UCL

Peerlive

Incentive Mechanisms for QoS on P2P Systems

Raúl Leonardo Landa Gamiochipi

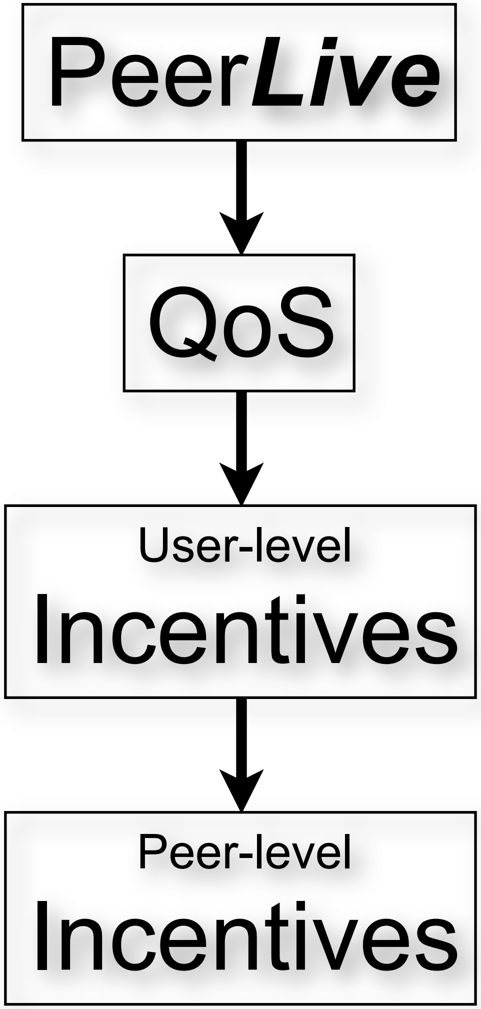
Department of Electronic and Electrical Engineering
University College London

What is PeerLive?

- Goal:
  - Near **real-time video** distribution using P2P technology

- Techniques:
  - QoS Overlays
  - Synthetic coordinate systems
  - Market-driven resource allocation
  - Reputation systems

Why Incentive Mechanisms?



Peer*Live*

QoS

User-level Incentives

Peer-level Incentives

What must peers do?

- The PeerLive incentive mechanism should help peers decide:
  - Which fragment requests to answer
  - In which order to answer them
  - How to measure peer contributions (QoS-aware)
  - How to reciprocate previous interaction (QoS-aware)

State of the Art: Incentives for QoS

- Contribution-based peer selection
  - Peer contribution ranking (Chuang, Habib 2006)
  - Creating downloader coalitions (Epema, Iosup et. al. 2007)

- Reciprocation for queue management
  - Using trust scores as queue priorities (Grothoff, 2003)
  - Token Stealing Algorithm (Pai, Mohr 2006)

Why Market-based incentives?

- Complex QoS conditions
  - Strategic, heterogeneous users
  - Changing network conditions
- Completely distributed
- Catallaxy (Friedrich Hayek, The Use of Knowledge in Society - 1945)

PeerLive: A Network of Local Markets

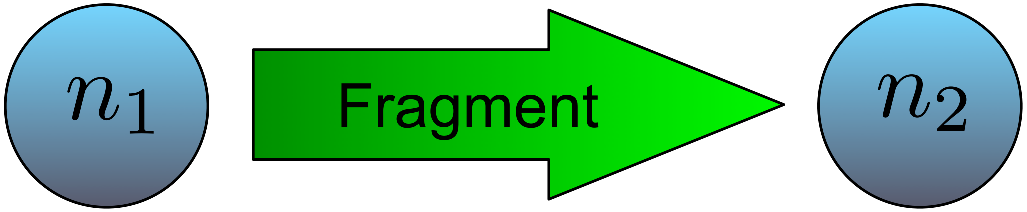
- Peer neighbourhoods are *local markets*
  - Every node “buys” and “sells” bandwidth to its neighbours
  - Prices balance supply and demand (Kearns et. al. 2004)
- Peers pay currency for the transmission of fragments

