Peer-to-Peer Networks for Self-Organizing Virtual Communities

NSF Proposal IIS-0118767

July 2001

Jamie Callan
Ramayya Krishnan
Alan Montgomery

Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

Project Summary

Locating information on the Internet is a leading usability problem. Often desired information exists but is difficult to find. A method for addressing this problem in the physical world is to create a community. Communities tend to be organized around individuals with similar needs and characteristics. The commonalties within a community provide a context in which to evaluate retrieved information, increasing the chance that information will be relevant to the specific information need. For example, a biomedical researcher interested in locating current genetic research on Parkinson's Disease will have much different requirements than a patient who has been diagnosed with this disease. A search by each of these individuals using a general-purpose search engine would be treated in the same way, since syntactically the searches may be identical. However, if the biomedical researcher presents the request to other biomedical researchers the obtained information is likely to be much more relevant for his or her needs. Communities help preserve the information need context that is missing in general purpose search engines.

Virtual communities have developed on the Internet, organized around particular topics, to serve purposes similar to more traditional communities in the physical world. The existing approaches to virtual community formation tend to be top down or ad-hoc. Virtual community hosting sites such as geocities.com do not support community formation based on content or type of information retrieved. Online communities must find an efficient means of locating individuals for which their information is highly valuable, or else only large-scale online communities will be practical. Recent failures of online business communities (e.g., Chemdex.com in the Life Sciences industry) have been attributed to the high costs of member acquisition.

We propose a new approach to forming virtual communities that is based on peer-to-peer computer networks. Peer-to-peer computer networks enable members to communicate directly with one another. The communications links between network nodes form a dynamic network topology. Nodes are aware of which other nodes they are connected to, and nodes can disconnect from one part of the network (one set of nodes) and reconnect to another part of the network (another set of nodes) when it is convenient or advantageous to do so. Large-scale peer-to-peer networks offer the possibility of *self-organizing communities*, in which nodes recognize and create relatively stable connections to other nodes with similar interests/requirements/capabilities. A given node might participate in several such virtual communities at a given time, for example to satisfy different types of information needs.

Peer-to-peer networks do not currently include capabilities that support formation of virtual communities, for example, methods of identifying other nodes with similar information needs or that publish similar information. The proposed research will address this need by placing at each node a software agent with more intelligent decision-making and learning capabilities. The agent will be able to decide which other agents it should connect to, how to allocate its resources (e.g., bandwidth) to these connections, and when faced with an information retrieval task, determine the agents it should query to maximize the "quality" of the response. The agent will develop knowledge about the other computer agents in the network to which it is connected. These agents will be designed using a decision theoretic formulation in which they weigh the expected benefits of the information against their expected costs, using evidence such as expected reliability, responsiveness, and likelihood of containing desired content

The proposed research will develop and evaluate centralized and local approaches to acquiring information about the characteristics of nodes in a peer-to-peer network; a particularly challenging type of information is information about the content at each node, but we will also develop and evaluate methods for acquiring information about other node characteristics, such as reliability and responsiveness. The proposed research will develop decision-theoretic utility functions that enable nodes to optimize their decision-making with respect to their specific priorities. This work will lead to a better understanding of i) information gathering and decision-making in large-scale peer-to-peer networks; ii) network conditions in which optimizing individual utility is consistent with globally desirable conditions such as reduced network congestion; and iii) network characteristics that foster spontaneous formation of virtual communities through the individual decision-making of nodes about where to connect and how to relay messages in the network.

The research will establish research methodologies and produce a simulation environment for studying large-scale peer-to-peer networks in a systematic manner. This simulation environment will also be used in graduate professional Masters degree programs, to provide hands-on experience with peer-to-peer networking, virtual communities, and virtual marketplaces.

C. Project Description, Goals, and Objectives

1. Introduction

Locating information on the Internet is a leading usability problem. Often the information may exist, yet an individual is unable to locate the needed information. Alternately the user may issue a general search query and be overwhelmed with irrelevant information. A method for addressing this problem in the physical world is to create a community. Communities tend to be organized with individuals with similar needs and characteristics. Sociologists have longed recognized that people cluster toward others who are similar to themselves (i.e., homophily, [Lazarsfeld and Merton, 1954]). Therefore a good starting place to find information is within your own community, or to at least use the community as a starting point from which to begin your search.

The commonalties within a community provide a similar context to evaluate the information that is retrieved, increasing the chance the information will be relevant. For example, a biomedical researcher interested in locating current genetic research on Parkinson's Disease will have much different requirements than a patient who has been diagnosed with this disease. Yet, a search by each of these individuals using a general-purpose search engine would be treated in the same way, since syntactically the searches may be identical. However, if the biomedical researcher presents his request to other biomedical researchers the information that is retrieved is likely to be much more relevant than a general query.

Communities help preserve the context of a search that is missing in general purpose search engines. Communities present a solution that has evolved in social groups that have long existed in the physical world to help organize content based upon its context. However, even in physical communities locating information or exchanging information with other community members is not easily accomplished. A new development on the Internet that helps to address this deficiency are peer-to-peer networks [Oram, 2001]. Peer-to-peer networks provide individuals a mechanism for organizing themselves into communities. They feature protocols that are decentralized in nature and do not mask the topology (i.e., connection structure) of the computers linked together to form the community.

The same types of problems arise in creating an online community as a physical community: How can the community locate potential community members and how can information be shared efficiently? Our solution is the creation of self-organizing communities. These self-organizing communities will consist of intelligent computer agents that aid in relaying information and actively seeking out other community members that are similar to themselves. A recent network protocol called Gnutella [Oram, 2001; Wego.com, 2001] will be adapted for this purpose. Gnutella offers the ability to monitor and alter network topology, which provides an ideal environment for studying self-organizing communities.

Our proposal is to embed intelligent decision-making and learning capabilities in these agents to enable them to process information that they collect or obtain about the network. Using the decision making capabilities, an agent can decide which other agents it should connect to, how to allocate its resources (e.g., bandwidth) to these connections, and when faced with an information retrieval task, determine the agents it should query to maximize the "quality" of the response. The agent can develop knowledge about the other computer agents in the network to which it is connected. These agents will be designed using a decision theoretic formulation in which they weigh the expected benefits of the information against their expected costs [Montgomery, Hosanager, Krishnan, and Clay, 2001], using evidence such as expected reliability, responsiveness, and likelihood of containing desired content [Callan, et. al., 1995; Gravano, et. al., 1999].

