Peer-to-Peer video streaming

Main features of p2p video streaming systems

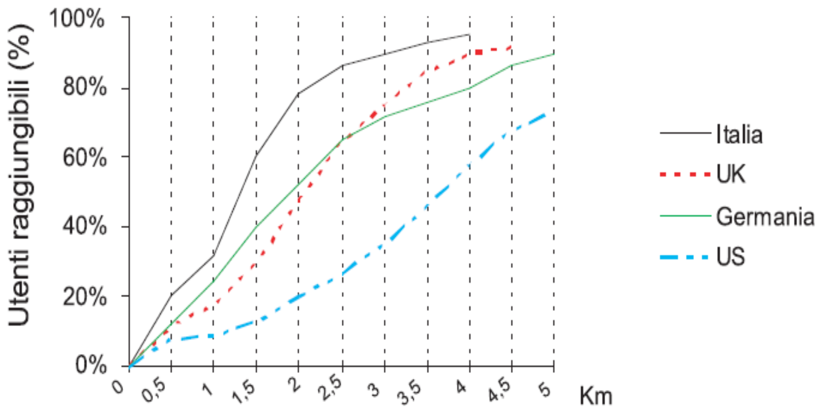
- 1. distribution of media
- 2. building overlay topology
- 3. types of overlay topology
- 4. sources
- 5. coordination and startup
- 6. join
- 7. leave
- 8. performance
- 9. cryptography

Technological requirements of p2p video streaming

- Residential large-bandwidth Internet access: ADSL and subsequent technology enable the distribution of p2p video streaming services
- Powerful user’s personal computers: the computational power of average-performance personal computer is enough to allow users Potenza dei personal computer di fascia medio-bassa, che permette di realizzare le funzionalità di nodo della rete peer-to-peer

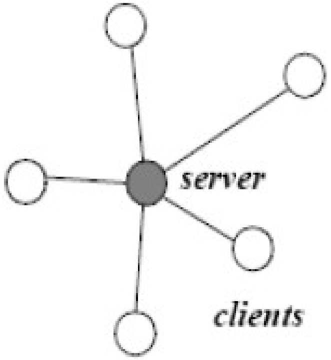
ADSL technologies

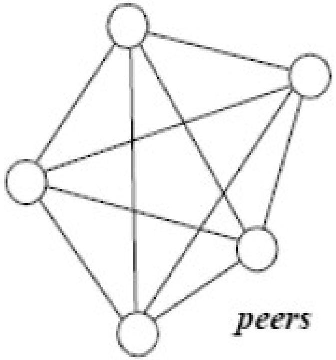
- ADSL technologies use the traditional copper twisted pair to provide IP communications to users
- The transmission speed depends, among other factors, on the quality of the twisted pair, on the interference power, and on the cable's length
- Traditional twisted pairs can provide a satisfactory transmission speed if they are not excessively long; a cable length of about 3km or less can be adequate
- In Italy the average length of twisted pairs is shorter than in other European countries; this is an advantage for ADSL technologies



Peer-to-Peer (P2P) systems

- P2P systems do not adopt the classical client-server architecture of many distributed applications
- In P2P systems, each participating node, referred to as *peer*, can acto both as server and client





Peer-to-peer: pros e cons

- The main advantage of p2p systems is that they allow users to reach easily a large number of contents, that usually are stored in the computers of peers
- P2p systems can scale and reach a very large size: some typical scalability limits of client-server architectures do not hold for p2p systems
- Some disadvantages:
- P2P systems are easily attacked and, as a consequence, their performance can be degraded heavily
- A significant performance issue is the upload bandwidth of the peer's connection
- The issue of the rights of distributed contents is serious: legal problems can arise if peers distribute material protected by copyright
-

Distribution of media

- end-to-end overlay,
  - each peer exchanges contents and also contributes to the management of the overlay topology. All nodes play the same role in the p2p system.
- proxy-based overlay,
  - The p2p system implements a hierarchy, in such a way that only some selected nodes route the contents to the final peer (they are called also *multicast nodes*); in this way, multicast nodes form an overlay backbone network, and the overlay network is divided into two layers. Peers receive contents from the backbone and send contents to the backbone, but they do not exchange directly contents.

Distribution of media

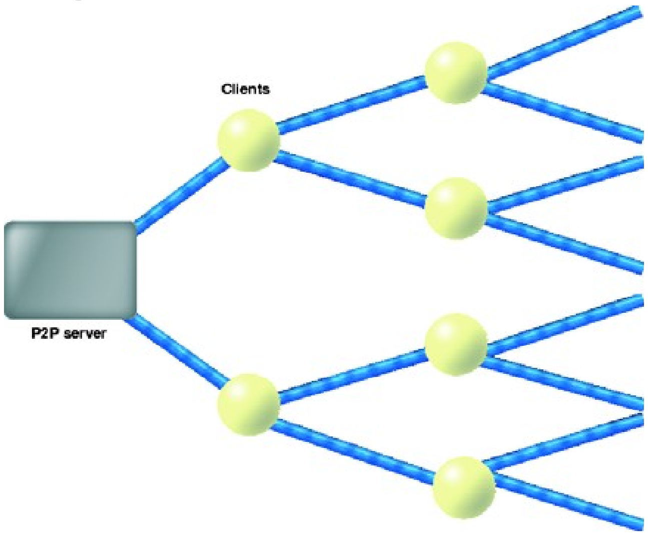
- The proxy-based simplifies the management of the overlay topology (only the overlay backbone has a mesh topology, while from each backbone network a tree (star) overlay network reaches peers.
- Multicast nodes can be peers, which have been promoted by the system; this can happen for nodes with adequate processing power and communications bandwidth
- In end-to-end systems peers need to exchange much information to maintaing the consistency of the overlay network's topology
- End-to-end overlay and proxy-based overlay systems are also referred to as application level multicast (ALM) and overlay multicast, respectively

