



Insert title here...

Bandwidth-sharing in LHCONE

an analysis of the problem

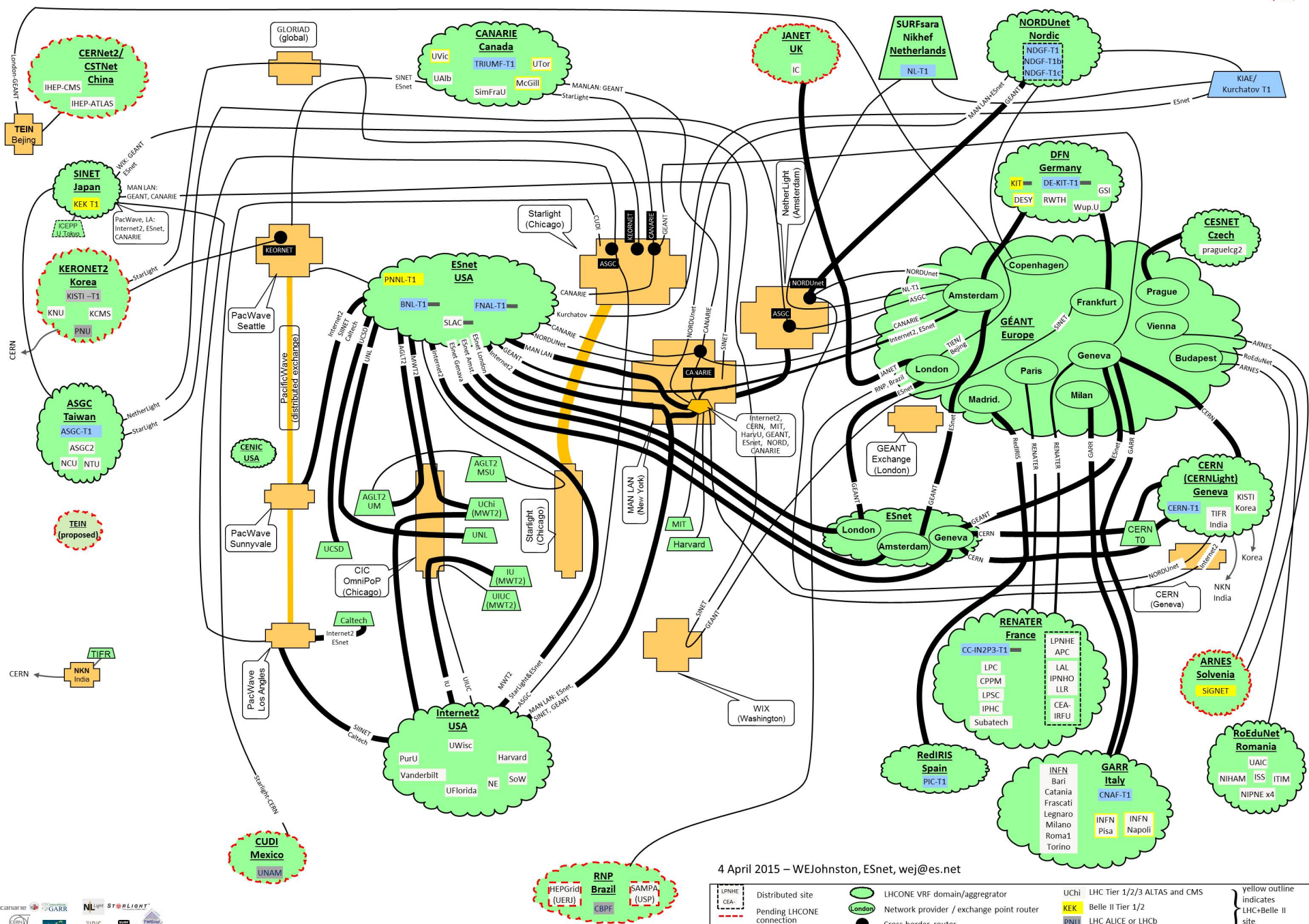
Run-2 vs. Run-1

- Relative reduction in CPU, disk, people
 - Efficiency is becoming more important
- Network now seen as more reliable
 - Stable, performant, well-provisioned
 - Now used as if free and infinite
- Computing models: more relaxed data-placement
 - T2 -> T2 transfers
 - Xrootd data federations for WAN access (CMS: AAA)
 - Significantly more fluid than originally planned

