Adaptive Routing in Distributed Decentralized Systems: NeuroGrid, Gnutella & Freenet

Sam Joseph Ph.D., NeuroGrid

Introduction

NeuroGrid is an adaptive network system developed by the author, and was introduced in an earlier study[2] that attempted to compare NeuroGrid with some other well-known Peer to Peer (P2P) systems, Gnutella and Freenet. However, inaccuracies in the simulation meant that the FreeNet (freenet.sourceforge.net) simulation was flawed, and this paper presents results based on a more accurate model of how Freenet operates. The previous study focused on static network designs with different connectivity patterns and routing algorithms. This type of approach is appropriate for systems such as Gnutella (gnutella.wego.com) which do not make extensive use of adaptive features. Freenet and NeuroGrid are two P2P systems that do, each successive search changing the connectivity of the system, and the knowledge that each node possesses about the contents of other nodes. Freenet also features an aggressive caching system which has the additional benefit of creating multiple copies of frequently requested data. This paper looks at some simulations of NeuroGrid and Freenet.

P2P technology is not new, but the term is new and recent applications in systems like Napster and Freenet have drawn attention to the area of decentralized computing. It is not clear that we yet have a useful definition of P2P, but "not client-server" could be a good starting point, one of the most important concepts being that all computers in a P2P network should have both server and client functionality. P2P search tends to differ from traditional routing in that it is not so much concerned with ensuring point to point communication [10] as resource discovery. P2P is related to work in ad-hoc network formation [11], and the properties of the Freenet system were explored in [3], which highlights the relation to similar work on request proxying in [12]. Work in the area of web-caching [14], [1] is strongly related to Freenet, although Freenet steps round many of the constraints considered in the field by assuming that data can be cached on any machine. The Whois++ system [4] is related to the NeuroGrid system, in that it provides a mechanism for forwarding queries to distributed servers on the basis of the content of those servers, but appears to lack NeuroGrid's adaptive node discovery mechanism whereby every successful query improves the overall system routing ability.

The question of how this work relates to agent systems, particularly mobile agent systems, is open to debate. Agents are often assumed to operate in distributed environments, and as such the ability to be able to locate desired resources is clearly of value. One could go further, and suggest that all the nodes in a p2p network such as NeuroGrid are themselves agents, in as much as they learn from experience, communicate with one another, even behave "autonomously" [6]. However, there does not seem much to be gained from applying such a label, so let us consider the elements of the p2p network simply as nodes, leaving any relevance to agents down to the general needs of any "agent" that must operate in a distributed environment. For a review of distributed search and its relevance to agents see [5].

This document is structured as follows. First of all we shall take a more detailed look at the mechanisms behind three different P2P networks, Gnutella, NeuroGrid and Freenet. Simulation results are presented for both NeuroGrid & Freenet, with various simulation extensions being introduced and explained. The document closes with a consideration of the various issues raised and pointers towards various future studies.

Gnutella

Gnutella is P2P network system that relies on broadcast search in order to locate items. One of Gnutella's innovative features is the way that it discovers new nodes as a by-product of the search process, and subsequently uses this information to generate new connections when existing ones fail. In this sense Gnutella is highly adaptive, each node connecting and reconnecting as necessary. However the search process itself is a simple broadcast search (see fig. 1), in that a node will forward a search query to all connected nodes, unless it can provide a match itself (fig.1; N005), or the message's Time-to-live (TTL) counter has reached zero (fig.1; N004). Additionally each message includes a globally unique ID (GUID) and nodes maintain a list of recently seen GUIDs that allow them to screen out messages they have already seen and thus prevent loops (fig.1; N007).

