Improving the Scalability of Logarithmic-Degree DHT-based Peer-to-Peer Networks ⋆

Bruno Carton[1] and Valentin Mesaros[2]

[1] CETIC, rue Clément Ader 8, B-6041 Charleroi, Belgium
bruno.carton@cetic.be
[2] Université catholique de Louvain, place Sainte Barbe 2,
B-1348 Louvain-la-Neuve, Belgium
valentin@info.ucl.ac.be

Abstract. High scalability in Peer-to-Peer (P2P) systems has been achieved with the emergence of the networks based on Distributed Hash Table (DHT). Most of the DHTs can be regarded as exponential networks. Their network size evolves exponentially while the minimal distance between two nodes as well as the routing table size, i.e., the degree, at each node evolve linearly or remain constant. In this paper we present a model to better characterize most of the current logarithmic-degree DHTs. We express them in terms of *absolute* and *relative* exponential structured networks. In relative exponential networks, such as Chord, where all nodes are reachable in at most H hops, the number of paths of length inferior or equal to H between two nodes grows exponentially with the network size. We propose the `Tango` approach to reduce this redundancy and to improve other properties such as reducing the lookup path length. We analyze `Tango` and show that it is more scalable than the current logarithmic-degree DHTs. Given its scalability and structuring flexibility, we chose `Tango` to be the algorithm underlying our P2P middleware.

1 Introduction

Over the past few years, Peer-to-Peer (P2P) networks have become an important research topic due to their interesting potentials such as self-organization, decentralization and scalability. A P2P network is principally characterized by its structuring policy and the lookup protocol employed. Not long after the emergence of the first popular P2P networks, Napster and Gnutella, it was realized that scalability in these networks was an important issue. A better alternative are the P2P networks based on DHT (Distributed Hash Table). These networks are self-organized, fully distributed and highly scalable. Furthermore, given that each node has a well defined routing table, the lookup for any node/item can be accomplished within a relatively small number of hops. As the network size increases *exponentially*, the maximum lookup length as well as the routing table size at each node (i.e., the degree) increase *linearly* like in Chord [1], Pastry [2] and Tapestry [3], or even remain constant like in Koorde [4] and DH [5].

The DHT based P2P networks are also called *structured networks*, since they follow a well defined structure. A closer look to their structure allowed us to notice that

⋆ This work was funded at CETIC (www.cetic.be) by the Walloon Region (DGTRE) and the E.U. (ERDF and ESF), and at UCL by the Information Society Technologies programme of the European Commission, Future and Emerging Technologies under IST-2001-33234 PEPITO.

most of the logarithmic-degree DHTs fall into two main categories, depending on the nodes' view of the network (we defer the definition of node's view to Section 2). We call them *absolute* and *relative structured exponential networks*. A first contribution of this paper is the description of a model to better characterize the exponential structured networks as absolute and relative. Related to this work is the research described in [6] where a model based on the concept of k-ary search is proposed for reasoning about DHT networks. Their model addresses only relative structured exponential networks, while ours is more general, addressing the absolute networks, too.

Our model allowed us to observe that in the relative exponential structured networks the fingers of a node are not totally exploited. Hereinafter we denote the "fingers" of a node n to be the single-hop connections of n, and hence representing the entries in the routing table of node n. In Section 3 we propose an approach, that we called `Tango`, to structure the relative exponential networks for increasing their scalability. `Tango` reduces the redundancy in the multiplicity of paths between two nodes of a relative exponential network and, as such, it reduces the path length between the nodes. The `Tango` approach is the second and the main contribution of this paper. In Section 4 we compare `Tango` with DKS [7], and with the DH constant-degree network.

2 Structured exponential network

A structured exponential network is a network built incrementally using well-defined steps. It is composed of nodes linked together via directed edges according to structuring rules, and characterized by an exponential factor k which is the number of instances of network Net_i used to define the subsequent network Net_{i+1}. The network Net_1 is the initial network composed of one node. At step i, network Net_i is built by using k instances of network Net_{i-1} linked to one another.

