono memorizzati

Le dimensioni di un sistema di p2p video streaming

- 1. struttura della distribuzione
- 2. costruzione della topologia
- 3. topologia
- 4. sorgenti
- 5. coordinamento e startup
- 6. join
- 7. leave
- 8. performance
- 9. crittografia

1. Struttura della distribuzione

- end-to-end overlay,
 - in cui ogni client, end-host, scambia il contenuto con gli altri e gestisce la topologia per la distribuzione. In questo caso la definizione di peer è appropriata, infatti tutti i nodi hanno la stessa funzione all'interno del sistema.
- proxy-based overlay,
 - in cui si definisce una struttura gerarchica e solo alcuni nodi, i multicast nodes, sono utilizzati per la distribuzione, e formano una backbone overlay. In questo caso la maggior parte dei peer riceve solamente il contenuto senza inoltrarlo, quindi sono foglie nella topologia di distribuzione. Il collegamento tra i nodi della backbone overlay e i peer finali può essere a sua volta in modalità multicast, anche multicast IP se i client fanno parte di una stessa rete, oppure unicast.

Struttura della distribuzione

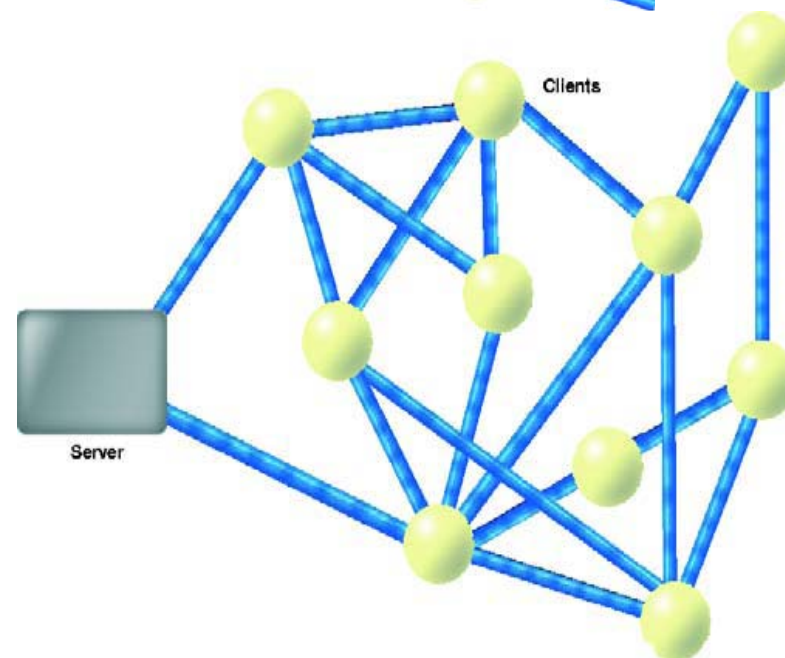
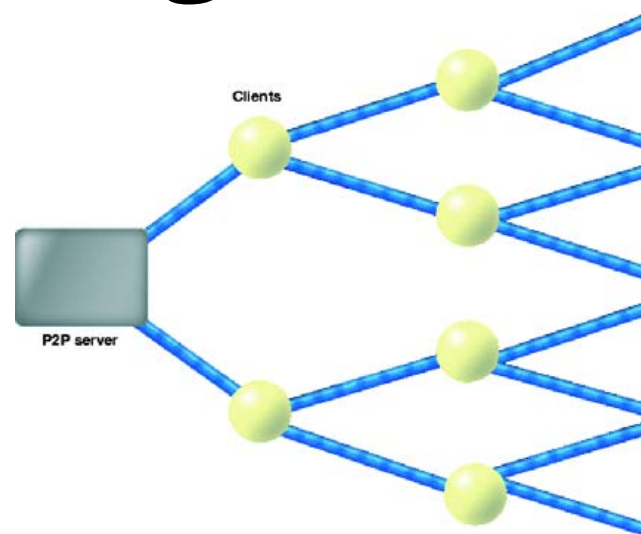
- La struttura proxy-based ha il vantaggio di diminuire l'overhead necessario per la gestione della topologia in quanto è limitata solo ad un sottoinsieme di peer, i multicast node
- I multicast node possono essere dei normali client con banda in upload adeguata oppure veri e propri server.
- Avere SCX (ScatterCast proXies), multicast nodes che sono parte della rete gestita dagli ISP, permette una maggior scalabilità della rete rispetto ai sistemi end-to-end dove non c'è controllo da parte del gestore sullo sviluppo della rete overlay
- I sistemi end-to-end hanno lo svantaggio di aumentare il ritardo, in quanto la conoscenza della rete è distribuita, ed è necessario uno scambio di informazioni di controllo cospicuo, ma sono più adatti per gestire situazioni in cui la topologia varia velocemente
- I sistemi end-to-end overlay e proxy-based overlay vengono chiamati anche application level multicast (ALM) e overlay multicast

2. Costruzione della topologia

- diretta:
 - quando un nodo entra nel sistema gli viene assegnato un parent a cui collegarsi per recuperare il contenuto. In questo modo il nuovo nodo è inserito immediatamente all'interno del processo di distribuzione senza passare da fasi intermedie. Molto spesso con questo sistema sono costruite topologie ad albero
- indiretta a maglia:
 - in una prima fase i peer si scambiano informazioni sulla loro posizione e sulla disponibilità dei contenuti, successivamente sfruttando questa conoscenza possono reperire dai loro vicini i dati
- indiretta a cluster gerarchici:
 - i nodi prima sono suddivisi in livelli gerarchici e in cluster, e successivamente sfruttando questa suddivisione è creato un albero di distribuzione che attraversa tutti i livelli della gerarchia

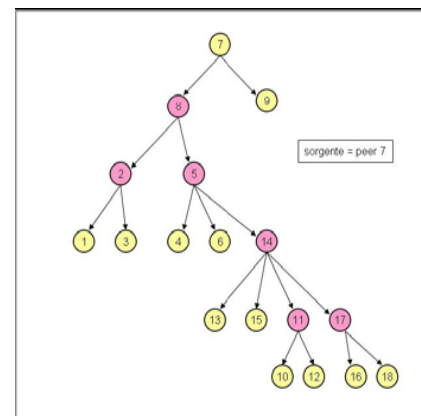
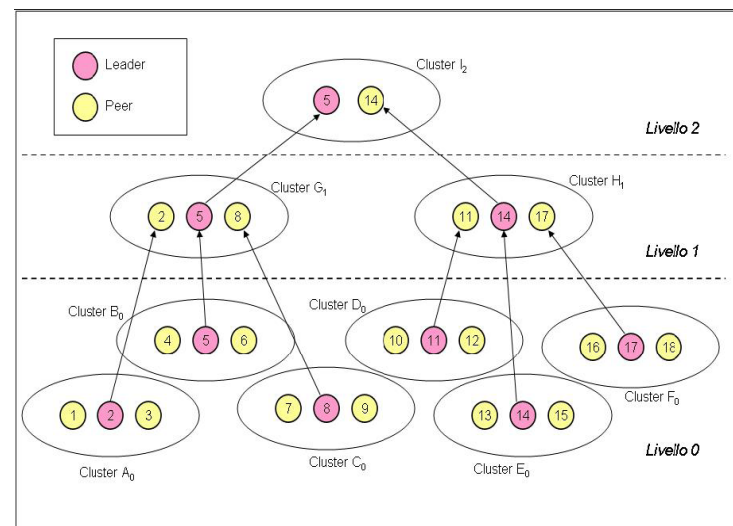
3. Topologia

- *albero singolo*: costruito in maniera diretta e indiretta;
- *foresta*: più alberi con la stessa radice;
- *generica*: in cui non è costruita una topologia ben definita (per



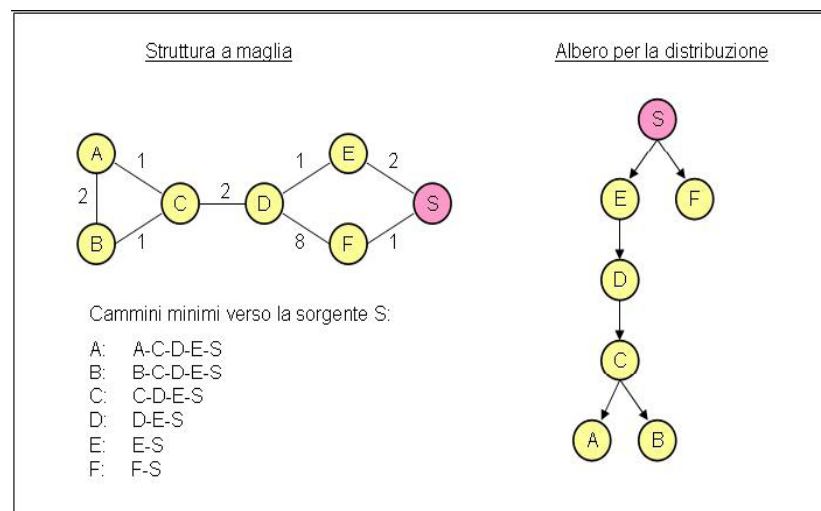
3. Topologia ad albero con costruzione indiretta a cluster

- Si parte dal nodo radice, che si trova a livello più alto, e questo invia il contenuto a tutti i suoi clustermate, cioè gli altri peer che si trovano nei suoi stessi cluster in ogni livello
- Ogni peer che riceve il contenuto lo inoltra a tutti i suoi clustermate, a meno che non abbia ricevuto il media da un peer del suo cluster
- Per cui in ogni cluster un solo peer invia il contenuto agli altri
- In figura si mostrano una struttura gerarchica a cluster con tre livelli e dimensioni del cluster pari a 3



3. Albero con costruzione indiretta maglia

- Data la maglia, cioè la conoscenza di alcuni peer presenti nel sistema, i percorsi per la distribuzione del media sono calcolati usando un algoritmo di tipo distance vector
- Ogni membro mantiene nelle sue tabelle il percorso e il costo del percorso per raggiungere ogni altro membro, informazioni che sono scambiate con i messaggi di update
- L'instradamento è source-based
- La metrica per definire il costo di un collegamento è scelta in base all'applicazione
- di riferimento, ad esempio potrebbe essere il ritardo del collegamento o
- l'inverso della banda disponibile.