Existing approaches to community formation are top down. Virtual community hosting sites such as geocities.com do not provide native support for community formation based on content or type of information retrieved. If a potential member is not aware of the community, they will be left out. Recent failures of online business communities (e.g., Chemdex.com in the Life Sciences industry) have been attributed to the high costs of member acquisition. Online communities must find an efficient means of locating individuals for which their information is highly valuable. This is the opposite problem of the search engine, which seeks to provide highly valuable information to an individual. Our proposal addresses this problem since nodes in the network are intelligent agents that learn about how they should be positioned within the network, i.e., determine the community or communities to

which they should belong in a "bottom-up" manner, and that can reposition themselves dynamically based upon current information needs.

The proposal is organized as follows. In Section 2 we review the Gnutella protocol since it will be used extensively in our research. In Section 3, we discuss proposed research. First, we present a utility-based approach to demonstrate how a decision-theoretic framework can be used in the design of intelligent agents in a peer-to-peer network. Second, we outline informational retrieval techniques that enable agents to make better decisions about the locations of desired content in the network. Third, we consider issues related the relationships that these agents develop (i.e., learn) within the network using criteria such as reputation and position. Section 4 outlines a simulation study to evaluate the ability of our proposal to improve quality of information retrieval while simultaneously minimizing "load" on the network. The study will compare decentralized and centralized architectures to gather and disseminate information used by the agents in decision-making. Sections 5 and 6 summarize the goals and objectives of our research. Sections 7 and 8 conclude by addressing administrative issues.

2. Peer-to-Peer Architectures: An Analytic Review

Peer-to-peer networks have created large communities of connected computers (hereafter referred to as *nodes*). These networks permit the shared use of resources: content, storage, CPU cycles, and bandwidth. Unlike traditional Web-based protocols, each node can be either a provider of a resource, a consumer of a resource, or a link in the chain that enables the discovery of resources. There are several specific benefits that peer-to-peer networks have over conventional search engines like Google:

1. Information is current since it is retained by the data provider and not cached by a central directory. Contrast this with search engines in wide use on the web today that index content using spiders. Their indices are only as current as the last spider visit to the information provider.
2. The topology of the network can be used for learning about a node's "neighborhood" and deciding how to respond to "neighbors". An appropriately-organized network can help preserve the context of the search.
3. Peer-to-peer networks are decentralized and do not possess a single centralized server which can serve as a point of failure or a bottleneck

Decentralization of the administration and coordination of nodes in the network is to varying degrees a distinctive feature of these systems. Examples of these peer-to-peer protocols and systems include Napster and Gnutella for file sharing, the SETI project at Berkeley for shared use of CPU cycles, and Freenet for secure information storage and retrieval. A complete listing of these new peer-to-peer systems may be found at http://www.oreillynet.com/pub/q/p2p_category. Given the focus of this proposal on information sharing in decentralized, virtual communities we review the principal features of the Gnutella protocol. Our objective is to present the key conceptual ideas underlying the protocol – and not the technical details of the protocol—in order to motivate the problems that need to be addressed to realize decentralized, virtual marketplaces.

2.1. The Gnutella Protocol

Gnutella is a protocol designed to facilitate decentralized search and discovery of information in a network. Gnutella implementations support three types of behaviors in nodes. As a client, the protocol permits a node (i.e., the computer agent in our introduction) in the Gnutella network to post queries. As a relay, it enables a node to forward to a query to other nodes to which it is directly connected. As a server, it permits a node to search its repository and respond to a query. Each of these behaviors is depicted in Figure 1, which depicts the sequence of message exchanges in a Gnutella network. Node 1, labeled *information searcher*, generates a query and submits it to node 2. Node 2 does not have content that matches the query requirements. It does not respond but relays the query to nodes 3 and 4 to which it is connected. Node 3 does not have information that matches the query either. Since it is not connected to any other node it neither responds nor relays the query request. Node 4 does have content that matches the query. It sends back meta-data (but not the content itself) containing information about the content and its IP address to node 2 which in turn sends the response back to node 1, the originator of the query. At this point node 1 can establish an HTTP-based connection to node 4 and download the information it needed. In this manner, through

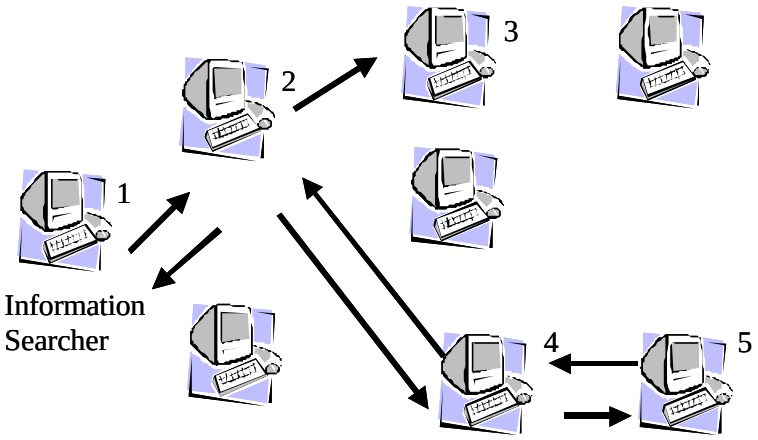

Figure 0. A Gnutella-based Decentralized Virtual Marketplace.

the shared use of bandwidth and CPU cycles provided by the nodes that serve as relays, information can be located and shared between nodes in the network.

A naïve application of this approach will encounter several problems. Chief among these are the delay in getting information from peers in the network versus obtaining the information from an efficiently implemented index. There are also problems caused by two shortcomings of the protocol itself. The first shortcoming is the constraint on the extent to which a node can relay a query. In most Gnutella implementations, this is limited by a parameter called "time to live" (TTL), which is set to 7. Each time a Gnutella packet containing a query is forwarded, its TTL counter is decremented by 1. This implies that searches in Gnutella are bounded and the responses to a query are dependent on the connectivity (i.e., the links to other nodes) of the first node (the *entry point* into the network) to which the query was submitted. This has led to clients making queries to multiple entry points resulting in network congestion with no obvious improvement in response quality. Further, given that peers may leave and join the network, no guarantees can be provided about the ability to replicate the response set to a query even if the query was submitted to the same entry point into the network.

