P2P Systems - Final Project

Master Degree in Computer Science,
Computer Science and Networking
Business Informatics
Academic Year 2014/2015

Simulating the Freenet 0.7 Darknet

1 Goal of the Project

Freenet 0.7 [2] is a peer-to-peer network where each user specifies which other users are allowed to connect to its peer: these peers correspond to his/her trusted friends. In this way each peer is only directly visible to its friends and this enables to obscure the participation of a peer in the network to other nodes. Peer-to peer networks that limit connections to friend-to-friend interactions are sometimes called darknets, so *Freenet 0.7* is also referred as *Dark Freenet*.

Each *Freenet 0.7* node is created with a unique, immutable identifier and a randomly generated initial *location key* whose values range in [0; 1). The identifier is used by the users to identify the friend peers, i.e. those it wants to connect to, while the location is used for routing. The identifier may correspond, for instance, to the Facebook identifier of a user.

In *Freenet 0.7* the basic routing strategy is to greedily forward a request to the neighbour whose location is closest to the key. The exact specification of the routing algorithm is presented in the slides of the course *P2P Systems*, lecture 16/12/2014. To make the routing algorithm find the data faster, Freenet attempts to cluster nodes with similar locations. Since the connections between nodes cannot be modified, the nodes periodically consider swapping their locations by evaluating whether the swap brings to a better clustering. The exact formula for deciding weather to swap two nodes is defined in the slides of the course *P2P Systems*, lecture 16/12/2014.

The project requires the analysis and the implementation of the basic functionalities of the Freenet 0.7 system, namely, node join, location swapping, content insertion and retrieval.

2 Implementation

It is required to exploit *Peersim* [1] to simulate the behaviour of *Freenet 0.7*. Peersim is a scalable simulator designed for the evaluation of large scale peer to peer systems. It is preferable to exploit the *event based* version of Peersim.

The project requires to develop a simulation implementing the following functionalities of *Freenet 0.7*:

- *join operation*: the joining peer is associated with a unique identifier and with a location key. Then it defines a set of connections to peers whose identifier is specified by the user. For the sake of simplicity, this operation must be implemented by a controller/initializer (i.e. no join logic inside the protocol). The controller must generate a unique location key for each peer p, while the identifier of the peer and those of its neighbours are present in an input file (for instance the file containing the Facebook dataset). These peers define the neighborhood of the peer p and should be saved in a data structure which is implemented by the Linkable protocol. Note that this procedure replaces the WireRandom initialization procedure exploited in the MidTerm.

- *put operation*: a peer may insert content in the network, each content is paired with a key. Each insertion request is routed and stored on a node of the system

- *content retrieval*: the user inserts a key and the system retrieves the content paired with that key.

- *periodic swap*: each peer periodically perform a swap operation whose goal is to optimize the greedy routing. Note that a swapping of the nodes implies also a swap of the content stored on the nodes involved in the swap.

Implement these functionalities, then perform a quantitative evaluation of the system. Present a set of plots evaluating:

- the average length of the routing paths

- the characteristics of the overlay, such as the clustering coefficient, the diameter of the network, the distribution of the nodes degrees.

Just as an example, a plot may report the average length of routing paths as a function of the network size, for different distribution of the keys in the network.

As far as concerns the friendship relations required to define the network, it is possible to exploit a Facebook data set, reporting the friendship relationships defined in this social network. The dataset reports an edge list where each edge defines a relation. The link to the dataset is reported on the Moodle.

Keys may be generated according different probability distributions (for instance uniform, power law,...). The generation of the content paired with the key may be avoided because not relevant for the evaluation of the functionalities to be implemented.

3 Project Submission Rules

The project must be developed individually. The material to be submitted for the evaluation is the following one:

- a pdf document reporting the code of all the JAVA classes defined to set up the simulation.
- a report (pdf document) describing the main features of the project. The report should include:
  - a general description of the project design choices
  - a set of plots reporting the outcome of the experiments
  - a description of the JAVA classes defined for the simulation

The report and the code must be submitted both electronically, through the Moodle, and at the reception desk of the Department of Computer Science. The project must be submitted a week before the date of the oral examination (if required). The oral examination will regard both the discussion of the project and a review of the topics presented in the course. The oral examination is waived for the the students who have passed the Mid and Final Term. The discussion of the project consists in the presentation of a short demo, which can be run on the personal laptop, and a general discussion of the choices made in the implementation of the system.

Do not hesitate to contact us by e-mail (ricci@di.unipi.it, alessandro.lulli@gmail.com, andreadesalve@gmail.com) or during the question time, each Thursday 15.00 PM-18.00 PM.

References

[1] *The Peersim Simulator* http://peersim.sourceforge.net/.

[2] N. S. Evans, C. Gauthier Dickey, C. Grothoff, *Routing in the Dark: Pitch Black* In Proceedings of the 23rd Annual Computer Security Applications Conference (ACSAC). IEEE Computer Society, 2007.