3. Foresta (alberi multipli)

- In una topologia ad alberi multipli, sono creati più alberi disgiunti, ma con la stessa sorgente, verso gli altri nodi, per aumentare la ridondanza nei percorsi di rete
- Così è superato il limite dell'albero singolo, infatti il fallimento di un singolo nodo provoca l'interruzione della distribuzione su un solo albero
- I peer, facendo parte anche di altri alberi, non sono quindi esclusi da tutto il contenuto
- Questa proprietà aumenta la robustezza dell'albero ed è chiamata tree diversity
- Gli alberi sono costruiti in modo che siano corti ma ramificati.
- Alberi corti consentono di minimizzare il tempo di latenza e la probabilità di rimanere isolati a causa dei leave dei parent
- L'outdegree del nodo, cioè il numero di figli che può supportare è fortemente limitato dalla sua banda in upload

3. Foresta (alberi multipli)

- Un altro approccio nell'utilizzo degli alberi multipli è il seguente:
 - si può suddividere lo stream da trasmettere in più sub-stream che sono tra loro non sovrapposti, così da ridurre il rate da trasmettere su un singolo albero
 - Infatti il problema principale che bisogna superare è la banda asimmetrica della maggior parte dei client
 - Non è necessario ricevere tutti i sub-stream

3. Topologia generica

- Ogni nodo deve gestire due diverse funzionalità:
- scambio delle informazioni con i vicini per stabilire una maglia tra i peer, e scambio dei contenuti
- Ogni video stream è suddiviso in segmenti, chunk
- I nodi possono scambiarsi i segmenti
- La dimensione dei segmenti dipende dal bit rate dello stream ma in genere non supera il valore di 10 KByte
- Ogni nodo mantiene una lista dei segmenti che ha a disposizione, la Buffer Map (BM), e un'altra lista che permette di rintracciare i peer che hanno un certo segmento, la membership cache

3. Topologia generica

- La Buffer Map è l'elenco dei segmenti che un nodo ha disponibili, entro un
- certo ritardo, e contiene:
 - offset, cioè l'identificativo del primo segmento (2 o 4 byte);
 - length della buffer map;
 - una stringa di 1 e 0 per indicare la disponibilità dei segmenti.
- Ad esempio se si considerano segmenti lunghi 1 secondo, la BM può contenere 120 segmenti, essendo gli altri troppo lontani nel tempo per interessare gli altri peer
- La Buffer Map è una finestra a scorrimento di 120 bit, che vale 1 se il segmento è presente e 0

3. Topologia generica

- Il processo di costruzione dell'albero è sostituito da un processo di scheduling
- Una volta che il nodo ha a disposizione la Buffer Map può rintracciare i peer supplier a cui richiedere i segmenti mancanti
- Il nodo, avendo a disposizione la sua BM e quella dei suoi partner, deve stabilire da quali nodi prelevare i vari segmenti con un algoritmo di scheduling che rispetti il vincolo di dead-line del segmento
- Questo è un problema della classe NP-hard per cui sono sviluppate delle approssimazioni subottime

3. Topologia generica

- DONet/Coolstreaming utilizza un algoritmo euristico:
 - per ogni segmento mancante si contano i supplier disponibili
 - seguendo il principio intuitivo per cui è più difficile recuperare un segmento se esso ha meno sorgenti, è scaricato prima il segmento mancante con un solo supplier, poi quelli con due e così via
 - Nel caso siano disponibili più supplier è scelto prima quello con banda maggiore
 - L'elenco dei segmenti che devono essere recuperati da uno stesso supplier è memorizzato in un lista con struttura uguale alla BM, così essi saranno richiesti nello stesso momento ed in ordine

3. Topologia generica

- La chiave per il buon funzionamento dell'algoritmo è la stesura della mCache
- Nei messaggi di refresh sono contenuti:
 - seq num (numero di sequenza del pacchetto)
 - id (identificativo del nodo all'interno delle rete overlay)
 - N_partner (il numero di partner che ha attualmente, cioè di peer con cui scambia le buffer map e quindi i segmenti)
 - TTL (validità residua del pacchetto in secondi)

3. Topologia generica

- Alla ricezione di un messaggio di refresh, un nodo aggiorna la mCache, aggiornando o aggiungendo una riga per il nodo mittente, con le informazioni ricevute a cui associa anche un time stamp (last update time), il tempo locale in cui è stato ricevuto quell'aggiornamento
- Periodicamente ogni nodo cerca altri partner scegliendoli a caso dalla sua mCache, così il numero di partner rimane stabile (pari ad M), nonostante i leave, e i nodi sono in grado di trovare partner migliori
- La qualità di un partner j per il nodo i dipende dalla velocità con cui riesce a scambiare informazioni, è pari a $\max(r_{ij}, r_{ji})$, dove r_{ij} è il numero di segmenti scambiati per unità di tempo tra i partner i e j

3. Topologia generica

- Il numero M di partner è un parametro molto importante perchè ha delle conseguenze sull'overhead prodotto, aumentando M cresce il traffico per lo scambio delle buffer map, ma se M è maggiore cresce anche la percentuale di segmenti che arriva entro la loro deadline
- In genere si usano valori di M compresi tra 2 e 5
- Rispetto alle topologie ad albero, questo meccanismo, non avendo nessuna struttura da rispettare, si adatta meglio ai contesti in cui i peer eseguono join o

4. Sorgenti

- Le sorgenti sono i nodi tramite i quali il contenuto è immesso in rete
- Le sorgenti possono essere
 - singole
 - multiple.
- La maggior parte dei sistemi utilizza una sola sorgente iniziale del contenuto.
- Nel caso si faccia riferimento a sistemi che supportano la video-conferenza le sorgenti sono sempre multiple in quanto tutti i nodi che partecipano attivamente contribuiscono, ma ogni sorgente invia un contenuto diverso

5. Coordinamento e startup

- Nelle strutture overlay, centralizzate o distribuite, i nodi devono scambiare informazioni sullo stato dei peer per monitorare il processo di distribuzione dei contenuti
- Bisogna individuare i guasti dei nodi e porvi rimedio in maniera rapida, e si possono prendere delle decisioni per migliorare le prestazioni cambiando la topologia del sistema o la topologia dell'albero.
- Quando il controllo è centralizzato i peer non fanno altro che inviare delle "misure" al server che gestisce l'organizzazione
- Nei casi di controllo distribuito i peer devono trasmettere a tutti gli altri peer anche le informazioni di cambio della topologia
 - In reti con numerosi utenti vi è lo svantaggio di dover gestire una quantità elevata di informazioni sovraccaricando la rete di traffico di overhead
 - Questo problema può essere limitato utilizzando un controllo parzialmente distribuito, come nei sistemi proxy-based, in cui solo un sottoinsieme di peer ha la facoltà di prendere decisioni sulla gestione della topologia e quindi invia traffico di controllo

5. Sistemi centralizzati

- Nei sistemi centralizzati tutte le richieste giungono al server di startup ed è il server che stabilisce dove inserire il nuovo client nella topologia
- Insieme alla richiesta del contenuto, il nuovo peer invia nella maggior parte dei casi anche delle informazioni che riguardano il client, per far in modo che il server costruisca una topologia dei peer utilizzando delle informazioni aggiuntive
- Ad esempio, può essere utile conoscere la banda in uplink che un peer ha a disposizione, per capire quanti figli può accettare, o i ritardi tra un peer e l'altro per scegliere meglio gli accoppiamenti
- E' il server che prende tutte le decisioni
- Inoltre i peer possono periodicamente inviare al server dei dati sull'andamento della distribuzione per poter operare delle scelte di modifica della topologia, a causa di cambiamenti della rete
- In questa architettura gli scambi tra peer non sono ammessi, tranne quelli che riguardano lo scambio del contenuto
- In generale, un protocollo distribuito è più scalabile, ma il controllo centralizzato consente di conoscere l'intera topologia localizzando velocemente i contenuti e gestendo join e leave più rapidamente

5. Sistemi distribuiti

- Usando un protocollo distribuito i nodi non devono conoscere l'intera topologia, ma basta che ne conoscano soltanto una parte, così il protocollo risulta di più facile implementazione risultando più scalabile
- In CoolStreaming, la quantità di overhead è pari all'1% del traffico utile; inoltre le prestazioni restano pressochè le stesse aumentando il numero di peer
- I peer che appartengo alla maglia scoprono gli altri vicini tramite dei messaggi di discover e mantengono aggiornate le tabelle tramite messaggi di refresh, per seguire la dinamicità della rete underlay (proprietà di self-organizing e self-improving)

5. Sistemi misti

- Nei protocolli parzialmente distribuiti i peer scambiano una quantità di informazioni proporzionale al loro ruolo e non tutti i peer partecipano al processo di costruzione della topologia
- Questi protocolli si applicano in sistemi proxy-based
- Facendo riferimento ai sistemi gerarchici a cluster, ogni peer deve comunicare con tutti i suoi clustermate e solo con quelli
- Le comunicazioni al di fuori dei cluster sono gestite dai leader che devono memorizzare la topologia interna del loro cluster e scambiarla con i clustermate del livello superiore
- I peer si scambiano periodicamente informazioni di refresh, per poter individuare i cambiamenti all'interno della topologia e pianificare lo sviluppo della struttura e della topologia