The second shortcoming in Gnutella is the "unintelligent" relaying of requests to other nodes in the network. A node upon receipt of a query relays a valid request (i.e., a request with a positive TTL value) to all other nodes to which it is connected. This results in wastage of shared resources such as bandwidth and does not improve the response set obtained by an information consumer. A related shortcoming is the inability to "allocate" resources such as bandwidth in an economically efficient manner. For instance, a node could get swamped with relay requests and not have any bandwidth left to serve as a reliable and efficient information provider. A node should be able to make decisions – either at the aggregate level or at a fine grained level – of how to allocate resources to the 3 different behaviors of information discovery, providing information in response to download requests and relaying requests. This decision-making should be accomplished in such a manner that local decisions made by each node maximize the "quality" of the responses to an information query while minimizing network congestion.

The discussion of Gnutella thus far has been set in the context of information discovery and sharing in a network of Gnutella-compliant nodes. An equally important issue relates to growth and evolution of a network. At present, an individual seeking to make their computer a node in a Gnutella compliant network chooses randomly to connect to one or more of a set of hosts advertised in sites such as gnutellahosts.com. There is no attempt made to "locate" (i.e., connect to appropriate nodes of an existing network) based on factors such as locality of content provided by the nodes and multi-dimensional factors such as reliability of nodes (i.e., uptime of nodes, available bandwidth, reputation for providing content in a honest manner and so on). Further, once connected, there is no principled means of evolving the connections by disconnecting from nodes or connecting to other nodes based on compiled experience of relaying or sharing information with these nodes.

Inherent in the Gnutella protocol is the ability to monitor and change the topography of the network. This provides a valuable source of information about the network. Specifically, when a search is routed through a Gnutella node, the node has access to the query string from its neighbors. The query terms can be used to catalog the interests of a

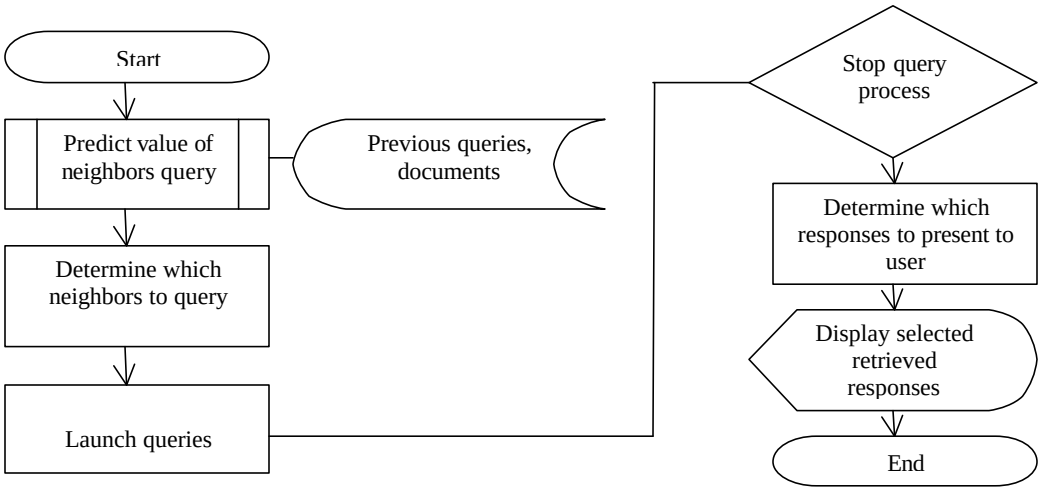

Figure 0. Flow chart of the operational decisions made by an intelligent node.

neighbor and compare them with the interests of the node for future reference. Additionally, successful searches can be recorded, providing information about which neighbors would be good references in the future.

Research in sociology [Wellman and Berkowitz, 1988] and economics [Jackson and Wolinsky, 1996] on social networks offers many insights into these issues. Jackson and Wolinsky [1996] consider conditions under which self-interested individuals can form or sever links to create stable and efficient networks. Their work is stylized and theoretical but demonstrates that under the right circumstances self-organizing communities will thrive. Our proposal is to develop decision-theory based technology to create decentralized virtual communities for information discovery and retrieval.

3. Research Issues

Large-scale use of peer-to-peer protocols such as Gnutella requires support for more intelligent decision-making capabilities by individual nodes. Support for more intelligent decision making can be provided at the network level, by adding nodes that provide a more global view of network resources. Support can also be provided at the individual node level by enabling nodes to learn to relay messages more effectively. Each approach improves network efficiency and increases the probability that each information need is satisfied in a predictable manner. They also enable additional capabilities with respect to reliability and authenticity of information providers, which peer-to-peer networks currently lack.

We propose to conduct research along several dimensions, outlined below, with the goal of providing support for more intelligent and reliable decision making in peer-to-peer networks. Our research will address centralized services that provide network-level information, learning capabilities within individual nodes, and methods of making decisions using multiple sources of evidence and multiple decision metrics. We first present a decision-theoretic formulation of the agent decision problem and use that to organize the specific activities in our proposed research program.

3.1. A Decision-Theoretic Formulation for Intelligent Agents in a Peer-to-Peer Network

To improve the performance of the Gnutella network, individual nodes need to better anticipate the value and cost of a search. The Gnutella network can be inundated with requests, which can significantly slow searches across the entire Gnutella network. As discussed in Section 2, if a Gnutella node initiates a search, this node is not intelligent in either selecting whom to query or in deciding how many resources to devote to such a search. Instead requests are simply passed onto all neighbors, who in turn pass the request onto their neighbors. Clearly nodes need to be more selective in the searches that they generate and the requests that they forward. Our goal is to look for a

solution that does not require nodes to act altruistically, but to identify goals for an individual node that are consistent with reducing network traffic.