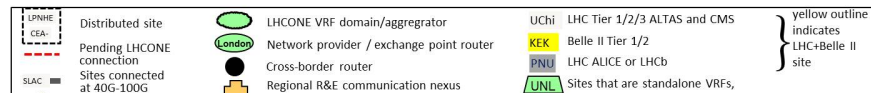
Scheduling the network

- Data-transfer needs are growing
 - Cannot assume the network will remain free and infinite forever
- Computing models are evolving
 - Increasingly close interaction with h/w resources
 - Broader range of resource-types (opportunistic, cloud, volunteer) with different storage & I/O characteristics
- Can see the value of scheduling network use
 - Just-in-time data-placement, deadline management
 - Possibilities that come from deterministic behaviour, not necessarily just more speed

LHCONE: A global infrastructure for the High Energy Physics (LHC and Belle II) data management



4 April 2015 – WEJohnston, ESnet, wej@es.net



Schedule, but how?

- How to allocate bandwidth fairly & efficiently to users with different & time-dependent needs?
- Candidate technologies, won't discuss here
 - Virtual circuits, multi-path flows
 - Bandwidth guarantees can be hard or soft
 - Networking groups are making progress towards a technical solution
- Then you get a new problem: oversubscription
 - Common to all successful middleware:
 - phase 1: make it possible, phase 2: stop users abusing it!

A good bandwidth-sharing system?

- Automatic, lightweight
 - Set up ‘circuits’ automatically, but only where needed
 - Participation not mandatory (casual or low-load users)
- Elastic, responsive
 - Shares can grow and shrink over time
 - Change on timescale of ~ 1 hour to follow needs
- Efficient, fair
 - Allows maximal use of bandwidth at all times
 - Short-term & long-term: no starvation, no hogging
- ~~Fixed quotas~~

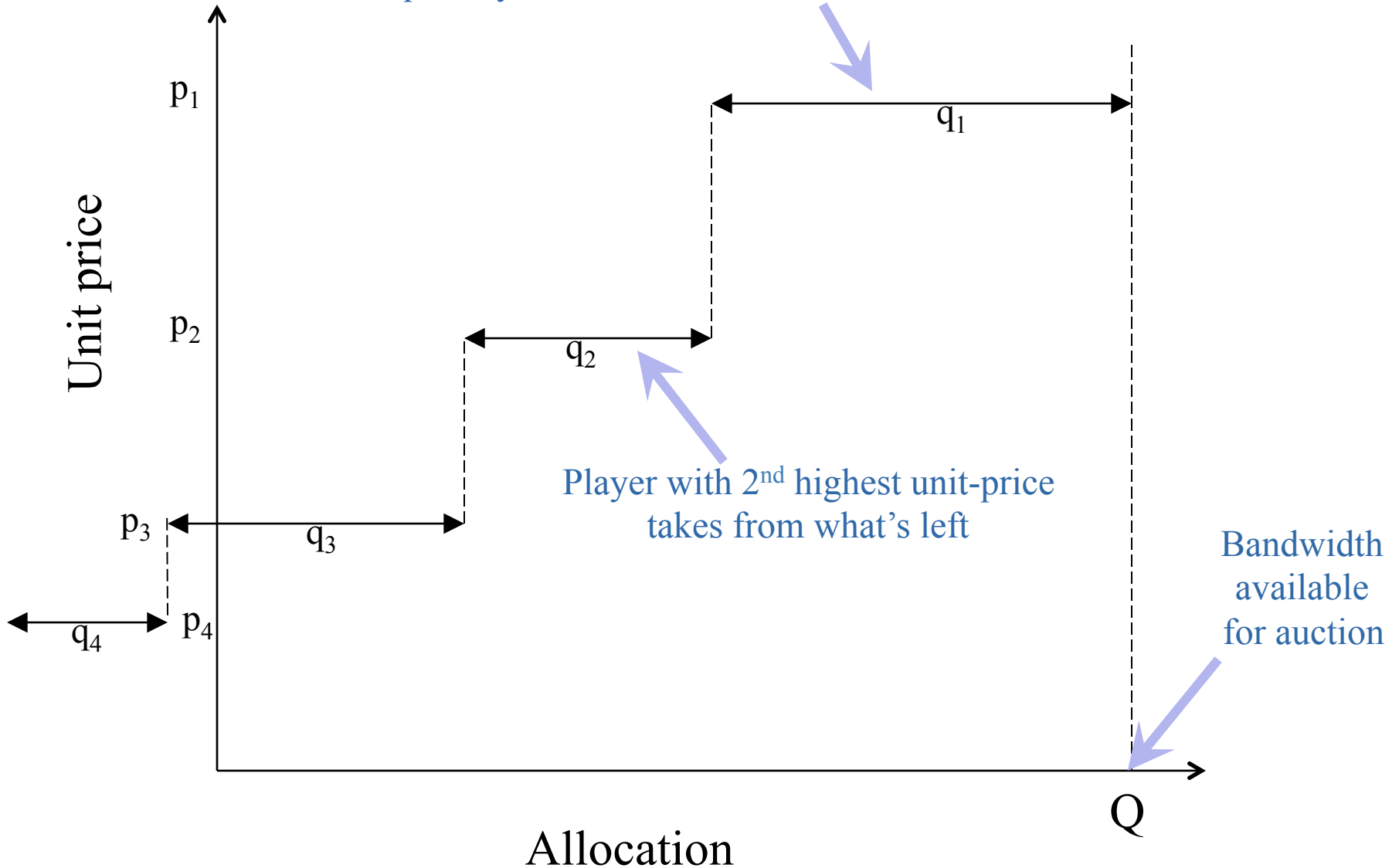
High-level requirements: we want...