PeerLive: A Network of Local Markets

- The “wealth” of a peer is stored in accounts on all its trading partners

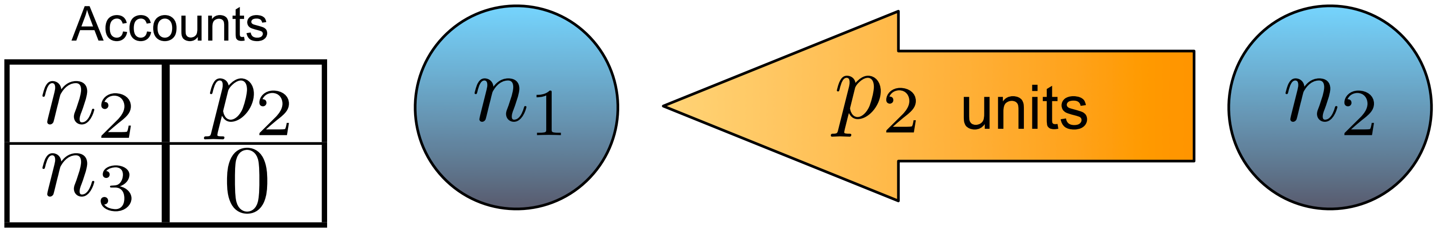
Accounts

n_2	0
n_3	0



PeerLive: A Network of Local Markets

- The “wealth” of a peer is stored in accounts on all its trading partners



Accounts

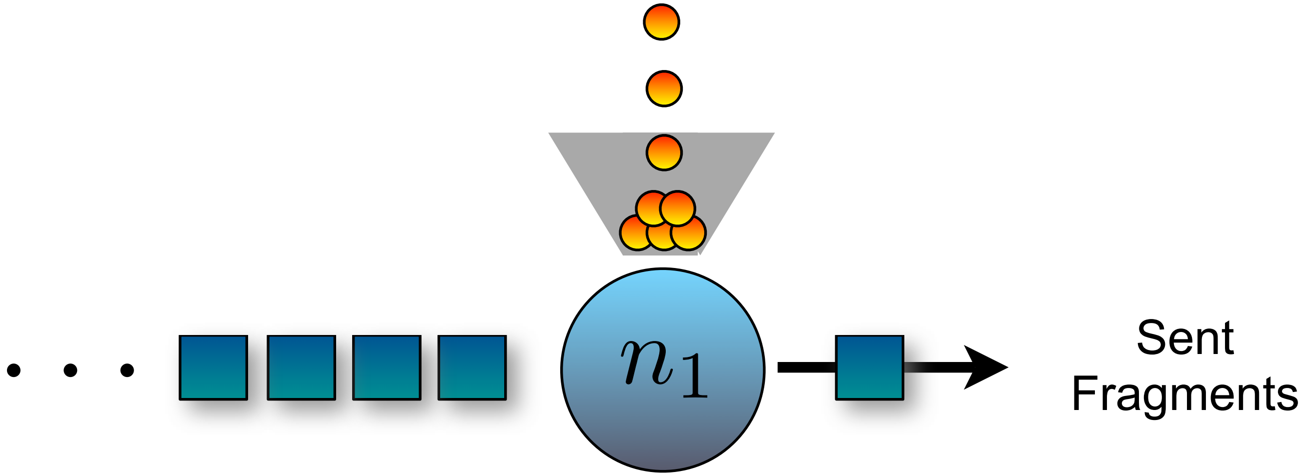
n_2	p_2
n_3	0

Peer 2 ***sold*** peer 1 a fragment for a cost p2

Peer 1 ***pays*** by increasing the account of peer 1 by p2

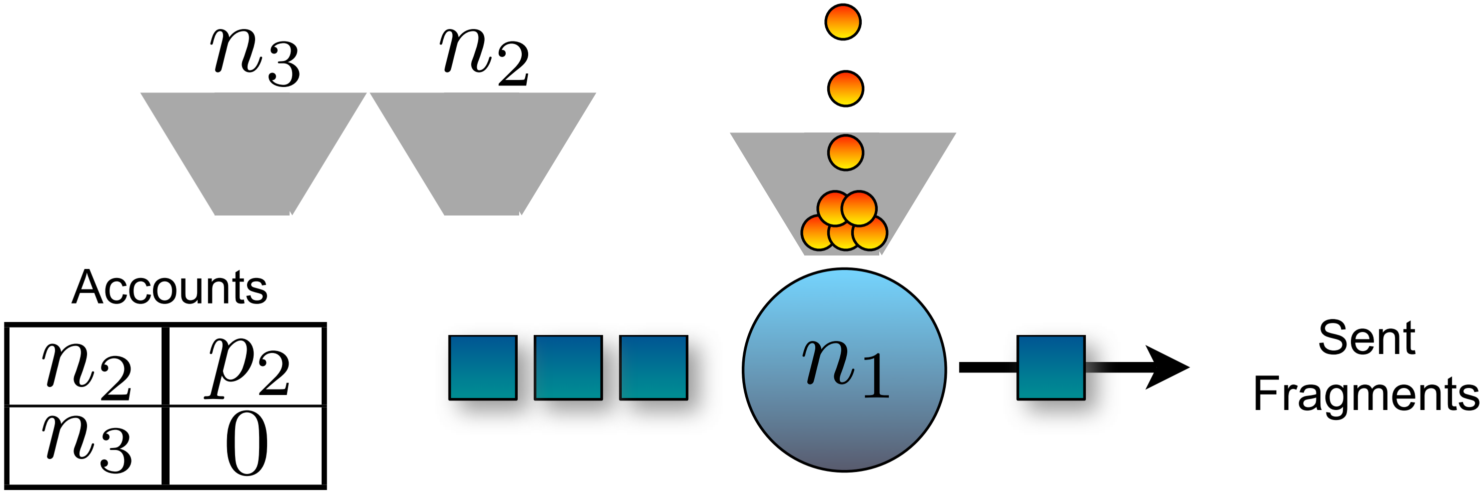
Making Bandwidth a Commodity

- Peers have a Shared token bucket to control outgoing bandwidth
  - Tokens represent a given number of bytes
  - Tokens are added periodically, according to peer capacity, and consumed when fragments are sent



Making Bandwidth a Commodity

- Additionally, peers have *Private token buckets* for each one of the peers they trade with
  - Tokens can be bought. This transfers them from the Shared bucket to the Private Buckets.
  - This is equivalent to reserving bandwidth.

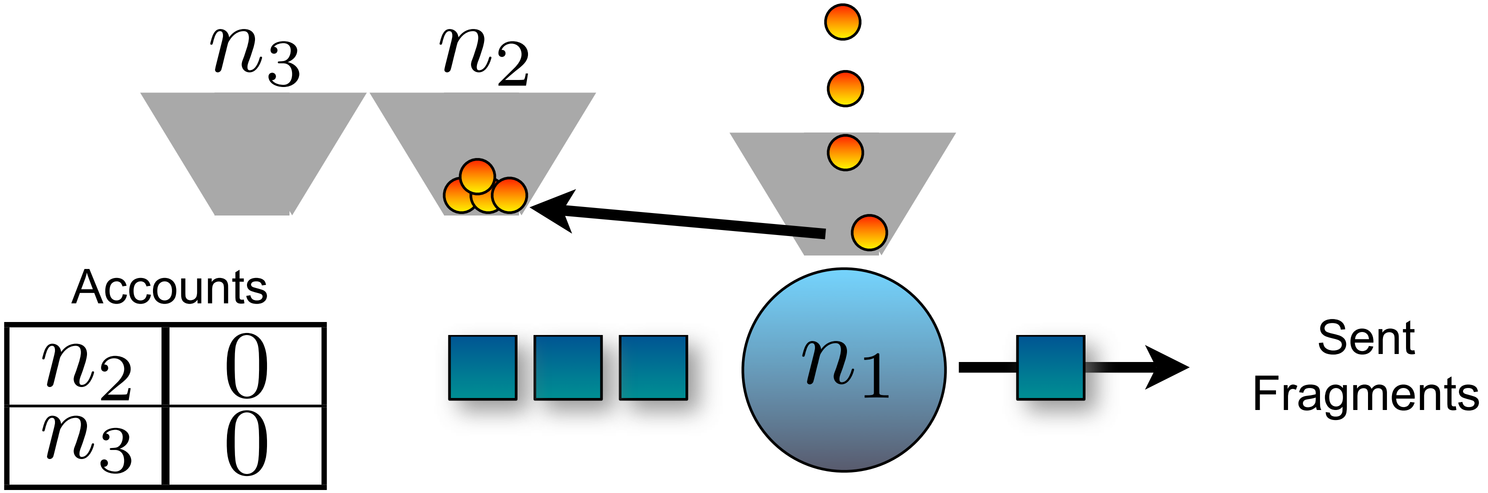


Accounts

n_2	p_2
n_3	0

Making Bandwidth a Commodity

- Additionally, peers have *Private token buckets* for each one of the peers they trade with
  - Tokens can be bought. This transfers them from the Shared bucket to the Private Buckets.
  - This is equivalent to reserving bandwidth.



