Peer-to-Peer Systems and Security

Interim Report

Network Size Estimation

Group 41
Ankur Sinha
Saquib Shah

1. Process Architecture:

Our process architecture deploys single threaded architecture with event loop using the “asyncio” package available from the standard library of Python. We propose this methodology since it is considered to be less vulnerable and error prone, and has less overhead.

2.1. Inter-Module Protocol and Message Format:

We have finalised (from [1] and [2]) on using the methodology mentioned in GNUnet’s paper titled “Efficient and Secure Decentralized Network Size Estimation” [1]. The algorithm computes the estimation using the identity of the peer and the random key T with the matching prefixes of its own and other messages received, over the last few iterations (typically, last 64 iterations), in short.

This protocol has been chosen because it employs Proof-of-Work mechanism to maintain a good level of security preventing malicious nodes from causing huge deviations in the size estimation. In addition, it is highly efficient as it has a complexity of O(1).

The main components of the message format are as follows: *S* corresponds to the starting time of the particular round, typically rounded off to the beginning of the particular hour. Proximity *p* is calculated by the hashing of starting time and public key. Ás mentioned earlier, PoW mechanism helps in keeping the system secure from malicious nodes, if any. Signature is calculated by hashing the other fields of the message. It is used to validate the fact that the public key has been derived from the private key.

Offset	Contents
0	Message header magic code
4	Hop-Count (updated at each peer)
8	Signed data header magic code
16	Time *S* of the round
24	Proximity *p* in bits
28	Public key (2048 bit RSA)
288	Proof-of-Work
296	Signature (signing bytes 8–295)

Table 1: Message Format [1]

2.2. Authentication of Peers:

The authentication of peers is done by implementing some security strategies in the form of PoW and signatures as mentioned earlier. PoW needs to be solved for each peer to keep the malicious nodes out or minimum to an extent that they do not cause a huge deviation. In addition, signatures have been included to verify if the public key has been derived from the private key.

2.3. Exception and Error Handling:

All exceptions are handled with appropriate messages that are comprehensible such that any further changes or troubleshooting can be done with ease. In addition, in case of any severe connection issues or violation of protocols, there would be a restart of the connection.

References:

[1] Evans, Polot, Grothoff - *Efficient and Secure Decentralized Network Size Estimation*

[2] Van de Bovenkamp, Kuipers, Van Mieghem - *Gossip-based counting in dynamic networks*