

Self-Configuration Protocols for Small-Scale P2P Networks

James Z. Wang Matti Vanninen
*Department of Computer Science
Clemson University, Box 340974
Clemson, SC 29634-0974, USA
+1-864-656-7678
{jzwang, mvann}@cs.clemson.edu*

Abstract

This paper proposes several protocols, including the random, egoist, altruist, greedy and optimizer protocols, for individual peers to self-configure the P2P overlay network. Our simulation studies show that the optimizer protocol for the P2P overlay network self-configuration tries to minimize the difference between the physical network distance and the logical network distance of two nodes by creating a virtual link along the network path having the shortest physical distance in the network, is superior to other protocols in terms of the network diameter and average node eccentricity.

1. Introduction

In a large-scale unstructured P2P network, it is possible to take advantage of the network heterogeneity to improve the P2P search efficiency using Superpeers [1] or index nodes [2, 3], which are more powerful than their peers, to cache the indices of other peers and act as search-hubs. However, these approaches are not suitable for more homogeneous environments, such as workstations in a corporate network, where generally no natural super-node exists. Thus, flooding-based search is a necessity in small-scale P2P systems without index servers.

In such small-scale unstructured P2P networks, search efficiency relies on the quality of the overlay network [4], which can be determined by its diameter and node eccentricity. In general, a P2P overlay network with shorter network diameter and lower node eccentricity is more efficient for flooding-based P2P search. In this paper, we propose several protocols for individual peers to self-configure the P2P overlay network. To determine which protocol is the best for self-configuration of the small scale P2P overlay network, we evaluate these protocols using simulation by comparing the quality of the self-configured overlay networks. The rest of the paper is organized as follows. After we propose the P2P self-configuration protocols and discuss the rationale behind these protocols

in Section 2, we use simulation to evaluate the proposed protocols in Section 3. Finally we present our concluding remarks and discuss future work in Section 4.

2. P2P self-configuration protocols

A P2P self-configuration protocol is used by peers to form virtual connections with some existing peers when they join the P2P network.

2.1 P2P self-configuration

When a node initially joins the P2P network, it enters the JOIN state. A node in the JOIN state calls procedure `node_join` depicted in Figure 1 to gather the current state information about the P2P overlay network.

```
Procedure node_join  
begin  
  send INIT_REQ message to any existing node;  
  wait for INIT_ACK message;  
  if no response received, return failure;  
  if INIT_ACK message is received  
    send PING to each node in the network;  
    wait for responses until 2*T time passes;  
    record the arrival times of all PONGs;  
    set state UNSTABLE;  
  endif  
end
```

Figure 1: Procedure node_join.

In this procedure, we assume each node in the P2P overlay network maintains a view of the network. An INIT_REQ message is used for the joining node to initially query any node in the P2P network. An INIT_ACK message, containing the responding node's view of the network and the overlay network parameters (M, N, T), will be sent to the joining node, where M and N are minimum and maximum connections for a node respectively, and T is a timeout value which is normally the largest round-trip physical network latency between any pair of nodes in the

P2P network. After receiving the INIT_ACK message, the joining node pings all other nodes in the network to collect the physical network latencies, and then enters the UNSTABLE state in which it uses procedure **node_connect** depicted in Figure 2 to select at least M peers as its neighbors in the P2P overlay network.

```

Procedure node_connect
begin
  while # of links to this node is less than M
    select the best suitable node in the network;
    send a CONN_REQ message to selected node;
    wait for response until 2*T time passes;
    if CONN_ACK message is received
      establish virtual connection;
    if CONN_REJ message is received or timeout
      mark the responding node as NO_CONN;
  end while
  broadcast UPDATE message to neighbors;
  set state STABLE;
end

```

Figure 2: Procedure node_connect.

In this procedure, a node not entering the STABLE state tries to establish at least M virtual links with other peers in the P2P overlay network. It selects the best suitable nodes not marked as NO_CONN to establish the links based on the following information:

- The distance in hops to the node.
- The latency of routing messages to the node through the P2P overlay network (logical distance).
- The latency of sending messages the node through the underlying physical network (physical distance).
- The number of connections to the node.
- The capacity of the node for further connections.

