A Visualization-Driven Approach to Overlay-Underlay Engineering* **

Vinay Aggarwal[1], Anja Feldmann[1], Marco Gaertler[2], Robert Görke[2], and Dorothea Wagner[2]

[1] Deutsche Telekom Laboratories / TU Berlin, Germany
{vinay.aggarwal, anja.feldmann}@telekom.de
[2] Universität Karlsruhe (TH), Germany
{rgoerke,gaertler,wagner}@ira.uka.de

Abstract. Overlay applications are popular as they provide high-level functionality by masking the intrinsic complexity of the underlay network. However, overlays rely on the underlay to provide them with basic connectivity. Therefore, the intrinsic features of the underlay network determine the efficiency of the overlay. Accordingly, studying the interdependency of the overlay and underlay networks leads to a better understanding of overlay application behaviour. We present a visualization-driven analysis technique for evaluating the overlay architecture with respect to the underlay, inspired by the goal of overlay engineering. Using Gnutella as a case study, our analysis confirms that Gnutella topology differs from a randomly generated network and that there is an implicit correlation between the overlay and underlay topologies.

1 Introduction

In recent times, the design of many real-world applications has changed from a monolithic structure to modular, yet highly customizable services. As an implementation from scratch is usually too time-consuming and expensive, these services are superimposed as an overlay on an already existing underlay infrastructure.

A well-known example arises in logistics. The highways and streets we use everyday constitute a huge transport network. However, traffic in this network is far from structured. In fact, countless companies and institutions rely on this network to accomplish their regular shipping of commodities and services, and by doing so, they cause the traffic on the road network to develop in certain patterns. In technical terms the road network constitutes an *underlay network* while the commodity exchange network of a set of companies implicitly building upon this network forms an *overlay network*. The overlay network uses the underlay to actually realize its tasks.

Another underlay network of prime interest is the Internet, which serves as the workhorse of countless data transfers, multimedia services and filesharing protocols. Almost anytime we use the Internet, we participate in some overlay network that uses the physical Internet (comprised of routers, links, cables, wires) to actually convey the data packets. Interestingly enough, the Internet itself is an overlay built over the telephone network underlay. Within the Internet, a particular breed of overlays that has received a lot of attention lately are peer-to-peer (P2P) applications [25], which range from file-sharing systems like Gnutella and Bittorrent, to real-time multimedia streaming like IPTV, to VoIP phone systems like Skype and GoogleTalk. To get a

* The authors gratefully acknowledge financial support from the European Commission within FET Open Projects DELIS (contract no. 001907) and the DFG under grant WA 654/13-3.

** An electronic version including colored images is obtainable from http://i11www.iti.uni-karlsruhe.de/algo/people/rgoerke/publications/pdf/afggw-vdaoue-08.pdf.

better perspective on the importance of overlays in our life, consider that the mentioned Internet applications currently comprise almost 75% of Internet traffic [15] and will continue to dominate the Internet in the next generation, collectively referred to as Web2.0. Such is the effect of overlays in today's Internet, that many studies [27] have contemplated that the underlay (physical Internet infrastructure) will just become a bit-pipe and almost all services employed by us will be realised through overlays, which will interact with the underlay as well as different overlays running over the same underlay.

Clearly, there is a crucial interdependence between overlay and underlay networks. In particular, the emergence of overly networks heavily affects and poses new requirements on the underlay. The major advantage of overlays is that they provide high-level functionality while masking the intrinsic complexity of the implementation realized in the underlay structure. However, this abstraction entails a certain trade-off, namely independence versus performance. To gain a deeper understanding of the interdependency between the overlay and the underlay, this trade-off needs to be included in the corresponding analysis.

Due to the explosive growth of P2P file sharing applications with respect to total Internet traffic [25], there has been an unprecedented interest in their analysis [2, 3, 23]. There have also been attempts to investigate the overlay-underlay correlations in P2P systems. Using game theoretic models, [18] studies the interaction between overlay routing and traffic engineering within an Autonomous System (AS). An analysis of routing around link failures [23] finds that tuning underlay routing parameters improves overlay performance. Most investigations tend to point out that the overlay topology does not appear to be correlated with the underlay (e.g., [2]), but the routing dynamics of the underlay do affect the overlay in ways not yet well understood. To address the apparent lack of overlay-underlay correlation, some schemes, e.g. [19, 20], have been proposed. More recently, [3] has made a case for collaboration between ISPs and P2P systems as a win-win solution for both.

In this paper, we approach the problem of modelling overlay-underlay correlations using a visualization-driven approach [7], to analyze the overlay in the context of the underlay network. We briefly survey techniques from analytic visualzation and present the used approaches in Section 2. In the following Section 3 our theoretical model is introduced, alongside some examples. We then demonstrate the application of our technique on a case study to study the correlation of Gnutella with the AS network, as well as to compare Gnutella with a random network in Section 4. We first explain how we sample the P2P network, followed by a comparison of the P2P network with random networks. After pointing out several possible directions for sensitivity analysis of the artificial generation of communications, we conclude in Section 5.

2 Analytic Visualization

Sets of data arising in a diversity of research areas exhibit a growing degree of complexity and depth of properties. In addition to their automatic procession, one of the central emerging challenges is the combination of extracting relevant information from the data and at the same time, representing it in a way well-perceivable by a human. Traditionally these two aspects are handled separately: data analysis and its more spe-

cific subject of network analysis concerns itself with the identification and the computation of relevant pieces of information; while visualization focuses on perceivable representations of networks. However, a fusion of the corresponding techniques supports the intuitive perception of known facts and enables the discovery of novel and yet unknown characteristics. Many network analysis techniques highly benefit from, or even depend on such information about structural properties of a network, in order to properly guide or find starting points for an analysis.

Along the more general issues in the field of information visualization, visualizations of large networks naturally (and specifically) suffer a trade-off between the level of detail and the amount of visible information. In other words, a detailed representation of a graph often antagonizes the immediate perceptibility of abstract analytic information. A key task of the analytic visualization of complex networks is to tackle the task of detailing a visualization while supporting high readability.

2.1 Decompostition Techniques

A common approach many visualization techniques rely on, is to decompose the input network in order to extract the general structure which then serves as a blueprint for the drawing technique. Generally this leads to the task of partitioning or clustering the network, which is a field of ample diversity. For an overview of clustering techniques we refer the reader to [8].

