582615: Overlay and peer-to-peer Networks, Autumn 2011

Exercise 1
Example solutions

Questions

1. What are overlay networks and why are they needed? What kind of applications are they best suited for? (3 points)

An overlay network is a network that is built on top of an existing network. The overlay therefore relies on the so called underlay network for basic networking functions, namely routing and forwarding. Today, most overlay networks are built in the application layer on top of the TCP/IP networking suite. Overlay technologies are best suited for overcoming some of the limitations of the underlay, and at the same time offering new routing and forwarding features without changing the routers. Good examples of applications of overlay networks are video streaming systems such as PPLive (overcoming the missing IP multicast support in the underlay), anonymous storage networks such as Freenet (overcoming the lack of anonymity in the underlay) and Internet telephony systems such as Skype (overcoming scalability issues and bypassing firewalls in the underlay).

2. Give short answers to the the following questions: (5 points)

a) What does BitTorrent use HTTP for?

BitTorrent uses HTTP for client-tracker communication. HTTP seeding is also supported by a number of clients.

b) Explain the structure of a .torrent file. Explain the function of each part of the file.

See http://wiki.theory.org/BitTorrentSpecification#Metainfo_File_Structure.

c) What is choke algorithm?

Choke algorithm is an algorithm that each BitTorrent client runs periodically (eg. every 10s) in order to do peer selection. As a leecher, the client orders its list of peers according to the speed that the peers uploaded data to the client since the last time the choke algorithm was run. The client unchokes N (eg. 3) fastest peers from the list and chokes the rest (thus the name), the client only uploads data to the unchoked peers until the next time the choke algorithm is run. Optimistic unchoke algorithm also runs periodically, but more seldom than the normal choke algorithm (eg. every 30s). The algorithm unchokes one extra peer chosen by random and keeps that peer unchoked until the algorithm is run again. Optimistic unchoking is done in order to give new peers the possibility to get their first pieces, and in order to test the speeds of the peers to find the fastest uploaders.

d) What is the difference between the new and the old choke algorithms in the seed mode?

The old choking algorithm in seed mode works the same way as the leecher algorithm described in the previous question, with the exception that the download speed instead of upload speed is used for ordering the peer list.

In the new choke algorithm the peer list is ordered according to the last unchoke time, most recently unchoked peers first. Together with the periodical “optimistic unchoke” -like selection of one random peer to be unchoked, this leads to the situation where we can say the choking criteria to be random. More specifically, there is a queue of N (eg. 4) unchoked peers, and periodically the queue is updated by inserting a random peer at the top of the queue, and by removing the node at the tail of the queue.

e) What are the benefits and drawbacks of rarest-first piece selection compared to sequential piece selection?

Rarest first piece selection prevents rare pieces becoming even rarer, or even extinct in the network, and thus the availability of all the pieces stays fairly consistent. This alleviates the “last pieces problem”. The “last pieces problem” means that finishing a download gets delayed because it takes such a long time to get the last remaining rare pieces. “Rarest first” piece selection also means that the clients are usually interested in different pieces even if they start their download at the same time. This means that the clients will have pieces that the other clients don't have, and thus piece exchange between the clients can occur.

In sequential piece selection the pieces towards the end of the file are rare, and may easily get extinct if the seed leaves the network. These rare pieces may cause the “the last pieces problem”. If a number of clients arrive at the same time in the sequential piece selection case, all these clients will be interested in the same pieces and thus piece exchange between the clients is unlikely to happen, leaving all the uploading to be done by the seed.

The benefit of sequential piece selection is that in the case of a media file, it is possible to start viewing or listening the file before the download has completely finished. This is not possible with the traditional rarest first piece selection.