Bandwidth-sharing in LHCONE

an analysis of the problem

Spoiler…

- Question:
  - Can we use market-mechanisms (auctions etc) to address bandwidth-sharing at LHCONE?
- Answer:
  - Yes, with a caveat: market-mechanisms work well with real money, much harder with fake ‘budgets’

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

- 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 & shrink 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)
- To avoid the possibility of ‘gaming the system’
- One option, the ***Progressive Second-price Auction***
  - A.Lazar, N.Semret, *Design, analysis and simulation of the progressive second price auction for network bandwidth sharing*

Second-price auctions

- Highest bid wins, price set by the 2nd-highest bid
  - Sounds odd, but is very common. E.g. Sothebys, Christies, the classic ‘English auction’
  - Two bidders raise in increments until one drops out, the other then wins, but price set by the one who stops bidding

- Key property: Best strategy is to bid the truth
  - Bid too high: may win, but at unacceptable cost
  - Bid too low: may lose at a price less than you are willing to pay
  - C.f. first-price auction: best strategy is to guess other bids, then bid between that and your true value
  - Useful when need to maximise ‘social utility’, not ‘profit’.

How does the PSP work?

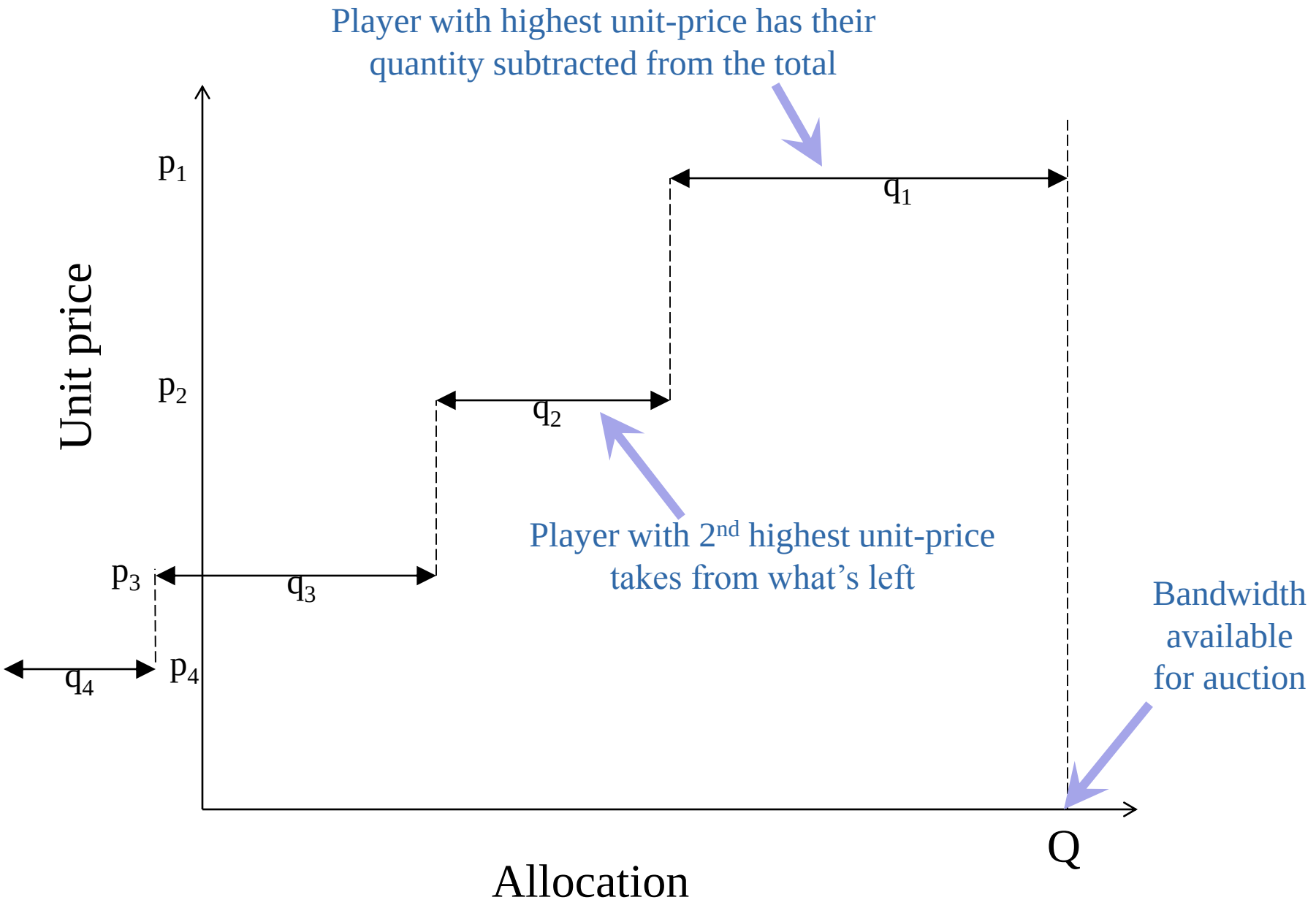
1. Network offers bandwidth **Q** on a given link
2. Bidders have a given budget (varies per bidder)
    3. Bidders specify quantity & unit-price: ($\mathbf{q_i}$,$\mathbf{p_i}$)
    4. PSP calculates allocations & total costs: ($\mathbf{a_i}$,$\mathbf{c_i}$)
    5. PSP sends all allocations/costs to all bidders
    6. Bidders revise their bids
7. Repeat until 3-6 until convergence

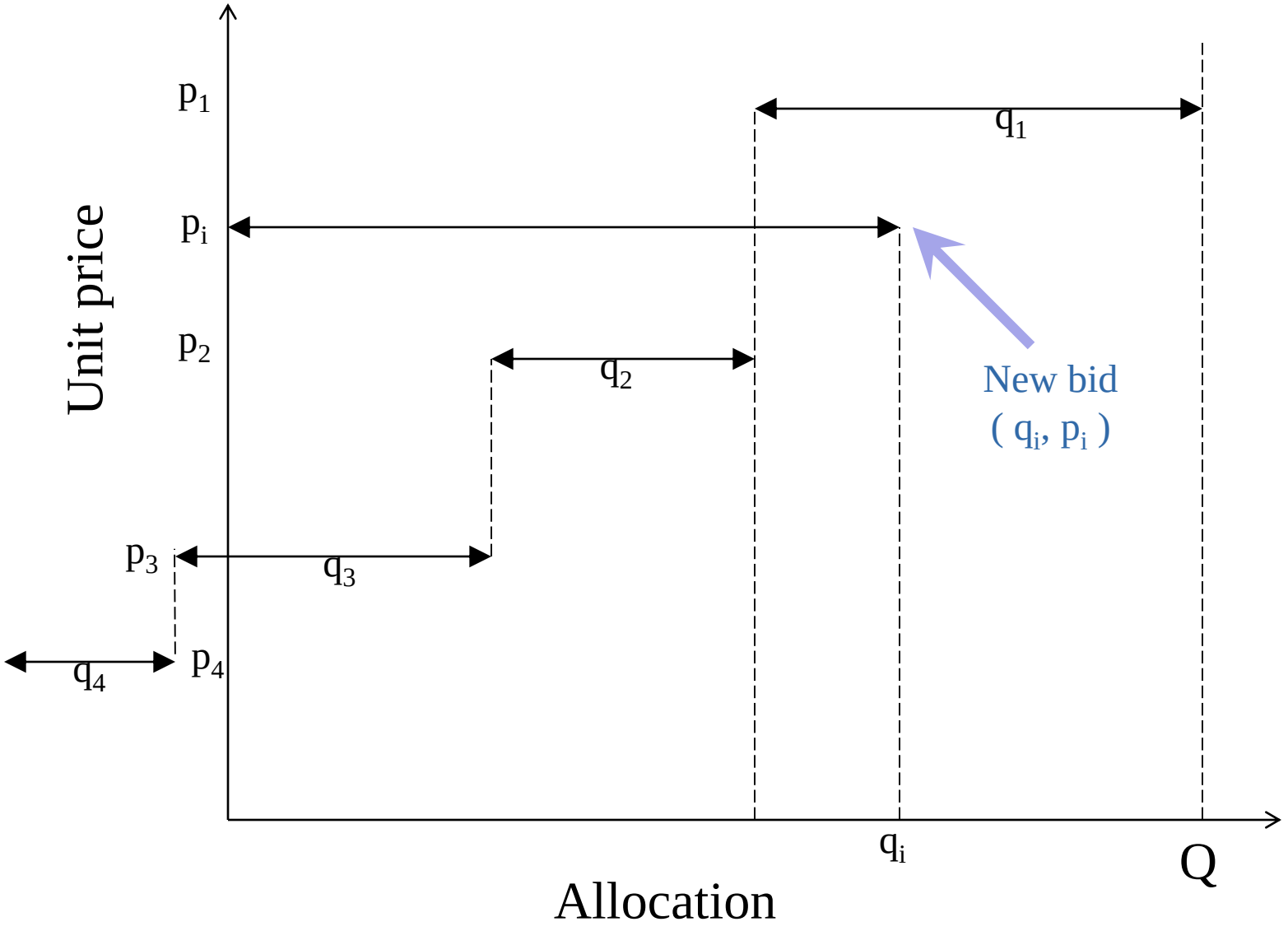
- Convergence guaranteed for rational bidders