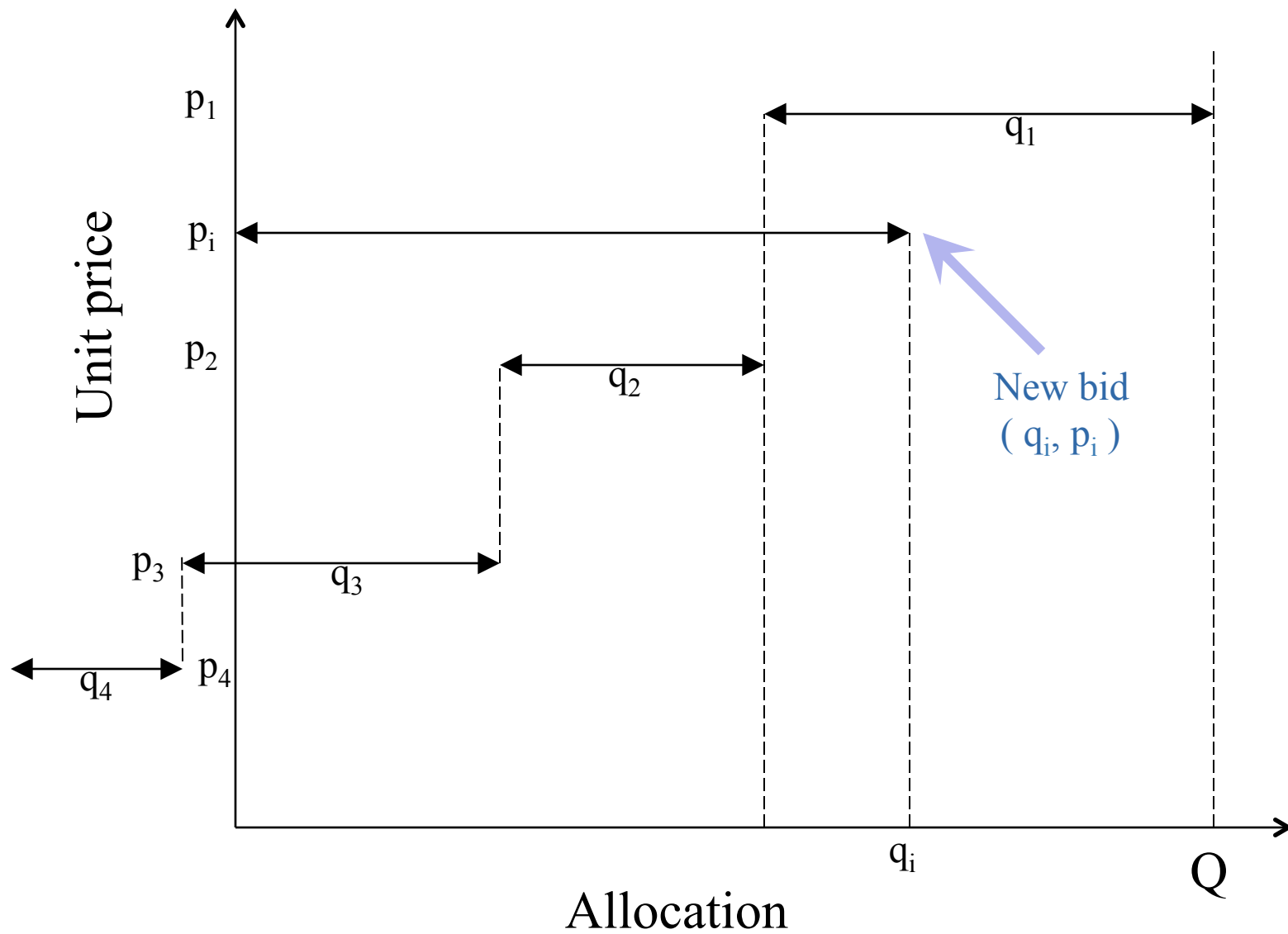
- A way for users to tell LHCONE their needs
 - At any time, across the whole of the LHCONE network
- To resolve over-subscription, quickly & fairly
- Technology to implement the shares
 - (out of scope of this talk)
- Candidate solution, the *Progressive Second-price Auction* (PSP)