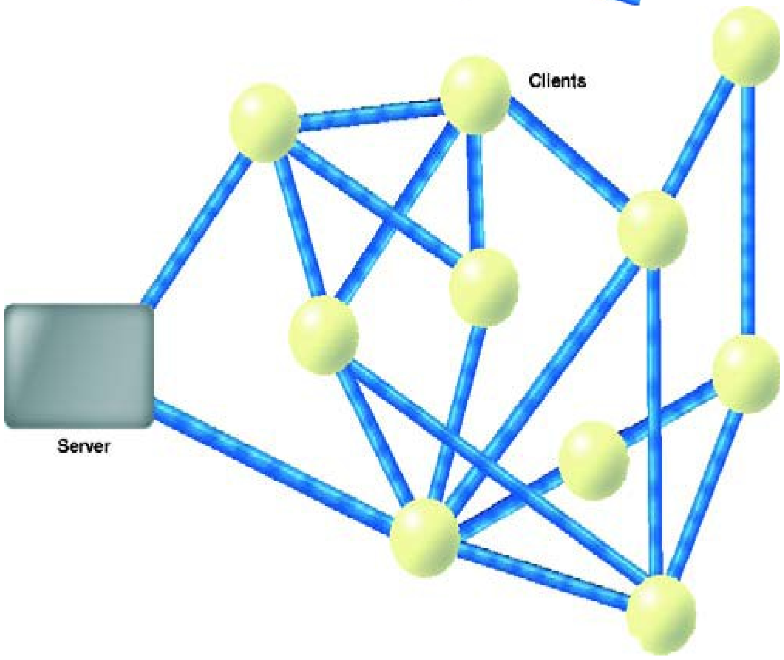
Construction of the overlay topology

- direct:
  - When a peer enters the system, it is immediately assigned a parent peer to which it must connect. This is the typical strategy implemented in tree and forest overlay topologies.
- Indirect, mesh:
  - When a peer enters the systems, a signaling phase starts, by which it is connected to one or more peers, in a mesh fashion. The peer can communicate with multiple peers.
- Indirect, hierarchical clusters:
  - Peers are divided into hierarchical levels, and they are also grouped into clusters. Only a subset of the peers in one cluster is allowed to coommunicate directly with peers in another cluster. Therefore, an inter-cluster overlay topology and an intra-cluster overlay topology are created. In some practical cases, the inter-cluster topology is a tree.

Topologies

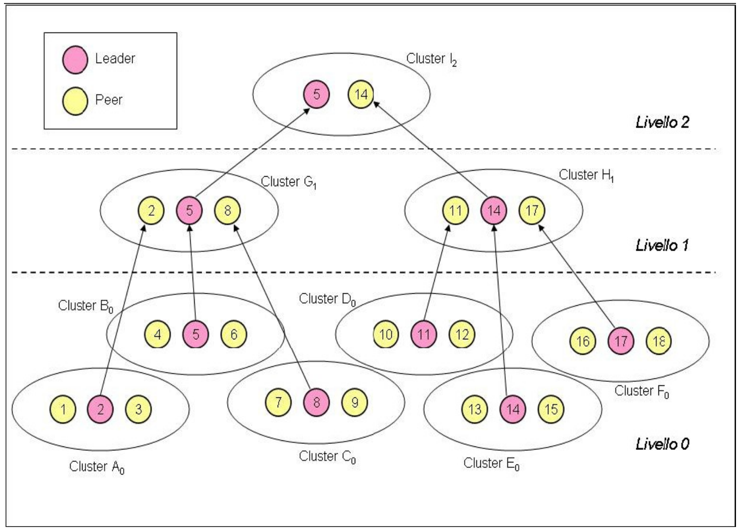
- *Single tree*
- *Forest*: multiple trees, with the same root node
- *Generic (mesh)*

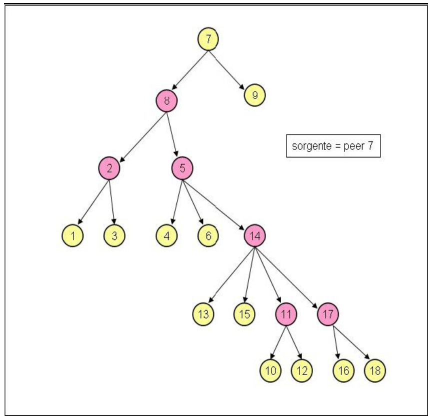




Tree topology with indirect construction, cluster-based

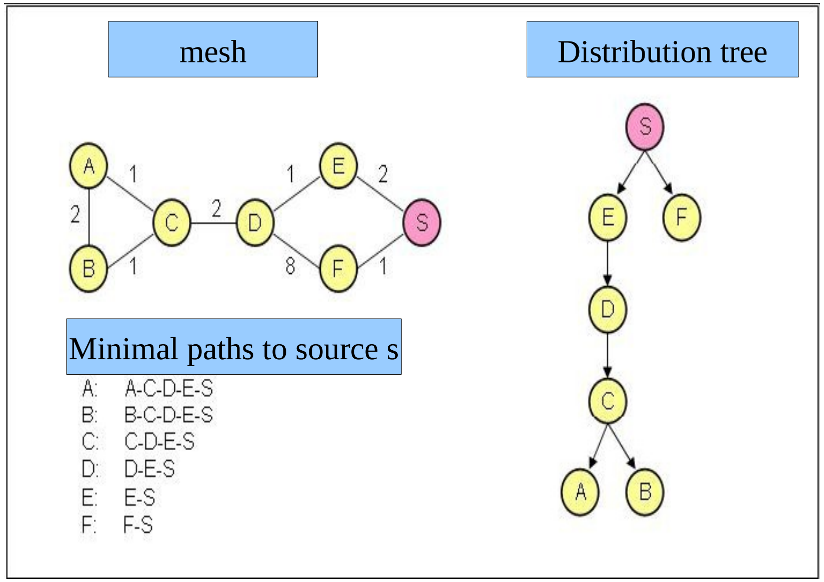
- The root node sends the content to all its clustermates, that is, to all the nodes in its clusters
- A peer receiving the content distributes it to all its clustermates, if it did not receive the content from a clustermate
- Thus, in each cluster only one peer distributes the content to the clustermate nodes
- In the figure a hierarchical cluster topology with 3 levels (level 0, 1, and 2) is shown
- The source of the content is node 7, and it is shown how a distribution tree for the content is built





Tree-based distribution, with indirect mesh-based construction of the topology

- In this kind of systems, a basic mesh overlay topology is firstly constructed
- On this topology, paths to reach the source of a content can be calculated, for example with a classic distance-vector routing algorithm
- A peer keeps a table of the minimal path to reach all the other peers, along with a conventional measure of the cost of each path (for example, path length)
- Tables are continually updated by inter-peer signalling communications
- For the distribution of a content, the source calculates a multicast tree, built over the basic mesh topology
- The elementary metric to measure the path's cost is the number of hops
- More advance metrics as delay can be implemented