Accounts

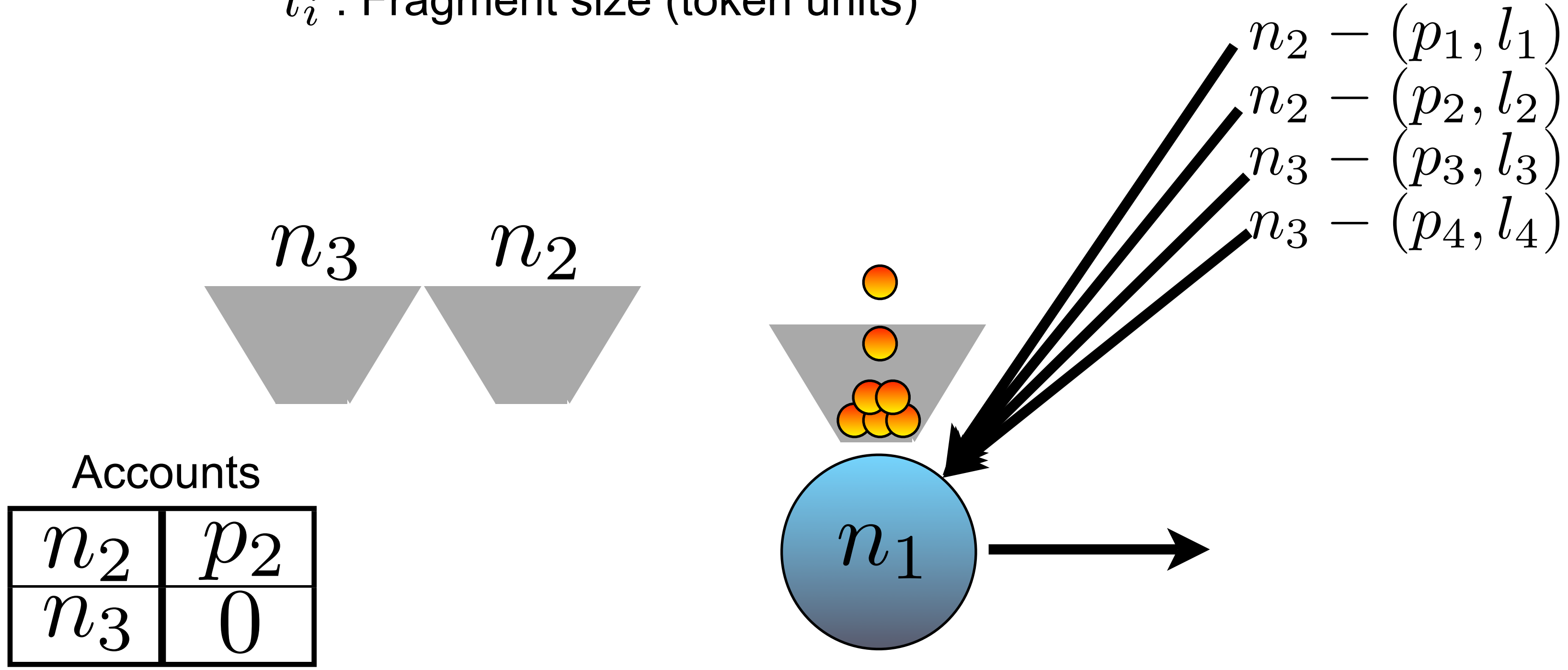
n_2	0
n_3	0

PeerLive: Token Stealing Algorithm with Price Mapping

- When requests arrive from peers with active accounts:
  1. They are ordered on decreasing account balance order

p_i : Currency offered for fragment

l_i : Fragment size (token units)



Accounts

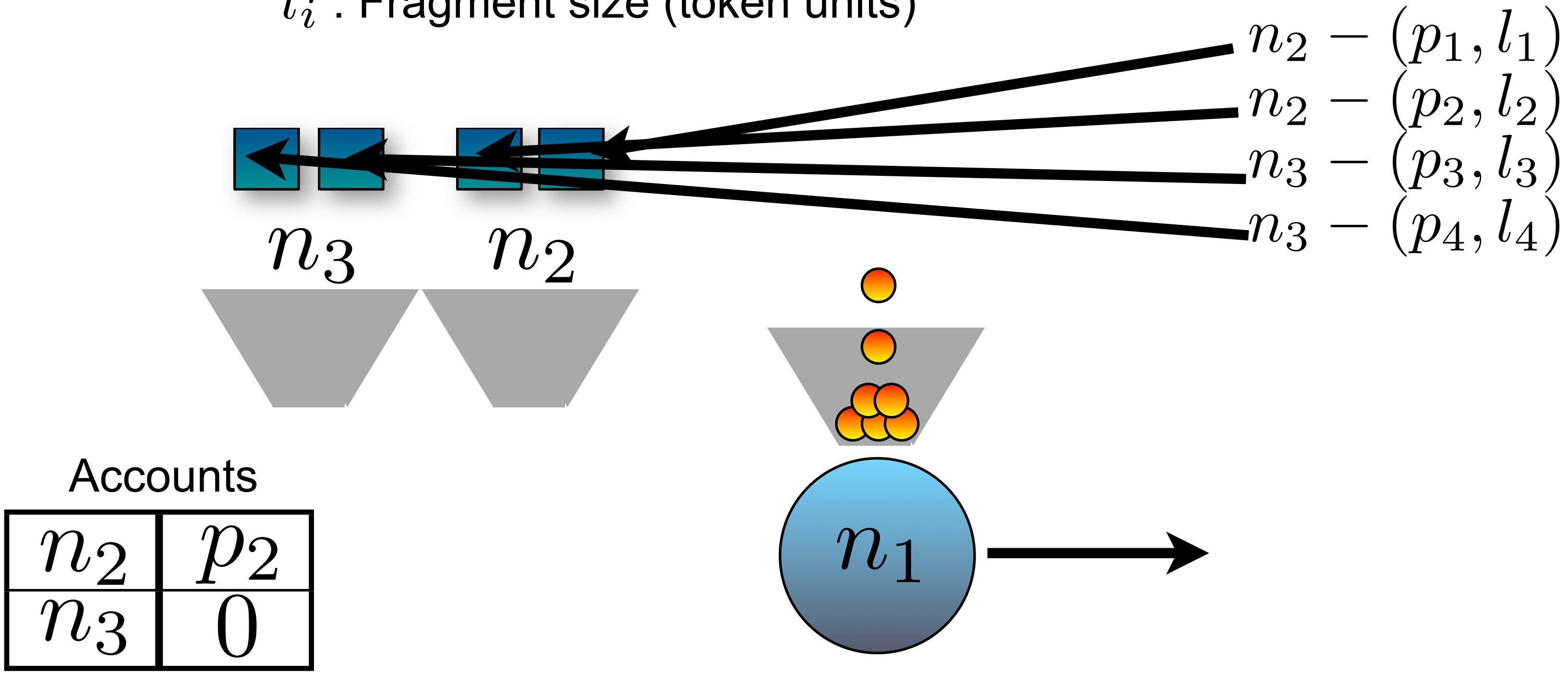
n_2	p_2
n_3	0

PeerLive: Token Stealing Algorithm with Price Mapping

- When requests arrive from peers with active accounts:
  1. They are ordered on decreasing account balance order

p_i : Currency offered for fragment

l_i : Fragment size (token units)

