

Analytical Model for Semantic Overlay Networks in Peer-to-Peer Systems

BERTALAN FORSTNER, HASSAN CHARAF
Department of Automation and Applied Informatics
Budapest University of Technology and Economics
H-1111, Budapest, Goldmann György tér 3.
HUNGARY

Abstract: - The hit rate of basic Peer-to-Peer protocols can be increased by using semantic metadata to transform the network structure or construct an overlay network. These semantic protocols benefit from the observation that the pattern of the stored or queried documents depends on the fields of interest of the nodes in the network. However, the performance of such networks highly depends on boundary conditions. These conditions determine both the probability of a successful query and the network traffic generated by the queries. In this paper, we will propose an analytical model that calculates the success rate of a query in a peer-to-peer system with a semantic overlay network. With the model, it is possible to optimize these systems by finding the appropriate values for the boundary conditions when designing the network or at runtime. The model has also been validated with a series of simulations.

Key-Words: - Modeling, Peer-to-Peer, Semantic overlay networks, Simulation

1 Introduction

The hit rate of basic peer-to-peer (P2P) protocols can be increased by using semantic metadata to transform the network structure or construct an overlay network. The new protocols introduce new connections between nodes with similar fields of interests, these are the semantic connections. The SemPeer protocol [1], for example, improves the performance of a P2P network by constructing semantic profiles for the nodes, based on the stored documents' metadata followed by a comparison of the semantic similarities on generalized topics. These topic generalizations are made with an adaptive algorithm that utilizes a thoroughly tested full-language taxonomy.

However, the performance of such networks highly depends on boundary conditions such as the number of off-topic queries or the number of semantic and standard connections per node. Off-topic query means that a node searches for a document that does not belong to the node's fields of interest. Another parameter, the TTL (Time-to-Live), highly determines the amount of network traffic. This value describes how many hops a query is propagated in the network.

In a SemPeer network, the nodes can issue queries for documents in their field of interest or for off-topic materials. With the observation of different nodes it is possible to gather statistics on this behavior. Let $P_{issued,offtopic}$ stand for the probabilities of off-topic queries and $P_{issued,topic}$ for the queries in the node's fields of interest. In view of the approximate value of these parameters, we will give a formula to calculate the probability of a successful query for a given node. This

formula will help to construct the optimal TTL strategy or it provides the number of semantic links to maximize the results. A node can use this information during its lifecycle to reconfigure itself for optimal performance.

2 Backgrounds and Related work

Considerable research effort has been involved in the examination of the performance of networks with client-server architecture [2]. There are some models elaborated to analyze the throughput, response time, and other parameters of the network. However, there are only very few papers concerning these issues of P2P networks. The main research directions can be characterized by the following types of network models.

The aspects of connection distribution of the large-scale P2P networks are modeled in [3]. This work describes the measures that affect the quality of service of the network, such as network latency or the short-circuit effect. However, it does not answer questions such as the probability of success or the influencing parameters.

We found a quite useful model in [4]. The main goal of this model was to capture network throughput for three different classes of P2P networks. The one that describes the P2P architecture of distributed indexing with flooding architecture is suitable to obtain probabilistic results for Gnutella networks. However, it can hardly be transformed to use with clustered SemPeer networks, but we can use it to validate new models in extreme cases, as we will do it later in this paper.

The model in [5] gives a formula to calculate the cost of a query, but the way in which the clustering in the

propagation tree is estimated is not sophisticated enough to be used with semantic overlay networks, nor can the formula give an approximation to the success rate. Similarly, [6] deals only with the traffic measurement aspects.

After examining the available models we found that there are no such analytical P2P models that can be used to describe semantic overlay networks. In order to promote the design of these advanced protocols, we decided to prepare an advanced analytical model for them.

2.1 Semantic overlay networks

2.1.1 The SemPeer protocol

To improve the performance of Peer-to-Peer networks, we designed a protocol, SemPeer [1], that constructs a semantic layer over unstructured P2P network. We use this protocol mainly later in the paper to validate the analytical model. In this section, we will give an overview of the protocol in order to subserve the understanding of the results.