Forest

- A forest topology is contructed as a number (>=1) of trees, which share the same source
- This topology is more reliably than the simple tree: if a peer leaves, a single tree would disconnect, while in a forest topology the overall distribution can be available (at least partially) even if a peer leaves (in a forest topology paths are redundant)
- This property of forest topologies is called *tree diversity*
- Trees with small depth and large fanout should be preferred, because delay is smaller than with deep treees with small fanout
- However, a large fanout of nodes can be obtained only if the upload capacity of peers is large

Forest

- Forest topologies can be used successfully to distribute multimedia streaming in a p2p fashion
  - The stream can be divided into sub-streams (this can be done if the stream is coded with a multi-layered coder or a multiple description coder)
  - Each substram is assigned to a different tree
  - In this way, even if one of the tree disconnected due to the leave of one peer, peers would still receive the other substreams

mesh

- A peer must perform two basic activities
  - Signaling with other peers to establish and maintain the mesh
  - Exchange contents
- In the case of p2p video streaming systems, we assume that a stream is divided into segments (chunks)
- Peers exchange segments
- A typical maximum size of a segment is about 10 KByte
- Each peer maintains a list of the segments that it has received (the *Buffer Map*, BM)
- Through signalling with it neighbor peers (in the overlay mesh) each peer maintains also the *mCache*, a table of the segments that it neighbor peers already have

mesh

- The Buffer Map is a list of the segments that a peer has; this list is time-limited (it spans over a limited time interval)
- The Buffer Map contains the following parameters:
  - offset, the index of the first segment (typically 2-4 bytes);
  - length of the buffer map;
  - A vector of binary flags (1/0) to indicate the availability of each segment in the limited time interval
- For example, if a segments contains 1 s of content, the BM could keep information about 120 segments (time interval would be two minutes)
- For each second in the time interval, a binary flag indicates if the corresponding segment is available

mesh

- Through its BM, a peer can determine which segments it needs
- In order to determine which peers to contact to get the missing segments, a peer must also have the Bms of a number of peers (partner peers), usually, its neighbor peers in the overlay mesh topology
- When this information is available, the peer runs a scheduling algorithm, to schedule requests to other peers
- The Bms of partner nodes are obtained through signalling

mesh

- The DONet/Coolstreaming system adopts an heuristic scheduling algorithm:
  - For each missing segment, the peer examines the Bms of its partner peers, and counts the number of possible suppliers
  - Requests are scheduled starting for the segment with the smallest number of suppliers, then for the segment with the second-smallest number of suppliers, and so on
  - Clearly, this scheduling algorithm improves reliability
  - If more than 1 segment must be required to the same supplier, it is done with a single request

mesh

- The construction of the mcache is critical
- Peers exchange refresh messages specifying:
  - seq num (the sequence number of the available segments)
  - id (identification code of the peer in the system)
  - N_partner (number of partners of the peer)
  - TTL (Time To Live: residual life time of the segment)

mesh

- When a peer receives a refresh message, it updated the its mCache
- Periodically, a peer seeks for other partners, if it decides that it has not enough partners
- A peer can also measure the “quality” of a partner, by observing how many segments it can succesfully exchange with it

mesh

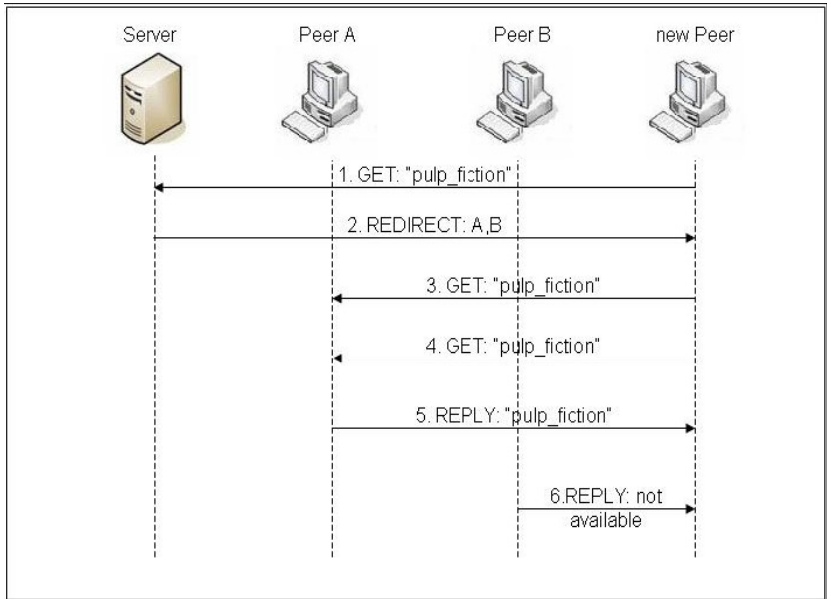
- The target number of partners of a peer is a critical system parameter.
- A large number of partners increases reliability
- However, a large number of partners increases significantly signaling traffic
- Therefore, an optimal value of M exists
- From practical trials, it has been concluded that a number of partners ranging from 2 to 5 is a reasonable choice
- In Coolstreaming, the overhead traffic adds up to about 1% of the total traffic
- Mesh topologies react to node leaves better than forest topologies

joins

- The join process is invoked when a peer joins the system
- In a tree topology with direct construction, the system must assign a parent to the peer
- In more complex topologies the system must assign to the joining peer a number of parents (forest) or a number of partner peers (mesh)
- In tree-based systems, the new peer will become a leave of the tree
- Subsequently, the peer may be promoted and get closer to the tree root

Joins in COOPNET

- In the case of direct construction of the overlay topology, the new peer must be connected to exactly one parent peer
- A server is dedicated to collect join requests and to redirect the requesting peer to potential parent peers
- Coopnet is a forest-based system:
  - The new peer contacts the server; the peer can also place a request for a specific content and provide information such as its internet bandwidth
  - The server sends to the requesting peer a list of peers that the new peer must contact



leave

- The leave process is invoked when a peer leaves the system
- In a tree topology, all the childs of a leaving peer are disconnected
- The system must provide a new parent for the orphan peers

leave

- Failure detection: the system needs to implement procedures to understand as soon as possible that a node has leaved
- There are two basic ways to do it:
  - keep-alive: nodes periodically exchange keep-alive messages
  - packet loss rate monitoring (CoopNet): child nodes monitor the percentage of lost packets, if it grows above a predefined threshold, the parent node is declared dead

