NAPA WINE

A Bandwidth-Aware Scheduling Strategy for P2P-TV Systems

Ana Couto, **Emilio Leonardi**,
Marco Mellia, Michela Meo

Politecnico di Torino

P2P 2008 – Aachen
11 september 2008

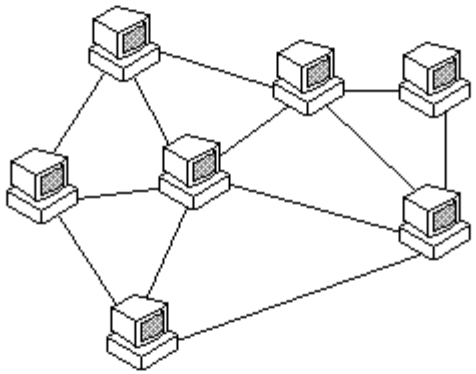
TELECOMMUNICATION NETWORKS GROUP

Outline

- System description and Objectives
- *Bandwidth-aware* schemes:
  - *Bandwidth-aware* scheduling strategy for chunk delivery
  - *Bandwidth-aware* setting of neighborhood size
- Conclusions and future work

The system

- Push mesh-based P2P-TV system:
  - Peers are arranged according to a general mesh overlay topology
  - Video content is cut in slices called chunks
  - Each chunk is independently distributed over the topology



Scheduling algorithm at nodes

- Chunks are distributed through the network using a swarm like (epidemic) approach
  - As soon as, a peer obtains a new chunk c, it will offer c to its neighbors
  - Chunks are not propagated perfectly in order; however chunk timing is critical (due to the application requirements)
  - Each chunk has a deadline after which it is not useful (this deadline is related to the play-out buffer)

More in detail

- Peers have infinite download bandwidth and *finite* upload bandwidth
- Each peer manages:
  - a list of neighbors with whom chunks can be exchanged
  - a window of chunks that can be distributed to neighbors
- Each peer decides which chunk distribute to which neighbor based on partial knowledge (limited to its neighborhood)

Objectives

- Problem: Find strategies for chunk distribution and neighborhood selection that make the system efficient (low delays, low losses)
- In homogeneous scenarios:
  - Most deprived peer - random usefull chunk (throughput optimal - by Massoulie et al [Infocom 07])
  - Random peer - Latest usefull chunk (delay optimal, by Bonald et al. [Sigmetrics 2008])

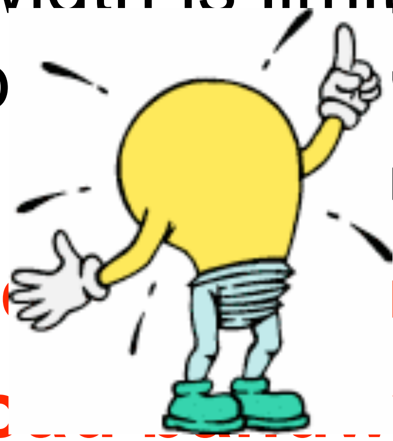
Objectives (cnt)

- How to schedule in more realistic heterogeneous scenarios where peers with different upload bandwidth coexist?
- Basic idea:

since peer bandwidth is limited in the upload,

upload cap ... mines system

... ce



take into ac ... erogeneity;

exploit uplo ... idth at best

Bandwidth-aware strategies

Exploit upload bandwidth at best:

Make peers contribute to the content distribution based on their upload bw

- Peer selection when delivering a chunk:
  distribute the chunk to the neighbor with the largest upload bandwidth
- Neighborhood selection:

  assign a larger number of neighbors to the peers with larger upload bandwidth

System description

- Chunks have constant size *L*
- Peer *p* has upload bandwidth *u(p)*
- Each peer maintains a list of neighbors and knows the neighbors upload bandwidth
- Each peer has a small sliding window of *w* chunks that can be distributed to its neighbors
- Each peer knows the content of its neighbors' windows

Further assuptions

- Each peer selects *k* random neighbors “at random”
- Since links are bidirectional, the neighbor-list of a peer contains, on average, 2*k* entries
- Churning is neglected

BA scheduling strategy

- The peer has to locally decide
  - the chunk to distribute
  - the destination neighbor
- Choice of the chunk:
  - *Latest chunk first* is the most convenient (already proposed in the literature by Bonald et al.)
- Choice of the neighbor:
  - *Bandwidth-aware* is our proposal

BA scheduling strategy

- *Bandwidth-aware* strategy:

  Select the peer with a probability proportional to its upload bandwidth

- BA tends to first deliver the chunk to peers that can contribute more to chunk diffusion
  - Chunk *j* is sent to peer *p* only if *p* does not have *j*
  - If chunk *j* cannot be delivered, the preceding chunk in the window is chosen

Simulation setting

- Two simulation phases:
  1. Generation of a random topology
  2. Chunk distribution dynamics
- Chunks are generated at a rate of λ=1 chunk/s, with size L=0.3 Mb (0.3Mbps)
- Source upload bandwidth is $u(S)$ = 1Mbps
- Peers are divided into three classes:
  - P_1 with upload rate [0.18,0.22]Mbps
  - P_2 with upload rate [0.3,0.4]Mbps
  - P_3 with upload rate [3,4]Mbps

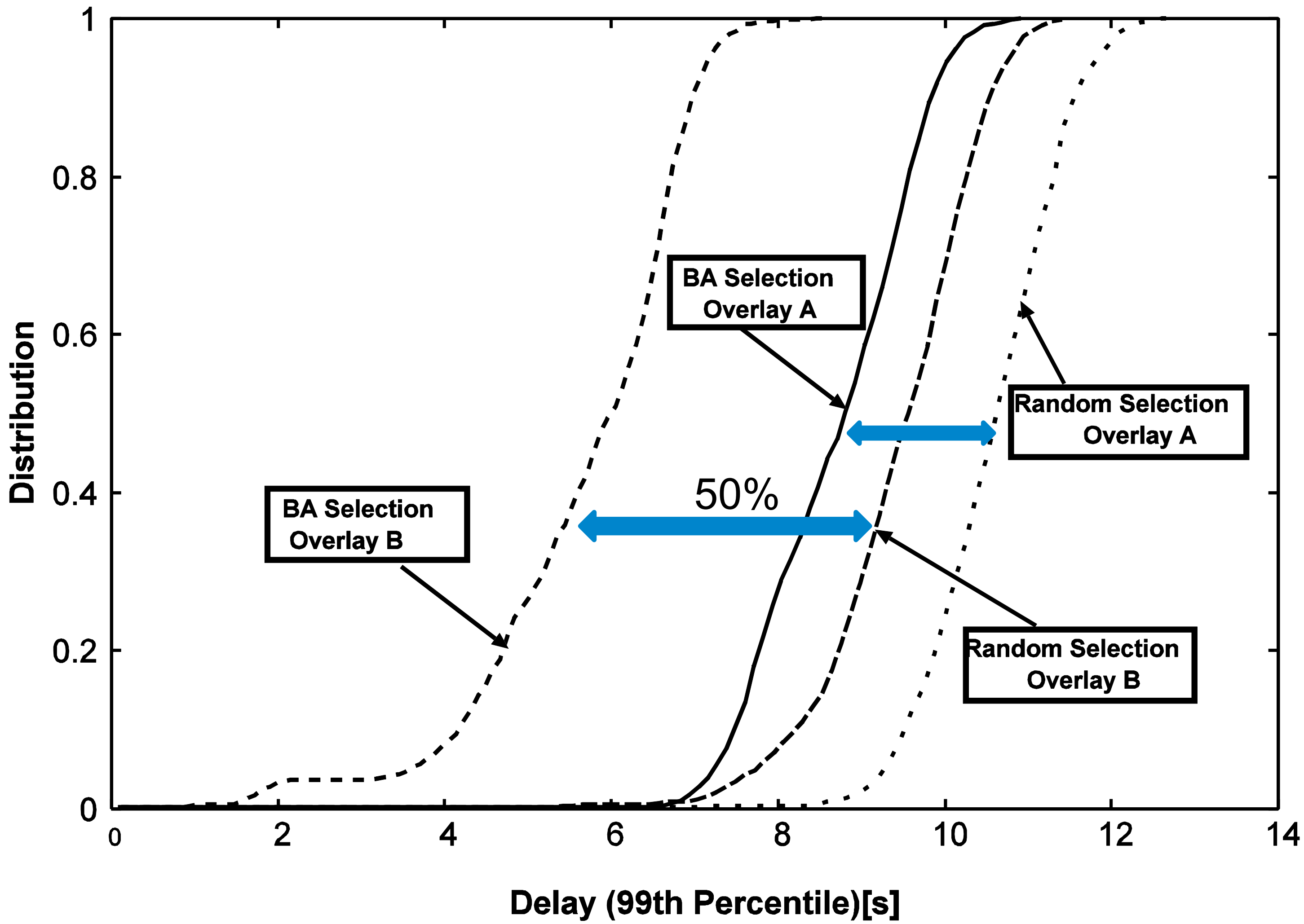
Simulation setting

- Performance indices:
  - *Delay*: Time interval between chunk generation at the source and chunk reception at the peer; mean value and distribution of the 99-th delay percentiles over all peers
  - *Losses*: Percentage of lost chunks