SemPeer uses thoroughly tested algorithms (for example [7][8]) to extract metadata from the stored documents at each node. From these metadata each node constructs a semantic profile that describes its fields of interest. Then the nodes will be connected by a certain number of semantic links, that is, links to nodes with similar fields of interest. Each time a node is obtaining a query hit message, it compares its own semantic profile with that of the originator node, and adds the connection to the list of semantic links when the similarity is rather high, and the link satisfies some circumstances to avoid clustering in the propagation tree. The queries are flooded in the standard (in our case, for example, Gnutella) protocol layer, and also propagated on the semantic links at the same time. For that reason there is no need for preliminary information on the query.

The performance of the P2P network can highly be increased using the SemPeer protocol. However, it can be fine tuned by setting the protocol parameters according to the network properties and query patterns. These parameters will be described in more depth in the next sections.

2.1.2 The Shortcuts

Another proposal is the Shortcuts [9] overlay network which can be regarded as a greedy algorithm that maintains a list of shortcuts for nodes with probably similar fields of interest. The protocol supposes that nodes that answered queries in the past will be able to answer other queries in the future. For this reason, each node sends out its queries through these links first.

Unlike SemPeer, this protocol does not use any quantitative measure to determine the similarity of the nodes; the decisions are made based on the number of queries answered by each node in the past. Consequently, as the analytical model and the simulations will show us, the SemPeer protocol can perform better.

However, shortcuts have good load distribution properties. The overall load is reduced, and more load is redistributed towards the peers that make heavy use of the system. In addition, shortcuts help to limit the scope of queries. Shortcuts are scalable, and incur very little overhead.

As the SemPeer protocol is more sophisticated, we will use it primarily to show the synthesis of our analytical model. However, both the summarized protocols and also Gnutella will be used to validate the results.

3 Clustering in the propagation tree

To understand our new model, a phenomenon has to be explained. Clustering in the propagation tree means that due to the high connectedness of the nodes, a query can arrive to a node more than once. The high connectedness can be caused by the small amount of nodes in the network or in a specific field of interest. The high number of connections per node also causes clustering. In case of semantic protocols such as SemPeer, this clustering in the propagation tree is much higher, because the nodes with similar fields of interest form clusters from a subset of the whole network. This kind of clustering has to be avoided in order to lower the network traffic and increase the performance of the network.

To be able to describe the connectivity of a graph in a formal way, we use a modified version of the clustering coefficient graph measure introduced by Watts and Strogatz [10].

The original formula can be introduced as follows. Firstly, we define a graph in terms of a set of n vertices $V = \{v_1, v_2, \dots, v_n\}$ and a set of edges E , where e_{ij} denotes an edge between vertices v_i and v_j . Below we assume that v_i , v_j and v_k are members of V .

Secondly, we define the *neighborhood* N for a vertex v_i as its immediately connected neighbors as follows:

$$N_i = \{v_j\}: e_{ij} \in E \quad (1)$$

The degree k_i of a vertex is the number of vertices in its neighborhood $|N_i|$.

The clustering coefficient C_i for a vertex v_i is the proportion of links between the vertices within its neighborhood divided by the number of links that could possibly exist between them. For a directed graph, e_{ij} is distinct from e_{ji} , and therefore for each neighborhood N_i there are $2k_i(k_i - 1)$ links that could exist among the

vertices within the neighborhood. Thus, the clustering coefficient is given as:

$$C_i = \frac{|\{e_{jh}\}|}{2k_i(k_i-1)} : v_j, v_h \in N_i, e_{jh} \in E \quad (2)$$

This measure equals 1 if each neighbor connected to v_i is also connected to each other vertex within the neighborhood, and 0 if no vertex connected to v_i is adjacent to any other vertex connected to v_i .

Because of the nature of the Gnutella-based protocols, the high connectivity of the nodes with similar semantic profiles could lead to a very high clustering coefficient. This results in a query that arrives multiple times in different ways to some of the nodes in the group. Because of the connectedness, fewer nodes can be reached by a query, and also unnecessary computational resources are required. This can be described in a more formal manner as follows.

