Swarming Techniques to Improve Live Streaming Performance in the PeerLive System

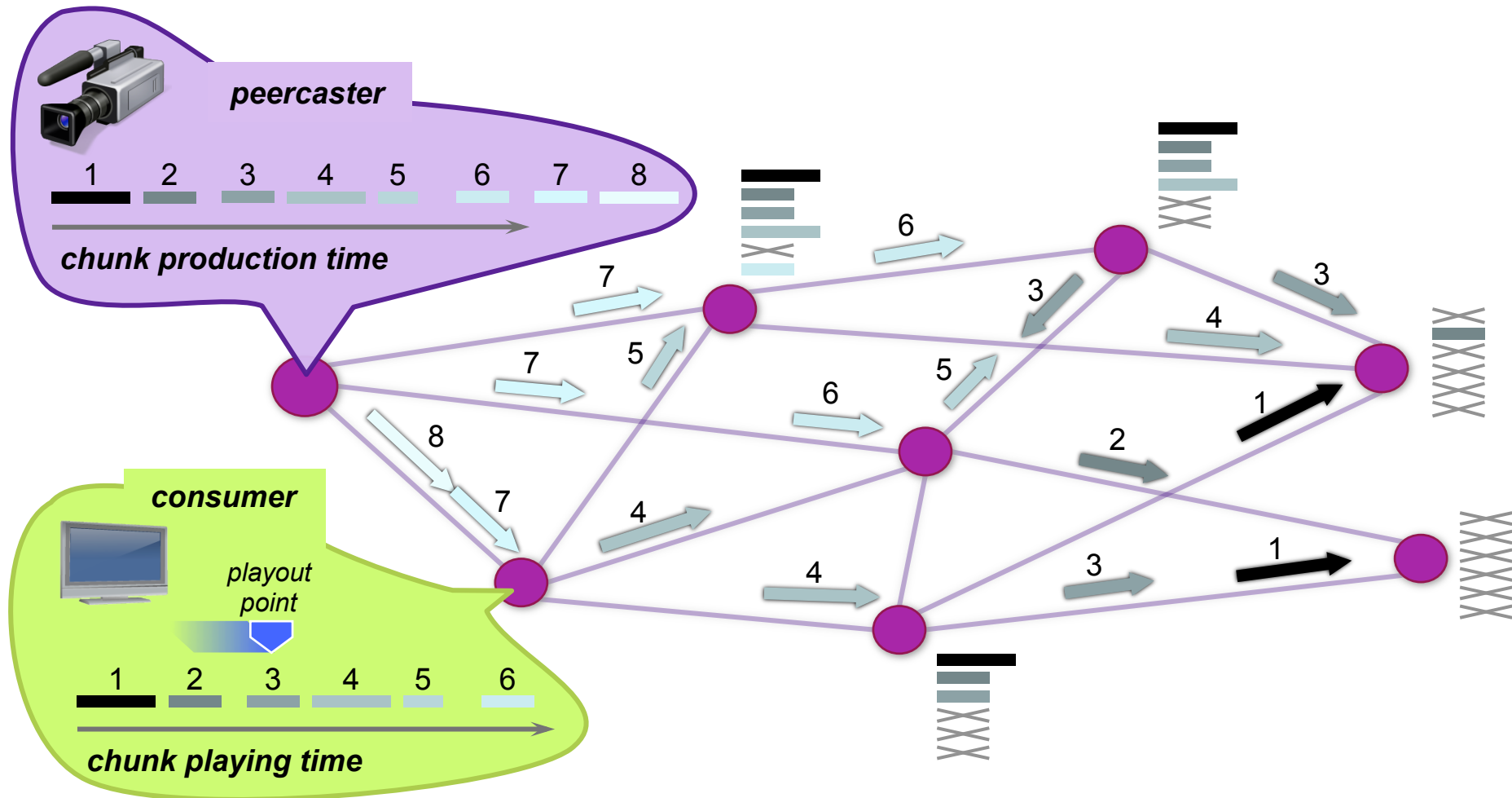Eleni Mykoniati (e.mykoniati@ee.ucl.ac.uk)

