Adaptive Overlay Topology for Mesh-Based P2P-TV Systems

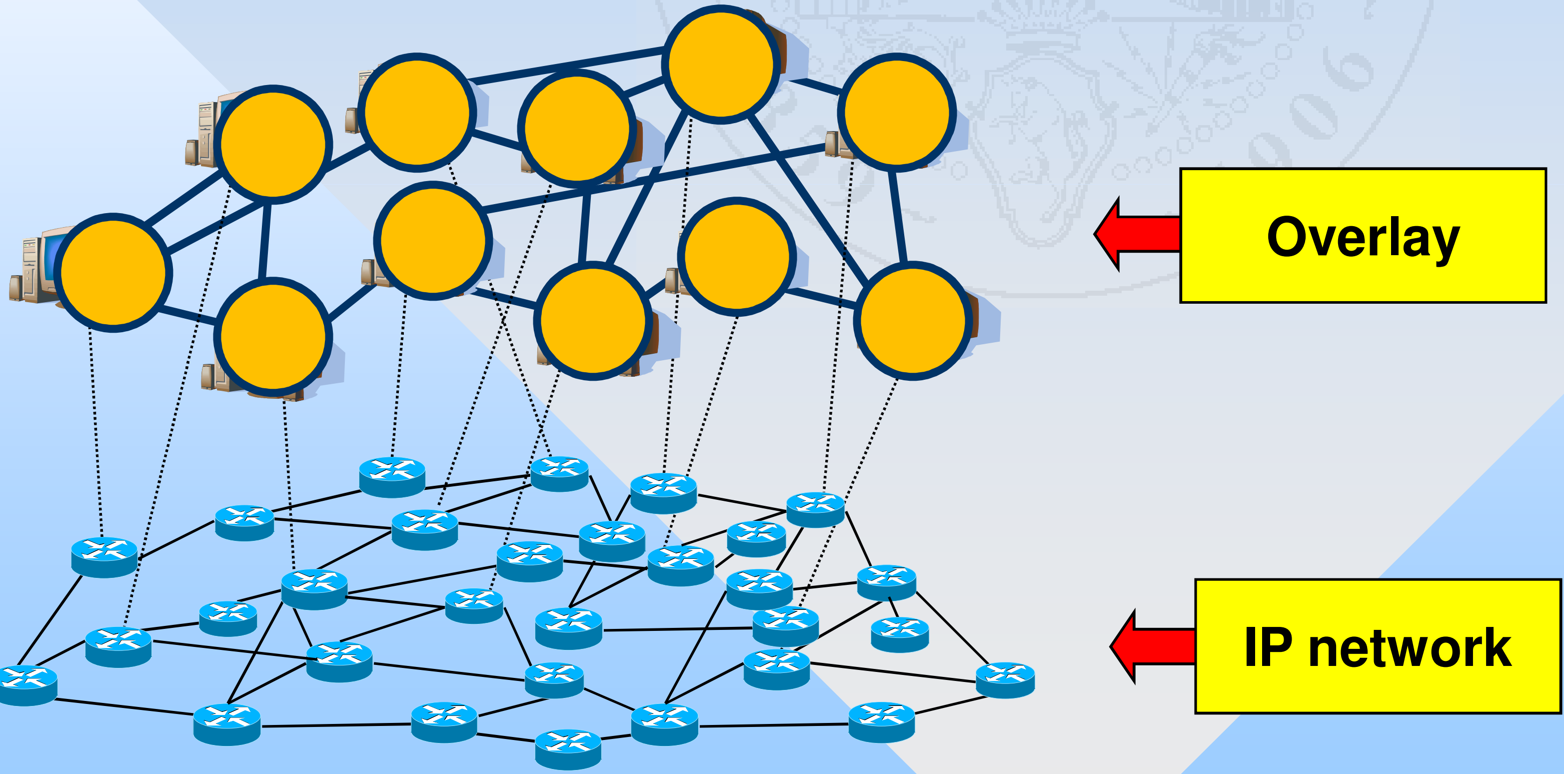
Richard Lobb (Univ. Canterbury, NZ)
Ana Paula Couto da Silva (Univ. Juiz de Fora, BR)
Emilio Leonardi
Marco Mellia
Michela Meo