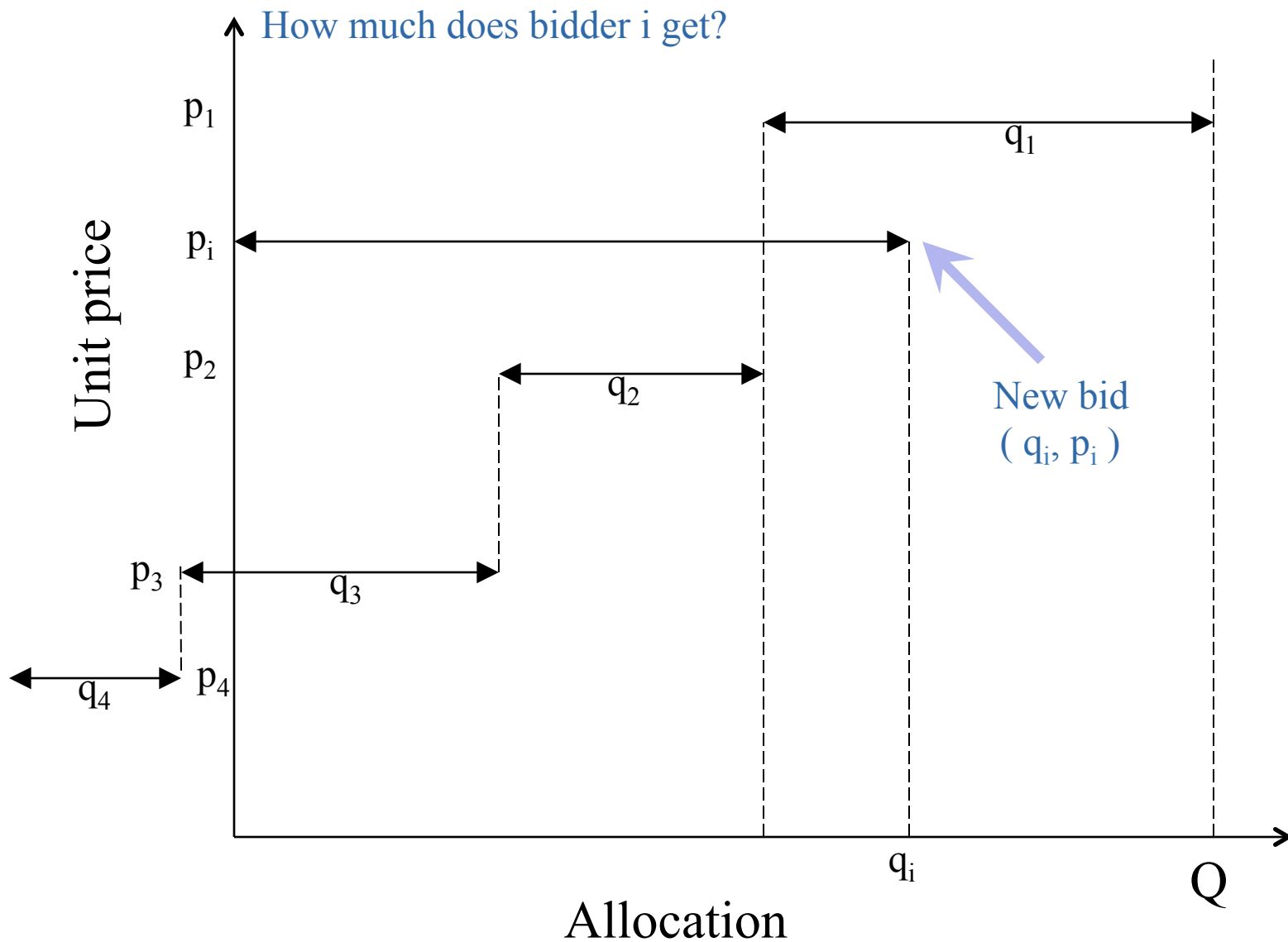
How does it work?

