# Peer-to-Peer Protocol Evaluation in Topologies Resembling Wireless Networks. An Experiment with Gnutella Query Engine

Balázs Bakos, Gergely Csúcs, Lóránt Farkas and Jukka K. Nurminen
Nokia Research Center, P.O. Box 392, H-1461 BUDAPEST, Hungary
{balazs.bakos, lorant.farkas}@nokia.com
Budapest University of Technology and Economics, P.O. Box 91, H-1521 BUDAPEST, Hungary
wizard@aut.bme.hu
Nokia Research Center, P.O. Box 407, FIN-00045 NOKIA GROUP, Finland
jukka.k.nurminen@nokia.com

*Abstract*–**Peer-to-peer (P2P) protocols implemented in wireless networks and mixed scenarios would generate new applications and value added services. In order to analyze the performance of such protocols the specific topologies of current and future wireless networks should be taken into account.**

**In our paper we analyze the performance of a Gnutella-type protocol query engine on random mesh (Random), semi-random mesh (Semi), connected stars (ConStars) and connected mesh (ConMesh) application layer network topologies. In addition we discuss and simulate for the considered topologies the effect of adaptive time-to-live (TTL) techniques. We take into account in our analysis the document replication phenomenon.**

**Within this paper we limit ourselves to the simulation of static network topologies.**

**The results of this study show that Semi and ConStars are the most useful topologies. Semi is the best topology for heterogeneous devices, e.g. all of the nodes are mobile phones. ConStars is useful when the network consists of devices with different capabilities, e.g. mobile phones and PC-s. If the node degrees are chosen appropriately the bandwidth needs can reasonably small making content sharing peer-to-peer applications feasible in pure wireless networks and mixed scenarios.**

**Keywords and abbreviations: peer-to-peer (P2P), adaptive time-to-live (TTL), content sharing, Gnutella, wireless communication, performance analysis, topology, underlying physical network, application layer network.**

## I. Introduction

Up to now the successful peer-to-peer applications have been running on the wired Internet. As mobile phones and other small devices are increasingly starting to resemble computers, the question can be formulated how the peer-to-peer activities could be scaled down to such devices. There are three typical resources that can be shared via a P2P system: computing power, storage capacity and user presence. Computing power is not the strongest part of current handheld devices, but the latter two resources fit well into the concept of upcoming cellular phones.

Content sharing applications such as Napster [1], Gnutella [2], and FastTrack [3] are among the most successful peer-to-peer (P2P) applications. While a lot of the early use has been infringing copyrights we can envision that such systems could also be increasingly used to transfer user-created and other legal content.

A key difference between Napster, FastTrack, and Gnutella is the level of centralization. The Gnutella protocol has been proposed for file sharing without the need for any centralized administration or server [2]. Contrary to Napster, which is fully centralized [1], and FastTrack, which is partly centralized [3], all Gnutella peers are equal and no centralized servers are used [2]. In addition, Gnutella network is completely unstructured, as the joining of nodes follow simple loose rules [2, 12]. The drawback of the decentralization is increasing load to the network. The drawback of unstructured topology is inefficient use of links. According to the estimation in [4] the two main P2P networks, Kazaa and Gnutella, now make up 40 to 60 percent of all Internet traffic.

The main goal of our research is to evaluate the network load caused by content sharing P2P applications in wireless networks. The rationale behind this work is that once we know which topology has good performance properties, we can add topology maintenance functionality to the protocol so that it will create and maintain the ideal application layer network topology.

In this paper we have focused on the Gnutella protocol. The main reasons were its simplicity and its availability in terms of algorithm and implementation. According to the measurement in [17] the search functionality of Gnutella protocol generates slightly more than half of the total amount of Gnutella traffic. This result encouraged us to investigate in particular the querying engine of the Gnutella protocol. A peer-to-peer protocol simulator has been developed to study

- the effect of P2P network topology to Gnutella performance,
- how adaptive time-to-live (TTL) techniques affect the performance of Gnutella for the considered topologies.

The other aspects related to Gnutella and the underlying physical network are not taken into account in this phase of our work.

This paper has 7 sections. In section 2 we review the Gnutella network topology as reported in the literature. The third section focuses on the underlying network topologies along with maintenance issues of their application-level replicas. The fourth section contains a review of the ongoing research regarding Gnutella performance and P2P content sharing protocols' performance, more generally. Section 5 presents the simulation environment and the simulation results. Section 6 contains the discussion of the results. Section 7 concludes the discussions and draws further research directions to be considered.

## II. Reported Gnutella network topology

The topology of the wired Gnutella network has been widely studied. The topology has a self-organizing character dependent on the user behavior and on the different solutions adopted by different Gnutella implementations. The topology evolution is perceptible not only on small scale but on statistical scale as well. The specific aspects and ongoing research activity related to it are briefly described in the following.

Measurements effectuated in November-December 2000 by Jovanovic et al. [5] indicated a type 2 power-law topology (frequency of node degree proportional to the node degree to the power of an exponent) of the Gnutella with node degree (number of links that have a node as their initial vertex) exponent of -1.6 during November and of -1.4 during December, meaning that the average node degree statistically changed in the given time period. The experiments done by Ripeanu et al. [6] in the same year but over a longer period indicate a systematic shift in time from the type 2 power-law topology to a multimodal one. This is an indication of a strong evolution of the network in the given time period. These two measurement results indicate that Gnutella network changes its statistics even during as short time interval as a month.

Annexstein et al. report in [7] the adversive effect of variable link latencies on the t-horizon size (related to the number of reachable nodes) of networks exposed to the so-called short-circuiting phenomenon (applies to Gnutella). Their results indicate a pronounced diminution of the t-horizon in the case of networks like Gnutella.

Ripeanu et al. also analyze in [6] the difference between Gnutella network topology and the underlying physical topology of wired networks. According to their conclusion, the self-organizing Gnutella network does not use efficiently the underlying physical infrastructure and this leads to a serious traffic overhead.

From this short review it can be concluded that most of the researchers focus their analysis on the existing wired Gnutella topologies and on their performance.

## III. Wireless Network Topologies

Four physical network topologies are identified to which the further discussions will be related.

### *A. Random Mesh (Random)*

Random is the most basic topology. Nodes may participate in the network without having any connections at all, and the nodes do not consider any parameter of a given connection (reachable nodes, etc). It is expected to appear when there is no maintenance available in the network or the link maintenance algorithms fail.

When inspecting dynamic behavior, preserving Random topology in the application layer needs no maintenance.

### *B. Semi-Random Mesh (Semi)*

The Semi topology ensures that every node is actively participating in the network. Participation means here having at least one connection.

Even if there are no isolated nodes the semi topology can result into separate islands. The probability of having separate islands and the number of islands depend on the average and standard deviation of the node degree.

If there is only one island, the Semi becomes a Connected Mesh, which is the next discussed topology.

Semi represents the idea that joining a network requires some kind of action; a node has to initiate a connection if it wants to be considered a participant of the network.

Semi is expected to appear when several isolated networks exist: one wired network, ad-hoc networks, Bluetooth networks or combined situations with no connection between the individual networks.

### *C. Connected Mesh (ConMesh)*

ConMesh assumes that all nodes belong to one connected network. Each node is connected to a randomly chosen participant of the existing network. This method builds a tree into which additional connections can be added.

We expect ConMesh to be appropriate in 2 different cases, namely: two different underlying technologies are used (local protocol – Bluetooth; external connection protocol – GPRS) or only one technology is used (Bluetooth; ad-hoc networks).

### *D. Connected Stars (ConStars)*

ConStars is made of core nodes that typically form a ConMesh topology, and of leaf nodes connected to the core nodes.

ConStars topology is likely to arise in practice in two cases. One scenario is that the core nodes are not sharing any documents. The resulting network thus resembles an IP network with separate hosts and routers. This is the case of the current 2.5-generation mobile networks (e.g. GPRS) and of the future wireless networks implementing mobile IP. The GPRS traffic always passes through one specific router, the

GPRS Gateway Support Node (GGSN) [18].

In the second scenario core nodes are also sharing files and the core nodes and leaf nodes run the same software. It is not determined beforehand which nodes will participate in the core network. This decision can be made by the users (enthusiasts can always choose to be core nodes) or by the protocol itself (based on bandwidth and uptime data). This topology does not match any physical topology.

## IV. Gnutella performance analysis in the literature

According to [6], a P2P content sharing application should have the following features:

- anonymity (protecting user privacy),
- reliability (robustness to external attacks),
- ability to operate in a dynamic environment and
- performance and scalability.

Anonymity and reliability in terms of robustness to external attacks are discussed in a number of contributions that are not strictly related to P2P applications [8]. There are also several contributions that discuss anonymity in the context of P2P applications [9, 14, 16].

Chu et al. [10] measure the locality of content of Gnutella and Napster networks. Their result indicate a log-quadratic distribution of the content for both Napster and Gnutella, which is prone to caching as much as Zipf's distribution, which approximates web content locality.

Saroiu et al. [11] and Chu et al. [10] measure the node availability during short and long time periods. Their results match each other: in average short session durations occur.

Lv et al. study in [12] the search performance, generated traffic and query overhead in unstructured P2P networks for different topologies, replication techniques and content popularity distributions, met in the most popular P2P content sharing protocols.

Our approach follows the approach in [12], but we are mainly interested in the behavior of P2P message propagation in a bandwidth-aware environment. This is unique in terms of granularity: we experiment with average node degrees are varying from 2 to 4, which is significantly lower than that from [13]. This is expected to occur in wireless environment having limited bandwidth.

## V. Simulation results

### A. Simulator details

For the performance analysis we developed a small and flexible simulation framework. The simulator is implemented with Java. A benefit of the Java implementation is that the same code works both in the simulator and in Java enabled mobile devices. The simulator has a skeleton implementation of a file-sharing application with well-defined interfaces.

### B. Validation and test network size

For validation purposes, we have reproduced the results of Hong [14], a random mesh with average node degree of 3. Twenty replicates of each document were stored in the network and 300 queries were issued requesting random (but existing) documents. The outputs were the path length and number of nodes contacted until the first hit was found.

When the network size was 1000 as in [13] the increment rate abruptly dropped after the 5th step. This is a sign that the network was saturated. The nodes of a 1000-node network typically received multiple copies of the same query (3 - 4 on the average). Since the duplicated queries are dropped by the Gnutella protocol less new packets were created.

With the larger networks (with 10000 and 100000 connections) the reach of a TTL=7 query is 3665 and 2831. We can compare now the results of our experiment with those of [13]. The study [13] states that if we assume no duplicated queries, TTL=7 and constant node degree N, then for N=4, 4372 packets will be generated and for N=5, 27305 packets will be generated. The number of generated packets coincides with the number of reached nodes. For a given TTL (7 in our case) the saturation can be avoided for any kind of graph if we choose N, its average node degree so that the idealistic graph from [13] with node degree N+1 does not saturate for the chosen TTL. The limit is 8742 nodes. Therefore we have chosen to have 10000 nodes in the further experiments.

### C. Inputs for the simulator

The following set of assumptions applies to all that follows, except the adaptive TTL analysis:

- 10000 nodes with identical capabilities participate in the network,
- 30 separate documents are stored in the network,
- each document has 50 replicates (a given document is present on 0.5% of the nodes),
- a TTL value of 7 is chosen, the default value in most Gnutella implementations [15],
- 300 queries starting from random locations and searching for random, but existing, documents are executed in each setup,
- the goal is to reach a hit rate of 95%.

In case of ConStars an even distribution would mean 2.5 (of 50) replicates per document falling in the core network. On the other hand, the case of "stronger" nodes (with long uptime) sharing more documents is also of interest. We used as a measure the percentage of documents in the core network (pdcn).

The simulated cases were 0:50 (no documents in core network – pdcn=0%), 1:49 (pdcn=2%), 2:48 (close to the uniform distribution – pdcn=4%), 5:45 (pdcn=10%), 10:40 (pdcn=20%) and 20:30 (pdcn=40%). The core network nodes can be thought of as document replicators, for faster access. We did not simulate cases with higher pdcn-s because then the network would have lost the peer-to-peer character and

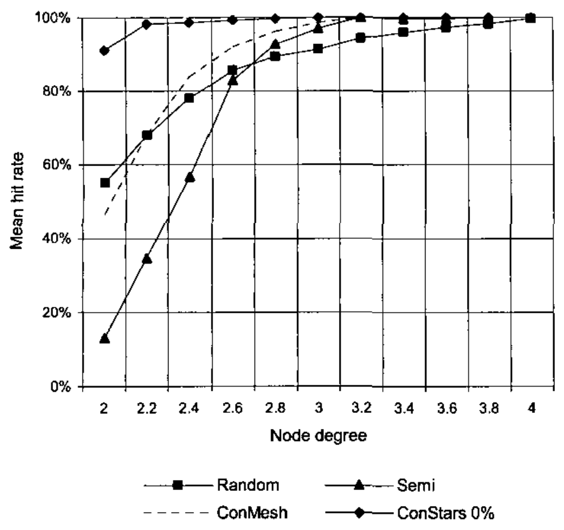

Fig. 1. Hit rate comparison for different network topologies

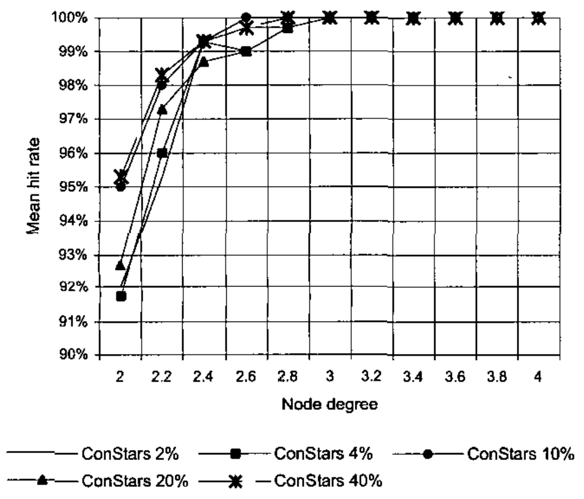

Fig. 2. Hit rate comparison of ConStars with different pdcn values

turned more and more into a centralized, client-server architecture.

*D. Hit rate test*

In order to find optimal average node degrees, we have simply covered our interest-domain (node degrees ranging from 2 to 4) with a resolution of 0.2. Fig. 1. and 2. show the hit rate of the chosen networks. For ConStars we tested several pdcn values.

We expect that as the node degree increases, a given number of connections in the network generally produce a higher hit rate for every topology. However, the topology

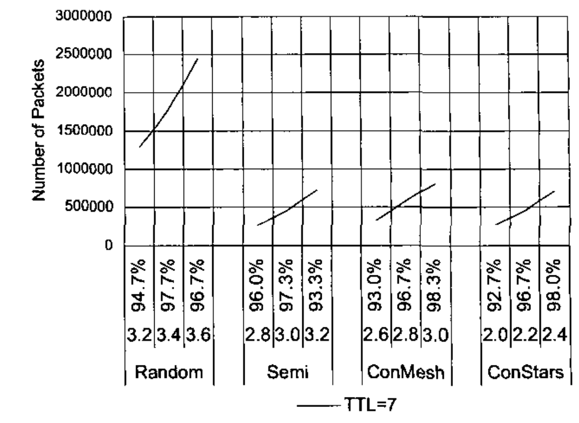

Fig. 3. Traffic of feasible networks

generation hazard can lead to the situation when for a given topology a reduction is produced in fact. In order to gather statistically correct data, in the hit rate test phase we ran 30 queries on 10 outcomes of the topology generation for each topology parameter set, instead of 300 queries over one representative. This way the statistics of both the topology and query generation random event will be averaged out.

At low values of the node degree Semi has the worst behavior, a lot worse than any other topology.

The difference between Random and ConMesh is not as spectacular, but the former always performs slightly worse. ConStars and ConMesh with pdcn=0 have comparable behavior.

For the case of ConStars Fig 2. indicates that above 2.2 it reaches the target hit rate for every value of the pdcn. If we choose higher values for the target, slight differences can be observed as a function of the pdcn. However these differences are not systematic within the tested region of pdcn.

The hit rate test results can be used to select the node degree region for each topology for deeper analysis. In the analysis of the following sections we generate networks with topologies using three node degrees: the smallest degree that satisfies the 95% goal hit rate and the neighbors, with offsets of ±0.2. The hit rates from the next figures differ from those in Fig. 1. and 2. because in this case we deal with individual outcomes of the topology generation random event.

*E. Traffic test*

Fig. 3. presents the generated traffic measured in number of packets for the four topologies. From the traffic viewpoint the Random topology is the worst-case alternative. ConMesh and ConStars have approximately the same behavior. Semi produces approximately the same traffic as ConMesh, and much less than Random.

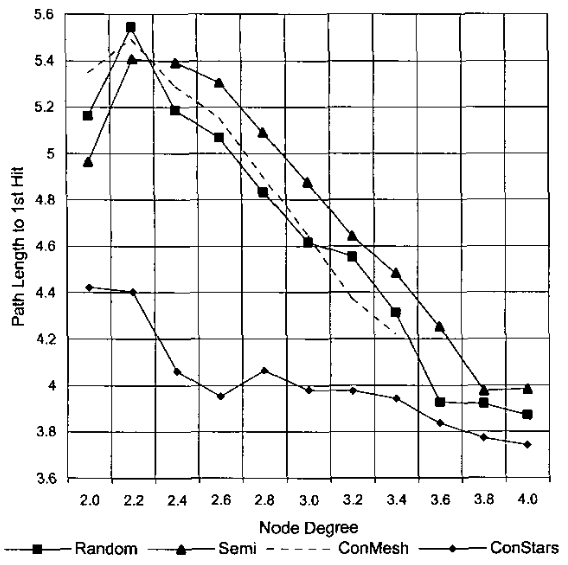

Fig. 4. Average path length to first hit

## F. *Average path length test*

The purpose of this test has been to give an insight in the distribution of path lengths to first hit. If the average path length to first hit is much below the selected TTL value, then one has to consider choosing a lower value for the TTL, otherwise the query will generate large unnecessary traffic, resulting in network load. Fig. 4. confirms the necessity of a careful selection of the TTL. The diagram shows that most queries do not need TTL=7, so we have run simulations with all TTL-s from 1 to 7 as shown in Table I.

## G. *Effect of adaptive TTL*

Fig. 4. clearly indicated that the typical TTL (7) was not needed by most queries. Fig. 5. shows the number of generated packets in each TTL step for 4 different topologies.

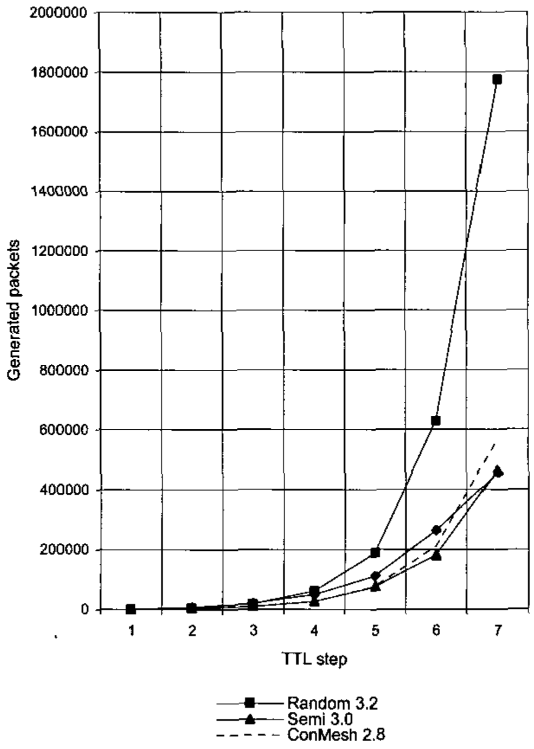

Fig. 5. Generated packets in individual TTL steps, non-adaptive case

Considering that more than 80% of total packets are generated in the last two TTL steps, and at least 50% of total traffic is generated in the last (7th) step, as indicated in Fig. 5., the traffic could be reduced using adaptive TTL schemes. One can expect large reduction of payload traffic especially from TTL steps 5, 6 and 7.

TABLE I
TTL PATTERNS FOR OPTIMAL TRAFFIC

| Type | Avg**. Degree | HitRate | Packets from TTL1...7* | | | | | | | Total packets | Compared to TTL=7 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Random | 3.2 | 94.7% | 0 | 0 | 0 | 45713 | 68880 | 113285 | 123557 | 351435 | 28% |
| Random | 3.4 | 97.7% | 0 | 4556 | 0 | 56068 | 78636 | 90175 | 71017 | 300453 | 17% |
| Random | 3.6 | 96.7% | 0 | 5045 | 0 | 62322 | 94024 | 86435 | 81563 | 329389 | 13% |
| Semi | 2.8 | 96.0% | 862 | 0 | 0 | 0 | 49506 | 52036 | 67349 | 169753 | 62% |
| Semi | 3.0 | 97.3% | 0 | 0 | 9752 | 0 | 64373 | 58081 | 71217 | 203422 | 44% |
| Semi | 3.2 | 93.3% | 0 | 0 | 0 | 34364 | 61282 | 55791 | 47675 | 199111 | 28% |
| ConMesh | 2.6 | 93.0% | 803 | 0 | 0 | 0 | 52389 | 73262 | 70169 | 196623 | 59% |
| ConMesh | 2.8 | 96.7% | 0 | 0 | 10449 | 0 | 64989 | 79946 | 66155 | 221538 | 39% |
| ConMesh | 3.0 | 98.3% | 0 | 0 | 12221 | 0 | 80161 | 94733 | 69511 | 256626 | 32% |
| ConStars | 2.0 | 92.7% | 0 | 0 | 19883 | 0 | 64852 | 0 | 89439 | 174175 | 63% |
| ConStars | 2.2 | 96.7% | 0 | 0 | 20346 | 0 | 86264 | 0 | 100156 | 206766 | 45% |
| ConStars | 2.4 | 98.0% | 0 | 0 | 21810 | 0 | 111492 | 57900 | 44213 | 235415 | 34% |

* 0 means that the given TTL step should be left out for optimal traffic

** In case of ConStars the Average Node Degree applies to Core Nodes only

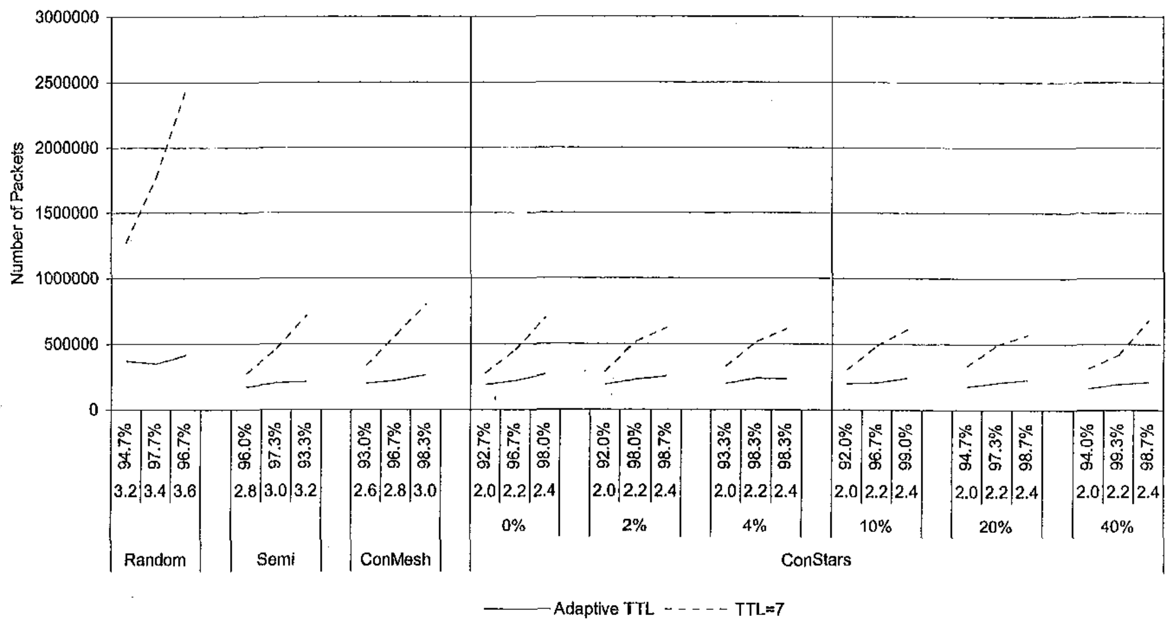

Fig. 6. The effect of adaptive TTL

Table I compares the generated traffic for the different topologies. The second column indicates the average node degree that has been considered. The third column indicates the hit rate obtained by using the given topology. The combined column indicates how many packets have been generated by a query at each hop. This number of packets consists of query packets and query hit packets. The parameter in this case is the TTL value set in the query message.

The packet generation algorithm with respect to the setting of the TTL field has been the following:

- set the smallest initial value for the TTL except the values that have been omitted (marked with '0')
- if no hit has been obtained and current TTL value is smaller than 7, increase the TTL value with one,
- if no hit has been obtained and current TTL value is 7, stop,
- if hit has been obtained, stop.

The algorithm to selecting the omitted TTL values has been the following: omit those TTL values that generate small number of hits compared to the rest of TTL values. More precisely, it has been tested that the values TTL=5, 6 cannot be omitted, otherwise efficiency will be only slightly better than the static case with TTL=7. However, the added efficiency by the values TTL=1, 2, 3, 4 is less obvious, therefore a full evaluation has been performed through consecutively eliminating these values (16 combinations). Table I contains for each pair of topology/node degree the optimal combination, i.e. the combination that has the greatest efficiency compared to the static case (TTL=7). For example, the Random topology with node degree 3.4 behaves the most efficiently if TTL steps 1 and 3 are omitted.

As Table I shows some topologies respond better (Random for example), or worse (ConStars) to the use of adaptive TTL, the 'goodness' being measured through the efficiency compared to the static case (TTL=7), the last column of Table I.

For improved understanding Fig. 6. shows a comparison of traffic with and without adaptive TTL for the different topologies. The figure also indicates that in the case of ConStars the required traffic slightly decreases when pdcn is increased in the core network, for all tested node degrees.

## VI. Discussion

In this section we briefly discuss and compare the tested topologies with respect to the aspects highlighted in the previous section.

The main results for each topology are collected in Table II. In addition to summarizing the results of section V, the table contains an estimate of the bandwidth needed per node for query traffic. [17] indicated that a network of 8000 nodes and 1/min frequency of searching would require an average bandwidth of 127 kBit/s for every peer participating in the Gnutella network. The values of 2.1 up to 4.0 kbps needed for various topologies using adaptive TTL, as shown in table II, are significantly smaller so scalability is not as critical a problem as reported in [17].

### *A. Random*

Random has a fragmented topology with isolated nodes and isolated network regions. The necessary node degree is the highest of all considered topologies.

In terms of generated traffic the Random topology has the worst behavior.

Adaptive TTL has a major effect. A reduction with a factor

TABLE II

COMPARISON OF TOPOLOGIES

| Topology | Node degree for 95% hit rate | Packets without adaptive TTL | Traffic using adaptive TTL as percentage of original traffic | Generated traffic/node for 1 query/min/node, without/with adaptive TTL | Main-tenance | Usage |
|---|---|---|---|---|---|---|
| Random | 3.4 | 1767370 | 17% (13%-28%) | 23.5 kbps/4.0 kbps | No | No use |
| Semi | 2.8 | 273795 | 62% (28%-62%) | 3.65 kbps/2.26 kbps | Simple | Good for homo-genous devices |
| ConMesh | 2.8 | 568046 | 39% (32%-59%) | 7.6 kbps/2.9 kbps | Compli-cated | No use |
| ConStars | 2.2 | 346666 | 45% (34%-63%) | 4.6 kbps/2.1 kbps | Mode-rate | Useful for heterogeneous networks (phones + PCs) |

of more then 5 (17%) can be obtained in terms of generated traffic, which is the highest reduction among all topologies. For target values slightly different from ours the reduction remains in the same order of magnitude.

Random is simple and does not need maintenance.

### A. *Semi*

Semi doesn't have isolated nodes, only isolated network regions. However, even with small node degrees the probability of isolated network regions is small. Its hit rate analysis resulted in a value of 2.8 as necessary average node degree, a low value in comparison to Random. In terms of generated traffic Semi has the best behavior.

Adaptive TTL techniques can lead to a reduction of more than 2 times of the generated traffic. The maintenance of a Semi is simple. It is a promising topology.

### B. *ConMesh*

ConMesh is a topology made of a single network. There are no isolated nodes or network regions.

In our experiment it needs a moderate node degree (2.8) to fulfill the target (statistically the necessary value is 3.0) and the generated traffic is average. Adaptive TTL techniques can lead to a reduction of 1.8 up to 3 times of the generated traffic.

If graph connectedness for the case of single node failure is to be achieved for any node degree, assorted data about the second neighbor should be stored in each node, grouped by the direct neighbors. The replacement of a failing node's connections generates considerable traffic. As a result ConMesh is the most difficult topology to maintain. Basically it differs very little from Semi and the maintenance costs resulting from the requirement of avoiding isolated islands are too high to make ConMesh a useful topology.

### D. *ConStars*

ConStars as well has no isolated nodes. The necessary node degree is 2.2, the lowest of each topology. Generated traffic is low, adaptive TTL leads to a reduction of up to 2 times of the generated traffic, which is the lowest.

A remarkable aspect of this topology is that the load is unequally divided: the star nodes are heavily loaded while the leaf nodes carry much less traffic. This can be useful in cases when the network consists of heterogeneous devices, such as mobile phones as leaf nodes and personal computers as star nodes.

Maintenance of a ConStars network is interpretable in the core network only. The steps taken for maintenance depend on the core network topology. As shown in Fig. 5., the distribution of document replicates between core and leaf nodes barely affects traffic within the tested region of pdcn (0%-40%).

## VII. CONCLUSIONS

In this paper, we have focused on the query traffic generated by peer-to-peer networks having static topologies. The nodes were identical, well-behaved peers equipped with a file sharing application. Four different topology types and different values of the average node degree have been tested.

We have highlighted some aspects of the networks for comparison. Qualitative attributes (complexity of maintenance) were also briefly dealt with but we have mainly focused on the quantitative properties.

The necessary average node degree has been determined for each topology, and we have tried to find a way to reduce overall traffic. An adaptive TTL scheme is constructed that results in significant reduction of traffic applied to any of the inspected topologies.

Our analysis definitely confirmed that regardless of the topology adaptive TTL has a significant effect on overall traffic. We have found that TTL steps 5-6-7 have the largest effect on average traffic reduction. The gain depends on the deviation of the node degree rather than on the topology itself. Accordingly Random profits the most from applying adaptive TTL meanwhile Semi and ConMesh produce comparable results.

The topology of the Gnutella network has fundamental effect to the protocol performance. Our analysis shows that only two of the four topologies are useful: Semi and ConStars. Semi is the best topology for a network with similar devices, e.g. all of the nodes are mobile phones. ConStars would be useful when the network consists of devices with different capabilities, e.g. mobile phones and PCs. If the node degrees are chosen appropriately for each topology and if the protocol implementation is able to maintain the topologies close to the optimum the bandwidth needs can be reasonably small.

A limitation of this study, as always with simulation analysis, is that the results are subject to parameter selection and random variation. We tried to choose the network sizes and other parameters in a way that resembles a real-world application. Performing more simulations random variations could be reduced but the added accuracy is not likely to change the key conclusions.

In summary this study shows that sharing content between mobile phones using Gnutella or similar protocols may be within reach. Further work will be needed in searching for efficient topology maintenance algorithms, for the analysis of the dynamic network case, adaptive document replication, indexing and adaptive caching, document popularity, and technology-specific aspects regarding the mobile environment and wireless communication systems mainly from the viewpoints of scarce bandwidth, addressing and peer discovery.

## ACKNOWLEDGMENT

We would like to express our thanks to Szabolcs Fodor, Nokia Research Center, Dr. Charaf Hassan and Kálmán Marossy, Budapest University of Technology and Economics, for fruitful discussions that enhanced the quality of this paper. Their help is gratefully acknowledged.

## REFERENCES

[1] The Napster protocol, http://opennap.sourceforge.net/napster.txt

[2] The Gnutella protocol specification v0.4, document revision 1.2, http://www9.limewire.com/ developer/gnutella_protocol_0.4.pdf

[3] http://www.fasttrack.nu

[4] Peer-to-peer file sharing – The impact of file sharing on service provider networks, white paper, December 2000, http://www.sandvine.com

[5] M. A. Jovanovic, F. S. Annexstein, and K. A. Berman, Modeling peer-to-peer network topologies through "small-world" models and power laws, 9th Telecommunications Forum TELFOR2001, Belgrade, November 2001

[6] M. Ripeanu, I. Foster, and A. Iamnitchi, Mapping the Gnutella network: properties of large-scale peer-to-peer systems and implications for system design, 1st International Workshop on Peer-to-Peer Systems (IPTPS'01), Cambridge, USA, March 2001

[7] F. S. Annexstein, K. A. Berman, and M. A. Jovanovic, Latency effects on reachability in large-scale peer-to-peer networks, SPAA'2001, Heraklion, Greece, July 2001

[8] M. J. Freedman, R. Morris, Tarzan: a peer-to-peer anonymizing network layer, CCS'02, Washington DC, USA, November 2002

[9] V. R. Scarlata, B. N. Levine, and C. Shields, Responder anonymity and anonymous peer-to-peer file sharing, ICNP-01, pp. 272-280, Riverside, California, USA, November 2001

[10] J. Chu, K. Labonte, and B. N. Levine, Availability and locality measurements of peer-to-peer file systems, July 2002, vol. 4868 of Proceedings of SPIE

[11] S. Saroiu, P. K. Gummadi, and S. D. Gribble, A measurement study of peer-to-peer file sharing systems, technical report # UW-CSE-01-06-02, Department of Computer Science & Engineering, University of Washington, Seattle, USA

[12] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, Search and replication in unstructured peer-to-peer networks, preprint 2001, unpublished, http://citeseer.nj.nec.com/lv02search.html

[13] J. Ritter, Why Gnutella can't scale. No, really, Darkridge, http://www.darkridge.com/%7Ejpr5/doc/gnutella.html

[14] A. Oram (editor), Peer-to-Peer – Harnessing the power of disruptive technologies, O'Reilly, 2001 pp. 203-241, pp. 354-380

[15] M. Ripeanu, Peer-to-peer architecture case study: Gnutella network, technical report, University of Chicago, http://people.cs.uchicago.edu/~matei/PAPERS/

[16] I. Clarke, Freenet: a distributed anonymous information storage and retrieval system, http://freenetproject.org/cgi-bin/twiki/view/Main/ICSI

[17] M. Portmann et al., The cost of peer discovery and searching in Gnutella peer-to-peer file sharing protocol, IEEE International Conference on Networks (ICON'01), Bangkok, October 2001

[18] Ch. Betstetter, H. Vögel, and J. Eberspächer, GSM phase 2+ general packet radio service GPRS: architecture, protocols and air interface, IEEE Communications Surveys, vol.2. No.3, 1999, pp. 2-14