Our proposal is to use a decision-theoretic formulation of the problem faced by a node in the Gnutella network. Specifically, we propose to introduce a value or utility measure so that a node can better assess the gains of a search and weigh them against its costs. We do not impose monetary costs, nor do we consider the cost of the search to others on the network, but instead focus on the node's own opportunity costs. Our purpose is to establish conditions in which optimizing individual utility is consistent with reducing network congestion. For example, a user may prefer to wait ten seconds to search five neighbors versus a search of thousands of sites that takes more than 90 seconds.

Implicit in this choice is a tradeoff of accuracy – operationalized as the precision and recall of the response – for speed. But it also suggests that users would willingly reduce the number of nodes searched, and consequently network congestion, to improve their own performance. We consider the following design elements involved with the construction of an intelligent node: which neighbors to query, how long to search, and which results to report to a user. The decision flow is given in Figure 2. The network consists of tens or hundreds of thousands of nodes that could be queried; the problem is to determine the node or nodes to search. To solve this problem we introduce the notion that the node has prior information to estimate the likelihood that a node has information relevant to a query. This information could be based on previous retrievals, or on knowledge about queries that have been relayed by this node in the past, or from another node in the network that maintains a specialized type of directory service. Clearly, there is a great deal of information that can be brought to bear on where to search. Similarly the agent can balance search time against the expected gains of search and intelligently decide how long to search.

An additional design problem for the intelligent node is which of the set of responses from a query to present to the user. Imagine that a search retrieves hundreds of responses, and that the user's requirement is met if one relevant response is identified. Every additional alternative response presented will force the user to expend cognitive effort. Given that consumers are cognitive misers, additional time spent on cognitive activity is more taxing than simply waiting. Presenting all retrievals to a consumer is clearly not the best solution, since this is very burdensome for the user. The other extreme of showing only the "best" retrieval is also unsatisfactory, since it is unlikely that the agent can be confident enough given the stochastic element in predicting utility. Therefore, we propose that the agent score query responses for their expected utility, sort these scores by utility, and offer a user fewer but more relevant choices by eliminating unlikely or redundant alternatives. Fewer alternatives reduce the cognitive burden to the consumer, however this reduction also comes at the expense of eliminating potentially beneficial alternatives from the presentation set.

To place a structure on the utility model we assume that utility from a query can be decomposed into the sum of the value derived from its features or attributes. This is a traditional economic approach to modeling consumer behavior and has been well researched in the economics and marketing literature [Lancaster 1966]. This provides an explicit mechanism for balancing the chance of finding a better match against query time and or costs borne by the user. Our approach allows for continuous as well as discrete features attributes. A primary benefit of the compensatory approach is that it can better capture tradeoffs that consumers make. For example, the information content in a bandwidth heavy GIF illustration in a document may compensate for the added time it takes to download the document.

To show how this framework could be used to construct a more intelligent node we formalize our arguments following the approach developed in our previous work on intelligent shopbot design [Montgomery, Hosanager, Krishnan, and Clay, 2001]. The utility of set of queries presented to a consumer (V) equals the utility of derived from the query (U) less the disutility from waiting (T) and the cognitive costs of evaluating the query (C):

$$V = U - xT - lC$$

Response time (T) is the time associated with making queries to Q nodes in parallel, which depends upon the slowest query, hence $T=\max(t_1,t_2,\ldots,t_Q)$. x represents the cost to the consumer of waiting one second. The cognitive effort associated with evaluating the set of alternatives is proportional to the comparisons that must be made [Shugan, 1980]. The number of comparisons equals the number of unique pairs of attributes/response combinations, $C=(A-1)(P-1)$ where A is the number of attributes of a given document and P is the set of responses obtained from

making queries to Q nodes. Alternatively this could be interpreted as the time to evaluate a retrieved offer. Since waiting and cognitive effort decrease utility their parameters are positive, x, ?>0.

The utility derived directly from the query responses equals the maximum utility of the responses from the presentation set, since we presume the user is only interested in selecting one item from this set. To simplify the statistical structure we assume that the node has N neighbors and contacts each of the neighbors directly, which in the Gnutella protocol means that TTL (time to live) is set to 1. Additionally, we assume that Q nodes are queried and their retrieved queries are sorted by their expected utility and the best P queries are presented to the user:

$$U = \max(U_{1:Q}, U_{2:Q}, \ldots, U_{P:Q})$$

Where i:Q denotes the ith ordered variate from a set of Q random variables, for example 1:Q denotes the maximum in this set. The utility of an individual response is the sum of a deterministic (U_i) and stochastic component (e_i):

$$U_i = \sum_{j=1}^{A} b_{ij} a_{ij} + e_i = \overline{U}_i + e_i$$

where b_{ij} denotes the weight and a_{ij} denotes the value of the jth attribute of the ith response. The selection of the attributes and the estimation of their weights will be discussed further in section 3.3.

The stochastic component is due to unobservable factors or random evaluation error by the user. We assume that the errors follow an extreme value distribution, which is a traditional assumption in choice models [Ben-Akiva and Lerman, 1985]. The mean and variance of the errors are ?? and p^2?^{2}/6, respectively, where ? is Euler's constant (?˜.57722). The parameter ? measures the extent that utility can be predicted. If ?=0 then utility is known with certainty. In our application the attributes of a query response could be attributes related to information content (e.g., relevance as determined by matching query terms, length in bytes of the document and so on) as well as factors such as the reputation of the information provider.

Using the property of the extreme value distribution and our other definitions we can write the expected utility of the presentation set after the query responses are retrieved as follows:

$$E[V] = q\left[\ln\left(\sum_{i=1}^{P} \exp\{\overline{U}_{i:Q} / q\}\right) + g\right] - x \max(t_1, \ldots, t_Q) - l(P-1)(A-1) \quad (1)$$

This equation allows us to establish the optimal number of responses to present to the user. As a simplification if all query have the same expected utility the optimal number of responses, P* to present is:

$$P^* = \frac{q}{l(A-1)}$$

Notice that the size of the optimal set size increases as cognitive costs decrease or the variance of utility (?) increases.

