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, driven 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 on an already existing underlay infrastructure as an overlay.

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 [17], which range from file-sharing systems like Gnutella and BitTorrent, to real-time multimedia streaming, to VoIP phone systems like Skype and GoogleTalk.

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

* 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.
underlay. The major advantage of overlays is that they provide high-level functionality while masking the intrinsic complexity of 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 [17], there has been an unprecedented interest in their analysis [1, 2, 15]. There have also been attempts to investigate the overlay-underlay correlations in P2P systems. Using game theoretic models, [10] studies the interaction between overlay routing and traffic engineering within an AS. An analysis of routing around link failures [15] 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., [1]), 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. [11, 12], have been proposed. More recently, [2] 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 unique visualization-driven approach [4] which relies on the concept of cores, to analyze the overlay in the context of the underlay network. We introduce our theoretical model with examples in Section 2, followed by the introduction of a new technique for analytic visualization in Section 3. 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 a sensitivity analysis of the random network to generate and better understand P2P network models, we conclude in Section 5.

2 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 be embedded into 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 \(O := (G, G', \phi, \pi)\), where

\[- G = (V, E, \omega) \text{ and } G' = (V', E', \omega') \text{ are two weighted graphs with } \omega : E \rightarrow \mathbb{R} \text{ and } \omega' : E' \rightarrow \mathbb{R}, \]
\[- \phi : V \rightarrow V' \text{ is a mapping of the nodes of } G \text{ to the node set of } G', \text{ and} \]
\[- \pi : E \rightarrow \{p | p \text{ is a (un-/directed) path in } G'\} \text{ is a mapping of edges in } G \text{ to paths in } G' \text{ such that } \{\text{source}(\pi\{u, v\}), \text{target}(\pi\{u, v\})\} = \{\phi(u), \phi(v)\}. \]

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 1. As direct communications in the overlay, which corresponds to the edges of \(G\), is realized by routing information along certain paths in the \(G'\), 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 is given in 2.

Definition 2. Given an overlay \(O := (G = (V, E, \omega), G' = (V', E', \omega'), \phi, \pi)\). The induced underlay \(\hat{O} := H := (V'', E'', \omega'')\) is a weighted graph, where
Both networks $G$ and $G'$ with the mapping $\phi$.

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

Fig. 1. Example of an overlay $O := (G, G', \phi, \pi)$. The mapping $\phi$ is represented by dash lines between nodes in $G$ and $G'$.

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

The weight function $\omega''$ is also called appearance weight.

The definition of $\omega''$ is given in the Iverson Notation [9]. The term inside the squared parentheses is a logical statement and depending on its value, the term evaluate 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.

2.1 Analysis

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 2 with the corresponding information about the degrees in Table 1. We use the degree, which is a popular feature, for illustration, however, these observations carry over to other characteristics. 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 informations can be read off the overlying 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
Fig. 2. 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 $\pi$ is done via shortest-path scheme.

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

Table 1. Table with degree information of the examples given in Figure 2. The weighted degree corresponds to the weighted degree in the underlay network (UN) and the induced underlay (IU), respectively.
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 3), we briefly demonstrate our model and methodology with some experimentally generated examples.

2.2 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 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 \( \pi \). 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 \( \phi \) to be the identity function. We define two overlays: the first one \( O_1 \) (uniform sampling) uses uniformly at random selected pairs of
nodes for communication, while in the second overlay $O_2$ (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 3. As can be clearly seen, the short-cuts, i.e.,

![Fig. 3. 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 one red node and select the other uniformly at random. In both cases, the thickness of an edge corresponds to the appearance weight.](image)

edges that connect two nodes that have a distance of order two, 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 $O_2$, 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).

3 Analytic Visualization

In the following section, we describe two visualization techniques that greatly help in the identification key features. Both highlight a given hierarchical decomposition of the network while displaying all nodes and edges. They have been successfully applied to the network of Autonomous Systems (AS), which is an abstraction of the physical Internet, yet are highly flexible and can be easily adjusted to other networks.

We use the concept of cores [3, 16] for the required hierarchical decomposition of the network. 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 $\ell$, if it belongs to the
The nature of the above layout technique 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 [8]. This approach, coined LunarVis lays out each set of a decomposition – which are the shells in our case – individually inside the segments of an annulus. 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. Combined with well-perceivable attributes, such as the size and the color of a node, these layouts offer remarkable readability of the decompositional connectivity and are capable of revealing subtle structural characteristics.