Consider a set of nodes where the clustering coefficient equals zero, i.e. no neighbors are connected with each other (Figure 1.a). In this case the number of nodes that a query can reach is written as

$$M = \sum_{i=1}^{TTL} k^i \quad (3)$$

In Eq. (3), TTL represents the Time-To-Live parameter: a query should be propagated through TTL hops. Now we consider the worst case, when the clustering coefficient equals 1. In this case the neighboring nodes form a fully connected directed graph, thus, the number of nodes reached by a query are decreased to k (Figure 1.b)

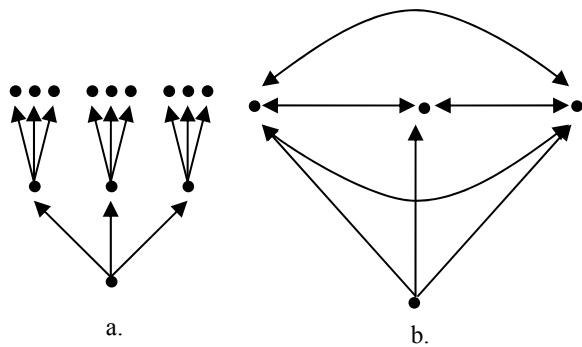


Figure 1: Directed graphs with extreme clustering coefficients, $k=3$, $TTL=2$. a.: $C=0$ b.: $C=1$

In a standard Gnutella network, the coefficient is nearly zero as the graph can be regarded as a random mesh. In case of SemPeer, this measure can be quite high, depending on the popularity of the given group. It can even happen that after certain queries the standard SemPeer reaches a saturation point: the clustering coefficient reaches a value, where the number of nodes reached by a query is strongly decreased, superseding

the benefit from the intelligent neighbor selection. Under bad circumstances, it can occur that the SemPeer protocol delivers fewer positive answers than Gnutella does.

In the case of much clustered subnetworks not only the high connectedness of the neighbors causes a problem, but also some other types of links. Firstly, the links *backward* in the propagation tree appear. In this case a node forwards a message back to a node that already propagated it in an earlier time. Secondly, links between nodes in the same level (*sibling links*, in our wording) appear. This causes that node A forwards a query to node B which received the query after the same number of hops (hop number) than node A did. The third type of counterproductive connections is a link to neighbors of a sibling node (*skew links*). In that case a node receives the same query with the same hop number from different nodes.

All three types can be seen on the graph representation marked with dotted lines in Figure 2.

These links can certainly occur in the propagation tree of any P2P network, however, the semantic overlay networks often transform the graph in such way that the number of these counterproductive links increases because of the small world phenomenon.

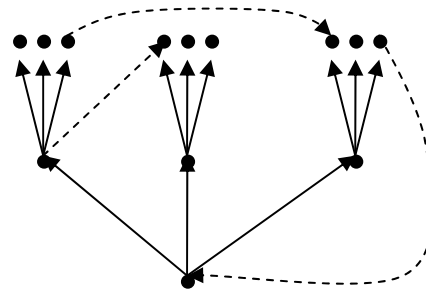


Figure 2: The dotted links decreases the number of reached nodes

To be able to measure this kind of connectedness, we introduce a modified clustering coefficient. This measure must be 0 if the nodes reached by a query constitute a tree, and it approaches 1 as the number of the counterproductive links increases.

Let $\{E_r^*\}$ stand for the set of counterproductive links in the propagation tree of a query initiated by the node v_r . For the sake of simplicity, we assume that each node has the same number of neighbors. In that case the modified clustering coefficient for the node r is

$$C_r = \frac{|\{E_r^*\}|}{\sum_{m=1}^{TTL} \left[k^m \left(\sum_{n=0}^{m-1} \binom{m-1}{n} k^n + k^m - 1 \right) \right] + \sum_{m=1}^{TTL-1} k^m (k^{m+1} - k)} \quad (4)$$

The explanation of this formula is the following. The denominator in Eq. 4 is the maximum number of the three different types of counterproductive links in a

query. These three types are detailed below. The maximum number of backward links is

$$\sum_{m=1}^{TTL} \left[k^m \left(\sum_{n=0}^{m-1} (k^n) \right) \right] \quad (5)$$

because there are k^m nodes reached by the query at the m^{th} step from issuing, and such a node can be connected with all the nodes visited in previous steps. The count of all the possible sibling links is

$$\sum_{m=1}^{TTL} [k^m (k^m - 1)] \quad (6)$$

because a node reached in step m can be connected to the other nodes ($k^m - 1$) reached in the very same step. Finally, the maximum number of skew links is counted to be

$$\sum_{m=1}^{TTL-1} k^m (k^{m+1} - k) \quad (7)$$

Recall that Eq. 3 gives the number of nodes reached by a query. The defined number cannot be reached because of the effect of clustering in the propagation tree: the same message is delivered to a node more than once. In [3], there is an experiment with the Gnutella protocol, where the proportion of the practical and theoretical number of the reached nodes is computed, based on snapshots of the Gnutella network. The reach is measured with different TTL values. In the typical experiment, there are 10.000 nodes with average 4 connections each (Table 1). We were able to reproduce these results with the GXS Simulator [14]. We found that the increase of the proportion at higher TTLs is owed to the fact that the theoretical number of the reached nodes approaches or overwhelms the number of nodes in the network.