Additionally, we can use Equation 1 to determine how many and which nodes in the network to query and how long to wait based upon our prior expectations about U_i and t_i. Consider the following scenario that there are 100 neighbors that can be searched and U_{iI} follows a standard normal distribution. Additionally, suppose that t_i follows an exponential distribution with a mean of 1 second. We set ?=1, ?=.75, and ?=.1, to represent a fair degree of uncertainty on the part of our agent to predict utility. The utility for querying varying numbers of nodes is illustrated in Figure 3. Notice in this example the optimal number of nodes that the node should query is four. Querying too many neighbors is sub-optimal. This is not due to the utility or the value of the documents retrieved, but simply that every search requires additional time. The expected time to search four nodes is a little more than two seconds while to search 30 nodes is about four seconds. However, the added gains in the utility of the documents are outweighed by the implicit cost of the additional search time. This result also demonstrates the fact that if a node is

intelligently designed it can reduce network traffic while at the same time increasing the satisfaction of users over current implementations.

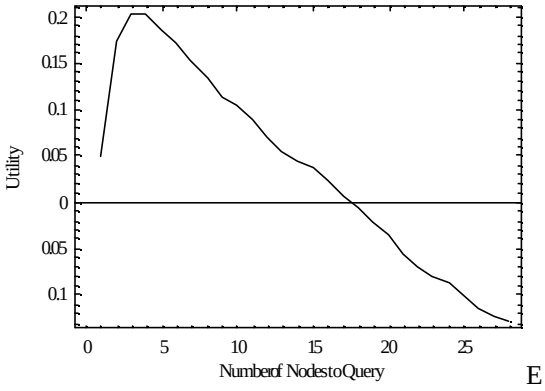

Figure 2. The utility that corresponds to the number of nodes searched. In this simulation the optimal number of nodes to search is 4.

Equation 1in Section 3.1 defines utility in terms of a deterministic component – modeled as the linear weighted sum of attributes (a_{ij}) – and a stochastic component. The implementation of Equation 1 requires the availability of information about these attributes. We will focus on three attributes relevant to the decision making, namely the content available at other nodes, the response time distribution of a node, and its reputation (e.g., that a node indeed provides what it claims in a query response). The compilation of this information is an important task in the proposed research program. A related issue is the manner – centralized vs. decentralized – in which the compiled information is made available to decision making nodes. Finally, the procedure to determine the weights ($ß_{ij}$) or the relative importance placed on these attributes need to be determined. These issues are discussed in more detail in the following sections.

3.2. Content-Based Resource Ranking

Large-scale peer-to-peer networks are not practical unless nodes are able to route messages to nodes that either a) have a high probability of providing desired content, or b) have a high probability of being connected via a short path to nodes providing desired content. We characterize these as routing the message directly to a desired content provider or into a neighborhood containing a desired content provider. This is an instance of a content-based resource-ranking problem, in which a set of resources (nodes) is ranked by the likelihood of containing content matching the information need described by a particular query. Content-based ranking is an important component of the utility function (Equation 1 in Section 3.1) used to make decisions in our proposed approach.

Content-based resource ranking can be provided by either global or local methods. We propose to develop and compare both global and local approaches. Each is described below.

3.2.1. Directory Services for Content-Based Resource Ranking

One solution is to extend the peer-to-peer protocol to include nodes that provide *directory services* to other nodes. A directory service matches *information requests* with appropriate *information provider* nodes, thereby reducing network traffic and increasing the probability that an information request will be successful (find desired content). Napster is a familiar example of a directory service, one that is tailored for a particular type of content (mp3 files) and in which the directory service has complete information about the catalogs (content) at each node.

We propose a more open architecture with the following characteristics:

- No restrictions on content type;
- Multiple, competing directory services;
- Information providers are not forced to divulge complete information about their catalogs; and
- Use of the directory service by nodes requesting information is optional.

Our goal is to extend the Gnutella architecture by giving requesting nodes another, more content-based method of determining where in the network to send their messages. Instead of posting messages at essentially random points in the network, a node could contact a centralized directory service, obtain a list of nodes with content similar to the request, and then submit the request directly to those nodes. The intent is for messages to be submitted to nodes in the right *neighborhoods* of the network, so that little propagation is required before the message reaches a node with matching content. Network traffic is reduced, the probability of success is increased, and identical requests are more likely to be handled in an identical manner.

Directory services need to know what information is available at each node. A directory service can require complete information (as does Napster), which requires the explicit cooperation of the information provider, or it can require only partial information, which may or may not require cooperation from the information provider. Cooperative models can be less desirable in diverse, multi-party environments, because they can be deceived easily, they fail to include nodes that cannot or will not provide requested information, and they can be difficult to extend to new types of information or services.

We propose to extend *query-based sampling* to the task of creating descriptions of the content available at each node in a peer-to-peer network. Query-based sampling is a technique whereby the topic areas covered by a resource are discovered by submitting queries and examining the results that the resource returns [Callan, et al, 1999; Callan & Connell, accepted]. Experimental results with text databases ranging in size from about three thousand document titles and abstracts (about 3 megabytes) to about one million full text documents (about 3 gigabytes) demonstrate that accurate resource descriptions can be learned by examining the results of a relatively small number of queries (e.g., 75-100).

Query-based sampling can also be applied in a peer-to-peer environment such as Gnutella. Queries can be submitted directly to specific nodes, and the time-to-live (TTL) can be set low enough to prevent the message from being forwarded to other nodes. Nodes respond either with a success message, also providing a reference to the object that matches the query; or they respond with a failure message. If a request is successful, the matching object can be fetched and its contents analyzed to determine more information about the contents provided by the node, and also to verify that the node provides the type of content that it claims to provide.

A Gnutella network is a challenging environment in which to apply query-based sampling. Query-based sampling works well when queries obtain full-text documents, because the contents of fetched documents guide the selection of subsequent query terms used for sampling the resource. The resource description converges quickly to the subject area covered by the resource. However, it is an open problem how best to sample resources that return other results, such as images, or stock quotes, or airline schedules. We will develop extensions to query-based sampling that adapt to the types of information returned by a node, for example varying among frequency-based and random selection of query terms from known sources (e.g., a dictionary) and previously fetched objects. Even relatively foolish sampling strategies will converge eventually, because there is a limited, albeit possibly large, vocabulary of terms that will be matched at a given site. The research goal is to develop methods that learn relatively accurate resource descriptions from only a small amount of interaction with the node.

