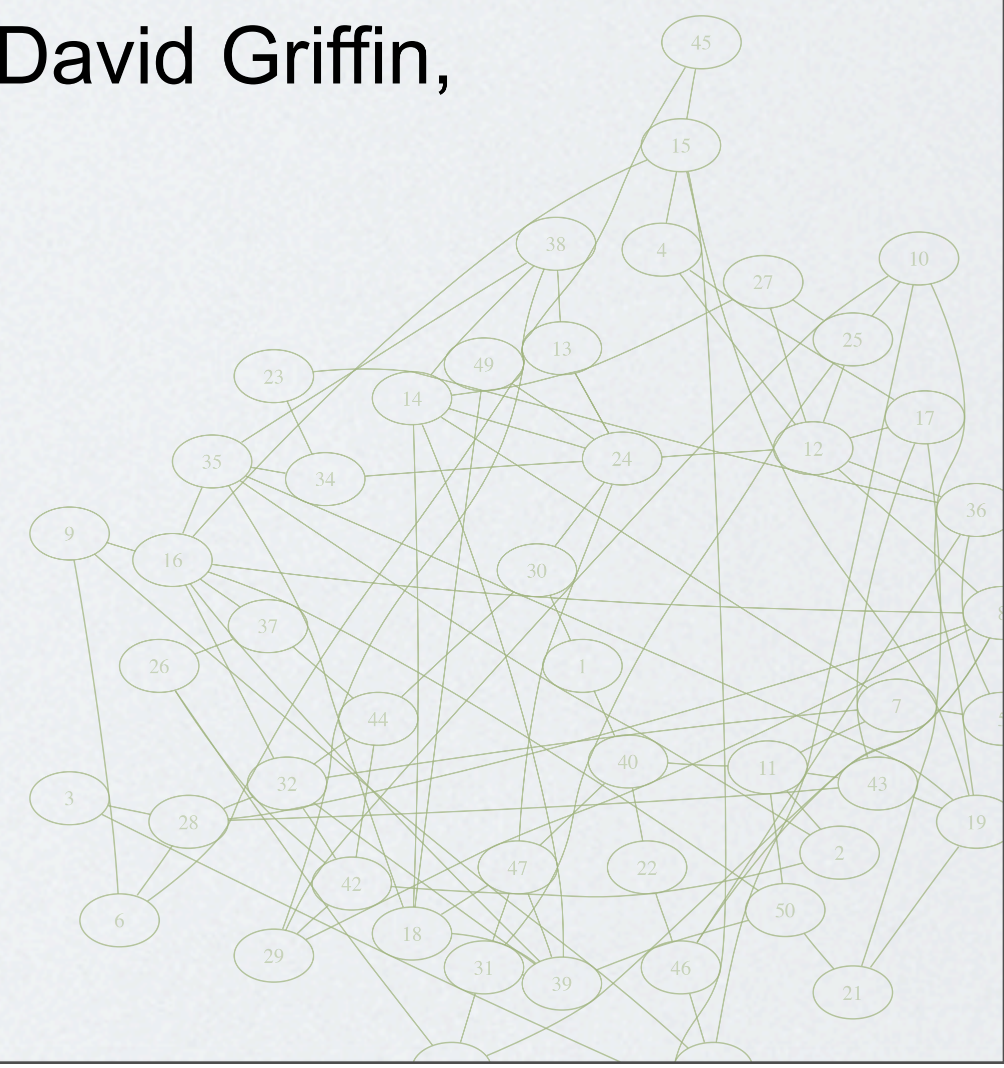
An Algorithm for Distributed Resource Allocation in QoS Overlays based on Vickrey-like Auctions

Raúl Landa, Richard Clegg, Eleni Mykoniati, David Griffin, Miguel Rio

Networks and Services Research Laboratory

Department of Electronic and Electrical Engineering

University College London



Our Objective

- Auctions are a well-known way of performing distributed resource allocation in networking
- However, they can suffer from slow convergence
- We address this by proposing algorithms to
  - **Resolve auctions faster**, by auctioning many items at the same time
  - **Reduce the number of wasted bids**, by allowing peers to estimate the probability of a bid being successful

Sealed-envelope, Highest-losing bid Auction (SHA)

- An auctioneer i sells N_i indistinguishable items.
- Each bidder j sends a set $(b^{ji}_{m_{ji}}, b^{ji}_{(m_{ji}-1)}, \ldots, b^{ji}_2, b^{ji}_1)$ of bids b^{ji} for all the items it is interested in, each for a value v_{ij}.
- Auctioneers rank all the received bids in increasing order, and the top N_i win the items.
- Each winning peer pays the value of the ***highest losing bid*** for its item.

Revelation Properties of the SHA

- We have proved analytically that truthful revelation of value is a dominant strategy equilibrium:
  - Conjecture, for each bidder, a *bidding strategy*

$$\left(b^{ji}_{m_{ji}}(v_{ij}), b^{ji}_{m_{ji}-1}(v_{ij}), \ldots, b^{ji}_2(v_{ij}), b^{ji}_1(v_{ij})\right)$$

  - Formulate peer utility as a function of this strategy

$$\phi_j = \int_0^{b^{ji}_1} m_{ji}(v_{ij} - y) f_V(y) dy + \sum_{k=1}^{k=m_{ji}-1} \int_{b^{ji}_k}^{b^{ji}_{k+1}} (m_{ji} - k)(v_{ij} - y) f_V(y) dy$$

  - Find the strategy that maximizes peer utility

$$b^{ji}_{m_{ji}} = b^{ji}_{m_{ji}-1} = \ldots = b^{ji}_2 = b^{ji}_1 = v_{ij}$$

Estimating SHA Outcomes

- Use of statistics of the k-th smallest value of a multi-item statistical sample (***order statistics***)
- Express $\mathbb{P}({V^i}_{(k)} \leq v_{ij})$ in terms of order statistics

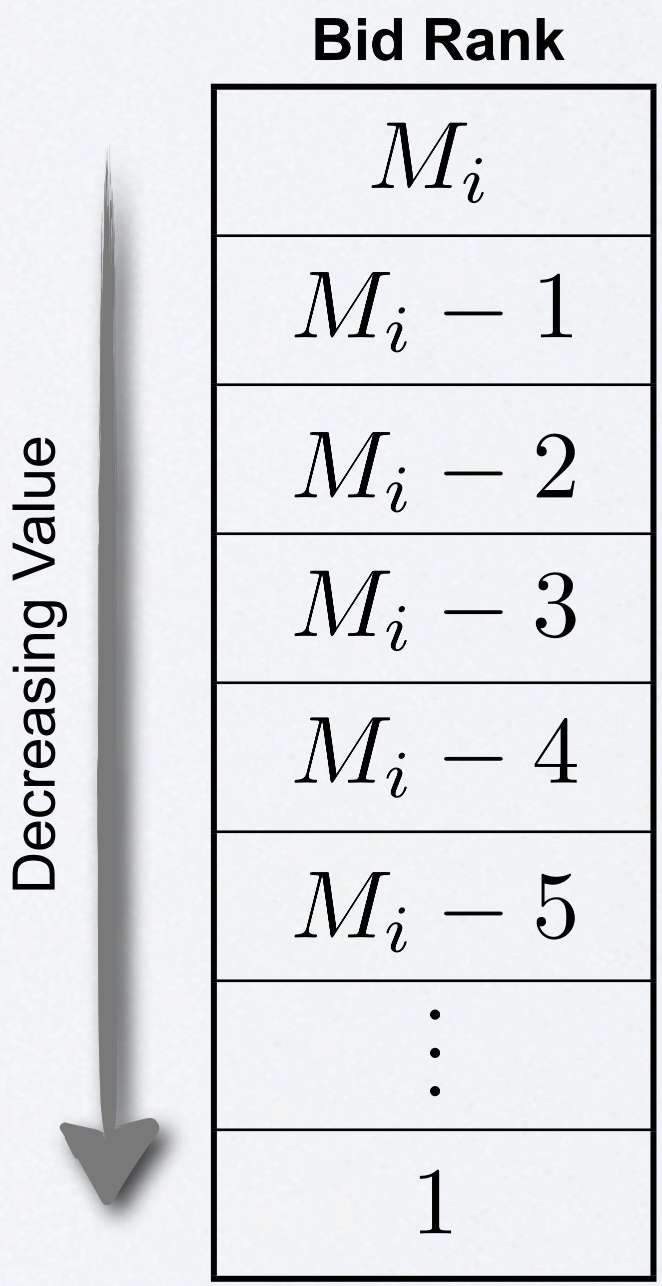
$$\mathbb{P}({V^i}_{(k)} \leq v_{ij}) = \int_0^{v_{ij}} \frac{m_{ji}!}{(k-1)!(m_{ji}-k)!} F_{V^i}(\omega)^{k-1}(1-F_{V^i}(\omega))^{m_{ji}-k} f_{V^i}(\omega) d\omega$$