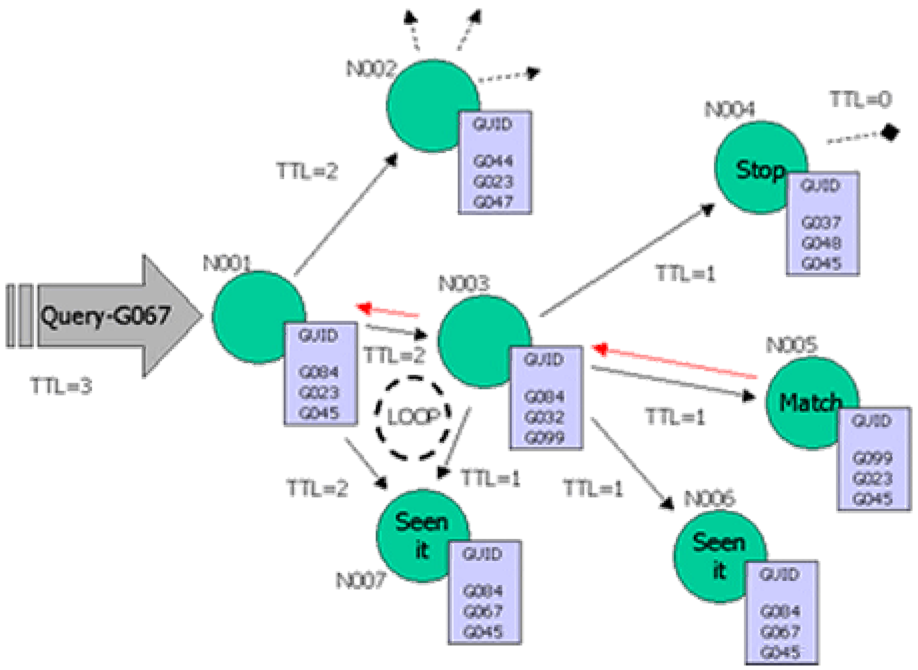

Figure 1. Gnutella Broadcast Search, the GUID tables are shown holding individual search identifiers that prevent loops. A search query would consist of some set of keywords which would be matched against each nodes knowledge base in order to determine if a match was present. The nature of the matching algorithm is left to the discretion of individual nodes.

Each node also maintains a table of where search queries came from so that search responses can be routed back along the path they came from. As is clear from fig.1, the broadcast search operates in a parallel fashion, the query spreading through the network to the extent that connectivity and TTL will allow. The query will be stopped when a match or possible loop occurs, but regardless of matches occurring in one location, the search will be fully propagated in others (fig.1; N002). Some Gnutella simulation results can be found in [9].

NeuroGrid

NeuroGrid was originally designed as an alternative routing model for Gnutella, and the simulations presented in this paper test NeuroGrid as a routing methodology for a P2P network. The current NeuroGrid implementation, http://www.neurogrid.com, takes a slightly different approach in that searches are not automatically forwarded in parallel, and rather presents search options to the user (for a description see the protocol document [7]). Nonetheless, the NeuroGrid routing methodology is applicable in both sorts of system. Each NeuroGrid node facilitates search of the network by forwarding queries to a subset of nodes that it believes may possess matches to the search query. NeuroGrid operates under the assumption that objects in the network (e.g. documents) are referenced by a number of 'keywords'. Each node maintains a knowledge base of keyword-node associations that are based on the nodes belief about the contents of remote nodes. So, for example, given that a node receives an incoming search consisting of keywords A, B, and C, the node will consult its knowledge base and retrieve any remote nodes that are associated with these keywords. The nodes retrieved from the knowledge base are ranked depending on the degree of match to the search query; there are a number of possible matching algorithms of which one is described in the NeuroGrid whitepaper [8]. The top M matches are selected and the query is forwarded to these nodes.

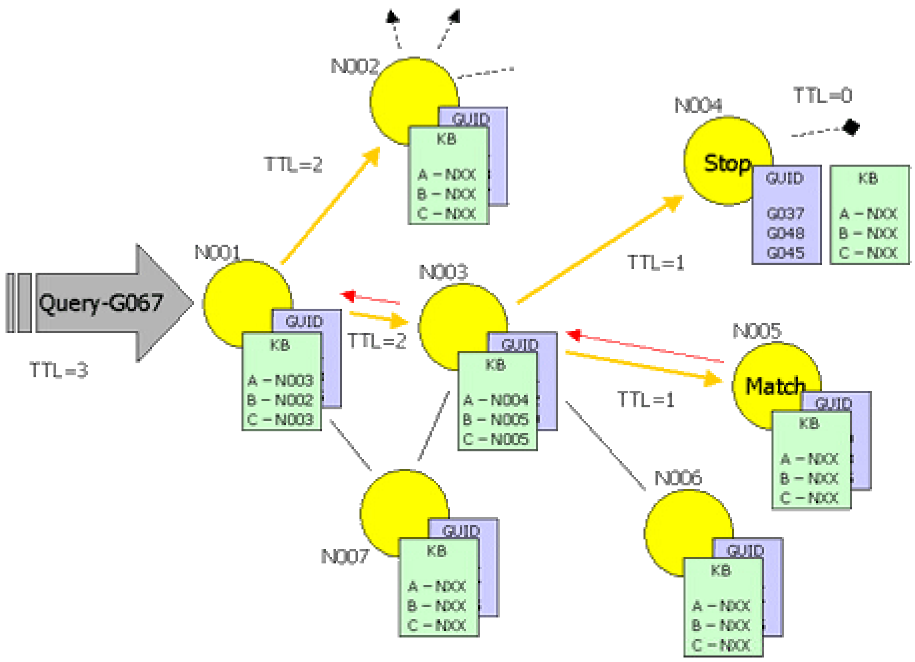

Figure 2. NeuroGrid Knowledge-based Search, in addition to the GUID tables, individual nodes knowledge bases (KB) are shown. Note that the query is now not forwarded down all possible connections, but only down to a subset of possible remote nodes (shown by the gold arrows). Otherwise the system is conceptually similar to Gnutella.