It is *not* a goal to characterize every information object available at an information provider. It is enough to know that a site is a provider of financial information, or perhaps financial information about high-tech companies. Our past research indicates that relatively accurate resource selection can be provided using only partial descriptions of resources [Callan, 2000]. Partial resource descriptions are not sufficient for “needle in the haystack” queries, in which just one resource has just one document on a subject, but such queries are rare in practice. However, when these situations do arise, the Gnutella protocol provides a failsafe capability. If the directory service can provide a list of nodes in the right “neighborhood” of the network, the message will be propagated by message-passing and is likely to reach the “right” node.

Once partial resource descriptions are available, a directory service can use resource-ranking algorithms such as the CORI and gGlOSS to rank nodes by how well their (sampled) content matches a specific information need. gGlOSS is a vector-space similarity measure between queries and resource representations [Gravano, et. al., 1999;]. CORI is based on Bayesian inference networks [Callan, et. al., 1995; Callan, 2000]. Both algorithms can be applied with partial resource descriptions acquired by query-based sampling, although the CORI algorithm is somewhat more consistent [Callan, et. al., submitted] and has been applied in a greater range of operating environments (e.g., [Larkey, et. al., 2000]). The proposed research will study how well these resource-ranking algorithms work with descriptions of Gnutella resources, and adapt them to the Gnutella environment as necessary.

This approach to creating directory services creates an open environment in which multiple, competing directory services can flourish, and in which information seekers can make their own decisions about which services to use. It allows directory services to specialize in particular subject areas or in sites providing particular types of services, for example by using domain knowledge to create tailored sampling strategies. General directory services will provide access to a broad range of content types, including emerging communities that have not yet reached the critical mass that would warrant special purpose directory services serving particular communities.

In summary, our research goals are to provide robust, open-architecture methods of creating directory services that will make very large-scale peer-to-peer networks practical. The directory services will reduce the amount of message passing required to find information, increase the probability of finding desired information (if it is available on the network), and increase the probability that two nodes submitting identical requests find the same information. The presence of directory services also enable other capabilities, for example, independent verification that a particular node usually provides the type of content it claims to provide.

3.2.2. Content-Based Resource Ranking by Individual Nodes

A second solution is for individual network nodes to *learn* to direct messages more effectively, based on their past experience routing messages. This idea was introduced initially in the FreeNet peer-to-peer architecture [Clarke, 2000; Clarke, et. al., 2000]. When a new message is similar to a past message, it is routed to the node that was able to provide matching content in the past.

FreeNet's matching function is based on a cryptographic hash, which protects the privacy of message contents. This matching function is possible because FreeNet nodes are expected to also cache the *results* of queries that they have seen in the past. A node may enter the network as a provider of "foo", but if it receives the message "supercalifragilistic" (perhaps because another node thought "foo" and "supercalifragilistic" were similar), the node is expected to forward the message and *to cache the response*, so that in the future it will also become a provider of "supercalifragilistic".

We propose a more open-architecture approach, in which fewer expectations are placed on the behavior of other nodes, and in which individual nodes are free to select whatever matching function they choose. This behavior is embedded within our decision theoretic approach to the design of the node's intelligent agent.

The Gnutella protocol does not require or prevent learning within individual nodes. We will study the effects on network traffic when some or all of the nodes learn from the traffic that they route. Learning nodes will represent other nodes by a list of messages to which they have responded positively in the past. We call these *local learned resource representations*, to emphasize the fact that two nodes A and B may have very different views of a third node C, based on the messages they have exchanged with C. Each node will be free to implement its own method of determining "similarity" between new messages and learned local resource descriptions. We will also explore traditional metrics for resource ranking, such as the CORI and gGlOSS algorithms [Callan, et. al., 1995; Gravano, et. al., 1999; Callan, 2000], but they may require significant adaptation or may be completely unsuitable because they were they were developed for resource descriptions of considerably longer length (e.g., 10,000 words or more). It is an open question how many messages from a node must be seen (i.e., how long its resource description must be) in order to route messages to it reliably. We will also explore methods of determining similarity among shorter unstructured text objects, such as cosine similarity [Salton and McGill, 1983], and metrics based on language modeling [Ponte and Croft, 1998].

Learning at the level of individual nodes provides low-level adaptability in the network, but it is not foolproof. In the Gnutella peer-to-peer protocol an intermediate node does not actually see the content promised by an information provider; it sees only a message in which a node claims to have matching content. Individual nodes learn what each node *claims* to provide, but those claims may be false. We address this issue below, in the Reputation section.

In summary, our research goals are to develop learning methods that can be applied to individual nodes in a peer-to-peer network such as Gnutella so as to reduce the amount of message passing required to find information, increase the probability of finding desired information (if it is available on the network), and increase the probability that two nodes submitting identical requests find the same information. Nodes will not be required to cache content provided by other nodes, and they will be free to choose whatever similarity functions they choose, encouraging development of improved methods.

3.3. Learning About Reputation

While content–related attributes (as discussed in the previous section) are an important determinant of utility, reputation of the information provider is an attribute that we propose be part of the utility function. By reputation, we mean the "audited" capability of an information provider to provide information they claim to be able to provide in response to a query. The objective here is to prevent the sort of problems that arise with the misuse of technologies such as meta tags that result in high rankings for documents that are not relevant to a search query. There are two approaches that we propose to investigate in this context.

The first approach is centralized and requires the directory service discussed in the previous section to sample content in order to ensure that the information provider indeed has information that they claim to have available. Based on this sampling, it develops a numeric rating. Such a rating may be modeled as a probability $P_i(D_j/R(Q))$, interpreted as the probability that provider i has document D_j given a response R to a query Q. These conditional probabilities can be updated based on experience using a Bayesian [Carlin and Louis, 2000] framework. Such probabilities can also be used to estimate the likelihood that a node will be a trusted provider of other documents D_k ($k \neq i$) in its collection.