TTL	Reach
1	100%
2	100%
3	98%
4	78%
5	66%
6	81%
7	91%

Table 1: Short-circuiting effects on a random topology (nodes = 10000, edges = 40000)

4 Success rate

4.1 Homogeneous Network

Using the clustering coefficient presented in the previous chapter, we will calculate the probability of a successful query in a homogenous network. The homogeneity of the

network in this context means that there are only semantic or only non-semantic links in the network.

We regard the P2P network as a directed graph. The fields of interest of a user represented by a node can be determined with a single topic. We assume t different topics, the same number of nodes (V_t) and documents (D_t) with each topic, and every node having D_n documents. The documents and the initiating links between the nodes are selected randomly with uniform distribution.

We can approximate the probability of a successful query in case of Gnutella as shown below:

$$P_{Success, Gnutella} = 1 - \left(1 - \frac{1}{tD_t} \right)^{D_n E_q} \quad (8)$$

In this formula, we compute the probability of not finding the requested document at any reached node, and subtract it from 1. The expression in parenthesis is the probability of selecting any disinterested document from all the documents that exist in the network. E_q is the number of reached nodes:

$$E_q = \sum_{i=1}^{TTL} [(1-C)k]^i \quad (9)$$

where C , the modified clustering coefficient is the average of all the C_r s.

A link in the propagation tree can be counterproductive with the following probability:

$$P = \frac{\sum_{m=l}^{TTL} \left[k^m \left(\sum_{n=0}^{m-1} (k^n) + k^m - 1 \right) \right] + \sum_{m=1}^{TTL-1} k^m (k^{m+1} - k)}{t V_t \sum_{i=0}^{TTL} k^i} \quad (10)$$

In this formula, we calculated the ratio of the number of possible counterproductive links in the propagation tree of a node and the count of all the links in every node's independent propagation trees. This shows the probability that a link in the network acts as a counterproductive link in a node's propagation tree. This formula is valid because we regard the basic Gnutella network as a random mesh. From that the average number of productive connections per node follows:

$$(1-C)k = k * \left(1 - \frac{\sum_{m=1}^{TTL} \left[k^m \left(\sum_{n=0}^{m-1} (k^n) + k^m - 1 \right) \right] + \sum_{m=1}^{TTL-1} k^m (k^{m+1} - k)}{t V_t \sum_{i=0}^{TTL} k^i} \right) \quad (11)$$

From (11), C can be approximated obviously by the following fraction:

$$C = \frac{\sum_{m=1}^{TTL} \left[k^m \left(\sum_{n=0}^{m-1} (k^n) + k^m - 1 \right) \right] + \sum_{m=1}^{TTL-1} k^m (k^{m+1} - k)}{t V_t \sum_{i=0}^{TTL} k^i} \quad (12)$$

In the case of SemPeer, we regard the steady state when nodes are connected only to other nodes from the same cluster. Therefore, a search is performed only in the set of documents related to only one topic, but the clustering coefficient also rises, because the multiplier t in the denominator of the Formula 12 decreases to 1. The approximate probability of a successful query is as follows:

$$P_{\text{Success,SemPeer}} = 1 - \left(1 - \frac{1}{D_t}\right)^{D_n E_q} \quad (13)$$

4.2 Inhomogeneous Network

Let P_i stand for the probability that a node on the i^{th} level of the propagation tree has the same topic as the originator node. To be able to approximate this value, we introduce P_{SemLink} , which denotes the probability that an outlink points to a node with similar topic. In reasonable cases this can be regarded as a system parameter and should be defined according to the ratio of the issued relevant and off-topic questions. Practically, this parameter is the ratio of semantic outlinks and the total number of outlinks per node. $P_{\text{SameTopic}}$ is the probability that a random (off-topic) connection points to a node with the same field of interest as the originator. When we suppose that there are t different fields of interest with uniform distribution, this value equals $1/t$. The P_0 value obviously equals 1.