So for example, looking at Figure 2, we see that Node N001 will generate a subset of nodes N002 and N003 for a query with terms A, B, and C. Node N003 generates a subset of nodes N004 and N005, and thus the query reaches node N005 where a match is obtained. Note that the number of nodes chosen from the recommended subset is an adjustable NeuroGrid parameter, which gives the equivalent of the Gnutella routing procedure if set equal to the number of connections present at each node.

NeuroGrid's search procedure will be effective to the extent that nodes possess knowledge bases that reflect the distribution of documents through the network itself. A simple way to set this up initially is to give each node knowledge about the contents of its immediate neighbours. NeuroGrid nodes also utilize the results of searches in order to update their knowledge bases and add new connections to the nodes that provide results to search queries. The best analogy is to think of the nodes as humans, that know something about what their friends know about, and when asked can put you in touch with a friend, who may well be able to put you in touch with a friend who ... and so on. Simulation results for NeuroGrid, based on the idea that a network that has undergone multiple searches will consist of nodes linked on the basis of shared content can be found in [9].

Just to emphasize the point, the **adaptive** aspect of NeuroGrid comes in to play when a search is successful and the querying node updates its knowledge base to associate the remote node (that supplied the matching document), with the document keywords. The querying node also establishes a direct link to the remote node, adding to the systems initial random connections, leading to a gradual increase in connectivity. The important side effect of this is that all of the nodes in the system gradually become a little smarter, being able to route queries to more and more nodes. One might expect that this would ultimately be very expensive, with connections and knowledge bases growing indefinitely; however it is not the problem one might think as we shall see in the simulation results.

Freenet

Freenet is another well-known P2P system that uses a steepest ascent depth first search. Freenet has many features such as information producer & consumer anonymity, encryption security, and aggressive caching functionality. In this paper, we focus on the search functionality. In Freenet every document (item) in the network is associated with a single key generated by a hash function that should make each key unique to a particular document. Each Freenet node also has a knowledge base that stores a list of relationships between keys and nodes. Freenet routes queries by creating a ranking of the different keys that it knows about, in terms of how similar they are to the search key, and then forwards the query to the node that is associated with the most similar key (fig.3; N001). If this does not lead to a match (fig.3; N002) it forwards the query to its second choice, and so on, thus the search proceeds in a greedy serial fashion.

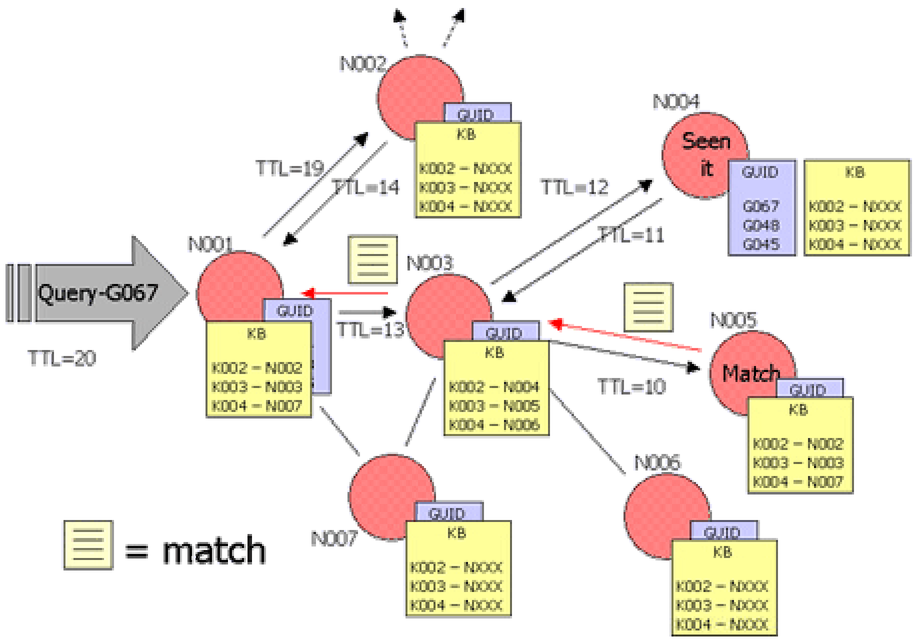

Figure 3. Freenet Search, in addition to the GUID tables, individual nodes knowledge bases (KB) are shown. Freenet maintains GUID tables and uses the message TTLs in a similar manner to other P2P architectures.

When a match is discovered, the search request is returned along the query path, along with the matching item itself. The matching item is cached, or rather replicated to all the nodes along the route of the query response (fig.3; N005, N003, N001), and in order to ensure anonymity, any of the nodes along this route can choose at random to claim that they are the source of the matching item. The original node will then update its knowledge base such that the search key becomes associated with the node that claimed to be the source of the match, adapting system knowledge and increasing the system connectivity.