BA vs Random: Simulated scenario

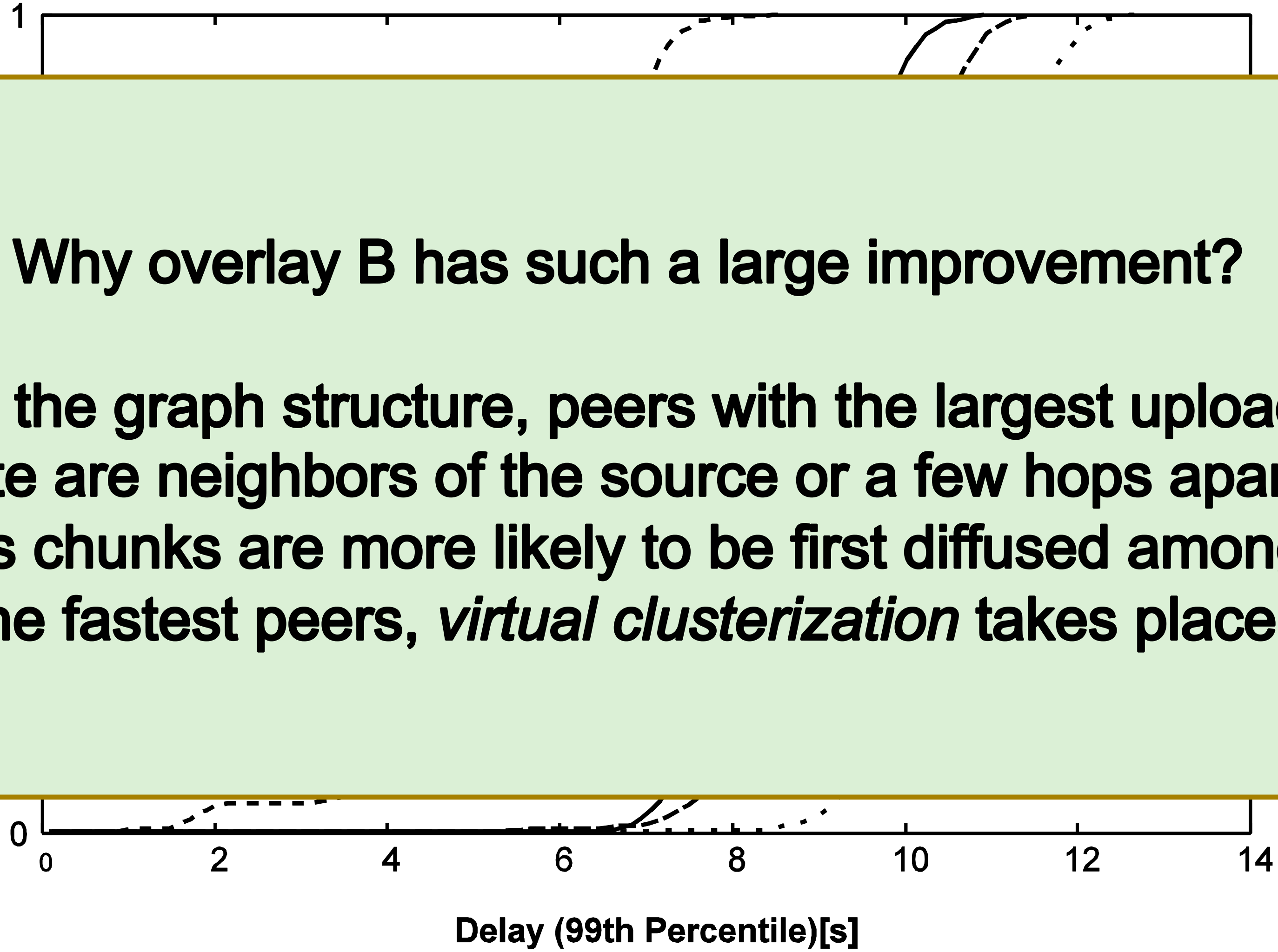
- **Compare BA with random peer selection**
- Overlay with 1,000 peers
  - 7% in P_3 -- high upload bandwidth
  - 27% in P_1 -- low upload bandwidth
- 25 different overlays are randomly selected
- Results are obtained by averaging over 50 independent simulation runs
- 200 chunks are distributed
- Peer window size is equal to 5