We identify two methods for connecting all k instances of Net_{i-1} at the i^{th} step : absolute and the relative connections. They lead to absolute and relative structured exponential network, respectively. We illustrate both methods for a network of size 64, built in four steps, and parameterized by an exponential factor $k = 4$. Each node is identified both numerically by using a unique identifier ranging from 0 to 63, and graphically by using k shapes (i.e., light square, light circle, bold square and bold circle). The shape organizes the nodes within the network whereas the size of the shape determines the network building step. Small shapes stand for instances of Net_1, medium shapes for instances of Net_2, and large shapes for instances of Net_3. The network instance of Net_4 regroups the four network instances of Net_3. However, for simplicity, Net_4 is not marked in the figures. In order to distinguish the fingers of the reference node from the other nodes, we represent them as non-gray numbers whereas the other nodes are in gray. Moreover, we introduce the $\oplus$ and the $\ominus$ operators. In a network of size S, we define the operators as $m \oplus n = (m + n) \bmod S$, and $m \ominus n = (m - n + S) \bmod S$.

An absolute structured exponential network is represented in Figure 1 (left). In such a network, each node has the same view of the network. For instance, all nodes see that nodes ranging from 0 to 15 are sitting in the large light square. That is, if a node sees that a node m is sitting in a given shape then all the nodes see that m is sitting in that given shape. In such a network, at the i^{th} step, the $k - 1$ fingers of a node n

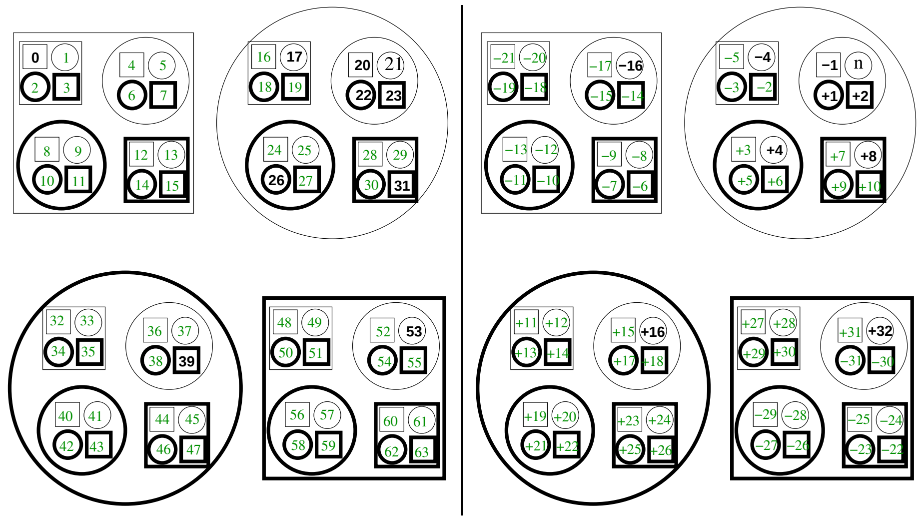

Fig. 1. (left) Absolute structured exponential network of size 64 with $k = 4$. (right) View of a node n in a relative structured exponential network of size 64 with $k = 4$.

are pointing to the $k - 1$ other instances of Net_{i-1}. Moreover, it does not matter to which node inside each Net_{i-1} n points to. For instance, in the network represented in Figure 1 (left), the fingers of node 21 at the third step can be any instance of nodes a,b,c where $a \in [16 \ldots 19]$, $b \in [24 \ldots 27]$, and $c \in [28 \ldots 31]$.

A relative structured exponential network differs from an absolute one by the fact that the view of the network owned by a particular node is relative to its position within the network. For instance, nodes sitting in the large light square are found at distance $dist$ from the reference node, with $-21 \leq dist \leq -6$. Moreover, in a relative exponential network, a node n has to point precisely to the nodes occupying relatively the same positions in the $k - 1$ other instances of Net_{i-1}. For instance, as represented in Figure 1(right), the fingers of node n at the third step are $n \ominus 16$, $n \oplus 16$, $n \oplus 32$.

Most logarithmic-degree DHT-based P2P networks can be expressed either in terms of an absolute or in terms of a relative structured exponential network. For instance, Pastry and Tapestry can be seen as instances of the absolute structured exponential network by instantiating the employed alphabet to the shapes used in Figure 1. On the other hand, Chord and DKS can be seen as instances of the relative structured exponential network.