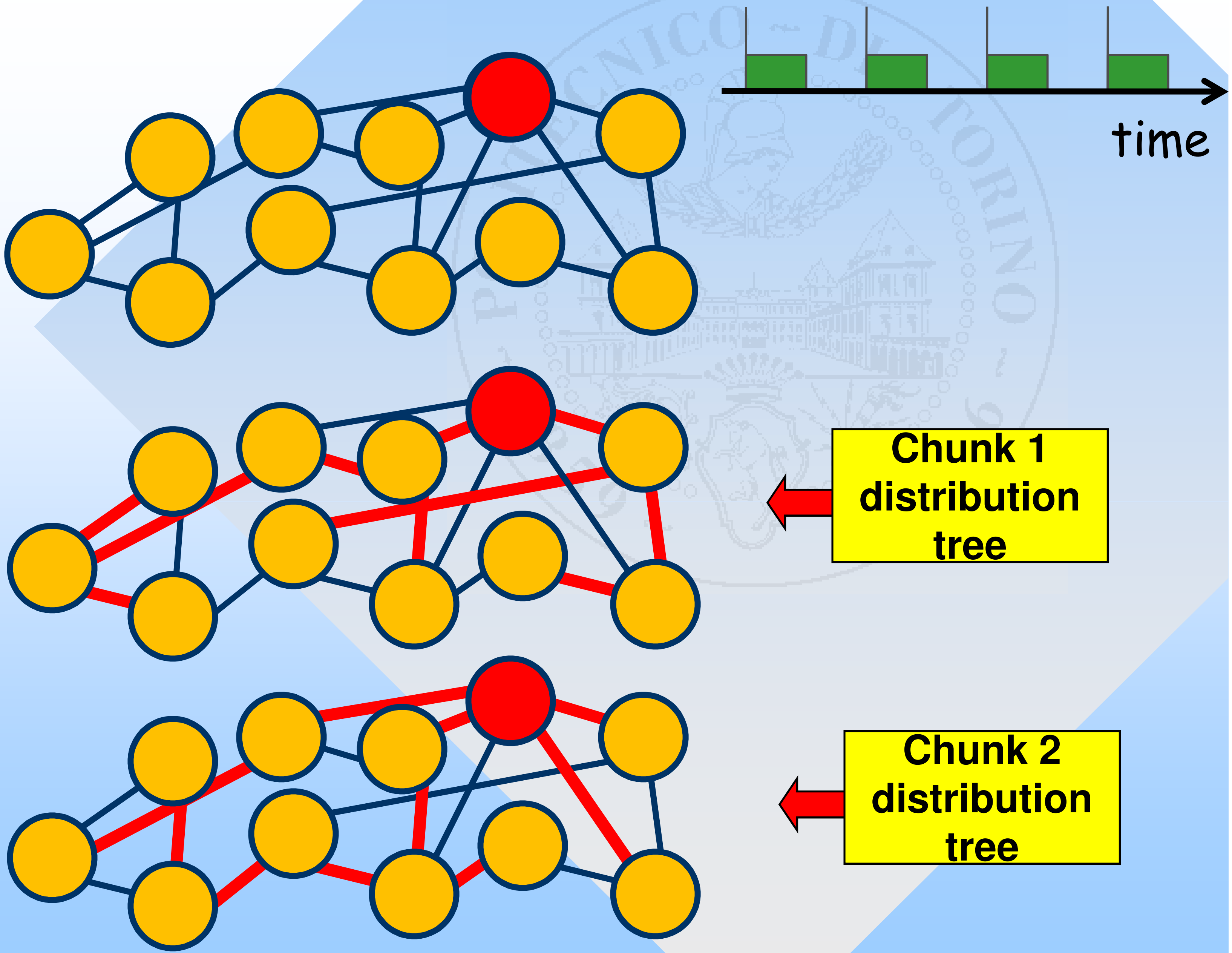
Politecnico di Torino, Italy

NAPA WINE

Mesh-Based P2P-TV Systems

- Distribute TV channels over the Internet in a P2P fashion
- Video is generated by a *source* and received by users with short delay
- The video information is organized in small *chunks* that are individually and independently distributed by the participating peers

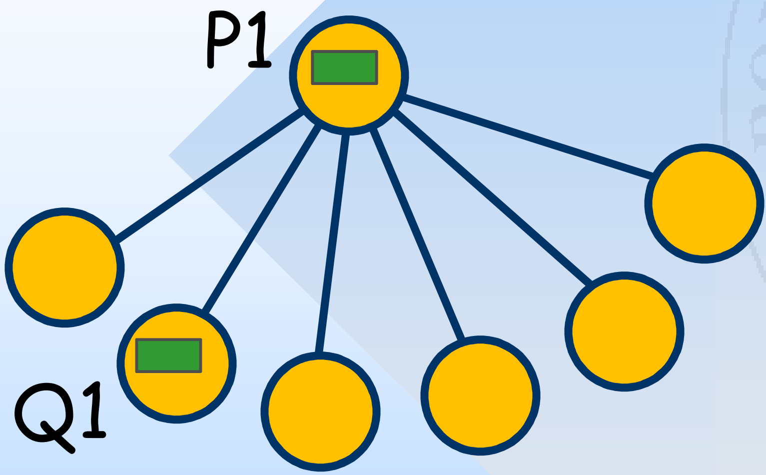

Overlay
IP network


time
Chunk 1 distribution tree
Chunk 2 distribution tree

Chunk distribution process

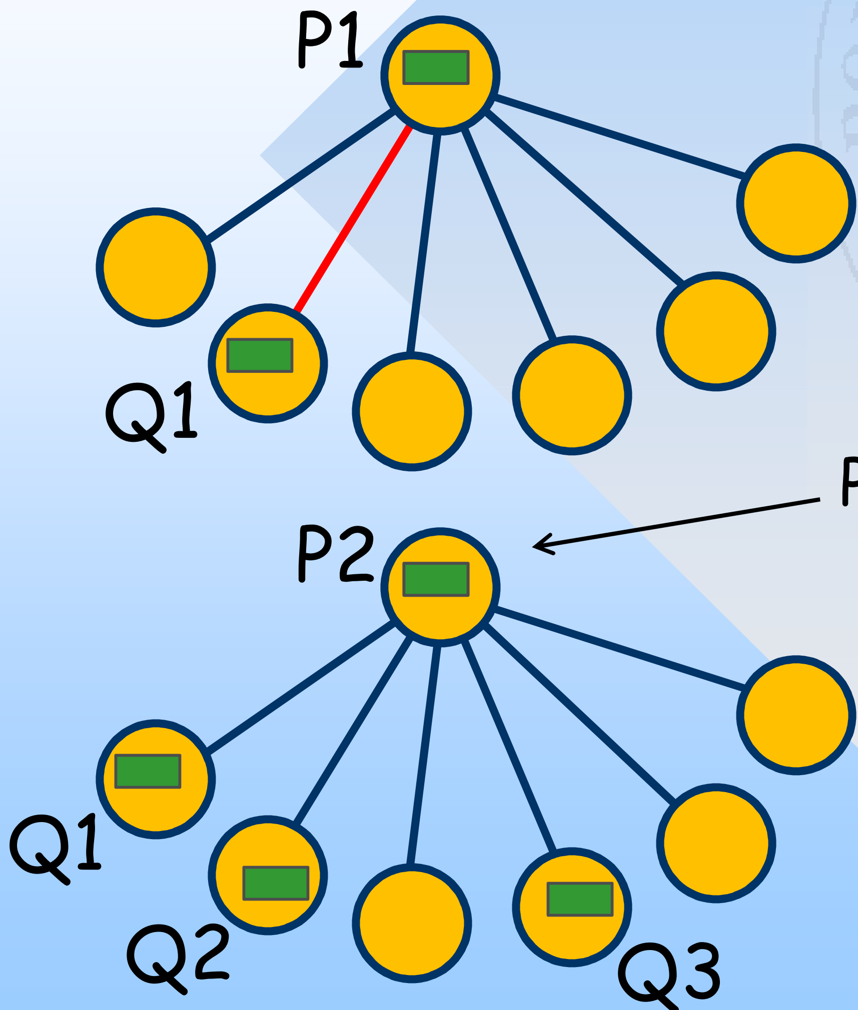
- Each peer maintains a (small) transmission window with chunks to be redistributed to other peers
- A scheduling process decides which chunk to distribute to which neighboring peer
- The most critical phase of the chunk distribution is the initial one (when it is rare)

