Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

HS 2013 Prof. R. Wattenhofer

Distributed Systems Part II

Exercise Sheet 8

1 DHT

As we have discussed in the lecture, most distributed hash tables (DHTs) are built on the same idea: a binary search tree. Each leaf of the tree is represented by a peer, nodes (including the root) do not really exist. A peer knows only a small subset of all the other existing peers. But if a peer knows the address of another peer it can always contact the other peer. The network is unreliable, messages can be lost, altered or arrive out of order.

Introduce new messages, protocols, restrictions or other ideas to protect a DHT from various Byzantine attacks. If you need to make assumptions, write them down. Each answer should be about 5-10 sentences.

a) *Wrong lookup*: A search for a key roughly requires $O(\log n)$ steps. In each step one new peer is included in the search. One way to search is to send a message through the DHT. The message contains the searched key and the address of the peer that started the search. The message is forwarded from one peer to the next such that it always gets a bit nearer to its (yet unknown) destination. Once the destination is reached, the final peer answers.

A Byzantine peer sends the message either in the wrong direction, or to a non-existing peer.

b) *Incorrect routing updates*: Each peer maintains a routing table containing the addresses of about $\log n$ other peers. The peers send update messages to each other in order to keep their routing tables up to date.

A Byzantine peer sends false updates, e.g, it tries to place dead links in a routing table.

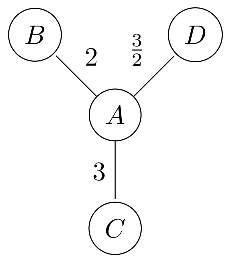
c) *Partitioning*: If a peer wants to join a DHT it has to make contact with a peer that is already part of the DHT. The new peer can then ask the old peer about other nodes of the DHT and insert itself at an appropriate place.

A Byzantine peer builds up his own private DHT by sending wrong messages to joining peers. It gives joining peers only the addresses of peers in his own isolated net.

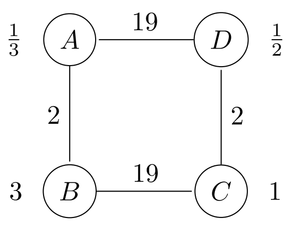
2 Selfish Caching

a) For each of the following caching networks, compute the social optimum, the pure Nash equilibria, the price of anarchy (PoA) as well as the optimistic price of anarchy($OPoA$):

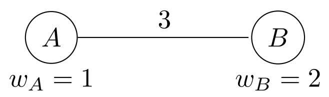
i. $\alpha = 4$, $w_A = w_B = w_C = w_D = 1$



ii. $\alpha = 10$

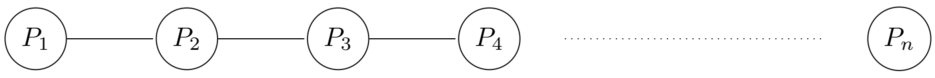


b) Write down the bi-matrix that corresponds to the following caching network given that $\alpha = 10$. (You may assume that not having access to the file incurs cost of 100.)



Compute the pure, and the mixed Nash equilibria.

c) Consider a line topology of n peers with $w_i = 1$ for all peers i, and $d_e = 1$ for all network edges e. Compute the price of anarchy as a function of the placement cost α for large n ($n \to \infty$)!



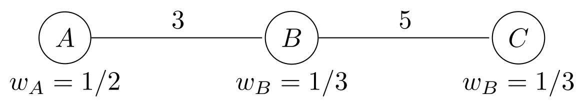
3 Selfish Caching with variable placement cost

The selfish caching model introduced in the lecture assumed that every peer incurs the same placement cost α, i.e., we assumed that $\alpha_i = \alpha$ for all peers i where α_i is the placement cost that peer i has to pay for caching the file locally. However, this is a simplification of the reality. A peer with little storage space could experience a much higher placement cost than a peer who has terabytes of free disc space available. In this exercise, we omit the simplifying assumption and allow variable placement costs α_i.

a) What are the Nash Equilibria in the following caching networks given that

i. $\alpha_A = 1$, $\alpha_B = 2$, $\alpha_C = 2$,

ii. $\alpha_A = 3$, $\alpha_B = 3/2$, $\alpha_C = 3$?



Does any of the above instances have a dominant strategy profile? What is the PoA of each instance?

b) The PoA of a class $\mathcal{C}$ is defined as the maximum PoA over all instances in $\mathcal{C}$. Let

- $\mathcal{A}^n_{[a,b]}$ be the class of caching networks with n peers, $a \leq \alpha_i \leq b$, $w_i = 1$, and $d_e = 1$ for all edges e,
- $\mathcal{W}^n_{[a,b]}$ be the class of networks with n peers, $a \leq w_i \leq b$, $\alpha_i = 1$, and $d_e = 1$ for all network edges e.

Show that $PoA(\mathcal{A}^n_{[a,b]}) \leq \frac{b}{a} \cdot PoA(\mathcal{W}^n_{[\frac{1}{b},\frac{1}{a}]})$ for all $n > 0$.

4 Matching Pennies

Tobias and Stephan like to gamble, and came up with the following game: Each of them secretly turns a penny to heads or tails. Then they reveal their choices simultaneously. If the pennies match Tobias gets both pennies, otherwise Stephan gets them.

Write down this 2-player game as a bi-matrix, and compute its (mixed) Nash equilibria!

5 P2P File Sharing

a) Tit-for-tat (T4T) denotes the strategy in a repeated two-player game to reply to the opponent's action with the same action. E.g., if the opponent previously was cooperative, the agent is cooperative. If not, the agent is not. Explain the problems that arise when we want to apply the T4T strategy to a P2P file sharing system.

b) Successful file sharing protocols such as BitTorrent exploit the fact that peers downloading the same file (or collection of files) can be organized so that they can trade different parts of the same file among each other. The set of peers that are in the process of downloading the same file f is called the *swarm* of f. If a peer wants to download a file f it enters the swarm of f, gets some trading capital, and exchanges parts of f with the other peers in the swarm.

In the following, let T_{ij} denote the number of file blocks transferred from peer i to peer j in the given P2P file sharing system. Furthermore, you may assume for simplicity that the file blocks are at any time distributed well among the peers. In particular, assume that every peer who is contacted by a freeloader always has a file block in which the freeloader is interested. Moreover, assume that the freeloader can trade any previously gathered file block b for a new one with a peer i unless it has received b from i earlier.

i. The *pairwise* Δ*-strict T4T* strategy is the strategy of a peer i in a file sharing system to only upload a file block to a peer j if $T_{ij} - T_{ji} < \Delta$. Consider a swarm of a file with m blocks where all peers trade according to a pairwise Δ-strict T4T strategy. How large does a swarm need to be at least such that a freeloader can download the entire file for free?

ii. Assume that a trusted authority keeps a global balance of each swarm participant's contribution, i.e., it announces any peer i's contribution $\beta_i = \sum_j (T_{ij} - T_{ji})$ upon request. The *global* Δ*-strict T4T* strategy is the strategy of a peer i in a file sharing system to only upload a file block to a peer j if $\beta_j < \Delta$. If all peers employ a global Δ-strict T4T strategy, how large can the file size m be at maximum, respectively how large must the swarm size n be at minimum, such that a freeloader can get the entire file for free?

iii. Explain why a trusted authority is hard to implement in a real P2P system, and give an alternative to keeping a global balance in P2P file-sharing systems!