# Ant-inspired Query Routing Performance in Dynamic Peer-to-Peer Networks

Mojca Ciglarič and Tone Vidmar

University of Ljubljana,
Faculty of Computer and Information Science,
Tržaška 25, Ljubljana 1000, Slovenia
{mojca.ciglaric, tone.vidmar} @fri.uni-lj.si

## Abstract

P2P Networks are highly dynamic structures since their nodes – peer users keep joining and leaving continuously. In the paper, we study the effects of network change rates on query routing efficiency. First, the problem background is described and abstract system model is defined. The system characteristics and behavior are analyzed and abstracted with a set of measurable metrics. The paper studies Mute query routing protocol and compares its behavior to previously suggested routing protocols. The chosen routing technique makes use of cached metadata from previous answer messages (analogy to ants laying feromone). The paper also discusses mechanisms for broken path detection and metadata maintenance. Further, simulations in various dynamic network environments are presented and discussed: the degree of network dynamics varies from one node departure and node join per ten queries generated to five node departures and joins per one generated query. Several metrics are used to clarify the protocol behavior even with high rate of node departures, but it is shown that above a certain threshold it literally breaks down and exhibits considerable efficiency degradation.

## 1. Introduction

One of inherent properties observed in P2P network topologies is their dynamics – they change all the time. While new members are joining, the others are leaving; some stay connected only for a few minutes while others only leave after weeks or even months of activity. In our laboratory, we have been researching peer-to-peer query routing protocols since Gnutella network started to deteriorate due to excessive message overhead two or three years ago. Since then, many new unstructured peer-to-peer protocols were proposed, but essentially they all suffered from either lack of anonymity or ineffective search strategies. In our recent research we have proposed new query routing strategies to reduce the cumulative query traffic without degradation of end user's experience, particularly average system response times, however by now we did not focus on anonymity issues.

Outside of the research community, new P2P clients and protocols are emerging almost daily, some of them only to disappear again in a few months while only a few are able to attract enough users to keep running longer. Besides an effective user interface, a typical winner provides at least some kind of anonymity that should protect users and the whole system from interventions by RIAA and like organizations. One of them is Mute [10], a novel ant-inspired unstructured peer-to-peer system, which uses its own virtual addresses instead of internet addresses and avoids direct contact between downloader and uploader.

The paper studies Mute's ant-inspired query routing protocol and compares its behavior to our previously suggested routing protocols. A detailed analysis in Sections 2 and 3 reveals that although there are some differences, the Mute routing essential rules are comparable to our choosy routing. The system characteristics and behavior are analyzed and abstracted with a set of measurable metrics in Section 3. Further, Section 4 presents the course of simulations in a range of dynamic network environments: the degree of network dynamics varies from one node departure and node join per ten queries generated to five node departures and joins per one generated query. The paper concludes with the discussion of the results and their meaning for future P2P systems implementations.

The main purpose of the paper is to show that Mute essentially uses choosy routing and by means of

1-4244-0054-6/06/$20.00 

simulations predict how will the persistence of its users (i.e. the network dynamics) influence its routing performance. We believe that the paper will be of interest to all the readers who want to know more about peer-to-peer systems structure, architecture and mechanisms, as well as those implementing such systems and finding their own ways of achieving better response times and higher degree of anonymity.

## 2. Background and related work

In unstructured peer-to-peer networks all the peer nodes have equal roles and functions. Two-layered systems are also popular: here, some nodes (usually stronger or with better communication links) are superior and act as proxies for regular peers or subordinate nodes. Ordinary nodes can only connect to one or a small number of the supernodes, and the supernodes communicate with the rest of the overlay on their behalf since the subordinate nodes can only send queries to their supernodes. The subnetwork of super nodes can also be viewed as an unstructured peer-to-peer network if we consider a supernode with the whole set of subordinate nodes as a single node. Basic mechanism for query message routing is usually based on flooding, which is robust and reliable but also exhibits high redundancy and creates very high network load (as first observed in the Gnutella file-sharing network [1]). More effective routing strategies are based on locally saved routing metadata from previous queries and answers.

Another category of peer-to peer systems are the so-called structured systems (for example systems based on distributed hash tables – DHTs), where certain rules about the overlay structure and the file placement exist. The peer nodes can not freely decide which files to store and where to connect to the overlay. Because of these rules the whole system can operate more efficiently than an unstructured one, but the users usually prefer the latter since they want to keep control over their hard disks. The unstructured systems pose their own problems and challenges which are different from the problems in the structured systems. This paper focuses on routing in unstructured systems.