Simulation

A simulation of the NeuroGrid and Freenet systems was performed. The networks both contained 1000 nodes, and each node was randomly assigned 3 documents from a pool of 1000 documents. In the Freenet simulation, the documents themselves were each associated with a single numerical key, the document being represented by the square of the key. In NeuroGrid each document was randomly assigned 2 keywords from a pool of 200. In both simulations each node was connected to three others that were selected at random, and each node received information about the keys held by their neighbouring nodes. For Freenet the initial TTL was set to 20, while for NeuroGrid the initial TTL was 7 and the forwarding subset was set to 2. Both simulations were run for 40,000 searches for random documents, executed in sequence starting from a randomly selected node that did not contain the document in question.

The current study was designed to allow a preliminary comparison to see whether the adaptive aspects of the two approaches would converge to a stable state. After 10000 searches both networks had converged to a stable state in terms of the TTL of first match, and number of messages generated per search. Performance on various measures were recorded throughout the study and the top two graphs in figure 4 show how the number of messages generated by each search changed as the networks adapted to the continuing search activity.

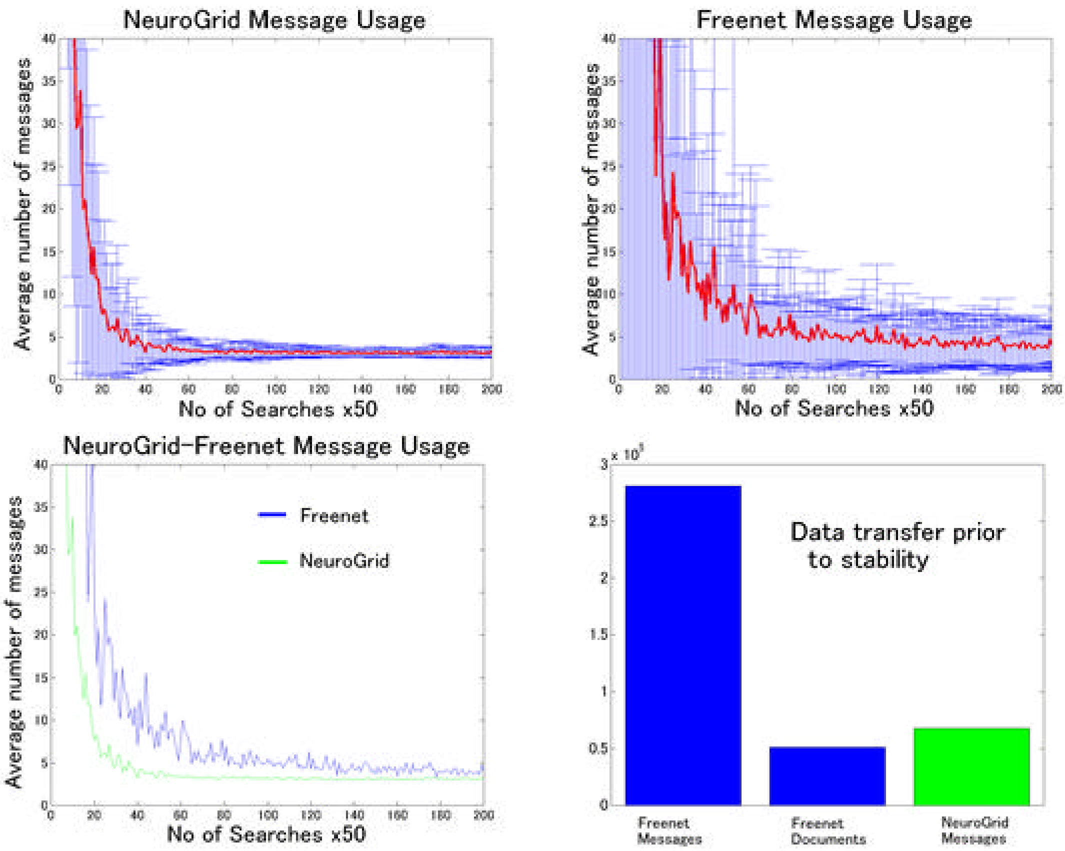

Figure 4. Top two graphs show messages generated by searching averaged over blocks of 50 searches. The x axis indicates the number of searches carried out so far. The error bars are single standard deviations. The bottom left

graph shows the average message usage of the two systems without error bars, and the chart on the bottom right shows the amount of data transfer that took place prior to network convergence.