Chunk distribution process



After a delay
ChunkSize/Bw1
1 more peer can contribute

Chunk distribution process



After a delay
ChunkSize/Bw1
1 more peer can contribute

P2 has 3 times the Bw of P1

After the same delay
ChunkSize/Bw1
3 more peers can contribute

Upload bandwidth matters…

- Peers that can contribute more to the chunk distribution (high bw peers) should be
  - → favored during the scheduling process

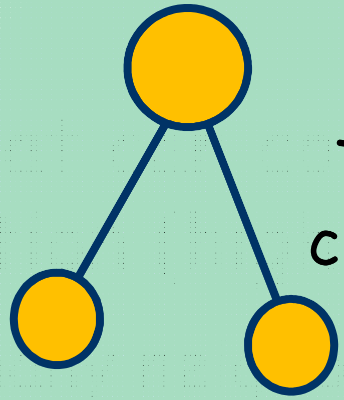
Upload bandwidth matters…

- Peers that can contribute more to the chunk distribution (high bw peers) should be
  → favored during the scheduling process

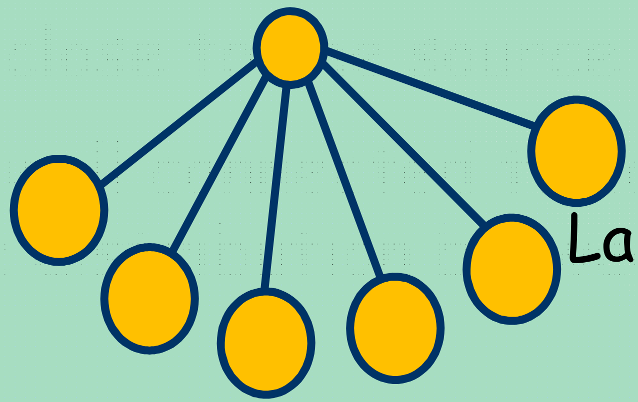
High bw peers should be preferentially served first

Scheduling is not enough

- Pee
  dist
  - → H
  - → B
  - → B
    th



High bw, small out-degree:
Too few neighbors to distribute the chunk to, the bw is not well exploited



Low bw, large out-degree:
Large neighborhood is useless for distribution (little bw),
while it is costly to manage

Scheduling is not enough

- Peers that can contribute more to the chunk distribution (high bw peers) should also
  - → Have many neighbors (they can use their bw)
  - → Be close to the sour
  - → Be well connected to the distribution betw

Overlay construction and maintenance are crucial

Our proposal key points

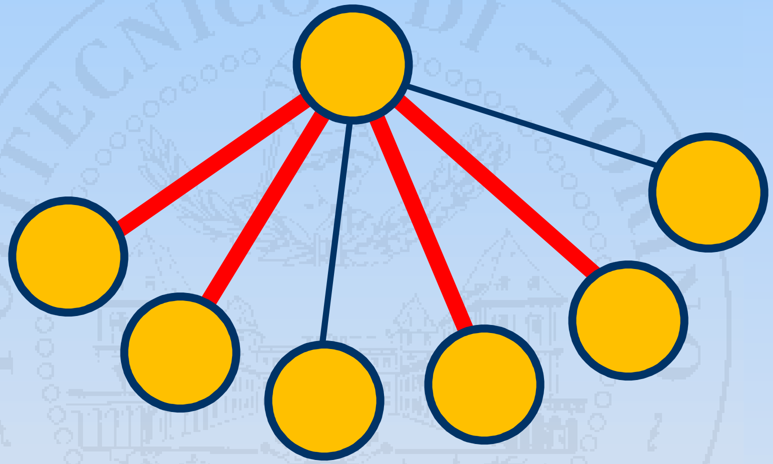
- Overlay maintenance algorithm that
  - → Automatically *adapts the out-degree* (neighborhood size) to the actual capacity of the peer to contribute to chunk distribution
  - → Automatically makes high bw peers highly connected and close to the source
  - → Does not require the *explicit* estimation of peer bw

Set a time window

Every time window…

…compute the

fraction of used links

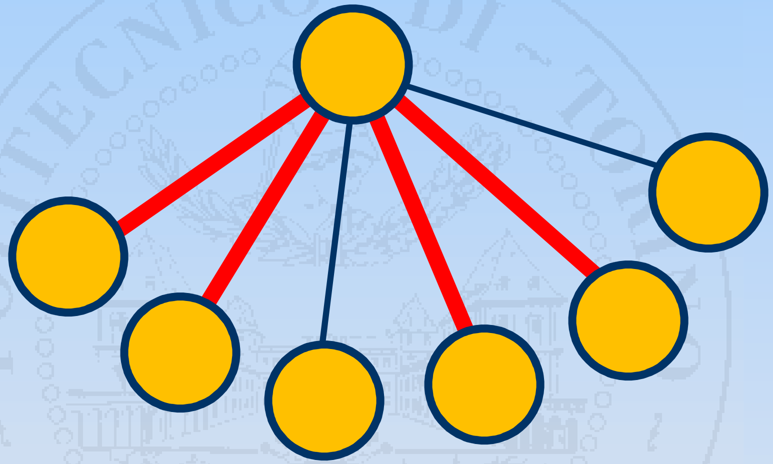


Use this value as an indication of peer capacity to contribute to chunk distribution

- Large? **Increase out-degree** (the peer can contribute more)
- Small? **Decrease out-degree** (the neighborhood is too large)

Set a time window.