Some p2p video streaming systems

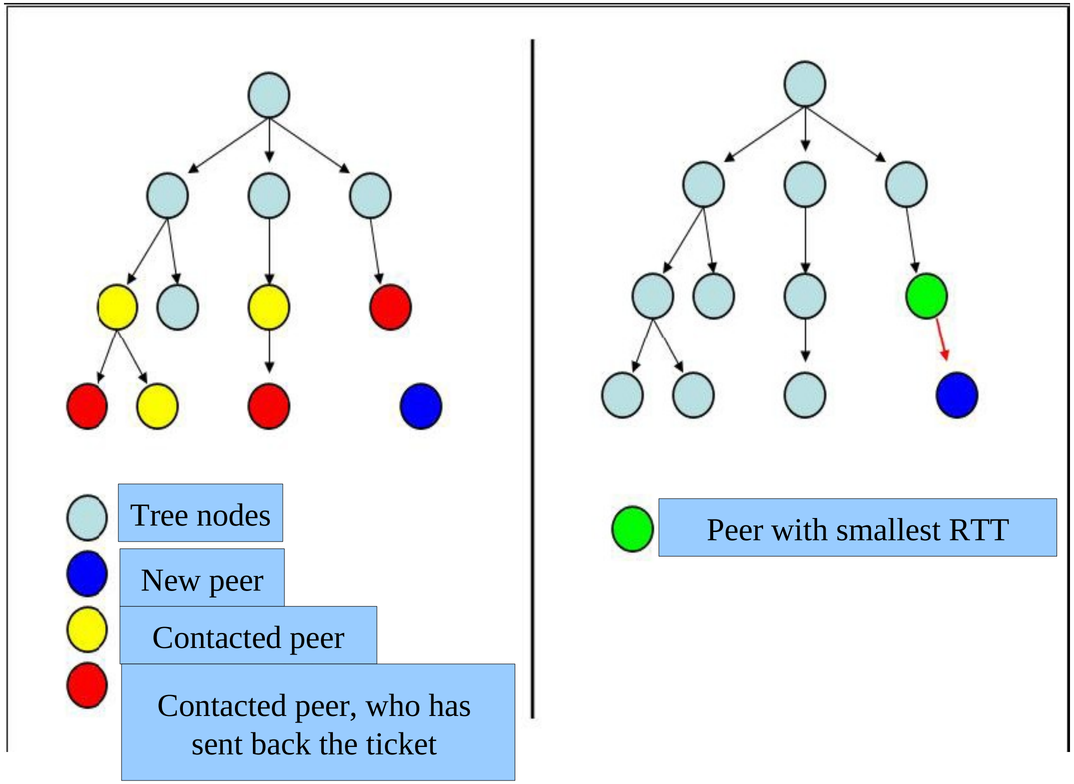
name	distribution	Topology construction	Topology	sources	7.2 failure detection
vidTorrent	end-to-end	Indirect mesh	forest	one	keep-alive
CoolStreaming	end-to-end	Indirect mesh	mesh	one	keep-alive
CoopNet	server centrale	direct	forest	one	packet loss monitoring
GridMedia	end-to-end	Indirect mesh	mesh	one	keep-alive
Narada	end-to-end	Indirect mesh	tree	one	keep-alive
Nice	proxy-based	Hierarc. clust.	tree	one	keep-alive
Overcast	proxy-based	direct	tree	one	keep-alive
PPLive	end-to-end	Indirect mesh	mesh	one	
Scattercast	proxy-based	Indirect mesh	tree	>= 1	keep-alive
SplitStream	end-to-end	Indirect mesh	forest	one	keep-alive
Zig-zag	proxy-based	Hierarc. clust.	tree	one	keep-alive
PeerCast	end-to-end	direct	tree	one	keep-alive

VidTorrent

- VidTorrent implements a forest topology where each tree transports a different sub-stream
- A peer, in order to receive the full stream, must be inserted in all the subtrees
- Trees are constructed in such a way that peers with larger transmission capacity are closer to the root of the tree
- The parent-child association is done by extimating the communication bandwidth and round trip delay between the two nodes

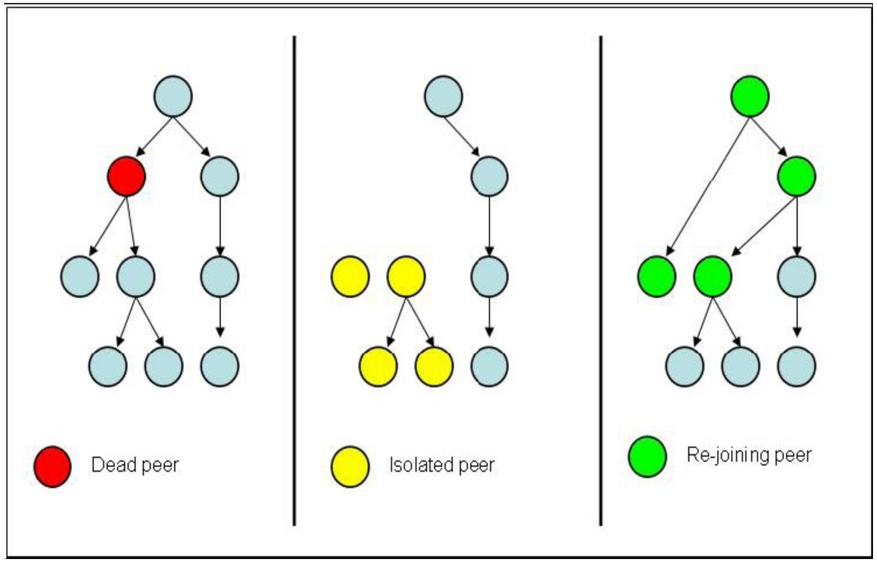
Join in VidTorrent

- The join procedure must be performed for each sub-tree
- The new peer musty contact a know randez-vous point in the system
- The randez-vous server replies with a list of peers to be contacted
- The new peer contacts these peers; in the responses, additional peers may be present
- The new peer sends to these peers two packets, to get a rough estimation of bandwidth and delay
- If the result of the measurement is acceptable, the contacted peer sends to the new peer an acceptation ticket
- The new peer, among all tickets, chooses the one with the smallest delay