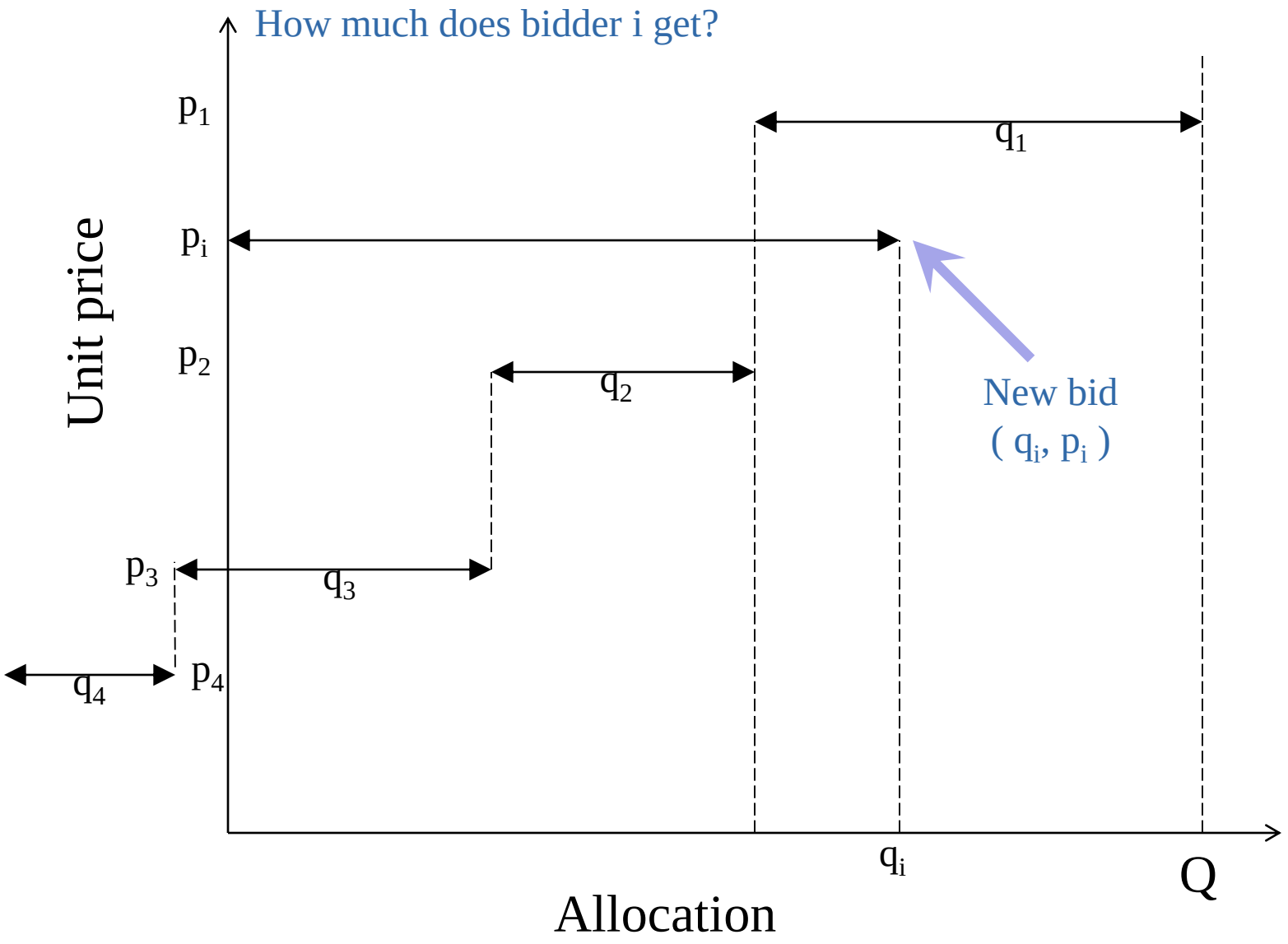
Rationality, cost…

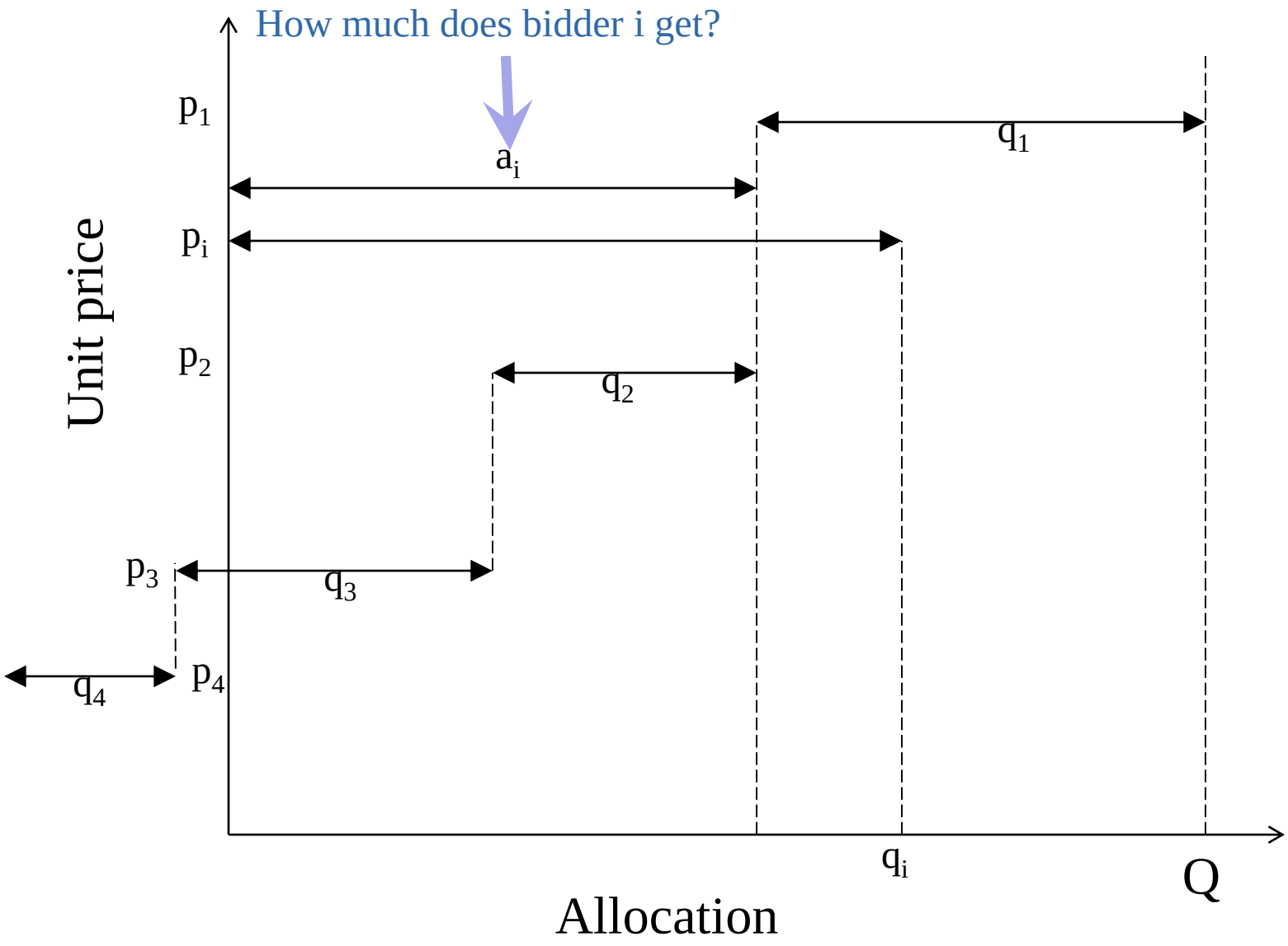
- Rational bidders
  - Economics: Law of Diminishing Marginal Returns
  - The more b/w you have, the less you are willing to pay to get even more

