ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

HS 2012 Prof. R. Wattenhofer / S. Welten

Distributed Systems Part II

Solution to Exercise Sheet 9

1 DHT

There are no "correct" and "wrong" answers for this task. Just answers that are better than others. The answers given here are just examples of potential solutions.

a)
- The peer starting the search sends many messages in different directions, hoping they take a different path.
- Each peer receiving the message sends an acknowledgement and the address of the next peer back to the one peer that started the search. The starting peer notices if the message gets lost or takes the wrong path. It then reacts, for example by restarting the search. The drawback of this solution are the additional connections that have to be made.
- Assuming we have authentication. Every process receiving the message sends back an acknowledgement to the previous peer. The previous peer then forwards the message to its predecessor. If a peer does not receive two times an acknowledgement (from the correct peers), then its message might have been lost. In this case the peer sends the message again, but in another direction.

b)
- When receiving an update for the routing table, the peer first contacts the new entries to ensure they really exist.
- A peer only accepts an update if received from enough other peers, for example if received $f + 1$ times.
- A peer keeps its old table for some time. Should the new table prove to be bad (e.g. messages are often delayed), then the peer switches back to the old table.

c)
- Some trusted servers are used for joining. But this would mean that the DHT is no longer fully decentralized.
- The joining peer contacts $3f + 1$ servers. These servers run a consensus-protocol. The result of the consensus is the place and the routing table of the joining peer.
- The joining peer contacts $f + 1$ servers. If they do not agree, the peer randomly chooses another set of $f + 1$ servers. This works only if f is small compared to the total number of servers.

2 P2P File Sharing

a) **Temporal imbalance.** If two peers, each of which has data the other is interested in, want to trade then at least one of them has to start sending data before it has received anything in return. It takes the risk that the trading partner does not stick to the deal, and cuts the connection. If the two peers have the prospect of continuing trading in the future,

this problem can be relaxed. However, if a peer lacks only one file block to complete its download, it has no incentive at all to return the favor after a trading partner has provided it with the missing block.

Bootstrapping. In swarm-based P2P file sharing systems like Bittorrent, a peer that starts a download has no trading capital yet when it enters the swarm. In order to use a T4T strategy a new peer has to be provided with some file blocks for free before it can start proper trading. This can be exploited by a freeloader in that it always claims to be a new peer.

Moreover, statistics of today's file-sharing systems reveal that there are typically only a few peers that publish most of the content. Thus there is an inherent, and probably desired imbalance among the users that could not be supported by a strict T4T regime.

b) i. The freeloader can receive Δ different file blocks from each peer in the swarm. Hence the swarm size has to be at least $\lceil m/\Delta \rceil$.

ii. The swarm size n does not matter. The freeloader can only get Δ file blocks for free in total. Thus, it can only download the entire file for free if $\Delta \geq m$.

iii. The P2P paradigm means that all peers are equally privileged, equipotent participants in the application. To have a central authority contradicts this understanding of a P2P system. Thus, the central authority has to be emulated by the peers themselves. Unfortunately, this is not an easy task, and most reputation system that try to implement it are susceptible to cheating. One problem is that the actions that affect the reputation—or in the case of a Bittorrent-like system, the "global" balance—are typically observed only by very few peers. For instance, if a peer A claims that it did not receive a certain file block from peer B although B claims so it is hard for a third peer C to tell which peer is lying. One cumbersome possibility is to make majority votes among a large fraction of all the peers. However this is inefficient, brings a large overhead, and is still susceptible to colluding freeloaders, or Sybil attacks. There are promising attempts that use cryptographic primitives to install a distributed trusted authority through multiparty computation.

In the realm of P2P file-sharing, an alternative to a global balance that achieves the same effect is to choose the file blocks that a peer is allowed to get for free depending on a requesting peer's Id, or IP-address. If all peers use the same rule when computing this set of free blocks a freeloader will only get these blocks. One example would be to provide a peer with IP x with the blocks in the range $[\mathcal{H}(x) \mod m, \mathcal{H}(x) \mod m + \Delta]$, where $\mathcal{H}$ is a hash function, e.g. SHA-1.