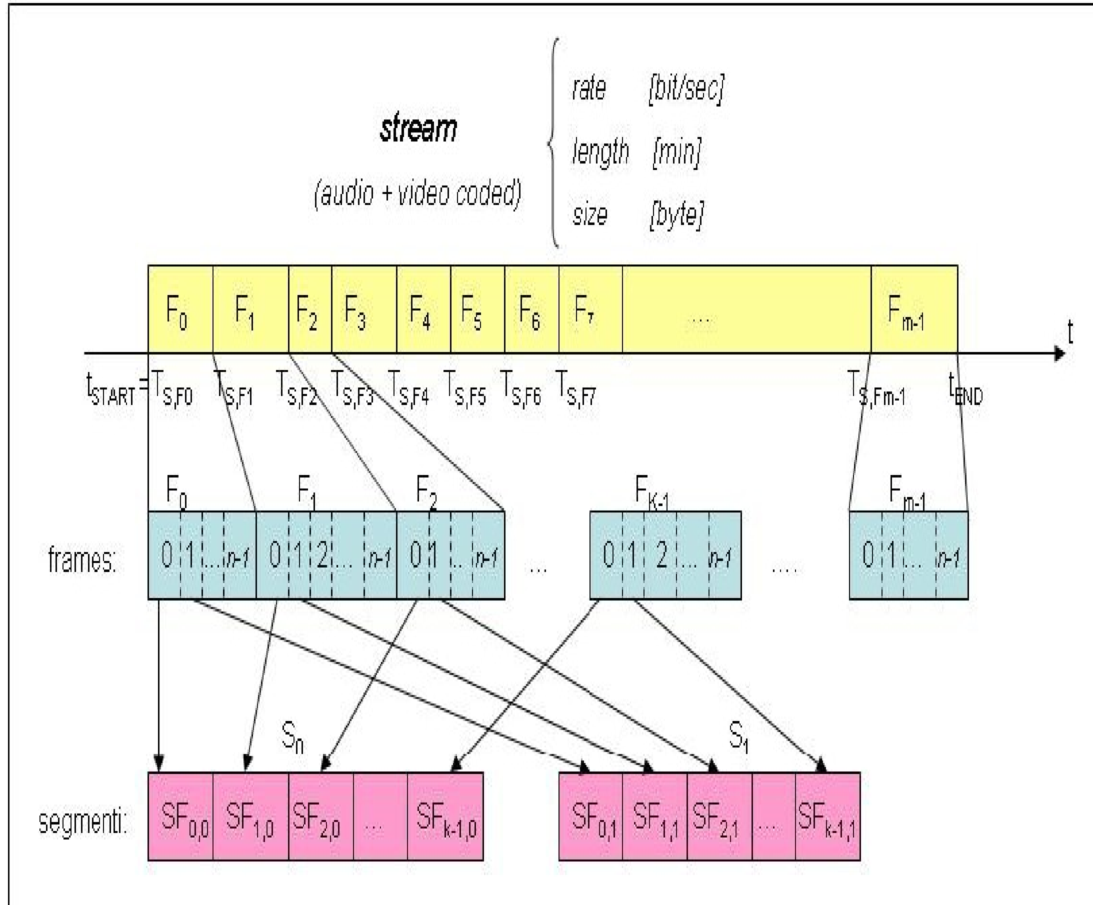
Leave in VidTorrent

- When a tree is disconnected, isolated peer contact the randez-vous point to get a new parent
- The procedure is very similar to the join procedure of a new node



Dividing a stream in substreams

- VidTorrent can divide a stream in substreams
- This division is hierarchical: frames, subframes and segments
- This stream structure can accommodate scalable encodings of media streams