While P2P network overlay changes, the metadata becomes obsolete and should be removed or replaced. However the processes of obsolete metadata detection and establishing new routes take time and the in this paper we suggest some directions for its estimation.

Detailed description of the choosy routing protocol can be found in our previous papers [2], [3] and [9], together with the discussion of related routing techniques. P2P overlay network topology and its properties (power law and small world) were researched by several authors, among others in [4], while other P2P-related issues can be found in [6-8]. In [5] Bu and Towsley suggest a topology generator which produces suitable topologies and which we have also used in our simulations.

MUTE File Sharing system [10] is a new peer-to-peer network that provides easy search-and-download functionality while also protecting its users' privacy. Its routing mechanism is inspired by ant behavior. When ants search for food, they mark their trail by laying feromone. When an ant finds food, it follows its own trail back to the anthill. When other ants run into the feromone trail, they give up their own search and follow the trail. The more ants walk the trail, the more feromone the trail receives and ants tend to follow the strongest scented trail. If an ant without food follows the trail and comes back to the nest, it turns around and walks in the opposite direction. Mute system uses ants philosophy for file search, except that it adds the sense of direction to the "feromone" metadata. Now the simplified rule set employs only two rules:

- If not carrying food, walk on food-directed trail or randomly. Mark trail with nest-directed feromone.
- If carrying food, walk on nest-directed trail and mark ground with food-directed feromone.

In a peer-to-peer network, the trail represents a path – a set of nodes through which a message (query or answer) need to travel. Each node only knows to which of its neighbors the message should be sent to reach a certain destination. Nodes can not see a complete network overlay topology neither its parts. Message routing can only be performed using metadata stored locally – our feromone trail.

## 3. Model and simulations

Every peer node shares a set of files. File $F_i$ is described by its metadata $m_i$, a set of metadata elements: $m_i = \{k_1, k_2, \ldots\}$: name, type, size, keywords, hash etc. Query $Q_i$ is a message, identified by a globally unique identifier and containing a subset of available metadata elements: $Q_i = \{k_m, k_n, \ldots\}$. The available files are not equally popular. A measure of file popularity $q_i$ [9] is defined as percent of queries looking for file $F_i$ (i.e. matching its metadata according to some matching function). With the term repetitive query we refer to subsequent queries with a positive match to the same file, which does not necessary mean that the repetitive queries contain same keywords and/or other metadata elements.

Each node is able to generate query messages about the files, receive and forward query messages from other nodes, generate answers, and receive and forward answer messages from other nodes. Each node is able to generate and store metadata on received messages. Messages can

only be passed on to one or a subset of neighbors, chosen by a routing mechanism. Answers return to the query originator over the same path. When a matching file is found, the node generates answer message containing complete file metadata.

### 3.1. Choosy query routing with or without metadata exchange

Each node that passes on an answer message also caches the metadata and the neighbour ID in order to use it for routing later when a similar query is issued elsewhere in the overlay. This is a formal description of an ant, carrying food and following the trail back to the nest. The cached metadata represents feromone, marking the food direction for other ants. However the cached answers are never passed back to another node again - they are merely used to route the queries.

If the nodes could send all their metadata to their neighbour nodes in appropriate time intervals, this would help them build efficient routes in the overlay faster. When no known route exists, the query still needs to be flooded.

When storing query and answer metadata as routing info, the nodes also keep track of the time needed from forwarding a query until the answer message came back. Metadata on forwarded queries represents feromone, marking the direction back to the nest – these will be needed by the forthcoming answer messages. Further, when routing next query over the same route, a node should estimate when an answer is expected to come back, allowing some extra time for unpredictable delays.

If the answer message does not arrive within that time, the path is considered broken. Therefore, a query flood should be triggered. Since the nodes still keep track of query GUIDs (globally unique identifiers), the flood is not multiplied at each node - each node forwards each query only once (in each direction). If at the node X a query was routed to the neighbor A earlier, now it is sent to all other neighbors of X with exception of the neighbor A and the originating neighbor node. This way, the query achieves the best response time possible and also reaches the same set of nodes as if it was flooded from the beginning.

In the ants world, a broken path implies that an alternative path or food source should be found and ants - scouts are sent out again in all directions.