Interestingly both NeuroGrid and Freenet simulations converge to a similar number of messages per search. The error bars in the top two plots of figure 4 show that initially the number of messages required to complete a search is erratic, but gradually the averages stabilize leaving both NeuroGrid and Freenet requiring approximately 3 messages to complete any given search. The two are compared in the bottom left plot, and while NeuroGrid appears to stabilise more quickly, it is clear from the error bars in the top two graphs that it is difficult to separate the two. The bottom right graph shows the total number of data transfers in the networks prior to their convergence. Both NeuroGrid and Freenet have settled down after 10000 searches and the data transfer figures are the sum of the data movements up to search 10000.

It is no clear that it is entirely appropriate to compare the two models in this way. The two networks contain the same number of nodes, and start with the same random connection pattern, have 1000 unique documents, and have the same starting distribution of documents. However each of Freenet's documents is described by a single unique key. In NeuroGrid each document is described by 2 keywords taken from a pool of 200 keywords. One can postulate that NeuroGrid would not perform so well (in terms of stabilizing with fewer data transfers) if each document was described by a single unique keyword, or if the keywords of a document rarely overlapped, since NeuroGrid relies upon overlapping keywords to route effectively.

The next two graphs show the number of nodes that NeuroGrid and Freenet query over time. Initial searches cover at least a tenth of the overall network , but this falls off in both cases until searches lead to the query of only 3 or 4 nodes, the search being routed directly to a node with the right data. For Freenet new data is inserted based on the key relationships, and special announce messages are used to insert new nodes. There is currently no such structure in place for NeuroGrid which implies that it may be difficult to break a stabilized network out of the various cliques that will form around particular subject areas. Further study is required to see how well NeuroGrid can adapt to the addition of new nodes and data.

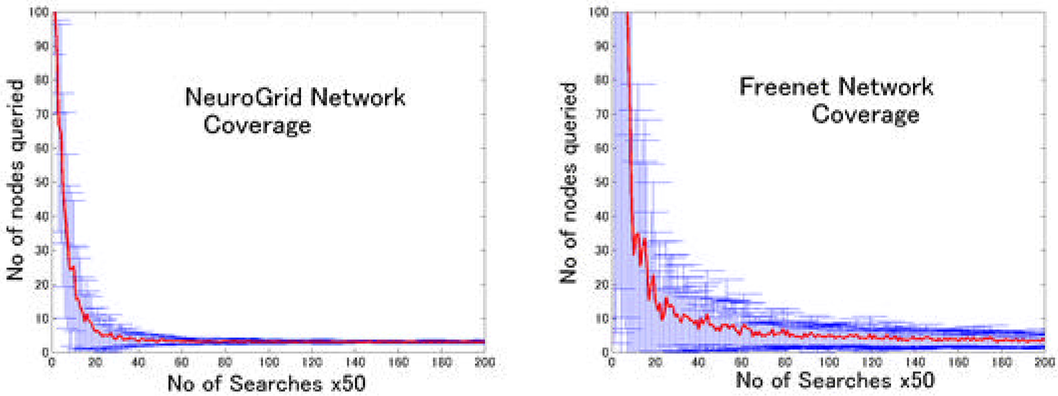

Figure 5. Coverage of network during search, i.e. no of nodes reached by any single query.

The two graphs below show the average TTL values of queries when a match is discovered. Predictably, initial searches have very low TTLs since a large number of nodes must be queried before a document is found. As time goes on the network adapts and is able to route queries more successfully, and TTLs converge to 6 in the case of Neurogrid and 18 for Freenet. All NeuroGrid searches in this simulation start with a TTL of 7, whereas Freenet searches start with a TTL of 20, and the fact that the asymptotic values are similar reflects the fact that after network adaptation queries only need travel a short distance before finding a match.

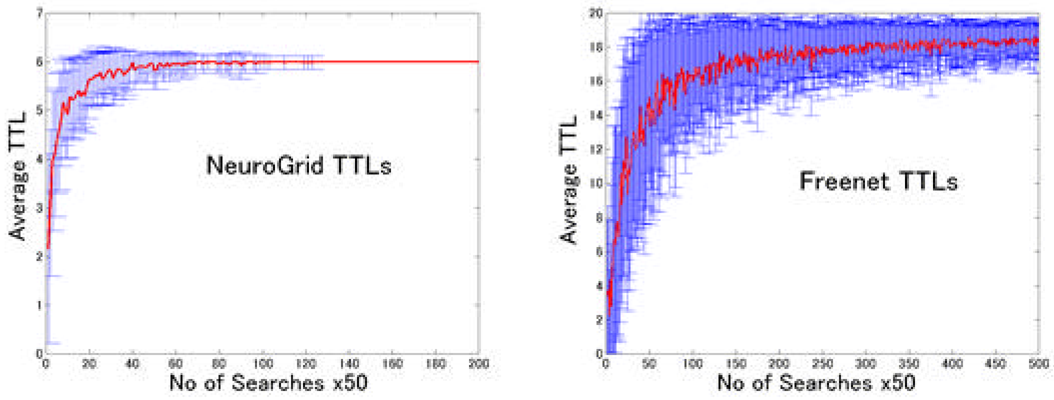

Figure 6. Average TTL of message when first search match is reached.