BA vs Random



BA Selection
Overlay A
BA Selection
Overlay B
Random Selection
Overlay A
Random Selection
Overlay B
50%
Distribution
Delay (99th Percentile)[s]
0
0.2
0.4
0.6
0.8
1
0
2
4
6
8
10
12
14

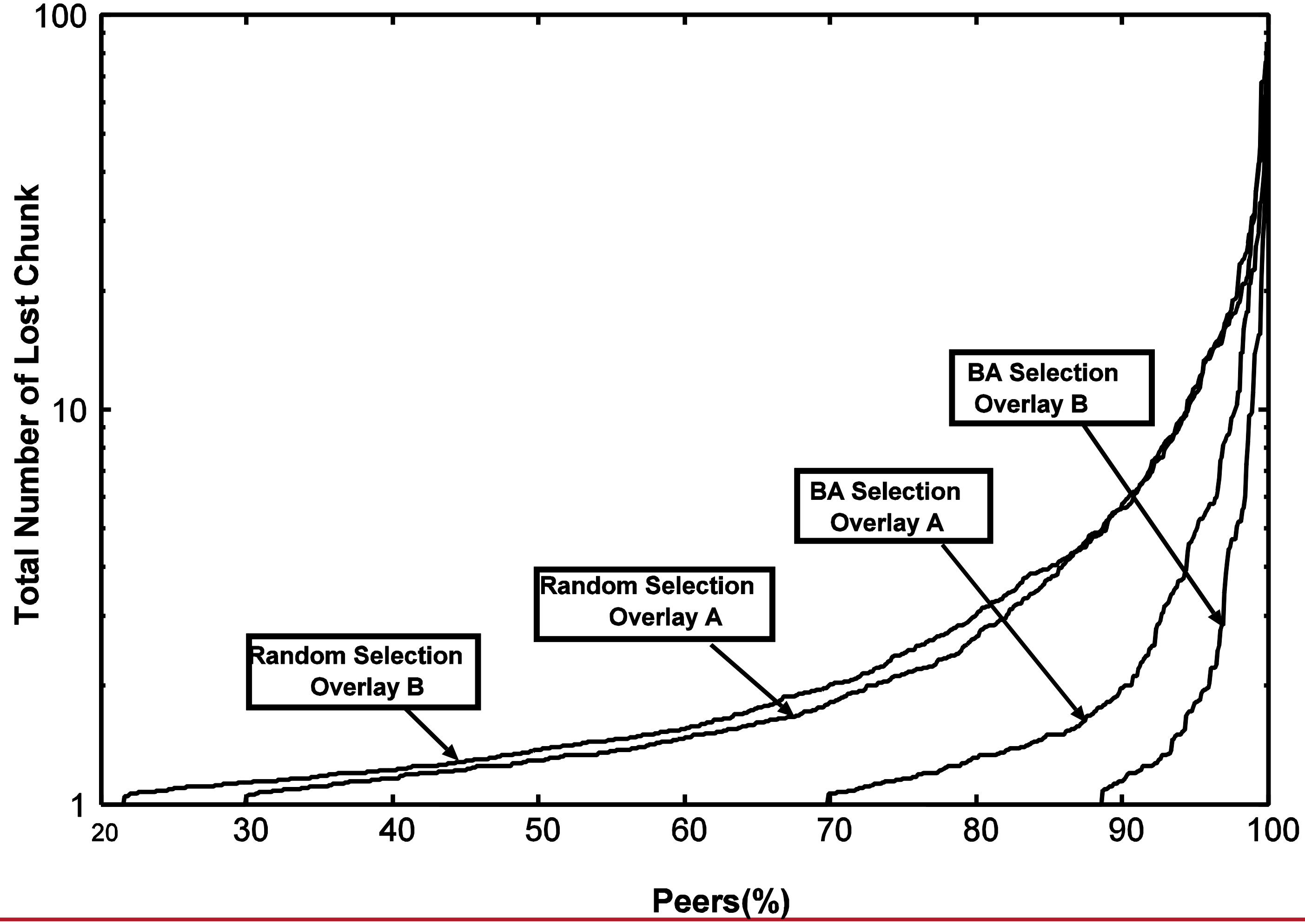
BA vs Random



Why overlay B has such a large improvement?

In the graph structure, peers with the largest upload rate are neighbors of the source or a few hops apart. As chunks are more likely to be first diffused among the fastest peers, *virtual clusterization* takes place!

BA vs Random: Losses



BA vs Random

Further observation:

- The window size has limited impact:
  - losses slightly decrease with the window size
  - marginal impact on the delay

BA neighborhood size setting: Variable Neighborhood

- By increasing the degree, delay and losses decrease
- How to deal with signaling issues?
  - Signaling bandwidth increases with the degree
  - Cost of managing the neighborhood increases with the degree
- So far, any peer *p* selects *k* neighbors at random
  - *k* should be large to improve performance
  - *k* should be small to reduce signaling overhead

BA neighborhood size setting: Variable Neighborhood

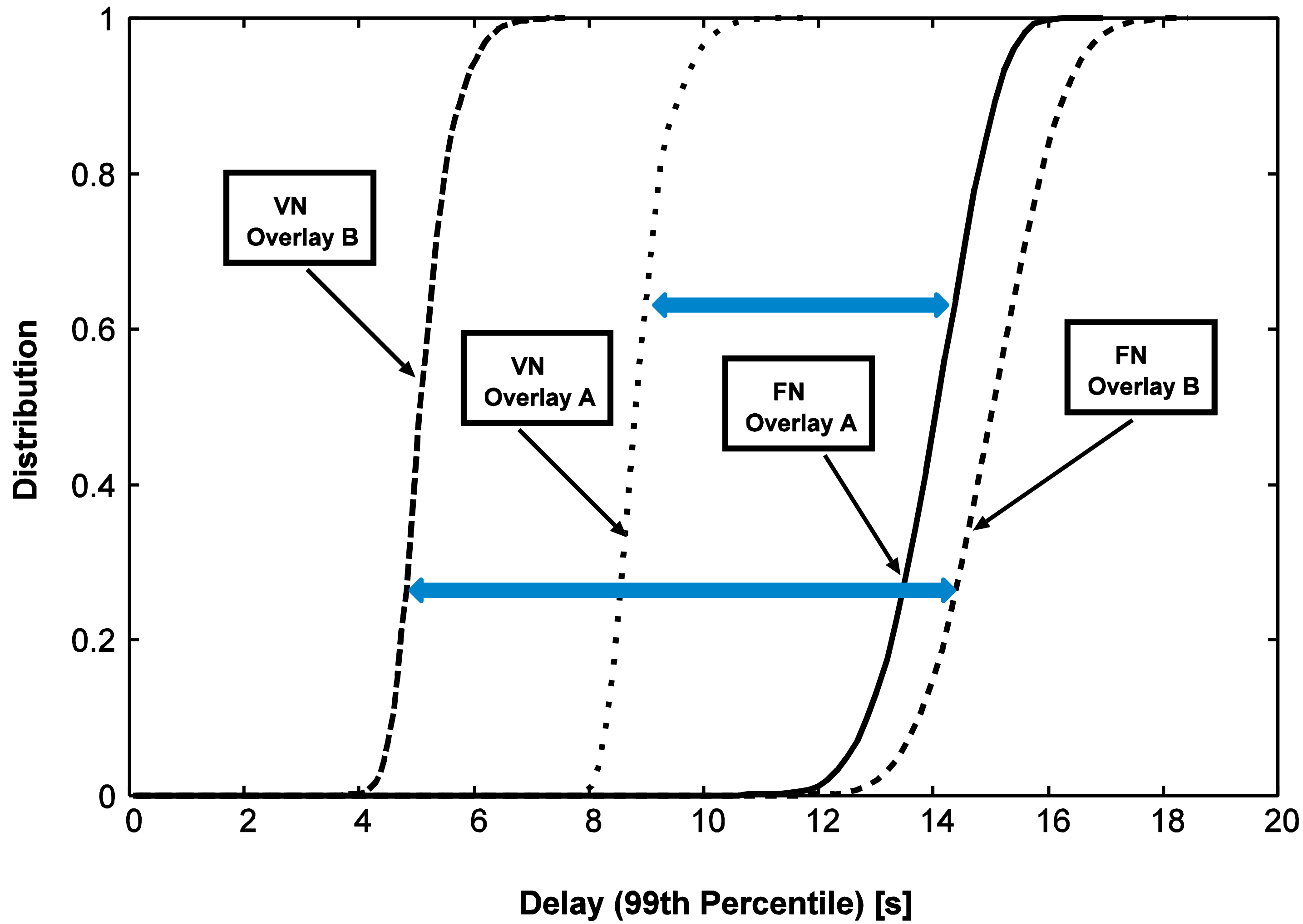
- Decide the peer neighborhood size (peer degree) based on the peer upload bandwidth
- Only peers with large bandwidth, that can contribute more and can actually exploit a large neighborhood, have many neighbors

The number of neighbors is chosen so that the time to deliver a chunk to all neighbors is larger than the chunk inter-generation time

Variable neighborhood: Simulated scenario

- Overlay with 10,000 peers
  - 7% in P_3 -- high upload bandwidth
  - 27% in P_1 -- low upload bandwidth
- 20 different overlays are randomly selected
- Results are obtained by averaging over 15 independent simulation runs
- 500 chunks are distributed
- Peer window size is equal to 5

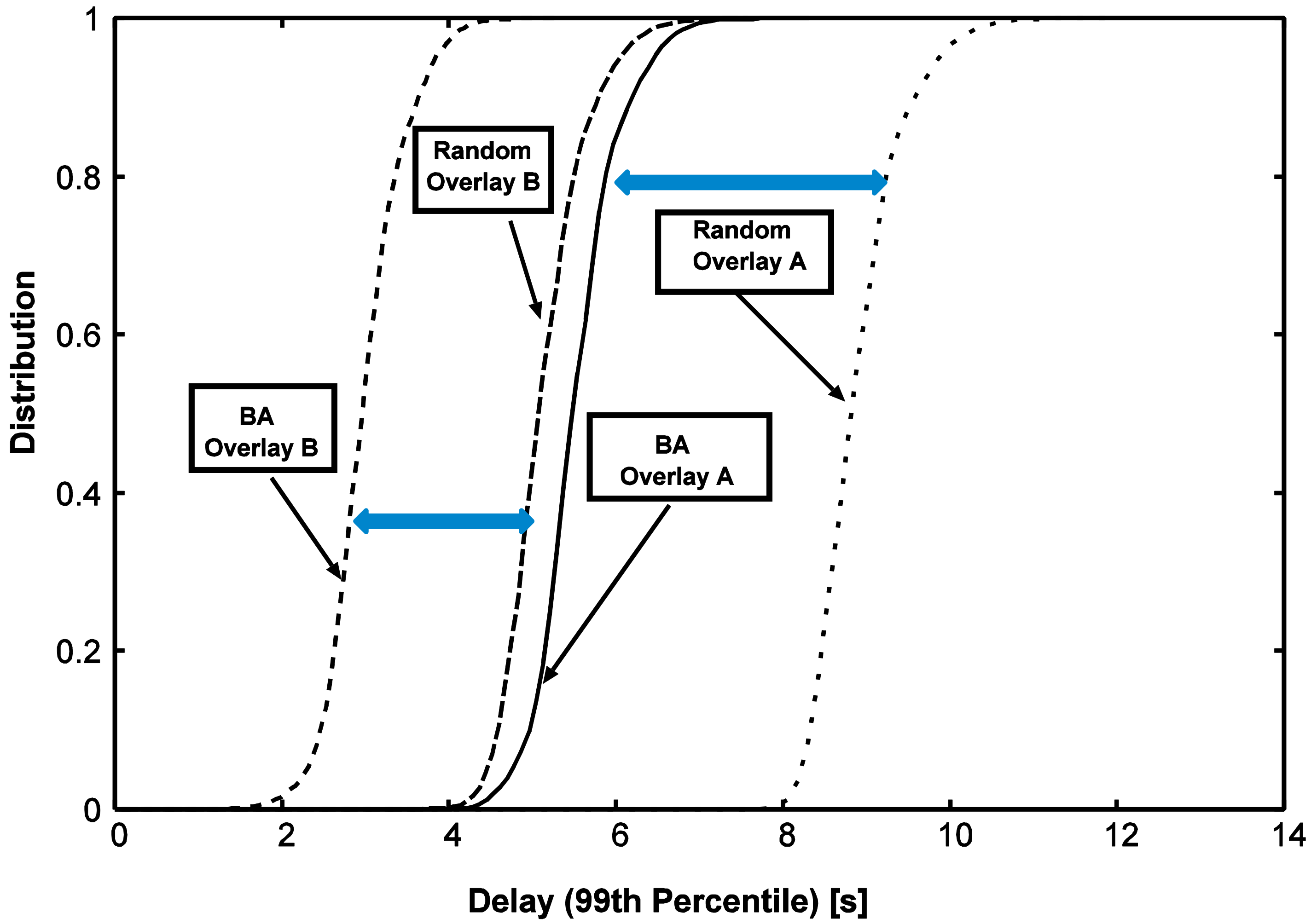
Variable neighborhood (rnd selection)



NAPA-WINE

TELECOMMUNICATION NETWORKS GROUP

Coupling the two BA schemes



Conclusions

- Exploiting peer heterogeneity, favoring peers with largest upload bandwidth improves the overall performance of the system

- Moreover, favoring altruistic peers can represent an incentive to collaboration.

Work in progress

- Algorithms and modeling:
  - Development of fluid models for a set of scheduling algorithms (including BA)
  - New scheduling strategies that take into account video coding: the different perceptual importance of different segments of the stream and the use of FEC
  - Development of a simulator for P2P streaming systems

- Thanks for your attentions

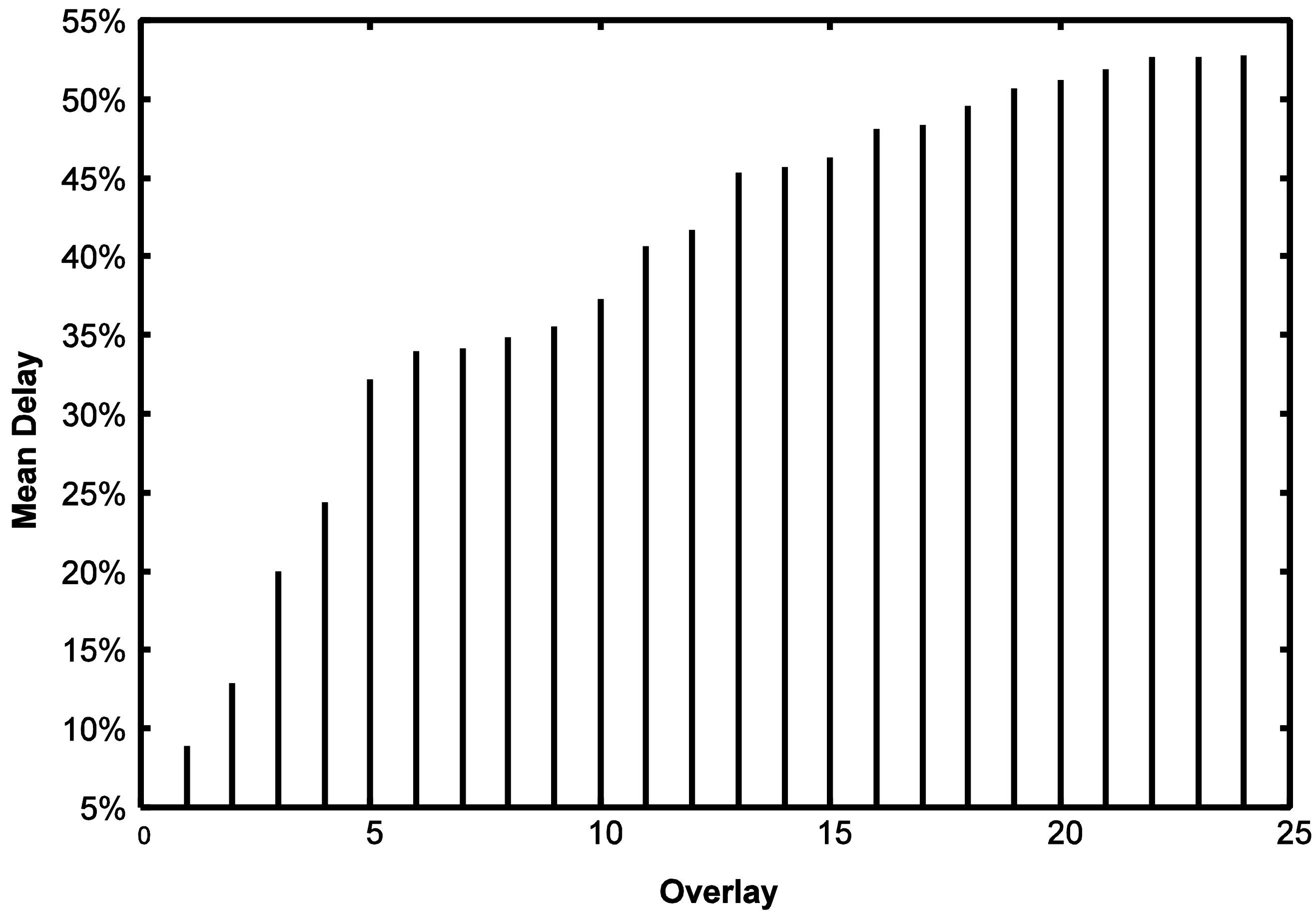
- Information about the napa-wine project are available at:

  http://www.napa-wine.eu

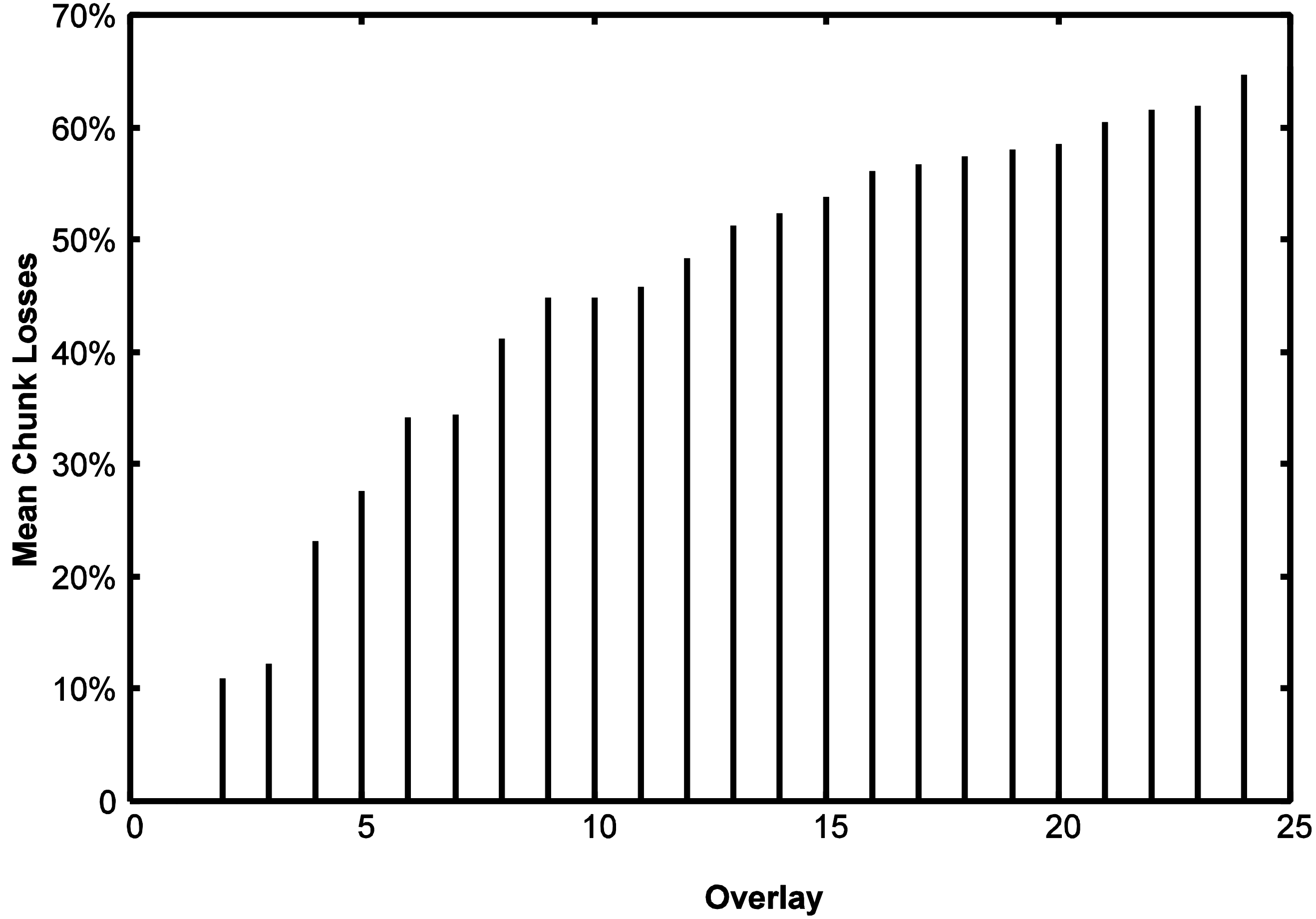
Work in progress

- We are developing a P2P streaming simulator:
  - Introduction of churning and dynamical overlay maintenance
  - Development of a simulator that integrates a layer 3 structure below the overlay
  - Differentiation of chunks based on their perceptual importance

Impact of the degree: Mean delay improvement



Impact of the degree: Mean losses improvement



NAPA WINE

BA vs Random: Losses