Selfish Caching in Distributed Systems

A Game-Theoretic Analysis

B. Chun
M. Barreno
K. Chaudhuri
C. Papadimitriou
H. Wee
J. Kubiatowicz

Tutor: Prof. Gerhard Weikum

Odysseas Papapetrou odysseas@mpii.de

Overview

- Selfish-caching problem
  - Peers need access to the resource
  - Fetch or cache
- Game Theory
- Solve the problem - Basic approach
- Payment-enhanced approach
- Conclusions

Game Theory - Introduction

- The name of the game
- Interesting point: Will the system stabilize?
- Steps
  - Game modelling (model as a competitive game)
  - Let the system evolve (simulation)
  - See if it reaches a stable state (we call this stable state a Nash equilibrium)
  - Evaluate and retry

Game Theory – Some Definitions

- **Rational players:** Act for their own profit
- **Each player** has a set of possible actions
- **Cost** of each action is common knowledge
- **Social Cost:** Sum of the cost for all the players
- **Pure Strategy:** A (rule-like) representation of the player's behaviour: if (condition) then (action)

  ex. if(cost<10$) then buy it

Game Theory – Some Definitions

- **Pure Strategy Nash Equilibrium:** No player can benefit by altering his strategy (if the others keep their strategies unchanged)

- **Optimal Solution OR Social Optimum:** The set of strategies that minimize the social cost (or maximize the social payoff)

Game Theory – Some Definitions

- **Price of Anarchy (PoA):** Ratio of the worst Nash Equilibrium social cost to the social optimum (the price we have to pay for being decentralized)

$$PoA = \frac{SocialCost(WorstNE)}{SocialCost(SocialOptimum)}$$

- **Optimistic Price of Anarchy (OPoA):** Ratio of the best Nash Equilibrium social cost to the social optimum

$$OPoA = \frac{SocialCost\,(BestNE\,)}{SocialCost\,(SocialOpti\,mum)}$$

Basic Approach

- Modelling the system as a (1-resource) game
  - Players: Peers, Resources: Documents
  - Possible Functions: Caching or Fetching
  - Configuration of a doc: The peers that cache the doc.
  - Strategy of a peer: The documents it will cache
  - Personal Cost: The cost to Fetch or Cache a doc.
  - Social Cost: The sum of personal costs for all players

Basic Approach

- **Selfish behaviour:** Each peer only cares about minimizing its own cost (or maximizing its own payoff)
- See if and where the system stabilises (Nash Equilibrium)
- Evaluate the Nash Equilibrium (compare to the optimal solution)
- How can the Nash Equilibrium be improved

Costs

- Personal Costs
  - **Placement-caching cost (independent of #demands w_{ij})**
  - **Remote Fetching cost (represented by a network distance matrix) for each time we fetch → multiply with #demands w_{ij}**

$$C_i(S) = a_{ij} S_i + w_{ij} d_{il(i,j)} (1 - S_i)$$

- Social Cost

$$C(S) = \sum_{i=0}^{n-1} C_i(S)$$

- Social Optimum
  - The set of strategies that minimize the Social Cost…

$$C(S_0) = \min_S C(S)$$

Nash Equilibrium

Proof that a Nash Equilibrium exists:

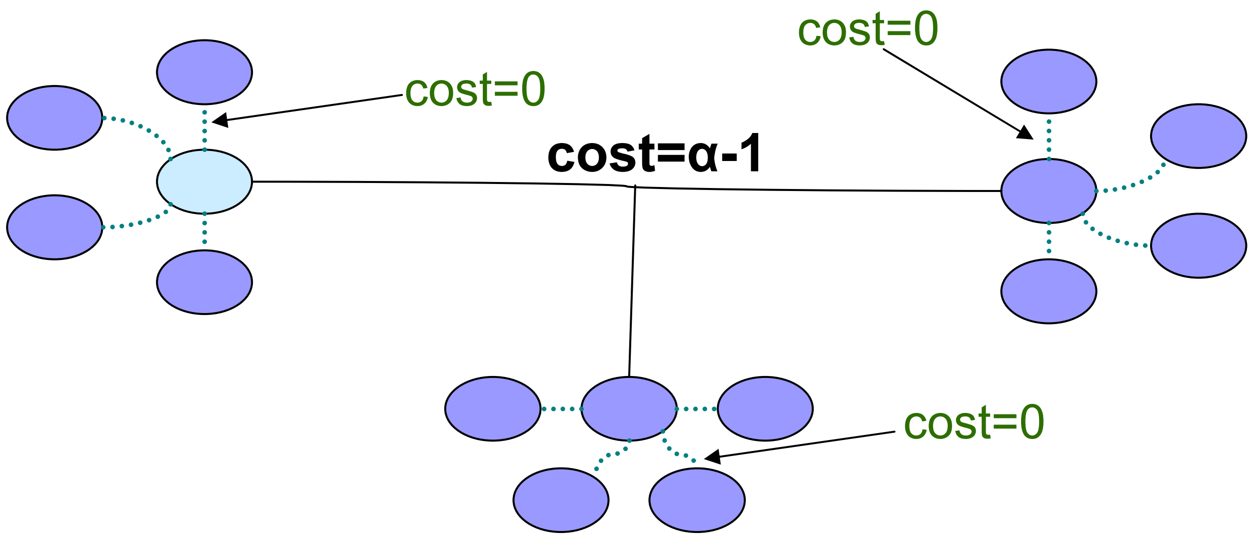
- Ignore nodes with zero demand

$$\beta_x = \frac{a}{w_{xj}}$$

- Order list of β_x ascending
- Start caching objects on peers with lower β_x. Each time remove all peers from β_x list that can get the object from the new caches.
- The system becomes stable when the list is empty (all peers are pleased, none of the peers wishes to cache or de-cache)

Nash Equilibrium

- Nash Equilibrium depends on the topology
- PoA also depends on the topology but can be bad (asymptotically approaching *O(n)* for large dimensions)



Placement cost is always α

but…

- Nash Equilibrium may ***seem*** good for the peers, but is not good for the system

- Intuition: Force the peers to cooperate

  Find a way so that the peers share the placement cost (cost of caching)…

Payment Model

- Money builds a protocol…
- Since P2P cannot force them, we can give them incentive to collaborate
- Proposal: Biding on someone to cache a document
  - Cache
  - Read from remote source
  - Pay one (bid) to cache it for you
  - Rule: If he caches, you have to pay!!!
- Who to pay? How much to pay? What to cache?
- What changes now

Payment Model

- Strategy now becomes: $(v_i, b_i, t_i) \in \{N, \Re_+, \Re_+\}$ (who to bid on, how much, what is your threshold to cache)
- Who to bid on? How much?
  - The peers will bid as much as they can, so that they will actually have profit (compared to when fetching or caching themselves):

  cost(peer,doc)>=cost(peer,bid_cache)+bid

  - The peers will cache only when they will actually have profit:

  cost(peer,doc)>=cost(peer,cache)-sum(all_bids)

Payment Model

- Game result: $\{(I_i, v_i, b_i, R_i)\}$

{(Replicate or not, player v that receives my bid for caching, payment I make to v, payments I receive)}

- Personal cost:

$$C_i(S) = a_{ij} I_i + w_{ij} d_{il(i,j)} (1 - I_i) + b_i I_{vi} - R_i I_i$$

- Social cost: Sum of personal costs (the bids and payments are zero-sum → do not affect the cost)

$$C(S) = \sum_{i=0}^{n-1} C_i(S)$$

Payment Model

- All the basic-game Nash Equilibria are Equilibria in the payment game too
- The PoA in the payment game is at least equal to the PoA in the basic game
- An Equilibrium for a given topology for the payment game can be even worse than the respective Equilibrium in the basic game Example

Payment Model

The optimistic PoA in the payment game is always 1 (social optimal configuration is always a Nash Equilibrium)

- Proof: Find a set of thresholds t and bids b that stabilise the social optimal configuration
  - Get the optimal configuration
  - Distribute the threshold cost for each peer that caches the object to the peers that read the object from it

Payment Model

- Payment Model Vs Basic Model
  - Payment gives players intensives to cache ➔ generally leads to better solutions
  - **In some cases** the basic model can reach a better Nash Equilibrium than the payment model
  - Payment model has always optimistic price of anarchy = 1 ➔ promising