This model allows us to state that networks built with the relative and the absolute approaches scale at the same rate. Indeed, let S_i be the size of network Net_i and H_i be the maximum number of hops to reach any node in Net_i. Then, for both structures we have $S_i = k * S_{i-1}$ with $S_1 = 1$, $H_i = i - 1$, and a number of $(k-1) * (i-1)$ fingers at each node. Moreover, this model allows us to state that if at the i^{th} step, a node n points to node m, then in an absolute network, the networks reachable in at most i hops by n and m, using all the fingers established in the first i steps, are identical while they differ in a relative network. This difference is at the foundation of the `Tango` definition and its propention to increase finger utilization.

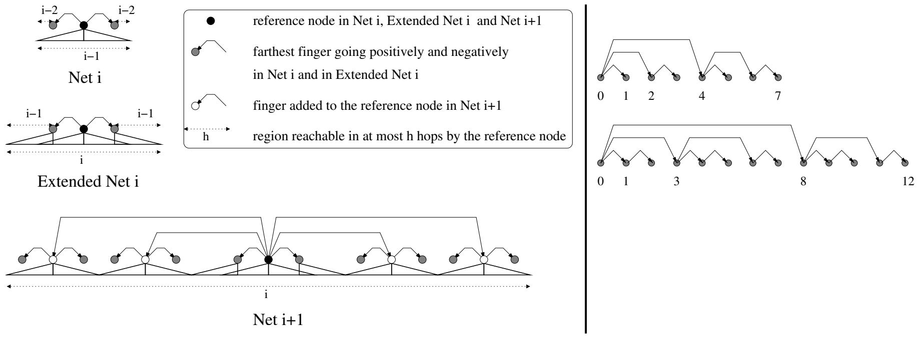

Fig. 2. (left) Network building pattern in `Tango` where $k = 5$. (right) Paths from node 0 to all the other nodes in a Chord network of size 8 and in a `Tango` network of size 13.

3 `Tango`: a novel approach for reducing unexploited redundancy

In a relative exponential network we can identify two types of redundancy. The first one results from the commutative property of the addition operation and from the fact that each node owns, relatively, the same fingers. For example, in Chord, node 0 can reach node 6 via node 4 (6=0+4+2) and also via node 2 (6=0+2+4). The second type of redundancy results from the underutilization of fingers.

To have a clear explanation, we introduce the notion of positive and negative regions of a given node n. A node m is found in the positive region of node n iff $m \ominus n < n \ominus m$, otherwise, node m is found in the negative region of node n.

We propose `Tango`, an approach to address the second type of redundancy, and thus increasing network scalability by taking into account that the networks reachable in at most i hops by n and its fingers added at step i, using all the fingers established in the first i steps, are different in a relative network. Indeed, the region covered in at most i hops via the farthest finger added in the positive (resp. negative) region at step i and the region covered in at most i hops via the closest finger added in the positive (resp. negative) region at step $i + 1$ overlap partially. For example, the regions reachable in at most 3 hops by node 21 via node 29 (i.e., from 24 to 39) and via node 37 (i.e., from 32 to 47) overlap. Let a valid path between two nodes in a network instance Net_i be any path between these nodes whose length is at most $i - 1$ hops. In a relative network, all these overlap regions increase exponentially the number of valid paths between two nodes. Moreover, the cumulated size of the overlap, i.e., the amount of unexploited redundancy in an instance of Net_i grows exponentially with i.

3.1 `Tango` definition

In order to prevent overlapping, the region comprised between the farthest finger added in the positive (resp. negative) region at step i and the closest finger added in the positive (resp. negative) region at step $i + 1$ has to be equal to the size of the network instance Net_i. This improvement is graphically expressed in Figure 2 (left) for a network characterized by $k = 5$. One can notice that Net_{i+1} is composed of 5 blocks. There are 4 instances of Net_i and 1 instance of $Extended\ Net_i$, which is the network reachable by the reference node in at most i hops by using the fingers defined in Net_i.

Let k_i^+ and k_i^- be the number of fingers added in the positive and, respectively, the negative regions of a node at step i. Hence, knowing that at each construction step i there are $k-1$ fingers added to a node, we obtain $k = k_i^+ + k_i^- + 1$.