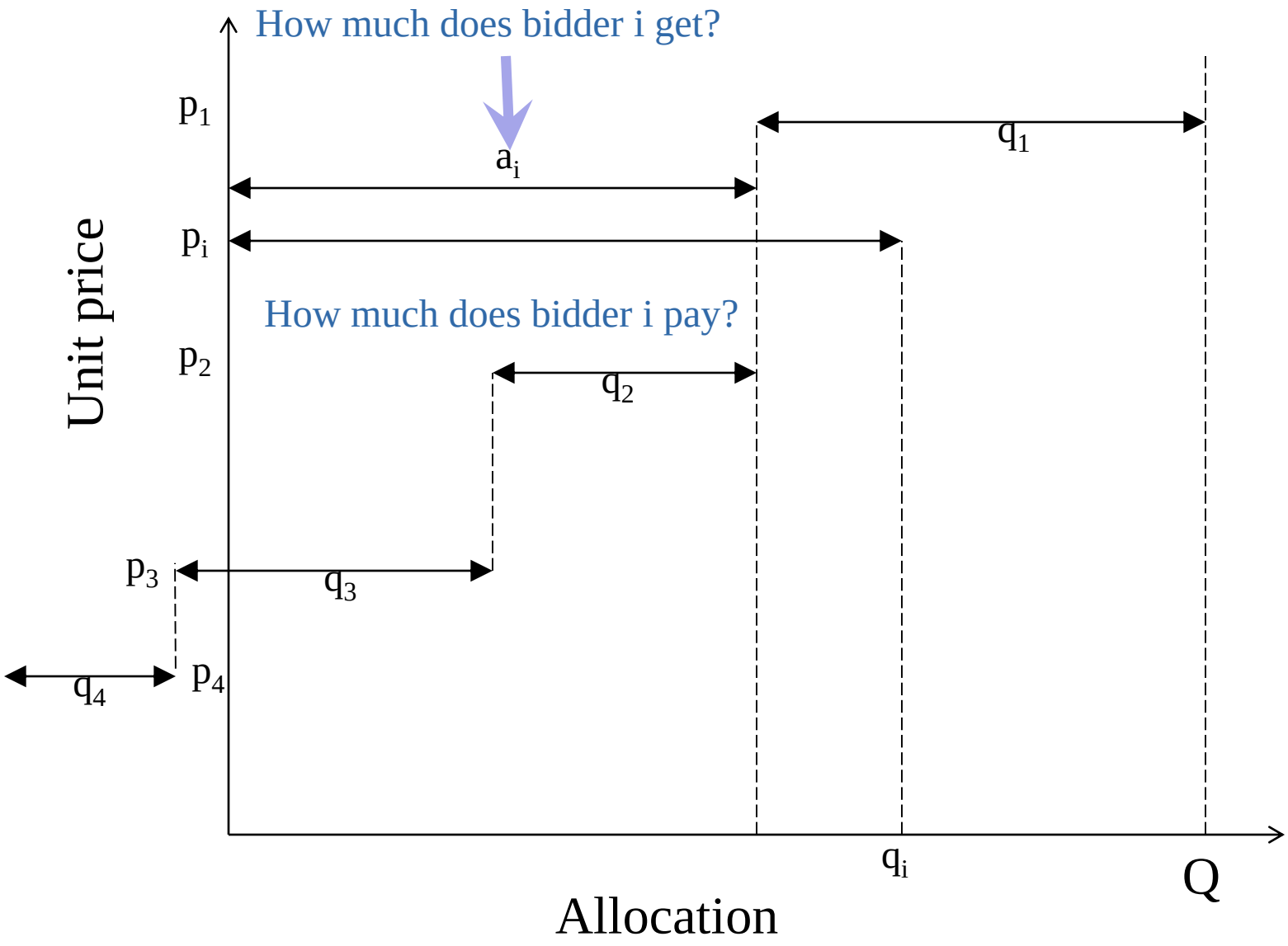
- Cost based on ‘exclusion compensation principle’
  - Bidder *i* inconveniences other bidders, who lose a slice of bandwidth because they are outbid by *i*
  - Cost to *i* set based on how much other bidders lose out because *i* exists