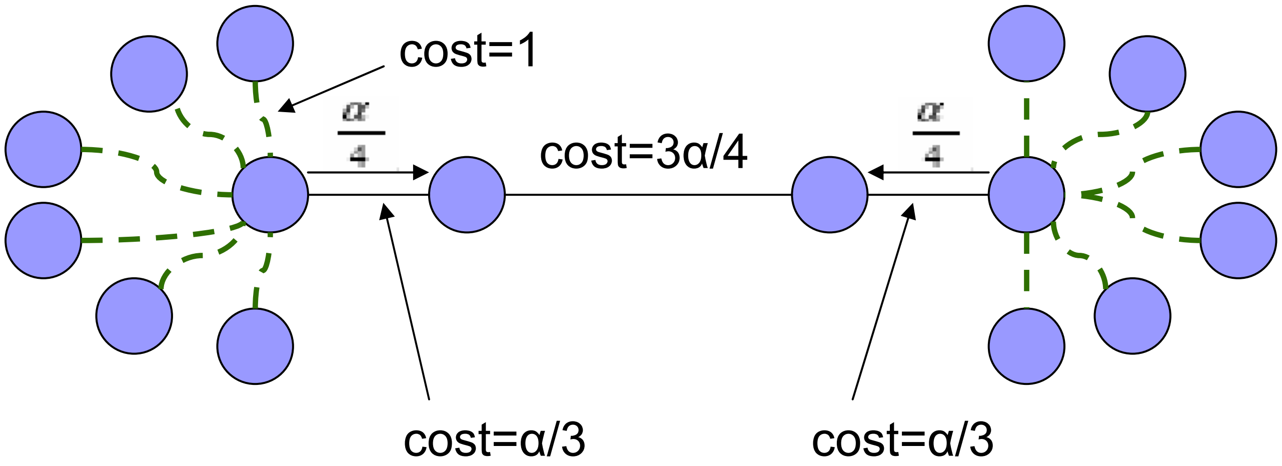
Conclusions

- Defined the Selfish Caching Problem as a game
- Solved it
- Enhanced it with Payments
- Optimistic PoA lower in payment model but NE is not always better compared to the basic model

Future work

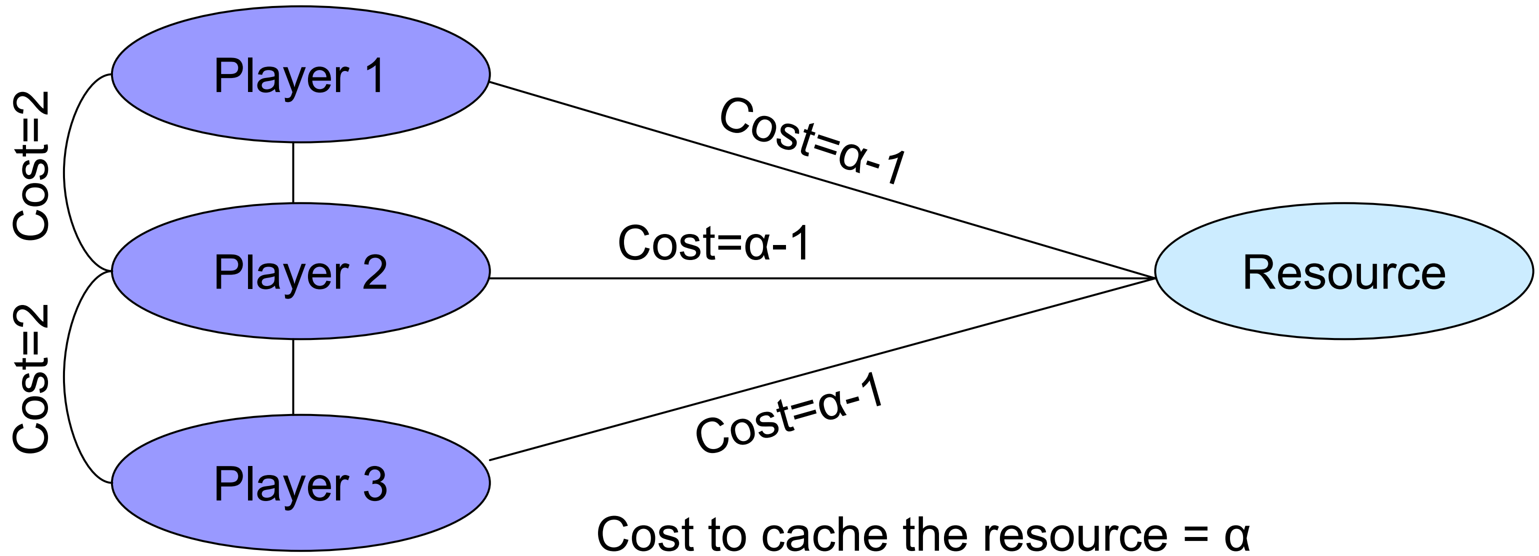
- Work on different configurations:
  - Capacitated game
  - Peers with different demand-rate for docs.
  - Peers with different placement costs
  - Aggregation effect
  - Server congestion
- Large scale simulations with realistic weights and different topologies

