Peer-to-peer networks – (due till April 1, 2009)

Exercise 5.1:

You have been hired as pirate buster against the Gnutella protocol by the content producing industry.

1) Your costumer suggests to generate packets with a maximum value for the TTL field. What is this supposed to cause and why will it be unsuccessful?

Solution: The idea is to flood the network with packets which are duplicated again and again. As the growth of flooding without avoidance of duplicates is exponential, the network would theoretically be down very fast.

But since nodes store packet IDs, the network will not generate a large number of duplicates. Each client could also check, whether the TTL is smaller or equal to a maximum radius and discard the packet if it does not meet the standard.

2) Can you think of ways to attack the network without violating the protocol?

Solution: Some clients can pretend to offer a file by responding on every request. In fact this is done every time a query is issued to the Gnutella-network, today. Enter a random number and you will get a file with that number as filename. If you are looking for a file with a well-known hash, you can of course check the hash upon reception.

Peer-to-peer networks – (due till April 16, 2008)

Exercise 5.1:

You have been hired as pirate buster by the content producing industry.

3) The next attempt to attack could be to insert fake clients into the network. Can you think of ways how to harm the network in this case?

Solution: A fake client could flood the network with requests. It could also reset the TTL to its maximum value to increase the load of the network but this would increase the usage only to a small degree. It could discard regular requests from other users though this would be no significant problem for a flooding protocol.

Exercise 5.2: Distributed Hash Tables

1) In P2P networks, files are usually identified by their hash values. Nodes are responsible for an entire range of values. What happens if a new node joins the network or if it leaves?

Solution: The range of hash values is split at the random number of the new node. According to the convention of the protocol, the new client could e.g., be responsible for the lower part, the old node for the upper rest.

Peer-to-peer networks

Exercise 5.2: Distributed Hash Tables

2) How does a P2P system ensure that every node stores the same amount of data? Can that be ensured at all?

Solution: The split point mentioned in 1) will usually not be in the middle of the range. So a slightly unbalanced distribution of files to nodes can be expected. But due to the probabilistic nature of allocating nodes and files to the hash space, large clusters of file in the range of a single node are unlikely. A problem might, however, be the file size. A large file might be a burden for a single node.

3) Everything seems so well-balanced. Can you think of situation or modes of use of the network which are less well-balanced?

Solution: The distribution of nodes and data is relatively balanced. But some files will be very popular while others may never be searched for. A node with a popular file could be flooded with too many requests (esp. true for dial-up connections).

Peer-to-peer networks

Exercise 5.2: Distributed Hash Tables

4) A node was shutdown by a sudden power failure. How would you store information redundantly in order to be more robust against the loss of data?

Solution: The ranges could overlap. Then, one or more nodes are responsible for the same range of values and they would store information redundantly. In order to implement a simply double-node overlap, a new node N has to query the predecessor P1 and P1's predecessor P2. Then, N has to adopt the range from P2 to N. The upper limit of P2's range must be limited from the successor S of N to N itself. The lower range of S is limited from P2 to P1.

Exercise 5.3: Gnutella analysis

Install a protocol analyzer like „Wireshark“ in your own home-network and run a Gnutella client. Which kinds of packets do you see? Check whether they conform to the protocol definitions presented in the lecture.

(A demonstration was shown in the previous tutorial)