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 [7], 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 [7].
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 [14], 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 [1]. 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 behavior 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 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 [13] 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 communication edges to a graph with ASes as nodes and the likely underlay communication 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 2 as follows. The overlay $\mathcal{O} = (G, G', \phi, \pi)$ 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 the AS level. The mapping $\phi$ corresponds to the IP to AS mapping, while $\pi$ is the 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 ($\omega''$) 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 5 shows visualizations of the direct overlay communication of both the Gnutella network and a random network. Employing the LunarVis [8] technique described in Section 3, these drawings focus on the decompositional properties of the core hierarchy. Numerous observations can be made by comparing the two visualizations. 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 6 visualizes the induced underlay communication of both the Gnutella network and a random network, employing the same technique and parameters as in Figure 5. The drawings immediately indicate the much smaller number of ASes and overlay nodes in the Gnutella network. As a consequence, more heavy edges (red) exist and the variance in the appearance weight (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.
Fig. 6. Visualization of the core decomposition of the induced underly communication network. These drawing use the same parameters as Figure 5

Fig. 7. Comparison of occurring communication in the P2P network and the Random network, using visualization, see Section 3.

For a closer comparison, Figure 7 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
Fig. 8. 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.

in the periphery in both models. Moreover, this pattern is more pronounced for Gnutella in the direct overlay 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 [18,5], 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 8 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 8, 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 analysis show that the Gnutella network differs from random networks and there appears to be some correlation of Gnutella topology with the Internet underlay.

### 4.3 Sensitivity Analysis to Refine the Model

The analyses conducted in Section 4.2 and the conclusions drawn, solely rely on analytic visualizations. Based on these we now aim at a deeper understanding of the properties of the underlay communication the P2P network induces. Modifying the generation process for the random networks in ways suggested by our observation, we are now able to conduct a sensitivity analysis, in order to find parameters for the random network that lead to a more aligned edge-coreness distribution with the observed P2P network.

It is both reasonable to assume that many nodes are in lower shells (customer ASes) and that heavy nodes (ultrapeers) are in higher shells. Therefore we consider two modifications: The low coreness communication restricts the IP-spaces that are available for communications to those hosted by ASes with low coreness. Analogously the high coreness communication uses only IPs located in ASes with high coreness. For reasons of space and simplicity we present only the plots of two of our various experiments. In order to model the routing in the Internet more accurately, we considered the AS network as directed and thus had to adjust the coreness calculation properly. As a rule of thumb, the values roughly double compared to the original scenario described in Section 4.2. Figure 9 shows the plots that correspond to the right four diagrams in Figure 8. Again a data point is plotted for each edge in the induced underlay, with coordinates that correspond to its appearance weight (x-axis) and to its minimum/maximum incident node coreness (y-axis). The corresponding plots of the degree distributions are omitted as they did not differ much.

At a first glance we can observe that the restriction to low coreness communication does not yield a significant difference to the corresponding plot of our initially unrestricted random network (Figure 8 lower right). Although the distributions are shaped in a highly similar manner, they differ in the maximum occurring appearance weight. On the other hand, the high coreness communication exhibits a very different pattern. Its distributions are more similar to those from Gnutella than the random ones. A very interesting observation, is that although communicating IPs are located in ASes with high coreness, some routing path uses low-coreness ASes.

Interpreting these findings, we conclude that the observed part of Gnutella mainly corresponds to a large part of the network spanned by the ultrapeers and only few leaf nodes are included. Typically ultrapeer nodes maintain a connection to a certain (small) number of leaf nodes. On the other hand, the leaf nodes possess only slow Internet connections and connect to the well-performing ultrapeers, who in turn shield them from a large amount of P2P traffic, yet enable them to locate and share content. The well-know effect of rampant free-riding corroborates our interpretation. More precisely, the phenomenon refers to the fact that a large number of nodes remain online for very short durations, share no content, and are only interested in finding content, while a small percentage of nodes participate in the network for very long durations, and provide most of the content sought in the network. Hence, they participate in much more communications as compared to the other P2P nodes.
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 it 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 artifacts 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.

Acknowledgements

We thank you.

References