Example



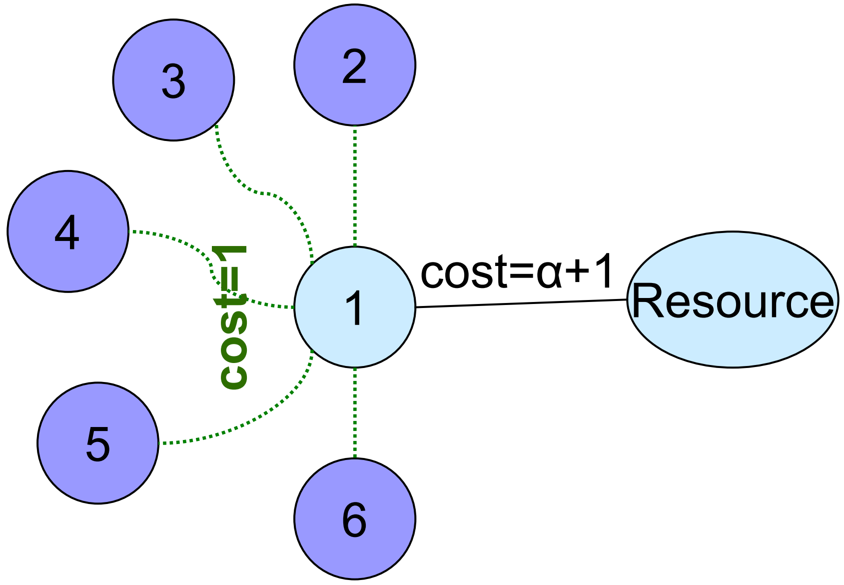
A Nash Equilibrium in the payment-enhanced game which is worse than any respective Nash Equilibrium in the basic game

Selfish Behaviour

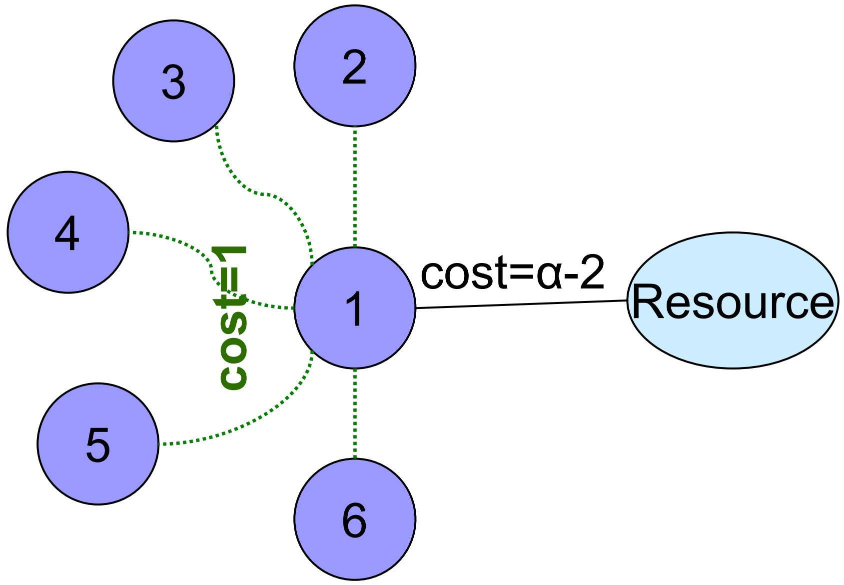


Selfish Behaviour: None of the players caches the resource since caching costs α and they can all fetch it for α-1. But is that the optimal for the system?