Every time window…
…compute the *fraction of used links*



Control the no. of unused links

$$\alpha_L U < \overline{U} < \alpha_H U$$

no. of *used* links

no. of *unused* links

Implicit estimation of bandwidth

- Fraction of used links is employed to adapt the out-degree to the capacity of the peer to contribute
- The peer out-degree is then used as an implicit estimation of its bandwidth
  - → For scheduling
  - → When choosing/dropping peers

Growing the neighborhood

- New peers are added choosing among the set of neighbors' neighbors
- Within this set, peers are chosen with a probability proportional to the *desirability function*
  - → The *desirability f*
    with out-degree
  - → Out-degree is thus
    the peer upload ba

Need to exchange the list of neighbors and their out-degree

Growing the neighborhood

- At peer p, neighborhood size L(p) increases by
  - → kL(p): k is the *growth factor*
- Initially, L(p) is small
  - → Low bw peers are not congested
- Initially, $k=k_i$
  k linearly decreases to $k_f < k_i$
  - → High bw peers quickly grow their neighborhood

Shrinking the neighborhood

- Cull from the neighborhood a number of links within the set of unused links, so that
- Set $\underbrace{\overline{U} = \frac{\alpha_L + \alpha_H}{2} U}_{\text{no. of desired unused links}}$

$$\alpha_L U < \overline{U} < \alpha_H U$$

no. of *unused* links

no. of *used* links

Chunk scheduler

- Choose the latest useful chunk (latest chunk needed by some neighbor)
- Send chunk to neighbor q with probability proportional to the desirability function

$$p(q) = \frac{D(q)}{\sum_{r \in N(c,p)} D(r)}$$

set of neighbors needing the chunk

$$D(q) = \sqrt{L(q)}$$

desirability function

Scenario & evaluation

- Discrete event simulator

Download P2PTVSim from www.napa-wine.eu

- 10,000 chunks (1000s at 10 chunk/s rate)
- N=10,000 peers, partitioned in 4 classes
  - → Class 1 (10%): Bw=5Mbps
  - → Class 2 (40%): Bw=1Mbps
  - → Class 3 (40%): Bw=0.5Mbps
  - → Class 4 (10%): Bw=0Mbps

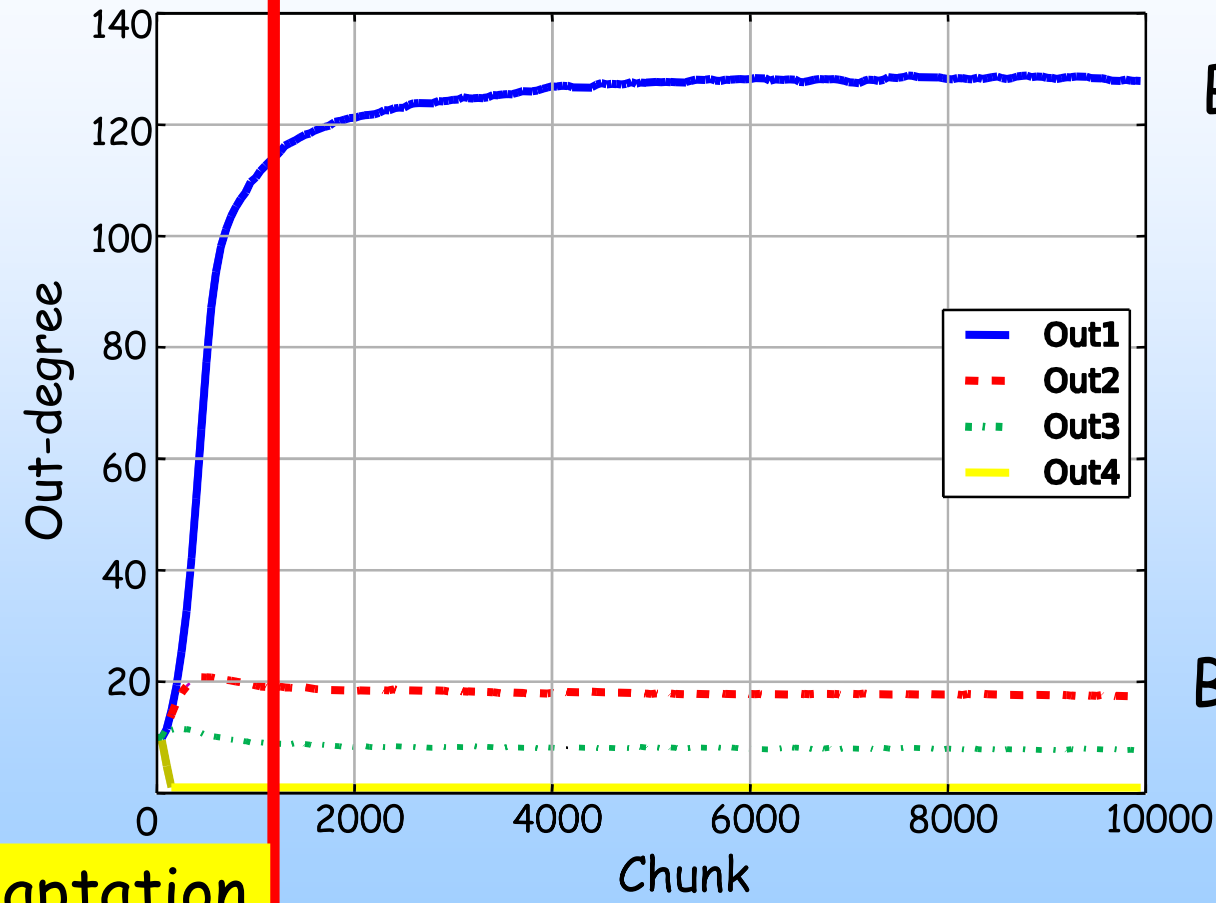
E[Bw]=1.1Mbps
(video rate=1Mbps)