In addition, it may be noted that the current version of NeuroGrid does not perform caching of any kind, and so while it may have converged faster, it gives none of the additional benefits of Freenet, such as the movement of data to the areas of demand in the network. One immediately appealing possibility in NeuroGrid would be to automatically cache the found documents on the node that originated the search, and have the nodes that it queried learn about it's interests, since this would provide additional sources of requested data.

It is interesting to look at the state of the networks once they have stabilised. For NeuroGrid the average node connectivity is 5.8 +/- 0.5 and the average size of a node's knowledge table (the number of keywords it has entries for) is 19.1 +/- 0.98, compared with an initial connectivity of 3 and knowledge table size of approximately 17.27 +/- 0.85. The Freenet network has a slightly higher and a wider range of node connectivity, the average being 8.23 +/- 2.07. The knowledge table is of comparable size (18.67 +/- 3.13), but again the range is broader. The average number of documents stored in each node after stabilization is 52.16 +/- 30.94. NeuroGrid nodes maintain the same documents they started with.

It is further illuminating to consider a Gnutella network that has the properties of the stabilised NeuroGrid network. A Gnutella network with a connectivity of 6 and a TTL of 7 generates 5901.5 +/-19.1 messages for each search, and generally retrieves all possible matches in a 1000 node network, since almost every node is queried. If we consider the data transfer statistics for the NeuroGrid network we see that in this single example the network stabilised after approximately 60000 messages, not much more than 10 single queries on a Gnutella network of the same size. This does not in any way prove that a NeuroGrid network is necessarily more efficient than a Gnutella network since there are a number of factors that must be taken into account. The first is that this is a single study of a network with only a 1000 nodes. The second is that there is no aspect of the simulation that addresses the real network issues of node failure, document movement and connection failures. Some network realism issues are addressed in the extended simulations below, however, further investigation is certainly required.

Extended Simulations

Various aspects of the simulations carried out above were unrealistic in many respects, and so extended simulations were employed in order to address some of these issues. One of the potential criticisms of the simulations above is the initial random connectivity of the network. Although some kind of power law connectivity pattern (as seen in real networks [15]), where the majority of nodes are only connected to a few others in their local neighbourhood, and a few have long range connections, would make for more realistic networks, it has already been noted [3] that Freenet generates a power law connectivity pattern (or small world network) as a natural consequence of its connection formation process. Perhaps more important than setting up a particular network structure for test, is to observe the kinds of networks that form as nodes and connections are added to the system.

These next set of simulations use a ring topology, where the nodes are arranged in a one dimensional line, connected to form a ring, and initial connectivity is set by connecting each node to four other nodes, the two neighbouring nodes on either side. Both Freenet and NeuroGrid change this connectivity structure as successful queries bring back information about more remote nodes.

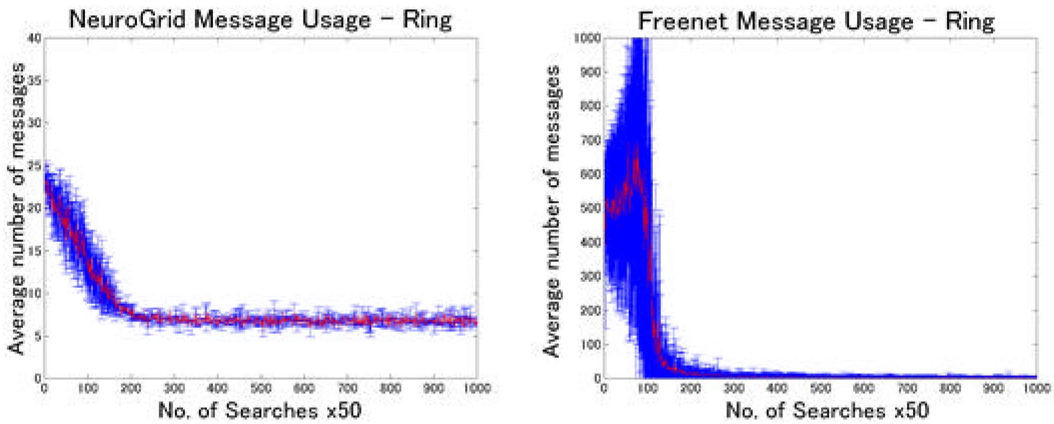

Figure 7. Message Usage against ongoing searches in NeuroGrid and Freenet networks that start from a ring topology

Comparing fig. 7 with fig. 5 above we see the impact of the ring topology is to make the initial number of messages required by NeuroGrid drop considerably, although the stable state is at a higher number of messages. Both NeuroGrid and Freenet take much longer to stabilize, and Freenet also sees a spike in the number of messages prior to stabilization, although the error bars are so large it would be hard to draw any definite conclusions, other than that the behaviour prior to stabilization in the ring topology is more erratic than in the randomly connected Freenet.