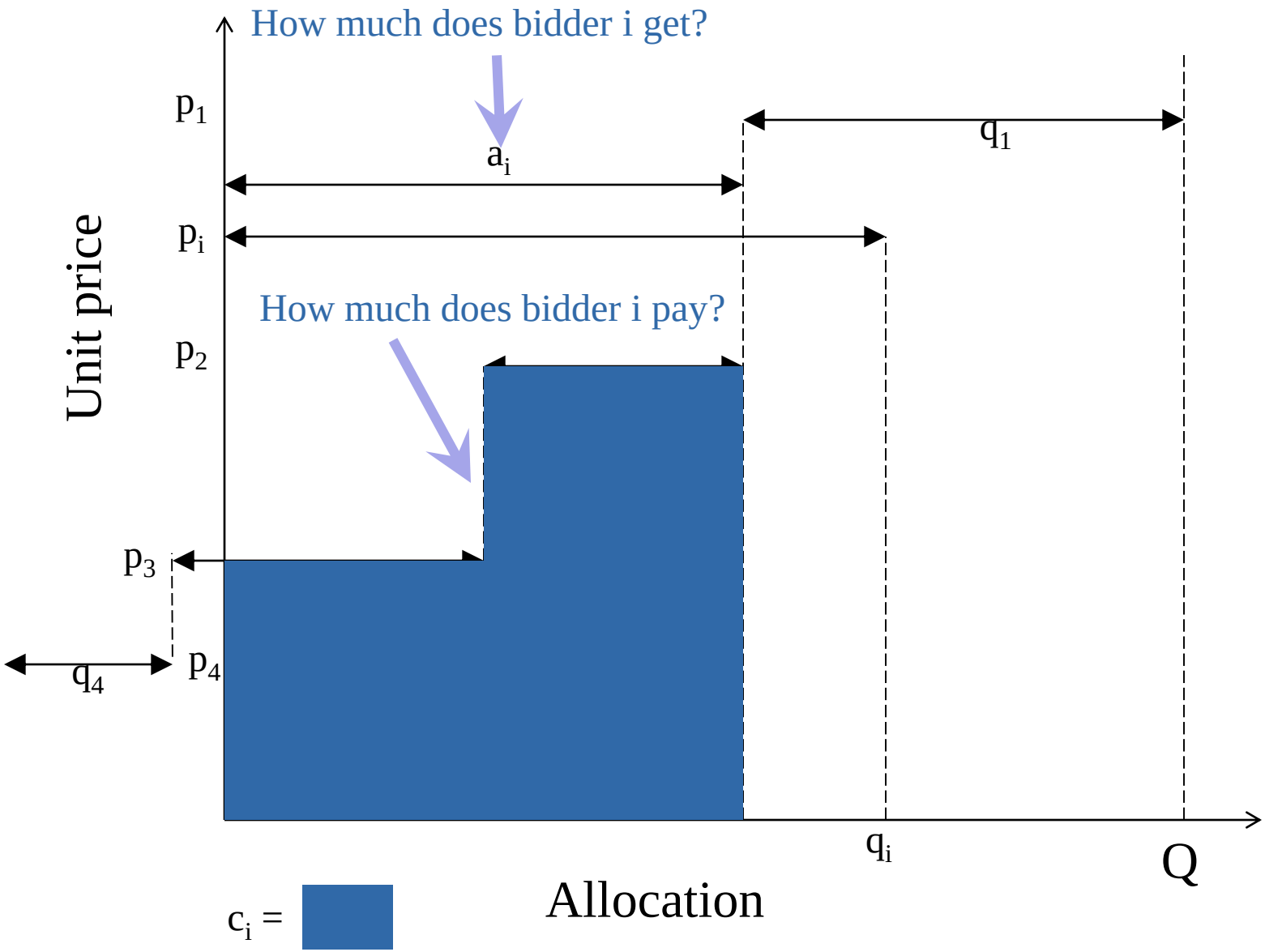

Player with highest unit-price has their quantity subtracted from the total
Unit price
p1
p2
p3
p4
q1
q2
q3
q4
Player with 2nd highest unit-price takes from what's left
Bandwidth available for auction
Q
Allocation