Other parameters

- Chunk size 0.1Mb, 10 chunks/s
- Playout delay=5s (50 chunks)
- Start with L(p)=10
- Time window size: 50 chunks
- k_i=0.4, k_f=0.1, after 750 chunks
- a_L=0.1, a_H=0.3

THE ADAPTIVE OVERLAY

Neighborhood



Bw=5Mbps

Bw=1Mbps

short adaptation time (2 min)

Neighborhood

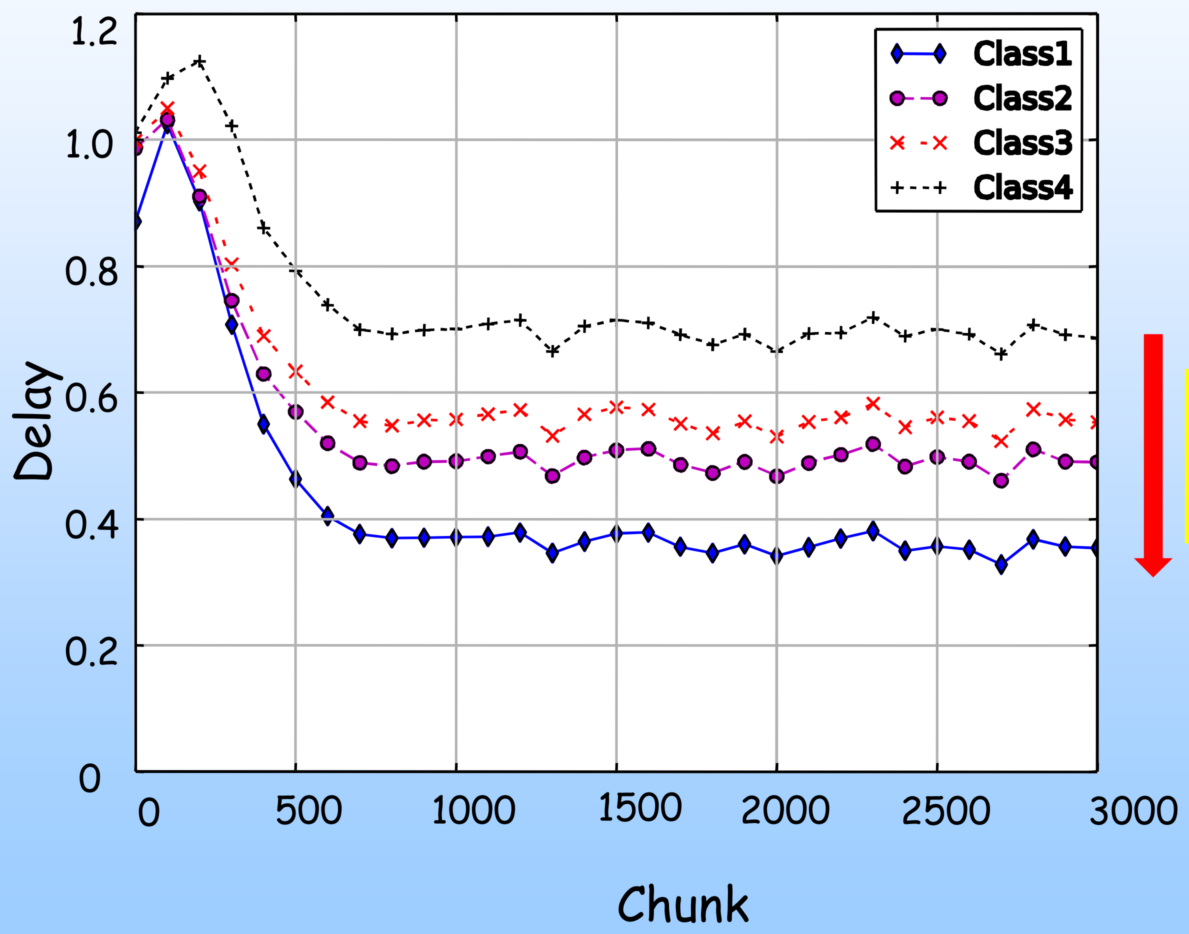
Per class: percentage of links to other classes

From\To	Class 1	Class 2	Class 3	Class 4	Degree
Source	33.6	44.0	21.6	0.9	116
Class 1	25.8	42.9	29.2	2.1	120.4
Class 2	19.6	35.7	36.7	8.0	14.5
Class 3	19.4	31.7	37.4	11.4	6.9

Peers prefer to be connected to high bw peers (10% of total only)