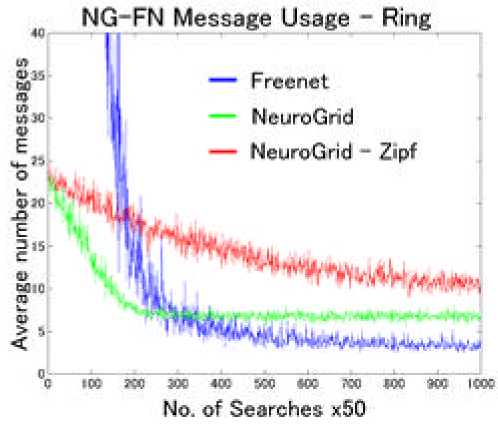

Figure 8. Message Usage against ongoing searches in NeuroGrid and Freenet networks that start from a ring topology, and also the effect of giving keyword-document relations a Zipf distribution

Also of note is that under the ring topology starting conditions the Freenet system converges to a stable state where the messages required on average are less than NeuroGrid (fig. 8). This suggests that while Freenet may be able to produce a more efficient search process after stabilization, a price has to be paid in data movement before that stable state is reached. The ring topology starting conditions lead to an order of magnitude increase in the amount of data movement required by Freenet (~10^6 messages, ~10^5 documents), in contrast the ring topology constraint does not greatly change the data transfer in NeuroGrid (constant at ~10^5 messages). Still, it would be desirable to run all of the simulations multiple times to get some better indication of how much of these effects are due to starting conditions. The only constraint is processing time, so we can look forward to some more detailed studies in a subsequent document.

Another potential criticism of the earlier simulations is that the random assignment of keywords to documents for NeuroGrid is unrealistic. Thus a Zipf law [2] distribution was employed whereby the likelihood of a document being assigned a keyword was an inverse function of the keyword rank (the keyword rank being determined from the keyword ID, 1, 2, 3, ... etc.). This issue does not affect Freenet which defines its own document-key hash relationship. As can be seen from fig. 8 NeuroGrid's performance is worse under the Zipf distribution than the random case, although the amount of data transfer is still much less than that under Freenet.

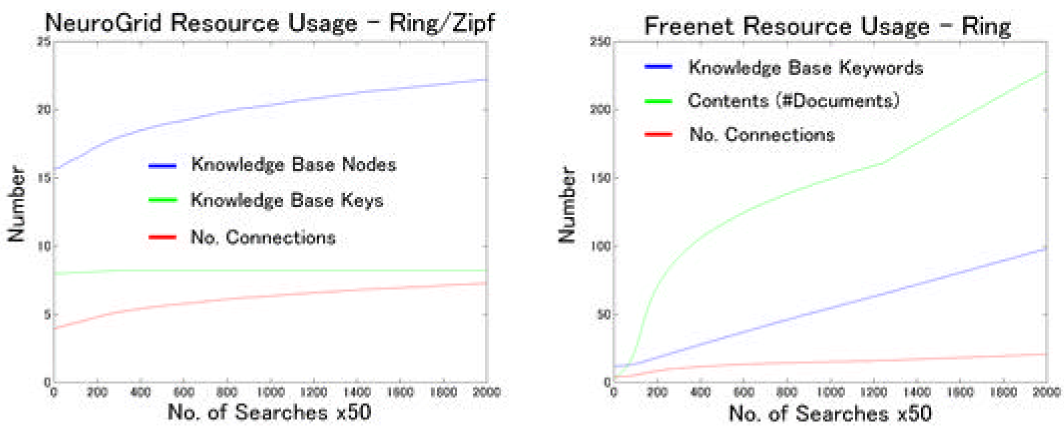

Figure 9. Resource Consumption against ongoing searches in NeuroGrid and Freenet networks that start from a ring topology, specifically #connections, knowledge base size and contents (#documents stored in each node)

This final pair of figures (fig. 9) show us how NeuroGrid and Freenet use system resources as searches go on. Note that the Freenet graph scale is an order of magnitude larger than the NeuroGrid graph. There seems to be a general increase in the amount of resources by both systems as searches go on, with the exception of the keys in the NeuroGrid knowledge base, which probably reach a ceiling because the Zipf distribution means that the majority of documents will be represented by a relatively small set of keys. The consumption of the various resources seem to be less than linear across the board, and it could easily be controlled by setting resource limits (as Freenet does). The important thing is to be able to work out what set of resource limits will allow the network to converge to a stable state, and what would actually prevent that state from being achieved. Further analysis will provide more detailed dependencies of stable state achievement on resource availability.