$$P_i = P_{i-1} P_{\text{SemLink}} + P_{i-1} \bar{P}_{\text{SemLink}} P_{\text{SameTopic}} + \bar{P}_{i-1} \bar{P}_{\text{SemLink}} P_{\text{SameTopic}} = P_{i-1} P_{\text{SemLink}} + P_{i-1} \bar{P}_{\text{SemLink}} P_{\text{SameTopic}} \quad (14)$$

With these formulas, we can approximate the probability of finding a given document in two cases: when the document is off-topic for the originator node or when it is not. When searching for a document that is in the field of interest of the originator node, the probability can be calculated as follows:

$$P_{\text{success,topic}} = 1 - \prod_{i=1}^{TTL} \left(1 - \frac{1}{D_t}\right)^{D_n P_i ((1-C)k)^i} \quad (15)$$

When the node issues a search for an off-topic question, the probability of a positive answer will be:

$$P_{\text{success,offtopic}} = 1 - \prod_{i=1}^{TTL} \left(1 - \frac{1}{tD_t}\right)^{D_n \bar{P}_i ((1-C)k)^i} \quad (16)$$

When the pattern of the issued queries is known, that is, $P_{\text{issued,offtopic}}$ stands for the probabilities of off-topic queries and $P_{\text{issued,topic}}$ for the queries in the node's field of interest, the aggregated value will give the probability of a successful query:

$$P_{\text{success}} = P_{\text{issued,topic}} P_{\text{success,topic}} + P_{\text{issued,offtopic}} P_{\text{success,offtopic}} \quad (17)$$

Now it is possible to state the result, which is a key step in calculating the probability of a successful query with

different parameter variables in a semantic peer-to-peer network.

Proposition. In the case the pattern of the issued queries is known, the following formula holds:

$$P_{\text{success}} = P_{\text{issued,topic}} \left(1 - \prod_{i=1}^{TTL} \left(1 - \frac{1}{D_t}\right)^{D_n P_i ((1-C)k)^i}\right) + P_{\text{issued,offtopic}} \left(1 - \prod_{i=1}^{TTL} \left(1 - \frac{1}{tD_t}\right)^{D_n \bar{P}_i ((1-C)k)^i}\right) \quad (18)$$

5 Model validation

5.1 Comparison with other models

As described in the introduction, we compared our model to the one introduced in [4] when validating the results of the Gnutella case. The simplified version of the model described in that paper proposes the following formula for the probability of a successful query:

$$P_{\text{Success,Gnutella}} = \frac{TTL}{(V_t t)^{1/\beta}} \quad (19)$$

where $\beta > 1$ is a parameter related to the connectivity of the topology formed by the peers. As in the case of the Gnutella architecture (distributed indexing with flooding architecture) the value TTL^β equals to the number of nodes reached by a query, and we assumed the clustering zero in the random graph, the β parameter can easily be calculated from our parameters from the following formula:

$$\beta = \log_{TTL} E_q \quad (20)$$

The comparison with this model in the reasonable cases gave the same probability as the model (see also the next subsection). We have not found such a model that can be used for comparison in the semantic overlay network cases.

5.2 Model validation with simulation results

To validate our results and examine the behavior of our protocol, we have evaluated a series of simulations on the GXS Peer-to-Peer Simulator [14]. Practically, GXS is a single-threaded message dispatching utility operating in batch mode. A simulation itself is a script file containing a sequence of default (built-in) and protocol-specific (user-implemented) commands; and the results are dumped into a stream. GXS uses the concept of steps, which means that it handles a whole set of parallel events timed to a given step number. The simulator supports two types of events: message arrival and expiration of timers. The events are handled in a first-come-first-served manner, since shuffling of

'parallel' events has no statistically recognizable effect on the results.

We assigned a field of interest to every simulated node, and generated documents with different keywords in these areas. After the initialization phase, every node starts to query for documents that it does not have.

The series of simulations involved typical cases in network size and node behavior. In Table 1 we conclude the results of a specific test case with different protocols and parameters, and compare the results with the predictions of the model. The network size was 48.000 nodes, there were 20.000 documents in the system, and each node hold an average of 20 document and 5 links. The TTL value was 5. In case of the Shortcuts protocol, the size of the shortcut list was 5 and 10. The average ratio of the off-topic queries was 20%. In case of the Shortcuts protocol, the success rate at 1 hops distance is indicated, because in more hops distance it has the same value as Gnutella.