When a peer is selected by this joining node, a CONN_REQ message will be sent to the selected peer to request for establishing a virtual link. A CONN_ACK will be sent back to the requesting node if the selected peer can accept more connections. The payload of the CONN_ACK message also contains the latest view of the P2P overlay network of the selected peer. If the selected peer can not accept any new connection, it will respond with a CONN_REJ message also containing its latest view of the overlay network. No response from a node means that node has probably quitted the P2P network, while a node responded with CONN_REJ usually reaches its connection capacity. The joining node will mark these nodes as NO_CONN so that no further attempt to link with them will be made. Once a node establishes M links in the P2P overlay network, it broadcasts an UPDATE message to its neighbors and enters the STABLE state. A UPDATE message contains information about all the nodes and

connections, known to the source node, which have been added or removed from the P2P overlay network. The UPDATE message will be disseminated across all nodes in the P2P overlay network. When a node is in the STABLE state, it should constantly communicate with its neighbors so that the proper overlay network is always maintained.

2.2 Neighbor selection protocols

Neighbor selection protocol determines the topology of the P2P overlay network. In this paper, the following protocols are proposed for peers to select their neighbors.

- **Random:** The simplest protocol is to select peers randomly to form virtual links. This protocol is used by original Gnutella P2P network [5].
- **Egoist:** This approach first finds the set of nodes with the largest number of connections, and then the joining node sends a CONN_REQ message to a node having the highest capacity for further connections among these highly connected nodes, hoping the joining node will be connected to the center of the overlay network.
- **Altruist:** This approach selects the nodes with the lowest number of virtual connections to try to balance the number of virtual links among the peers, hoping to bring the nodes on the edge of the overlay network to the centroid of the network.
- **Greedy:** This approach always tries to connect to the nodes having the lowest physical network latencies in order to build a low diameter P2P overlay network in terms of message routing latency.
- **Optimizer:** This approach considers not only the physical but also the logical network latency between peers. Given two nodes i and j , the physical network latency from node i to node j (LP_{ij}) is the time for node i to send a ping message directly to node j through underlying physical network; the logical network latency from node i to node j (LL_{ij}) is the time for a message to travel from node i to node j by routing through the P2P overlay network. For three peers i , j and k , we define $O_{ijk} = |LP_{ij} + LP_{jk} - LP_{ik}|$, where LP_{ij} , LP_{jk} , and LP_{ik} are physical network latencies between those nodes. It is reasonable that $LP_{ij} + LP_{jk}$ is viewed as the logical network distance between node i and node k if we build a P2P overlay network i - j - k (i.e., j is the node that connects both node i and node k). Thus O_{ijk} can be viewed as the latency overhead of routing messages using overlay network i - j - k . Furthermore, we define the cost of any node i connecting to a node j as $C_{ij} = \sum O_{ikj}$, where k is any node in the network other than node i and node j . Our optimizer protocol uses connecting cost C_{ij} as the metric for neighbor selection, i.e., the joining node selects the peer that results in the least connecting cost as its neighbor.

2.3 Message format

Three types of messages are used in our proposed protocols. The KEEPALIVE and PING messages are point-to-point message used for handshaking or physical network latency probing. The broadcast messages (query messages and UPDATE messages) are forwarded by a node to all neighbors except the incoming node. The responding messages are only forwarded along the original query path. A common message format is depicted in Table 1.

Table 1: Message format

ID	Unique ID of the message
Source	IP address of the source node
Destination	IP address of the destination node
Type	Type of the message
Payload	Payload of the message (optional)

3. Performance study

The quality of a P2P overlay network can be evaluated using the network diameter and the average node eccentricity. The diameter of a P2P overlay network is measured as the maximum latency for routing messages between any two nodes through the shortest distance path on the P2P overlay network. The network diameter should be short so that any query message can reach all nodes quickly. The eccentricity of a node is its logical network distance to the centroid of the P2P overlay network. The average eccentricity of all nodes in the overlay network should be low so that queries can be responded promptly.