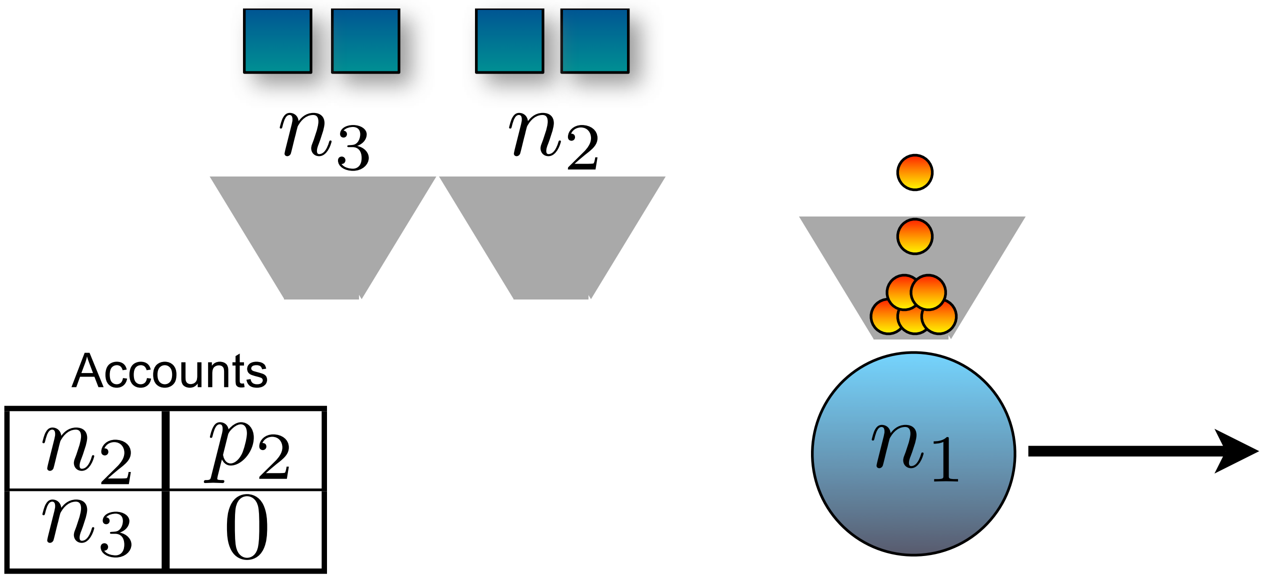


Accounts

n_2	p_2
n_3	0

PeerLive: Token Stealing Algorithm with Price Mapping

- When requests arrive from peers with active accounts:
    1. They are ordered on decreasing account balance order

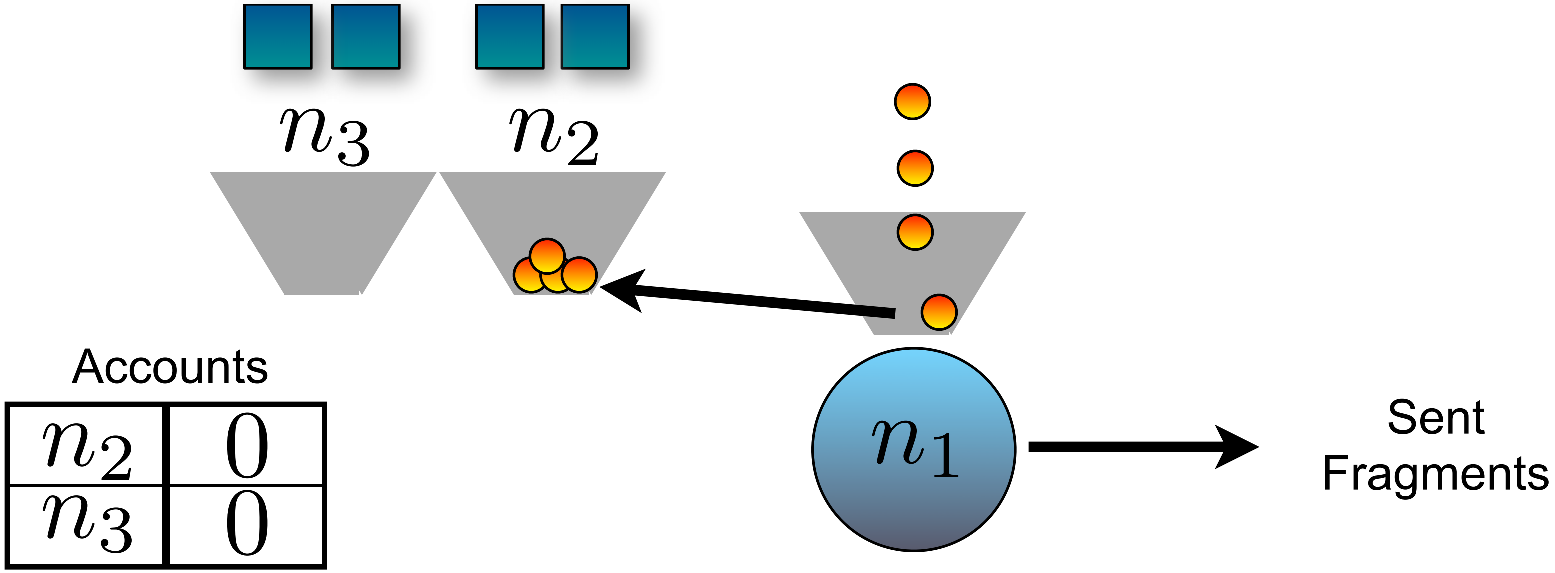


Accounts

n_2	p_2
n_3	0

PeerLive: Token Stealing Algorithm with Price Mapping

- When requests arrive from peers with active accounts:
  1. They are ordered on decreasing account balance order
  2. Their currency is used to buy tokens