6. Join

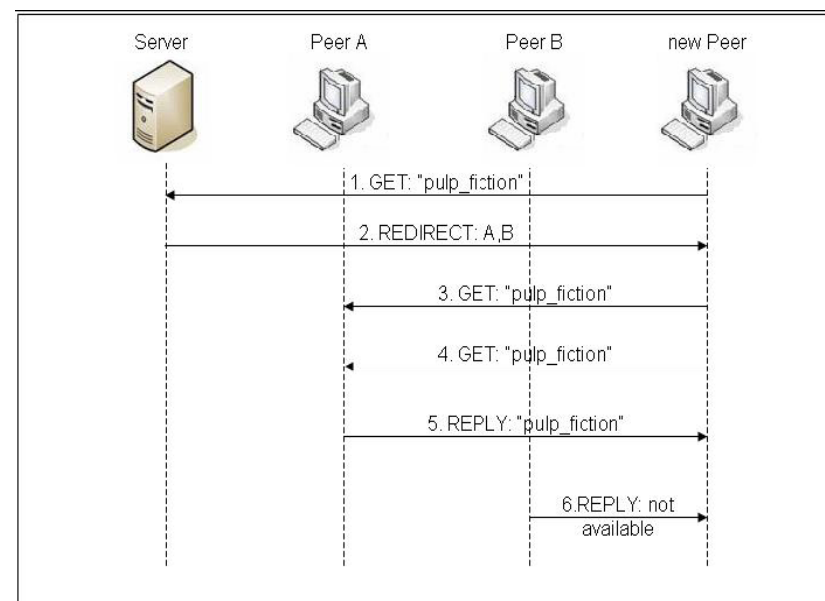
- Il processo di join è invocato quando un nuovo utente si collega al sistema ed effettua la richiesta del contenuto
- Se l'albero è costruito in maniera diretta, al nuovo nodo deve essere associato un parent da cui reperire il contenuto
- Nel caso di strutture più complesse il nuovo nodo deve essere anche inserito all'interno della topologia
- Il processo di join deve essere eseguito velocemente per non far attendere l'utente al momento della richiesta del servizio
- Nei sistemi con topologia ad albero il nuovo peer sarà un foglia dell'albero di
- Distribuzione
- Questo perchè al nuovo nodo è data la possibilità di ricevere subito il media, senza modificare la topologia esistente
- Successivamente è possibile modificare la sua posizione se il peer ha le caratteristiche giuste per essere un parent

6. Join

- In seguito ad un join, cioè ad un cambiamento nella rete overlay tramite l'introduzione di un nuovo nodo, molto spesso è necessario rilanciare nuove procedure
- Ad esempio nel caso di costruzione a cluster, se viene superata la dimensione massima del cluster, bisogna dividere in due quel cluster (cluster split)

6. Join, assegnamento diretto, il caso di COOPNET

- Nei sistemi che costruiscono la topologia in maniera diretta il nuovo peer ha bisogno solo di essere associato ad un parent.
- Quando il nuovo client si collega al server riceve le informazioni per poter reperire il parent a cui collegarsi
- Facendo riferimento a Coopnet, sistema con costruzione di una foresta in maniera diretta, l'operazione di join avviene nel modo seguente
 - Il nuovo nodo (new Peer) contatta il server e fa la sua richiesta. Nel messaggio di richiesta il client può fornire al server anche altre informazioni, quali la banda disponibile per l'upload
 - Il server, se non può gestire lui stesso la richiesta, invia un messaggio di re-direzione al client con una lista, con i relativi indirizzi IP, di peer (nella Figura i client A e B) a cui chiedere la trasmissione del contenuto. Il server deve memorizzare una lista di client per ogni file. Questa lista viene aggiornata ad ogni nuova richiesta
 - Il client contatta gli altri peer e riceve da loro il contenuto, poi si mette a disposizione per distribuire, in futuro, questo contenuto a nuovi nodi che arriveranno



7. Leave

- Il processo di leave è scatenato quando un nodo lascia il sistema. I suoi nodi figli rimangono orfani, cioè non ricevono più il media. Quindi il sistema, per essere efficiente, deve essere in grado di gestire questi eventi velocemente per rendere minimo il periodo di indisponibilità
- Nel caso di topologia ad albero la leave di un nodo ha effetti più marcati, in quanto isola tutto il sottoalbero con radice in quel nodo, dal processo di distribuzione.
- Il sistema deve trovare un nuovo parent per tutti i nodi rimasti orfani, cercando un nuovo parent per ogni nodo in maniera indipendente, oppure lasciando inalterata la struttura del sottoalbero

7. Leave

- La leave di un utente può avvenire in due modalità:
 - node failure, se la morte del nodo è improvvisa. Alcune possibili cause sono il crash, lo spegnimento o la disconnessione dalla rete
 - graceful departure, se il nodo avvisa il sistema della sua volontà di abbandonare e la rete può eseguire una procedura per sostituire quel nodo senza interruzione del servizio.
- Quasi tutti i sistemi usano una tecnica reattiva, cioè dopo che è avvenuto il node failure il sistema si assume il compito di riconfigurare la topologia.
- In questo caso i nodi orfani cercano di collegarsi o alla radice, che li tratta come join nodes, o al grandparent

7. Leave

- Failure detection: nel caso di node failure il sistema ha bisogno di una tecnica che gli permetta di accorgersi della failure
- Esistono due modalità:
 - keep-alive, periodicamente i vari nodi scambiano messaggi per confermare la propria presenza e operatività. Scaduto un time-out senza che siano stati ricevuti messaggi di heart-beat, il nodo è dichiarato morto ed è attivata la procedura per il ripristino della topologia
 - packet loss rate monitoring, tecnica utilizzata da CoopNet, in cui i nodi □figli controllano la percentuale di pacchetti persi e nel caso essa salga sopra una certa soglia il parent è dichiarato morto

Alcuni sistemi a confronto

NOME	1 struttura della distribuzione	2 costruzione topologia	3 topologia	4 sorgenti	5 coordinamento e startup	6 join	7.1 leave	7.2 failure detection	8 performance	9 crittografia	10 accademico commerciale
vidTorrent	end-to-end	indiretta a maglia	alberi multipli	singola	distribuito / startup centr.	annuncio ai vicini	react. - back to the root	keep-alive	segmentazione, underlay miming	no	acc
CoolStreaming	end-to-end	indiretta a maglia	richiesta diretta a più peers	singola	distribuito / startup centr.	annuncio ai vicini	react. - abandon	keep-alive	segmentazione	no	com
CoopNet	server centrale	diretta	alberi multipli	singola	centralizzato / startup centr.	foglia - assegn. diretto	react. - back to the root	packet loss monitoring	underlay miming, bandwidth matching, MDC, segmentaz.	no	acc
GridMedia	end-to-end	indiretta a maglia	richiesta diretta a più peers	singola	distribuito / startup centr.	annuncio ai vicini		keep-alive	push-pull, porta UDP nota	no	com
Narada	end-to-end	indiretta a maglia	albero singolo	singola	distribuito / startup centr.	annuncio ai vicini	react. - abandon	keep-alive	underlay miming	no	acc
Nice	proxy-based	cluster gerarchici	albero singolo	singola	parz. distribuito / startup centr.	foglia - cluster di livello 0	react. - back to grandparent	keep-alive	underlay miming	no	acc
Overcast	proxy-based	diretta	albero singolo	singola	parz. distribuito / startup centr.	foglia - assegn. diretto	react. - back to grandparent	keep-alive	backup tree	no	com
PPLive	end-to-end	indiretta a maglia	richiesta diretta a più peers	singola	distribuito / startup centr.	annuncio ai vicini	react. - abandon		segmentazione	no	com
Scattercast	proxy-based	indiretta a maglia	albero singolo	multipla	distribuito / startup centr.	foglia - proxy più "vicino"	react. - aggiornamento degli alberi	keep-alive		no	acc
SplitStream	end-to-end	indiretta a maglia	foresta	singola	distribuito / startup centr.	foglia - livello più alto	react. - back to grandparent	keep-alive	MDC, segmentazione	no	acc
Zig-zag	proxy-based	cluster gerarchici	albero singolo	singola	parz. distribuito / startup centr.	foglia - cluster di livello 0	react. - back to grandparent	keep-alive	eq. degree/banda	no	acc
PeerCast	end-to-end	diretta	albero singolo	singola	distribuito / startup centr.	foglia - livello più alto	react. - back to grandparent/root	keep-alive		no	acc

VidTorrent

- La documentazione di riferimento è la Master Thesis di Ilia Mirkin "Reliable Real-time Stream Distribution Using an Internet Multicast Overlay"
- Il codice sorgente e alcune slide di presentazione sono reperibili all'indirizzo:
<http://viral.media.mit.edu/index.php?page=vidtorrent>
- Il software è ancora in fase di sviluppo, inoltre il nome non è ancora definitivo
- Il sistema utilizza una struttura overlay sulla quale sono costruiti alberi multipli ognuno dei quali trasporta un sub-stream indipendente ricavato dallo stream principale
- La struttura è dinamica e self-healing
- Ogni albero è costruito in maniera tale che i nodi con maggior banda in upload si trovino ai livelli più alti, cioè abbiano un maggior numero di nodi figli
- L'associazione tra parent e figlio è fatta stimando la banda disponibile con un meccanismo di probing aggressivo e supponendo una vicinanza geografica sulla base del round trip time, RTT, tra i due peer