3.1 Simulation model

We use a discrete event simulation to evaluate the P2P overlay networks self-configured using different protocols. Similar to the simulation models used in [3, 4], we model the node join and exit activities as a standard M/M/ ∞ queue, where an arrival in M/M/ ∞ queue represents a node joining the P2P network, the node lifetime is presented as the time spent in the queue, and the number of nodes in P2P network is assumed to be the queue length. The arrival of

nodes follows the distribution $p(\text{arrival} < t) = 1 - e^{-\frac{t}{\lambda_1}}$ and the node lifetime has distribution $p(\text{lifetime} < t) = 1 - e^{-\frac{t}{\lambda_2}}$, where λ_1 and λ_2 are mean arrival time and lifetime of the peers. By Little's Law, the expected number of nodes in the P2P network is $N = \frac{\lambda_2}{\lambda_1}$.

We assume that the minimum and maximum numbers of connections for a node are 3 and 7, respectively. The physical network latency between a pair of nodes within a segment follows an exponential distribution with a mean value of 3000 ns. The physical network latency between a pair of nodes across different segments follows an exponential distribution with mean value of 30000 ns. These latency values were obtained by an experimental

study of actually pinging network nodes in Clemson University campus network. We assume the entire network is divided into 5 segments. The simulation ran for 1000000 ticks with an expected number of nodes $N = \frac{\lambda_2}{\lambda_1}$. During each tick, nodes may arrive with a probability based on the arrival distribution, or depart if their time to live reaches zero. Without loss of generality, we assume the mean interval of node arrivals is 100 ticks. The expected number of nodes ranges from 100 to 3500, which are typical sizes for corporate networks. It is no doubt that the P2P network can reach the expected number of nodes after 1000000 ticks. We repeated the simulation for five times and measured the network diameter and average node eccentricity under various sizes of network in each time. The average results of five runs were reported.

3.2 Network diameter

In this simulation, we evaluate the quality of the overlay networks self-configured using different protocols in terms of the network diameter. The simulation results are depicted in Figure 3.

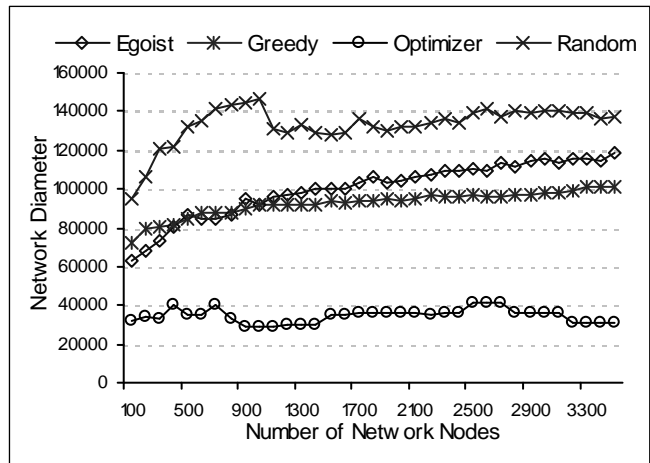


Figure 3: Network diameter under various number of network nodes.

As expected, the P2P overlay network built by using the random protocol has the largest network diameter. Both egoist and greedy approaches built P2P overlay networks with much shorter diameters. The optimizer approach is the best self-configuration protocol in terms of the network diameter. The diameter of the P2P overlay network self-configured using the optimizer protocol is shorter than half of that configured using the egoist protocol or greedy protocol. Although the optimizer approach may generate higher network traffics since more information is disseminated through the peers, the drastically reduced network diameter not only results in much quicker query responses but also reduces network traffics. Another good feature of the optimizer protocol, as shown by our simulation results, is that it creates virtual links along the

shortest physical distance between a pair of nodes. This ensures that the diameter of the self-configured overlay network does not vary too much when the number of network nodes varies as long as the underlying physical network is not changed. The results for the altruist approach are not reported because using that protocol can not consistently build a connected overlay network, resulting in infinite network diameters.