- Find the ranks of the order statistics for which a peer wins a given number of auctions

$$k_0 = \max(1, M_i - N_i + 1)$$

$$k_1 = \min(k_0 + m_{ji} - 1, M_i)$$

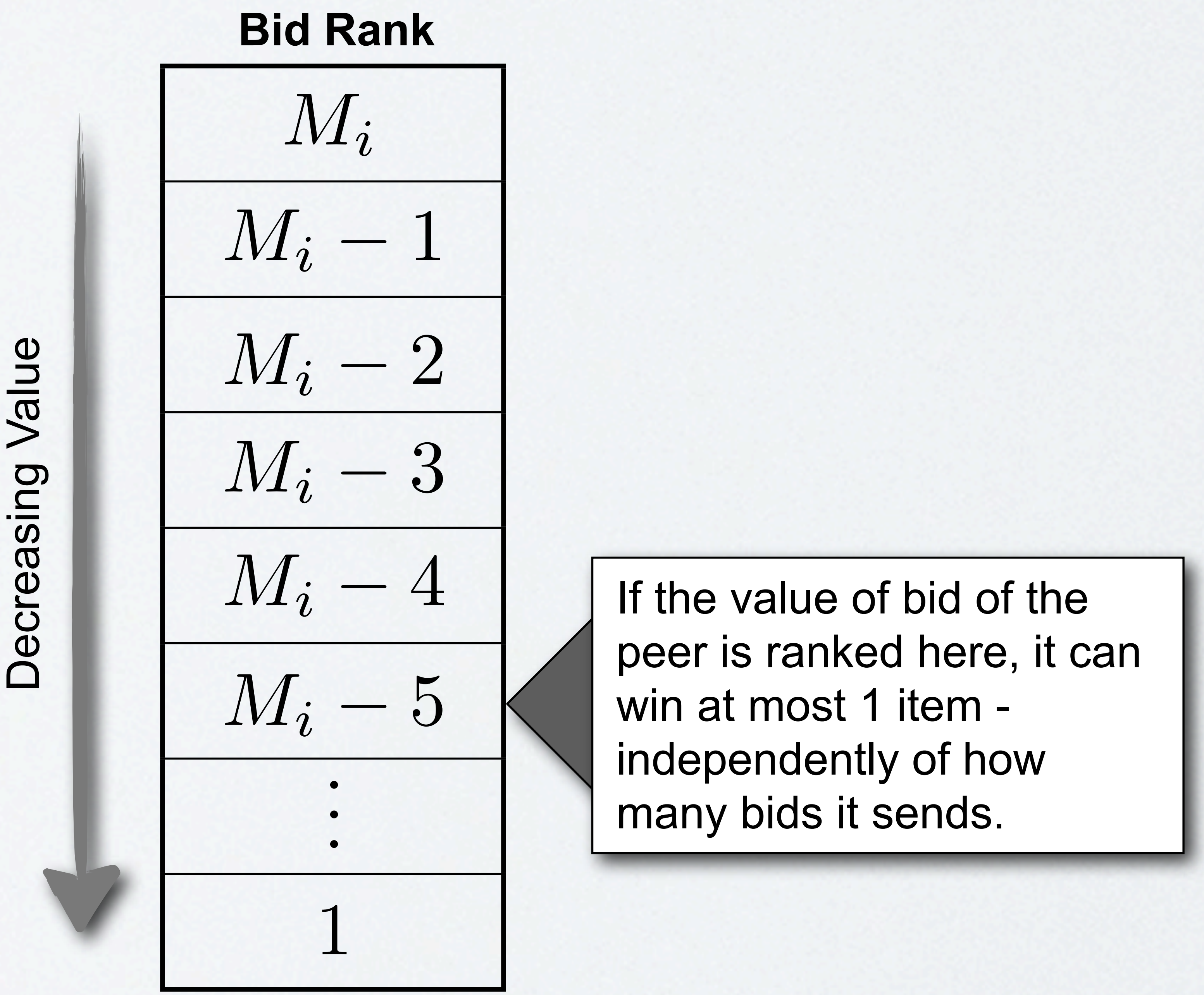
Estimating SHA Outcomes



- For instance, assume that there are 6 items for sale ($N_i = 6$). Then:

$$k_0 = M_i - 5$$

Estimating SHA Outcomes

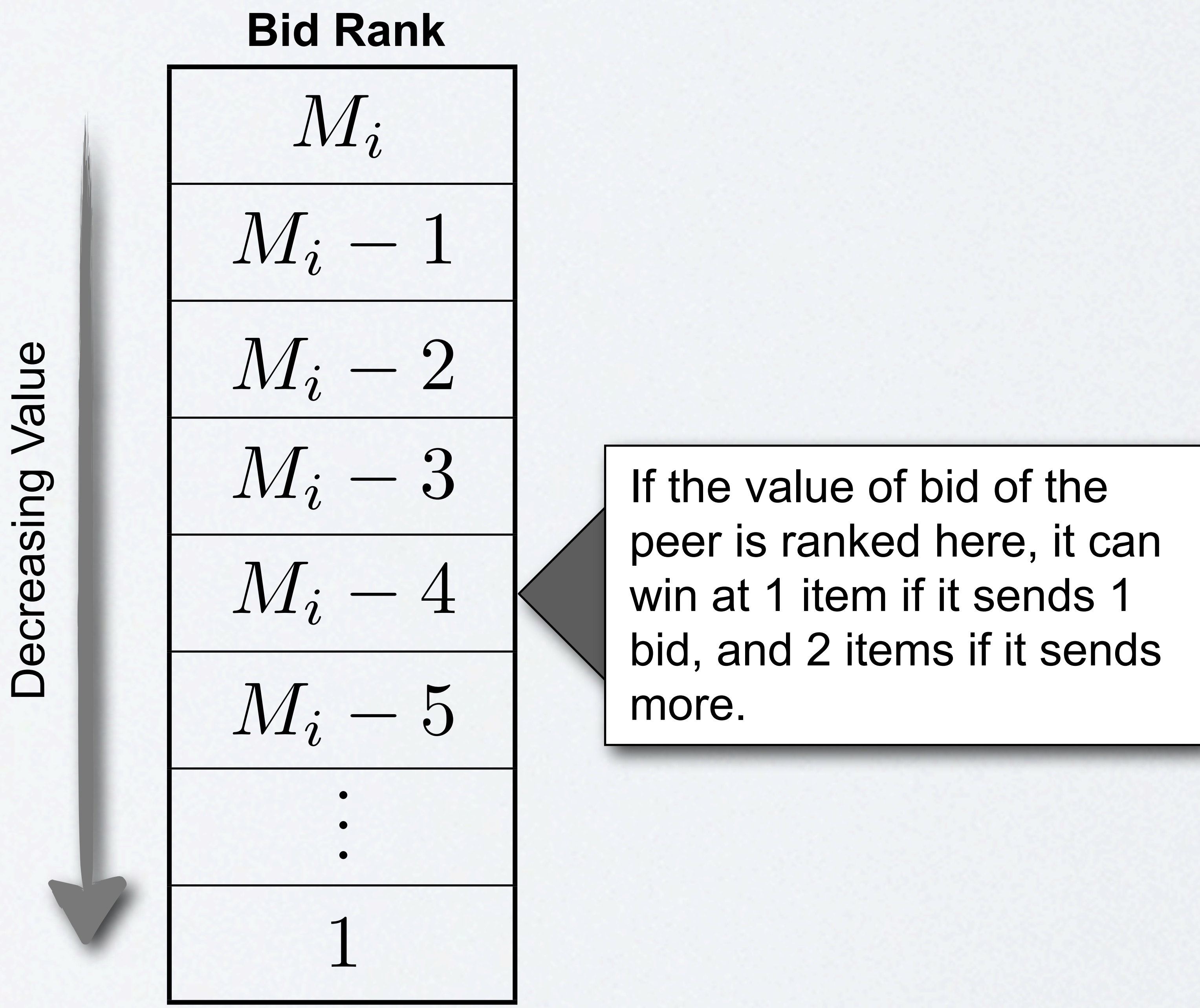


- For instance, assume that there are 6 items for sale ($N_i = 6$). Then:

$$k_0 = M_i - 5$$

- k_1 will depend on how many bids the peer sends.

Estimating SHA Outcomes

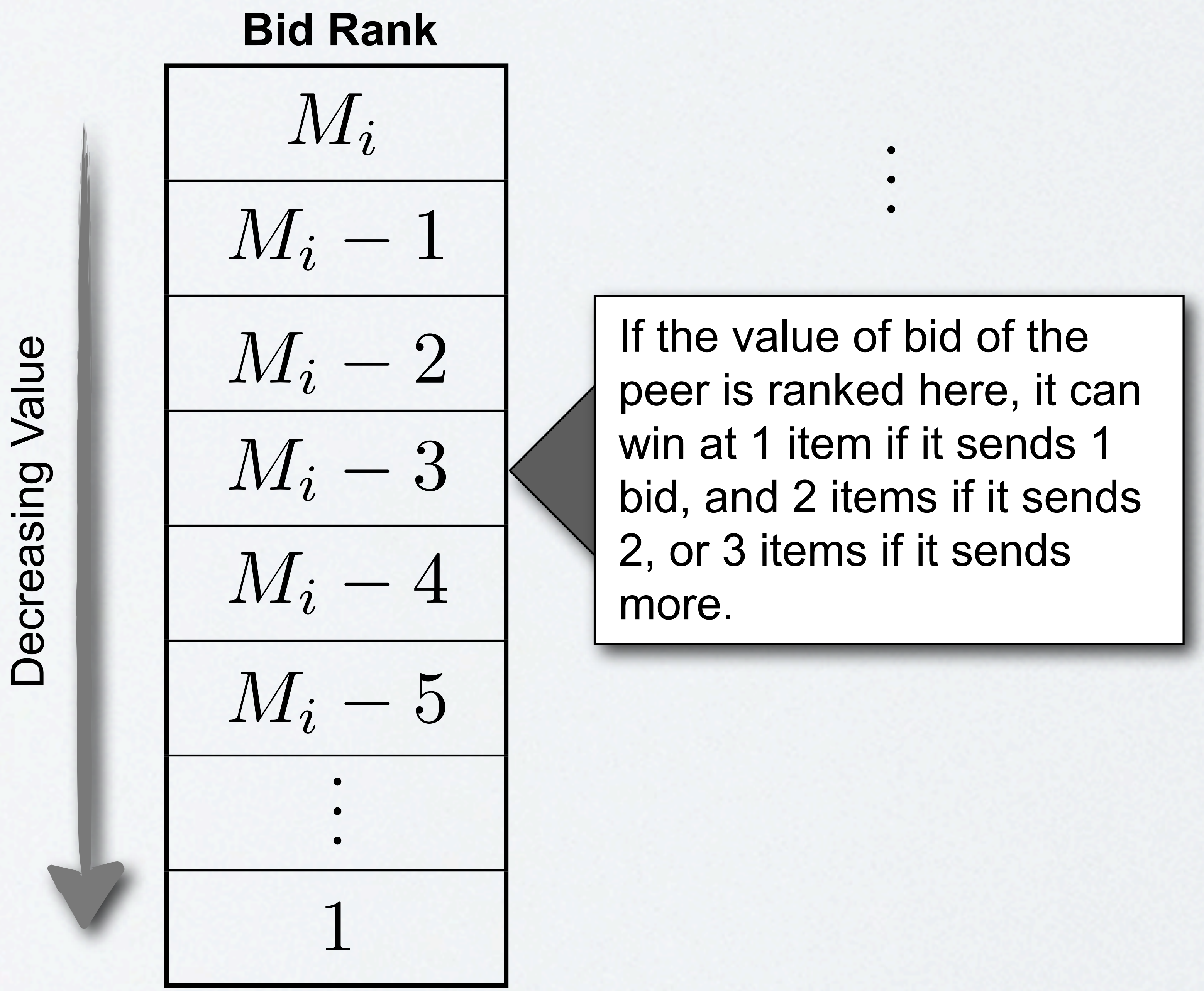


- For instance, assume that there are 6 items for sale ($N_i = 6$). Then:

  $$k_0 = M_i - 5$$

- k_1 will depend on how many bids the peer sends.

Estimating SHA Outcomes



- For instance, assume that there are 6 items for sale ($N_i = 6$). Then:

  $$k_0 = M_i - 5$$

- k_1 will depend on how many bids the peer sends.

Estimating SHA Outcomes

- We can now calculate the expected number of items won, given that N_i are bid for at a value of v_{ij}.

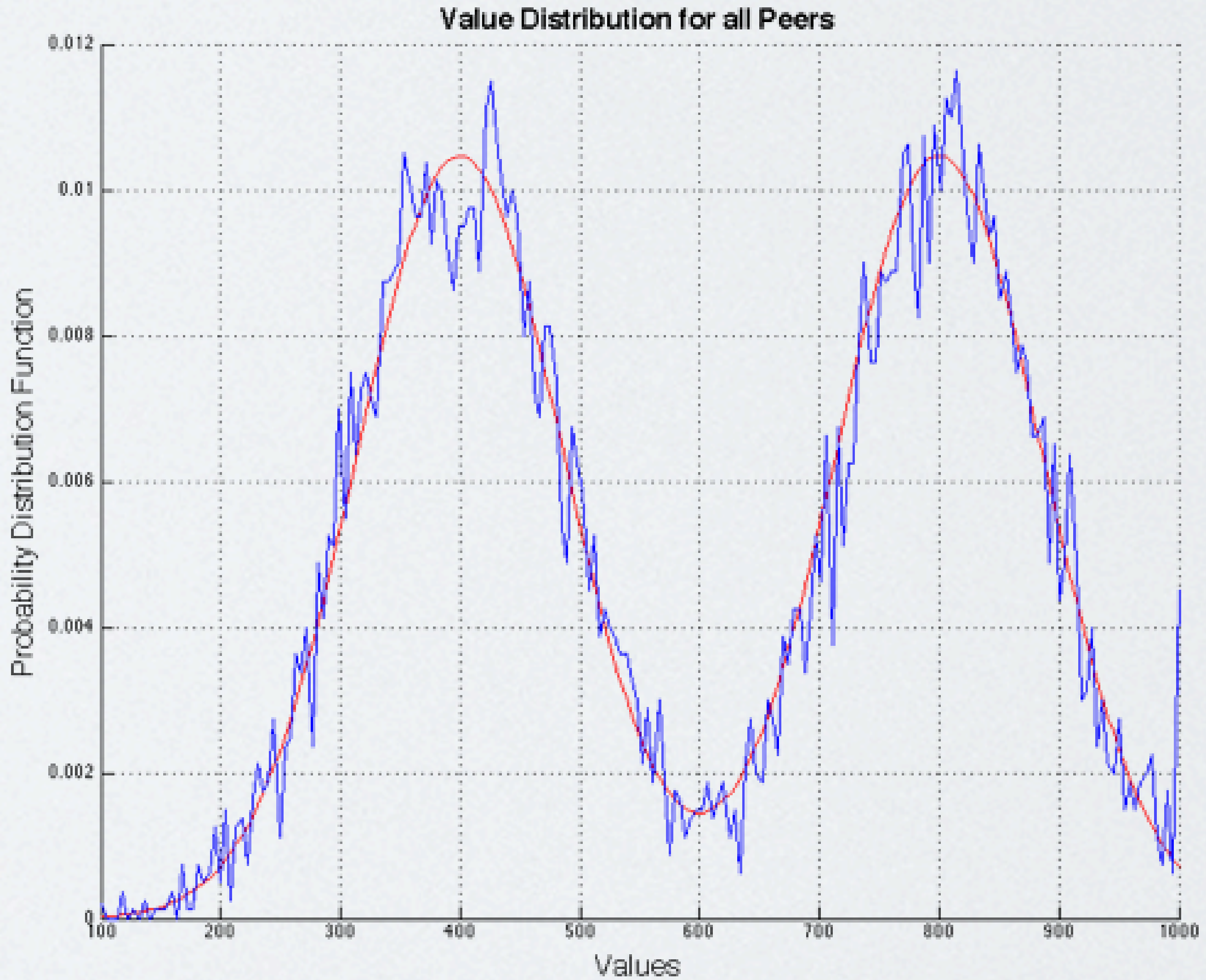
$$\mu(v_{ij}, N_i, m_{ji}) = \sum_{k=k_0(v_{ij},N_i)}^{k_1(v_{ij},N_i)} I(F_{V^i}(v_{ij}); k, m_{ji} - k + 1)$$

- Where $I(x; k, n)$ is the *regularized, incomplete beta function*.