VidTorrent

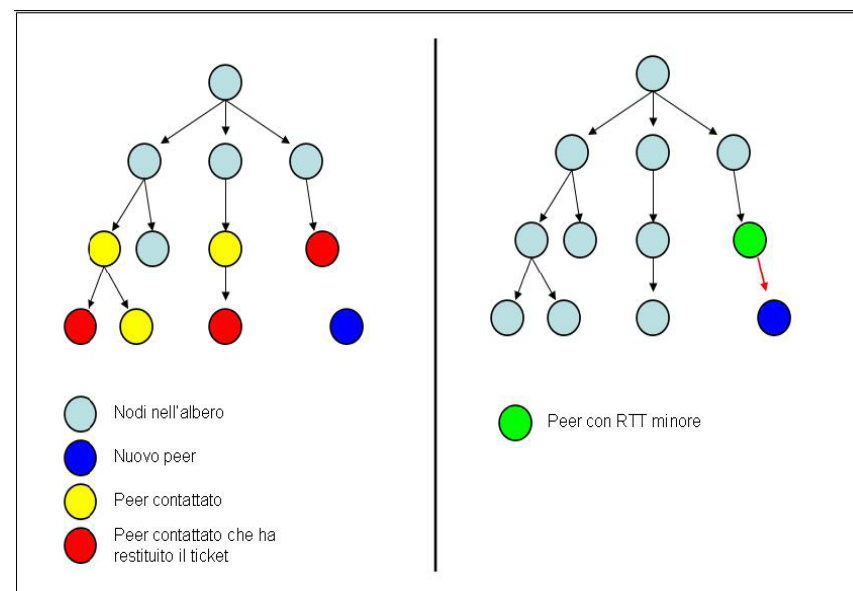
- Struttura end-to-end, tutti peer scambiano informazioni per gestire la topologia e partecipano al processo di distribuzione
- La costruzione della topologia avviene in maniera indiretta instaurando in una prima fase una struttura a maglia
- La topologia è ad alberi multipli, in cui ogni albero trasporta un sub-stream indipendente ricavato dallo stream principale
- Ogni albero è costruito in modo che i peer con maggior banda a disposizione si trovino ai livelli più alti
- La sorgente del media è unica
- Il coordinamento è distribuito
- Lo startup è centralizzato
- Il join avviene tramite annuncio agli altri peer e selezione del "miglior parent candidato", il quale è scelto in base alle stime di: banda disponibile tra i due con un meccanismo di probing e ritardo con stima del round trip time
- Gestione leave di tipo reactive: il peer rimasto isolato contatta direttamente la root oppure altri peer, di cui aveva conoscenza, per scatenare una re-join

VidTorrent

- Lo stream video è suddiviso dalla sorgente in più sub-stream indipendenti e non sovrapposti, cioè le stesse informazioni non sono contenute in più sub-stream
- Ogni sub-stream è distribuito mediante un albero, generando una topologia a foresta per l'intero stream
- Esiste un'associazione uno a uno tra sub-stream e albero
- Un peer, per poter ricevere l'intero stream, deve far parte di tutti gli alberi
- Se la sorgente suddivide il media in n sub-stream, il peer partecipa alla distribuzione in n alberi diversi, che compongono la foresta

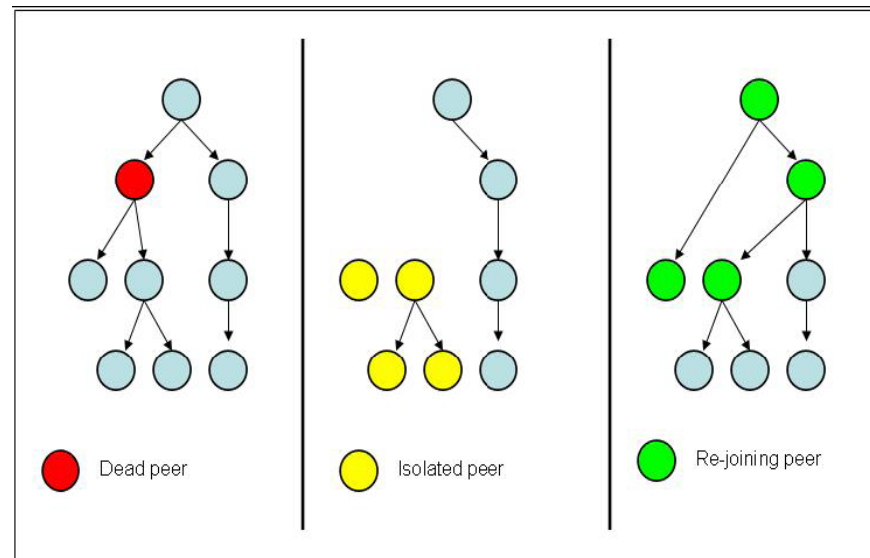
Join in VidTorrent

- Il join deve essere effettuato per ogni albero, cioè per ogni sub-stream
- Il processo di join in un albero avviene nel seguente modo:
- Il nuovo peer contatta il rendezvous-point, il quale risponde con una lista di peer che potrebbero essere adatti a fare da parent per quel sub-stream
- Il nuovo peer contatta questi peer candidati ed, eventualmente, riceve da essi informazioni su altri peer
- Il nuovo peer invia ad ognuno dei parent candidati due pacchetti, di lunghezza nota L [bit], consecutivi, tramite i quali il parent è in grado di stimare la banda disponibile tra i due
- Se questo valore stimato è sopra una soglia minima restituisce al nuovo peer un ticket di accettazione
- Tra i parent candidati da cui ha ricevuto il ticket il nuovo peer sceglie come parent quello con round trip time minore
- Il meccanismo di stima della banda è inefficiente, perchè soggetto alla variabilità delle condizioni di congestione delle rete, in quanto utilizza una sola misurazione per effettuare la stima. Ha il vantaggio di non sovraccaricare la rete di messaggi di controllo e di non aumentare in maniera sensibile il ritardo necessario al nuovo peer per collegarsi.
- Questa stima può essere calcolata anche in momenti successivi al join per monitorare le condizioni attuali delle rete e per apportare eventuali cambiamenti che possano migliorare le prestazioni.



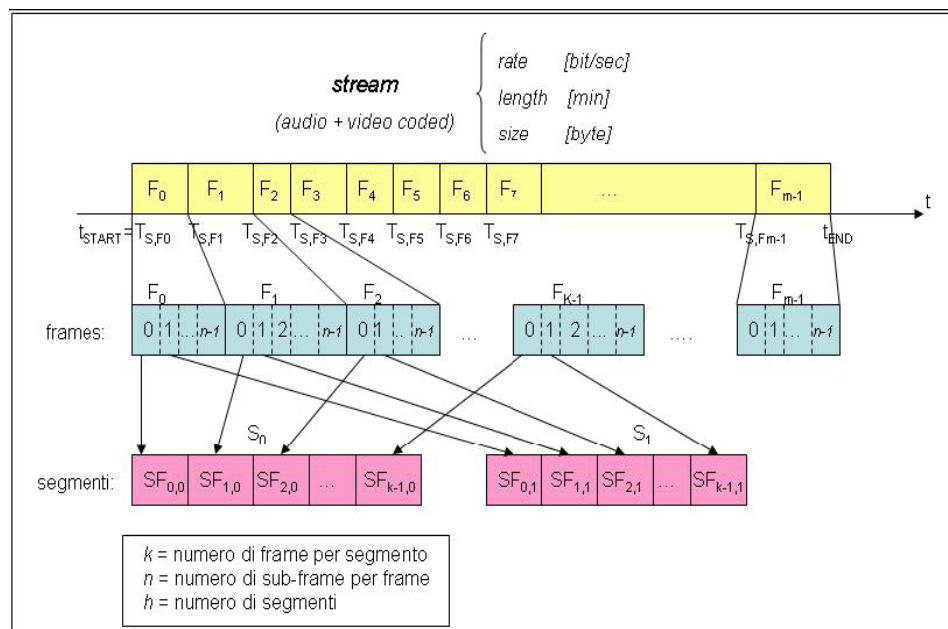
Leave in VidTorrent

- Nel caso di leave, cioè quando un albero rimane isolato, è eseguita una re-join contattando direttamente il rendezvous-point oppure altri peer noti grazie alle informazioni scambiate in precedenza
- La procedura di re-join è del tutto simile al join
- La struttura del sotto-albero, cioè la parte di albero che si dirama a partire dal nodo isolato, rimane inalterata e "migra" con il nodo