A small difference between Mute routing and choosy routing is in the meaning of home-direction feromone or, more formally said, metadata created by the nodes after forwarding a query. In Mute's jargon, a message traveling from Alice to Bob leaves a trail of clues "To get to Alice, use this path" on the intermediate nodes. In choosy routing, a query message $Q_i$ leaves a trail of clues "To get home, answer to $Q_i$ should use this path". This kind of metadata is less meaningful but nevertheless it gets each answer home safely. However it cannot be used by consequent queries or answers and can therefore get erased after we are done with the query.

Another difference is the mechanism of limiting the query depth. Our choosy routing limits the query lifetime by a well-known TTL mechanism, while Mute expands it with a rather complicated Utility counter scheme. Although its use makes sense, it does not affect the results presented in the remaining of this paper.

The simulations, described in Section IV, were performed using choosy routing with metadata exchange. Although Mute does not employ such a mechanism, the simulation results are suitable if we ignore the number of metadata exchange messages in the total network traffic, since our metrics mostly relate to the stable system state. In a stable state, the majority of nodes are configured, which means that they can route most of the messages. In this case, most of metadata exchange messages do not bring new routing information to target nodes, with exception of newly joined nodes which are not configured yet. However we still have to stress that introducing metadata exchange mechanism is an open chance to further improve Mute's efficiency.

One could debate that we should explore the idea of similarity of queries, files, or routing paths. However if we carefully consider the protocol, we can see that these are not the issue as long as the matching function is well defined. If two queries are similar, they will have a positive match with overlapping sets of files, but nevertheless each query will have to reach the target node independently. The same is true for similar files: they will give positive match with overlapping queries but at last each query will choose the nearest matching file. The route similarity is not the issue because the query always chooses the best path and when it is not available any more, the query is flooded and thus finds the new best path.

### 3.2. Metrics

The simplest metric from the system's point of view is the total number of message hops (HT) in the whole simulation period (including cold start), however more relevant is the average value in the stable system state (HS), when the routes are already configured.

R is the average number of nodes reached by a query and M is the average node load (the number of forwarded query messages).

A somewhat modified definition of a query price from

[9] is

$$C=\frac{Total\ query\ hops}{Nodes\ reached},$$

while in [6] the percent of redundant hops P and query efficiency D are defined as

$$P=\frac{Total\ hops\ per\ query-R}{Total\ hops\ per\ query}\ \text{and}\ D=\frac{All\ query\ hops}{Effective\ hops},$$

where a hop is effective when it reaches a node with the matching file.

Another group of metrics are user-related: the number of time intervals (AT) before an answer is received, the number of hops from answer node to the source node (AH), and a share of answered queries (QA) for the queries where an answer can be found within the TTL radius (our routing with metadata exchange should always find such an answer). For better clarity, all the metrics are reviewed in Table 1.

Our goal is to achieve the values of C, P, M, R, HS, HT, D, AT and AH as low as possible, and still have QA close to 100%.

| METRIC | MEANING | DESIRED VALUE |
|---|---|---|
| C | Query price | Low |
| P | Number of redundant hops | Low |
| M | Average node load (number of forwarded queries) | Low |
| R | Average number of nodes reached by a query | Low |
| HS | Average number of hops per query in a stable state | Low |
| HT | Total number of query hops (including cold start) | Low |
| D | Query efficiency | Low |
| AT | Answer time (number of time intervals from issuing a query to receiving the first answer) | Low |
| AH | Answer hops (length of path from answer node to source node) | Low |
| QA | Percent of answered queries | Close to 100% |

**Table 1: The metrics describing system behavior.**

## 4. Routing in a dynamic environment

Yang [7, 8] implies that in common P2P file-sharing networks, the rate of query generation is roughly 10 times higher than the rate of network changes. For observations of degradation in routing efficiency, node departures are most important. However to keep the overlay connected and stable over longer simulation runs, as well as to capture real system properties, we also introduce new nodes into the topology. New nodes are joining at the same rate as the old ones are leaving. Let us define network change rate *NR* as the average number of node joins or departures per query. For example, *NR* = 0.1 means that one node joins and one node leaves the overlay after ten queries are issued.

In our simulation environment we generate one query per time step and to evaluate different routing techniques we have used *NR* = 0.1, however here we want to compare system metrics with higher *NR* values.