joint work with R.G.Clegg, D.Griffin, R.Landa, L.Latif, S.Spirou, M.Rio
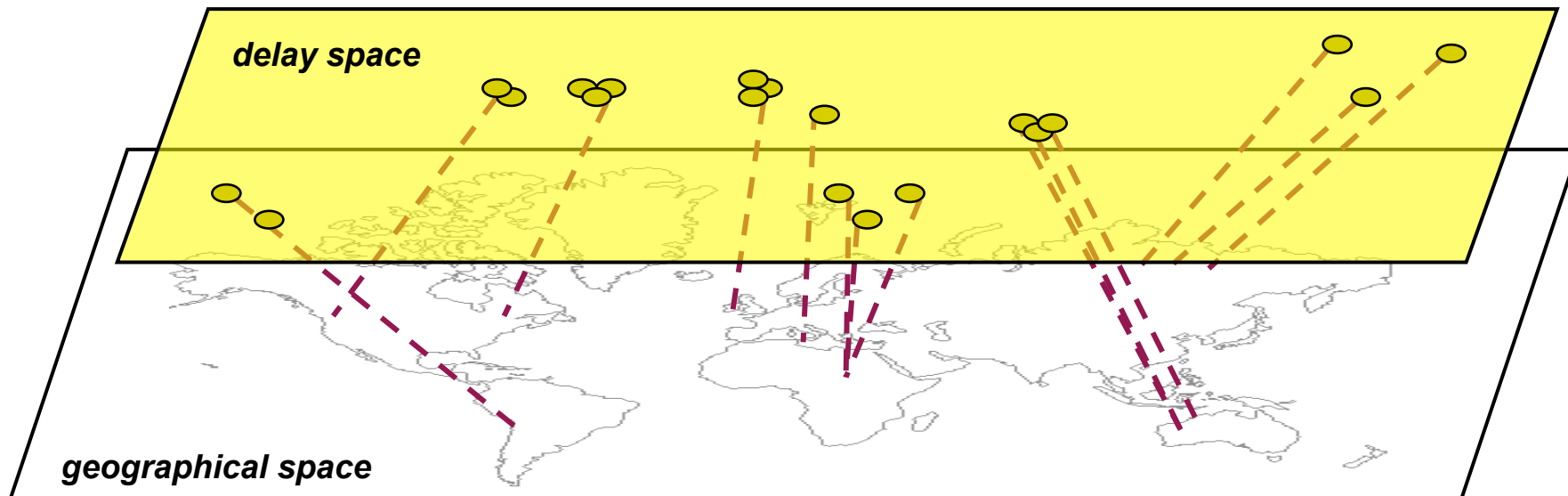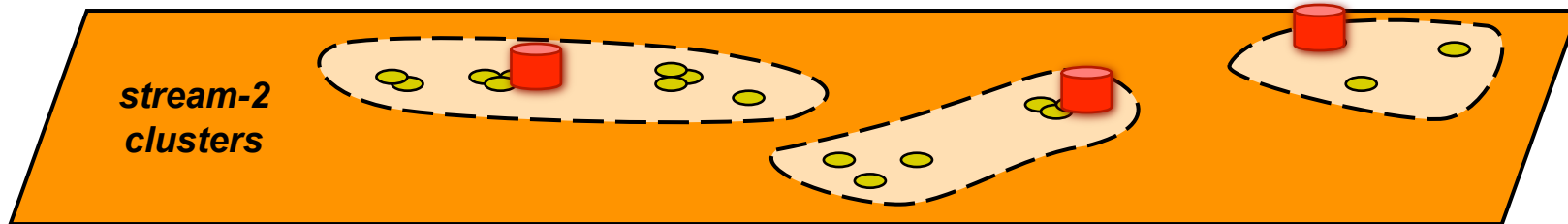
EPSRC-funded PeerLive project
Department of Electronic and Electrical Engineering
University College London