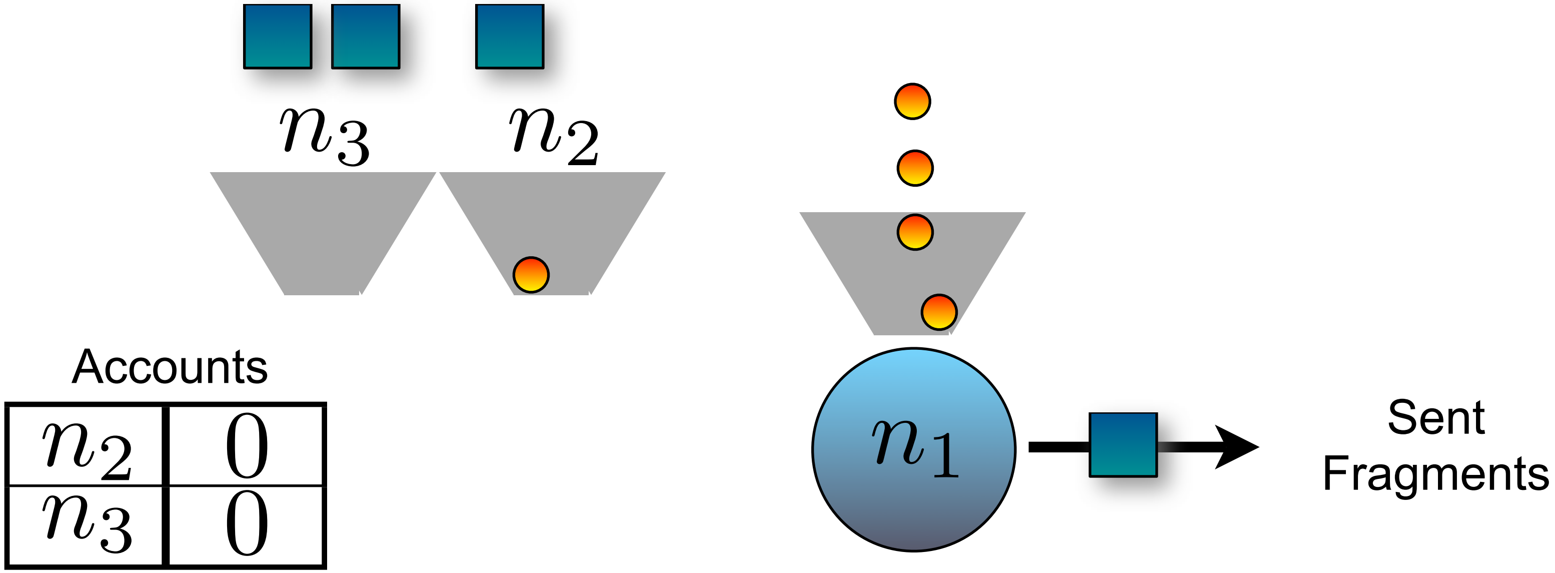


Accounts

n_2	0
n_3	0

PeerLive: Token Stealing Algorithm with Price Mapping

- When requests arrive from peers with active accounts:
  1. They are ordered on decreasing account balance order
  2. Their currency is used to buy tokens
  3. If the tokens are enough to send the request, it is serviced (tokens are consumed).

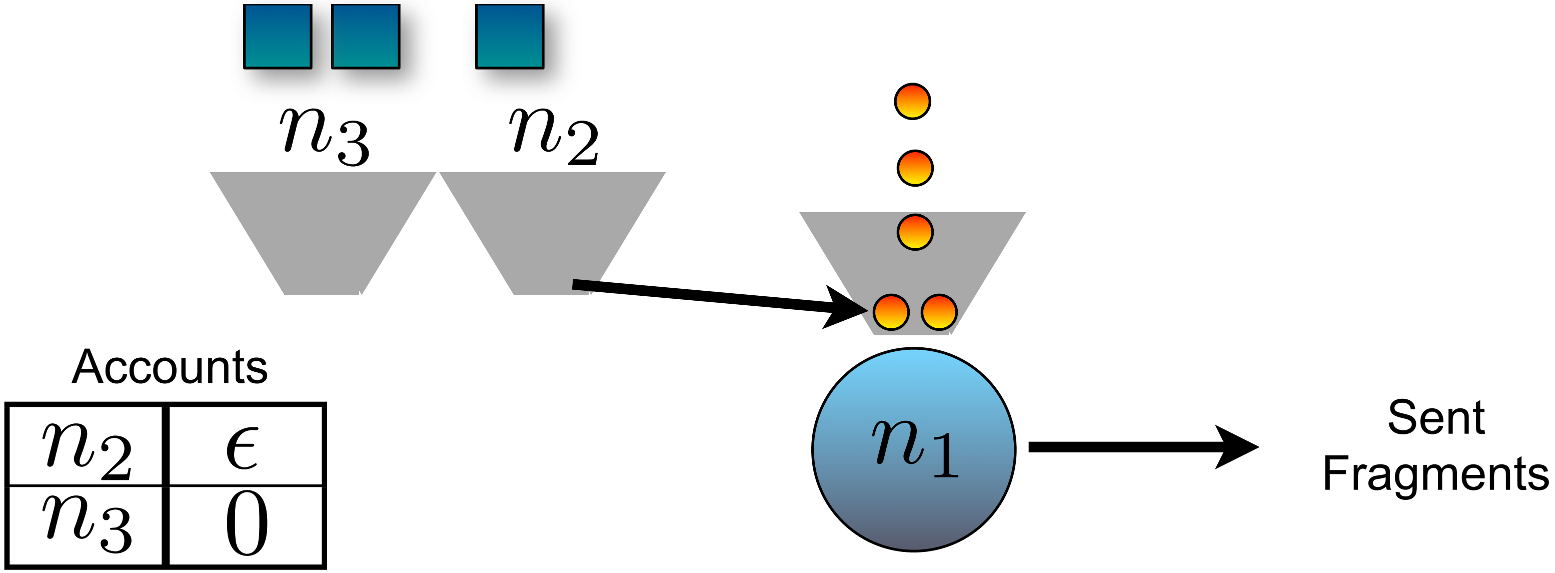


Accounts

n_2	0
n_3	0

PeerLive: Token Stealing Algorithm with Price Mapping

- When requests arrive from peers with active accounts:
    1. They are ordered on decreasing account balance order
    2. Their currency is used to buy tokens
    3. If the tokens are enough to send the request, it is serviced (tokens are consumed).
    4. Non-consumed tokens are sold back to the shared bucket.