- The standard deviation can be found equivalently.

Simulation

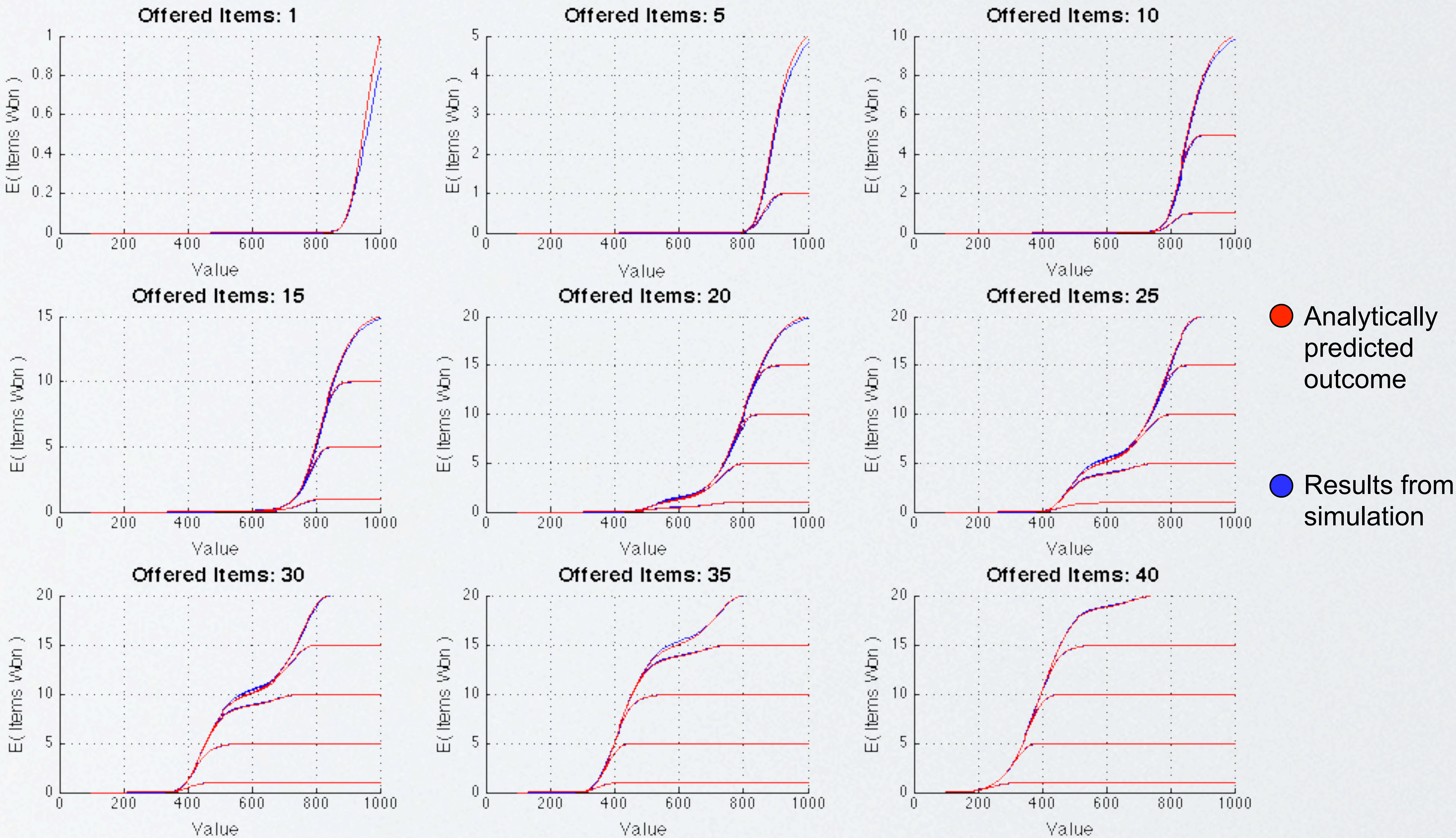
- To test the analytic results presented before, we use an artificial bimodal value distribution as shown
- Bids are generated according to this distribution, and the expected number of items won is recorded for discrete values of v_{ji}.