Let $d_{i,j}^+$ (resp. $d_{i,j}^-$) be the distance at which the j^{th} positive (resp. negative) finger of the i^{th} step should be placed. Let S_i^+ (resp. S_i^-) be the size of the positive (resp. negative) region of a reference node at step i. Equations 1 establish the size growth and the fingers positioning in `Tango`. One can note that for $k_i^- = 0$, the `Tango` network corresponds to an improved version of Chord, and DKS with an arity k. The reader can refer to Section 4 for a comparison between Chord, DKS and `Tango`.

$$\begin{array}{lll} d_{2,j}^{\pm} = j & j \in [1 \ldots k_2^{\pm}] & S_1^{\pm} = 0 \\ d_{i,j}^{\pm} = d_{i,j-1}^{\pm} \pm S_{i-1} & j \in [1 \ldots k_i^{\pm}],\ i > 2 & S_i^{\pm} = S_{i-1}^{\pm} + d_{i,k_i^{\pm}}^{\pm} \quad i > 1 \\ d_{i,0}^{\pm} = d_{i-1,k_{i-1}^{\pm}}^{\pm} & i > 2 & S_i = S_i^+ + S_i^- + 1 \quad i > 0 \end{array} \tag{1}$$

3.2 Key-based routing

The purpose of key-based routing is to route a message tagged with key Key to the node responsible of Key. Let p_n^+ (resp. p_n^-) be the first node encountered in the positive (resp. negative) region of n. The responsibility of a node n is defined in Equation 2.

Beside the node responsibility, there is the finger responsibility defining the node to which a message should be forwarded to. In `Tango` we split the finger responsibility of a given finger F in negative and positive sides[1]. Than, let the focused network be an instance Net_l and let $Sp_{i,j}^{\pm}$ (resp. $Sn_{i,j}^{\pm}$) be the sizes of the positive (resp. negative) finger responsibility as defined in Equations 3 and 4. The finger responsibility $R_{i,j}^{\pm}$ of finger located at position $P_{i,j}^{\pm}$ related to the distance $d_{i,j}^{\pm}$ are defined in Equation 5. Hence, by using its finger $F_{i,j}$, a node can cover the region $R_{i,j}$ in at most $i-1$ hops.

$$R_n = \left[n \ominus \left(\left\lfloor \frac{k_2^+}{k-1} \right\rfloor * (n \ominus p_n^- \ominus 1) \right) \ldots \left(\left\lceil \frac{k_2^-}{k-1} \right\rceil * (p_n^+ \ominus n \ominus 1) \right) \oplus n \right] \tag{2}$$

$$Sp_{i,j}^- = S_i^- \,;\, Sp_{l,k_l^+}^+ = S_i^+ \,;\, Sp_{i,k_i^+}^+ = S_{i+1}^+ \,;\, Sp_{i,j}^+ = S_i^+ \tag{3}$$

$$Sn_{i,j}^+ = S_i^+ \,;\, Sn_{l,k_l^-}^- = S_i^- \,;\, Sn_{i,k_i^-}^- = S_{i+1}^- \,;\, Sn_{i,j}^- = S_i^- \tag{4}$$

$$R_{i,j}^{\pm} = \left[P_{i,j}^{\pm} \ominus Sn_{i,j}^{\pm} \ldots P_{i,j}^{\pm} \oplus Sp_{i,j}^{\pm} \right] \tag{5}$$

[1] The denomination of `Tango` comes from its ability to have positive routing steps followed by negative routing steps and vice versa.

3.3 `Tango` in a sparse and dynamic network

In a sparse network, the position of a finger F (i.e., P) of a node n may correspond to a missing node. In that case, n points to the node responsible of P. Hence, the nodes are playing the finger role of the missing nodes laying within their responsibility. In order to preserve the lookup efficiency, each node adapts its routing table in order to reach the same part of the network in the same number of hops as it would have been done by each missing nodes within its responsibility. That is why in `Tango`, we define the finger position $P(n)$ and the finger node $F(n)$ of a node n as in Equation 6, where $j \in [1 \ldots k_i^-]$, $g \in [1 \ldots k_i^+]$ and $i \in [1..l]$.

$$\begin{array}{ll} P_{i,j}^-(n) = R_n.inf \ominus d_{i,j}^- & F_{i,j}^-(n) = m \quad s.t. \quad P_{i,j}^-(n) \in R_m \\ P_{i,g}^+(n) = R_n.sup \oplus d_{i,g}^+ & F_{i,g}^+(n) = m \quad s.t. \quad P_{i,g}^+(n) \in R_m \end{array} \tag{6}$$

To deal with the dynamics in a `Tango` network, the algorithms of join, fault tolerance and correction on use defined in DKS can be applied directly to `Tango`. Moreover, due to the symmetry provided by the `Tango` networks featured with $k^+ = k^-$, the correction on use can be made more efficient. For more details, the reader can refer to [8].

4 Analysis

In this section we shortly compare `Tango` with DKS, and with the Distance Halving constant-degree network. For more details, the reader can refer to [8].

4.1 `Tango` vs. DKS

DKS generalizes Chord to allow a tradeoff between the maximum lookup length in the network (i.e., the diameter) and the size of the routing table at each node (i.e., the degree). The structure of DKS characterized by $k = 2$ is the same as the one of Chord. `Tango` also supports the tradeoff between the diameter and the degree. Moreover, the network covered with `Tango` is much larger than the network covered with DKS and Chord, while keeping the same network diameter and the same degree at a node. That is, in `Tango` the exponential factor is bigger than in Chord and DKS. From Equation 1, one can deduce the size of the network covered with `Tango` at a step $i > 2$ together with the roots (i.e., z_1 and z_2) of its characteristic equation. For a step i sufficiently high, the exponential factor in `Tango` can be approximated to z_1, where $k < z_1 < k + 1$, and thus we obtain $S_i \approx {z_1}^{i-1}$. Although, the search cost in `Tango` is O($logN$), for apprx. the same network size the highest search cost in `Tango` is 75% of the one in Chord.

$$S_i = (k+1) * S_{i-1} - S_{i-2} \qquad z_1 = \frac{k+1+\sqrt{(k+1)^2-4}}{2}, \; z_2 = \frac{k+1-\sqrt{(k+1)^2-4}}{2} \tag{7}$$

$$S_i = k^{i-1} + \sum_{j=1}^{i-2} k^{i-j-2} * (d_{j,k_j^+}^+ + d_{j,k_j^-}^-) \qquad\qquad i > 2 \tag{8}$$

In order to compare the size growth in `Tango` and DKS, one can define the network size covered by `Tango` at the i^{th} construction step as in Equation 8. Note that the first

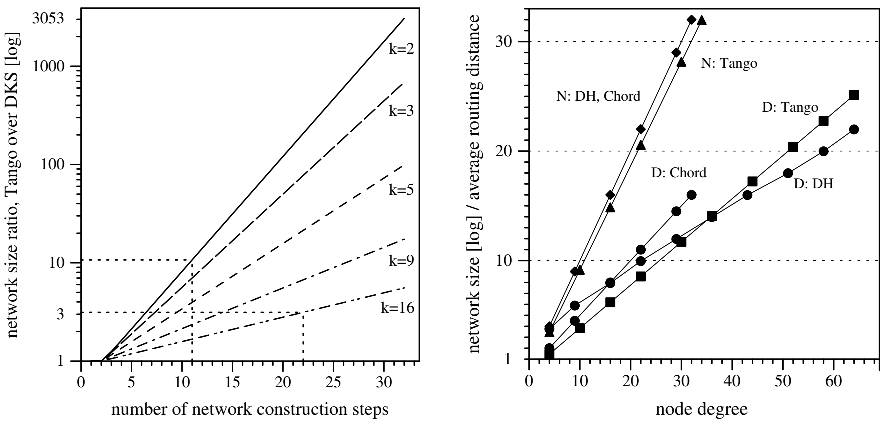

Fig. 3. (left) Ratio between the network sizes covered by `Tango` and DKS, with the same number of fingers at different construction steps. (right) Network size [`N`] and average routing distance [`D`] of `Tango`, DH and Chord, with respect to different values of node degree.

term of the equation corresponds to the network size covered by DKS at the i^{th} step, i.e., k^{i-1}. The second term, which also increases exponentially, corresponds to the difference between the two network sizes; it actually represents the cumulated unexploited redundancy in DKS. In Figure 2 (right) we present an example of how `Tango` covers a larger network than Chord (DKS, $k = 2$) even at the very early building steps. With a routing table of size 3, a node in Chord can cover a network of size 8 in 3 hops, whereas in `Tango`, in 3 hops, a node can cover a larger network, i.e., of size 13.