Multiservice Networks, Cosener's House, July 2008

live streaming & related performance aspects

peercaster

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

chunk production time

consumer

playout point

chunk playing time

| 1 | 2 | 3 | 4 | 5 | 6 |

- *playout lag* = how far is the playout point from the livest chunk at the peercaster
- *playout continuity* = the percentage of chunks not received in time for the playout
- *startup delay* = time elapsed until the playout starts

improving playout lag – locality and clusters

stream-2 clusters

stream-1 clusters

delay space

geographical space

- discover the nearest local tracker for a particular stream

- *Distributed Overlay Anycast Table (DOAT)* – an infrastructure to retrieve the nearest local tracker given the stream identifier
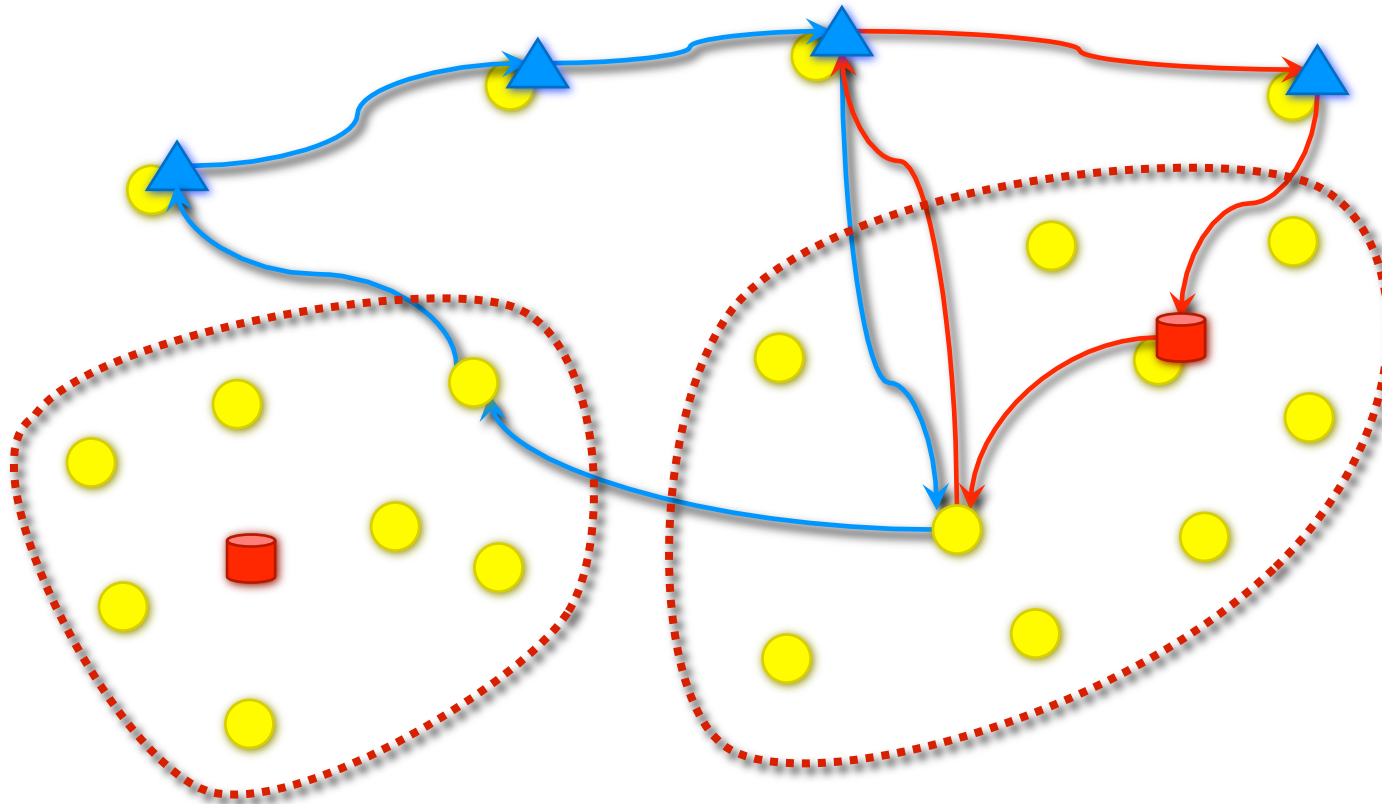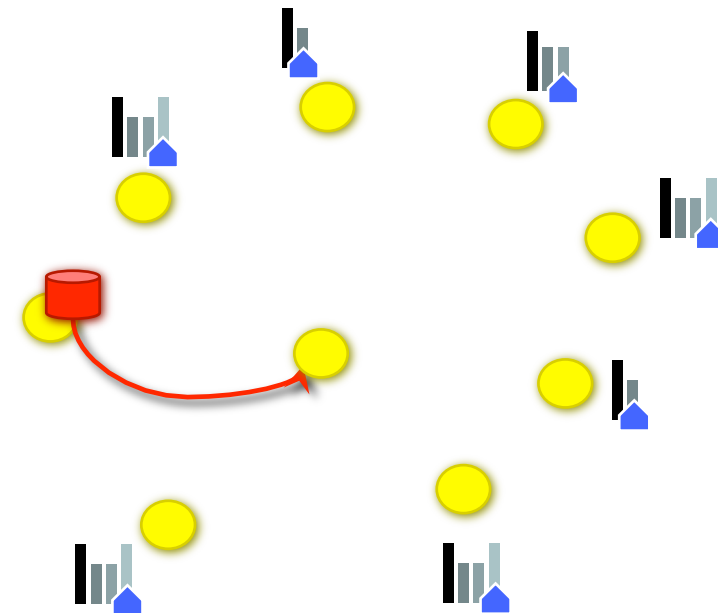
- DOAT is designed to route queries in hops of logarithmically decreasing distances until it hits the nearest local tracker entry