Accounts

n_2	ϵ
n_3	0

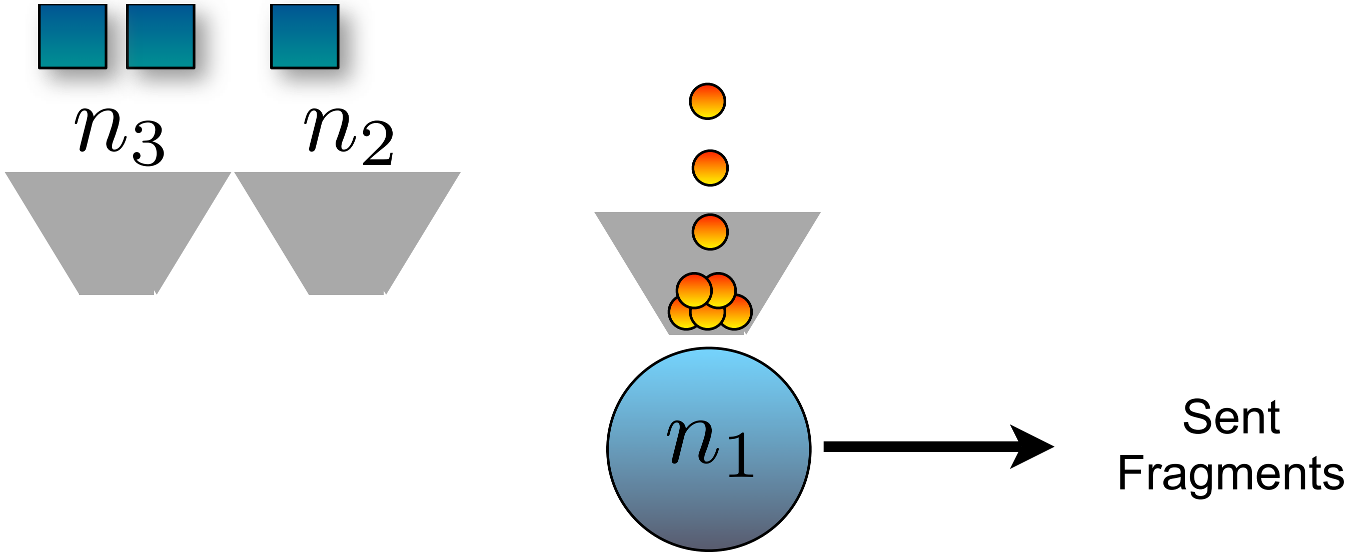
PeerLive: Token Stealing Algorithm with Price Mapping

- Peers now compete with each other for the tokens left on the shared bucket
    1. Their benefit per token unit (t_i) is calculated

$$t_2 = \frac{p_2}{l_2}$$

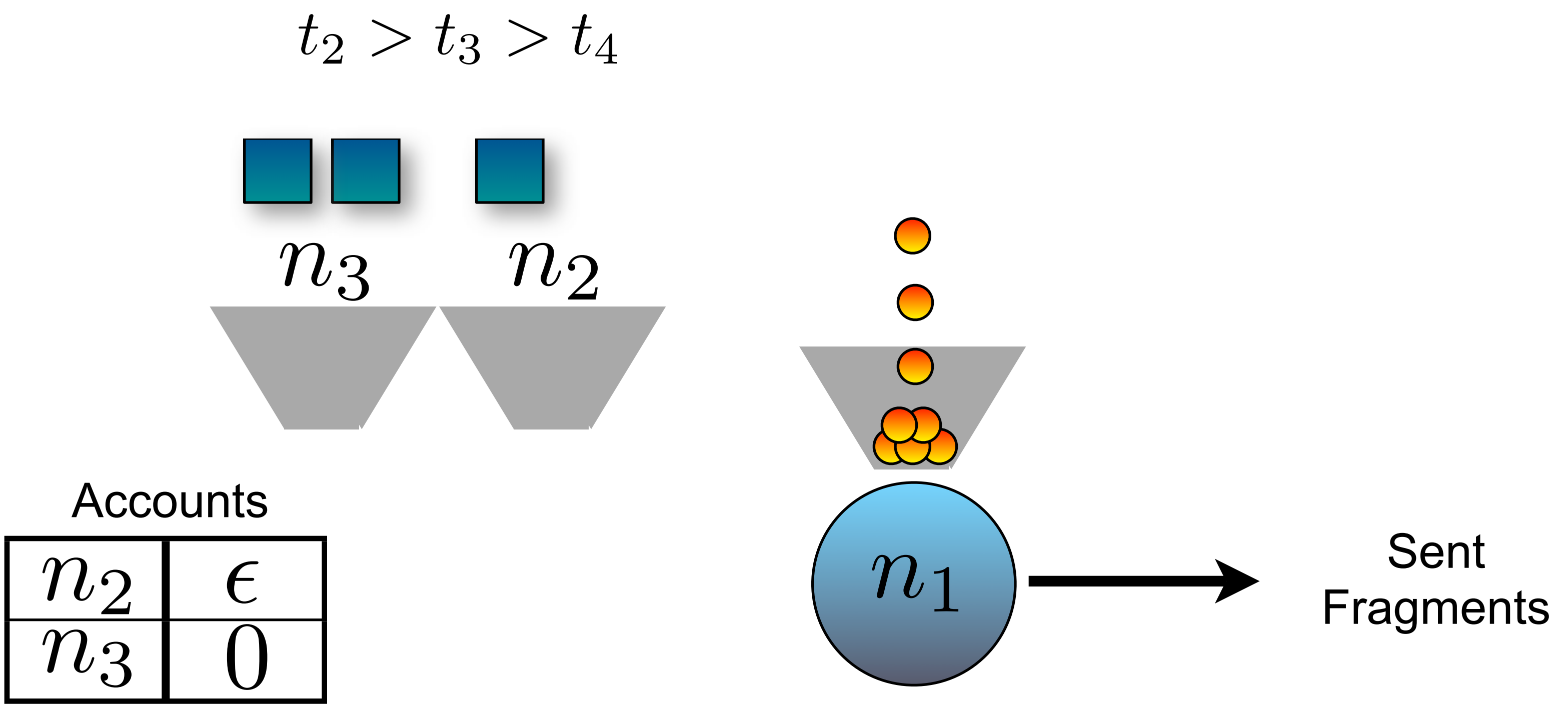
$$t_3 = \frac{p_3}{l_3}$$

$$t_4 = \frac{p_4}{l_4}$$



PeerLive: Token Stealing Algorithm with Price Mapping

- Peers now compete with each other for the tokens left on the shared bucket
  1. Their benefit per token unit is calculated
  2. They are ordered in decreasing order of benefit