A concept related to density-based clustering is *cores* [5, 24], which yield a hierarchical decomposition of the network and can be computed in linear runtime with a simple implementation, in contrast to many other clustering techniques. Briefly, the *k*-core of an undirected graph is defined as the unique subgraph obtained by recursively removing all nodes of degree less than k. A node has coreness ℓ, if it belongs to the ℓ-core but not to the (ℓ+1)-core. The ℓ-shell is the collection of all nodes having coreness ℓ. The core of a graph is the non-empty *k*-core such that the $(k+1)$-core is empty. Generally the core decomposition of a graph results in disconnected sub-graphs, but in the case of the Internet Autonomous System (AS) network we observe that all *k*-cores stay connected, which is a good feature regarding connectivity. Cores have been frequently used for network analysis, e. g., [10, 11], due to the fact that cores can be computed efficiently in linear time and space.

2.2 Approaches to Analytic Visualization

In the past, several layout techniques have been developed driven by the ambitious goal to properly visualize complex networks such as the AS network. Two important approaches are the landscape metaphor [7] and network fingerprinting [4], examples of which are shown in Fig. 1 and Fig. 2, respectively. Introduced by Baur et al., the former modifies a conventional layout technique by a framework of underlying constraints that are based on analytic properties. The global shape of the network is induced by the position of structurally important elements, which automatically conceal inferior parts. Thus, it reflects the 'landscape' of importance, either in two or three dimensions. The latter approach, LaNet-vi [4] uses analytic properties to define a suitable global shape, which in this case consists of concentric rings of varying thickness, one for each level of the *core*-decomposition (see Section 2.3). Then, the elements of the network

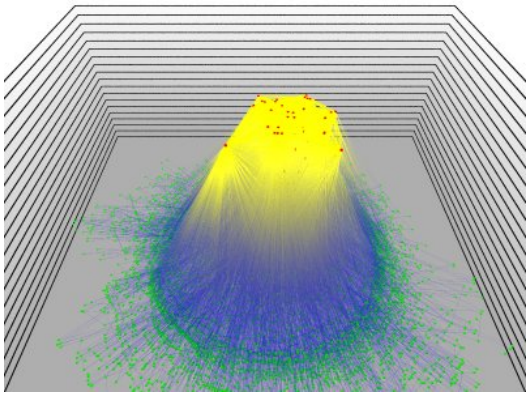


Fig. 1. A 2.5-dimensional layout of the AS network, utilizing the landscape metaphor [7].

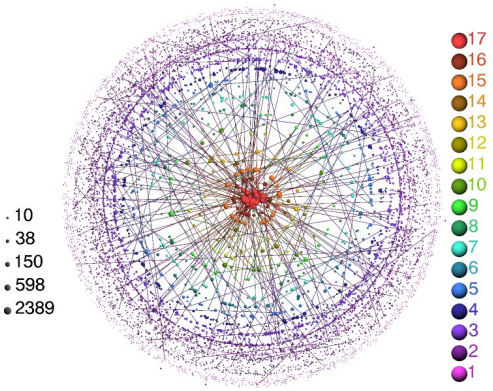


Fig. 2. A fingerprint of the AS network made with LaNet-vi [4].

are placed within these bounds while the overall readability is achieved by showing only a small sample of the edge set.

The nature of the above layout techniques is popularly referred to as a *network fingerprint*. Such pseudo-abstract visualizations offer great informative potential by setting analytic characteristics of a network into the context of its structure, revealing numerous traits at a glance. A fingerprint drawing technique that focuses on the connectivity properties of a network decomposition has been presented in [13]. This approach, coined *LunarVis* lays out each set of a decomposition individually inside the segments of an annulus. An example using departments as the decomposition of a large set of coworkers is given in Figure 3. The rough layout of LunarVis is defined by analytic properties of the decomposition, allowing the graph structure to determine the details. By virtue of a sophisticated application of force-directed node placement, individual nodes inside annular segments reflect global and local characteristics of adjacency while the inside of the annulus offers space for the exhibition of the edge distribution.

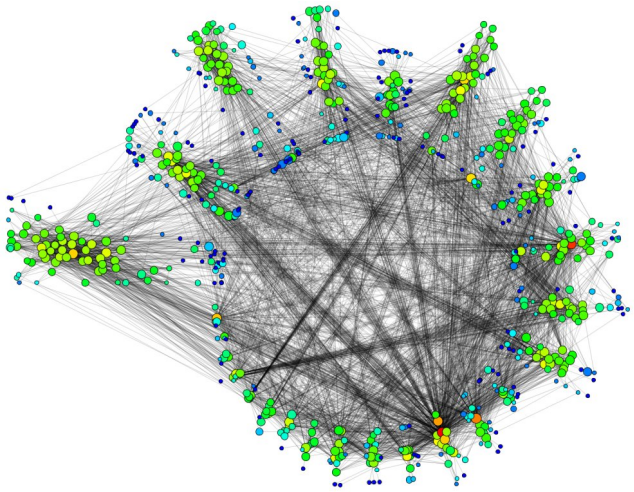


Fig. 3. Visualization of a network of email contacts. Nodes represent computer scientists at Universität Karlsruhe and edges are email contacts. The network is decomposed by departments, color indicates the degree of the nodes, (red = high), while their size indicate their betweenness centrality.

Figure 4 is a visualization of the NLANR web caching hierarchy, created with the tool Plankton [1], which displays all nodes and edges of the NSF-sponsored web caching network. This is an example of a highly application-specific approach of analytic visualization. Although it has the look and feel of classic force-directed methods (for an overview see [6]), it exploits the strongly hierarchical nature of this particular network, and its relatively small size to directly determine a node's position. The low asymptotic complexity of the algorithm allows for an interactive emphasis

of geographical or topological properties, and for the visualization of temporal evolutions. Figure 5 displays 50 stock prices from the Frankfurt stock index over a period

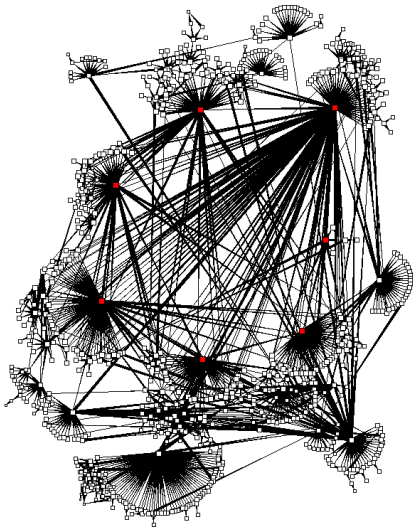


Fig. 4. Visualization of the growth and topology of the NLANR caching hierarchy with Plankton [1]