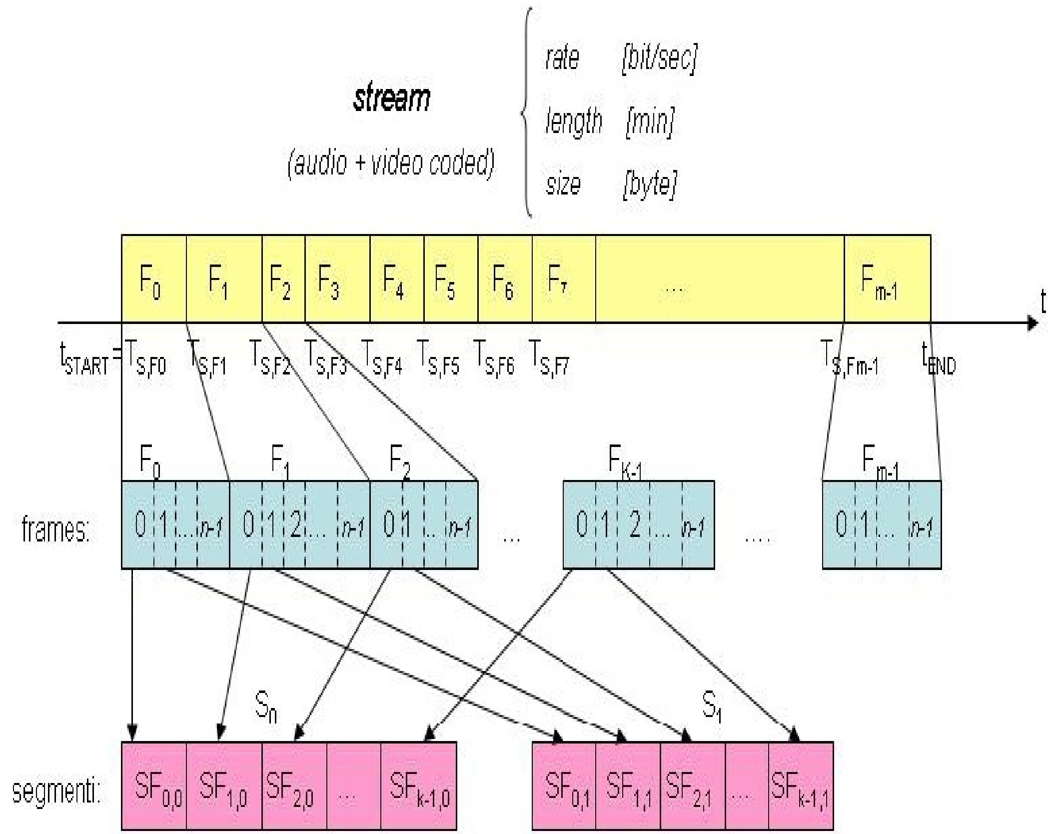
k: Number of frames per segment

n: Number of subframes per frame

h: Number of segments

Dividing a stream in substreams

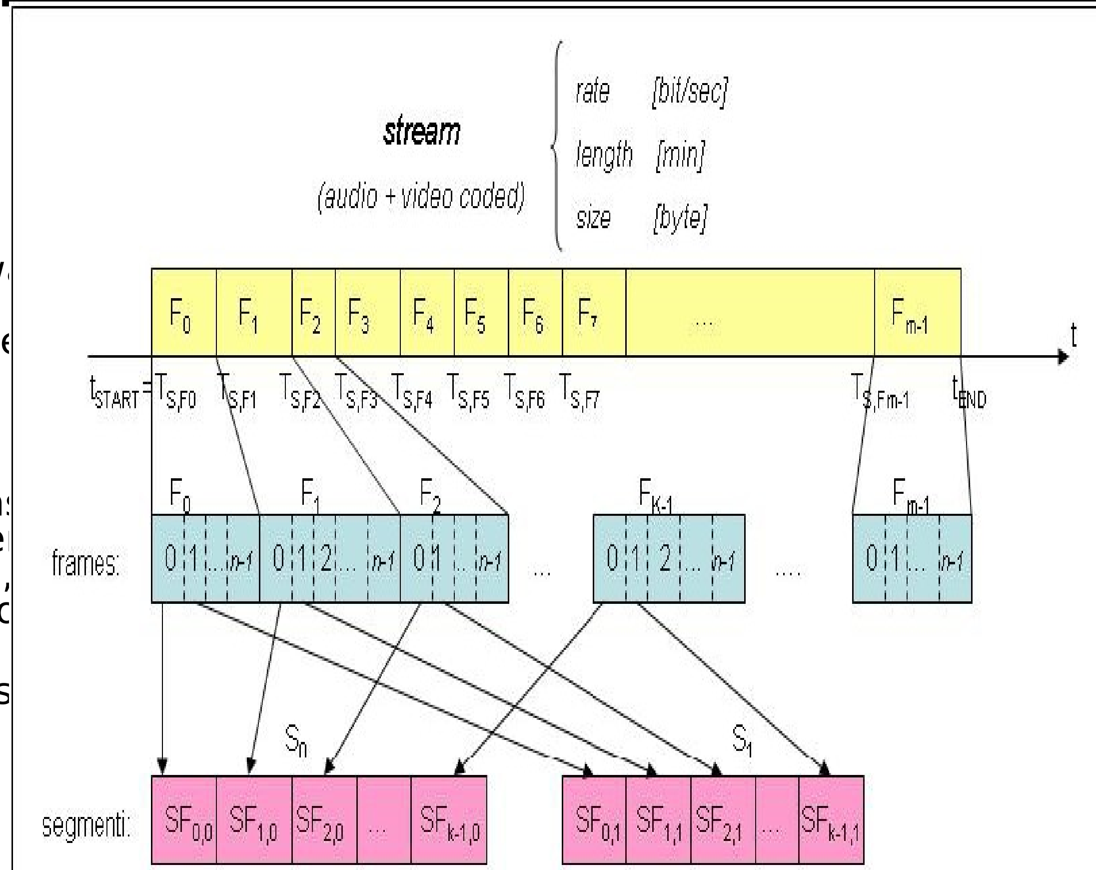
- For a Multiple Description coding, the ith Frame Fi è is divided in subframes (SFi,j)
- In order to replay the content, a peer must receive at least one subframe per frame; receiving more subframes per frame improves quality
- The source interleaves segments, in such a way that each segment contains only one subframe per frame
- Thus, the subframes of a frame are distributed over a number of consecutive segments
- Each segment carries data from k frames



k: Number of frames per segment
n: Number of subframes per frame
h: Number of segments

Suddivisione dello stream

- The stream is divided into m frames, each frame is divided into n subframes
- Segments are built in such a w that segment S0 contains all subframes 0 of the first k frame
- Segment S1 contains all subframes 1 of the first k frames, and so on
- If k = 3, each segment contain 3 subframes, and if the numbe of subframes per frame is n=4, then 4 segments are needed tc transmit the first k frames
- The fifth segment, S4, contains all the subframes 0 of the frames from 3 to 5, and so on