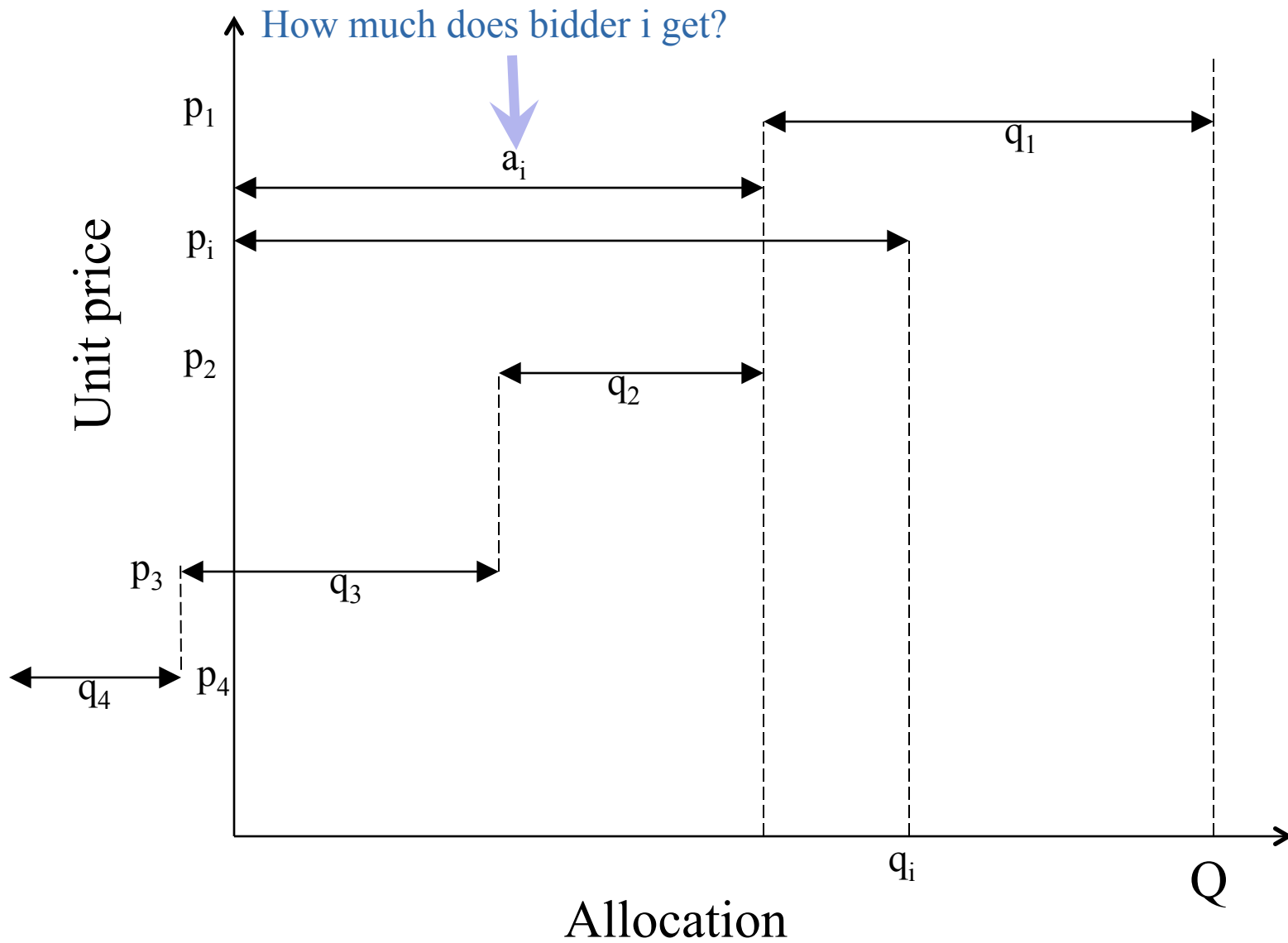
1. Network offers bandwidth Q on a given link
 2. Bidders have fixed budget (varies per bidder)
 3. Bidders specify quantity & unit-price: (q_i, p_i)
 4. PSP calculates allocation & total cost: (a_i, c_i)
 5. PSP sends all allocations/costs to all bidders
 6. Bidders revise their bids, submit them again
 7. Repeat until 3-6 until convergence
- Convergence guaranteed for rational bidders