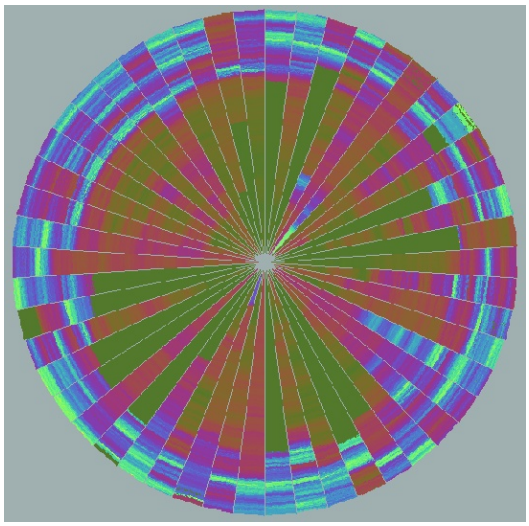


Fig. 5. Circle Segments [14] are used for visualizing multidimensional data sets. Here, about 265000 data values are drawn.

of 10 years. Thus, no actual graph is depicted, however, the drawing techniques from the field of information visualization are related to the approaches in network analysis since they share a set of crucial goals such as the readability of analytic properties of the input data. Naturally, these two fields influence each other (for an introduction into information visualization, see [16] or [28]). The so called *pixel-per-value* technique in Figure 5 fills each segment with one dimension of the data (i.e. one stock item), starting from the inside and coloring pixels according to the stock value.

2.3 Employed Techniques

The techniques presented in the past section have been applied in numerous tasks, serving as an aid in network analyses. In this section, we describe the two visualization techniques in more detail, that helped us immensely in the identification of key features during our analyses. The two techniques, LunarVis [13] and the landscape metaphor [7], both highlight a given hierarchical decomposition of the network while displaying all nodes and edges. We use the concept of cores for the required hierarchical decomposition of the network.

The first technique employing the concept of cores was proposed by Baur et. al. in [7]. More precisely, their algorithm lays out the graph incrementally starting from the innermost shell, iteratively adding the lower shells. Their implementation uses core decomposition and a combination of spectral and force-directed layout techniques. A successful application of this visualization technique compares actual AS graphs with generated AS graphs. The obtained layouts clearly reveal structural differences between the networks. In the following we shall mosly use top-down views of these 2.5-dimensional drawings.

Roughly speaking, the algorithm of LunarVis splits up into three distinct phases, the first of which sets out the abstract layout attributes of the annular layout, such as the

number of segments, their dimension and their placement. In our analysis we will again use core decomposition. Based on these abstract layout attributes, a heuristic computation of suitable parameters follows, which will then be employed in the third and final step. This last, and by far the most intricate and computationally demanding step can be regarded as an iterative, segment-wise application of spring forces. These forces determine the final placement of each single node based on neighborhood attraction and repulsion both inside and between segments.

In the end, the annulus is scaled to the desired angular range and radial spreading and finally edges are drwans as straight lines with a high degree of transparency. Optionally, the size of a node and its color may serve as additional dimensions of information, yet ample use of these potentially overburdens a visualization.

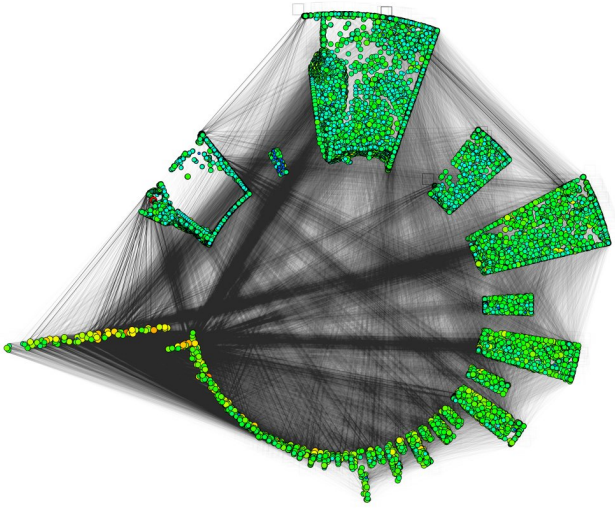


Fig. 6. An example visualization of the core decomposition (segments) of the AS network using LunarVis. Each nodes represent an AS with size and color reflecting the size of its IP-space. Angular and radial extent of a segment reflect the number of nodes and intra-shell edges respectively.

Combined with well-perceivable attributes, such as the size and the color of a node, the layouts made with LunarVis offer remarkable readability of the decompositional connectivity and are capable of revealing subtle structural characteristics. The key to this is the fact that the individual position of a node indicates both its affiliation inside its own segment as well as its tendency of neighborhood to other segments. Figure 6 shows an example of a LunarVis drawing of the AS network. Mentioning only one of the observable insights, note the extremely large AS (upper left red node) in the minimum shell.

3 Modelling Underlays and Overlays

In this section, we introduce our model and methodology for analyzing the relation between under- and overlays as well as a first discussion about different modelling aspects.

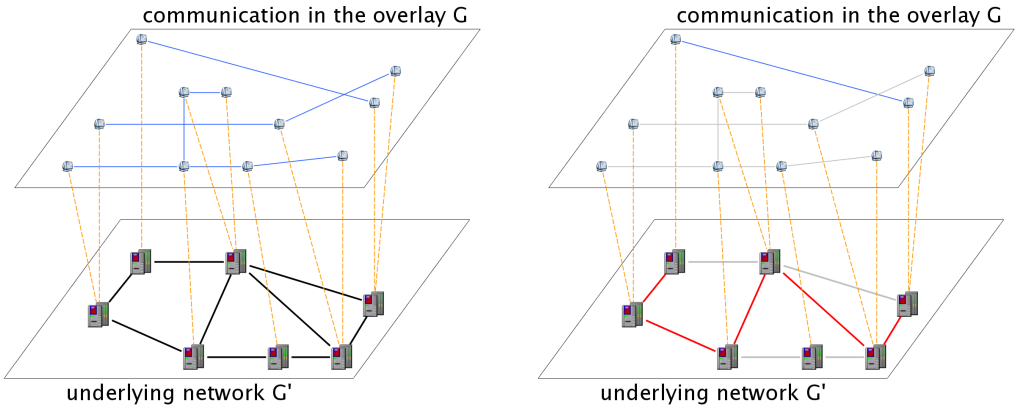
Basically, an overlay consists of network structure that is embedded onto another one. More precisely, each node of the overlay is hosted by a node in the underlay and every edge of the overlay induces at least one path between the hosting nodes (in the underlay) of its end-nodes. The formal definition is given in Definition 1.

Definition 1. *An* overlay *is given by a four-tuple* $\mathcal{O} := (G, G', \phi, \pi)$*, where*

- $G = (V, E, \omega)$ *and* $G' = (V', E', \omega')$ *are two weighted graphs with* $\omega \colon E \to \mathbb{R}$ *and* $\omega' \colon E' \to \mathbb{R}$*,*
- $\phi \colon V \to V'$ *is a mapping of the nodes of* G *into the nodeset of* G'*, and*

– $\pi\colon E \to \{p \mid p \text{ is a (un-/directed) path in } G'\}$ *is a mapping of edges in* G *to paths in* G' *such that* $\{\text{source}(\pi(\{u,v\})), \text{target}(\pi(\{u,v\}))\} = \{\phi(u), \phi(v)\}$.

The interpretation of Definition 1 is that G models the overlay network itself, the graph G' corresponds to the hosting underlay, and the two mappings establish the connection between the two graphs. An example is given in Figure 7. Since direct



(a) Both networks G and G' with the mapping ϕ.

(b) Highlighting one edge e in G and the corresponding path $\pi(e)$ in G'.

Fig. 7. Example of an overlay $\mathcal{O} := (G, G', \phi, \pi)$. The mapping ϕ is represented by dash lines between nodes in G and G'.

communications in the overlay, which corresponds to the edges of G, are realized by routing information along certain paths in G', not all parts of the underlay graph are equally important. In order to focus on the relevant parts, we associate an *induced underlay* with an overlay. The corresponding definition below.

Definition 2. *Given an overlay* $\mathcal{O} := (G = (V, E, \omega), G' = (V', E', \omega'), \phi, \pi)$. *The* induced underlay $\widetilde{\mathcal{O}} := H := (V'', E'', \omega'')$ *is a weighted graph, where*

– $V'' := \{v \in V' \mid \exists\, e \in E\colon \pi(e) \text{ contains } v\}$,
– $E'' := \{e' \in E' \mid \exists\, e \in E\colon \pi(e) \text{ contains } e\}$, *and*
– $\omega''(e') := \sum_{e \in E} \omega(e) \cdot [e' \text{ contained in } \pi(e)]$.

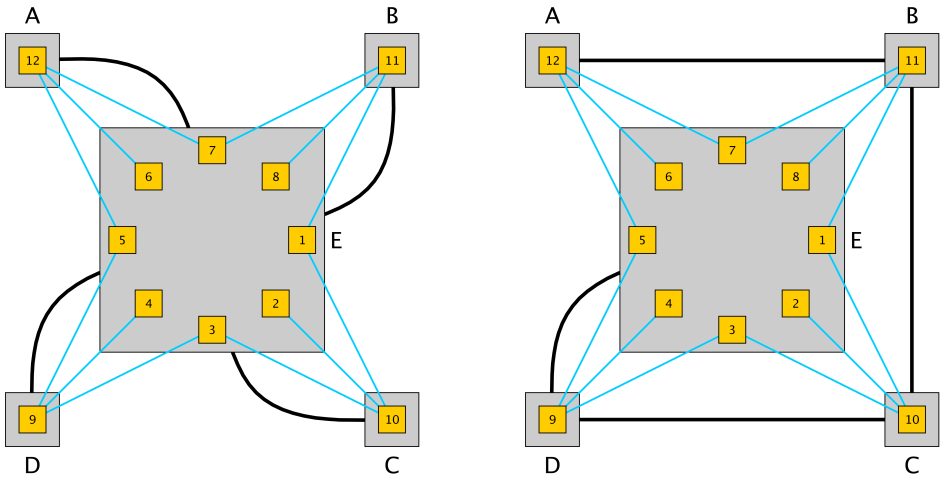
The weight function ω'' *is also called* appearance weight.

The definition of ω'' is given in the Iverson Notation [17]. The term inside the squared parentheses is a logical statement and depending on its value, the term evaluates to 1 if its value is true, and to 0 otherwise. In other words, the induced underlay corresponds to the subgraph of the underlay graph that is required to establish the communication in the overlay graph. Note that the defined weight can be interpreted as the load caused by the communication and thus is independent of a weighting in the underlay network.

3.1 Analysis of Overlays

In the analysis of overlays, we focus on two important aspects: the identification of key features with respect to the underlay and the comparison of different overlays.

The first part, the identification of key features, consists of standard tasks of network analysis, e. g., determining important and relevant nodes or edges, clustering nodes with similar patterns, and detecting unusual constellations. As existing techniques can be applied to the overlay network and the induced underlay, these standard tasks are reasonably well understood in the case of the analysis of a single network. However, these techniques do not incorporate the relationship between the two networks. An example showing such dependencies is given in Figure 8 with the corresponding information about the degrees in Table 1. For illustration, we use the degree,



(a) star topology (b) path topology

Fig. 8. Examples of two overlays where only the topology in the underlay network G' changes. Nodes in the overlay network are numbered with integers and edges are drawn blue, while nodes in the underlay network are labeled with characters and edges are drawn black. In both cases the routing π is done via shortest-path scheme.

property	A	B	C	D	E
number of hosting nodes	1	1	1	1	8
number of edge in the overlay network having an end-node in the node	3	3	3	3	12
UN weighted degree (star top.)	1	1	1	1	4
UN weighted degree (path top.)	1	2	2	2	1
IU weighted degree (star top.)	3	3	3	3	12
IU weighted degree (path top.)	3	9	15	21	12

Table 1. Table with degree information of the examples given in Figure 8. The weighted degree correspondes to the weighted degree in the underlay network (UN) and the induced underlay (IU), respectively.

which is a popular feature, however, these observations carry over to other characteristics as well. First note that the number of hosting nodes and the number of communications a node in the underlay participates in gives a first impression about its role in the network. Both pieces of information can be read off the overlay graph G. However, they are completely independent from the routing structure in the underlay. As the example illustrates, the degree of a node (in the induced underlay) heavily depends

on the routing structure. In the case of the star topology, both the weighted degree in the underlying network and in the induced underlay are fairly similar, here they are even proportional and clearly identify the center node of the star to be central for the network. The situation drastically changes when using a path topology. Although all communications start/terminate at node E, it is not very central. The nodes C and D take on very active roles, due to the fact that most/all communication has to be routed through them. In many cases, the information provided by the induced underlay sufficiently codes the relation between the overlay and underlay networks, while still enabling us to use standard notation of network analysis. On the other hand, there are some scenarios where the provided view is too coarse. For example, it could make a difference, whether a heavy edge is caused by a single heavy communication or by a multitude of small communications or, conversely, whether all communication of a node in the induced underlay have only one target in the overlay or are distributed over many targets.

One motivation for identifying key features is to build a proper model that can be used for extensive simulations. For example, simulations are used to predict scaling behavior or to experimentally validate heuristics, enhancements, or novel techniques. As such, it is a major issue to structurally compare different overlays with each other. On the one hand, our model already reflects all dependencies between the underlay and the overlay network and, thus, it does not require the underlay network, embedding, or routing to be fixed for different instances. On the other hand, due to this elaboration of our model, a simple matching of nodes or edges will not suffice. Our idea is to match key features. For example, one can try to match the appearance weight of an edge with structural properties of its end-nodes. If both overlays have a sufficient number of such matches, it is reasonable to assume that they are created by the same mechanism.

Both parts, the identification of key features and the comparison of overlays, benefit from proper analytic visualizations that emphasize relevant aspects of the corresponding networks. Before presenting two visualization techniques (Section 2), we briefly demonstrate our model and methodology with some experimentally generated examples.

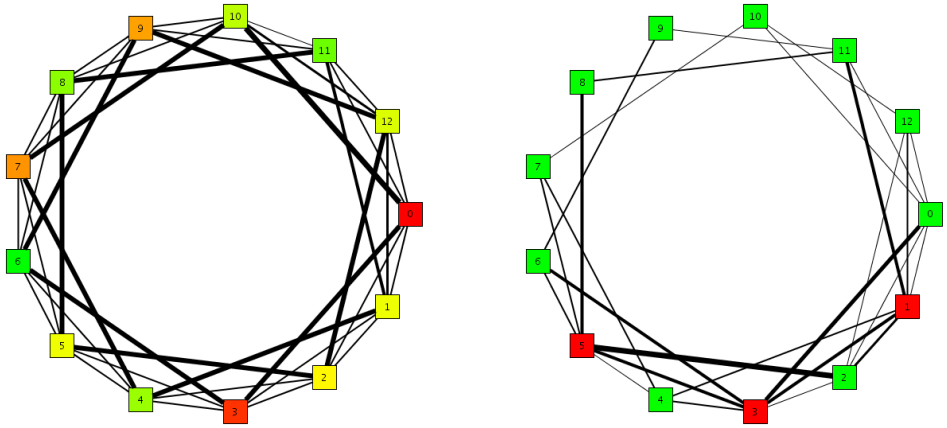
3.2 Examples

In the following, we demonstrate our model and methodology with simple examples. Before looking at a specific overlay, we give two further intuitions.

First, assume a fixed given underlying network. The overlay communication can thus be interpreted as a sampling process of pairs in the underlay. Depending on the application, different patterns occur. For example, in services such as Internet broadcast, one can expect few highly active nodes, which correspond to the hosts of the service while the majority of nodes participate in only a few communications. Using the induced underlay, we can extract such patterns and reconstruct the sampling parameters. Second, assume the underlying network is unknown and acts as a black box, i. e., no information about routing policy and so on is available. By choosing uniformly at random a sample with sufficiently many communications as the overlay, we can not only discover the underlay, but also partly reverse engineer the routing mechanism of π. In the special case that the overlay network is complete, i. e., every pair

of node is connected, the appearance weight of the induced underlay is proportional to the (edge-)betweenness of the original underlying network.

As an example, we consider an underlying network with 17 nodes and a 3-cycle topology, i. e., nodes are cyclic-ordered and each node is connected to 3 of its immediate predecessors and successors. Traffic is routed using shortest path scheme. For simplicity, we set the nodeset of the overlay network to the nodeset of the underlay and thus ϕ to be the identity function. We define two overlays: the first one $\mathscr{O}_1$ (*uniform sampling*) uses uniformly at random selected pairs of nodes for communication, while in the second overlay $\mathscr{O}_2$ (*star-like sampling*) the communication takes place between three predefined nodes and all other nodes chosen uniformly at random. The resulting induced underlays are displayed in Figure 9. As can be clearly seen, the short-



(a) uniform sampling (b) star-like sampling

Fig. 9. Example of induced underlays for different overlay networks in the same underlying network. In the left figure, the communication is uniformly at random distributed over the network and the color codes the (relative) amount of participation. In the right figure, all communications use at least on red node and select the other uniformly at random. In both cases, the thickness of an edge corresponds to the appearance weight.

cuts, i. e., edges that connect two nodes that have a distance of order three, have the largest appearance weight and all other edges have relatively small weights for the uniform sampling. This is not surprising as the appearance weight corresponds to the betweenness of edges. The situation drastically changes, when modifying the sampling mechanism. As in the case of the induced underlay of $\mathscr{O}_2$, the edges relatively close to the initial set have large weights and edges far away have small weights or do not appear at all. For example, the non-existence of the edges $\{9,10\}$ is due to the fact that no shortest path between a red node and any other node uses that edge. On the other hand, the edge $\{6,7\}$ is contained in a shortest path, namely between 3 and 7. However, its absence reveals certain aspects of the underlay routing, i. e., the routing between 3 and 7 will either use the path $(3,4,7)$ or $(3,5,7)$, but never the path $(3,6,7)$.

The above examples showcase the ability of this model to scrutinize an overlay application in the context of the required underlay functionality. This is particularly powerful, if a lot of background knowledge about the underlay is at hand. Analytic visualizations lend themselves well to this kind of analysis, since they are capable of visually correlating diverse analytic properties.

4 Case Study: Overlay Graphs of P2P systems

In this section, we exemplify our analysis technique with a case study of a P2P overlay. For our analysis we choose Gnutella [12], an unstructured file-sharing system which relies on flooding connectivity pings and search queries to locate content. Each message carries a TTL (time to live) and message ID tag. To improve scalability, nodes are classified in a two-level hierarchy, with high-performance ultrapeer nodes maintaining the overlay structure by connecting with each other and forwarding only the relevant messages to a small number of shielded leaf nodes. Responses to pings and queries are cached, and frequent pinging or repeated searching can lead to disconnection from network. More details about Gnutella can be found at [12].

4.1 Sampling and Modelling the P2P Network

In order to analyze the overlay structure, we first need to identify a representative set of connections, called edges, between nodes in the P2P network. To reduce the bias in our sample, we identify edges where neither of the two end-nodes is controlled by us. We refer to such nodes as remote neighbor servents.

Due to message caching and massive churn in P2P networks (we measured the median incoming/outgoing connection duration to be $0.75/0.98$ seconds), a simple crawling approach using pings, e.g., as employed in [22], is not sufficient. However, pings identify nodes that should have been remote neighbor servents at some point.

We thus deploy a combination of active and passive techniques to explore the Gnutella network [2]. Our passive approach consists of an ultrapeer that participates in the network and is attractive to connect to. It shares 100 randomly generated music files (totalling 300 MB in size) and maintains 60 simultaneous connections to other servents. The passive approach gives us a list of active servents. The active approach consists of a multiple-client crawler that uses ping with TTL 2 to obtain a list of candidate servents. Since queries are difficult to cache, we use queries with TTL 2 to obtain a set of remote neighbor servents. These servents are then contacted actively to further advance the network exploration. This approach allows us to discover P2P edges that existed at a very recent point of time. When interacting with other servents, our crawler pretends to be a long-running ultrapeer, answering incoming messages, sharing content, and behaving non-intrusively. This pragmatic behaviour avoids bans. The client uses query messages with broad search strings, e.g., mp3, avi, rar to obtain maximum results. We then combine active and passive approaches by integrating the crawler into the passive ultrapeer.

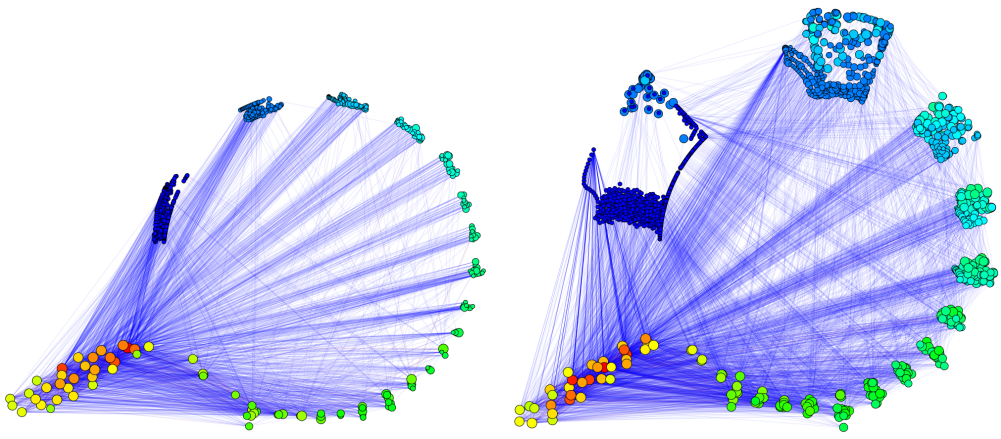
Using this setup, we sample the Gnutella network for one week starting on April 14, 2005. The ultrapeer logs $352,396$ sessions and the crawler discovers $234,984$ remote neighbor servents, a figure significantly higher than most reported results during this period. For each edge of the Gnutella network we map the IP addresses of the Gnutella peers to ASes using the BGP table dumps offered by Routeviews [21] during the week of April 14, 2005. This results in 2964 unique AS edges involving 754 ASes, after duplicate elimination and ignoring P2P edges inside an AS. For the random graph we pick end-points at the IP level by randomly choosing two valid IP addresses from the whole IP space. These edges are then mapped to ASes in the same manner as for the Gnutella edges. This results in 4975 unique edges involving 2095 ASes for

the random network at the AS graph level. The different sizes of the graphs are a result of the generation process: we generate the same number of IP pairs for random network as observed in Gnutella, and apply the same mapping technique to both data sets, which abstracts the graph of IPs and direct communcation edges to a graph with ASes as nodes and the likely underlay communcation path as edges. This way, the characteristics of Gnutella are better reflected than by directly generating a random AS network of the same size as Gnutella network.

For our analysis, we apply the model and methodology from Section 3 as follows. The overlay $\mathscr{O} = (G, G', \phi, \pi)$ as given in Definition 1 uses the direct communication in Gnutella as graph G, the graph G' corresponds to the hosting Internet, in our case at the AS level. The mapping ϕ corresponds to the IP to AS mapping, while π denotes routing in the AS network. Apart from the already introduced induced underlay, we also investigate the network of direct overlay communication, yet abstracted to the level of ASes in order to be comparable to the induced underlay. Note that in a simplified model, where each communication causes uniform costs, the appearance weight in the induced underlay (ω'') corresponds to the total load caused by the overlay routing in the underlay network. As exact traffic measurements on each underlay link are non-trivial, this can be interpreted as an estimate of the actual load on underlay links due to the overlay traffic.

4.2 Overlay-Underlay Correlation in a P2P system

Figure 10 shows visualizations of the direct overlay communication of both the Gnutella network and a random network. Employing the LunarVis [13] technique described in Section 2, these drawings focus on the decompositional properties of the core hierarchy. Numerous observations can be made by comparing the two visualiza-



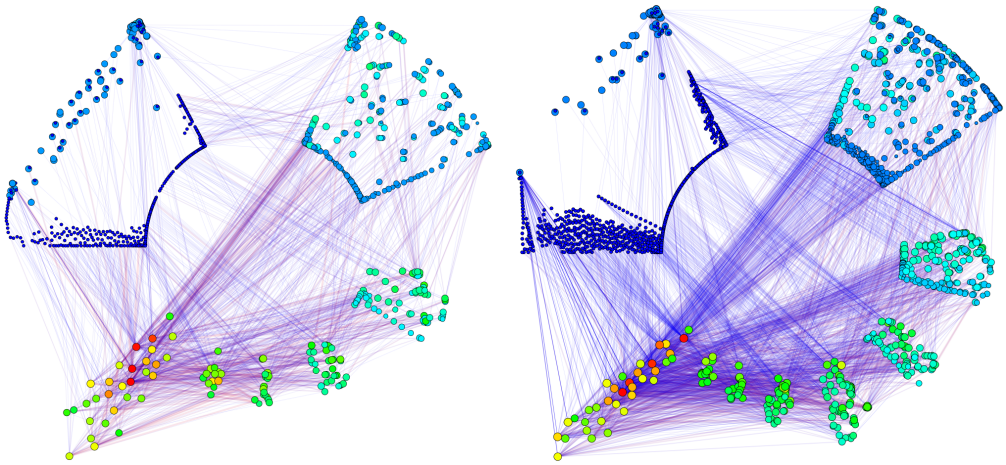
(a) P2P network (b) Random network

Fig. 10. Visualization of the core decomposition of the overlay communication networks. Core-shells are drawn into annular segments, with the 1-shell at the upper left. Angular and radial extent of a segment reflect the number of nodes and intra-shell edges respectively. Inside each shell nodes are drawn towards their adjacencies. Colours represent the degree of a node while the size represents their betweenness centrality. Edges are drawn with 10% opacity and range from blue (small weight) to red (large weight).

tions. Note, first, the striking lack of intra-shell edges for all but the maximum shell in

the Gnutella network (small radial extent). This is also true for edges between shells, as almost all edges are incident to the maximum shell. This means that almost always at least one communication partner is in the maximum shell, a strongly hierarchical pattern that the random network does not exhibit to this degree. Note furthermore that in Gnutella, betweenness centrality (size of a node) correlates well with coreness, a consequence of the strong and deep core hierarchy, whereas in the random network the two- and even the one-shell already contain nodes with high centrality, indicating that many peerings heavily rely on low-shell ASes. The depth of the Gnutella hierarchy (26 levels) strongly suggests a strongly connected network kernel of ultrapeers, which are of prime importance to the connectivity of the whole P2P network. However, note that the distribution of degrees (node colors) does not exhibit any unusual traits and that no heavy edges are incident to low-shell ASes, in either network.

Figure 11 visualizes the induced underlay communication of both the Gnutella network and a random network, employing the same technique and parameters as in Figure 10. The drawings immediately indicate the much smaller number of ASes and



(a) P2P network (b) Random network

Fig. 11. Visualization of the core decomposition of the induced underly communication network. These drawings use the same parameters as Figure 10

overlay nodes in the Gnutella network. As a consequence, more heavy edges (red) exist and the variance in the appearance weigth (edge color) is more pronounced. This is because of the fact that not all the ASes host P2P users (this is in accordance with our measurements in Section 4.1), as is the case for the random network. Again, the distributions of degrees do not differ significantly.

For a closer comparison, Figure 12 shows a top-down view of the visualizations of communication edges in Gnutella and random network. The visualization technique places nodes with dense neighborhoods (tier-1 and tier-2 ASes) towards the center, and nodes with lesser degrees (tier-3 customer ASes) towards the periphery. We can observe that while both networks have many nodes with large degrees in the center, the random network possesses several nodes with large degree in the periphery. Gnutella, on the other hand, has almost no nodes with large degree in the periphery in both models. Moreover, this pattern is more pronounced for Gnutella in the direct overlay

(a) Gnutella (b) random network

Fig. 12. Comparison of occuring communication in the P2P network and the Random network, using visualization, see Section 2.

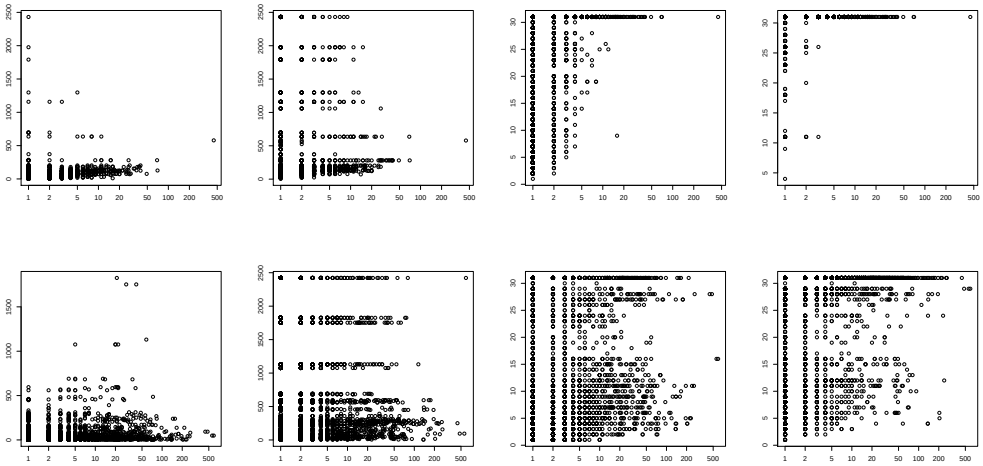


Fig. 13. Comparing appearance weight with minimum and maximum degree and coreness of the corresponding end-nodes in Gnutella and the random network. Each data point represents an edge, the x-axis denotes the appearance weight and the y-axis reflects the degrees (coreness) of the end-nodes. All axes use logarithmic scale.

communication model, while the random network is largely similar in both models. In other words, it appears that Gnutella peering connections tend to lie in ASes in the core of the Internet where there may be high-bandwidth links available.

To further corroborate our observations, we investigate structural dependencies between the induced underlay communication model and the actual underlay network, by comparing the appearance weight with node-structural properties of the corresponding end-nodes in the original underlay. We focus on the properties degree and coreness, as both have been successfully applied for the extraction of customer-provider relationship as well as visualization [26, 10], due to the ability of these properties to reflect the importance of ASes. We systematically compare the weight of an edge with the

minimum and maximum degree and coreness of its end-nodes. Figure 13 shows the corresponding plots.

From the plots of minimum and maximum degree, it is apparent that the appearance weight of an edge and its end-nodes' degrees are not correlated in both the Gnutella and the random network, as no pattern is observable. Also, the distributions are similar as the majority of edges are located in the periphery of the network where the maximum degree of the end-nodes is small. We thus hypothesize that the relation of load in the P2P network and node degree in the underlying network is the same in both the Gnutella and the random network. In other words, the Gnutella network does not appear to be significantly affected by the node degree of underlay nodes.

However, considering the coreness reveals interesting observations. From the graphs of minimum and maximum coreness in Figure 13, we can observe that although there is no correlation in either of the two networks, their distributions are different. In the random network the distributions are very uniform, which is a reflection of its random nature. But in the case of Gnutella almost no heavy edge is incident to a node with small coreness, as can be seen in the minimum-coreness diagram. Positively speaking, most edges with large appearance weights are incident to nodes with large minimum coreness. Interpreting coreness as importance of an AS, these Gnutella edges are located in the backbone of the Internet, an important observation. The same diagram for the random network does not yield a similar significant distribution, thus denying a comparable interpretation. For instance, in the random network, there exist edges located in the periphery that are heavily loaded. As an aside, backbone edges need not necessarily be heavily loaded in either network.

All these observations and analyses show that the Gnutella network differs from random networks and there appears to be some correlation of the Gnutella topology with the Internet underlay.

4.3 Engineering Approaches to Generate Appropriate Communication Models

The analyses conducted in Section 4.2 suggest that a modified generation process is necessary for a more appropriate simulation of Gnutella communication. Speaking in terms of engineering, this closes the cycle of development and leads to a stage of redesigning.

There are different reasons for the observed behaviour, some of which are outside our scope of modelling. For example, the overall communication might be random, but respecting a certain popularity of user content. Thus, naturally, users having much popular content participate in more communications than others. By introducing a bias in the random generation, e.g. assigning weights to nodes, we can incorporate such preferences. A disadvantage of this approach is that it is even harder to obtain realistic estimates for such bias than collecting structural information about the P2P network itself.

As a variation, we propose bias based on graph-structural properties of the overlay and underlay networks, which corresponds to aspects of the infrastructure (instead of content). Such a structural property could be coreness, which reflects the importance of a node in the AS network. Thus, preferring nodes in the overlay that are located in ASes with low coreness models the selection of end-users/customers. Such an analysis has

been conducted in [9]. Furthermore variations of this kind can constitute new protocols for which the performance can be evaluated using our model.

5 Conclusion

In this paper, we present a novel model and technique to analyze the overlay in the context of the underlying network. The major focus of our analysis is the identification of key features as well as the structural comparison between different overlays. More precisely, we transform the overlay to a corresponding subgraph in the underlying network that is crucial for the functionality required by the overlay.

The driving force behind this work is the engineering of overlays which is demonstrated using a case study of the real-world Gnutella network. On the one hand, our analysis reveals differences between the measured Gnutella and experimental mimics that are founded on the same principles and prerequisites. On the other hand, by repeatedly modifying and adjusting the corresponding generation process, based on the insights obtained through detailed analysis and visualization, we are able to deepen our understanding of the real-world instance. In addition, we identify certain artefacts that incite further research. More precisely, our extensive case study incorporates existing visualization techniques for the underlying Internet and establishes that while overlay networks like Gnutella use an arbitrary neighborhood selection process, their topology differs from randomly generated networks.

Our methodology of analyzing the overlays and underlays supported by analytic visualizations offers a powerful and flexible tool in the general engineering process of overlays, which will continue to dominate many spheres of life for times to come!

References

1. Plankton: Visualizing NLANR's web cache hierarchy. http://www.caida.org.
2. Vinay Aggarwal, Stefan Bender, Anja Feldmann, and Arne Wichmann. Methodology for Estimating Network Distances of Gnutella Neighbors. In *GI Jahrestagung*, pages 219–223, 2004.
3. Vinay Aggarwal, Anja Feldmann, and Christian Scheideler. Can ISPs and P2P Systems Cooperate for Improved Performance? In *ACM SIGCOMM Computer Communication Review, 37(3)*, 2007.
4. J. Ignacio Alvarez-Hamelin, Luca Dall'Asta, Alain Barrat, and Alessandro Vespignani. Large scale networks fingerprinting and visualization using the k-core decomposition. In *NIPS*, 2005.
5. Vladimir Batagelj and Matjaž Zaveršnik. Generalized Cores. Preprint 799, IMFM Ljublana, Ljubljana, 2002.
6. G. Battista, P. Eades, R. Tamassia, and I. Tollis. *Graph Drawing - Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.
7. Michael Baur, Ulrik Brandes, Marco Gaertler, and Dorothea Wagner. Drawing the AS Graph in 2.5 Dimensions. In *Proceedings of the 12th International Symposium on Graph Drawing (GD'04)*, volume 3383 of *Lecture Notes in Computer Science*, pages 43–48. Springer, January 2005.
8. Marco Gaertler. Clustering. In Ulrik Brandes and Thomas Erlebach, editors, *Network Analysis: Methodological Foundations*, volume 3418 of *Lecture Notes in Computer Science*, pages 178–215. Springer, February 2005.
9. Marco Gaertler, Robert Görke, Dorothea Wagner, Anja Feldmann, and Vinay Aggarwal. Modelling Overlay-Underlay Correlations Using Visualization. *Telektronikk*, 2008. to appear.
10. Marco Gaertler and Maurizio Patrignani. Dynamic Analysis of the Autonomous System Graph. In *IPS 2004 – Inter-Domain Performance and Simulation*, pages 13–24, March 2004.
11. Christos Gkantsidis, Milena Mihail, and Ellen W. Zegura. Spectral Analysis of Internet Topologies. In *Proceedings of Infocom'03*, 2003.
12. Gnutella v0.6. `http://en.wikipedia.org/wiki/Gnutella/`.
13. Robert Görke, Marco Gaertler, and Dorothea Wagner. LunarVis - Analytic Visualizations of Large Graphs. In *Proceedings of the 15th International Symposium on Graph Drawing (GD'07)*, Lecture Notes in Computer Science. Springer, 2008. to appear.

14. Bradley Huffaker, Jaeyeon Jung, Evi Nemeth, Duane Wessels, and K. Claffy. Visualization of the growth and topology of the NLANR caching hierarchy. *Comput. Netw. ISDN Syst.*, 30(22-23):2131–2139, 1998.
15. Ipoque. `http://www.ipoque.com`.
16. Daniel A. Keim. Visual exploration of large data sets. *Commun. ACM*, 44(8):38–44, 2001.
17. Donald E. Knuth. Two notes on notation. *American Mathematical Monthly*, 99:403–422, 1990.
18. Y. Liu, H. Zhang, W. Gong, and D. Towsley. On the interaction between overlay routing and traffic engineering. In *IEEE INFOCOM*, 2005.
19. A. Nakao, L. Peterson, and A. Bavier. A Routing Underlay for Overlay Networks. In *SIGCOMM*, 2003.
20. S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Topologically aware overlay construction and server selection. In *IEEE INFOCOM*, 2002.
21. University of Oregon Routeviews Project. `http://www.routeviews.org/`.
22. S. Saroiu, K. Gummadi, and S. Gribble. A measurement study of p2p file sharing systems. In *Multimedia Computing and Networking*, 2002.
23. S. Seetharaman and M. Ammar. On the interaction between dynamic routing in the overlay and native layers. In *IEEE INFOCOM*, 2006.
24. Stephen B. Seidman. Network Structure and Minimum Degree. *Social Networks*, 5:269–287, 1983.
25. R. Steinmetz and K. Wehrle. *P2P Systems and Applications*. Springer Lecture Notes in CS, 2005.
26. Lakshminarayanan Subramanian, Sharad Agarwal, Jennifer Rexford, and Randy H. Katz. Characterizing the Internet Hierarchy from Multiple Vantage Points. In *Proceedings of Infocom'02*, pages 618–627, 2002.
27. Telco 2.0 notes. `http://ikisai.wordpress.com/2006/10/08/telco-20-notes-verbose/`.
28. Colin Ware. *Information visualization: perception for design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.