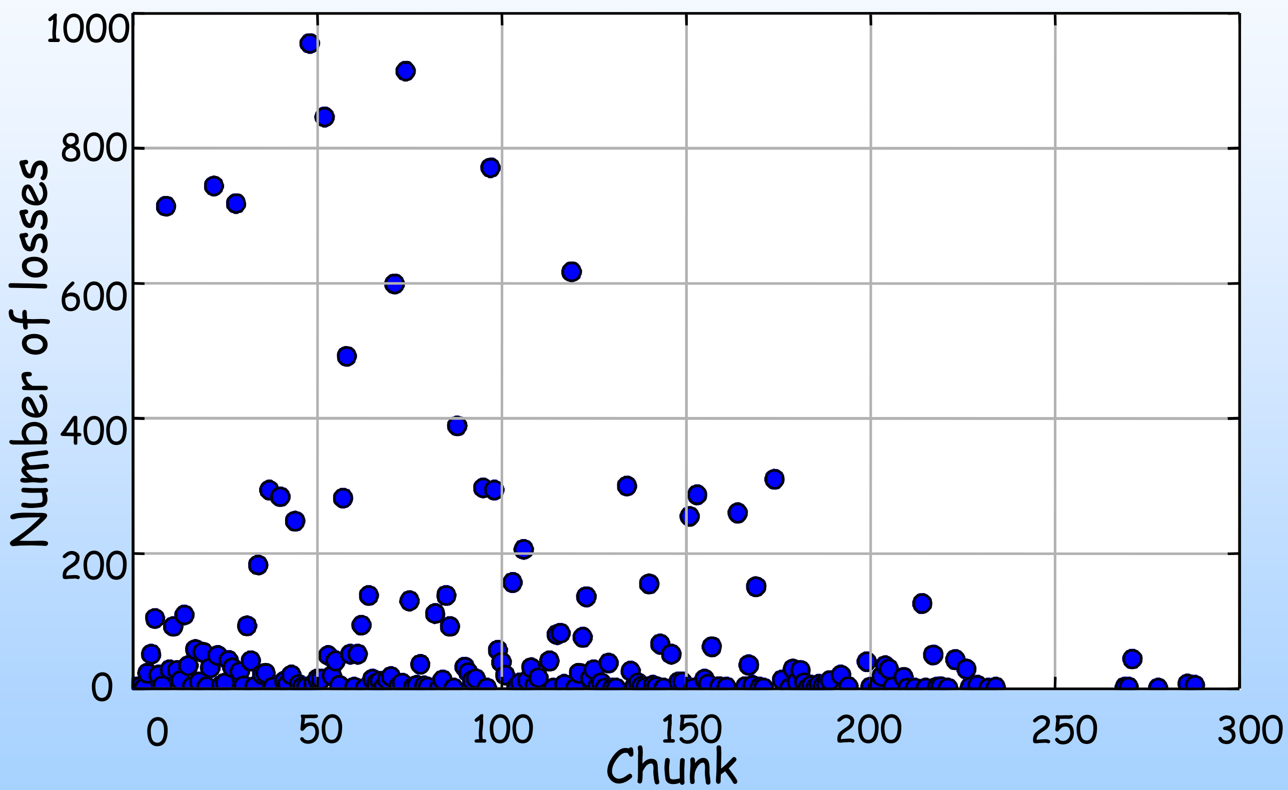
PERFORMANCE

Delay



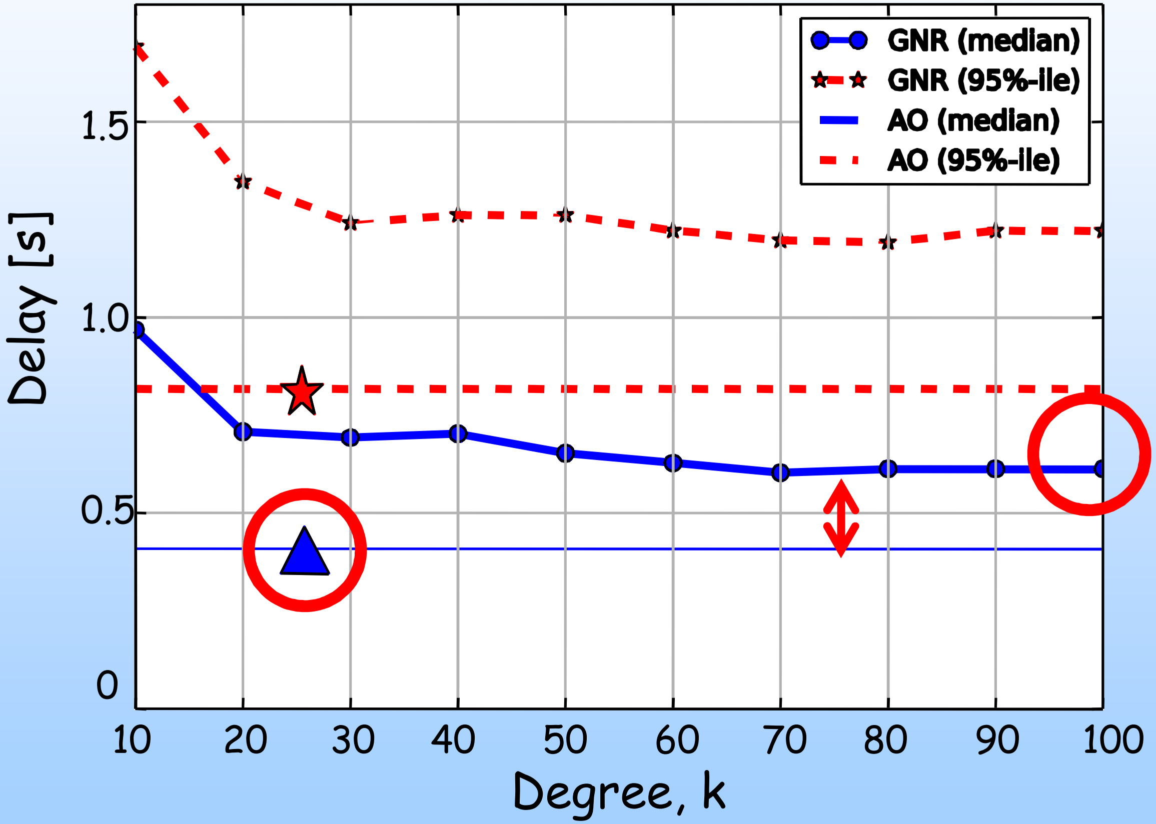
High bw peers are favored

Losses