Considerations

Considering that the introduction of a Zipf distribution of keywords over documents lead to a drop in NeuroGrid performance, it would be desirable to see how increases in the number of keywords affected the system, as well as looking at remedial action such as restricting keyword to the more meaningful portion of the Zipf distribution, as is effected by information retrieval approaches like TFIDF (term frequency inverse document frequency) [13]. It would be interesting to see whether document clustering made this problem better of worse. It seems plausible that to the extent that regularities exist between the types of keywords used to describe documents and their locations, then the NeuroGrid routing approach could exploit them to improve search efficiency. Although we might expect the network to take longer to stabilize if node contents were polarized. There is also an issue of node contents changing over time; if follows intuitively that in a network where documents where moved at random, and at a much faster rate than search can take place, then their will be little to gain from the NeuroGrid approach. In that case, one must either exhaustively search the network, or provide some rules about where documents are placed and relocated, as in the Freenet model.

Clearly further study of the NeuroGrid system and its dynamics is required, particularly how its react to node deletion and addition. Other possibilities include simulating data transfer failures, and investigating how convergence time relates to network size and other factors. In terms of making the NeuroGrid simulation more realistic it would be good to improve the way in which documents are assigned to nodes, such that documents with similar keywords appeared on the same nodes in the NeuroGrid starting network, as mentioned earlier. However, it is not immediately clear what kind of distribution would best simulate that found in real world networks, such as the world wide web. There is also the issue that the initial connection pattern should follow some kind of power law, rather than ring, or random topology, but it would seem expedient to concentrate first on ways in which nodes join the network (c.f. [3]) and look at the emerging patterns of connections, rather than trying to impose a particular structure.

Conclusion

In conclusion we can say that (1) Gnutella uses a fairly inefficient search process, but that this is the default approach, particularly when node identities can not be determined reliably; both Freenet and NeuroGrid rely on being able to consistently determine node identities. (2) NeuroGrid and Freenet appear to support a more efficient search procedure then Gnutella, and NeuroGrid does this with far fewer data transfers being required to achieve convergence. (3) NeuroGrid's lighter data transfer requirements come at the cost of having no caching functionality which is important in providing Freenet with some of its attractive qualities, and the requirement that documents have associated keywords.

References

[1] G. Barish and K. Obraczka "World Wide Web Caching: Trends and Techniques". IEEE Communications Magazine Internet Technology Series, (2000)

[2] A. Bogomolny, "Benford's Law and Zipf's Law" http://www.cut-the-knot.com/do_you_know/zipfLaw.html (2001).

[3] I. Clarke, O. Sandberg, B. Wiley & T. W. Hong "Freenet: A Distributed Anonymous Information Storage and Retrieval System" in Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability, LNCS 2009, ed. by H. Federrath. Springer: New York (2001).

[4] P. Deutsch, R. Schoultz, P. Faltstrom, and C. Weider, "Architecture of the WHOIS++ service". RFC 1835, http://www.ietf.org/rfc/rfc1835.txt (1995).

[5] Nick Gibbins, "Agent-based Computing and General Distributed Search", http://citeseer.nj.nec.com/13222.html , First Year PhD Report.

[6] S. R. H. Joseph, "Why Autonomy makes the Agent". In "Agent Engineering" Eds. Liu, J, Zhong, N, Tang, Y.Y and Wang P. World Scientific Publishing (2001).

[7] S. R. H. Joseph, "NeuroGrid Protocol" http://www.neurogrid.net/ng-protocol.html (2001).

[8] S. R. H. Joseph, "NeuroGrid White Paper" http://www.neurogrid.net/WhitePaper0_3.html (2001).

[9] S. R. H. Joseph "NeuroGrid Simulation" http://www.neurogrid.net/ng-simulation.html (2001).

[10] A. R. Mikler, V. Honovar & J.S.K. Wong "Analysis of Utility-Theoretic Heuristics for Intelligent Adaptive Network Routing" In: Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96). San Diego: AAAI Press. (1996).

[11] M.K. Reiter & A.D. Rubin, ``Anonymous web transactions with Crowds''. Communications of the ACM 42(2), 32-38 (1999).

[12] E. M. Royer & C-K. Toh "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks". IEEE Personal Communications, *April:* 46-55 (1999).

[13] G. Salton and C. Buckley, "Term-weighting Approaches in Automatic Text Retrieval". In "Readings in Information Retrieval" Eds. K. Sparck Jones and P. Willett, Morgan Kauffman (1997).

[14] J. Wang "A Survey of Web Caching Schemes for the Internet". ACM Computer Communication Review, 29(5):36--46, (1999).

[15] D. Watts and S. Strogatz, "Collective dynamics of `small-world' networks," Nature 393, 440-442 (1998).