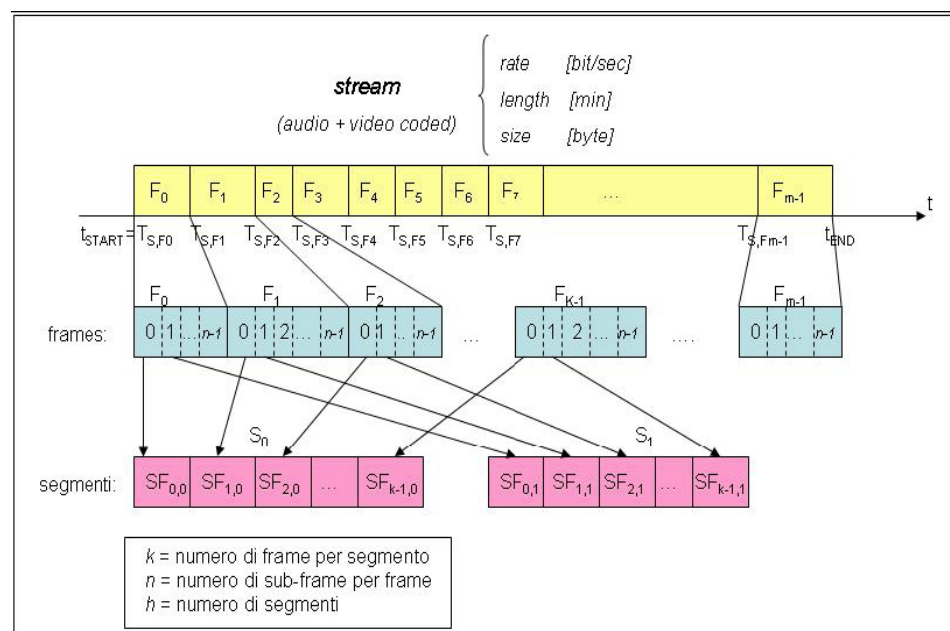
Suddivisione dello stream

- VidTorrent può supportare il trasporto di uno stream suddiviso in substream
- Da un punto di vista logico, la suddivisione di uno stream può essere rappresentata in modo gerarchico, per esempio
 - Segmenti
 - Frame
 - Subframe
- La suddivisione può rappresentare diversi layer di una codifica scalabile, diverse descrizioni di una codifica multiple-description oppure diverse “zone” di un video



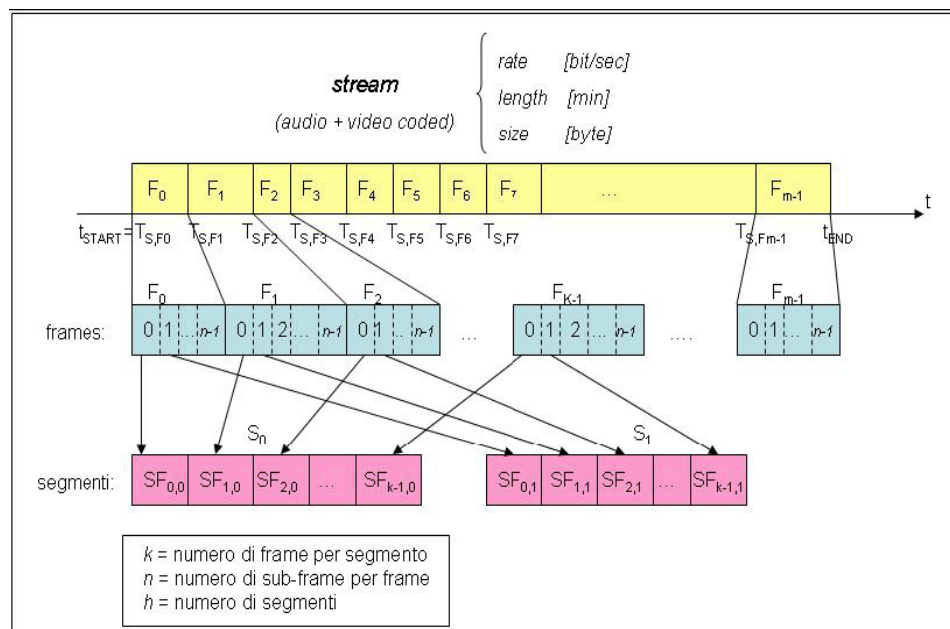
Suddivisione dello stream

- Nel caso di una codifica di tipo Multiple Description, la Frame F_i è suddivisa sub-frame ($SF_{i,j}$)
- Per riprodurre il contenuto è necessario che il client riceva almeno una sub-frame di ogni frame; ricevere più sub-frame di una stessa frame aumenta la qualità della riproduzione
- della perdita dei pacchetti in rete.
- Le sub-frame sono raggruppate dalla sorgente, mediante interleaving, in segmenti, in modo che ogni segmento contenga una sola sub-frame di una frame
- le sub-frame di una frame sono distribuite su segmenti consecutivi diversi
- Ogni segmento contiene lo stesso numero di frame, pari a k



Suddivisione dello stream

- La traccia è suddivisa in m frame, ogni frame è suddivisa in n sub-frame
- Il numero di sub-frame per frame è fisso
- I segmenti sono costruiti in modo che il segmento S_0 contenga tutte le sub-frame 0 delle prime k frame
- Il segmento S_1 contiene tutte le sub-frame con indice 1 delle prime k frame, e così via
- Se il numero di frame per segmento è $k = 3$, allora ogni segmento conterrà 3 sub-frame, e se il numero di sub-frame per frame è $n = 4$ allora saranno necessari 4 segmenti per trasmettere per intero le prime k frame
- Il segmento successivo, S_4 , conterrà le sub-frame 0 delle frame da 3 a 5, il segmento S_5 le sub-frame 1 delle stesse frame e così via

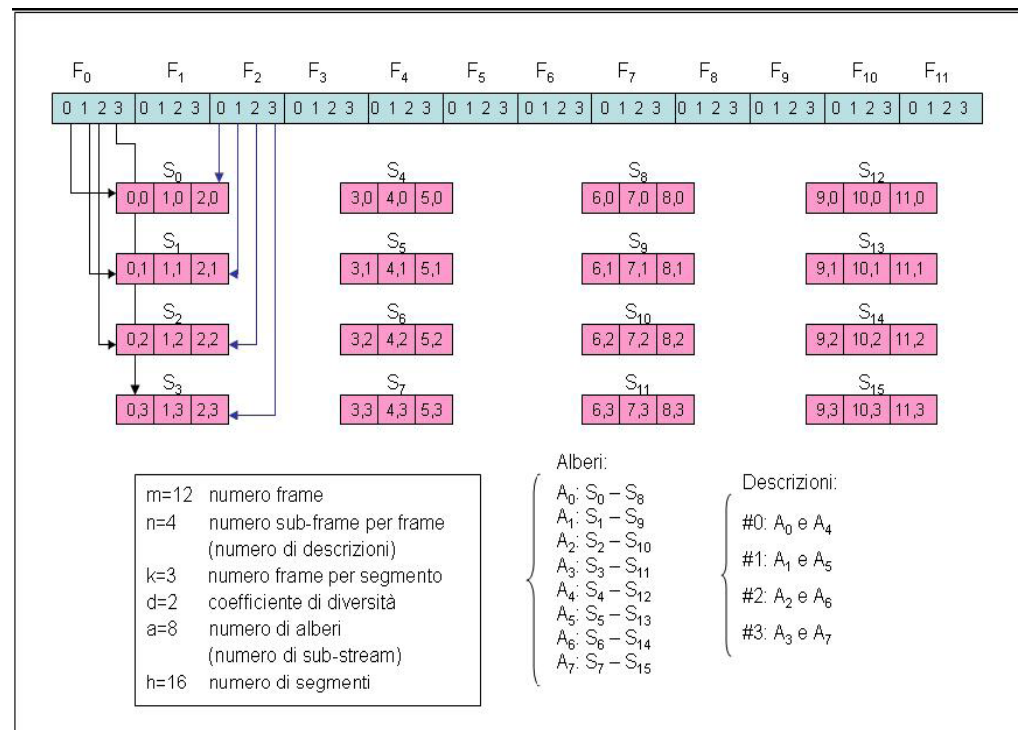


I sub-stream

- I segmenti sono associati ciclicamente ai vari sub-stream
- Un sub-stream è una sequenza di segmenti che sono trasmessi su uno stesso albero nell'ordine con cui sono generati
- All'interno del sistema il contenuto è distribuito su una topologia overlay a foresta con più alberi, ognuno dei quali ha radice nella sorgente e trasporta un sub-stream

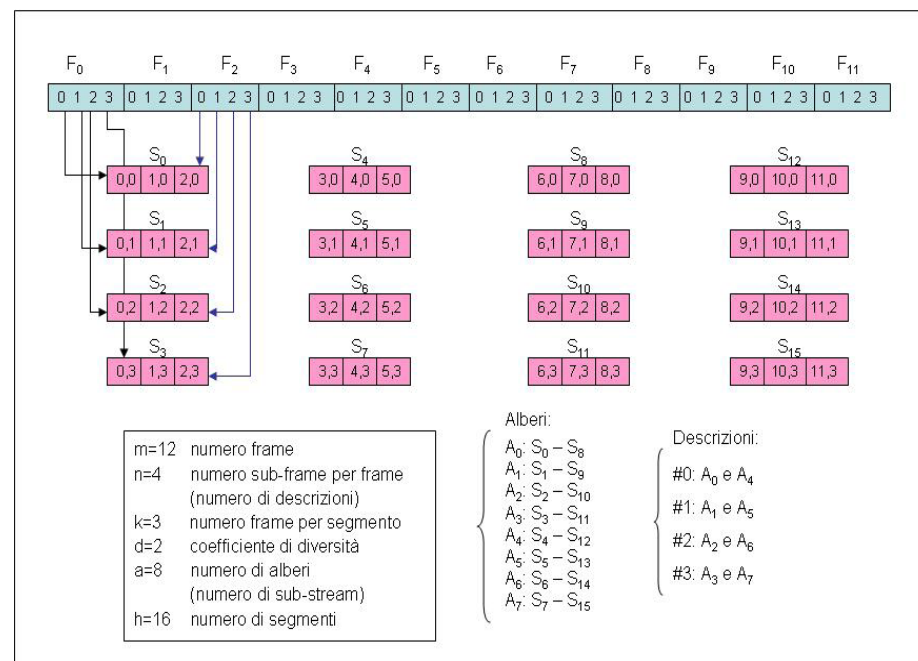
La diversità

- Un sub-stream può coincidere con una descrizione o essere una parte di essa
- Infatti, ogni descrizione può essere distribuita ai vari peer in un unico albero o su più alberi diversi, dipende dal parametro d , chiamato coefficiente di diversità
- Nel primo caso, corrispondente a $d = 1$, vi è un'associazione uno a uno tra alberi e descrizioni, mentre vi è sempre un'associazione uno a uno tra alberi e sub-stream.
- La ricezione di più sub-stream, e quindi di descrizioni, provoca un aumento della qualità della riproduzione
- Nel secondo caso, quando $d > 1$, una singola descrizione è distribuita tramite più alberi
- Il numero di descrizioni è sempre pari al numero di sub-frame per frame
- Ma il numero di sub-stream e quindi di alberi è un multiplo del numero di descrizioni.