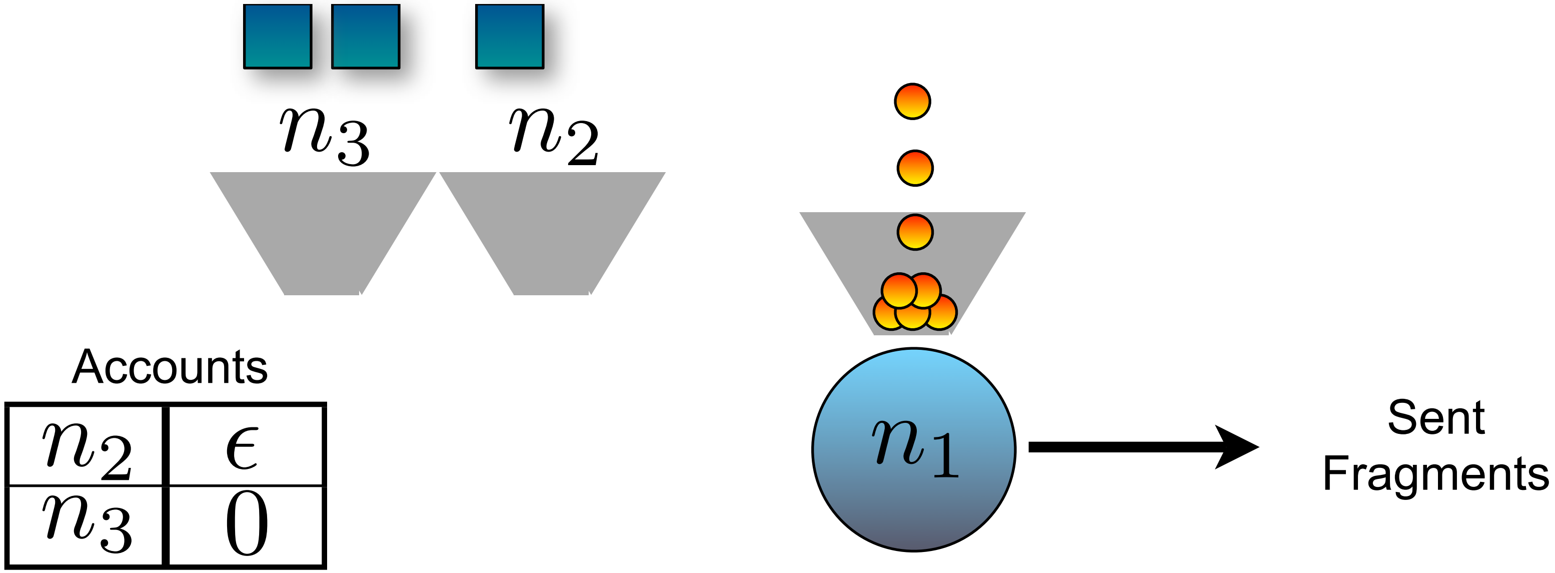


Accounts

n_2	ϵ
n_3	0

PeerLive: Token Stealing Algorithm with Price Mapping

- Peers now compete with each other for the tokens left on the shared bucket
  1. Their benefit per token unit is calculated
  2. They are ordered in decreasing order of benefit
  3. The requests are served until no tokens remain in the shared bucket

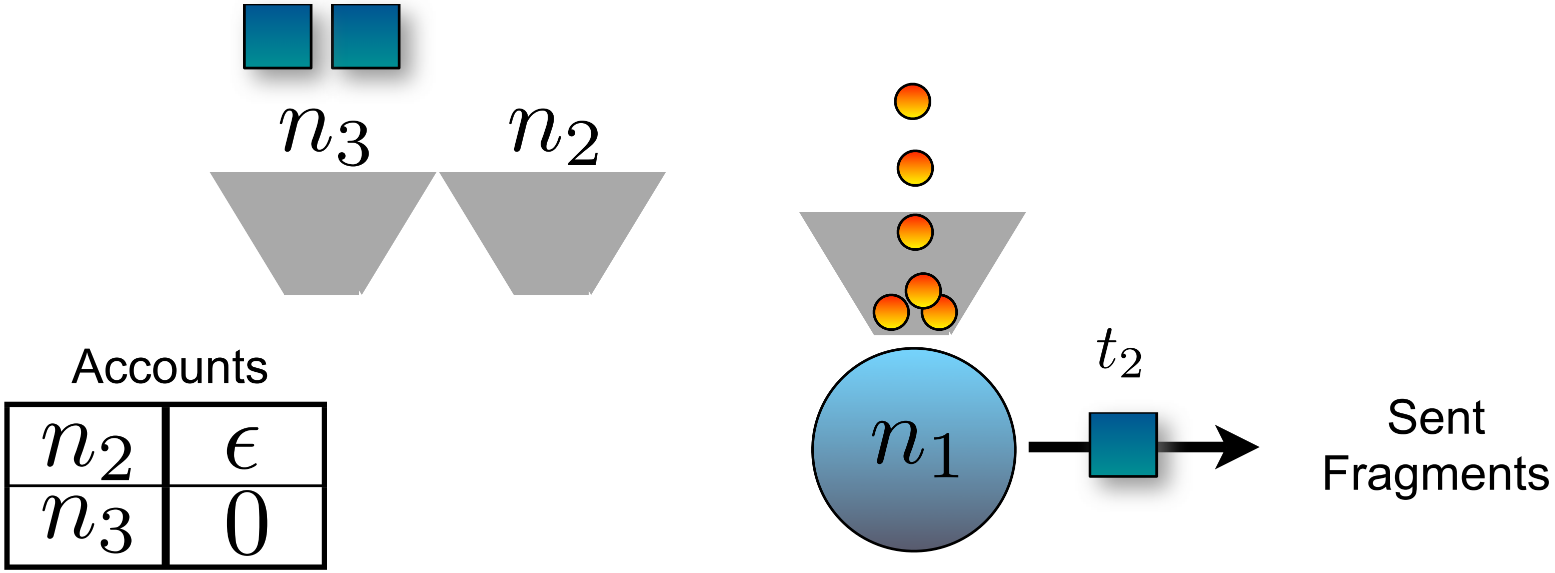


Accounts

n_2	ϵ
n_3	0

PeerLive: Token Stealing Algorithm with Price Mapping

- Peers now compete with each other for the tokens left on the shared bucket
  1. Their benefit per token unit is calculated
  2. They are ordered in decreasing order of benefit
  3. The requests are served until no tokens remain in the shared bucket