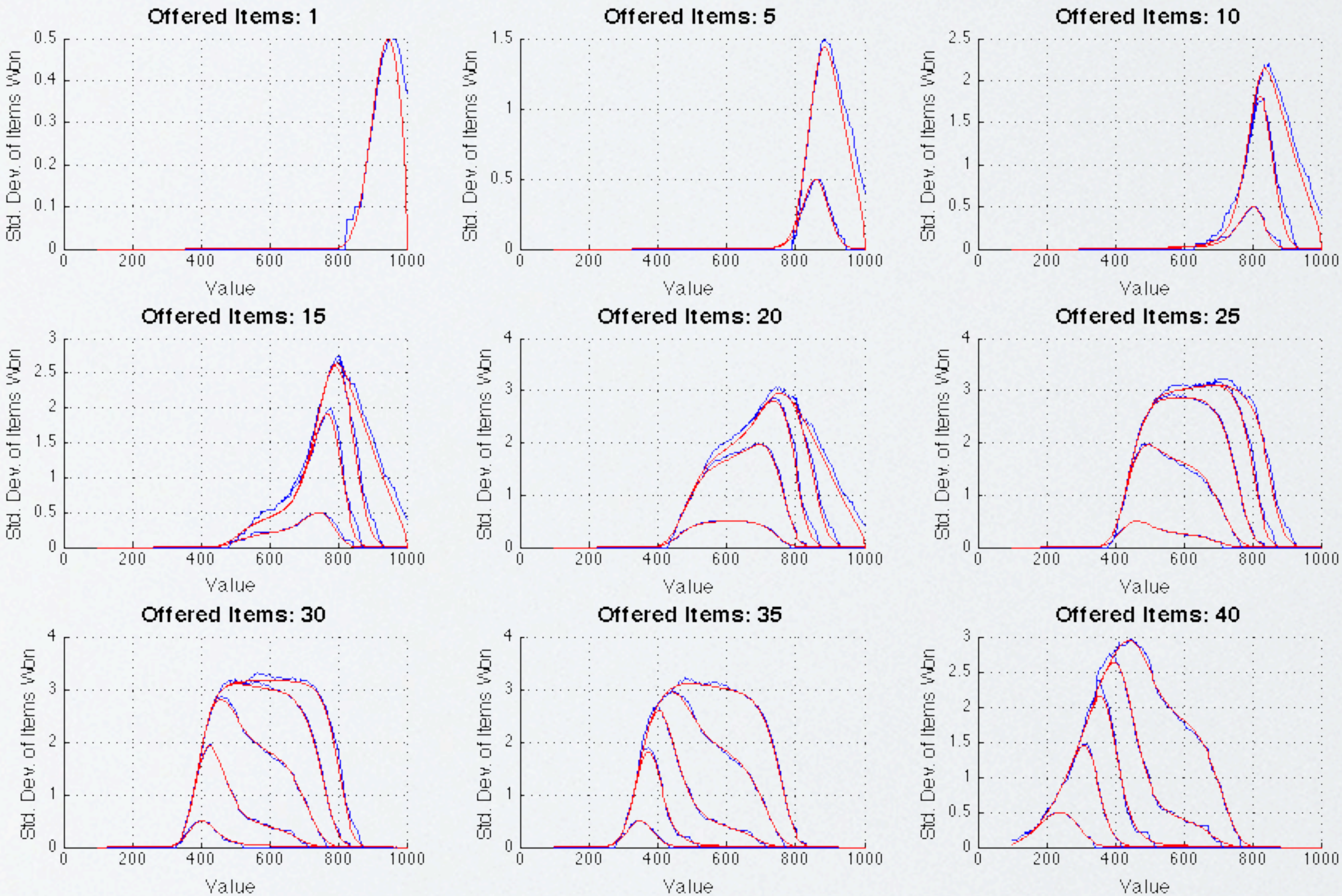
Analytically predicted outcome

Results from simulation

Estimating the number of bids won



Estimating the variability in bids won



Analytically predicted outcome

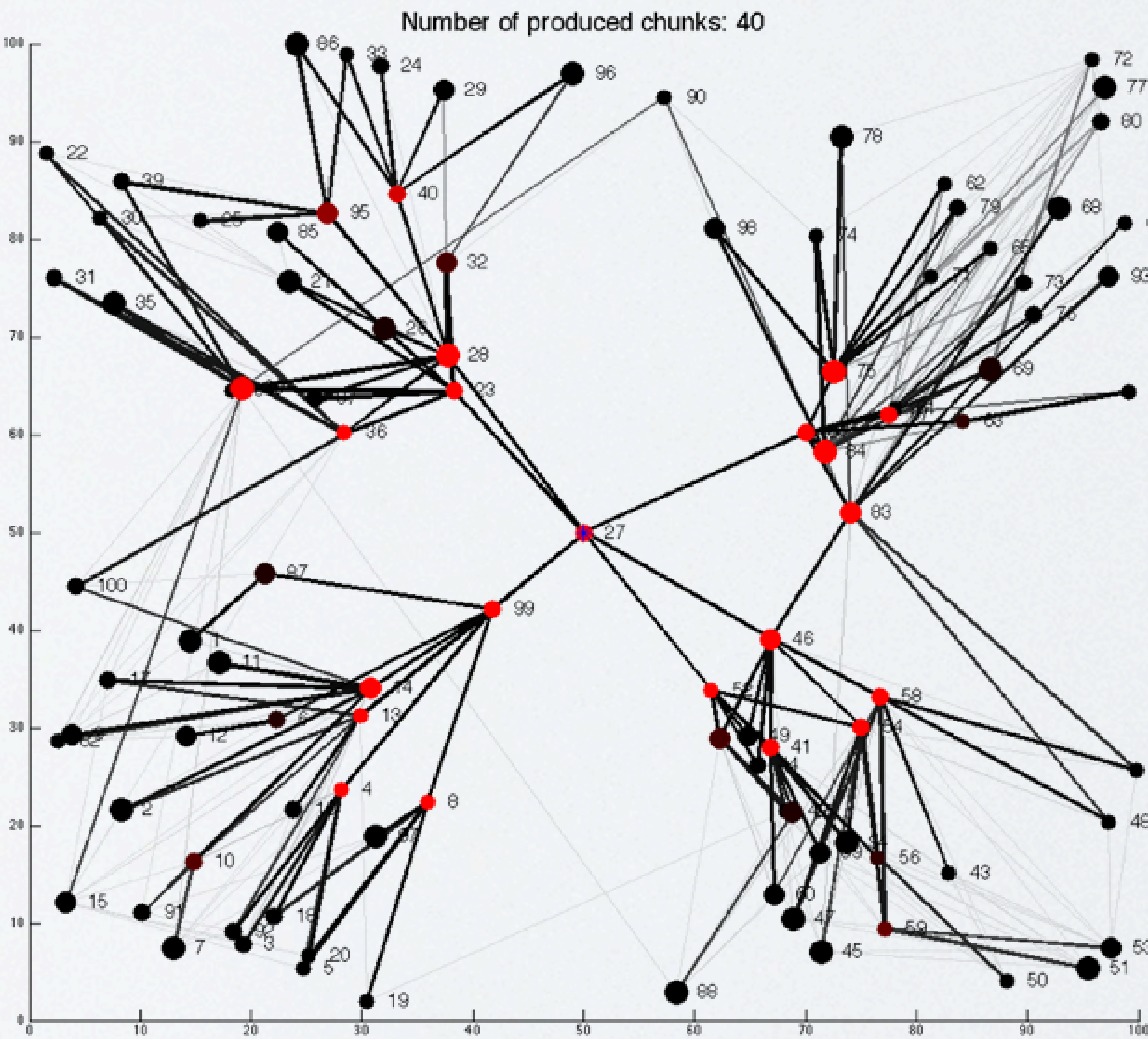
Results from simulation

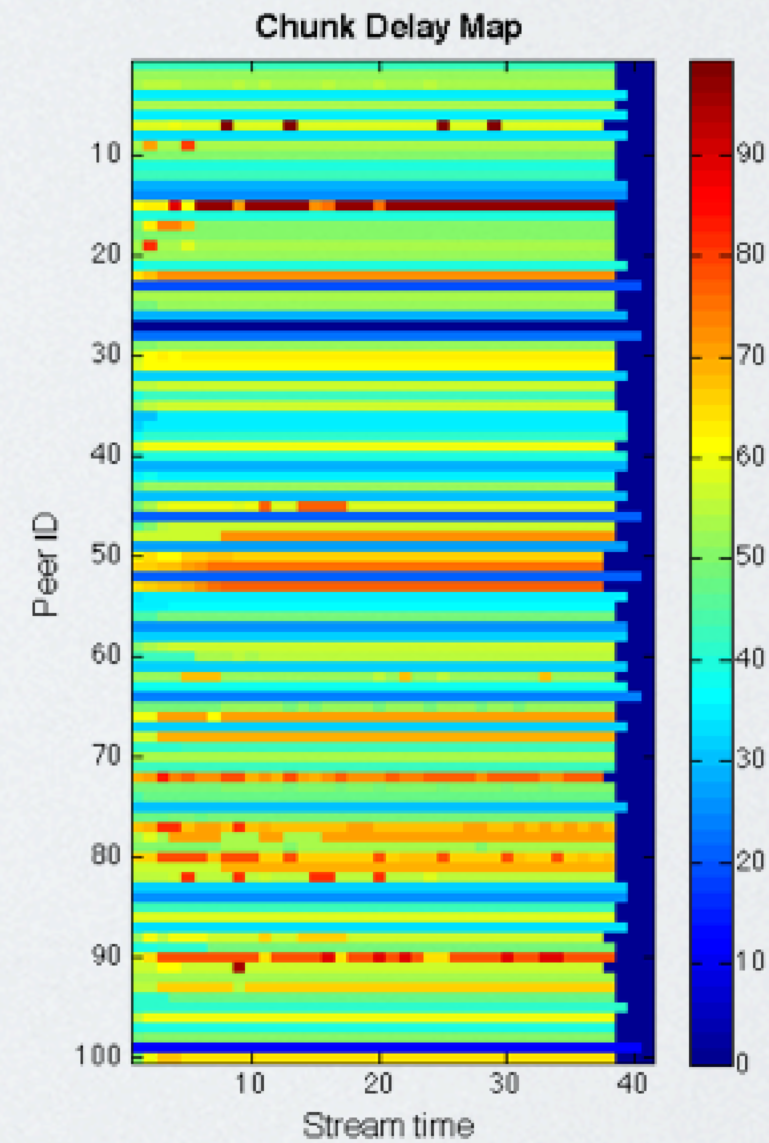
SHA and P2P Live Streaming Overlays

- We use auctions to perform capacity/delay tradeoffs in P2P streaming overlays.