Our expectations are as follows. As long as the dynamics is within certain limits, its effect on the network behavior will be negligible. But when the NR will go over the threshold value, the average query traffic (i.e. the number of hops per query) will grow rapidly. Many paths will be broken and the subsequent queries will have to be partially flooded after the original path will not return the expected answer. So we can be pretty sure that the traffic (HS) will go up and average response times (AT) will grow to some extent, while we cannot be sure about the answer distance (AH) – it may as well stay within the boundaries of previous average values. With more query floods, node load (M) will grow together with redundancy (P), while query efficiency will get worse.

## 5. Results and discussion

Our first observation was that there were no significant changes when we varied NR step by step from 0.1 (one node departure / join per ten queries) to 1 (one node departure / join per query). Only when we used NR of 5 (5 node departures / joins per query) we observed the expected degradation of routing efficiency. Figure 1 represents the most obvious degradation – the increase in average number of hops per query. Figure 2 shows the average response times, which are slightly elevated due to the delayed floods, while in Figure 3 we can see that even the percent of answered queries decreases with very high values of NR, possibly because the overlay can sometimes fall apart into two or more disconnected components and some files are not accessible any more.

Figure 4 shows the average node load (M) – the number of query messages to be passed on to the neighbor nodes per time unit (i.e. per one query generated in the overlay). As soon as a few floods are necessary, the

average load quickly increases. Sometimes even more important is the maximum node load, because it can cause congestion and drastic decrease of response times. In our case, the maximum node load is over 1200 for NR = 5, while for other values of NR it hovers around 800.

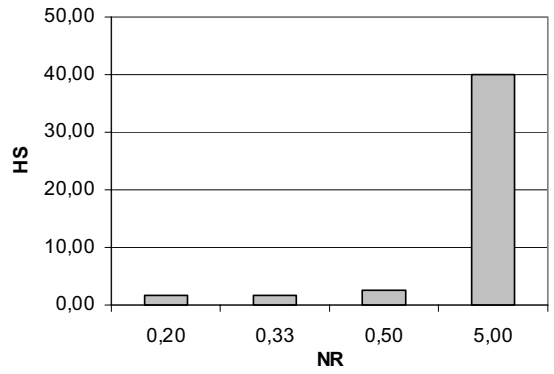


**Figure 1. Average number of hops per query in a stable system state (HS), for different values of network change rate (NR).**