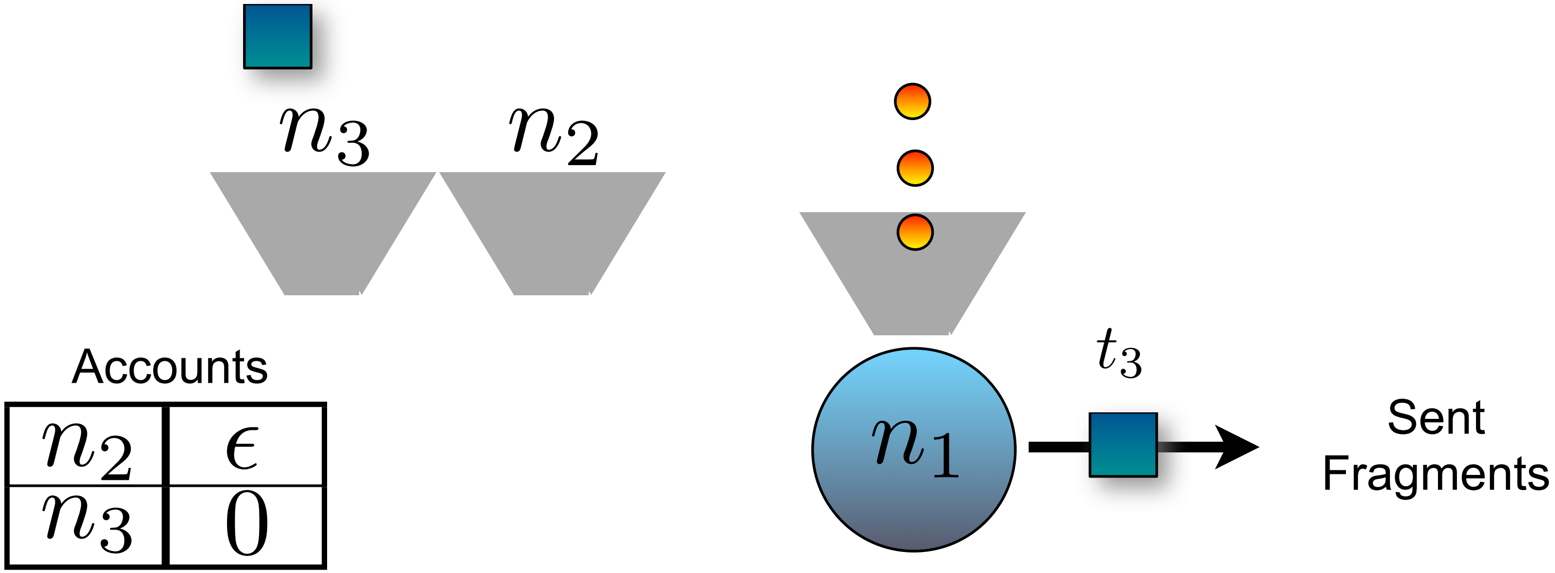


Accounts

n_2	ϵ
n_3	0

PeerLive: Token Stealing Algorithm with Price Mapping

- Peers now compete with each other for the tokens left on the shared bucket
  1. Their benefit per token unit is calculated
  2. They are ordered in decreasing order of benefit
  3. The requests are served until no tokens remain in the shared bucket

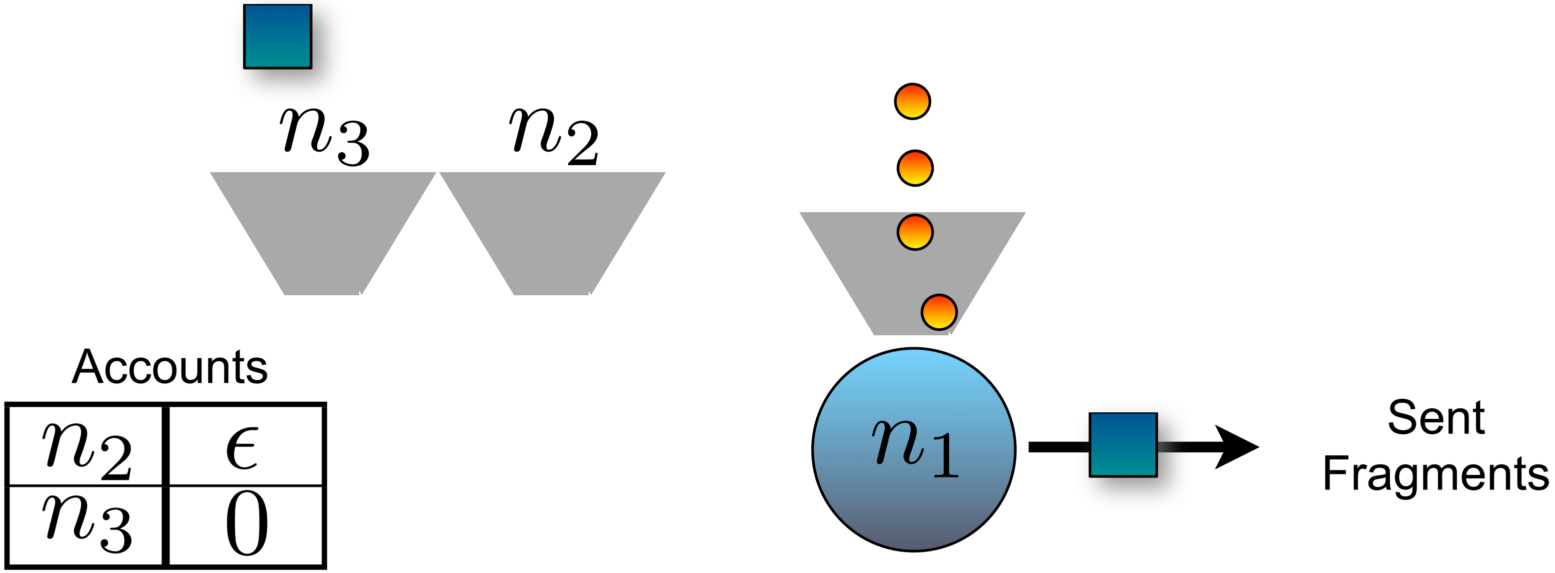


Accounts

n_2	ϵ
n_3	0

PeerLive: Token Stealing Algorithm with Price Mapping

- Peers now compete with each other for the tokens left on the shared bucket
  1. Their benefit per token unit is calculated
  2. They are ordered in decreasing order of benefit
  3. The requests are served until no tokens remain in the shared bucket
  4. Non-consumed tokens are left in the shared bucket for the next round.



Accounts

n_2	ϵ
n_3	0

PeerLive: Summary

- Bandwidth contributions build up peer accounts
- This accumulated wealth is used to get priority access for peer service
- Price reflects fragment value, including QoS
- Support for *strategic* peers

PeerLive: Future Work

- Supply-demand matching protocol
- Market convergence optimality
- Reputation system
- Currency circulation

Thank You!

- ***www.peerlive.org***

- Any questions?