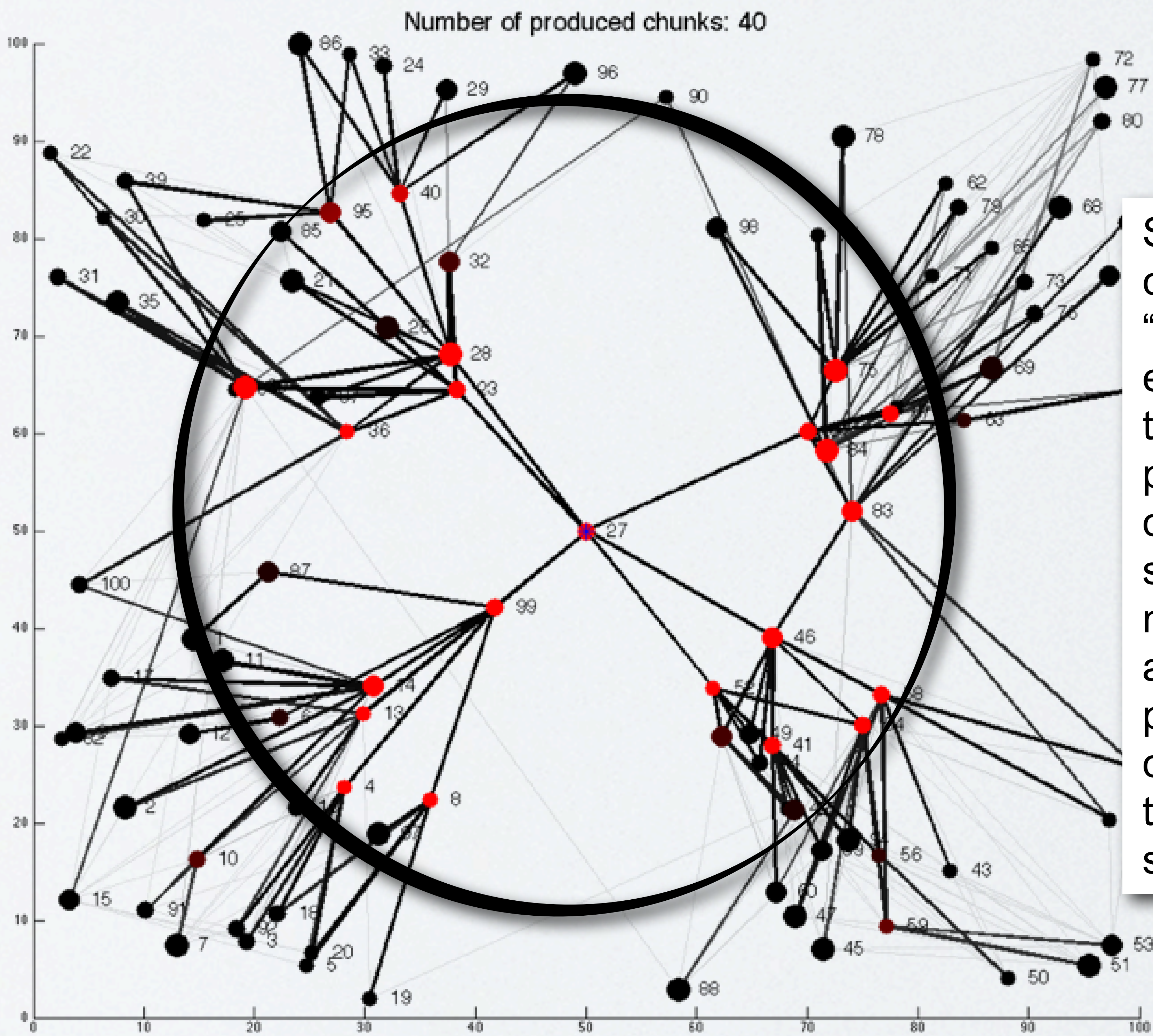
- For each chunk they are interested in,
  - Peers send a bid to the peer that gives them the greatest value for that chunk;
  - If the expected number of times the chunk will be received is smaller than 1, the process is repeated.
  - The process takes peer capacity, delay and load into account.

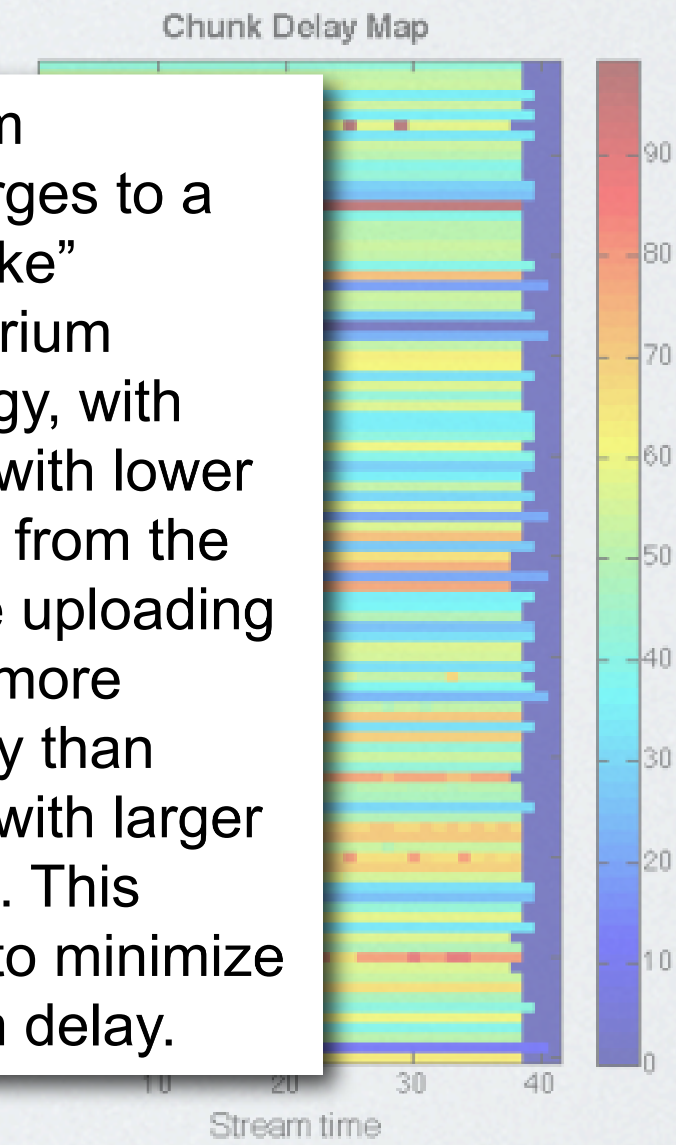
De-centralised Overlay Topology Construction





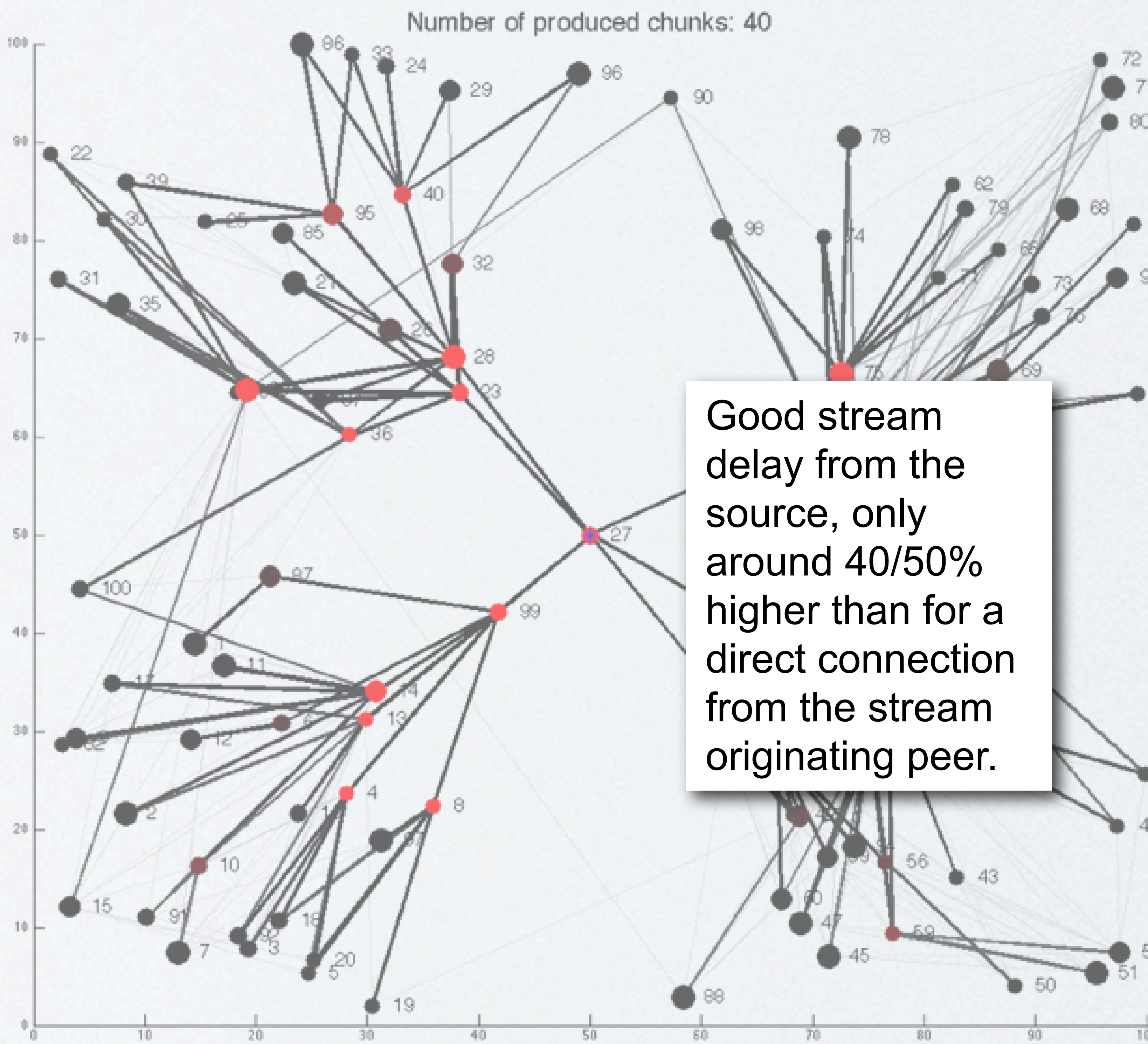
De-centralised Overlay Topology Construction



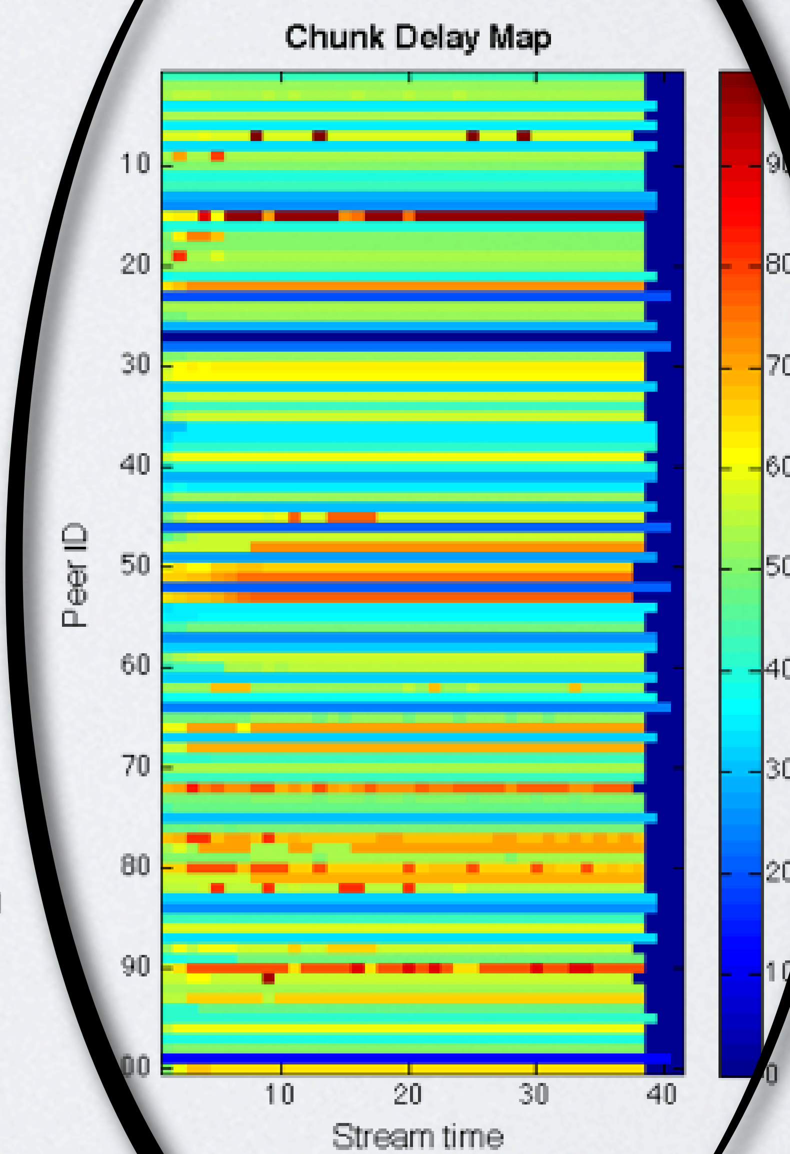


System converges to a “tree-like” equilibrium topology, with peers with lower delays from the source uploading much more actively than peers with larger delays. This tends to minimize stream delay.

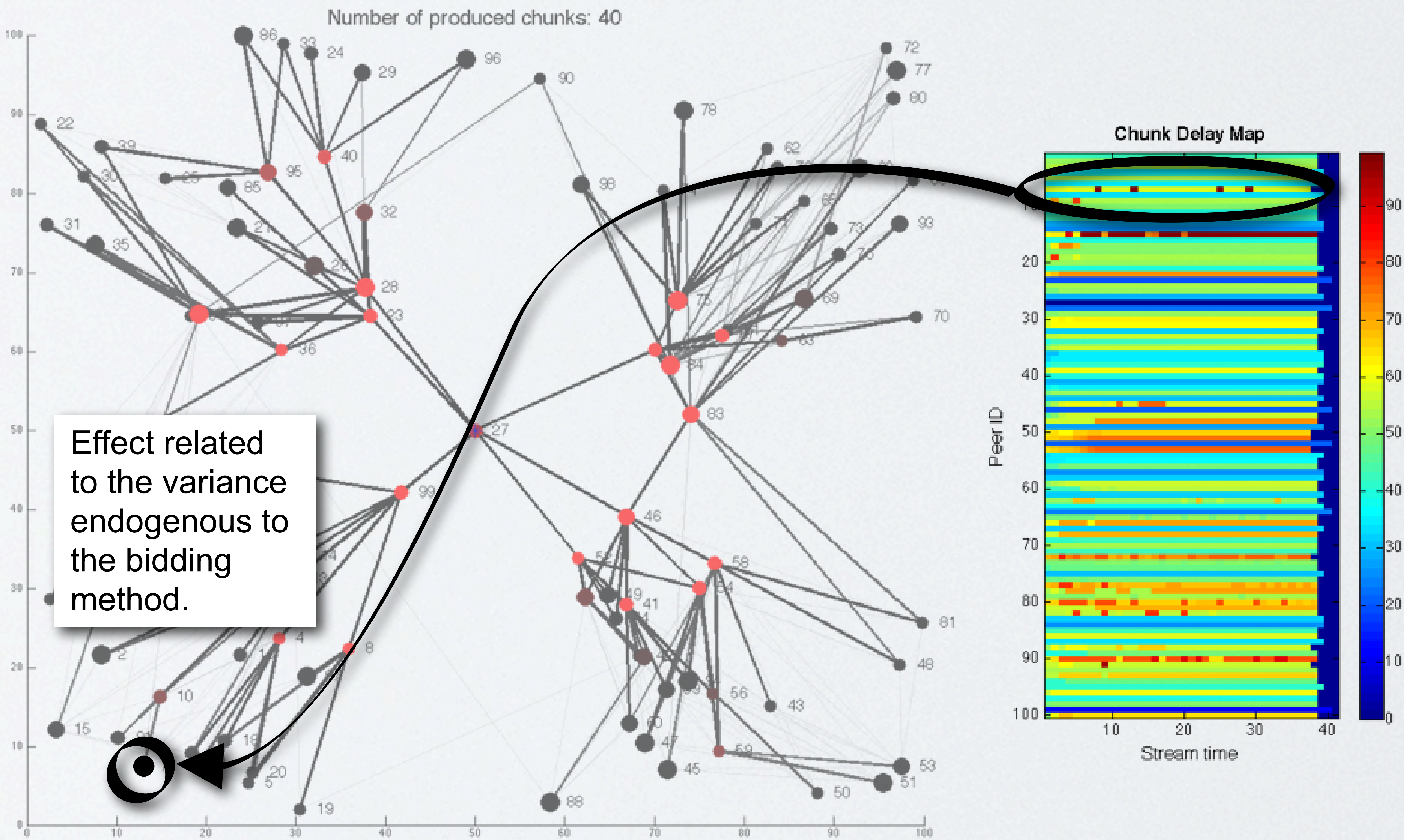
De-centralised Overlay Topology Construction



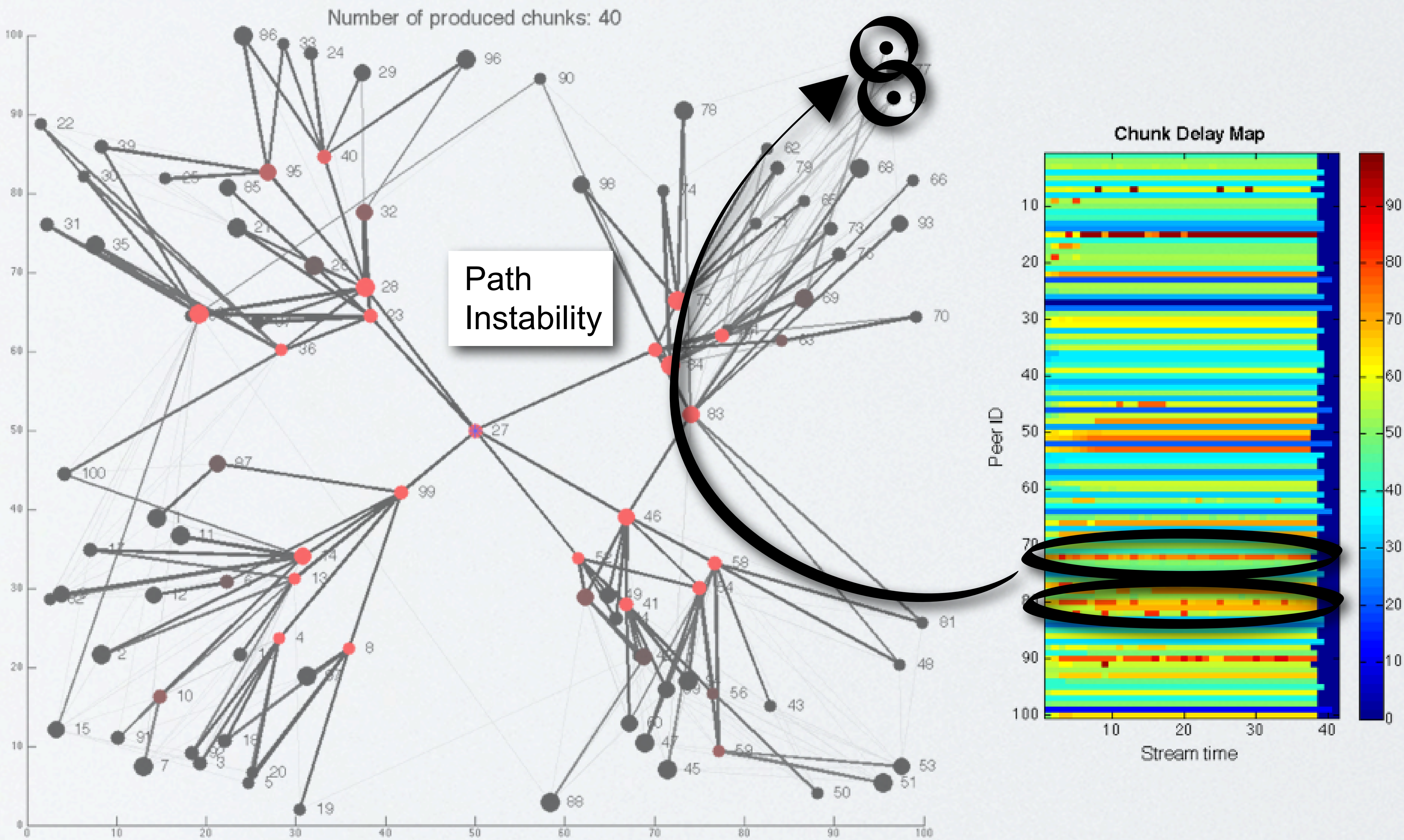
Good stream delay from the source, only around 40/50% higher than for a direct connection from the stream originating peer.



Future Work: Lost Chunks ("Slips")



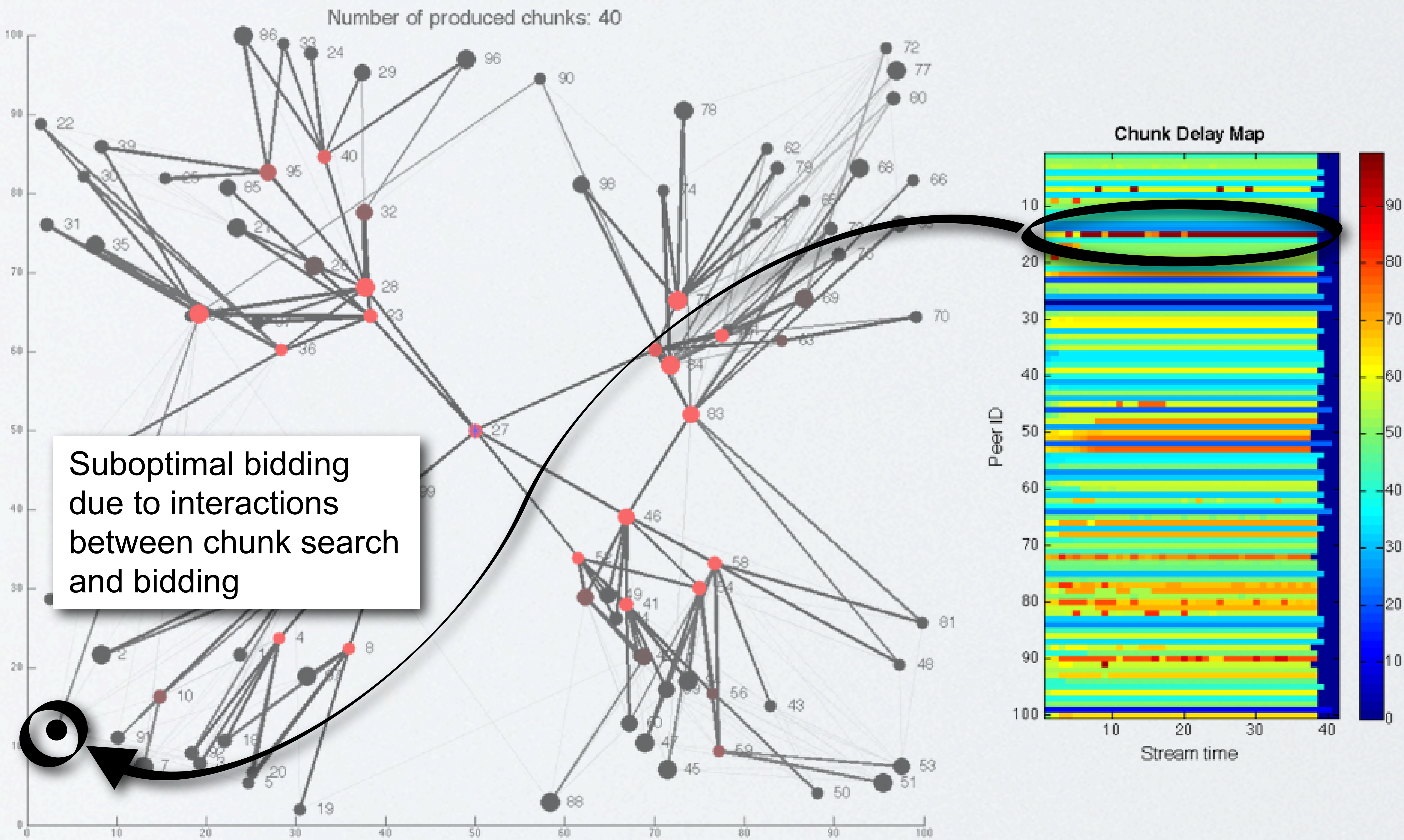
Future Work: High Chunk Jitter



Thank You!

Questions?

High Chunk Delay



Suboptimal bidding due to interactions between chunk search and bidding