Only during the initial transient

Improvement



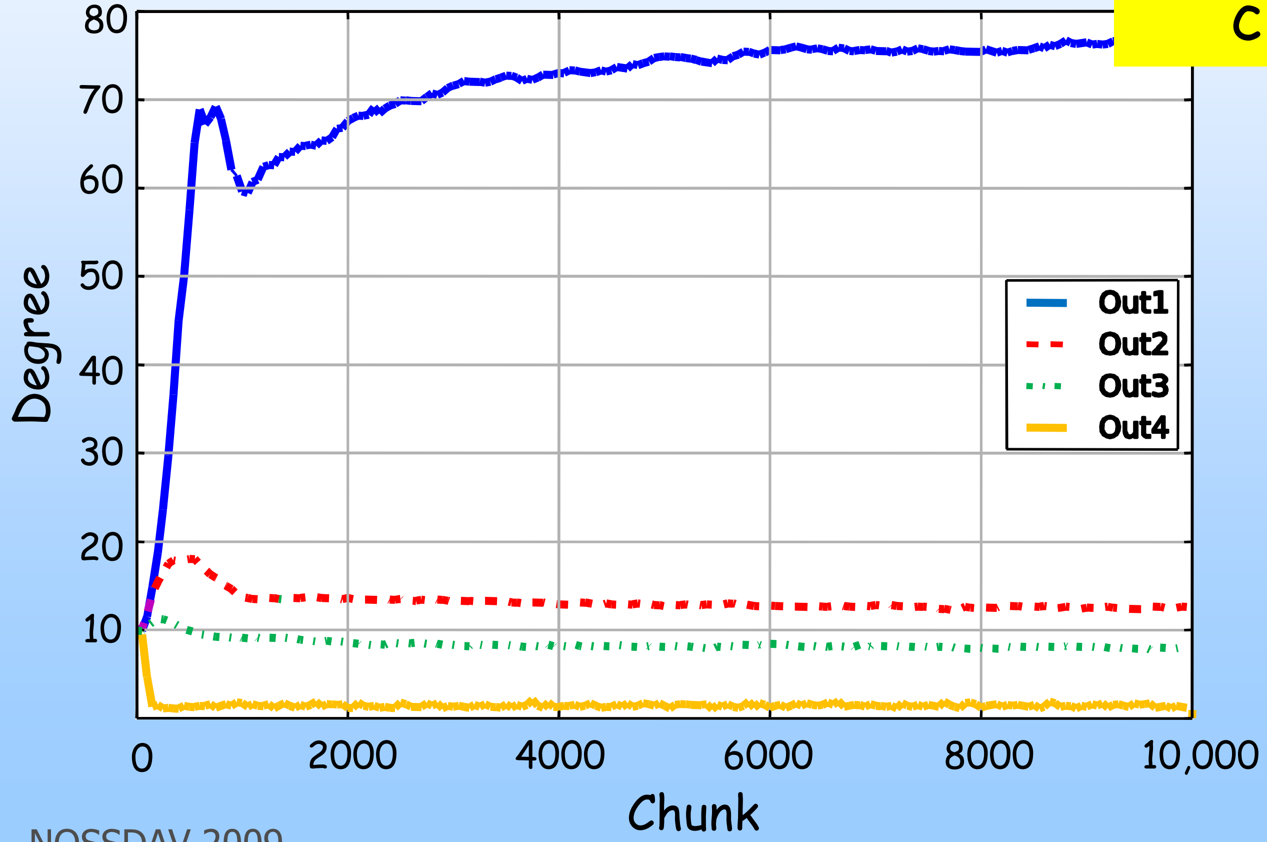
High bw peers
- more neigh.
- highly connected