The alternative approach is to adapt ideas from collaborative filtering in recommendation systems used in electronic commerce [Schafer et al., 2000]. Recall that in a decentralized Gnutella network only the information seeker eventually downloads the content from an information provider. Relayers of queries or responses do not process the content and can only develop knowledge about the *claims* made by information providers. The proposal is to have the nodes that are information seekers provide ratings of reputation (since they have downloaded the content, they can determine if the provider had made truthful claims) to the directory service or alternatively maintain them locally. These ratings supplied by multiple nodes about information nodes can be pooled together using a Bayesian updating approach to create a reputation matrix that can then be used by nodes in making decisions about utility.

3.4. Learning About Responsiveness

Responsiveness is a measure of the availability of a node in the network. Recall that the utility function discussed in Section 3.1 has a negative weight on waiting time. When a collection of nodes are queried simultaneously (i.e., parallel threads), the time $T=\max(t_1,t_2,\ldots,t_Q)$ is the maximum response time of one of the nodes in the set determines when responses can be compiled, processed and presented to the information seeker. Prior to making a decision about the nodes to query, information about responsiveness needs to be compiled. We propose to model responsiveness as a response time distribution.

Information about previous responses could be made available in either a centralized manner or decentralized manner. The advantage of a centralized approach is that the response time monitor can poll the nodes in the network and compile the response time distribution. As in our example in Section 3.1, information about the distribution (e.g., exponential distribution with a mean of 1 second) can be distributed to the decision making nodes. The disadvantage with the centralized approach for *collecting* response time distributions is that this is not the response

time observed by neighboring nodes when they pose query requests or relay requests. In the worst case, you could have nodes responding more promptly to the response monitor than to *real* relay or query requests.

We propose to have local nodes develop response time distributions conditional on the type of request made to other nodes. A response time distribution is modeled as probability density function R(i,j,k,t) that is interpreted as probability that the response time of node i to node j for task k is t seconds. Such response time distributions can be used in two ways. First, it can be used by the node to make decisions about the set of nodes to query using the utility function approach that has been discussed. Second, it can be pooled by a centralized service to give new nodes a way to estimate what the response time distribution is likely to be from any arbitrary node in the network.

3.5. Learning About the Utility Parameters: How Much Weight Should an Attribute Be Given?

Initially the value ascribed to attribute weights ($ß_{ij}$ in Equation 1 of Section 3.1) could be given reasonable starting values based upon a representative user. For example, the weight of the score between a query string and the retrieved document could be set to unity and all others to zero. Our expectation is that each user could have much different preferences and perhaps want to incorporate attributes that are unique to their interests. As the user makes choices the agent can learn about the user's preferences. Formally, the choice of which item a user chooses to view from the presentation set offered can be treated as a logit choice model with random coefficients. We propose the use of Bayesian learning to update the values of the $ß_{ij}$. There has been a great deal of work in econometrics and marketing on the use such individualized models [Rossi, McCulloch, and Allenby 1996] in probit and logit modeling that can be applied to this problem.

4. Evaluation of the Research

The proposed research will develop centralized and local approaches to acquiring information about the characteristics of nodes in a peer-to-peer network, and it will develop decision-theoretic utility functions that enable nodes to optimize their decision-making with respect to their specific priorities. The research will be evaluated at the component-level, and at the network level, using a combination of simulation and testing within Gnutella networks constructed for that purpose. Component-level testing will determine how effectively a particular method performs, for example, how effectively a particular local learning strategy discovers the contents of neighboring nodes under various conditions. Network-level testing will investigate the effects of various node configurations on a network of Gnutella nodes, for example, whether the presence of nodes that learn about the contents of neighboring nodes actually reduces network traffic or improves consistency in satisfying identical queries submitted by nodes in different parts of a network.

The evaluations will be built around a simulation testbed that contains a population of Gnutella nodes with varying characteristics. The 100 gigabyte TREC Very Large Corpus (VLC2) database [Hawking, et al., 1999] will be automatically partitioned into a set of databases with desired characteristics, as is common in distributed information retrieval research [Callan, 2000], to provide content for nodes in the testbed. The VLC2 database consists of about 20 million Web pages, which will support experiments with up to about 10,000 nodes. The VLC2 data was chosen because we want to use data that others can obtain, so that others can replicate the experiments or use the same testbed for related research.

Other node characteristics, for example what evidence is used in making decisions and how it is acquired, can be configured relatively simply. For example, a node can use one of three content-based criteria when deciding where to submit a query initially: i) ask a centralized directory service for advice, ii) use the node's own past experience gained from observing messages it relays, and iii) random selection (current Gnutella). The set of node configurations is finite and small; the choices described in this proposal represent 24 combinations.

The number of networks that can be created from a set of nodes is essentially infinite for any reasonable population of nodes. We will develop software that can generate network configurations automatically, based upon probability distributions describing network characteristics such as number of neighbors per node, probability of being connected (initially) to a node with similar content, reputation of a given node and probability of being reliable.

The final element of an evaluation is information needs (queries) and relevance judgements for retrieved information. The TREC VLC2 corpus includes queries and relevance judgements, but too few to be useful for our purposes. A common approach to generating queries automatically is to select a document randomly, weight its terms by a *tf* or *tf.idf* formula, and select the top *N* terms. The "matching" documents for the query can be determined in two ways: i) those that satisfy a Boolean query, or ii) those that would be in the top *D* documents if the entire network were treated as a single, large database.

Once node content is determined, node decision-making is configured, node information needs are determined, and nodes are connected in a network, a simulation can begin. The effectiveness and efficiency of a network operating under a particular set of conditions can be determined with a variety of metrics, including how well nodes satisfy their priorities with respect to Precision, Recall, reliability, and response time, and the cost to the network in satisfying requests, measured by the path lengths of successful searches and the number of messages propagated per search. We are particularly interested in how measurements change over time in a network with nodes that learn. The degree to which the network supports formation of virtual communities can be measured by changes over time in the distances between nodes with similar content, or with similar information needs.

No single simulation will provide the results that we seek. However, testing under a variety of different population assumptions and network configurations will identify methods of gathering evidence and making decisions that tend to improve results and lead to stable network behavior.

5. Summary

This proposal advocates the use of decentralized communities for information discovery and retrieval. The following problems are addressed in this research.

A The efficiency and effectiveness of information search and discovery is maximized for each individual query while minimizing the load on the network.