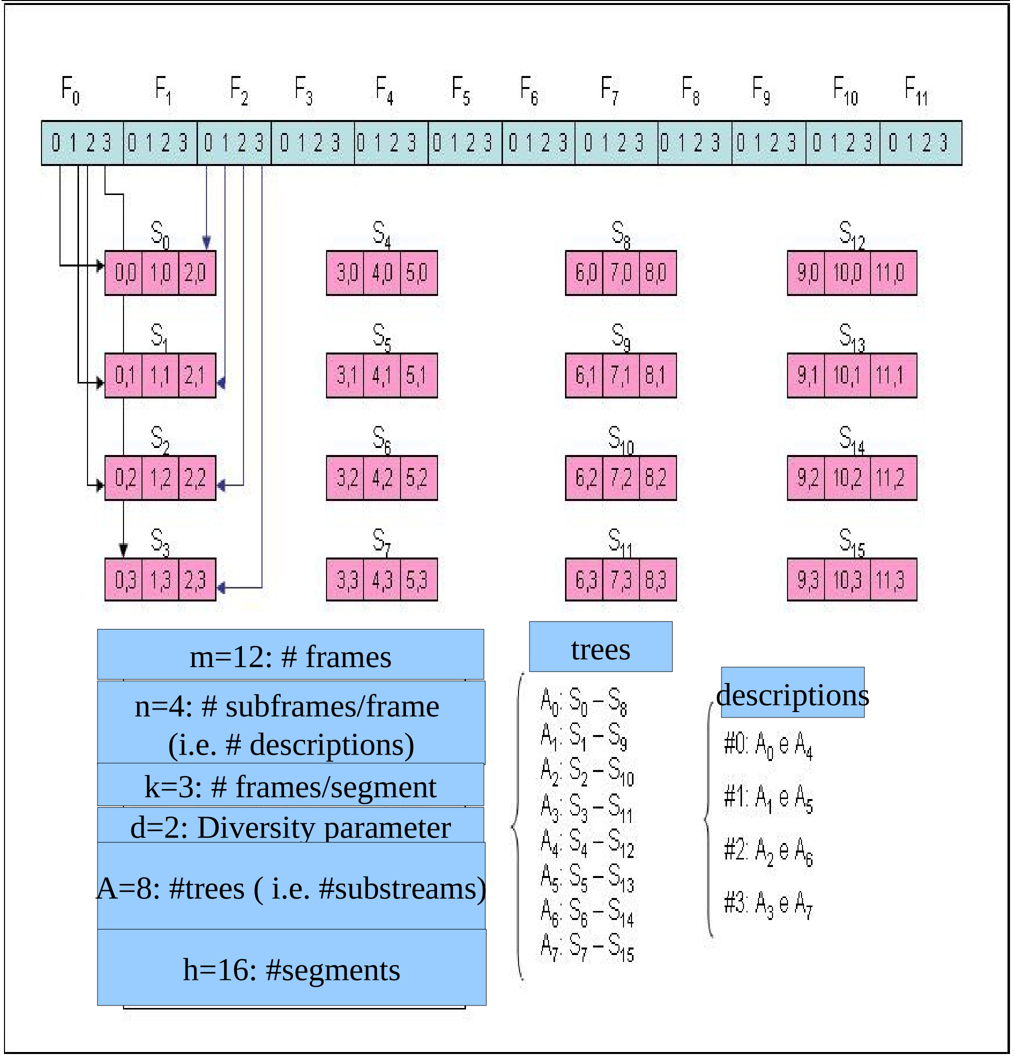
k: Number of frames per segment
n: Number of subframes per frame
h: Number of segments

substreams

- Segments are assigned cyclically to the substreams
- A substream is a sequence of segments transmitted on a given tree
- Each tree transports a different substream

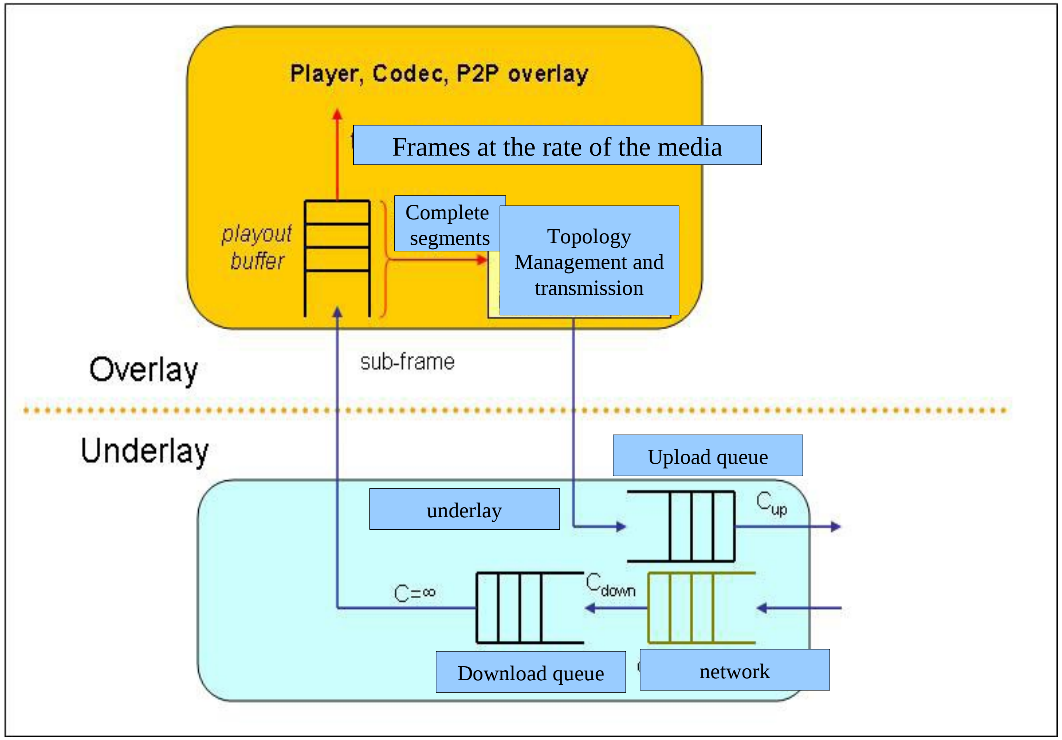
Diversity

- A substream can coincide with a description, or it can be part of a description
- A description can be distributed to peers through a single tree, or by multiple trees (d trees)
- In the first case (d=1) there is a one-to-one mapping between trees and description
- If d>1, a description is carried by multiple (d) trees
- Segments are assigned cyclically to substreams (trees)



The client

- Subframes are received at the downstream speed CDOWN
- Subframes are sent to the overlay component of the client and they are reassembled into segments
- Complete segments are sent to other peers
- Naturally, if the client must distribute the content to M partners, then it must send M copies of the content
-