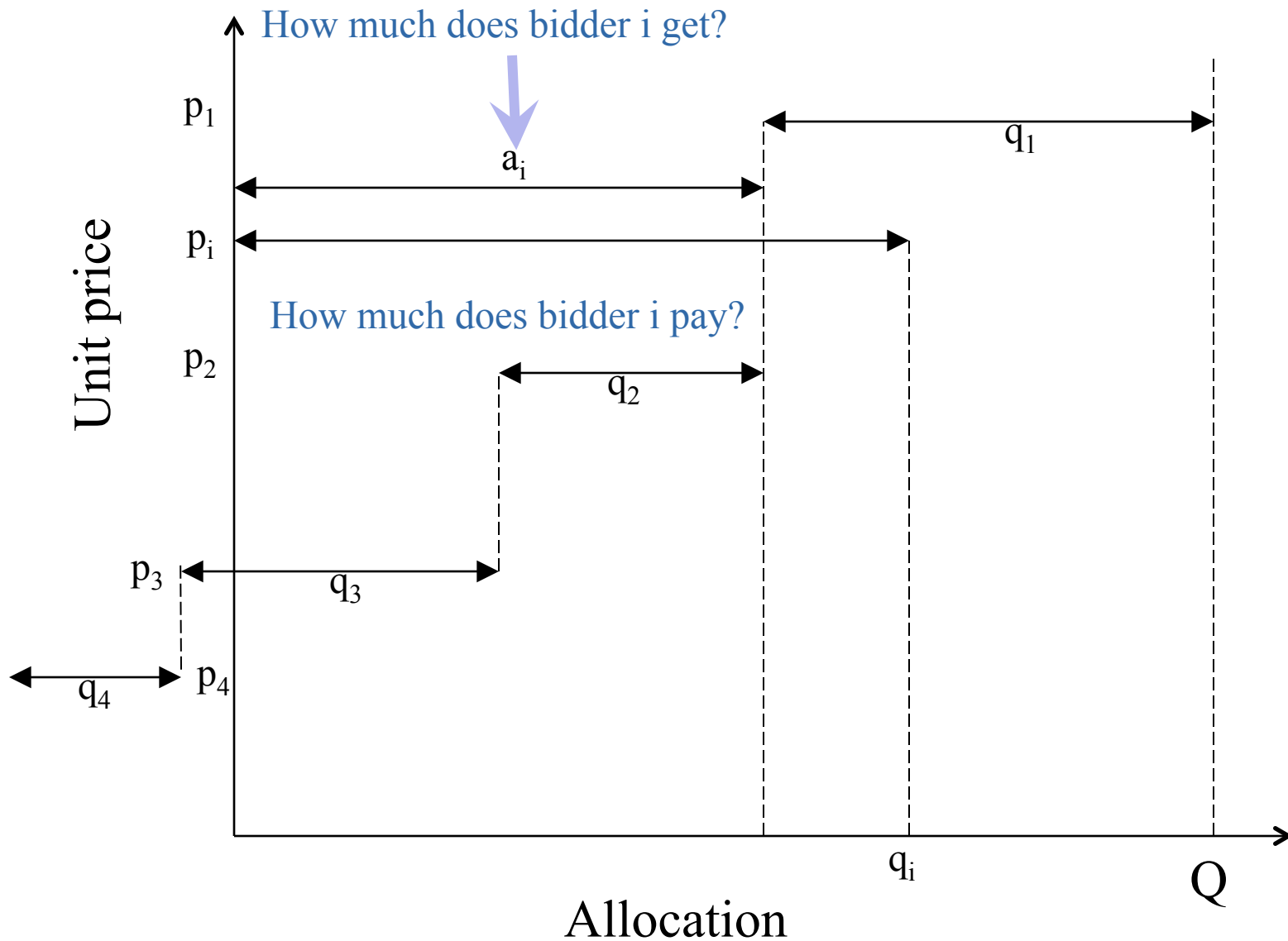
Player with highest unit-price has their
quantity subtracted from the total