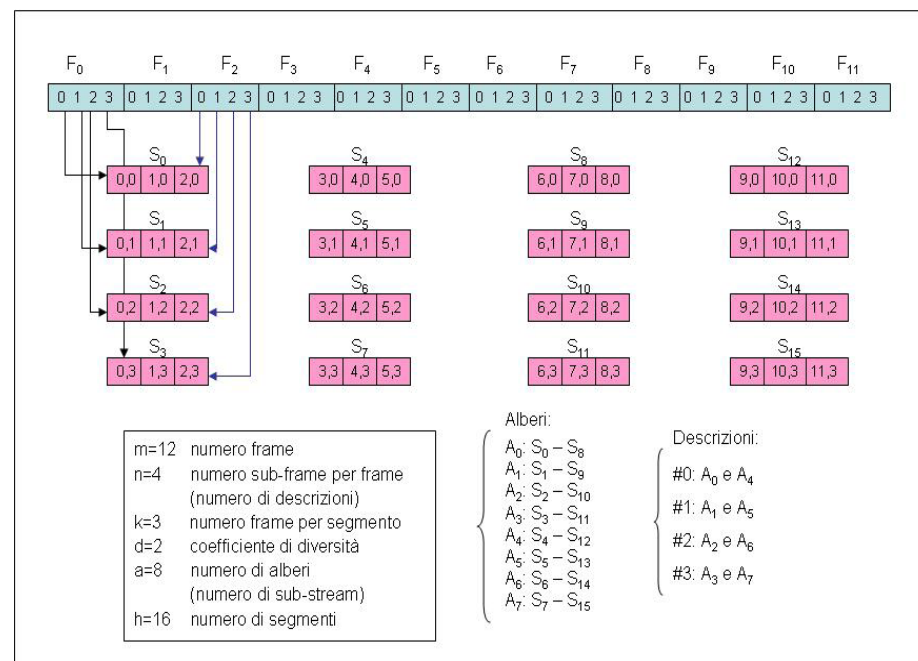
Diversità

- I segmenti sono associati ciclicamente ai vari sub-stream. Un sub-stream è una sequenza di segmenti che sono trasmessi su uno stesso albero nell'ordine con cui sono generati
- All'interno del sistema il contenuto è distribuito con una topologia a foresta con più alberi, ognuno dei quali ha radice nella sorgente e trasporta un sub-stream
- Un sub-stream può coincidere con una descrizione o essere una parte di essa
- Infatti ogni descrizione può essere distribuita ai vari peer in un unico albero o su più alberi diversi, dipende dal parametro d , chiamato coefficiente di diversità



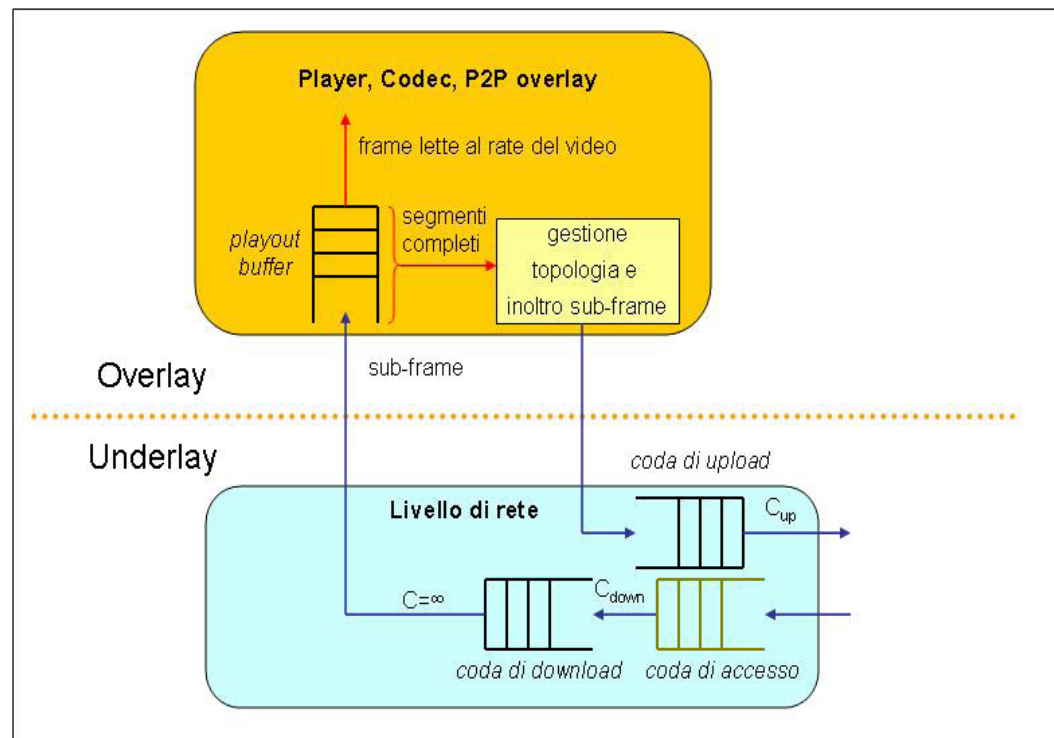
Diversità

- Nel primo caso, corrispondente a $d = 1$, vi è un'associazione uno a uno tra alberi e descrizioni, mentre vi è sempre un'associazione uno a uno tra alberi e sub-stream
- Nel secondo caso, quando $d > 1$, una singola descrizione è distribuita tramite più alberi
- Il numero di descrizioni è sempre pari al numero di sub-frame per frame, ma il numero di sub-stream e quindi di alberi è un multiplo del numero di descrizioni
- Per cui, ricordando che m è il numero di frames in cui è suddiviso il contenuto, k il numero di sub-frame per segmento e n il numero di sub-frame per frame, si ha che il numero di descrizioni è pari a n , il numero di alberi è $t=dn$, k frame creano n segmenti, per cui il numero totale di segmenti è



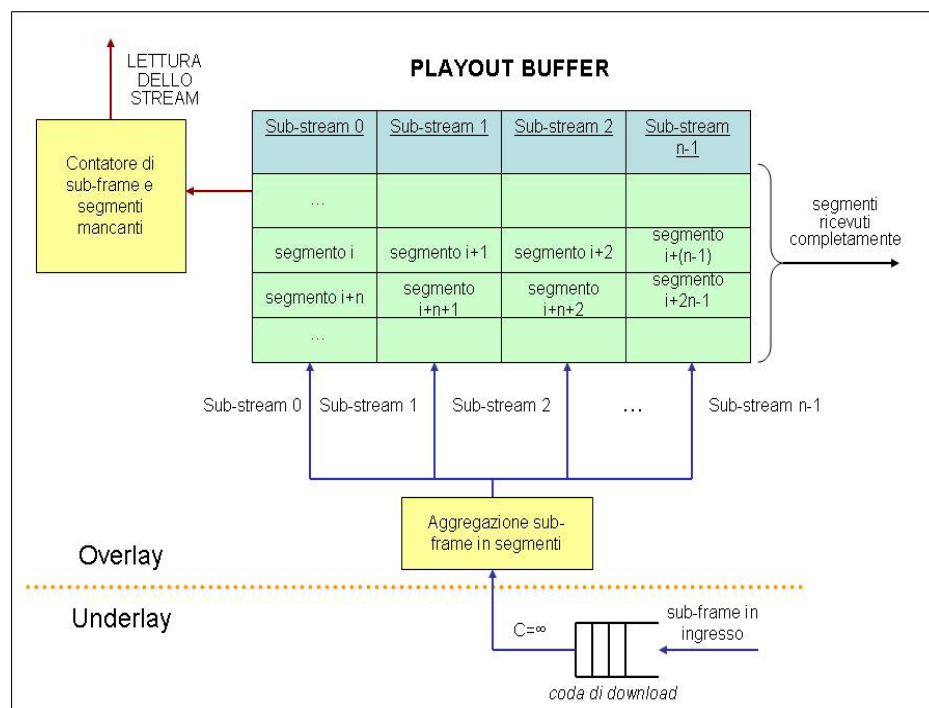
Il client

- Le sub-frame provenienti dai parent sono ricevute nella coda di accesso
- Le sub-frame sono consegnate alla coda di download con velocità CDOWN
- Le sub-frame della coda di download sono inviate al livello superiore, il playout-buffer, che le ordina e le riaggrega in segmenti.
- I segmenti completi sono inviati ai figli non appena sono stati ricevuti
- Il segmento è inoltrato inviando singolarmente e in sequenza le sub-frame che lo compongono, le sub-frame sono duplicate per ogni peer figlio a cui devono essere inviate
- La sub-frame del playout-buffer sono inviate al player aggregate in frame
- Le frame sono lette al rate