- delay coordinates are mapped with a space-filling curve to a one-dimensional coordinate
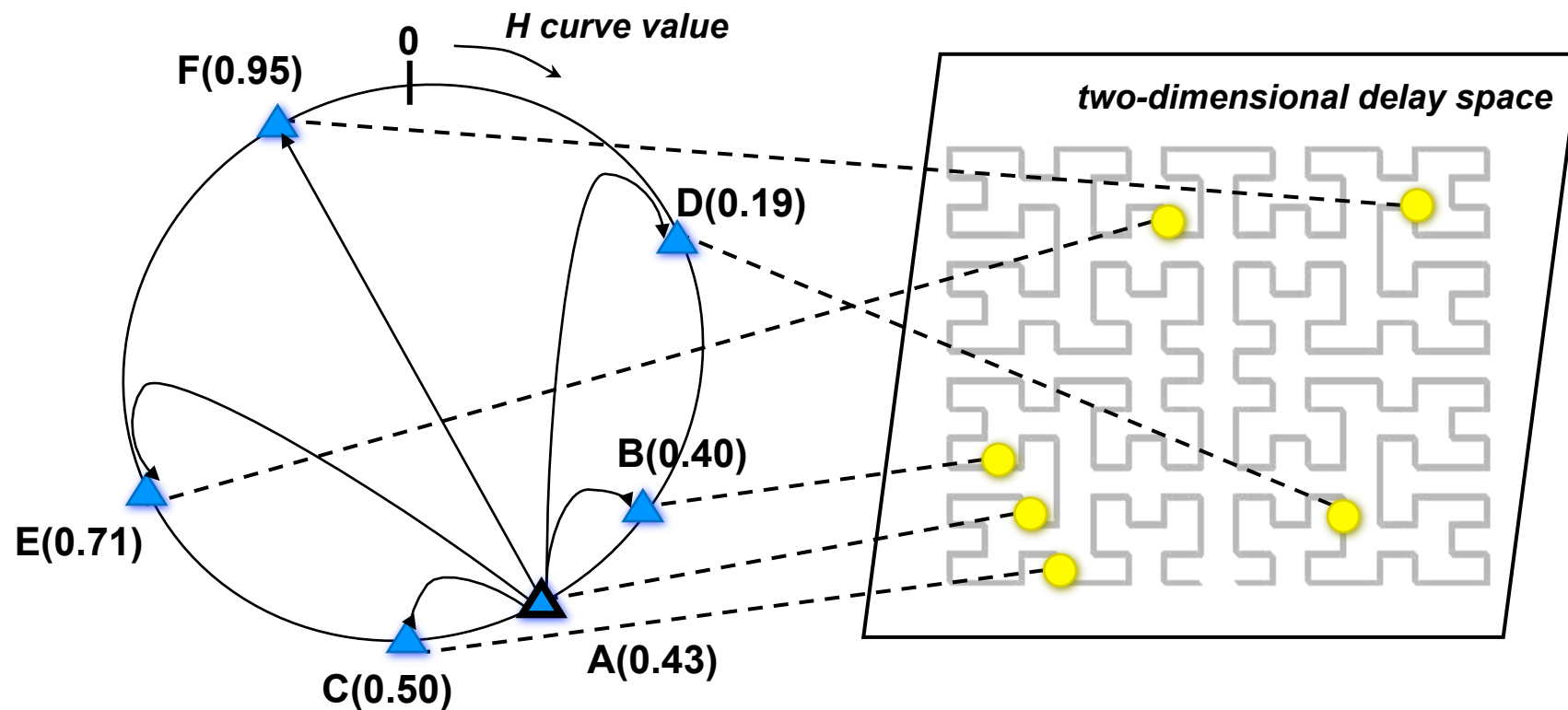
*discover content availability
by exchanging explicit information*
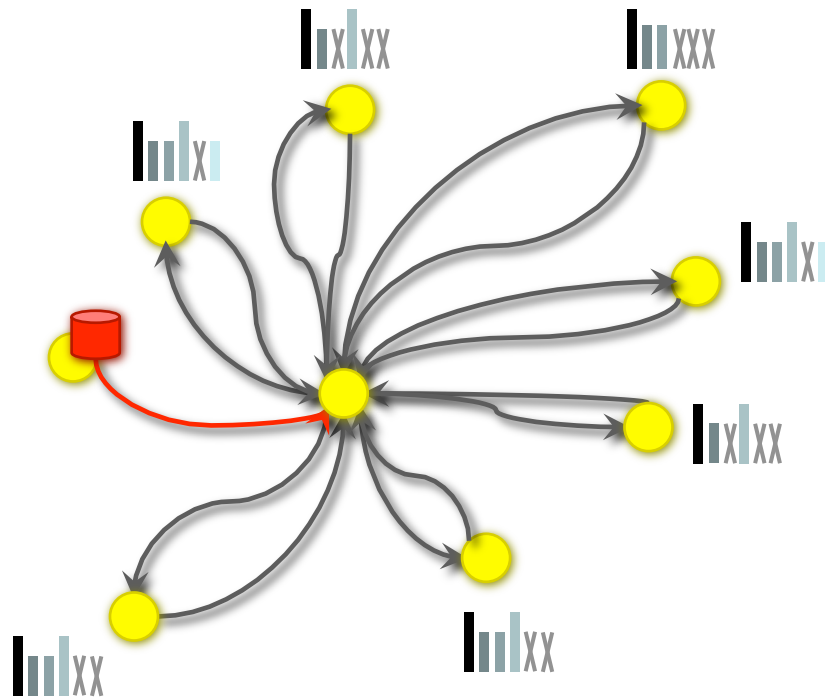
*deducing content availability
based on announced playout lag*