1 This requires new measures about the distribution of content on the network and information about factors such as reliability and reputation. A decision procedure for the nodes (agents in the network) will be developed to allow the nodes to use these measures and information to maximize the utility of the information searcher. This can be done by determining the entry points for a search query into the network, selectively responding to queries and relaying messages, and making decisions about how to allocate internal resources to respond to peer requests. An important issue in this is understanding under what conditions centralized gathering and dissemination of information dominates decentralized or local gathering and dissemination of information required by nodes for decision making.

2 Minimizing load on the network requires that nodes become intelligent learning nodes and use compiled experience with responses to previously relayed queries to learn decision rules about how to process relay requests. In this manner over time, queries will be relayed (or rather routed) to those nodes in the network with the highest probability of improving the efficiency and effectiveness of the information search query.

Without these solutions, peer-to-peer network traffic grows exponentially, the probability of satisfying information needs drops, and search results appear unreliable because two identical search requests return different results depending upon where they were inserted into the network.

6. Expected Results

The proposed research will develop and evaluate centralized and local approaches to acquiring information about the characteristics of nodes in a peer-to-peer network; a particularly challenging type of information is information about the content at each node, but we will also develop and evaluate methods for acquiring information about other node characteristics, such as reliability and responsiveness. The proposed research will develop decision-theoretic utility functions that enable nodes to optimize their decision-making with respect to their specific priorities. This work will lead to a better understanding of i) information gathering and decision-making in large-scale peer-to-peer networks; ii) network conditions in which optimizing individual utility is consistent with globally desirable conditions such as reduced network congestion; and iii) network characteristics that foster spontaneous formation of virtual communities through the individual decision-making of nodes about where to connect and how to relay messages in the network.

The research will establish research methodologies and produce a simulation environment for studying large-scale peer-to-peer networks in a systematic manner. This simulation environment will also be used in graduate professional Masters degree programs, to provide hands-on experience with peer-to-peer networking, virtual communities, and virtual marketplaces (discussed below).

References

- M. Ben-Akiva and S. R. Lerman. *Discrete Choice Analysis: Theory and Application to Travel Demand*, The MIT Press, Cambridge, Massachusetts, 1985.
- J. Callan. "Distributed information retrieval." In W. B. Croft, editor, *Advances in Information Retrieval* (pp. 127-150). Kluwer Academic Publishers. 2000
- J. Callan and M. Connell. "Query-based sampling of text databases." *ACM Transactions on Information Systems*. Accepted.
- J. Callan, M. Connell, and A. Du. "Automatic discovery of language models for text databases." In *Proceedings of the ACM-SIGMOD International Conference on Management of Data*, (pp 479-490), Philadelphia: ACM. 1999.
- J. P. Callan, Z. Lu and W. B. Croft. "Searching distributed collections with inference networks." In *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 21-28). ACM. 1995.
- J. Callan, A.L. Powell, J.C. French, and M. Connell. "The effects of query-based sampling on automatic database selection algorithms." *Twenty Fourth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* (Under review.)
- Bradley P. Carlin, Thomas A. Louis. *Bayes and Empirical Bayes Methods for Data Analysis*, CRC Press, 2000.
- A. E. Cha. "E-power to the people." *The Washington Post*, p. A01, May 18, 2000.
- I. Clarke. "Freenet - Front page." http://freenet.sourceforge.net/, August 2000.
- I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. "Freenet: A distributed anonymous information storage and retrieval system." *ICSI Workshop on Design Issues in Anonymity and Unobservability*", Berkeley, California, July 25-26, 2000.
- L. Gravano, H. Garcia-Molina and A. Tomasic. "GlOSS: Text-Source Discovery over the Internet." *ACM Transactions on Database Systems*, 24(2) (pp. 229-264). ACM. 1999.
- Donald Gross, Carl M. Harris, *Fundamentals of Queueing Theory*, Wiley Series in Probability and Mathematical Statistics, 1997.
- D. Hawking, N. Craswell and P. Thistlewaite. "Overview of TREC-7 Very Large Collection track." In *The Seventh Text REtrieval Conference (TREC-7)* (pp. 91-104). NIST Special Publication 500-242. 1999.
- M. O. Jackson and A. Wolinsky, "A Strategic Model of Social and Economic Networks", *Journal of Economic Theory* (Vol. 71, pp. 44-74), 1996.
- K. J. Lancaster, "A new approach to consumer theory" (Vol 74., pp. 132-57), *Journal of Political Economy*, 1966.
- L. Larkey, M. Connell, and J. Callan. "Collection selection and results merging with topically organized U.S. patents and TREC data". In *Proceedings of the 9th International Conference on Information and Knowledge Management (CIKM)* (pp. 282-289). ACM. 2000.
- P. Lazarsfeld and R. Merton, "Friendship as Social Process: A Substantive and Methodological Analysis", in *Freedom and Control in Modern Society* (pp. 18-66), eds. M. Berger et al., New York: Van Nostrand. 1954.
- A. Oram, *Peer-to-Peer: Harnessing the Power of Disruptive Technologie, O'reilly Publishing, 2001.*
- A. Montgomery, K. Hosanagar, R. Krishnan, K. Clay, "Intelligent Shopbot Design", Technical Report, Institute for Complex Engineered Systems and Heinz School, Carnegie Mellon University, 2001.
- J. Ponte and W. B. Croft. "A Language Modeling Approach to Information Retrieval." In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 275-281). ACM. 1998.
- P. E. Rossi, R. E. McCulloch, and G. M. Allenby, "On the Value of Household Information in Target Marketing" (Vol. 15, pp. 321-340), *Marketing Science*, 1996.
- G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill. 1983.
- Schafer, J.B.; Konstan, J.; and Riedl, J. forthcoming, "Electronic commerce recommendation applications," Journal of Data Mining and Knowledge Discovery. Retrieved December 9, 2000 from www.cs.umn.edu/Research/GroupLens/ECRA.pdf.
- S. M. Shugan, "The Cost of Thinking", *Journal of Consumer Research* (Vol. 7, pp. 99-111), 1980.
- Wego.com Incorporated. "Welcome to gnutella." http://gnutella.wego.com/, August 2000.
- B. Wellman and S. Berkowitz, "Social Structure: A Network Approach", Cambridge University Press, Cambridge, 1988.