Playout buffer

- All'arrivo delle sub-frame esse sono associate al segmento, all'albero e al sub-stream corrispondenti
- Il peer esegue il forward di interi segmenti. Non appena il playout buffer riceve tutte le sub-frame appartenenti ad un segmento, esso inoltra il segmento a tutti i figli
- Le frame, quando devono essere riprodotte dal player, sono estratte dal playout buffer al ritmo previsto
- Il player inizia a leggere le frame quando il buffer è pieno oltre una certa soglia che è considerata superata quando almeno una descrizione raggiunge il numero minimo di segmenti completi
- Una soglia più alta diminuisce la probabilità di freezing points ma aumenta il ritardo di playout
- E' definito il tempo di buffering, T_b , come l'intervallo di tempo che intercorre tra la ricezione della prima sub-frame del video e l'inizio della riproduzione del video stesso,

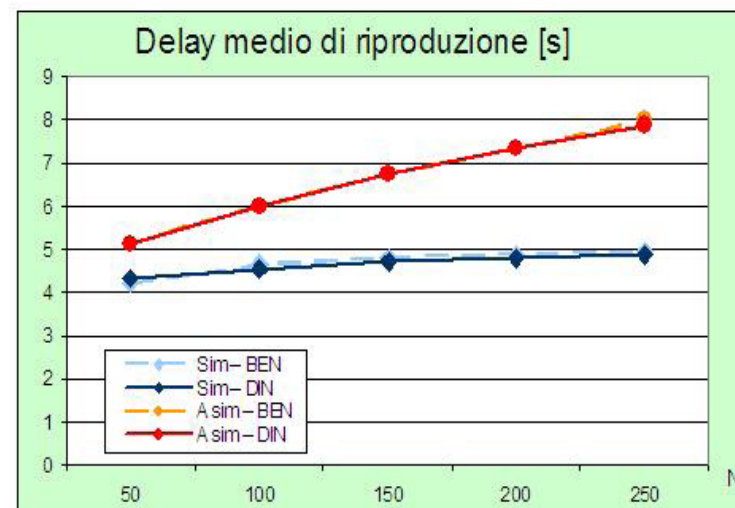


Parametri per valutazione prestazioni vidtorrent

- Simmetrico: $C_{down}=7\text{Mbit/s}$, $C_{up}=7\text{Mbit/s}$
- Asimmetrico: 7 / 1
- Traccia da 875 kbit/s formata da 4 descrizioni
- 4 alberi, diversità 1
- 1 segmento ha 20 frame e dura 0.8 s
- $N=250$ utenti in regime stazionario
- Permanenza 15 minuti
- $T_{rejoin}=100\text{s}$
- Soglia di riproduzione = 5 segmenti (4 s)

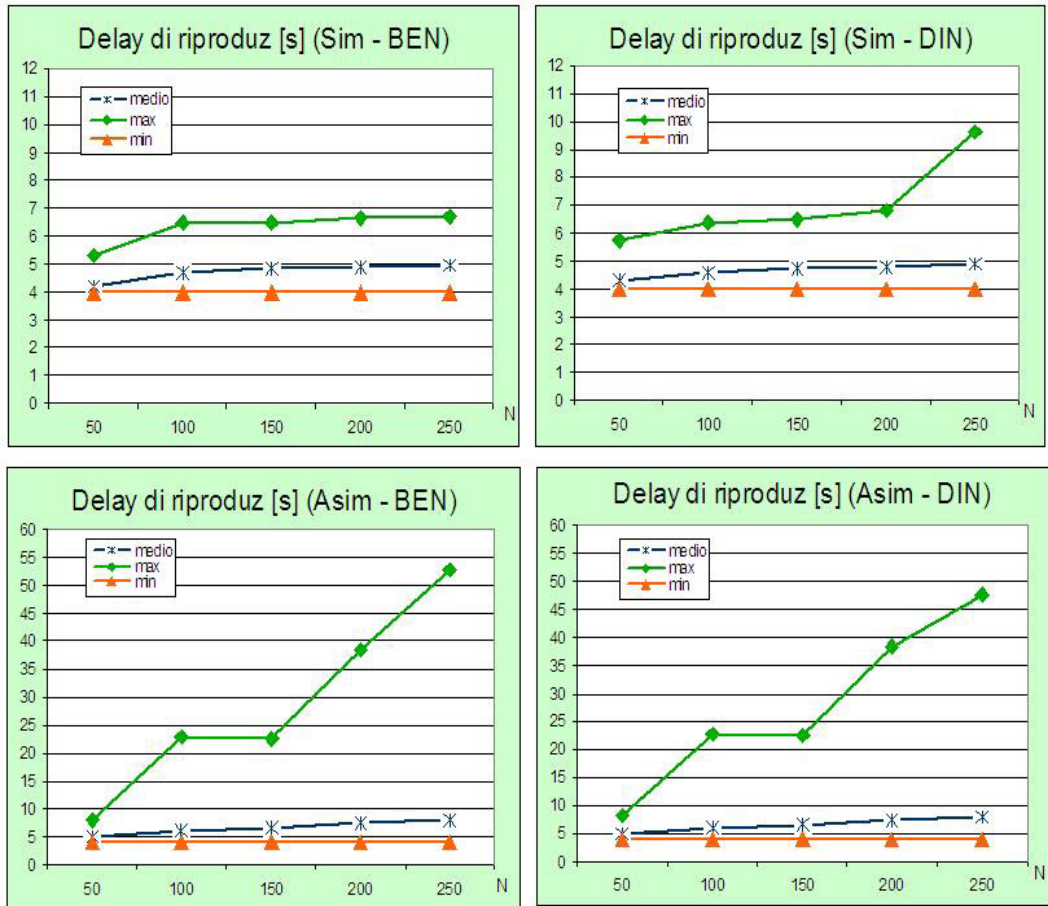
Prestazioni

- Tutti i grafici di prestazione qui riportati sono stati ottenuti con il simulatore di sistemi p2p video streaming sviluppato nel Broadband & Multimedia Internet & Security LAB (BBMIS)
- Le curve rilevanti in questo studio hanno label “Sim-DIN” (caso simmetrico) e “Asim-DIN” (caso asimmetrico)

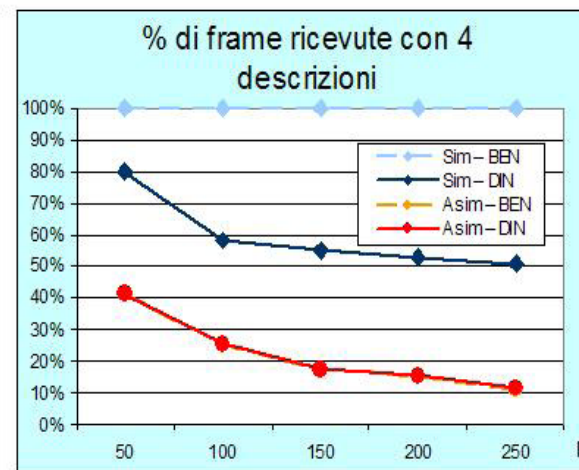
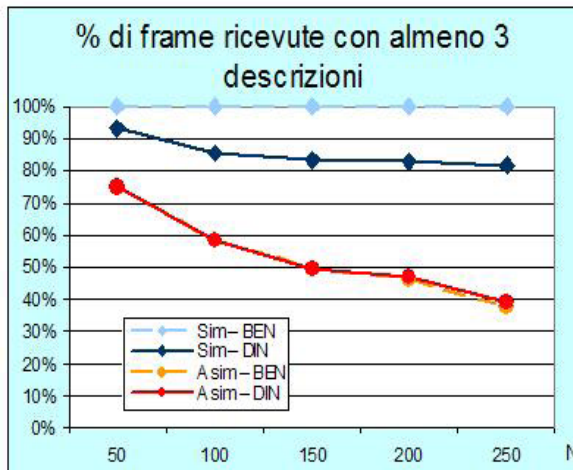
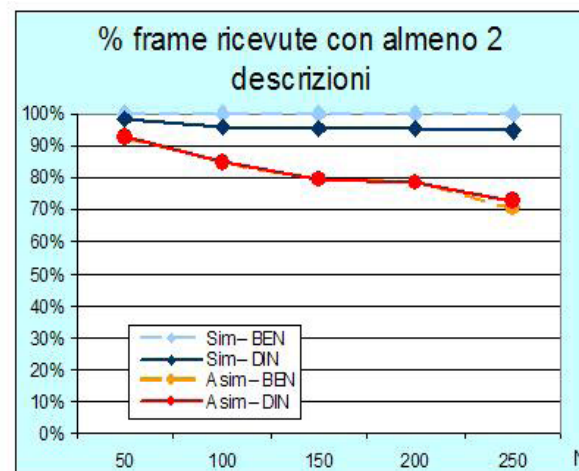
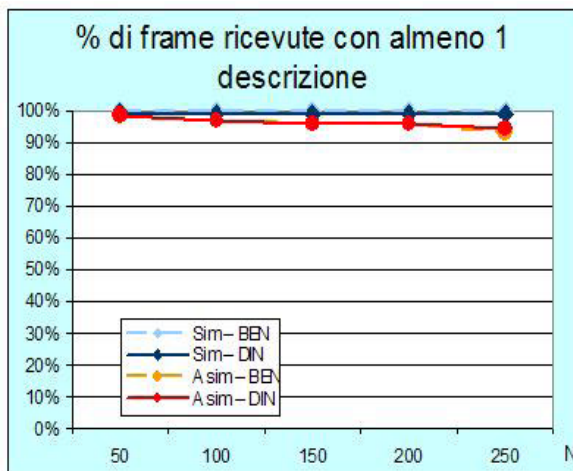


© BBMIS LAB 2008

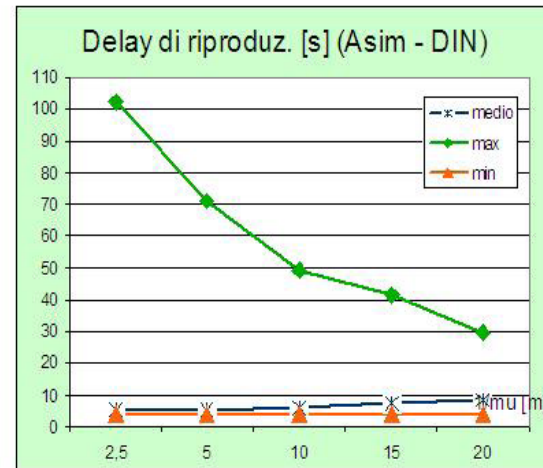
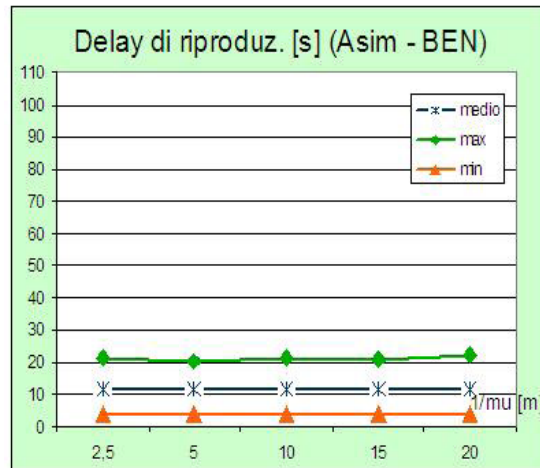
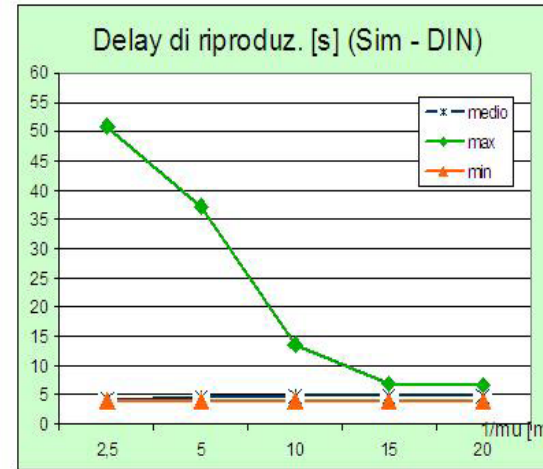
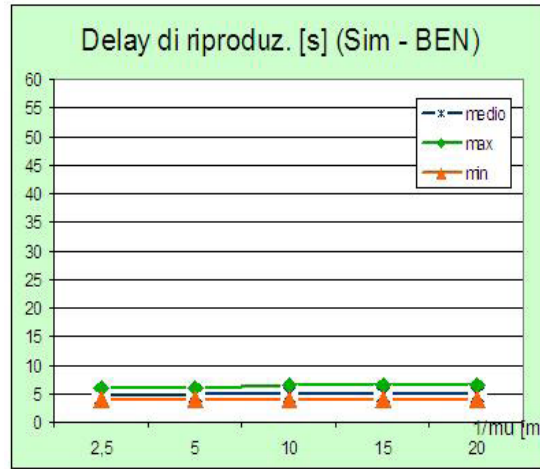
Ritardo di riproduzione in funzione del numero di peer



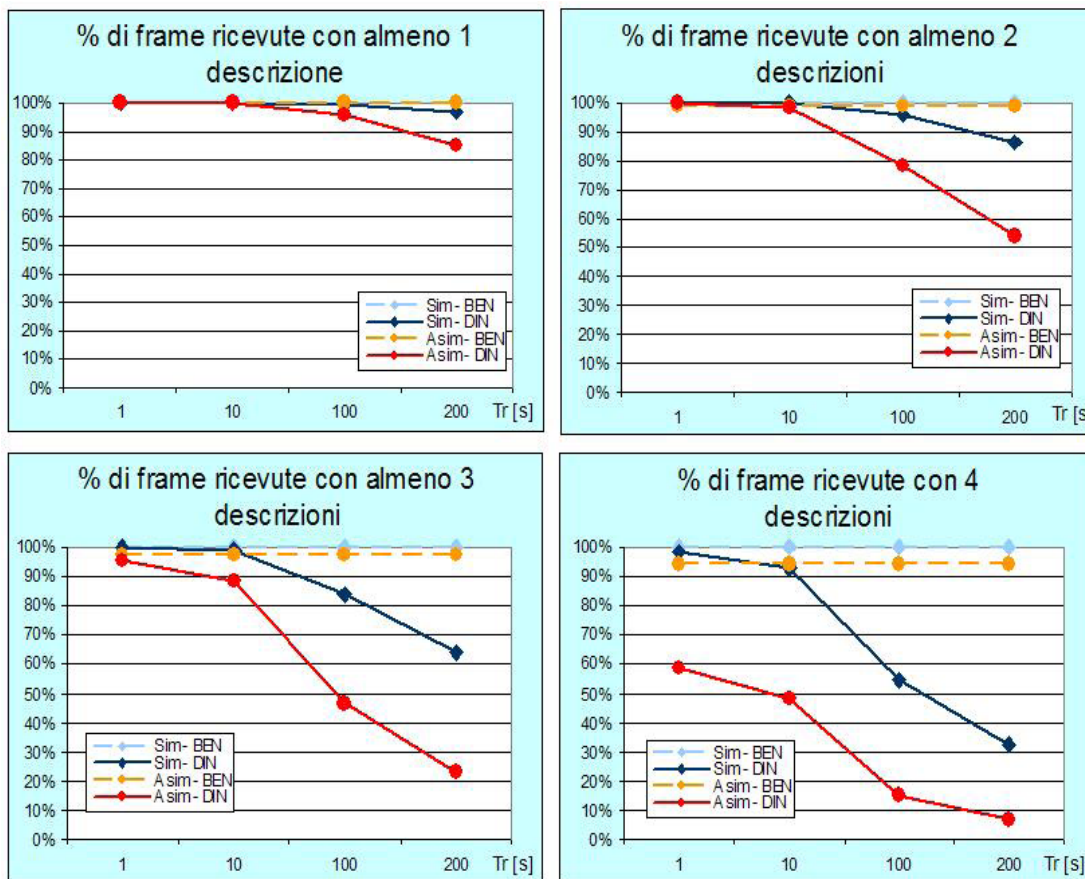
% frame ricevute in tempo



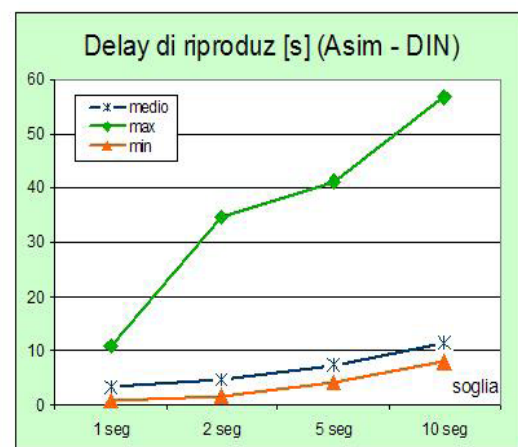
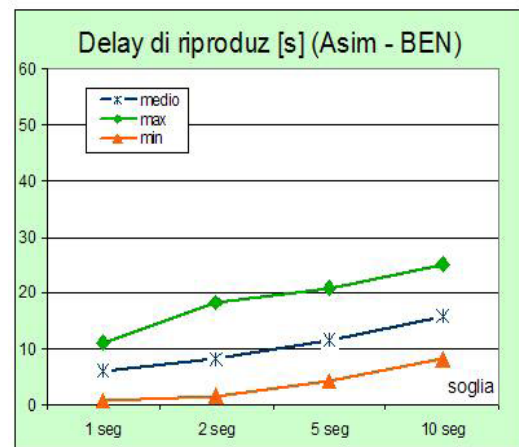
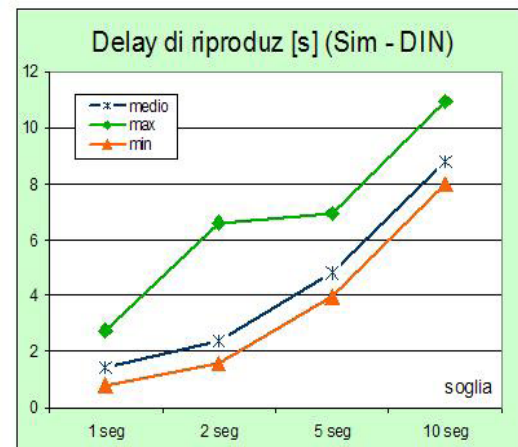
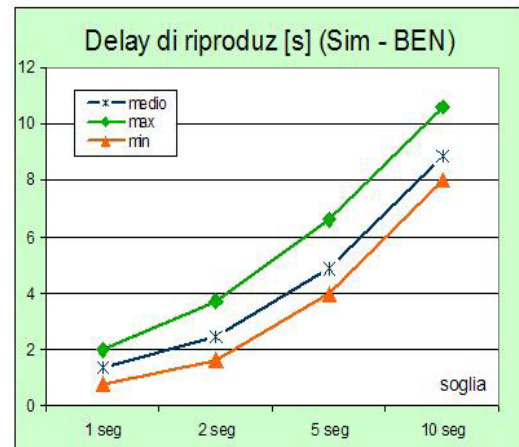
Ritardo di riproduzione in funzione del tempo medio di permanenza dei peer nel sistema



% frame ricevute in tempo in funzione del tempo di rejoin



Ritardo di riproduzione in funzione della soglia di riproduzione



% frame ricevute in tempo in funzione della soglia di riproduzione