- unlike file distribution, in live streaming there is a correlation in the sequence of received chunks: to preserve playout continuity a peer strives to receive all chunks before the playout point

- the playout point at a given time determines the playout lag, which can be extrapolated at any other future time, without the need for exchanging explicit state

- P2P systems require that the peers contribute enough upload capacity to accommodate their download rates => the more capacity the higher the sustainable video quality

- more available capacity => higher probability to find a non-saturated peer => faster system response to demand and churn, less protocol overhead

- ***Non-Consuming Peers (NCPs)*** – peers that have no interest in viewing a stream, yet they have incentives to act as multipliers, contributing upload resources

UCL
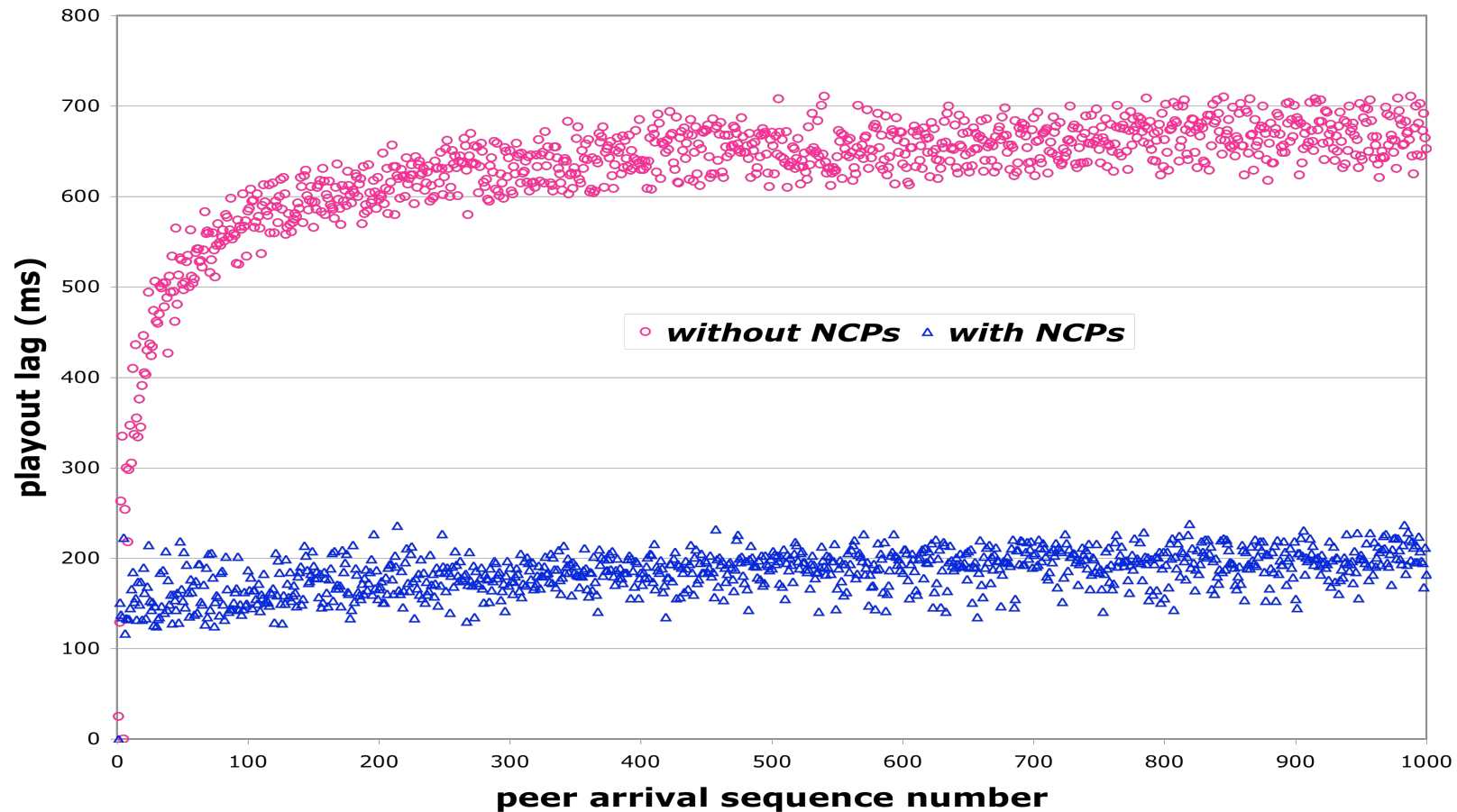
- built a simulator modelling overlay links using artificial and trace-based delay coordinates – *pointers to delay/capacity traces are always welcome* ☺
- implemented DOAT, clustering algorithm, and swarming strategies and have preliminary results
- refining peer swarming strategies and algorithms
- comparing with alternative approaches
- implementing a prototype

Thank you!