3.3 Node eccentricity

In this simulation, we evaluate the quality of P2P overlay network self-configured using different protocols in terms of the average node eccentricity. The eccentricity of a node in a P2P network is its logical network distance to the centroid of the overlay network. In a P2P network with n nodes, the mean shortest logical network distance of any node i to other nodes is $\mu_i = \frac{1}{n-1} \cdot \sum_{j \neq i} d_{ij}$, where d_{ij} is the

shortest logical network distance between node i and node j . The variance of the shortest logical network distances for node i to other nodes is $s_i = \sum_{j \neq i} (d_{ij} - \mu_i)^2$. The centroid

of a P2P overlay network is the node having the minimum variance of the shortest logical network distances to other nodes in the network. After identifying the centroid of the overlay network, we measure the logical network distances of all nodes to this center node and report the average results in Figure 4.

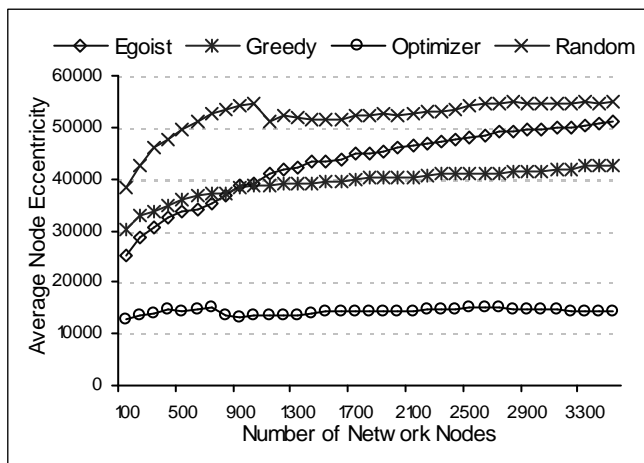


Figure 4: Average node eccentricity under various number of network nodes.

As the results for the network diameter, the random approach is the worst P2P self-configuration protocol in terms of the average node eccentricity. The egoist and greedy protocols both build overlay networks with much less average node eccentricity than the random protocol does. The results for the altruist approach are not reported because the overlay networks self-configured using the

altruist approach are not consistently connected, resulting in infinite node eccentricities.

Once again, the results in Figure 4 demonstrate that the overlay network configured using the optimizer protocol has the lowest average node eccentricity. The average node eccentricity of the P2P overlay network self-configured using the optimizer protocol is less than 1/3 of the average node eccentricity of the overlay network configured using either egoist or greedy protocol.

4. Conclusion and future studies

In this paper, we propose several protocols, including the random, egoist, altruist, greedy and optimizer protocols, for individual peers to self-configure the overlay network in small-scale P2P network environments. Based on our simulation studies, the random approach results in the worst P2P overlay network in terms of the network diameter and average node eccentricity, while both egoist and greedy approaches build much better overlay networks in terms of these two metrics. The optimizer protocol considers both the physical and logical network latencies for self-configuration. It tries to minimize the difference between the physical and logical network distance of two nodes by creating virtual links along the shortest physical network distance path in the P2P overlay network. The simulation results show that the optimizer protocol is superior to other protocols in terms of the network diameter and average node eccentricity. We are currently studying the cost of the proposed self-configuration protocols in terms of network traffic overhead for constructing the P2P overlay network. We are also investigating self-configuration protocols that require peers to cache information about only a small set of network nodes, reducing the cost of the overlay network configuration.

5. References

- [1] A. Singla and C. Rohrs, Ultrapeers: Another Step Towards Gnutella Scalability, <http://RFC-Gnutella.sourceforge.net/Proposals/Ultrapeer>, Dec 2001.
- [2] Y. Chawathe et al. Making Gnutella-like systems scalable. *In ACM SIGCOMM 2003*, pages 407–418, August 2003.
- [3] James Z. Wang, Matti Vanninen, A Novel Self-Configuration Mechanism for Heterogeneous P2P Networks, *In Proceedings of IEEE/WIC/ACM Conference on Intelligent Agent Technology (IAT'04)*, Beijing, September 2004.
- [4] G. Pandurangan, P. Raghavan, and E. Upfal. Building lowdiameter p2p networks. *In Proceedings of STOC 2001*, pages 492-499, Crete, Greece, 2001.
- [5] The Gnutella Protocol Specification v0.4. http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf.