Enabling Co-operation between ISPs and P2P systems

Technical report
No. 2, March 2007

Vinay Aggarwal, Anja Feldmann
Deutsche Telekom Laboratories, Berlin
{vinay.aggarwal, anja.feldmann}@telekom.de

Christian Scheideler
TU Munich
scheideler@in.tum.de
1 Abstract

Peer-to-peer (P2P) systems, which are realized as overlays on top of the underlying Internet routing architecture, contribute a significant portion of today's Internet traffic. While the P2P users are a good source of revenue for the Internet Service Providers (ISPs), the immense P2P traffic also poses a significant traffic engineering challenge to the ISPs. This is because P2P systems implement their own routing in the overlay topology, which is largely independent of the Internet routing, and thus impedes the ISP's traffic engineering capabilities. On the other hand, P2P users are primarily interested in finding their desired content quickly, with good performance. But as the P2P system has no access to the underlying network, it either has to measure the path performance itself or build its overlay topology agnostic of the underlay. This situation is disadvantageous for both the ISPs and the P2P users.

To overcome this, we propose a solution where the ISP offers an "oracle" to the P2P users. When the P2P user supplies the oracle with a list of possible P2P neighbors, the oracle ranks them according to their proximity to the user. This can be used by the P2P user to choose neighbors in close proximity or on higher bandwidth links, and therefore improve its performance. The ISP can use this mechanism to better manage the immense P2P traffic, e.g., to keep it inside its network, or to direct it along a desired path. The improved network utilization will also enable the ISP to provide better service to its customers.
# Table of contents

1. Abstract ........................................................................................................... 2

2. Table of contents .............................................................................................. 3

3. Introduction ........................................................................................................ 4
   3.1 An oracle service ............................................................................................ 5
   3.2 Realizing an oracle service .............................................................................. 6

4. Evaluation methodology ..................................................................................... 8
   4.1 Metrics .......................................................................................................... 8
   4.2 Simulation setup ............................................................................................. 10

5. Preliminary results ............................................................................................ 12
   5.1 Gnutella simulations ..................................................................................... 15

6. Summary ............................................................................................................ 17

7. References ......................................................................................................... 18
3 Introduction

P2P systems have recently gained a lot of attention from the Internet users and the research community. Popular applications that use P2P systems include file sharing systems [1] such as BitTorrent, eDonkey, Kazaa, Gnutella as well as VoIP systems such as Skype and GoogleTalk. P2P systems are so popular that they contribute more than 50% to the overall network traffic [2, 3, 4].

However, the wide-spread use of such P2P systems has put the ISPs in a dilemma! On one hand, P2P system applications have resulted in an increase in revenue for ISPs, as they are one of the major reasons cited by Internet users for upgrading their Internet access to broadband [5]. On the other hand, ISPs find that P2P traffic poses a significant traffic engineering challenge [3, 6]. P2P traffic often starves other applications like Web traffic of bandwidth, and swamps the ISP network. This is because most P2P systems rely on application layer routing based on an overlay topology on top of the Internet, which is largely independent of the Internet routing and topology [7].

To construct an overlay topology, unstructured P2P networks usually employ an arbitrary neighbor selection procedure [4]. This can result in a situation where a node in Munich downloads a large content file from a node in Sydney, while the same information may be available at a node in Berlin. It has been shown that P2P traffic often crosses network boundaries multiple times [7, 8]. This is not necessarily optimal as most network bottlenecks in the Internet are assumed to be either in the access network or on the links between ISPs, but not in the backbones of the ISPs [9]. Besides, studies have shown that the desired content is often available “in the proximity” of interested users [10]. This is due to content language and geographical regions of interest. Since a P2P user is primarily interested in finding his desired content quickly with good performance, we believe that increasing the locality of P2P traffic will benefit both ISPs and P2P users.

To better understand the origin of the problem, let us consider how routing works in the Internet and P2P systems. In the Internet, which is a collection of Autonomous Systems (ASes), packets are forwarded along a path on a per-prefix basis. This choice of path via the routing system is limited by the contractual agreements between ASes and the routing policy within the AS (usually shortest path routing based on a fixed per link cost) [11].

P2P systems, on the other hand, setup an overlay topology and implement their own routing [12] in the overlay topology which is no longer done on a per-prefix basis but rather on a query or key basis. In unstructured P2P networks queries are disseminated, e.g., via flooding [13] or random walks while structured P2P networks often use DHT-based routing systems to locate data [4]. Answers can either be sent directly using the underlay routing [4] or through the overlay network by retracing the query path [13]. By routing through the overlay of P2P nodes, P2P systems hope to use paths with better performance than those available via the Internet [12, 14]. But the benefits of redirecting traffic on an alternative path, e.g., one with larger available bandwidth or lower delay,
are not necessarily obvious. While the performance of the P2P system may temporarily improve, the available bandwidth of the newly chosen path will deteriorate due to the traffic added to this path. The ISP then has to redirect some traffic so that other applications using this path receive enough bandwidth. In other words, P2P systems reinvent and reimplement a routing system whose dynamics should be able to interact with the dynamics of the Internet routing [15]. Consider a situation where a P2P system imposes a lot of traffic on an ISP network. This may cause the ISP to change some routing metrics and therefore some paths (at the routing layer) in order to improve its network utilization. This can however cause a change of routes (at the application layer) by the P2P system, which may again trigger a response by the ISP, and so on.

In summary we identify the following drawbacks:

- The ISP has limited ability to manage its traffic and therefore incurs potentially increased costs for its interdomain traffic, as well as for its inability to do traffic engineering on its internal network.
- The P2P system has limited ability to pick an optimal overlay topology and therefore provide optimal performance to its users, as it has no prior knowledge of the underlying Internet topology. It therefore has to either disregard or reverse engineer it.
- Different P2P systems have to measure the path performance independently.

While we do not know of a P2P network that tries to reverse-engineer the Internet topology, there are some proposals that suggest that P2P networks should bias their overlay topology by choosing neighbors that are close in the sense of high-throughput or low latency, e.g., [16, 17] or that are within the same AS, e.g., [18]. Therefore we propose a general solution that allows ISPs and P2P users to cooperate in such a way that both benefit.

### 3.1 An oracle service

Let’s consider how unstructured P2P networks tend to maintain their topologies. New P2P nodes usually retrieve a list of members of the P2P network either via a well known Web page, a configuration file, or some history mechanism [1, 4]. They then pick some subset of these as possible neighbors either randomly [13] or based on some degree of performance measurement [16]. If the chosen neighbor cannot serve the new node it might redirect the new node by supplying an alternative list of P2P members.

Instead of the P2P node choosing neighbors independently, the ISP can offer a service, which we call the oracle, that ranks the potential neighbors according to the distance to the P2P node. This ranking can be seen as the ISP expressing preference for certain P2P neighbors.
Possible coarse-grained distance metrics are:
- inside/outside of the AS
- number of AS hops according to the BGP path [11]

For P2P nodes within the AS the oracle may further rank the nodes according to:
- geographical information such as: same point of presence (PoP), same city
- performance information such as: expected delay, bandwidth

This ranking can then be used by the P2P node to select a closeby neighbor although there is no obligation. The benefit to P2P nodes of all overlays is multifold:

1. they do not have to measure the path performance themselves;
2. they can take advantage of the knowledge of the ISP;
3. they can expect improved performance in the sense of low latency and high throughput as bottlenecks [9] can be avoided.

That P2P networks benefit by increasing traffic locality has also been shown by Bindal et. al [18] for the case of BitTorrent.

The benefit to the ISPs is that they can influence the neighborhood selection process of the P2P network to, e.g., ensure locality of traffic flows and therefore, again have the ability to manage the flow of their traffic. This will also allow them to provide better service to their customers and ensure fairness for other applications like Web traffic, etc.

The oracle is available to all overlay networks. Furthermore, as an open service, it can be queried by any application and is not limited to file-sharing systems. Hence, querying the oracle does not necessarily imply participation in file sharing systems.

3.2  Realizing an oracle service

It may seem rather challenging to build such an oracle in a scalable manner, but much more complicated services, e.g. DNS, already exist. The oracle service can be realized as a dedicated server on each ISP that can be queried using a UDP-based protocol or run as a Web service. It can rely on a semi-static database with the ISP's prefix and topology information. Furthermore, the oracle server only needs to roughly rank the IP nodes and therefore does not need to reveal more information about its network than can anyhow be inferred by reverse-engineering the ISP network via measurements [19].
While the oracle service is not yet offered by the ISPs, P2P nodes have the chance of using a similar, but slow, service to gain some of the oracle benefits already using the "whois" service [20]. It enables the P2P node to retrieve information about possible P2P neighbors such as the AS and some geographic information. This information can then be used by the P2P node to bias its neighbor selection. But purely using the "whois" service does not enable the ISP to express its preference and therefore does not yet enable cooperation.

Overview of Paper

One may argue that biasing the neighborhood selection process adversely affects the structural properties of the overlay topology. Therefore we in this paper propose metrics for evaluating the impact of using the oracle on the overlay as well as the underlay topology. We show, relying on graph based simulations and measured Internet topologies, that the resulting P2P overlays maintain their graph properties like small diameter, small mean path length, but the densely connected subgraphs are now local to the ISPs. To study the impact on an actual P2P network we run extensive simulations of the Gnutella protocol within the packet level network simulator SSFNet [21] and again find that the resulting topologies maintain their graph properties but that the ISP now has the ability to influence the P2P topology.
4 Evaluation methodology

In this section, we propose metrics for evaluating the effectiveness of the idea of using an oracle and describe how we derive representative topologies for our simulations from the Internet AS topology.

4.1 Metrics

As a basic model for our investigations, we model the AS-graph as a complete bi-directed graph

\[ G = (V, E) \text{ with cost function } c : E \to \mathbb{R}^+ \text{ associated with the edges. Every node represents an AS, and for every pair (u, v), let } c(u, v) \text{ denote the overall cost of routing a message from AS u to AS v (which depends on the routing policies of the ASes such a message may traverse).} \]

Given a set of peers P, let AS : P \to V define how the peers are mapped to the ASes and b : P \to \mathbb{R}^+ denotes the bandwidth of the Internet connections of the peers. The overlay network formed by the peers is given as a directed graph H = (P, F) in which every edge (p, q) \in F has a cost of c(AS(p), AS(q)). The quality of H can be measured using several metrics.

Degree

The degree of a peer is defined as the number of its outgoing connections. Ideally, every peer should have a large number of connections to other peers within its AS so as to favor communication within the AS, while connections to other ASes should be limited to avoid high communication costs and high update costs as peers enter/leave the network.

Hop count diameter

Another parameter that should be small is the hop count diameter of the overlay graph H. The hop count diameter D of H is the maximum over all pairs p, q \in P of the minimum length of a path (concerning the number of edges) from p to q in H. It is well-known that any graph of n nodes and degree d has a hop count diameter of at least \( \log_{d-1} n \), and that dynamic overlay networks such as variants of the de Bruijn graph [22] can get very close to this lower bound, a very nice property. However, even though the hop count diameter may be small, the AS diameter (i.e., the distance between two P2P nodes when taking the underlying AS-graph G with cost function c into account) can be very large.

AS diameter

The AS diameter of H is defined as the maximum over all pairs p, q \in P of the minimum cost of a path from p to q in P, where the cost of a path is defined as the sum of the cost of its edges. Ideally,
we would like both the hop count diameter and the AS diameter to be as small as possible. Research in this direction was pioneered by Plaxton et al. [23], and the (theoretically) best construction today is the LAND overlay network [24].

Surprisingly, the best AS diameter achievable when avoiding many P2P connections to other ASes can be better than the best AS diameter achievable when all P2P connections go to other ASes. Consider the simple scenario in which the cost of a P2P edge within the same AS is 0 and that between two different ASes is 1. Let the maximum degree of a peer be d. In scenario 1, we require all edges of a peer to leave its AS, and in scenario 2, we only allow one edge of a peer to leave its AS. In scenario 1, the best possible AS diameter is \( \log_d n \) (see our comments above). However, in scenario 2 one can achieve an AS diameter of just \( \frac{\log_{d/2} (n/(d - 1))}{d-1} \). For this, organize the peers into cliques of size \( (d-1) \) within the ASes (we assume that the number of peers in each AS is a multiple of \( d-1 \)). We can then view each clique as a node of degree \( d-1 \). It is possible to connect these nodes with a graph of diameter close to \( \frac{\log_{d/2} (n/(d - 1))}{d-1} \), giving the result above.

**Flow conductance**

Having a small hop count diameter and AS diameter is not enough. A tree, for example, can have very low hop count and AS diameter. Yet, it is certainly not a good P2P network, since one single faulty peer is sufficient to cut the network in half. Ideally, we would like to have a network that is well-connected so that it can withstand many faults and can route traffic with low congestion. A standard measure for this has been the expansion of a network. However, it seems that the expansion of a network cannot be approximated well. The best known algorithm can only guarantee an approximation ratio of \( O(\sqrt{\log n}) \) [25]. Therefore, we propose an alternative measure here that we call the flow conductance of a network (which is related to the flow number proposed in [26]).

Consider a directed network \( G = (V, E) \) with edge bandwidths \( b : E \rightarrow \mathbb{R}^+ \). For every node \( v \in V \), let \( b(v) = \sum_{e \in E(v)} b(e) \), where \( E(v) \) is the set of edges leaving \( v \). Furthermore, for any subset \( U \subseteq V \) let \( b(U) = \sum_{v \in U} b(v) \). Consider the concurrent multicommodity flow problem \( M_0 \) with demands \( d_{v,w} = b(v) \cdot b(w)/b(V) \) for every pair \( v, w \) of nodes. That is, we consider the heavy-traffic scenario in which each node aims at injecting a flow into the system that is equal to its bandwidth, and the destinations of the flows are weighted according to their bandwidth. The flow conductance \( \lambda \) measures how well the network can handle this scenario, or more formally, the flow conductance is equal to the inverse of the largest value of \( \lambda \) so that there is a feasible multicommodity flow solution for the demands \( d_{v,w} \) in \( G \). It is easy to show that for any network \( G \), \( 0 \leq \lambda \leq 1 \), and the larger the \( \lambda \), the better is the network. As an example, for uniform link bandwidths the flow conductance of the \( n \times n \)-mesh is \( \Theta(1/n) \) and the flow conductance of the hypercube of size \( n \) is \( \Theta(1/\log n) \).
Interestingly, one can significantly lower the number of inter-AS edges without losing much on the flow conductance. Suppose we have \( m \) peers with bandwidth \( b \) that can have a maximum degree of \( d \). Consider a class of networks \( G(n) \) of degree \( d \) and size \( n \) with monotonically increasing flow conductance \( C(n) \). Connecting the \( m \) peers by \( G(m) \) gives a network with flow conductance \( C(m) \). Suppose now that every peer can establish only one inter-AS edge with bandwidth \( b/2 \), and the remaining bandwidth can be used for intra-AS edges. In this case, let us organize the peers into cliques of size \( d - 1 \) within the ASes (we assume that the number of peers in each AS is a multiple of \( d - 1 \)) and interconnect the cliques so that they form \( G(m/(d - 1)) \). Then it is not difficult to see that the resulting network has a flow conductance of \( 2C(m/(d - 1)) \). Hence, compared to arbitrary networks we lose a factor of at most 2.

**Summary**

We have proposed measures that are useful for P2P systems and our results demonstrate that it is possible to have a highly local topology with an AS diameter and a flow conductance that is comparable to the best non-local topologies. Hence, worst-case communication scenarios can be handled by local topologies (i.e., topologies with many intra-AS connections) essentially as well as by non-local topologies. In addition, we expect local topologies to be far better cost-wise for serving P2P traffic in practice than non-local topologies, which we aim to validate through experiments.

**4.2 Simulation setup**

The basis for our simulations is the current AS topology of the Internet, as it can be derived from the BGP routing information. We use BGP data from more than 1300 BGP observation points including those provided by RIPE NCC, Routeviews, GEANT, and Abilene. This includes data from more than 700 ASes as on November 13, 2005. Our dataset contains routes with 4,730,222 different AS-paths between 3,271,351 different AS-pairs. We derive an AS-level topology from the AS-paths. If two ASes are next to each other on a path, we assume that they have an agreement to exchange data and are therefore neighbors. We are able to identify 58,903 such edges. We identify level-1 providers by starting with a small list of providers that are known to be tier-1. An AS is added to the list of level-1 providers if the resulting AS-subgraph between level-1 providers is complete, that is, we derive the AS-subgraph to be the largest clique of ASes including our seed ASes. This results in the following 10 ASes being referred to as level-1 providers: 174, 209, 701, 1239, 2914, 3356, 3549, 3561, 5511, 7018. While this list may not be complete, all found ASes are well-known tier-1 providers. There are 7,994 ASes that are neighbors of a level-1 provider, which we refer to as level-2. All other 13,174 ASes are grouped together into the class other.

Since we do not know how many P2P nodes are in each AS, and we may want to study smaller subsets to be able to compute the complex graph properties in reasonable time, we randomly sub-sample the AS-topology by keeping all level-1 ASes and their interconnections, and selecting a
fraction of the level-2 and level-3 ASes while keeping their proportion the same as in the original data. Hereby, we first select the level-2 ASes and keep their interconnections. Only then do we select the level-3 ASes from among the ASes that are reachable in our subgraph. Most level-1 ASes traditionally are expected to serve more customers than level-2 and level-3 ASes. At the same time there are more level-3 than level-2 than level-1 ASes. Thus we distribute the P2P clients among the ASes in the following ad-hoc manner: a P2P node has equal probability to pick an AS from each level. This results in a 1/3 : 1/3 : 1/3 split of the nodes among the AS levels. This way a level-1 AS serves many more P2P nodes than a level-3 AS. All the topologies used in our experiments have been derived in this manner by randomly subsampling the AS topology derived from the BGP table dumps.
5 Preliminary results

We rely on the system in [27] as the basis for our experiments as it provides us with support for operations on overlay graphs. In the future we plan to use the system to also study the reliability and robustness of the overlay networks as they evolve over time.

For our evaluation we consider five graphs, each with 300 ASes and 4,372 P2P nodes, which results in an average of 14.6 nodes per AS. Each graph consists of 4 level-1 ASes, 100 level-2 ASes and 196 level-3 ASes. We place 375 nodes within each level-1 AS, 15 nodes within each level-2 AS, and 7 nodes within each level-3 AS.

We establish P2P neighbor relationships by randomly picking one of the P2P nodes and let it establish a neighborship either

unbiased: to a single randomly chosen P2P node or

biased: to one from a list of candidates.

The unbiased case corresponds to a P2P protocol with arbitrary neighbor selection, while the biased case corresponds to a P2P node giving a list of potential neighbors to the oracle, and the oracle helping it pick an optimal neighbor. We simulate the simplest of such oracles where it either chooses a neighbor within the querying node's AS if such a one is available, or a node from the nearest AS (considering AS hop distance). We experiment with different sizes of the oracle's choice list.

We experimented with establishing 1,000 up to 40,000 neighbor relationships in total. Given that for random graphs, the threshold for the number of edges to ensure connectivity is at log n/2 times the number n of nodes, it is not surprising that we need roughly 18,000 edges to ensure that the simulated graph is connected. Increasing the number of edges beyond this number does not change the graph properties noticeably. Accordingly, we concentrate on results for 20,000 peerings.

We run four experiments for each of the five AS graphs where the oracle is used for each neighbor relationship with candidate lists of length 1, 10, 50, 100, 200, 375, resulting in 120 experiments. Note that a list length of 1 corresponds to the unbiased case.
Figure 1: Error plots showing comparison of metrics with increasing size of Oracle list
The results we obtained are as follows. First, we check whether the overlay graphs remain connected using biased neighbor selection. It is possible that due to a heavy bias, the graph disintegrates into disconnected components which are themselves well connected. We experimentally verify that all resulting graphs remain connected, thereby not impacting the reachability of the overlay graph.

The next question is if the mean degree of the P2P nodes changes. We find that the mean degree value of 9.138 of an unbiased graph changes to 8.8 in biased graphs with list size 200, see Figure 1(a). The small change in node degree implies that we do not affect the structural properties of the overlay graph seriously.

One may expect that our biased neighborhood selection increases the diameter and mean path length, as it prefers "closeby" neighbors. Yet, in all experiments the hop count diameter of the overlay graph stays at 7 or 8 hops and the AS diameter of the underlying AS graph stays at 5 hops. Neither does the average path length in the overlay graph increase significantly, see Figure 1(b). Therefore we can conclude that the biased neighborhood selection does not negatively impact the overlay connectivity graph.