Tot no. links smaller of a factor 4

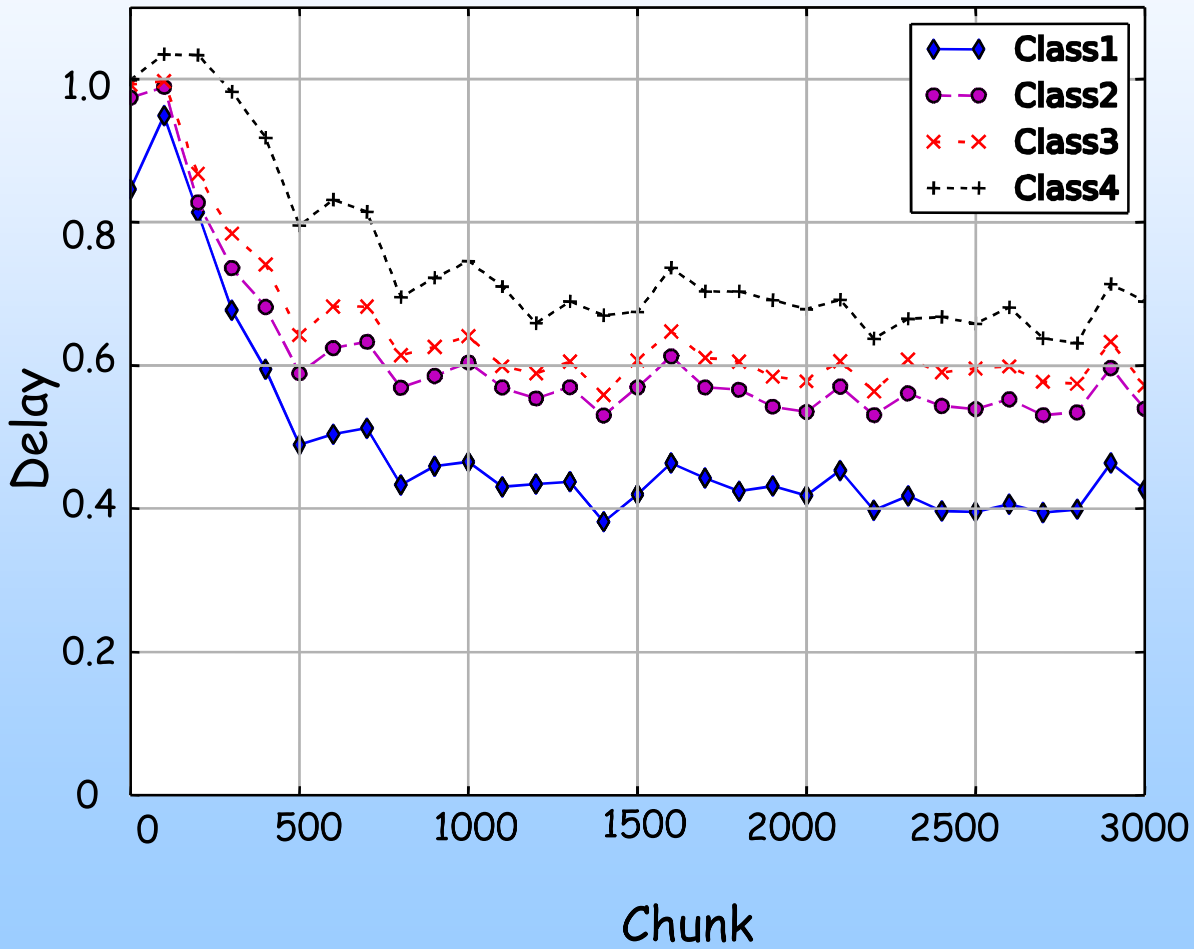
Churning

- 50% of the peers disconnect after a random time in [10,100]s

Robust to churning



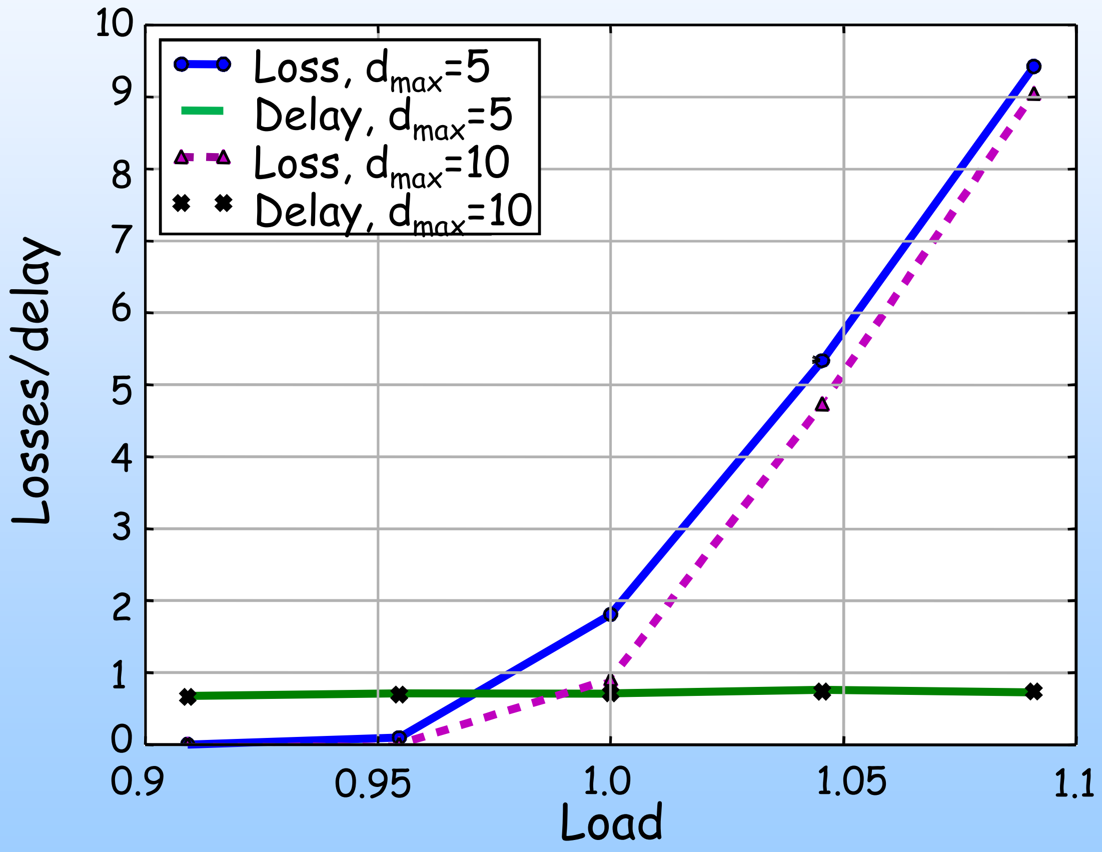
Churning


Class1
Class2
Class3
Class4
1.0
0.8
0.6
0.4
0.2
0
Delay
0
500
1000
1500
2000
2500
3000
Chunk

Conclusions

- Effective in *adapting* the overlay to peer heterogeneity
  - → Neighborhood size and topology
- Robust to varying conditions
  - → Churning
  - → Available bw variations
- Simple and does not need bw estimation
- Requires limited exchange of information
  - → List of neighbors and their out-degree

Impact of load



Sensitivity to parameters

α_H

α_L

	0.2	0.3	0.4	0.5	0.6	0.7
0.1	0.438	0.453	0.478	0.478	0.463	0.478
0.2		0.408	0.438	0.468	0.498	0.473
0.3			0.398	0.443	0.453	0.468
0.4				0.393	0.433	0.468
0.5					0.393	0.423
0.6						0.398

Delay decreases with large no. unused links