Nash Equilibrium

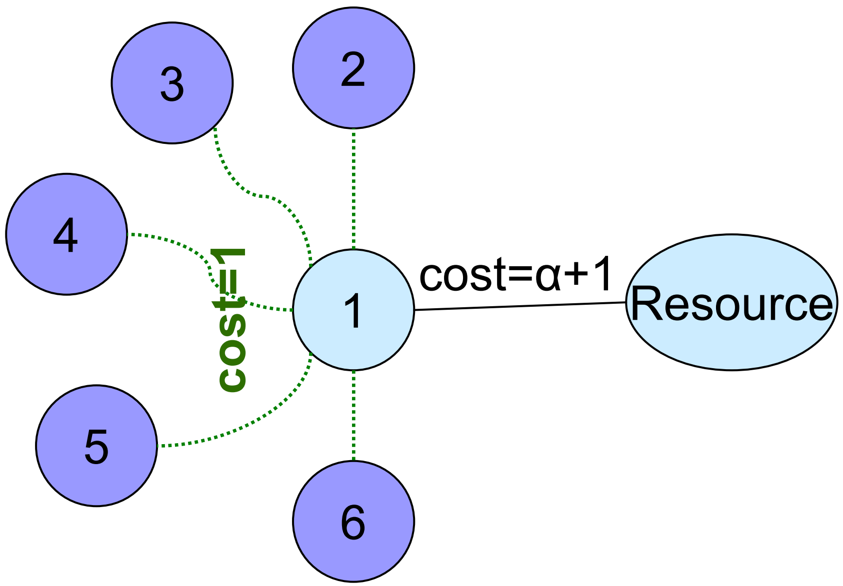


Caching cost=α for all peers
Peer 1 caches the resource
All the others get the data from peer 1
Social cost = α+6 = optimal

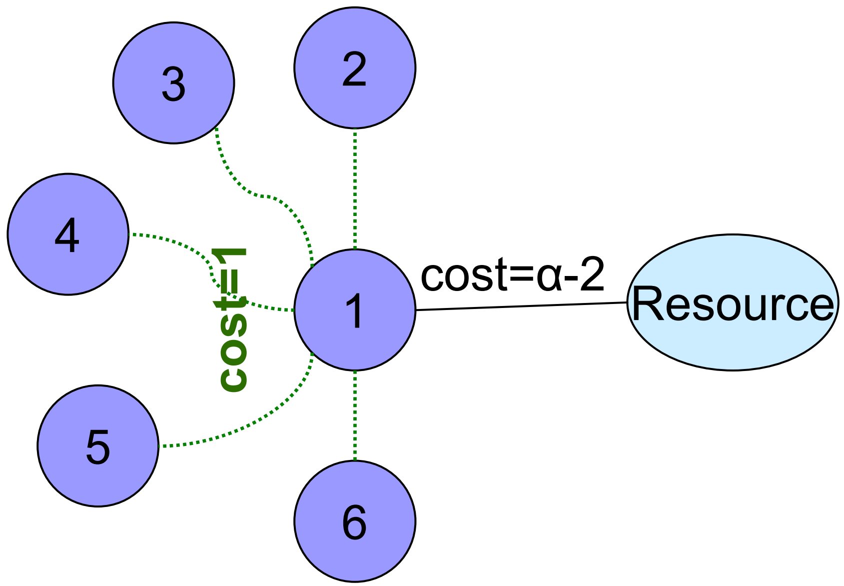


Caching cost=α for all peers
Peer 1 can fetch the resource cheaper
All the peers fetch the resource
No peer caches the resource
Social cost = (α-2)+(α-2+1)x5
Optimal when peer 1 caches the resource
Optimal social cost = α+5