On the other hand we find that locality improves significantly as captured by the average AS-distance of P2P neighbors. Figure 1(c) shows how the AS-distance improves with the ability of the P2P node to choose a nearby neighbor. A lower AS-distance should correspond to lower latency. This is also reflected in the number of P2P neighbor connections that stay within each of the ASes, see Figure 1(d). Without consulting the oracle, only 4% of the edges are local to any of the ASes. The use of the oracle increases locality by a factor of 7 from 697 to 5088 (in a total of 20,000 peerings), even with a rather short candidate list of length 10. With a candidate list of length 200, more than half of the edges, 59%, stay within the AS. We find that the effects are even more pronounced for smaller networks. This demonstrates how much the oracle increases the ability of the AS to keep traffic within its AS and with a refined oracle to better manage the P2P traffic. These results also imply the benefit to the user, as traffic within the AS is less likely to encounter network bottlenecks than inter-AS traffic.

The remaining question is if the network maintains its ability to route traffic with low congestion. Since the run time requirements of our algorithm for computing a lower bound for the flow conductance of a graph is \(O(n^4)\), we can currently only estimate the flow conductance for small graphs.

Being able to calculate the conductance of smaller graphs only is not a big problem as in case of Gnutella [13], we can calculate the conductance of the graph of ultrapeers, which is naturally much smaller than the entire Gnutella connectivity graph. We constructed unbiased as well as biased graphs with 10 nodes and 21 edges, respectively 18 nodes and 51 edges. Both graphs are generated on a topology with 6 ASes.
The expected flow conductance of the unbiased graphs is 0.505 for the 10 node graph and 0.533 for the 18 node graph (see Section 2). We experimentally verify that both unbiased graphs support a conductance of at least 0.5. Also, we find that the penalty for the two biased graphs is less than a factor of 2. The 10 node biased graph supports a flow conductance of at least 0.3, and the 18 node graph, of at least 0.25. We furthermore observe that subgraphs of the biased graphs support a higher flow conductance which indicates that the connectivity within the ASes is good. This will likely result in a performance boost if the desired content can be located within the proximity of the interested user. The locality of biased graphs increases to 50% (for 10 nodes), respectively 80% (for 18 nodes) compared to 20% in the unbiased graphs.

5.1 Gnutella simulations

After implementing the Gnutella protocol in SSFNet [21], we modified the neighbor selection procedure to take advantage of the oracle [28]. When a Gnutella node bootstraps, it gets a list of Gnutella node addresses it may connect to. The node then consults the oracle, which picks a node within the querying node’s AS if it exists, or a random node otherwise. The node then establishes a Gnutella peering with this oracle-preferred node. On running experiments on a 1000 node topology distributed among 25 ASes, we observed that the use of the oracle improved the scalability of Gnutella. The negotiation traffic of Ping/Pong [13] messages was reduced by a factor of 2, and the search Query messages were reduced by a factor of 3. This was due to enhanced locality in overlay topology, which enabled messages to traverse lesser overlay hops. We observed that the Gnutella topology is well correlated with the Internet AS topology, where the nodes within an AS form a dense cluster, with only a few connections going to nodes in other ASes, see Figure 2. The overall overlay network is still connected, while the diameter and average path length increase only slightly. In a real Testlab with 6 routers, 6 switches, 21 computers, and 45 running Gnutella servers, we found that all the search queries that were satisfied in the unmodified Gnutella protocol, were also satisfied with biased neighbor selection. Experiments on larger test environments are underway. We are conducting broader experiments in SSFNet to compare response times, latency and message distributions, and also implementing the oracle scheme on Planetlab nodes using Gnutella and Bittorrent.
Figure 2: Overlay topology of Gnutella network
6 Summary

P2P systems build their overlay topology largely agnostic of the Internet underlay. To overcome this, we propose to use an oracle hosted by the ISPs, so that ISPs and P2P users can cooperate for improved performance. This oracle can be queried by P2P nodes while choosing neighbors and it will rank the possible neighbors of the querying node according to a locality indication. We propose metrics to evaluate the effectiveness of using an oracle and show that using the oracle allows the overlay topologies to maintain the graph properties like small diameter, small mean path lengths and node degree, while at the same time, tremendously increasing their network locality (lesser mean AS distance, larger number of intra-AS peerings). Even the ability of the network to route arbitrary traffic patterns with low congestion, while reduced, is still reasonable. Initial simulations on Gnutella also report similar results along with improved scalability. In a next step we plan to design simple, provably good, and experimentally well-behaved distributed algorithms for P2P neighborhood selection that take full advantage of such an oracle.
7 References