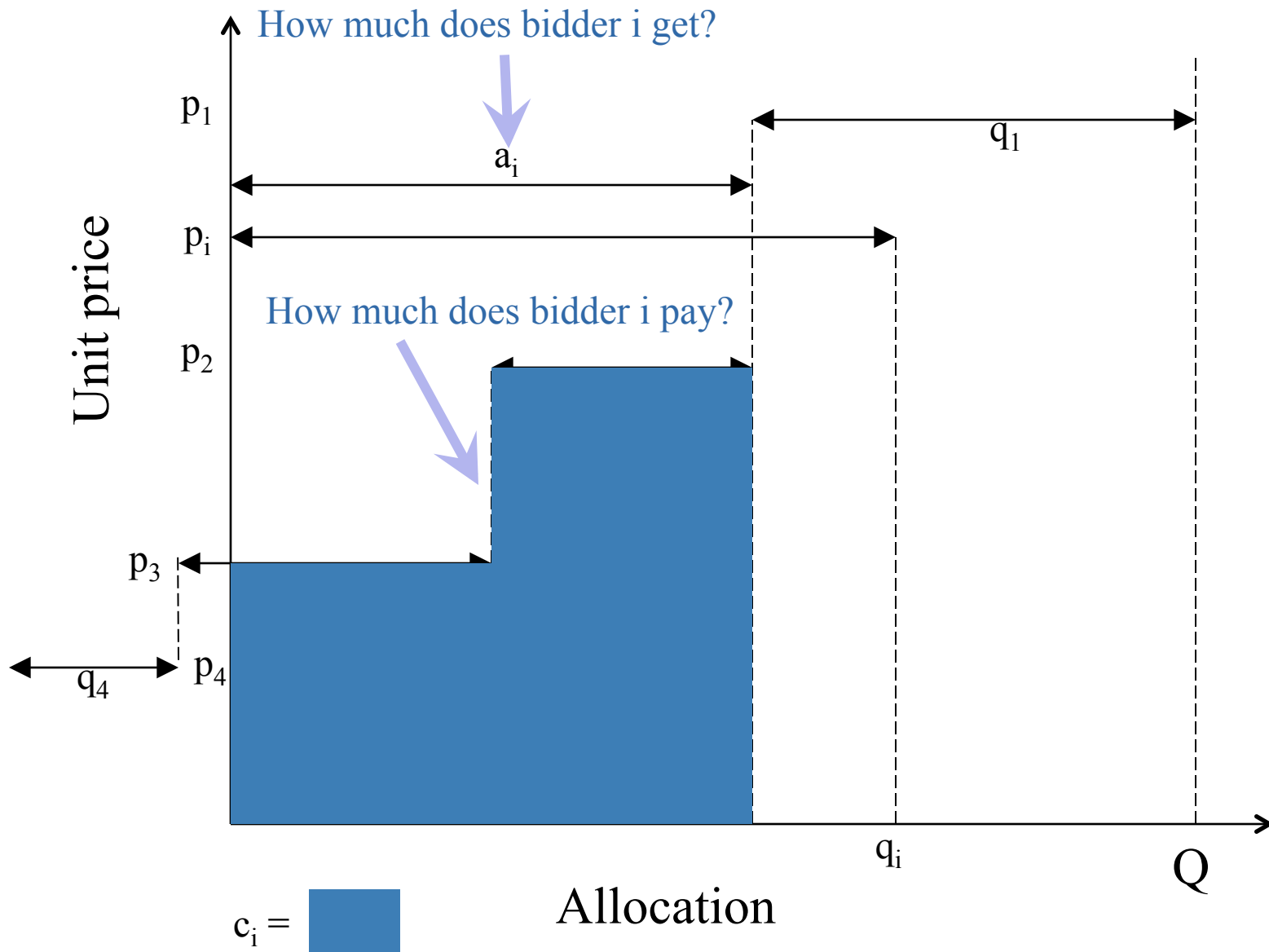












PSP on LHCONe?

- PSP extends naturally to multiple links
 - Decentralized, independent auction on each link
 - Bidders have fixed global budget
 - Best strategy is to bid for same bandwidth on each link
 - Bidder offers a price per link dependent on the competition for that link
 - Can show that it still converges if bidders are rational
- Repeat auction whenever conditions change
 - After some ‘lease-time’, to prevent chaos - ~ 1 hour?

Practicalities: budgets...

- Real world, real money
 - Budgets set by bidders
 - PSP guarantees the bidders converge on a solution
- LHCONe, HEP experiments, fake budget
 - Must ensure ‘fake money’ has real value in the auction
 - Must have just enough to express needs coherently
 - How to set the budgets?
 - Budget spent ~every time you win a slice of an auction, need to reset/adjust periodically, to keep the bidders solvent
 - Similar problem to allocating batch quota on shared farms?

How, and how often, to update budgets?

- Reset budget per-auction?
 - No incentive not to spend entire budget every time
 - Can lead to wasteful bidding, where not needed
- Carry-over of unspent budget?
 - Budget-hoarding => undesirable/‘unfair’ outcomes
- Excess budget
 - Blocking tactics, bidding for a link you don’t need
 - Some way to penalize for under-used circuits?
- Needs simulation, with various bidding strategies
 - Budget adjustment must not destroy auction fairness

Conclusion: principles

- Bandwidth-allocation at LHCONE requires a mechanism which is fair, efficient, lightweight, responsive and automatic
- The Progressive Second-Price auction offers this
 - Users negotiate among themselves how much bandwidth they should get
 - Repeat auction as needed, follow fluctuations automatically
 - Network providers get clear statement of what users want at any point in time
 - No negotiations between experiments & network providers

Conclusion: practicalities

- Fake budget complicates things
 - Setting initial budgets, refreshing budgets periodically
 - Similarities to batch quotas?
 - (can we just charge real money instead?)
- Bidding strategies
 - Coupled to how budgets are managed
 - Possible learning behaviour in repeat auctions
 - Need to understand how budget allocation interacts with bidding strategy to keep the auction truthful