Peer to Peer Systems

Master Degree in Computer Science,
Computer Science and Networking, Business Informatics
Academic Year 2015/2016

Assignment (may replace the mid-term)

An Analysis of the Ping-Pong Gnutella Protocol

Deadline 30-11-2015

The goal of this assignment is to verify the understanding of the fundamental concepts of the basics of the Gnutella PING-PONG protocol. Write a simulation of the Gnutella PING/PONG protocol and compare the following strategies:

1. *basic solution, on-demand PING*: each time a peer receives a *PING message* from another peer P, it sends back a *PONG Message* containing its identifying data, and propagates the *PING* to all its neighbours. The *PING* is propagated through a *TTL-enhanced* flooding. The *PONG messages* received as a reply to the propagated *PING* are sent back to P as well.

2. *pong caching strategy, periodic PING*: each peer periodically sends a *PING* and caches the received replies. Each peer may define a different period. When a peer receives a *PING* from a neighbour, it replies by sending a set of *PONG* taken from its cache.

3. *optimized PONG caching*: when receiving a *PING*, if the cache includes less then k *PONG*, broadcast the *PING* , otherwise drop the *PING* and exploit *PONG* in the cache. When receiving a *PONG*, store it in the cache pairing it with a timestamp. Periodically, filter the oldest entries in the cache.

You can look for further material (scientific papers, slide of lectures,...) available in the network, to support the full understanding of the system.

Compare the different solutions by evaluating the number of messages exchanged in the network. In the optimized PONG caching solution, consider the behaviour of the system for different values of k.

Exploit the Peersim Simulator [1] or write your own simulation in JAVA. In Peersim, peers are represented by JAVA objects (nodes) and communication between nodes is simulated by exploiting a global message queue, which is shared by all the nodes. This allows to execute simulations with millions of

peers. Instead, if you write your own simulation, a peer can be modelled by a JAVA process running in its own instance of the JAVA Virtual Machine. All the JVMs may run on the same physical machine. JAVA processes may communicate through sockets, redirected through the loopback address. Even if this solution is more realistic, it suffers from a limited scalability (10-20 JAVA processes at most on a single machine).

The assignment is optional and it may replace the mid-term. It must be done individually and its deadline is 30 November 2015. If the student receives a positive evaluation for this assignment, he/she will the receive a final term, at the end of the course. If the evaluation of both the mid and the final term will be positive, the student will be relieved from the oral exam. Please enrol in the course, if you do not have done it yet, because the assignment must be submitted through Moodle and its evaluation will be notified through the Moodle as well.

References

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