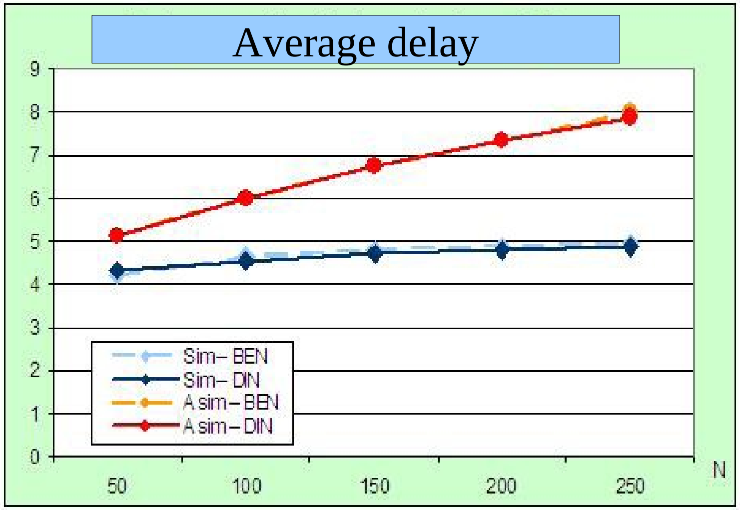


Performance of VidTorrent

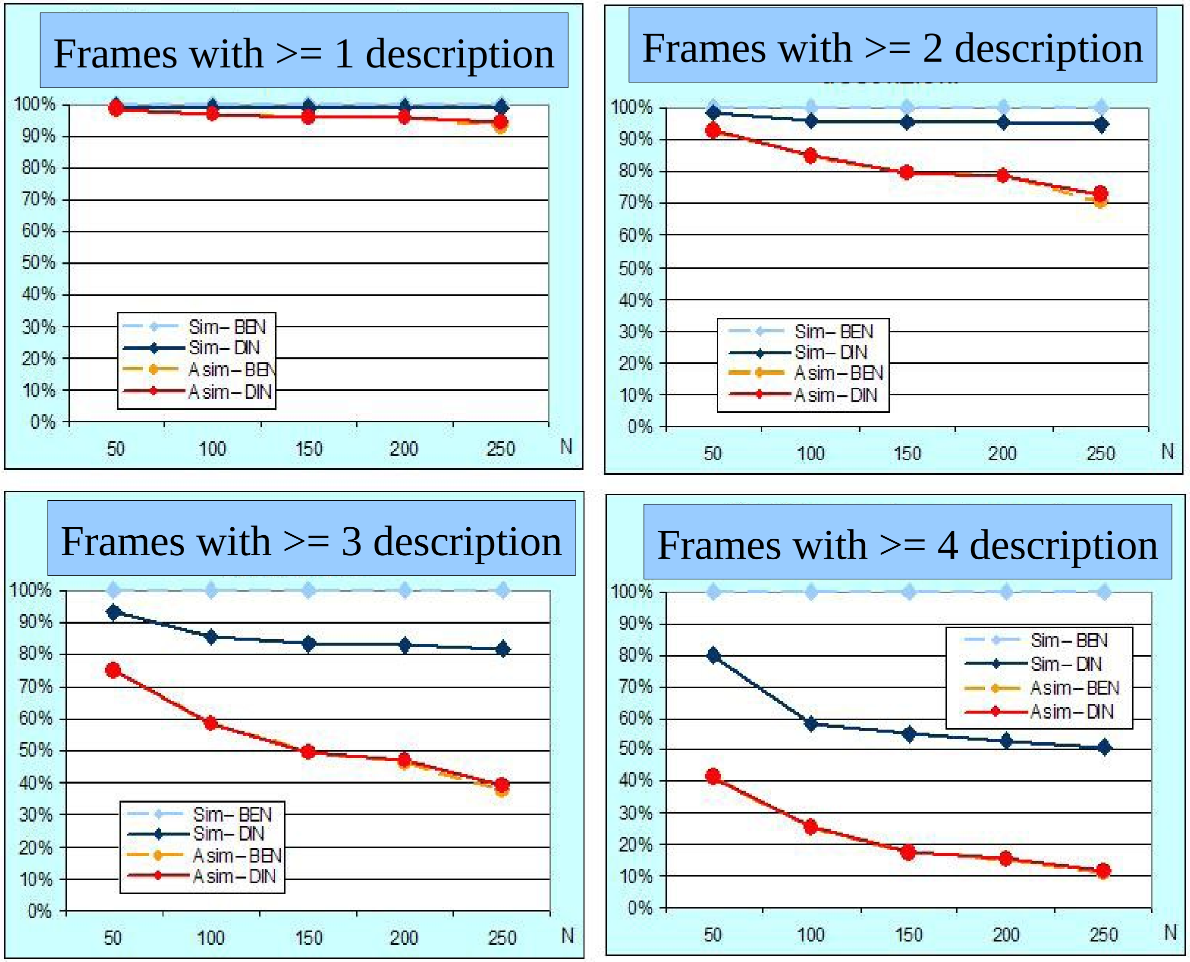
- Symmetric access: Cdown=7Mbit/s, Cup=7 Mbit/s
- Asymmetric access: 7/1
- 875 kbit/s video trace, 4 descriptions
- 4 trees, diversity 1
- 1 segment has 20 frames and it carries 0.8 s
- N=250 users (average)
- Average stay time of users: 15 minutes
- Trejoin=100s
- Depth of playout buffer = 5 segments (4 s)

Performance

- Relevant curves have label “Sim-DIN” (symmetric) and “Asim-DIN” asymmetric)

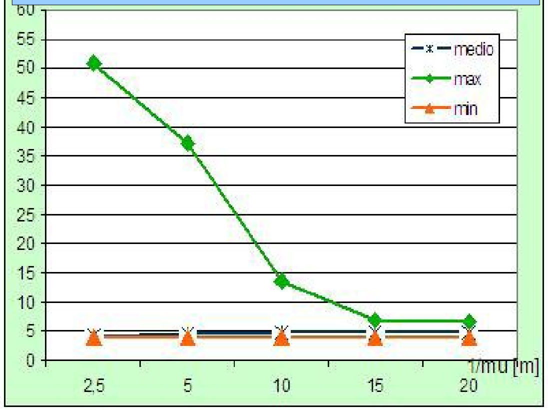


% frames received in time

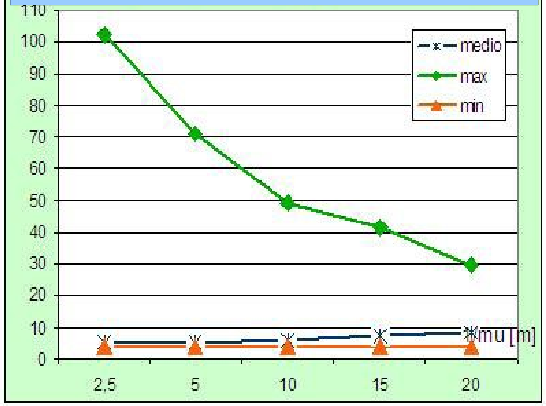


Delay as a function of the average peer stay time

Delay (SIM_DIN)



Delay (ASYM_DIN)



% frames received in time as a function of the rejoin time