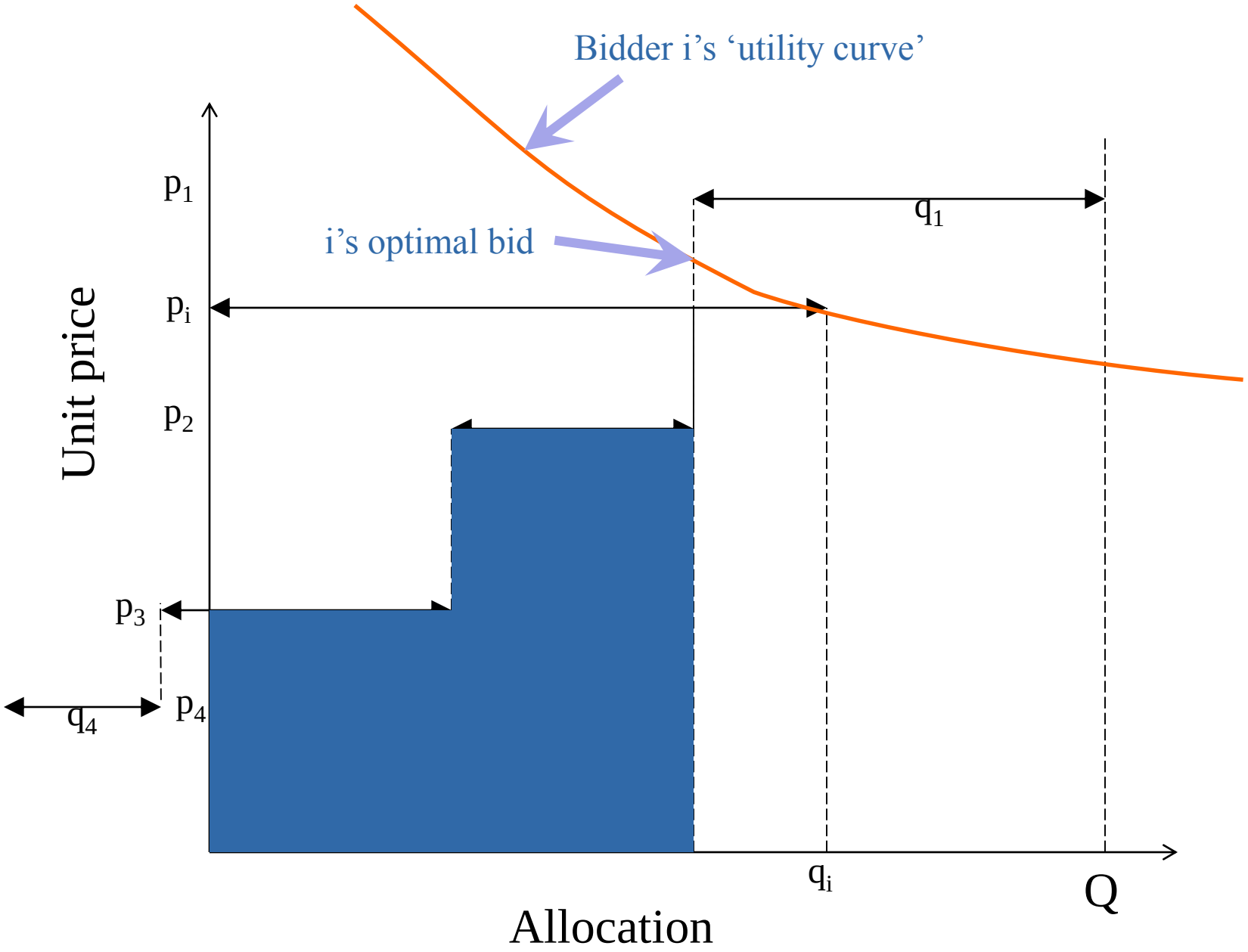

Unit price
p1
pi
p2
p3
p4
q1
q2
q3
q4
New bid
(qi, pi)
qi
Q
Allocation


How much does bidder i get?
Unit price
p_1
p_i
p_2
p_3
p_4
q_1
q_2
q_3
q_4
New bid
(q_i, p_i)
q_i
Q
Allocation


How much does bidder i get?
p_1
a_i
q_1
p_i
p_2
q_2
p_3
q_3
q_4
p_4
Unit price
q_i
Q
Allocation


How much does bidder i get?
p_1
a_i
q_1
p_i
Unit price
How much does bidder i pay?
p_2
q_2
p_3
q_3
q_4
p_4
q_i
Q
Allocation


How much does bidder i get?
p_1
a_i
q_1
p_i
Unit price
How much does bidder i pay?
p_2
p_3
q_4
p_4
q_i
Q
Allocation
c_i =


Bidder i's 'utility curve'
i's optimal bid
p_1
p_i
p_2
p_3
p_4
q_1
q_4
q_i
Q
Unit price
Allocation

PSP on LHCONE?

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

Does that satisfy requirements?

- Automatic, lightweight?
  - Create circuits only where needed (i.e. where competition exists)
  - May choose to not create circuits if natural performance satisfies requirements, trigger new auction if conditions change…
- Elastic, responsive?
  - Auction converges quickly, even for several tens of users
  - Can make convergence faster with ‘reserve price’
- Efficient, fair? Two sub-problems
  - How to set budgets?
  - How to cater for non-bidders/non-participants (casual users)?

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
  - How to set the budgets?
    - One-time allocation not enough, auctions recur as conditions change
    - 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 => create circuits where you don't really need them
  - Can bid to block opponents if you don't need your full budget
- Excess budget may be harmful
  - Blocking tactics, bidding for a link you don't need
  - Some way to penalize for under-used circuits?
- Carry-over of unspent budget?
  - Budget-hoarding => undesirable/'unfair' outcomes
  - Carry-over with exponential decay? Fixed ceiling?

How, and how often, to update budgets?

- Repeat auctions: possible learning of strategy
  - Learn which links your opponent favours, and when
  - May or may not be a problem, depends if result is co-operative or destructively competitive
- Needs simulation, with various bidding strategies
  - Budget adjustment must not destroy auction fairness
- Initial budget, and refresh policy, decided by WLCG?

How to represent LHCONE?

- Topology: by individual service provider?
  - PSP allows independent auctions, so each region can auction their part of the network separately
  - Introduces another budget-related problem:
    - With real money, the bidder decides how they are limited.
    - With fake money, need to co-ordinate between independent auctions to ensure no bidder overspends their total budget
    - i.e. with fake budgets, cannot truly decentralize the auction

Casual users, non-bidders?

- What fraction of b/w to auction per link?
  - If offer 100%, can sell it all => excludes casual users
  - If offer (e.g.) only 80%, casual users can fit into the other 20%
  - But then: how can someone who *wants* 100% get it if nobody else is using it, but they're only granted 80%?
  - Soft-guaranteed b/w may help: allow giving more if no competition, less if competition arises from new users
    - Can trigger a new auction if conditions change too much
    - Casual users who use too much should ask for a budget

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
  - Budget must have ‘real value’ to the bidders
  - Bidding strategies can interact with long-term budget management
  - Need to understand how, to keep the auction useful (truthful)
- Network topology/representation?
  - How to cater for casual users vs. the desire to use all of link?
  - How to respond to less than total demand?
- Needs more work, nonetheless, PSP shows promise as a bandwidth allocation mechanism for LHCONE