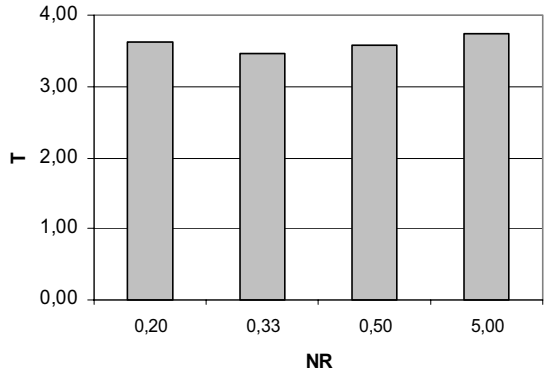


**Figure 2. Average response times for different values of NR.**

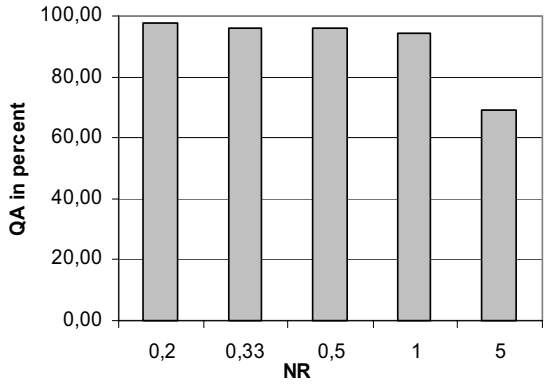


**Figure 3. Percent of the answered queries for different values of NR.**

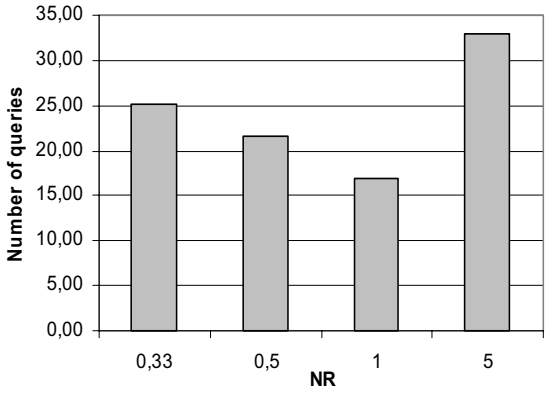


**Figure 4. Average node load (M) for different values of NR.**

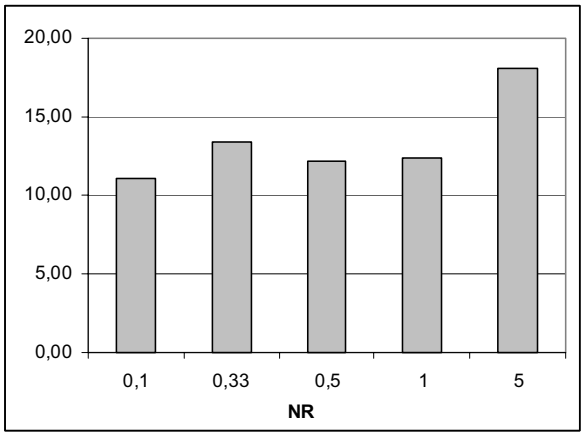


**Figure 5. Query efficiency (D) for different values of NR. Lower values mean more efficiency.**

Figure 5 shows the values of query efficiency - D. Since D is defined as the ration between the number of all query hops and the number of efficient query hops (those that reach the node with matching file), higher value of D shows that there was more query hops for needed for reaching one node that could generate an answer. We do not show graphs for other metrics since they are quite similar to those presented above.

The results obtained from simulations are expected and confirm our assumptions about system behavior. What surprised us a bit is the threshold value. In a real P2P system, users usually join when they want to get some new files. There are also the altruistic users, who stay longer and share files also when they do not need anything from the system. But in the worst case, when every user is

selfish and just joins to make one query and then disconnects (in this case NR is exactly 1), the network behavior stays below the threshold value. This is very good news since it tells us that the routing protocol is robust enough and can be used even in the most dynamic environment.

## 6. Conclusions and further work

In the paper, we presented the problem of dynamic nature in P2P network overlays. With focus on a novel Mute peer-to-peer system, we compared its ant-inspired routing to our previously suggested routing schemes. We briefly described the abstract system model, both routing mechanisms and their similarities as well as differences, the mechanism for keeping metadata up-to-date and a set of metrics, describing the system behavior under certain routing protocol. Further we explained the issues related to the network dynamicity and illustrated the effect of the system nodes leaving and new ones joining as a basic property in a highly dynamic environment.

By means of simulation we have confirmed our assumptions. The main finding, based on the simulation results, is that Mute routing mechanism is robust and predictable within the boundaries of NR that can be expected in the common P2P file sharing networks. In the future, we will do our best to include in our simulation program also those Mute features that seemed less important at the first glance and repeat the whole set of simulations. We are also planning to expand the simulation environment to the level where any distributed search protocol could be easily plugged in and simulated or compared to other protocols.

## References


[1] M. Ripeanu, I. Foster, A. Iamnitchi, "Mapping the Gnutella Network: Properties of Large-Scale Peer-to-Peer Systems and Implications for System Design". *IEEE Internet Computing*, Vol. 6(1), 2002.

[2] M.Ciglarič, "Content networks: distributed routing decisions in presence of repeated queries". *Int. j. found. comput. sci.*, 2004, Vol. 15, no. 3, pp. 555-566.

[3] M.Ciglarič, Towards More Effective Message Routing in Unstructured Peer-to-Peer Overlays, *IEE Proc. Communications*, 2005, Vol. 152, No. 5, pp. 673-678.

[4] M. A. Jovanovic, F. S. Anexstein, K. A. Berman, Modeling Peer-to-Peer Network Topologies through "Small-World" Models and Power Laws, *Proc. IX. Telecommunications Forum TELFOR*, 2001.

[5] T. Bu, D. Towsley, On Distinguishing between Internet Power Law Topology Generators, Proc. INFOCOM 2002.

[6] Q. Lv et al.: Search and Replication in Unstructured Peer-to-peer Networks, Proc. 16th ACM Intl. Conf. Supercomputing ICS'02.

[7] B. Yang, H. Garcia-Molina, Comparing Hybrid Peer-to-Peer Systems, Proc. Very Large Databases VLDB, 2001.

[8] B. Yang and H. Garcia-Molina, Efficient Search in Peer-to-Peer Networks, ICDCS 2002. http://dbpubs.stanford.edu/pub/2001-47.

[9] M. Ciglarič, Problems in Unstructured P2P Systems, to be published in Electrotechnical Review, 2006.

[10] Mute system homepage: http://mute-net.sourceforge.net/, December 2005.