To better understand the relation between `Tango` and DKS, in Figure 3 (left) we plotted the ratio between the network sizes covered in `Tango` and DKS at each construction step ranging from 1 to 32, for five different values of k. One can note that for a given k, the ratio between the network sizes is growing exponentially at each step. It is also interesting to note that the growth ratio of the ratio decreases as k increases. However, since increasing k leads to increasing the resource consuming and the maintenance cost, it is likely that relative small values of k will be employed.

4.2 `Tango` vs. constant-degree networks

A constant-degree network is a network whose size can increase exponentially, while the node degree remains fixed and the diameter increases logarithmically. Some examples are those based on the de Bruijn graph, such as Koorde and DH. In our analysis we were interested in the average routing distance and the network size for `Tango` ($k = 3$) and DH with respect to different node degrees. We also plot them for Chord to have a third party reference. To compute the average routing distance for DH we used the μ_d formula for de Bruijn graphs given in [9] and doubled it to achieve load balancing, as suggested in [5]. As shown in Figure 3 (right), for the same node degrees (inferior to 34), and almost the same network size, `Tango` provides lower values for the average routing distance than DH.

5 Conclusion

First, in this paper we presented a model to better characterize the structure of the current logarithmic-degree P2P exponential structured networks, such as Tapestry, Pastry, Chord and DKS, in terms of absolute and relative exponential structured networks.

On the other hand, we proposed the `Tango` approach to better structure the relative exponential networks to increase their scalability by exploiting the redundancy in the lookup paths. We showed that `Tango` is more scalable than the current logarithmic-based DHTs. We analyzed the structure of `Tango` with respect to the one of DKS and, implicitly, to the one of Chord. Particularly, we observed that, for small values of the exponential factor k, `Tango` is much more scalable than DKS (and Chord), while for big values of k the scalability of the two networks is more comparable. However, since increasing k leads to increasing the resource consuming and the maintenance cost, it is likely that relative small values of k will be employed. We also analyzed `Tango` with respect to DH, a constant-degree network. We observed that, for networks with relative large node degrees, the average routing distance in `Tango` and DH are comparable.

Given its structuring flexibility and its scalability potential, we chose `Tango` to be the algorithm underlying our recently released P2P middleware [10], and demo applications: PostIt and Matisse [10]. As future work, we plan to address the redundancy in `Tango` resulting from the commutative property of the finger addition operation.

Acknowledgments

We thank Peter Van Roy and Kevin Glynn from UCL, Belgium, and Luc Onana from KTH, Sweden, for their constructive comments.

References

1. I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications. In *ACM SIGCOMM*, August 2001.
2. A. Rowstron and P. Druschel. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In *ICDSP*, November 2001.
3. B. Zhao, J. Kubiatowicz, and A. Joseph. Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing. Technical Report CSD-011141, U.C. Berkeley, April 2001.
4. F. Kaashoek and D. Karger. Koorde: A Simple Degree-optimal Hash Table. In *IPTPS*, February 2003.
5. M. Naor and U. Wieder. Novel Architectures for P2P Applications: the Continous-Discrete Approach. In *ACM SPAA*, June 2003.
6. S. El-Ansary and L. Onana et al. A Framework for Peer-to-Peer Lookup Services based on k-ary Search. Technical Report TR-2002-06, SICS, May 2002.
7. L. Onana and S. El-Ansary et al. DKS(N, k, f): A Family of Low Communication, Scalable and Fault-Tolerant Infrastructures for P2P Applications. In *CCGRID2003*, May 2003.
8. B. Carton, V. Mesaros, and P. Van Roy. Improving the Scalability of Logarithmic-Degree Peer-to-Peer Networks. Technical Report RR-2004-01, UC-Louvain, January 2004.
9. D. Loguinov and A. Kumar et al. Graph-Theoretic Analysis of Structured Peer-to-Peer Systems: Routing Distances and Fault Resilience. In *SIGCOMM*, August 2003.
10. P2PS v 1.0, Peer-to-Peer System Library, October 2003. Universtité catholique de Louvain, and CETIC, Belgium. www.mozart-oz.org/mogul/info/cetic_ucl/p2ps.html.