Social Cost and Social Optimum

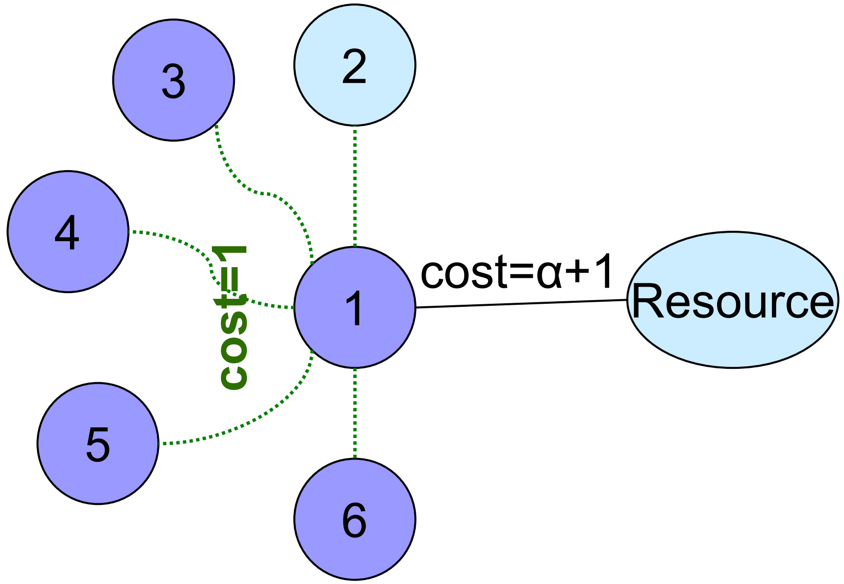


Costs:
Caching cost=α for all the peers
1: α 4: 1
2: 1 5: 1
3: 1 6: 1
Social cost=α+5 Optimal cost= α+5



Costs:
Caching cost=α for all the peers
1: α 4: α-1
2: α-1 5: α-1
3: α-1 6: α-1
Social cost=6α-5 Optimal cost= α+5

(Optimistic) Price of Anarchy



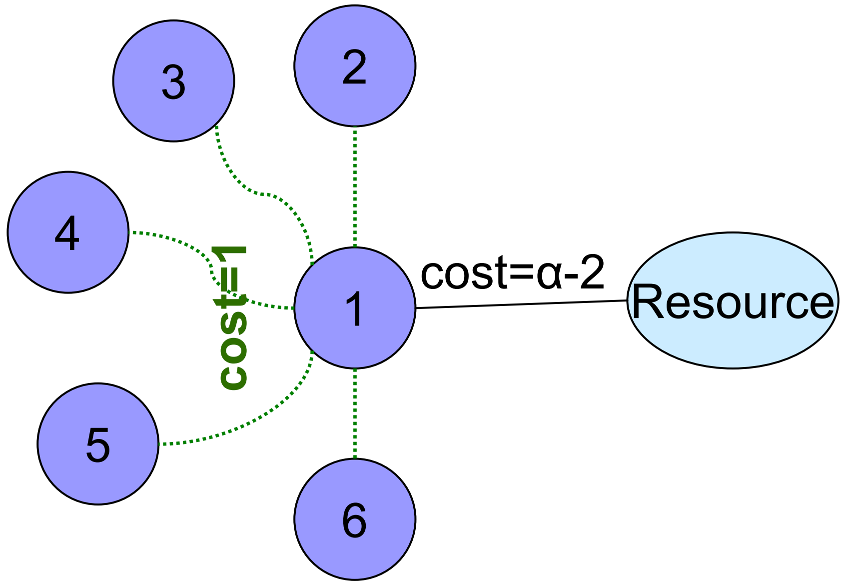
Worst case of NE:

When 2 (or 3 or 4 or 5 or 6) caches

Social cost=α+9

Optimal cost=α+5 PoA=(α+9)/(α+5)

OPoA=1 (the optimal solution can be reached)



Worst case of NE:

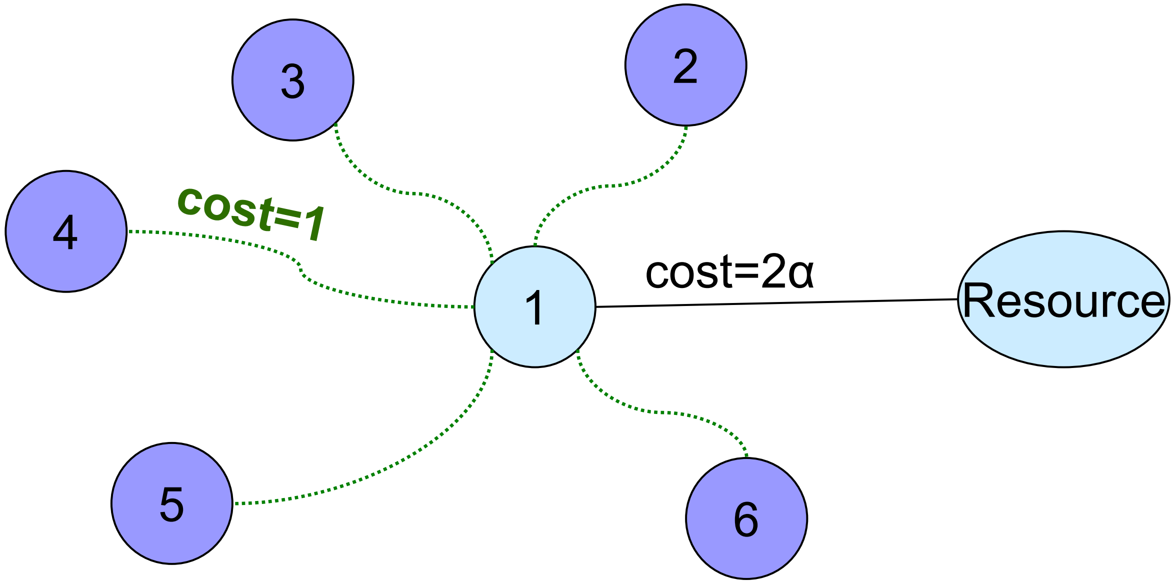
When no peer caches

Social cost= 6α-5

Optimal cost=α+5 PoA=(6α-5)/(α+5)

OPoA=PoA (the optimal solution can NOT be reached)

Costs



Placement cost (caching cost)=α

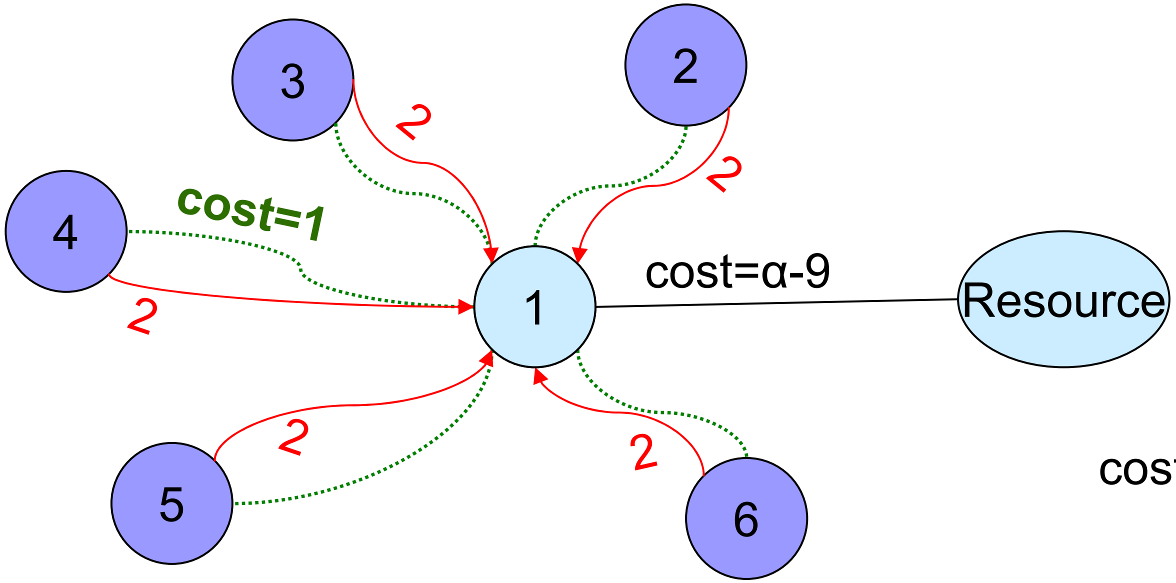
Peer 1 caches the resource

cost(1)=α cost(2)=1 cost(3)=1

cost(4)=1 cost(5)=1 cost(6)=1

Social cost=α+5

Costs with bidding…



cost to cache =α

Peer 1 caches the resource

cost(1) to cache =α – 2 – 2 – 2 – 2 – 2 (from bidding) = α – 10

cost(2)=1+bid(1)=1+2=3

cost(3)=1+bid(1)=1+2=3

cost(4)=1+bid(1)=1+2=3

cost(5)=1+bid(1)=1+2=3

cost(6)=1+bid(1)=1+2=3

Social cost=α+15 -10 = α + 5