Table 1.

Protocol/Scenario	$P_{success,simulation}$	$P_{success,model}$
Gnutella	0,546	0,546
SemPeer w/1 sl*	0,577	0,590
SemPeer w/2 sl*	0,677	0,706
SemPeer w/3 sl*	0,803	0,828
SemPeer w/4 sl*	0,863	0,844
SemPeer w/5 sl*	0,849	0,833
Shortcuts w/5 sl*	0,063	0,059
Shortcuts w/10 sl*	0,119	0,114

*sl: semantic link

The appropriateness of the model is quite good, and the digression is less than 5 percent. In Figure 3, the prediction of the model and the simulations results are shown. From the model, it follows that in this particular case the system performance is maximized when there are four semantic links out of five connections per node, and the answer rate is approximately 0.85.

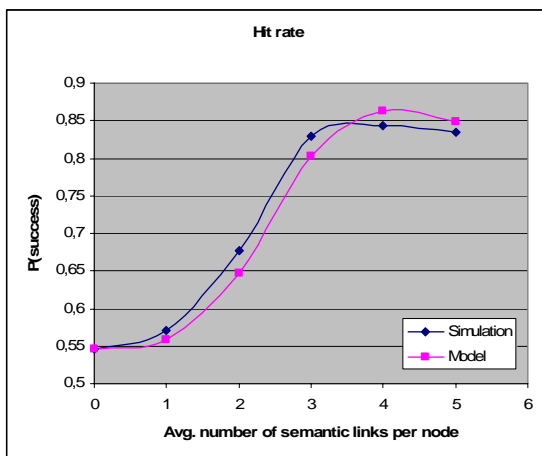


Figure 3. Comparing the results of the simulator and the model

6 Results

The performance and communication costs of P2P systems with semantic overlay networks highly depend on boundary parameters that should be defined when designing the system or even at runtime. In this paper, we presented an analytical model that can predict the success rate of these networks with different parameters, making possible to fine-tune the system without more computational expense.

With the model, it is also possible to compare the performance of different semantic overlay networks without running a huge amount of simulations.

The paper also discusses case studies that present the use of model results to optimize a network.

The model can be extended further to use with structured overlay networks.

References

- [1] Bertalan Forstner, Hassan Charaf, "Neighbor Selection in Peer-to-Peer Networks Using Semantic Relations", WSEAS Transactions on Information Science & Applications, 2(2), February 2005. ISSN 1790-0832, pp. 239-244
- [2] D. A. Menasc'e, V. A. F. Almeida, and L. W. Dowdy, "Capacity Planning for Web Services: metrics, models, and methods". Prentice Hall, 2001.
- [3] M. A. Jovanovic, F. S. Annexstein, and K. A. Berman, "Modeling peer-to-peer network topologies through "small-world" models and power laws," IX Telecommunications Forum, TELFOR, 2001.
- [4] Z. Ge, D. R. Figueiredo, S. Jaiswal, J. Kurose, and D. Towsley, "Modeling Peer-Peer File Sharing System" In Proceedings of INFOCOM 2003, San Francisco, USA, Apr 2003.
- [5] B. Yang and H. Garcia-Molina. Efficient search in peer-to-peer networks. In *Proc. of the 22nd Int. Conf. on Distributed Computing Systems*, July 2002.
- [6] Analyzing peer-to-peer traffic across large networks, Sen, S.; Jia Wang; Networking, IEEE/ACM Transactions on Volume 12, Issue 2, April 2004
- [7] H. Assadi, "Construction of a Regional Ontology from Text and Its Use within a Documentary System International Conference on Formal Ontology and Information Systems", FOIS-98, IOS Press, Amsterdam (WebDB-2000), Springer-Verlag, Berlin, 2000,
- [8] J.-U. Kietz, A. Maedche and R. Volz, "Semi-Automatic Ontology Acquisition from a Corporate Intranet", Proc. Learning Language in Logic Workshop (LLL-2000), ACL, New Brunswick, N.J., 2000
- [9] K. Sripanidkulchai, B. Maggs, H.Zhang, "Efficient content location using interest-based locality in peer-to-peer systems", Infocom, 2003
- [10] D. J. Watts, S. H. Strogatz, "Collective Dynamics of 'Small-World' Networks", Nature, 393 (1998), pp. 440-442